Interactive Streaming of Stored Multiview Video Using Redundant Frame Structures

Gene Cheung, *Senior Member, IEEE*, Antonio Ortega, *Fellow, IEEE*, and Ngai-Man Cheung, *Member, IEEE*

***Abstract*—While much of multiview video coding focuses on the rate-distortion performance of compressing all frames of all views for storage or non-interactive video delivery over networks, we address the problem of designing a frame structure to enable interactive multiview streaming, where clients can interactively switch views during video playback. Thus, as a client is playing back successive frames (in time) for a given view, it can send a request to the server to switch to a different view while continuing uninterrupted temporal playback. Noting that standard tools for random access (i.e., I-frame insertion) can be bandwidth-inefficient for this application, we propose a redundant representation of I-, P-, and "merge" frames, where each original picture can be encoded into multiple versions, appropriately trading off expected transmission rate with storage, to facilitate view switching. We first present ad hoc frame structures with good performance when the view-switching probabilities are either very large or very small. We then present optimization algorithms that generate more general frame structures with better overall performance for the general case. We show in our experiments that we can generate redundant frame structures offering a range of tradeoff points between transmission and storage, e.g., outperforming simple I-frame insertion structures by up to 45% in terms of bandwidth efficiency at twice the storage cost.**

***Index Terms*—Media interaction, multiview video coding, video streaming.**

I. Introduction

MULTIVIEW video consists of sequences of spatially correlated pictures captured simultaneously and periodically by multiple closely spaced cameras. The cameras can be parts of a real physical camera system [1] or a virtual camera setup inside a computer [2] using API manipulation of graphics cards [3]. Much of the previous research in multiview video focuses on compression: to design coding techniques exploiting temporal (across time) and spatial (across view) correlation to encode all frames of all views of a multiview sequence in a rate-distortion optimal manner [4]–[7].

Manuscript received December 21, 2009; revised May 17, 2010; accepted August 08, 2010. Date of publication August 26, 2010; date of current version February 18, 2011. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Alex C. Kot.

G. Cheung is with the National Institute of Informatics, Tokyo 101-8430, Japan (e-mail: cheung@nii.ac.jp).

A. Ortega is with the Department of Electrical Engineering, University of Southern California, Los Angeles, CA 90089-2564 USA (e-mail: antonio.ortega@sipi.usc.edu).

N.-M. Cheung is with the Department of Electrical Engineering, Stanford University, Stanford, CA 94305 USA (e-mail: ncheung@stanford.edu).

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

Digital Object Identifier 10.1109/TIP.2010.2070074

Compression of all multiview data in an interdependent way is a sensible objective when the intended application is storage of the entire data set, or *non-interactive* delivery over networks, i.e., where, as in TV broadcast [8], the clients' potential interaction with the received multiview content does not effect how and what data is delivered from the server.

In this paper, we focus instead on the problem of *interactive multiview video streaming* (IMVS). In this problem, after one representation of a multiview sequence is pre-encoded and stored at the server, streaming clients *interactively* request desired views for successive video frames in time. Each client requests and plays back one single view at a time out of possibly many available views, meaning that the requested data corresponds to only a small subset out of a large set of available multiview data at the server. The encoding is done once at the server for a possibly large group of clients, each of which can navigate the content by playing it back (in time) while switching views, thus resulting in a different traversal of views across time for each user. Our goal is to provide a desired level of view interactivity with minimum expected transmission bandwidth cost. The extent of view interactivity is determined by the *view switching period* T; i.e., view switching can only take place at multiples of T frames.

A natural approach to enable this kind of interactive view switching is to make use of standard random access tools, i.e., making every Tth frame (in all views) an I-frame. Our work is based on the observation that *random access and view switching are fundamentally different functionalities*, and thus efficient tools for one problem may not provide the best solution for the other. For random access to a frame, one can make no assumptions about which frames are available at the decoder; independently coded I-frames are therefore well suited for this purpose. View switching, on the other hand, arises when temporal playback is not interrupted; i.e., successive frames in time are displayed, but one wishes to switch point of view. The key difference with respect to random access is that the decoder has access to some of the frames (possibly from a different view) immediately preceding in time the requested frame. Thus, since consecutive frames in different views tend to be correlated, using an I-frame for switching can be inefficient in terms of bandwidth.

The main focus of our work is then to study alternatives for view switching that are more bandwidth-efficient than simple I-frame insertions. Note that our proposed tools *do not* support random access, and thus we are not advocating using these tools *instead* of random access tools such as I-frame insertion. Rather, we propose to consider view switching and random access as two explicitly different functionalities, supported by different tools. It will then be up to the system designer to select the

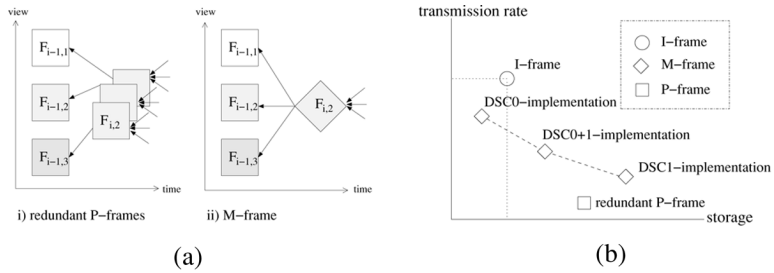


Fig. 1. Examples and transmission rate/storage tradeoffs at one switching point using I-, various M-, and redundant P-frames. (a) Redundant P-frames and M-frame. (b) Rate/storage tradeoffs at one switching point.

appropriate setting for a given application. For example, one may select a parameter T for view switching and separately allow random access at every T'th frame, where typically $T \ll T'$.

One can see that the design of a multiview representation to permit interactive view switching can involve a tradeoff between expected transmission rate and required storage space. For simplicity consider the $T = 1$ case, i.e., we require a user's ability to switch views at any time, but restrict allowable switching to view j from a neighboring view k in a one-dimensional camera array setup, i.e., $j - 1 \leq k \leq j + 1$. Since temporal playback is not interrupted, at time i one of the previous frames $F_{i-1,k}$'s of time $i - 1$ (for at most three different views k) will be available at the decoder. Thus, one possible way to support switching would be the following: for each possible decoded frame $F_{i-1,k}$, differentially encode one P-frame to represent original picture $F^o_{i,j}$ at time i and view j using $F_{i-1,k}$ as a predictor—we call this approach *redundant P-frames*. An example is shown in Fig. 1(a)(i) where three P-frames representing the same picture $F^o_{i,2}$ are constructed, each using a different predictor. Doing so for every view j will increase the number of decoding paths (frame representations) at each switching instant by a factor of three, resulting in a tree structure of size $O(3^N)$ when there are N switching instants in between two I-frame insertion points. So while performing this redundant P-frame encoding for every view at every switching instant would lead to a structure with minimum transmission cost (only bandwidth-efficient P-frames are used), the size of the resulting structure is impractically large.

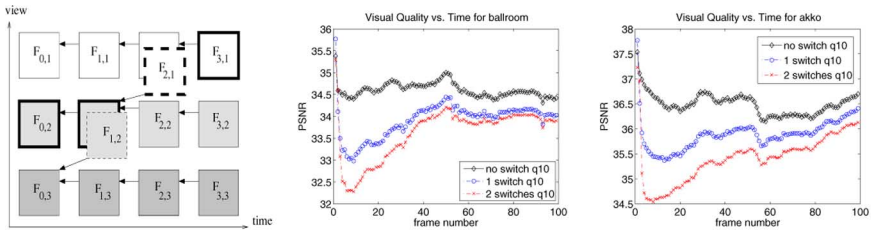
At the other extreme, one can construct a *single* quantized representation of original picture $F^o_{i,j}$ ($F_{i,2}$ in Fig. 1(a)(ii)) for all possible decoder states; i.e., a frame (we call *merge frame* or *M-frame*) that can be correctly and identically decoded *no matter from which* $F_{i-1,k}$ the user is switching (frame $F_{i-1,1}$, $F_{i-1,2}$ or $F_{i-1,3}$ in Fig. 1(a)(ii)). Doing so will keep the number of frame representations (and hence decoding paths) at a switching instant to a minimum K for K total views. 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 employ differential coding, using only *one* predictor from the previous frame set $F_{i-1,k}$'s (whichever one is available at the decoder) and produces the exact same reconstruction regardless of which predictor was used. Example implementations of M-frames include SP-frames in H.264 [9] and different distributed source coding (DSC) techniques [10], [11].[1] Different implementations of M-frames typically result in different tradeoffs between storage and transmission costs, as shown in Fig. 1(b).[2] Note, however, that all implementations of M-frame must have larger transmission rate than a P-frame, since by definition, an M-frame must merge *multiple* (more than one) decoding paths each with a different predictor—a more stringent reconstruction requirement than a P-frame which has only one decoding path with one known predictor. If we encode an M-frame using the most storage-efficient implementation available—the left-most point on the convex hull[3]—for every view at every switching point, this leads to the most storage-efficient frame structure. However, as shown in Fig. 1(b), this structure has high transmission rate, since the size of an M-frame is in general much larger than a P-frame.

Note that the underlying assumption of the two approaches above is that *coding drift* is not desirable and must be avoided, either by preserving decoding paths via redundant P-frames, or by perfectly merging decoding paths via a single M-frame that reconstructs to the exact same frame representation no matter which decoded frame is available at the decoder for prediction. A simple experiment can show that coding drift is indeed non-negligible. Suppose we first encode a closed loop prediction representation for frames within the same view, then add cross-view P-frames to facilitate view switches, as shown in Fig. 2(a). For example, to switch from view 2 to view 1 at instant 2, we send the bold outlined frames in Fig. 2(a), even though $F_{3,1}$ is predictively coded using a different version of $F_{2,1}$ as predictor instead of the transmitted $F_{2,1}$, resulting in coding drift in $F_{3,1}$ onwards. (Due to quantization, the reconstructed image from decoding path ending with cross-view P-frame will be different from image from path ending with same-view P-frame of same view and instant.) Using H.263+ for source coding, we see in Fig. 2(b) and (c) that due to drift, a single view switch resulted in 1.2 dB loss in visual quality (PSNR), while two consecutive switches resulted in well over 2 dB loss for both `ballroom` and `akko&kayo` sequences. Given view switches are a likely event in an IMVS scenario, our discussed approaches of redundant

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

[2]Though in Fig. 1(b) `DSC1` has higher transmission rate *and* storage than redundant P-frames at one switching point, due to its ability to merge all decoding paths into one frame, `DSC1` does not lead to exponential number of decoding paths and storage in the entire structure.

[3]We discuss a storage-efficient implementation of an M-frame using DSC—`DSC0` in Fig. 1(b)—in Section III.



(a) (b) (c)

Fig. 2. Experiments showing effects of coding drift on visual quality during view switches. (a) Frame structure where pictures in the same view are encoded using I- and subsequent P-frames, then additional cross-view P-frames (dotted edges) are added to faciliate view switches. Bold-edge frames are transmitted to provide view switch from view 2 to 1 at instant 2. Drift begins at $F_{3,1}$ due to predictor mismatch. (a) Structure with drift, (b) `ballroom`, (c) `akko&kayo`.

P-frames and M-frames to circumvent coding drift are reasonable if one chooses to preserve quality across view switches.[4]

From the above discussion, it is clear that more practical multiview representations lie between the two extremes, containing the optimal combination of I-frames (for random access), P-frames (for low transmission rate) and M-frames (for storage efficiency), that optimally trade off transmission and storage costs. In this paper, we develop heuristics and optimization algorithms that construct good frame structures for IMVS using available I-, P, and implementations of M-frames as building blocks. In our experiments with several multiview video datasets, we show that our algorithm can offer a range of tradeoff points between transmission and storage costs, and that the generated frame structures outperform the simple I-frame insertion approach. In particular, we show that in some cases our algorithm generates frame structures reducing expected transmission rate by up to 45% compared to I-frame insertion approach, at twice the storage costs.

The outline of the paper is as follows. We first review related work in Section II. We then overview our IMVS system and models and describe three example implementations of M-frames in Section III. We formulate the problem of optimally generating redundant frame structures for IMVS in Section IV. For intuition, we first discuss two classes of heuristic structures for the extreme cases when the likelihood of switching view is very high or very small, and then based on the intuition developed, we discuss a greedy optimization that performs reasonably well for all values of view switching probabilities in Section V. To further improve performance, we extend the greedy optimization into a provably optimal recursive one (under some simplifying conditions) in Section VI at the cost of increased complexity. Finally, we discuss our experiments and conclude in Sections VII and VIII, respectively.

II. Related Work

A. Interactive Media Streaming

The communication paradigm for our IMVS work 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 sharp contrast to *non-interactive media streaming* scenarios like terrestrial digital TV broadcast [8], where the entire media set is delivered server-to-client before a client interacts with the received data set (e.g., switching TV channel, etc). Interactive media streaming has the advantage of reduced bandwidth utilization since only the requested media subset is transmitted and is used for a wide range of media modalities. One example is interactive browsing of JPEG2000 images [12], [13], where a small spatial area, selected by a user, of a possibly very large image encoded using discrete wavelet transform, is transmitted successively via incremental quality layers. Yet another example is video playback with flexible decoding [14], where a video frame is DSC encoded using both past and future frames as predictors (side information), so that frames can be sent either forward or backward in time per client's request, and the client can simply decode and play back the video in the transmission order with no excess buffering.

B. Interactive Light Field Streaming

In the case of *light fields* [15], where a subset of a densely sampled 2-D array of images is used to interpolate a desired view using image-based rendering (IBR) [16], the notion of interactive media streaming has been investigated extensively [17]–[22]. These works are motivated by the very large size of the original image set—on the order of tens of Gigabytes [23]—so that sending the entire set before view navigation will create intolerably large delay to the user. To exploit inter-view spatial correlation among nearby views for coding gain, many works employed *disparity compensation* while providing some level of random access. Two representative works are [17] and [19], which used DSC and SP-frame-like lossless coding respectively to accommodate different decoding paths. We first note that both proposals in [17] and in [19] are captured in our M-frame abstraction of "single frame reconstruction using one of multiple predictors". Hence, these are viewed as implementations of an M-frame in our formulation, using which our proposed optimization can generate structures containing the optimal combinations. We also note that [17], [19] do not consider multiple decoding paths resulting from the use of redundant P-frames to represent an original picture, hence [17], [19] provide no mechanism to systematically further lower transmission cost by making use of extra storage capacity, if available.

[4]A generalization is to use an *imperfect* M-frame to merge multiple decoding paths, such that the resulting reconstructed frame depends on the predictor available at the decoder, but any two reconstructed frames (using two different predictors) differ by no more than a chosen threshold $\hat{d}$ to bound coding drift in the system. Theoretically, the same optimization presented in this paper can be used to find optimal combinations of redundant P-frames and such an imperfect M-frame. Implementation of an imperfect M-frame, however, is left for future work.

[21], [22] first assumed that each coding block of an image is encoded as INTRA, INTER or SKIP as done in H.263 [24]. Then, for a requested INTER coding block to be correctly decoded, all blocks in its dependency chain[5] that are *not* already in the client cache must be transmitted, creating a cost both in transmission rate and decoding complexity. The notion of transmitting and decoding multiple blocks before displaying a single one is called *rerouting* in our previous work [25]. Our findings in [25] showed that rerouting provides marginal performance gain in the IMVS scenario when view switching period T is reasonably large and when redundant P-frames are already used (multiple-frame representation of a single picture is not considered in [21], [22]). Hence, we will assume rerouting is not used during formulation in Section IV.

In summary, though we focus on the IMVS scenario in developing our optimization framework in this paper, we can theoretically retarget our optimization for interactive light field streaming: previous switching techniques like [17], [19] can be abstracted into different implementations of M-frames, and rerouting [21], [22] can be easily incorporated into the optimization as done in our earlier work [25]. Unlike these previous work on light fields, our optimization can in addition systematically trade off transmission rate with storage by controlling the number of decoding paths created by redundant P-frames. An in-depth investigation specifically for light field streaming using our optimization is left for future work.

C. Interactive Multiview Video Streaming

Most of the previous research in multiview video has focused on efficient compression of all frames of all views exploiting temporal (across time) and spatial (across view) correlation [4]–[6]. This approach makes obvious sense when the application is compact storage of the entire multiview content, or non-interactive media streaming applications as described earlier.

For interactive streaming of stored multiview video, the two-layer approach proposed in [26], [27] 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 between transmission rate and storage using combinations of redundant P-frames and M-frames in our frame structure.

The most similar work to our proposal is [28], 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 explore the best possible tradeoffs among these quantities in one structure given an interaction model.

In our previous work, we formally posed the IMVS problem as a combinatorial optimization in [2], proved its NP-hardness, and provided two heuristics-based algorithms to find good frame structures while allowing unlimited rerouting for IMVS. [25] is a more thorough and analytical treatment of the same problem with limited rerouting, 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 [11]. Preliminary results of using I-, P-, and DSC frames in an IMVS optimized structure is presented in [29]; this paper is a generalization of [29] where the optimization is posed as a search for the best combination of I-, P-, and generalized M-frames.

D. Distributed Source Coding

DSC [30] has been studied extensively in the past few years for low-complexity video encoding, e.g., [31], [32]. In addition, DSC has been investigated for a variety of applications, ranging from scalable coding [33], [34], error resilience video transmission [35], [36], distributed compression of multiview image/video [37], [38], to hyperspectral image compression [39] (see [40] for a recent survey). In this paper, we extend our previous work [10], [11] to apply DSC to facilitate interactive view switching, and this is significantly different from other DSC applications. Relevant information theoretic results for this setup were developed in [41], [42].

III. System Model, Interaction Model & M-Frame Implementations

To transition to later sections describing our core contribution on frame structure optimization for IMVS, as introduction material we first overview the IMVS system model, the assumed interaction model between an observer and the multiview content, and describe three different implementations for merge frames (M-frames).

A. System Model

The system model we consider for our IMVS problem is shown in Fig. 3. A *Multiview Video Source* simultaneously captures multiple pictures of K different views at regular intervals. An example of a multiview sequence of two views across four time instants is shown in Fig. 5(a). A *Video Server* sequentially grabs the captured uncompressed pictures from Multiview Video Source and encodes them offline, using an optimized frame structure $\mathcal{S}$ of I-, P-, and different implementations of M-frames for each picture batch of all K views across T' capturing intervals. The Video Server stores the structured representation of the sequence locally, using which the server serves multiple streaming clients subsequently. An alternative approach of real-time encoding a path traversal tailor-made for each streaming client's interactivity is computationally prohibitive if the number of clients is large.

In the sequel, we will use the term *frame* to denote a specific coded version of a picture and the term *picture* for the corresponding original captured image. A frame can be an intra-coded *I-frame*, a differentially coded *P-frame* with a single predictor, or a conceptual merge frame (*M-frame*) that

[5]By dependency chain we mean all the blocks that need to be decoded before the requested block can be decoded, i.e., an INTRA block followed by a succession of disparity compensated INTER blocks.

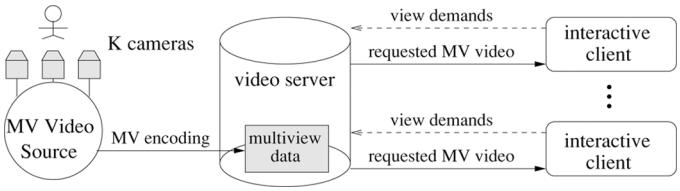


Fig. 3. Interactive multiview video streaming system.

uses a known set of predictors at encoding time. I-frame is used for random access. For view switching, P-frame offers the lowest possible transmission rate but increases the storage required as the number of decoding paths multiplies over time. M-frame offers the merging of multiple decoding paths into one single representation but at a higher transmission cost than P-frame.

Available M-frame *implementations* include SP frames in H.264 [9] or DSC [10], [11]. Each implementation itself can be composed of one or more encoded *components*, each encoded using a different coding technique. Each implementation of M-frame must satisfy the *M-frame reconstruction condition*: the exact same decoded version must be reconstructed at the decoder no matter which one of a known set of predictors is available at the decoder buffer. Three example implementations of M-frames using DSC are described in detail in Section III.C. Our system will have representation redundancy, in the sense that there may be more than one coded version (frame) to represent a given picture.

B. View Interaction Model

We assume a view interaction model where, upon watching any decoded version of the picture $F^o_{i,j}$, corresponding to time instant i and view j, an interactive client will request a coded version of picture $F^o_{i+1,k}$ of view k and *next* time instant $i+1$, where view k is between $j-1$ and $j+1$,[6] with *view transition probability* $\alpha_{i,j}(k)$; we call this interactivity *forward view switching*.[7] Another possible interactivity for multiview video is to freeze video in time and switch view (*static view switching*); this interactivity can be efficiently supported by novel usage of DSC [11]. Incorporation of static view switching into our optimization framework is left for future work.

Note that a significant difference between our setting and that of general light field streaming [17]–[22] is that in the latter case the user is free to explore a static scene in all directions, while here we play forward in time with only limited switching possibilities (i.e., among neighboring views). The implication is that only a limited number of previous frames $F_{i-1,k}$'s of different views k's (three views in our setting) could have been decoded when a current frame $F_{i,j}$ is requested, and so redundant P-frame representation of a picture is more practical in IMVS than general light field streaming. (However, redundant P-frames can be practical for light field streaming if a more restricted interaction model—only vertical and horizontal view switches are permitted, for example—is adopted.)

Though our interactive model presumes a client's desire to switch views at single-frame level ($T=1$), our model encompasses the more general case of a view switching period $T>1$. In the more general case, a "frame" $F_{i,j}$ in our model can represent T consecutive actual frames of view j (a carefully chosen I-, P- or M-frame determined by our optimization followed by $T-1$ consecutive P-frames of the same view).

C. Implementations of M-Frames

We discuss three novel implementations of M-frames using DSC [11]. Each implementation represents a different tradeoff between transmission and storage cost as illustrated in Fig. 1(b).

1) DSC Implementation 0 of M-Frame: The first DSC implementation (`DSC0`) of M-frame is straightforward: construct a single DSC component $W^0_{i,j}$ for all possible transitions into view j of instant i from frames $F_{i-1,k}$'s of previous instant using algorithm in [10]. In other words, given side information of previous frames $F_{i-1,k}$'s, encode a DSC image for target $F^o_{i,j}$ using codec in [10], such that the same image can be reconstructed no matter which one of previous frames $F_{i-1,k}$'s is present at the decoder buffer. The size of implementation `DSC0` (lone DSC component $W^0_{i,j}$) is modeled as $r^{W0}_{i,j}(d)$, where d is the maximum view index difference between DSC component $W^0_{i,j}$ and a predictor. Size of a DSC image in general is inversely proportional to the amount of correlation between the target and the weakest correlated predictor [10]. An example of `DSC0` is shown in Fig. 4(b).

The transmission cost of `DSC0` is simply $r^{W0}_{i,j}(d)$. In other words, the transmission cost of `DSC0` is the size of the lone DSC component $r^{W0}_{i,j}(d)$ itself.

`DSC0` represents the most storage-efficient one of three implementations of M-frames using DSC discussed here; because this approach exploits correlation between successive frames (using a previous frame as side information), both the transmission and storage costs are smaller than a corresponding I-frame as shown in Fig. 4(a).

2) DSC Implementation 1 of M-Frame: The second DSC implementation (`DSC1`) of M-frame is the following: first construct multiple closed-loop differentially coded components $P_{i,j}$'s corresponding to all possible transitions from frames $F_{i-1,k}$'s of previous instant, then construct a DSC component $W^1_{i,j}$ of the *same view* j and *same instant* i from the constructed $P_{i,j}$'s. Closed-loop differentially coded components $P_{i,j}$'s are essentially P-frames encoded using motion/disparity compensation, while the DSC component is encoded using DSC codec described in [10]. See Fig. 4(c) for an example of `DSC1`.

The storage cost of `DSC1` $r^{M1}_{i,j}(\mathbf{u})$, given predictor set $\mathbf{u}$, is the size of all differentially coded components $P_{i,j}$'s plus the size of the DSC component $W^1_{i,j}$:

$$r^{M1}_{i,j}(\mathbf{u}) = \sum_{u_l \in \mathbf{u}} r^P_{i,j}(u_l) + r^{W1}_{i,j} \tag{1}$$

where $r^P_{i,j}(u_l)$ is the size of the differentially coded component $P_{i,j}$ given predictor u_l, and $r^{W1}_{i,j}$ is the size of the DSC component using predictors of the same time and view. The trans-

[6]One can of course easily generalize our 2-neighbor view switching to closest-V-neighbor view switching.

[7]All video streaming systems today offer forward playback of video, hence forward view switching is a natural extension.

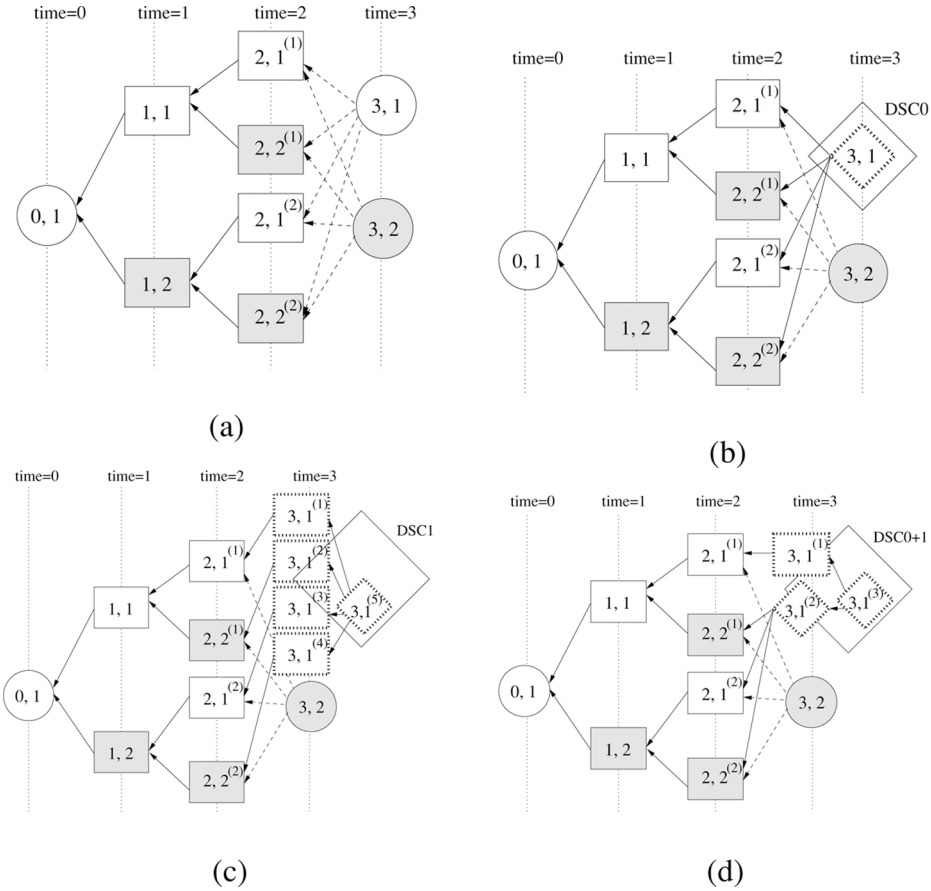


Fig. 4. Examples of DSC implementations of M-Frame for two views. I-, P-, and M-frames are denoted as solid circles, solid squares and solid diamonds, respectively. Differentially coded components and DSC components are denoted as dotted squares and dotted diamonds, respectively. (a) I- and P-frames only, (b) I-, P-frames and `DSC0`, (c) I-, P- frames and `DSC1`, (d) I-, P- frames and `DSC0+1`.

mission cost $t_{i,j}^{M1}(u_l)$ given predictor u_l can be written similarly: $r_{i,j}^{P}(u_l) + r_{i,j}^{W1}$. With the additional closed-loop differentially coded components $P_{i,j}$'s, storage cost of `DSC1` is almost surely larger than `DSC0`. However, given the DSC component and its predictors are of the same instant and view, a large correlation exists between the target and the weakest predictor. This results in a much smaller $W_{i,j}^{1}$ than $W_{i,j}^{0}$. For predictor u_l with strong correlation to target $F_{i,j}^{o}$ (e.g., same view as $F_{i,j}^{o}$), only a small differential component $P_{i,j}$ plus a small DSC component $W_{i,j}^{1}$ is required, resulting in smaller transmission cost, i.e., $r_{i,j}^{P}(u_l)+r_{i,j}^{W1} < r_{i,j}^{W0}$. This is in contrast to `DSC0`, where a large coding penalty $r_{i,j}^{W0}$ (size corresponding to the least correlated predictor in $\mathbf{u}$) must be paid for all predictors u_l's.

3) DSC Implementation `0+1` *of M-Frame:* We can combine the two discussed DSC constructs into a hybrid one (`DSC0+1`): construct multiple closed-loop differentially coded components $P_{i,j}$'s and subsequent DSC component $W_{i,j}^{1}$ as done in `DSC1`, then replace a subset of differentially coded components $P_{i,j}$'s that correspond to unlikely view switches with a single DSC component $W_{i,j}^{0}$. See Fig. 4(d) for an example where three differentially coded components $P_{3,1}$'s in Fig. 4(c) are replaced by a DSC component $W_{3,1}^{0}$. The size of a DSC component $W_{i,j}^{0}$ with multiple predictors is larger than a differentially coded component $P_{i,j}$, hence the transmission cost of each replaced decoding path is larger. On the other hand, the combined storage cost of the multiple replaced differentially coded components $P_{i,j}$'s is likely larger than a single DSC component $W_{i,j}^{0}$, hence `DSC0+1` offers a tradeoff of transmission and storage that is in between `DSC0` and `DSC1`.

IV. Problem Formulation

We now formulate our IMVS problem as a combinatorial optimization problem. We first present necessary definitions in Section IV.A. We then define IMVS formally in Section IV.B. We present an alternative expression for the Lagrangian formulation, useful during algorithm development, in Section IV.C.

A. Definitions

1) Redundant Frame Structure: Suppose we are given a multiview sequence of K views and N time switching instants,[8] and corresponding view transition probabilities $\alpha_{i,j}(k)$'s as discussed in Section III.B. For simplicity of discussion we assume for now that video starts with a single view K^o. Fig. 5(a) shows an example multiview sequence where $K = 2$, $N = 3$ and $K^o = 1$.

Given a multiview sequence, one can construct a *redundant frame structure* $\mathcal{S}$, comprised of I-, P-, and Θ different implementations of M-frames, denoted as $I_{i,j}$'s, $P_{i,j}$'s and $M_{i,j}^{\theta}$'s

[8]Given random access and view switching intervals T' and T, respectively, where $T' > T$, we have $N = \lfloor (T')/(T) \rfloor - 1$.

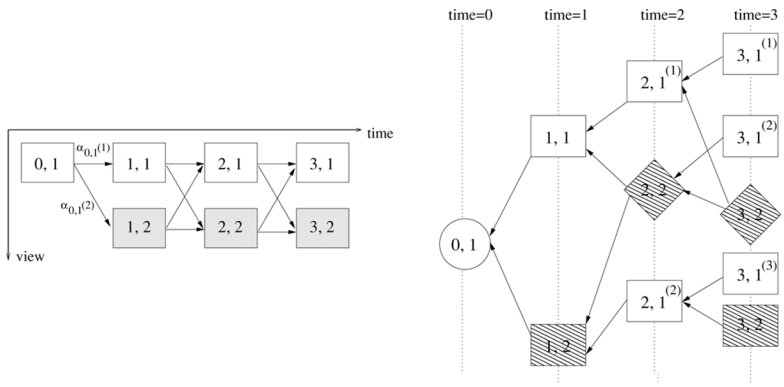


Fig. 5. Example of redundant frame structure. I-, P-, and M-frames are drawn as circles, rectangles and diamonds, respectively. A single edge from $F_{i+1,k}$ to $F_{i,j}$ in (b) means a P-frame $P_{i+1,k}$ is differentially coded using reference frame $F_{i,j}$. A set of edges from $F_{i+1,k}$ to a set of frames $F_{i,j}$'s means an M-frame $M_{i+1,k}$ is predictively coded using reference frames $F_{i,j}$'s. (a) Multiview video sequence. (b) Redundant frame structure.

TABLE I
SUMMARY OF NOTATIONS FOR IMVS PROBLEM FORMULATION

Variable	Description
N	number of switching instants
K, K^o	number of views, starting view
$B(\mathcal{S})$, $C(\mathcal{S})$	storage cost, transmission cost of structure $\mathcal{S}$
$F^o_{i,j}$, $F_{i,j}$	original picture, encoded frame of instant i and view j
$I_{i,j}$, $P_{i,j}$	I-frame, P-frame of instant i and view j
$M^\theta_{i,j}$	M-frame of implementation θ at instant i and view j
$q(F_{i,j})$	frame display probability of frame $F_{i,j}$
$r^I_{i,j}$	size of I-frame $I_{i,j}$
$r^P_{i,j}(k)$	size of P-frame $P_{i,j}$ with predictor of view k
$r^M_{i,j}(\theta, \{F_{i-1,k}\})$	storage size of M-frame of implementation θ given predictors $\{F_{i-1,k}\}$
$t(F_{i,j})$	transmission cost of frame $F_{i,j}$
$t^M_{i,j}(\theta, \{F_{i-1,k}\}, F_{i-1,k})$	transmission cost of M-frame of implementation θ given predictors $\{F_{i-1,k}\}$ and view-switching from $F_{i-1,k}$

(M-frame of implementation θ), respectively, to represent the sequence and enable IMVS. Note first that we do not specify whether the predictions for P-frames are motion- and/or disparity-compensated; our abstraction only aims to capture the dependencies among frames and not the particular encoding tool used. Note also that we specify only a storage and transmission cost here for each θ of Θ implementations of an M-frame $M^\theta_{i,j}$. Our abstraction and subsequent optimization apply generally to all implementation of M-frames, including the three discussed in Section III.C. Without loss of generality, however, we do assume that the first construct $M^0_{i,j}$ is *the most storage-efficient* among all Θ implementations—i.e., it corresponds to the left-most tradeoff point on the convex hull of transmission rate versus storage, as illustrated by `DSC0` in Fig. 1(b). A structure representing the example multiview sequence in Fig. 5(a) is shown in Fig. 5(b).

A frame structure $\mathcal{S}$ forms a *directed acyclic graph* (DAG) with I-frames as start nodes (nodes without ancestors). In Fig. 5(b), I-frame $I_{0,1}$ is the lone start node of structure $\mathcal{S}$. $\mathcal{S}$ is redundant in that an original picture $F^o_{i,j}$ can be represented by multiple frames $F_{i,j}$'s. In Fig. 5(b), $F^o_{2,1}$ is represented by two P-frames $P^{(1)}_{2,1}$ and $P^{(2)}_{2,1}$. From frame $F_{i,j}$ to a feasible view switch k, $j-1 \le k \le j+1$, without loss of generality, we assume structure $\mathcal{S}$ contains one target frame $F_{i+1,k}$ such that $F_{i,j} \leftarrow F_{i+1,k}$. This ensures structure $\mathcal{S}$ is feasible; i.e., a server can schedule transmission of $F_{i+1,k}$ when viewer requests view k after observing $F_{i,j}$.

Given view transition probabilities $\alpha_{i,j}(k)$ (which are intrinsic to the problem) and a frame structure $\mathcal{S}$, we will next formally define storage cost $B(\mathcal{S})$ and transmission cost $C(\mathcal{S})$. See Table I for a summary of notations.

2) Frame Display Probabilities: For ease of discussion, we first define *frame display probabilities* $q(F_{i,j})$'s—the probabilities that frames $F_{i,j}$'s are sent by server to be displayed at the viewer upon requests. We can compute $q(F_{i,j})$'s from front of the structure $\mathcal{S}$, I_{0,K^o}, to the back as follows using view transition probabilities, $\alpha_{i,j}(k)$:

$$q(I_{0,K^o}) = 1$$
$$q(F_{i+1,k}) = \sum_{F_{i,j} | F_{i,j} \leftarrow F_{i+1,k}} q(F_{i,j})\alpha_{i,j}(k). \qquad (2)$$

In words, starting with initial I-frame I_{0,K^o} of display probability one, the display probability $q(F_{i+1,k})$ of frame $F_{i+1,k}$ is the sum of display probabilities of previous frames $F_{i,j}$'s that can switch view to $F_{i+1,k}$, scaled by transition probabilities $\alpha_{i,j}(k)$.

3) Storage Cost: For a given frame structure $\mathcal{S}$, we can define the corresponding *storage cost*, $B(\mathcal{S})$, by simply adding the storage required by all encoded frames in $\mathcal{S}$:

$$B(\mathcal{S}) = \sum_{F_{i,j} \in \mathcal{S}} |F_{i,j}| \tag{3}$$

where $|F_{i,j}|$ denotes the storage required by frame $F_{i,j}$. For an I-frame $I_{i,j}$, rate only depends on the frame itself, and so we denote $|I_{i,j}| = r^I_{i,j}$. In contrast, the rate for a P-frame $P_{i,j}$ depends also on the frame used for prediction. Assuming $P_{i,j}$ is differentially encoded using predictor $F_{i-1,k}$, the corresponding rate will be $|P_{i,j}| = r^P_{i,j}(k)$; i.e., we assume that $|P_{i,j}|$ depends only on the *view* of predictor $F_{i-1,k}$. For example, we expect that a more accurate prediction can be obtained if $j = k$, and so in general $r^P_{i,j}(j) \leq r^P_{i,j}(k)$, for $k \neq j$.

As discussed in Section III.C, the size of an M-frame $M^\theta_{i,j}$ depends on implementation θ. Further, for a given implementation θ, size of an M-frame $M^\theta_{i,j}$ also depends on the set of frames from which observer can view-switch to $M^\theta_{i,j}$. We hence write the size of an M-frame $|M^\theta_{i,j}|$ as $r^M_{i,j}(\theta, \{F_{i-1,k}\})$—a function of both the implementation θ and the set of predictors $\{F_{i-1,k}\}$ that precedes M-frame $M^\theta_{i,j}$ in structure $\mathcal{S}$. We will assume in general that size M-frame $|M^\theta_{i,j}|$ is smaller if the predictor set is smaller:

$$r^M_{i,j}(\theta, \mathbf{u}') \leq r^M_{i,j}(\theta, \mathbf{u}) \forall \mathbf{u}', \mathbf{u}, \theta, i, j \text{ s.t. } \mathbf{u}' \subset \mathbf{u} \tag{4}$$

As previous discussed, we will also assume that the first of Θ implementations is the most storage-efficient, i.e.,

$$r^M_{i,j}(0, \{F_{i-1,k}\}) \leq \min_{0 \leq \theta < \Theta} \left\{ r^M_{i,j}(\theta, \{F_{i-1,k}\}) \right\} \quad \forall \{F_{i-1,k}\}, i, j \tag{5}$$

We will discuss how $r^I_{i,j}$'s, $r^P_{i,j}(k)$'s and $r^M_{i,j}(\theta, \{F_{i-1,k}\})$'s were generated in our experiments in Section VII.

4) Transmission Cost: Similar to storage cost, given a structure $\mathcal{S}$ we can define a corresponding transmission cost $C(\mathcal{S})$ for $\mathcal{S}$ as the sum of individual transmission costs $t(F_{i,j})$'s:

$$C(\mathcal{S}) = \sum_{F_{i,j} \in \mathcal{S}} t(F_{i,j}) \tag{6}$$

Transmission cost $t(F_{i,j})$ of frame $F_{i,j}$ depends on the frame type: if $F_{i,j}$ is I- or P-frame, it is just the frame size $|F_{i,j}|$ itself scaled by frame display probability $q(F_{i,j})$. If $F_{i,j}$ is an M-frame, then depending on which previous frame $F_{i-1,k}$ a view transition has arrived from, the transmission cost $t^M_{i,j}(\theta, \{F_{i-1,k}\}, F_{i-1,k})$ would be different. See (7) at the bottom of the page.

For example, from Fig. 5(b) the transmission cost $t(M^\theta_{3,2})$ is the sum of costs $q(P^{(1)}_{2,1})\alpha_{2,1}(2)t^M_{3,2}(\theta, \{P^{(1)}_{2,1}, M^\theta_{2,2}\}, P^{(1)}_{2,1})$ and $q(M^\theta_{2,2})\alpha_{2,2}(2)t^M_{3,2}(\theta, \{P^{(1)}_{2,1}, M^\theta_{2,2}\}, M^\theta_{2,2})$ for the two possible transitions from $P^{(1)}_{2,1}$ and $M^\theta_{2,2}$, respectively.

B. Optimization Problem Defined

We can now formally define the search for the optimal redundant frame structure for IMVS as a combinatorial optimization problem: find a structure $\mathcal{S}$, using a combination of I-, P-, and Θ implementations of M-frames, in feasible space[9] Φ that possesses the smallest possible expected transmission cost $C(\mathcal{S})$ while a storage constraint $\bar{B}$ is observed. We denote this optimization problem as IMVS*:

$$\min_{\mathcal{S} \in \Phi} C(\mathcal{S}) \text{ s.t. } B(\mathcal{S}) \leq \bar{B} \tag{8}$$

Though (8) differs from the definition in [2], a similar proof can be easily constructed to show that IMVS* is NP-hard. Given the computational difficulty of (8), we focus next on solving the corresponding Lagrangian optimization for given Lagrange multiplier λ instead[10]:

$$\min_{\mathcal{S} \in \Phi} J(\mathcal{S}) = C(\mathcal{S}) + \lambda B(\mathcal{S}) \tag{9}$$

C. Alternative Expression for Lagrangian Cost

It turns out that one of the complexities of the IMVS* problem—optimal selection of implementation given Θ available ones for a given chosen M-frame—can be solved separably without loss of optimality when the optimal frame structure is sought in (9). More precisely, when evaluating the Lagrangian objective of (9), given previous frame set $\{F_{i-1,k}\}$ as predictors and target view frame $F_{i,j}$, one can find the optimal

[9]The feasible space is the set of structures that enable an observer, upon viewing displayed frame $F_{i,j}$, to switch to a displayed frame $F_{i+1,k}$ of desired view k, where transitions have been constrained depending on desirable application characteristics, e.g., j and k may be constrained to be neighboring views.

[10]Technically, the corresponding Lagrangian should be $C(\mathcal{S}) + \lambda[B(\mathcal{S}) - \bar{B}]$. When λ and $\bar{B}$ are fixed, however, it is equivalent to (8).

$$t(F_{i,j}) = \begin{cases} q(F_{i,j}) r^I_{i,j}, & \text{if } F_{i,j} \text{ is I} - \text{frame} \\ q(F_{i,j}) r^P_{i,j}(k), & \text{if } F_{i,j} \text{ is P} - \text{frame and } F_{i-1,k} \leftarrow F_{i,j} \\ \sum_{F_{i-1,k} | F_{i-1,k} \leftarrow F_{i,j}} q(F_{i-1,k}) \alpha_{i-1,k}(j) t^M_{i,j}(\theta, \{F_{i-1,k}\}, F_{i-1,k}), & \text{if } F_{i,j} \text{ is M}-\text{frame and } \{F_{i-1,k}\} \leftarrow F_{i,j} \end{cases} \tag{7}$$

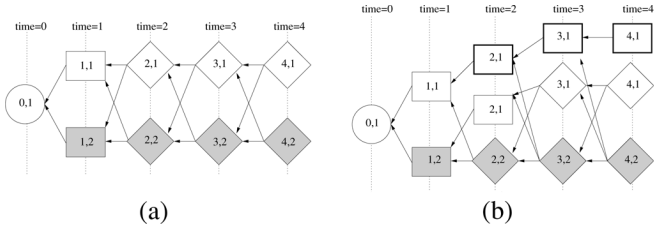


Fig. 6. Examples of frame structures generated using path-based heuristic. (a) Minimum storage frame structure. (b) Minimum storage structure plus added sub-path. Additional path in (b) is shown in thick-line rectangles.

implementation θ^* among Θ that minimizes the Lagrangian cost $l^M(F_{i,j})$ of M-frame $F_{i,j}$ as follows:

$$l^M(F_{i,j}) = \min_{0\le\theta<\Theta} \left\{ \lambda r_{i,j}^M(\theta, \{F_{i-1,k}\}) + \sum_{F_{i-1,k}|F_{i-1,k}\leftarrow F_{i,j}} q(F_{i-1,k}) \times \alpha_{i-1,k}(j) t_{i,j}^M(\theta, \{F_{i-1,k}\}, F_{i-1,k}) \right\}. \quad (10)$$

Equation (10) says that for a given M-frame with the same previous frame set $\{F_{i-1,k}\}$ and target frame $F_{i,j}$, one can find the optimal implementation θ^* for M-frame $F_{i,j}$ in Lagrangian sense, independent of other parts of the overall structure $\mathcal{S}$. Hence, the selection of the optimal implementation for M-frame can be solved simply as a sub-routine using (10) when the Lagrangian cost of the M-frame is needed during frame structure search in (9).

Given the derivation for Lagrangian cost of an M-frame in (10), we can now write the Lagrangian cost of structure $\mathcal{S}$ simply as a sum of Lagrangian costs, $l(F_{i,j})$'s, of individual frames $F_{i,j}$'s:

$$J(S) = \sum_{F_{i,j}\in\mathcal{S}} l(F_{i,j}) \quad (11)$$

$$l(F_{i,j}) = \begin{cases} (q(F_{i,j}) + \lambda)|F_{i,j}|, & \text{if } F_{i,j} \text{ is I} - \text{or P} - \text{frame} \\ l^M(F_{i,j}), & \text{if } F_{i,j} \text{ is M} - \text{frame} \end{cases} \quad (12)$$

We will focus on minimizing the Lagrangian cost of structure $\mathcal{S}$ in form (12) in later sections.

V. Heuristic Approaches

In a nutshell, good frame structures should contain the "right" mixture of redundant P-frames (for bandwidth efficiency) and M-frames (for storage efficiency) for a given Lagrangian multiplier λ. To gain intuition as to what frame structures are good, we first consider two heuristics, *path-based heuristic* and *tree-based heuristic*, to construct mixtures of redundant P-frames and M-frames, *without performing any explicit Lagrangian optimization*. We show that these heuristics, though simple, perform well when the view transition probabilities $\alpha_{i,j}(k)$'s are very small and very large, respectively. The resulting intuition will guide us to a greedy structure optimization in Section V.C. For the sake of keeping the heuristics simple, we first assume the most storage-efficient implementation $\theta = 0$ is always used when an M-frame is selected. The recursive algorithm in Section VI will consider the more general case when Θ implementations of an M-frame are available.

A. Path-Based Heuristic Structures

We first consider the case when the view transition probabilities $\alpha_{i,j}(k)$'s are very small; i.e., an observer will very likely remain in the same view throughout all potential N view switches. To construct a good redundant frame structure $\mathcal{S}$ for this case, we start with the *minimum storage frame structure*, as discussed in Section I, one where an M-frame is used at each switching point. Fig. 6 shows an example of this structure for $K = 2$ and $N = 4$. Note that this structure has no redundant representation; each picture is represented by only one decoded frame. Note also that this structure is optimal regardless of transition probabilities when λ is sufficiently large.

This minimum storage frame structure comes with high transmission cost. To reduce transmission cost by incrementally increasing redundancy in the structure (thereby increasing storage), one can do the following. Because an observer is very likely to stay in a path of the same view throughout, a simple heuristic is to locate the most likely transition $q(F_{i,j})\alpha_{i,j}(k)$ from frame $F_{i,j}$ to a view k in the structure, and add a sub-path of P-frames for this transition till the end of the structure. In Fig. 6(b), a sub-path $\{P_{2,1}, P_{3,1}, P_{4,1}\}$ is added to the transition from $P_{1,1}$ to same view 1. Note that in this case, the original M-frame $M_{2,1}$ is subsequently replaced by a P-frame, because the sub-path addition has caused $M_{2,1}$'s set of previous frame predictors to reduce to a single predictor.

We compare the performance of structures generated using the path-based heuristic to structures generated by a full optimization procedure to be described in Section VI. The performance of each structure is shown as a data point in Fig. 7(a), representing the tradeoff between the expected transmission rate and storage required to store the structure. We see that when the transition probabilities are low, the path-based heuristic indeed produced structures (`path-lo`) that closely approximates performance of the optimal (`opt-lo`), demonstrating that the use of the path abstraction indeed is appropriate. When the transition probabilities are high, however, the performance of the path-based heuristic (`path-hi`) and the optimal (`opt-hi`) are

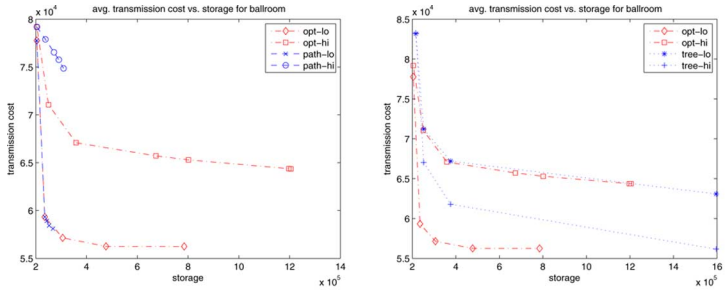


Fig. 7. Comparison of transmission rate/storage tradeoff between path-based heuristic and full optimization, and between tree-based heuristic and full optimization, respectively. (a) Rate/storage tradeoff for path-based heuristic. (b) Rate/storage tradeoff for tree-based heuristic.

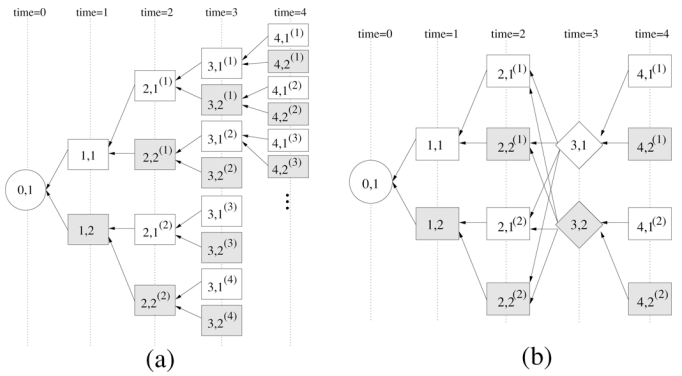


Fig. 8. Examples of frame structures generated using path-based heuristic. (a) Minimum transmission frame structure. (b) Minimum transmission structure of limited depth.

quite far apart. This motivates us to derive a different heuristic when the transition probabilities are high.

B. Tree-Based Heuristic Structure

When the view transition probabilities $\alpha_{i,j}(k)$'s are very large, e.g., when it is just as likely for an observer to stay in the same view as to switch to the neighboring view(s), we use a different heuristic. We now start from the *minimum transmission frame structure* as described in Section I, one where only P-frames are used at every switching point. This is essentially a full-tree of depth N with the lone I-frame I_{0,K^o} as the root of the tree; Fig. 8(a) shows such a full-tree of depth $N = 4$. The tree has exponential $O(3^N)$ number of frames—3 is the size of the set of neighboring views an observer can switch to at a switching point—which is not practical for large N. However, this redundant structure is optimal in Lagrangian cost when λ is sufficiently small.

As similarly done before for the minimum storage structure, to lower the storage cost by incrementally reducing the redundancy in the minimum transmission structure, we do the following. Because each transition is equally likely, the probability of arriving at a certain P-frame of a given tree depth is exactly the same as arriving at any other P-frame of the same depth. That means the Lagrangian costs of individual P-frames at the same tree depth are roughly the same (assuming all P-frames are of roughly the same size). On the other hand, as P-frames of one tree depth transition to P-frames of the next tree depth, the total transmission cost of P-frames of the next tree depth remains the same (still one P-frame per transition), while the storage has increased by factor 3. That means Lagrangian cost of the P-frame set at deeper tree depth are strictly more expensive than cost of the P-frame set at shallower tree depth. A reasonable heuristic then is to eliminate P-frames of tree depth $\geq d$ by replacing P-frames at tree depth d with M-frames. More specifically, starting with the minimum transmission structure, one can replace P-frames at depth $N/2+1$ with M-frames for one structure—cutting the tree into halves—resulting in smaller storage but higher transmission cost. For more tradeoff between storage and transmission, one can replace P-frame at depth $N/3+1$ and $2N/3+1$ with M-frames for another structure—cutting the tree into thirds, etc. Fig. 8(b) shows a structure where P-frames at depth $N/2+1=3$ of the full free in Fig. 8(a) are replaced by M-frames.

We now compare the performance of structures generated using the tree-based heuristic to that of the full optimization procedure, shown in Fig. 7(b). We see that when the transition probabilities are high, the tree-based heuristic produced structures (`tree-hi`) approximate that of the optimal (`opt-hi`), proving empirically that the use of the tree abstraction is indeed appropriate. On the other hand, when the switching probabilities are low, the performance of the tree-based heuristic (`tree-lo`) compared to the optimal (`opt-lo`) is inferior. This suggests that an optimal structure must use a combination of paths and trees corresponding to different transition probabilities to optimize the rate/storage tradeoff. We discuss such an optimization next.

C. Greedy Structure Optimization

In this section, we derive a greedy structure optimization based on our observations of Sections V.A and V.B. While in Sections V.A and V.B we started with a complete structure (minimum storage or minimum transmission, respectively) and then incrementally made heuristic modifications to the structure to lower transmission or storage cost, here we iteratively build a structure from front to back, i.e., starting with an initial I-frame I_{0,K^o}, we construct frames $F_{1,j}$'s at view switching instant 1, then frames $F_{2,j}$'s at switching instant 2 and so on. At each switching instant i, the key question we need to answer is: given frames $F_{i-1,k}$'s constructed at previous switching instant $i-1$, how do we optimally construct frames $F_{i,k}$'s to minimize (12) for given Lagrange multiplier λ?

Suppose that given constructed frames $F_{i-1,k}$'s at switching instant i, we identify the n_o most likely view switches from $F_{i-1,k}$'s, and construct a P-frame to fulfill each of the n_o view switches. We then construct K M-frames for K total views and assign the remaining view switches from $F_{i-1,k}$'s to them. Let $P(n_o)$ be the sum of probabilities of these n_o most likely view switches. If a P-frame at instant i has average size r_i^P, and an M-frame has average transmission and storage cost of t_i^M and r_i^M respectively, then we can find the optimal n_o that minimizes the Lagrangian cost at instant i as follows:

$$\min_{n_o} \left\{ P(n_o) r_i^P + (1-P(n_o)) t_i^M + \lambda \left(n_o r_i^P + K r_i^M \right) \right\}. \quad (13)$$

We can now grow a structure $\mathcal{S}$ from front to back greedily starting from the initial I-frame I_{0,K^o} using (13) at every switching instant. Note that the algorithm is greedy in that at each instant i, the locally optimal ith "slice" of the structure is chosen with no lookahead into future switching instants $k > i$.

We now compare the performance of structures generated using the greedy optimization to that of the full optimization procedure, shown in Fig. 9. We see that the greedy optimization produced structures (`heu-hi, heu-mid, heu-lo`) approximate that of the optimal (`opt-hi, opt-mid, opt-lo`) on most data points whether the view transition probabilities are high, medium or low. There are, however, points that are far from the optimal due to the greedy nature of (13). We next investigate an optimal algorithm, extending on the greedy optimization here to take into account future switching instants as well via recursion, generating the best structures possible in a Lagrangian sense.

VI. Recursive Optimization

We now extend the previous greedy structure optimization in Section V.C to a recursive algorithm to generate best possible structures for `IMVS`*. We first derive an optimal algorithm for `IMVS`* with exponential running time in Section V.C. We then discuss methods to reduce its complexity for practical use in Sections VI.B and VI.C.

A. Optimal Algorithm

We first provide an overview of the algorithm to develop intuition. Similar to the greedy algorithm in Section V.C, we build a frame structure from front to back one "slice" at a time, starting

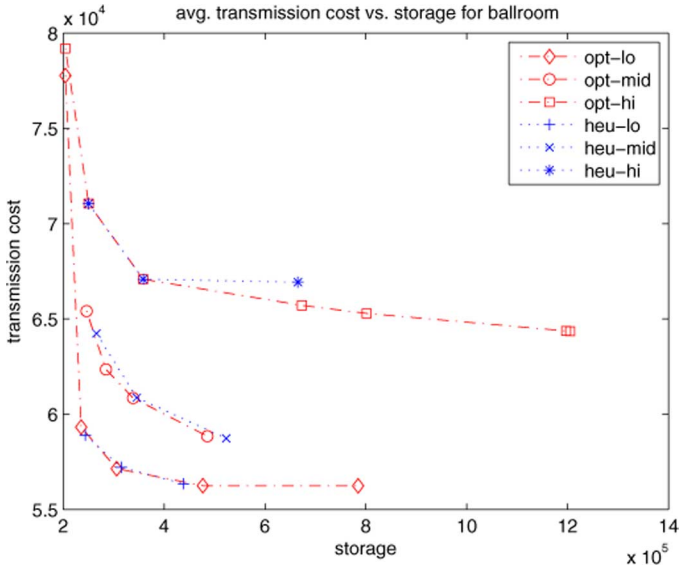


Fig. 9. Comparison of transmission rate/storage tradeoff between greedy optimization and full optimization.

with an initial I-frame I_{0,K^o}. We define $\mathcal{S}_i$ as a *partial structure* constructed from I_{0,K^o} up till switching instant i; $\mathcal{S}_0$ is simply $\{I_{0,K^o}\}$. For given partial structure $\mathcal{S}_{i-1}$, each frame $F_{i-1,k}$ in $\mathcal{S}_{i-1}$ has a display probability $q(F_{i-1,k})$ computable front-to-back using (2). Each frame $F_{i-1,k}$ will switch to view j with probability $q(F_{i-1,k})\alpha_{i-1,k}(j)$ either via a P-frame $P_{i,j}$ predicted from $F_{i-1,k}$, or via an M-frame $M_{i,j}$, each with different local Lagrangian costs at instant i. An optimal structure $\mathcal{S}_i$ at switching instant i given $\mathcal{S}_{i-1}$ has the smallest sum of: i) local Lagrangian costs at instant i, and ii) future (recursive) minimum Lagrangian costs stemming from $\mathcal{S}_i$.

We first define $\mathbf{s}_{i,j}$ as the *local structure* at instant i for view j; each $\mathbf{s}_{i,j}$ receives view transitions from frames $F_{i-1,k}$'s in $\mathcal{S}_{i-1}$ to frames $F_{i,j}$'s of view j. $\mathbf{s}_{i,j}$ can be a set of M-frames $M_{i,j}$'s, a set of P-frames $P_{i,j}$'s each coded with different reference frames $F_{i-1,k}$'s in $\mathcal{S}_{i-1}$, or a combination of both. Partial structure at the next instant $\mathcal{S}_i$ is simply a concatenation of previous partial structure $\mathcal{S}_{i-1}$ and chosen local structures $\mathbf{s}_{i,j}$'s for all view j's with appropriate coding dependencies attached.

We now define the minimum Lagrangian cost $L_i(\mathcal{S}_{i-1})$ from switching instant i till last switching instant N, given partial structure $\mathcal{S}_{i-1}$, as the sum of *local Lagrangian cost* $l_{i,j}(\mathcal{S}_{i-1}, \mathbf{s}_{i,j})$ of all view j's at instant i, and recursive cost $L_{i+1}(\mathcal{S}_i)$, where structure $\mathcal{S}_i$ is a concatenation of $\mathcal{S}_{i-1}$ and $\mathbf{s}_{i,j}$'s as discussed:

$$L_i(\mathcal{S}_{i-1}) = \min_{\mathbf{s}_{i,j}} \left\{ \sum_{j=1}^{K} l_{i,j}(\mathcal{S}_{i-1}, \mathbf{s}_{i,j}) + L_{i+1}(\mathcal{S}_i) \right\}. \quad (14)$$

Note that all feasible local structures $\mathbf{s}_{i,j}$'s must be searched exhaustively to find the global minimum in (14).

Local Lagrangian cost $l_{i,j}(\mathcal{S}_{i-1}, \mathbf{s}_{i,j})$ for view j can be divided into costs of M-frames $\mathbf{s}_{i,j}^M$ and of P-frames $\mathbf{s}_{i,j}^P$ in local structure $\mathbf{s}_{i,j}$, where $\mathbf{s}_{i,j}^M \cup \mathbf{s}_{i,j}^P = \mathbf{s}_{i,j}$, to receive transitions to view j from previous frames $F_{i-1,k}$'s:

$$l_{i,j}(\mathcal{S}_{i-1}, \mathbf{s}_{i,j}) = l_{i,j}^M \left(\mathcal{S}_{i-1}, \mathbf{s}_{i,j}^M \right) + l_{i,j}^P \left(\mathcal{S}_{i-1}, \mathbf{s}_{i,j}^P \right). \quad (15)$$

The local Lagrangian cost $l^M_{i,j}(\mathcal{S}_{i-1}, \mathbf{s}^M_{i,j})$ is simply the sum of individual Lagrangian cost of each M-frame $M_{i,j}$ in $\mathbf{s}^M_{i,j}$ using (10):

$$l^M_{i,j}\left(\mathcal{S}_{i-1}, \mathbf{s}^M_{i,j}\right) = \sum_{M_{i,j} \in \mathbf{s}^M_{i,j}} l^M(M_{i,j}). \tag{16}$$

As discussed in Section IV.C, the optimal $\theta^*$of Θ implementations of an M-frame given predictor set $\{F_{i-1,k}\}$ and encoding target $F_{i,j}$ is found locally using (10) without loss of global optimality.

The local Lagrangian cost $l^P_{i,j}(\mathcal{S}_{i-1}, \mathbf{s}^P_{i,j})$ for P-frames $P^{(x)}_{i,j}$'s, on the other hand, is the sum of each transition probability into a unique $P^{(x)}_{i,j}$ plus λ times the size of $P^{(x)}_{i,j}$:

$$l^P_{i,j}(\mathcal{S}_{i-1}, \mathbf{s}^P_{i,j}) = \sum_{F_{i-1,k} | F_{i-1,k} \leftarrow P^{(x)}_{i,j}, P^{(x)}_{i,j} \in \mathbf{s}^P_{i,j}} [q(F_{i-1,k})\alpha_{i-1,k}(j) + \lambda] r^P_{i,j}(k). \tag{17}$$

Display probability for M-frame $M_{i,j}$ is the sum of display probabilities of previous predictor frames $F_{i-1,k}$'s scaled by transition probabilities $\alpha_{i-1,k}(j)$'s:

$$q(M_{i,j}) = \sum_{\forall F_{i-1,k} | F_{i-1,k} \leftarrow M_{i,j}} q(F_{i-1,k})\alpha_{i-1,k}(j), \quad M_{i,j} \in \mathbf{s}^M_{i,j}. \tag{18}$$

We claim that a call $L_0(\mathcal{S}_0)$, $\mathcal{S}_0 = \{I_{0,K^o}\}$, using (14) solves the Lagrangian (9) optimally. We state this result formally as a theorem below and then provide a simple proof.

Theorem 1: Using initial argument $\mathcal{S}_0 = \{I_{0,K^o}\}$, optimization (14) returns an optimal solution to IMVS*.

Proof: We prove this by induction. For the base case of recursion depth $N = 1$, it is clear that starting from any initial structure slice $\mathcal{S}_0$ at instant 0, (14) finds the optimal solution by exhaustively searching all feasible local structures $\mathbf{s}_{1,j}$'s with no recursive term $L_2(S_1)$. For the inductive case, suppose (14) is optimal for recursive depth N; i.e., $L_i(\mathcal{S}_{i-1})$ yields the optimal structure of depth N given $\mathcal{S}_{i-1}$. For recursive depth of $N+1$, we can do the following. Given initial slice $\mathcal{S}_0$, exhaustively search all feasible set of $\mathbf{s}_{1,j}$'s, and for each set, evaluate its local Lagrangian cost plus recursive cost $L_2(\mathcal{S}_1)$, where $\mathcal{S}_1$ is simply a combination of $\mathcal{S}_0$ and set of $\mathbf{s}_{1,j}$'s. By assumption, $L_2(\mathcal{S}_1)$ returns optimal structure of depth N given $\mathcal{S}_1$. Given $\mathbf{s}_{1,j}$'s are exhaustively searched, this procedure gives the optimal solution for depth $N+1$. Note that this is exactly how (14) performs its optimization for depth $N+1$. Hence, the inductive case is also proven, and (14) is globally optimal. □

Though optimal, (14) is nevertheless exponential in running time; each of the n potential transitions into instant i and view j, can be mapped to either a P- or an M-frame, hence there are at least $\Omega(2^n)$ local structures $\mathbf{s}_{i,j}$'s for view j alone. For general n, exhaustive search is simply not tractable, and we will hence discuss strategies to reduce the overall complexity next.

B. Complexity Reduction 1: Simplify Local Structure Selection

To reduce complexity in (14), in this section we derive a methodology to systematically reduce the number of local structures $\mathbf{s}_{i,j}$'s that are searched at each function invocation $L_i(\mathcal{S}_{i-1})$ in (14) to reduce complexity. Towards that goal, we first state the following result:

Lemma 1: If local Lagrangian cost of switching from X frames $F_{i-1,k}$'s in partial structure $\mathcal{S}_{i-1}$ to a single M-frame $M_{i,j}$ in local structure $\mathbf{s}_{i,j}$ at instant i and view j is no larger than corresponding local Lagrangian costs of switching to X constructed P-frames $P^{(x)}_{i,j}$'s, $x = 1, \ldots, X$, then resulting global Lagrangian cost of switching to $M_{i,j}$ is also no larger than resulting global Lagrangian costs[11] of switching to $P^{(x)}_{i,j}$'s.

In other words, if the local cost of merging X transitions to an M-frame is no larger than the cost of X corresponding P-frames, then surely the merging decision of using M-frame is also globally optimal. The proof for Lemma 1 is stated as follows.

Proof of Lemma 1: We show that for any set of subtrees of P-frames stemming from constructed $P^{(x)}_{i,j}$'s in $\mathbf{s}_{i,j}$, one can construct a corresponding subtree of P-frames stemming from $M_{i,j}$ in $\mathbf{s}_{i,j}$ such that the resulting global Lagrangian cost is no larger. For given set of subtrees $\mathcal{T}^{(x)}$'s below $P^{(x)}_{i,j}$'s, we construct corresponding subtree $\mathcal{U}$ for $M_{i,j}$ by taking the *union* of $\mathcal{T}^{(x)}$'s; i.e., for each P-frame $P_{y,z}$ in $\mathcal{T}^{(x)}$, we add a corresponding $P_{y,z}$ of the same size to $\mathcal{U}$ if $\mathcal{U}$ does not already have $P_{y,z}$ constructed. (See Fig. 10 for an example.) First, we know $|\mathcal{U}| \leq |\mathcal{T}^{(1)}| + \cdots + |\mathcal{T}^{(X)}|$ since union of sets is no larger then sum of sets. Further, transmission cost from $P^{(x)}_{i,j}$ to any frame $P_{y,z}$ in $\mathcal{T}^{(x)}$ is exactly the same from $M_{i,j}$ to its corresponding $P_{y,z}$ in $\mathcal{U}$, hence the transmission cost of using $\mathcal{U}$ over $\mathcal{T}^{(x)}$'s can be no worse. Since by assumption $M_{i,j}$ alone induces no larger local Lagrangian cost than $P^{(x)}_{i,j}$'s, we conclude $M_{i,j}$ and $\mathcal{U}$ also induce no larger global cost than $P^{(x)}_{i,j}$'s and $\mathcal{T}^{(x)}$'s. □

To make good use of Lemma 1, we construct the following greedy heuristic to simplify the selection of local structure $\mathbf{s}_{i,j}$ for view j given invocation $L_i(\mathcal{S}_{i-1})$. First, we create an M-frame $M_{i,j}$ to which all viable switches to view j from frames $F_{i-1,k}$'s in $\mathcal{S}_{i-1}$ would transition. Then we identify the frame $F^*_{i-1,k}$ in $\mathcal{S}_{i-1}$, whose corresponding local Lagrangian cost associated with transition to M-frame $M_{i,j}$, $F^*_{i-1,k} \rightarrow M_{i,j}$, would decrease the most if $F^*_{i-1,k}$ transitions to a P-frame $P^*_{i,j}$, $F^*_{i-1,k} \rightarrow P^*_{i,j}$, instead. We create $P^*_{i,j}$, reassign $F^*_{i-1,k}$ to transition to $P^*_{i,j}$, and compute global Lagrangian cost using (14) to test global optimality. The process repeats to find the most locally beneficial $F^*_{i-1,k}$ in $\mathcal{S}_{i-1}$ and add corresponding P-frame $P^*_{i,j}$ to $\mathbf{s}_{i,j}$ until no cost-reducing $F^*_{i-1,k}$ remains.

From the description of the greedy heuristic, it is clear that at most $n+1$ local structures $\mathbf{s}_{i,j}$'s, where n is the number of viable transitions to view j from frames $F_{i-1,k}$'s in $\mathcal{S}_{i-1}$, are recursively computed for global optimality using (14). The stopping condition—that no more P-frame $P_{i,j}$ can be constructed to lower local Lagrangian cost—is guaranteed to be optimal within the set of $n+1$ local structures given Lemma 1. (The corollary of Lemma 1 is that local Lagrangian cost cannot be further decreased by constructing more P-frames means that global Lagrangian cost *also* cannot be further decreased by constructing more P-frames.) Further, it turns out that restricting testing of

[11]This lemma assumes that each P-frame $P_{i,j}$ encoded using predictor $F_{i-1,k}$ is of size $r^P_{i,j}(k)$ as described in Section IV.A.

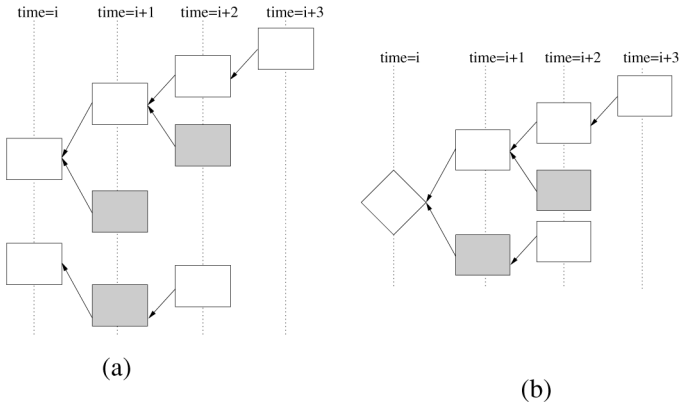


Fig. 10. Example of union construction of M-frame subtree. (a) Multiple P-frames and subtrees. (b) M-frame and subtree.

1) For each view j, create M-frame $M_{i,j}$, and assign all viable transitions from frames in $\mathcal{S}_{i-1}$ to view j via $M_{i,j}$. Compute global Lagrangian cost using (14).
2) Identify frame(s) $F^*_{i-1,k} \in \mathcal{S}_{i-1}$ with corresponding view-j transition $F^*_{i-1,k} \rightarrow M_{i,j}$ that would decrease local Lagrangian cost the most by switching to new P-frame $P^{(x)}_{i,j}$, or to new M-frame $M'_{i,j}$, instead.
3) If such frame(s) $F^*_{i-1,k}$ exists, create corresponding $P^*_{i,j}$ or $M'_{i,j}$, reassign $F^*_{i-1,k}$, and compute global cost using (14).
4) If such frame(s) $F^I_{i-1,k}$ does not exist, Stop.

Fig. 11. Complexity reduction 1: Simplify selection of local structure $\mathbf{s}_{i,j}$'s during invocation $L_i(\mathcal{S}_{i-1})$.

0) Select optimization depth $h \in \mathcal{I}$.
1) Initialize sliding index $m = 0$, and partial frame structure $\mathcal{S}^o$ to be starting $\{I_{0,K^o}\}$.
2) Using $\mathcal{S}^o$ as argument, perform optimization (14) up to maximum depth h; i.e., extend partial structure $\mathcal{S}^o$ to $\mathcal{S}^*$, $\mathcal{S}^o \subset \mathcal{S}^*$, in instants $i = 1 + m, \ldots, \min(h + m, N)$.
3) Commit frames of the first optimization instant $1 + m$ (first structure "slice") in derived structure $\mathcal{S}^*$ in step 2 to $\mathcal{S}^o$.
4) If $m < N - 1$, increment m by 1. Goto step 2. Else, stop and return $\mathcal{S}^o$.

Fig. 12. Complexity reduction 2: Sliding window of maximum recursion depth h during invocation $L_i(\mathcal{S}_{i-1})$.

global optimality to only these $n + 1$ local structures remains globally optimal if the only implementation of M-frame is an I-frame implementation, and frame sizes of I- and P-frames do not change across time and across view. The intuition is that given the difference in frame sizes between M- and P-frames remains the same across time, there is nothing extra to be gained in future instants $> i$ that cannot already be obtained locally at instant i. See Appendix I for a detailed proof of this special case.

More generally, one can create another M-frame $M'_{i,j}$ instead of P-frame $P_{i,j}$ when reassigning $F_{i-1,k}$'s in $\mathcal{S}_{i-1}$ to transition to newly created frames to lower local Lagrangian cost. For completeness, Fig. 11 describes the general complexity reduction procedure 1 to select local structures $\mathbf{s}_{i,j}$'s during invocation $L_i(\mathcal{S}_{i-1})$.

C. Complexity Reduction 2: Sliding Window of Recursion Depth h

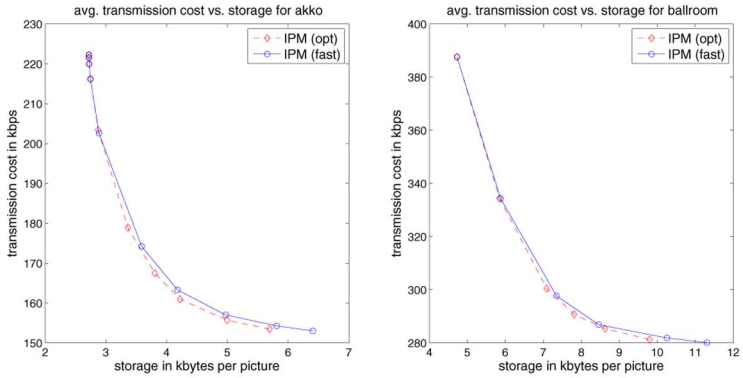
Though the number of next-level recursive calls from each invocation is now linear due to procedure in Section VI.B, the number of total calls is still exponential in the depth of the recursion N. Hence, we propose a *sliding-window* strategy of maximum lookahead depth $h < N$ as shown in Fig. 12. Essentially, the algorithm performs optimization (14) up to maximum depth h starting with initial I-frame, $\mathcal{S}^o = \{I_{0,K^o}\}$, as argument, adds the first "slice" of the optimized partial frame structure to $\mathcal{S}^o$, and then use updated $\mathcal{S}^o$ as the new argument to solve (14) again up to maximum depth h, and so on. We will show in Section VII that the sliding-window strategy produces good approximated results.

VII. Experimentation

A. Experimental Setup

To gather multiview video data for our experiments, we used three neighboring views[12] from 100-frame sequences `akko&kayo` and `ballroom` at 320 × 240 resolution and 30 fps and 25 fps, respectively. The distance between neighboring cameras for `akko&kayo` and `ballroom` were 5 cm and 20 cm respectively; using both data sets can test our algorithm with multiview sequences of different characteristics. To generate data for `DSC0` and `DSC1` implementations of M-frames, we used the algorithm in [11], which is based on H.263 tools (e.g., half-pel motion estimation). We selected QP such that

[12]We believe the general trends in the tradeoff between transmission rate and storage will remain the same for larger number of views, due to our assumption that an observer can only switch to a neighboring view. Hence, any intermediate view (i.e., neither the left-most nor right-most view) will behave like the middle view of the three views used in our experiments.



(a) (b)

Fig. 13. Tradeoffs between expected transmission rate and storage per picture for $T' = 24$, $T = 3$, comparing performance between full recursive optimization (`opt`) and optimization with finite sliding window of size h (`fast`). (a) `akko&kayo`, (b) `ballroom`.

I-, P-, and DSC-frames were reconstructed to the same quality (around 34 dB).

More precisely, for each sequence, we generated encoding rates $r^I_{i,j}$'s, $r^P_{i,j}(k)$'s, as inputs to our optimization algorithm as follows. Each $r^I_{i,j}$ was obtained when we encoded picture $F^o_{i,j}$ as I-frame. $r^P_{i,j}(j)$'s were obtained when we encoded picture $F^o_{i,j}$ in a single-view sequence where each $F_{i,j}$ was motion-compensated from $F_{i-1,j}$. For $r^P_{i,j}(k)$'s, $k \neq j$, we first generated four zigzagged sequences as follows:

1) $z_{lc} = \{I_{0,1}, P_{1,2}, P_{2,1}, P_{3,2}, \ldots\}$.
2) $z_{cr} = \{I_{0,2}, P_{1,3}, P_{2,2}, P_{3,3}, \ldots\}$.
3) $z_{cl} = \{I_{0,2}, P_{1,1}, P_{2,2}, P_{3,1}, \ldots\}$.
4) $z_{rc} = \{I_{0,3}, P_{1,2}, P_{2,3}, P_{3,2}, \ldots\}$.

For each $r^P_{i,j}(k)$, we simply located the zigzagged stream z that contained the sub-sequence $\{F_{i-1,k}, P_{i,j}\}$ and assigned the corresponding coding rate.

For transition probabilities $\alpha_{i,j}(k)$'s, we assume frame $F_{i,1}$ remains at the same view $F_{i+1,1}$ with probability $1-\alpha$, and transitions to neighboring views $F_{i+1,0}$ and $F_{i+1,2}$ with probability $\alpha/2$ each. Frame $F_{i,0}(F_{i,2})$ transition to the single neighboring view $F_{i,1}$ with the same probability α.

B. Experimental Results

1) Approximation Using Sliding-Window Strategy: We first examine how closely the fast sliding-window strategy (`fast`) discussed in Section VI.C approximates the original algorithm without the sliding window (`opt`). For random access period of $T' = 24$ and switch period of $T = 3$—hence optimization window depth of $N = \lfloor (24)/(3) \rfloor - 1 = 7$—we generated tradeoff points between expected transmission rate and storage required per picture using `opt`, shown in Fig. 13 for the two test sequences. Lagrangian multiplier λ was swept from 0.01 to 10.24 to induce different tradeoffs. We also generated tradeoff points for `fast` when the lookahead depth was $h = 5$. We see that the convex hull of `fast` closely resembled that of `opt`, demonstrating that the sliding-window strategy performs numerically close to the original in practice. For this case when $N = 7$, we found experimentally that performance as a function of lookahead improves noticeably up to $h = 5$, beyond which the improvement is marginal. We conjecture that the reason is that sufficient number of frame structures are already considered for optimization when $h = 5$ (including the "full tree" frame structure with K^5 leaf nodes). Hence, lookahead $h = 5$ is used for the rest of the results section.

2) Algorithm Performance Comparison 1: Different Transition Probabilities: For algorithm comparison, using first a random access period of $T' = 30$ frames and switching period of $T = 3$ frames, we plotted the tradeoff points for our algorithm using I- and P-frames only (`IP`) and I-, P- and M-frames (`IPM`), for the `akko&kayo` and `ballroom` sequences in Fig. 14(a) and (b), respectively. To induce different transition probabilities $\alpha_{i,j}(k)$'s, α was set to 0.1 and 0.2 for two different trials. We also plotted the performance of the random access approach (`RA-I`) where I-frames were inserted at all switching points for view switching.

We first see that `RA-I` was represented by a single point; because the I-frame insertion algorithm was fixed, it had one corresponding fixed transmission and storage cost, and therefore could not take advantage of extra storage capacity if available to lower transmission cost.

Second, we see that our algorithm found tradeoff points in `IP`and `IPM` that were to the lower left of `RA-I`; i.e., our algorithm generated frame structures that offered lower transmission rates than *and* required smaller storage than `RA-I`. This is particularly noticeable when M-frames were used in addition to I- and P-frames, where our algorithm generated frame structures with 38% and 20% smaller expected transmission rate respectively for the two sequences, while requiring comparable storage.

Third, unlike `RA-I`, our algorithm offered a range of tradeoff points to take advantage of extra storage when available to further decrease expected transmission rate. In particular, at twice the storage of `RA-I`, our algorithm generated frame structures with 45% and 32% smaller expected transmission rate than `RA-I` for `akko&kayo` and `ballroom`, respectively. The performance improvement of `IPM` over `RA-I` was more dramatic for `akko&kayo` than `ballroom`; we conjecture that this was due to relatively smaller sizes of P-frames compared to I-frames in `akko&kayo`, as a result of the closely spaced cameras.

Fourth, we see that using M-frames (`IPM`) in addition I- and P-frames (`IP`) did generate better tradeoff points at all storage

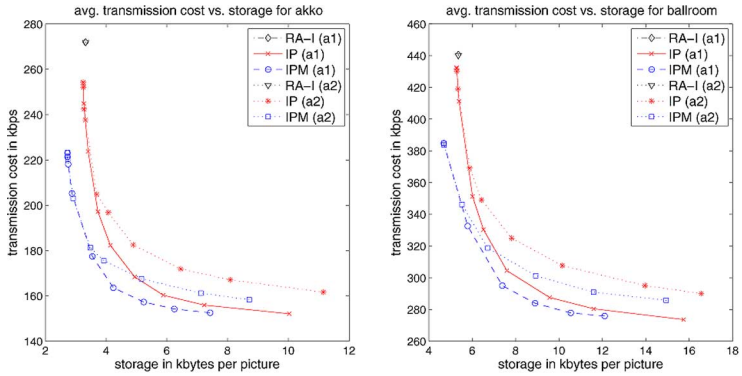


Fig. 14. Tradeoffs between expected transmission rate and storage per picture for $T' = 30$, $T = 3$, comparing performance among random acesss I-frame insertion (RA-I), I- and P-frames only (IP), and I-, P-, and M-frames (IPM) for different view switching probability α. (a) akko&kayo, (b) ballroom.

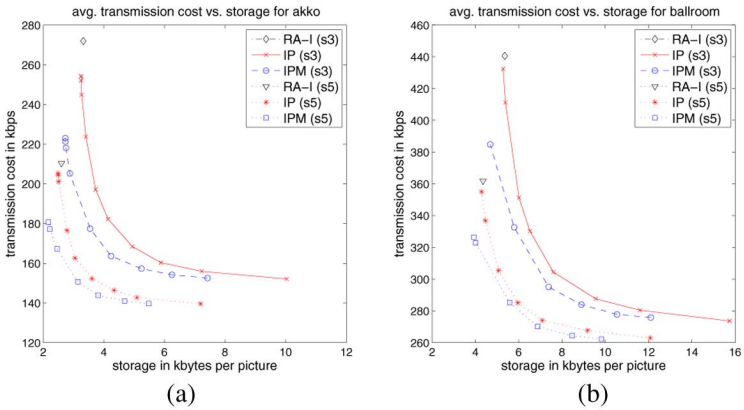


Fig. 15. Tradeoffs between expected transmission rate and storage per picture for $T' = 30$, $\alpha = 0.1$, comparing performance among random acesss I-frame insertion (RA-I), I- and P-frames only (IP), and I-, P-, and M-frames (IPM) for different view switching period T. (a) akko&kayo, (b) ballroom.

requirements. The differences are largest at stringent storage requirement, when M-frames are used more often then redundant P-frames.

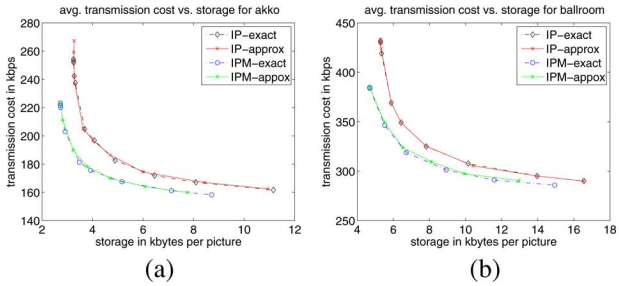
Finally, we see that as the switching probabilities increased, the performance of our algorithm deteriorated gradually (curves of IP and IPM shifted to the upper-right). This is expected, as more transitions mean more inter-view predicted frames are required, resulting in higher transmission rates. The improvement over RA-I, however, remains dramatic.

3) Algorithm Performance Comparison 2: Different View Switching Periods: We repeated the same experiment again, where this time we varied the view switching period T from 3 to 5. The results are shown in Fig. 15(a) and (b) for the two test sequences. We observe that larger switching period means smaller transmission rate and storage in general; this is intuitive since staying in the same view more often means same-view predicted P-frames that are very bandwidth-efficient were used more extensively between transitions. Nevertheless, our algorithm did generate similar patterns of tradeoff points that are superior to RA-I, and using M-frames did perform better than optimization using I- and P-frames only.

4) Algorithm Performance Comparison 3: Frame Rate Approximation: One source of complexity in our algorithms is the acquisition of data—specifically, frame sizes for I-, P and different implementations of M-frames—for input into our optimization. To reduce the burden of data collection, we investigated the use of *estimated data* as representatives for the entire data set, in particular, use statistics of frames in the first switching instant for optimization of the entire sequence. For example, use of P-frame size $r^P_{1,j}(k)$ of instant 1 and view j for all $r^P_{i,j}(k)$'s, $\forall i$. Fig. 16 shows the performance comparison of using approximated rates versus exact rates for both akko&kayo and ballroom sequences. Our results show that for both cases when M-frames was (IPM) and wasn't used (IP) in the optimization, using estimated data gave almost as good a performance. Hence, in a scenario where encoding complexity is a concern (though the proposed system is for stored-and-playback applications), one can choose to use estimated data for optimization with minor loss in performance.

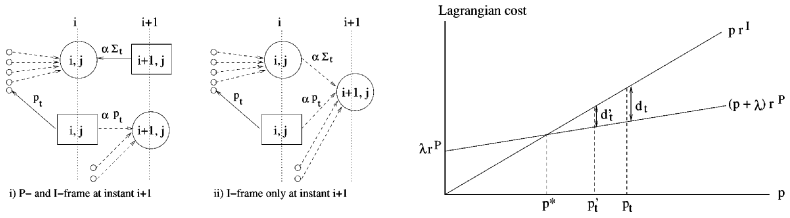
VIII. Conclusion

In this paper, we motivated the need for an interactive multiview video streaming (IMVS) system, where an observer can periodically send feedbacks to the server requesting desired views out of many available. In response, the server will send corresponding pre-encoded video data to the observer for decoding and display without interrupting forward video playback. Observing that one can trade off expected transmission rate with a modest increase in storage when designing the pre-encoded frame structure, we formulated a combinatorial



(a) (b)

Fig. 16. Tradeoffs between expected transmission rate and storage per picture for $T' = 30$, $\alpha = 0.2$, comparing performance between I- and P-frames only (`IP`) and I-, P-, and M-frames (`IPM`) using frame rate approximation. (a) `akko&kayo`, (b) `ballroom`.



(a) (b)

Fig. 17. Structure constructions and corresponding Lagrangian costs for proof of optimality of greedy procedure. (a) Possible structures at instant $i + 1$. (b) Lagrangian cost versus transition probability.

optimization problem, where the optimal structure contains the best mixture of I-frames (for random access), P-frames (for low transmission rate) and merge or M-frames (for low storage), trading off transmission rate with storage. We presented fast heuristic-based strategies, as well as a near-optimal recursive algorithm as potential solutions to the optimization problem. Our results show that when compared to the simple I-frame insertion strategy, our proposed frame optimization can reduce expected transmission rate by up to 45% at twice the storage.

APPENDIX

We prove here that the greedy procedure described in Section VI is optimal when the only implementation of M-frame is an I-frame, and the sizes of P- and I-frames do not change across time and across view. Let r^I and r^P be the sizes of I- and P-frames respectively for all views and all time. Under these assumptions, the greedy procedure will iteratively select transitions with the largest probabilities for P-frame constructions, until the resulting local Lagrangian cost exceeds that of the previous iteration.

We prove the optimality of the greedy procedure by contradiction. Suppose at instant i, an alternative procedure selects at iteration t a transition with probability p'_t that is not the transition with the tth largest probability p_t, i.e., $p'_t < p_t$. Suppose further that this alternative procedure eventually leads to a smaller overall Lagrangian cost compared to the greedy procedure. First, due to the different transition selection for P-frame construction at iteration t, the greedy procedure has a smaller local Lagrangian cost over the alternative one at this instant i by $(p_t - p'_t)(r^I - r^P)$, as shown in Fig. 17(b). The difference in transition selection also leads to different display probabilities for the resulting M-frame (I-frame), Σ_t and Σ'_t, for the greedy and alternative procedure, respectively, where $\Sigma'_t - \Sigma_t = p_t - p'_t$. Note also that $\Sigma'_t \geq p_t$ and $\Sigma_t \geq p'_t$.

At instant $i + 1$, the transition probabilities to the same view j from M-frame and the P-frame created from iteration t of instant i are $\alpha\Sigma_t$ and αp_t respectively for the greedy procedure, and $\alpha\Sigma'_t$ and $\alpha p'_t$ respectively for the alternative procedure. This difference in transition probabilities at instant $i + 1$ due to the alternative procedure can lead to a lower local Lagrangian cost than the greedy procedure for same view j at instant $i + 1$, *if* the optimal structure contains a P-frame for the M-frame in instant i and an I-frame for the P-frame constructed in iteration t in instant i, as illustrated in Fig. 17(a)(i). We call this structure `P + I`. Any other structure for the alternative procedure would lead to the same or worse local Lagrangian cost than a comparable structure for the greedy procedure. In contrast, the optimal structure for the greedy procedure at instant $i + 1$ can be either `P + I`, or a structure where only an I-frame is constructed to handle both transitions from M-frame and P-frame of tth largest probability in instant i—we call this structure `I-only` (see Fig. 17(a)(ii) for an illustration). We bound the gain in local Lagrangian cost by the alternative procedure for the two cases as follows.

Suppose the optimal structure for the greedy procedure at instant $i+1$, like the alternative procedure, is also `P + I`. Because the alternative procedure leads to a larger transition probability $\alpha\Sigma'_t > \alpha\Sigma_t$ into the P-frame in instant $i + 1$, it will induce a smaller local Lagrangian cost at instant $i + 1$ compared to the greedy procedure. The magnitude of the gain, however, is bounded by $(\alpha\Sigma'_t - \alpha\Sigma_t)(r^I - r^P) = \alpha(p_t - p'_t)(r^I - r^P)$.

Suppose now the optimal structure for the greedy procedure at instant $i + 1$ is instead `I-only`. First, that means that it is more costly to create a P-frame for transition of probability $\alpha\Sigma_t$

stemming from the M-frame of instant i than to merge this transition to an existing I-frame:

$$(\alpha\Sigma_t + \lambda)r^P \geq \alpha\Sigma_t r^I$$
$$\lambda \geq \alpha\Sigma_t \left(\frac{r^I - r^P}{r^P}\right).$$

We can now bound the cost improvement of the P+I structure of the alternative procedure over the I-only structure of the greedy procedure as follows:

$$\begin{aligned}
&\alpha\Sigma_t' r^I - (\alpha\Sigma_t' + \lambda)r^P \\
&\quad = \alpha\Sigma_t' \left(r^I - r^P\right) - \lambda r^P \\
&\quad \geq \alpha\Sigma_t'(r^I - r^P) - \alpha\Sigma_t \left(\frac{r^I - r^P}{r^P}\right) r^P \\
&\quad = (\alpha\Sigma_t' - \alpha\Sigma_t)(r^I - r^P) = \alpha(p_t - p_t')(r^I - r^P).
\end{aligned}$$

Hence, the gain is also bounded by $\alpha(p_t - p_t')(r^I - r^P)$. The same analysis will show that the gain for transition to other views k's, $k \neq j$, scaled by probability $1 - \alpha$, is also bounded similarly. Hence, alternative procedure does not lead to a lower Lagrangian cost than the greedy procedure at instant $i+1$. By induction, it does not lead to a lower Lagrangian cost in future instants as well. Therefore, the alternative procedure does not lead to a smaller Lagrangian cost and the optimality of the greedy procedure is proven. □

REFERENCES

[1] T. Fujii, K. Mori, K. Takeda, K. Mase, M. Tanimoto, and Y. Suenaga, "Multipoint measuring system for video and sound—100 camera and microphone system," presented at the IEEE Int. Conf. Multimedia and Expo, Toronto, Canada, Jul. 2006.

[2] G. Cheung, A. Ortega, and T. Sakamoto, "Coding structure optimization for interactive multiview streaming in virtual world observation," presented at the IEEE Int. Workshop on Multimedia Signal Processing, Cairns, Queensland, Australia, Oct. 2008.

[3] OpenGL: The Industry's Foundation for High Performance Graphics. [Online]. Available: http://www.opengl.org

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

[5] M. Flierl, A. Mavlankar, and B. Girod, "Motion and disparity compensated coding for multiview video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 11, pp. 1474–1484, Nov. 2007.

[6] S. Shimizu, M. Kitahara, H. Kimata, K. Kamikura, and Y. Yashima, "View scalable multiview coding using 3-D warping with depth map," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 11, pp. 1485–1495, Nov. 2007.

[7] J. Kim, P. Lai, J. Lopez, A. Ortega, Y. Su, P. Yin, and C. Gomila, "New coding tools for illumination and focus mismatch compensation in multiview video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 11, pp. 1519–1535, Nov. 2007.

[8] Digital Video Broadcasting. [Online]. Available: http://www.dvb.org/

[9] M. Karczewicz and R. Kurceren, "The SP- and SI-frames design for H.264/AVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 637–644, Jul. 2003.

[10] N. Cheung and A. Ortega, "Distributed source coding application to low-delay free viewpoint switching in multiview video compression," presented at the Picture Coding Symp., PCS'07, Lisbon, Portugal, Nov. 2007.

[11] N.-M. Cheung, A. Ortega, and G. Cheung, "Distributed source coding techniques for interactive multiview video streaming," presented at the 27th Picture Coding Symp., Chicago, IL, May 2009.

[12] D. Taubman and R. Rosenbaum, "Rate-distortion optimized interactive browsing of JPEG2000 images," presented at the IEEE Int. Conf. Image Processing, Barcelona, Spain, Sep. 2003.

[13] JPEG2000 Interactive Protocol (Part 9—JPIP). [Online]. Available: http://www.jpeg.org/jpeg2000/j2kpart9.html

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

[15] M. Levoy and P. Hanrahan, "Light field rendering," in *Proc. SIGGRAPH'96*, Aug. 1996, pp. 31–42.

[16] H.-Y. Shum, S. B. Kang, and S.-C. Chan, "Survey of image-based representations and compression techniques," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 11, pp. 1020–1037, Nov. 2003.

[17] A. Jagmohan, A. Sehgal, and N. Ahuja, "Compression of lightfield rendered images using coset codes," in *Proc. 37th Asilomar Conf. Signals, Systems and Computers*, Nov. 2003, vol. 1, pp. 830–834.

[18] A. Aaron, P. Ramanathan, and B. Girod, "Wyner-Ziv coding of light fields for random access," presented at the IEEE Int. Workshop on Multimedia Signal Processing, Siena, Italy, Sep. 2004.

[19] P. Ramanathan and B. Girod, "Random access for compressed light fields using multiple representations," presented at the IEEE Int. Workshop on Multimedia Signal Processing, Siena, Italy, Sep. 2004.

[20] C.-L. Chang and B. Girod, "Receiver-based rate-distortion optimized interactive streaming for scalable bitstreams of light fields," presented at the 2004 IEEE Int. Conf. Multimedia and Expo, Taiwan, Jun. 2004.

[21] I. Bauermann and E. Steinbach, "RDTC optimized compression of image-based scene representation (part I): Modeling and theoretical analysis," *IEEE Trans. Image Process.*, vol. 17, no. 5, pp. 709–723, May 2008.

[22] I. Bauermann and E. Steinbach, "RDTC optimized compression of image-based scene representation (part II): Practical coding," *IEEE Trans. Image Process.*, vol. 17, no. 5, pp. 724–736, May 2008.

[23] M. Levoy and K. Pulli, "The digital michelangelo project: 3-D scanning of large statues," in *Proc. SIGGRAPH'00*, Aug. 2000, pp. 131–144.

[24] Video Coding for Low Bitrate Communication, ITU-T, Recommendation H.263, Feb. 1998.

[25] G. Cheung, A. Ortega, and N.-M. Cheung, "Generation of redundant coding structure for interactive multiview streaming," presented at the 17th Int. Packet Video Workshop, Seattle, WA, May 2009.

[26] A. M. Tekalp, E. Kurutepe, and M. R. Civanlar, "3DTV over IP: End-to-end streaming of multiview video," *IEEE Signal Process. Mag.*, vol. 24, no. 6, pp. 77–87, Nov. 2007.

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

[28] J.-G. Lou, H. Cai, and J. Li, "A real-time interactive multi-view video system," presented at the ACM Int. Conf. Multimedia, Singapore, Nov. 2005.

[29] G. Cheung, N.-M. Cheung, and A. Ortega, "Optimized frame structure using distributed source coding for interactive multiview streaming," presented at the 2009 IEEE Int. Conf. Image Processing, Cairo, Egypt, Nov. 2009.

[30] D. Slepian and J. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. Inf. Theory*, vol. IT-19, pp. 471–480, Jul. 1973.

[31] R. Puri and K. Ramchandran, "PRISM: A new robust video coding architecture based on distributed compression principles," presented at the 2002 Allerton Conf. Communications, Control, and Computing, Urbana-Champaign, IL, Oct. 2002.

[32] B. Girod, A. Aaron, S. Rane, and D. Rebollo-Monedero, "Distributed video coding," *Proc. IEEE, Special Issue on Advances in Video Coding and Delivery*, vol. 93, no. 1, pp. 71–83, Jan. 2005.

[33] H. Wang, N.-M. Cheung, and A. Ortega, "A framework for adaptive scalable video coding using Wyner-Ziv techniques," *EURASIP J. Appl. Signal Process.*, 2006, DOI 10.1155/ASP/2006/60971.

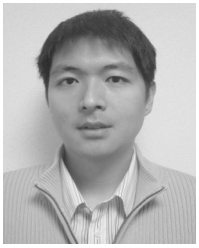
[34] Q. Xu, V. Stankovic, and Z. Xiong, "Layered Wyner-Ziv video coding for transmission over unreliable channels," *EURASIP J. Signal Process. Special Issue on Distributed Source Coding*, vol. 86, no. 11, Nov. 2006.

[35] A. Sehgal, A. Jagmohan, and N. Ahuja, "Wyner-Ziv coding of video: An error-resilient compression framework," *IEEE Trans. Multimedia*, vol. 6, no. 2, pp. 249–258, Apr. 2004.

[36] J. Wang, V. Prabhakaran, and K. Ramchandran, "Syndrome-based robust video transmission over networks with bursty losses," presented at the Proc. Int. Conf. Image Processing (ICIP), Atlanta, GA, 2006.

[37] X. Zhu, A. Aaron, and B. Girod, "Distributed compression for large camera arrays," presented at the Workshop on Statistical Signal Processing, St. Louis, MO, 2003.

[38] X. Guo, Y. Lu, F. Wu, W. Gao, and S. Li, "Distributed multi-view video coding," presented at the Conf. Visual Communications and Image Processing (VCIP), San Jose, CA, 2006.
[39] N.-M. Cheung, C. Tang, A. Ortega, and C. S. Raghavendra, "Efficient wavelet-based predictive Slepian-Wolf coding for hyperspectral imagery," *EURASIP J. Signal Process. Special Issue on Distributed Source Coding*, vol. 86, no. 11, Nov. 2006.
[40] C. Guillemot, F. Pereira, L. Torres, T. Ebrahimi, R. Leonardi, and J. Ostermann, "Distributed monoview and multiview video coding," *IEEE Signal Process. Mag.*, vol. 24, no. 5, pp. 67–76, Sep. 2007.
[41] S. Draper and E. Martinian, "Compound conditional source coding, Slepian-Wolf list decoding, and applications to media coding," presented at the Int. Symp. Information Theory (ISIT), Nice, France, 2007.
[42] N.-M. Cheung and A. Ortega, "Compression algorithms for flexible video decoding," presented at the Visual Communications and Image Processing (VCIP), San Jose, CA, 2008.



Gene Cheung (M'00–SM'07) received the B.S. degree in electrical engineering from Cornell University, Ithaca, NY, in 1995, and the M.S. and Ph.D. degrees in electrical engineering and computer science from the University of California, Berkeley, in 1998 and 2000, respectively.

He was a senior researcher with Hewlett-Packard Laboratories, Japan, Tokyo, from 2000 to 2009. He is currently an Assistant Professor with the National Institute of Informatics, Tokyo, Japan. His research interests include media representation and network transport, single- /multiple-view video coding and streaming, and immersive communication and interaction. He has published more than 15 international journal and 50 conference publications.

Dr. Cheung has served as an Associate Editor of the IEEE TRANSACTIONS ON MULTIMEDIA since 2007, and serves as area chair of the IEEE International Conference on Image Processing (ICIP) 2010, and as technical program co-chair of the International Packet Video Workshop 2010. He was a co-recipient of the Top 10% Paper Award in IEEE International Workshop on Multimedia Signal Processing 2009.

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

In 1994 he joined the Electrical Engineering-Systems Department at the University of Southern California (USC), Los Angeles, where he is currently a Professor. He currently serves as Associate Chair of EE-Systems and as was previously a director of the Signal and Image Processing Institute at USC. His research interests are in the areas of multimedia compression, communications and signal analysis. His recent work is focusing on distributed compression, multiview coding, error tolerant compression, wavelet-based signal analysis and information representation in wireless sensor networks. His work at USC has been or is being funded by agencies such as NSF, NASA, DOE, as well as a number of companies. Over 25 Ph.D. students have completed their Ph.D. thesis work under his supervision at USC, and his work has led to over 250 publications in international conferences and journals.

Dr. Ortega has been Chair of the Image and Multidimensional Signal Processing (IMDSP) technical committee and a member of the Board of Governors of the IEEE Signal Processing Society (2002). He has been technical program co-chair of ICIP 2008, MMSP 1998 and ICME 2002. He has been an Associate Editor for the IEEE TRANSACTIONS ON IMAGE PROCESSING, the *IEEE Signal Processing Letters* and the EURASIP *Journal on Advances in Signal Processing*. He has received the NSF CAREER award, the 1997 IEEE Communications Society Leonard G. Abraham Prize Paper Award, the IEEE Signal Processing Society 1999 Magazine Award, and the 2006 EURASIP Journal of Advances in Signal Processing Best Paper Award. He is a Fellow of the IEEE and a member of ACM.

Ngai-Man Cheung (M'08) received the Ph.D. degree from the University of Southern California (USC), Los Angeles, in 2008.

He is currently a postdoctoral researcher with the Information Systems Laboratory, Stanford University, Stanford, CA. He was a research associate with Hong Kong University of Science and Technology (HKUST) from 2008 to 2009. His research interests are multimedia signal processing and compression.

Dr. Cheung has received paper awards from the EURASIP *Journal of Advances in Signal Processing*, IEEE International Workshop on Multimedia Signal Processing (MMSP) 2007, IS&T/SPIE VCIP 2008, and the USC Department of Electrical Engineering.