IN-NETWORK VIEW RE-SAMPLING FOR INTERACTIVE FREE VIEWPOINT VIDEO STREAMING

Laura Toni[1]*, Gene Cheung[2], Pascal Frossard[1]

[1] Signal Processing Laboratory (LTS4), EPFL, Switzerland
[2] National Institute of Informatics, Japan

ABSTRACT

Interactive free viewpoint video offers the possibility for each user to independently choose the views of a 3D scene to be displayed at the decoder. The visual content is commonly represented by N texture and depth map pairs that capture different viewpoints. A server selects an appropriate subset of $M \leq N$ views for transmission, so that the user can freely navigate in the corresponding window of viewpoints without being affected by network delay. During navigation, a user can synthesize any intermediate virtual view image in the navigation window via depth-image-based rendering (DIBR) using two nearby camera views as references. When the available bandwidth is too small to transmit all camera views typically used to synthesize views in the navigation window, we propose to synthesize intermediate virtual views as new references for transmission—*a re-sampling of viewpoints for the 3D scene*—so that the synthesized view distortion within the navigation window is minimized. We formulate a combinatorial optimization problem to find the best set of M virtual views to synthesize as new references, and show that the problem is NP-hard. We approximate the original problem with a new reference view equivalence model and derive in this case an optimal dynamic programming algorithm to determine the best set of M views to be transmitted to each user. Experimental results show that synthesizing virtual views as new references for client-side view synthesis can outperform simple selection from camera views by up to 0.73dB in synthesized view quality.

Index Terms— multiview video, video streaming, network processing

1. INTRODUCTION

Interactive free viewpoint video applications endow users with the ability to choose and display any virtual viewpoint of a 3D scene that is originally represented by images captured by an array of N cameras. Specifically, a virtual image of viewpoint u can be synthesized at decoder via *depth-image-based rendering* (DIBR) [1] using texture and depth maps of left and right reference views v_L and v_R, where $v_L \leq u \leq v_R$. Many practical challenges in interactive free viewpoint systems have already been addressed in works such as [2, 3, 4, 5, 6, 7, 8, 9], which propose novel coding and/or streaming strategies at the server to optimize the tradeoff between transmission rate and visual distortion, while providing the promised view-navigation ability at the client. As an example, in order to deal with network delay, the system in [3] transmits a subset of camera views enabling synthesis of a window W of possible virtual views (with the window size proportional to the round trip time (RTT) between
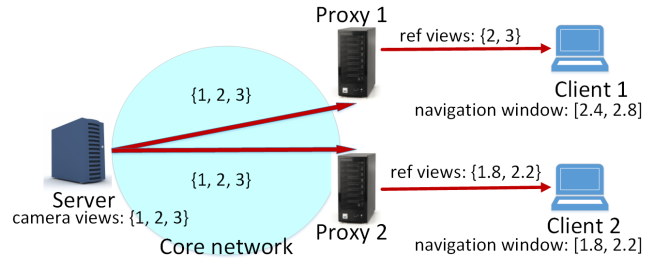

Fig. 1. Proxy architecture for interactive free viewpoint video, where proxies construct new reference views for client-side virtual view synthesis.

server and client), so that the client can freely navigate in the 3D scene. The interactive free viewpoint applications, however, rely on a system that is capable of delivering proper reference views to each client in real-time in order to enable free navigation, even if network resources are limited.

In this paper, we argue that, in the case where the available bandwidth is too small for the server to transmit all camera views conventionally used to synthesize virtual views in the navigation window, there exists a better strategy than simply transmitting a subset from N available camera views as reference for client-side view synthesis. Instead, *one can re-sample the viewpoints of the 3D scene* and synthesize intermediate virtual views as new references to be transmitted to the decoder, so that the synthesized view distortion within the navigation window is further minimized. In particular, we consider a distributed proxy architecture, where an interactive free viewpoint service is provided by a network of *proxies* at the edges of the core network, as shown in Fig. 1. A distant server has pre-encoded and currently stores a free viewpoint video in N camera viewpoints $\mathcal{V} = \{1, \ldots, N\}$. The feasible range $\mathcal{U} = \{1, 1 + \delta, \ldots, N\}$ of synthesized virtual views $u = k\,\delta$ for navigation is between leftmost and rightmost camera views 1 and N, where k is a positive integer and δ is a pre-determined fraction that describes the minimum view spacing between neighboring virtual views.

When a client initiates an interactive streaming session, the requested N-view video is first pre-fetched (if not locally cached) to a proxy that directly serves the clients. In general, service personalization at proxies[1] instead of server means a smaller RTT and hence a smaller navigation window W. With a RTT of T seconds between the proxy and the client, and a maximum speed ρ at which a user can navigate to neighboring virtual views, one can compute a *navigation window* $W(u) = [u - \rho T, u + \rho T]$, given that the user has selected

---
[1] Proxy-based service also means computation load at the centralized server is greatly relieved.

virtual view u at some time t_0. By transmitting reference views for synthesis of virtual views in $W(u)$ at time $t_0 + T/2$, the client can experience zero-delay view navigation at time $t_0 + T$ (see [3] for details). However, there often exists a bottleneck link between proxy and client[2], such that the proxy is unable to continuously transmit N camera views to the client. Hence, each proxy must select or construct $M < N$ reference views for transmission to the client depending on its interaction with the system.

We study the problem of selecting the optimal M reference views—ones that lead to the minimum distortion for all synthesized virtual views in window $W(u)$ subject to a bandwidth constraint—for the proxy to transmit. We formulate a combinatorial optimization problem to find this best set of M reference views, and subsequently show that it is unfortunately NP-hard. However, with help of a new *equivalent view distance* model—where synthesized view distortion using a distorted reference view is approximately the same as the synthesized distortion using an undistorted but further-away reference view—we can reformulate a simplified problem that becomes solvable optimally in polynomial time, using a computation-efficient dynamic programming (DP) algorithm. We test our algorithm in experiments where proxy-based virtual view synthesis outperforms simple proxy-based camera view forwarding by up to 0.73dB in PSNR. This confirms the importance of proper reference view selection in free viewpoint applications along with the benefit of in-network processing for interactive video systems.

The outline of the paper is as follows. We first provide an illustration of the reference view re-sampling problem and formalize the problem in Section 2. In Section 3, we derive our novel view re-sampling DP algorithm that is based on a new equivalent view distance model. Finally, experimental results and conclusion are provided in Section 4 and 5, respectively.

2. SYNTHETIC VIEW SELECTION PROBLEM

2.1. Illustrative example

As a simple illustration of our reference view selection problem, we consider that the server in Fig. 1 stores three coded camera views: $\{1, 2, 3\}$ and that the bottleneck links between proxy-client pairs can support transmission of only two views. We assume that a virtual view can have a non-integer index $i.x$, which corresponds to a position between camera views i and $i + 1$. If client 1 requests navigation window $[2.4, 2.8]$, the servicing proxy can simply forward the closest camera views 2 and 3. However, if client 2 requests navigation window $[1.8, 2.2]$, transmitting camera views 1 and 3 results in large synthesized view distortions due to the large distance between reference and virtual views (called *reference view distance* in the sequel). Instead, the proxy can synthesize virtual views 1.8 and 2.2 using camera views 1, 2, 3 and send them to the client 2 as new reference views for the virtual view window $[1.8, 2.2]$. Because of the smaller reference view distance, this novel strategy may result in smaller synthesized view distortion. However, the virtual view synthesis process introduces distortion into the new reference views 1.8 and 2.2. This has to be carefully considered when choosing which views to synthesize at the proxy.

2.2. Synthesized View Distortion

Since our goal is to minimize distortion of virtual views within a defined window W, we first sought to properly characterize the synthesized view distortion function. We first assume that the received

---

[2]In practice, the last-mile access network is often the bottleneck in real-time media distribution.

N camera views v from server at the proxy are distorted due to compression, which can be simply evaluated as $d(v)$. Further, a synthesized view u is distorted due to pixel rounding and disocclusion hole filling procedures [10, 11] during DIBR-based view synthesis [1] at proxies or clients. Specifically, we assume that the rendered image distortion $D_u(v_L, v_R, d(v_L), d(v_R))$ of virtual view $u \in \mathcal{U}$ synthesized using left and right reference views v_L and v_R, with respective distortion $d(v_L)$ and $d(v_R)$, depends on two factors: i) reference view distances $|v_L - u|$ and $|v_R - u|$, and ii) distortions of the reference views $d(v_L)$ and $d(v_R)$. Note that for v_L and v_R to be valid left and right reference views for synthesis of virtual view u, we require $v_L \leq u \leq v_R$.

In particular, we assume that the distortion $D_u()$ has the property of *monotonicity in reference view distance*, as done in [12]:

$$D_u(v_L, v_R, d_L, d_R) \leq D_u(v_L, v'_R, d_L, d_R), \quad v_R \leq v'_R$$
$$D_u(v_L, v_R, d_L, d_R) \leq D_u(v'_L, v_R, d_L, d_R), \quad v'_L \leq v_L \quad (1)$$

(1) states that, if the right (left) reference view moves further away from the virtual view u (and all other parameters are the same), then the synthesized view distortion cannot decrease. We can similarly define the property of *monotonicity in reference view distortion*: if the distortion in the right (left) reference view is larger, then the synthesized view distortion cannot be smaller.

2.3. Problem Formulation

We now formulate our view re-sampling problem to choose M optimal reference views. Among the possible views in the feasible range $\mathcal{U}$, a proxy selects the "optimal" subset $\mathcal{T} \subseteq \mathcal{U}$ for a client with virtual view window $W = [u_L^o, u_R^o]$, such that the bandwidth requirement is observed, *i.e.*, $|\mathcal{T}| \leq M$. By optimal, we mean a subset $\mathcal{T}$ that minimizes the distortion $\mathcal{D}(\mathcal{T})$ experienced over all virtual views in W:

$$\mathcal{D}_W(\mathcal{T}) = \sum_{u \in W} \min_{v_L, v_R \in \mathcal{T} | v_L \leq u \leq v_R} [D_u(v_L, v_R, d(v_L), d(v_R))]$$
$$(2)$$

where, for each virtual view u, the best pair of reference views in $\mathcal{T}$ that minimize synthesized distortion are selected. The parameter $d(v)$ is the compression distortion if v is a camera view; $d(v)$ is the distortion of a synthesized view if v is itself a virtual view synthesized at the proxy using camera views $\mathcal{V}$:

$$d(v) = \min_{v_L, v_R \in \mathcal{V} | v_L \leq v \leq v_R} [D_v(v_L, v_R, d(v_L), d(v_R))] \quad (3)$$

Hence the best pair for virtual view u might not always be the reference views that are the closest to u, since the reference views themselves might be distorted from view synthesis. Finally, the reference view selection problem can be formalized as a combinatorial optimization problem:

$$\min_{\mathcal{T} \subseteq \mathcal{U}} \mathcal{D}_W(\mathcal{T}) \quad \text{s.t. } |\mathcal{T}| \leq M \quad (4)$$

2.4. NP-Hardness of Optimization Problem

The optimization problem (4) is unfortunately a NP-hard problem, as it can be proved via a reduction from the well-known NP-hard problem *set cover* (SC) [13]. In SC, there is a universe $\mathcal{S}$ of items s, and a collection $\mathcal{C}$ of sets c, each covering a subset of items. The SC problem is to select a subset $\mathcal{C}' \subset \mathcal{C}$ of size no larger than K, so that each item $s \in \mathcal{S}$ is covered by at least one set $c \in \mathcal{C}'$.
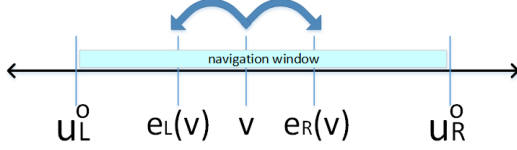
Fig. 2. Illustration of reference view equivalence: view v with distortion $d(v)$ is replaced by $e_L(v)$ ($e_R(v)$) with no distortion when used as a left (right) for synthesis of virtual view in window $W = [u_L^o, u_R^o]$.

Fig. 3. Illustration of possible reference view selections in optimal solution.

We briefly outline how our problem (4) includes SC as special case, one where only compression-distorted camera views $\mathcal{V}$ are selected as references. We map each item s to a virtual view u in navigation window W. There is only one camera view $v \in \mathcal{V}$ to the left of W that must be selected as left reference. We map each set c to a camera view $v \in \mathcal{V}$ to the right of W, and add a default right camera view v'. Serving as right reference, each camera view v can reduce the synthesized distortion by 1 (over default v') for the virtual view u if its corresponding item s is covered in the set c corresponding to v, and by 0 otherwise. If by selecting K right reference views the total distortion can be reduced by $|\mathcal{S}|$ over default v', then the selected views v corresponding to the desired sets c in SC.

3. SYNTHETIC VIEW SELECTION ALGORITHM

3.1. Reference View Equivalence

As the original problem (4) is NP-hard, we introduce an additional assumption called *reference view equivalence* on the synthesized view distortion function. It leads to a simplified problem—one that is optimally solvable in polynomial time. Specifically, we say that the synthesized view distortion $D_u()$, as function of left and right reference views v_L and v_R and reference view distortions $d(v_L)$ and $d(v_R)$, has the reference view equivalence property if it can be written in the following form:

$$
\begin{aligned}
D_u(v_L, v_R, d(v_L), d(v_R)) &= D_u(e_L(v_L), e_R(v_R), 0, 0) \\
&\stackrel{\text{def}}{=} D_u^0(e_L(v_L), e_R(v_R)) \quad (5)
\end{aligned}
$$

where $e_L(v_L) < v_L$ and $v_R < e_R(v_R)$. In words, the adverse effect of reference view distortions $d(v_L)$ and $d(v_R)$ on synthesized view quality is equivalent to an *increase* in reference view distances by $|e_L(v_L) - v_L|$ and $|v_R - e_R(v_R)|$, respectively. As an illustration, when synthesizing virtual views $u \in W = [u_L^o, u_R^o]$, in Fig. 2 the view v is replaced by $e_L(v)$ or $e_R(v)$ when used as left or right reference, respectively. For the sake of simplicity, we denote by $D_u^0()$ the synthesized view distortion $D_u()$ when reference views contain no distortion.

In practice, the empirical synthesized view distortion may not satisfy this property exactly. We thus approximate the equivalent reference view distances $e_L(v)$ and $e_R(v)$ for v as the values that minimize the square error over the portions of the navigation window $W = [u_L^o, u_R^o]$ when v can be used as left and right references:

$$
\begin{aligned}
e_R(v) &= \arg\min_{e|e>v} \sum_{u=u_L^o}^{\min\{v, u_R^o\}} \left[D_u(v_L^o, v, 0, d(v)) - D_u^0(v_L^o, e) \right]^2 \\
e_L(v) &= \arg\min_{e|e<v} \sum_{u=\max\{u_L^o, v\}}^{u_R^o} \left[D_u(v, v_R^o, d(v), 0) - D_u^0(e, v_R^o) \right]^2
\end{aligned}
\quad (6)
$$

where v_L^o and v_R^o are the nearest camera views to the left and right of W.
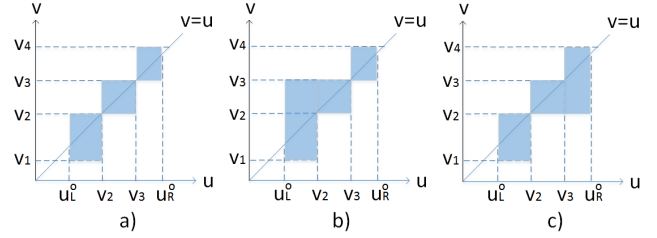
3.2. Optimization Algorithm

Under the assumption of reference view equivalence, we can now derive our optimal reference view selection algorithm through a series of lemmas. Without loss of generality, an optimal solution $\mathcal{T}^*$ to (4) results in a partitioning of window $W = [u_L^o, u_R^o]$ into a sequence of *virtual view segments* $\mathbf{s}^* = \{s_1^*, s_2^*, \ldots\}$, where virtual views u within each k-th segment $s_k^* = [u_k^*, u_{k+1}^*)$ use the same left and right reference views, $v_L(k), v_R(k) \in \mathcal{T}^*$, respectively, for view synthesis. The first lemma shows that if $v_R(k)$ is not the same as $v_R(k+1)$, then the optimality of $\mathbf{s}^*$ implies that setting $v_R(k)$ at the segment boundary $u_{k+1}^* - \delta$ is optimal:

Lemma 1 *For consecutive segments s_k^* and s_{k+1}^*, if right reference view of s_k^* is different from that of s_{k+1}^*, i.e., $v_R(k) \neq v_R(k+1)$, then $v_R(k) = u_{k+1}^* - \delta$ is optimal.*

Proof We prove by contradiction. Suppose $v_R(k) \neq v_R(k+1)$. $v_R(k) < u_{k+1}^* - \delta$ is not possible, since a valid right reference $v_R(k)$ for segment s_k^* means $v_R(k) \geq u_{k+1}^* - \delta$. Suppose instead $v_R(k) > u_{k+1}^* - \delta$, or equivalently $v_R(k) \geq u_{k+1}^*$, since δ is the minimum virtual view spacing. If the equivalent right reference view of $v_R(k)$ is larger than that of $v_R(k+1)$, *i.e.* $e_R(v_R(k)) > e_R(v_R(k+1))$, then having segment s_k^* selects $v_R(k+1)$ as right reference—valid since $v_R(k+1) \geq u_{k+2}^* - \delta > u_{k+1}^* - \delta$—will result in no larger total synthesized view distortion by monotonicity of reference view distance, hence $v_R(k) = v_R(k+1)$—a contradiction.

Suppose instead $e_R(v_R(k)) \leq e_R(v_R(k+1))$. If $v_R(k) \geq u_{k+2}^* - \delta$, then segment s_{k+1}^* can select right reference $v_R(k)$ with no increase in total distortion by monotonicity of reference view distance—a contradiction. If $v_R(k) < u_{k+2}^* - \delta$, then virtual views $\{u_k^*, \ldots, v_R(k)\}$ in s_{k+1}^* can be made into a new segment $s^\#$ with $v_R(k)$ as right reference view with no increase in total distortion. $s^\#$ has right reference $v_R(k)$ at segment boundary—a contradiction. $\square$

We now state the corresponding lemma for left reference views as well. The proof is similar and hence omitted.

Lemma 2 *For consecutive segments s_k^* and s_{k+1}^*, if left reference view of s_k^* is different from that of s_{k+1}^*, i.e., $v_L(k) \neq v_L(k+1)$, then $v_L(k+1) = u_{k+1}^*$ is optimal.*

The two lemmas effectively limit the search space of possible reference views we need to consider during optimization: from segment s_k^* to s_{k+1}^*, one of the following three cases must be true: i) s_k^* and s_{k+1}^* share the same left reference view, *i.e.*, $v_L(k) = v_L(k+1)$; ii) s_k^* and s_{k+1}^* share the same right reference view, *i.e.*, $v_R(k) = v_R(k+1)$; or iii) right reference view of s_k^* is the left reference view of s_{k+1}^*, *i.e.*, $v_R(k) = v_L(k+1)$. These three scenarios are illustrated in Fig. 3, where reference view v is plotted against virtual view u. In each case, four reference views $v_1, \ldots, v_4$ are selected.

In Fig. 3(b), segments $s_1^* = [u_L^o, v_1)$ and $s_2^* = [v_1, v_2)$ share a right reference view v_3. In Fig. 3(c), segments $s_2^* = [v_2, v_3)$ and $s_3^* = [v_3, u_R^o)$ share a left reference view v_2. There is no sharing of left / right reference views among segments in Fig. 3(a).

Given this corollary, we now derive a dynamic programming (DP) algorithm to find the best reference views as follows. Let $\Phi(u_L, v_L, k)$ be the minimum total synthesized view distortion for virtual view range $[u_L, u_R^o]$, given the best left reference view (in terms of equivalent reference view distance) left of u_L is v_L, and a budget of k references is available. $\Phi(u_L, v_L, k)$ can be defined recursively as follows for the $k \geq 1$ case[3]:

$$\min_{v > u_L} \left\{ \min \left[\begin{array}{l} \sum_{u=u_L}^{v} D_u^0(e_L(v_L), e_R(v)) + \Phi(v + \delta, \{v_L, v\}_L^*, k - 1), \\ \min_{0 \leq n \leq k-1} \Psi(u_L, v_L, v, n) + \Phi(v + \delta, v, k - n - 1) \end{array} \right] \right\} \quad (7)$$

where $\{v_L, v\}_L^* = \arg\max_{z \in \{v_L, v\}} e_L(z)$ returns the view with the closer left equivalent reference in the pair, and $\Psi(u_L, v_L, v_R, n)$ is the minimum total synthesized view distortion for virtual view range $[u_L, v_R]$, given the best left reference view left of u_L is v_L, and there is a budget of n for left reference views. It is defined as:

$$\min_{z | u_L < z < v_R} \left\{ \sum_{u=u_L}^{z} D_u^0(e_L(v_L), e_R(v_R)) + \Psi(z + \delta, z, v_R, n - 1) \right\} \quad (8)$$

In words, (7) states that $\Phi(u_L, v_L, k)$ is the minimum of two possible choices, for all possible right reference views v, $v > u_L$:

1. Total distortion from u_L to v using v_L and v as respective left and right references, *plus* a recursive cost starting from virtual view $v + \delta$ with best left reference to date $\{v_L, v\}_L^*$.

2. Total distortion from u_L to v using n different left reference views and v as right reference view, *plus* a recursive cost starting from virtual view $v + \delta$.

(8), accounting for right reference view sharing, states that $\Psi(u_L, v_L, v_R, n)$ is the sum of distortion for virtual view range $[u_L, z]$ using v_L and v_R as left and right reference views, *plus* the recursive distortion cost for the remaining range $[z + \delta, v_R]$ given v_R is *still* the right reference used, for all possible z.
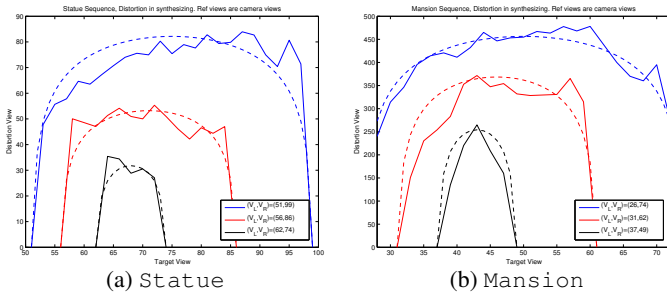
4. EXPERIMENTAL RESULTS

(a) Statue   (b) Mansion

Fig. 4. Comparison of empirical synthesized view distortion with our distortion model for sequences Statue and Mansion. Parameters (γ, ξ, β) used are $(9, 70, 4)$ and $(83, 300, 4)$, respectively.

We conducted experiments to show the performance gain by synthesizing reference views in the proxies selected by our proposed DP algorithm. We used the two sequences Stature and Mansion for simulation experiments, where 51 cameras captured respective 3D scenes with uniform spacing between the cameras: 5.33mm and

---
[3] $\Phi(u_L, v_L, k)$ when $k = 0$ is a simplified version of (7) and is omitted.

10mm for Statue and Mansion, respectively [14]. During optimization, we used the following model for distortion $D_u^o(v_L, v_R)$ of synthesized view u given left and right reference view v_L and v_R:

$$= \gamma \exp\left[\frac{v_R - v_L}{\xi}\right] \log\{1 + \beta[\alpha(v_R - u) + (1 - \alpha)(u - v_L)]\} \quad (9)$$

where $\alpha = (u - v_L)/(v_R - v_L)$ and γ, ξ, and β are parameters that depend on the particular video sequence. We can see in Fig. 4, where means square error (MSE) is plotted against synthesized view location for different pairs of left and right references, that there is a reasonably good fit between model (dotted lines) and empirical data (solid lines) for two texture-plus-depth image sequences Statue and Mansion. In the case when left and right reference views are distorted by d_L and d_R, we used

$$D_u(v_L, v_R, d_L, d_R) = D_u^0(v_L, v_R) + \zeta\left[(1 - \alpha)d_L + \alpha d_R\right] \quad (10)$$

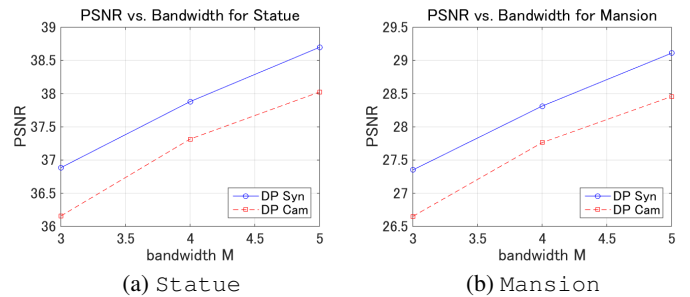where, from curve fitting, we set $\zeta = 0.8$ for our experiments.

(a) Statue   (b) Mansion

Fig. 5. Comparison of empirical synthesized view distortion with our distortion model for sequences Statue and Mansion. Parameters (γ, ξ, β) used are $(9, 70, 4)$ and $(83, 300, 4)$, respectively.

We now provide experimental results when $\mathcal{V} = \{1, 2, 3, \ldots, 7\}$, $\mathcal{U} = \{1, 1 + 1/8, \ldots, 7\}$, and the range of interest is $[U_L, U_R] = [1.75, 6.25]$. Results are provided in terms of mean PSNR of virtual views within the navigation window, given the selected M reference views. We compare our scheme of synthesizing M virtual views as new references (DP_Syn) with the case when only camera views $\mathcal{V}$ are available for selection, also using our DP algorithm (DP_Cam). We see in Fig. 5 that for both sequences, DP_Syn outperforms DP_Cam as the bandwidth constraint M ranges from 3 to 5. In particular, DP_Syn has a maximum of 0.73dB gain in PSNR over DP_Cam. The difference is largest when M is small, namely when the available bandwidth is small relative to the viewpoint samples within the navigation window. This is the situation where viewpoint re-sampling is most important.

5. CONCLUSION

During interactive free viewpoint video streaming when a server / proxy must reduce the number of reference views transmitted to a client due to bandwidth constraint, we argue that instead of simply choosing a subset of N pre-encoded camera views, one can synthesize intermediate virtual views as new references to further reduced client-side synthesized view distortions. Though in general the problem of finding the optimal set of reference views to synthesize is NP-hard, we show that using an equivalent view distance assumption, the problem becomes solvable in polynomial time via a dynamic programming algorithm. Experimental results show that synthesizing reference views can improve image quality by up to 0.73dB in PSNR and outline benefits of in-network processing in interactive applications.

6. REFERENCES

[1] D. Tian, P.-L. Lai, P. Lopez, and C. Gomila, "View synthesis techniques for 3D video," in *Applications of Digital Image Processing XXXII, Proceedings of the SPIE*, 2009, vol. 7443 (2009), pp. 74430T–74430T–11.

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

[3] 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.

[4] L. Toni, N. Thomos, and P. Frossard, "Interactive free viewpoint video streaming using prioritized network coding," in *Proc. IEEE Int. Workshop on Multimedia Signal Processing*, Sept 2013.

[5] 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.

[6] C.G. Gurler and M. Tekalp, "Peer-to-peer system design for adaptive 3D video streaming," *IEEE Commun. Mag.*, vol. 51, no. 5, pp. 108–114, May 2013.

[7] B. Hu, V. Zhao, and G. Cheung, "Incentive analysis for cooperative interactive multiview video streaming," in *EURASIP Signal Processing: Image Communication*, July 2014, vol. 29, no.6, pp. 641–666.

[8] T. Fujihashi, Z. Pan, and T. Watanabe, "UMSM: A traffic redution method on multi-view video streaming for multiple users," in *IEEE Transactions on Multimedia*, January 2014, vol. 16, no.1, pp. 228–241.

[9] D. Ren, G. Chan, G. Cheung, and P. Frossard, "Coding structure and replication optimization for interactive multiview video streaming," in *IEEE Transactions on Multimedia*, November 2014, vol. 16, no.7, pp. 1874–1887.

[10] Y. Mao, G. Cheung, and Y. Ji, "Image interpolation during DIBR view synthesis using graph Fourier transform," in *3DTV-Conference*, Budapest, Hungary, July 2014.

[11] S. Reel, G. Cheung, P. Wong, and L. Dooley, "Joint texture-depth pixel inpainting of disocclusion holes in virtual view synthesis," in *APSIPA ASC*, Kaohsiung, Taiwan, October 2013.

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

[13] M. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman, 1979.

[14] C. Kim, H. Zimmer, Y. Pritch, A. Sorkine-Hornung, and M. Gross, "Scene reconstruction from high spatio-angular resolution light fields," *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH)*, vol. 32, no. 4, pp. 73:1–73:12, 2013.