Multiple Description Coding & Recovery of Free Viewpoint Video for Wireless Multi-Path Streaming

Zhi Liu, *Student Member, IEEE*, Gene Cheung, *Senior Member, IEEE*,
Jacob Chakareski, *Member, IEEE*, and Yusheng Ji *Member, IEEE*

Abstract*—By transmitting texture and depth videos captured from two nearby camera viewpoints, a client can synthesize via depth-image-based rendering (DIBR) any freely chosen intermediate virtual view of the 3D scene, enhancing the user's perception of depth. During wireless network transmission, burst packet losses can corrupt the transmitted texture and depth videos and degrade the synthesized view quality at the client. In this paper, we propose a multiple description coding system for multi-path transmission of free-viewpoint video, with joint inter-view and temporal description recovery capability. In particular, we encode separately the even frames of the left view and the odd frames of the right view, and transmit them as one description on one path. The second description comprises the remaining frames in the two views and is transmitted over a second path. If the receiver receives only one description due to burst loss in the other path, the missing frames in the other description are partially reconstructed using our frame recovery procedure. First, we construct two recovery candidates for each lost pixel in a frame. The first candidate is generated via temporal super-resolution from its predecessor and successor frames in the same view. The second candidate is generated via DIBR from the received frame of the same time instance in the other view. Next, we select the best pixel candidates one patch at a time, where an image patch corresponds to a neighborhood of pixels with similar depth values in the 3D scene. Near-optimal source and channel coding rates for each description are selected using a branch-and-bound method, for given transmission bandwidth on each path. Experimental results show that our system can outperform a traditional single-description / single-path transmission scheme by up to 5.5dB in Peak Signal-to-Noise Ratio (PSNR) of the synthesized intermediate view at the client.

I. Introduction

The popularity of stereoscopic video, where two texture images captured from two closely spaced cameras are shown respectively to each of the viewer's eyes in order to induce a perception of depth in the 3D scene, is indisputable. However, it is known that *motion parallax* [1], where the viewer's head movement triggers a corresponding shift in the viewing perspective of the observed scene, represents an even stronger stimulus of depth perception [2]. With stereoscopic video, the same two views are shown to the viewer's two eyes regardless of how much the viewer moves his head. This results in physical objects in the 3D scene appearing as unnatural flat layers, which is undesirable.

One technology to enable motion parallax is *free viewpoint video* [3]. At the sender, a large 1D array of closely spaced cameras synchronously captures texture and depth images[1] of the same 3D scene from slightly different viewing angles. The sender then transmits texture and depth maps of two adjacent captured views—a format known as *texture-plus-depth* [6]—that are closest to the viewer's viewing perspective of the scene, as governed by his head movement that is dynamically tracked over time [7]. (The two transmitted views are denoted as left and right views in the sequel.) The viewer can then synthesize any intermediate virtual view that corresponds to his present viewpoint of the scene via DIBR [8], using texture and depth maps of the two captured views as references. This results in an enhanced 3D depth perception via the aforementioned motion parallax.

If the communication path between the sender and receiver is over wireless links that are known to be burst-loss prone due to shadowing, slow channel fading, and interference [9], then the resulting packet losses of texture and depth video data are difficult to overcome and can severely affect the synthesized view quality. This is especially true since the interactivity of free viewpoint video mandates stringent playback deadline requirements at the receiver [10]. Therefore, packet loss recovery strategies based on automatic retransmission

[2]This work was supported in part by JSPS Grant-in- Aid for Scientific Research A (23240011) and JSPS Research Fellowships for Young Scientists.

[3]The authors Zhi Liu, Gene Cheung, and Yusheng Ji are with The Graduate University for Advanced Studies, National Institute of Informatics, 2-1-2, Hitotsubashi, Chiyoda-ku, Tokyo 101−8430, Japan (e-mail:{liuzhi,cheung,kei}@nii.ac.jp).

[4]The author Jacob Chakareski is with The University of Alabama, Tuscaloosa, AL, USA (e-mail:jakov@jakov.org).

[1]Texture image is a digital (color) image that includes color information (e.g., red (R), green (G), or blue (B)) for each pixel. A depth image comprises per-pixel distances between physical objects in the 3D scene and the capturing camera. It can be either captured directly via a depth sensor [4] or estimated from neighboring texture images using stereo-matching algorithms [5].

request [11], which exhibit round-trip-time delays, are not applicable.

To tackle this challenge, we propose a novel *multiple description coding* (MDC) system for multi-path streaming of free-viewpoint video, with joint inter-view and temporal description recovery capability. Specifically, we construct description D_1 to comprise four sub-streams of data that are encoded separately. They are the even frames of the texture and depth maps of the left view and the odd texture and depth frames of the right view. Similarly, the odd texture and depth frames of the left view and even texture and depth frames of the right view comprise the second description D_2. Each description is transmitted over a disjoint network path. Furthermore, appropriate quantization parameters (QP) and channel coding rates are selected for the sub-streams comprising the two descriptions using an efficient *branch-and-bound* (BB) algorithm.

Like MDC for single-view video [12], if the receiver receives one description but loses the other during transmission, the sole received description can be independently decoded, resulting in reduced, but still acceptable, video quality. Yet, unlike single-view video MDC [12], our MDC is carefully designed so that a lost frame in one description can be partially reconstructed using available frames in the received description, exploiting both temporal and inter-view correlation. Our recovery approach comprises two methods.

In the first method, denoted as *temporal super-resolution* (TSR), for a given lost right-view texture frame[2] $\mathbf{x}_t^r$ at time instant t, we exploit temporal correlation in received neighboring frames $\mathbf{x}_{t-1}^r$ and $\mathbf{x}_{t+1}^r$ in time to interpolate the missing pixels in $\mathbf{x}_t^r$. Yet, unlike traditional TSR methods like [13] where only block-based motion estimation (ME) is performed, we exploit available depth information in the corresponding depth frames $\mathbf{z}_{t-1}^r$ and $\mathbf{z}_{t+1}^r$ to partition the missing texture block into foreground and background sub-blocks for separate ME, leading to a more accurate per-pixel motion field. Finally, when copying the reference sub-blocks to reconstruct the missing block in $\mathbf{x}_t^r$, depending on the sharpness of the sub-block boundary in the reference texture block, we optionally perform *overlapped motion compensation* (OMC) to synthesize a more naturally looking image.

The second method, denoted as DIBR, exploits the inter-view correlation between the received left-view texture frame $\mathbf{x}_t^l$ and missing frame $\mathbf{x}_t^r$. Then, given that most missing pixels in $\mathbf{x}_t^r$ have two recovery candidates (TSR and DIBR), we select the better candidate for each texture pixel at a patch level, where an image patch is a neighborhood of pixels with similar depth values. This ensures consistency of selected candidates within the same object. Through extensive experimentation, we demonstrate that our system outperforms a single-description / single-path transmission scheme by up to 5.5dB in PSNR of the synthesized intermediate view at the receiving client.

The rest of the paper is structured as follows. We first discuss related work in Section II. Then, we describe our free viewpoint video streaming system and our MDC coding scheme in Section III. We present our frame recovery procedure via TSR and DIBR in Section IV and V, respectively. The candidate pixel selection procedure is described at the end of Section V. Next, we discuss our data transport optimization in Section VI. Finally, experimentation and conclusions are presented in Section VII and VIII, respectively.

II. Related Work

We divide our related work review into four sections. We first review related work in multiview video coding and MDC in Sections II-A and II-B respectively. We discuss related work in TSR in Section II-C and then conclude with a discussion of error-resilient streaming of free viewpoint video in Section II-D.

A. Multiview and Free Viewpoint Video Coding

Multiview Video Coding (MVC) [14] is an extension of the single-view video coding standard H.264/AVC [15], where multiple texture maps from closely spaced capturing cameras are encoded into one bitstream. Early work in MVC [14, 16] focused on exploiting the signal redundancy across views using *disparity compensation* for coding gain—matching of code blocks between neighboring view images for efficient signal prediction. However, given that temporal redundancy has already been exploited via MC, and neighboring temporal frames tend to be more similar than neighboring inter-view frames due to the typically high frame rate of captured videos, it was shown that additional coding gain afforded by disparity compensation is noticeable, but not dramatic (around 1dB in PSNR [14]). Given that our goal is loss-resilient video streaming, for simplicity we perform only temporal motion compensation (MC) in our MDC scheme.

The texture-plus-depth format of free viewpoint video [6] is another multi-view representation that encodes texture and depth maps captured from multiple nearby viewpoints, so that a user can also choose intermediate virtual viewpoints between a pair of neighboring captured views for free viewpoint image rendering. Depth maps possess unique piecewise smooth signal characteristics that can be exploited for coding gain [17–20]. Since we focus on error-resilient streaming of free viewpoint video, for simplicity, we employ the standard H.264 video codec for coding texture and depth maps. Usage of more advanced coding tools is left as future work.

B. Multiple Description Coding

MDC has been proposed for multi-path streaming of single-view video [12, 21–24]. In particular, in [12] the

[2]Frame recovery for left-view texture frame $\mathbf{x}_t^l$ can be performed similarly. Due to its piecewise smooth characteristics, recovery of depth frame $\mathbf{z}_t^r$ is done using DIBR only as described in Section V.

even and odd frames of a video are encoded separately into two descriptions; we follow the same paradigm in our MDC design as well. However, the recovery of a lost description in [12] relies on conventional block-based ME using temporal neighboring frames [25], which does not result in accurate recovery of the motion field per pixel. In contrast, we propose a sub-block-based ME scheme, where a block can potentially be divided into foreground sub-block and background sub-block using available depth information. As a result, we can recover more accurate per-pixel motion information at comparable complexity.

Note that in the multiple description literature for single-view video, there exist studies [23,26] that generalize the number of descriptions to $N > 2$ sent over disjoint network paths. However, it has been shown [22, 23] that video coding performance of a system based on $N > 2$ descriptions drops dramatically, due to the inefficiency of motion compensated video coding when the temporal distance between the target frame and the predictor frame is larger than two [27,28]. This will hold true for free viewpoint video as well, given that the prediction structure in our coding scheme is similar to the single-view video case. Thus, though in theory employing $N > 2$ descriptions is possible, we encode only two descriptions in our proposed system.

The work in [29] exploited a hierarchical B-frame structure to construct multiple descriptions. In contrast, in our work only I- and P-frames are considered, which has the advantage of minimum decoding delay[3]—important for free viewpoint video streaming, where a user can interactively switch views in real-time as the video is played back. Moreover, [29] studied the single-view video scenario instead of free viewpoint, and in their context the focus is on reconstruction of the single-view video at higher quality, when multiple frame-subsampled versions of the same content are received. In contrast, in our MDC work we focus on how a lost description can be recovered by exploiting inter-view and temporal correlation in the received description.

C. Temporal Super Resolution

TSR interpolates frame $\mathbf{x}_t$ at time t using its two temporal neighbors $\mathbf{x}_{t-1}$ and $\mathbf{x}_{t+1}$, by exploiting their temporal correlation. TSR is used in applications such as temporal down-sampling for low-bitrate video streaming [30]. The most common method for TSR remains block-based ME and MC. For example, [31] proposed to perform forward ME from frame $\mathbf{x}_{t-1}$ to $\mathbf{x}_{t+1}$ and backward ME from $\mathbf{x}_{t+1}$ to $\mathbf{x}_{t-1}$, and then selects the better option. The shortcoming of [31] is that it cannot guarantee at least one candidate per missing pixel in the target frame. In our MDC scheme, because DIBR does not provide inter-view recovery candidates for all missing pixels (due to disocclusion, out-of-view problems, etc.), we must construct a temporal recovery candidate per-pixel in the missing frame. We thus elect the bidirectional ME (BME) approach taken in [13,25], described in Section IV. Note, however, that we perform sub-block ME and OMC using the available depth information, which is not considered in [13,25].

Finally, we note that exploiting spatio-temporal correlation in the context of stereoscopic video coding using distributed source coding principles have been examined in [32–34], where side information video frames are generated by combining disparity-based and temporal-based data recovery.

D. Error-resilient Free Viewpoint Video Streaming

While the problem of error-resilient streaming of single-view video has been extensively studied, error-resilient streaming of free viewpoint video is an emerging topic. In [35], a scheme to minimize the expected synthesized view distortion based on reference picture selection (RPS) [36] at the block level was proposed for depth maps only. In a follow-up work, [37] extended the idea proposed in [35] to encoding of both texture *and* depth maps. Lastly, in [38] the work is extended to the case where optimization of source coding rate (via an optimal selection of QP) is included into the error-resilient streaming framework. However, in [35, 37,38] a simple independent and identically distributed (iid) packet loss model is adopted, while in wireless networks it is more common to observe burst packet loss events [9]. This motivates our current work on MDC of free viewpoint video for transmission over multiple independent network paths.

III. Multiple-path Free Viewpoint Video System

A. Free Viewpoint Video Streaming System

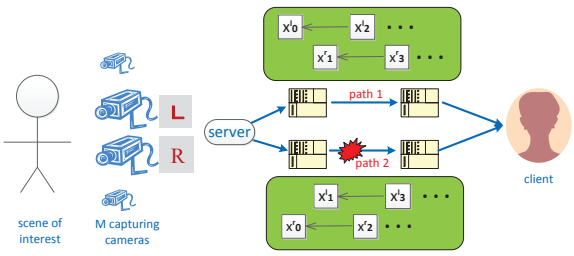


Fig. 1. Overview of our streaming system for free viewpoint video encoded in two descriptions for transmission over two disjoint paths.

Our system is illustrated in Fig. 1. We assume that there are two disjoint network paths available for transmission of free viewpoint video content to the client. For example, a multi-homed wireless client can have two network interfaces such as 3G cellular and 802.11 Wi-Fi that connect to two orthogonal communication networks [39]. Another example is a community of wireless clients [40] in proximity of each other that collaboratively

[3]A B-frame is correctly decoded only after the past and future predicted frames are correctly decoded, resulting in decoding delay.

pool their wireless network resources together for a high-priority task. Yet another example is multi-source video streaming [41], where the video content resides in both a remote server and a nearby peer who has cached the content and can help with the distribution. In any of these cases, the free viewpoint video content can be transmitted to the client simultaneously over two disjoint network paths. At the same time, we assume that the client sends periodic feedback to the sender(s) over these two paths, so that the sender(s) knows the intermediate virtual view requested at any time. The two disjoint network paths will in general be characterized by different transmission bandwidth and packet loss statistics. Since the paths are disjoint, packet loss events on one link are independent from loss events on the other.

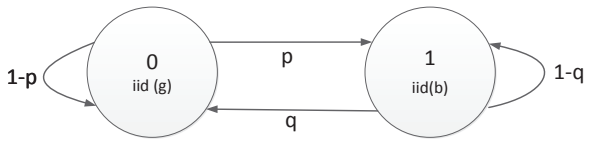


Fig. 2. Gilbert-Elliott packet loss model: transitions between the two states (good - 0 and bad - 1) with probabilities p and q. The packet loss probabilities in good and bad states are g and b, respectively.

We assume that each network path exhibits end-to-end burst packet loss characteristics modeled by a *Gilbert-Elliott* (GE) model [42]. Burst packet losses are common in wireless links due to shadowing, slow path fading, and interference [9]. As illustrated in Fig. 2, a GE model has state transition probabilities p and q to switch between its good (0) and bad (1) states. A group of packets transmitted during a good or bad channel state experience *independent and identically distributed* (iid) packet losses, with probabilities g and b, respectively.

B. Free Viewpoint Video Representation

We assume that the free viewpoint video content is encoded in the now popular *texture-plus-depth* format [6]. In a nutshell, an array of closely spaced cameras capture texture and depth maps (images) from different viewpoints (see [43] for an example camera setup). Depending on the intermediate virtual view currently requested by the client (based on currently tracked head position [1], for example), texture and depth maps from the *two* nearest camera viewpoints (left and right views) will be encoded for transmission[4]. We further assume that the two transmitted views are rectified in a pre-processing step [45].

Using texture and depth maps from two captured views as references, a novel image as observed from an intermediate virtual view chosen by the client can be synthesized via DIBR. This is essentially a pixel-to-pixel mapping procedure that translates texture pixels in the reference camera views to the virtual view image, where the mapped locations are determined by known camera parameters and the corresponding depth pixel values. Spatial regions in the virtual view that are occluded by foreground objects and thus not visible in the reference views are called *disocclusion holes*. They are in general difficult to fill; there exist depth-based inpainting methods in the literature [46–48] that provide satisfactory solutions in typical cases. Using the two closest camera views as references for DIBR ensures that the sizes of the resulting disocclusion holes in the virtual image are small.

[4]Using more than two captured views typically does not increase the synthesized view quality noticeably, while using only a single captured view for synthesis leaves large disocclusion holes, resulting in poor synthesized view quality, as shown for example in [44]. Thus, we also assume that two and only two captured views are transmitted.

C. Multiple Description Construction

We encode texture and depth videos from the left and right views as follows. We first perform standard MC predictive video coding, such as H.264 [15], respectively on the odd and even frames of the left-view texture video, $\mathbf{x}^l_1, \mathbf{x}^l_3, \ldots$, and $\mathbf{x}^l_0, \mathbf{x}^l_2, \ldots$, thereby creating two streams $\mathbf{X}^l_o$ and $\mathbf{X}^l_e$. Similarly, we encode the odd and even frames of the left-view depth video, as well as the odd and even frames of the right-view texture and depth video, into the corresponding streams $\mathbf{Z}^l_o$, $\mathbf{Z}^l_e$, $\mathbf{X}^r_o$, $\mathbf{X}^r_e$, $\mathbf{Z}^r_o$ and $\mathbf{Z}^r_e$. This procedure of encoding even and odd frames separately into different streams is reminiscent of previous MDC schemes for single-view video [12]. Note that since the temporal distance between the consecutively coded frames is two (rather than one frame as in conventional video coding), our MDC results in a slightly larger source coding rate.

Note also that because a depth frame provides only geometric information for viewpoint image rendering and is not itself observed directly by users, how to select QPs for texture and depth maps for optimal synthesized view quality is a non-trivial problem [49]. We will discuss our proposed QP selection for texture and depth videos in the left and right views in Section VI.

Given the encoded streams, we construct two descriptions D_1 and D_2 as follows. First, we bundle the streams $\mathbf{X}^l_e$, $\mathbf{Z}^l_e$, $\mathbf{X}^r_o$, and $\mathbf{Z}^r_o$ into description D_1; *i.e.*, D_1 is composed of the left-view even frames and right-view odd frames. Then, we bundle the remaining streams $\mathbf{X}^l_o$, $\mathbf{Z}^l_o$, $\mathbf{X}^r_e$, and $\mathbf{Z}^r_e$ into description D_2; *i.e.*, D_2 is composed of the left-view odd frames and right-view even frames. D_1 and D_2 are transmitted to the client via paths one and two, as illustrated in Fig. 1.

The descriptions are designed such that even if only one description is received, the client can reconstruct the missing frames of the other description by exploiting the inherent temporal and inter-view correlation that the descriptions feature. See Fig. 3 for an illustration. Specifically, for each pixel in a lost frame, we reconstruct two recovery candidates. The first candidate is reconstructed via TSR using neighboring temporal frames of the same

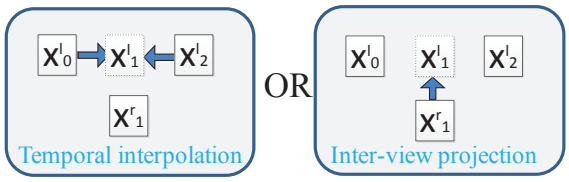


Fig. 3. Illustration of the recovery procedure.

view. The second candidate is reconstructed via DIBR using a frame of the same time instant in the opposing view. Given the recovery candidates, we then select the final reconstruction of the missing data at a patch level, where each image patch is a neighborhood of pixels with similar depth values. Doing so means we achieve reconstruction consistency among neighboring pixels of the same object.

IV. Temporal Super-Resolution-based Frame Recovery

Let texture frame $\mathbf{x}_t^r$ be lost during transmission. The TSR recovery procedure comprises a number of computational steps that are outlined in Figure 4 and are explained next.

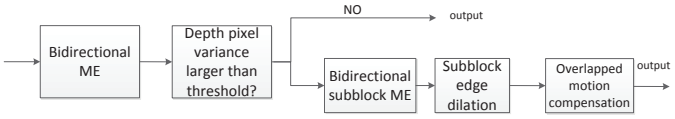


Fig. 4. Flow diagram of the proposed TSR-based frame recovery method.

A. Bidirectional Motion Estimation

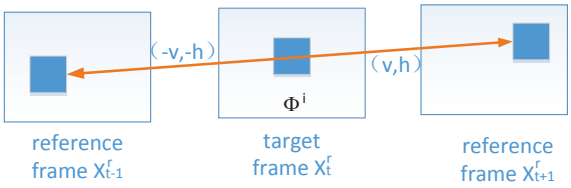


Fig. 5. Bidirectional motion estimation (BME) to recover missing block in target frame $\mathbf{x}_t^r$ via block matching in neighboring temporal reference frames $\mathbf{x}_{t-1}^r$ and $\mathbf{x}_{t+1}^r$.

We first perform BME at the block level. Specifically, for each given non-overlapping $K \times K$ pixel block Φ_p, specified by its upper-left corner pixel $p = (i, j)$ in the target missing frame $\mathbf{x}_t^r$, we search for two similar blocks in the reference frames $\mathbf{x}_{t-1}^r$ and $\mathbf{x}_{t+1}^r$ at locations $(i-v, j-h)$ and $(i+v, +h)$, respectively. In other words, we search for the two best-matched blocks in $\mathbf{x}_{t-1}^r$ and $\mathbf{x}_{t+1}^r$ such that a *half* of their temporal motion vector (MV) will place the block at location p in frame $\mathbf{x}_t^r$. Fig. 5 shows an example of BME.

Assuming that the *sum of absolute differences* (SAD) is used as a matching criteria, the best MV (v_p, h_p) for block $\Phi_{(i,j)}$ in the target frame $\mathbf{x}_t^r$ is given by:

$$(v_p, h_p) = \arg\min_{(v,h)} SAD\left(\mathbf{x}_{t-1}^r(\Phi_{(i-v,j-h)}),\ \mathbf{x}_{t+1}^r(\Phi_{(i+v,j+h)})\right) + \lambda\left(|v-\bar{v}_p| + |h-\bar{h}_p|\right) \quad (1)$$

where $(\bar{v}_p, \bar{h}_p)$ is the weighted average of the MVs of the causal neighboring blocks of Φ_p. The additional regularization term in (1) enforces piecewise smoothness of the motion field. Note that the search is performed at 1/2-pixel precision, interpolated from full-pixel resolution using bilinear filtering[5].

$\bar{v}_p$ is computed as

$$\bar{v}_p = \frac{\sum_{q\in\mathcal{N}_p} w_q v_q}{\sum_{q\in\mathcal{N}_p}^{s} w_q}, \qquad w_q = \exp\left\{-\frac{|\bar{z}_t^r(\Phi_p) - \bar{z}_t^r(\Phi_q)|}{\sigma^2}\right\}, \quad (2)$$

where $\mathcal{N}_p$ denotes the set of causal neighboring blocks of Φ_p, $\bar{z}_t^r(\Phi)$ denotes the arithmetic mean of depth values in block Φ of depth frame $\mathbf{z}_t^r$, and σ is a chosen parameter. $\bar{h}_p$ is written in the same form as $\bar{v}_p$ with h_q replacing v_q. Given unique MV (v_p, h_p) for block Φ_p in frame $\mathbf{x}_t^r$, we can compute the average of blocks $\mathbf{x}_{t-1}^r(\Phi_{i-v_p,j-v_p})$ and $\mathbf{x}_{t+1}^r(\Phi_{i+v_p,j+h_p})$, to reconstruct block Φ_p in $\mathbf{x}_t^r$.

Ideally, instead of block-level motion, *pixel-level* motion would provide more accurate information, since a given block can contain parts of more than one object with different motion vectors. However, finding pixel-level motion via optical flow techniques [50] is computationally expensive. To overcome the shortcomings of both block-based BME and optical flow, we propose an alternative *arbitrary-shaped sub-block BME* that uses the available information in depth frames $\mathbf{z}_{t-1}^r$ and $\mathbf{z}_{t+1}^r$.

Specifically, given a texture block in the reference frame $\mathbf{x}_{t-1}^r$, we first check if the variance of the corresponding depth block in depth frame $\mathbf{z}_{t-1}^r$ is large. If so, we partition the texture block into two sub-blocks along an edge similar to the corresponding depth block discontinuity. The partition edge in the reference texture block in frame $\mathbf{x}_{t-1}^r$ is then translated to a partition in the target block in missing frame $\mathbf{x}_t^r$, dividing the target block into sub-blocks. We then perform sub-block BME following the previously described BME procedure. Finally, OMC is optionally performed to avoid sharp sub-block boundaries in the reconstructed block.

B. Texture Block Partitioning

Given texture map $\mathbf{x}_{t-1}^r$ and depth map $\mathbf{z}_{t-1}^r$, block support Φ_p at pixel p—denoted by a sequence of offsets from p, *i.e.*, $(0,0), (0,1), \ldots, (K-1, K-1)$—can be partitioned into two non-overlapping sub-block supports Φ_p^1 and Φ_p^2 (*e.g.*,

[5] *Bilinear* interpolation is also used in H.264 [15] to increase the resolution from half-pel to 1/4-pel for a more accurate ME. For complexity reasons, we perform BME only at half-pel resolution.

foreground and background objects), where $\Phi_p = \Phi_p^1 \cup \Phi_p^2$ and $\emptyset = \Phi_p^1 \cap \Phi_p^2$. Hence the texture pixel block $\mathbf{x}_{t-1}^r(\Phi_p)$ is also the union set $\mathbf{x}_{t-1}^r(\Phi_p^1) \cup \mathbf{x}_{t-1}^r(\Phi_p^2)$.

The first step of macroblock partitioning is to compute the variance of the corresponding depth block $\mathbf{z}_{t-1}^r(\Phi_p)$. If the variance is smaller than a pre-defined threshold T_d (indicating how likely the block contains more than one object), the block will not be partitioned.

If the variance is larger than T_d, the depth block will be partitioned into two sub-blocks, each with depth pixels above and below the arithmetic mean $\bar{z}_{t-1}^r(\Phi_p)$, respectively. Assuming block $\mathbf{z}_{t-1}^r(\Phi_p)$ contains only one foreground object (small depth value) in front of a background (large depth value), this method can segment pixels into two correct sub-blocks. This statistical approach has been shown to be robust and has low complexity [51]. Finally, we perform a morphological closing to ensure that each partitioned sub-block represents a contiguous region.

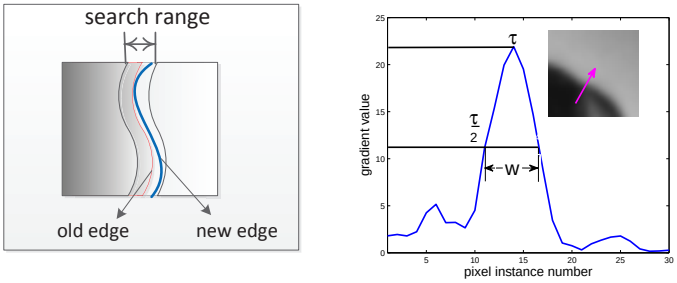
(a) Kendo (b) Pantomime (C) Pantomime

Fig. 6. Illustration showing texture and depth edges may not be perfectly aligned, where the depth edges (white lines) are detected using a 'Canny' edge detector.

In the ideal case, the texture map contains a superset of edges of the depth map. Thus, one can simply reuse the computed depth sub-block boundary for partitioning the texture block as well. However, a known problem in the texture-plus-depth representation [6] is that edges in texture and depth maps may not be perfectly aligned, due to noise in the depth acquisition process. Fig. 6 shows example spatial regions of texture maps overlaid with edges detected in the corresponding depth maps using a Canny edge detector (white lines). One can clearly see that the texture and depth edges are not perfectly aligned.

To circumvent the edge misalignment problem, we perform a simple dilation process. Specifically, we first copy the computed sub-block boundary to the texture block. We next perform edge detection in the texture block. Then, we perform dilation of the depth boundary—thickening of the edge—until a texture edge is found. Fig. 7 shows an example of dilation.

Using the discovered texture edge, the reference block in frame $\mathbf{x}_{t-1}^r$ is also partitioned into two sub-blocks.



(a) depth edge dilation (b) boundary illustration

Fig. 7. (a) edge dilation to identify corresponding texture edge for texture block partitioning. (b) a blurred boundary and the corresponding gradient function across boundary.

Then, the corresponding full block in frame $\mathbf{x}_t^r$ can be partitioned into two sub-blocks as well, by copying the texture edge in $\mathbf{x}_{t-1}^r$ using the MVs computed in Section IV-A.

C. Overlapped Sub-block Motion Estimation

For each partitioned sub-block Φ_p^i in $\mathbf{x}_t^t$, we find its best match in reference frames $\mathbf{x}_{t-1}^r$ and $\mathbf{x}_{t+1}^r$, as described in Section IV-A. The only difference is that now we use sub-blocks instead of full blocks. MVs for each sub-block are computed.

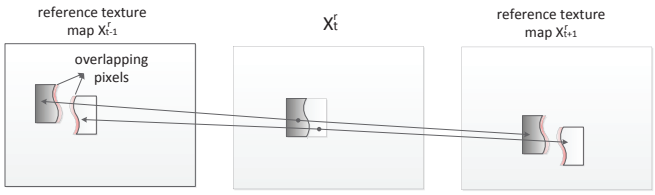


Fig. 8. Illustration of overlapping sub-blocks.

Optionally, we can now perform OMC for better reconstruction of the target block. Specifically, when copying a best-matched sub-block from the reference frame to the missing block in the target frame, we copy the sub-block plus l pixels across the sub-block boundary. The extra copied pixels will be alpha-blended with overlapping pixels copied from the opposing sub-block. See Fig. 8 for an illustration.

The width of the overlapping region l is determined by the sharpness of the texture edge (sub-block boundary) in the reference block of frame $\mathbf{x}_{t-1}^r$. The key insight here is that unlike a depth map which always has sharp edges, object boundaries in the texture map can be blurred due to out-of-focus, motion blur, etc. On the other hand, sub-block motion compensation tends to result in sharp sub-block boundaries. So to mimic the same blur across a boundary in the reference block in frame $\mathbf{x}_{t-1}^r$, we first compute a texture gradient function for a line of pixels in the reference block perpendicular to the sub-block boundary [52].

We then compute the width of the plateau corresponding to the sub-block boundary, which we define as the number of pixels across the plateau at half the peak τ of the gradient plateau. Finally, we set l to be a linear function of the computed width w (*i.e.* more blur, more overlap) as follows:

$$l = round\,(\varepsilon\, w)\ , \tag{3}$$

where ε is a chosen parameter. See Fig. 7(b) for an example of a blurred sub-block boundary, its corresponding gradient function across the boundary, and the width of the plateau w.

V. DIBR-based Frame Recovery and Pixel Selection Framework

Having described how using TSR we can reconstruct a recovery candidate for each pixel in a missing texture frame $\mathbf{x}_t^r$, we now discuss how using DIBR we can reconstruct another recovery candidate. In particular, in Sections V-A and V-B we first discuss how we reconstruct the missing depth map $\mathbf{z}_t^r$, which is easier given its known piecewise smooth characteristics. We then discuss how the corresponding texture map $\mathbf{x}_t^r$ can be reconstructed using the recovered depth map $\mathbf{z}_t^r$ in Section V-C. Finally, we propose a patch-level candidate selection scheme for the final missing texture map reconstruction by choosing between the two recovery candidates.

A. Depth Map Reconstruction

We first synthesize the missing right-view depth map $\mathbf{z}_t^r$ via DIBR [8] using the corresponding left-view depth map $\mathbf{z}_t^l$. Specifically, assuming that the captured camera views are rectified [45], each depth pixel $z_t^l(x,y)$ of row x and column y in the left-view depth map is mapped to a corresponding pixel $z_t^r(x,y')$ in the right-view depth map, where the new column index y' is computed as:

$$y' = y - round\left(\frac{1}{z_t^l(x,y)} * \gamma\right) \tag{4}$$

From (4), we note that the horizontal disparity (pixel translation) is governed by $1/(z_t^l(x,y))$ and the shift parameter γ, which depends on the physical distance between the two capturing cameras.

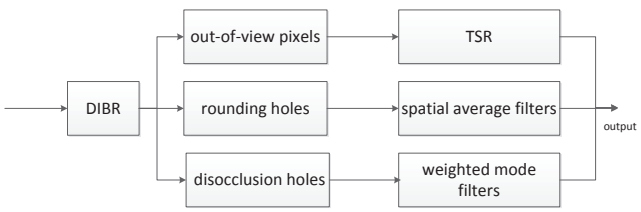


Fig. 9. Flow chart of the proposed depth map recovery method.

In general, depth pixels synthesized via DIBR are more reliable than color pixels, because while color pixels of the same object surface can contain different values at different viewpoints if the surface is non-Lambertian [53], depth pixels are not affected by the object's surface reflectance properties. Hence a depth pixel mapped from the left view to the right view is very likely to be correct. To recover all pixels in the right-view depth map, only missing pixels need to be completed using neighboring spatial and temporal information. We discuss in detail how the missing pixels are filled in this section. Fig. 9 shows the flow diagram of our depth map reconstruction procedure.

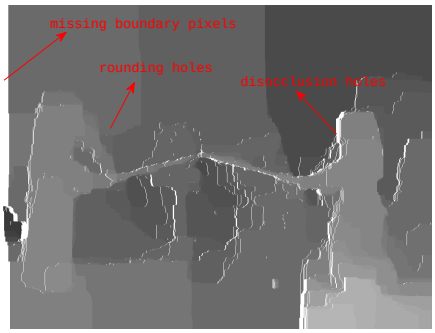


Fig. 10. Three kinds of holes in a synthesized depth map.

DIBR's simple pixel-to-pixel translational mapping results in three types of pixel holes illustrated in Fig. 10. First, there are *out-of-view pixels* in the right-view depth map $\mathbf{z}_t^r$ that are out-of-view in the left-view depth map $\mathbf{z}_t^l$. Second, due to the rounding operation in (4), there might not be any left-view depth map pixels that map to a given pixel location in a right-view depth map. These are called *rounding holes*. Finally, there are spatial regions in the synthesized right-view image that are occluded by foreground objects and therefore not visible in the reference view. These are called *disocclusion holes*.

Due to the operation of rounding to the nearest pixel column, carried out in (4), the thereby created rounding holes are characterized by being narrow in width. Because neighboring depth pixels around a rounding hole usually belong to the same physical object, they have very similar depth values. Hence, simple spatial average filtering can adequately fill in these rounding holes.

By definition, out-of-view pixels in $\mathbf{z}_t^r$ are not in the field of view in depth map $\mathbf{z}_t^l$, and so $\mathbf{z}_t^l$ contains no information to reconstruct out-of-view pixels in $\mathbf{z}_t^r$. Hence we fill out-of-view pixels in $\mathbf{z}_t^r$ by reusing the MVs computed in TSR for the texture candidates used to copy depth pixels from matched blocks in $\mathbf{z}_{t-1}^r$ and $\mathbf{z}_{t+1}^r$ to $\mathbf{z}_t^r$. We focus our discussion on the filling of disocclusion holes next.

B. Filling of Disocclusion Holes in a Depth Map

First, using MVs computed for a texture map during TSR described in Section IV, we initialize the depth values in these disocclusion holes by averaging the

corresponding reference blocks in neighboring temporal depth frames $\mathbf{z}^r_{t-1}$ and $\mathbf{z}^r_{t+1}$. The initialized depth values may not lead to a piecewise smooth solution. Thus, we next employ a *weighted mode filter* (WMF) [54] to sharpen the overly smoothed pixels.

Mathematically, for a pixel location p with neighbors $q \in \mathcal{N}_p$, we first compute a *relaxed histogram* $H(p,d)$ with index d as follows:

$$H(p,d) = \sum_{q \in \mathcal{N}_p} G_s(p-q) \;\; G_f(z^r_t(p) - z^r_t(q)) \;\; G_r(d - z^r_t(q)) \quad (5)$$

where $G_s(p-q)$ is a Gaussian term with the geometric distance between pixel locations p and q as its argument, $G_f(z^r_t(p) - z^r_t(q))$ is a Gaussian term based on the photometric distance between depth values $z^r_t(p)$ and $z^r_t(q)$, and $G_r(d - z^r_t(q))$ is a Gaussian term based on the error between bin index d and $z^r_t(q)$. Note that G_s and G_f are similarly computed in *bilateral filter* [55].

Having computed $H(p,d)$ for different bin indices d, the new depth value $z^r_t(p)$ is the index with the largest histogram value:

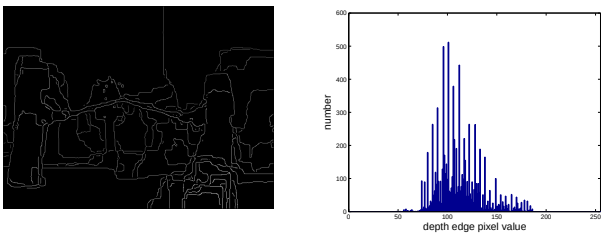
$$z^r_t(p) = \arg\max_d H(p,d) \quad (6)$$

C. Depth Image Based Rendering for Texture Maps

We apply the same procedure we used for reconstructing depth map $\mathbf{z}^r_t$, to generate recovery candidates for texture map $\mathbf{x}^r_t$ via DIBR. Rounding holes are also filled using spatial average filtering. Out-of-view pixels and disocclusion holes are left unfilled. They make up a small percentage of the total pixels, and these pixels will be reconstructed via TSR exclusively. We now discuss how we select between TSR candidates and DIBR candidates for the rest of the texture pixels.

D. Selection of Recovery Candidates

Given the constructed recovery candidates for pixels in a missing texture frame $\mathbf{x}^r_t$, we now describe a procedure to select candidates at a patch level. A patch roughly corresponds to a depth layer of a physical object, so that selecting candidates consistently in a patch would lead to a visually pleasing reconstructed image.



(a) detected edges (b) depth histogram

Fig. 11. Detected edges and depth histogram of detected edges for frame 6, view 3 of the Kendo sequence.

1) Image Segmentation: We first segment a missing texture map $\mathbf{x}^r_t$ into patches based on the reconstructed depth map $\mathbf{z}^r_t$. The algorithm is a variant of the Lloyd's algorithm in *vector quantization* (VQ) [56]. To initialize a segmentation, we first construct a histogram of depth values for the detected edge pixels (edges are detected using a Canny edge detector) and identify the K highest peaks $\hat{z}_k$'s. See Fig. 11 for an example depth image with detected edges in white and corresponding depth histogram of detected edge pixels. For each pair of adjacent peaks $\hat{z}_k$ and $\hat{z}_{k+1}$ in the histogram, we identify a depth value that is a minimum between the peaks and denote it as a boundary b_k. Using $K-1$ boundary values b_k's, we can segment the image into at least K patches, where a patch is a set of contiguous pixels with depth values within two boundaries b_k and b_{k+1}. Fig. 12 shows resulting patches (marked in brown) between two boundary points b_k and b_{k+1} after the segmentation.

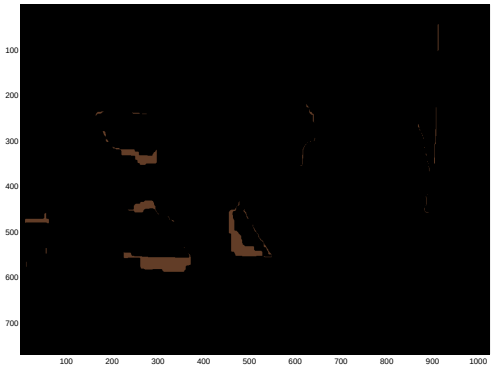


Fig. 12. Patches (in brown) between two boundary points after segmentation.

Having initialized patches, we then perform the following two steps, alternately, until convergence. In the first step, we solve for the *centroid* for each patch, which is the depth value that minimizes the MSE between the centroid and the depth values in the patch. In the second step, given the computed centroids of different patches, each pixel on the border of a patch can be associated with the centroid of a neighboring patch such that its squared error is further minimized. The iteration ends when neither of the two steps can further decrease the MSE.

2) Recovery Candidate Selection: To select recovery candidates between TSR and DIBR for a given patch with centroid c, we examine frames from the most recent correctly received descriptions to see if patches with centroids close to c have smaller reconstruction errors using TSR or DIBR. The idea is that patches with similar depth centroids are more likely to represent the same physical objects. Assuming the same object exhibits similar motion patterns (which affect the performance of TSR) and surface reflectance properties (which affect the performance of DIBR) over time, previous frames provide valuable side information for good selection of recovery candidates for a current frame.

VI. Data Transport Optimization

Having discussed the description recovery method in the previous sections, when the client receives only one description out of two, we describe now how we optimally select the source and channel coding rates for each description, given the available bandwidth and packet loss statistics associated with a transmission path, such that the client's expected video quality is maximized.

Within one description, the video frames (texture or depth) comprising a GOP are split into N sub-groups each with the same number of frames, as shown in Figure 13. Let n_i denote the total number of packets (source plus channel) that are transmitted for sub-group i. k_i denotes the number of source packets only for sub-group i, where texture and depth maps in the description are encoded using different QPs (to be discussed). $n_i - k_i$ packets in sub-group i are for FEC packets, generated as linear combinations of the corresponding k_i source packets. Correct delivery of any k_i of n_i transmitted packets will recover all k_i source packets.

For simplicity, we assume also that the playback deadline of the first frame of each sub-group is the transmission deadline for the whole sub-group. We now formulate an optimization problem for selecting QP Q, for every video frame comprising packet n_i of sub-group i.

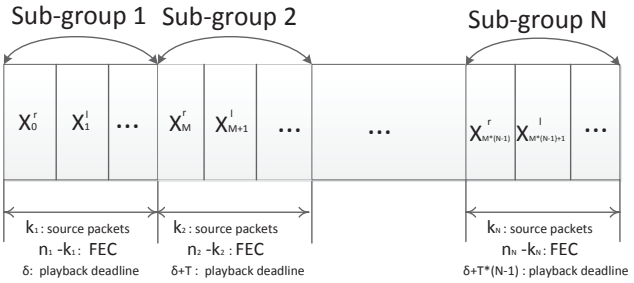


Fig. 13. Illustration of the video frame grouping.

We first introduce preliminaries needed for calculating the probabilities of correctly decoding the video frames given a GE packet loss model, presented also in [40]. We then show how to derive our objective function. Two optimization algorithms are described thereafter.

A. Preliminaries

Let $P(i)$ be the probability of having *at least* i consecutive transmissions during the good state of the GE model, given that transmission started in the bad state. Furthermore, let $p(i)$ be the probability of having *exactly* i good state transmissions between two bad state transmissions, given that transmission started in the bad state. We denote $P(i)$ and $p(i)$ as follows:

$$\begin{aligned} P(i) &= \begin{cases} 1 & \text{if } i = 0 \\ q(1-p)^{i-1} & \text{otherwise} \end{cases} \\ p(i) &= \begin{cases} 1-q & \text{if } i = 0 \\ q(1-p)^{i-1}p & \text{otherwise} \end{cases} \end{aligned} \tag{7}$$

Similarly, we define $Q(i)$ and $q(i)$ as the probability of *at least* i consecutive bad state transmissions, and the probability of *exactly* i bad state transmissions, given transmission starts in good state. Equations for $Q(i)$ and $q(i)$ will be the same as those for $P(i)$ and $p(i)$, with the parameters p and q interchanged.

We can now recursively define the probability $R(m,n)$ of *exactly* m bad state transmissions in n total transmissions, given that transmission started in the bad state:

$$R(m,n) = \begin{cases} P(n) & \text{for } m = 0 \text{ and } n \geq 0 \\ \sum_{i=0}^{n-m} p(i) R(m-1, n-i-1) & \text{for } 1 \leq m \leq n \end{cases} \tag{8}$$

Similarly, the probability $S(m,n)$ of *exactly* m good state transmissions in n total transmissions, given transmission starts in good state, is written in the same form as (8), with $Q(i)$ and $q(i)$ replacing $P(i)$ and $p(i)$ in (8), respectively.

B. System Constraints

Given that the transmission deadline for the first frame in a sub-group i is the delivery deadline for whole sub-group, we can derive the maximum number of packets l_j that can be transmitted by the first j sub-groups as follows:

$$l_j = (\delta + T \times (j-1))B \tag{9}$$

where δ is the initial buffer time, T is the playback duration of the frames in each sub-group, and B is the bandwidth of the transmission path in number of packets per second. This means that the total number of packets $\sum_{j=1}^{i} n_j$ expended for transmission of frames up to and including sub-group i cannot exceed the budget l_i, i.e.,

$$\sum_{j=1}^{i} n_j \leq l_i, \qquad \forall i \in \{1, \ldots, N\} \tag{10}$$

Otherwise, we assume that sub-group i is not correctly delivered since the transmitted packets do not meet the required playback deadline.

C. Probability of Correct Decoding

Due to the predictive nature of video coding, the probability β_i of correctly decoding the frames in sub-group i is a product of: i) the probability α_i of timely and correct recovery of all source packets in sub-group i, and ii) the probability β_{i-1} of correctly decoding the

frames in the previous sub-group $i-1$. Thus, we can write β_i as follows:

$$\beta_i = \beta_{i-1} * \alpha_i \tag{11}$$

We compute β_{i-1} as follows:

$$\beta_{i-1} \approx \prod_{j=1}^{i-1} \alpha_j \tag{12}$$

The assumption of independence of α_j's is an approximation; since the GE packet loss model has memory, and the GE state in which last packet was transmitted in sub-group $i-1$ can affect the probability of correct packet transmission of the following sub-group i. However, if the number of transmitted packets in each sub-group is large, the approximation is nonetheless a good one.

For each sub-group i, the initial state of the G-E model at transmission could be good or bad with different probabilities. We write the probability α_i of correctly recovering all source packets in sub-group i as a weighted sum of α_i^G and α_i^B, which are the probabilities of correctly receiving at least k_i of n_i transmitted packets, given that packet transmission begins at a good or bad state, respectively:

$$\alpha_i = \left(\frac{q}{p+q}\right)\alpha_i^G \;+\; \left(\frac{p}{p+q}\right)\alpha_i^B \tag{13}$$

Assuming first that transmission starts in the good state, m of n_i total packets can be transmitted in good state with probability $S(m, n_i)$. Source packets in sub-group i can be successfully recovered if at least k_i of n_i transmitted packets are correctly delivered. Among r received packets, $r \geq k_i$, r_G can be delivered packets in good state while $r - r_G$ can be delivered packets in bad state. We can hence write α_i^G as:

$$\alpha_i^G = \sum_{m=0}^{n_i} S(m, n_i) \sum_{r=k_i}^{n_i} \sum_{r_G=0}^{r} P_G(r_G, m)\; P_B(r - r_G, n_i - m) \tag{14}$$

where $P_G(x, y)$ and $P_B(x, y)$ are the probabilities of exactly x delivered packets in y iid trials, in the good and bad states, respectively. These quantities can be computed easily using binomial expansion and the packet loss probability g and b, respectively for the good and bad states. α_i^B can be derived similarly.

D. Optimization Problem

The objective we selected for optimization is the rendered virtual view image quality, where the virtual viewpont chosen for evaluation is the middle view between left and right captured views. Inserting superscript R to denote the right path[6], let β_i^R be the probability of correctly decoding frames in sub-group i of the *right* path. Furthermore, denote d_i to be the rendered virtual view's quality, if frames of sub-groups i of both paths are correctly decoded, and d_i^R the reandered view quality if frames of sub-group i of right path only are correctly decoded. d_i and d_i^R are dependent on the QPs used for the two paths: Q_T^R and Q_D^R for texture and depth maps of the right path, and Q_T^L and Q_D^L for texture and depth maps of the left path. We can now write the expected synthesized view quality for sub-group i as:

$$\begin{aligned} D_i \;=\; & \beta_i^R\,\beta_i^L\, d_i(Q_T^L, Q_D^L, Q_T^R, Q_D^R) + \\ & \beta_i^R\,(1-\beta_i^L)\, d_i^R(Q_T^R, Q_D^R) + (1-\beta_i^R)\,\beta_i^L\, d_i^L(Q_T^L, Q_D^L) \end{aligned} \tag{15}$$

We assume here that having frames lost in both descriptions (simultaneous burst loss events on two disjoint transmission paths) is rare and hence not considered.

We can now formally define the optimization problem as follows. The optimization variables are: i) QPs Q_T^R, Q_D^R, Q_T^L, Q_D^L, and ii) the number of transmitted packets n_i^L and n_i^R for each sub-group i in each path. The optimization is subject to the system constraints (10):

$$\max_{Q_T^R, Q_D^R, Q_T^L, Q_D^L, \{n_i^L\}, \{n_i^R\}} \sum_i D_i \quad \text{s.t.} \quad \begin{array}{l} \sum_{j=1}^{i} n_j^L \leq l_i^L, \;\; \forall i \in \{1, \ldots, N\} \\ \sum_{j=1}^{i} n_j^R \leq l_i^R, \;\; \forall i \in \{1, \ldots, N\} \end{array} \tag{16}$$

E. Optimization Algorithms

Solving (16) is complicated, as it involves variables from both transmission paths. We thus elect to solve for variables in one path at a time with variables in the other path fixed, then iterate until convergence.

Fixing a given set of variables in a single path (say the left path), we also iterate between right path QPs Q_T^R, Q_D^R and rates $\{n_i^R\}$ until convergence. When n_i^R's are fixed, we find the optimal QPs Q_T^R and Q_D^R as follows. We alternately perturb Q_T^R and Q_D^R locally in an attempt to increase the objective (16), while respecting the transmission rate constraints. We stop when no further attempt to increase the objective is possible. Given there are only two QPs, this iteration takes little time and converges quickly.

We now discuss two proposals for finding $\{n_i^R\}$, when QPs Q_T^R and Q_D^R are fixed. The first proposal finds the optimal n_i^R's, but has high complexity. The second proposal solves the problem approximately, but exhibits lower computational complexity. Alg. 1 outlines the overall optimization procedure.

1) Dynamic Programming Algorithm: One can search for the optimal n_i^R's to (16), for fixed QPs, using the following recursive algorithm. Let $\Delta_i^R(m)$ be the maximum quality for sub-group i to N, given that m total packets were transmitted for previous sub-groups 1 to $i-1$ *and* the previous groups are all decoded correctly. We know sub-group i must transmit at least k_i^R source packets and no more than $l_i^R - m$ total packets to observe the system constraint (10). We can thus write $\Delta_i^R(m)$ recursively as follows:

[6]Note that unlike previous superscripts l and r that denote left and right captured views, we use here L and R to denote left and right transmission paths.

$$\Delta_i^R(m) = \max_{n_i^R \in \{k_i^R,\ldots,l_i^R-m\}} \begin{array}{l} \alpha_i^R(n_i^R)\ \left[D_i^R + \Delta_{i+1}^R(m+n_i^R)\right] \\ +\ \left(1-\alpha_i^R(n_i^R)\right)\ \sum_{j|j\geq i}\beta_j^L d_j^L \end{array} \quad (17)$$

where correct recovery probability $\alpha_i^R(n_i^R)$ for sub-group i is a function of the number of transmitted packets n_i^R only, and D_i^R, the contribution from sub-group i of the right path, is $D_i^R = \beta_i^L d_i + (1-\beta_i^L)d_i^R$ from (15). Initial call $\Delta_1^R(0)$ would return the optimal solution to (16).

We note that the solution to $\Delta_i^R(m)$ can be stored in entry (i,m) of a *dynamic programming* (DP) table Γ, so that a repeated call to the sub-routine $\Delta_i^R(m)$ can be simply looked up, instead of being actually computed fully. Thus, the complexity of (17) is bounded by the size of the DP table Γ multiplied by the complexity of computing each table entry: $O(N\left(\sum_{i=1}^N l_i^R\right)\left(\max_{i=1}^N l_i^R\right))$.

2) Branch and Bound: Given fixed QPs, using (17) to find the optimal n_i^R's can still be expensive. We thus now present modifications to (17) using a BB method to further limit the search space.

We first compute the objective (15) for a naïve selection of n_i^R's, *e.g.*, equal loss protection where the proportion of FEC packets employed for each sub-group relative to source packets, $(n_i^R - k_i^R)/k_i^R$, is roughly the same for all sub-groups. We denote its objective value as D^e.

When (17) is called, for each possible value n_i^R, we first compute an *upper bound* $\Delta_i^{R,u}(m,n_i^R)$, which is the upper limit of quality given n_i^R is chosen for sub-group i. $\Delta_i^{R,u}(m,n_i^R)$ can be computed recursively similar to (17), but without any search for the optimal n_i^R's:

$$\Delta_i^{R,u}(m,n_i^R) = \begin{array}{l} \alpha_i^R(n_i^R)\ \left[D_i^R + \Delta_{i+1}^{R,u}(m+n_i^R)\right] \\ +\ \left(1-\alpha_i^R(n_i^R)\right)\ \sum_{j|j\geq i}\beta_j^L d_j^L \end{array} \quad (18)$$

$$\Delta_i^{R,u}(m) = \begin{array}{l} \alpha_i^R(l_i^R-m)\ \left[D_i^R + \Delta_{i+1}^{R,u}(m)\right] \\ +\ \left(1-\alpha_i^R(n_i^R)\right)\ \sum_{j|j\geq i}\beta_j^L d_j^L \end{array} \quad (19)$$

In words, (18) states that using the selected n_i^R yields recovery probability $\alpha_i^R(n_i^R)$ and increases the argument passed to future sub-groups by n_i^R. In contrast, (19) states that using *all* permissible packets $l_i^R - m$ for sub-group i will yield correct delivery probability $\alpha_i^R(l_i^R-m)$, *but* we do not increase the argument passed to future sub-groups to seek an upper-bound. Thus, the returned objective value for $\Delta_i^{R,u}(m,n_i^R)$ is from a selection of n_i^R's that may not be feasible (may not observe constraints (10)), and therefore is a *super-optimal* solution.

We use the upper bound $\Delta_i^{R,u}(m,n_i^R)$ as follows. If $\Delta_i^{R,u}(m,n_i^R) < D^e$, then we know that n_i^R cannot lead to a solution that is better than our naïve solution, and hence there is no need to recursively compute $\Delta_{i+1}^R(m+n_i^R)$ in (17), thereby reducing the computation cost.

Similarly, we can also compute the *lower bound* $\Delta_i^l(m,n_i)$, for each selected n_i in (17):

$$\Delta_i^{R,l}(m,n_i^R) = \begin{array}{l} \alpha_i^R(n_i^R)\ \left[D_i^R + \Delta_{i+1}^{R,l}(m+n_i^R)\right] \\ +\ \left(1-\alpha_i^R(n_i^R)\right)\ \sum_{j|j\geq i}\beta_j^L d_j^L \end{array} \quad (20)$$

$$\Delta_i^{R,l}(m) = \begin{array}{l} \alpha_i^R(l_i^R-r)\ \left[D_i^R + \Delta_{i+1}^{R,l}(m+r)\right] \\ +\ \left(1-\alpha_i^R(n_i^R)\right)\ \sum_{j|j\geq i}\beta_j^L d_j^L. \end{array} \quad (21)$$

The definition of $\Delta_i^{R,l}(m,n_i^R)$ in (20) is analogous to that of $\Delta_i^{R,u}(m,n_i^R)$ in (18). (21) returns a performance point when n_i^R is chosen randomly from the feasible range set $\{k_i^R,\ldots,l_i^R-m\}$. Note though that unlike the upper bound in (18), the solution produced by the lower bound in (20) is guaranteed to be feasible.

We use the computed lower bound $\Delta_i^{R,l}(m,n_i^R)$ as follows. If the difference between the upper bound $\Delta_i^{R,u}(m,n_i^R)$ and the lower bound $\Delta_i^{R,l}(m,n_i^R)$ is smaller than a threshold δ, then the permuted random solution produced by (20) is already good enough. Then, again there is no need to recursively compute $\Delta_{i+1}^R(m+n_i^R)$, and we can simply return the computed random solution instead. This also leads to computational savings.

Algorithm 1 Transport optimization

1: Assuming Q_T^R, Q_D^R, Q_T^L, Q_D^L are equal and use dynamic programming or branch and bound to compute the optimal n_i^R and n_i^L, let $\sum_i D_i^{old} = \sum_i D_i$
2: Fix Q_T^R and Q_D^R, alternately perturb Q_T^L and Q_D^L until there is no gain in the objective (16)
3: Fix the Q_T^L and Q_D^L computed in step 2, alternately perturb Q_T^R and Q_D^R until there is no gain in the objective (16), mark the newly computed objective as $\sum_i D_i^{new}$
4: **if** $\sum_i D_i^{new} \leq \sum_i D_i^{old}$ **then**
5: Exit (we have converged)
6: **else**
7: $\sum_i D_i^{old} = \sum_i D_i^{new}$
8: Return to Line 2
9: **end if**

VII. Experimentation

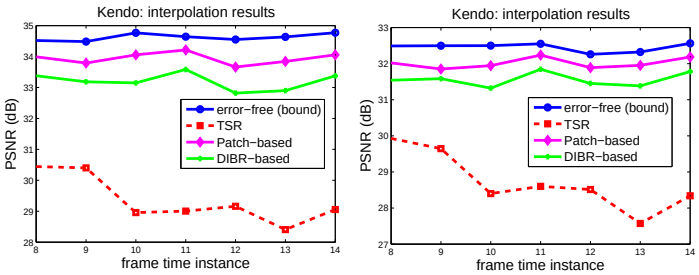
A. Experimental Setup

We evaluate the performance of our system, denoted as `Patch-based`, via extensive experiments. We used the 30fps MPEG free viewpoint test sequences `Kendo` and `Pantomime` from Nagoya University, where the texture and depth signals were encoded using H.264 JM18.0. The spatial resolution of `Kendo` and `Pantomime` is 512×384 and 640×480, respectively. The *Maximum transmission unit* (MTU) in the transmission network was set to 1500 bytes. Each GOP had 30 frames and was divided into three sub-groups. The initial video buffering time was set to 0.4s.

B. Lost Frame Recovery

We compare our frame recovery scheme to two competing schemes: `DIBR-based` and `TSR`. `DIBR-based` is the scheme proposed in [57], which recovers the lost texture and depth pixels first using DIBR, and then fills the remaining missing pixels using TSR. `TSR` recovers missing pixels using TSR only. For TSR in `DIBR-based`, `TSR`, and `Patch-based`, the block size was set to be 4×4, and the search was performed in 1/2-pixel accuracy. `error-free (bound)` is the synthesized intermediate view quality when both the left and right views are correctly delivered.

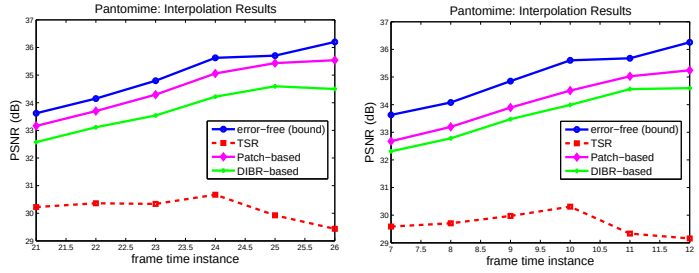
The recovery performance of these schemes on the content `Kendo` is shown in Fig. 14. View 1 and view 3 were the left and right views respectively during the experiment, and view 2 was used as the synthesized intermediate view. The x-axis denotes the frame number (index), and the y-axis measures the quality of the synthesized middle view. In Fig. 14(a), uncompressed video was used, and the video sequences used in Fig. 14(b) were encoded with QP=40.



(a) uncompressed video (b) video at QP=40

Fig. 14. Lost frame recovery results using different recovery methods for `Kendo`.

We observe that our proposed scheme `Patch-based` outperformed `DIBR-based` by up to 1.1dB. This is due to the more accurate sub-block motion estimation method and patch-level candidate selection method. Further, `Patch-based` outperformed `TSR` by up to 4.3dB. Comparing Fig 14(a) and (b), we see that larger QP leads to worse synthesized view quality as expected, but the performance trend remains consistent.



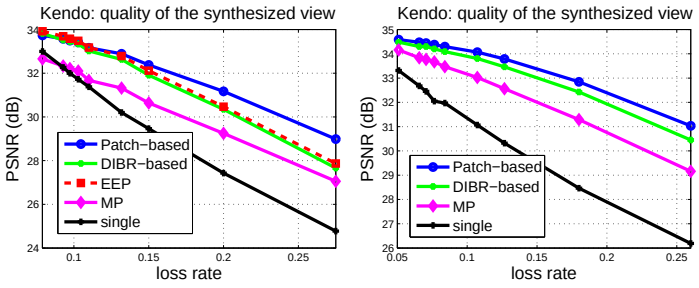
(a) uncompressed video (b) video at QP=23

Fig. 15. Lost frame recovery results using different recovery methods for `Pantomime`.

We also conducted comparison experiments for `Pantomime`. These results are shown in Fig. 15(a) and (b). In Fig. 15(a), uncompressed video was used, and the video sequences used in Fig. 15(b) were encoded with QP set to 23. We observe similar performance as for `Kendo`, where here `Patch-based` outperforms `DIBR-based` by up to 1.04dB.

C. Video Streaming

We conducted streaming experiments involving six competing schemes: `Patch-based`, `Patch-based SQP`, `single`, `DIBR-based`, `EEP`, and `MP`. `single` stands for the state-of-the-art single path / single description video transmission. Left- and right-view frames were sent in succession. At streaming time, the server will vary the amount of source packets by choosing the best source and channel coding rates via an exhaustive search. `Patch-based SQP` is a modified version of `Patch-based`, where the same QP is used for encoding of texture and depth maps on each path. `DIBR-based`, `EEP` and `MP` used two paths for video delivery, but `DIBR` used the `DIBR-based` recovery scheme to recover frames lost in the missing description, and `MP` used `TSR` as the recovery scheme. `EEP` used the same frame recovery scheme as `Patch-based`, but with equal error protection, which means the FEC packets were equally allocated to each subgroup. In `MP`, FEC packets were allocated to each subgroup optimally via an exhaustive search. `DIBR`, `EEP`, and `MP` used optimized QP for source coding. *Frame freeze* was used for the incorrectly decoded video frames, *i.e.* the user will play back the last correctly decoded frames if both descriptions are not correctly received.



(a) channel 1 (b) channel 2

Fig. 16. `Kendo`: Streaming results with different channel loss rates.

We first set the bandwidth for each path in the multi-path transmission scenario to be 400 kbps, and `single` had the combined bandwidth of the two paths. *i.e.*, 800 kbps. The streaming results are shown in Fig. 16. In Fig. 16(a), the GE parameters assumed were g=0.05, b=0.95, q=0.1 with p varied throughout the simulation to induce different loss rates. We observe that our proposed scheme outperformed all competing schemes, and the transmission schemes using multi-path outperformed `single`, although the latter is more efficient in terms of source coding. The reason for this outcome is that

if the communication channel enters a bad state, FEC cannot sufficiently protect lost data, and a lost frame can lead to a long error propagation. For multi-path transmission, the probability of both paths entering a bad state is quite low. Compared with `DIBR-based`, our proposed scheme `Patch-based` has better performance because of our advanced frame recovery scheme and source / channel rate optimization. `EEP`'s performance is worse compared with `Patch-based` because the FEC packets in `EEP` are not optimally allocated. To save space, we omit `EEP` in the following figures.

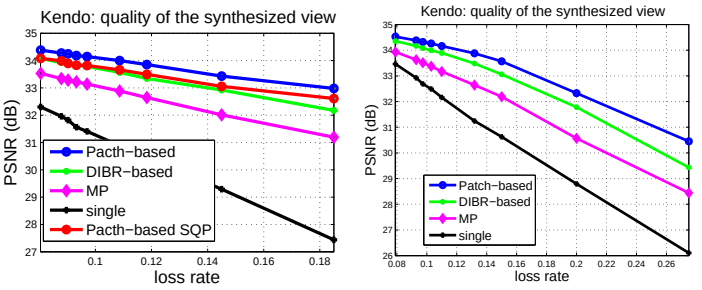
Then we changed the parameters of the GE model to be the following: g=0.02, b=0.98, q=0.05 with p varied to induce different loss rates. The results are shown in Fig. 16(b). Similar performance trend can be observed. `Patch-based` outperformed `single` by up to 4.2dB and 4.8dB in Fig. 16(a) and (b), respectively.

We also tested the cases when the two paths have asymmetric path loss rates, and the results are shown in Fig. 17(a). For the multi-path transmission, the GE parameters assumed for one path were g=0.05, b=0.95, q=0.1 p=0.0071, and GE parameters for the other path were g=0.02, b=0.98, q=0.05 with p varied throughout the simulation. Then we computed the expected loss characteristics for the two paths (expected bad state duration and expected loss rate) and selected comparable single-path GE parameters, g = 0.035, b = 0.965, and q = 0.1333 for `single`, so that the single path also has the same expected bad state duration and expected loss rate. In Fig. 17 (a), `Patch-based` outperformed `Patch-based SQP` by up to 0.4dB, which shows the advantages of using different QPs for texture and depth video encoding.

Next, we tested the case when the two paths have different transmission bandwidth, and the results are shown in Fig. 17(b), where all the paths were simulated using GE parameters g=0.05, b=0.95, q=0.1 with p varied to induce different loss rates. The bandwidth values of the two paths in the multi path scenario were set to 400 kbps and 500 kbps, respectively, and the bandwidth of *single* was 900 kbps. For both Fig. 17(a) and (b), similar performance could be observed as in Fig. 16. Relative to the single path / single description scheme, the performance gain of our system reaches up to 5.5dB and 4.4dB in Fig. 17(a) and (b), respectively.

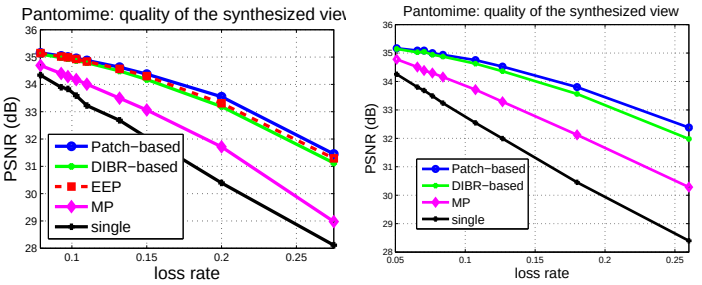
We also conducted the same experiments for `Pantomime`. In Fig. 18(a), the GE parameters assumed were g=0.05, b=0.95, q=0.1 with p varied to induce different loss rates. In Fig. 18 (b), the GE parameters assumed were g=0.02, b=0.98, q=0.05 with p varied. The bandwidth for each path in the multi-path scenario was 400 kbps and the bandwidth for `single` was 800 kbps. From the results, we can observe similar performance as for `Kendo`. The maximum performance gain relative to `single` is 3.4dB and 4.0dB in Fig. 18(a) and (b), respectively.

For `Pantomime`, we also tested the cases when the two paths have different loss conditions and different channel bandwidth values, as shown in Fig. 19. In Fig. 19(a),



(a) Asymmetric loss rates (b) Asymmetric bandwidth

Fig. 17. `Kendo`: Streaming results with asymmetric loss rates and bandwidth values.

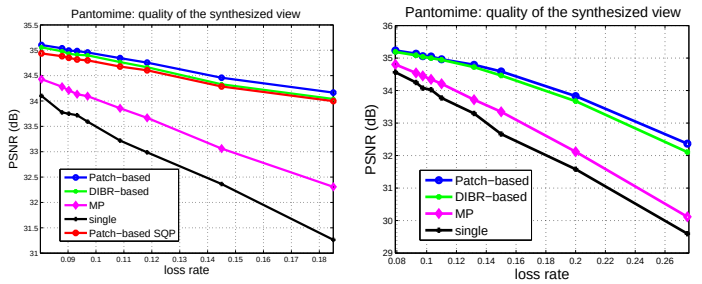


(a) Channel 1 (b) Channel 2

Fig. 18. `Pantomime`: Streaming results with different channel loss rates.

the GE parameters assumed for one of the paths were g=0.05, b=0.95, q=0.1 p=0.0071, and the GE parameters for the other path were g=0.02, b=0.98, q=0.05 with p varied to induce different loss rates. Then, the two paths' expected loss characteristics were used to construct a comparable single-path loss GE model for `single`, with parameters g = 0.035, b = 0.965, and q = 0.1333. We again varied p to control the overall loss rate in this case. From the simulation results, we could observe that our proposed scheme outperformed `single` by up to 2.9dB. In Fig. 19(a), `Patch-based` outperformed `Patch-based SQP` by up to 0.2dB.

When the two transmission paths have different bandwidth with the corresponding transmission channels simulated using the GE parameters g=0.05, b=0.95, q=0.1,



(a) Asymmetric loss rates (b) Asymmetric bandwidth

Fig. 19. `Pantomime`: Streaming results for asymmetric loss rates and bandwidth values.

and with p varied throughout the simulation, the results are shown in Fig. 19(b). The two paths were with 400 kbps and 500 kbps bandwidth respectively for the multi-path transmission schemes, and the single path transmission had 900 kbps bandwidth available. We observe that our proposed scheme can outperform `single` by up to 2.8dB.

VIII. Conclusion

Streaming of free viewpoint video in the texture-plus-depth format over wireless networks is a challenging problem due to the burstiness of the packet losses in wireless links and the stringent packet delivery deadlines of interactive video. In this paper, we propose to first encode the texture and depth signals of two camera-captured viewpoints into two independently decodeable descriptions for transmission over two disjoint wireless network paths. The source and channel coding rates for each description are optimized using an efficient branch-and-bound algorithm. In the event that a description is lost during transmission, missing frames in the lost description can be partially reconstructed using frames in the received description by exploiting the temporal and inter-view correlation of the transmitted viewpoints. Experimental results show that our proposed scheme can outperform a naïve single description / single path streaming solution by up to 5.5dB in PSNR.

References

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

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

[3] M. Tanimoto, M. P. Tehrani, T. Fujii, and T. Yendo, "Free-viewpoint TV," in *IEEE Signal Processing Magazine*, vol. 28, no.1, January 2011.

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

[5] M. Tanimoto, T. Fujii, and K. Suzuki, "Multi-view depth map of Rena and Akko & Kayo," ISO/IEC JTC1/SC29/WG11 MPEG Document M14888, Oct. 2007.

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

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

[8] D. Tian, P.-L. Lai, P. Lopez, and C. Gomila, "View synthesis techniques for 3D video," in *Applications of Digial Image Processing XXXII, Proceedings of the SPIE*, vol. 7443, (2009), February 2009, pp. 74 430T–74 430T–11.

[9] I.-H. Hou and P. R. Kumar, "Scheduling heterogeneous real-time traffic over fading wireless channels," in *IEEE INFOCOM*, San Diego, CA, March 2010.

[10] X. Xiu, G. Cheung, and J. Liang, "Delay-cognizant interactive multiview video with free viewpoint synthesis," in *IEEE Transactions on Multimedia*, vol. 14, no.4, August 2012, pp. 1109–1126.

[11] M. Podolsky, S. McCanne, and M. Vetterli, "Soft arq for layered streaming media," University of California, Berkeley, Tech. Rep. UCB/CSD-98-1024, November 1998.

[12] J. Apostolopoulos, "Error-resilient video compression via multiple state streams," *Proc. International Workshop on Very Low Bitrate Video Coding (VLBV'99)*, pp. 168–171, October 1999.

[13] B.-D. Choi, J.-W. Han, C.-S. Kim, and S.-J. Ko, "Motion-compensated frame interpolation using bilateral motion estimation and adaptive overlapped block motion compensation," *IEEE Trans. Cir. and Sys. for Video Technol.*, vol. 17, no. 4, pp. 407–416, Apr. 2007.

[14] P. Merkle, A. Smolic, K. Muller, and T. Wiegand, "Efficient prediction structures for multiview video coding," in *IEEE TCSVT*, vol. 17, no.11, November 2007, pp. 1461–1473.

[15] T. Wiegand, G. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," in *IEEE TCSVT*, vol. 13, no.7, July 2003, pp. 560–576.

[16] M. Flierl, A. Mavlankar, and B. Girod, "Motion and disparity compensated coding for multiview video," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no.11, November 2007, pp. 1474–1484.

[17] G. Shen, W.-S. Kim, S. Narang, A. Ortega, J. Lee, and H. Wey, "Edge-adaptive transforms for efficient depth map coding," in *IEEE Picture Coding Symposium*, Nagoya, Japan, December 2010.

[18] G. Cheung, W. s. Kim, A. Ortega, J. Ishida, and A. Kubota, "Depth map coding using graph based transform and transform domain sparsification," in *IEEE International Workshop on Multimedia Signal Processing*, Hangzhou, China, October 2011.

[19] J. Gautier, O. L. Meur, and C. Guillemot, "Depth map coding: exploiting the intrinsic properties of scenes and surface layout," in *Picture Coding Symposium 2012*, Krakow, Poland, May 2012.

[20] W. Hu, G. Cheung, X. Li, and O. Au, "Depth map compression using multi-resolution graph-based transform for depth-image-based rendering," in *Proc. of the IEEE International Conference on Image Processing*, Orlando, FL, September 2012.

[21] L. Golubchik, J. C. S. Lui, T. F. Tung, A. L. H. Chow, A. W.-J. Lee, G. Franceschinis, and C. Anglano, "Multi-path continuous media streaming: what are the benefits?" *Perform. Eval.*, vol. 49, no. 1/4, pp. 429–449, 2002.

[22] J. Chakareski, S. Han, and B. Girod, "Layered coding vs. multiple descriptions for video streaming over multiple paths," in *In Proc. of ACM Multimedia*, 2003, pp. 422–431.

[23] ——, "Layered coding vs. multiple descriptions for video streaming over multiple paths," in *Multimedia Syst.*, vol. 10, no. 4, 2005, pp. 275–285.

[24] V. Singh, S. Ahsan, and J. Ott, "Mprtp: Multipath considerations for real-time media," in *Proc. of ACM Multimedia Systems*, 2013.

[25] J. Zhai, K. Yu, J. Li, and S. Li, "A low complexity motion compensated frame interpolation method," in *IEEE International Symposium on Circuits and Systems*, Kobe, Japan, May 2005.

[26] P. Xia, S.-H. Chan, and X. Jin, "Optimal bandwidth assignment for multiple-description-coded video," *Multimedia, IEEE Transactions on*, vol. 13, no. 2, pp. 366–375, April 2011.

[27] S. Wenger, "Video redundancy coding in h.263+," in *Proceedings of AVSPN 97*, Aberdeen, U. K., 1997.

[28] S. Wenger, G. Knorr, J. Ott, and F. Kossentini, "Error resilience support in h.263+," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 8, no. 7, pp. 867–877, Nov 1998.

[29] C. Zhu and M. Liu, "Multiple description video coding based on hierarchical b pictures," *IEEE Trans. Circuits Syst. Video Techn.*, vol. 19, no. 4, pp. 511–521, 2009.

[30] J. Kim, Y. g. Kim, H. Song, T. y. Kuo, Y. Chung, and J. Kuo, "TCP-friendly internet video streaming employing variable frame-rate encoding and interpolation," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 10, no. 7, 2000, pp. 1164–1177.

[31] C. Wang, L. Zhang, Y. He, and Y.-P. Tan, "Frame rate up-conversion using trilateral filtering," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 20, no.6, June 2010, pp. 886–893.

[32] J. D. Areia, C. Brites, O. Pereira, and J. Ascenso, "Wynerziv stereo video coding using a side information fusion approach," in *in: IEEE International Workshop on Multimedia Signal Processing, Chania*, 2007, pp. 453–456.

[33] M. Ouaret, F. Dufaux, and T. Ebrahimi, "Multiview distributed video coding with encoder driven fusion," in *in Proceedings of the European Conference on Signal Processing (EUSIPCO 07*, 2007.

[34] G. Petrazzuoli, M. Cagnazzo, and B. Pesquet-Popescu, "Novel solutions for side information generation and fusion in multiview dvc," *EURASIP Journal on Advances in Signal Processing*, vol. 2013, no. 1, p. 154, 2013.

[35] B. Macchiavello, C. Dorea, M. Hung, G. Cheung, and W. t. Tan, "Reference frame selection for loss-resilient depth map coding in multiview video conferencing," in *IS&T/SPIE Visual Information Processing and Communication Conference*, Burlingame, CA, January 2012.
[36] G. Cheung, W.-T. Tan, and C. Chan, "Reference frame optimization for multiple-path video streaming with complexity scaling," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no.6, June 2007, pp. 649–662.
[37] B. Macchiavello, C. Dorea, M. Hung, G. Cheung, and W. t. Tan, "Reference frame selection for loss-resilient texture & depth map coding in multiview video conferencing," in *IEEE International Conference on Image Processing*, Orlando, FL, September 2012.
[38] ——, "Loss-resilient texture & depth map coding in multiview video conferencing," in *(accepted to) IEEE Transactions on Multimedia*, November 2013.
[39] J. Sørensen, J. Østergaard, P. Popovski, and J. Chakareski, "Multiple description coding with feedback based network compression," in *Proc. Globecom*. Miami, FL, USA: IEEE, Dec. 2010.
[40] G. Cheung, P. Sharma, and S. Lee, "Smart media striping over multiple burst-loss channels," in *IEEE Transactions on Selected Topics in Signal Processing*, vol. 1, no.2, August 2007, pp. 319–333.
[41] Y. Ding, Y. Yang, and L. Xiao, "Multi-path routing and rate allocation for multi-source video on-demand streaming in wireless mesh networks," in *IEEE INFOCOM*, Shanghai, China, April 2011.
[42] E. N. Gilbert, "Capacity of burst-noise channel," *Bell System Technical Journal*, vol. 39, pp. 1253–1265, 1963.
[43] 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," in *IEEE International Conference on Multimedia and Expo*, Toronto, Canada, July 2006.
[44] J. Chakareski, "Transmission policy selection for multi-view content delivery over bandwidth constrained channels," *Image Processing, IEEE Transactions on*, vol. 23, no. 2, pp. 931–942, Feb 2014.
[45] Y.-S. Kang, C. Lee, and Y.-S. Ho, "An efficient rectification algorithm for multi-view images in parallel camera array," in *3DTV-Conference*, IStanbul, Turkey, May 2008.
[46] I. Daribo and B. Pesquet-Popescu, "Depth-aided image inpainting for novel view synthesis," in *IEEE Multimedia Signal Processing Workshop*, Saint-Malo, France, October 2010.
[47] I. Ahn and C. Kim, "Depth-based disocclusion filling for virtual view synthesis," in *IEEE International Conference on Multimedia and Expo*, Melbourne, Australia, July 2012.
[48] S. Reel, G. Cheung, P. Wong, and L. Dooley, "Joint texture-depth pixel inpainting of disocclusion holes in virtual view synthesis," in *APSIPA ASC*, Kaohsiung, Taiwan, October 2013.
[49] G. Cheung, V. Velisavljevic, and A. Ortega, "On dependent bit allocation for multiview image coding with depth-image-based rendering," in *IEEE Transactions on Image Processing*, vol. 20, no.11, November 2011, pp. 3179–3194.
[50] D. Sun, S. Roth, and M. J. Black, "Secrets of optical flow estimation and their principles," in *IEEE CVPR*, San Francisco, CA, June 2010.
[51] I. Daribo, D. Florencio, and G. Cheung, "Arbitrarily shaped sub-block motion prediction in texture map compression using depth information," in *Picture Coding Symposium 2012*, Krakow, Poland, May 2012.
[52] J. Sun, Z. Xu, and H.-Y. Shum, "Image super-resolution using gradient profile prior," in *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008), 24-26 June 2008, Anchorage, Alaska, USA*. IEEE Computer Society, 2008.
[53] H.-Y. Shum, S.-C. Chan, and S. B. Kang, *Image-Based Rendering*. Springer, 2007.
[54] D. Min, J. Lu, and M. Do, "Depth video enhancement based on weighted mode filtering," *Image Processing, IEEE Transactions on*, vol. 21, no. 3, pp. 1176–1190, 2012.
[55] C.Tomasi and R.Marnduchi, "Bilateral filtering for gray and color images," in *Proceedings of the Sixth International Conference on Computer Vision*, Washington, DC, USA, 1998.
[56] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Kluwer Academic Publishers, 1992.
[57] Z. Liu, G. Cheung, J. Chakareski, and Y. Ji, "Multiple description coding of free viewpoint video for multi-path network streaming," in *2012 IEEE Global Communication Conference*, Anaheim, USA, Decemeber 2012.

Zhi Liu (S'11) received B.E., in computer science and technology from University of Science and Technology of China, China in 2009. Currently, he is a Ph.D candidate and JSPS research fellow in the National Institute of Informatics (NII) and the Department of Informatics, School of Multidisciplinary Science, The Graduate University for Advanced Studies (Sokendai). His research interest includes multiview video streaming, the wireless networks. He is a student member of IEEE, IEICE and IPSJ.

Gene Cheung (M'00—SM'07) received the B.S. degree in electrical engineering from Cornell University 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 in Hewlett-Packard Laboratories Japan, Tokyo, from 2000 till 2009. He is now an associate professor in National Institute of Informatics in Tokyo, Japan.

His research interests include image & video representation, immersive visual communication and graph signal processing. He has published over 130 international conference and journal publications. He has served as associate editor for IEEE Transactions on Multimedia from 2007 to 2011 and currently serves as associate editor for DSP Applications Column in IEEE Signal Processing Magazine and APSIPA journal on signal & information processing, and as area editor for EURASIP Signal Processing: Image Communication. He currently serves as member of the Multimedia Signal Processing Technical Committee (MMSP-TC) in IEEE Signal Processing Society (2012-2014). He has also served as area chair in IEEE International Conference on Image Processing (ICIP) 2010, 2012-2013, technical program co-chair of International Packet Video Workshop (PV) 2010, track co-chair for Multimedia Signal Processing track in IEEE International Conference on Multimedia and Expo (ICME) 2011, symposium co-chair for CSSMA Symposium in IEEE GLOBECOM 2012, and area chair for ICME 2013. He was invited as plenary speaker for IEEE International Workshop on Multimedia Signal Processing (MMSP) 2013 on the topic "3D visual communication: media representation, transport and rendering". He is a co-author of best student paper award in IEEE Workshop on Streaming and Media Communications 2011 (in conjunction with ICME 2011), best paper finalists in ICME 2011 and ICIP 2011, best paper runner-up award in ICME 2012, and best student paper award in ICIP 2013.

Jacob 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 Ecole Polytechnique Federale 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. He eagerly pursues ultrasound applications in telemedicine, remote sensing, and biomedicine, and is passionate about bridging science and technology via entrepreneurial activity.

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 Chancellor's 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 2003. 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 a special session on 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 Journal's 2013 special issue on P2P Cloud Systems. He is an Advisory Board member of Mainframe2, an innovative cloud computing start-up with a bright future. For more information, please visit http://www.jakov.org.

Yusheng Ji (M'94) received the B.E., M.E., and D.E. degrees in electrical engineering from the University of Tokyo, Tokyo, Japan, in 1984, 1986, and 1989, respectively. She joined the National Center for Science Information Systems, Japan in 1990. Currently, she is a Professor at the National Institute of Informatics, Japan, and the Graduate University for Advanced Studies, Sokendai, Japan. Her research interests include network architecture, traffic control, and performance analysis for quality of service provisioning in wired and wireless communication networks.

Dr. Ji is also a member of IEICE, IPSJ and ACM. She has held various positions, such as Board member of Trustees of IEICE, Steering Committee Member of Quality Aware Internet (QAI) SIG and Internet Architecture (IA) SIG, and Internet and Operation Technologies (IOT) SIG of IPSJ, Editor of IEEE Transactions on Vehicular Technology, Associate Editor of IEICE Transactions and IPSJ Journal, Guest Editor-in-Chief, Guest Editor and Guest Associate Editor of Special Sections of IEICE Transactions, and IPSJ Journal. She has also served as a TPC member of many conferences, including IEEE ICC, GLOBECOM, INFOCOM, VTC, WCNC, and PIMRC. She is/was the Wireless Networking Symposium Co-Chair of IEEE GLOBECOM 2012 and Optical Networks and Systems Symposium Co-Chair of IEEE GLOBECOM 2014.