View-Popularity-Driven Joint Source and Channel Coding of View and Rate Scalable Multi-View Video

Jacob Chakareski, Vladan Velisavljević, and Vladimir Stanković

***Abstract*—We study the scenario of multicasting multi-view video content, recorded in the video plus depth format, to a collection of heterogeneous clients featuring Internet access links of diverse packet loss and transmission bandwidth values. We design a popularity-aware joint source-channel coding optimization framework that allocates source and channel coding rates to the captured content, such that the aggregate video quality of the reconstructed content across the client population is maximized, for the given packet loss and bandwidth characteristics of the clients and their view selection preferences. The source coding component of our framework features a procedure for generating a view and rate embedded bitstream that is optimally decodable at multiple data rates and accounts for the different popularity of diverse video perspectives of the scene of interest, among the clients. The channel coding component of our framework comprises an expanding-window rateless coding procedure that optimally allocates parity protection bits to the source encoded layers, in order to address packet loss across the unreliable client access links. We develop an optimization method that jointly computes the source and channel coding decisions of our framework, and also design a fast local-search-based solution that exhibits a negligible performance loss relative to the full optimization. We carry out comprehensive simulation experiments and demonstrate significant performance gains over competitive state-of-the-art methods (based on H.264/AVC and network coding, and H.264/SVC and our own channel coding procedure), across different scenario settings and parameter values.**

***Keywords*—Joint source-channel multi-view video coding, view and rate scalable encoding, rateless codes, video multicast.**

I. Introduction

Multi-view video (MVV) has emerged as an exciting novel paradigm for interactive multimedia that has the potential to significantly augment our capacity to communicate and collaborate online. It is expected that MVV will usher in a new age of immersive communication that will affect our society broadly, by leading to innovative applications of higher productivity and quality of experience in entertainment, remote control and monitoring, telecommuting and telemedicine, and many other areas [1]. In brief, MVV enhances the sensation of immersion in the remote scene for the user, by allowing it to switch to different viewpoints dynamically [2].

 Jacob Chakareski is with the Department of Electrical and Computer Engineering, the University of Alabama, Tuscaloosa, Al 35487, USA. Vladan Velisavljević is with the Department of Computer Science and Technology, the University of Bedfordshire, Luton, UK. Vladimir Stanković is with the Department of Electronic and Electrical Engineering, the University of Strathclyde, Glasgow, UK.

Compared to its single-camera counterpart, MVV is characterized by an N-fold bandwidth and complexity expansion, since content needs to be captured from multiple perspectives simultaneously. To increase transmission efficiency, multicast delivery of such a content may be utilized, when multiple users may be interested to visually interact with the same scene simultaneously. This is the subject we study here.

In particular, we consider a scenario where MVV content is streamed to a collection of heterogeneous clients, characterized by different access link characteristics (bandwidth and packet loss). To lower the complexity of the system and improve its efficiency, we replace the individual (unicast) connections to every client with a single multicast distribution tree, as illustrated in Figure 1. To construct a single content distribution stream that can be reconstructed at every client at optimal video quality, at different data rates, we formulate a novel popularity-driven view and rate scalable encoding procedure that accounts for the different view selection preferences of the clients. Our source coding strategy is inspired by our recent work on view-rate scalable unicast multi-view streaming [3]. Furthermore, to combat packet loss on the access links of the clients, we map the view and rate scalable source stream onto optimal channel coding protection levels that we integrate into the source encoding process. Our joint source-channel coding approach delivers gains over competing reference methods, as our experiments show. In brief, our main contributions are

- A viewpoint-popularity-aware source coding for view and rate scalable multi-view video multicast that extends our prior work in [3] to rate-distortion optimized embedded source coding for multiple heterogeneous target client classes;
- A joint source-channel coding scheme that exploits rateless expanding-window random linear coding for unequal packet erasure protection and embedded source coding for reliable multi-view video multicast to heterogenous clients;
- A framework for optimizing the source and channel coding parameters under transmission rate constraints given view popularity distribution;
- Evaluation of the robustness of the proposed system in different application scenarios and comparison with prior source-channel coding methods, demonstrating considerable advances over the state-of-the-art.

The rest of the paper is organized as follows. We briefly describe the video plus depth (VpD) multi-view format that we use and review related work in Section II. In Section III, we describe the source and channel coding components of our framework. In Section IV, we formulate two constrained optimization problems of computing source and channel encoding

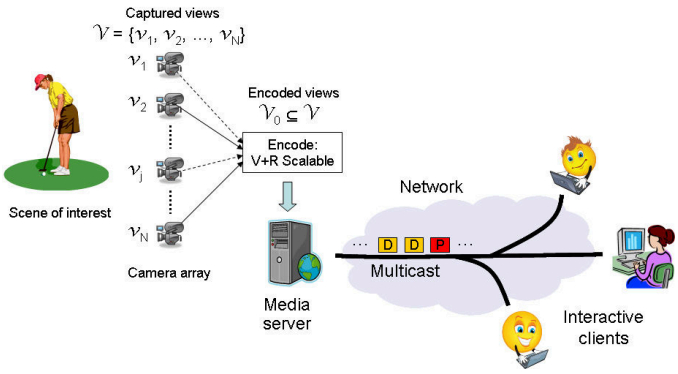

Fig. 1. Scalable multicast to multi-view clients that receive appropriate amounts of data (D) and parity (P) packets sent over the tree, to reconstruct desired viewpoints at optimal video quality.

rates such that the aggregate video quality over the client population is maximized, whereas in Section V, we evaluate the performance of our system and compare against reference methods. Finally, we conclude in Section VI.

II. Background

A. Video plus depth MVV

MVV features N captured viewpoints (video signals) to which a user can simultaneously switch. Observing the remote scene from other (virtual) viewpoints can be achieved via view interpolation. To this end, depth signals are recorded for every camera location using time-of-flight cameras [4]. In essence, a depth signal measures the distance of each object in the scene from the camera location. A virtual viewpoint is synthesized using the depth and video signals for the two nearest captured viewpoints using a procedure known as 3D warping[1] [5]. In general, depth signals can better handle scenery with multiple objects compared to mesh-based models that require dense image sampling around a single object.

B. Source coding

The study in [6] considered encoding VpD MVV with a single rate constraint known ahead of time. We face a more challenging problem here, since our clients are bandwidth heterogenous. This intuitively calls for a scalable coding solution that will deliver video quality proportional to the downlink bandwidth. A scalable or layered bitstream starts with a base layer and continues with a set of enhancement layers of progressively lower importance. H.264/SVC [7] is a recent scalable extension of the H.264/AVC video coding standard [8] that provides efficient scalability functionalities, at competitive video quality. Quality scalability, the focus of our paper, enables the use of a single stream to describe video content at different fidelity levels. In this way, the receivers that only receive a part of the stream can still reconstruct the content, though at lower quality. The more enhancement layers a receiver decodes the higher its reconstruction video quality becomes. State-of-the-art wavelet-based scalable video coders (see [9] and the references therein) that use motion-compensated temporal filtering usually provide better quality scalability features than SVC (e.g. fine rate granularity), but suffer from performance loss. However, a recent JPEG2000-compatible scalable wavelet-based codec proposed in [10] provides results close to those of H.264/SVC.

In [11], loss-resilient source coding of VpD MVV is studied, however, with no channel coding considerations. Similarly, [12] considers multicast of MVV, where the captured video and depth signals are SVC encoded, and each client is served two reference video and depth signals. It is shown that finding the optimal subset of scalable video and depth signal layers to transmit for each reference view, which maximize the clients' received video quality is an NP-complete problem. In contrast to our work, [12] uses only two views, compresses each view with SVC, and does not consider channel error control. Finally, in our earlier work [3] we have studied the problem of delivery of scalable multi-view content to a single user. The present paper extends our source coding framework from [3] to view-popularity-driven joint source-channel coding for scalable multi-view multicast to a collection of clients, where it is optimally matched with an error protection transmission method that we design. Here, we integrate the source coding, channel coding, and client heterogeneity and view interaction aspects into one unifying framework that aims to optimize the operation of the system end-to-end.

C. Channel coding

Random linear codes (RLC) are a class of rateless codes that are becoming increasingly popular for erasure protection over wireless networks due to their simple implementation, flexibility, and natural extension to multi-hop setups [13]. RLC are flexible for adaptation to video content and varying channel conditions via unequal error protection (UEP). In [14], the popular expanding window fountain (EWF) coding UEP approach [15] is applied to RLC, leading to an EW-RLC design based on the idea of creating a set of nested windows over the source data block.

D. Source-channel coding

UEP EW-RLC have been used for transmission of single-camera video, e.g., in [16], where EW-RLC are proposed as an application-layer forward error protection solution for transmission of H.264 AVC video over DVB-H networks. In addition, in [17], RLC is proposed for transmission of H.264 SVC video over LTE networks at the MAC layer, as a replacement of traditional ARQ. In [18], prioritized video streaming over lossy overlay networks using UEP-based RLC is proposed for single-view video. In [19], depth maps are used to recover lost texture maps for WWAN video streaming and source-channel optimization framework is formed to allocate the optimal amount of redundancy to texture and depth maps.

In [20], 3D video transmission over lossy networks is proposed that allocates different priorities to colour and depth

[1]Direct interpolation from closest video signals exhibits poor quality, since it cannot account for the scene's 3D geometry.

map stream based on their importance for the reconstruction of the content. In [21], a cross-layer optimization framework for scalable VpD video streaming is proposed with H.264 SVC for source coding and Reed-Solomon codes for packet-level erasure protection. In [22], joint source-channel coding of VpD content is considered, where H.264 AVC is used for compression of texture and depth information, while turbo codes are used for error protection. In [23], VpD video is protected using prioritized network coding [18] and multicast to heterogeneous clients in a multi-hop network. The optimization problem is posed taking into account different channel conditions, as well as video distortion and view popularity characteristics, and solved using the hill-climbing algorithm from [24]. Actual views are source encoded independently in an incremental fashion to form quality-scalable layers. In contrast to this work, in our system a layer can comprise multiple encoded viewpoints, at the same time, whose quality gradually improves from the lowest to the highest layer. This offers a considerably improved performance, as it enables a higher system flexibility and more effective view synthesis at the decoder, as observed in our experiments.

III. MVV Multicast System

A. View and rate scalable encoding

For encoding the captured MVV content, we extend the scalable coder developed in [3] that provides joint view and rate scalability. The coder generates an embedded bitstream that features video and depth signals of captured viewpoints. The encoding used for the selected views is based on shape-adaptive wavelets [25] followed by SPIHT [26] applied to the difference between the original frame and its prediction, for the same view. This prediction can be either (i) the previously quantized version of the same frame or (ii) a synthesized frame obtained using view interpolation techniques (e.g., depth-image-based rendering) with nearest left and right previously encoded views as reference. In (i), an already compressed view is refined using the best predictor thus achieving rate scalability. In turn, in (ii), a new captured view is inserted into the set of compressed views providing therefore view scalability. For each coding layer, the coder optimizes the coding strategy by selecting the best choice between (i) and (ii) for the best encoding view such that rate-distortion performance is maximized.

B. Forward error correction

Our EW-RLC scheme, illustrated in Figure 2, starts by selecting a window from which the encoded symbol will be generated. The window selection is independently performed for each encoded symbol and is governed by window selection probabilities $\Lambda = [\lambda_1, \ldots, \lambda_L]$ that are assigned ahead of time and known at both the encoder and decoder. Their selection is carried out according to the importance of the different source symbols and the available data rate. Note that $\sum_i^L \lambda_i = 1$.

[14] derives an expression for the decoding probability of window l. For completeness, we include here the main aspects of the formulation. Let K_l be the symbol length of window l,

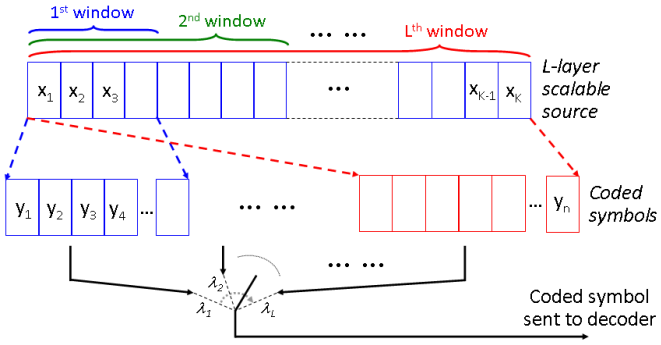

Fig. 2. EW-RLC: A scalable source is organized into L embedded windows of progressively increasing size. Window k comprises windows $1, \ldots, k$, for $k = 1, \ldots, L$. Each coded symbol is generated using RLC over a selected window, where λ_k denotes the probability of selecting window k. One window contains one or more source layers.

and let n_l denote the number of coded symbols, generated over window l, received by a client. Thus, $\mathbf{n} = (n_1, \ldots, n_L)$ denotes the vector of received coded symbols, for every window $l = 1, \ldots, L$, where $N = \sum_l n_l$ is the total number of received coded symbols. Then, the probability that a received sequence of coded symbols of length N features the distribution of received coded symbols per window specified by $\mathbf{n}$ is governed by the multinomial probability mass function[2], i.e.,

$$P_{\Lambda,N}(\mathbf{n}) = \frac{N!}{n_1! \cdots n_L!} \lambda_1^{n_1} \cdots \lambda_L^{n_L}. \qquad (1)$$

Given (1), the probability of successful decoding of window l can be computed as

$$P_l(N) = \sum_{\substack{(n_1,\ldots,n_L):\\ 0 \le n_1 \le \ldots \le n_L \le N \\ \sum_l n_l = N}} P_{\Lambda,N}(\mathbf{n}) P_l(N|\mathbf{n}), \qquad (2)$$

where $P_l(N|\mathbf{n})$ denotes the probability that window l can be decoded, given that the received sequence can be described by $\mathbf{n}$. It can be shown that $P_l(N|\mathbf{n})$ can be upper bounded by $I(n_l \ge K_l - K_{l-1})$, where $K_0 = 0$ and $I(\cdot)$ represents the indicator function that is equal to 1, if its argument is true, and zero, otherwise. An expression for $P_l(N|\mathbf{n})$ can be found in [14], where it is further shown that $I(n_l \ge K_l - K_{l-1})$ also represents a good estimate of $P_l(N|\mathbf{n})$.

C. Client population & view selection

There are N_c client classes characterized with distinct bandwidth and packet loss pairs. Let γ_j denote the fraction of the client population associated with class j. Let $\mathcal{V} = \{v_1, \ldots, v_N\}$ denote the discrete set of captured viewpoints[3]. We quantize the continuum of prospective views $[v_1, v_N]$ that a user can select to watch into a discrete set $\bar{\mathcal{V}} \supset \mathcal{V}$. Note

[2] Assuming independent channel symbol erasures during transmission.

[3] Note that $\mathcal{V}$ represents a mathematical abstraction that facilitates our analysis. Whenever we refer to encoding viewpoint v_i henceforth, we have in mind the encoding of the corresponding video frames captured from v_i.

that these views in $\bar{\mathcal{V}}$ may consist of both captured and virtual (i.e. synthesized) views. Now, let w_i denote the fraction of clients accessing viewpoint $V_i \in \bar{\mathcal{V}}$. The factor w_i can be considered as the popularity of V_i over the client population or the likelihood that a client selects V_i to watch.

We consider that the provider of the multi-view video application will have available the aggregate client access link packet loss and bandwidth characteristics described above. That is because today IP multicast video services are typically delivered by the same ISP providers through which the clients connect to the Internet[4]. An ISP provider will have such information readily available off-line and can easily update it dynamically, by monitoring data packets entering its network through an access link. Moreover, view switching capability is established at the ingress router through which clients connect to the Internet, at which point local statistics for the views' popularity can be collected, as well, before they are forwarded back to the encoding multicast server in an aggregated form. Thus, feedback implosion overwhelming the IP network cannot occur, as the individual view switching requests are not propagated further upstream.

IV. Source-Channel Coding

A. Preliminaries

The content is encoded progressively into L source layers. Let $\mathbf{R}^{(l)} = (R_1^{(l)}, \ldots, R_l^{(l)})$ denote the vector of encoding data rates cumulatively assigned to layers $1, \ldots, l$ by the time layer l is encoded. Similarly, let $\mathbf{V}^{(l)} = (\mathcal{V}_0^{(1)}, \ldots, \mathcal{V}_0^{(l)})$ denote the vector of captured viewpoint sets cumulatively represented in the scalable bitstream by the time layer $k = 1, \ldots, l$ is encoded. By construction, it holds that $\mathcal{V}_0^{(l)} \subseteq \mathcal{V}_0^{(l+1)}$. In the following, we will address two optimization problems of interest, in the context of the scenario we examine.

B. Source rate allocation

We are interested in minimizing the *expected* video distortion over the client population, computed as $\sum_i w_i D_{V_i}(\mathbf{R}^{(L)}, \mathbf{V}^{(L)})$, such that the base layer encoding rate $R_1^{(L)}$ and the aggregate encoding rate of the content $\sum_{l=1}^{L} R_l^{(L)}$ meet required minimum and maximum transmission rate constraints, $C_{\min}$ and $C_{\max}$, associated with the multicast session. The latter are motivated by the needs to ensure minimum video quality delivered to every client and match the available serving rate for the session. Note that we consider that the clients are characterized by heterogenous bandwidth values only, in this case. $D_{V_i}(\mathbf{R}^{(L)}, \mathbf{V}^{(L)})$ represents the distortion of view $V_i \in \bar{\mathcal{V}}$, given the rate allocation and view coding selection decisions for all L layers[5]. Formally, we aim to solve

$$\min_{\mathbf{R}^{(L)}, \mathbf{V}^{(L)}} \sum_i w_i D_{V_i}(\mathbf{R}^{(L)}, \mathbf{V}^{(L)}), \tag{3}$$

$$\text{s.t. } C_{\min} \leq R_1^{(L)}; \ \sum_{l=1}^{L} R_l^{(L)} \leq C_{\max}\,.$$

The viewpoint distortion in (3) is computed as an integral value over all clients watching that view. Concretely,

$$D_{V_i}(\mathbf{R}^{(L)}, \mathbf{V}^{(L)}) = \sum_{j=1}^{N_c} \gamma_j D_{V_i}^{j}(\mathbf{R}^{(L)}, \mathbf{V}^{(L)})\,, \tag{4}$$

where γ_j is the fraction of clients in class j, and $D_{V_i}^{j}(\mathbf{R}^{(L)}, \mathbf{V}^{(L)})$ is the reconstruction error of viewpoint i for clients of that class, given $\mathbf{R}^{(L)}$ and $\mathbf{V}^{(L)}$. We compute $D_{V_i}^{j}(\mathbf{R}^{(L)}, \mathbf{V}^{(L)})$ via an expression derived in [3] that takes advantage of an accurate synthesized view distortion model that we derived in [27, 28]. Without loss of generality, we consider that γ_j is independent of the viewpoint index i.

To solve (3), we design the following optimization procedure. At initialization, the coder selects the left-most and right-most views to comprise the initial set of encoded views, i.e., $V^{(0)} = \{v_1, v_N\}$. It then sets the assigned (encoding) rates to the corresponding video and depth frames to zero, i.e., $R_{f_i,i}^{(0)} = 0, i \in V^{(0)}, f_i \in \{v, d\}$. Next, for every two consecutive coding layers l and $l+1$, the coder selects the best assignment of the incremental (layer) rates R_l and R_{l+1}, given its rate allocation carried out for layers $0 \leq k < l$. For simplicity, we consider that $R_l = \Delta R, \forall l$. Our optimization is implemented as a minimization of the cost function in (3), via an exhaustive search over all prospective assignments of R_l and R_{l+1} to encoding video or depth frames f_i and f_j associated with views i and j, at encoding layers l and $l+1$, where i and j could be new or already encoded views. Note that optimizing over two layers jointly represents a good trade-off between optimization performance[6] and computational complexity. Furthermore, we observed that expanding the optimization horizon to four layers does not provide significant additional benefits.

An algorithmic description of our source coding optimization is provided in Algorithm 1. We denote the action of rate assignment to view $i \in V^{(l)}$ as *refinement*, because the corresponding frame f_i is encoded predictively with respect to its version $\hat{f}_i$ present in the compressed bitstream comprising layers $1, \ldots, l-1$. That is, we encode the difference $f_i - \hat{f}_i$. The thereby created new bits are merged to the embedded code associated with frame f_i in the compressed bitstream, thus, allowing for refining the reconstruction quality of $\hat{f}_i$, at decoding. We denote the action of rate assignment to a new view as *insertion*, since a new view $i \in \mathcal{V}$ is inserted in $V^{(l)}$. In this case, the associated video or depth frame f_i is encoded predictively, using as a reference a synthesized version of the frame $\tilde{f}_i$, interpolated using the nearest left and right views in

[4]For example, FiOS IPTV by Verizon and Xfinity IPTV by Comcast.

[5]That is, $D_{V_i}(\mathbf{R}^{(L)}, \mathbf{V}^{(L)})$ represents the error of reconstructing viewpoint V_i from the compressed bitstream, given $\mathbf{R}^{(L)}$ and $\mathbf{V}^{(L)}$.

[6]Considering only one layer in isolation cannot exploit the benefit of allocating rate to both video and depth frames of the same viewpoint.

$V^{(l)}$. The exhaustive search computes the cost function in (3) for every possible assignment of R_l and R_{l+1} to refinement or insertion of $f_i, f_j \in \{v, d\}$, for $i, j \in \mathcal{V}$. It then selects the action that results in the smallest cost value, to generate coding layers l and $l+1$ that are then integrated into the embedded bitstream. In addition, the assigned rates $R^{(l)}_{f_i,i}$ and $R^{(l+1)}_{f_i,i}$, for $f_i \in \{v, d\}, i \in \mathcal{V}$ are updated to account for the incremental allocation of R_l and R_{l+1}. Similarly, the sets V^l and V^{l+1} are updated accordingly. When the optimization in Algorithm 1 completes, it results in an embedded stream with optimal source rate $R^{(L)*}$ and view selection $V^{(L)*}$.

Algorithm 1 View-popularity-driven scalable source coding

Initialize $V^{(0)} = \{v_1, v_N\}$; $R^{(0)}_{v,i} = R^{(0)}_{d,i} = 0, i \in V^{(0)}$; $l = 1$
repeat
  for $i \in \mathcal{V}$ and $f_i \in \{v, d\}$ **do**
    if $i \in V^{(l)}$ **then**
      Encode($f_i - \hat{f}_i$); $V^{(l+1)} = V^{(l)}$
      for $j \in \mathcal{V}$ and $f_j \in \{v, d\}$ **do**
        if $j \in V^{(l+1)}$ **then**
          Encode($f_j - \hat{f}_j$)
        else
          Encode($f_j - \tilde{f}_j$)
        end if
        Compute the cost function in (3)
        Record the result in $D(i, j, f_i, f_j)$
      end for
    else
      Encode($f_i - \tilde{f}_i$); $V^{(l+1)} = V^{(l)} \cup \{i\}$
      for $j \in \mathcal{V}$ and $f_j \in \{v, d\}$ **do**
        if $j \in V^{(l+1)}$ **then**
          Encode($f_j - \hat{f}_j$)
        else
          Encode($f_j - \tilde{f}_j$)
        end if
        Compute the cost function in (3)
        Record the result in $D(i, j, f_i, f_j)$
      end for
    end if
  end for
  $(i, j, f_i, f_j)^* = \arg\min D(i, j, f_i, f_j)$
  $R^{(l)}_{f_i,i} = R^{(l-1)}_{f_i,i}, i \in \mathcal{V}, f_i \in \{v, d\}$
  if $i^* \in V^{(l-1)}$ **then**
    $V^{(l)} = V^{(l-1)}$
  else
    $V^{(l)} = V^{(l-1)} \cup \{i^*\}$
  end if
  $R^{(l)}_{f^*_i,i^*} = R^{(l)}_{f^*_i,i^*} + \Delta R$
  $R^{(l+1)}_{f_i,i} = R^{(l)}_{f_i,i}, i \in \mathcal{V}, f_i \in \{v, d\}$
  if $j^* \in V^{(l)}$ **then**
    $V^{(l+1)} = V^{(l)}$
  else
    $V^{(l+1)} = V^{(l)} \cup \{j^*\}$
  end if
  $R^{(l+1)}_{f^*_j,j^*} = R^{(l+1)}_{f^*_j,j^*} + \Delta R$
  $l = l + 2$
until $l \leq L$

Our principle when designing Algorithm 1 was simplicity. Thus, we opted not to formulate a solution to (3) via more sophisticated techniques, e.g., dynamic programming [29], since due to the complexity of (3), the latter would not lead to better solutions.

C. Source and channel rate allocation

Here, we consider that the clients' access links may also exhibit heterogeneous packet loss. Thus, the multi-view multicast layers need to be protected against its impact on video quality. In particular, now, the reconstruction error of a viewpoint V_i will also depend on the assignment of forward error correction (FEC) packets to each of the layers, carried out by the server. In the following, for simplicity and without loss of generality, we assume that one source layer is put in one transmission window. Formally, let $\mathbf{R}^{(L)}_p = (R^p_1, \ldots, R^p_L)$ denote the rate of protection (parity) packets assigned to every window. We are interested in computing $\mathbf{R}^{(L)}$ and $\mathbf{R}^{(L)}_p$ jointly, inclusive of $\mathbf{V}^{(L)}$, such that the aggregate video quality over the client population is maximized. In this case, the overall data rate of the L windows needs to meet the multicast session's transmission rate constraints. Thus, we write

$$\min_{\mathbf{R}^{(L)}, \mathbf{R}^{(L)}_p, \mathbf{V}^{(L)}} \sum_i w_i D_{V_i}(\mathbf{R}^{(L)}, \mathbf{R}^{(L)}_p, \mathbf{V}^{(L)}) \tag{5}$$

$$\text{s.t. } C_{\min} \leq \left(R^{(L)}_1 + R^p_1\right); \sum_{l=1}^{L} \left(R^{(L)}_l + R^p_l\right) \leq C_{\max}.$$

Similarly to (4), $D_{V_i}(\mathbf{R}^{(L)}, \mathbf{R}^{(L)}_p, \mathbf{V}^{(L)})$ is computed using

$$D_{V_i}(\mathbf{R}^{(L)}, \mathbf{R}^{(L)}_p, \mathbf{V}^{(L)}) = \sum_{j=1}^{N_c} \gamma_j D^j_{V_i}(\mathbf{R}^{(L)}, \mathbf{R}^{(L)}_p, \mathbf{V}^{(L)}), \tag{6}$$

where $D^j_{V_i}(\mathbf{R}^{(L)}, \mathbf{R}^{(L)}_p, \mathbf{V}^{(L)})$ denotes in this case the expected reconstruction error of viewpoint i for client class j, given $\mathbf{R}^{(L)}, \mathbf{R}^{(L)}_p$, and $\mathbf{V}^{(L)}$, which can be computed as

$$D^j_{V_i}(\mathbf{R}^{(L)}, \mathbf{R}^{(L)}_p, \mathbf{V}^{(L)}) = \sum_{l=0}^{L} P_{1:l}(N) D^j_{V_i}(1 : l | \mathbf{R}^{(L)}, \mathbf{V}^{(L)}), \tag{7}$$

where $P_{1:0} = 1 - P_1(N)$, $P_{1:L} = \prod_{i=1}^{L} P_i(N)$, and for $l = 1, \ldots, L-1$, $P_{1:l} = \prod_{i=1}^{l} P_i(N)(1 - P_{l+1}(N))$. Note that $P_{1:l}(N)$ is the probability that the first l layers are decoded correctly, while layer $l+1$ is not decoded correctly. Furthermore, $D^j_{V_i}(1 : l | \mathbf{R}^{(L)}, \mathbf{V}^{(L)})$ represents the reconstruction distortion of viewpoint i for client class j when the first l transmitted layers are decoded correctly, given the source rate allocation and view selection $(\mathbf{R}^{(L)}, \mathbf{V}^{(L)})$. Here, $D^j_{V_i}(0 | \mathbf{R}^{(L)}, \mathbf{V}^{(L)})$ denotes the reconstruction error when no received layers are decoded correctly. This quantity depends on the error concealment strategy used by the clients. We note that the probabilities $P_{1:l}(N)$ directly depend on the amount of protection added, that is, $\mathbf{R}^{(L)}_p$ (see Section III), hence the right-hand side of (7) is also a function of $\mathbf{R}^{(L)}_p$.

D. Optimization methods

Note that our source encoding procedure produces an embedded bitstream of fine granularity. Therefore, given our channel encoding procedure from Section III-B, solving (5) can be carried out by determining the partition of the embedded bitstream across its L windows, illustrated in Figure 2, that is determining the source coding rate per layer, and computing the corresponding window selection probabilities λ_i. Let $s_l(\mathbf{R}^{(L)}, \mathbf{R}_p^{(L)})$ be the number of source symbols in window l. In the following, for clarity, we denote $s_l(\mathbf{R}^{(L)}, \mathbf{R}_p^{(L)})$ simply as s_l. Then, (5) can be reformulated as

$$\min_{\{s_l\},\{\lambda_l\}} \sum_i w_i D_{V_i}(\{s_l\},\{\lambda_l\}) \tag{8}$$

$$\text{s.t. } C_{\min} \le N_1;\ \sum_{l=1}^{L} \lambda_l = 1; N_L \le C_{\max},$$

where N_l is the cumulative number of symbols that can be generated by channel coding of the source data in windows $k = 1, \ldots, l$. In our implementation, we solve (8) by quantizing the probabilities λ_i using a step size of 0.1, which was empirically found to provide good trade-off between complexity and performance, and then applying either full search or local search algorithm.

1) Full search: The full search method is based on computing the objective in (8) for all combinations of $\{s_i\}$ and quantized $\{\lambda_i\}$, given $R^{*(L)}$ and $V^{*(L)}$. This is possible, since s_i and L need to be kept small due to the complexity of RLC decoding. The computational complexity of this optimization step is $O(|\{s_i\}|^{s_{\max}/\Delta_s} \cdot |\{\lambda_i\}|^{1/\Delta_\lambda})$, where Δ_s and Δ_λ denote the step sizes for the prospective s_i and λ_i values, and $s_{\max}$ represents the maximum possible value that an s_i can attain. Note that though our optimization features non-trivial complexity, that does not preclude its deployment in practice, as it is not expected to operate in real time, in the application we consider. Still, we present next a low-complexity method that approximates the exact solution closely.

2) Low-complexity local search: Instead of searching over all possible combinations of $\{s_i\}$ and quantized $\{\lambda_i\}$, we design a local search algorithm that significantly reduces the computation time. Our local search procedure is summarized in Algorithm 2. $\Delta\lambda$ and Δs denote the step change for the λ_i and s_j parameters, respectively. The algorithm starts by setting $s_j = N_j$ and $\lambda_j = 0$, for all j, save for $\lambda_1 = 1$. Then, for each distribution of the λ_i's, the algorithm decreases the s_j's as long as there is improvement. Once no further improvement can be obtained, the λ_i's are updated, and the s_j's are further decreased. Ultimately, when no further improvement can be achieved, the algorithm terminates.

Algorithm 2 Low-complexity local search

Initialize $[\lambda_1, \ldots, \lambda_L] = [\lambda_1^*, \ldots, \lambda_L^*] = [1, 0, \ldots, 0, 0]$
Initialize $[s_1, \ldots, s_L] = [s_1^*, \ldots, s_L^*] = [N_1, \ldots, N_L]$
Initialize $D_{max} = \infty$
for $i = 1$ to $L - 1$ **do**
  FLAG1 = 0
  repeat
    $\lambda_i = \lambda_i - \Delta\lambda; \lambda_{i+1} = \lambda_{i+1} + \Delta\lambda$
    if $\sum_j \lambda_j = 1$ **then**
      for $j = L$ to 1 **do**
        FLAG2 = 0
        repeat
          $s_j = s_j - \Delta s$
          Compute the cost function of (8)
          Assign the result to D
          if $D_{max} < D$ **then**
            $D_{max} = D; s_j^* = s_j; \lambda_i^* = \lambda_i$
            FLAG2 = 1; FLAG1 = 1
          else
            break
          end if
        until $s_i \le \Delta s$
      end for
    end if
    if FLAG1 = 0 **then**
      break
    end if
  until $\lambda_i \le \Delta\lambda$
end for
Return $[s_1^*, \ldots, s_L^*], [\lambda_1^*, \ldots, \lambda_L^*], D_{max}$

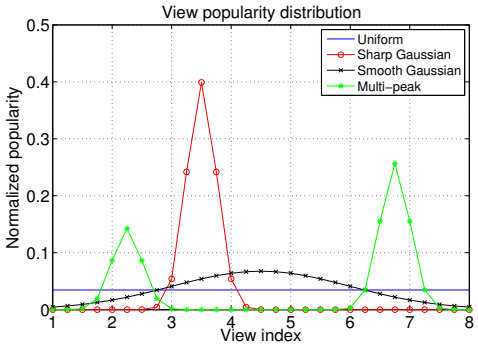

Fig. 3. Client popularity distribution: Uniform (blue), sharp Gaussian (red), smooth Gaussian (black) and multi-peak Gaussian (green).

V. Experiments

We carry out a comprehensive evaluation of various performance aspects of our system and its relation to multiple reference schemes. We carefully examine the impact of the multi-view content and the client population characteristics on the coding efficiency of all schemes under comparison. To evaluate the performance of our source-channel coding system, we either use analytical expressions given in Section III or carry out experiments in a custom-built Matlab simulator that we developed to this end, which is clear from the context. In our simulations, we assume that there is one receiver per class. Extension to multiple receivers per class is straightforward. Uniform view popularity distribution is always assumed unless otherwise stated.

A. Content, client, and channel characteristics

We use the multi-view video sequences Ballet and Breakdance provided by Microsoft Research [30]. They both fea-

ture 8 camera viewpoints capturing video signals of spatial resolution of 768×1024 pixels and temporal rate of 15 frames-per-second. The data sets include estimated depth video sequences for each camera, at the same spatial resolution and temporal rate. We adopt the depth-image-based rendering (DIBR) algorithm from [30] to synthesize virtual views based on encoded reference viewpoints, at a user. The captured 8 views are indexed as integers, $1, \ldots, 8$, whereas the allowed synthesized views comprise the encoded ones plus 3 virtual views between each pair of camera viewpoints (indexed as non-integers) amounting, thus, to a total of 29 views. We represent $D^J_{V_i}(\cdot)$ for a synthetic V_i as the PSNR of its interpolated video signal[7].

We consider that the clients' view popularity distribution, characterized by the weights w_i, can attain one of the following four types. First, a Gaussian function with a peak at the view indexed as 3.5 and variance of 0.25 (distance between two neighboring virtual views) is selected to correspond to a narrow interval of interest in user view selection. Second, a smoother Gaussian function with a peak at the view 4.5 and variance of 1.5 models a wider interval of interest. The third distribution corresponds to a multi-peak function comprising two sharp Gaussian functions both with variance of 0.25 centered at 2.25 and 6.75, respectively. Finally, a uniform popularity distribution where w_i are constant models a non-preferential user view selection – all views are equally popular. These four popularity distributions are graphically shown in Figure 3.

Since like digital fountain codes, EW-RLC represents a universal channel coding scheme for erasure channels [14, 15, 31], its performance is affected only by long-term average packet loss rate. Therefore, it suffices to examine only the number of received packets at the receiver for each coding window, and thus a conventional packet erasure channel model is used in our experiments.

B. Reference techniques

With *H.264/SVC*, we denote a reference system based on H.264/SVC and our EW-RLC scheme designed in Section III-B. In terms of source coding, it applies H.264/SVC across the video signal frames and the depth signal frames of the captured viewpoints, independently for every time instance, to enable random access to the encoded content for a user. The MGS configuration used for H.264/SVC exhibits 4 coding layers, each split into 4 additional sub-layers. Our EW-RLC scheme forms two windows s_1 and s_2 that comprise the base layer and the base plus enhancement layer of the encoded content, respectively. The symbol size is set to 1024 bytes, and one symbol is put in one transmission packet, which is common for RLC packetization [14, 16]. With *Toni et al.*, we denote the system proposed in [23]. It applies RLC in an incremental fashion to an embedded collection of viewpoints that are source-encoded independently using the standard video codec H.264. In the context of source coding, our performance measure is the objective function in (3), and in the context of joint source-channel coding, our performance measure is the objective function in (5).

[7]Relative to interpolation from non-compressed reference views.

C. Source coding efficiency

First, we examine the setup considered in Section IV-B. That is, we study the end-to-end performance of the competing techniques under examination in this paper, in the absence of packet loss (and thus channel coding). Specifically, in Figure 4 (for the content Breakdance) and Figure 5 (for the content Ballet), we compare the compression efficiency[8] of our source coding component (for the four popularity distributions shown in Figure 3) and *H.264/SVC*. The graphs in both figures demonstrate that knowing the clients' view preferences can improve coding efficiency in most cases, sometimes by more than 1dB. We also demonstrate that our method outperforms the standard H.264/SVC codec by more than 2dB.

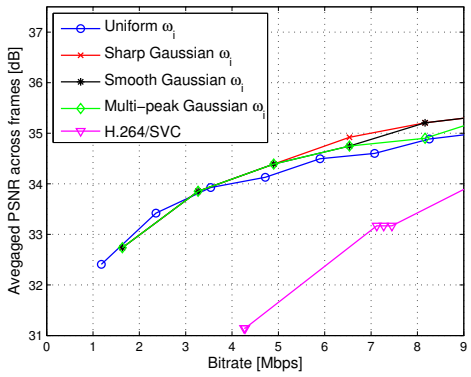

Fig. 4. Compression efficiency (Breakdance): Proposed method with uniform (blue), sharp Gaussian (red), smooth Gaussian (black) and multi-peak Gaussian (green) $\{w_i\}$ and *H.264/SVC* (magenta).

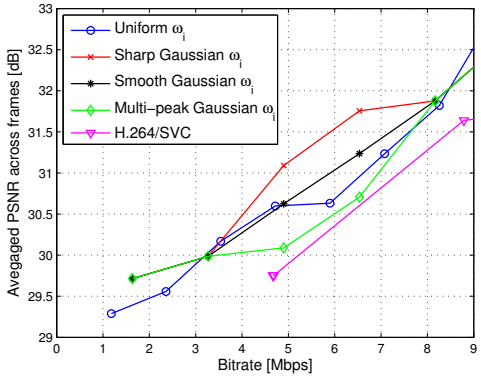

Fig. 5. Compression efficiency (Ballet): Proposed method with uniform (blue), sharp Gaussian (red), smooth Gaussian (black) and multi-peak Gaussian (green) $\{w_i\}$ and *H.264/SVC* (magenta).

[8]Measured as the average Y-PSNR of the reconstructed content across the client population versus the encoding rate of the content.

D. Source-channel coding performance

Here, we carry out multiple experiments. First, we consider multicast to two client classes, where the client access links are characterized as packet erasure channels. The two client classes comprise a high-rate (HR) class and a low-rate (LR) class. Thus, in our EW-RLC scheme from Section III-B, we construct two embedded windows that comprise the scalable source base layer only, in the case of window 1, and the scalable source base and enhancement layers, in the case of window 2. In these experiments, we first examine the impact of the multicast transmission rate, the packet erasure rate, and the client class distribution, expressed through the factors γ_i, on the end-to-end performance of our framework and *H.264/SVC*. Then, we examine the sensitivity of our optimization framework to a mismatch in the values of γ_i. That is, we optimize with respect to one set of γ_i values, however, the actual distribution on which we evaluate performance is different. Next, we present end-to-end performance results examining the impact of the view popularity distribution, followed by another set of experiments where three client classes are examined. Finally, we examine the difference in performance between Algorithm 2 and the full search method from Section IV-D, and study the relative performance of *Toni et al*. In all our experiments, each client class is assigned a different downlink bandwidth value, but equal packet erasure rate. Given the nature of the error protection codes we use, this setup is equivalent to fixing the client class bandwidth and varying the erasure rate across the classes.

1) Impact of transmission rate: Figure 6 and Figure 7 show the value of the objective function in (5) vs. the available multicast rate to HR clients, for a packet loss rate of 5%. The data rate at which the content is streamed to LR clients is half of that for the HR clients. Analytical expressions from Secs III and IV are used to evaluate system performance. It can be seen that only for rates $>$ 9.5Mb/sec the SVC scheme delivers the content to the LR users. This is due to the relatively high encoding rate of the SVC base layer. Only at very high rates (above 12MB/sec) the SVC scheme becomes marginally better than our solution.

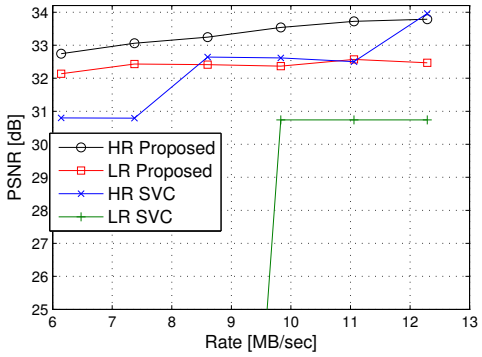

Fig. 6. Average video quality vs. HR client class multicast rate (Breakdance).

2) Impact of loss rate and $\{\gamma_i\}$*:* Figure 8 and Figure 9 show the average video quality for each client class, for three different γ_1 values. The transmission rate to the HR client class is set to 9.5Mbps and the transmission rate to the LR client class is set to 4.9Mbps. Each data point of a graph in Figure 8 and Figure 9 is obtained by optimizing the source-channel coding for that specific γ_1. Analytical expressions from Secs III and IV are used to evaluate system performance. It can be seen that our system significantly outperforms *H.264/SVC* for heterogenous client populations. Moreover, the proposed scheme maintains steady performance, irrespective of γ_1.

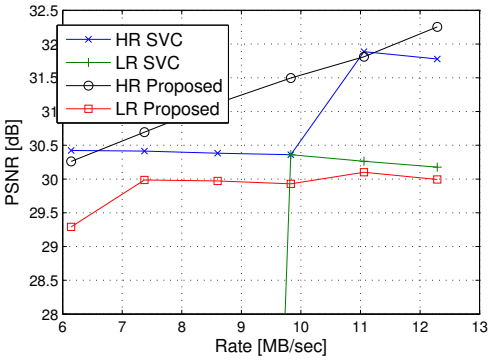

Fig. 7. Average video quality vs. HR client class multicast rate (Ballet).

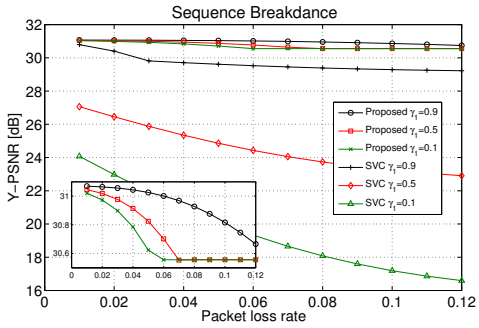

Fig. 8. Average video quality vs. packet loss rate for three different γ_1 values (Breakdance). The inset shows the zoomed-in high PSNR region.

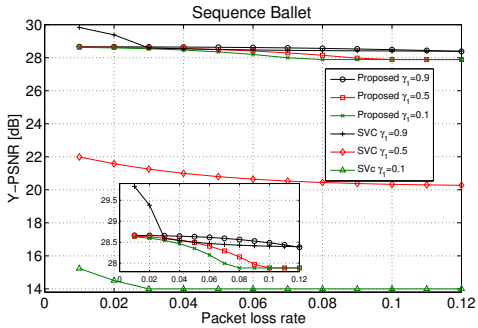

Fig. 9. Average video quality vs. packet loss rate for three different γ_1 values (Ballet). The inset shows the zoomed-in high PSNR region.

3) $\{\gamma_i\}$ mismatch: Figure 10 and Figure 11 examine the sensitivity of our optimization to an incorrect γ_1 value. That is, the joint source-channel coding is optimized with respect to one value of γ_1, however, the one used in practice, when the content is delivered, is actually different. Thus, we have a mismatch between the considered and actual values of γ_1. In these experiments, we optimize our system for $\gamma_1 = 0.1$ or $\gamma_1 = 0.9$, and examine its performance, expressed through the value of the objective function in (5), for $\gamma_1 = 0.5$. For a reference, we include in Figure 10 and Figure 11 the corresponding performance graphs in the absence of γ_1 mismatch. Analytical expressions from Secs III and IV are used to evaluate system performance. It can be seen that our system is robust to parameter mismatch, experiencing no more than a 1dB performance degradation, for all simulated examples. This is due to averaging over all client classes and all 29 views. Moreover, a rate-optimal solution that maximizes the total number of received packets usually provides a solution close to the distortion-optimal one irrespective of γ. Note that the 'mismatch' curves do not necessarily show monotonic behavior, since different non-optimal schemes are used for different packet loss rates.

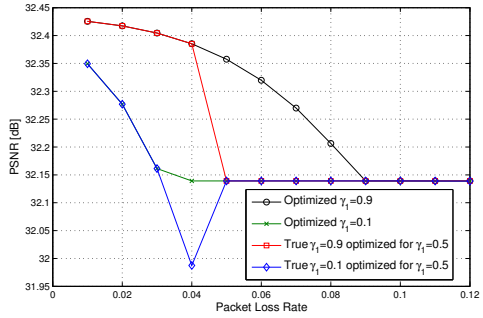

Fig. 10. Objective (5) vs. packet loss rate (Breakdance): γ_1 mismatch.

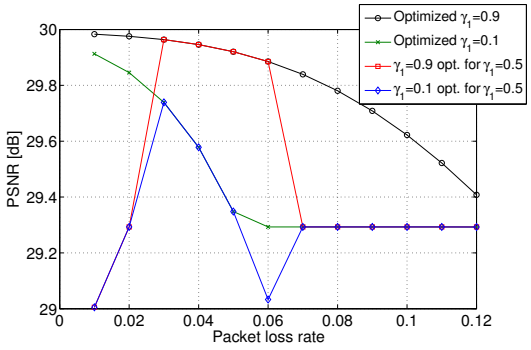

Fig. 11. Objective (5) vs. packet loss rate (Ballet): γ_1 mismatch.

4) View popularity: The following results examine the effect of the clients' viewpoint popularity distribution. We use four client classes, where all γ_i's are set to 0.25. The transmission rate is set to 1.5, 2, 3, and 4 Mb/sec, for Class 1, 2, 3, and 4, respectively. Figures 12-15 show video quality achieved at each viewpoint for Class 1 and 4, for the two video sequences at packet loss rate of 0.1 (10%). The source-channel rate allocation is optimized via the full-search technique from Section IV-C, and the EW-RLC window selection probabilities λ_i's are selected such that the client class data rate constraints are met. All results are averaged after 1000 simulations. One can see from Figures 12 and 14 that the sharp Gaussian (peaky) distribution has a clear PSNR peak while the multi-peak Gaussian distribution has two obvious peaks, in the case of Class 1. This outcome occurs because of the low coding rate that is available in the source coding optimization process for Class 1 so that only the pronounced views have been allocated non-zero rates. For the same reason, the resulting PSNR values for the uniform and peaky view-popularity distributions overlap in Figure 14, in the case of views 4-8. These phenomena are less visible in the case of Class 4, as illustrated in Figures 13 and 15, because of the higher operational data rate that is available then, which resulted in allocation of non-zero rate to multiple views in the optimization process.

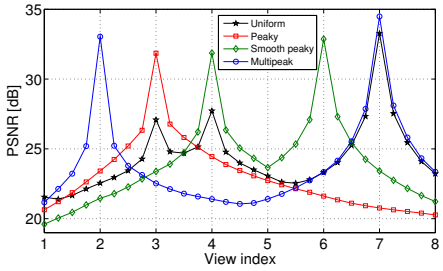

Fig. 12. Video quality per viewpoint for different popularity distributions (Ballet: Client class 1).

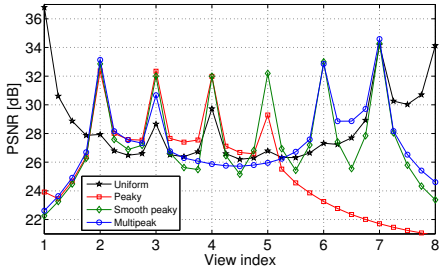

Fig. 13. Video quality per viewpoint for different popularity distributions (Ballet: Client class 4).

5) Three client classes: Figures 16 and 17 show the value of the objective function in (5) vs. the packet erasure rate, in the case of three client classes (L1, L2, and L3). The three downlink bandwidth values associated with the client classes are 1.25Mbps (L1), 2.45Mbps (L2), and 9.8Mbps (L3). λ_1 and λ_2 are set to 0.3 and 0.6. While *H.264/SVC* cannot support in this case class L1 clients for packet loss rates greater than 0.02, our system provides three levels of acceptable video quality for L1, L2, and L3 clients, across a large range of packet loss rate values, as seen from the figures. Note that the

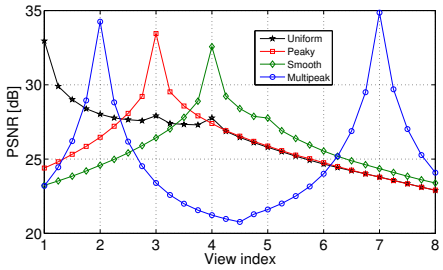

Fig. 14. Video quality per viewpoint for different popularity distributions (Breakdance: Client class 1).

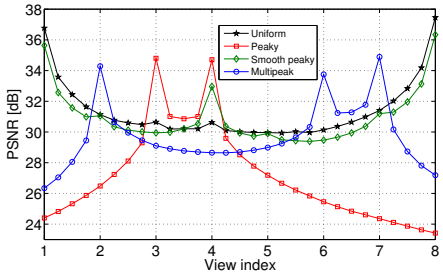

Fig. 15. Video quality per viewpoint for different popularity distributions (Breakdance: Client class 4).

performance of *H.264/SVC* is better for the highest class at very low erasure rates due to the fact that at very high source rates, SVC outperforms the proposed scheme.

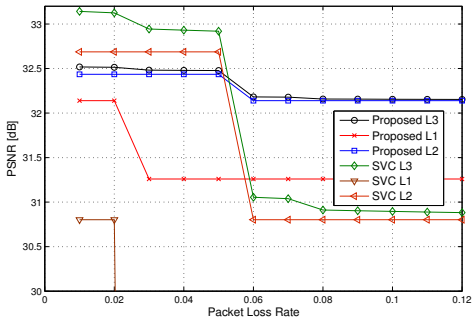

Fig. 16. PSNR [dB] for each client class vs. packet loss rate (Breakdance): Three client classes.

6) Local vs. full search: In Figure 18 and Figure 19, we compare the two optimization methods we designed for channel coding in Section IV-D: full search and low-complexity local search, for two and three client classes. The client class downlink bandwidth values are selected as 4.9Mbps and 9.5Mbs (two classes), and 1.25Mbps, 2.45Mbps, and 9.8Mbps (three classes). The γ_i's are all set to 1/2 and 1/3, for the case of two and three classes, respectively. It can be seen that the proposed local search method always finds an allocation that delivers average video quality that is practically identical to that for the full search method, in the case of two client

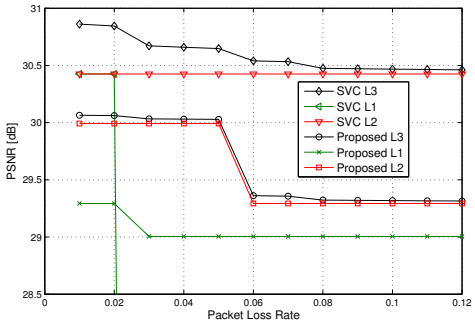

Fig. 17. PSNR [dB] for each client class vs. packet loss rate (Ballet): Three client classes.

classes. When the number of classes is three, the performance degradation due to the local search optimization does not exceed 0.4dB, as seen from Figure 18 and Figure 19. The performance gap stems from the higher likelihood that the local-search method will end up in a local minimum in this case.

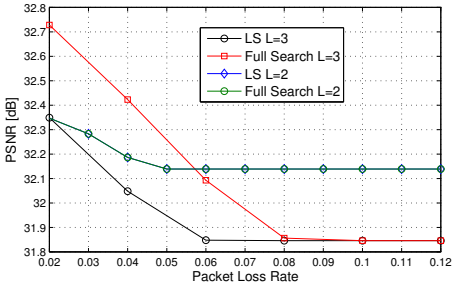

Fig. 18. Objective (5) vs. packet loss rate (Breakdance): Local vs. full search.

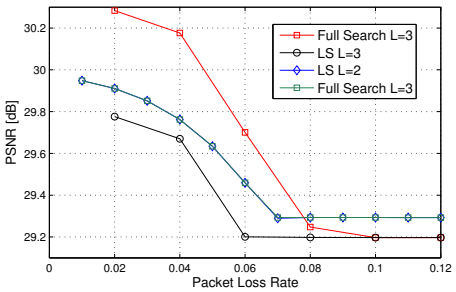

Fig. 19. Objective (5) vs. packet loss rate (Ballet): Local vs. full search.

We measure the execution time of our optimization algorithms in order to assess their complexity in this context. For three client classes, the local search algorithm found the solution after 38 seconds, while the exhaustive search method required 38 minutes. We measured these quantities on an Inter Core 2 CPU 6700 2.66GHz processor with 2GB RAM running MATLAB2011 on Windows XP OS. We anticipate that their

values will be much lower in the case of C/C++ implementation of the two optimization methods from Section IV-D.

7) Comparison to [23] for multiple client classes: Figure 20 shows the achieved average video quality – the objective function in (5) – in the case of four client classes, as a function of the aggregate transmission rate for all four layers. We examine three prospective erasure rates in this case. The benchmark method here represents the system *Toni et al.* that was introduced earlier. The γ_i's are set to 0.25. (Note that in [23], the γ distribution is not explicitly taken into account since the transmission is over a peer-to-peer network.) For the benchmark scheme, similarly to [23], we form four source layers such that the first layer contains H.264-compressed captured Views 1 and 8, layer 2 contains Views 3 and 6, layer 3 Views 2 and 5, and layer 4 Views 4 and 7. One source layer is put in each RLC window. The source-channel rate allocation is found using exhaustive search under transmission rate constrains for each client class. The λ_i's are set to ensure that individual transmission rate constraints for each class are satisfied.

Each point is obtained after averaging over 1000 simulations. The lowest aggregate transmission rate corresponds to 40, 60, 80, 100 packets in layers 1-4, respectively, and the highest to 80, 120, 160, and 220 packets.

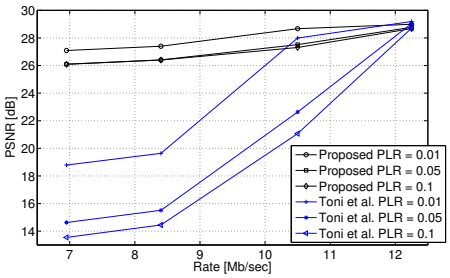

Fig. 20. Objective (5) vs. rate (Ballet): four client classes.

One can see that the proposed scheme significantly outperforms the benchmark method at low and medium transmission rates, and across all erasure rates, while having alike performance at high transmission rates. Competitive performance of the benchmark scheme at the high end of transmission rates is due to the better performance of H.264/SVC at high encoding rates, in the case of a transmission-error-free environment.

The video quality achieved by our method (P) and the benchmark (B) for every client class (C1, C2, C3, and C4) is shown in Figure 21 as a bar graph. The four numbers across every group of bars represent the transmission rate constraints associated with every layer $l = 1, \ldots, 4$ in Mb/sec, while the numbers on the horizontal axis represent the corresponding aggregate transmission rate of all four layers. Note that the benchmark scheme does not succeed to deliver any layer to client classes 1 and 2 at the lowest two aggregate transmission rates, and it still fails to do that for the next aggregate transmission rate point (10.5 Mbps) in the case of class 1. On the other hand, it delivers the highest quality to class C4 in the case when transmission bandwidth is plentiful (the last rate

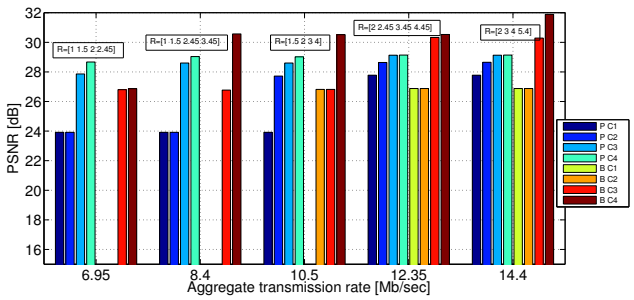

Fig. 21. View-averaged PSNR [dB] for the four classes vs. rate (Ballet).

point examined on the horizontal axis). Our solution instead provides a much better balance in terms of video quality distribution across the four client classes, for every aggregate bandwidth value examined in Figure 21, e.g., even the client class with the smallest transmission bandwidth (C1) is ensured basic quality in all cases.

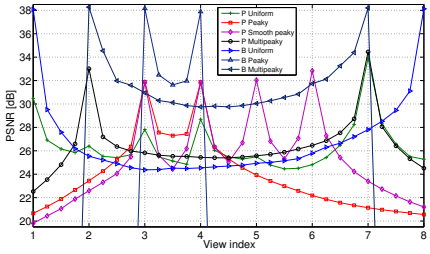

Fig. 22. PSNR [dB] per view for four client popularity distributions (Ballet - client class 1).

Figure 22 shows the video quality per view for client class 1. The transmission rates are set to 2, 2.45, 3.4 and 4.45Mb/sec for the four client classes, respectively. In the case of the uniform view popularity distribution, one can see that the benchmark scheme selects View 1 and View 8 as reference views and encoded them at very high quality. The remaining viewpoints in between exhibit much lower quality, as they can only be synthesized via DIBR (as the transmission bandwidth is limited, they cannot be encoded as well) and the reconstruction quality of such views reduces considerably as their distance from the reference views increases. In the case of non-uniform view popularity distributions, *Toni et al.* again selects to encode the most popular captured views only, which leads to poor reconstruction for the remaining viewpoints at the client, as seen from Figure 22.

In contrast to this, the proposed scheme with uniform distribution leads to a minor variation in reconstruction quality across all reconstruction viewpoints (captured and virtual). This is because the eight captured (actual) views are always encoded and sent and three synthetic viewpoints are generated between each two neighboring actual views, making the distance between the synthetic viewpoints and the captured viewpoints small (hence DIBR is very effective) and uniform across all viewpoints (hence low quality variations). On the

other hand, the proposed scheme with other, non-uniform, distributions places more weight on the popular views resulting in the increased coding rate for compressing these views which leads to a higher reconstruction quality of these views as well as increased quality of the neighboring synthetic views. This can be seen by the PSNR peaks at the popular viewpoints and graceful quality degradation when the viewpoint is moved from the popular ones. This is a desirable feature, since in practice a smooth quality transition between viewpoints is expected.

Figure 23 compares the cases of three and four client classes (L), versus the packet loss rate. It can be seen that while the proposed scheme shows graceful degradation as more classes are introduced, the benchmark scheme degrades significantly with the addition of new classes. The total transmission rate, the sum of the transmission rates of all layers, is 8.4Mb/sec and all γ_i's are set to 1/3 and 1/4 for both cases, $L = 3$ and $L = 4$, respectively. For $L = 4$ the transmission rates for the four layers are 1, 1.5, 2.45, and 3.45Mb/sec while for $L = 3$ the rates are 2.35, 2.6, and 3.45Mb/sec. Exhaustive search is used to find the optimal source-channel rate allocation.

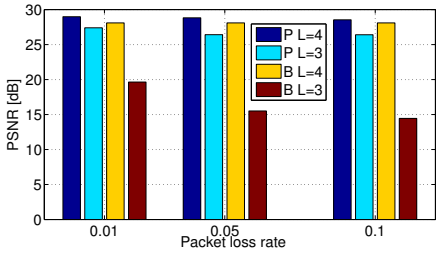

Fig. 23. Objective (5) vs. PLR (Ballet): three vs. four client classes.

VI. Conclusion

We studied the scenario of multicasting VpD MVV to a collection of clients. To address their heterogeneity, we designed a scalable joint source and channel coding scheme for which we formulated a view-popularity-driven source-channel rate allocation and view packing optimization problem that aims at maximizing the expected video quality over all clients, under transmission rate constraints and the clients' view selection preferences. We have shown that our system is superior to state-of-the-art reference systems based on H.264/SVC and the channel coding technique we designed, and H.264 and network coding. Finally, we developed a faster local-search-based method that still optimizes the source and channel coding components of our system jointly, however, at lower complexity, and exhibits only a marginal loss relative to the original exhaustive-search optimization.

Acknowledgment

The authors acknowledge the many constructive comments of the reviewers and the Associate Editor.

References

[1] J. G. Apostolopoulos, P. A. Chou, B. Culbertson, T. Kalker, M. D. Trott, and S. Wee, "The road to immersive communication," *Proc. of the IEEE*, vol. 100, no. 4, pp. 974–990, Apr. 2012.

[2] C. Zhang, Z. Yin, and D. Florencio, "Improving depth perception with motion parallax and its application in teleconferencing," in *Proc. of the IEEE Multimedia Signal Proc. (MMSP)*. Rio de Janeiro, Brazil: IEEE, Oct. 2009.

[3] J. Chakareski, V. Velisavljević, and V. Stanković, "User-action-driven view and rate scalable multiview video coding," in *IEEE Trans. Image Processing*, vol. 22, Sept. 2013, pp. 3473–3484.

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

[5] Y. Morvan, D. Farin, and P. H. N. de With, "Multiview depth-image compression using an extended H.264 encoder," in *Advanced Concepts for Intelligent Vision Systems, Lecture Notes in Computer Sciences*, vol. 4678, 2007, pp. 675–686.

[6] G. Cheung and V. Velisavljević, "Efficient bit allocation for multiview image coding & view synthesis," in *IEEE International Conference on Image Processing*, Hong Kong, September 2010.

[7] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the H.264/AVC standard," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no.9, September 2007, pp. 1103–1120.

[8] T. Wiegand, G. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no.7, July 2003, pp. 560–576.

[9] P. Eder, D. Engel, and A. Uhl, "JPEG2000-based scalable video coding with MCTF," Universität Salzburg Technical Report 2007-04, October 2007.

[10] T. André, M. Cagnazzo, M. Antonini, and M. Barlaud, "JPEG2000-compatible scalable scheme for wavelet-based video coding," *EURASIP J. Image and Video Proc.*, vol. 2007, no. 1, pp. 1–11, 2007.

[11] B. Macchiavello, C. Dorea, E. Hung, G. Cheung, and W. Tan, "Loss-resilient coding of texture and depth for free-viewpoint video conferencing," in *arxiv.org.1305.5464*, May 2013.

[12] A. Hamza and M. Hefeeda, "Energy-efficient multicasting of multiview 3d videos to mobile devices," in *ACM Transactions on Multimedia Computing and Applications*, vol. 8, no.3s, September 2012, pp. 45:2–45:25.

[13] D. Lun, M. Medard, R. Koetter, and M. Effros, "Efficient bit allocation for an arbitrary set of quantizers," in *Physical Communication*, vol. 1, no.1, 2008, pp. 22–30.

[14] D. Vukobratović and V. Stanković, "Unequal error protection random linear coding for erasure channels," in *IEEE Trans. Communications*, vol. 60, May 2012, pp. 1243–1252.

[15] D. Vukobratović, V. Stanković, D. Sejdinović, L. Stanković, and Z. Xiong, "Expanding window fountain codes for scalable video multicast," in *IEEE Trans. Multimedia*, vol. 11, June 2009, pp. 1094–1104.

[16] S. Nazir, D. Vukobratović, V. Stanković, I. Andonović, K. Nybom, and S. Gronroos, "Unequal error protection for data partitioned h.264/avc video broadcasting," in *Multimedia Tools and Application, Springer*, March 2014.

[17] D. Vukobratović, C. Khirallah, V. Stanković, and J. Thompson, "Random network coding for multimedia delivery services in LTE/LTE-advanced," in *IEEE Transactions on Multimedia*, vol. 16, no.1, January 2014, pp. 277–282.

[18] N. Thomos, J. Chakareski, and P. Frossard, "Prioritized distributed video delivery with randomized network coding," in *IEEE Trans. Multimedia*, vol. 13, August 2011, pp. 776–787.

[19] Z. Liu, G. Cheung, V. Velisavljevic, E. Ekmekcioglu, and Y. Ji, "Joint source/channel coding for wwan multiview video multicast with cooperative peer-to-peer repair," in *18th Int. PacketVideo Workshop 2010*, Hong Kong, China, December 2010.

[20] C. Hewage, S. Nasir, S. Worrall, and M. Martini, "Prioritized 3d video distribution over ieee 802.11e," in *Future Network and Mobile Summit, 2010*, Florence, Italy, June 2010.

[21] J. Liu, Y. Liu, S. Ci, and R. Yao, "3d visual experience oriented cross-layer optimized scalable texture plus depth based 3d video streaming over wireless networks," in *Journal of Visual Communication and Image Representation*, vol. 25, no.5, July 2014, pp. 1209–1221.

[22] A. Vosoughi, P. Cosman, and L. B. Milstein, "Joint source-channel coding and unequal error protection for video plus depth," in *IEEE Signal Processing Letters*, vol. 22, August 2014, pp. 31–34.

[23] L. Toni, N. Thomos, and P. Frossard, "Interactive free viewpoint video streaming using prioritized network coding," in *International Workshop on Multimedia Signal Processing*, Pula, Italy, October 2013.

[24] E. Kurdoglu, N. Thomos, and P. Frossard, "Scalable video dissemination with prioritized network coding," in *StreamComm-2011*, Barcelona, Spain, July 2011.

[25] M. Maitre, Y. Shinagawa, and M. Do, "Wavelet-based joint estimation and encoding of depth-image-based representations for free-viewpoint rendering," in *IEEE Transactions on Image Processing*, vol. 17, no.6, June 2008, pp. 946–957.

[26] A. Said and W. A. Pearlman, "A new, fast, and efficient image codec based on set partitioning in hierarchical trees," *IEEE Trans. on Circuits and Systems for Video Tech.*, vol. 6, pp. 243–250, 1996.

[27] V. Velisavljević, J. Chakareski, and G. Cheung, "Bit allocation for multiview image compression using cubic synthesized view distortion model," in *Proc. 2nd Int'l Workshop on Hot Topics in 3D (Hot3D)*. Barcelona, Spain: IEEE, Jul. 2011.

[28] G. Cheung, V. Velisavljević, and A. Ortega, "On dependent bit allocation for multiview image coding with depth-image-based rendering," *IEEE Trans. Image Processing*, vol. 20, no. 11, pp. 3179–3194, Nov. 2011.

[29] R. Bellman, *Dynamic Programming*. Princeton University Press, 1957.

[30] C. Zitnick, S. Kang, M. Uyttendaele, S. Winder, and R. Szeliski, "High-quality video view interpolation using a layered representation," in *ACM SIGGRAPH and ACM Trans. on Graphics*, Los Angeles, CA, Aug. 2004, pp. 600–608.

[31] J. Byers, M. Luby, and L. Mitzenmacher, "A digital fountain approach to reliable distribution of bulk data," in *IEEE J. Select. Areas Commun.*, vol. 20, no.8, October 2002, pp. 1528–1540.

PLACE PHOTO HERE

J acob Chakareski completed the M.Sc. and Ph.D. degrees in electrical and computer engineering at the Worcester Polytechnic Institute (WPI), Worcester, MA, USA, Rice University, Houston, TX, USA, and Stanford University, Stanford, CA, USA.

He is an Assistant Professor of Electrical and Computer Engineering at the University of Alabama. He was a Senior Scientist at École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland, where he conducted research, supervised students, and lectured. He also held research positions with Microsoft, Hewlett-Packard, and Vidyo, a leading provider of Internet telepresence solutions. Chakareski has authored one monograph, three book chapters, and over 100 international publications, and holds 5 US patents. His current research interests include graph-based information processing, computer networks, immersive communication, and social computing. Dr. Chakareski is a member of Tau Beta Pi and Eta Kapa Nu. He is a recipient of the Technical University Munich Mobility Fellowship, the University of Edinburgh Chancellors Fellowship, and fellowships from the Soros Foundation and the Macedonian Ministry of Science. He was the recipient of the Texas Instruments Graduate Research Fellowship at Rice University, the Swiss NSF Ambizione Career Development Award, the Best Student Paper Award at the SPIE VCIP 2004 Conference, and the Best Paper Award of the Stanford Electrical Engineering and Computer Science Research Journal for 2004.

He actively participates in technical and organizing committees of major IEEE conferences. He was the Publicity Chair of the Packet Video Workshop 2007 and 2009 and the Workshop on Emerging Technologies in Multimedia Communications and Networking at ICME 2009. He has organized and chaired special sessions on the next generation wireless multimedia at the IEEE International Symposium on Wireless Pervasive Computing 2008 and telemedicine at MMSP 2009. He was the Technical Program Co-Chair of Packet Video 2012 and the General Co-Chair of the IEEE SPS Seasonal School on Social Media Processing 2012. He was a Guest Editor of the Springer PPNA Journals 2013 special issue on P2P Cloud Systems. He was invited panelist at the Globecom 2013 Workshop on Quality of Experience for Multimedia Communications and invitee at the Microsoft Research Faculty Summit 2014. He is a Technical Program Area Chair for ICME 2015 and Demo/Expo Chair for ICME 2016. He is a Guest Editor of the IEEE Trans. on Circuits and Systems August 2015 special issue on Mobile Visual Computing in the Cloud. He is a Senior Member of the IEEE. He is an Advisory Board member of Mainframe2, an innovative visual cloud computing start-up with a bright future. For more information, please visit http://www.jakov.org.

PLACE PHOTO HERE

V ladan Velisavljević (M06-SM12) received the B.Sc. and M.Sc. (Magister) degree from the University of Belgrade, Serbia, in 1998 and 2000, respectively, and the Master and Ph.D. degree from EPFL, Lausanne, Switzerland, in 2001 and 2005. From 1999 to 2000, he was a member of academic staff at the University of Belgrade. In 2000, he joined the Audiovisual Communications Laboratory (LCAV) at EPFL as teaching and research assistant, where he was working on his Ph.D. degree in the field of image processing. In 2003, he was a visiting student at Imperial College London. From 2006 to 2011, Dr. Velisavljevic was a Senior Research Scientist at Deutsche Telekom Laboratories, Berlin, Germany. Since October 2011, he is Senior Lecturer (Associate Professor) at the University of Bedfordshire, Luton, UK, and Acting Head of the Centre for Wireless Research. He has co-authored around 50 research papers published in peer-reviewed journals and conference proceedings and he has been awarded or filed 4 patents in the area of image and video processing. He is the Lead Guest Editor for the special issue on visual signal processing for wireless networks in IEEE JSTSP in February 2015 and he co-organized a special session at IEEE ICIP-2011 on compression of high-dimensional media data for interactive navigation. Dr. Velisavljevic was a TPC co-chair of the Multimedia Computing and Communications Symposium (MCC) at ICNC-2013 and ICNC-2014 and an IEEE ComSoc Multimedia Technical Committee member and Associate Editor for R-Letters 2011-2014. He also serves as a reviewer for a number of IEEE, Eurasip, ACM and Elsevier journals and as a TPC member for a number of conferences. His research interests include image, video and multiview video compression and processing, wavelet theory, multi-resolution signal processing and distributed image/video processing.

PLACE PHOTO HERE

V ladimir Stanković (M03-SM10) received the Dr.-Ing. (Ph.D.) degree from the University of Leipzig, Leipzig, Germany in 2003. From 2003 to 2006, he was with Texas A&M University, College Station, first as Research Associate and then as a Research Assistant Professor. From 2006 to 2007 he was with Lancaster University. Since 2007, he has been with the Dept. Electronic and Electrical Engineering at University of Strathclyde, Glasgow, where he is currently a Reader. He has co-authored 4 book chapters and over 160 peer-reviewed research papers. He was an IET TPN Vision and Imaging Executive Team member, Associate Editor of IEEE Communications Letters, member of IEEE Communications Review Board, and Technical Program Committee co-chair of Eusipco-2012. Currently, he is Associate Editor of IEEE Transactions on Image Processing and IEEE Transactions on Communications. His research interests include user-experience driven image/signal processing and communications and source/channel/network coding.