Delay-Cognizant Interactive Streaming of Multiview Video with Free Viewpoint Synthesis

Xiaoyu Xiu, *Student Member, IEEE,* Gene Cheung, *Senior Member, IEEE,* Jie Liang, *Member, IEEE*

Abstract

In interactive multiview video streaming (IMVS), a client receives and observes one of many available viewpoints of the same scene, and periodically requests from server view-switches to neighboring views, as the video is played back in time uninterruptedly. One key technical challenge is to design a frame coding structure that facilitates periodic view-switching, and achieves an optimal tradeoff between storage cost and expected transmission rate. In this paper, we first propose three significant improvements over existing IMVS system, and then study the corresponding frame structure optimization. First, using depth-image-based rendering, the new IMVS system enables free viewpoint switching, *i.e.*, by encoding and transmitting both texture and depth maps of captured views, a client can select and synthesize any virtual view from an almost continuum of viewpoints between the left-most and right-most captured views. Second, IMVS system adopts a more realistic Markovian view-switching model with memory that more accurately captures user behaviors than previous memoryless models. View-switching model is used in predicting client's future view-switching patterns. Third, assuming that the round-trip-time (RTT) delay during server-client communication is non-negligible, during an IMVS session, IMVS system additionally transmits redundant frames RTT into future playback, so that zero-delay view-switching can be achieved. Given these improvements, we formalize a new join optimization of the frame coding structure, transmission schedule, and quantization parameters of the texture and depth maps of multiple camera views. We propose an iterative algorithm to achieve fast and near-optimal solutions. The convergence of the algorithm is also demonstrated. Experimental results show that the proposed optimized rate allocation method requires 38% less transmission rate than the fixed rate allocation scheme. In addition, with the same storage, the transmission rate of the optimized frame structure can be up to 55% lower than that of I-frame-only structure, and 27% lower than that of the structure without distributed source coding (DSC) frames.

Index Terms

multiview video, video streaming, media interaction, view synthesis

I. Introduction

Multiview video are videos of the same scene captured time-synchronously by multiple closely spaced cameras from different observation viewpoints. If a viewer can naturally and interactively select one out of many available captured views for observation on a 2D display as the video is played back, viewer can experience a perception

X. Xiu and J. Liang are with the School of Engineering Science, Simon Fraser University, Burnaby, BC, V5A 1S6, Canada. Phone: 778-782-5484. Fax: 778-782-4951. E-mail: {xxa4, jiel}@sfu.ca.

G. Cheung is with National Institute of Informatics, 2-1-2, Hitotsubashi, Chiyoda-ku, Tokyo, Japan 101–8430. E-mail: cheung@nii.ac.jp.

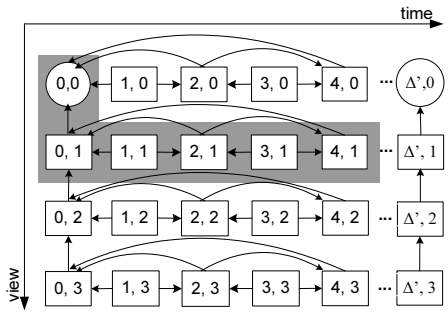


Fig. 1. Example of MVC frame structure, where circles and rectangles denote I- and P-frames, respectively. Each frame $F_{t,v}$ is marked by its time instant t and view v. The frames in the shaded box represent the ones decoder can access during one navigation.

of depth via *motion parallax*; *e.g.*, shifting of a viewer's head can trigger rendering of the correspondingly shifted observed view of the scene [1], [2]. Several prototypes of such multiview video systems [1], [3] have demonstrated an improved viewing experience via this view-switching media interaction.

Much of previous research on multiview video focuses mainly on multiview video coding (MVC) [4], *i.e.*, how to efficiently compress *all* captured videos in a rate-distortion optimal manner, by exploiting the inherent correlation among nearby frames across time and view. However, MVC frame structures are not suitable for *interactive multiview video streaming* (IMVS) [5], [6], [7], where a client periodically selects and requests the aforementioned view-switches from a remote server, and the server in response transmits the requested single-view video for uninterrupted playback at the client. This is because typical MVC frame structures are not designed to provide sufficient decoding flexibility to support this periodic view-switching interaction; hence multiple frames usually need to be transmitted in order for a desired frame to be correctly decoded, resulting in large bandwidth consumption. As an illustration, Fig. 1 shows one MVC frame structure proposed in [4], where I-frames are periodically inserted every Δ' frames to permit some level of random access. In order to facilitate view-switches every Δ frames, the structure in Fig. 1 can be generated with Δ' set to Δ. However, for a small desired view-switching period Δ, this leads to high transmission costs due to frequent I-frame insertion. Alternatively, one can first select a compression-efficient frame structure with $\Delta' \gg \Delta$, and then send to client all the frames required to enable decoding of frames in a single requested view after a view-switch. For instance, letting $F_{t,v}$ denote a frame at time instant t and view v, in order to switch from frame $F_{2,1}$ to frame $F_{3,2}$, given the frames available at decoder buffer in the shaded region in Fig. 1, server would send frames $F_{0,2}$, $F_{2,2}$, $F_{3,2}$ and $F_{4,2}$ to client, but only frame $F_{3,2}$ is displayed. Besides a large resulting transmission rate spike during the view-switch, this also incurs an unwanted overhead in decoding complexity.

Recently, frame structure optimizations [5], [6], [7] for IMVS have been studied. The goal is to design frame structures at encoding time that facilitate periodic view-switching during an IMVS streaming session, and optimally trade off expected IMVS transmission rate and storage required to store the structure. Optimized IMVS frame structures have shown significant reduction in expected transmission rate over naïve frame structures of comparable

sizes. However, the underlying IMVS system that deploys these structures is still simplistic and has several shortcomings. First, the available views for a client to select were limited by the few camera-captured views pre-encoded at server, thus a view-switch could appear abrupt and unnatural to a viewer. Second, when devising view-switching model to predict client's future view-switching patterns, previous IMVS system assumes a memoryless model that is statistically independent in time. However, it has been shown [8] that viewers exhibit temporal dependencies when switching views. Third, previous IMVS system assumes server-client communication takes place over idealized zero-delay network. In a realistic packet-switched network such as the Internet with non-negligible round trip time (RTT) delay, server's responding upon receipt of each client's requested view will mean each client's requested view-switch will suffer at least one RTT delay, hampering interactivity of the viewing experience.

In this paper, we first propose three significant improvements over existing IMVS system, and then study the corresponding frame structure optimization. First, leveraging on the recent advances in depth-image-based rendering (DIBR) [9] that enable synthesis of a virtual intermediate view between two captured views using depth information, new IMVS system encodes *both* the texture and depth maps of captured views into a *video-plus-depth* coding format [10], each at the respective optimized quantization parameter (QP). To enable free-viewpoint view-switching [11], [12], *i.e.*, synthesizing virtual views from an almost continuum of viewpoints between the left-most and right-most captured views, the server transmits texture and depth maps of *two* nearest captured views to the client. This represents a major improvement in interactive viewing experience over previous IMVS systems that are limited to streaming and rendering of captured views only.

Second, given free viewpoint selection is available to clients, new IMVS system adopts a more realistic Markovian view-switching model with memory that more accurately captures user behaviors than previous memoryless models. Third, assuming that the round-trip-time (RTT) delay during server-client communication is non-negligible, during an IMVS session, IMVS system additionally transmits redundant frames RTT into future playback. Doing so means client can enjoy zero-delay view-switching during an IMVS streaming session.

Given these improvements in the new IMVS system, we formalize the joint optimization of the frame encoding structure, transmission schedule, and QPs of the texture and depth maps, and propose an iterative algorithm to achieve fast and near-optimal solutions. Convergence of the proposed algorithm is also demonstrated. Note that though the DIBR tool [9], [10] and view-switching model with memory [3] have both been studied as individual pieces in the literature, this paper is the first attempt to incorporate them into IMVS coding structure optimization. As we shall see in the rest of the paper, it is a non-trivial extension of the previous IMVS work [5], [6], [7] to take into consideration three practical components in IMVS system. Experimental results show that our proposed rate allocation method reduces transmission rate over fixed texture/depth rate allocation methods by up to 38%. In addition, for the same storage, transmission rate of the frame structure generated by our proposed algorithm can be up to 55% lower than that of I-frame-only structures, and 27% lower than that of the structure without DSC frames.

The outline of the paper is as follows. We first overview related work in Sec. II. We then discuss the IMVS system, source model of encoding multiview video, our generalized media interaction model with memory for

view-switching and network delay model in Sec. III. In Sec. IV, we formulate the problem of finding the optimal frame structure, transmission schedule and QPs for encoding texture and depth maps in a network-delay-cognizant manner. In Sec. V, we develop an iterative optimization algorithm to efficiently find a solution for the proposed IMVS problem. Simulation results and conclusion are given in Sec. VI and Sec. VII, respectively.

II. Related Work

We divide our discussion on related work into three sections. We first articulate the difference between interactive and non-interactive media streaming. We then discuss related work in multiview video streaming. Finally, we differentiate our contributions in this paper relative to our earlier work on IMVS.

A. Interactive and Non-Interactive Media Streaming

The communication paradigm for IMVS is one where the server continuously and reactively sends appropriate media data in response to a client's periodic requests for data subsets; we call this paradigm *interactive media streaming*. This is in contrast to *non-interactive media streaming* scenarios like terrestrial digital TV broadcast, where the entire media set is delivered server-to-client before a client interacts with the received data set (*e.g.*, switching TV channel). Interactive media streaming has the advantage of reduced bandwidth utilization since only the requested media subset is transmitted. It is used for a wide range of media modalities, such as interactive light field [13], interactive image browsing [14], flexible video playback [15]. For multiview video, MVC [4], [16] discussed in the Introduction where multiple captured views are compressed efficiently together into a single stream would be suitable for non-interactive media streaming. In contrast, special frame structures need to be designed for the periodic view-switching nature of IMVS [7]. This is the focus of previous IMVS work and this paper as well.

B. Interactive Multiview Video Streaming

For interactive streaming of stored multiview videos, the two-layer approach proposed in [3], [17] can be one solution, where coarse and fine quality layers of several views are grouped and pre-encoded. During actual streaming, a subset of views of low quality plus two views of high quality, carefully selected based on user's behavioral prediction, would then be sent to the client. All transmitted views were subsequently decoded, and the highest quality views that matched the user's at-the-moment desired views were displayed. While the intended IMVS application is the same, our approach is different in that we focus on the optimal tradeoff among transmission rate, storage and view synthesis distortion using combinations of redundant P-frames and DSC frames in our frame structure.

The most similar work to our IMVS work is [8], which developed three separate frame structures to support three types of interactivity: view switching, frozen moment and view sweeping. While the authors recognized the importance of a "proper tradeoff among flexibility (interactivity), latency and bandwidth cost", no explicit optimization was performed to find the best tradeoffs of these quantities in one structure.

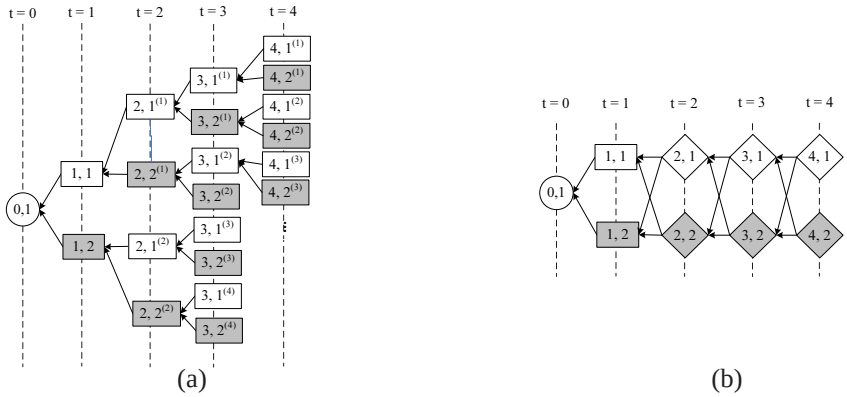


Fig. 2. Two extreme examples of frame structure to enable view-switching for two views (white and grey) for $\Delta = 1$. I-, P- and M-frames are represented by circles, rectangles and diamonds, respectively. (a) P-frames only at switching points; (b) M-frames only at switching points.

C. Previous Work in IMVS

The problem of frame structure optimization for IMVS has been recently studied, where the goal is to design frame structures at encoding time that facilitate periodic view-switching during an IMVS session, while trading off expected transmission rate and storage required to store the structure. The encoding must be performed *without* knowing the exact view trajectory a client will take at stream time [5], [6], [7]. To see intuitively the tradeoff involved, consider the following two extreme examples. For simplicity we let $\Delta = 1$, but restrict allowable switches to only neighboring views on a 1D camera array setup; i.e, only clients observing views k's, $j-1 \leq k \leq j+1$, at time $i-1$ can switch to view j at time i. To encode frame $F_{i,j}$, since temporal playback is not interrupted, at time i one of the previous frames $F_{i-1,k}$'s (for at most three different views k) will be available at the decoder. Thus, one way to support view-switching is to differentially encode one P-frame $P_{i,j}$ for each possible decoded frame $F_{i-1,k}$ in the decoder buffer. We call this approach *redundant P-frames*—redundant in that an original picture $F^o_{i,j}$ is represented by multiple coded versions $P_{i,j}$'s. An example structure to allow view-switching between two views is shown in Fig. 2(a) where only P-frames $P_{i,j}$'s are encoded at view-switching points, each using a predictor $F_{i-1,k}$ of previous instant. As shown in Fig. 2(a), this approach will increase the number of decoding paths at each switching instant by a factor of two, resulting in a tree structure of size $O(2^N)$ if there are N switching instants between two I-frames. So although this approach would lead to a structure with minimum transmission cost (only bandwidth-efficient P-frames are used), the size of the coding structure is impractically large.

At the other extreme, one can construct a *single* coded version of the original picture $F^o_{i,j}$ for all possible decoder states, *i.e.,* a frame (we call *merge frame* or *M-frame*) that can be correctly decoded *no matter* which $F_{i-1,k}$ is in the decoder buffer; see Fig. 2(b) for an example. Obviously, an independently coded I-frame would fit the M-frame reconstruction constraint, but more generally, one can conceive other implementations of M-frame that exploit correlation between the set of possible predictors $F_{i-1,k}$'s and the target $F^o_{i,j}$ for coding gain. Example

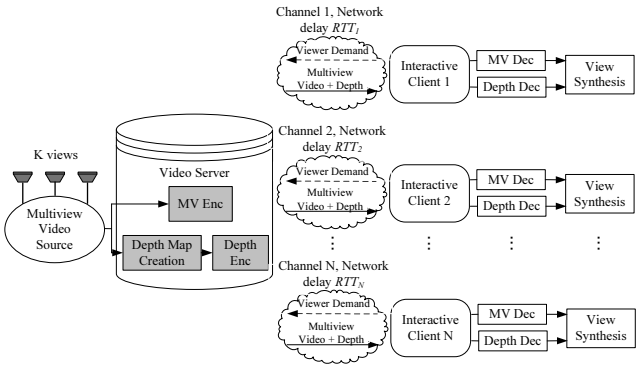


Fig. 3. System overview of the proposed IMVS system.

implementations of M-frames include SP-frames in H.264 [18] and different DSC techniques [19], [20][1]. In general, different implementations of M-frames induce different tradeoffs between storage cost and transmission rate [7]. However, any implementation of M-frame must necessarily have larger transmission rate than a P-frame, since by definition, an M-frame must be encoded under the uncertainty of which one frame in the set of possible predictors $F_{i-1,k}$'s would be available at decoder buffer at stream time. Hence, a structure that uses M-frames exclusively at all view-switching points has high transmission rate but small storage cost (since each original picture is represented by a single coded version).

In our earlier IMVS work, we had posed the IMVS problem as a combinatorial optimization in [21], proved its NP-hardness, and provided two heuristics-based algorithms to find good frame structures for IMVS. A more thorough and analytical treatment of the same problem was given in [5], using only I- and P-frames in the structure. We have also developed two novel DSC implementations to serve as M-frames for IMVS in [20]. Preliminary results of using I-, P- and DSC frames in an IMVS optimized structure is presented in [6]; [7] is a generalization of [6] where the optimization is posed as a search for the best combination of I-, P- and generalized M-frames.

Different from our most recent work in [7] where the number of views available for clients' selections is limited to the set of captured views, in this paper, we focus on coding structure optimization for a new IMVS system that transmits texture and depth maps of captured views, thereby providing free viewpoint synthesis at decoder. For transmission over communication networks with non-negligible round-trip time (RTT) delay, IMVS system transmits frames of possibly selected viewpoints RTT into future playback, so that clients can experience zero-delay view-switching. Given the new IMVS system, our goal is to optimize multiview video frame structure, transmission schedule and QP to encode texture and depth maps for transmission at stream time.

III. System and Media Interaction Model

To facilitate understanding of our contributions in this paper, we first overview the system model for IMVS. We then describe the source model for coded multiview videos, and DIBR used for synthesizing virtual views using

[1]In the context of DSC, "predictor" frames are used as side information for decoding.

coded texture and depth maps of neighboring views. We then discuss a general view-switching model of finite memory that captures user's behavior in selecting (possibly virtual) views. Finally, we discuss our network model that considers the RTT delays between streaming server and clients.

A. System Model for IMVS

The system model we consider for IMVS is shown in Fig. 3, where a *multiview video source* captures time-synchronized videos of a 3D scene from K evenly spaced, horizontally shifted cameras in a 1D array. A *video server* sequentially grabs captured texture and depth maps from the multiview video source[2], and encodes the texture and depth maps separately into the same optimized frame structure $\mathcal{T}$ of I-, P- and M-frames, at their respective optimized QPs. In other words, the same permutation of I-, P- and M-frames used to encode texture maps at one QP, will be used also to encode depth maps using a different QP separately. The video server stores a single data structure $\mathcal{T}$, using which the server can provide IMVS service for multiple clients. An alternative approach of live encoding a unique view traversal of frames for each client's interactively chosen navigation path is computationally prohibitive if the number of clients is large.

A client can request a view-switch every Δ frames, where the requested view can be a captured view or an intermediate virtual view between two captured views. The availability of a large number of virtual views—an almost continuum of views between left-most and right-most captured views—enables finer grain view-switches compared to previous IMVS systems [5], [6], [21], where the available views were limited by the number of capturing cameras, and each view-switch was an abrupt jump from one camera view to another. To facilitate synthesis of a virtual view at the client side, the server always transmits *both* texture and depth maps of the closest left and right captured views. The client then interpolates the requested virtual view using received texture and depth maps via DIBR (to be explained in Sec. III-C). Further, we assume I-frames are inserted every Δ' frames, $\Delta \ll \Delta'$, for all K captured views for some pre-defined level of random access.

Since the same optimized frame structure is used to encode both texture and depth maps of multiview video source, for ease of discussion, we will use the term *picture* to denote both texture and depth maps of the corresponding captured image, and the term *frame* to denote the specific coded version of texture and depth maps of an image. Further, given view-switch period[3] Δ, we use $F^o_{i,j}$ and $F_{i,j}$ to denote a picture and a frame of view j at *view-switch instant* $i\Delta$, *i.e.*, the time at which a client selects her i-th view-switch location.

B. Multiview Video Source Model

As done in [7], in this paper, a picture can be coded as an intra-coded I-frame with no predictor, a differentially coded P-frame with a single predictor, or a conceptual M-frame with *multiple* predictors known at encoding time.

[2]Depth maps can be estimated from texture maps using stereo-matching algorithms [22], or captured directly using time-of-flight cameras [23].

[3]In more general case of $\Delta > 1$, a picture $F^o_{i,j}$ represents Δ consecutive pictures of view j from time $i\Delta$ to time $(i+1)\Delta - 1$, and a frame $F_{i,j}$ represents Δ consecutive actual frames of view j, including a carefully chosen I-, P- or M-frame determined by our optimization algorithm followed by $\Delta - 1$ consecutive P-frames predicted from the same view.

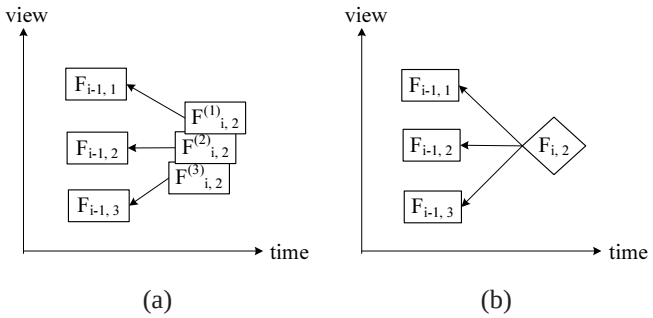


Fig. 4. Examples of (a) redundant P-frames and (b) M-frame.

I-frame is used for random access. For view-switching, either redundant P-frames or M-frames are used. Redundant P-frames mean one differentially coded P-frame is constructed for each potential predictor (last frame in a decoding path from which a view-switch is possible). M-frame, on the other hand, has a single frame representation for multiple potential predictors; reconstruction property of M-frame guarantees that the exact same frame can be correctly decoded no matter which one of a set of predictor frames known at encoding time is actually available at the decoder's buffer at stream time. Redundant P-frames offer the lowest transmission rate possible while increasing the storage required as the number of decoding paths multiplies over time. An M-frame has a single frame representation and hence smaller storage, but at a higher transmission rate than P-frame.

Fig. 4 shows an example tradeoff between transmission rate and storage for redundant P-frames and M-frame from three different predictors. The redundant P-frames in Fig. 4(a) need three different coded versions of picture $F^o_{i,2}$, one for each of three different predictors $F_{i-1,1}$, $F_{i-1,2}$ and $F_{i-1,3}$, whereas in Fig. 4(b) only one M-frame is needed to get the same coded version no matter which of the three predictors is available at the decoder.

An M-frame can be implemented using one of many available coding techniques such as SP-frames in H.264 [18] and DSC frames [19], [20]. In this paper, we implement an M-frame using DSC [20], due to its demonstrably superior coding performance over SP-frames. We overview the encoding of a DSC frame as follows. First, motion information from each of the predictor frames is encoded. Then, transform coefficients of the motion residuals in Discrete Cosine Transform (DCT) domain from each prediction are compared. Because most significant bits (MSB) of the transform coefficients are likely to be the same for all residuals, only the least significant bit (LSB) bit-planes that are different among the residuals require encoding. In particular, given the target is the I-frame, the LSB difference between each residual and the target is interpreted as channel noise, and channel coding (such as low-density parity check codes (LDPC) used in [19], [20]) of sufficient strength is employed so that the largest noise in all residuals can be removed. By encoding multiple motion information and LDPC codes for LSB bit-planes, the exact same frame can be recovered no matter which predictor frame is available at decoder's buffer. By exploiting correlation between predictor frames and the target, DSC frame has much smaller size than the independently coded I-frame.

C. Depth-image-based Rendering

Depth-image-based rendering (DIBR) is the process of synthesizing novel intermediate virtual views of a 3D scene from the texture and depth maps of neighboring anchor viewpoints. DIBR-based view synthesis can be implemented as follows. First, the original texture pixels of one anchor view are projected into the 3D space, using the associated depth map. Then, those 3D points are re-projected into the image plane of the virtual view. This concatenation of 2D-to-3D projection and 3D-to-2D projection is usually called *3D warping* [24]. Since the number of disoccluded pixels in a virtual view synthesized using texture and depth map from one single viewpoint is large, texture and depth maps of two adjacent views are often used for DIBR [12]. If two texture pixels from left and right anchor viewpoints map to the same virtual view pixel, pixel blending is performed, where the weights for the left and right corresponding pixels are inversely proportional to the distance from the virtual viewpoint to the left and right anchor viewpoints. It is possible that no texture pixels from either the left or right anchor viewpoints map to a particular virtual view pixel due to occlusion. In this case, the missing pixels are usually filled by image inpainting methods from neighboring projected pixels [22]. In general, large distance between the left and right anchor views could increase the number of disoccluded pixels in the virtual view, leading to worse DIBR-based view synthesis quality [12].

Note that though the DIBR view synthesis tool adopted in the paper interpolates an intermediate image using only texture and depth maps of two neighboring coded views, it is also possible to use texture and depth maps of other time instants and other views to impose time and view consistency in view generation as done in [25]. Our proposed coding structure optimization can easily be adapted to a more advanced view synthesis tool, however, and hence our use of a simple DIBR tool is sufficient to illustrate our core contribution of coding structure optimization.

D. Probabilistic View-switching Model

Without loss of generality, we first denote K evenly spaced captured views by $1, \ldots, K$. Between every pair of adjacent captured views i and $i+1$, we in addition define a set of K' evenly spaced virtual view positions that can also be requested by clients, *i.e.*, $i + \frac{j}{K'+1}$, $j \in \{1, \ldots, K'\}$, separated by *view spacing* $d = 1/(K'+1)$. The total number of views available for client's selection is hence expanded to $K'(K-1)+K$. Fig. 5 shows an example of multiview sequence where $K = 4$ and $K' = 1$ ($d = 0.5$). Note that all available discrete *view-switch positions* (virtual and captured) available for client's selection are multiples of d. In the sequel, we will say that a view-switch position $v = kd$, $k \in \mathcal{Z}^+$, $1/d \leq k \leq K/d$, has *view coordinate* k, where k is view-switch position v expressed in multiples of view spacing d.

We design a view-switch model to allow a client to periodically request a view-switch every Δ frames from view-switch position v to another view-switch position v', where the difference $|v' - v|$ is no larger than Ld, $L \in \mathcal{Z}^+$, where the pre-defined *View difference bound* L limits the speed of view transition.

To optimize multiview video frame structure at encoding time without knowledge of clients' eventual chosen view trajectories at stream time, we propose the following probabilistic model to capture the view-switching trend of a typical client. Suppose a client is watching view coordinate k at view-switch instant $i\Delta$, *after* watching view

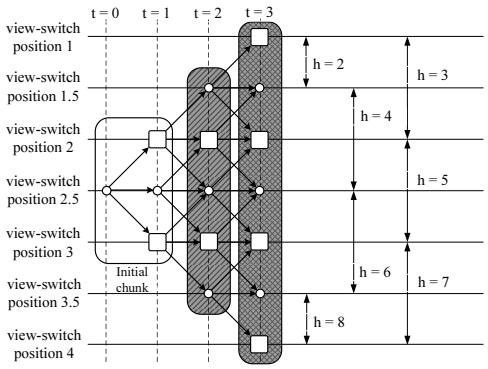


Fig. 5. Example of progressive view-switch for $K = 4$ captured views (rectangles) with $K' = 1$ intermediate view (circles) between two captured views ($d = 0.5$), view difference bound $L = 1$, initial view $v^0 = 2.5$, view-switching period $\Delta = 1$ and $RTT = \Delta - \epsilon$. View-switch positions in the shadeless box and shaded boxes with different patterns represent the ones covered by the initial chunk and structure slices at time 2 and 3 respectively. Each double-end arrow delimits the range of possible view-switches covered by one structure slice after receiving a view-switch coordinate feedback h from client.

coordinate k' at instant $(i-1)\Delta$. The probability[4] that she will select view coordinate l at instant $(i+1)\Delta$ is $\Omega_{k',k}(l)$, $l \in \{\max(1/d, k-L), \ldots, \min(K/d, k+L)\}$:

$$\Omega_{k',k}(l) = \begin{cases} \Phi(l-(2k-k')), & \max(1/d, k-L) < l \\ & < \min(K/d, k+L) \\ \sum_{n=l}^{\infty} \Phi(n-(2k-k')), & l = \min(K/d, k+L) \\ \sum_{n=-\infty}^{l} \Phi(n-(2k-k')), & l = \max(1/d, k-L) \end{cases} \quad (1)$$

where $\Phi(n)$ is a symmetric *view-switching probability function* centered at zero; see Fig. 6(a) for an example. In words, (1) states that the probability $\Omega_{k',k}(l)$ that a client selects view coordinate l depends on both the current view coordinate k *and* previous selected coordinate k'; the probability is the highest at position $k+(k-k')$ where the client continues in view-switch direction $k-k'$. If l is a boundary coordinate, 1 or K, or at the view difference bound $k \pm L$, then the probability $\Omega_{k',k}(l)$ needs to sum over probabilities in view-switching probability function $\Phi(n-(2k-k'))$ that fall outside the feasible views as well, as shown in Fig. 6(b), where the right-most boundary view is requested given $K = 3$ and $K' = 1$, *i.e.*, $l = 3/d$.

E. Network Delay Model

Round trip time (RTT) delay is the time required for a packet to travel from a client to the server and back. In our IMVS scenario, RTT delay represents the minimum server-client interaction delay experienced by a client from the time she sends a view-switch request, to the time the effected video due to the request is received. Here, we assume there are different RTTs between the video server and different clients, though RTT of each server-client

[4]Given all available view-switch positions (captured and virtual) for client's selection are integer multiples of view spacing d, we can define the view-switch probability function $\Omega_{k',k}(l)$ in discrete domain, where k', k and l are all view coordinates.

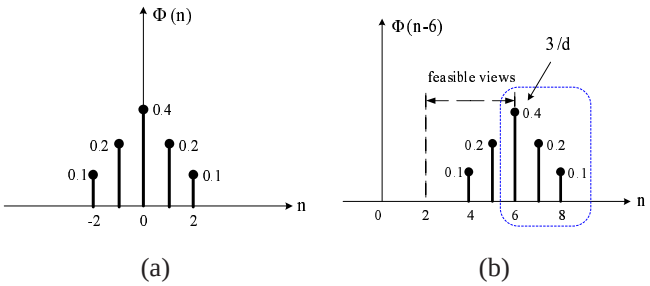


Fig. 6. Example of view-switch probability function for $K = 3$ captured views with $K' = 1$ intermediate view between two neighboring captured views ($d = 0.5$). (a) original $\Phi(n)$; (b) shifted function $\Phi(n-6)$.

pair remains constant (each of server-to-client and client-to-server transmission takes exactly half of RTT) once video streaming starts. In addition, we assume all RTTs do not exceed an upper-bound $RTT_{\max}$. There is much work in the literature in estimating RTT in typical packet-switched networks [26], [27], but is outside the scope of this paper. We will simply assume the probability density function (PDF) of RTT, $\psi(x)$, is known *a priori* at video encoding time.

IV. Problem Formulation

Having described the functionalities of the new IMVS system and models in Sec. III, we now formulate the IMVS problem as an optimization problem: given pre-defined storage and distortion constraints, design an optimal frame structure and associated transmission schedule, and select optimal QPs for texture and depth map coding, that minimize the expected server transmission rate, while providing clients with zero-delay view-switching interactivity in IMVS. In Sec. IV-A, we first develop a network-delay-cognizant transmission protocol for transmitting frames in a coding structure for IMVS, so that each client can enjoy zero-delay view-switching given her unique server-client RTT. We then provide definitions of optimization variables, search space, constraints and objective in Sec. IV-B. Finally, we formally define the IMVS optimization problem in Sec. IV-C.

A. Network-delay-cognizant Transmission Protocol

Previous IMVS works [5], [6], [7] do not properly address the problem of network delay; hence a view-switch request from a client will suffer at least one RTT delay in addition to the system's inherent Δ-frame view-switch interval[5]. In this section, we develop a transmission protocol for network-delay-cognizant view-switching, so that a client can play back the video in time and perceive no *additional* view-switching delay (beyond the system's Δ-frame view-switch interval), even when RTT is non-negligible. The key idea is to send additional data to cover all possible view-switch positions to be requested by a client one RTT into the future beyond the requested view.

[5]View-switch interval Δ for IMVS systems can be set very small (on the order of every 3 to 5 frames), and hence an additional RTT delay on the Internet of up to hundreds of milliseconds can be detrimental to the interactive multiview video experience.

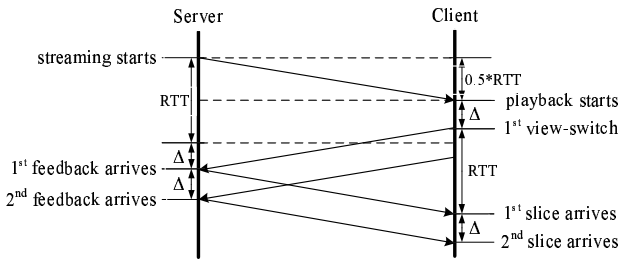


Fig. 7. Timing diagram during server-client communication.

TABLE I

SUMMARY OF NOTATIONS FOR IMVS PROBLEM FORMULATION

Notations	Description
K, K'	number of captured views, number of virtual views between two neighboring captured views.
L, v^o	view difference bound, starting view-switch position.
RTT, δ	RTT delay, number of view-switches that a structure slice covers into future after receiving a client feedback.
$\mathcal{T}$, $\mathbf{Q}=[Q_t,Q_d]^T$	frame structure, QPs for encoding texture and depth map.
$B(\mathcal{T},\mathbf{Q})$, $C(\mathcal{T},\mathbf{Q})$, $D(\mathbf{Q})$	storage cost, transmission cost, distortion cost of a frame structure $\mathcal{T}$ and QPs $\mathbf{Q}$.
$F^o_{i,j}$, $F_{i,j}$	original picture, coded frame of view-switch instant $i\Delta$ and view j.
$I_{i,j}$, $P_{i,j}$, $M_{i,j}$	I-frame, P-frame, M-frame of view-switch instant $i\Delta$ and view j.
$\Xi_i(\delta)$, $c(\Xi_i(\delta))$	the set of frames, the center view coordinate for decoding at view-switch instant $i\Delta$.
$p(\Xi_i(\delta))$, $q(F_{i,j},\delta)$	transmission probability of a slice $\Xi_i(\delta)$, a frame $F_{i,j}$, given δ.
$\psi(x)$, $\Psi(\delta)$	probability density function of RTT, probability mass function of δ.
$t^\delta(\mathcal{T},G(\delta),\mathbf{Q})$	transmission rate of a frame structure $\mathcal{T}$ and QPs $\mathbf{Q}$, given schedule $G(\delta)$.
$D^c_j(Q_t)$, $D^s_k(\mathbf{Q})$	average distortion of frames of captured view j, virtual view k, given QPs $\mathbf{Q}$.

Following the illustration in Fig. 7, we first discuss timing events during server-client communication in IMVS system assuming constant transmission delay (as discussed in Sec. III-E). The server first transmits an *initial chunk* of coded multiview data to the client, arriving at the client $\frac{1}{2}RTT$ time later. Upon receipt of the initial chuck at time 0, the client starts playback, and makes her first view-switch Δ-frame time later. Her first view-switch decision (feedback) is transmitted immediately after the view-switch, and arrives at server at time $\frac{1}{2}RTT+\Delta$. Responding to the client's first feedback, server immediately sends a *structure slice*, arriving at the client $\frac{1}{2}RTT$ time later, or RTT time after the client transmitted her feedback. More generally then, the client sends feedbacks in interval of Δ-frame time, and in response, server sends a structure slice corresponding to each received feedback every Δ-frame time. We assume there are no packet losses during packet transmission.

Notice that from the time the client starts playback to the time the first structure slice is received from server, $\Delta+RTT$ time has elapsed. Therefore, before the arrival of the first structure slice, the number of view-switches, δ, that the initial chunk must enable is

$$\delta=\left\lfloor\frac{\Delta+RTT}{\Delta}\right\rfloor \tag{2}$$

For simplicity, we assume that IMVS session starts from a known initial position v^o with view coordinate k^o, *i.e.*,

$v^o = k^o d$ and $k^o \in \mathcal{Z}^+$, $1/d \leq k^o \leq K/d$. Given each subsequent view-switch can maximally alter view coordinate by $\pm L$, initial chunk must contain data enabling view-switches to view coordinates $\mathcal{V}_i = \{k \mid \max(1/d, k^o - iL) \leq k \leq \min(K/d, k^o + iL), k \in \mathcal{Z}^+\}$ at view-switch instants $(i\Delta)$'s, where $0 \leq i \leq \delta$.

Because subsequent structure slices arrive every Δ-frame time, each structure slice only needs to enable one more view-switch for the client to continue video playback in time and enjoy zero-delay view-switching. Notice that because each structure slice arrives at the client RTT time after the client sent her view-switch feedback, the view-switch enabled by the structure slice corresponding to the client's feedback sent at instant $t = i\Delta$ is the first view-switch *after* time $t + RTT$, *i.e.*, view-switch at instant $(i + \delta)\Delta$. In other words, given client's view coordinate selection h at instant $i\Delta$, the *span* of view-switch coordinates $\mathcal{V}_{i+\delta}$ that a structure slice must cover for the view-switches at instant $(i + \delta)\Delta$, is $\mathcal{V}_{i+\delta} = \{k \mid \max(1/d, h - \delta L) \leq k \leq \min(K/d, h + \delta L), k \in \mathcal{Z}^+\}$.

This protocol—transmitting multiple views for the sake of client's selection of a single view in the future—is in stark contrast with the protocol in [5], [6], [7], where only one single view is transmitted corresponding to each client's request. Fig. 5 illustrates a view-switching example for $K = 4$, $K' = 1$, $L = 1$, $v^o = 2.5$, $\Delta = 1$ and $RTT = \Delta - \epsilon$ for small $\epsilon > 0$. The initial chunk contains only enough multiview data to enable $\delta = 1$ view-switch, spanning view-switch coordinates $\mathcal{V}_1 = \{4, 5, 6\}$. If the client first selects view-switch coordinate $h = 4$ at time 1, then the first structure slice must span view-switch coordinates $\mathcal{V}_{1+\delta} = \{3, 4, 5\}$. Instead, if the client first selects view-switch coordinate $h = 6$, then the corresponding slice must span view-switch coordinates $\mathcal{V}_{1+\delta} = \{5, 6, 7\}$.

B. Definitions for IMVS Optimization

Before formally defining the IMVS optimization problem, we first define optimization variables (frame structure, associated transmission schedules and QPs), and storage, transmission and distortion costs corresponding to a set of variables. See Table I for a summary of notations.

1) Redundant Frame Structure: One can construct a *redundant frame structure* $\mathcal{T}$, comprised of I-, P- and M-frames, denoted as $I_{i,j}$'s, $P_{i,j}$'s and $M_{i,j}$'s respectively, to represent the captured multiview video frames at view-switch instant $i\Delta$'s and view j's for IMVS. Each frame not located at view-switch instants (not shown in our graphical model) is a P-frame predicted from a frame of the same view and previous time instant. Note that while we already discussed one concrete DSC implementation of M-frame in Sec. III-B, our abstraction and subsequent optimization can apply more generally to any implementation of M-frame. Fig. 8 shows one example frame structure for multiview sequence in Fig. 5.

A frame structure $\mathcal{T}$ forms a *directed acyclic graph* (DAG) starting with an I-frame if initial view-switch position is a captured view, or an I-frame and a P-frame predicted from the I-frame if initial view-switch position is a virtual view, as starting nodes. In Fig. 8, I-frames $I_{0,2}$ and P-frame $P_{0,3}$ are two starting nodes of structure $\mathcal{T}$ for synthesizing virtual view 2.5. $\mathcal{T}$ is "redundant" in the sense that an original picture $F^o_{i,j}$ can be represented by more than one frame $F_{i,j}$. In Fig. 8, original picture $F^o_{3,4}$ is represented by two P-frames, $P^{(1)}_{3,4}$ and $P^{(2)}_{3,4}$, each encoded using a different predictor, $P_{2,4}$ and $P_{3,3}$, respectively. Depending on which predictor is available at decoder during stream time, different coded frames $F_{i,j}$'s can be transmitted to enable correct decoding and (slightly different)

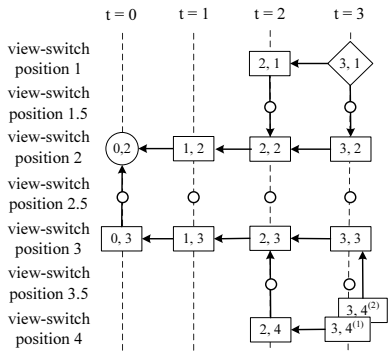


Fig. 8. Example frame structure for $K = 4$ captured views with $K' = 1$ intermediate view (small circles) between two neighboring captured views (view spacing $d = 0.5$), view difference bound $L = 1$, initial view $v^0 = 2.5$, view-switching period $\Delta = 1$ and $RTT = \Delta - \epsilon$. I-, P- and DSC-frames are represented by large circles, rectangles and diamonds, respectively.

reconstruction of original picture $F^o_{i,j}$. This is done to lower transmission rate by exploiting correlation between the requested picture and frames in the decoder buffer, and to avoid coding drift [5].

2) Structure Slice: As discussed in Sec. IV-A, depending on the view-switch coordinate h selected by client at view-switch instant $(i-\delta)\Delta$, a set of frames of different captured viewpoints will be transmitted for possible decoding at view-switch instant $i\Delta$. Given δ, we define *structure slice* $\Xi_i(\delta)$, with *center coordinate* $c(\Xi_i(\delta))$, as a set of frames to enable selection of view-switch coordinates in span $\{k \mid \max(1/d, c(\Xi_i(\delta)) - \delta L) \leq k \leq \min(K/d, c(\Xi_i(\delta)) + \delta L), k \in \mathcal{Z}^+\}$ at view-switch instant $i\Delta$. Center coordinate $c(\Xi_i(\delta))$ is the client's selected view coordinate h at view-switch instant $(i-\delta)\Delta$.

Consider the example in Fig. 8, where initial chunk contains frames $I_{0,2}$, $P_{0,3}$, $P_{1,2}$ and $P_{1,3}$ to cover view-switches to positions 2, 2.5 and 3 at time 1. If the client selects view-switch coordinate $h = 4$ (view 2) at time 1, then the corresponding structure slice transmitted is $\Xi^{(1)}_2(1) = \{P_{2,1}, P_{2,2}, P_{2,3}\}$ with $c(\Xi^{(1)}_2(1)) = 4$, to cover possible view-switches to positions 1.5, 2 and 2.5 at time 2. Instead, if client remains in coordinate $h = 5$ (view 2.5) at time 1, then the structure slice $\Xi^{(2)}_2(1) = \{P_{2,2}, P_{2,3}\}$ will be sent to decoder with $c(\Xi^{(2)}_2(1)) = 5$, for the possible switches to positions 2, 2.5 and 3 at time 2. Notice that different slices can contain the same frames, and can also contain different number of frames.

3) Transmission Schedule: Which slice $\Xi_i(\delta)$ of structure $\mathcal{T}$ is transmitted for view-switch instant $i\Delta$ depends on slice $\Xi_{i-1}(\delta)$ transmitted previously (for differential coding), and client's selected view-switch coordinate h at view-switch instant $(i-\delta)\Delta$. We can formalize the association among $\Xi_{i-1}(\delta)$, h and $\Xi_i(\delta)$ via a *transmission schedule* $G(\delta)$. More precisely, $G(\delta)$ dictates which structure slice $\Xi_i(\delta)$ will be transmitted for client's selection at view-switch instant $i\Delta$, given previous transmitted slice $\Xi_{i-1}(\delta)$ and client's selected view-switch coordinate h at view-switch instant $(i-\delta)\Delta$:

$$G(\delta) : (\Xi_{i-1}(\delta), h) \Rightarrow \Xi_i(\delta), \quad \begin{array}{l} \max(1/d, c(\Xi_{i-1}(\delta)) - L) \leq h \\ \leq \min(K/d, c(\Xi_{i-1}(\delta)) + L) \end{array} \tag{3}$$

where center coordinate of $\Xi_i(\delta)$ is $c(\Xi_i(\delta)) = h$. In what follows, we denote a scheduled transmission from slice $\Xi_{i-1}(\delta)$ to slice $\Xi_i(\delta)$, with client's selected view-switch coordinate h at instant $(i-\delta)\Delta$, as $(\Xi_{i-1}(\delta), h) \overset{G(\delta)}{\Rightarrow} \Xi_i(\delta)$.

Note that for a given structure $\mathcal{T}$ and slice $\Xi_{i-1}(\delta)$ available for decoding at view-switch instant $(i-1)\Delta$, if client selects view coordinate h at view-switch instant $(i-\delta)\Delta$, there may exist different decodable slices $\Xi_i(\delta)$'s, and hence different transmission schedules $G(\delta)$'s, that enable all reachable view-switch coordinates $\mathcal{V}_i$ at instant $i\Delta$. Our optimization will hence consider not just optimal structure $\mathcal{T}$, but also optimal schedule $G(\delta)$ for the chosen structure $\mathcal{T}$.

4) Feasible Structure Space: Based on the above discussion, we can define a *feasible frame structure* $\mathcal{T}$ given δ as one where every reachable view-switch coordinate, as constrained by the view-switching model (Sec. III-D), can be requested by a client every Δ-frame interval and be executed with zero-delay using $\mathcal{T}$. Mathematically, we say that $\mathcal{T}$ is feasible given δ if there exists at least one *feasible schedule* $G(\delta)$, such that each sequence of client's permissible selection of view-switch coordinates, $h_1, h_2, \ldots$, will lead to a corresponding scheduled transmission of decodable slices $\Xi_{i+\delta}(\delta), \Xi_{i+1+\delta}(\delta), \ldots$, such that center coordinate and view span of each slice $\Xi_{i+\delta}(\delta)$ are $c(\Xi_{i+\delta}(\delta)) = h_i$ and $\mathcal{V}_{i+\delta} = \{k \mid \max(1/d, h_i - \delta L) \leq k \leq \min(K/d, h_i + \delta L), k \in \mathcal{Z}^+\}$, respectively. Center coordinates and view spans of slices defined above ensure all reachable view-switch coordinates can be selected by client at instants $(i+\delta)\Delta$, $(i+1+\delta)\Delta$, etc.

More generally, RTT between server and client can take on different values resulting in different δ's. In what follows, we define *feasible space* Θ as the set of all feasible frame structures $\mathcal{T}$'s, where a feasible structure $\mathcal{T}$ is one where there exists at least one feasible schedule $G(\delta)$ for each possible δ.

5) Structure Slice Probability and Frame Transmission Probability: To properly define transmission cost, we first define *structure slice probability* $p(\Xi_i(\delta))$ as the probability that structure slice $\Xi_i(\delta)$ for decoding at instant $i\Delta$ is transmitted, given schedule $G(\delta)$. Considering the structure slices $\Xi_i(\delta)$'s in the initial chunk, where $0 \leq i \leq \delta$, are always sent to client, this probability could be computed recursively using view transition probability $\Omega_{k',k}(l)$:

$$p(\Xi_i(\delta)) = \begin{cases} 1, & 0 \leq i \leq \delta \\ \sum\limits_{\Xi_{i-1}(\delta) \in \mathcal{G}} p(\Xi_{i-1}(\delta)) \sum\limits_{c'} \Omega_{c', c(\Xi_{i-1}(\delta))}(c(\Xi_i(\delta))), & i > \delta \end{cases} \tag{4}$$

where $\mathcal{G} = \{\Xi_{i-1}(\delta) \mid (\Xi_{i-1}(\delta), c(\Xi_i(\delta))) \overset{G(\delta)}{\Rightarrow} \Xi_i(\delta)\}$. In words, (4) states that $p(\Xi_i(\delta))$ is the sum of probability of each slice $\Xi_{i-1}(\delta)$ switching to slice $\Xi_i(\delta)$, scaled by the slice probability of $\Xi_{i-1}(\delta)$ itself, $p(\Xi_{i-1}(\delta))$, given schedule $G(\delta)$ dictates slice transmission in frame structure $\mathcal{T}$.

Further, we define *frame transmission probability* $q(F_{i,j}, \delta)$ as the probability that a frame $F_{i,j}$ is transmitted from server to client, which can be calculated using the defined structure slice probability (4):

$$q(F_{i,j}, \delta) = \sum_{\Xi_i(\delta) | F_{i,j} \in \Xi_i(\delta)} p(\Xi_i(\delta)) \tag{5}$$

In words, the transmission probability of a frame $F_{i,j}$ is the sum of probabilities of slices $\Xi_i(\delta)$'s that include $F_{i,j}$.

6) Storage Cost: For a given frame structure $\mathcal{T}$ and the associated QPs for texture and depth images, Q_t and Q_d, we can define the corresponding *storage cost* by simply adding up the sizes of all the frames $F_{i,j}$'s in $\mathcal{T}$, *i.e.*,

$$B(\mathcal{T},\mathbf{Q}) = \sum_{F_{i,j}\in\mathcal{T}} |F_{i,j}(\mathbf{Q})| = \sum_{F_{i,j}\in\mathcal{T}} \Big(|F^t_{i,j}(Q_t)| + |F^d_{i,j}(Q_d)|\Big) \tag{6}$$

where $\mathbf{Q}$ is the pair of QPs for texture and depth maps $\mathbf{Q} = [Q_t, Q_d]^T$, $|F_{i,j}|$ is the size of frame $F_{i,j}$ which depends on the specific QPs $\mathbf{Q}$, $F^t_{i,j}$ and $F^d_{i,j}$ denote the texture and depth maps of frame $F_{i,j}$, respectively.

7) Transmission Cost: Given a frame structure $\mathcal{T}$ and the associated QPs $\mathbf{Q}$, we can define the corresponding *transmission cost*. First, given the relationship between δ and RTT in (2), one can see that the same transmission schedule $G(\delta)$ for a given frame structure $\mathcal{T}$ can be applicable to a range of RTT's, $(\delta-1)\Delta \le RTT < \delta\Delta$; *i.e.*, the same slice $\Xi_i(\delta)$ of structure $\mathcal{T}$ can be transmitted for view-switch instant $i\Delta$. Therefore, to facilitate the definition of transmission cost, we map the PDF of RTT, $\psi(x)$, into a discrete probability mass function (PMF) of an integer number δ of view-switch interval Δ, $\Psi(\delta)$, by integrating $\psi(x)$ over the range $[(\delta-1)\Delta, \delta\Delta)$:

$$\Psi(\delta) = \int_{(\delta-1)\Delta}^{\delta\Delta} \psi(x), \qquad 1 \le \delta \le \delta_{\max} \tag{7}$$

Where $\delta_{\max} = \lfloor(\Delta + RTT_{\max})/\Delta\rfloor$. Then, given schedules $G(\delta)$'s for possible δ's, transmission cost $C(\mathcal{T}, G(), \mathbf{Q})$ of a frame structure $\mathcal{T}$ associated with QPs $\mathbf{Q}$ is defined as the expected transmission cost, *i.e.*,

$$C(\mathcal{T}, G(), \mathbf{Q}) = \sum_{\delta=1}^{\delta_{\max}} \Psi(\delta) \;\; t_\delta(\mathcal{T}, G(\delta), \mathbf{Q}) \tag{8}$$

where $G()$ denotes the set of schedules $G(\delta)$'s for all δ's.

For a given schedule $G(\delta)$, individual transmission cost $t_\delta(\mathcal{T}, G(\delta), \mathbf{Q})$ of structure $\mathcal{T}$ and QPs $\mathbf{Q}$ depends on view transition probability $\Omega_{k',k}(l)$, which can be calculated by adding up the sizes of all frames $F_{i,j}$'s in $\mathcal{T}$, scaled by the corresponding frame transmission probability (5):

$$t_\delta(\mathcal{T}, G(\delta), \mathbf{Q}) = \sum_{F_{i,j}\in\mathcal{T}} q(F_{i,j}, \delta)\ |F_{i,j}(\mathbf{Q})| \tag{9}$$

8) Distortion Cost: Since clients can request captured or synthesized views for observation, we define *distortion cost* as the average distortion of all captured and synthesized views available in the system. For distortion of a picture in a captured view, we use the Mean-Squared-Error (MSE) between the original and coded versions of the texture maps of the picture. On the other hand, since no captured image is available for a virtual view, we synthesize an image using the uncompressed textures and the depth images of neighboring captured views as reference to calculate its MSE. We denote Λ as the discrete set of available QPs for texture and depth coding.

Notice that the distortion of both captured views and virtual views are mainly influenced by the chosen QPs $\mathbf{Q}$, and independent of a particular frame structure $\mathcal{T}$. For example, in Fig. 8, captured view 4 at time 3 can be reconstructed with roughly the same distortion using either a P-frame $P^{(1)}_{3,4}$ or $P^{(2)}_{3,4}$. Let $D^c_j(Q_t)$ be the average distortion of frames at all view-switch instants of captured view j given the texture QP Q_t, and $D^s_k(\mathbf{Q})$ be the average distortion of synthesized frames at all view-switch instants of virtual view k given the texture/depth QPs

$\mathbf{Q}$ used for both neighboring captured views. Distortion cost $D(\mathbf{Q})$ is then given by

$$D(\mathbf{Q}) = \frac{1}{(K-1)K' + K} \left(\sum_{j=1}^{K} D_j^c(Q_t) + \sum_{j=1}^{K-1} \sum_{k=1}^{K'} D_{j+\frac{k}{K'+1}}^s(\mathbf{Q}) \right) \tag{10}$$

From (6) and (10), it can be seen that coarse QPs $\mathbf{Q}$ result in smaller frame size of texture and depth coding, $|F_{i,j}^t(Q_t)|$ and $|F_{i,j}^d(Q_d)|$, and larger distortion $D(\mathbf{Q})$. This means that given a storage constraint, a frame structure can afford more redundant representations of one picture using redundant P-frames to lower transmission rate, at the expense of sacrificed visual quality. Alternatively, finer QPs $\mathbf{Q}$ can lower the distortion, but the increased frame size will lead to less redundancy (more M-frames) used in a frame structure, resulting in larger transmission rate.

C. Optimization Definition

We can now formally define our IMVS problem as a combinatorial optimization problem as follows.

Problem Definition 4.1: Given a number of captured views, the IMVS optimization problem is to find a structure $\mathcal{T}$ using a combination of I-, P- and M-frames, and associated schedules $G(\delta)$'s for possible δ's, as well as texture/depth QPs $\mathbf{Q}$, that minimize the transmission cost $C(\mathcal{T}, G(), \mathbf{Q})$ while both a storage constraint $\bar{B}$ and a distortion constraint $\bar{D}$ are observed. Mathematically, this optimization problem is given by:

$$\begin{array}{ll} \min\limits_{\mathcal{T}\in\Theta, G(), \mathbf{Q}\in\Lambda} & C(\mathcal{T}, G(), \mathbf{Q}) \\ s.t. & B(\mathcal{T}, \mathbf{Q}) \leq \bar{B}, \quad D(\mathbf{Q}) \leq \bar{D} \end{array} \tag{11}$$

It is instructive to compare our new joint optimization formulation with that in [7]. On one hand, the objective of both formulations is to minimize transmission rate subject to a storage constraint. On the other hand, our formulation is different from [7] in two respects. First, to enable IMVS with free viewpoint synthesis, our joint optimization considers the optimal bit rate allocation between texture and depth maps since both types of maps need to be transmitted for view synthesis at decoder, while [7] considers coding of texture maps only. Correspondingly, an additional distortion constraint is considered in our formulation to identify the optimal texture/depth QPs. Second, in our formulation, we consider structure optimization for variable network delays. As we will see in Sec. V-B, it turns out that different delays contribute to scheduled transmission of different coded frames. This is in contrast to structure optimization in [7] where only one logical schedule exists for a given structure.

V. Algorithm Development

In this section, we develop algorithms to select a good frame structure, associated transmission schedules, and texture/depth QPs for the IMVS optimization problem in (11). We first propose an iterative procedure by alternately optimizing structure $\mathcal{T}$ and associated schedule set $G()$ only, then QPs $\mathbf{Q}$ only, while keeping the other set of variables fixed. We then present a greedy algorithm to optimize a frame structure $\mathcal{T}$ and schedule set $G()$ given QPs $\mathbf{Q}$. Finally, we present a low-complexity algorithm to update QPs $\mathbf{Q}$ for a given frame structure $\mathcal{T}$ and schedule set $G()$.

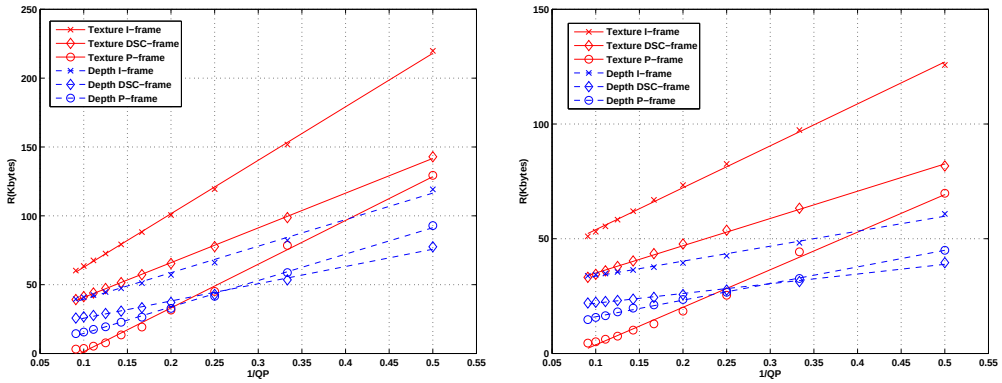


Fig. 9. Relationship between R and $1/QP$ of I-, P- and DSC-frames, for the texture and depth coding of sequence (a) `Dog` and (b) `Pantomime`.

A. Two Sub-Problems

To simplify the optimization, we divide the overall IMVS optimization problem into two simpler sub-problems, optimizing one set of variables while keeping the other set fixed. We formalize the definitions of the two sub-problems as follows.

Problem Definition 5.1: Given chosen texture/depth map QPs $\mathbf{Q}^{(k)}$ at iteration k that satisfy distortion constraint $\bar{D}$, the IMVS optimization problem degenerates to *sub-problem one*: find structure $\mathcal{T}$ and associated schedule set $G()$ to minimize transmission cost $C(\mathcal{T}, G(), \mathbf{Q}^{(k)})$, subject to storage constraint $\bar{B}$, *i.e.*,

$$\begin{aligned} \min_{\mathcal{T}\in\Theta, G()} \quad & C(\mathcal{T}, G(), \mathbf{Q}^{(k)}) \\ s.t. \quad & B(\mathcal{T}, \mathbf{Q}^{(k)}) \leq \bar{B} \end{aligned} \tag{12}$$

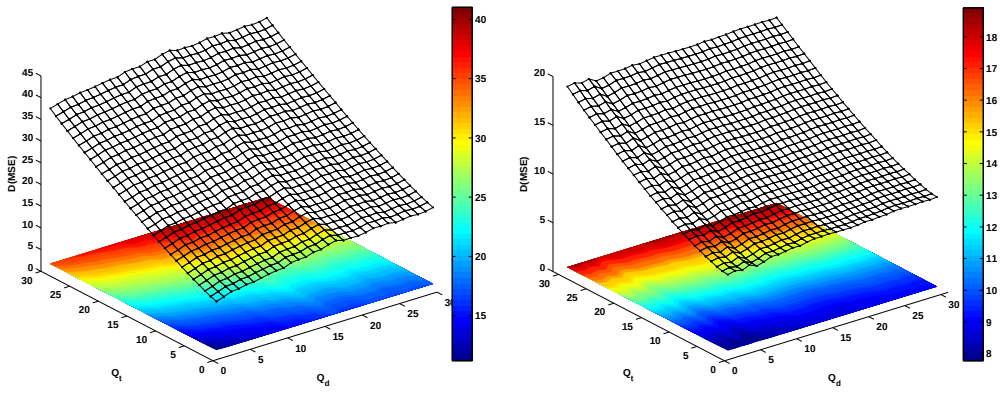
Notice that since the quality of view synthesis depends only on QPs $\mathbf{Q}^{(k)}$ and not on the particular chosen structure $\mathcal{T}$, the distortion constraint can be ignored in this sub-problem.

Problem Definition 5.2: Given a fixed frame structure $\mathcal{T}^{(k)}$ and associated schedule set $G^{(k)}()$ at iteration k, the IMVS optimization problem degenerates to *sub-problem two*: find QPs $\mathbf{Q}$ for texture and depth coding, such that the expected transmission cost $C(\mathcal{T}^{(k)}, G^{(k)}(), \mathbf{Q})$ is minimized while observing both the storage constraint $\bar{B}$ and the distortion constraint $\bar{D}$:

$$\begin{aligned} \min_{\mathbf{Q}\in\Lambda} \quad & C(\mathcal{T}^{(k)}, G^{(k)}(), \mathbf{Q}) \\ s.t. \quad & B(\mathcal{T}^{(k)}, \mathbf{Q}) \leq \bar{B}, \quad D(\mathbf{Q}) \leq \bar{D} \end{aligned} \tag{13}$$

Based on the two sub-problems, we can summarize the iterative procedure as:

1) Initialize a pair of texture/depth QPs $\mathbf{Q}^{(0)}$ satisfying the distortion constraint $\bar{D}$, and set $k = 0$.
2) Fix $\mathbf{Q}^{(k)}$, and optimize structure $\mathcal{T}$ and associated schedule set $G()$ to minimize the transmission rate in sub-problem one (12). For $k > 0$, stop if the pre-defined convergence criterion is satisfied.
3) Fix $\mathcal{T}^{(k)}$ and $G^{(k)}()$, and find $\mathbf{Q}$ that minimizes the transmission rate in sub-problem two (13).
4) Go to step 2 and set $k \leftarrow k + 1$.



(a) (b)

Fig. 10. Relationship between distortion D and quantization parameters (Q_t, Q_d) of sequence (a) `Dog` and (b) `Pantomime`.

As we can see above, the crux of the iterative procedure is to solve the two sub-problems separately. In the following, we will propose a greedy frame structure and schedule optimization algorithm and a QP update algorithm, both with low complexity, to separately address the two sub-problems.

B. Frame Structure & Schedule Optimization

We first present a frame structure and schedule optimization algorithm given fixed QPs $\mathbf{Q}^{(k)}$. Though (12) differs in some respects from the frame structure optimization problem in [21], a similar proof can be easily constructed to show that sub-problem one is also NP-hard. Given the computational complexity of (12), we first convert the storage-constrained problem (12) into the following unconstrained problem:

$$\begin{aligned} \min_{\mathcal{T}\in\Theta, G()} \quad & J(\mathcal{T}, G(), \mathbf{Q}^{(k)}) = C(\mathcal{T}, G(), \mathbf{Q}^{(k)}) + \lambda B(\mathcal{T}, \mathbf{Q}^{(k)}) \\ & = \sum_{F_{i,j}\in\mathcal{T}} \left(\sum_{\delta} \Psi(\delta) q(F_{i,j}, \delta) + \lambda \right) |F_{i,j}(\mathbf{Q}^{(k)})| \end{aligned} \tag{14}$$

where the Lagrangian multiplier λ is a fixed parameter that represents the tradeoff between transmission rate and storage, and $J(\mathcal{T}, G(), \mathbf{Q}^{(k)})$ is the Lagrangian cost. To find the optimal λ that minimizes transmission cost while observing storage constraint in (12), a bisection-search method is used over a predefined range $[\lambda_{min}, \lambda_{max}]$.

To solve (14) efficiently for given λ, we use a greedy approach to find near-optimal frame structure and associated schedules. In a nutshell, we iteratively build one "structure layer" t_i and "schedule layer" $g_i()$ one view-switch instant at a time from front to back. Structure layer t_i is comprised of frames $F_{i,j}$'s of all captured views j's at instant $i\Delta$, and schedule layer $g_i()$ consists of local schedules $g_i(\delta)$'s for all possible δ's, each mapping a structure slice $\Xi_{i-1}(\delta)$ in structure layer t_{i-1} to a structure slice $\Xi_i(\delta)$ in t_i, given client's view-switching feedback at instant $(i-\delta)\Delta$. At each view-switch instant $i\Delta$, the key question is: given structure $\mathcal{T}_{i-1}$ and schedule set $G_{i-1}()$ constructed up to instant $(i-1)\Delta$, how to optimally construct structure layer t_i and schedule layer $g_i()$ to minimize (14)?

To construct locally optimal structure layer t_i at view-switch instant $i\Delta$, we first initialize structure layer t_i^0 with K M-frames, one for each of K captured views. More precisely, for each M-frame $M_{i,j}$ of captured view j, we

assign all frames $F_{i-1,k}$'s in structure layer t_{i-1} of instant $(i-1)\Delta$ that can switch to view j at instant $i\Delta$, as predictors of $M_{i,j}$. Since an M-frame is not a redundant representation (one frame per captured picture), the initial structure layer has minimum storage of all possible layers.

Corresponding to initial structure layer t_i^0, we construct initial schedule $g_i^0(\delta)$ given δ as follows. We first designate structure slices $\Xi_i(\delta)$'s using created M-frames, where each slice $\Xi_i(\delta)$ of center coordinate $c(\Xi_i(\delta)) = h$ has enough M-frames to enable view-switches to coordinates $\mathcal{V}_i = \{k \mid \max(1/d, h-\delta L) \leq k \leq \min(K/d, h+\delta L), k \in \mathcal{Z}^+\}$. Then, given client's view coordinate selection h at instant $(i-\delta)\Delta$, an initial schedule $g_i^0(\delta)$ will map any previously designated structure slice $\Xi_{i-1}(\delta)$ in t_{i-1} with center coordinate $c(\Xi_{i-1}(\delta)) = h'$, $\max(1/d, h-L) \leq h' \leq \min(K/d, h+L)$, to the same $\Xi_i(\delta)$, *i.e.*, $(\Xi_{i-1}(\delta), h) \overset{g_i^0(\delta)}{\Rightarrow} \Xi_i(\delta)$.

However, large M-frame sizes in initial structure layer t_i^0 lead to large transmission cost. To reduce transmission cost, we incrementally add the most "beneficial" redundant P-frames one at a time—beneficial meaning one that reduces the Lagrangian cost—thereby increasing storage. We terminate when no more beneficial redundant P-frames can be added.

In details, we describe the algorithm as follows. First, as initial structure layer t_i^0, we construct one M-frame for each captured view j at view-switch instant $i\Delta$. We then designate structure slices $\Xi_i(\delta)$'s and determine the corresponding schedules $g_i^0(\delta)$'s as described earlier, and compute the resulting local Lagrangian cost in (14). Given the initial structure and schedule layers, we improve t_i and $g_i()$ by iteratively making local structure augmentations: selecting one candidate from a set of structure augmentations at each iteration, which offers the largest decrease in local Lagrangian cost. The possible augmentations are:

- Add new P-frame $P_{i,j}$ to t_i, predicted from existing frame $F_{i,k}$ of neighboring view k of *same* instant $i\Delta$.
- Add a new P-frame $P_{i,j}$, predicted from an existing frame $F_{i-1,k}$ in t_{i-1} of the *previous* instant $(i-1)\Delta$.
- Select a different predictor $F_{i,k}$ of the *same* instant $i\Delta$ for an already constructed P-frame $P_{i,j}$ in t_i.

Notice that the last augmentation does not increase the number of representations of a given captured view, while each of the first two increases the number of frame representations by one P-frame.

Using constructed structure layer t_i^l in the l-th iteration, we build up the corresponding schedule $g_i^l(\delta)$ given δ by minimizing transmission rate. More specifically, given a client's selected view coordinate h at view-switch instant $(i-\delta)\Delta$ and a structure slice $\Xi_{i-1}(\delta)$ in t_{i-1} with center coordinate $c(\Xi_{i-1}(\delta)) = h'$, $\max(1/d, h-L) \leq h' \leq \min(K/d, h+L)$, we designate structure slice $\Xi_i(\delta)$ by finding the set of frames $F_{i,j}$'s in t_i^l, which possesses the smallest size of transmitted frames while enabling all reachable coordinates $\mathcal{V}_i = \{k \mid \max(1/d, h-\delta L) \leq k \leq \min(K/d, h+\delta L), k \in \mathcal{Z}^+\}$ at view-switch instant $i\Delta$. This can be mathematically expressed as

$$g_i^l(\delta) : \min_{\Xi_i(\delta) \in \mathcal{G}'} \sum_{F_{i,j} \in \Xi_i(\delta)} |F_{i,j}| \tag{15}$$

where $\mathcal{G}' = \{\Xi_i(\delta) \mid (\Xi_{i-1}(\delta), h) \Rightarrow \Xi_i(\delta)\}$.

Note that different from frame optimization methods in [7] where there is only one logical schedule for a given structure due to the assumption of zero network delay, in the proposed greedy algorithm, we need to optimize

multiple schedules $g_i(\delta)$'s for different δ's in the schedule layer given a structure layer t_i at instant $i\Delta$, each corresponding to clients' view-switch feedbacks at different instants $((i-\delta)\Delta)$'s.

The above process repeats to find the most locally beneficial augmentation at each iteration, update the corresponding schedules by (15) and compute the local Lagrangian cost in (14), until no more Lagrangian cost reduction can be found. Note that after updating the local schedules at each iteration, it is possible that some frames in t_i are not used by any view-switch. In this case, those unused frames will be removed from the structure.

C. Optimal Quantization Parameters Update

We next present a low-complexity algorithm to optimally update QPs $\mathbf{Q}$ for given structure $\mathcal{T}^{(k)}$ and schedule set $G^{(k)}()$, as defined by the second sub-problem (13). To find the optimal solution of the constrained optimization problem (13), the naïve approach of exhaustively searching all candidates $\mathbf{Q}$'s that satisfy both storage constraint $\bar{B}$ and distortion constraint $\bar{D}$ is too expensive in practice. Instead, we develop a strategy to update QPs by first studying rate-quantization (R-Q) and distortion-quantization (D-Q) characteristics of multiview videos.

1) R-Q Model Analysis: During the last decades, the relationship between rate and QP of video coding has been extensively studied for applications such as rate control. Based on the experiments on a large number of multiview video sequences, we adopt the modified linear R-D model for H.263 in [28], where the rate R of a coded frame is modeled as $R(QP) = X/QP + L$. Here X is a constant and L is an offset indicating the overhead bits.

Fig. 9 shows the relationship between coded bits R and $1/QP$ of one I-, P- and M-frame, on the texture and depth coding respectively of the sequences `Dog` and `Pantomime`. As shown in Fig. 9, R is linearly correlated with $1/QP$ no matter if the frame in question is coded using I-, P, or M-frame. As a consequence, the storage cost $B(\mathcal{T}, \mathbf{Q})$ in (6) of a given frame structure $\mathcal{T}$ can be written as a function of QPs $\mathbf{Q}$ as:

$$B(\mathcal{T}, \mathbf{Q}) = \sum_{F_{i,j} \in \mathcal{T}} \left(\frac{X^t_{i,j}}{Q_t} + L^t_{i,j} + \frac{X^d_{i,j}}{Q_d} + L^d_{i,j} \right) = \frac{X_1}{Q_t} + \frac{X_2}{Q_d} + L \tag{16}$$

where $X^t_{i,j}$ and $L^t_{i,j}$, $X^d_{i,j}$ and $L^d_{i,j}$ are the individual parameters of texture and depth components for frame $F_{i,j}$, and $X_1 = \sum_{F_{i,j} \in \mathcal{T}} X^t_{i,j}$, $X_2 = \sum_{F_{i,j} \in \mathcal{T}} X^d_{i,j}$ and $L = \sum_{F_{i,j} \in \mathcal{T}} (L^t_{i,j} + L^d_{i,j})$ are the corresponding parameters of the overall structure $\mathcal{T}$. In our experiments, instead of calculating the specific parameters for each frame $F_{i,j}$, X_1, X_2 and L of a given structure $\mathcal{T}$ can be directly estimated from a number of available R-Q points (no fewer than 3) by the least-square solution of the following linear problem:

$$\mathbf{A}\,[X_1, X_2, L]^T = \mathbf{B} \tag{17}$$

where matrix $\mathbf{A}$ and column vector $\mathbf{B}$ are composed of row vectors $[1/Q_t, 1/Q_d, 1]$'s and the storages of each available R-Q point respectively.

2) D-Q Model Analysis: To the best of our knowledge, the relationship between view synthesis distortion and texture/depth QPs has not been formally studied in the literature. However, in our experiments, we observed that the distortion cost defined in (10) is roughly correlated with texture and depth QPs through a linear model, *i.e.*,

$$D(\mathbf{Q}) = Y_1 Q_t + Y_2 Q_d + Z \tag{18}$$

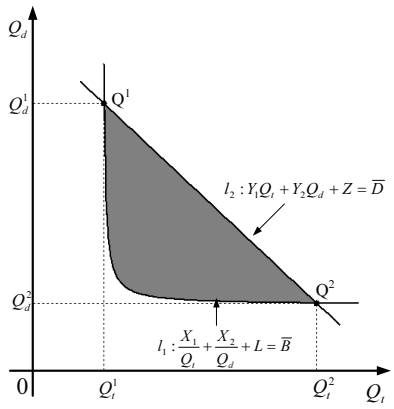


Fig. 11. Region of valid QP candidates for the second sub-problem.

where Y_1, Y_2 and Z are constants. Fig. 10 shows the relationship between average distortion D and QPs $\mathbf{Q}$ of the sequences `Dog` and `Pantomime`, where $K = 4$ and $K' = 4$.

3) Quantization Parameters Update: Based on the R-Q model (16) and D-Q model (18), given the storage and distortion constraints $\bar{B}$ and $\bar{D}$ in (13), the set of valid QPs (Q_t, Q_d)'s can be shown to be the shaded region in Fig. 11. In Fig. 11, l_1 and l_2 are two boundaries of the valid region, which are determined by the corresponding constraints $\bar{B}$ and $\bar{D}$ respectively, and $\mathbf{Q}^1 = [Q_t^1, Q_d^1]^T$ and $\mathbf{Q}^2 = [Q_t^2, Q_d^2]^T$ are two intersection points between l_1 and l_2, with $Q_t^1 < Q_t^2$ and $Q_d^2 < Q_d^1$.

We introduce the following lemma, which can lead to a closed-form solution to optimally update QPs in (13).

Lemma 5.1: Given a fixed structure $\mathcal{T}^{(k)}$ and schedule set $G^{(k)}()$ at the k-th iteration of the algorithm, the optimal QPs $\mathbf{Q}$ of sub-problem (13) is located at the boundary line l_2, corresponding to the distortion constraint $\bar{D}$.

Proof: The proof is given in Appendix I. ■

The conclusion of Lemma 5.1 suggests that we can now limit the search range of optimal QPs for given structure $\mathcal{T}^{(k)}$ and schedule set $G^{(k)}()$ to a line on which the distortions of all QPs are identically equal to the distortion constraint $\bar{D}$. Further, it turns out that with the help of Lemma 5.1, we can even derive a closed-form solution to update QPs of sub-problem two at each iteration, without any search process. More specifically, we first simplify sub-problem two (13) to a single-constraint problem, stated formally as a theorem below.

Theorem 5.1: Given structure $\mathcal{T}^{(k)}$ and schedule set $G^{(k)}()$, the optimization of transmission $C(\mathcal{T}^{(k)}, G^{(k)}(), \mathbf{Q})$ in terms of $\mathbf{Q}$, with storage constraint $\bar{B}$ and distortion constraint $\bar{D}$, is mathematically equivalent to the following univariate optimization problem:

$$\begin{aligned} \min_{Q_t} \quad & A_1/Q_t + A_2/(\bar{D} - Z - Y_1 Q_t) + S \\ s.t. \quad & Q_t^1 \le Q_t \le Q_t^2 \end{aligned} \tag{19}$$

TABLE II

PROCEDURES TO ITERATIVELY FIND OPTIMAL FRAME STRUCTURE, TRANSMISSION SCHEDULE AND QUANTIZATION PARAMETERS

1) Initialize texture/depth quantization parameters $\mathbf{Q}^{(0)}$ satisfying the distortion constraint $\bar{D}$. Set $k = 0$, and specify values of λ_{min}, λ_{max}, and a tolerance ε as the convergence criterion.

2) Fix $\mathbf{Q}^{(k)}$ for any $k \geq 0$. Search the suitable trade-off parameter λ^* over $[\lambda_{min}, \lambda_{max}]$ for the given storage constraint $\bar{B}$. Generate optimal structure $\mathcal{T}^{(k)}$ and schedule set $G^{(k)}()$ based on the greedy structure and schedule generation algorithm, which achieves the following unconstrained minimum given λ^*:

$$\begin{aligned} \min_{\mathcal{T}^{(k)} \in \Theta, G^{(k)}()} \quad & J(\mathcal{T}^{(k)}, G^{(k)}(), \mathbf{Q}^{(k)}) \\ & = C(\mathcal{T}^{(k)}, G^{(k)}(), \mathbf{Q}^{(k)}) + \lambda^* B(\mathcal{T}^{(k)}, \mathbf{Q}^{(k)}) \end{aligned}$$

For $k > 0$, if $C(\mathcal{T}^{(k)}, G^{(k)}(), \mathbf{Q}^{(k)}) > C(\mathcal{T}^{(k-1)}, G^{(k-1)}(), \mathbf{Q}^{(k)})$, continue to use frame structure and schedule at the previous iteration, *i.e.*, set $\mathcal{T}^{(k)} = \mathcal{T}^{(k-1)}$, $G^{(k)}() = G^{(k-1)}()$. If $\left\| (C(\mathcal{T}^{(k)}, G^{(k)}(), \mathbf{Q}^{(k)}) - C(\mathcal{T}^{(k-1)}, G^{(k-1)}(), \mathbf{Q}^{(k)})) / C(\mathcal{T}^{(k)}, G^{(k)}(), \mathbf{Q}^{(k)}) \right\| \leq \varepsilon$, stop the iteration and output $(\mathcal{T}^{(k)}, G^{(k)}(), \mathbf{Q}^{(k)})$ as optimal result. Otherwise, go to step 3 to continue the iteration.

3) Fix $\mathcal{T}^{(k)}$ and $G^{(k)}()$. Randomly select l ($l \geq 3$) quantization parameter pairs $\mathbf{Q}_i = [Q_{t,i}, Q_{d,i}]^T$, $i = 1, 2, \ldots, l$, calculate the transmission cost C_i and distortion cost D_i for the given $\mathcal{T}^{(k)}$ and $G^{(k)}()$. Then, estimate parameters A_1, A_2, Y_1, Y_2 and Z by solving two linear problems as (17). Update $\mathbf{Q}^{(k)}$ to $\mathbf{Q}^{(k+1)}$ based on (20) so that $\mathbf{Q}^{(k+1)}$ can achieve the following constrained minimum:

$$\begin{aligned} \min_{\mathbf{Q}^{(k+1)} \in \Lambda} \quad & C(\mathcal{T}^{(k)}, G^{(k)}(), \mathbf{Q}^{(k+1)}) \\ s.t. \quad & B(\mathcal{T}^{(k)}, \mathbf{Q}^{(k+1)}) \leq \bar{B}, \quad D(\mathbf{Q}^{(k+1)}) \leq \bar{D} \end{aligned}$$

4) Go to step 2, $k \leftarrow k + 1$.

where

$$\begin{aligned} A_1 = & \sum_{\delta} \Psi(\delta) \left(\sum_{F_{i,j} \in \mathcal{T}^{(k)}} q(F_{i,j}, \delta) X^t_{i,j} \right) \\ A_2 = & \left(\sum_{\delta} \Psi(\delta) \left(\sum_{F_{i,j} \in \mathcal{T}^{(k)}} q(F_{i,j}, \delta) X^d_{i,j} \right) \right) Y_2 \\ S = & \sum_{\delta} \Psi(\delta) \left(\sum_{F_{i,j} \in \mathcal{T}^{(k)}} q(F_{i,j}, \delta) (L^t_{i,j} + L^d_{i,j}) \right) \end{aligned}$$

Proof: By using Lemma 5.1 and taking R-Q model (16) and D-Q model (18) into (8), optimization sub-problem (13) can be found to become (19). ■

Note that after applying Lemma 5.1, the denominator of the second component in (19) denotes that the corresponding view synthesis distortion of the optimized QPs just satisfies the distortion constraint $\bar{D}$, while the constraint $Q^1_t \leq Q_t \leq Q^2_t$ guarantees the storage constraint $\bar{B}$ is satisfied.

Given the convexity of the target function in (19) over the range $[Q^1_t, Q^2_t]$, we next evaluate the minimization by

taking derivative of (19) with respect to Q_t [6]. The optimal[7] (Q_t^*, Q_d^*) is then found to be

$$\begin{aligned} Q_t^* = & \begin{cases} \frac{\sqrt{A_1}(\bar{D}-Z)}{\sqrt{A_1}Y_1+\sqrt{A_2}Y_1}, & Q_t^1 \le \frac{\sqrt{A_1}(\bar{D}-Z)}{\sqrt{A_1}Y_1+\sqrt{A_2}Y_1} \le Q_t^2 \\ Q_t^1, & \frac{\sqrt{A_1}(\bar{D}-Z)}{\sqrt{A_1}Y_1+\sqrt{A_2}Y_1} < Q_t^1 \\ Q_t^2, & \frac{\sqrt{A_1}(\bar{D}-Z)}{\sqrt{A_1}Y_1+\sqrt{A_2}Y_1} > Q_t^2 \end{cases} \\ Q_d^* = & (\bar{D} - Z - Y_1 Q_t^*)/Y_2 \end{aligned} \tag{20}$$

Note that similar to (16), to reduce the computational complexity, instead of using the specific expression in (19), all the necessary parameters A_1, A_2, Y_1, Y_2 and Z to calculate (Q_t^*, Q_d^*) in (20) can be directly derived by solving the linear problem (17) from multiple quantization parameters (Q_t, Q_d)'s for the fixed $\mathcal{T}^{(k)}$ and $G^{(k)}()$. The optimal QPs $\mathbf{Q}$ in Step 3 of the proposed iterative optimization algorithm can be then updated using (20).

At the end of this section, for the completeness of description, we describe in Table II the complete procedures to iteratively optimize frame structure, associated schedule and quantization parameters, based on the proposed greedy frame structure and schedule generation algorithm in Sec. V-B and QP update algorithm in Sec. V-C.

We claim that the proposed iterative joint optimization of frame structure, transmission schedule and quantization parameters in Table II is guaranteed to converge, which is stated formally as a theorem below.

Theorem 5.2: Based on the greedy frame structure and schedule generation and QP update algorithms, the convergence of the proposed iterative optimization is guaranteed.

Proof: The proof is given in Appendix II. ■

VI. Simulation Results

A. Simulation Setup

To gather multiview video data for our experiments, we encoded the first 90 frames of sequences `Dog`, `Pantomime` and `Champaign` [29] of 4 captured views ($K = 4$), at resolution 1280×960 and 30 frames per second (fps). Each sequence has different characteristic and camera setup, *e.g.*, different distance between neighboring capturing cameras, and capturing objects with various range of depth values. The MPEG depth estimation reference software (DERS) 3.0 [25] is used to generate the depth maps. We set the number of synthesized views between neighboring captured views to be $K' = 4$; hence there are $(K-1)K' + K = 16$ views available for clients' selection. To synthesize the virtual views, the view synthesis reference software (VSRS) 2.0 [22] is used for DIBR. Note that different multiview video characteristics can affect the amount of geometric errors caused by depth map estimation, which is reflected by the resulting view synthesis distortion of DIBR. To generate data for DSC implementation of M-frames, we use the algorithm in [20], developed using H.263 tools (with half-pixel motion estimation). In addition, the random access period Δ' and view-switch period Δ are set to be 30 and 3, respectively. The Lagrangian multiplier λ is swept from 0.01 to 40.96 to induce different tradeoffs between storage and transmission rate.

[6]Though we focus on the optimization of (19) in terms of Q_t, the same process can be carried on Q_d as well, correspondingly, (19) is formulated as function of Q_d

[7]The QPs of a practical video codec are chosen from a discrete set of values, while the R-Q and D-Q models we developed in this paper are in continuous domain. Therefore, there is an inherent rounding error for the resulting optimal QPs, Q_t^* and Q_d^*.

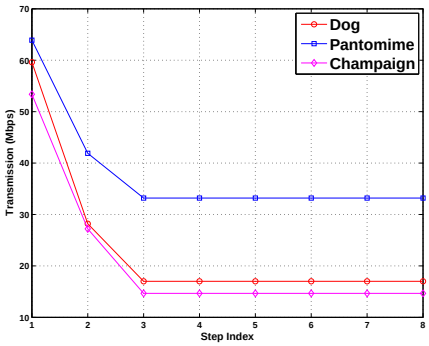


Fig. 12. Convergency rate of the iterative procedure, where three sequences are encoded at 320KBytes with distortion $D = 23$ for `Dog`, $D = 11$ for `Pantomime`, and $D = 36$ for `Champaign`.

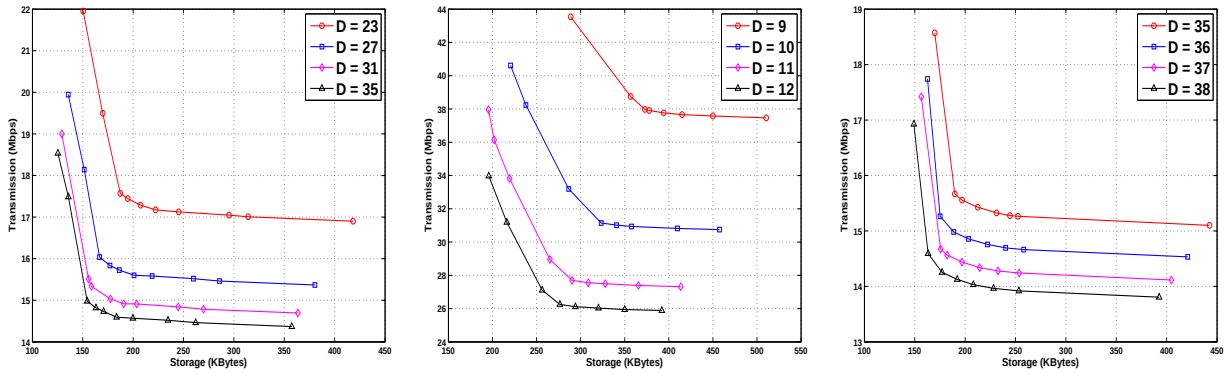
For view-switching interactive model, we set view difference bound $L = K' + 1 = 5$, which means the distance between two consecutive view-switches cannot exceed that between two neighboring captured views. In addition, we assume the view-switching probability function in the form $\Phi(n) = \phi_1 - \phi_2\|n\|$, $-5 \leq n \leq 5$, where $\phi_2 = (11\phi_1 - 1)/30$ such that $\sum_n \Phi(n) = 1$. Note that ϕ_1 is the probability that client switches to the view coordinate where she remains in the same view-switch direction as previous view-switch, and ϕ_2 is the decreased probability when she transitions to neighboring view coordinates in other view-switch directions. By changing ϕ_1, we can model the behaviors of video streaming clients with different view-switching habits.

For the PDF of RTT delay, we assume an uniform distribution with upper bound $RTT_{max} = 5\Delta$ (500ms) for simplicity, *i.e.*, $\psi(x) = 1/(5\Delta)$, $x \in (0, 5\Delta)$. Correspondingly, the PMF $\Psi(\delta)$ of δ could be calculated from (7) as $\Psi(\delta) = 0.2$, $1 \leq \delta \leq 5$.

B. Simulation Results

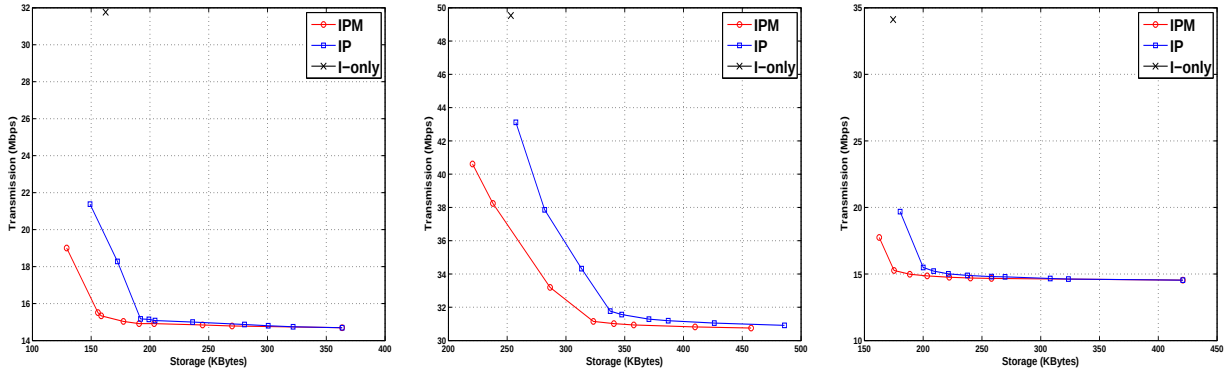
1) Convergency Speed of Iterative Optimization: We first examine how fast the proposed iterative joint optimization algorithm of frame structure, transmission schedule and QPs could converge. In Fig. 12, supposing $\phi_1 = 0.2$, we plot the change of transmission rate at each step of the proposed iterative algorithm, where the storage constraint is 320KBytes for all three sequences, the distortion constraint is set to be 23, 11 and 36 for `Dog`, `Pantomime` and `Champaign` respectively. In Fig. 12, the points with step indices $2i - 1$ and $2i$ represent the transmission rates after Step 2 and Step 3 at the i-th loop of the iterative procedure, respectively. First, we can see that as shown in the proof of Theorem 5.2 in Appendix II, the transmission rate is a non-increasing function at each step of the iterative algorithm. Second, we can observe that when applied to different sequences, the proposed iterative algorithm can coverage to the optimal solution very fast within 2 iterations, demonstrating the efficiency of the proposed algorithm to real multiview video data.

2) Algorithm Performance Comparison with Different Distortion Constraints: We next study the change of transmission rate resulting from our optimized structure, schedule and QPs when we vary storage and distortion constraints. When ϕ_1 is set to be 0.2, we generated tradeoff points between storage and transmission rate for different view synthesis distortion, shown in Fig. 13. First, for a given distortion, we see an inverse proportional relationship



(a) (b) (c)

Fig. 13. Tradeoff between storage and transmission rate with different distortion constraints: (a) `Dog`; (b) `Pantomime`; (c) `Champaign`.



(a) (b) (c)

Fig. 14. Tradeoff between storage and transmission rate using different coding configurations, for a given distortion constraint: (a) `Dog` with $D = 31$; (b) `Pantomime` with $D = 10$; (c) `Champaign` with $D = 36$.

between transmission rate and storage, because larger storage budget means more frame structure redundancy, resulting in more bandwidth-efficient P-frames used in a frame structure. Second, we observe that in general larger distortion means a smaller transmission rate and storage. This is also expected, since better view synthesis quality means smaller quantization distortion, leading to comparatively large frame sizes of encoded texture and depth maps, which are expensive for both storage and transmission bandwidth. In addition, for the same storage, the transmission rate drops more slowly as the increase of distortion constraint.

3) Algorithm Performance Comparison Using Different Source Coding Models: Then, we analyze the effects of different video source coding models on the performance of frame structures generated by our proposed algorithm. In Fig. 14, we plot the tradeoff points between storage and transmission rate for our algorithm using I-, P- and M-frames (`IPM`), using I- and P-frames (`IP`) and using only I-frames (`I-only`), when ϕ_1 is set to be 0.2 and distortion constraint is set to be 31, 10 and 36 for `Dog`, `Pantomime` and `Champaign` respectively. First, we observe that `I-only` has a single tradeoff point, because placing I-frames at all switching points results in no flexibility to trade off between storage and transmission rate, therefore could not take advantage of extra storage if

available to lower transmission rate.

Second, for the same storage, `IPM` offers lower transmission rate by up to 51.7% for `Dog`, 22.8% for `Pantomime` and 55.3% for `Champaign`, due to the optimal usage of redundant P- and DSC-frames. The performance improvement of `IPM` over `I-only` for `Pantomime` is much smaller than `Dog` and `Champaign`. This is due to the relatively small size of I-frames in `Pantomime`, as a result of almost textureless background region in the sequence, which introduces adequate spatial redundancy for efficient intra prediction.

Third, we observe that structures using M-frames can offer a noticeable improvement over those using I-frames, with transmission rate saving up to 27.5% for `Dog`, 11.3% for `Pantomime` and 23.9% for `Champaign`. The improvement is larger at stringent storage constraint (high transmission rate), because DSC-frames are more often used by the optimized frame structure to lower overall storage.

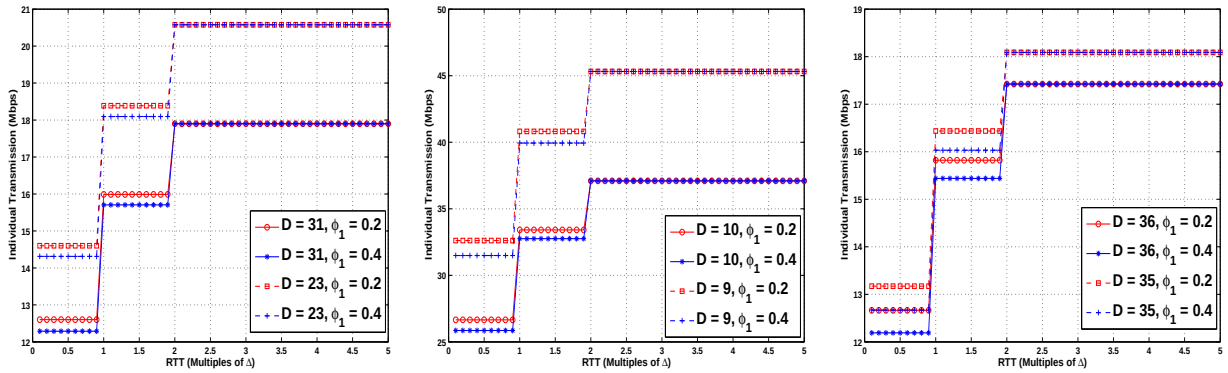
4) Algorithm Performance Comparison with Different RTT Delays: We then evaluate the impact of RTT delays on the performance of frame structures optimized from the proposed algorithm. Given the corresponding storage and distortion constraints, Fig. 15 depicts the change in expected transmission rate with the increase of RTT delay when the same frame structure generated from the proposed algorithm is individually operated on server-client channels with different RTT delays. More specifically, we first optimize the frame structure, schedule and QPs to lower the expected transmission rate with respect to the PDF of RTT, $\psi(x)$, subject to given storage and view synthesis distortion constraints as discussed in Sec. V. We then compare the corresponding individual transmission rate (9) when the resulting frame structure performs on different specific RTT delays. To induce different view-switching probability $\Phi(n)$, ϕ_1 is set to be 0.2 and 0.4 for two trials.

First, we see that transmission rate is a non-decreasing step function with the increase of RTT delay, and in general, larger RTT delay results in more transmission bandwidth consumption. This is intuitive: given a frame structure $\mathcal{T}$, all RTT delays RTT's, $(\delta-1)\Delta \leq RTT \leq \delta\Delta$ will use the same transmission schedule $G(\delta)$, leading to the same coded frames delivered from video server for each transmission, while in overall larger RTT delay means more view-switch positions to cover at each structure slice, resulting in larger transmission rate.

Second, we can observe that transmission rate cannot be further increased when $RTT \geq 2\Delta$. This can also be easily explained: when $RTT \geq 2\Delta$, one client is able to reach all $(K-1)K' + K = 16$ available view-switch positions within one RTT no matter which view-switch position she choose one RTT before. Correspondingly, each structure slice needs to cover all $K = 4$ captured views, resulting in a constant transmission rate.

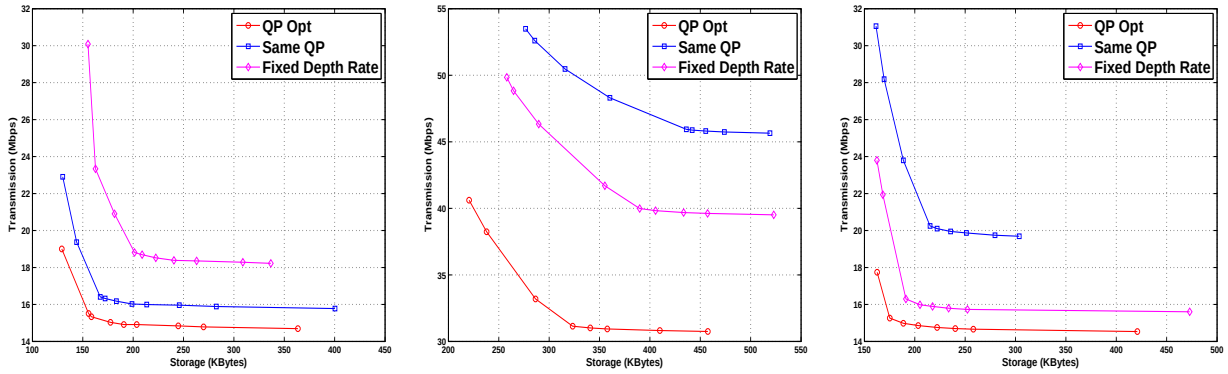
Third, as ϕ_1 is increased, the transmission rate of the optimized frame structure decrease. This is expected: larger ϕ_1 means higher probability that client remains at the same view-switch direction, which also increases the probability that client stays at the same view after one view-switch; therefore, more P-frames predicted from the previous frames of the same view are used in the structure, resulting in lower transmission rate. Moreover, the frame structure has the same transmission rate when $RTT \geq 2\Delta$, independent on ϕ_1, because of transmission of all captured views at each slice structure.

5) Improvement of Texture/Depth QP Optimization: Finally, we verify the effectiveness of the proposed quantization optimization algorithm for texture and depth map coding. Using the same distortion constraints for three



(a) (b) (c)

Fig. 15. Transmission rate of a frame structure versus RTT delay: (a) `Dog` at 360KBytes; (b) `Pantomime` at 420KBytes; (c) `Champaign` at 420KBytes.



(a) (b) (c)

Fig. 16. Tradeoff between storage and transmission rate using different selection methods for texture and depth QPs, for a given distortion constraint: (a) `Dog` with $D = 31$; (b) `Pantomime` with $D = 10$; (c) `Champaign` with $D = 36$.

sequences in Fig. 14, Fig. 16 compares the tradeoff points between storage and transmission rate generated by our quantization optimization algorithm (`QP Opt`) with two anchor results. The first (`Same QP`) uses the same QP to encode both texture and depth maps. The second (`Fixed Depth Rate`) is a constant rate allocation method with a pre-defined depth rate equal to 50% of texture rate. We observe that `QP Opt` consistently outperforms the other two methods for all test sequences, while `Fixed Depth Rate` is better than `Same QP` for `Pantomime` and `Champaign`, but worse for `Dog`. For example, at a storage of 300 KBytes `QP Opt` yields a transmission rate reduction over `Same QP` about 8%, 38% and 26%, and over `Fixed Depth Rate` about 19%, 32% and 7%, for `Dog`, `Pantomime` and `Champaign` respectively. It illustrates the importance of joint texture and depth quantization optimization.

VII. Conclusion

In this paper, we propose three major technological improvements to existing IMVS works. First, in addition to camera-captured views, we make available additional virtual views between each pair of captured views for

clients' selection, by transmitting both texture and depth maps of neighboring captured views and synthesizing intermediate views at decoder using DIBR. Second, we construct a Markovian view-switching model that more accurately captures viewers' behaviors. Third, we optimize frame structures and schedule the transmission of frames in a network-delay-cognizant manner, so that clients can enjoy zero-delay view-switching even over transmission network with non-negligible RTT.

We formalize the joint optimization of the frame encoding structure, transmission schedule, and QPs of the texture and depth maps, and propose an iterative algorithm to achieve fast and near-optimal solutions. Experimental results show that our proposed rate allocation method can lower transmission rate by up to 38% over naïve schemes. In addition, for the same storage, using our generated frame structures can lower transmission rate by up to 55% compared to I-frame-only structures, and up to 27% compared to structures without M-frames.

APPENDIX I
PROOF OF LEMMA 5.1

As shown in Fig. 11, given storage and distortion constraints $\bar{B}$ and $\bar{D}$ in (13), both two boundary lines l_1 and l_2 of the valid QP region are monotonically decreasing functions. Therefore, for any point $\mathbf{Q}^a = [Q_t^a, Q_d^a]^T$ in the valid region, we can always identify one unique point $\mathbf{Q}^b = [Q_t^b, Q_d^b]^T$ on l_2 such that $Q_t^b = Q_t^a$ and $Q_d^b \geq Q_d^a$. On the other hand, given a frame structure $\mathcal{T}^{(k)}$ associated with schedule set $G^{(k)}()$ and a texture quantization parameter Q_t, transmission cost function $C(\mathcal{T}^{(k)}, G^{(k)}(), \mathbf{Q})$ is strictly decreasing function in terms of depth quantization parameter Q_d. So, we can have $C(\mathcal{T}^{(k)}, G^{(k)}(), \mathbf{Q}^a) \geq C(\mathcal{T}^{(k)}, G^{(k)}(), \mathbf{Q}^b)$. This proves that the optimal quantization solution $\mathbf{Q}$ in (13) is located on l_2.

APPENDIX II
PROOF OF THEOREM 5.2

In Table II, given frame structure $\mathcal{T}^{(k-1)}$, schedule $G^{(k-1)}()$ and QP $\mathbf{Q}^{(k)}$ at step 2 of the k-th iteration, we compare the new result of $\mathcal{T}^{(k)}$ and $G^{(k)}()$ to that of $\mathcal{T}^{(k-1)}$ and $G^{(k-1)}()$ such that $C(\mathcal{T}^{(k)}, G^{(k)}(), \mathbf{Q}^{(k)}) \leq C(\mathcal{T}^{(k-1)}, G^{(k-1)}(), \mathbf{Q}^{(k)})$. This means the transmission cost function is noincreasing at step 2. On the other hand, given frame structure $\mathcal{T}^{(k)}$, schedule set $G^{(k)}()$ and QP $\mathbf{Q}^{(k)}$ of the k-th iteration, we searches at step 3 the entire space of all possible QPs, Λ, for the best solution $\mathbf{Q}^{(k+1)}$ to lower transmission cost, *i.e.*, $\mathbf{Q}^{(k)} \in \Lambda$. This means $C(\mathcal{T}^{(k)}, G^{(k)}(), \mathbf{Q}^{(k+1)}) \leq C(\mathcal{T}^{(k)}, G^{(k)}(), \mathbf{Q}^{(k)})$. Hence, we prove that the transmission cost function is a noincreasing function at each step of the proposed iterative optimization algorithm.

Since both Θ and Λ are finite space, the noincreasing nature of the transmission cost function guarantees that the proposed iterative algorithm is surely to converge.

REFERENCES

[1] C. Zhang, Z. Yin, and D. Florencio, "Improving depth perception with motion parallax and its application in teleconferencing," in *IEEE International Workshop on Multimedia Signal Processing*, Rio de Jeneiro, Brazil, October 2009.

[2] S. Reichelt, R. Haussler, G. Futterer, and N. Leister, "Depth cues in human visual perception and their realization in 3D displays," in *Proceedings of SPIE*, Orlando, FL, April 2010.

[3] E. Kurutepe, M. R. Civanlar, and A. M. Tekalp, "Client-driven selective streaming of multiview video for interactive 3DTV," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no.11, November 2007, pp. 1558–1565.

[4] P. Merkle, A. Smolic, K. Muller, and T. Wiegand, "Efficient prediction structures for multiview video coding," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no.11, November 2007, pp. 1461–1473.

[5] G. Cheung, A. Ortega, and N.-M. Cheung, "Generation of redundant coding structure for interactive multiview streaming," in *Seventeenth International Packet Video Workshop*, Seattle, WA, May 2009.

[6] G. Cheung, N.-M. Cheung, and A. Ortega, "Optimized frame structure using distributed source coding for interactive multiview streaming," in *IEEE International Conference on Image Processing*, Cairo, Egypt, November 2009.

[7] G. Cheung, A. Ortega, and N.-M. Cheung, "Interactive streaming of stored multiview video using redundant frame structures," in *IEEE Transactions on Image Processing*, vol. 20, no.3, March 2011, pp. 744–761.

[8] J.-G. Lou, H. Cai, and J. Li, "A real-time interactive multi-view video system," in *Proc. ACM Multimedia*, Nov. 2005, pp. 161–170.

[9] H.-Y. Shum, S.-C. Chan, and S. B. Kang, *Image-Based Rendering*. Springer, 2007.

[10] P. Merkle, A. Smolic, K. Mueller, and T. Wiegand, "Multi-view video plus depth representation and coding," in *IEEE International Conference on Image Processing*, San Antonio, TX, October 2007.

[11] T. Fujii and M. Tanimoto, "Free viewpoint TV system based on ray-space representation," in *Proc. of SPIE*, vol. 4864, 2002, p. 175.

[12] A. Kubota, A. Smolic, M. Magnor, M. Tanimoto, T. Chen, and C. Zhang, "Multi-view imaging and 3DTV," in *IEEE Signal Processing Magazine*, vol. 24, no.6, November 2007, pp. 10–21.

[13] A. Aaron, P. Ramanathan, and B. Girod, "Wyner-Ziv coding of light fields for random access," in *IEEE International Workshop on Multimedia Signal Processing*, Siena, Italy, September 2004.

[14] D. Taubman and R. Rosenbaum, "Rate-distortion optimized interactive browsing of JPEG2000 images," in *IEEE International Conference on Image Processing*, Barcelona, Spain, September 2003.

[15] N.-M. Cheung, H. Wang, and A. Ortega, "Video compression with flexible playback order based on distributed source coding," in *IS&T/SPIE Visual Communications and Image Processing (VCIP'06)*, San Jose, CA, January 2006.

[16] S. Shimizu, M. Kitahara, H. Kimata, K. Kamikura, and Y. Yashima, "View scalable multiview coding using 3-D warping with depth map," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no.11, November 2007, pp. 1485–1495.

[17] A. M. Tekalp, E. Kurutepe, and M. R. Civanlar, "3DTV over IP: End-to-end streaming of multiview video," in *IEEE Signal Processing Magazine*, November 2007.

[18] M. Karczewicz and R. Kurceren, "The SP- and SI-frames design for H.264/AVC," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no.7, July 2003, pp. 637–644.

[19] N. Cheung and A. Ortega, "Distributed source coding application to low-delay free viewpoint switching in multiview video compression," in *Proc. of Picture Coding Symposium, PCS'07*, Lisbon, Portugal, Nov. 2007.

[20] N.-M. Cheung, A. Ortega, and G. Cheung, "Distributed source coding techniques for interactive multiview video streaming," in *27th Picture Coding Symposium*, Chicago, IL, May 2009.

[21] G. Cheung, A. Ortega, and T. Sakamoto, "Coding structure optimization for interactive multiview streaming in virtual world observation," in *IEEE International Workshop on Multimedia Signal Processing*, Cairns, Queensland, Australia, October 2008.

[22] M. Tanimoto, T. Fuji, K. Suzuki, N. Fukushima, and Y. Mori, "View synthesis algorithm in view synthesis reference software (vsrs) 2.0," in *ISO/IECJTC1/SC29/WG11*, Februray 2008.

[23] S. Gokturk, H. Yalcin, and C. Bamji, "A time-of-flight depth sensor—system description, issues and solutions," in *Conference on Computer Vision and Pattern Recognition Workshop (CVPRW)*, Washington, DC, June 2004.

[24] W. Mark, L. McMillan, and G. Bishop, "Post-rendering 3D warping," in *Symposium on Interactive 3D Graphics*, NY, April 1997.

[25] M. Tanimoto, T. Fuji, M. P. Suzuki, and M. Wildeboer, "Depth estimation reference software (DERS) 3.0," in *ISO/IECJTC1/SC29/WG11*, April 2009.

[26] S. Wee, W. Tan, J. Apostolopoulos, and M. Etoh, "Optimized video streaming for networks with varying delay," in *IEEE International Conference on Multimedia and Expo*, Lausanne, Switzerland, August 2002, pp. 89–92.

[27] D. Bertsekas and R. Gallager, Eds., *Data Networks*. Prentice Hall, 1992.

[28] C. Wong, O. C. Au, B. Meng, and H. Lam, "Novel H.26X optimal rate control for low-delay communications," in *International Conference on Information Communications and Signal Processing*, December 2003, pp. 90–94.

[29] "Test sequences," http://www.tanimoto.nuee.nagoya-u.ac.jp/.