Optimizing Multiview Video Plus Depth Prediction Structures for Interactive Multiview Video Streaming

Ana De Abreu, *Member, IEEE*, Pascal Frossard, *Senior Member, IEEE*, and Fernando Pereira, *Fellow, IEEE*

***Abstract*—Several multiview video coding standards have been developed to efficiently compress images from different camera views capturing the same scene by exploiting the spatial, the temporal and the interview correlations. However, the compressed texture and depth data have typically many interview coding dependencies, which may not suit *interactive multiview video streaming* (IMVS) systems, where the user requests only one view at a time. In this context, this paper proposes an algorithm for the effective selection of the interview prediction structures (PSs) and associated texture and depth quantization parameters (QPs) for IMVS under relevant constraints. These PSs and QPs are selected such that the visual distortion is minimized, given some storage and point-to-point transmission rate constraints, and a user interaction behavior model. Simulation results show that the novel algorithm has near-optimal compression efficiency with low computational complexity, so that it offers an effective encoding solution for IMVS applications.**

***Index Terms*—Interactive multiview video streaming (IMVS), multiview video plus depth, popularity model, prediction structure, view synthesis.**

I. Introduction

In interactive multiview video streaming (IMVS) systems, multiple cameras capture the same scene from different viewpoints and then a user can interactively select the viewpoint of his/her preference within a certain navigation range. A common data format used in these systems is *multiview video plus depth* (MVD), where for each texture frame of a captured view there is an associated depth map, which is required for intermediate view rendering purposes. Due to the huge amount of redundant information present in MVD data, several multiview video coding[1] standards can be used to efficiently compress MVD data with prediction structures (PSs) that exploit the interview, spatial, and temporal dependencies, which is usually the solution for this type of content [1]. While maximizing the redundancy reduction, in particular the interview redundancies, is the main target for applications where all the views are stored and transmitted together, this may not be optimal for IMVS applications where the user requests only one view at a time from a large set of available views, where distant views can considerably differ in the scene content. Indeed, the numerous coding dependencies in these PSs may bring significant rate penalties as a request for one view typically implies the transmission of data from many other views that the requested view depends on. As a result, the optimal prediction structure (PS) for IMVS applications should be different from other multiview applications, it needs to offer an appropriate trade-off between transmission and storage cost. However, most previous works in multiview video coding have been focused on exhaustively exploiting the inherent correlation among the views to improve the overall compression efficiency of multiview video systems, without paying particular attention to the transmission aspects, [1], [2], and thus IMVS challenges have so far not been much addressed.

In this work, our main goal is to find the optimal interview PS and associated texture and depth quantization parameters (QPs) to encode a set of views in the context of IMVS. In the considered IMVS system, given a user's view switch request the minimal information is sent for view rendering, meaning the requested view or reference views to render the missing viewpoint. We consider depth-image-based rendering (DIBR) techniques in order to render new views from encoded texture and depth maps.

To better adapt the coding model to the video content along time, we characterize the user interaction behavior with a view popularity model [3]–[5], assuming a random access interactivity model, where users can switch to any viewpoint in the multiview system and not only to neighboring views. This is appropriate for many application scenarios, notably sports events, where the users suddenly and largely change the point of view. Transmission and coding rates, and distortion models are proposed in the context of IMVS. Finally, a greedy algorithm is proposed to find the optimal interview PS and QPs for the texture and depth maps. The optimal PS and QPs should minimize the distortion in a system where the point-to-point transmission bandwidth and the storage capacity are scarce resources. It is important to mention that the proposed algorithm is not specific of any coding standard, provided that we are using a temporal and interview predictive coding solution for both texture and depth maps. Experimental results show that the proposed algorithm is able to identify a near-optimal PS in the sense of minimizing the

Manuscript received April 25, 2014; revised October 13, 2014 and February 13, 2015; accepted February 13, 2015. Date of publication February 25, 2015; date of current version March 18, 2015. This work was supported by Fundação para a Ciência e a Tecnologia, under the Grant SFRH/BD/51443/2011. The guest editor coordinating the review of this manuscript and approving it for publication was Prof. Thrasos Pappas.

A. De Abreu is with the Signal Processing Laboratory (LTS4), École Politechnique Fédérale de Lausanne (EPFL), CH-1015 Lausanne, Switzerland, and also with the Instituto Superior Técnico, Universidade de Lisboa-Instituto de Telecomunicações (IST/UTL-IT), 1049-001 Lisbon, Portugal (e-mail: ana.deabreu@epfl.ch).

P. Frossard is with the Signal Processing Laboratory (LTS4), École Politechnique Fédérale de Lausanne (EPFL), CH-1015 Lausanne, Switzerland (e-mail: pascal.frossard@epfl.ch).

F. Pereira is with the Instituto Superior Técnico, Universidade de Lisboa-Instituto de Telecomunicações (IST/UTL-IT), 1049-001 Lisbon, Portugal (e-mail: fp@lx.it.pt).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JSTSP.2015.2407320

[1]In this work multiview video coding refers to a general codec for multiview video, otherwise when referring to the Multiview Video Coding standard the acronym MVC is used.

distortion while trading off the transmission and storage costs. At the same time, our PS and associated QPs selection algorithm leads to a complexity reduction up to 72% compared to an exhaustive searching approach.

Contrarily to our previous work [4], where only texture frames are considered and only coded views are available for user request, here also virtual views can be requested, increasing the navigation range and smoothness offered to the user. In addition, both the PSs and the corresponding texture and depth maps QPs are optimized in this work, while in [4] we assumed that the QPs (for the texture information) were known in advance by fixing the maximum acceptable distortion.

To achieve its objectives, this paper is organized as follows: Section II overviews the related work. Section III outlines the main characteristics of the IMVS system under consideration. Section IV describes the transmission and coding rate, and distortion models adopted. Then, the optimization problem to find the optimal interview PS and associated QPs given some system constraints is formulated in Section V. In Section VI, a greedy algorithm is proposed to efficiently solve the optimization problem, previously formulated. Section VII presents and analyses the performance results demonstrating the benefits of the proposed solution considering both the MVC and 3D-HEVC coding standards and finally, the conclusions and further work are presented in Section VIII.

II. Related Work

IMVS systems challenges have not been much addressed in the literature, where the focus has mainly been put on the overall video compression efficiency [1], [2]. However, some prediction structure (PS) selection mechanisms have been recently proposed for IMVS systems, with the goal of providing multiview video with flexible viewpoint switching by trading off the transmission rate, storage capacity and/or decoding complexity.

In [6], a group of GOPs (GoGOP) concept is introduced, where interview prediction is restricted to the views from the same GoGOP. This approach offers a low-delay random accessibility, as well as low-transmission bandwidth cost but leads to lower compression efficiency.

To save transmission bandwidth, different interview prediction structures are proposed in [7] to code different versions of a multiview set, in order to satisfy different RD performances. However, this approach brings a high storage cost at the server, as its gain depends on the number of PSs used to encode the multiview sequence.

In [8], a *user dependent multiview video streaming for Multi-users* (UMSM) system has been proposed to reduce the transmission rate due to redundant transmission of overlapping frames, in a system with multiple users. In UMSM, the overlapping frames (potentially requested by two or more users) are encoded together and transmitted by multicast, while the non-overlapping frames are transmitted to each user by unicast. This approach is only useful when several users are watching the same video at the same time instant. However, in this paper, we are interested in a unicast transmission, where a one-to-one communication is established between each user and the server. In addition, if in [8] a random interactivity model is assumed, where a user can switch to any viewpoint, as it is assumed in this work, UMSM must transmit all the views to each user, which is exactly what we would like to generally avoid in this work, where the transmission bandwidth is scarce.

In [11], [12] and [13], the authors have studied the PSs that facilitate a continuous view-switching by trading off the transmission rate and the storage capacity. The authors have considered a coding system with redundant P- and DSC-frames (distributed source coding), which is unfortunately not compliant with common decoders.

A different approach is followed in [14] and [15] where, given a rate constraint, a set of views is selected at the sender side for encoding. Then, the set of views are transmitted to the users from where they may select an encoded view or synthesize an uncoded view. Differently, we consider the case where only minimal information is sent to the user to render the requested view, instead of a large set of views.

In [9], the authors target the transmission bandwidth problem alone in a 3DTV context by predicting the head position of the users. They propose to transmit a small number of views at high quality, such that two views are available for each predicted head position to render the viewing angle in a stereoscopic display. After, to conceal the effect of the prediction errors, low quality versions of all the views are also transmitted. Similarly, in [10] a joint tracking and compression scheme is proposed to accurately predict future head positions. In contrast, we consider in our work an IMVS system where only one view is requested at a time by the users and thus only the required information to render the requested view is transmitted to the user.

In [3], a PS selection mechanism has been proposed for distortion-minimized IMVS while trading off transmission rate and storage cost. However, even when the set of possible PSs proposed is limited, the complexity of this PS selection mechanism exponentially grows with the number of views in the multiview set, which is a major disadvantage. In [4], a constructive algorithm is proposed to speed up the PS search with low compression performance penalty in IMVS systems based on Multiview Video Coding (MVC).

Finally, the authors in [16] consider the trade-off between flexibility, latency, and bandwidth when proposing a system to provide interactive multiview video service in real time. In this system, three prediction structures are proposed in order to offer three different interactive experiences to the users. However, the trade-off presented is found empirically after carrying out a user study, for which no specific information is given and without solving any explicit optimization.

In this work, we build on our previous work [4] and consider now the case where both texture and depth maps are available for each coded view. This advanced data representation permits the synthesis of virtual views at the user. Second, we derive an optimization algorithm to find the optimal or very close to optimal PS and associated QPs for texture and depth maps, in a system where there is a trade-off between the transmission rate and storage capacity, and the distortion of the decoded views. This proposed algorithm only requires a temporal and interview predictive coding solution, it is not specific of a particular coding standard. Finally, we build a highly adaptive framework where the optimized interview PS may vary at GOP level, in

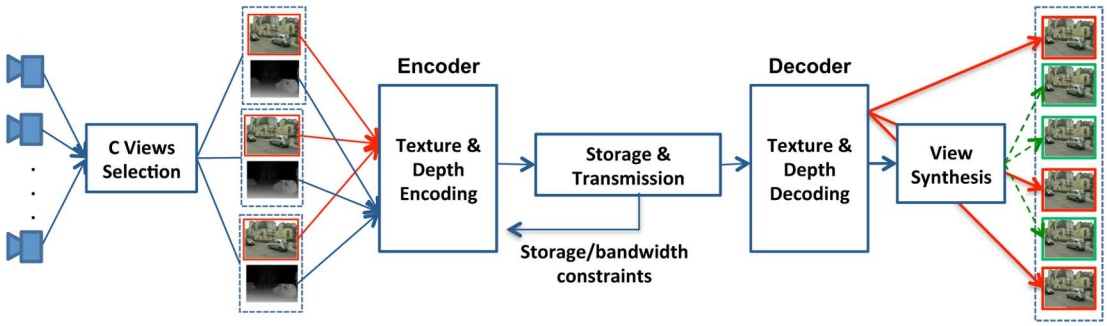

Fig. 1. General IMVS system architecture. Coded and virtual views are represented by images connected by continuous and dashed arrows to the texture decoder and view synthesis blocks, respectively.

order to better adapt the coding model to the video content along time.

III. IMVS Framework

In this work, we consider an IMVS system, where multiview video coding standards are used to compress texture and depth data for a limited set of views. The users may not only be interested in the coded viewpoints but also in intermediate viewpoints derived from a pair of textures and depths views. The most relevant characteristics of the IMVS system model considered in this work are described below.

A. Depth-based Multiview Model

We consider an IMVS system where a set of C views, $\mathcal{C} = \{1, \cdots, C\}$, is encoded at the sender side. For each coded view $c \in \mathcal{C}$, texture and depth maps are available, allowing the generation of intermediate virtual viewpoints at the decoder with an appropriate synthesis algorithm. This set of coded views may be different from the set of captured ones as the rate may be limited and/or the position of the cameras capturing the scene may not always be the optimal one. Between each pair of consecutive coded views, some virtual view positions may be available for user request at a minimum guaranteed quality. The set of all V virtual views is defined as $\mathcal{V} = \{1, \cdots, V\}$. With the help of the depth-image based rendering (DIBR) solution [17], [18], these views are synthesized using the closest right and left coded views, denoted as $\{r_v, l_v\}$ for $r_v, l_v \in \mathcal{C}$ and $v \in \mathcal{V}$ or simply as $\{r, l\}$ for $r, l \in \mathcal{C}$. In DIBR, pixels from the right and left reference views are appropriately projected to an intermediate virtual viewpoint position, using the available depth information. Then, the projected pixels from the reference views are merged together, e.g., using a linear weighting function that considers the distance between reference and virtual views [19]. This approach reduces the occurrence of disocclusions as unknown regions for the first reference view are filled with information from the second reference view. Overall, the global number of views available for user request is defined by $\mathcal{U} = \{1, \cdots, U\}$, where $U = C + V$.

Multiview video coding is applied to both texture and depth components of the set of coded views. In this paper, two coding standards are adopted. In the first coding solution, both the texture and depth components are (independently) coded using the Multiview Video Coding (MVC) standard, a backward compatible extension of the H.264/AVC standard [1]. In the second coding solution, the texture and depth components will be jointly coded exploiting the inter-component dependencies, notably between the depth and the texture, using the 3D-HEVC standard, one of the 3D extensions of the HEVC standard [20].

Based on predefined storage and bandwidth constraints, both texture and depth images are encoded using the same optimized PS at their respective QPs, $Q = (Q_t, Q_d)$, where t and d stand for texture and depth, respectively. The QPs are typically different for the texture and depth data [21], as they have completely different impact on the final texture quality, and thus lead to different RD trade-offs. It is important to remember that, while decoded textures are directly offered to the users, decoded depths are not; they only serve to generate virtual views (thus also influencing their texture quality). All coded data is stored in a server and eventually transmitted when requested. The server provides an IMVS service to multiple users. We assume that when a coded view is requested by the user, only the texture information is transmitted. On the other hand, when a virtual view is requested, both the texture and depth maps of the closest right and left coded views are transmitted by the server, if not already available at the user, so that the user can synthesize the requested virtual viewpoint. This transmission model ensures the backward compatibility with traditional video decoders, by only offering texture information to users unable to synthesize virtual viewpoints. The same general IMVS system model can be considered in the stereo video case, where instead of one, two views are requested by the user, notably considering that different stereo displays may use different baseline distances. Then, if the two requested views are coded views, only their texture information is transmitted to the user. However, if one or both requested views are virtual views, texture and depth maps of the closest right and left coded views of one or both viewpoints need to be transmitted. Fig. 1 illustrates the general IMVS system architecture, where the coded and virtual views are represented by frames connected by continuous and dashed arrows to the decoder and view synthesis blocks, respectively.

B. Interactivity Model

In our system, the user interactivity model works as follows:

1) Random Access: We consider an IMVS system with random access characterized by view-switches from/to any viewpoint in the multiview set, virtual or coded, but occurring only at the anchor frames of the coded views. Anchor frames are frames that do not use temporal prediction for encoding, although they do allow interview prediction from other views in the same time instant [22]. Generally, random access is

guaranteed by coding the anchor frames of the independently encoded views (*key views*) in Intra mode (I-slices for all frame).

2) View Popularity: To model the user interaction, a view popularity factor, p_u^g, is considered to express the probability that a user selects view $u \in \mathcal{U}$ at the switching time instant (i.e., at the anchor frames) of a group of pictures (GOP) g. We assume that the probability p_u^g, $\forall u \in \mathcal{U}$, depends on the popularity of the views or on the scene content itself but not on the view previously requested by the user. This may be the case for sports scenes for example, where a user may be following the moves of his/her favorite player but at a certain time decides to change to the most popular view, which is done independently of his/her current position. We assume a *static view temporal popularity model*, meaning that all the GOPs of a given view have the same probability of being requested by the users, although this may be easily modified if the temporal characteristics of the content are considered.

C. Coding Model

Multiview video plus depth coding considers both the temporal and interview correlations to increase the RD performance, reducing the redundancy among different views at the same time instance and among subsequent frames in time in the same view position. In this work, the same temporal and interview PS is used for coding both the texture and depth maps of the set of coded views $\mathcal{C}$. The temporal and interview coding models to be considered for the optimization of the texture and depth common PS have the following characteristics:

1) Temporal Coding Model: As commonly done in the literature, we assume a fixed temporal PS for each view (texture and depth maps), with hierarchical B-frames/slices [23], where B-frames are hierarchically predicted from other B or anchor frames. Fig. 2 illustrates a typical hierarchical B-frames PS with 4 temporal layers, denoted with a sub-index from 0 to 3. The arrows in the figure indicate the reference frames used for the prediction of the various B frames. To control the quantization steps in the temporal domain, and thus the distortion, a cascading quantization parameters (CQP) [2] strategy is used. In this strategy, the full set of texture and depth QPs, $Q = (Q_t, Q_d)$, for the anchor frames are encoded with a small QP (high quality), since they are used as references for the prediction of frames in higher temporal layers. Then, the QPs of the frames in higher temporal layers are assigned by increasing the previous temporal layer QP with a pre-defined ΔQ, which may also be different for texture and depth. Here, we assume that even if Q_t and Q_d can take different values, their value distribution is the same for all the views in a particular GOP, as they vary at GOP level. Therefore, for a given PS there is only one Q_t and one Q_d that are used for all the texture and depth maps of the views, respectively. This assumption reduces the complexity of the Q search and it is not far from reality as the content of the various views from the same captured scene tends to be very similar and as a consequence the optimal QPs should also be similar among the views.

2) Interview Coding Model: The interview coding models considered here are based on the two most commonly used interview PSs in multiview video coding standards, namely IBP and IP [22]. In these PSs, hierarchical B-frames are used in the temporal domain while the IBP or IP modes are used for the anchor frames, determining the interview coding model of both anchor and non-anchor frames. Although the use of interview coding in the non-anchor frames is optional, here we use it as it has been shown to improve the RD performance in typical sequences [22]. The set of PSs under consideration are illustrated in Fig. 3, where only one key view is usually considered, typically a lateral view, in order to reduce interview redundancy. However, as high compression efficiency is not the only objective in IMVS systems, we allow here more than one key view in the two basic PSs (IBP and IP) to reduce the coding dependencies that penalize the system performance in interactive streaming. This is illustrated in Fig. 4 in the case of two IP PSs, one with only one key view (view 0) and the other with two key views (views 0 and 2). We further show (in gray) the frames that need to be transmitted in order to decode a GOP in view 3. It can be seen that, due to the interview coding dependencies, for the PS with only one key view (maximum compression efficiency) all the frames from the previous views (views 0, 1 and 2) need to be transmitted together with the requested view 3, while for the PS with two key views, only views 2 and 3 need to be transmitted. Finally, for benchmarking purposes, we also consider the simulcasting structure where all the views are key views (I PS).

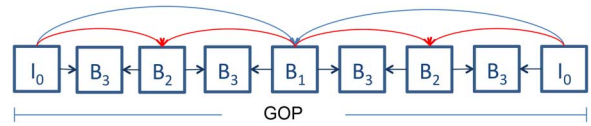

Fig. 2. Hierarchical B-frames with four temporal layers.

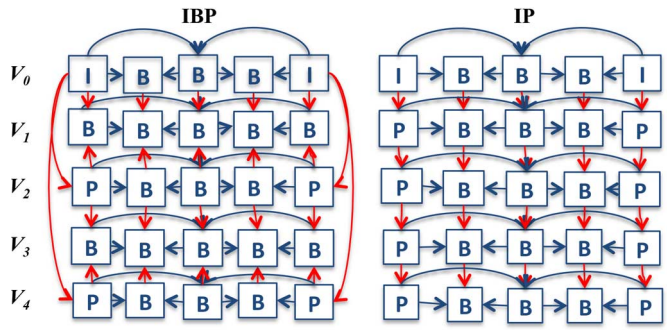

Fig. 3. IBP and IP interview prediction structures (PSs) for five views. The temporal prediction structure is based on hierarchical B-frames for all the views.

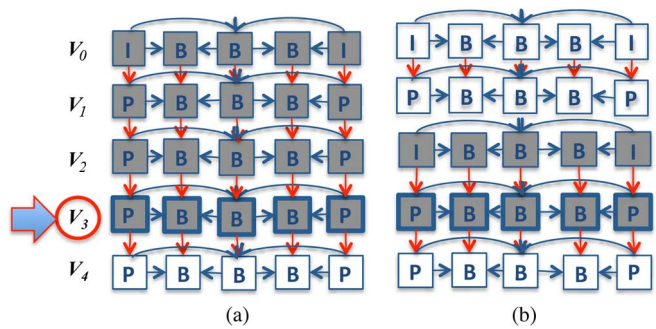

Fig. 4. Interview coding dependencies example. Two IP PSs are illustrated along with the coding dependencies: (a) with only one key view (view 0) and (b) with two key views (views 0 and 2). The frames that need to be transmitted, in order to decode a GOP from view 3 are shown in grey.

IV. Rate and Distortion Modeling

To fully characterize the IMVS system, we now define the rate and distortion models considered in this paper. In the following, we use F to denote the texture of a frame and $\mathcal{F}$ to denote both the texture and depth of a frame. Both F and $\mathcal{F}$ terms are used to refer to frames fully covered by a single type of slice, namely I-, P- or B-slices.

A. Coding Rate

The coding rate (CR) is defined as the total number of bits per unit of time necessary to code both the texture and depth maps of a multiview sequence and it may be computed as:

$$CR = f\frac{\sum_{g=1}^{G}\sum_{c=1}^{C}\sum_{n=1}^{N} n_b\left(\mathcal{F}_{c,n}^{g}(PS_g, Q_g)\right)}{GNC} \quad (1)$$

where f is the frame rate in frames per second, G the total number of GOPs per view, C the number of coded views, N the number of frames per view in a GOP (we assume that all the views have the same GOP size) and $n_b(\mathcal{F}_{c,n}^{g})$ the number of bits used to code frame $\mathcal{F}_{c,n}^{g}$ of view $c \in \mathcal{C}$ at time instant n in GOP g. The number of bits necessary to code frame $\mathcal{F}_{c,n}^{g}$ depends on the PS and the set of QPs used to code the texture and depth on each particular GOP g, PS_g and $Q_g = (Q_t^g, Q_d^g)$. It is important to mention that since we consider that the PS may vary on a GOP basis, also the texture and depth QPs should vary in order to better match the system constraints. Typically, a PS with only one key view, meaning a maximum number of inter-view dependencies, and a coarse quantization should result in higher compression efficiency or lower coding rate, CR, in (1).

B. Transmission Rate

The transmission rate (TR) is here associated to a point-to-point connection where a dedicated video stream is transmitted between two network nodes. This transmission model is useful in content on-demand scenarios where users act independently; hence there are not many streams that could be shared between them as normally the probability that two or more users request the same video stream at the same time is very low. The TR depends on the PS considered, in particular on the interview PS. For instance, in order to decode a particular frame, other frames from the same time instant but from different views might have to be transmitted and processed before decoding the requested view. This is illustrated in Fig. 4, where an example of the effect of interview dependencies is presented. In addition, the TR also depends on which view is requested by the user, notably whether it is a coded or virtual view. If the requested view is a coded view, $c \in \mathcal{C}$, only its texture information has to be transmitted. Otherwise, if the requested view is a virtual view, $v \in \mathcal{V}$, both the texture and depth maps of the closest right and left coded views have to be transmitted, if not already available, so that the user can synthesize the requested virtual viewpoint. This is illustrated in Fig. 5, where user A requests a virtual view (view 2) while user B asks for a coded view (view 5). Then, coded views 1 and 3 (texture and depth maps) have to be transmitted to user A, in order to synthesize the requested virtual view, while for user B, only the texture information of view 5 has to be sent.

Before defining the transmission rate TR, we need to define the so-called frame- and GOP-dependency path size. Similar

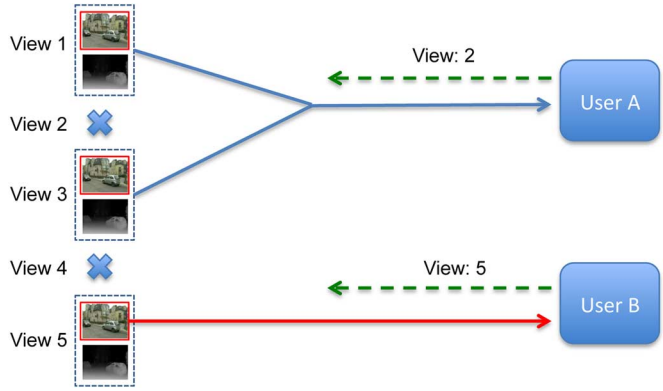

Fig. 5. Transmission model example where views $\{1,2,3,4,5\} \in \mathcal{U}$; $\{1,3,5\} \in \mathcal{C}$ and $\{2,4\} \in \mathcal{V}$. User A requests virtual view 2 and User B coded view 5 (dashed arrows). Coded views 1 and 3 have to be transmitted to user A, in order to synthesize the requested virtual view, while for user B only the texture information of view 5 need to be sent.

to the transmission cost defined in [11], the frame-dependency path size $\phi(F_{u,n}^{g})$ corresponds to the number of bits that have to be transmitted to be able to decode or synthesize a particular texture frame from view $u \in \mathcal{U}$. The definition of $\phi(F_{u,n}^{g})$ depends on whether $F_{u,n}^{g}$ corresponds to a frame from a coded view, $F_{c,n}^{g}$, for all $c \in \mathcal{C}$, or from a virtual view, $F_{v,n}^{g}$, for all $v \in \mathcal{V}$. If $F_{u,n}^{g}$ corresponds to a frame from a coded view, $F_{c,n}^{g}$, $\phi(F_{c,n}^{g})$ is recursively defined as:

$$\phi\left(F_{c,n}^{g}\right) = n_b\left(F_{c,n}^{g}\left(PS_g, Q_t^g\right)\right) + \sum_{\hat{c}\in\{c-1,c+1\}} \phi\left(F_{\hat{c},n}^{g}\right) + \sum_{\hat{n}\in\{1,\cdots,N\}\setminus n} \phi\left(F_{c,\hat{n}}^{g}\right) \quad (2)$$

where $F_{\hat{c},n}^{g}$ and $F_{c,\hat{n}}^{g}$ are the spatial and temporal reference frames for $F_{c,n}^{g}$, respectively. The frame $F_{\hat{c},n}^{g}$ corresponds to the reference frame of $F_{c,n}^{g}$ from the same time instant but from one of the two neighboring views (depending on the interview PS), while frame $F_{c,\hat{n}}^{g}$ is a reference frame from the same view $c \in \mathcal{C}$ and GOP g, but at different time instant. In (2) each frame is considered once, so redundancy is avoided.

On the other hand, if $F_{u,n}^{g}$ corresponds to a frame from a virtual view, $F_{v,n}^{g}$, the texture and depth data of the closest right and a left coded view, $\mathcal{F}_{r,n}^{g}$ and $\mathcal{F}_{l,n}^{g}$, for $r, l \in \mathcal{C}$, need to be transmitted and decoded in order to synthesize frame $F_{v,n}^{g}$. Therefore, $\phi(F_{v,n}^{g})$ becomes:

$$\phi\left(F_{v,n}^{g}\right) = \phi\left(\mathcal{F}_{r,n}^{g}\right) + \phi\left(\mathcal{F}_{l,n}^{g}\right) \quad (3)$$

where, $\phi(\mathcal{F}_{r,n}^{g})$ and $\phi(\mathcal{F}_{l,n}^{g})$ are still the frame dependency paths of frames $\mathcal{F}_{r,n}^{g}$ and $\mathcal{F}_{l,n}^{g}$, where both texture and depth data are considered. Remember that here we consider that texture and depth data use the same optimized PS, so that the coding dependencies are the same for both data types. Then, $\phi(\mathcal{F}_{r,n}^{g})$ and $\phi(\mathcal{F}_{l,n}^{g})$ are recursively defined as in (2); for instance, in the case of $\phi(\mathcal{F}_{r,n}^{g})$ we have:

$$\phi\left(\mathcal{F}_{r,n}^{g}\right) = n_b\left(\mathcal{F}_{r,n}^{g}\left(PS_g, Q_g\right)\right) + \sum_{\hat{c}\in\{c-1,c+1\}} \phi\left(\mathcal{F}_{\hat{c},n}^{g}\right) + \sum_{\hat{n}\in\{1,\cdots,N\}\setminus n} \phi\left(\mathcal{F}_{c,\hat{n}}^{g}\right) \quad (4)$$

The frame-dependency path size for the left reference view, $\phi(\mathcal{F}^g_{l,n})$, is similarly defined.

As a consequence, the number of bits required to decode or synthesize all the frames in a GOP g of a particular view $u \in \mathcal{U}$, named GOP-dependency path size ϕ^g_u, is defined as:

$$\begin{aligned}\phi^g_u &= \sum_{n=1}^{N} \phi\left(F^g_{u,n}\right) \\ &= \begin{cases} \sum_{n=1}^{N} \phi\left(F^g_{c,n}\right), & \text{if } u = c \in \mathcal{C} \\ \sum_{n=1}^{N} \phi\left(F^g_{v,n}\right), & \text{if } u = v \in \mathcal{V} \end{cases}\end{aligned} \tag{5}$$

Finally, we compute the overall *expected point-to-point transmission rate, TR*, as:

$$TR = f\frac{\sum_{g=1}^{G} E\left\{\phi^g_u\right\}}{GN} \tag{6}$$

where $E\{\phi^g_u\}$ is the expectation of the GOP-dependency path size ϕ^g_u, which is defined as $E\{\phi^g_u\} = \sum_{u=1}^{U} p^g_u \phi^g_u$, considering the view popularity model, p^g_u, to express the user preferences for the various views in a particular GOP, common for all the views. Therefore, assuming that the texture and depth QPs are fixed, by increasing the GOP-dependency path size (by increasing the number of interview dependencies), the TR increases. This is the opposite of what happens for CR.

C. Distortion

The average distortion for GOP g in view u, D^g_u, corresponding to the coding noise associated to the quantization process, is taken as the temporal average of the distortion per frame in GOP g, $D^g_{u,n}$:

$$D^g_u = \frac{\sum_{n=1}^{N} D^g_{u,n}}{N} \tag{7}$$

If the view $u \in \mathcal{U}$ corresponds to a coded view, $c \in \mathcal{C}$, its distortion D^g_c depends only on the texture QP, Q^g_t. Otherwise, if u is a virtual view, $v \in \mathcal{V}$, its distortion D^g_v, depends both on the texture and depth QPs, $Q_g = (Q^g_t, Q^g_d)$, used to encode the right and left reference views; $r, l \in \mathcal{C}$. Remember that Q_g is constant for all the corresponding GOP in the various views. The distortion perceived by the user for a particular GOP g takes the value D^g_u with probability p^g_u (considering the view popularity factor). Then, the expected distortion in a specific GOP g, D_g, for the multiview sequence is defined as:

$$\begin{aligned}D_g &= \sum_{u=1}^{U} p^g_u D^g_u \\ &= \sum_{c=1}^{C} p^g_c D^g_c\left(Q^g_t\right) + \sum_{v=1}^{V} p^g_v D^g_v\left(Q_g\right)\end{aligned} \tag{8}$$

Note that the distortion of both coded and virtual views, D^g_c and D^g_v, mainly depends on the QPs of the coded or reference views and not on the PS chosen.

We measure the distortion due to different coding choices in order to select the best coding strategy. To quantify the distortion of the coded views, D^g_c, we measure the mean-squared-error (MSE) between the original view and its coded version. Regarding the distortion of the virtual views, D^g_v, typically there are no original frames available to compute the same metric or any full reference objective quality metric. A commonly used solution available in the literature consists in computing a virtual reference view from the uncompressed texture and depth data of the closest right and left coded views. Then, this synthetic view is taken as benchmark to evaluate the distortion, e.g. the MSE, of the same view synthesized from the decoded reference views [24]. Alternatively, one could use a distortion model for the virtual views, instead of computing it explicitly using the available data. However, it is hard to build good distortion/quality models due to the numerous dependencies of the synthetic views, so we preferred to compute the distortion using the view synthesized from the uncompressed data.

Finally, the expected distortion for the overall multiview sequence is defined as:

$$D = \frac{\sum_{g=1}^{G} D_g}{G} \tag{9}$$

Hereafter, for the sake of simplicity, we use the terms distortion and transmission rate when referring to the expected distortion and expected point-to-point transmission rate per sequence, respectively.

V. Problem Formulation

After describing the main characteristics of our IMVS system, we shall now formulate the optimization problem. The problem addressed here is to find the optimal texture and depth interview PS per GOP, $PS^* = \{PS^*_1, PS^*_2, \cdots, PS^*_G\}$, together with their associated optimal texture and depth QPs, $Q^* = \{Q^*_1, Q^*_2, \cdots, Q^*_G\}$ to encode a predefined set of views, minimizing the distortion D while considering the following storage and bandwidth related constraints:

- *Storage constraint*—For convenience, we express the storage capacity of the system as a rate, CR, notably as the total number of bits per unit of time used to code all the views, considering both texture and depth. The constraint states that the coding rate shall not exceed the maximum storage capacity of the system, CR_{max}.
- *Bandwidth constraint*—Moreover, the transmission rate, TR, for each user is limited by the maximum data rate supported by the network for any user, namely TR_{max}.

In summary, the optimization problem may be written as follows:

$$\{PS^*, Q^*\} = \arg\min_{PS,Q} D(Q) \tag{10}$$

such that,

$$\begin{aligned} CR(PS, Q) &\le CR_{max} \quad \text{Storage constraint} \\ TR(PS, Q) &\le TR_{max} \quad \text{Bandwidth constraint} \end{aligned}$$

where CR, TR and D are calculated as in (1), (6) and (9), respectively. When all the GOPs have the same probability of being requested by the user, meaning a static view temporal popularity model is assumed, the optimization problem defined in (10) can be independently solved for each GOP. Then, the optimal PS per GOP g, PS^*_g and associated texture and depth QPs, Q^*_g, corresponds to those minimizing the GOP distortion, D_g, as defined in (8):

$$\left\{PS^*_g, Q^*_g\right\} = \arg\min_{PS_g, Q_g} D_g(Q_g) \tag{11}$$

such that,

$$CR_g = \frac{f}{NC}\sum_{c=1}^{C}\sum_{n=1}^{N} n_b\left(\mathcal{F}_{c,n}^{g}\left(PS_g, Q_g\right)\right) \leq CR_{max}$$

$$TR_g = \frac{f}{N} E\left\{\phi_u^g\right\} \leq TR_{max}$$

where the expressions for the storage and bandwidth constraints are calculated from (1) and (6), respectively.

For the sake of simplicity, we assume that an optimal bitrate allocation (eventually at GOP level) between texture and depth is known. A different texture and depth rate ratio is expected for different sequences, as it has been shown to be content dependent [25], [26].

$$CR_d^g \leq CR_{d,max} \quad CR_t^g \leq CR_{t,max} \tag{12}$$

Solving the combinatorial optimization problem defined in (11) can be very computationally intensive, notably if exhaustive search (ES) is applied. Indeed, the number of possible interview PSs exponentially grows with the number of views in the multiview set, and for each PS multiple texture and depth QPs configurations are possible. Therefore, in the following section we propose a greedy algorithm that finds near-optimal PSs and associated texture and depth QPs, with remarkably reduced complexity, able to minimize the distortion under storage and bandwidth constraints.

VI. Optimization Algorithm

In this section, we propose a novel optimization algorithm that is able to find, for each GOP over all views, with a reduced complexity, a near-optimal PS with associated texture and depth QPs, given some IMVS system constraints. To significantly reduce the overall complexity regarding an exhaustive search (ES) approach, we propose a greedy optimization solution, which basically reduces the set of considered PSs without significant compression performance penalty. With this approach, the problem in (11) is solved by breaking it down into a series of stages, S_i, which are successively solved, one after the other. To better understand these different stages and how they depend on each other, we adopt a graph to embody all this information. Then, for each GOP over all views, the optimization problem in (11) is solved based on this stage graph.

A. Stage Graph Creation

The stage graph defines the various phases of the solution for the problem in (11). Each stage S_i includes a set of associated states representing the possible PS solutions at each phase of the proposed algorithm. These PSs are then processed in order to find the best PS and QPs in the stage, denoted as PS_i^{g*} and Q_i^{g*}. The states of consecutive stages are linked if they contain a similar sub-structure, which is defined in terms of key views position. In the following, we describe the two main steps in the stages graph creation process, notably the states and links definition.

1) States Definition: We define the states in our stages graph in terms of the number of key views in the interview prediction structure. Thus, the states in a particular stage correspond to the PSs with the same number of key views (e.g., $1, 2, \cdots, C$), in different positions of the multiview set. We start by including in the first stage, S_1, all possible PSs (for the IBP and IP PSs considered in this paper) with only one key view. This corresponds to the solutions with the maximum number of interview coding dependencies, thus associated with maximum compression efficiency and also maximum transmission rate in an IMVS system. Then, we gradually increase the number of key views in the PSs as we move towards the following stages, until the last stage, S_C, where all the C views are independently encoded. This corresponds to the absence of interview coding dependencies, hence minimum compression efficiency and minimum transmission rate in an IMVS system. Therefore, for fixed texture and depth QPs, by moving from stage S_1 to stage S_C, we are, in general, moving along solutions from a maximum TR (minimum CR) to a minimum TR (maximum CR), as the redundancy between the views increases i.e., the number of interview dependencies decreases.

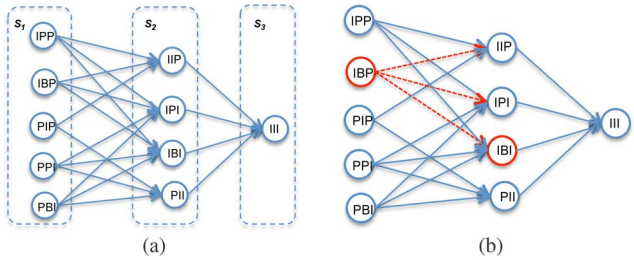

Fig. 6. Example of a three stages graph definition and PS selection. (a) Stages states and link definition; (b) Sub-stage $SS_2 = \{\text{IIP}, \text{IPI}, \text{IBI}\}$, given that $PS_1^{g*} = \text{IBP}$.

2) Links Definition: To link the states of two consecutive stages, we assume that the optimal PS, PS_i^{g*}, in a particular stage S_i for a specific GOP g, determines the optimal position of the i key views in the final optimal PS. This means, for example, that the optimal PS in S_1 determines the positioning of one of the key views in the optimal PS solution. Therefore, a link is defined between two states j, k, associated to $PS_{i-1,j}^{g}$ and $PS_{i,k}^{g}$ from stages S_{i-1} and S_i, if the $i-1$ key views in $PS_{i-1,j}^{g}$ keep their position in $PS_{i,k}^{g}$. This is illustrated in Fig. 6(a) where the different states are represented by circles and the links are defined between PSs of consecutive stages that preserve the key views position. The set of PSs in stage S_i linked to a same PS in stage S_{i-1}, $PS_{i-1,j}^{g}$, is called a sub-stage of S_i and denoted as $SS_{i,j}$, given $PS_{i-1,j}^{g} \in S_{i-1}$. In this work, there is only one sub-stage relevant for each stage, this means the one corresponding to the optimal PS in the previous stage. Therefore, to shorten the sub-stage notation, here a sub-stage of S_i is denoted as SS_i, which is associated to PS_{i-1}^{g*}, while $|SS_i|$ stands for the number of states in SS_i. For instance, in Fig. 6(b), the IIP, IPI and IBI PSs define SS_2, given that $PS_{i-1}^{g*} = \text{IBP}$. In the particular case of stage S_1, $SS_1 = S_1$, as there is no previous stage.

B. Iterative PS and Texture and Depth QPs Selection

The stages of the graph are successively processed for each GOP of the multiview sequence, starting with stage S_1, until the adopted stopping criterion is fulfilled, meaning that the best PS for a particular GOP g, PS_g^*, (defined over all the coded views) has been found together with the optimal texture and depth QPs, Q_g^*. At each stage S_i, only the PSs in the sub-stage

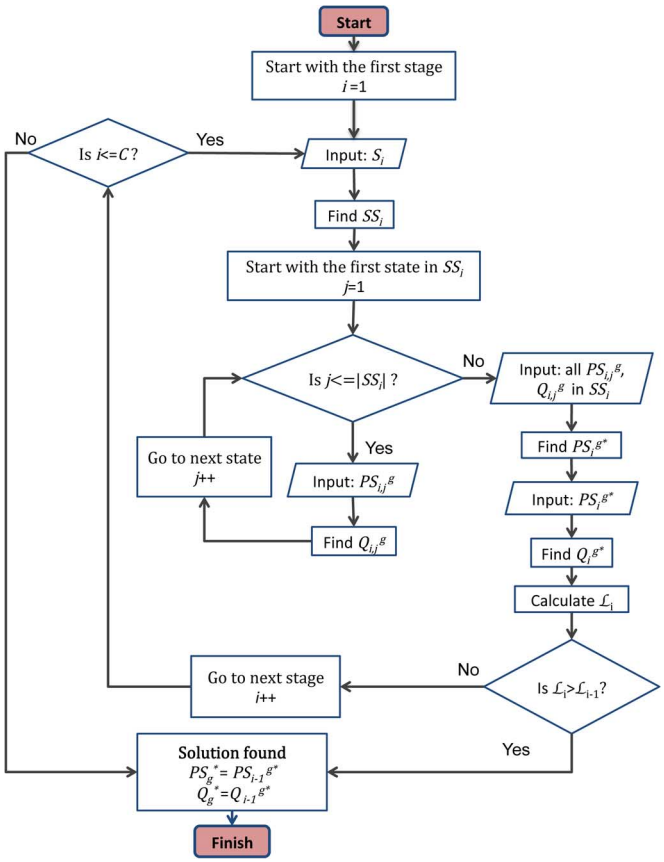

Fig. 7. Flowchart of the proposed optimization algorithm, after the stage graph creation.

SS_i, given PS_{i-1}^{g*}, are processed to find the optimal PS and Q, this means PS_i^{g*} and Q_i^{g*}. The flowchart in Fig. 7 summarizes our optimization algorithm, after the creation of the stage graph.

The optimal PS for GOP g and sub-stage SS_i, PS_i^{g*}, and associated optimal texture and depth QPs, Q_i^{g*}, are found by alternatively solving the problem in (11) for the PSs and QPs in SS_i. In particular, the following steps are followed for each S_i, starting with S_1:

1) Initialize $Q_{i,j}^g$: For each PS in sub-stage SS_i, find $Q = (Q_t, Q_d)$ that satisfies the texture and depth components of the storage constraint, as defined in (12). We denote it as $Q_{i,j}^g$, which is associated to $PS_{i,j}^g$ from sub-stage SS_i and state $j \in SS_i$, as Q may take different values for different PSs in SS_i. By initializing $Q_{i,j}^g$, for each PS in SS_i, $PS_{i,j}^g$, such that it satisfies one of the problem constraints in (11), we are trying to find a set of texture and depth QPs that is close enough to the optimal one. Here, we have only considered the storage constraint, but the bandwidth constraint could have been also used if preferred.

2) Find Optimal PS, PS_i^{g*}: Here, we optimize the problem in (11) only for the PSs in SS_i, while the texture and depth QPs set is kept fixed for each PS. In particular, we consider $Q_{i,j}^g$ (Section VI-B-1) as the QPs set for each PS in SS_i. Hence, the problem addressed here is to find the optimal PS in SS_i for the GOP g, PS_i^{g*}, that minimizes the GOP distortion D_g given some storage and bandwidth constraints:

$$PS_i^{g*} = \arg\min_{PS_{i,j}^g} D_g\left(PS_{i,j}^g\right), \quad \forall PS_{i,j}^g \in SS_i \tag{13}$$

such that,

$$CR_g \leq CR_{max} \quad TR_g \leq TR_{max}$$

where we do not consider the texture and depth components of the CR independently, as for each PS we have already found the texture and depth QPs fulfilling the storage constraint for the texture and depth maps (Section VI-B-1).

To solve the combinatorial problem in (13), we apply the Lagrangian relaxation approach, where according to [27] the constraints are first relaxed by adding them into the objective function with an associated weight (the Lagrangian multiplier). In our case, we move the storage and bandwidth constraints, as in (13), to the objective function with the Lagrangian multipliers, $\{\lambda, \mu\} \geq 0$. Each Lagrangian multiplier represents a penalty to be added to a solution that does not satisfy the considered constraints. Then, the problem in (13) is relaxed as follows:

$$\mathcal{J}_i\left(PS_{i,j}^g, \lambda, \mu\right) = \min_{PS_{i,j}^g} \{D_g - \lambda(CR_{max} - CR_g) - \mu(TR_{max} - TR_g)\} \tag{14}$$

In (14), we have eliminated the constraints from (13), but the number of variables has increased with the number of eliminated constraints or the number of Lagrangian multipliers used. To find the optimal values for the Lagrangian multipliers, λ and μ, we solve the Lagrangian dual problem [27]:

$$\{\lambda^*, \mu^*\} = \arg\max_{\lambda,\mu} \mathcal{J}_i \tag{15}$$

Finally, considering only the PSs in SS_i, the best PS for GOP g and stage S_i, PS_i^{g*}, is the one minimizing (14) for the optimal Lagrangian multipliers obtained in (15).

3) Find Optimal Q, Q_i^{g*}: Given the optimal PS in SS_i and GOP g, PS_i^{g*}, the problem addressed here is to find the optimal set of texture and depth QPs, Q_i^{g*} minimizing the distortion given some storage and bandwidth constraint:

$$Q_i^{g*} = \arg\min_{Q} D_g(Q) \tag{16}$$

such that,

$$CR_d^g \leq CR_{d,max} \quad CR_t^g \leq CR_{t,max} \quad TR_g \leq TR_{max}$$

Differently from 13, here we consider the texture and depth components of the CR independently, as we need to find the set of texture and depth QPs, Q_i^{g*}, satisfying these constraints, while in 13, for each PS, we have already selected the set Q satisfying both of the CR constraints.

As in Section VI-B-2, to solve the problem in (16), we apply the Lagrangian relaxation approach with the Lagrangian multipliers, $\{\alpha, \beta, \gamma\} \geq 0$:

$$\mathcal{L}_i(Q, \alpha, \beta, \gamma) = \min_{Q} \{D_g - \alpha\left(CR_{t,max} - CR_t^g\right) - \beta\left(CR_{d,max} - CR_d^g\right) - \gamma(TR_{max} - TR_g)\} \tag{17}$$

In order to find the optimal α, β and γ values, we solve the following Lagrangian dual problem:

$$\{\alpha^*, \beta^*, \gamma^*\} = \arg\max_{\alpha,\beta,\gamma} \mathcal{L}_i \tag{18}$$

Then, the best Q for GOP g and stage S_i, Q_i^{g*}, is the one minimizing (17) for the optimal Lagrangian multipliers obtained in (18).

The optimal PS, PS_i^{g*}, found in Section VI-B-2 using $Q_{i,j}^g$, with high probability, is not changed after modifying the texture and depth QPs to the optimal ones, Q_i^{g*}. This is due to the similarities between PSs compared in each stage of our algorithm. This statement is further justified in Section VI-D. Therefore, there is not need to recalculate the optimal PS of the current sub-stage for the new texture and depth QPs, Q_i^{g*}.

Before moving to the following stage, the stopping criterion needs to be checked. This is explained in the following.

C. Stopping Criterion Checking

The decision to process the next stage in the graph or to stop the PS selection algorithm at the current stage depends on the fulfillment of the following stopping criterion. If $\mathcal{L}_i$ is larger than $\mathcal{L}_{i-1}$, then stop the optimization algorithm as $PS_g^* = PS_{i-1}^{g*}$ and $Q_g^* = Q_{i-1}^{g*}$ define the locally optimum solution, since moving to the next stage will increase the Lagrangian cost, which is not desirable. In other words, by moving from stage S_{i-1} to stage S_i, at a fixed quality, we are in general moving to solutions with higher coding rate and lower transmission rate, as the number of interview coding dependencies decreases from one stage to the other. Then, an increase of $\mathcal{L}_i$, as defined in (17), means that the distortion has increased in order to satisfy the storage capacity constraint. Thus, as we move forward to the following stages, after the first increase of the $\mathcal{L}_i$ value, we expect $\mathcal{L}_i$ to monotonically increase, as the CR will become higher (for a fixed quality level). As a result, moving to upcoming stages, after the rise of the Lagrangian cost $\mathcal{L}_i$, will only increase the complexity of the algorithm with no benefits in terms of reduced distortion.

It is important to mention that, if S_i is the last stage of the graph, $S_i = S_C$, and $\mathcal{L}_i < \mathcal{L}_{i-1}$ then the locally optimal solution is defined by the current solution $PS_g^* = PS_i^{g*}$ and $Q_g^* = Q_i^{g*}$.

Following this approach we achieve a major reduction on the complexity associated to solving the optimization problem at the price of slightly losing optimality. Although this greedy algorithm determines the optimal PS (and associated optimal Q) at a sub-stage level, the final PS may not be the global optimal one, as at each stage some PSs are ignored. Remind that under the assumptions made, there is only one sub-stage relevant for each stage, this means the one corresponding to the optimal PS in the previous stage. However, a good performance is expected as when adding a new key view at each stage, it is very unlikely that the previous k views do not maintain their optimal position in the multiview set. This argument becomes stronger as k becomes larger as the key views positions providing higher gain are chosen in the first stages of the algorithm. This is confirmed with the experimental results.

D. Sub-stage Optimal PS and Q Relationship

We discuss here why it is reasonable to claim that, at each stage of our greedy algorithm, the optimal PS, PS_i^{g*}, tends to be independent of the level of quality or the QPs used to encode the texture and depth maps, $Q = (Q_t, Q_d)$. This is important to justify the decision taken in Section VI-B of not recalculating the optimal PS for the obtained Q_i^{g*}.

For the PSs considered in this work, the various views are different in terms of the type of coding used at the anchor frame time, meaning an I- P- or B-frame (meaning frames with I, P or B slices). Empirically, we have seen that for the same coding conditions, each type of frame, I, P (in anchor frame position) and B (in anchor and non-anchor frame position) tends to have the same number of bits as another frame of the same coding type in a different view but at the same time instant. This is true because, we compare frames with similar motion characteristics, as they are frames from the same time instant and the scene is typically captured with equidistant cameras. Then, when we compare the CR between the PSs with the same quality and the same number of key views, as it is done at each stage of the graph of our greedy algorithm, the number of bits required for each PS is very similar. In general, the CR values are closer for IP PSs than for IBP PSs, as IP PSs are more similar than IBP PSs. In particular, for the same number of key views, IP PSs have the same number of P anchor frames while IBP PSs may have different number of B and P frames, depending on the position of the key views. On the other hand, when, for a particular quality level, we compare the TR between the PSs with the same number of key views (from the same graph stage), the expected number of bits per unit of time that is needed to decode a view may change from one PS to another. This occurs since the relative position of the different views in the multiview set and the view popularity distribution have a great impact on the TR value (please, refer to Section IV-B). However, as explained before, the PSs compared have, most of the time, the same frame types, as they are PSs from the same graph stage and almost the same number of bits for each frame type. Therefore, as the QP decreases (increases), we expect that the proportion of the increase (decrease) of the transmission rate is the same for all the PSs in the same graph stage. This means that the TR difference between PSs at different QPs is very much constant, making the optimal PS independent of the QP selected.

This can be better understood through an example. Let us consider the multiview sequence *Poznan_Hall2* [28] where $C = 7$ coded views and one virtual view between each pair of coded views are considered, for a total of $U = 7 + 6 = 13$ views. Let us also assume a uniform popularity distribution, which means $p_u^g = 1/13$, $\forall u \in \mathcal{U}$. The seven views available at the server side are encoded using the MVC reference software JMVC v8.2 [29] with all the possible IBP PSs with one key view (corresponding to the IBP PSs in the first stage of our greedy algorithm), where the texture QP, Q_t varies and the depth QP is kept fixed, $Q_d = 42$ (related to the optimal Q_d values shown in Table ?? for *Poznan_Hall2* sequence). Figs. 8(a) and 8(b) show the relationship between CR and Q_t and TR and Q_t, respectively, for all IBP PSs with a single key view this means in stage S_1. The charts show that for different IBP PSs the CR is very

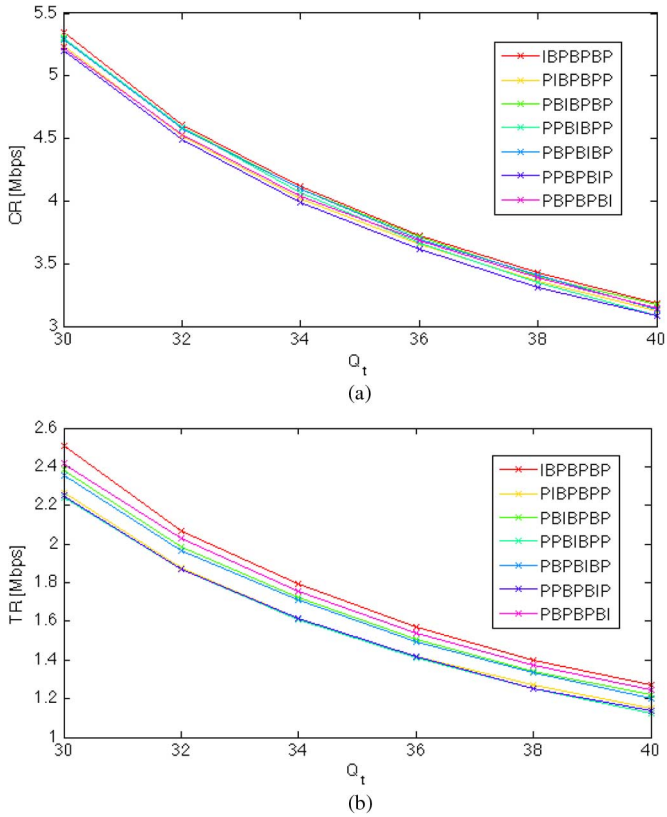

Fig. 8. Relationship between (a) CR and Q_t, and (b) TR and Q_t for IBP PSs with one key view. *Poznan_Hall2* [28] sequence is considered, where a total of $U = 13$ views are available for request ($C = 7$ and $V = 6$).

similar, where the number of P and B views may be different as they depend on the position of the key views. For TR, PSs with less B-frames as anchor frames tend to have better performance. However, the curves representing the efficiency of the PSs (in terms of CR or TR) are rather parallel, for both CR and TR, which means that the efficiency difference between the PSs is independent of the quality level. Therefore, the optimal PS in a particular stage of our greedy algorithm is very much independent of the quality level. The same behavior has been observed for IP PSs and for PSs with more than one key view.

VII. Performance Assessment

This section presents the test conditions and performance results obtained in different scenarios when the PS and associated texture and depth maps QPs search is performed with our proposed algorithm.

A. Content and Coding Test Conditions

As multiview video coding standards, we have considered the MVC, with the reference software JMVC v8.2 [29], and the 3D-HEVC, with the reference software HTM 6.2 [30]. As multiview data, we have used the sequences *Poznan_Hall2* [28] (1920 × 1080, 25 Hz), *Pantomime* [31] (1280 × 960, 30 Hz), *Book Arrival* [32] (1024 × 768, 16.67 Hz), *GT_Fly* [33] (1920 × 1080, 25 Hz) and *Undo Dancer* [34] (1920 × 1080, 25 Hz). Fig. 9 illustrates some frames of the considered sequences. While *Poznan_Hall2*, *Pantomime* and *Book Arrival* are real captured scenes, *GT_Fly* and *Undo Dancer* are computer-generated scenes. For all sequences, a GOP size of 8 frames has been adopted as specified in JCT-3V common test conditions [35]. In the temporal domain, the CQP strategy has been used with a fixed ΔQ equal to 0, 3 and 1 when the temporal layer was equal to 0, 1 and larger than 1, respectively. This is a common ΔQ setting for multiview test sequences. For each sequence, the following conditions have been considered:

- *Poznan_Hall2* [28]—$C = 7$ coded views and $V = 6$ virtual views, each located between two coded views. The seven coded views correspond to the views captured by the first seven cameras. The cameras are horizontally arranged with a fixed distance between neighboring cameras of approximately 13.75 cm.
- *Pantomime* [31]—$C = 10$ coded views and $V = 9$ virtual views, each located between two coded views. The ten coded views correspond to the captured views $\mathcal{C} = \{34 - 43\}$. The cameras are horizontally arranged with a fixed stereo distance.
- *Book Arrival* [32]—$C = 5$ coded views and $V = 4$ virtual views, each located between two coded views. The five coded views correspond to the captured views $\mathcal{C} = \{6, 7, 8, 9, 10\}$. The cameras are horizontally arranged with a spacing of 6.5 cm.
- *GT_Fly* [33]—The five available views are taken as coded views, $\mathcal{C} = \{1, 2, 3, 5, 9\}$, and $V = 4$ virtual views, $\mathcal{V} = \{4, 6, 7, 8\}$. In this sequence, cameras are equidistantly arranged but the camera separation changes with time in order to preserve the 3D perception of the various scenes types: "landscape-view" and "near-view" scenes.
- *Undo Dancer* [34]—As for *GT_Fly*, the five available views are taken as coded views, $\mathcal{C} = \{1, 2, 3, 5, 9\}$, and $V = 4$ virtual views, $\mathcal{V} = \{4, 6, 7, 8\}$. The cameras for this sequence are horizontally arranged with a fixed distance of 20 cm between neighboring views; this means that there are 80 cm of separation between the captured views 5 and 9.

For the sequences *Poznan_Hall2*, *Pantomime* and *Book Arrival* not all the depth maps for the coded views are provided. Therefore, we used the MPEG depth estimation reference software (DERS) [36] to generate the missing depth maps of these three sequences. In addition, we used the MPEG view synthesis reference software (VSRS) [37] based on DIBR, to synthesize the virtual views of all the considered sequences.

Depending on the content characteristics, this means after visual inspection, we have assigned different view popularity distributions to different sets of frames in the considered sequences. The view popularity distributions assumed here are: uniform (equally distributed popularity among the views), exponential (most popular views are located at the left end of the multiview set), inverted exponential (most popular views are located at the right end of the multiview set), Gaussian (most popular views are located at the center of the multiview set) and U-quadratic (most popular views are located at the borders of the multiview set) [3], [4]. Table I shows the frame sets encoded for each sequence and the different popularity distributions assumed for each set. For instance, for the sequence *GT_Fly* two types of scenes have been considered, one where the region of interest of the scene is at the right end of the multiview set (Fig. 9(g)) and another one where the major attention is ex-

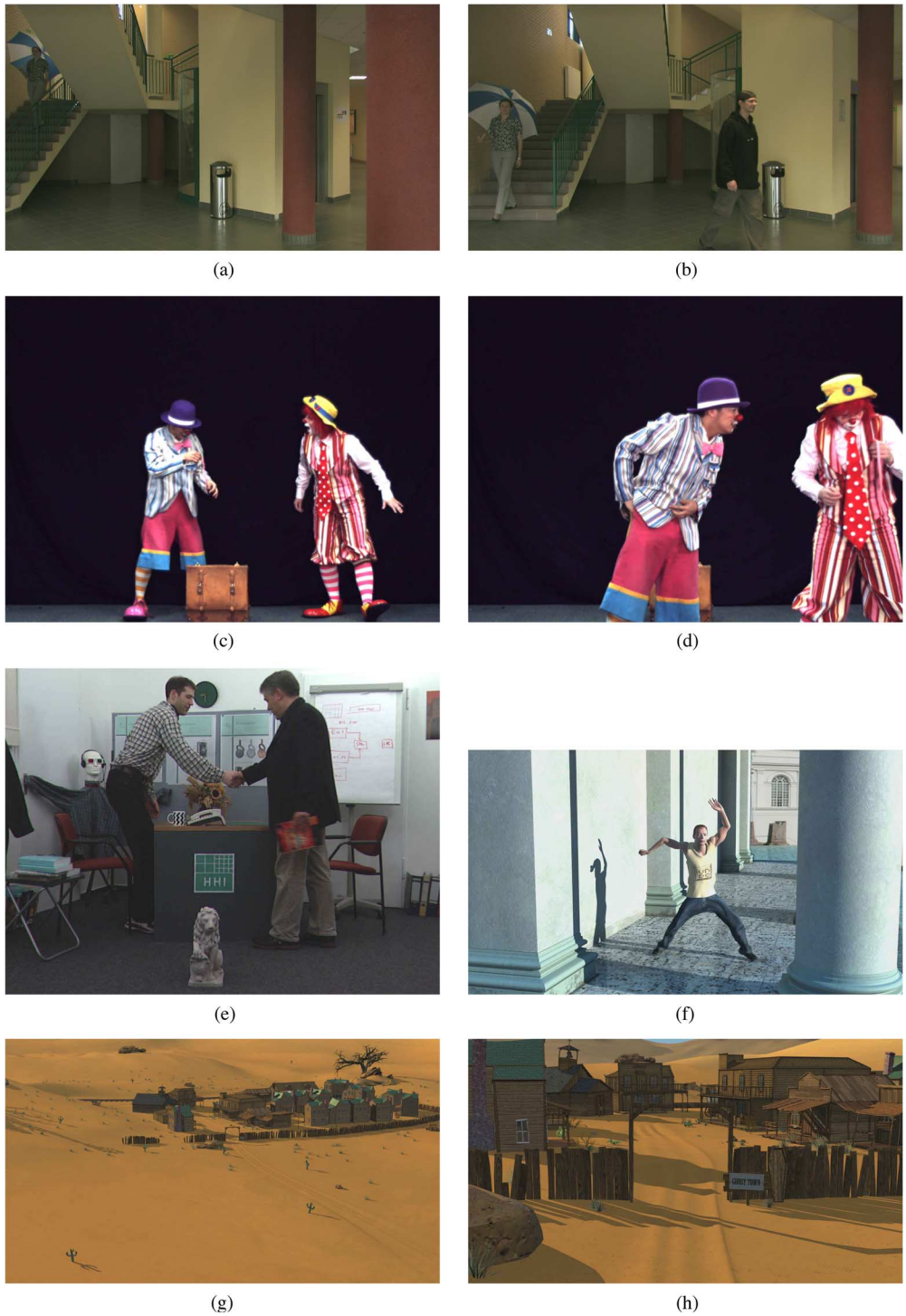

Fig. 9. Content characteristics examples for the frame sets for each test sequence: (a) and (b) *Poznan_Hall2*, (c) and (d) *Pantomime*, (e) *Book Arrival*, (f) *Undo Dancer* and (g)*GT_Fly*. (a) *Poznan_Hall2* sequence, coded view 0, frame 40. (b) *Poznan_Hall2* sequence, coded view 0, frame 200. (c) *Pantomime* sequence, coded view 37, frame 1. (d) *Pantomime* sequence, coded view 37, frame 370. (e) *Book Arrival* sequence, coded view 8, frame 50. (f) *Undo Dancer* sequence, coded view 1, frame 56. (g) *GT_Fly* sequence, coded view 3, frame 1. (h) *GT_Fly* sequence, coded view 3, frame 135.

pected to be at the center of the scene (Fig. 9(h)). Therefore, we have assumed the inverted exponential and the Gaussian distributions for the first and second sets of frames, respectively. A similar reasoning has been applied to the other sequences when selecting the different sets of frames and their associated popularity distribution. As the sequences *Book Arrival* and *Undo Dancer* are very homogeneous in time in terms of the position of the region of interest of the scene only one set of frames (frames 0–50) has been considered. Sample frames of the considered frame sets for each sequence are presented in Fig. 9. We also considered, for *Book Arrival* sequence and its unique set of frames, two different popularity distributions (Gaussian and uniform) to conclude about their impact on the PS and QPs selection.

TABLE I
TEST CONDITIONS: ENCODED FRAME SETS AND POPULARITY DISTRIBUTION FOR EACH TEST SEQUENCE

Sequence	Frame sets	View Pop. Distribution
Poznan_Hall2	0-50	Exponential
	100-150	Gaussian
	150-200	U-Quadratic
Pantomime	0-50	Gaussian
	350-400	Inverted exponential
Book Arrival	0-50	Gaussian
	0-50	Uniform
GT_Fly	0-50	Inverted exponential
	125-175	Gaussian
Undo Dancer	0-50	Gaussian

TABLE II
TEST SCENARIOS: BANDWIDTH AND STORAGE CAPACITY FOR EACH SEQUENCE

Sequence	TR_{max} [Mbps]	$CR_{t,max}$ [Mbps]	$CR_{d,max}$ [Mbps]	CR_{max} [Mbps]
Poznan_Hall2	1	2	2	4
Pantomime	1.8	4.5	1	5.5
Book Arrival	0.7	1	0.7	1.7
GT_Fly	3.7	5.5	1.3	6.8
Undo Dancer	3	5	1	6

B. Storage and Transmission Constraints

Given the different sequence characteristics, the best PS and associated texture and depth maps QPs have been found for various scenarios defined in terms of bandwidth and storage capacity. These scenarios are specified in Table II for each sequence under consideration. The defined TR_{max} and CR_{max} values were chosen in order to have a good video quality in terms of PSNR (30–40 dB). These values are different for the various sequences due to the particular content characteristics and image size. Regarding the allocation of the texture and depth coding rate, $CR_{t,max}$ and $CR_{d,max}$, we empirically found the appropriate ratio of the rate that provided the lowest expected distortion, as defined in (9). For instance, for the *Book Arrival* sequence the best percentage of rate allocated to the depth, CR_d, would be around 40% of the available bitrate budget. These values are consistent with the texture and depth maps rate allocation results available in the literature [25], [26], where they observe that the optimal bitrate ratio is significantly different depending on the sequence characteristics.

C. Results and Analysis

In Table III and IV, the optimal PSs and associated texture and depth maps QPs are shown for each sequence and set of frames when MVC and 3D-HEVC are used as codecs, respectively. We compare the performance of our proposed algorithm with the exhaustive search (ES) approach, which guarantees to find the global optimal PS, this means the PS minimizing the distortion while fulfilling the storage and bandwidth constraints. In the exhaustive search approach, at each stage of our graph, all the PSs and possible QPs are evaluated, while in our optimization algorithm only the PSs in each sub-stage are considered. Due to the content similarity and fixed view popularity distribution, the PS_g^* and Q_g^* found for all GOPs, of each frame set, were always the same. Therefore, in Table III and IV only one PS_g^* and Q_g^* are shown per frame set and sequence.

TABLE III
MVC GREEDY AND EXHAUSTIVE SEARCH SOLUTIONS: RESULTS AND PERFORMANCE COMPARISON

Sequence	Frame sets	ES (Q_t, Q_d)	Greedy (Q_t, Q_d)	$\Delta\mathcal{L}$, ΔT [%]
Poznan_Hall2	0-50	IIBIBPP	IIBIBPP	0, 71
$C = 7$, $V = 6$		(36, 41)	(36, 41)	
	100-150	PBIIIBP	PBIIIBP	0, 71
		(37, 41)	(37, 41)	
	150-200	IIBPBIP	IIBPBIP	0, 72
		(37, 42)	(37, 42)	
Pantomime	0-50	PPPPIIPPPP	PPPPIIPPPP	0, 65
$C = 10$, $V = 9$		(35, 34)	(35, 34)	
	350-400	PPPPPPPPII	PPPPPPPPII	0, 64
		(36, 34)	(36, 34)	
Book Arrival	0-50	PIIPP	PIIPP	0, 42
$C = 5$, $V = 4$		(33, 35)	(33, 35)	
	0-50	PIPIP	PPIIP	2.2, 42
		(33, 36)	(33, 36)	
GT_Fly	0-50	PPPII	PPPII	0, 42
$C = 5$, $V = 4$		(39,33)	(39,33)	
	125-175	PPPII	PPPII	0, 41
		(39,33)	(39,33)	
Undo Dancer	0-50	PPPII	PPPII	0, 42
$C = 5$, $V = 4$		(35,27)	(35,27)	

TABLE IV
3D-HEVC GREEDY AND EXHAUSTIVE SEARCH SOLUTIONS: RESULTS AND PERFORMANCE COMPARISON

Sequence	Frame sets	ES (Q_t, Q_d)	Greedy (Q_t, Q_d)	$\Delta\mathcal{L}$, ΔT [%]
Poznan_Hall2	0-50	IIBPIBP	IIBPIBP	0, 50
$C = 7$, $V = 6$		(33, 40)	(33, 40)	
	100-150	PBIIIBP	PBIIIBP	0, 50
		(34, 40)	(34, 40)	
	150-200	IIBPPBI	IIBPPBI	0, 51
		(34, 40)	(34, 40)	
GT_Fly	0-50	PPIII	PPIII	0, 57
$C = 5$, $V = 4$		(28, 26)	(28, 26)	
	125-175	PPIPI	PPIPI	0, 44
		(29, 26)	(29, 26)	
Undo Dancer	0-50	PPIPI	PPIPI	0, 43
$C = 5$, $V = 4$		(28, 26)	(28, 26)	

The comparison between the proposed greedy algorithm and the ES approach is done here in terms of the Lagrangian cost as specified in (18), and the computational complexity, measured as CPU execution time. We use the normalized difference of the Lagrangian, $\Delta\mathcal{L}$, and the difference of execution time, ΔT, both in percentage. In particular, $\Delta\mathcal{L} = (\mathcal{L}_G - \mathcal{L}_{ES}) * 100/\mathcal{L}_G$ and $\Delta T = (T_{ES} - T_G) * 100/T_{ES}$, where the indexes ES and G are used to differentiate the Lagrangian and execution time obtained with exhaustive search and with our proposed greedy algorithm, respectively. The closer $\Delta\mathcal{L}$ is to zero, the closer the

obtained PS solution is to the optimal solution in terms of RD performance. Moreover, the closer ΔT is to 100%, the larger is the complexity reduction obtained with the proposed algorithm compared to exhaustive search.

In general, the results obtained with the 3D-HEVC codec (Table IV) are very similar to the ones obtained with the MVC codec (Table III), which shows how our proposed selection algorithm is independent of the specific codec used. The differences are due to the higher efficiency of the 3D-HEVC codec compared with MVC, obtaining PSs with lower optimal $Q = (Q_t, Q_d)$, and to the limitations of the 3D-HEVC reference software HTM. In the 3D-HEVC codec only 2 or 3 views can be coded, which limits the possible PSs as at least one key view should be available for every 3 coded views. For instance, in the case of $C = 5$ the only two possible PSs with 3D-HEVC with one key view are: PBIBP and PPIPP. On the other hand, the MVC reference software provides more freedom when selecting the number and position of the key views.

As it can be seen from Table III and IV, the proposed algorithm is able to identify the global optimal PS ($\Delta\mathcal{L} = 0\%$) or near-optimal PS ($\Delta\mathcal{L} = 2.2\%$) with a complexity reduction of up to 72%, in comparison with the ES algorithm. The variation of the complexity reduction with the sequences is due to the number of coded views considered and the number of key views we are able to allocate, given the CR and TR constraints. The larger the number of coded views and allocated key views, the larger the complexity reduction is, as the number of PSs considered with our algorithm, compared with the ones considered with the ES approach, gets smaller. This is the case of *Poznan_Hall2* sequence, where our algorithm achieves a lower complexity reduction when 3D-HEVC is used compared to when MVC is used, as the possible PSs are fewer with the 3D-HEVC codec than with the MVC codec.

In general, we can observe an alignment of the optimal PSs with the popularity models, where for both the greedy and the ES algorithms, the chosen PSs allocate the key views to the most popular viewpoint positions. For instance, for the *Book Arrival* sequence, and the same set of frames, different allocations of the key views are proposed for the two popularity models considered, namely Gaussian and uniform. This is not so obvious for the *GT_Fly* and *Undo Dancer* sequence, where for all the view popularity distributions the optimal key views take the lateral position in the multiview set. This is due to the non-uniform distribution of the coded and virtual views. For instance, when the MVC codec is used, the optimal chosen key views are the two coded views 5 and 9, which serve as reference views to render the virtual views considered. To render virtual views $\{6,7,8\} \in \mathcal{V}$ coded views 5 and 9 are needed as reference views, while virtual view $4 \in \mathcal{V}$ requires coded view 5 as the right reference view. Therefore, since six ({4,5,6,7,8,9}) out of nine available views for user request need coded views 5 and/or 9, it is expected that they should be independently encoded, as they contribute with most of the transmission bitrate.

Different from common PSs in the multiview compression literature, the best PSs, shown in Table III and IV, have more than one key view. This solution results from the trade-off between minimizing the transmission rate (associated to PSs with less interview dependencies) and maximizing the compression efficiency (associated to PSs with more interview dependencies). These results indicate that a pure compression efficiency objective is not ideal in IMVS systems.

Though experiments have been done with the available data sets, which have a limited number of views or a small navigation range, similar results are expected in real IMVS applications where a large number of views should be available for user request and distant views considerably differ in their scene content.

VIII. Conclusion

We have proposed an algorithm that efficiently selects a near-optimal interview PS and associated texture and depth QPs, at GOP level, when the MVD data format is used for IMVS systems. We consider an IMVS system where storage capacity and transmission rate are limited resources. The proposed algorithm is able to reduce the set of relevant PSs, compared with an exhaustive search approach, without significant RD performance penalty. To evaluate the performance of the proposed algorithm, the multiview video coding standards MVC and 3D-HEVC have been considered and simulation results have shown that the global optimal or near-optimal PS can be obtained with the proposed algorithm, while the associated complexity is considerably reduced.

Future work may focus on the extension of the current optimization algorithm to systems where the reference views used to synthesize the considered virtual views are not restricted to be the closest ones, but their choice can be RD optimized to further improve the performance of our algorithm. Also, the implementation of a non-static view temporal popularity model is left for future work.

References

[1] P. Merkle, A. Smolic, K. Muller, and T. Wiegand, "Multi-view video plus depth representation and coding," in *Proc. IEEE Int. Conf. Image Processing*, San Antonio, TX, USA, September 2007.

[2] Y. Liu, Q. Dai, Z. You, and W. Xu, "Rate-prediction structure complexity analysis for multi-view video coding using hybrid genetic algorithms," in *Proc. SPIE, Visual Commun. Image Process.*, San Jose, CA, USA, Jan. 2007.

[3] A. De Abreu, P. Frossard, and F. Pereira, "Optimized MVC prediction structures for interactive multiview video streaming," *IEEE Signal Process. Lett.*, vol. 20, no. 6, pp. 603–606, Jun. 2013.

[4] A. De Abreu, P. Frossard, and F. Pereira, "Fast MVC prediction structure selection for interactive multiview video streaming," in *Proc. Picture Coding Symp.*, San Jose, CA, USA, Dec. 2013.

[5] A. Fiandrotti, J. Chakareski, and P. Frossard, "Popularity-aware rate allocation in multi-view video coding," in *Proc. IEEE Int. Conf. Visual Commun. Image Process.*, Huang Shan, An Hui, China, July 2010, invited paper.

[6] H. Kimata, M. Kitahara, K. Kamikura, Y. Yashima, T. Fujii, and M. Tanimoto, "System design of free viewpoint video communication," in *Proc. Int. Conf. Comput. Inf. Technol.*, Wuhan, China, Sep. 2004.

[7] Y. Liu, Q. Huang, S. Ma, D. Zhao, and W. Gao, "RD-optimized interactive streaming of multiview video with multiple encodings," *J. Vis. Commun. Image Represent.*, vol. 21, no. 5–6, pp. 523–532, Jul. 2010.

[8] T. Fujihashi, Z. Pan, and T. Watanabe, "UMSM: A traffic reduction method on multi-view video streaming for multiple users," *IEEE Trans. Multimedia*, vol. 16, no. 1, pp. 228–241, Jan. 2014.

[9] E. Kurutepe, M. Civanlar, and A. Tekalp, "Client-driven selective streaming of multiview video for interactive 3dtv," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 11, pp. 1558–1565, Nov. 2007.

[10] C. Zhang and D. Florłncio, "Joint tracking and multiview video compression," pp. 77 440P–77 440P-8, 2010 [Online]. Available: http://dx.doi.org/10.1117/12.863066

[11] G. Cheung, A. Ortega, and T. Sakamoto, "Coding structure optimization for interactive multiview streaming in virtual world observation," in *Proc. IEEE Int. Workshop Multimedia Signal Process.*, Cairns, Queensland, Australia, Oct. 2008, pp. 450–455.
[12] G. Cheung, A. Ortega, and N.-M. Cheung, "Interactive streaming of stored multiview video using redundant frame structures," *IEEE Trans. Image Process.*, vol. 20, no. 3, pp. 744–761, Mar. 2011.
[13] X. Xiu, G. Cheung, and J. Liang, "Delay-cognizant interactive streaming of multiview video with free viewpoint synthesis," *IEEE Trans. Multimedia*, vol. 14, no. 4, pp. 1109–1126, Aug. 2012.
[14] G. Cheung, V. Velisavljevic, and A. Ortega, "On dependent bit allocation for multiview image coding with depth-image-based rendering," *IEEE Trans. Image Process.*, vol. 20, no. 11, pp. 3179–3194, Nov. 2011.
[15] J. Chakareski, V. Velisavljevic, and V. Stankovic, "User-action-driven view and rate scalable multiview video coding," *IEEE Trans. Image Process.*, vol. 22, no. 9, pp. 3473–3484, Sep. 2013.
[16] J.-G. Lou, H. Cai, and J. Li, "A real-time interactive multi-view video system," in *Proc. ACM Multimedia*, New York, NY, USA, Nov. 2005.
[17] M. Schmeing and X. Jiang, *Depth Image Based Rendering*, P. S. Wang, Ed. Berlin/Heidelberg, Germany: Springer , 2011.
[18] S. Zinger, L. Do, and P. H. N. de With, "Free-viewpoint depth image based rendering," *J. Vis. Commun. Image Represent.*, vol. 21, no. 5–6, pp. 533–541, Jul. 2010.
[19] Y. Mori, N. Fukushima, T. Fujii, and M. Tanimoto, "View generation with 3D warping using depth information for FTV," in *Proc. 3DTV Conf.: The True Vis.—Capture, Transmiss., Display of 3D Video*, Istanbul, Turkey, May 2008.
[20] K. Muller, H. Schwarz, D. Marpe, C. Bartnik, S. Bosse, H. Brust, T. Hinz, H. Lakshman, P. Merkle, F. Rhee, G. Tech, M. Winken, and T. Wiegand, "3D high-efficiency video coding for multi-view video and depth data," *IEEE Trans. Image Process.*, vol. 22, no. 9, pp. 3366–3378, Sep. 2013.
[21] K. Klimaszewski, K. Wegner, and M. Domanski, "Distortions of synthesized views caused by compression of views and depth maps," in *Proc. 3DTV Conf.: True Vis.—Capture, Transmiss., Display of 3D Video*, Potsdam, Germany, May 2009.
[22] P. Merkle, A. Smolic, K. Muller, 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.
[23] H. Schwarz, D. Marpe, and T. Wiegand, "Analysis of hierarchical B pictures and MCTF," in *Proc. IEEE Int. Conf. Multimedia and Expo*, Toronto, ON, Canada, July 2006.
[24] K. Muller, A. Smolic, K. Dix, P. Merkle, and T. Wiegand, "Coding and intermediate view synthesis of multiview video plus depth," in *Proc. IEEE Int. Conf. Image Process.*, Cairo, Egypt, Nov. 2009, pp. 741–744.
[25] E. Bosc, V. Jantet, M. Pressigout, L. Morin, and C. Guillemot, "Bit-rate allocation for multi-view video plus depth," in *Proc. 3DTV Conf.: True Vision—Capture, Transmiss., Display of 3D Video*, Antalya, Turkey, May 2011.
[26] E. Bosc, P. Riou, M. Pressigout, and L. Morin, "Bit-rate allocation between texture and depth: Influence of data sequence characteristics," in *Proc. 3DTV-Conf.: True Vis.—Capture, Transmiss., Display of 3D Video*, Zurich, Switzerland, Oct. 2012.
[27] D. P. Bertsekas, *Nonlinear Programming*. Belmont, MA, USA: Athena Scientific, 1999.
[28] M. Domański, T. Grajek, K. Klimaszewski, M. Kurc, O. Stankiewicz, J. Stankowski, K. Wegner, and M. Wildeboer, "Poznań multiview video test sequences and camera parameters," Xian, China, ISO/IEC JTC1/SC29/WG11 MPEG 2009/M17050, Oct. 2009.
[29] JMVC 8.2 Software. [Online]. Available: garcon.ient.rwth-aachen.de
[30] HTM 6.2 Software. [Online]. Available: https://hevc.hhi.fraunhofer.de/svn/svn 3DVCSoftware/tags/HTM-6.2/
[31] "Tanimoto Laboratory test sequences for MVC-FTV," [Online]. Available: http://www.tanimoto.nuee.nagoya-u.ac.jp/
[32] I. Feldmann, M. Mueller, F. Zilly, R. Tanger, K. Mueller, A. Smolic, P. Kauff, and T. Wiegand, "HHI Test Material for 3D Video," Archamps, France, ISO/IEC JTC1/SC29/WG11 MPEG 2008/M15413, April 2008.
[33] J. Zhang, R. Li, H. Li, D. Rusanovskyy, and M. Hannuksela, "Ghost Town fly 3DV sequence for purposes of 3DV standardization," Geneva, Switzerland, ISO/IEC JTC1/SC29/WG11 MPEG2010/M20027, Mar. 2011.
[34] D. Rusanovskyy, P. Aflaki, and M. M. Hannuksela, "Undo Dancer 3DV sequence for purposes of 3DV standardization," Geneva, Switzerland, ISO/IEC JTC1/SC29/WG11 MPEG2010/M20028, Mar. 2011.
[35] D. Rusanovskyy, K. Müller, and A. Vetro, "Common test conditions of 3DV core experiments," Geneva, Switzerland, ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, Oct. 2013.
[36] M. Tanimoto, T. Fujii, and K. Suzuki, Depth Estimation Reference Software (DERS) 5.0 Xian, China, ISO/IEC JTC1/SC29/WG11 M16923, Oct. 2009.
[37] M. Tanimoto, T. Fujii, and K. Suzuki, "view synthesis algorithm in view synthesis reference software 2.0 (VSRS 2.0)," Lausanne, Switzerland, ISO/IEC JTC1/SC29/WG11 M16090, Feb. 2008.

Ana De Abreu, photograph and biography not available at the time of publication.

Pascal Frossard, photograph and biography not available at the time of publication.

Fernando Pereira, photograph and biography not available at the time of publication.