# Multiple Description Coding of Free Viewpoint Video for Multi-Path Network Streaming

Zhi Liu[†], Gene Cheung[†], Jacob Chakareski[⋆], and Yusheng Ji[†]
[†]National Institute of Informatics, The Graduate University for Advanced Studies, Tokyo, Japan 101-8430
[⋆]École Polytechnique Fédérale de Lausanne, 1015 Lausanne, Switzerland
Email: {liuzhi, cheung, kei}@nii.ac.jp, jakov.cakareski@epfl.ch

***Abstract*—By transmitting texture and depth videos from two adjacent captured viewpoints, a client can synthesize via depth-image-based rendering (DIBR) any intermediate virtual view of the scene, determined by the dynamic movement of the client's head. In so doing, depth perception of the 3D scene will be created through motion parallax. Due to the stringent playback deadline of interactive free viewpoint video, burst packet losses in the texture and depth video streams caused by transmission over unreliable channels are difficult to overcome and can severely degrade the synthesized view quality at the client. We propose a multiple description coding (MDC) of free viewpoint video in texture-plus-depth format that will be transmitted on two disjoint network paths. Specifically, we encode even frames of the left view and odd frames of the right view separately as one description and transmit it on path one. Similarly, we encode odd frames of the left view and even frames of the right view as the second description and transmit it on path two. Appropriate quantization parameters (QP) are selected for each description, such that its data rate matches optimally the available transmission bandwidth on each of the two paths. If the receiver receives one description but not the other due to burst loss on one of the paths, it can still partially reconstruct the missing frames in the loss-corrupted description using a computationally efficient DIBR-based recovery scheme that we design. Extensive experimental results show that our MDC streaming system can outperform the traditional single-path single-description transmission scheme by up to 7dB in Peak Signal-to-Noise Ratio (PSNR) of the synthesized intermediate view at the receiving client.**

## I. Introduction

The popularity of stereoscopic video, where two texture images captured from slightly shifted cameras are shown separately to each of the viewer's eyes in order to induce depth perception of the 3D scene through binocular vision, is indisputable. However, it is known that *motion parallax* [1], where the viewer's head movement triggers a corresponding shift of viewing angle of the observed scene, represents an even stronger stimulus of depth perception. 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 an undesirable visual effect where physical objects in 3D scene appear as unnatural flat layers.

One technology to enable motion parallax is *free viewpoint video* [2]. At the sender, a large 1D array of closely spaced cameras synchronously capture 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* [3]—that are closest to the viewer's changing viewing perspective of the scene, as governed by his head movement that is dynamically tracked over time [4]. The viewer can then synthesize any intermediate view that corresponds to his present viewpoint of the scene, by using the texture and depth maps of the two captured views as anchors, via a *depth-image-based rendering* (DIBR) technique like 3D warping [5]. This results in an enhanced depth perception of the 3D scene via the aforementioned motion parallax.

If the communication path between the sender and receiver is burst-loss prone, which frequently occurs in wired networks (due to network congestion and packet queue overflow) and in wireless links (due to slow channel fading), then the resulting packet losses of texture and depth video 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. Therefore, packet loss recovery strategies based on automatic retransmission request (ARQ), which exhibit round-trip-time (RTT) delays, are not applicable.

To address this shortcoming, we propose a novel *multiple description coding* (MDC) scheme for free viewpoint video in texture-plus-depth format that is transmitted on two disjoint network paths. Specifically, we construct description $D_1$ to be 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 separately encoded odd frames of texture and depth maps of the left view and even frames of texture and depth maps of the right view comprise the second description $D_2$. Each description is transmitted on a disjoint network path, and appropriate quantization parameters (QP) are selected for each description in order to optimally match its data rate to the available transmission bandwidth on the path. If the receiver receives $D_1$, but not $D_2$, then it can reconstruct the missing frames in the second descrip-

[1]This work is supported in part by JSPS Grant-in-Aid for Scientific Research A (23240011), the work of J. Chakareski has been supported by the Swiss National Science Foundation under Ambizione grant PZ00P2-126416.

[1]A depth image is a pixel accurate measure of the distance between physical objects in the 3D scene and the capturing camera. It can be captured directly via a time-of-flight camera, or estimated using neighboring texture images using stereo-matching algorithms.

tion using a computationally efficient DIBR-based recovery scheme that we design. In particular, candidate pixels in a missing frame are first generated using DIBR and temporal super-resolution, separately. Then, a candidate-pixel selection procedure is employed to reconstruct the final frame presented to the client. Through experiments, we demonstrate that our MDC-based streaming system outperforms the conventional single-path single-description transmission scheme by up to 7dB in Peak Signal-to-Noise Ratio (PSNR) of the synthesized intermediate view at the receiving client.

The rest of the paper is organized as follows. We first discuss related work in Section II. Then, we describe our free viewpoint video streaming system in Section III. We present our MDC coding scheme and DIBR-based frame recovery procedure in Section IV. We discuss our data transport optimization in Section V. We experimentally investigate the performance of our video streaming system in Section VI. Finally, concluding remarks are provided in Section VII.

## II. Related Work

Multi-view video coding (MVC) [6] is an extension of the single-view video encoding standard, where multiple texture maps from closely spaced capturing cameras are encoded into one bitstream. Texture-plus-depth format of free viewpoint video [3] is a further extension, where by encoding texture and depth maps from multiple viewpoints, a user can now choose from an almost continuum of intermediate viewpoints between encoded views for display. Thereby, the 3D visual experience will be additionally enhanced.

While encoding the texture-plus-depth video format is already an intensively studied subject [3, 7], error resilient transmission of free viewpoint video is an emerging area. The few related studies thus far include the following. [8] proposed a new frame type called *unified distributed source coding frame* in order to enable discrete view-switching and to stop error propagation due to unrecoverable packet loss simultaneously. [9] considered the use of reference frame selection (RFS) in depth video coding to mitigate the adverse effects of synthesized view distortion due to packet losses, at the cost of additional coding overhead. In the present paper, we instead propose an MDC approach to loss-resilient transmission of free viewpoint video that is similar in spirit to the seminal work on multi-path transmission of MDC of single view video [10]. However, in [10] lost frame recovery is carried out via traditional temporal super-resolution methods based on motion search [11]. On the other hand, our loss frame recovery method employs DIBR to warp the received view to the view perspective of the lost view, to serve as its main estimate. [**?**] proposed a scalable MDC for 3D video that also separately encodes even and odd frames, but our work is a notable improvement due to our proposed description recovery mechanism in the event of burst losses.

## III. Multipath Free Viewpoint Video System

We first describe the proposed multi-path streaming system for free viewpoint video that is illustrated in Fig. 1. We assume that the server has available two disjoint network

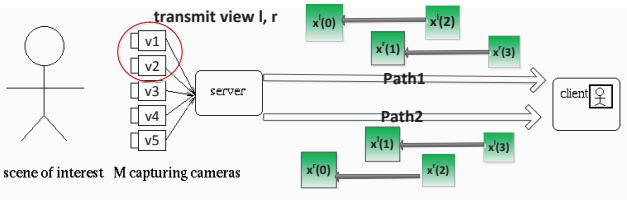



Fig. 1. Overview of the multi-path free-viewpoint video transmission system. Multiple descriptions are constructed for transmission over two disjoint paths.

transmission paths to a client. For instance, a wireless client can have two network interfaces such as 3G cellular and 802.11 Wi-Fi that connect to two orthogonal communication networks [12]. Similarly, a multi-homed client can connect to the Internet via two different ISPs that will typically route data to the same destination via different network paths [13]. In both of these cases, a streaming server can transmit the free viewpoint data to the client simultaneously over the two independent networks. The two disjoint network paths that will be employed to this end will in general be characterized by different transmission bandwidth values and packet loss statistics. Since the paths are disjoint, packet loss events on one link are independent from loss events on the other.

We consider that the free viewpoint content is encoded in the now popular *texture-plus-depth* format [3], where texture and depth maps of two appropriately chosen viewpoints (called left and right views in the sequel) are encoded and transmitted from the server to the client. In particular, the free viewpoint video is encoded into two descriptions $D_1$ and $D_2$. If the receiver receives one of the two transmitted descriptions, then it can reconstruct the content at coarse quality. If it receives both descriptions, then it can reconstruct the content at high quality. MDC is different from layered encoding, where an enhancement layer is correctly decoded only if the base layer is already correctly decoded. At the client, a novel intermediate virtual view can be synthesized using texture and depth maps of the two encoded views via depth-image-based rendering (DIBR) techniques like 3D warping [5].

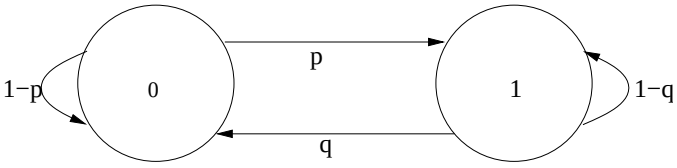



Fig. 2. Gilbert-Elliott loss model: The channel transitions between its two states (good - 0 and bad - 1) with probabilities $p$ and $q$. The packet loss probabilities in good and bad states are $g$ and $b$, respectively.

We assume that each network paths exhibit packet losses characterized by a Gilbert-Elliott (GE) model. 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. Given that a packet is transmitted during a bad or good state, the packet will be lost with probabilities $g$ and $b$, respectively.

## IV. Source Coding & Data Recovery

### A. Multiple Description Construction

We first describe how we construct two descriptions from the captured texture and depth maps by the 1D array of closely spaced cameras. For the left view, we perform standard motion-compensated (MC) predictive video coding, such as H.264 [14], respectively on the even and odd frames of the texture video of the left view, $x^l(0), x^l(2), \ldots$, and $x^l(1), x^l(3), \ldots$, thereby creating two streams $X^l_e$ and $X^l_o$. Similarly, we encode the even and odd frames of the depth video data associated with the left view, as well as the even and odd frames of the texture and depth video of the right view, into the corresponding streams $Z^l_e$, $Z^l_o$, $X^r_e$, $X^r_o$, $Z^r_e$, and $Z^r_o$. Note that since the temporal distance between consecutive coded frames is two (rather than one frame as in conventional video coding), our MDC results in a slightly larger coding rate. Note also that because a depth frame only comprises smooth surfaces with sharp edges, its required encoding bitrate represents only 10% to 15% of the texture video bitrate, for the same quantization parameter (QP).

Given the encoded streams, we construct two descriptions of the free viewpoint content, as follows. First, we map the streams $X^l_e$, $Z^l_e$, $X^r_o$, and $Z^r_o$ onto the first description $D_1$; i.e., the first description is composed of the even frames of the left view and the odd frames of the right view. Then, we map the remaining streams $X^l_o$, $Z^l_o$, $X^r_e$, and $Z^r_o$ onto the second description $D_2$; i.e., the second description is composed of the odd frames of the left view and the even frames of the right view. The first and second description $D_1$ and $D_2$ are transmitted to the client via path one and two, respectively.

### B. DIBR-based Frame Recovery

When only one description is received, e.g., $D_1$, we can partially recover the lost frames in the second description $D_2$, as follows. For each missing frame $x^r(t)$, frame $t$ of the right view texture video, we first synthesize via DIBR the missing frame $\bar{x}^r(t)$ using as anchors the corresponding texture and depth maps of the left view at the same time instant, $x^l(t)$ and $z^l(t)$. Specifically, each pixel $x^l(t, m, n)$ of row $m$ and column $n$ in the left texture map can potentially be mapped to a corresponding pixel $x^r(t, m, n - z^l(t, m, n) * \gamma)$ in the right texture map, where the horizontal shift is determined by the disparity[2] value $z^l(t, m, n)$ and a shift parameter $\gamma$, the latter of which is a function of the physical distance between the two capturing cameras.

There are two shortcomings of this simple pixel-to-pixel translational mapping. First, multiple texture pixels in $x^l(t)$ can map to the same pixel location in $\bar{x}^r(t)$. In that case, one typically chooses the pixel with the largest corresponding disparity value [15], which is the one with the smallest depth value and closest to the capturing camera. Hence, this pixel will occlude the further-away pixels. The second problem is that there might not be any pixel $x^l(t)$ that can map to a given pixel location in $x^r(t)$. Physically, this means that the pixel location in $x^r(t)$ was occluded in $x^l(t)$ and so no pixel correspondence can be established, a phenomenon known as *disocclusion*. To solve this problem, we propose the following computationally efficient procedure.

**Temporal super-resolution**: The basic idea is to to fill in a missing pixel $\bar{x}^r(t)$ using *temporal super-resolution* (TSR) techniques such as [11], where a missing frame $x^r(t)$ is interpolated using information from its two temporal neighbors, $x^r(t-1)$ and $x^r(t+1)$. While complex methods like optical flow [16] can provide excellent TSR performance, instead, we design a more efficient method that employs block search [11]. In a nutshell, given a $(2n+1) \times (2n+1)$ target block in frame $x^r(t+1)$, we search for its best match in frame $x^r(t-1)$, offset by the *motion vector* $(v, h)$. This is typically done by computing the sum of the absolute differences (SAD) between the pixels of the target block and its prospective match in frame $x^r(t-1)$. The candidate match with the smallest overall SAD is then selected as the winner. Formally, this block matching procedure can be written as

$$\min_{(v,h)} \sum_{(i,j)\in\mathcal{B}} |x^r(t-1, m+v+i, n+h+j) - x^r(t+1, m+i, n+j)|, \quad (1)$$

where $\mathcal{B} = \{(-n, -n), \ldots, (n, n)\}$ defines the support of the candidate block. Note that the search is performed in 1/4-pixel precision that is also employed by the H.264 standard [14] in motion estimation.

Given a motion vector $(v, h)$, we can copy a target block in $x^r(t+1)$ to the location in $\hat{x}^r(t)$ that is offset by the *half-vector* $(v/2, h/2)$. By performing this block-based search around a local neighborhood of disoccluded pixels in synthesized $\bar{x}^r(t)$, we can now have candidate pixels in $\hat{x}^r(t)$ to fill in the holes. The same process is used for forward block search: given a target block in $x^r(t-1)$, we find the best-matched block in $x^r(t+1)$ with vector $(v, h)$, and compute candidate pixels in $\tilde{x}^r(t)$ with the half-vector $(v/2, h/2)$.

To choose between candidate pixels in $\hat{x}^r(t)$ and $\tilde{x}^r(t)$, for hole-filing in $\bar{x}^r(t)$, we start from the boundary of the disoccluded region in $\bar{x}^r(t)$, and find the boundary pixels in $\hat{x}^r(t)$ and $\tilde{x}^r(t)$ that are the most similar to neighboring filled-in pixels in $\bar{x}^r(t)$. We iteratively select candidate pixels until all the disoccluded pixels are filled. The underlying assumption that we employ here is that missing pixels are most likely to be similar to surrounding pixels.

The same procedure is used to recover odd texture frames of the left view in lost description $D_2$. To recover odd depth frames of left view and even depth frames of the right view, synthesized frames $\bar{z}^l(t-1)$ and $\bar{z}^r(t)$ can be computed identically. For candidate pixels in disoccluded regions in $\bar{z}^l(t-1)$ and $\bar{z}^r(t)$, instead of performing motion search again, we reuse the already computed half-vectors in the corresponding texture maps to compute $\hat{z}^l(t-1)$, $\tilde{z}^l(t-1)$, $\bar{z}^r(t)$, and $\bar{z}^r(t)$ to reduce the computational complexity.

[2]Because there is typically a one-to-one correspondence between depth and disparity, we will assume without loss of generality that disparity values (rather than depth values) are actually encoded.

Note that given that we use TSR only to fill-in missing pixels in the DIBR-based synthesized view, the underlying assumption of our DIBR-based frame recovery procedure is that available pixels in synthesized view $\bar{x}^r(t)$ are more accurate than TSR-generated $\hat{x}^r(t)$ and $\tilde{x}^r(t)$ candidates. It turns out that this is indeed true in practice because: i) the same object surface observed from slightly different viewpoints very often reflects similar amounts of light (called *Lambertian* in the literature), resulting in very similar luminance values; and ii) motion search between target frame $x^r(t+1)$ and frame $x^r(t-1)$ is very hard to be pixel-accurate when the search block is large. On the other hand, if the search block is too small, there are typically too many good matches and the true pixel motion cannot be easily identified.

## V. Multi-Path Transmission

Let $R_i$, $i \in \{1,2\}$, denote the number of source packets associated with the two descriptions $D_1$ and $D_2$ for one Group of Pictures (GOP). These quantities depend on the QPs $Q_1$ and $Q_2$ employed to encode the two descriptions, respectively. Let $B_1$ and $B_2$ denote the available transmission bandwidth on the two network paths. That means that $B_i - R_i(Q_i)$, for $i = 1, 2$, FEC packets (such as the Reed-Solomon codes) will be used for loss protection of source packets on path $i$, where full recovery is possible if any $R_i(Q_i)$ of $B_i$ packets are received correctly. Our goal here is to select the source rate associated with each description such that the expected decoded video quality is maximized.

### A. Preliminaries

We first formally define the mathematical quantities, as done in [8], which are useful for a GE packet loss model. Let $P(i)$ be the probability of having *at least* $i$ consecutive transmissions in the good state in the GE model, given transmission starts in bad state. Furthermore, let $p(i)$ be the probability of having *exactly* $i$ good state transmissions between two bad state transmissions, given transmission starts in bad state. We write $P(i)$ and $p(i)$ as follows:

$$\begin{aligned} P(i) &= \begin{cases} 1 & \text{if } i = 0 \\ q(1-p)^{i-1} & \text{otherwise} \end{cases} \\ p(i) &= \begin{cases} 1-q & \text{if } i = 0 \\ q(1-p)^{i-1}p & \text{otherwise} \end{cases} \end{aligned} \tag{2}$$

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 transmission starts in bad state:

$$R(m,n) = \begin{cases} P(n) & \text{for } m = 0 \text{ and } n \geq 0 \\ \sum_{i=0}^{n-m} p(i) R(m-1, n-i-1) & \text{for } 1 \leq m \leq n \end{cases} \tag{3}$$

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 (3), with $Q(i)$ and $q(i)$ replacing $P(i)$ and $p(i)$ in (3), respectively.

### B. Source Coding Rate Optimization

We first write the probability $\alpha_i$ of correctly receiving description $i$ as a weighted sum of $\alpha_i^G$ and $\alpha_i^B$, which are the probabilities of correctly receiving $D_i$, given that packet transmission started during a good or bad state, respectively.

$$\alpha_i = \left(\frac{q}{p+q}\right)\alpha_i^G + \left(\frac{p}{p+q}\right)\alpha_i^B \tag{4}$$

Assuming transmission starts in the good state, $m$ of $B_i$ total packets can be transmitted in good state with probability $S(m, B_i)$. GOP can be successfully received if at least $R_i$ of $B_i$ packets are correctly delivered, and these $R_i$ packets can be a sum of $r_G$ and $R_i - r_G$ delivered packets in good and bad states respectively. We can hence write $\alpha_i^G$ as:

$$\alpha_i^G \approx \sum_{m=0}^{B_i} S(m, B_i) \sum_{r=R_i}^{B_i} \sum_{r_G=0}^{r} P_G(r_G, m) P_B(r - r_G, B - m) \tag{5}$$

where $P_G(x,y)$ and $P_B(x,y)$ are the probabilities of exactly $x$ delivered packets in $y$ tries in good and bad state respectively, which can be computed easily using binomial expansion and packet loss probability $g$ and $b$, respectively for good and bad channels. $\alpha_i^B$ can be derived similarly.

If we now assume QP $Q_i$ leads to a visual quality $d_i$ in PSNR for description $D_i$, then the source rate optimization for each description $D_i$ becomes:

$$\max_{Q_i} \ \alpha_i d_i \quad \text{s.t.} \quad R_i(Q_i) \leq B_i \tag{6}$$

Because $Q_i$ takes on a small finite set of values, (6) can be solved via exhaustive search with reasonable complexity.

## VI. Experimentation

### A. Simulation Setup

To evaluate the performance of our framework in a typical network loss environment, we carry out a number of experiments based on the following parameter setup. For source coding, we use H.264 JM18.0 as the encoder to encode the `Kendo` multiview video test sequence at 1024*768 spatial resolution. The original frame rate is 30Hz.

The maximum transmission unit (MTU) size is set to 1500 bytes. We examine two cases for the transmission bandwidth on the network paths. First, we consider that each path has the same transmission bandwidth of 480kbps. Then, we also examine the scenario where the two paths exhibit asymmetric transmission bandwidth values of 480 and 720 kbps. Varying degrees of measured packet loss from 8.5% to 15.7% were simulated on each path by varying the transition probability $q$ of the respective G-E models. The loss rates of the good and bad states $g$ and $b$ were respectively set to 5% and 80%, while the average sojourn time in a bad state is set to 10. Texture and depth videos associated with views 1 and 3 are sent to the client on the two paths. An event driven simulation is used and

each performance point in the network simulation experiments is the average result of over 100,000 experimental trials.

### B. Simulation Results

In Figs. 3 and 4, we respectively show the texture and depth maps affiliated with frame 8 of View 3. They are employed to reconstruct the texture map (the actual image shown to the viewer) of the same frame, but associated with View 1, using our DIBR-based interpolation technique from Section IV-B. The resulting interpolated frame 8 of View 1 is shown in Fig. 5. We counted the number of disoccluded pixels ("holes") in the interpolated frame in order to understand how often our interpolation technique cannot be applied in this case. They comprise 4% of the overall number of pixels in the frame.

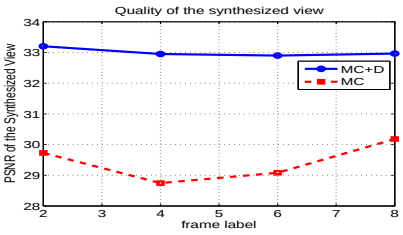
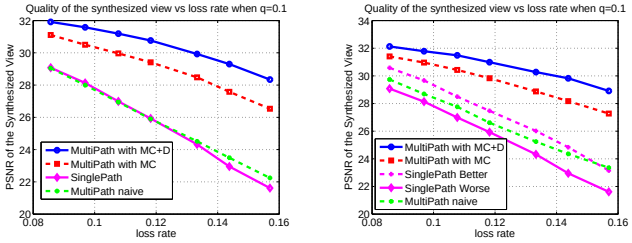



Fig. 6. Quality of the synthesized View 2

We first investigate the interpolation performance of our proposed interpolation method, denoted henceforth as `MC+D`, relative to the approach that only employs motion estimation and motion compensation for the same goal, denoted henceforth as `MC`. These results are shown in Fig. 6, where we interpolate the 2nd, 4th, 6th and 8th frame of View 2. First, we can observe that the gains over `MC` are substantial and range between 3 to 4.1dB. Furthermore, we can also observe that the performance of the proposed scheme is consistent regardless of the content's complexity in terms of temporal motion, since the synthesized view quality remains constant over different video frames.

Next, we compare the resulting video quality of synthesized View 2 at the receiver in the case of four different transmission systems. `MultiPath with MC+D` is a system that employs the multipath transmission scheme and the interpolation method proposed in the present paper. `MultiPath with MC` is a scheme that also employs the proposed multipath transmission approach, however, it only considers motion estimation and compensation for interpolation of lost frames. `SinglePath` is a baseline scheme that transmits all the data over a single network path between the server and the client. To make the comparison meaningful, the transmission bandwidth of the single path employed by `SinglePath` is equal to the composite bandwidth of the two transmission paths employed by the multi-path techniques. It should be mentioned that `SinglePath` exhibits the highest compression efficiency, since the content associated with a view is not encoded separately into odd and even frames. Finally, `MultiPath naive` is a scheme that employs multi-path transmission that does not interleave view data across paths, i.e., one view is sent on one path, exclusively. In each simulation run, we adapt the quantization parameters of the encoded I and P frames such that the data rate of the content fits the effective network bandwidth $r_i$ on each path. The reader should recall that FEC coding is applied in order to reduce the effective packet loss rate on each path, as described in Section V. Therefore, the overall network bandwidth $B_i$ on path $i$ is divided between $r_i$ and the remaining fraction $B_i - r_i$ that is employed for FEC packet loss resilience.

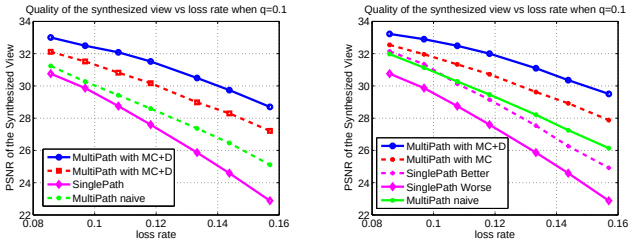



(a) Symmetric path loss (b) Asymmetric path loss

Fig. 7. Quality of synthesized View 2 for symmetric path bandwidth.

In Fig. 7(a), we show the resulting synthesized view quality for the four techniques under comparison, as a function of the average packet loss rate on each of the two network paths. We can see that `MultiPath with MC+D` outperforms the other schemes with a significant margin. In particular, gains of 1.8dB, 6.7dB, and 6.1dB are achieved over `MultiPath with MC`, `SinglePath`, and `MultiPath naive`, respectively. In the case of single-path transmission, if the communication channel enters a bad state, the lost data cannot be recovered by FEC decoding. However, for multipath transmission, the probability of both paths exhibiting a bad state is quite low. Therefore, the lost data on one path can be recovered using the received data on the other. Yet, the substantial gains of close to 2dB over `MultiPath with MC` illustrate the additional benefits of the proposed interpolation technique.

We also studied the case of asymmetric path loss, by introducing additional packet loss of 1.5% on one of the paths. The corresponding results are shown in Fig. 7(b), where the x-axis represents the loss rate of the network path with the higher loss rate. `SinglePath Worse` in Fig. 7(b) corresponds to the case of single-path transmission over the higher-loss path and `SinglePath Better` to the case of single-path transmission over the less lossy path.




(a) Symmetric path loss (b) Asymmetric path loss

Fig. 8. Quality of synthesized View 2 for asymmetric path bandwidth.

Next, we increased the bandwidth of one path to 720kbps, and therefore the combined bandwidth of the two paths is 1.2Mbps now. We then repeated the same experiments from

Fig. 3. Texture map of frame 8, View 3.

Fig. 4. Depth maps of frame 8, View 3.

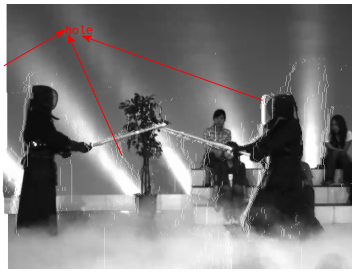


Fig. 5. Texture map of Frame 8 of View 1 interpolated using the corresponding depth and textures maps of View 3.

above on this asymmetric bandwidth scenario. These results are shown in Fig. 8. We observed similar gains in this case that range up to 1.8dB, 6.7dB, and 3.5dB relative to `MultiPath with MC`, `SinglePath`, and `Multipath naive`, respectively, as shown in Fig. 8(a). We also considered the possibility of asymmetric packet loss rate of 1.5% between the two paths, as in our previous experiments. These results are shown in Fig. 8(b). Since we obtained equivalent results, irrespective of which of the two paths exhibited the smaller loss rate, we included in Fig. 8(b) only one set of them, for the case when the smaller bandwidth path also exhibits a smaller loss rate. Note that similar gains of 1.7dB, 4.6dB, 6.7dB, and 3.4dB are observed in this scenario over `MultiPath with MC`, `SinglePath Better`, `SinglePath Worse`, and `Multipath naive`, compared to the corresponding results examined in Fig. 7(b) for the symmetric path bandwidth case. However, it should be pointed out that all transmission techniques improved their end-to-end performance in this scenario due to the higher overall bandwidth, relative to the case of symmetric path bandwidth studied in Fig. 7. In particular, it can be observed that the naive multi-path transmission technique `Multipath naive` apparently profited the most from the higher available bandwidth on one of the paths in the present scenario. Its performance has notably improved relative to the results shown in Fig. 7(b), as we can see from Fig. 8(b). The same is also true when the performance of `Multipath naive` is cross-compared between Fig. 8(a) and Fig. 7(a).

## VII. Conclusion

We have presented a system for multi-path transmission of free viewpoint video. In order to deal with burst loss in the network, we encode the content into multiple descriptions that are sent on two disjoint network paths. The description construction is such that it facilitates the generation of synthetic viewpoints based on an effective view interpolation technique that we design as part of our framework. Our DIBR view recovery scheme outperforms existing motion-compensation based techniques with a margin of close to 2dB in video quality, when they also employ multi-path transmission, and with an even more impressive gain of close to 7dB, when they employ single-path transmission.

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

[3] 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.

[4] 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.

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

[6] 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.

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

[8] Z. Liu, G. Cheung, and Y. Ji, "Unified distributed source coding frames for interactive multiview video streaming," in *(accepted to) IEEE ICC*, Ottawa, Canada, June 2011.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] A. Akella, J. Pang, B. Maggs, S. Seshan, and A. Shaikh, "A comparison of overlay routing and multihoming route control," in *Proc. SIGCOMM*. Portland, OR, USA: ACM, Aug./Sep. 2004, pp. 93–106.

[14] 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.

[15] 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.

[16] D. Sun, S. Roth, and M. J. Black, "Secrets of optical flow estimation and their principles," in *IEEE CVPR*, San Francisco, CA, June 2010.