On Dependent Bit Allocation for Multiview Image Coding with Depth-Image-Based Rendering

Gene Cheung *Senior Member, IEEE*, Vladan Velisavljević *Member, IEEE*, Antonio Ortega *Fellow, IEEE*

Gene Cheung Email Address: cheung@nii.ac.jp
National Institute of Informatics, 2-1-2, Hitotsubashi, Chiyoda-ku, Tokyo, Japan 101–8430

Vladan Velisavljević Email Address: vladan.velisavljevic@telekom.de
Deutsche Telekom Laboratories, Ernst-Reuter-Platz 7, 10587 Berlin, Germany

Antonio Ortega Email Address: antonio.ortega@sipi.usc.edu
Department of Electrical Engineering, University of Southern California, Los Angeles, CA 90089

Abstract

The encoding of both texture and depth maps of a set of multi-view images, captured by a set of spatially correlated cameras, is important for any 3D visual communication system based on depth-image-based rendering (DIBR). In this paper, we address the problem of efficient bit allocation among texture and depth maps of multi-view images. More specifically, suppose we are given (1) a coding tool to encode texture and depth maps at the encoder, and (2) a view synthesis tool to construct intermediate views at the decoder using neighboring encoded texture and depth maps. Our goal is to determine how to best select captured views for encoding and distribute available bits among texture and depth maps of selected coded views, such that visual distortion of desired constructed views is minimized. First, in order to obtain at the encoder a low complexity estimate of the visual quality of a large number of desired synthesized views, we derive a cubic distortion model, based on basic DIBR properties, whose parameters are obtained using only a small number of viewpoint samples. Then, we demonstrate that the optimal selection of coded views and quantization levels for corresponding texture and depth maps is equivalent to the shortest path in a specially constructed three-dimensional trellis. Finally, we show that using the assumptions of monotonicity in predictor's quantization level and distance, sub-optimal solutions can be efficiently pruned from the feasible space during solution search. Experiments show that our proposed efficient selection of coded views and quantization levels for corresponding texture and depth maps outperforms an alternative scheme using constant quantization levels for all maps (commonly used in video standard implementations) by up to 1.5dB. Moreover, the complexity of our scheme can be reduced by at least 80% over the full solution search.

I. Introduction

Recent development of imaging technology has led to research on higher dimensional visual information processing beyond traditional two-dimensional (2D) images and single-view video, aiming at improving user's visual experience and offering new media navigation functionalities to consumers. One notable example is *multiview*

video [1], where a scene of interest is captured by a large 2D array of densely spaced, time-synchronized cameras from different viewpoints [2]. Thus, the resulting captured data has a much larger number of dimensions compared to traditional media; i.e., pixel location (i, j) at time t from camera location (x, y). In this work, we focus on the more constrained scenario where the scene of interest is *static*, and the capturing cameras are placed in a *1D horizontal array*. Hence we can drop the temporal dimension t and vertical camera shift y, and focus on a set of still images instead of video sequences. The media interaction promised for users is the ability to interactively choose viewpoint images for observation anywhere along the horizontal x-axis. We refer to this more constrained scenario as *multiview imaging* in the sequel[1].

In a typical multiview imaging scenario, a sender creates and transmits a multiview representation—composed of viewpoint images taken by the aforementioned spatially correlated cameras—of a physical scene of interest, so that a receiver can construct images of the scene from viewpoints of his own choosing for display. To efficiently encode the multiview image sequence for a given bit budget, the sender can employ disparity compensation coding tools such as those used in multiview video coding (MVC) [3] to exploit inter-view correlation among the N captured views. The receiver can subsequently decode images (texture maps) in the encoded sequence for display. See Fig. 1 for an overview of multiview imaging communication system. The available viewpoint images for the receiver are the same encoded set of N images at the sender, plus possibly intermediate images between coded images interpolated using methods[2] such as *motion compensated frame interpolation* (MCFI) [4], [5]. Because typical MCFI schemes, with no available geometric information about the scene, assume simple block-based translational motion which in general is not true for multiview images, the interpolated quality tends to be poor.

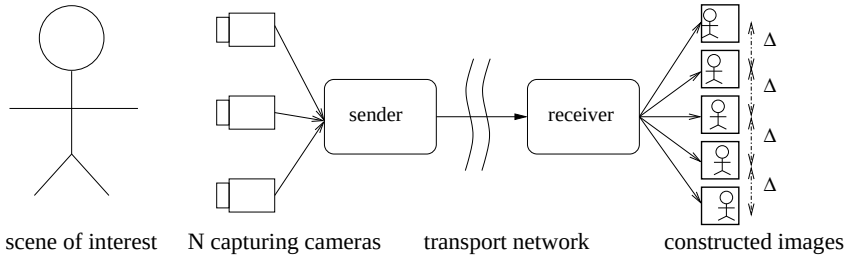


Fig. 1. Overview of a multiview imaging communication system. N cameras in a 1D array capture images of a scene from different viewpoints. The sender selects a multiview representation, compresses it and transmits it to the receiver. The receiver decodes the compressed images, and if depth maps are available, synthesizes intermediate views via DIBR that are spaced Δ apart.

One method for the receiver to improve the quality of interpolated intermediate viewpoint images that are not explicitly coded at the sender is to use depth-image-based rendering (DIBR) [6]. The idea is for the sender to

[1]The analysis and bit allocation algorithm presented in this paper for multiview images serve as a foundation for the more complex multiview video case. For example, for low-motion video, bit allocation algorithm proposed here can be used to select quantization levels for the first temporal frames of different views, which are then reused across time for the duration of the Group of Pictures (GOP) in the video.

[2]Though multiview images have disparity instead of motion, in theory interpolation methods based on motion compensation can also be used for multiview images.

encode *depth information* (distance between camera and the physical object in the scene corresponding to each captured pixel) for some viewpoints. Depth can be estimated [7] or recorded by special hardware [8]. A receiver can then synthesize additional intermediate views from received neighboring texture and depth maps using DIBR techniques such as 3D warping [9], [10]. Conveying both texture and depth maps—commonly called the *texture plus depth* representation—for a large multiview image sequence to the receiver, however, means a large amount of data must be transmitted. A natural resource allocation question hence arises: given a disparity-compensation-based coder at the sender and a DIBR view synthesis tool at the receiver, what is the "best" multiview representation of a scene for a given transmission bit budget?

More specifically, we address the following bit allocation problem for DIBR in this paper. Suppose the receiver desires to "construct" multiview images (either decode images from the coded bitstream or interpolate images from neighboring decoded images) from viewing locations that are integer multiples of a given view spacing Δ. How should the sender select captured views for coding, and select quantization levels of corresponding texture and depth maps of chosen captured views, to minimize distortion of all Δ-spaced constructed views (decoded or interpolated) at the receiver for a given coding bit budget? We focus on the scenario where the desired constructed views at the receiver are very dense (small Δ), thus offering the receiver maximum flexibility to choose virtually any viewpoints for his/her observation of the scene. From a coding perspective, dense constructed views also means that an alternative multiview representation[3] that synthesizes all required intermediate views at the sender and encodes all the generated texture maps, will require very large bit expenditure, even at coarse quantization. Hence given a small synthesized view spacing Δ, a practical multiview representation with reasonable bit budget must only encode (possibly a subset of) captured views, and rely on DIBR at the receiver to synthesize many desired intermediate views between two coded frames.

To address this bit allocation problem, the first practical difficulty is how to estimate, at the sender, the visual distortion of Δ-spaced intermediate views that would be synthesized at the receiver using neighboring encoded texture and depth maps. One obvious method to estimate synthesized distortion between two coded views at the encoder is to actually synthesize the entire set of intermediate views with spacing Δ and calculate their distortions. For small spacing Δ, however, this can be exceedingly expensive computationally.

Instead, in this paper we derive a *cubic distortion model*, based on basic properties of DIBR, in order to calculate, at low computation cost, the distortions of all synthesized intermediate images between two coded views. Specifically, given the model, we can either: i) deduce model parameters using several sample synthesized views to estimate the distortion of all required intermediate views between two coded frames, or ii) estimate the average distortion of all required intermediate views using a single image sample at the mid-point between the two coded frames. We note that, to the best of our knowledge, we are the first to estimate DIBR-synthesized view distortion of a set of densely spaced viewpoints between two coded views using a small number of image samples.

[3]When the required view spacing and/or the available bit budget is very large, a feasible multiview representation can indeed synthesize all intermediate views at the sender and encode them as regular frames. See [11] for this related bit allocation problem when the optimal representation can be a mixture of synthesized views interpolated and coded at the sender, and views synthesized at the receiver via DIBR.

Armed with our cubic distortion model, the second practical difficulty is to select an appropriate subset of captured views for coding for a given desired rate-distortion (RD) tradeoff. This is difficult because depending on the efficiency of the chosen coding and view synthesis tools, and the complexity of the captured scene, different optimal selections are possible. For example, if the captured scene is complex and requires detailed depth maps for good view interpolation, then encoding texture and depth maps of all captured views may be a good selection. On the other hand, if the captured scene is relatively easy to interpolate intermediate views at high fidelity, then synthesizing even some captured views instead of coding them can offer better RD tradeoff. Hence the issue of coded view selection is a critical one in multiview bit allocation and must be optimized for good RD performance.

In this paper, we propose a bit allocation algorithm that finds the optimal subset among captured views for encoding, and assigns quantization levels for texture and depth maps of the selected coded views. We first establish that the optimal selection of coded views and associated quantization levels is equivalent to the shortest path in a specially designed three-dimensional (3D) trellis. Given that the state space of the trellis is enormous, we then show that using lemmas derived from monotonicity assumptions in predictor's quantization level and distance, sub-optimal states and edges in the trellis can be pruned from consideration during shortest path calculation without loss of optimality. Experiments show that our proposed selection of coded views and quantization levels for corresponding texture and depth maps can outperform an alternative scheme using constant quantization levels for all texture and depth maps (commonly used in video standard implementations) by up to 1.5dB. Moreover, our search strategy reduces at least 80% of the computations compared to the full solution search that examines every state and edge in the 3D trellis.

The paper is organized as follows. After discussing related work in Section II, we derive the cubic distortion model used to estimate distortion of densely spaced synthesized views in Section III. We then formulate our bit allocation problem in Section IV. We introduce the monotonicity assumptions and propose an efficient bit allocation algorithm in Section V. We present our experimental results in Section VI. Finally, we conclude in Section VII.

II. Related Work

We divide the discussion of related work into four parts. We first motivate the value of "texture + depth" representation of a 3D static scene studied in this paper. Having established "texture + depth" is an important representation, we discuss recent advances in coding tools for texture and depth maps for multiview images and video, and new view synthesis tools using DIBR. Then, we discuss recent analysis and models for distortion of images synthesized via DIBR. Finally, we discuss related work on bit allocation for image/video coding in general.

A. Representations of 3D Static Scenes

In general, one can construct many different viable representations of a static scene for image-based rendering of any viewpoint at the receiver, including layered depth images [12], light field [13], lumigraph [14] and view-dependent texture mapping (VDTM) [15]. See [16], [9] for excellent surveys of representations proposed in the literature. For a chosen representation, coding optimization can then be performed to trade off reconstructed view

distortion with encoding rate. As a concrete example, [17] considered two representations: VDTM and model-aided predictive coding. For VDTM, [17] first constructed a 300^3-voxel model, using 257 captured images around a single object of interest (e.g., a stuffed toy animal). Given the model information, the receiver can first render the shape of the single object, then stitch texture patches on the model surface for image reconstruction. Tradeoff between synthesized view distortion and coding rate can be achieved by varying the number of bits used to encode the voxel model and the texture patches. For model-aided predictive coding, an image is first predicted by warping multiple reference images given a geometry model [18]. Prediction residuals are subsequently coded using conventional transform coding techniques. Coding rate can be reduced via coarser quantization during residual coding.

In contrast, "texture + depth" format [6]—the focus of this paper—has one texture and depth map at each captured viewpoint, where each depth map is a 2D representation of the 3D surface in the scene of interest. Image or video sequence encoded in the "texture + depth" format can enable the decoder to synthesize novel intermediate views via depth-image-based rendering (DIBR) techniques such as 3D warping [19].

"Texture + depth" format has several desirable properties. First, unlike the mesh-based geometrical model in [15] that can take hours to compute [17], depth maps can be either estimated simply using stereo-matching algorithms [7], or captured directly using time-of-flight cameras [8]. Second, depth maps can better handle complicated scenery with multiple objects, while a mesh-based model often requires dense image sampling around the single object of interest for good construction quality. Finally, "texture + depth" format is more adaptable to dynamic scene where objects change positions and shapes over time. For these and other reasons, "texture + depth" is currently the chosen format for 3D scene representation in the free viewpoint video (FTV) working group in MPEG.

Given that the "texture + depth" format is an important representation for multiview image/video, in this paper we propose a bit allocation strategy to select captured texture and depth maps for encoding at the appropriate quantization levels, so that the synthesized distortion at intermediate views of close spacing Δ is minimized. We believe we are the first in the literature to address this important problem formally; the natures of previous geometry representations (e.g., [17]) are sufficiently different from "texture + depth" format that previous empirical and theoretical optimizations do not carry over.

B. Motion / Disparity Compensation Coding Tools and DIBR View Synthesis Tools

For efficient representation of multiview images and video, novel coding tools and frame structures for texture map encoding [20], [21], [22] have been proposed in order to exploit inter-view correlation for coding gain. Similarly, new coding algorithms tailored specifically for depth maps [23], [24] have been proposed, leveraging on their unique smooth-surface and sharp-edge properties. While new coding tools are important in their own right, the associated bit allocation problem for DIBR—how bits should be optimally distributed among texture and depth maps for the chosen coding tools for maximum fidelity of reconstructed views—is not addressed in these works. We provide this missing piece in our work by solving the following two key problems: i) how to estimate distortions of a large number of synthesized intermediate views between two coded frames at the encoder at low complexity, and ii) how to optimally select a subset of captured views for coding using the optimal amount of bits for texture and depth

maps. We emphasize the generality of our proposal: our bit allocation strategy can operate no matter which of the above mentioned tools are chosen for texture and depth maps encoding.

With the advent of the *texture plus depth* representation for multiview images / video [6], enabling DIBR-based view synthesis at the decoder using received texture and depth maps, new 3D warping algorithms [9], [10] have been proposed recently in the literature. Virtual view interpolation has also been an useful tool for 3D video systems [25]; several interpolation methods based on disparity techniques have been studied in [26]. Instead of developing new view synthesis tools, our goal is to find the RD-optimal bit allocation given chosen coding tool at the encoder and DIBR-based view synthesis tool at the decoder.

C. Synthesized Distortion Model and Analysis

There has been work [27], [28], [29] studying the relationship between synthesized view distortion and lossy compression of depth map. Because distortion of depth maps creates geometric errors that ultimately affect synthesized view constructions, [28], [29] proposed new metrics based on synthesized view distortion (instead of depth map distortion) for mode selection at a block level during H.264 encoding of depth maps. Our work is different in that we find the optimal quantization parameters for texture and depth maps at the frame level. Moreover, we find the optimal subset of captured views for coding for given desired RD tradeoff.

For a two-view-only video sequence, [27] constructed a theoretical view synthesis distortion model and derived two quantization parameters, one for all texture maps and one for all depth maps, that minimize the theoretical distortion. In contrast, our proposed bit allocation scheme selects quantization parameters for individual texture and depth maps in a multi-view image sequence. Selecting one quantization parameter for every frame (rather than one for a large group of frames as done in [27]) means we can take *dependent quantization* into consideration, where a coarsely quantized predictor frame would lead to worse prediction, resulting in higher distortion and/or rate for the predicted view. In terms of modeling, unlike the complex model in [27] which requires derivation of a large number of parameters, we first derive a simple cubic distortion model (to be discussed in Section III) to model the synthesized distortion between two coded views. Then, for every pair of coded views, we construct a finite number of synthesized image as samples to deduce the four cubic polynomial coefficients specifically for this pair of coded views during the solution search. While our *operational* approach avoids a priori modeling errors (beyond our cubic distortion model), the task of data collection can be overwhelming. Hence, our focus is on complexity reduction, so that only a minimal data set is required to find the optimal solution.

D. Bit Allocation for Image / Video Coding

Operational approaches for optimal bit allocation among independent [30] and dependent [31] quantizers have been studied for single-view video coding. More recently, [32] has extended the trellis-based optimization technique in [31] to multi-view video coding where texture maps of different frames can be coded using different quantization parameters. [32] did not consider view synthesis when allocating bits to texture maps, while our work considers

bit allocation for two types of resource—texture and depth maps—for chosen subset of captured views for coding, such that the resulting distortion of both encoded and synthesized views at the decoder is minimized.

The most similar prior research to our work is the work on bit allocation for single-view video with frame skip [33], [34], [35], which studies the problem of selecting a subset of captured frames in a video to code at optimal amount of allocated bits. The frames skipped at the encoder are interpolated at the decoder using optical flow analysis. The key differences between the two problems are the following. First, for our multiview problem, both texture and depth maps for a coded view need to be coded, possibly at different quantization levels, leading to a more complicated resource allocation problem (and leading naturally to a 3D trellis, to be discussed in Section IV). Second, depth map encoding is an *auxiliary bit expenditure* that does not improve the reconstruction of the coded view itself, but improves the construction quality of intermediate views synthesized at the decoder using the coded view's texture and depth maps. There is no such "auxiliary" bit expenditure in the problem addressed in [33], [34], [35][4].

This paper extends our previous work [36], [11] on bit allocation among texture and depth maps for DIBR as follows. In [36], to evaluate the distortion of synthesized intermediate views, a small number of evenly spaced samples are chosen a priori, and the encoder synthesizes intermediate frames at all these sample locations for evaluation. In this paper, assuming the viewer desires dense viewpoint images of small spacing Δ, we derive a cubic distortion model, so that only a few intermediate view samples are constructed to estimate the distortion of all Δ-spaced synthesized intermediate views between two coded frames. Further, we validate our monotonicity assumption on predictor's quantization level and distance empirically. In [11], we studied the bit allocation problem where the required reconstructed view spacing Δ is large, so that synthesizing texture maps of intermediate views at the encoder and coding them is a viable multiview representation. The optimization proposed in [11] has high complexity, however. In this paper, we focus instead on the case when Δ is small, so that synthesizing all required intermediate views at encoder and encoding them requires too many bits and is not a viable option. By excluding this possibility, the search strategy presented here is much simpler than [11].

III. Viewpoint Sampling for Modeling of Synthesized View Distortion

The goal of a DIBR-based multiview imaging communication system is to construct high-quality images of a static scene observed from densely spaced viewpoints at the receiver. We optimize quality of all constructed views at the receiver by selecting captured views for coding and allocating bits among texture and depth maps of the selected coded views at the sender. We search for the optimal selection of coded views and bit allocation among selected views in an *operational* manner, meaning that we iteratively try different allocations and evaluate their quality (in a computationally efficient manner), until we converge to an optimal operating point and terminate the solution search.

[4]It is theoretically possible to have auxiliary bit spending that improves the interpolation quality of skipped frames in a single-view video, e.g., bits that improve optical flow prediction in the skipped frames. This was not studied in the cited previous works. If such expenditure does exist, our proposed search strategy can be used to solve this bit allocation problem for single-view video coding with frame skip as well.

To evaluate the merit of different bit allocations across texture and depth maps of coded views for this purpose, the sender needs to assess the quality of intermediate views synthesized using the encoded texture and depth maps of two neighboring coded views v_i and v_j. Denote by $d^s_{v_i,v_j}$ the sum of distortion of all desired intermediate views between coded views v_i and v_j. Then, $d^s_{v_i,v_j}$ can be written as a sum of individual synthesized view distortions $d^s_{v_i,v_j}(v)$'s at intermediate viewpoints v's, $v_i < v < v_j$:

$$d^s_{v_i,v_j} = \sum_{n=1}^{U_{v_i,v_j}(\Delta)} d^s_{v_i,v_j}(v_i + n\Delta) \quad (1)$$

$$U_{v_i,v_j}(\Delta) = \left\lceil \frac{v_j - v_i}{\Delta} \right\rceil - 1 \quad (2)$$

where Δ, as discussed in Section I, is the desired viewpoint spacing of constructed views at the receiver. $U_{v_i,v_j}(\Delta)$ is the number of desired intermediate views between viewpoints v_i and v_j (excluding v_i and v_j). In practice, each $d^s_{v_i,v_j}(v)$ can be computed as the mean square error (MSE) between the DIBR-synthesized images at viewpoint v using uncompressed texture and depth maps at v_i and v_j, and using compressed texture and depth maps at the same v_i and v_j. Since Δ is assumed to be small, the summation in (1) has many terms, and the computation of $d^s_{v_i,v_j}$ at the sender requires DIBR view synthesis of many images at many v's. Further, $d^s_{v_i,v_j}$ differs for different quantization levels chosen for the texture and depth maps of v_i and v_j; coarsely quantized texture and depth maps for v_i and v_j will naturally lead to poorer synthesized view quality. Requiring the sender to compute (1) for $d^s_{v_i,v_j}$ multiple times for different combinations of quantization levels during its solution search for optimal bit allocation is clearly too computationally expensive.

Hence, there is a need for a low-complexity methodology, so that the sender can estimate synthesized view distortions of many viewpoints between two coded frames, without first explicitly synthesizing all required intermediate views and then calculating their distortions. In addition, the methodology must maintain generality, so that its synthesized distortion estimate is reasonably accurate for a generic class of DIBR-based view synthesis tools. We discuss how we derive such a methodology next.

A. Derivation for Cubic Synthesized Distortion Model

The key to the derivation is to identify what constitutes reasonable assumptions about synthesized distortions of intermediate viewpoints between two coded frames using a DIBR-based view synthesis tool. Suppose we want to synthesize an intermediate view v between left coded view v_i and right coded view v_j. For simplicity of derivation, we assume $v_i = 0$ and $v_j = 1$. In general, a pixel in view v can be mapped to a corresponding texture image pixel in view 0 using the depth map of view 0, assuming known intrinsic and extrinsic camera parameters [37]. For simplicity, assume further that the capturing cameras are physically located in purely horizontally shifted locations, so that a pixel at a certain coordinate (k', y) in view v corresponds to a horizontally shifted pixel coordinate (k, y) in the left texture map. Denote by $g_0(v)$ the *geometric error* of pixel (k', y) at view v due to depth map distortion at view 0. In other words, $g_0(v)$ is the *offset* in number of (horizontal) pixels away from the true corresponding pixel

coordinate (k, y) in the left texture map, due to left depth map distortion, resulting in erroneous pixel coordinate $(k+g_0(v), y)$ instead. In [28], it is shown that $g_0(v)$ grows linearly with view location v; i.e., $g_0(v) = b_0\ v$, $b_0 > 0$.

Now suppose we model a row of pixels $X_0(k)$'s in the texture map of view 0 as a Gauss-Markov process; i.e.,

$$X_0(k+1) = \rho X_0(k) + w_0(k) \qquad\qquad 0 < \rho < 1 \tag{3}$$

where $w_0(k)$ is a zero-mean Gaussian variable with variance σ_0^2. One can argue that Gauss-Markov process is a good first-order model for pixels of the same physical object in a scene of interest.

Due to geometric error g, an erroneous pixel $X_0(k+g)$ at location $k+g$ in the texture map of view 0 is used for DIBR instead of the true corresponding pixel $X_0(k)$ for view synthesis. The expectation of the resulting squared error is:

$$\begin{aligned}
d_0^s(g) &= E[|X_0(k+g) - X_0(k)|^2] \\
&= E[|\rho^g X_0(k) + \rho^{g-1} w_0(k) + \rho^{g-2} w_0(k+1) + \ldots + w_0(k+g-1) - X_0(k)|^2] \\
&= E[|(\rho^g - 1) X_0(k) + \sum_{t=1}^{g} \rho^{g-t} w_0(k+t-1)|^2] \\
&= (\rho^g - 1)^2 E[X_0(k)^2] + \sigma_0^2 \sum_{t=1}^{g} \rho^{2(g-t)} \quad \leq \quad (g+1)\sigma_0^2
\end{aligned}$$

where $E[X_0(k)^2] = R_0(0) = \sigma_0^2$ is the autocorrelation $R_0(\tau) = \sigma_0^2 \rho^\tau$ of process $X_0(k)$ evaluated at $\tau = 0$. The inequality holds for $0 < \rho < 1$. Given that $g_0(v)$ is linear with respect to v, we now see that the expected squared error $d_0^s(g)$ at view v due to the left depth map distortion, $d_0^s(g_0(v))$, is also linear: $d_0^s(g_0(v)) = d_0^s(v) = (b_0\ v{+}1)\sigma_0^2$. Similarly, we can write the expected squared error due to the right depth map distortion as: $d_1^s(v) = (b_1(1{-}v){+}1)\sigma_1^2$.

In typical DIBR view synthesis, a pixel $Y(v)$ in synthesized view v, $0 < v < 1$, is a weighted sum of two corresponding pixels $X_0(k)$ and $X_1(l)$ from the left and right anchor views, where the weights, $(1-v)$ and (v), depend linearly on the distances to the two anchor views; i.e., $Y(v) = (1{-}v)X_0(k){+}(v)X_1(l)$. Due to the left and right depth map distortions, a pixel in synthesized view v becomes $\hat{Y}(v) = (1{-}v)X_0(k{+}g_0(v)){+}vX_1(l{+}g_1(v))$. Thus, the squared error $d_{0,1}^s(v) = |\hat{Y}(v) - Y(v)|^2$ in the synthesized pixel due to distortion in the left and right depth maps can be derived as follows:

$$\begin{aligned}
d_{0,1}^s(v) &= E[|\hat{Y}(v) - Y(v)|^2] \\
&= E[|(1-v)X_0(k+g_0(v)) + (v)X_1(l+g_1(v)) - (1-v)X_0(k) - (v)X_1(l)|^2] \\
&= E[|(1-v)\left(X_0(k+g_0(v)) - X_0(k)\right) + (v)\left(X_1(l+g_1(v)) - X_1(l)\right)|^2] \\
&= (1-v)^2 d_0^s(v) + (v)^2 d_1^s(v) \ + \\
&\quad + (v)(1-v)E[(X_0(l+g_0(v)) - X_0(k))(X_1(l+g_1(v)) - X_1(l))] \\
&= (1-v)^2(b_0\ v+1)\sigma_0^2 + (v)^2(b_1(1-v)+1)\sigma_1^2 \\
&= \underbrace{(b_0\sigma_0^2 - b_1\sigma_1^2)}_{c_3}\ v^3 \ + \ \underbrace{((1-2b_0)\sigma_0^2 + (b_1+1)\sigma_1^2)}_{c_2}\ v^2 \ + \ \underbrace{((b_0-2)\sigma_0^2)}_{c_1}\ v \ + \ \underbrace{(\sigma_0^2)}_{c_0}
\end{aligned} \tag{4}$$

where we assume pixels in the left and right texture maps $X_0(k)$ and $X_1(l)$ are independent processes, and c_i's are the cubic polynomial coefficients. We now see that $d^s_{0,1}(v)$ is in general a cubic function with respect to the intermediate view location v.

Notice that if the left and right Markov-Gauss processes are of the same object, then $b_0 = b_1$ and $\sigma_0^2 = \sigma_1^2$. The cubic term equals to zero, and we have a quadratic function:

$$d^s_{0,1}(v) = (2 - b_0)\sigma_0^2 v^2 + (b_0 - 2)\sigma_0^2 v + \sigma_0^2 \tag{5}$$

Taking the derivative of $d^s_{0,1}(v)$ with respect to v and setting it equal to 0, we see the maximum distortion occurs at mid-point $v = 1/2$. We can hence conclude the following: if distortions in left and right depth maps are not severe, then DIBR will be performed using corresponding pixels in the left and right texture maps of the same object for majority of pixels in the synthesized view, and the resulting distortion is quadratic. This is what was observed experimentally in [38] as well. If distortions in left and right depth maps are severe enough that DIBR erroneously uses pixels of different objects for interpolation for majority of pixels in the synthesized view, then the distortion becomes cubic.

Note that in addition to (4), there are secondary non-linear effects on the synthesized distortion $d^s_{v_i,v_j}(v)$ due to: i) occlusion of different spatial regions with respect to the viewpoint v determined by complex scene geometry, ii) pixel coordinate rounding operations used in the view synthesis (i.e., a 3D-warped point is usually displayed at the nearest integer pixel location in the synthesized view), and iii) statistical discrepancies in texture maps, as discussed previously. We consider these effects secondary and focus instead on the major trend outlined by the cubic distortion model. For the sake of simplicity, we model the sum of these effects as a small noise term[5] $n(v)$.

B. Sampling for Cubic Distortion Model

Though we have concluded that the appropriate distortion model as a function of intermediate view v is a cubic function, we still need to find coefficients c_i's that characterize cubic polynomial function $\tilde{d}^s(v) = c_0 + c_1 v + c_2 v^2 + c_3 v^3$ for given coded texture and depth maps at anchor views v_i and v_j. Our approach is sampling: synthesize a small number of images at intermediate views v_k's between v_i and v_j and calculate corresponding distortions $d^s_{v_i,v_j}(v_k)$'s, so that using samples (v_k, d^s_k)'s, we can compute coefficients c_i's in some optimal fashion. We present two sampling methods below.

In the first method, we use S even-spaced samples (v_k, d^s_k)'s between v_i and v_j to derive "optimal" coefficients c_i's in the cubic polynomial. For each data point (v_k, d^s_k), we can express the distortion d^s_k as a cubic function $c_0 + c_1 v_k + c_2 v_k^2 + c_3 v_k^3$ plus error e_k; i.e., in matrix form, we write:

[5]Size of the noise will be larger if the quality of the obtained depth maps are poor and/or if the captured images are not perfectly rectified. Nonetheless, we stress that even in those cases, the derived cubic distortion model is still accurate up to a first-order approximation, especially when the capturing cameras are physically very close to each other.

$$\underbrace{\begin{bmatrix} 1 & v_1 & v_1^2 & v_1^3 \\ 1 & v_2 & v_2^2 & v_2^3 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & v_S & v_S^2 & v_S^3 \end{bmatrix}}_{\mathbf{V}} \underbrace{\begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix}}_{\mathbf{c}} = \underbrace{\begin{bmatrix} d_1^s \\ d_2^s \\ \vdots \\ d_S^s \end{bmatrix}}_{\mathbf{d}^s} + \underbrace{\begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_S \end{bmatrix}}_{\mathbf{e}} \tag{6}$$

By optimal, we mean coefficients c_i's lead to the smallest squared errors $\mathbf{e}$ possible. Using linear regression [39], optimal c_i's can be calculated simply:

$$\mathbf{c}^* = \underbrace{(\mathbf{V}'\mathbf{V})^{-1}\mathbf{V}'}_{\mathbf{V}^+}\mathbf{d}^s \tag{7}$$

where $\mathbf{V}^+$ is the Moore-Penrose pseudo-inverse of $\mathbf{V}$.

The constructed cubic distortion model will be used to calculate the sum of synthesized distortions between the two coded views v_i and v_j, $\tilde{d}^s_{v_i,v_j}$, as follows:

$$\tilde{d}^s_{v_i,v_j} = \sum_{n=1}^{U_{v_i,v_j}(\Delta)} \tilde{d}^s(v_i + n\Delta) \tag{8}$$

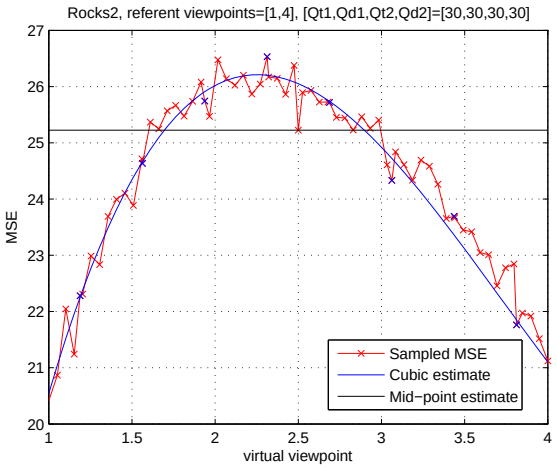
Clearly, $\tilde{d}^s_{v_i,v_j}$ in (8) is an approximation to the true synthesized distortion $d^s_{v_i,v_j}$ in (1) at much reduced computation complexity. As an example, we see that in Fig. 2, using the cubic distortion model we constructed curves (blue) using eight samples each. We see that in both cases, the cubic model captures the general trend of the actual distortions (red) quite well. In addition, we see that for fine quantization levels of depth maps in Fig. 2(a), the curve does behave more like a quadratic function, as predicted by our model. Extensive empirical evidence showing the accuracy of the model is provided in Section VI.

Notice that in the first sampling method, we need S samples to find the four coefficients $c_0, \ldots, c_3$ in the cubic distortion model. It is recommended [39] that the number of samples S required should be at least multiples of the number of parameters; in our experiments we use eight samples. This still translates to a non-negligible computation overhead. To further reduce computation, in the second sampling method we only sample at the mid-point $(v_i+v_j)/2$ between two coded views, and scale it by the number of desired intermediate views $U_{v_i,v_j}(\Delta)$ to obtain an estimate $\hat{d}^s_{v_i,v_j}$, i.e,:

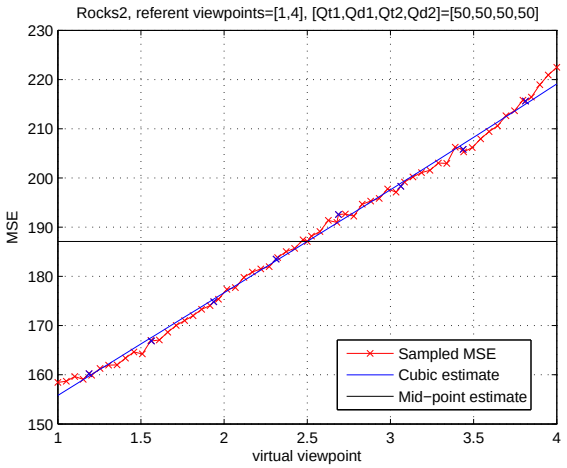
$$\hat{d}^s_{v_i,v_j} = U_{v_i,v_j}(\Delta) \ * \ d^s_{v_i,v_j}((v_i+v_j)/2). \tag{9}$$

As previously discussed, if distortions in left and right depth maps are small, then we expect a quadratic function with peak at mid-point, and this mid-point sampling method captures the maximum distortion. If distortions in left and right depth maps are very large, this mid-point sampling method is no longer guaranteed to be accurate. However, the distortions in such extreme cases are very large anyway, and they will not be selected as operational parameters for optimal bit allocation.

In the sequel, we will assume that whenever the synthesized distortion $d^s_{v_i,v_j}$ between two coded views v_i and v_j needs to be computed in our solution search, we will invoke either (8) for $\tilde{d}^s_{v_i,v_j}$ or (9) for $\hat{d}^s_{v_i,v_j}$ as a low-complexity estimate. We will investigate in Section VI the accuracy of both sampling methods experimentally.



(a) Synthesized MSE vs. viewpoint for $QP = 30$



(b) Synthesized MSE vs. viewpoint for $QP = 50$

Fig. 2. Synthesized distortion is plotted against viewpoint location for different quantization levels for `Rocks2` sequence [40]. Cubic distortion model (blue), mid-point (black) and actual synthesized distortion at 0.05 view spacing are shown.

IV. Formulation

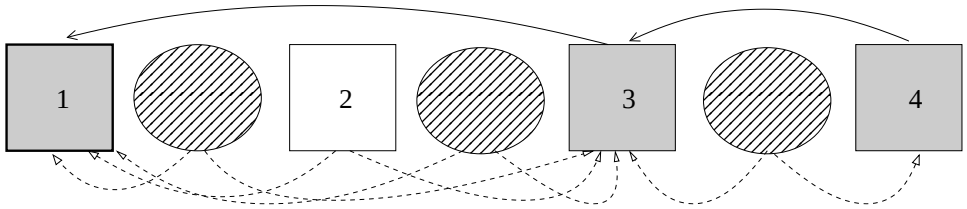


Fig. 3. Example of multiview image sequence. Coding dependencies among coded views (gray) are shown in solid arrows. View synthesis dependencies between an intermediate (patterned) view and two neighboring coded views (gray), and between an uncoded view (white) and two neighboring coded views (gray), are shown in dotted arrows. Coded and uncoded views are $\mathcal{J} = \{1, 3, 4\}$ and $\mathcal{J}' = \{2\}$, respectively. Note that each patterned ellipsoid represents many desired intermediate views at spacing Δ between two neighboring captured views.

We now formulate our bit allocation problem formally as follows. A set of camera-captured views $\mathcal{N} = \{v_1, \ldots, v_N\}$ in a 1D-camera-array arrangement, and a desired constructed view spacing Δ, are specified a priori as input to the optimization. For mathematical simplicity, we will assume that each captured view v_n can be expressed as a positive integer multiple of Δ, i.e. $v_n = n\Delta$, $n \in \mathcal{Z}^+$. Captured views $\mathcal{N}$ are divided into K *coded views*, $\mathcal{J} = \{j_1, \ldots, j_K\}$, and $N - K$ *uncoded views* $\mathcal{J}' = \mathcal{N} \setminus \mathcal{J}$. Coded views are captured views that are selected for encoding by the sender. Uncoded views are synthesized at the receiver along with *intermediate views* (views that the user desires viewing but are not explicitly captured by cameras at the sender). The first and last captured views in $\mathcal{N}$ must be selected as coded views; i.e., $v_1, v_N \in \mathcal{J} \subseteq \mathcal{N}$. Texture and depth maps of a coded view j_k are encoded using quantization level q_{j_k} and p_{j_k}, respectively. q_{j_k} and p_{j_k} take on discrete values from quantization

level set $\mathcal{Q} = \{1, \ldots, Q_{\max}\}$ and $\mathcal{P} = \{1, \ldots, P_{\max}\}$, respectively, where we assume the convention that a larger q_{j_k} or p_{j_k} implies a coarser quantization.

Uncoded views and intermediate views are synthesized at the receiver, each using texture and depth maps of the closest left and right coded views. We assume inter-view differential coding is used for coded views as done in [21]. That means there exists dependency between an uncoded view and two neighboring coded views, between an intermediate view and two neighboring coded views, and between two neighboring coded views (due to differential coding). Fig. 3 shows an example. The first view is always coded as an I-frame. Each subsequent coded view j_k—frames 3 and 4 in Fig. 3—is coded as P-frame using previous coded view j_{k-1} as predictor for disparity compensation. Each uncoded or intermediate view depends on two neighboring coded views.

A. Signal Distortion

Given the coded view dependencies, we can now write the distortion D^c of the coded views as a function of the texture map quantization levels, $\mathbf{q} = [q_{j_1} \ldots, q_{j_K}]$:

$$D^c(\mathbf{q}) = d^c_{j_1}(q_{j_1}) + \sum_{k=2}^{K} d^c_{j_k, j_{k-1}}(q_{j_k}, q_{j_{k-1}}) \tag{10}$$

which states that the distortion $d^c_{j_1}$ of starting viewpoint j_1 (coded as I-frame) depends only on its own texture quantization level q_{j_1}, while the distortion of a P-frame $d^c_{j_k}$ depends on both its own texture quantization level q_{j_k} and its predictor j_{k-1}'s quantization level $q_{j_{k-1}}$. A more general model [31] is to have P-frame j_k depend on its own q_{j_k} and all previous quantization levels $q_{j_1}, \ldots, q_{j_{k-1}}$. We assume here that truncating the dependencies to $q_{j_{k-1}}$ only is a good first-order approximation, as done in previous works such as [41].

Similarly, we now write the distortion of the synthesized views D^s (including uncoded views $\mathcal{J}'$ and intermediate views) as a function of $\mathbf{q}$ and depth quantization levels, $\mathbf{p} = [p_{j_1}, \ldots, p_{j_K}]$:

$$D^s(\mathbf{q}, \mathbf{p}) = \sum_{k=1}^{K-1} d^s_{j_k, j_{k+1}}(q_{j_k}, p_{j_k}, q_{j_{k+1}}, p_{j_{k+1}}) \tag{11}$$

where $d^s_{j_k, j_{k+1}}$ is the sum of synthesized view distortions between coded views j_k and j_{k+1}, as described in (1), given the texture and depth map quantization levels (q_{j_k}, p_{j_k}) and $(q_{j_{k+1}}, p_{j_{k+1}})$ for coded views j_k and j_{k+1}. In words, distortion of the synthesized views depends on both the texture and depth map quantization levels of the two spatially closest coded views.

B. Encoding Rate

As done for distortion, we can write the rate of texture and depth maps of coded views, R^c and R^s, respectively, as follows:

$$R^c(\mathbf{q}) = r^c_{j_1}(q_{j_1}) + \sum_{k=2}^{K} r^c_{j_k,j_{k-1}}(q_{j_k}, q_{j_{k-1}}) \tag{12}$$

$$R^s(\mathbf{q},\mathbf{p}) = r^s_{j_1}(q_{j_1}, p_{j_1}) + \sum_{k=2}^{K} r^s_{j_k,j_{k-1}}(q_{j_k}, p_{j_k}, q_{j_{k-1}}, p_{j_{k-1}}) \tag{13}$$

(12) states that the encoding rate for texture map of a coded view, $r^c_{j_k}$, depends on its texture map quantization level, q_{j_k}, and its predictor's level, $q_{j_{k-1}}$. In contrast, (13) states that the encoding rate for depth map, $r^s_{j_k}$, depends on both the texture and depth map quantization levels, q_{j_k} and p_{j_k}, and its predictor's texture and depth map levels, $q_{j_{k-1}}$ and $p_{j_{k-1}}$. Note that though we assume depth maps are coded independently from texture maps in experimental Section VI, there does exist correlation between texture and depth maps, and one can devise joint texture/depth map coding schemes that exploit this correlation for coding gain [42]. Our formulation is sufficiently general to include the case when depth maps are differentially coded using texture maps as predictors.

C. Rate-distortion Optimization

Given the above formulation, the optimization we are interested in is to find the coded view indices $\mathcal{J} \subseteq \mathcal{N}$, and associated texture and depth quantization vector, $\mathbf{q}$ and $\mathbf{p}$, such that the Lagrangian objective L_λ is minimized for given Lagrangian multiplier $\lambda \geq 0$:

$$\min_{\mathcal{J},\mathbf{q},\mathbf{p}} L_\lambda = D^c(\mathbf{q}) + D^s(\mathbf{q},\mathbf{p}) + \lambda\left[R^c(\mathbf{q}) + R^s(\mathbf{q},\mathbf{p})\right] \tag{14}$$

For clarity of later presentation, we will in addition define *local Lagrangian cost* for a differentially coded view j_k as follows. Let $\phi_{j_k,j_{k-1}}(q_{j_k}, p_{j_k}, q_{j_{k-1}}, p_{j_{k-1}})$ be the Lagrangian term for coded view j_k given quantization levels of view j_k and its predictor view j_{k-1}, i.e., the sum of distortion $d^c_{j_k,j_{k-1}}(q_{j_k}, q_{j_{k-1}})$ and penalties $\lambda r^c_{j_k,j_{k-1}}(q_{j_k}, q_{j_{k-1}})$ and $\lambda r^s_{j_k,j_{k-1}}(q_{j_k}, p_{j_k}, q_{j_{k-1}}, p_{j_{k-1}})$ for texture and depth maps encoding. $\phi_{j_k,j_{k-1}}$ will be used to mathematically describe the two key monotonicity assumptions in the next section.

V. Bit Allocation Optimization

We first discuss how the optimal solution to (14) corresponds to the shortest path (SP) in a specially constructed 3D trellis. Nevertheless, the complexity of constructing the full trellis is large, and hence we will discuss methods to reduce the complexity using monotonicity assumptions of predictor's quantization level and distance. Using the assumptions, only a small subset of the trellis needs to be constructed and traversed as the modified SP search algorithm is executed.

A. Full Trellis & Viterbi Algorithm

We first show that the optimal solution to (14) can be computed by first constructing a three-dimensional (3D) trellis, and then finding the SP from the left end of the trellis to the right end using the famed Viterbi Algorithm (VA).

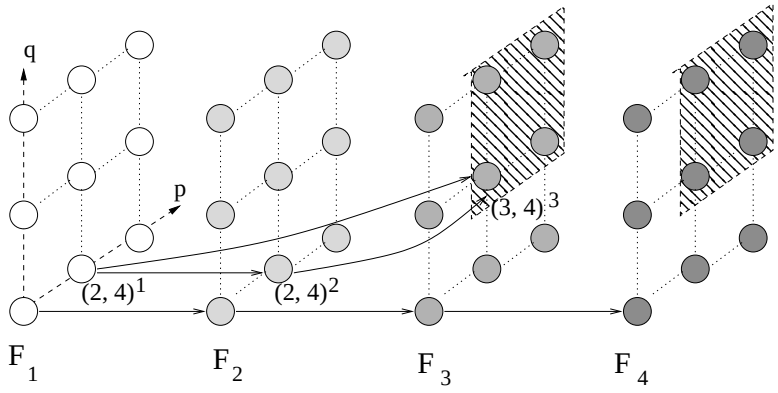


Fig. 4. Optimization 3D Trellis.

We can construct a trellis—one corresponding to the earlier example is shown in Fig. 4—for the selection of coded view indices $\mathcal{J}$, texture and depth quantization levels $\mathbf{q}$ and $\mathbf{p}$ as follows. Each captured view $v_n \in \mathcal{N}$ is represented by a *plane* of states, where each state represents a pair of quantization levels $(q_{v_n}, p_{v_n})^{v_n}$ for texture and depth maps. States in the first plane corresponding to the first view v_1 will be populated with Lagrangian costs $\phi_{v_1}(q_{v_1}, p_{v_1})$'s for different level pairs $(q_{v_1}, p_{v_1})^{v_1}$'s. Each directed edge from a state $(q_{v_1}, p_{v_1})^{v_1}$ in the first plane to a state in the second plane $(q_{v_2}, p_{v_2})^{v_2}$ of neighboring captured view $v_2 \in \mathcal{N}$ will carry a Lagrangian cost $\phi_{v_2,v_1}(q_{v_2}, p_{v_2}, q_{v_1}, p_{v_1})$ *and* synthesized view distortions $d^s_{v_1,v_2}(q_{v_1}, p_{v_1}, q_{v_2}, p_{v_2})$. Selecting such edge would mean captured views v_1 and v_2 are both selected as coded views in $\mathcal{J}$. Each directed edge from a state $(q_{v_1}, p_{v_1})^{v_1}$ in the first plane to a state $(q_{v_n}, p_{v_n})^{v_n}$ in a further-away plane of captured view $v_n \in \mathcal{N}$ will carry similar Lagrangian cost $\phi_{v_n,v_1}(q_{v_n}, p_{v_n}, q_{v_1}, p_{v_1})$ and synthesized view distortions $d^s_{v_1,v_n}(q_{v_1}, p_{v_1}, q_{v_n}, p_{v_n})$. Selecting such edge would mean captured view v_1 and v_n are both selected as coded views in $\mathcal{J}$ with no coded views in-between.

We state without proof that the SP from any state in the left-most plane to any state in the right-most plane, found using VA, corresponds to the optimal solution to (14). However, the number of states and edges in the trellis alone are prohibitive: $O(|\mathcal{Q}||\mathcal{P}|N)$ and $O(|\mathcal{Q}|^2|\mathcal{P}|^2N^2)$, respectively. Hence the crux of any complexity reduction method is to find the SP by visiting only a small subset of states and edges. Towards that goal, we first discuss monotonicity assumptions next.

B. Monotonicity in Predictor's Quantization Level

Motivated by a similar empirical observation in [31], we show here the *monotonicity in predictor's quantization level* for both Lagrangian $\phi_{j_k,j_{k-1}}$ of coded view j_k, and synthesized view distortion $d^s_{j_k,j_{k+1}}$ of intermediate views between coded views j_k and j_{k+1}. The assumption is formally stated as follows:

> The Lagrangian term $\phi_{j_k,j_{k-1}}(q_{j_k}, p_{j_k}, q_{j_{k-1}}, p_{j_{k-1}})$ for coded view j_k given the predictor view j_{k-1} and the synthesized view distortion $d^s_{j_k,j_{k+1}}$ is a monotonically non-decreasing function of the predictor's quantization levels. That is,

$$\begin{aligned}
\phi_{j_k,j_{k-1}}(q_{j_k},p_{j_k},q_{j_{k-1}},p_{j_{k-1}}) &\leq \phi_{j_k,j_{k-1}}(q_{j_k},p_{j_k},q^+_{j_{k-1}},p_{j_{k-1}}) \qquad (15)\\
\phi_{j_k,j_{k-1}}(q_{j_k},p_{j_k},q_{j_{k-1}},p_{j_{k-1}}) &\leq \phi_{j_k,j_{k-1}}(q_{j_k},p_{j_k},q_{j_{k-1}},p^+_{j_{k-1}})\\
d^s_{j_k,j_{k+1}}(q_{j_k},p_{j_k},q_{j_{k+1}},p_{j_{k+1}}) &\leq d^s_{j_k,j_{k+1}}(q^+_{j_i},p_{j_k},q_{j_{k+1}},p_{j_{k+1}}) \qquad (16)\\
d^s_{j_k,j_{k+1}}(q_{j_k},p_{j_k},q_{j_{k+1}},p_{j_{k+1}}) &\leq d^s_{j_k,j_{k+1}}(q_{j_k},p^+_{j_k},q_{j_{k+1}},p_{j_{k+1}})\\
d^s_{j_k,j_{k+1}}(q_{j_k},p_{j_k},q_{j_{k+1}},p_{j_{k+1}}) &\leq d^s_{j_k,j_{k+1}}(q_{j_k},p_{j_k},q^+_{j_{k+1}},p_{j_{k+1}})\\
d^s_{j_k,j_{k+1}}(q_{j_k},p_{j_k},q_{j_{k+1}},p_{j_{k+1}}) &\leq d^s_{j_k,j_{k+1}}(q_{j_k},p_{j_k},q_{j_{k+1}},p^+_{j_{k+1}}),
\end{aligned}$$

where q_v^+ (or p_v^+) implies a larger (coarser) quantization level than q_v (or p_v).

In words, (15) states that if predictor view j_{k-1} uses a coarser quantization level in texture or depth map, it will lead to a worse prediction for view j_k, resulting in a larger distortion and/or coding rate, and hence a larger Lagrangian cost $\phi_{j_k,j_{k-1}}$ for $\lambda \geq 0$. Similarly, (16) makes a statement for monotonicity of the synthesized view distortion. A coarser texture quantization (larger q_{j_i} or $q_{j_{i+1}}$) results in a higher synthesized distortion $d^s_{j_i,j_{i+1}}$; since a synthesized pixel is a linear combination of two corresponding pixels in the left and right coded texture map (as discussed in Section III-A), a larger quantization error in the left or right texture pixel will translate to a larger error in the synthesized pixel as well. A coarser depth quantization (larger p_{j_i} or $p_{j_{i+1}}$) leads to a larger geometric error and results in a larger synthesized distortion $d^s_{j_i,j_{i+1}}$ (also discussed in Section III-A). We will provide empirical evidence of this monotonicity assumption in Section VI.

C. Monotonicity in Predictor's Distance

We can also express monotonicity of Lagrangian cost $\phi_{\zeta,\xi}$ of coded view ζ given predictor view ξ, $\xi < \zeta$, and synthesized view distortion $d^s_{\zeta,\xi}(v)$ at intermediate view v between coded views, that is $\xi < v < \zeta$, with respect to the *predictor's distance* to a coded view used for differential coding or synthesis. For $\phi_{\zeta,\xi}$, we first assume further-away predictor view ξ^- for coded view ζ, $\xi^- < \xi$, has the same quantization levels as view ξ. Similarly, for $d^s_{\zeta,\xi}(v)$, we assume further-away predictor views ζ^- and ξ^+, $\zeta^- < \zeta$ and $\xi^+ > \xi$, have the same quantization levels for synthesized view v as respective levels of views ζ and ξ. We can then formulate the following monotonicity assumption:

> The Lagrangian term $\phi_{\zeta,\xi}(q_\zeta,p_\zeta,q_\xi,p_\xi)$ for coded view ζ given predictor view ξ, and the synthesized view distortion $d^s_{\zeta,\xi}(q_\zeta,p_\zeta,q_\xi,p_\xi)(v)$ for intermediate view v given closest left and right coded view ζ and ξ, are monotonically non-decreasing functions of the predictor's distance. That is,

$$\begin{aligned}
\phi_{\zeta,\xi}(q_\zeta,p_\zeta,q_\xi,p_\xi) &\leq \phi_{\zeta,\xi^-}(q_\zeta,p_\zeta,q_\xi,p_\xi) \qquad (17)\\
d^s_{\zeta,\xi}(q_\zeta,p_\zeta,q_\xi,p_\xi)(v) &\leq d^s_{\zeta,\xi^+}(q_\zeta,p_\zeta,q_\xi,p_\xi)(v) \qquad (18)\\
d^s_{\zeta,\xi}(q_\zeta,p_\zeta,q_\xi,p_\xi)(v) &\leq d^s_{\zeta^-,\xi}(q_\zeta,p_\zeta,q_\xi,p_\xi)(v),
\end{aligned}$$

where ζ^- implies $\zeta^- < \zeta$, and ξ^+ implies $\xi^+ > \xi$.

In other words, (17) states that a further-away predictor, with the same quantization levels as the original predictor, provides a worse prediction for differential coding, hence a larger Lagrangian term $\phi_{\zeta,\xi}(q_\zeta, p_\zeta, q_\xi, p_\xi)$. (18) states that, for synthesized view distortion $d^s_{\zeta,\xi}(q_\zeta, p_\zeta, q_\xi, p_\xi)(v)$, a further-away predictor means a larger distance $v - v_i$ between predictor frame at view v_i and predictee frame at view v. That means a larger geometric error $g_{v_i}(v)$, as discussed in Section III-A, which again leads to a larger synthesized distortion. This assumption has also been shown valid in [43] using the Markov random field prior model, and we will verify it empirically in Section VI. We note that while monotonicity in predictor's quantization level has been used extensively [31], [32], [35], we are the first in the literature to exploit monotonicity in predictor's distance for bit allocation.

D. Reducing Complexity

Given the described monotonicity assumptions, we now derive lemmas that will be used to construct a fast SP search algorithm. Let $\Phi_{v_n}(q_{v_n}, p_{v_n})$ be the shortest sub-path (minimum Lagrangian cost sub-path) from any states of first view to state $(q_{v_n}, p_{v_n})^{v_n}$ of captured view v_n. The first lemma eliminates *sub-optimal states* $(q_{v_n}, p_{v_n})^{v_n}$'s, given computed $\Phi_{v_n}(q_{v_n}, p_{v_n})$'s, using monotonicity in predictor's quantization level.

Lemma 1: For given texture map quantization level p_{v_n}, if at state plane of captured view v_n, $\Phi_{v_n}(q^+_{v_n}, p_{v_n}) > \Phi_{v_n}(q^*_{v_n}, p_{v_n})$, $\forall q^+_{v_n} > q^*_{v_n}$, then sub-paths up to states $(q^+_{v_n}, p_{v_n})^{v_n}$, $\forall q^+_{v_n} > q^*_{v_n}$, cannot belong to an end-to-end SP.

In other words, Lemma 1 states that if sub-path cost to state $(q^+_{v_n}, p_{v_n})$ with coarse texture quantization level $q^+_{v_n}$ is already larger than sub-path cost to state $(q^*_{v_n}, p_{v_n})$ with fine texture quantization level $q^*_{v_n}$, then state $(q^+_{v_n}, p_{v_n})$ is globally sub-optimal. A simple proof is provided in the Appendix.

Lemma 1 also holds true for depth quantization level p_{v_n}: given q_{v_n}, if $\Phi_{v_n}(q_{v_n}, p^+_{v_n}) > \Phi_{v_n}(q_{v_n}, p^*_{v_n})$, $\forall p^+_{v_n} > p^*_{v_n}$, then states $(q_{v_n}, p^+_{v_n})^{v_n}$'s, $\forall p^+_{v_n} > p^*_{v_n}$, are globally sub-optimal and can be pruned.

The next lemma eliminates *sub-optimal edges* stemming from state $(p_{v_n}, q_{v_n})^{v_n}$ of captured view v_n to a state in further-away coded view ξ, $\xi > v_n$, using monotonicity in predictor's distance.

Lemma 2: Given start state $(q_{v_n}, p_{v_n})^{v_n}$ of captured view v_n, end state $(q_\xi, p_\xi)^\xi$ of captured view ξ, and in-between captured view v_{n+1}, $v_n < v_{n+1} < \xi$, if cost of traversing state $(q_{v_n}, p_{v_n})^{v_{n+1}}$ of view v_{n+1}, $\phi_{v_{n+1},v_n} + d^s_{v_n,v_{n+1}}$, is smaller than a lower-bound cost of skipping view v_{n+1}, $\sum_{x=1}^{U_{v_n,v_{n+1}}(\Delta)} d^s_{v_n,\xi}(v_n + x\Delta)$, then edge $(q_{v_n}, p_{v_n})^{v_n} \rightarrow (q_\xi, p_\xi)^\xi$ cannot belong to an end-to-end SP.

In other words, Lemma 2 states that if from state $(q_{v_n}, p_{v_n})^{v_n}$ of captured view v_n, traversing state $(q_{v_n}, p_{v_n})^{v_{n+1}}$ of captured view v_{n+1} with same quantization levels is cheap in Lagrangian cost compared to a lower-bound cost of skipping captured view v_{n+1}, en route to destination state $(q_\xi, p_\xi)^\xi$, then skipping captured view v_{n+1} using edge $(q_{v_n}, p_{v_n})^{v_n} \rightarrow (q_\xi, p_\xi)^\xi$ is sub-optimal. A simple proof is provided in the Appendix.

The corollary of Lemma 2 is that if the said condition holds, then edges $(q_{v_n}, p_{v_n})^{v_n} \rightarrow (q_{\xi^+}, p_{\xi^+})^{\xi^+}$, $\forall q_{\xi^+} \geq q_\xi, p_{\xi^+} \geq p_\xi$, where ξ^+ means all indices larger than ξ, also cannot belong to the SP. The reason is: synthesized distortion $d^s_{v_n,\xi}(v)$ of intermediate view v using coded view v_n and ξ as predictors is surely no larger than $d^s_{v_n,\xi^+}(v)$ using coded view v_n and further-away coded view ξ^+ with same or coarser quantization levels. Hence the said

condition must hold also for $(q_{\xi^+}, p_{\xi^+})^{\xi^+}$ as well, and the same argument as proof 2 follows to rule out edge $(q_{v_n}, p_{v_n})^{v_n} \rightarrow (q_{\xi^+}, p_{\xi^+})^{\xi^+}$. As an example, in Fig. 4 if the cost of traversing state $(2,4)^2$, $\phi_{2,1} + d^s_{1,2}$, is smaller than $\sum_{x=1}^{U_{1,2}(\Delta)} d^s_{1,3}(1 + x\Delta)$, then edges from $(2,4)^1$ to all states on the shaded region, including $(3,4)^3$ of view 3, can be eliminated.

E. Bit Allocation Algorithm

Algorithm 1 Bit Allocation Algorithm

1: $n \leftarrow 1$. $\Phi_{v_1}(q_{v_1}, p_{v_1}) \leftarrow \phi(q_{v_1}, p_{v_1})$, for all states $(q_{v_1}, p_{v_1})^{v_1}$ of first captured view v_1.
2: $q^*_{v_n} \leftarrow \arg\min_{q_{v_n}} \Phi_{v_n}(q_{v_n}, p_{v_n})$, for each p_{v_n} of view v_n. Eliminate states $(q^+_{v_n}, p_{v_n})^{v_n}$'s, $q^+_{v_n} > q^*_{v_n}$.
3: $p^*_{v_n} \leftarrow \arg\min_{p_{v_n}} \Phi_{v_n}(q_{v_n}, p_{v_n})$, for each q_{v_n} of view v_n. Eliminate states $(q_{v_n}, p^+_{v_n})^{v_n}$'s, $p^+_{v_n} > p^*_{v_n}$.
4: For each survived state $(q_{v_n}, p_{v_n})^{v_n}$ of view v_n, evaluate forward sub-paths to states $(q_{v_{n+1}}, p_{v_{n+1}})^{v_{n+1}}$'s of neighboring captured view v_{n+1}.
5: For each survived state $(q_{v_n}, p_{v_n})^{v_n}$ of view v_n, using state $(q_{v_n}, p_{v_n})^{v_{n+1}}$ of neighboring captured view v_{n+1}, evaluate sub-paths forward: i.e.,
6: $\zeta \leftarrow$ neighboring captured view of v_{n+1}, where $\zeta > v_{n+1}$. Length-$P_{\max}$ vector $\mathbf{Q}_{\lim} \leftarrow [Q_{\max}, \ldots, Q_{\max}]$.
7: **for** each state $(q_\zeta, p_\zeta)^\zeta$, s.t. $q_\zeta \leq \mathbf{Q}_{\lim}(p_\zeta)$, **do**
8: **if** $\phi_{v_{n+1}, v_n} + d^s_{v_n, v_{n+1}} > \sum_{x=1}^{U_{v_n, v_{n+1}}(\Delta)} d^s_{v_n, \zeta}(v_n + x\Delta)$ **then**
9: Evaluate possible path to state $(q_\zeta, p_\zeta)^\zeta$ with edge $(q_{v_n}, p_{v_n})^{v_n} \rightarrow (q_\zeta, p_\zeta)^\zeta$.
10: **else**
11: $\mathbf{Q}_{\lim}(p^+_\zeta) \leftarrow q_\zeta - 1$, $\forall p^+_\zeta \geq p_\zeta$.
12: **end if**
13: **end for**
14: If $\zeta \neq v_N$ and $\mathbf{Q}_{\lim}$ is non-zero vector, increment ζ to next neighboring captured view and goto step 7.
15: If $n < N$, increment n and repeat step 2 to 14.

We now describe a bit allocation algorithm, shown in Algorithm 1, exploiting the lemmas derived in previous section to reduce complexity from the full trellis. The basic idea is to construct a subset of the trellis on the fly as the algorithm is executed, and to try to rule out as many states and edges in the constructed trellis subset as early as possible. Starting from the left-side of trellis, for each captured view v_n, using computed sub-paths to states $(q_{v_n}, p_{v_n})^{v_n}$'s with sub-path Lagrangian costs $\Phi_{v_n}(q_{v_n}, p_{v_n})$'s[6], we first eliminate states with larger Lagrangian costs Φ_{v_n}'s and coarser texture quantization levels $q^+_{v_n}$'s than a minimum state $(q^*_{v_n}, p_{v_n})$, given p_{v_n}. Same procedure is applied for the depth quantization levels $p^+_{v_n}$'s given fixed q_{v_n}. These sub-optimal states are eliminated due to lemma 1.

[6]Lagrangian costs $\Phi_{v_1}(q_{v_1}, p_{v_1})$'s of first coded view v_1 are simply $\phi_{v_1}(q_{v_1}, p_{v_1})$'s.

In step 4, for each survived state $(q_{v_n}, p_{v_n})^{v_n}$ of view v_n, we *evaluate* all forward sub-paths to states $(q_{v_{n+1}}, p_{v_{n+1}})^{v_{n+1}}$'s of the next captured view v_{n+1}. By "evaluate", we mean comparing the sum of $\Phi_{v_n}(q_{v_n}, p_{v_n})$ and $\phi_{v_{n+1},v_n} + d^s_{v_n,v_{n+1}}$ to the cost of the best sub-path to $(q_{v_{n+1}}, p_{v_{n+1}})^{v_{n+1}}$ to date, $\Phi_{v_{n+1}}(q_{v_{n+1}}, p_{v_{n+1}})$, for each state $(q_{v_{n+1}}, p_{v_{n+1}})^{v_{n+1}}$. If the former is smaller, $\Phi_{v_{n+1}}(q_{v_{n+1}}, p_{v_{n+1}})$ will be updated accordingly.

In step 5, for each survived state $(q_{v_n}, p_{v_n})^{v_n}$, we next evaluate feasible edges to states $(q_\zeta, p_\zeta)^\zeta$'s of captured views ζ's, $\zeta > v_{n+1}$. Feasible edges are ones that satisfy $\phi_{v_{n+1},v_n} + d^s_{v_n,v_{n+1}} > \sum_{x=1}^{U_{v_n,v_{n+1}}(\Delta)} d^s_{v_n,\zeta}(v_n + x\Delta)$. We stop when there are no more forward feasible edges. We can identify the shortest end-to-end path by finding the minimum cost state $(q_{v_N}, p_{v_N})^{v_N}$ of view v_N and tracing it back to view v_1.

VI. Experimentation

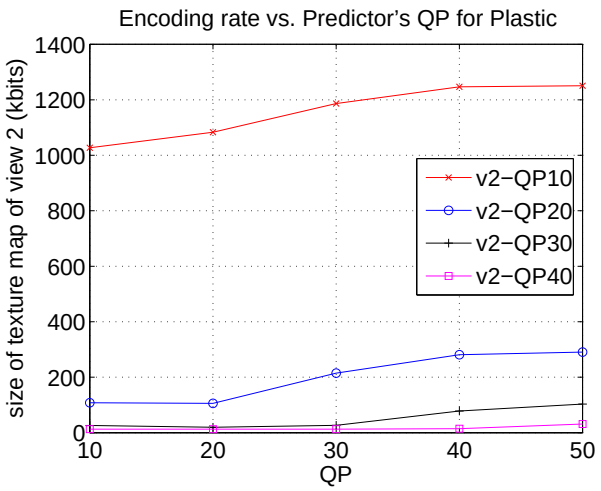
We start the experimentation section by providing empirical evidence to justify our assumption of monotonicity in predictor's quantization level and distance. We then evaluate the quality of our estimate of intermediate synthesized view distortion using our proposed cubic distortion model. Finally, we show the effectiveness of our proposed bit allocation strategy.

For test data sets, we used four Middlebury still image sequences [40], `Plastic`, `Lampshade1`, `Rocks2` and `Bowling2` of size 1270×1110, 1300×1110, 1276×1110 and 1330×1110, respectively. We assumed captured camera views were $\{1, 2, 3, 4, 5\}$, and desired constructed view spacing Δ at the decoder was 0.05. For all our experiments, we used H.264 JM16.2 [44] video codec to encode texture and depth maps (texture and depth maps were encoded independently from each other). The available quantization levels for both texture and depth maps were $\mathcal{Q} = \mathcal{P} = \{25, 30, \ldots, 50\}$. Rate controls were disabled in JM16.2, and software modifications were made so that a particular quantization level can be specified for each individual frame.

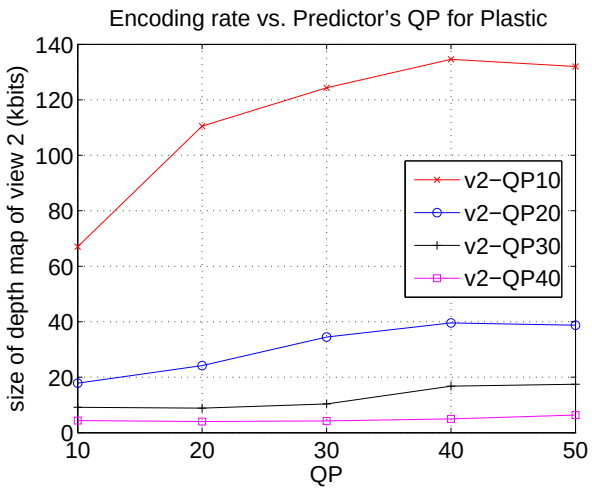
For DIBR virtual view synthesis at the decoder, we used a simple algorithm presented in [38]. A synthesized view is obtained by projecting two (left and right) captured anchor views to the chosen synthesis viewpoint such that the texture map pixels are warped according to the disparity information recorded in the intensities of the depth map pixels captured at the same viewpoint. The pixels projected from the two anchor views to the same coordinate at the synthesis viewpoint are blended using a view-dependent linear weighted sum of the two pixel intensities, where the weight factors are proportional to the proximity of the source anchor view. At the synthesized view pixel coordinates, where one of the two projections is unavailable due to occlusion or out-of-frame pixel location, the pixels are synthesized using the single available intensity, whereas the pixels unavailable from any of the anchor views are filled in a post-processing in-painting or interpolation step.

A. Validation of Monotonicity Assumptions

We first provide empirical evidence to show that the assumption of monotonicity in predictor's quantization level and distance are indeed valid. Using the `Plastic` sequence, we first plotted the texture map coding rate of captured view 2, using captured view 1 as predictor, as function of view 1's quantization level (quantization level of view 2 was kept constant for each curve). In Fig. 5(a), we see that for all curves, texture map coding rate of view

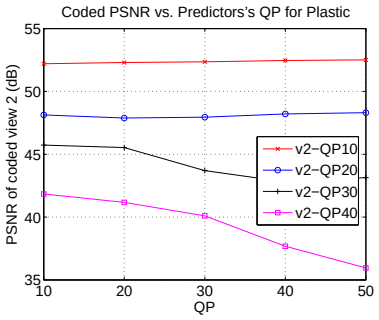


(a) Texture map coding rate vs. predictor's QP

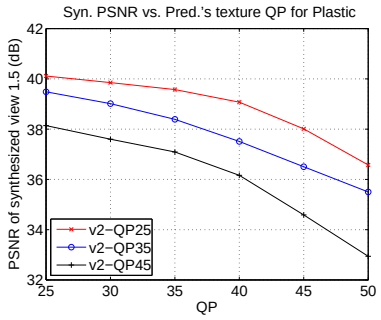


(b) Depth map coding rate vs. predictor's QP

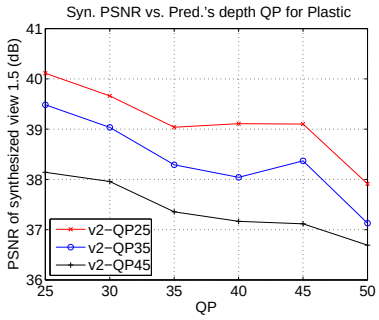
Fig. 5. Encoding rate of texture and depth map of coded view 2 are plotted against predictor view 1's quantization level for `Plastic` sequence. Each curve is generated using constant quantization level for view 2.



(a) Coded PSNR vs. pred.'s QP



(b) Syn. PSNR vs. pred.'s texture QP

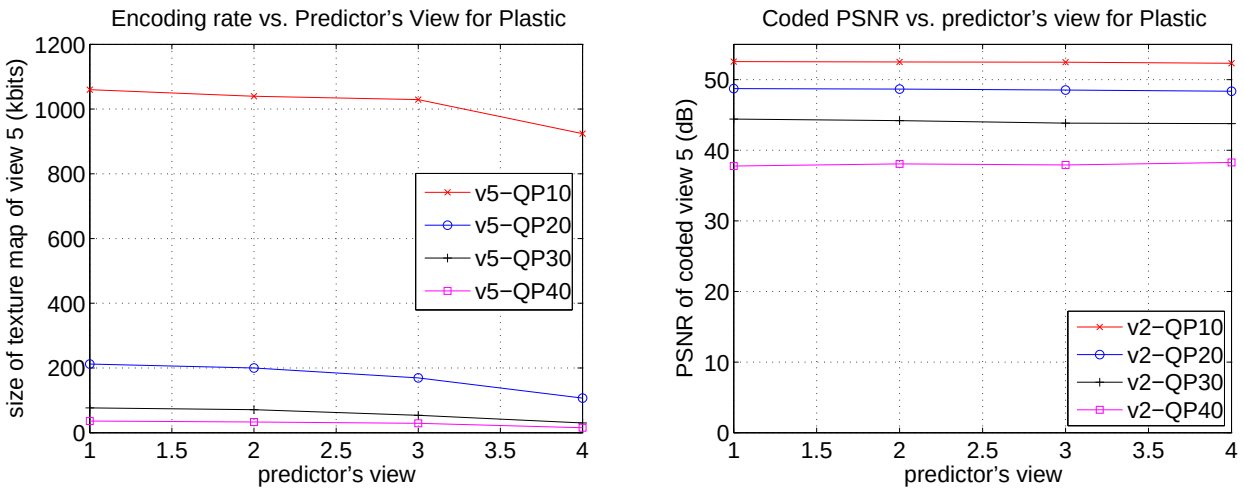


(c) Syn. PSNR vs. pred.'s depth QP

Fig. 6. Visual quality of coded view 2 and of synthesized view 1.5 are plotted against predictor view 1's quantization levels for `Plastic` sequence. Each curve is generated using constant quantization level(s) for coded view 2.

2 increased as view 1's quantization level became larger (coarser). In Fig. 5(b), we see the same trend for depth map coding rate of view 2 as function of predictor view 1's quantization level. This agrees with our intuition that a coarsely quantized predictor (view 1) creates a poor prediction for the predictee (view 2), and hence to maintain the desired quality at the predictee (controlled by its quantization parameter), more bits must be spent.

We also plotted PSNR (Peak Signal-to-Noise Ratio, a common objective measure for image quality) of coded view 2 as function of predictor view 1's quantization level in Fig. 6(a). We see that for all curves, PSNR either remained roughly constant, or decreased (distortion increased) as view 1's quantization level became coarser. This also agrees with our intuition that the image quality of the predictee (view 2) is mostly controlled by its quantization level, hence we expect no or small negative change in the predictee's visual quality as the quality of the prediction



(a) Texture map coding rate vs. predictor's view

(b) Coded view PSNR vs. predictor's view

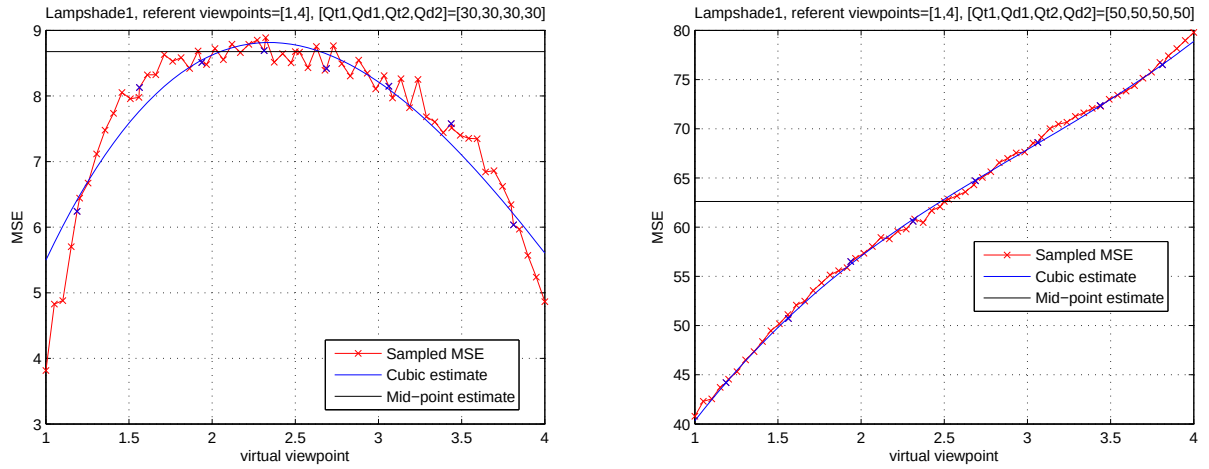
Fig. 7. Texture map encoding rate and Visual quality of coded view 5 are plotted against predictor's view for `Plastic` sequence. Each curve is generated using constant quantization level(s) for all coded views.

deteriorates. Since Lagrangian cost is a weighted sum of distortion and coding rate, given empirical evidence showing distortion and coding rate increase as a function of predictor's quantization level, we can conclude that our assumption of Lagrangian cost monotonicity of predictor's quantization level (15) is shown to be valid.

We also plotted PSNR of synthesized view 1.5 as a function of predictor view 1's *texture map* quantization level in Fig. 6(b), and as a function of predictor view 1's *depth map* quantization level in Fig. 6(c) . (Quantization levels of view 1's other map and view 2's texture and depth map were kept constant for each curve.) For Fig. 6(b), we see clearly that for all curves, PSNR decreased as view 1's texture map quantization level became coarser. Fig Fig. 6(c), though the curves are not strictly decreasing at all points, the similar downward trend is undeniable. This agrees with our intuition that a poorer predictor leads directly to a poorer synthesized view. Hence we can conclude that our assumption of synthesized distortion monotonicity of predictor's quantization level (16) is justified.

To validate our assumption of monotonicity of predictor's distance, we first plotted texture map coding rate of view 5 as function of predictor's view in Fig. 7(a). (Quantization levels of predictor's and view 5's texture maps were kept at the same constant for each curve.) We see that as the predictor's view became closer, the texture map coding rate of view 5 became smaller. Though not shown, depth map coding rate of view 5 also showed the same behavior. This agrees with our intuition that a closer predictor provides better prediction, leading to a smaller coding rate.

In Fig. 7(b), we plotted the PSNR of coded view 5 as function of predictor's view. As discussed earlier, intuitively the quality of the predictee (view 5) is controlled mostly by its quantization parameter, so we expect almost no change in the predictee's visual quality as we move the predictor frame closer to the target frame. The experimental data does confirm our intuition. Given these evidences, we can conclude that the empirical evidence supports our



(a) Synthesized MSE vs. viewpoint for $QP = 30$

(b) Synthesized MSE vs. viewpoint for $QP = 50$

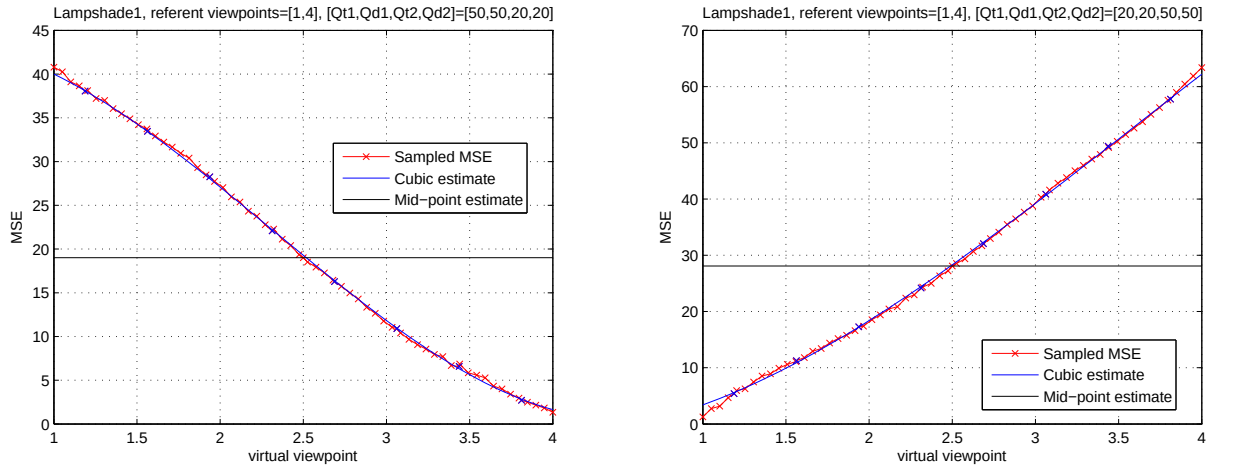
Fig. 8. Synthesized distortion is plotted against viewpoint location for different quantization levels for `Lampshade1` sequence. Cubic distortion model (blue), mid-point (black) and actual synthesized distortion at 0.05 view spacing are shown.

assumption of Lagrangian cost monotonicity of predictor's distance (17).

B. Accuracy of Cubic Distortion Model

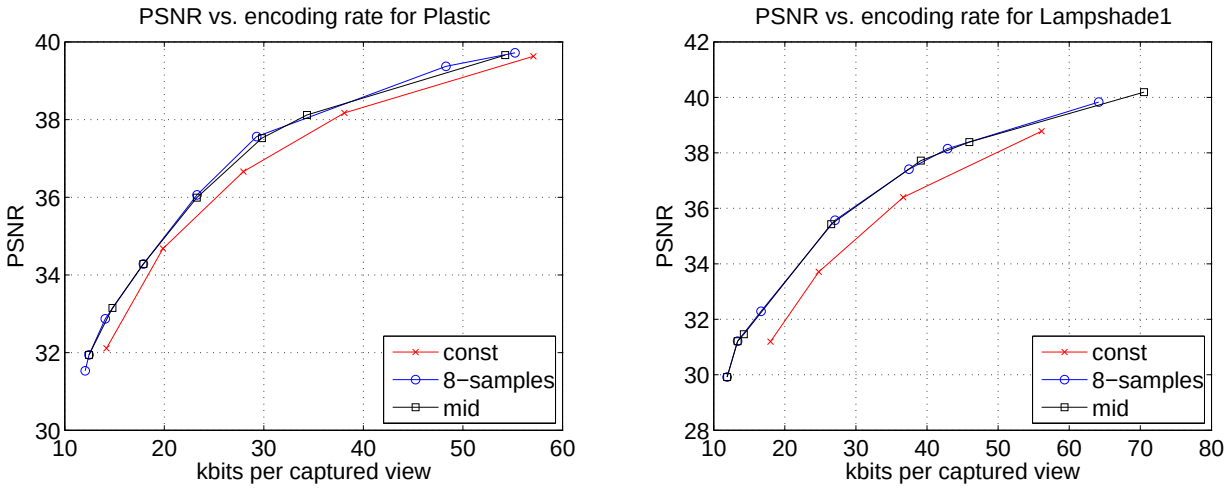
To demonstrate the accuracy of our proposed cubic synthesized distortion model, in addition to Fig. 2, we plotted the synthesized view distortion interpolated using coded views 1 and 4 of the `Lampshade1` sequence as function of viewpoint location in Fig. 8(a) and (b) for two different sets of quantization levels: $QP = 30$ in (a) and $QP = 50$ in (b). The actual computed MSE of the synthesized view, as compared to the "clean" synthesized view when interpolated using two nearest uncompressed captured views, is shown in red. The constructed cubic distortion model is shown in blue. We first observe that, though there was a non-negligible noise term $n(v)$ in the measured MSE due to secondary effects such as occlusion, rounding, etc., there is undeniably a trend that is either concave (i.e., increased, then decreased distortion) or strictly increasing/decreasing. Second, we see visually that for both plots, our proposed distortion model did track this trend of synthesized distortion as function of viewpoint, demonstrating the accuracy of our model. For Fig. 8(a), when the depth map quantization levels are relatively fine, the distortion curve is close to parabolic in shape, as predicted in Section III.

We also plotted the synthesized distortion as function of viewpoint location when the quantization levels of the left and right coded views were different. In Fig. 9(a), quantization level for the left view was set coarser than the right, while in Fig. 9(b), quantization level for the right view was set coarser than the left. In both cases, we see that our proposed cubic distortion model tracked the trend of measured MSE accurately, showing the accuracy of our model.



(a) Synthesized MSE vs. viewpoint for $QP_1 > QP_4$ (b) Synthesized MSE vs. viewpoint for $QP_1 < QP_4$

Fig. 9. Synthesized distortion is plotted against viewpoint location for different quantization levels for `Lampshade1` sequence. Cubic distortion model (blue), mid-point (black) and actual synthesized distortion at 0.05 view spacing are shown.



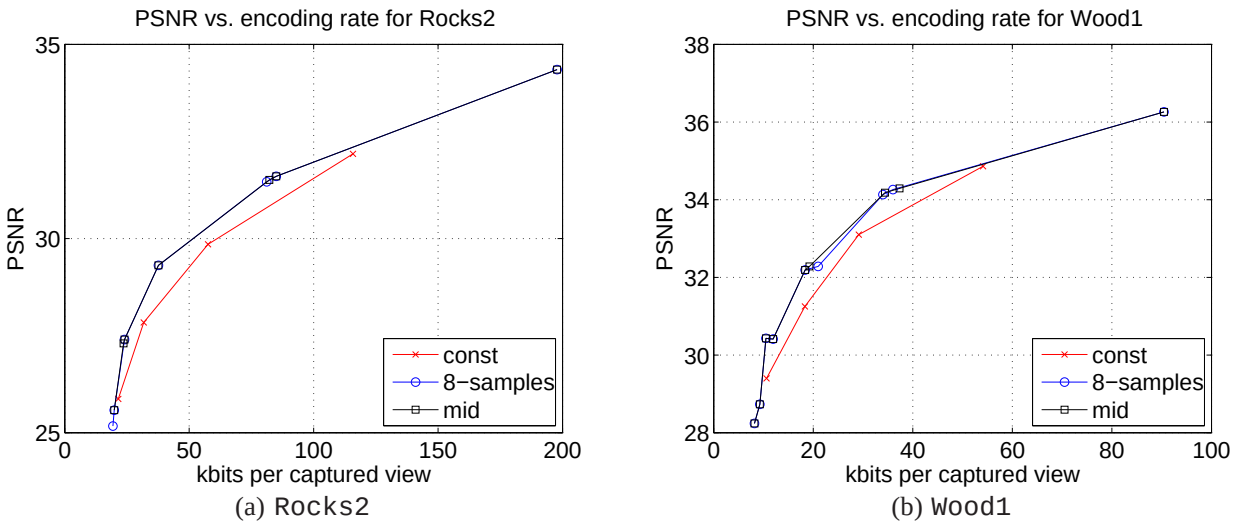
(a) `Plastic` (b) `Lampshade1`

Fig. 10. Performance Comparison between Optimal and Constant-QP Coded View and Quantization Level Selection Schemes

C. Comparing RD Performance of Bit Allocation Strategies

We tested the performance of our proposed bit allocation strategy using both sampling methods discussed in Section III-B, S samples to construct the cubic model (`8-samples`) and a single mid-point sample (`mid`) to bound average synthesized distortion, for the four Middlebury image sequences. We also tested a simple constant-QP scheme `const` that selects all captured views $\mathcal{N}$ for coding, i.e., $\mathcal{J} = \mathcal{N}$, and assigns a constant quantization level to all pixel and depth maps of coded views.

In Fig.10, we see the performance of the bit allocation strategies for `Plastic` and `Lampshade1`, shown as



(a) `Rocks2` (b) `Wood1`

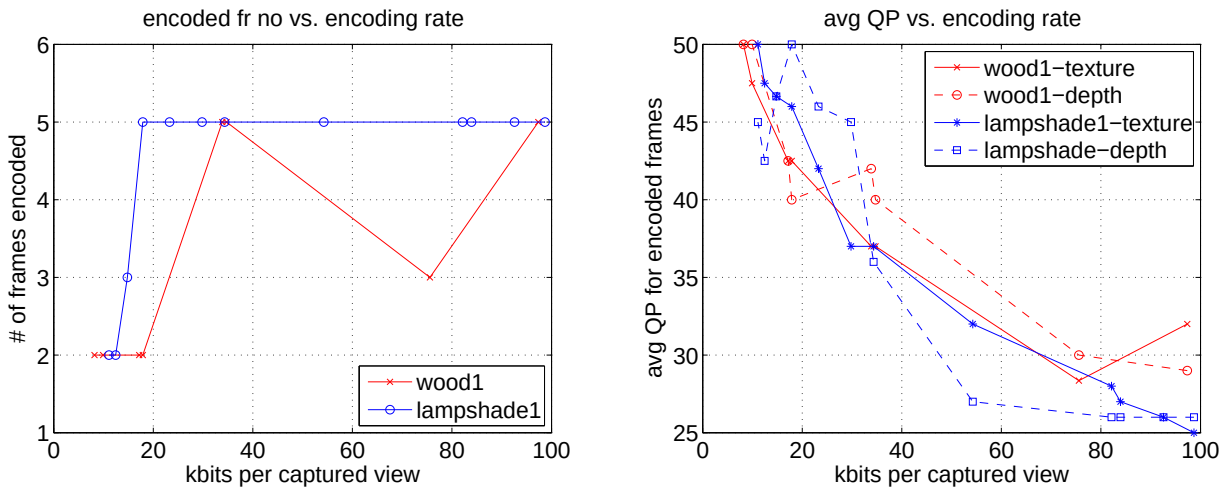
Fig. 11. Performance Comparison between Optimal and Constant-QP Coded View and Quantization Level Selection Schemes

PSNR versus bitrate per captured view (including both pixel and depth maps). First, we see that both `8-samples` and `mid` have better RD performance than `const` over all bitrate regions—by up to 0.80dB and 1.51dB for `Plastic` and `Lampshade1`, respectively. This shows that correct selection of quantization levels per frame is important. Second, as bitrate decreased, `8-samples` and `mid` selected fewer captured views for coding and relied instead on decoder's view synthesis of captured views (four left-most points in `Plastic` and three left-most points in `Lampshade1` of `8-samples` represented selections of uncoded views). This is also the region where `8-samples` and `mid` out-performed `const` the most, hence selection of captured views for coding is also important for best RD performance. Finally, we observe that the RD performance differences between `8-samples` and `mid` are very small. Hence for complexity reasons, the less complex `mid` would be more preferable than `8-samples` in practice.

When generating RD curves using `8-samples`, we tracked the amount of computation performed using our solution search strategy compared to a full trellis search approach. Essentially, we counted the number of times local Lagrangian cost $\Phi_{v_n}(q_{v_n}, p_{v_n})$ is potentially updated in both search strategies, where in `8-samples` evaluations are avoided when nodes and edges are pruned during search in the 3D trellis. We found the computation savings ranged from 80% to 99% , with the maximum saving occurring at the right-most RD point.

In Fig. 11, we see the RD performance of competing bit allocation schemes for sequences `Rocks2` and `Wood1`. We see that the general trend is similar to the earlier two sequences; i.e., performance gain of our bit allocation strategy `8-samples` and `mid` over constant-QP scheme `const` is more pronounced at low bitrate, when captured views are skipped. The maximum gain in PSNR for these two sequences are 1.05dB and 0.95dB, respectively. We see also that the two sampling methods `8-samples` and `mid` produced very similar results.

To take a closer look at the solutions generated by our algorithm `mid`, we constructed Fig. 12. First, Fig. 12(a)



(a) Number of Encoded frames vs. encoding rate (b) Average QP vs.encoding rate

Fig. 12. Number of captured frames selected for encoding and average QP for selected encoded frames as function of encoded bitrate for `wood1` and `lampshade1`.

shows the number of captured views selected by `mid` for encoding as function of the encoding bitrate for image sequence `wood1` and `lampshade1`. We observe that at lower bitrate region, fewer number of views were selected for encoding. This is intuitive, since fewer number of encoded views leads to smaller bit expenditure in general. This is also the region where `mid` out-performed `const` the most. This shows that when bitrate is more a concern than synthesized view quality, selecting the right subset of captured frames for encoding is very important for good RD performance.

In Fig. 12(b), we plotted the average QP of the selected encoded views in solutions generated by `mid` as function of bitrate for `wood1` and `lampshade1`. We see that as bitrate decreased, the average QP became coarser for both texture and depth maps, which is intuitive. We see also that in general, `mid` deemed texture maps as slightly more important than depth maps, resulting in finer QP for texture than depth in most generated solutions. Finally, we observe that the depth map QP curves are not strictly monotonic; i.e., there are cases when the QP became *finer* as the bitrate decreased. These correspond to solutions where the texture map became coarser, or the number of captured views decreased. Hence, we can conclude that a strictly monotonic search to derive one solution from a neighboring one on the RD curve would not be RD-optimal.

VII. Conclusion

Towards the goal of finding a compact multview image representation, one that takes advantage of both efficient texture and depth map coding tools at encoder and view synthesis tool using depth-image-based rendering (DIBR) at decoder, in the paper we presented an algorithm to select captured views for coding and quantization levels of corresponding texture and depth maps in a rate-distortion (RD) optimal manner. We first derive a cubic distortion model that models synthesized view distortion between two coded views. We then show that using monotonicity in

predictor's quantization level and distance, search complexity can be drastically reduced without loss of optimality. Experiments show that our selection scheme outperformed a heuristic scheme by up to 1.5dB in PSNR for the same bitrate.

APPENDIX

We provide proofs for the two lemmas in Section V-D here.

Proof of Lemma 1: We prove by contradiction. Suppose shortest sub-path up to state $(q^+_{v_n}, p_{v_n})^{v_n}$, $q^+_{v_n} > q^*_{v_n}$, is part of an end-to-end shortest path. That means captured view v_n is a coded view; let $j_k = v_n$. If we replace sub-path to $(q^+_{j_k}, p_{j_k})^{j_k}$ with sub-path to $(q^*_{j_k}, p_{j_k})^{j_k}$, synthesized intermediate views to the right of j_k and coded view j_{k+1} that depend on view j_k's texture map will have no larger synthesized view distortion $d^s_{j_k,j_{k+1}}$ or Lagrangian cost ϕ_{j_{k+1},j_k}, if $q^*_{j_k}$ is used instead of $q^+_{j_k}$, by monotonicity in predictor's quantization level (15) and (16). Given $\Phi_{j_k}(q^+_{j_k}, p_{j_k}) > \Phi_{j_k}(q^*_{j_k}, p_{j_k})$, we see that replacing sub-path to $(q^+_{j_k}, p_{j_k})^{j_k}$ with sub-path to $(q^*_{j_k}, p_{j_k})^{j_k}$ will yield strictly lower Lagrangian cost. A contradiction. □

Proof of Lemma 2: We prove by contradiction. Suppose an optimal end-to-end path includes edge $(q_{v_n}, p_{v_n})^{v_n} \rightarrow (q_\xi, p_\xi)^\xi$. If we replace it with two edges $(q_{v_n}, p_{v_n})^{v_n} \rightarrow (q_{v_n}, p_{v_n})^{v_{n+1}} \rightarrow (q_\xi, p_\xi)^\xi$, the cost of traversing state $(q_{v_n}, p_{v_n})^{v_{n+1}}$, considering intermediate synthesized views v's, $v_n < v < v_{n+1}$, and captured view v_{n+1}, is smaller than not traversing it by assumption. Moreover, Lagrangian cost of coded view ξ and distortion of synthesized views to the right of view v_n that predicted from view v_n will not increase predicting view v_{n+1} instead with same quantization levels due to monotonicity of predictor's distance (17) and (18). Hence a path using the two replacement edges will yield strictly lower cost. A contradiction. □

REFERENCES

[1] B. Wilburn, M. Smulski, K. Lee, and M. A. Horowitz, "The light field video camera," in *Proc. Media Processors SPIE Electron. Imag.*, San Jose, CA, January 2002.

[2] T. Fujii, K. Mori, K. Takeda, K. Mase, M. Tanimoto, and Y. Suenaga, "Multipoint measuring system for video and sound—100 camera and microphone system," in *IEEE International Conference on Multimedia and Expo*, Toronto, Canada, July 2006.

[3] "Mpeg - technologies - introduction to multiview video coding," January 2008, iSO/IEC JTC 1/SC 29/WG 11 N9580.

[4] T. Chen, "Adaptive temporal interpolation using bidirectional motion estimation and compensation," in *IEEE International Conference on Image Processing*, Rochester, NY, September 2002.

[5] J. Zhai, K. Yu, J. Li, and S. Li, "A low complexity motion compensated frame interpolation method," in *IEEE International Symposium on Circuits and Systems*, Kobe, Japan, May 2005.

[6] P. Merkle, A. Smolic, K. Mueller, and T. Wiegand, "Multi-view video plus depth representation and coding," in *IEEE International Conference on Image Processing*, San Antonio, TX, October 2007.

[7] M. Tanimoto, T. Fujii, and K. Suzuki, "Multi-view depth map of Rena and Akko & Kayo," ISO/IEC JTC1/SC29/WG11 MPEG Document M14888, Oct. 2007.

[8] S. Gokturk, H. Yalcin, and C. Bamji, "A time-of-flight depth sensor—system description, issues and solutions," in *Conference on Computer Vision and Pattern Recognition Workshop (CVPRW)*, Washington, DC, June 2004.

[9] H.-Y. Shum, S.-C. Chan, and S. B. Kang, *Image-Based Rendering*. Springer, 2007.

[10] Y. Morvan, D. Farin, and P. H. N. de With, "Multiview depth-image compression using an extended H.264 encoder," in *Advanced Concepts for Intelligent Vision Systems, Lecture Notes in Computer Sciences*, vol. 4678, 2007, pp. 675–686.

[11] G. Cheung and V. Velisavljević, "Efficient bit allocation for multiview image coding & view synthesis," in *IEEE International Workshop on Multimedia and Signal Processing*, St. Malo, France, October 2010.

[12] J. Shade, S. Gortler, L. He, and R. Szeliski, "Layered depth images," in *ACM SIGGRAPH*, New York, NY, September 1998.

[13] M. Levoy and P. hanrahan, "light field rendering," in *ACM SIGGRAPH*, New Orleans, LA, August 1996.

[14] S. Gortler, R. Grzeszczuk, R. Szeliski, and M. Cohen, "The lumigraph," in *ACM SIGGRAPH*, New Orleans, LA, August 1996.

[15] P. Debevec, C. Taylor, and J. Malik, "Modeling and rendering architecture from photographs," in *ACM SIGGRAPH*, New Orleans, LA, August 1996.

[16] H.-Y. Shum, S. B. Kang, and S.-C. Chan, "Survey of image-based representations and compression techniques," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no.11, November 2003, pp. 1020–1037.

[17] M. Magnor, P. Ramanathan, and B. Girod, "Multi-view coding for image-based rendering using 3-D scene geometry," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no.11, November 2003, pp. 1092–1106.

[18] S. Seitz and C. Dyer, "view morphing," in *ACM SIGGRAPH*, New Orleans, LA, August 1996.

[19] W. Mark, L. McMillan, and G. Bishop, "Post-rendering 3D warping," in *Symposium on Interactive 3D Graphics*, New York, NY, April 1997.

[20] M. Flierl, A. Mavlankar, and B. Girod, "Motion and disparity compensated coding for multiview video," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no.11, November 2007, pp. 1474–1484.

[21] P. Merkle, A. Smolic, K. Muller, and T. Wiegand, "Efficient prediction structures for multiview video coding," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no.11, November 2007, pp. 1461–1473.

[22] T.-Y. Chung, I.-L. Jung, K. Song, and C.-S. Kim, "Multi-view video coding with view interpolation prediction for 2D camera arrays," *Journal of Visual Communication and Image Representation*, vol. 21, no. 5-6, pp. 474–486, July-August 2010.

[23] Y. Morvan, D. Farin, and P. H. de With, "Depth-image compression based on an R-D optimized quadtree decomposition for the transmission of multiview images," in *IEEE International Conference on Image Processing*, San Antonio, TX, September 2007.

[24] M. Maitre, Y. Shinagawa, and M. Do, "Wavelet-based joint estimation and encoding of depth-image-based representations for free-viewpoint rendering," in *IEEE Transactions on Image Processing*, vol. 17, no.6, June 2008, pp. 946–957.

[25] A. Smolic, K. Muller, K. Dix, P. Merkle, P. Kauff, and T. Wiegand, "Intermediate view interpolation based on multiview video plus depth for advanced 3D video systems," in *IEEE International Conference on Image Processing*, San Diego, CA, October 2008.

[26] J.-R. Ohm, E. Izquierdo, and K. Muller, "Systems for disparity-based multiple-view interpolation," in *IEEE International Symposium on Circuits and Systems*, vol. 5, Monterey, CA, 1998.

[27] Y. Liu, Q. Huang, S. Ma, D. Zhao, and W. Gao, "Joint video/depth rate allocation for 3D video coding based on view synthesis distortion model," in *Elsevier, Signal Processing: Image Communication*, vol. 24, no.8, September 2009, pp. 666–681.

[28] W.-S. Kim, A. Ortega, P. Lai, D. Tian, and C. Gomila, "Depth map distortion analysis for view rendering and depth coding," in *IEEE International Conference on Image Processing*, Cairo, Egypt, November 2009.

[29] ——, "Depth map coding with distortion estimation of rendered view," in *SPIE Visual Information Processing and Communication*, San Jose, CA, January 2010.

[30] Y. Shoham and A. Gersho, "Efficient bit allocation for an arbitrary set of quantizers," in *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 36, no.9, September 1988, pp. 1445–1453.

[31] K. Ramchandran, A. Ortega, and M. Vetterli, "Bit allocation for dependent quantization with applications to multiresolution and MPEG video coders," in *IEEE Transactions on Image Processing*, vol. 3, no.5, September 1994.

[32] J.-H. Kim, J. Garcia, and A. Ortega, "Dependent bit allocation in multiview video coding," in *IEEE International Conference on Image Processing*, Genoa, Italy, September 2005.

[33] H. Song and C.-C. J. Kuo, "Rate control for low-bit-rate video via variable-encoding frame rates," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, no.4, April 2001, pp. 1051–8215.

[34] G. Liebl, M. Kalman, and B. Girod, "Variable frame skipping scheme based on estimated quality of non-coded frames at decoer for real-time block-based video coding," in *IEEE International Conference on Multimedia and Expo*, Taipei, Taiwan, July 2004.

[35] S. Liu and C.-C. J. Kuo, "Joint temporal-spatial bit allocation for video coding with dependency," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no.1, January 2005, pp. 15–26.

[36] G. Cheung and V. Velisavljević, "Efficient bit allocation for multiview image coding & view synthesis," in *IEEE International Conference on Image Processing*, Hong Kong, September 2010.

[37] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003.

[38] P. Merkle, Y. Morvan, A. Smolic, D. Farin, K. Muller, P. de With, and T. Wiegand, "The effects of multiview depth video compression on multiview rendering," in *Signal Processing: Image Communication*, vol. 24, 2009, pp. 73–88.

[39] C. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[40] "2006 stereo datasets," http://vision.middlebury.edu/stereo/data/scenes2006/.

[41] G. Cheung, W.-T. Tan, and C. Chan, "Reference frame optimization for multiple-path video streaming with complexity scaling," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no.6, June 2007, pp. 649–662.

[42] I. Daribo, C. Tillier, and B. Pesquet-Popescu, "Motion vector sharing and bit-rate allocation for 3D video-plus-depth coding," in *EURASIP: Special Issue on 3DTV in Journal on Advances in Signal Processing*, vol. 2009 (2009), January 2009.

[43] W. Li and B. Li, "Virtual view synthesis with heuristic spatial motion," in *IEEE International Conference on Image Processing*, San Diego, CA, October 2008.

[44] T. Wiegand, G. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no.7, July 2003, pp. 560–576.