Optimized Receiver Control in Interactive Multiview Video Streaming Systems

Xue Zhang*, Laura Toni†, Pascal Frossard‡, Yao Zhao* and Chunyu Lin*
*Institute of Information Science, Beijing Jiaotong University, China
{xuezhang, yzhao, cylin}@bjtu.edu.cn
†Department of Electronic & Electrical Engineering, UCL, UK, l.toni@ucl.ac.uk
‡LTS4, EPFL, Switzerland, pascal.frossard@epfl.ch

***Abstract*—Multiview applications endow final users with the possibility to freely navigate within 3D scenes with minimum-delay. High-quality rendering of the scene is enabled by transmitting multiple high-quality camera views, which can be used to synthesize additional virtual views to offer a smooth navigation in the scene. When network resources are limited, the set of camera views needs to be properly selected by the client. The right tradeoff between coding artifacts (reducing the quality of camera views) and virtual synthesis artifacts (reducing the number of camera views sent to users) has to be optimized. Existing client adaptation logic strategies usually fail to properly consider the content characteristics and the client navigation properties in the view selection problem. We therefore propose an optimal representation selection for interactive multiview HTTP adaptive streaming (HAS), with a complete problem formulation to select the optimal set of camera views that optimize the navigation quality experienced by the user while satisfying the bandwidth constraints. We show that our optimization problem is NP-hard and develop an effective solution based on a dynamic programming algorithm with polynomial time complexity. Simulation results show significant navigation quality improvement compared to two baseline multiview adaptation logic solutions. This confirms that adaptation logics have to consider both video content and interactivity level of the user in the representation selection strategy.**

***Index Terms*—Dynamic adaptive streaming over HTTP, multiview video plus depth, representation set, multiview navigation, dynamic programming.**

I. Introduction

Last years have witnessed the advent of disruptive interactive and immersive video technologies, where a user can freely navigate within a 3D scene via images captured from multiple cameras. This is possible due to the free-viewpoint technology, where a virtual viewpoint can be synthesized at decoder via depth-image-based rendering (DIBR) [1] using texture and depth maps of camera views, namely anchor views. The quality of the synthesized viewpoints generally increases with both the quality of the anchor views and the similarity or proximity between the anchor views and the synthesized views. The optimization of the quality at the client therefore corresponds to the proper selection of the camera views and their encoding rates in resource constrained settings.

HTTP adaptive streaming (HAS), the universal technology for video streaming over the Internet, offers the possibility for users to adaptively select among different versions (different coding rates and resolutions) of video streams that have been pre-encoded and stored on a server. It provides an ideal framework for interactive multiview (MV) navigation, where each media client can choose different views along with different encoding rates, in order to maximize the video quality. Most of the research efforts in optimizing the client behavior have however focused on classical video streaming applications, while interactive MV streaming is highly dependent on particular factors like view synthesis artifacts and switching delays. In this work, we fill this gap and propose an optimal adaptation strategy for MV interactive users.

In more details, we consider the scenario of MV video sequences stored at the main server of the service provider (e.g., Netflix, YouTube). Each view corresponds to a sequence of texture images and depth maps captured by a given camera. Each view is pre-encoded into different representations. Each representation is then decomposed into temporal segments (usually 2s long) and stored at the server. The client then requests the best set of representations for the current segment based on both its level of interactivity and the available bandwidth. The best set of representations is defined as the one that permits to effectively reconstruct a navigation window at the client, namely a range of consecutive virtual views that can potentially be displayed by the client during the duration of the video segment. To achieve this goal, we provide a formal problem formulation to jointly optimize the subset of camera views and their encoding rates that should be requested by the client, among the ones available at the server. The proposed optimization leads to an optimal solution that takes into account both coding and virtual synthesis artifacts that affect the navigation quality. Since our optimization problem is NP-hard, we propose an effective solution based on dynamic programming (DP) algorithm to reach optimality with polynomial time complexity. Our adaptation logic strategy has been compared with a few heuristic algorithms from the literature and simulation results show significant gains (in terms of navigation quality) under different streaming scenarios. This means that the proposed optimization framework is able to find the right combination of representations that exploits at best the available resources for the considered client. This reflects into a better usage of the available network resources and into higher satisfaction of the final users.

Only a few works in the literature have studied the optimization of HAS systems for MV streaming. The optimization

of the representations to store at the server for HAS MV streaming under simplified assumptions on the client control strategy has been studied in [2]. In this work, we rather target the optimization at the client side for HAS MV as [3], [4]. In [3], the client adjusts the downloading bit rate by varying the number of anchor views but under the constraint of equal coding rate for all camera views. Equal rate across views is a limiting constraint in multiview systems [5]. A two-step rate adaption approach is further proposed in [4], where the reference views are chosen and then the optimal bit rate is selected. We rather propose to optimize jointly the reference views and the coding rates, which permits to find better tradeoff and to reach significantly higher performance in most settings by carefully considering the video content characteristics, the user behavior and the network availability altogether.

The rest of this paper is organized as follows. In Section II, we present the system model and the MV representation selection optimization problem. Our solution is given in Section III. Simulation results are provided in Section IV and conclusions are given in Section V.

II. Problem Formulation

A. System model

We consider the MV-based HAS system depicted in Fig. 1. At the server side, the set of camera views $\mathcal{V} = \{1, 2, 3, \cdots, N\}$ per video sequence are coded. Each texture image is pre-encoded into different *representations*. Without loss of generality, we consider one spatial resolution and multiple encoding rates. Let $\mathcal{R}$ be the set of coding rates for each camera view, and

$$\mathcal{T} = \{(v_i, r_i)\}_i, \text{with } v_i \in \mathcal{V}, r_i \in \mathcal{R} \tag{1}$$

be the set of representations stored at the main server and made available to clients[1]. The pair (v_i, r_i) identifies a representation of camera view v_i encoded at rate r_i, $\forall 1 \leq v_i \leq N$. Since accurate depth information has high importance for view synthesis but relatively low coding rate cost, depth maps are encoded once at high quality. The texture image is encoded at rate r_i. The resulting representations are divided into a set of segments with equal playback duration τ (typically 2s long).

At the client side, the set of viewpoints that can be displayed is $\mathcal{U} = \{1, 1+\Delta, 1+2\Delta, \cdots, 2, \cdots, N\}$, where $\Delta \in [0, 1)$ is the minimum space between two adjacent virtual views. We consider that v represents any camera view, while u identifies any either virtual viewpoint or camera view that can be displayed during the navigation. Any virtual viewpoint u can be rendered using a pair of left and right reference view v_L and v_R, $v_L < u < v_R$ and $v_L, v_R \in \mathcal{V}$, via a DIBR technique.

For each navigation segment, the client sends a downloading request to the server. At the downloading opportunity t, S_k is the segment eventually displayed by the client, while $S_{k+\ell}$ is the segment to be downloaded. There is therefore a mismatch between the downloading and displaying time of $T = \ell\tau$ seconds. Assuming that the last viewpoint displaying in S_k at t is denoted by $u \in \mathcal{U}$, and that ρ is the maximum speed at which a user can navigate to adjacent views, $w(u) = [u - \rho T, u + \rho T] = [U_L(u), U_R(u)]$ is the range of viewpoints that can potentially be displayed by the user in $S_{k+\ell}$. We call this range the *navigation window*. In order to guarantee a zero-delay view-switching, the adaptation logic has to select the best set of representations such that any viewpoint in the navigation window can be reconstructed on time at the client.

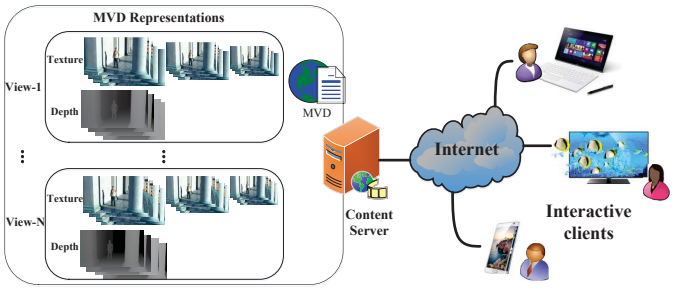


Fig. 1. Multiview-based HAS system

B. Navigation Distortion

We now evaluate the distortion experienced by a client downloading the set of representations $\mathcal{T}_d \subseteq \mathcal{T}$ while navigating in the window w,[2] as illustrated in Fig. 2. Each viewpoint u in the navigation window will be displayed at a quality $d_u(v_k, r_k, v_{k+1}, r_{k+1})$, where v_k, v_{k+1} are the left and right reference views and r_k, r_{k+1} are the corresponding coding rates respectively, with $(v_k, r_k), (v_{k+1}, r_{k+1}) \in \mathcal{T}_d$. This means that a user, given $\mathcal{T}_d$, navigates in the navigation window w at the quality

$$D(\mathcal{T}_d, w) = \sum_{k=1}^{|\mathcal{T}_d|-1} \sum_{u \in w} d_u(v_k, r_k, v_{k+1}, r_{k+1}) \tag{2}$$

where we assume that consecutive camera views in $\mathcal{T}_d$ are used as anchor views for all virtual viewpoints between them [6].

The objective of the adaptation logic is then to seek for the best subset $\mathcal{T}_d^*$ that minimize the distortion on the navigation window of interest, subject to the bandwidth constraints.

C. Optimization Problem Formulation

We can now formulate a navigation-aware optimization problem for MV adaptive streaming. Given a complete representation set of MVs, the navigation window of interest for the user and the available network bandwidth between the server and user, we want to determine which representations should be downloaded such that the navigation distortion experienced at the end-user side is minimized. More formally, a particular

[1] We consider the same $\mathcal{T}$ for all videos, but our work can be easily extended to the case of unequal $\mathcal{T}'s$

[2] For the sake of clarity, in the following we do not explicit the dependency of w on the viewpoint u displayed at t.

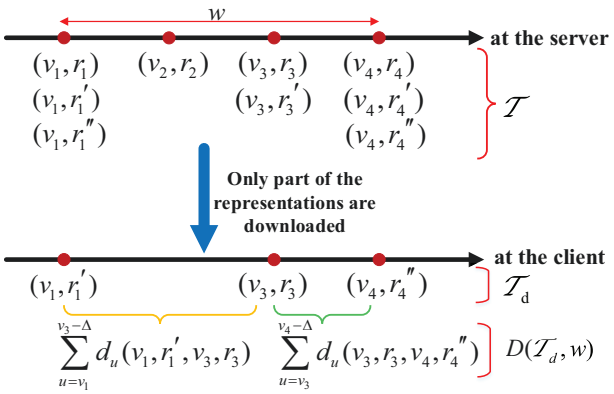


Fig. 2. Example of representation set available at the server ($\mathcal{T}$), representation set downloaded by one final user ($\mathcal{T}_d$), with the associated distortion experienced while navigating the scene.

client searches for

$$\mathcal{T}_d^* : \arg\min_{\mathcal{T}_d \subseteq \mathcal{T}} D(\mathcal{T}_d, w) \quad \text{s.t.} \sum_{\forall i:(v_i,r_i)\in\mathcal{T}_d} r_i \leq c \tag{3}$$

where c corresponds to the bandwidth constraint.

The optimal MV representations selection problem in (3) is unfortunately NP-hard. This can be proven by noting that the reduced case of $|\mathcal{R}| = 1$ is shown as a camera view selection problem. This special problem can be formulated as a *set cover* (SC) problem [6], which is a NP-hard. Optimizing jointly camera view subsets and encoding rates is no easier than solving the SC problem, thereby the problem in (3) is also NP-hard in general cases.

III. Optimal Representation Selection

We present here an effective solution based on a DP algorithm to solve the optimization of (3) with a polynomial time complexity.

Given a representation $(v,r) \in \mathcal{T}_d$, we define the aggregate distortion $\Phi(v,r,c)$ as the minimum distortion experienced between $\max\{U_L, v\}$ and U_R, when a remaining rate budget c is available for additional reference views. We can write the iterative property as follows:

$$\Phi(v,r,c) = \min_{(v_i,r_i),v_i>v} \left\{ \sum_{u=\max\{U_L,v\}}^{v_i-\Delta} d_u(v,r,v_i,r_i) + \Phi(v_i,r_i,c-r_i) \right\} \tag{4}$$

The equation (4) states that, when one of the optimal representation (v_i,r_i) is selected for download between $[v,U_R]$, the range of views $[v,U_R]$ is decomposed into two ranges $[v,v_i)$ and $[v_i,U_R]$. All viewpoints in the first range will be synthesized by the pair of camera views (v,v_i). In the second range $[v_i,U_R]$, other camera views can be selected for downloading with a total bitrate budget of $c-r_i$ as depicted in Fig. 3.

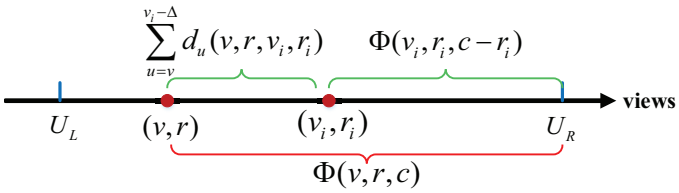


Fig. 3. The recursion property in the DP solution

Evaluating

$$\min_{\{v_L \leq U_L, r_L\}} \Phi(v_L, r_L, C - r_L) \tag{5}$$

leads to the solution of the problem optimization in (3) under the assumptions that i) only one camera view at one coding rate is selected in $\mathcal{T}_d$, ii) the most left camera view in $\mathcal{T}_d$ is such that $v_L \leq U_L$. These conditions are satisfied for most common 3D sequences [6], therefore (5) solves (3) in almost all multicamera scenarios. Due to the recursion shown in (4), (5) can be evaluated by DP.

We can deduce the computational complexity of our solution in (4) from a bound on the size of DP table and the cost in computing each table entry. For the sake of clarity in the notation, let the number of selected reference views and the number of views covered in the navigation window be $N_v = |\mathcal{T}_d|$ and $N_u = (U_R - U_L)/\Delta + 1$, respectively. The size of the DP table Φ is no larger than $N_v \times N_c \times |\mathcal{R}|$, where N_c is the number of channel bandwidth values that can be experienced during the optimization. For each entry in the DP table, we need to consider at most $(N_v - 1) \times |\mathcal{R}| + 1$ candidate camera views and for each of them, we need to evaluate the distortion to compare. Hence the complexity in computing each entry over all navigation views is $\mathcal{O}((N_v - 1)|\mathcal{R}| + 1)N_u)$. Generally, the overall computation complexity of our proposed algorithm in (4) is $\mathcal{O}(N_v N_c |\mathcal{R}|((N_v - 1)|\mathcal{R}| + 1)N_u)$, which can be approximated by $\mathcal{O}(N_u N_c N_v^2 |\mathcal{R}|^2)$.

IV. Experimental Evaluation

In this section, we study the performance of our algorithm and we show the navigation quality gains offered by our proposed optimal MV adaptation logic.

A. Simulation framework

1) System Settings: We considered three multiview video sequences at $1080p$ resolution, namely "*Hall*", "*Shark*", and "*Dancer*". The sequences are highly heterogenous in terms of coding and view synthesis efficiency, and are thus representative of various video categories. "*Dancer*" for example is a very dynamic sequence highly affected by coding artifacts, while "*Hall*" is a quite static scene but with a 3D geometry that renders virtual view synthesis highly challenging. For each video sequence, we consider two sets of coding rates that can be stored at the server, namely $\mathcal{T}1$ and $\mathcal{T}2$ provided in Table I. We consider 50 segments (50τ seconds) for each video sequence. The adaptation logic is activated at each downloading opportunity by the client, whose available bandwidth varies

TABLE I
THE REPRESENTATION AVAILABLE SETS

	$\mathcal{T}1$	$\mathcal{T}2$
View Set	1 2 3 4 5 6 7 8 9	1 3 5 7 9
Encoding rate Set (Mbps)	0.2 0.3 0.5 1 2 3 4 6 8 10	0.2 0.5 2 4 6
Bandwidth (Mbps)	0.5 1 2 3 4 5 6 8 10	

over time following a Markovian model, as widely used in the literature [7]. We set the Markov transition matrix that allows transitions to adjacent states with probability $2p_c/3$ and two-state jumps with probability $p_c/3$. Finally, to emulate heterogenous clients, we generate different navigation paths following the dynamic MV navigation model in [8] with minimum space between two adjacent views $\Delta = 0.1$. More specifically, we simulate i) a *uniform* navigation, when the user has the same probability of displaying the current view, or switching to the left or right view, and ii) a *non-uniform* navigation, when the user has a probability p_n of displaying the current view and $(1-p_n)/2$ of switching to the left or right view.

In the following, for each realization of both the channel and user navigation path, the adaptation logic is activated over 50 downloading opportunities (one per segment as in the regime phase of HAS system). The resulting distortion is averaged over multiple realizations. It is worth noting that our simulation considers some approximations (infinite playback buffers, exact channel estimation, etc.) with respect to real HAS systems. But it does not impact on our objective in this paper, which is to demonstrate the benefit of considering content and interactivity information in the optimal representation selection for a HAS client in a stationary regime.

2) Synthesis Distortion Function: For a given navigation window w, we evaluate the average distortion as $(1/N_u)\sum_{u\in w} d_u$, with N_u being the number of viewpoints in the navigation window. We adopt the synthesis distortion model from [2], provided in the following for clarity:

$$d_u(v_L, v_R) = \alpha D_{\min} + (1-\alpha)\beta D_{\max} + [1-\alpha-(1-\alpha)\beta]D_I \quad (6)$$

where $D_{\min} = \min\{D_L, D_R\}$, $D_{\max} = \max\{D_L, D_R\}$, D_I is the inpainted distortion, and D_L, D_R are the distortions of left and right reference views, respectively. Here, $\alpha = \exp(-\xi|u - v_{\min}|$, and $\beta = \exp(-\xi|u - v_{\max}|)$, with $v_{\min} = v_L$, $v_{\max} = v_R$ if $D_L \leq D_R$, otherwise, $v_{\min} = v_R$, $v_{\max} = v_L$ if $D_L > D_R$. The parameters ξ and D_I can be evaluated by curve fitting. Similarly, we set the inpainting distortion to $D_I = 0.35$, and $\xi = \{0.35, 0.52, 1.32\}$ for "*Dancer*" ("sport-action" type of video), "*Shark*" ("cartoon" type of video), and "*Hall*" ("movie" type of video), respectively.

Finally, the distortion of the coded camera views follows the model

$$D_v = 1 - (a - \frac{b}{r_v + e}) \quad (7)$$

where r_v is the coding rate, a, b and e are parameters that depend on both the content characteristics and the resolution of the video and are set to fit experimental $(1-\text{VQM})$ data points, where VQM is Video Quality Metric [2]. Note that a visually pleasant video usually has a VQM score below 0.2 and a gain in VQM of 0.1 is a good quality improvement as shown in [9].

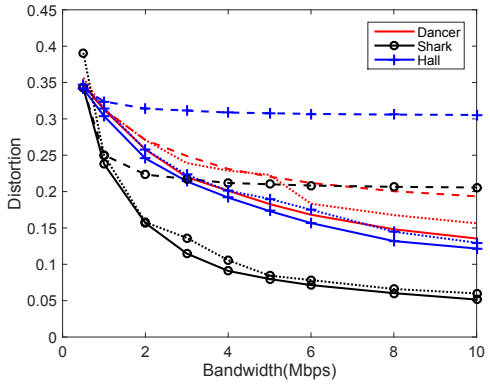


Fig. 4. Distortion comparison of a client having a navigation window $w = [1.5\ 8.5]$ with respect to bandwidth capacities using $\mathcal{T}1$. Solid lines show the performance of our optimization, while dotted lines and broken lines show the performance of "view-based adaptation logic" and "rate-based adaptation logic", respectively.

3) Baseline Algorithms: Our proposed algorithm is compared to two recent works: the one proposed in [3] is labeled in the following as "view-based adaptation logic", while the second one is extrapolated by [4] and labeled as "rate-based adaptation logic". The "view-based adaptation logic" optimizes the best subset of camera views given a total channel constraint, but under the constraint of equal coding rate for all selected camera views. In [4], a two-step algorithm is used, where in the first step the set of camera views is selected, and then the rate per camera view is optimized. The algorithm was originally limited to subsets of two or three camera views, and it has been extended here to the case of navigation window. This means that the "view-based adaptation logic" first selects two camera views for one segment that better cover the navigation window, then the coding rates for the selected camera views are optimized given the channel constraints. In our work, we rather consider a joint optimization of the camera views subset and the coding rates.

B. Simulation Results

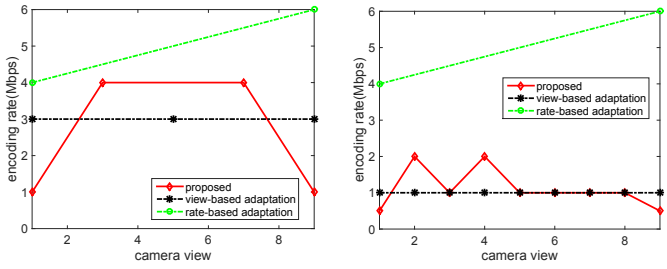
We first compare the proposed adaptation logic with the baseline in the case of a fixed navigation window. This particular scenario of low-interactivity is the most favorable one for the baseline algorithms, which do not fully take into account interactivity in their optimization.

In Fig. 4, the expected distortion as a function of the available bandwidth is provided for a navigation window $w = [1.5\ 8.5]$ when the representation set $\mathcal{T}1$ is available at the server. Simulation results are provided for our optimization (solid lines) as well as competitor algorithms, i.e., "view-based adaptation logic" (dotted lines) and "rate-based adaptation logic" (broken lines). It can be observed that, even in this particular static scenario, the proposed optimization always outperforms the baseline ones for any channel constraint with

a gain up to 0.05 with respect to "view-based adaptation logic" for the "*Shark*" and a gain up to 0.18 with respect to "rate-based adaptation logic" for the "*Hall*". This is because our method is able to find the right tradeoff between coding and synthesis artifacts.

To better understand this tradeoff, in Fig. 5 we provide the optimal representation sets for each video sequence for all algorithms, when the channel constraint is set to $c = 10Mbps$. Each point along the curves is an additional representation whose camera view index is indicated in the x-axis and its coding rate is indicated in the y-axis. For the "*Dancer*" sequence, the proposed optimization selects 4 views at medium-high rates to cover the navigation window, while a larger number of views at lower rates are selected for the "*Hall*" sequence. This is explained by the fact that "*Dancer*" sequence is highly affected by coding artifacts (due to the high-motion content) and not drastically by the synthesis artifacts (due to a simple scene geometry). On the contrary, "*Hall*" has the largest dissimilarity among adjacent camera views, making the synthesis process highly challenging. Therefore, many camera views are selected in such a way that virtual viewpoints are always synthesized by close-by anchor views. To meet the channel constraints, the camera views downloaded for "*Hall*" are the ones encoded at lower rate. Therefore, the joint optimization of both camera views and encoding rates leads to an unequal allocation of the $10Mbps$ available per sequence, based on the content characteristics. This unequal allocation is a key concept of our method and it is not achieved by the baseline methods. The view-based adaptation logic is limited to the same rate for all the views and most of the time selects many views but at low coding rate. This might be convenient for the "*Hall*" sequence but not for "*Dancer*". On the contrary, the rate-based adaptation leads to a limited number of downloaded views but at high coding rate. This can be a close to optimal camera selection for "*Dancer*" but not for the "*Hall*" sequence.

We now consider more interactive scenarios, where users navigate within the 3D scene. This leads to a variation of the navigation window over time. We simulate three types of navigation paths: (1) a uniform navigation with view 2.4 as first viewpoint; (2) a non-uniform navigation with $p_n = 0.3$ and view 2.4 as starting point; (3) a non-uniform navigation with $p_n = 0.6$ and view 5.1 as initial view. To better understand the temporal variation of distortion, we first depict the simulated distortion over time derived from a specific bandwidth constraint realization. This varied bandwidth is randomly generated by the channel model with $p_c = 0.5$ and indicated in the right y-axis of Fig. 6. For each segment downloaded progressively, the experienced distortion for the navigation window of interest is provided in the left y-axis of Fig. 6 for the "*Hall*" video sequence and with $\mathcal{T}1$ as the set of representations available at the server. As expected, the greater the available bandwidth the lower the distortion. Most importantly, despite the low channel bandwidth values, our adaptation logic outperforms the baseline ones since it is able to adapt its requests to the interactivity of users. Up to 0.06



(a) *Dancer* (b) *Hall*

Fig. 5. Comparison of selected optimal representation sets with navigation window [1.5 8.5] at $\mathcal{C} = 10Mbps$ using $\mathcal{T}1$.

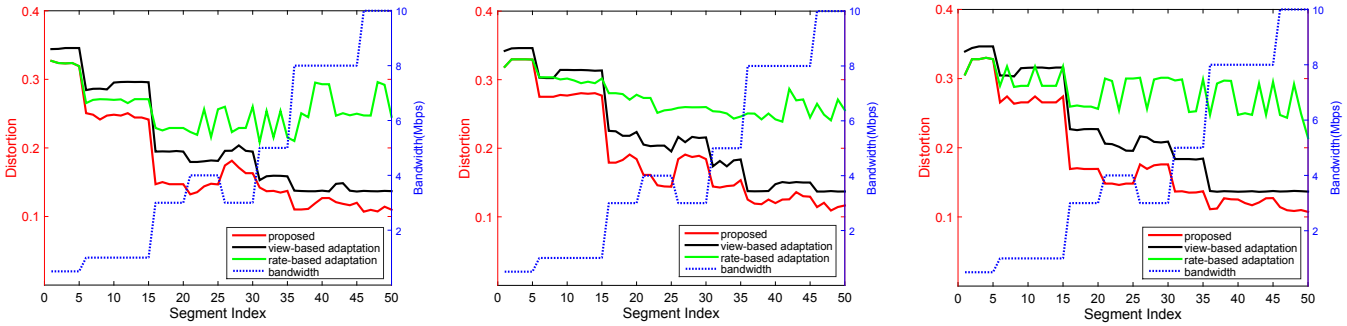
and 0.19 gains in the uniform case are achieved with respect to view-based and rate-based adaptation logic, respectively.

To conclude, we test our proposed adaptation logic for different representation sets stored at the main server as well as different video sequences. Dynamic channels are considered with $p_c = \{0.25, 0.5, 0.75, 0.9\}$. The results are shown in Fig. 7 and Fig. 8 respectively for $\mathcal{T}1$ and $\mathcal{T}2$ representation sets at the server. In both scenarios, the performance of proposed adaptation algorithm (solid curves) substantially outperforms that of two comparative algorithms (dashed curves) for all categories of clients. The gain is however more limited in the case of a more limited representation set (Fig. 8). This is expected since the small set $\mathcal{T}2$ reduces the search space in the optimization as well as the room for finding optimal solutions. However, in the second case of non-uniform navigation for the "*Hall*" sequence, when using $\mathcal{T}1$, the overall mean distortion reduction that we achieve is up to 0.03 with respect to the "view-based adaptation logic" and 0.1 with respect to "rate-based adaptation logic", while using $\mathcal{T}2$, we can also achieve the distortion reduction up to 0.03 and 0.09, respectively. We recall that a distortion reduction of 0.1 is considered to be a significant improvement.

In summary, the above results have shown that the navigation distortion can be reduced for clients in the interactive MV system when the optimal representation set is designed following our joint optimization logic. This shows the importance of taking into consideration the video content characteristics, bandwidth constraints, the users interactivity and representation sets available at the server when selecting the content to be downloaded by the client.

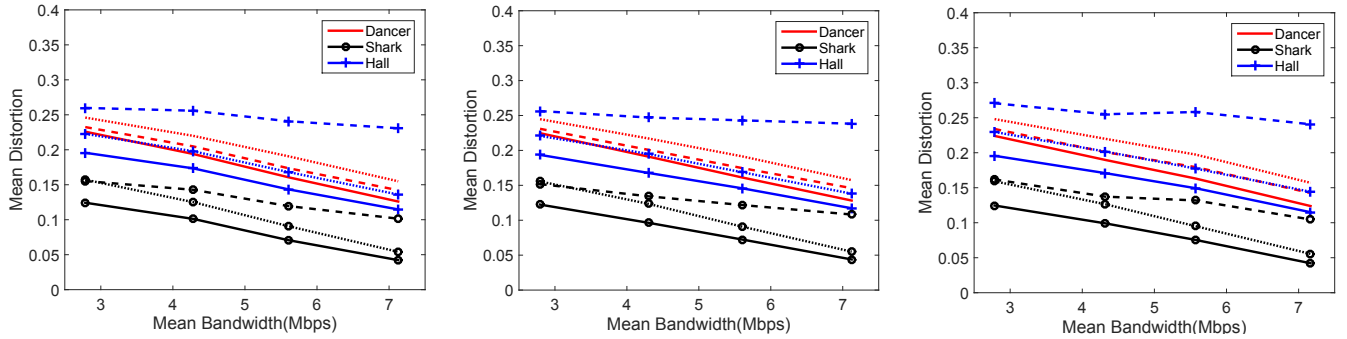
V. Conclusion

In this paper, we study a navigation-aware HAS logic optimization problem for interactive MV video systems in order to minimize the navigation distortion and view-switching delay. To the best of our knowledge, it is the first work about formal optimization of HAS-client controller for adaptive MV streaming that jointly select the best anchor views subsets and the corresponding encoding rates. Our algorithm properly takes into consideration both video content characteristics and user interactivity level and outperforms competitor algorithms in different scenarios. We show that it is necessary to find the proper tradeoff between view quality and number of reference



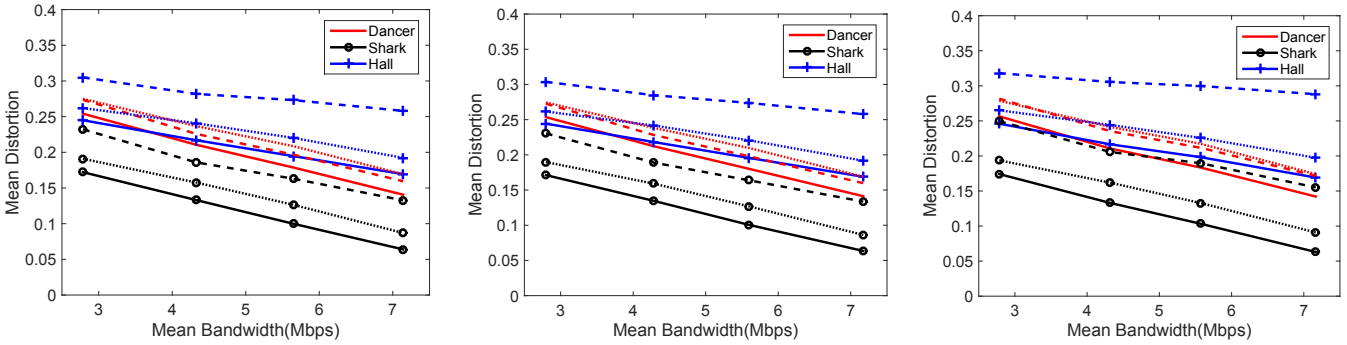
(a) *uniform* (b) *non-uniform with* $p_n = 0.3$ (c) *non-uniform with* $p_n = 0.6$

Fig. 6. Distortion comparison over time in different navigation distribution cases with a specific channel realization using $\mathcal{T}1$ for "*Hall*".



(a) *uniform* (b) *non-uniform with* $p_n = 0.3$ (c) *non-uniform with* $p_n = 0.6$

Fig. 7. Comparison of average distortion over time in different navigation distribution cases using $\mathcal{T}1$.



(a) *uniform* (b) *non-uniform with* $p_n = 0.3$ (c) *non-uniform with* $p_n = 0.6$

Fig. 8. Comparison of average distortion over time in different navigation distribution cases using $\mathcal{T}2$.

views in constrained-resource networks. Future work will consider the deployment of the proposed solution in realistic systems.

REFERENCES

[1] D. Tian, P.-L. Lai, P. Lopez, and C. Gomila, "View synthesis techniques for 3D video," *Proceedings of SPIE*, vol. 7443, pp. 74 430T–1–11, 2009.

[2] L.Toni and P. Frossard, "Optimal representations for adaptive streaming in interactive muti-view video systems," *ArXiv*, vol. abs/1609.04196, 2016.

[3] T. Su, A. Sobhani, A. Yassine, S. Shirmohammadi, and A. Javadtalab, "A DASH-based HEVC multi-view video streaming system," *Journal of Real-Time Image Processing*, pp. 1–14, 2015.

[4] A. Hamza and M. Hefeeda, "Adaptive streaming of interactive free viewpoint videos to heterogeneous clients," in *Proc. ACM Multimedia Systems Conf.*, Klagenfurt, Austria, May 2016.

[5] A. D. Abreu, L. Toni, N. Thomos, T. Maugey, F. Pereira, and P. Frossard, "Optimal layered representation for adaptive interactive multiview video streaming," *Journal of Visual Communication and Image Representation*, vol. 33, pp. 255–264, 2015.

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

[7] C. Zhou, C. W. Lin, and Z. Guo, "mDASH: A markov decision-based rate adaptation approach for dynamic http streaming," *IEEE Trans. Multimedia*, vol. 18, no. 4, pp. 738–751, 2016.

[8] L.Toni, T. Maugey, and P. Frossard, "Optimized packet scheduling in multiview video navigation systems," *IEEE Trans. Multimedia*, vol. 17, no. 9, pp. 1604–1616, 2015.

[9] M. Pinson and S. Wolf, "A new standardized method for objectively measuring video quality," *IEEE Trans. Broadcast.*, vol. 50, no. 3, pp. 312–322, 2004.