MULTI-STREAM SWITCHING FOR INTERACTIVE VIRTUAL REALITY VIDEO STREAMING

Gene Cheung [#], *Zhi Liu* [*], *Zhiyou Ma* [$], *Jack Z. G. Tan* [$]

[#] National Institute of Informatics, [*] Waseda University, [$] Kandao Technology

ABSTRACT

Virtual reality (VR) video provides an immersive 360 viewing experience to a user wearing a head-mounted display: as the user rotates his head, correspondingly different fields-of-view (FoV) of the 360 video are rendered for observation. Transmitting the entire 360 video in high quality over bandwidth-constrained networks from server to client for real-time playback is challenging. In this paper we propose a multi-stream switching framework for VR video streaming: the server pre-encodes a set of VR video streams covering different view ranges that account for server-client round trip time (RTT) delay, and during streaming the server transmits and switches streams according to a user's detected head rotation angle. For a given RTT, we formulate an optimization to seek multiple VR streams of different view ranges and the head-angle-to-stream mapping function simultaneously, in order to minimize the expected distortion subject to bandwidth and storage constraints. We propose an alternating algorithm that, at each iteration, computes the optimal streams while keeping the mapping function fixed and vice versa. Experiments show that for the same bandwidth, our multi-stream switching scheme outperforms a non-switching single-stream approach by up to 2.9dB in PSNR.

Index Terms— Video streaming, virtual reality, video coding

1. INTRODUCTION

The advent of technologies for camera rigs, fisheye lenses and image-stitching algorithms [1, 2] means that 360 *virtual reality* (VR) video can now be readily generated. A user equipped with a head-mounted display (HMD) such as Oculus Rift[1] or HTC Vive[2] can enjoy an immersive 360 viewing experience: as the user rotates his head to the left or right, correspondingly different *fields-of-view* (FoV) of the 360 VR video are rendered for observation. See Fig. 1 for an illustration. It has been shown [3] that such *motion parallax* visual effect—changing FoVs according to user's head position and rotation angle—is the strongest cue for human's depth perception in a 3D scene, and VR video enables this effect for any head rotation angle from 0 to 360.

[1]https://www3.oculus.com/en-us/rift/

[2]https://www.vive.com/jp/

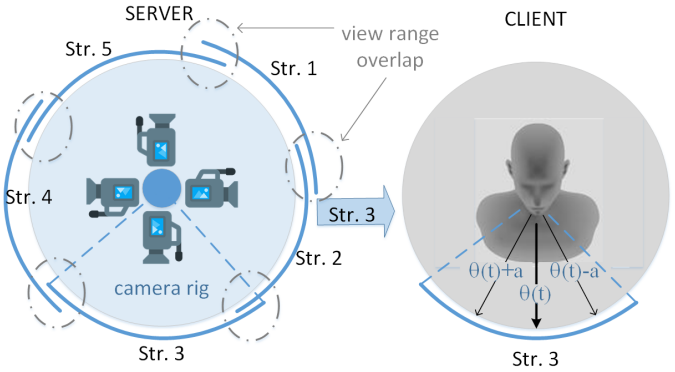


Fig. 1. Interactive VR video streaming system using 5 pre-encoded streams with overlapping view ranges. Corresponding to user's head rotation angle $\theta(t)$ and FoV $[\theta(t) - a, \theta(t) + a]$ at time t, stream 3 is selected and transmitted.

However, transmitting the entire 360 VR video in high quality over bandwidth-limited networks from a server to a client for real-time playback is challenging. Leveraging on previous works in *interactive multiview video streaming* (IMVS) [4, 5], we propose a multi-stream switching framework for 360 VR video streaming. The server pre-encodes a set of VR video streams, each covering a different view range of the original 360 video. During streaming, the server transmits and switches among the pre-encoded streams according to a user's detected head rotation angle.

By transmitting one video stream covering a limited view range at a time, the server can encode the stream at a higher quality than a single stream covering all 360 viewing angles for the same bandwidth constraint. However, to minimize the adverse effect of interaction delay in motion parallax—even in the face of non-negligible server-to-client *round trip time* (RTT) delay—each pre-encoded stream must cover a wide enough view range, so that a user's head with rotation angle starting in the view range center would not drift outside the view range in one RTT. This implies that the coded streams tend to overlap in view ranges, resulting in representation redundancies and high storage cost. Thus, *multi-stream switching can enable higher visual quality, at the expense of an increase in storage cost due to streams' view range overlaps.*

Thus, the technical challenge is, for a given RTT, to design multiple VR streams of different view ranges and the

head-angle-to-stream mapping function in order to minimize the expected distortion subject to bandwidth and storage constraints. We mathematically formalize this optimization and propose an alternating algorithm that, at each iteration, computes the optimal VR streams while keeping the mapping function fixed and vice versa. Experimental results show that for the same bandwidth constraint, our proposed multi-stream switching scheme outperforms a single-stream approach by up to 2.9dB in PSNR.

2. RELATED WORK

Using an array of cameras to capture a 3D scene synchronously from slightly shifted viewpoints, IMVS systems [4–8] study how the captured multi-view videos can be pre-encoded into multiple streams. A receiving user can periodically request switches to neighboring camera views, and the server in response switches video streams with minimum discruption to the user's viewing experience. To facilitate stream-switching, new frames like DSC frame [9] and merge frame (M-frame) [10] were proposed. Unlike IMVS [4–8], we optimize the division of 360 VR video into multiple streams covering different view ranges given a constant RTT. To the best of our knowledge, we are the first to study this problem for interactive VR video streaming formally.

There are recent studies on VR video streaming. Assuming that the 3D scene can be represented by a 3D mesh, [11, 12] proposed to first divide the mesh into 3D sub-meshes (tiles). During streaming, a user communicates the desired tiles to the server using MPEG-DASH-SRD [13], an extension of MPEG-DASH [14] to specify spatial relationships in media content. Unlike [11, 12], we assume the input to our optimization is a 360 VR video, not 3D mesh. Further, we take the effect of RTT on interaction delay into account explicitly during optimization (to be detailed in Section 4).

Assuming a camera rig with multiple cameras capturing a 360 view from different angles, [15] described a multiview video scheme that divides and codes captured camera views into two types: i) primary views at lower resolution that cover the entire 360 field-of-view, and ii) auxiliary views for the remaining camera views at high resolution. The two video types are coded using multilayer extensions of HEVC. The receiver then performs image stitching to compose a 360 VR view. Instead, we assume 360 VR video is composed at the sender, and the challenge is to design multiple video streams covering different view ranges for interactive streaming.

3. SYSTEM OVERVIEW

We overview the operations of our multi-stream switching framework for a given RTT. Denote by T the RTT between server and client. Denote by Δ the time interval between coded frames; $1/\Delta$ is the number of frames per second (fps). For simplicity, assume for now that all frames are intra-coded, so that streams can be switched at any frame. The server starts transmission of an initial video stream to the user at time $t = -T/2$, assuming the user begins at an initial head rotation angle $\theta(0)$. At time $t = 0$, the stream arrives at the client and playback begins. At time $t = \Delta$, the client transmits the first feedback $\theta(\Delta)$ of the user's head rotation angle to the server. This feedback $\theta(\Delta)$ arrives at the server at $t = T/2 + \Delta$, and the server decides the new stream to transmit corresponding to $\theta(\Delta)$ using a *mapping function* $f(\theta(\Delta))$. This new stream arrives at the client at time $t = T + \Delta$, exactly T seconds after feedback $\theta(\Delta)$ was generated. Hence, *the transmitted stream must accommodate the change in head rotation angle from* $\theta(\Delta)$ *to* $\theta(T + \Delta)$. See Fig. 2 for an illustration.

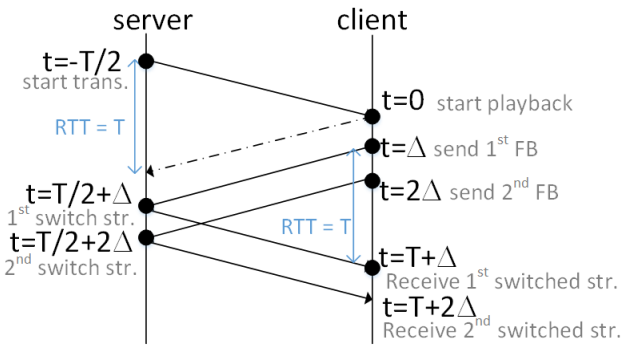


Fig. 2. Interaction between server and client where RTT is T and frame interval is Δ. A switched stream arrives T seconds after a feedback is sent.

Consider now the case when the VR streams are coded in Group-of-Pictures (GOP) of H frames each. This means that the frequency at which the server can switch streams is also every H frames. Compared to the previous case of intra-coded frames, each VR stream must now accommodate the change in head rotation angle in time interval $T + H\Delta$.

We next formulate the optimization problem to find the multiple VR streams and the mapping function $f(\,)$.

4. PROBLEM FORMULATION

4.1. View Interaction Model

We first define a *view interaction model* that models a typical view selection process during 360 VR video observation. Denote by $\theta[n]$ the *central view angle* at which an observer is watching straight ahead at discrete time n. For convenience, we define the duration of a discrete time interval to be Δ (time interval between frames), and RTT in discrete instants to be $T_s = T/\Delta$. We assume that $\theta[n] \in \{1, \ldots, K\}$ is also discrete, where $\theta[n]\, 2\pi/K$ is angle in radians between 0 and 2π. We assume a one-hop Markov view transition model, where the probability of an observer's angle $\theta[n+1] = j$ given $\theta[n] = i$ is $p_{i,j}$. Finally, we assume that the observer changes views only locally per instant, *i.e.*, $p_{i,j} = 0$ if $|i - j| > v_{\max}$.

At any instant n, the observer has a FoV of size $1 + 2a \ll K$ that defines the angular span a human observes at a time. Hence at time instant n, given central view angle $\theta[n]$, the

observer's FoV is $\mathbf{R}[n] = [\theta[n]-a, \theta[n]+a]$. It means that an observer will see visual distortion if the current video stream is not coded at high enough video quality in this range $\mathbf{R}[n]$.

4.2. Expected Distortion

We define the expected distortion an observer sees in a 360 VR video as he naturally rotates his head. We consider first the simple case when the GOP size is a single frame. First, we compute the *steady state probabilities* $\mathbf{q} \in \mathbb{R}^K$ assuming stationary view transition probabilities $p_{i,j}$ via the Perron-Frobenius Theorem[3]:

$$\mathbf{q}\,\mathbf{P} = \mathbf{q} \tag{1}$$

where $\mathbf{q}$ is the left eigenvector (row vector) corresponding to the eigenvalue 1 for matrix $\mathbf{P}$.

Denote by $\mathbf{1}_k$ the canonical row vector of length K with the only non-zero entry at position k equals to 1. T_s instants after an observer starts in central angle k, the angle distribution is $\mathbf{1}_k\mathbf{P}^{T_s}$. Because an observer's FoV size is $1+2a$, we multiply $\mathbf{1}_k$ by a binary circulant matrix $\mathbf{C}_a \in \{0,1\}^{K\times K}$ to account for FoV. For example, $\mathbf{C}_1$ for $K=5$ is:

$$\mathbf{C}_1 = \begin{bmatrix} 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 \end{bmatrix} \tag{2}$$

Suppose now that for central angle k, the server transmits stream $f(k)$ with distortion vector $\mathbf{d}_{f(k)}$, where $d_{f(k),l}$ is the distortion of angle l in stream $f(k)$. We can then write the expected distortion for this intra-coded streaming system as:

$$D(\{\mathbf{d}_i\}, f(\,)) = \sum_{k=1}^{K} q_k\ \mathbf{1}_k\mathbf{C}_a\mathbf{P}^{T_s}\mathbf{d}_{f(k)} \tag{3}$$

where the expected distortion D depends on *both* the distortion vectors $\mathbf{d}_i$ of different streams i *and* the mapping function $f(\,)$ from angles to streams.

If the 360 VR video streams are coded in GOP of H frames each, then the stream-switching delay becomes $T_s + H$, and the distortion term for each k needs to be computed for all H frames:

$$D(\{\mathbf{d}_i\}, f(\,)) = \sum_{k=1}^{K} q_k \sum_{h=0}^{H-1} \mathbf{1}_k\mathbf{C}_a\mathbf{P}^{T_s+h}\mathbf{d}_{f(k)} \tag{4}$$

4.3. Rate Constraints

Given distortion vector $\mathbf{d}_i$ of stream i, we define the coding rate as $r(\mathbf{d}_i) = \sum_{k=1}^{K} g(d_{i,k})$, where $g(d_{i,k})$ is in turn defined as a clipped Laplacian function with parameter σ:

$$g(d) = U(d_{\max} - d)\exp\left(-\frac{|d|}{\sigma^2}\right) \tag{5}$$

[3]https://en.wikipedia.org/wiki/Perron%E2%80%93Frobenius_theorem

where $U(\,)$ is a step function; *i.e.*, if $d \geq d_{\max}$, then rate $g(d)$ is 0. Parameter σ can be chosen according to the 360 VR video characteristics. Because distortion d is non-negative, we can drop the absolute value operator in practice.

Having defined $r(\mathbf{d}_i)$, we can define a storage constraint as follows. Denote by $\mathcal{S}$ the set of pre-encoded video streams, by Q the duration in time for the 360 video, and by B the storage budget in bits. We write the storage constraint as:

$$\sum_{i\in\mathcal{S}} r(\mathbf{d}_i) \leq B/Q \tag{6}$$

We can similarly define a transmission constraint for a transmission budget C in bps. Assuming a mapping function $f(\,)$ from angles to streams, we write:

$$\sum_{k=1}^{K} q_k r(\mathbf{d}_{f(k)}) \leq C \tag{7}$$

4.4. Objective Function

Assuming $H = 1$, collecting derived equations (3), (6) and (7), we write an unconstrained Lagrangian objective as:

$$\min_{\{\mathbf{d}_i\}, f(\,)} \sum_{k=1}^{K} q_k\ \mathbf{1}_k\mathbf{C}_a\mathbf{P}^{T_s}\mathbf{d}_{f(k)} + \lambda\sum_{i\in\mathcal{S}} r(\mathbf{d}_i) + \mu\sum_{k=1}^{K} q_k r(\mathbf{d}_{f(k)}) \tag{8}$$

where λ and μ are chosen parameters so that the storage constraint (6) and transmission constraint (7) are satisfied.

5. OPTIMIZATION ALGORITHM

We take an alternating optimization approach, where we optimize variables $\{\mathbf{d}_i\}$ and $f(\,)$ one at a time while keeping the other fixed. When $f(\,)$ is fixed, we take the derivative of the objective with respect to $d_{i,l}$ and set it to 0:

$$\sum_{k|f(k)=i} q_k \left[\mathbf{1}_k\mathbf{C}_a\mathbf{P}^{T_s}\right]_l + \underbrace{\left(\lambda + \mu \sum_{k|f(k)=i} q_k\right)}_{\gamma} \frac{\partial g(d_{i,l})}{\partial\, d_{i,l}} = 0$$

$$-\frac{1}{\gamma}\sum_{k|f(k)=i} q_k \left[\mathbf{1}_k\mathbf{C}_a\mathbf{P}^{T_s}\right]_l = \frac{\partial \exp\left(-\frac{d_{i,l}}{\sigma^2}\right)}{\partial\, d_{i,l}}$$

$$-\sigma^2 \log\left(\frac{\sigma^2}{\gamma}\sum_{k|f(k)=i} q_k \left[\mathbf{1}_k\mathbf{C}_a\mathbf{P}^{T_s}\right]_l\right) = d^*_{i,l} \tag{9}$$

where $[\,]_l$ denotes the l-th entry of a vector.

For intuition, we can check the boundary cases of (9) as follows. If angle l of stream i is not observed (summation in the argument of log is 0), then the left side of (9) evaluates to ∞, so we can set $d^*_{i,k}$ to $d_{\max}$. On the other hand, if angle l is observed with high probability (summation in the argument

of log is upper-bounded by 1), assuming σ^2/γ is also upper-bounded by 1, then $d^*_{i,l}$ is lower-bounded by 0.

When streams $\{d_i\}$ are fixed, we optimize $f(\,)$ simply as follows. For each angle k, we identify a stream i for k with the minimum expected transmission cost in (8).

5.1. Initialization

For a given number $|\mathcal{S}|$ of target streams, we perform initialization as follows. We evenly distribute the central angles of $|\mathcal{S}|$ streams in $\{1,\ldots,K\}$. For each stream i with central angle k, we set distortion $d_{i,l}$ to a constant d_1 for angle l where $|k-l| < T_s\, v_{\max}$; *i.e.*, angle l is reachable in T_s transitions. Otherwise, $d_{i,l} = d_{\max}$. d_1 is then adjusted so that the transmission constraint is met for this stream.

The number of streams $|\mathcal{S}|$ is varied to find a locally optimal solution.

6. EXPERIMENTS

6.1. Experimental Setup

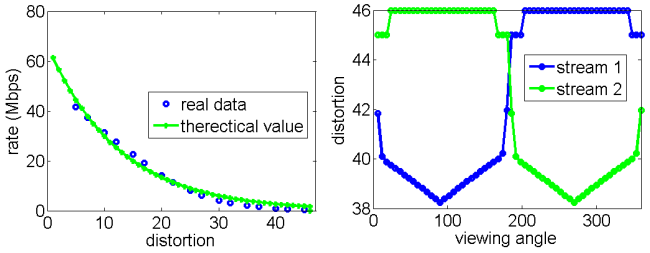
We use two 360 VR sequences captured by Kandao Technology[4], `indoor concert` and `outdoor walking`, for our experiments. Each video is 1 hour long at 30 fps. FoV is assumed to be $90°$, and $v_{\max}$ is $5°$. Video for one FOV has resolution 512×512. Number of discrete view angles K is 60, and RTT T_s is 3. We use a linear function to model angle transition probabilities: $p_{i,j}$ linearly decreases with $|i-j|$, and the slope of decrease is steeper at $\pi/2$ and $3\pi/2$, resulting in higher steady state probabilities q_k at these two angles.

As competitor we choose a non-switching scheme called *static*, which always sends an encoded video covering the entire 360 angles. For practical implementation, both our proposed scheme (called *adaptive*) and *static* use two QPs to encode each VR video stream; the two QPs are selected using *Lloyd-Max quantizer* [16] to approximate the theoretical Laplacian RD curve $r(d)$ shown in Fig. 3 (a). Test videos are first encoded at different QPs to generate empirical RD points, then the parameters of $r(d)$ are fitted.

6.2. Experimental Results

We assume two different channel bandwidths are available. We vary the available storage and show the tradeoff against visual quality (PSNR) in Fig. 4 for `indoor concert` and `outdoor walking`. Weight parameters λ and μ are tuned to satisfy bandwidth and storage constraints at each point. Each data point in Fig. 4 is marked by a square, circle or triangle to denote the optimal number of streams generated: 1, 2 and 3, respectively. *static* uses 1 stream (squares), and *adaptive* uses multiple streams (circles and triangles).

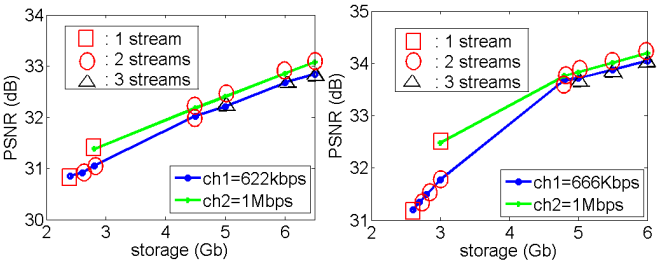
We observe that *adaptive* outperforms *static* for the two sequences—up to 2.9dB in PSNR at the same bandwidth but using more storage. For given channel bandwidth and storage, *adaptive* selects the optimal number of streams and view range for each stream via optimization of distortion vectors $\mathbf{d}_i$. Fig. 3 (b) (distortion versus viewing angle) shows the optimized distortion vectors $\mathbf{d}_i$ for two streams when the storage is 5Gb and bandwidth is 1Mbps. $d_{\max} = 46$ in this case, and the corresponding angle range is not encoded because there is zero probability of being observed (given our view interaction model). In contrast, viewing angles with high probabilities have low distortion values in $\mathbf{d}_i$. We observe also that the two steams overlap, as discussed in the Introduction, to guarantee good visual quality when user's head rotates in one RTT. Due to the low observe probability at the stream view range boundaries, the associated distortion values are relatively larger.



(a) R-D curve (b) distortion vs. view angle

Fig. 3. Illustration of R-D curve and streams' distortion vectors.

When storage is small, *static* and *adaptive* have the same performance for both $ch1$ and $ch2$. As more storage becomes available, relative performance of *adaptive* becomes better for both $ch1$ and $ch2$ when multiple streams are employed. On the other hand, by sending only one stream always, *static* cannot make use of extra storage to improve quality for a given channel bandwidth.



(a) `indoor concert` (b) `outdoor walking`

Fig. 4. PSNR versus storage for two competing schemes.

7. CONCLUSION

Transmitting 360 VR video in high quality over bandwidth-limited networks is difficult. In this paper, we pre-compute mulitple streams covering different overlapping view ranges at the server, and during streaming a single stream is selected corresponding to the user's tracked head rotation angle that minimizes the adverse effect of interaction delay. We formulate an optimization to find the optimal streams and the head-angle-to-stream mapping function simultaneously, solved via an alternating algorithm. Experimental results show that our multi-stream switching approach outperforms a single-stream approach by up to 2.9dB in PSNR.

[4]VR sequences will be made available at time of publication.

8. REFERENCES

[1] J. Jia and C.-K. Tang, "Image stitching using structure deformation," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, April 2014, vol. 30, no.4, pp. 617–631.

[2] J. Zaragoza, T.J. Chin, and Q.-H. Tran, "As-projective-as-possible image stitching with moving DLT," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, July 2014, vol. 36, no.7, pp. 1285–1298.

[3] S. Reichelt, R. Hausselr, 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 2010*, Orlando, FL, April 2010.

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

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

[6] T. Maugey, I. Daribo, G. Cheung, and P. Frossard, "Navigation domain partitioning for interactive multiview imaging," in *Special Issue on 3D Video Representation, Compression, Rendering, IEEE Transactions on Image Procesing*, September 2013, vol. 22, no.9, pp. 3459–3472.

[7] D. Ren, G. Chan, G. Cheung, V. Zhao, and P. Frossard, "Anchor view allocation for collaborative free viewpoint video streaming," in *IEEE Transactions on Multimedia*, March 2015, vol. 17, no.3, pp. 307–322.

[8] L. Toni, G. Cheung, and P. Frossard, "In-network view synthesis for interactive multiview video systems," in *IEEE Transactions on Multimedia*, May 2016, vol. 18, no.5, pp. 852–864.

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

[10] W. Dai, G. Cheung, N.-M. Cheung, A. Ortega, and O. Au, "Merge frame design for video stream switching using piecewise constant functions," in *IEEE Transactions on Image Processing*, June 2016, vol. 25, no.8, pp. 2896–2909.

[11] M. Hosseini and V. Swaminathan, "Adaptive 360 VR video streaming based on MPEG-DASH SRD," in *IEEE International Symposium on Multimedia*, San Jose, CA, December 2016.

[12] M. Hosseini and V. Swaminathan, "Adaptive 360 VR video streaming: Divide and conquer," in *IEEE International Symposium on Multimedia*, San Jose, CA, December 2016.

[13] O. Niamut, E. Thomas, and L. D'Acunto, "MPEG DASH SRD - spatial relationship description," in *ACM Conference on Multimedia Systems*, Klagenfurt, Austria, May 2016.

[14] T. Stockhammer, "Dynamic adaptive streaming over HTTP-standards and design principles," in *ACM Conference on Multimedia Systems*, San Jose, CA, February 2011.

[15] K. Sreedhar, A. Aminlou, M. Hannuksela, and M. Gabbouj, "Standard-compliant multiview video coding and streaming for virtual reality applications," in *IEEE International Symposium on Multimedia*, San Jose, CA, December 2016.

[16] Allen Gersho and Robert M. Gray, *Vector Quantization and Signal Compression*, Kluwer Academic Publishers, 1992.