

SERVER-SIDE RATE ADAPTATION FOR MULTI-USER 360-DEGREE VIDEO STREAMING

Chengming Liu

School of Communication Engineering
Shanghai University, China

Nuowen Kan, Junni Zou, Qin Yang, Hongkai Xiong*

School of Electronic Information and Electrical
Engineering, Shanghai Jiao Tong University, China

ABSTRACT

How to balance the tradeoff between the user experience and bandwidth utilization emerges a critical challenge for multi-user 360-degree video adaptive streaming. This paper studies the server-side rate adaptation strategy for multiple users which are competing for the server bandwidth capacity. A tile visibility probability model is established, by which the tiles are classified into predicted, marginal and invisible types. A fine-grained rate adaptation problem is formulated as a non-linear integer programming (NIP) problem, which aims at maximizing the video quality and navigation smoothness for multiple users. Thereafter, a steepest ascent algorithm with feasible starting point is developed to solve the proposed NIP problem in polynomial time. Finally, simulation results verify the performance of the proposed rate adaptation strategy.

Index Terms— 360-degree video, rate adaptation, server-side, field of view prediction, steepest ascent algorithm

1. INTRODUCTION

360-degree videos, one major virtual reality application, provide immersive sensation within a scene by using an omnidirectional camera to capture the scene from all directions. The user wears a head-mounted display (HMD) to watch the video and moves his head to change the viewing direction. As the bandwidth required to deliver 360-degree video is 4-6 times of traditional video with the same resolution [1], how to efficiently transmit such huge video content emerges a challenge for 360-degree video streaming.

Constrained by the field of view (FoV) (e.g., 90-degree vertically and 110-degree horizontally) of the HMD, the user at any given time can view partial video content. It means that the server can only stream the video portion corresponding to the user's current *viewport* (also called *user FoV* in this paper) to reduce the bandwidth consumption. Moreover, considering the user's viewport switch arising from fast head movement, the video portion outside the user FoV also requires to be delivered, but can be encoded at a very low rate. The user FoV depends on the position of the user's head, which needs to be

predicted before transmission. If the FoV prediction can reach 100% accurate, a simple strategy, i.e., streaming the predicted FoV with high quality and the outside area with low quality, can well function. However, it is hardly conduct absolutely accurate prediction, a margin area that compensates for the difference between the predicted and the real user FoV always needs to be transmitted at a relatively high rate to guarantee the quality of experience (QoE). The larger margins, the higher probability that the real FoV can be fully covered, and the more bandwidth required. This paper focuses on the adaptive streaming strategy for both the predicted FoV and the margins to balance the tradeoff between the user experience and bandwidth utilization.

HTTP adaptive streaming, on the basis of dynamic adaptive streaming over HTTP (DASH) [2], has become a desirable solution for bandwidth-limited 360-degree video delivery. Unlike the traditional video, the adaptive 360-degree video streaming not only provides representation selection for each video chunk, but offers multiple resolutions for each tile that is defined by spatial relationship description (SRD) [3], a new extension of DASH. Toni et al. in [4] studied tile-based adaptive streaming strategies and proposed a solution to determine the rate at which each tile is downloaded for maximizing the quality experienced by the user. Given the FoV and bandwidth estimation, Ghost et al. in [5] formulated different QoE metrics and designed a streaming algorithm for 360-degree video streaming. The authors in [6] proposed a viewport-adaptive 360-degree video streaming system, in which the front face is encoded in full quality while the other faces are encoded in low quality.

All the existing studies considered user-side rate adaption, where the user determines the best set of tile rates to download from the server based on bandwidth estimation and buffer occupancy. Such approaches have shown sub-optimal QoE performance when multiple users compete for the bandwidth [7] [8]. In this paper, we consider the server-side rate adaption for multiple users which are competing for the server bandwidth capacity with personal viewport requirement and bandwidth constraint. Moreover, a definition of the tile visibility probability is introduced, by which the tiles are classified into predicted, marginal and invisible tiles, and then a more fine-grained rate adaption strategy is proposed for maximizing the overall qualities of user experience. Our contributions

*The work has been partially supported by the NSFC grants No. 61622112, No. 61472234, No. 61720106001, No. 61529101, and the Program of Shanghai Academic Research Leader (17XD1401900).

are summarized as follows:

1) We design a tile visibility probability model to measure the probability that the tile is visible to the user, in which the FoV prediction accuracy, the head movement speed, and the chunk duration time are considered.

2) We formulate the server-side fine-grained rate adaptation problem as a non-linear integer programming (NIP) problem, which aims at maximizing the satisfaction for the received video and navigation smoothness of multiple users.

3) We develop a steepest ascent algorithm to solve the proposed NIP problem in polynomial time, where the feasible starting point is determined by the relaxed convex optimization problem.

2. SERVER-SIDE RATE ADAPTATION

In this section, we first model the tile visibility probability, and then formulate the server-side rate adaptation problem.

2.1. System Model

Consider a 360-degree spherical video that is projected into a rectangular panoramic video by using a certain projection method, e.g. equirectangular projection (ERP) [9]. The panoramic video sequence is segmented into a set \mathcal{T} of chunks of the same duration ℓ . Each panoramic frame within a chunk is divided into a set \mathcal{L} of tiles. Each tile is encoded into M representations, where $\mathcal{R} = \{R_1, R_2, \dots, R_M\}$ denotes the set of the coding rates satisfying $R_1 < R_2 < \dots < R_M$. Assume that the server sends the video simultaneously to a set \mathcal{K} of users. The bandwidth capacity of the server and each user k are B_s and B_k , respectively. For each chunk t , let $\mathcal{V}_{k,t}^p$ represent the set of *predicted tiles* that fully cover user k 's predicted FoV, and $\mathcal{V}_{k,t}^m$ be the corresponding set of *marginal tiles* that cover the difference between the predicted and real FoV. Further, let the combination $\mathcal{V}_{k,t} = \mathcal{V}_{k,t}^p \cup \mathcal{V}_{k,t}^m$ represent the set of *visible tiles* that is visible for user k when wearing the HMD, and $\bar{\mathcal{V}}_{k,t}$, i.e. the supplementary set of $\mathcal{V}_{k,t}$, be the set of *invisible tiles*, as shown in Fig. 1.

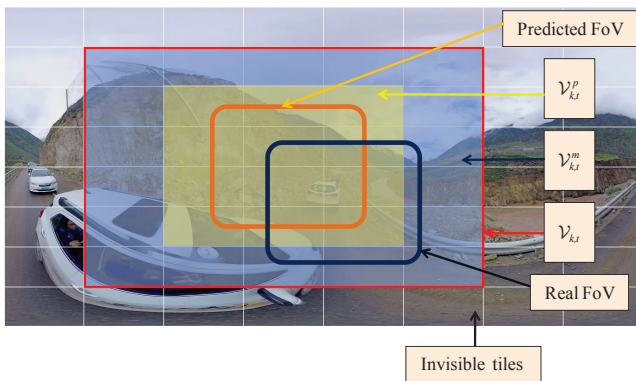


Fig. 1. Tile segmentation of a panoramic frame

2.2. Tile Visibility Probability

The tiles in a panoramic frame are classified into three types: predicted, marginal and invisible tile. The predicted tiles are streamed with high quality, while the marginal tiles are transmitted with moderate video quality. As for the invisible tiles, they are transmitted at the lowest rate R_1 to save the bandwidth. The set of predicted tiles can be easily determined by performing user FoV prediction as in [10]. The critical issue is how to find out an appropriate marginal tile set, as the margin size impacts both the quality and efficiency.

Let $P_{i,k,t}$ represent the probability that tile i of chunk t is visible to user k . The value of $P_{i,k,t}$ is the largest when the tile is located in the predicted FoV, then it decreases with the increase of the distance between the tile and the predicted FoV. For any tile i , $i \notin \mathcal{V}_{k,t}^p$, let d_i denote the distance between the center point of tile i and that of the predicted FoV. Besides the distance, $P_{i,k,t}$ has also relationship with user motion speed s , chunk length ℓ , and FoV prediction accuracy p [11]. For a long chunk or a fast user motion, the $P_{i,k,t}$ of the tiles outside the predicted FoV decreases very slowly because the user is more likely to watch the tiles beyond the predicted FoV.

Assume that the distance d_i meets the normal distribution with a zero mean, and a variance of $s \cdot \ell$, i.e., $d_i \sim \mathcal{N}(0, s \cdot \ell)$. Thus, we have

$$P_{i,k,t} = \begin{cases} p, & i \in \mathcal{V}_{k,t}^p, \\ p \cdot \int_{d_i}^{W/2} \frac{1}{\sqrt{2\pi s \ell}} \exp\left(-\frac{d^2}{2s \ell}\right), & \text{otherwise,} \end{cases}$$

where W is the width of 360-degree video in ERP format. Given a threshold α , for any tile i , $i \notin \mathcal{V}_{k,t}^p$, if its visibility probability satisfies $P_{i,k,t} \geq \alpha$, it is defined as a marginal tile. Otherwise, it is an invisible tile.

2.3. Problem Formulation

For each chunk t , the server seeks transmission rates for the predicted and marginal tiles to maximize the overall user satisfactions, subjected to both the server and users bandwidth capacity. Mathematically, the server-side rate adaptation problem can be formulated as the following P-1 problem:

$$\begin{aligned} \max_{\mathbf{R}} \quad & \sum_{k \in \mathcal{K}} \left(\sum_{i \in \mathcal{V}_{k,t}} U(R_{i,k,t}) P_{i,k,t} + w U\left(\min_{i \in \mathcal{V}_{k,t}^m} R_{i,k,t}\right) P_{i,k,t} \right) \\ \text{s.t.} \quad & \sum_{k \in \mathcal{K}} \sum_{i \in \mathcal{L}} R_{i,k,t} \leq B_s, \forall t \in \mathcal{T} \quad (1) \\ & \sum_{i \in \mathcal{K}} R_{i,k,t} \leq B_k, \forall k \in \mathcal{K}, \forall t \in \mathcal{T} \quad (2) \\ & R_{i,k,t} \in \{R_1, R_2, \dots, R_m\}, \forall i \in \mathcal{V}_{k,t}, \forall k \in \mathcal{K}, \forall t \in \mathcal{T} \quad (3) \\ & R_{i,k,t} = R_1, \forall i \in \bar{\mathcal{V}}_{k,t}, \forall k \in \mathcal{K}, \forall t \in \mathcal{T} \quad (4) \\ & R_{i,k,t} = R_{j,k,t}, \forall i, j \in \mathcal{V}_{k,t}^p, \forall k \in \mathcal{K}, \forall t \in \mathcal{T} \quad (5) \end{aligned}$$

$$R_{i,k,t} \leq R_{j,k,t}, \forall i \in \mathcal{V}_{k,t}^m, j \in \mathcal{V}_{k,t}^p, \forall k \in \mathcal{K}, \forall t \in \mathcal{T} \quad (6)$$

The user satisfaction function $U(\cdot)$ describes the satisfaction of the user for receiving a video, which is increasing and strict concave with respect to the video rate. The second term in the objective function is introduced to avoid uncomfortable degradation in QoE when the user switches the viewing direction from the predicted FoV to the margin area, where w is a weight. Constraints (3) define the optional quality levels for each visible tile. Constraints (4) set the quality level of all the invisible tiles to R_1 . Constraints (5) specify that all the tiles in the predicted FoV should have the same quality level. Constraints (6) ensure that the quality level in the margin area would not be higher than that in the predicted FoV.

Problem P-1 is a non-linear integer programming problem which is in general NP-hard. In the following, a steepest ascent algorithm [12] with the feasible starting point achieved from the relaxed convex problem is applied to find the solution in polynomial time.

3. STEEPEST ASCENT SOLUTION WITH CONVEX RELAXATION STARTINT POINT

Let $Q(\mathbf{R})$ represent the objective function of problem P-1, $B(\mathbf{R}) = \sum_{k \in \mathcal{K}} \sum_{i \in \mathcal{V}_{k,t}} R_{i,k,t}$ represent the total streaming rates of visible tiles for chunk t , and

$\mathbf{R} = (R_{1,1,t}, \dots, R_{L,1,t}, \dots, R_{1,K,t}, \dots, R_{L,K,t})$ be the possible rate combinations of all the visible tiles for all the users. Also, let \mathbf{R}_j^+ denote the rate R_i fixed to R_{i+1} on the j -th element of \mathbf{R} and others remain unchanged, where $i \in \{1, \dots, M-1\}$. The slope in the j -th direction at \mathbf{R} could be defined as $s_j(\mathbf{R}) = \frac{Q(\mathbf{R}_j^+) - Q(\mathbf{R})}{B(\mathbf{R}_j^+) - B(\mathbf{R})}$.

The steepest ascent algorithm iteratively finds the direction j in which the maximum slope of all the directions and the new \mathbf{R}_j^+ are achieved when the operation satisfies all the constraints of P-1. Finally, the algorithm converges after numerous iterations. To ensure the optimality of the achieved solution, we first describe three optimality conditions for the steepest ascent algorithm [13]:

1) *Cross-over*: For each j and $v > u$, the upper convex hull $Q_j^v(\mathbf{R})$ of achievable solutions with the j -th direction of \mathbf{R} fixed to R_v is greater than $Q_j^u(\mathbf{R})$ for sufficiently large $B(\mathbf{R})$.

2) *Cross-over ordered*: Suppose the cross-over rate $B^c(u, v, j)$ between two curves $Q_j^v(\mathbf{R})$ and $Q_j^u(\mathbf{R})$ is defined as the smallest rate such that $Q_j^u(\mathbf{R}) \geq Q_j^v(\mathbf{R})$ if $u > v$. Then $B^c(u, v, j)$ is greater than the rate of the first achievable solution on $Q_j^v(\mathbf{R})$ with rate greater than or equal to $B^c(v, u, j)$ for each j and $\omega > v > u$.

3) *Reachability*: For all j and u , the vectors corresponding to two adjacent achievable solutions on $Q_j^u(\mathbf{R})$ differ in exactly one index. Also, given a vector \mathbf{q} corresponding to the highest rate achievable solution less than or equal to $B^c(u, u-1, j)$ on $Q_j^u(\mathbf{R})$, the vector \mathbf{q}_j^+ gives the lowest

achievable rate solution on $Q_j^{u-1}(\mathbf{R})$ with rate greater than or equal to $B^c(u, u-1, j)$.

Theorem 1 [13] *The steepest descent algorithm will find all achievable (Q, B) values on the upper convex hull with $B(\mathbf{R})$ satisfied all constraints if the cross-over, cross-over ordering and reachability conditions hold.*

Fig. 2 shows three optimal Q-B curves which represent the $Q_j^u(\mathbf{R})$, $Q_j^{u-1}(\mathbf{R})$ and $Q_j^{u-2}(\mathbf{R})$, respectively. It is seen that $Q_j^u(\mathbf{R}) > Q_j^{u-1}(\mathbf{R}) > Q_j^{u-2}(\mathbf{R})$ for large $B(\mathbf{R})$, which means that the *cross-over* property is satisfied in this problem. Also, the *cross-over ordered* property is satisfied since $B^c(u, u-1, j) > B^c(u-1, u-2, j)$. However, the *reachability* cannot be guaranteed as there are too much directions in which \mathbf{q}_j^+ gives the lowest achievable solution. To meet the *reachability* property, a feasible starting point is required for this problem.

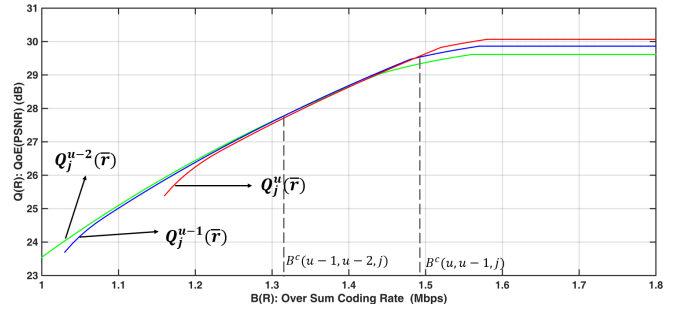


Fig. 2. Switching between optimal Q-B curves

Algorithm 1 Steepest Ascent Algorithm

Input: Feasible start point $\mathbf{R}_{\text{integer}}$

Output: Optimal solution \mathbf{R}^*

Initialisation : $\mathbf{R} = \mathbf{R}_{\text{integer}}$

1: Let the *active set* \mathcal{A} include the indices that the corresponded streaming rate of vector \mathbf{R} can be added.

LOOP Process

2: **while** \mathcal{A} is no-empty **do**

3: Calculate the $s_j(\mathbf{R})$ for all $j \in \mathcal{A}$

4: Get the index m that has the largest value of $s_j(\mathbf{R})$

5: If the tile of which coding rate corresponding to index m is in $\mathcal{V}_{k,t}^p$, let $\mathbf{R}' = \mathbf{R}_i^+$ for all $i \in \mathcal{V}_{k,t}^p$; Otherwise, let $\mathbf{R}' = \mathbf{R}_m^+$.

6: If \mathbf{R}' satisfy all the constrains of problem (1), $\mathbf{R} = \mathbf{R}'$, else, remove m from the active set.

7: **end while**

8: **return** \mathbf{R}

Let $\mathbf{R}_{\text{relax}}$ denote the solution of the relaxed convex problem of P-1, in which the discrete $R_{i,k,t}$ is relaxed to: $R_1 \leq R_{i,k,t} \leq R_M$. Further, let $\mathbf{R}_{\text{integer}} = \lfloor \mathbf{R}_{\text{relax}} \rfloor$ be the integers no larger than $\mathbf{R}_{\text{relax}}$, which is neighboring to the global

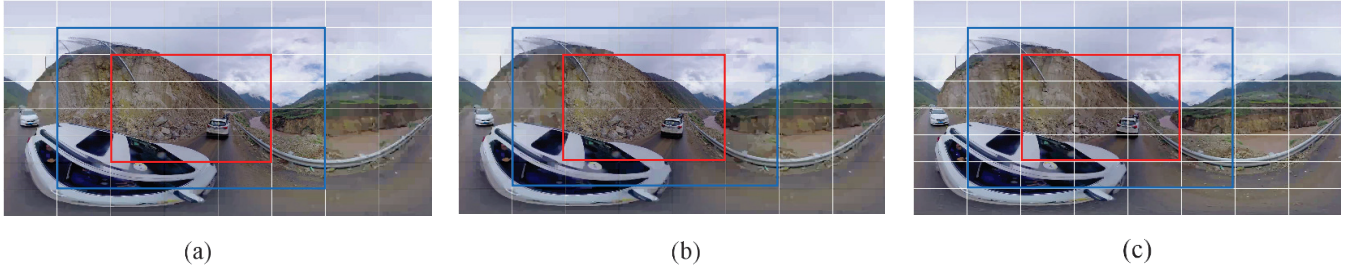


Fig. 3. Video frames achieved by different algorithms: (a) Proposed algorithm (b) Greedy algorithm (c) Baseline algorithm. (The red line area covers the predicted tiles and the blue one covers the visible tiles.)

optimal solution \mathbf{R}^* of P-1. Since $\mathbf{R}_{\text{integer}}$ is always near the global optimal point \mathbf{R}^* and the errors resulting from the conflict with the *reachability* property is very likely to be recovered. Taking the $\mathbf{R}_{\text{integer}}$ as the starting point for the steepest ascent algorithm will achieve the global optimal solution with a very high probability. The whole searching process is summarized in Algorithm 1.

4. SIMULATION RESULTS

We use the standard 360-degree test sequence, *Driving in Country* [14], with a resolution of 3840×1920 and a frame rate of 25 fps. The video in ERP format is divided into 8×8 tiles. We use High Efficiency Video Coding (HEVC) and provide six representations for each tile, with the QP of each representation being 17, 22, 27, 32, 37, 42. The parameter α is set to 0.1. 10 users are assumed to watch the videos with different predicted FoV.

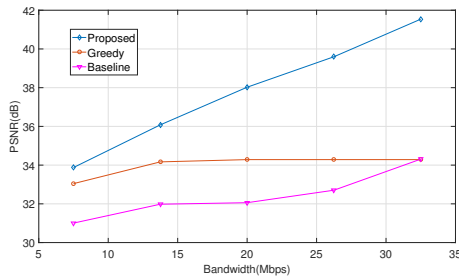


Fig. 4. Average PSNR of all the users

Fig. 3 shows the video frames achieved at the same user, where the baseline algorithm assumes that all the tiles have the same quality, while the greedy algorithm provides high quality levels only for the predicted tiles. In this case, the server bandwidth capacity is set to 32.5 Mbps, which is insufficient for delivering all the tiles with high quality. Therefore, all the tiles has to be streamed at a low rate by the baseline algorithm. For the greedy algorithm, only the predicted tiles has a high quality, resulting in a sharp quality degradation when the viewing direction switches to the margin area. In contrast,

with the proposed algorithm, not only the predicted FoV but the margin area are assigned high transmission rates.

Fig. 4 compares the average PSNR achieved at all the users. It is seen that with a very small server capacity, the greedy algorithm reaches the peak performance. The quality of the baseline algorithm increases slowly with the server capacity. compared with these two algorithms, the proposed algorithm has a significant improvement in video quality.

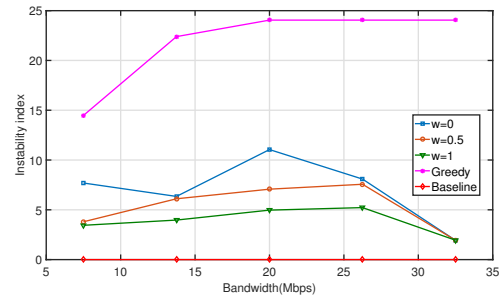


Fig. 5. Average instability index of all the users

Fig. 5 compares the average instability index of all the users where w in the proposed algorithm is set to 0, 0.5 and 1, respectively. The instability index of each user is defined as the standard deviation of the rates for all the visible tiles. It is observed that the smoothness of the baseline algorithm always remains stable as all the tiles have the same quality. For the proposed algorithm, the instability can be reduced by increasing the value of w .

5. CONCLUSION

In this paper, we have proposed a server-side rate adaptation algorithm for 360-degree video streaming where multiple users compete for the server bandwidth. The server chooses the optimal set of transmission rates for the tiles of each user. We divided the tiles into predicted, marginal and invisible types for a smooth video play when the user head moves from the FoV to other area. Simulation results show that the proposed algorithm outperforms the widely used baseline and greedy algorithm in average video quality and smoothness.

6. REFERENCES

- [1] Y. Bao, H. Wu, T. Zhang, A. A. Ramli, and X. Liu, "Shooting a moving target: motion-prediction-based transmission for 360-degree videos," in *IEEE International Conference on Big Data*, 2016, pp. 1161-1170.
- [2] I. Sodagar, "The MPEG-DASH standard for multimedia streaming over the internet," *IEEE MultiMedia*, vol. 18, no. 4, pp. 62-67, 2011.
- [3] O. A. Niamut, E. Thomas, L. D'Acunto, C. Concolato, F. Denoual, and S. Y. Lim, "MPEG DASH SRD: spatial relationship description," in *Proceedings of ACM International Conference on Multimedia Systems*, 2016.
- [4] S. Rossi and L. Toni, "Navigation-aware adaptive streaming strategies for omnidirectional video," in *Proceedings of IEEE International Workshop on Multimedia Signal Processing*, 2017.
- [5] A. ghosh, V. Aggarwal, and F. Qian, "A rate adaptation algorithm for tile-based 360-degree video streaming," *arXiv preprint arXiv: 1704.08215*, 2017.
- [6] X. Corbillon, G. Simon, A. Devlic, and J. Chakareski, "Viewport-adaptive navigable 360-degree video delivery," in *Proceedings of IEEE International Conference on Communications*, 2017.
- [7] Z. Li, X. Zhu, J. Gahm, R. Pan, H. Hu, A. C. Begen, and D. Oran, "Probe and adapt: rate adaptation for http video streaming at scale," *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 4, pp. 719-733, Apr. 2014.
- [8] S. Akhshabi, L. Anantkrishnan, C. Dovrolis, and A. C. Begen, "Server-based traffic shaping for stabilizing oscillating adaptive streaming players," in *Proceedings of ACM NOSSDAV*, 2013.
- [9] K. K.-Sreedhar, Al Aminlou, M. M. Hannuksela, and M. Gabbouj, "Viewport-adaptive encoding and streaming of 360-degree video for virtual reality applications," in *Proceedings of IEEE International Symposium on Multimedia*, 2016, pp. 583-586.
- [10] Y. Bao, T. Zhang, A. Pande, H. Wu, and X. Liu, "Motion-prediction-based multicast for 360-degree video transmissions," in *Proceedings of IEEE International Conference on Sensing, Communication, and Networking*, 2017.
- [11] P. R. Alface, J. F. Macq, and N. Verzijp, "Interactive omnidirectional video delivery: a bandwidth-effective approach," *Bell Labs Technical Journal*, vol. 16, no. 4, pp. 135-147, 2012.
- [12] G. P. Akilov and L. V. Kantorovich, *Functional Analysis*, 2nd edition. Pergamon Press, 1982.
- [13] Y. Sermadevi and S. S. Hemami, "Efficient bit allocation for dependent video coding," in *Proceedings of IEEE Data Compression Conference*, 2004.
- [14] W. Sun and R. Guo, "Test sequences for virtual reality video coding from letinvr," in *Joint Video Exploration Team of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, JVET-D0179*, 2016.