In-Network View Synthesis for Interactive Multiview Video Systems

Laura Toni Member, IEEE, Gene Cheung Senior Member, IEEE, and Pascal Frossard Senior Member, IEEE

Abstract—Interactive multiview video applications endow users with the freedom to navigate through neighboring viewpoints in a 3D scene. To enable such interactive navigation with a minimum view-switching delay, multiple camera views are sent to the users, which are used as reference images to synthesize additional virtual views via depth-image-based rendering. In practice, bandwidth constraints may however restrict the number of reference views sent to clients per time unit, which may in turn limit the quality of the synthesized viewpoints. We argue that the reference view selection should ideally be performed close to the clients, and we study the problem of in-network reference view synthesis such that the navigation quality is maximized at the clients. We consider a distributed cloud network architecture where data stored in a main cloud is delivered to end users with the help of cloudlets, i.e., resource-rich proxies close to the users. In order to satisfy last-hop bandwidth constraints from the cloudlet to the users, a cloudlet-re-samples viewpoints of the 3D scene into a discrete set of views (combination of received camera views and synthesized virtual views) to be used as reference for the synthesis of additional virtual views at the client. This in-network synthesis leads to better viewpoint sampling given a bandwidth constraint compared to simple selection of camera views, but it may however carry a distortion penalty in the cloudlet-synthesized reference views. We therefore cast a new reference view selection problem where the best subset of views is defined as the one minimizing the distortion over a view navigation window defined by the user under transmission bandwidth constraints. We show that the view selection problem is NP-hard, and propose an effective polynomial time algorithm using dynamic programming to solve the optimization problem under general assumptions that cover most of the multiview scenarios in practice. Simulation results confirm the performance gain offered by virtual view synthesis in the network. It shows that cloud computing resources provide important benefits in applications with limited resources.

Index Terms—Depth-image-based rendering, network processing, cloud-assisted applications, interactive systems.

I. INTRODUCTION

Interactive free viewpoint video systems [1] endow users with the ability to choose and display any virtual views of a 3D scene, given original viewpoint images captured by multiple cameras. In particular, a virtual view image can be synthesized by the decoder via depth-image-based rendering (DIBR) [2] using texture and depth images of two neighboring views that act as reference viewpoints. One of the key challenges in interactive multiview video streaming (IMVS) [3] is to transmit an appropriate subset of reference views from a potentially large number of camera-captured views, such that the client enjoys high quality and low delay view navigation even in resource-constrained environments [4]–[6].

In this paper, we propose a new paradigm to solve the reference view selection problem and capitalize on cloud computing resources to perform fine adaptation close to the clients. We consider a hierarchical cloud framework, where the selection of reference views is performed by a network of cloudlets, i.e., resource-rich proxies that can perform personalized processing at the edges of the core network [7], [8]. An adaptation at the cloudlets results in a smaller round-trip time (RTT), hence more reactivity than more centralized architectures. Specifically, we consider the scenario depicted in Fig. 1, where a main cloud stores pre-encoded video from different cameras, which are then transmitted to the edge cloudlets that act as proxies for final delivery to users. We assume that there is sufficient network capacity between the main cloud and the edge cloudlets for the transmission of all camera views, but there exists however a bottleneck of limited capacity between a cloudlet and a nearby user1. In this scenario, each cloudlet sends to a client the set of reference views that respect bandwidth capacities and enable synthesis of all viewpoints in the client’s navigation window. This window is defined as the range of viewpoints within which the user can freely navigate without any delay interacting with the cloudlet.

We argue that, in resource-constrained networks, resampling the viewpoints of the 3D scene in the network—i.e., synthesizing novel virtual views in the cloudlets that are transmitted as new references to the decoder—is beneficial compared to mere subsampling of the original set of camera views. We illustrate this in Fig. 1, where the main cloud stores three coded camera views: \{1, 2, 3\} while the bottleneck links

1In practice, the last-mile access network is often the bottleneck in real-time media distribution.
between cloudlet-user pairs can support the transmission of only two views. If user A requests a navigation window [2, 4, 2, 8], the cloudlet can simply forward the closest camera views 2 and 3. However, if user B requests the 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 cloudlet can synthesize virtual views 1.8 and 2.2 using camera views 1, 2, 3 and send these virtual views to the user B as new reference views for the navigation window [1, 8, 2, 2]. This strategy may result in smaller synthesized view distortion at the client due to the smaller distance to the reference views. However, the inter-network virtual view synthesis may also introduce distortion into the new reference views 1.8 and 2.2, which results in a tradeoff that should be carefully considered when choosing the views to be synthesized in the cloudlet.

Equipped with the above intuitions, we study the main tradeoff between reference distortion and bandwidth gain. Using a Gauss-Markov model, we first analyze the benefit of synthesizing new reference images in the network. We then formulate a new synthesized reference view selection optimization problem. It consists in selecting or constructing the optimal reference views that lead to the minimum distortion for all synthesized virtual views in the user’s navigation window subject to a bandwidth constraint between the cloudlet and the user. We show that this combinatorial problem can be solved optimally but is NP-hard. We then introduce a generic assumption on the view synthesis distortion which leads to a polynomial time solution with a dynamic programming (DP) algorithm. We then provide extensive simulation results for synthetic and natural sequences. They confirm the quality gain experienced by the IMVS clients when synthesis is allowed in the network, with respect to scenarios whose edge cloudlets can only transmit camera views. They also show that synthesis in the network allows to maintain good navigation quality when reducing the number of cameras as well as when cameras are not ideally positioned in the 3D scene. This is an important advantage in practical settings, which confirms that cloud processing resources can be judiciously used to improve the performance of applications that have limited network resources.

The remainder of this paper is organized as follows. Related works are described in Section II. In Section III, we provide a system overview and analyze the benefit of in-network view synthesis via a Gauss-Markov model to impart intuitions. The reference view selection problem is then formulated in Section IV. We propose general assumptions on view synthesis distortion in Section V and derive an additional polynomial time view selection algorithm. In Section VI, we discuss the simulation results, and we conclude in Section VII.

II. RELATED WORK

Prior studies addressed the problem of providing interactivity in IMVS, while saving on transmitted bandwidth and view-switching delay [3], [9]–[14]. These works mainly focused on optimizing the frame coding structure to improve interactive media services. In the case of pre-stored camera views, however, rather than optimal frame coding structures, interactivity in network-constrained scenario can be addressed by studying optimal camera selection strategies, where a subset of selected camera views is actually transmitted to clients such that the navigation quality is maximized and resource constraints are satisfied [4]–[6], [15]–[18]. In [18], a real-time multiview coding optimization is proposed for point-to-point interactive streaming. The coding scheme is based on a camera selection algorithm that predicts each user future requests by observing his/her head position. Rather than real-time coding, we mainly focus on pre-encoded video sequences with real-time processing at the cloudlets for on-demand interactive streaming. We note that although virtual view synthesis techniques have been studied intensively [19], [20], the view synthesis problem remains challenging, especially when the distance between cameras is large.

In [21], an optimal camera view selection algorithm in resource-constrained networks has been proposed based on the users’ navigation paths. In [22] a bit allocation algorithm over an optimal subset of camera views is proposed for optimizing the visual distortion of reconstructed views in interactive systems. Finally, in [23], [24] authors optimally organize camera views into layered subsets that are coded and delivered to clients in a prioritized fashion to accommodate for the network and clients heterogeneity. While in these works the selection is limited to camera views, we instead assume virtual view synthesis in the cloud network.

In-network adaptation strategies allow to cope with network resource constraints and are mainly categorized in i) packet-level processing and ii) modification of the source information. In the first category, packet filtering, routing strategies [25], [26] or caching of media content information [27] allow to save network resources while improving the quality experienced by clients. In the second category — in-network processing at the source level — the main objective is usually to process the source data in the network reducing both the communication volume and the processing required at the client side. Transcoding strategies might be collaboratively performed in peer-to-peer networks [28] or in the cloud [29]. Furthermore, source data can be compressed in the cloud [30], [31] to efficiently address users’ requests.

Rather than media processing in the main cloud, offloading resources to cloudlets might reduce the transmission latency [7], [8]. This is beneficial for delay-sensitive / interactive applications [32], [33]. Because of the proximity of cloudlets to users, cloudlet computing has been under intense investigation for cloud-gaming applications, as shown in [34] and references there in. The above works are mainly focused on multimedia processing, rather than on specific multiview scenarios. However, the use of cloudlets in delay sensitive applications motivates the idea of cloudlet-based view synthesis for IMVS. Cloud processing for multiview system is considered in [35]–[37]. In [37], view synthesis in the main cloud has been introduced to offload clients’ terminals. However, authors mainly address the problem of computational complexity at the

\footnote{We consider integer index \(i\) for any camera view, while we assume that a virtual view can have a non-integer index \(i.x\), which corresponds to a position between camera views \(i\) and \(i+1\).}
clients' terminal, knowing the view that clients will request. In our work, we exploit cloud computing resources to face bandwidth constraints in delay-sensitive applications, optimizing the quality navigation over a navigation window that takes into account the uncertainty of interactive clients. Also, to limit this uncertainty and still experience zero switching-delay in IMVS, we consider view synthesis at the edge cloudlets rather than at the main cloud. To the best of our knowledge, none of the work investigating cloud processing have considered the problem of multi-view interactive streaming under network resource constraints.

III. BACKGROUND

A. System Model

Let $\mathcal{V} = \{v_1, \ldots, v_N\}$ be the set of the $N$ camera viewpoints captured by the multiview system. For all camera-captured views, compressed texture and depth maps are stored at the main cloud, with each texture/depth pair encoded at the same rate using standard video coding tools like H.264 [38] or HEVC [39]. Since typically network limitations reside in the last-mile access network, we assume an abundant channel from the main cloud to the cloudlets, such that each cloudlet receives all camera-captured views$^3$. The possible viewpoints offered to the users are denoted by $\mathcal{U} = \{u_1, u_{1+1}/Q, \ldots, u_N\}$. The set $\mathcal{U}$ contains both synthesized views and camera views for navigation between the leftmost and rightmost camera views, $v_1$ and $v_N$. It is equivalent to offering views $u = k/Q$, where $k$ is a positive integer and $1/Q$ is a pre-determined fraction that describes the minimum view spacing between neighboring virtual views. We consider that any virtual viewpoint $u \in \mathcal{U}$ can be synthesized using a pair of left and right reference view images $v_L$ and $v_R$, $v_L < u < v_R$, via a known DIBR technique such as 3D warping [40]. View synthesis can be performed in-network (to generate new reference views) or at the user side (to warp) [40]. View synthesis can be performed in-network of the cloudlet is to serve the user with the best subset of the captured views in the window. Specifically, the goal is to synthesize all views in the window. Accordingly, the user can experience zero-delay view navigation at time $t_0 + T$ (see [13] for details) with optimized visual quality.

B. Analysis of Cloudulet-based Synthesized Reference View

To impart intuition on why synthesizing new references at in-network cloudlets may improve rendered view quality at an end user, we consider a simple model to capture correlations among neighboring views. Let $x_v$ be the signal on camera view $v$, modeled as a scalar random variable that represents all visual information acquired from camera view $v$. Similar to [41], [42], we assume that the information on neighboring views are correlated following a Gauss-Markov model; i.e., the variable $x_v$ is correlated with $x_{v-1}$ as follows:

$$x_v = x_{v-1} + e_v, \quad \forall v \geq 2$$

where $e_v$ is a zero-mean independent Gaussian variable with variance $\sigma_v^2$, and $x_1 = e_1$. A large $\sigma_v^2$ would mean views $v$ and $v_{v-1}$ are not similar. We can write $N$ variables $x_1, \ldots, x_N$ in matrix form:

$$Fx = e, \quad x = F^{-1}e$$

where

$$F = \begin{bmatrix}
1 & 0 & \ldots & 0 \\
-1 & 1 & 0 & \ldots \\
0 & -1 & 1 & 0 & \ldots \\
\vdots & \ddots & \ddots & \ddots \\
0 & \ldots & 0 & -1 & 1
\end{bmatrix},$$

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix}, \quad e = \begin{bmatrix} e_1 \\ \vdots \\ e_N \end{bmatrix}$$

Given $x$ is zero-mean, the covariance matrix $C$ can be computed as:

$$C = E[xx^T] = F^{-1}E[ee^T](F^{-1})^T$$

where $E[ee^T] = \text{diag}(\sigma_1^2, \ldots, \sigma_N^2)$ is a diagonal matrix. The precision matrix $Q$ is the inverse of $C$ and can be derived as follows:

$$Q = C^{-1} = (F^{-1} \text{diag}(\sigma_1^2, \ldots, \sigma_N^2) (F^{-1})^T)^{-1}$$

$$= F^T \text{diag}(\sigma_1^2, \ldots, \sigma_N^2) F^{-1}$$

which is a tridiagonal matrix.

When synthesizing a view $x_u$ using its neighbors $x_{u-1}$ and $x_{u+1}$, we would like to know the resulting precision. Without loss of generality, we write $x$ as a concatenation of two sets of variables, i.e. $x = [y \; z]$. For example, $y$ can be the target synthesized view and $z$ can be its reference views used for synthesis. It can be shown [43] that the conditional mean and precision matrix of $y$ given $z$ are:

$$m_{y|z} = m_y - Q_{yz}^*Q_{zy}(z - m_z)$$

$$Q_{y|z} = Q_{yy}$$

Consider now a set of four views $x_1, x_2, x_3, x_4$, where $x_1, x_2, x_3$ are camera views transmitted from the main cloud. Suppose further that the user window is $[1.8, 2.2]$, and that

$^3$Our framework can be easily extended to other network settings as well. For example, in the case of more constrained network resources between the main cloud and the cloudlets, only a subset of the captured views will be forwarded to the cloudlet, and the optimization problem is adapted accordingly.
only two reference views can be sent to clients because of bandwidth limitation. The cloudlet has to choose between using received \( x_4 \) as right reference, or synthesizing new reference \( x_3 \) using received \( x_2 \) and \( x_4 \). Using the discussed Gauss-Markov model (1) and the conditionals (5), we see that synthesizing \( x_3 \) using reference \( x_2 \) and \( x_4 \) results in precision:

\[
Q_{3\mid(2,4)} = Q_{33} = \frac{1}{\sigma_{3}^2} + \frac{1}{\sigma_{4}^2}
\]  

(6)

Taking the inverse of the right-hand side, to obtain the variance of \( x_3 \) of synthesized \( x_2 \) and \( x_4 \):

\[
1/Q_{33} \text{ is thus the additional noise variance when using new reference } x_3 \text{ to synthesize } x_2. \text{ We can then compute the conditional precision } Q_{2\mid(1,3)} \text{ given new reference } x_3:
\]

\[
Q_{2\mid(1,3)} = \frac{1}{\sigma_{2}^2} + \frac{1}{\sigma_{3}^2 + \frac{1}{\sigma_{4}^2}}
\]  

(7)

In comparison, if a user uses received \( x_4 \) as right reference, \( x_4 \) will accumulate two noise terms from \( x_2 \) to \( x_4 \):

\[
x_4 = x_2 + e_3 + e_4
\]  

(8)

The resulting conditional precision of \( x_2 \) given \( x_1 \) and \( x_4 \) is:

\[
Q_{2\mid(1,4)} = \frac{1}{\sigma_{2}^2} + \frac{1}{\sigma_{3}^2 + \sigma_{4}^2}
\]  

(9)

We now compare \( Q_{2\mid(1,3)} \) in (7) with \( Q_{2\mid(1,4)} \) in (9). We see that if \( \sigma_{3}^2 \) is very large relative to \( \sigma_{4}^2 \), then \( \left( \frac{1}{\sigma_{3}^2} + \frac{1}{\sigma_{4}^2} \right)^{-1} \approx \sigma_{4}^2 \), and \( Q_{2\mid(1,3)} \approx Q_{2\mid(1,4)} \). That means that in general it is worthwhile to synthesize new reference \( x_3 \). The reason can be interpreted from the derivation above: by synthesizing \( x_3 \) using both \( x_2 \) and \( x_4 \), the uncertainty (variance) for the right reference has been reduced from \( \sigma_{4}^2 \) to \( \left( \frac{1}{\sigma_{3}^2} + \frac{1}{\sigma_{4}^2} \right)^{-1} \), improving the precision of the subsequent view synthesis.

Finally, we note that for a wide baseline camera setup where the physical distance between each pair of neighboring reference cameras is large, the noise variance \( \sigma_{i}^2 \) is large for all \( i \). That means the precisions in (7) and (9) are both small. Thus while synthesizing a new reference view, network would still help according to our argument above, in general the synthesized view quality is poor, and the additional gain may be marginal.

IV. REFERENCE VIEW SELECTION PROBLEM

We now formalize the NP-hard synthesized reference view selection problem and introduce an assumption on the distortion of synthesized viewpoints.

A. Problem Formulation

Interactive view navigation means that a user can construct any virtual view \( u \) within a specified navigation window with zero view-switching delay, using viewpoint images transmitted from the main cloud as reference [13]. We denote this navigation window by \( [U_{L}^{0}, U_{R}^{0}] \) that depends on the user’s current observed viewpoint. If bandwidth is not a concern, for best synthesized view quality the edge cloudlet would send to the user all camera-captured views in \( V \) as reference to synthesize virtual view \( u \), \( \forall u \in [U_{L}^{0}, U_{R}^{0}] \). When this is not feasible due to limited bandwidth \( C \) between the serving cloudlet and the user, only a subset of views \( T \) is sent as set of reference views. At the client, the navigation window reconstructed from views in \( T \) leads to an aggregate distortion \( D(T) \) evaluated as

\[
D(T) = \sum_{u \in [U_{L}^{0}, U_{R}^{0}]} \min_{v \in \{L, R\}} \{ d_{u}(v_{L}, v_{R}, D(v_{L}), D(v_{R})) \}
\]

(10)

where \( D(v) \) is the distortion of viewpoint image \( v \), due to lossy compression for a camera-captured view, or due to DIBR synthesis for a virtual view, and \( d_{u}(v_{L}, v_{R}, D(v_{L}), D(v_{R})) \) is the distortion of the virtual view \( u \) synthesized using left and right reference views \( v_{L} \) and \( v_{R} \) with distortions \( D(v_{L}) \) and \( D(v_{R}) \), respectively.

In the case of limited network-resources, among all subsets \( T \subset U \) of synthesized and camera-captured views that satisfy the bandwidth constraint, the cloudlet must select the best subset \( T^{*} \) that minimizes the aggregate distortion \( D(T) \) of all virtual views \( u \in [U_{L}^{0}, U_{R}^{0}] \), i.e.,

\[
T^{*} : \arg \min_{T} D(T) \quad \text{s.t.} \quad |T| \leq C \quad \text{and} \quad T \subseteq U
\]

(11)

We note that (11) differs from existing reference view selection formulations [16], [17], [24] in that the cloudlet has the extra degree of freedom to synthesize virtual view(s) as new reference(s) for transmission to the user. In (10), for each virtual view \( u \) the best reference pair in \( T \) is selected for synthesis. Note that, unlike [16], the best reference pair may not be the closest references, since the quality of synthesized \( u \) depends not only on the view distance between the synthesized and reference views, but also on the distortions of the references.

B. Shared optimality of reference views assumption

We consider first an assumption on the synthesized view distortion \( d_{u}() \) called the shared optimality of reference views:

\[
\text{if } d_{u}(v_{L}, v_{R}, D(v_{L}), D(v_{R})) \leq d_{u}(v'_{L}, v'_{R}, D(v'_{L}), D(v'_{R})) \text{ then } d_{u'}(v_{L}, v_{R}, D(v_{L}), D(v_{R})) \leq d_{u'}(v'_{L}, v'_{R}, D(v'_{L}), D(v'_{R}))
\]

(12)

for max\{\( v_{L}, v'_{L} \)\} \leq u, u' \leq \min\{v_{R}, v'_{R}\}. In words, this assumption (12) states that if the virtual view \( u \) is better synthesized using the reference pair \( (v_{L}, v_{R}) \) than \( (v'_{L}, v'_{R}) \), then another virtual view \( u' \) is also better synthesized using \( (v_{L}, v_{R}) \) than \( (v'_{L}, v'_{R}) \). This means that if \( (v_{L}, v_{R}) \) is the best reference pair for a view \( u \), then \( (v_{L}, v_{R}) \) is also the best reference pair for view \( u' \in [v_{L} + 1/Q, v_{R} - 1/Q] \). In the following we show that the assumption holds in most 3D scenes.
Motivation on the shared optimality of reference views assumption: We see intuitively that this assumption is reasonable for smooth 3D scenes; a virtual view \( u \) tends to be similar to its neighbor \( u' \), so a good reference pair \((v_L, v_R)\) for \( u \) should also be good for \( u' \). We can also argue for the plausibility of this assumption as a consequence of two functional trends in the synthesized view distortion \( d_u() \) that are observed empirically to be true generally. For simplicity, consider for now the case where the reference views \( v_L, v_R, v_L', v_R' \) have zero distortion, i.e. \( D(v_L) = D(v_R) = D(v_L') = D(v_R') = 0 \). The first trend is the monotonicity in predictor’s distance [22]; i.e., the further-away are the reference views to the target synthesized view, the worse is the resulting synthesized view distortion. This trend has been successively exploited for efficient bit allocation algorithms [22], [44]. In our scenario, this trend implies that reference pair \((v_L, v_R)\) is better than \((v_L', v_R')\) at synthesizing view \( u \) if the pair is closer to \( u \), i.e.

\[
|u - v_L| + |v_R - u| \leq |u - v_L'| + |v_R' - u|
\]

(13)

where \( \max\{v_L, v_L'\} < u < \min\{v_R, v_R'\} \).

It is easy to see that if reference pair \((v_L, v_R)\) is closer to \( u \) than \((v_L', v_R')\), it is also closer to \( u' \), thus better at synthesizing \( u' \). Without loss of generality, we write new virtual view \( u' \) as \( u' = u + \delta \). We can then write:

\[
|(u + \delta) - v_L| + |v_R - (u + \delta)| = |u - v_L + v_R - u|
\]

\[
\leq |u - v_L' + v_R' - u|
\]

\[
\leq |(u + \delta) - v_L'| + |v_R' - (u + \delta)|
\]

(14)

where \( \max\{v_L, v_L'\} < u' \leq \min\{v_R, v_R'\} \).

Consider now the case where the reference views \( v_L, v_R, v_L', v_R' \) have non-zero distortions. In [45], another functional trend is empirically demonstrated, where a reference view \( v_L \) with distortion \( D(v_L) \) was well approximated as a further-away equivalent reference view \( v_L^\# < v_L \) with no distortion \( D(v_L^\#) = 0 \). Thus a better reference pair \((v_L, v_R)\) than \((v_L', v_R')\) at synthesizing \( u \) just means that the equivalent reference pair for \((v_L, v_R)\) is closer to \( u \) than the equivalent reference pair for \((v_L', v_R')\). Using the same previous argument, we see that the equivalent reference pair for \((v_L, v_R)\) are also closer to \( u' \) than \((v_L', v_R')\), resulting in a smaller synthesized distortion. Hence, we can conclude that the assumption of shared optimality of reference views is a consequence of these two functional trends.

We can graphically illustrate possible solutions to the optimization problem (11) under the assumption of shared optimality of reference views. Fig. 2(a) depicts the selected reference views for virtual views in the navigation window. In the figure, the \( x \)-axis represents the virtual views in the window \([U_L^0, U_R^0]\) that require synthesis. Correspondingly, on the \( y \)-axis, we have two piecewise constant (PWC) functions representing the left and right reference views selected to synthesize each virtual view \( u \) in the window, assuming that for each \( u \in [U_L^0, U_R^0] \) there must be one selected reference pair \((v_L, v_R)\) such that \( v_L \leq u \leq v_R \). A constant line segment—e.g., \( v = v_1 \) for \( U_L^0 \leq u \leq v_3 \) in Fig. 2(a)—means that the same reference is used for a range of virtual views. This graphical representation results in two PWC functions—left and right reference views—above and below the \( u = v \) line. The set of selected reference views are the unions of the constant step locations in the two PWC functions.

Under the assumption of shared reference optimality, we see that the selected reference views in Fig. 2(a) cannot be an optimal solution. Specifically, virtual views \( v_3 - 1/Q \) and \( v_3 \) employ references \([v_1, v_4]\) and \([v_2, v_5]\) respectively. However, if references \([v_1, v_4]\) are better than \([v_2, v_5]\) for virtual view \( v_3 - 1/Q \), they should be better for virtual view \( v_3 \) according to shared reference optimality in (12). An example of an optimal solution candidate under the assumption of shared reference optimality is shown in Fig. 2(b).

The shared reference optimality assumption thus reduces the number of reference sets that are candidate solutions of the reference views selection problem in (11), i.e., it reduces the search space in the optimization. However, even under this assumption the problem remains NP-hard, as proven in the Appendix. In the following section, we introduce a second assumption on the distortion function that allows us to solve the reference view selection problem in polynomial time.

V. OPTIMAL VIEW SELECTION ALGORITHM

We now introduce an additional assumption on the synthesized distortion function that holds in most common 3D scenes. Then we detail the DP algorithm to solve (11) and analyze the DP algorithm’s computation complexity.

A. Independence of reference optimality assumption

The second assumption on the synthesized view distortion \( d_u() \) is the independence of reference optimality, which states that the selection of the best left (right) reference view for viewpoint \( u \) does not depend on the selection of the right (left) reference view used during the synthesis. More formally:

if \( d_u(v_L, v_R, D(v_L), D(v_R)) \leq d_u(v'_L, v_R, D(v'_L), D(v_R)) \) then \( d_u(v_L, v'_R, D(v_L), D(v'_R)) \leq d_u(v'_L, v'_R, D(v'_L), D(v'_R)) \)

(15)

for \( \max\{v_L, v'_L\} \leq u \leq \min\{v_R, v'_R\} \). In words, the assumption (15) states that if \( v_L \) is a better left reference than \( v'_L \)
\[ \Lambda(v_1, v_2) = \begin{cases} v_1 & \text{if } d_u(v_1, v_R, D(v_1), D(v_R)) \leq d_u(v_2, v_R, D(v_2), D(v_R)) \\ v_2 & \text{otherwise} \end{cases} \]

\[ \Psi(u_L, v_L, v_R, n) = \begin{cases} \min_{v > u_L} \sum_{u = u_L}^{v - \frac{1}{2}} d_u(v_L, v_R, D(v_L), D(v_R)) + \Psi(v, v_R, n - 1) & \text{if } n \geq 1 \\ \sum_{u = u_L}^{v_R - \frac{1}{2}} d_u(v_L, v_R, D(v_L), D(v_R)) & \text{o.w.} \end{cases} \]

Fig. 3. Reference view assignments in (a) and (b) are optimal solution candidates under both assumptions. We name these two cases “shared-left” and “shared-right”, respectively.

Motivations on the independence of reference optimality assumption: This assumption essentially states that contributions towards the synthesized image from the two references are independent from each other, which is reasonable since each rendered pixel in the synthesized view is typically copied from one of the two references, but not both. We can also argue for the plausibility of this assumption as a consequence of the two aforementioned functional trends in the synthesized view distortion \(d_u(.)\) in Section IV. Consider first the case where the reference views \(v_L, v_R, v_L', v_R'\) have zero distortion. The monotonicity in predictor’s distance in (13) for a common right reference view becomes

\[ |u - v_L| + |v_R - u| \leq |u - v_L'| + |v_R - u| \rightarrow |u - v_L| \leq |u - v_L'| \] (16)

where \(\max\{v_L, v_L'\} < u < v_R\). Thus if \(v_L\) is preferred to \(v_L'\) for \(v_R > u\), it will hold also for \(v_R'\) as long as \(v_R' > u\). Consider now the case where the reference views \(v_L, v_R, v_L', v_R'\) have non-zero distortions. Introducing the equivalent reference views \(v_1^e, v_2^e\) with no distortion \(D(u_L^e) = 0\), the same argument of (16) holds for the equivalent reference views, leading to

\[ |u - v_L^e| \leq |u - v_L'|, \forall v_R > u. \]

We illustrate different optimal solution candidates to (11) now under both virtual view distortion assumptions to impart intuition. We see that the assumption of independence of reference optimality would prevent the reference view selection in Fig. 2(b) from being an optimal solution. Specifically, we see that both \(v_3\) and \(v_4\) are feasible right reference views for virtual views \(v_2 = 1/Q\) and \(v_2\). Regardless of which left references are selected for these two virtual views, if \(v_3\) is a strictly better right reference than \(v_4\), then having both virtual views select \(v_3\) as right reference will result in a lower overall distortion (and vice versa). If \(v_3\) and \(v_4\) are equally good right reference views resulting in the same synthesized view distortion, then selecting just \(v_4\) without \(v_3\) can achieve the same distortion with one fewer right reference view. Thus the selected reference views in Fig. 2(b) cannot be optimal.

We can thus make the following observation: as virtual view \(u\) increases, an optimal solution cannot switch right reference view from current \(v_R\) earlier than \(u = v_R\). Conversely, as virtual view \(u\) decreases, an optimal solution cannot switch left reference view from current \(v_L\) earlier than \(u = v_L - 1/Q\). As examples, Fig. 3 provides solutions of left and right reference views for virtual views in the navigation window. In the figure, on the \(x\)-axis are the virtual views \(u\) in the window \([U_L^0, U_R^0]\) that require synthesis. Correspondingly, on the \(y\)-axis are the left and right reference views (blue and green PWC functions respectively) selected to synthesize each virtual view \(u\) in the window.

We see that the reference view selections in Fig. 3(a) and Fig. 3(b) are optimal solution candidates to (11). Thus, the optimal reference view selections must be graphically composed of “staircase” virtual view ranges as shown in Fig. 3(a) and Fig. 3(b). In other words, either a shared left reference view \(v_L^s\) is used for multiple virtual view ranges \([u_i, u_{i+1}]\) where each range has the same \(v_L^s\) as left reference ("shared-left" case), or a shared right reference view \(v_R^s\) is used for multiple ranges \([u_i, u_{i+1}]\), where each range has \(v_R^s\) as its right reference ("shared-right" case). This motivates us to design an efficient DP algorithm to solve (11) optimally in polynomial time.

B. DP Algorithm

We first define a recursive function \(\Phi(u_L, v_L, k)\) as the minimum aggregate synthesized view distortion of views between \(u_L\) and \(U_R^n\), given \(v_L\) is the selected left reference view for synthesizing view \(u_L\), and there is a budget of \(k\) additional reference views. To analyse \(\Phi(u_L, v_L, k)\), we consider the two “staircase” cases identified by Fig. 3(a) and Fig. 3(b) separately, and show how \(\Phi(u_L, v_L, k)\) can be evaluated in each of the cases.

Consider first the “shared-left” case (Fig. 3(a)) where a shared left reference view is employed in a sequence of virtual view ranges. A view range represents a contiguous range of virtual viewpoints that employ the same left and
right reference views. The algorithm selects a new right reference view \( v, v > u_L \), creating a new range of virtual views \([u_L, v)\), as depicted in Fig. 4(a). Virtual views in range \([u_L, v)\) are synthesized using a shared left reference \( v_L \) and the newly selected reference view \( v \), resulting in distortion \( d_u(v_L, v, D(v_L), D(v)) \) for each virtual view \( u, u_L \leq u < v \). The aggregate distortion function \( \Phi(u_L, v_L, v, k) \) for this case is the distortion of views in \([u_L, v)\) plus a recursive term \( \Phi(v, \Lambda(v_L, v), k-1) \) to account for aggregate synthesized view distortions to the right of \( v \):

\[
\Phi(u_L, v_L, v, k) = \sum_{u=u_L}^{v-1} d_u(v_L, v, D(v_L), D(v)) + \Phi(v, \Lambda(u_L, v_L), k-1)
\]

(17)

where \( k-1 \) is the remaining budget of additional reference views, and \( \Lambda(v_1, v_2) \) chooses the better of the two left reference views, \( v_1 \) and \( v_2 \), for the recursive function \( \Phi(v) \). Formally, the left reference selection function \( \Lambda(v_1, v_2) \) is defined in (18). Given our two assumptions, we know that the selected left reference \( \Lambda(v_1, v_2) \) remains better for all other virtual views \( u \) in \([\max\{v_1, v_2\}, v]\).

We now consider the “shared-right” case (Fig. 3(b)) where a newly selected view \( v \) is actually a common right reference view for a sequence of virtual ranges from \( u_L \) to \( v \). This means that viewpoints in the range \([u_L, v)\) share the right reference view \( v \), but they may select left reference view(s) that differ from \( v_L \), since they do not necessarily share also the left reference view. It follows that other reference views can be selected in the range \([u_L + 1/Q, v - 1/Q]\), as depicted in Fig. 4(b). To select these remaining reference views, we define a companion recursive function \( \Psi(u_L, v_L, v_R, n) \) that returns the minimum aggregate synthesized view distortion from view \( u_L \) to \( v_R \), given that \( v_L \) is the selected left reference view, \( v_R \) is the common right reference view, and there is a budget of \( n \) other left reference views in addition to \( v_L \). We can write \( \Psi(u_L, v_L, v_R, n) \) recursively resulting in (19). In more details, (19) states that \( \Psi(u_L, v_L, v_R, n) \) is the synthesized view distortion of views in the range \([u_L, v)\), plus the recursive distortion \( \Psi(v, v, v_R, n-1) \) from view \( v \) to \( v_R \) with a reduced reference view budget \( n-1 \).

We can now put the two cases together into a complete definition of \( \Phi(u_L, v_L, k) \), resulting in (20). The equation (20) states that \( \Phi(u_L, v_L, k) \) examines each candidate reference view \( v, v > v_L \), which can be used either as right reference for synthesizing virtual views in \([u_L, v)\) with left reference \( v_L \) (“shared-left” case), or as a common right reference for a sequence of \( n+1 \) virtual view ranges within the interval \([u_L, v)\) (“shared-right” case).

When the remaining view budget is \( k = 1 \), in (20) \( \Phi(u_L, v_L, 1) \) simply selects a right reference view \( v, v \geq U_R^n \), which minimizes the aggregate synthesized view distortion for the range \([u_L, U_R^n]\):

\[
\Phi(u_L, v_L, 1) = \min_{v \leq U_R^n} \sum_{u=u_L}^{v-1} d_u(v_L, v, D(v_L), D(v))
\]

(21)

Having defined \( \Phi(u_L, v_L, k) \), we can identify the best \( C \) reference views by calling \( \Phi(U_R^n, v, C) \) repeatedly to identify the best leftmost reference view \( v, v \leq U_R^n \), and start the selection of the \( K-1 \) remaining reference views as follows

\[
\min_{v \leq U_R^n} \Phi(U_L^n, v, C - 1)
\]

(22)

C. Computation Complexity

Our proposed DP algorithm requires two different tables to be stored. The first time \( \Psi(u_L, v_L, v_R, n) \) is computed, the result can be stored in entry \([(u_L - U_L^n)/Q][(v_L - U_L^n)/Q][(v_R - U_R^n)/Q][n] \) of a DP table \( \Psi^* \), so that subsequent calls with the same arguments can be simply looked up. Analogously, the first time \( \Phi(u_L, v_L, k) \) is called, the computed value is stored in entry \([(u_L - U_L^n)/Q][(v_L - U_L^n)/Q]/k \) of another DP table \( \Phi^* \) to avoid repeated computation in future recursive calls.

We bound the computation complexity of our proposed algorithm (20) by computing a bound on the sizes of the required DP tables and the cost in computing each table entry. For notation convenience, let the number of reference views and synthesized views be \( S_n = (V - 1)/Q \) and \( S_u = (U_R^n - U_L^n)/Q \), respectively. The size of DP table \( \Phi^* \) is no larger than \( S_u \times S_v \times C \). The cost of computing an entry in \( \Phi^* \) using (20) over all possible reference views \( v \) involves the computation of the “shared-left” case with complexity \( O(S_u) \) and the one of the “shared-right” case with complexity \( O(C) \). Thus, each table entry has complexity \( O(S_u + S_v + K) \). Hence the complexity of completing the DP table \( \Phi^* \) is \( O(S_n^2 S_v^2 + S_vS_C^2) \). Given that in typical setting \( S_u \gg C \), the complexity for computing DT table \( \Phi^* \) is thus \( O(S_n^2 S_v^2 C) \).

We can perform similar procedure to estimate the complexity in computing DP table \( \Psi^* \). The size of the table in this case is upper-bounded by \( S_n \times S_v \times S_v \times C \). The complexity in computing each entry is \( O(S_v) \). Thus the complexity of computing DP table \( \Psi^* \) is \( O(S_n^2 S_v^2 C) \), which is the same as the \( \Phi^* \). Thus the overall computation complexity of our solution in (20) is also \( O(S_n^2 S_v^2 C) \).
\[ \Phi(u_L, v_L, k) = \min_{v > v_L} \left\{ \min \left[ \sum_{v = u_L}^{v - \frac{1}{2}} d_u(u_L, v, D(v_L), D(v)) + \Phi(v, \Lambda(v_L), v, k - 1), \min_{1 \leq n \leq k-1} \Psi(u_L, v_L, v, n) + \Phi(v, v, k - n - 1) \right] \right\} \]

(20)

TABLE I

<table>
<thead>
<tr>
<th>Viewpoints notation.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Camera ID as in [46], &quot;Bikes&quot;</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>Camera ID as in [46], &quot;Mansion&quot; and &quot;Church&quot;</td>
</tr>
<tr>
<td>25</td>
</tr>
<tr>
<td>Camera ID as in [46], &quot;Statue&quot;</td>
</tr>
<tr>
<td>50</td>
</tr>
<tr>
<td>Camera ID in our work</td>
</tr>
<tr>
<td>0</td>
</tr>
</tbody>
</table>

VI. SIMULATION RESULTS

A. Settings

We study the performance of our algorithm and we show the distortion gains offered by cloudlets-based virtual view synthesis. For a given navigation window \( [U_L^I, U_R^I] \), we provide the average quality at which viewpoints in the navigation window is synthesized. This means that we evaluate the average distortion of the navigation window as \( \frac{1}{N} \sum_{u = U_L^I}^{U_R^I} d_u \), with \( N \) being the number of synthesized viewpoints in the navigation window, and we then compute the corresponding PSNR. In our algorithm, we have considered the following model for the distortion of the synthesized viewpoint \( u \) from reference views \( V_L, V_R \)

\[
d_u(V_L, V_R, D_L, D_R) = \alpha D_{\text{min}} + (1 - \alpha) \beta D_{\text{max}} + \left[ 1 - \alpha - (1 - \alpha) \beta \right] D_I \tag{23}\]

where \( D_{\text{min}} = \min\{D_L, D_R\} \), \( D_{\text{max}} = \max\{D_L, D_R\} \), \( D_I \) is the inpainted distortion, and \( \alpha = \exp(-\gamma|u - V_{\text{min}}|/d) \), \( \beta = \exp(-\gamma|u - V_{\text{max}}|/d) \) with \( d \) is the distance between two consecutive camera views \( v_i \) and \( v_{i+1} \), \( V_{\text{min}} = V_I \) if \( D_L \leq D_R \), \( V_{\text{min}} = V_R \) otherwise, and \( V_{\text{max}} = V_I \) if \( D_L > D_R \), \( V_{\text{max}} = V_R \). The model can be explained as follows. A virtual synthesis \( u \), when reconstructed from \( (V_L, V_R) \) has a relative portion \( \alpha \in [0, 1] \) that is reconstructed at a distortion \( D_{\text{min}} \), from the dominant reference view, defined as the one with minimum distortion. The remaining portion of the image, i.e., \( 1 - \alpha \), is either reconstructed by the non-dominant reference view for a portion \( \beta \), at a distortion \( D_{\text{max}} \), or it is inpainted, at a distortion \( D_I \). The parameter \( \gamma \) and \( D_I \) depend on the scene geometry. While \( \gamma \) describes the reduction of the camera view correlation with the inter-camera distance, \( D_I \) corresponds to the inpainted distortion and depends on both the scene and the inpainting strategy.

The results have been carried out using 3D sequences “Mansion”, “Statue”, “Bikes” and “Church” [46], where 51 cameras acquire the scene with uniform spacing between the camera positions. The spacing between camera positions is 5.33 mm for “Statue”, 5 mm for “Bikes”, and 10 mm for “Mansion” and “Church”. Among all camera views provided for both sequences, only a subset represents the set of camera views \( V \) available at the cloudlet, while the remaining are virtual views to be synthesized. Table I depicts how the camera notation used in [46] is adapted to our notation. Finally, in Table II we provide the per-sequence parameters adopted in the theoretical model in (23) and derived by curve fitting.

In the following, we compare the performance achieved by virtual view synthesis in the cloudlets with respect to the scenario in which cloudlets only send to users a subset of camera views. We denote by \( T \) the subset of selected reference.
views when synthesis is allowed in the network, and by $T_{ns}$ the subset of selected reference views when only camera views can be sent as reference views, i.e., when synthesis is not allowed in the network. For both the cases of network synthesis and no network synthesis, the best subset of reference views is evaluated both with the proposed view selection algorithm and with an exact solution, i.e., an exhaustive search of all possible combinations of reference views. For the proposed algorithm, the distortion is evaluated both with experimental computation of the distortion, where the results are labeled “Proposed Alg. (Experimental Dist)”, and with the model in (23), results labeled “Proposed Alg. (Theoretical Dist)”. For all three algorithms, once the optimal subset of reference view is selected, the full navigation window is reconstructed experimentally and the mean PSNR of the actual reconstructed sequence is computed.

In the following, we first validate the distortion model in (23) as well as the proposed optimization algorithm. Then, we provide simulation using the model in (23) and study the gain offered by network synthesis.

### B. Performance of the view selection algorithm

In Fig. 5, we provide the mean PSNR as a function of the available bandwidth $C$ in the setting of a regular spaced camera set $V = \{0, 1, 2, \ldots, 5, 6\}$, and a navigation window $[0.75, 5.25]$ requested by the user. Results are provided for the “Mansion”, the “Statue”, “Bikes” and the “Church” sequences in Fig. 5(a), Fig. 5(b), Fig. 5(c), and Fig. 5(d), respectively. For the “Mansion” sequence, the proposed algorithm with experimental distortion perfectly matches the exhaustive search. Also the proposed algorithm based on theoretical distortion nicely matches the exhaustive search method, with the exception of the experimental point at $C = 4$ in the network synthesis case. In that experiment, the algorithm selects as best subset $T_s = \{0.75, 2, 4, 5.25\}$ rather than $T_s = \{0.75, 2, 3, 5.25\}$ selected by the exhaustive search. The good match is verified also for the other sequences in Fig. 5(b)–(d). Fig. 5(a) also shows the gain achieved in synthesizing reference views at the cloudlets. For $C = 2$, the optimal sets of reference views are $T_s = \{0.75, 5.25\}$ and $T_{ns} = \{0, 6\}$. The possibility of selecting the view at position 0.75 as reference view reduces the reference view distance for viewpoints in $[0.75, 5.25]$ compared to the case in which camera view 0 is selected. Thus, as long as the viewpoint 0.75 is synthesized at a good quality in the network, synthesizing in the network improves the quality of the reconstructed region of interest, when the bandwidth $C$ is limited. Increasing the channel capacity reduces the quality gain between synthesis and no synthesis at the cloudlets. For $C = 4$, for example, the virtual viewpoint 0.75 is used to reconstruct the views range $[0.75, 2]$ of the navigation window. Thus, the benefit of selecting 0.75 rather than 0 is limited to a portion of the navigation window and this portion usually decreases for large $C$. Similar considerations can be derived from Fig. 5(b)–(c), for the remaining sequences.

Table III further provides the SSIM values for the same settings as for the experiments in Fig. 5, i.e., navigation window $[0.75, 5.25]$ and subset optimized with the proposed algorithm based on theoretical distortion. The gain in synthesizing reference views at the cloudlets is also clear from the SSIM metric, which is a more visually-reliable metric than PSNR.

Finally, we show the gain in synthesizing at the cloudlet with synthetic images. In particular, Fig. 6 displays the viewpoint 5 of the “Statue” sequence (Fig. 6(a)) that is respectively i) synthesized from $\{0, 6\}$ when no synthesis is allowed at the cloudlets (Fig. 6(b)) and ii) synthesized from $\{0.75, 5.25\}$ in the case of synthesis allowed at the cloudlets (Fig. 6(c)). The red box in Fig. 6(a) highlights the area in Fig. 6(b) and Fig. 6(c), respectively, that is mostly affected by the synthesis. The achieved gain can be visually observed also for the sequence “Bikes” in Fig. 7.

We then compare in Fig. 8 the performance of the exhaustive search algorithm with our optimization method in the case of non-equally spaced cameras. The “Statue” sequence is considered with unequally spaced cameras set $V = \{0, 1.5, 2, 2.75, 4, 5, 6\}$, and a navigation window $[0.75, 5.25]$ at the client. Similarly to the equally spaced scenario, the performance of proposed optimization algorithm matches the one of the exhaustive search. This confirms the validity of our assumptions and the optimality of the DP optimization solution. Also in this case, a quality gain is offered by virtual view synthesis in the network, with a maximum gain achieved for $C = 2$, with optimal reference views $T_s = \{0.75, 5.25\}$ and $T_{ns} = \{0, 6\}$.

### C. Network synthesis gain

Now, we aim at studying the performance gain due to synthesis in the network for different scenarios. However, multiview video sequences (with both texture and depth maps) currently available as test sequences have a very limited

| TABLE III | SSIM VALUE ASSOCIATED TO THE RESULTS IN FIG. 5, I.E., SSIM EXPERIENCED OVER THE NAVIGATION WINDOW $[0.75, 5.25]$ WITH OPTIMIZED REFERENCE VIEW SET FOR DIFFERENT SEQUENCES. |
|-------|-------|-------|-------|-------|
| Mansion - Synthesis | 0.865 | 0.900 | 0.912 | 0.921 |
| Mansion - No Synthesis | 0.828 | 0.878 | 0.895 | 0.910 |
| Statue - Synthesis | 0.937 | 0.964 | 0.968 | 0.971 |
| Statue - No Synthesis | 0.942 | 0.958 | 0.964 | 0.967 |
| Bikes - Synthesis | 0.895 | 0.933 | 0.948 | 0.958 |
| Bikes - No Synthesis | 0.800 | 0.890 | 0.929 | 0.946 |
| Church - Synthesis | 0.766 | 0.839 | 0.875 | 0.905 |
| Church - No Synthesis | 0.662 | 0.760 | 0.822 | 0.855 |

| TABLE IV | OPTIMAL SUBSETS FOR THE SCENARIO OF FIG. 9. |
|-------|-------|-------|
| C | $T_s$ | $T_{ns}$ |
| 2 | $\{0.75, 5.25\}$ | $\{0, 6\}$ |
| 3 | $\{0.75, 3, 5.25\}$ | $\{0, 6\}$ |
| 4 | $\{0.75, 2, 4.5, 5.25\}$ | $\{0, 2, 4.6\}$ |
| 5 | $\{0.75, 2, 4, 5.25\}$ | $\{0, 2, 3.4, 6\}$ |
| 6 | $\{0.1, 2, 3, 4.5, 6\}$ | $\{0, 1, 2, 3, 4.6\}$ |
| 7 | $\{0.1, 2, 3, 4, 5, 6\}$ | $\{0, 1, 2, 3, 4, 6\}$ |
number of views (e.g., 8 views in the Ballet video sequences\(^4\)). Because of the lack of test sequences, we consider synthetic scenarios and we adopt the distortion model in (23) both for solving the optimization algorithm and evaluating the system performance. The following results are meaningful since we already validated our synthetic distortion model in the previous subsection. For the sake of brevity, in the following we show simulation results carried out in few main scenarios. We refer to [47] for a more complete set of results.

We consider the cases of equally spaced cameras \((\mathcal{V} = \{0,1,2,\ldots,5,6\}\)) and unequally spaced cameras \((\mathcal{V} = \{0,1,3,5,7,8\}\) and \(\mathcal{V} = \{0,2,3,4,7,8\}\)) capturing the scene of interest. In Fig. 9, we show the mean PSNR as a function of the available channel capacity \(C\) when the navigation window requested by the user is \([0.75, 5.25]\) and cameras are equally spaced. The distortion of the synthesized viewpoints is evaluated with (23), with \(\gamma = 0.2\), \(D_1 = 200\), and \(d = 25\). The case of synthesis in the network is compared with the one in which only camera views can be sent to clients. In Table IV, we show the optimal subsets \(\mathcal{T}_s\) and \(\mathcal{T}_{ns}\) associated to each simulation point in Fig. 9, where camera views indexes are highlighted in bold. We observe that the case with synthesis in the network performs best in terms of quality over the navigation window. When \(C = 2\), \(\mathcal{T}_s : \{0.75, 5.25\}\) for the network synthesis case, and \(\mathcal{T}_{ns} : \{0,6\}\), otherwise. However, the larger the channel capacity the less the need for sending virtual viewpoints. When \(C = 6\), for example, both camera views 0 and 1 can be sent, thus there is no gain in transmitting only view 0.75. Finally, when \(C = 7\) and all camera views can be sent to clients, \(\mathcal{T}_s = \mathcal{T}_{ns} = \mathcal{V}\), with \(\mathcal{V}\) being the set of camera views. As expected, sending synthesized viewpoints as reference views leads to a quality gain only in constrained scenarios in which the channel capacity does not allow to send all views required for reconstructing the navigation window of interest.

We now study the gain in allowing network synthesis

---

\(^4\)http://research.microsoft.com/en-us/um/people/sbkang/3dvideodownload/
TABLE V
Selected subset of reference views and associated quality for scenarios with $|U_L^0, U_R^0| = [0.75, 2.25]$, d = 25 mm, $\gamma = 0.2$, $D_{\text{max}} = 200$.

<table>
<thead>
<tr>
<th>$\mathcal{V}$ = {0, 1, 3, 5, 7, 8}, case a</th>
<th>$\mathcal{V}$ = {0, 2, 3, 4, 7, 8}, case b</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>$T_s$</td>
</tr>
<tr>
<td>2</td>
<td>(0.75, 7.25)</td>
</tr>
<tr>
<td>3</td>
<td>(0.75, 3, 7.25)</td>
</tr>
<tr>
<td>4</td>
<td>(0.75, 3, 5, 7.25)</td>
</tr>
<tr>
<td>5</td>
<td>(0.1, 3, 5, 7.25)</td>
</tr>
<tr>
<td>6</td>
<td>(0.1, 3, 5, 7, 8)</td>
</tr>
</tbody>
</table>

Fig. 9. PSNR (in dB) as a function of the channel capacity C for different channel capacity values C for a regular spaced camera set with varying distance among cameras, $\gamma = 0.3$, $D_f = 300$, navigation window [0.75, 5.25], and camera set $\mathcal{V}$ = {0, 1, 2, ..., 5, 6} (equally spaced cameras).

Fig. 10. PSNR (in dB) vs. $U_L^0$ for a camera set $\mathcal{V}$ = {0, 2, 3, 4}, navigation window $|U_L^0, 4|$, with $d = 50$, $\gamma = 0.2$, and $D_f = 200$.

We now consider a scenario in which the camera views position is not a priori given. In Fig. 11(a), we provide the mean PSNR as a function of the variance $\sigma_v^2$, which defines the randomness of the camera views positions when acquiring the scene. More in details, we consider a navigation window $|U_L^0, U_R^0| = [2, 6]$. We then define a deterministic camera views set $\mathcal{V}_D = \{0, 1, 2, ..., 6, 7\}$, which is the best camera view set since it is aligned with the requested viewpoint navigation window. For each value of $\sigma_v^2$, we generate a random cameras set $\mathcal{V}$ as $\mathcal{V} = \mathcal{V}_D + [n_0, n_1, ..., n_7]$, where each $n_i$ is a gaussian random variable with zero mean and variance $\sigma_v^2$ with $n_i$ and $n_j$ mutually independent for $i \neq j$. Thus, the larger $\sigma_v^2$, the larger the probability for the camera view set to be not aligned with the navigation window. For each realization of $\mathcal{V}$, we run our optimization for both the cases of allowed and not allowed synthesis and we evaluate the experienced quality. For each $\sigma_v^2$ value we simulate 400 runs and we provide in Fig.
11(a) the averaged quality. What it is interesting to observe is that even if camera views are not perfectly aligned with the navigation window of interest (i.e., even for large variance values) the quality degradation with respect to the case of \( \sigma_v^2 = 0 \) is limited, about 0.5 dB for \( C = 3 \), when network synthesis is allowed. On the contrary, when synthesis is not allowed in the cloudlet, the quality substantially decreases with \( \sigma_v^2 \), experiencing a PSNR loss of almost 1.5dB. This means that network synthesis can compensate for cameras not ideally positioned in the 3D scene, as in the case of user generated content systems.

Finally, we study performance of the cloudlet-based view synthesis for a varying number of acquiring cameras. In particular, given the set of equally spaced viewpoints \( \mathcal{U} \), we assume that one every \( L \) viewpoints in \( \mathcal{U} \) is a camera view, i.e., there are \( L - 1 \) virtual viewpoints between consecutive camera views. Being the viewpoints in \( \mathcal{U} \) equally spaced, say at distance \( d \), \( Ld \) is the distance between consecutive cameras. In the following, we provide the quality behavior for \( L \) ranging from 1 to 12. For each value of the sampling distance \( L \), we simulate a navigation window spanning a range of \( 20d \). The navigation window is selected uniformly at random and the optimization algorithm evaluates the best subset of reference views. The experienced quality is averaged over 400 runs and evaluated for different values of \( L \). In Fig. 11(b), we show the mean quality for the navigation as a function of the sampling distance \( L \), for the scenario with \( C = 2 \), \( d = 50 \), \( \gamma = 0.2 \), and \( D_I = 200 \) in (23). It is worth noting that for a user to navigate at given quality, a much higher value of sampling distance \( L \) can be used when network synthesis is allowed, with respect to the value of \( L \) required with no network synthesis. For example, a mean quality in the navigation of 33 dB is achieved with \( L = 5 \) when network synthesis is not allowed as opposed to \( L = 10 \) when allowing network synthesis. This means that when synthesis is allowed, half of the number of camera views can be used respect to the case in which no synthesis is allowed. Thus, view synthesis in the network allows to maintain a good navigation quality when reducing the number of cameras.

**VII. CONCLUSION**

When interactive multiview video systems face limited bandwidth constraints, we argue that synthesizing reference views in the cloud improve the quality of navigation at the client side. In particular, we propose a synthesized reference view selection optimization problem aimed at finding the best subset of viewpoints to be transmitted to the decoder as reference views. This subset is not limited to captured camera views as in previous approaches but it can also include virtual viewpoints. The problem is formalized as a combinatorial optimization problem, which is shown to be NP-hard. However, we show that, under the general assumption that the distortion of synthesized viewpoints is well-behaved, the problem can be solved in polynomial time via a dynamic programming algorithm. Simulation results validate the performance gain of the proposed method and show that synthesizing reference views can improve image quality at the client by up to 2.1dB in PSNR. We finally demonstrate that view synthesis in the network obviates to non optimal camera sampling and permits to increase the distance between camera views without affecting the quality of the navigation. This first work on virtual view synthesis in the cloudlets shows the potentiality of using cloud processing resources for interactive multimedia applications that are a priori quite greedy in terms of network resources.

**APPENDIX**

We now outline a proof-by-construction, showing that the reference view selection problem (11) is NP-hard under the shared optimality assumption. We reduce the known NP-hard set cover (SC) problem [48] to a special case of the reference view selection problem. In SC, we are given a set of items \( S \) (called the universe), together with a defined collection \( C \) of subsets of items in \( S \). The SC problem is to identify no more than \( K \) subsets from collection \( C \) that covers \( S \), i.e., a smaller collection \( C' \subseteq C \) with \( |C'| \leq C \) such that every item in \( S \) belongs to at least one subset in collection \( C' \).

We construct a special case of our reference view selection problem as follows. We first construct the set of undistorted reference views \( S = \{ 1, 2, \ldots, |S| \} \). We set \( Q = 2 \) and the navigation window to \([U^Q_L, U^Q_R]\) = \([1, |S| + \frac{1}{2}] \). For each virtual view \( i + \frac{1}{2} \) with \( i \in S \), it exists at least one view \( v > |S| \) such that the reference view pair \((V_L, V_R) = (i, v) \) leads to
a synthesized view distortion $d_{i+\frac{1}{2}}(i, r, 0, 0) = \bar{D} < \infty$. We call this view $v$ a matching right reference view. The selection of $|S|$ left references $1, \ldots, |S|$ consumes $|S|$ views worth of bandwidth already. See Fig. 12 for an illustration, where the low PWC function shows that view $i$ is selected as left reference view in the range $[i, i + \frac{1}{2}]$. We note that given this fixed selection of left reference views, any possible selection of right reference views will satisfy the shared optimality of reference views assumption.

For each subset $j$ in collection $C = \{1, \ldots, |C|\}$ in the SC problem, we now construct in addition a matching right reference view $|S| + j$, such that if item $i$ belongs to subset $j$ in the SC problem, then the synthesized distortion $d_{i+\frac{1}{2}}(i, |S| + j, 0, 0)$ at virtual view $i + \frac{1}{2}$ will be $\bar{D}$ given right reference view $|S| + j$ is used. Thus the selection of this right reference view $|S| + j$ will enable distortion $\bar{D} < \infty$ for all the virtual views $i + \frac{1}{2}$ (items $i$) in the subset $j$. In Fig. 12, we provide an example where $|S| = 5$, $C = 4$, and $C = \{(1, 2), (1, 3), (2, 4), (3, 4, 5)\}$. The corresponding binary decision we ask is: given channel bandwidth of $|S| + C$, is there a reference view selection such that the resulting synthesized view distortion is $|S|\bar{D}$ or less?

From construction, it is clear that to minimize overall distortion, left reference views $1, \ldots, |S|$ must be first selected in any solution with distortion $\infty$. Given remaining budget of $C$ additional views, if distortion of $|S|\bar{D}$ is achieved, then $C$ or fewer additional matching right reference views $|S| + j$ are selected to achieve synthesized distortion of $\bar{D}$ at each of the virtual view $i + \frac{1}{2}, i \in \{1, \ldots, |S|\}$. Thus these additional $C$ or fewer selected right reference views correspond exactly to $C$ subsets in collection $C$ in the SC problem that covers all items $i$ in the set $S$. Thus solving this special case of the reference view selection problem is no easier than solving the SC problem, and therefore the reference view selection problem is also NP-hard.

References


