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

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

***Abstract*—The encoding of both texture and depth maps of multiview images, captured by a set of spatially correlated cameras, is important for any 3-D 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 multiview images. More specifically, suppose we are given a coding tool to encode texture and depth maps at the encoder and 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 the 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 3-D trellis. Finally, we show that, using the assumptions of monotonicity in the predictor's quantization level and distance, suboptimal 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 1.5 dB. Moreover, the complexity of our scheme can be reduced by at least 80% over the full solution search.**

***Index Terms*— Bit allocation, depth-image-based rendering, 3-D image coding.**

I. Introduction

RECENT development of imaging technology has led to research in higher dimensional visual information processing beyond traditional 2-D images and single-view video, aiming at improving the user's visual experience and offering new media navigation functionalities to consumers. One notable example is the *multiview video* [1], where a scene of interest is captured by a large 2-D array of densely spaced time-synchronized cameras from different viewpoints [2]. Thus, the resulting captured data have a much larger number of dimensions compared with traditional media, i.e., pixel location (i,j) at time t from camera location (x,y). In this paper, we focus on the more constrained scenario where the scene of interest is *static*, and the capturing cameras are placed in a 1-D *horizontal array*. Hence, we can drop the temporal dimension t and the 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 simply as *multiview imaging* in the sequel.[1]

In a typical multiview imaging scenario, a sender creates and transmits a multiview representation, which is composed of viewpoint images taken by the aforementioned spatially correlated cameras, of a physical 3-D scene so that a receiver can construct images of the scene from viewpoints of his own choice 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 the 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 the *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 that, in general, is not true for multiview images, the interpolated quality tends to be poor.

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 encode *depth information* (distance between the camera and the physical object in the scene corresponding to each captured pixel) for some viewpoints. The depth can be estimated [7] or recorded by special

Manuscript received November 09, 2010; revised March 16, 2011; accepted May 02, 2011. Date of publication May 31, 2011; date of current version October 19, 2011. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Zhou Wang.

G. Cheung is with the National Institute of Informatics, Tokyo 101-8430, Japan (e-mail: cheung@nii.ac.jp).

V. Velisavljević is with the Deutsche Telekom Laboratories, 10587 Berlin, Germany (e-mail: vladan.velisavljevic@telekom.de).

A. Ortega is with the Department of Electrical Engineering, University of Southern California, Los Angeles, CA 90089 USA (e-mail: antonio.ortega@sipi.usc.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2011.2158230

[1] The analysis and the 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, the 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 in the video.

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

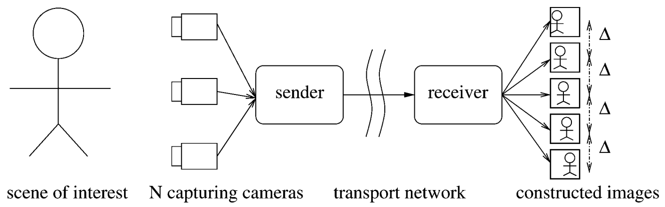


Fig. 1. Overview of a multiview imaging communication system. N cameras in a 1-D array capture images of a scene from different viewpoints. The sender selects a multiview representation, compresses it, and transmits it to the receiver. The receiver decodes the compressed images and, if depth maps are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

hardware [8]. A receiver can then synthesize additional intermediate views from received neighboring texture and depth maps using DIBR techniques such as 3-D warping [9], [10]. Conveying both texture and depth maps, which is called the *texture + depth* representation, for a large multiview image sequence to the receiver, however, means that a large amount of data must be transmitted. A natural resource allocation question hence arises, i.e., given a disparity-compensation-based encoder 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 encoding and select quantization levels of corresponding texture and depth maps of chosen captured views to minimize the 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 virtually choose 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 total 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 computationally expensive.

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 deduce model parameters using several sample synthesized views to estimate the distortion of all required intermediate views between two coded frames, or estimate the average distortion of all required intermediate views using a single image sample at the midpoint between the two coded frames. We note that, to the best of our knowledge, we are the first to estimate the DIBR-synthesized view distortion of a set of densely spaced viewpoints between two coded views using a small number of image samples.

Armed with our cubic distortion model, the second practical difficulty is to select an appropriate subset of captured views for encoding 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 it is relatively easy to interpolate intermediate views for the captured scene 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 of 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 (SP) in a specially designed 3-D trellis. Given that the state space of the trellis is enormous, we then show that, using lemmas derived from monotonicity assumptions in the predictor's quantization level and distance, suboptimal states and edges in the trellis can be pruned from consideration during the SP 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 1.5 dB. Moreover, our search strategy reduces at least 80% of the computations compared with the full solution search that examines every state and edge in the 3-D trellis.

This paper is organized as follows. After discussing related work in Section II, we derive the cubic distortion model used to estimate the 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.

[3] When the required view spacing and/or the available bit budget are 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.

II. Related Work

We divide the discussion of related work into four parts. We first motivate the value of "texture + depth" representation of a 3-D static scene studied in this paper. Having established that "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 3-D 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 [9] and [16] for excellent surveys of representations proposed in the literature. For a chosen representation, coding optimization can be then performed to trade off the reconstructed view distortion with the encoding rate. As a concrete example, Magnor *et al.* [17] considered two representations, i.e., VDTM and model-aided predictive coding. For VDTM, Magnor *et al.* [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 and then stitch texture patches on the model surface for image reconstruction. The tradeoff between the synthesized view distortion and the 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. The coding rate can be reduced via coarser quantization during residual coding.

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

The "texture + depth" format has several desirable properties. First, unlike the mesh-based geometrical model in [15], which can take hours to compute [17], depth maps can be either simply estimated using stereo-matching algorithms [20] or directly captured using time-of-flight cameras [8]. Second, depth maps can better handle a complicated scenery with multiple objects, whereas a mesh-based model often requires dense image sampling around the single object of interest for good construction quality. Finally, the "texture + depth" format is more adaptable to a dynamic scene where objects change positions and shapes over time. For these and other reasons, "texture + depth" is currently the chosen format for 3-D scene representation in the free-viewpoint video working group in the Motion Pictures Expert Group.

Given that the "texture + depth" format is an important representation for the 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 small spacing Δ is minimized. We believe that 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 the "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 the efficient representation of multiview images and video, novel coding tools and frame structures for texture map encoding [21]–[23] have been proposed in order to exploit inter-view correlation for coding gain. Similarly, new coding algorithms specifically tailored for depth maps [24], [25] 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, i.e., 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 paper by solving the following key problems: 1) how to estimate distortions of a large number of synthesized intermediate views between two coded frames at the encoder at low complexity; and 2) 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, i.e., our bit allocation strategy can be executed no matter which of the aforementioned tools are chosen for texture and depth maps encoding.

With the popularity of the texture + depth representation for the multiview images/video [6], enabling the DIBR-based view synthesis at the decoder using received texture and depth maps, new 3-D warping algorithms [9], [10] have been recently proposed in the literature. Virtual view interpolation has also been a useful tool for 3-D video systems [26]; several interpolation methods based on disparity techniques have been studied in [27]. Instead of developing new view-synthesis tools, our goal is to find the RD-optimal bit allocation, given the chosen coding tool at the encoder and the DIBR-based view-synthesis tool at the decoder.

C. Synthesized Distortion Model and Analysis

There has been work [28]–[30] studying the relationship between synthesized view distortion and lossy compression of the depth map. Because the distortion of depth maps creates geometric errors that ultimately affect synthesized view constructions, Kim *et al.* [29], [30] proposed new metrics based on the synthesized view distortion (instead of the depth-map distortion) for the mode selection at a block level during H.264 encoding of depth maps. Our paper is different in which 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 encoding for a given desired RD tradeoff.

For a two-view-only video sequence, Liu *et al.* [28] constructed a theoretical view-synthesis distortion model and derived two quantization parameters, i.e., one for all texture maps and one for all depth maps, which minimize the theoretical distortion. In contrast, our proposed bit allocation scheme selects quantization parameters for individual texture and depth maps in a multiview image sequence. Selecting one quantization parameter for every frame (rather than one for a large group of frames as done in [28]) means that 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 [28], which requires the 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 images 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 the 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 [31] and dependent [32] quantizers have been studied for single-view video coding. More recently, Kim *et al.* [33] have extended the trellis-based optimization technique in [32] to MVC, where texture maps of different frames can be coded using different quantization parameters. Kim *et al.* [33] did not consider the view synthesis when allocating bits to texture maps, whereas our paper considers the bit allocation for two types of resource, i.e., texture and depth maps, for the chosen subset of captured views for encoding, such that the resulting distortion of both encoded and synthesized views at the decoder is minimized.

The most similar prior research to our paper is the work on the bit allocation for the single-view video with frame skip [34]–[36], which studies the problem of selecting a subset of captured frames in a video to code at an optimal amount of allocated bits. The frames skipped at the encoder are interpolated at the decoder using the 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 naturally leading to a 3-D trellis, to be discussed in Section IV). Second, the 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 [34]–[36].[4]

This paper extends our previous work [11], [37] on bit allocation among texture and depth maps for DIBR as follows. In [37], 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 distortions of all Δ-spaced synthesized intermediate views between two coded frames. Furthermore, we validate our monotonicity assumption on the 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 are a viable multiview representation. The optimization proposed in [11] has high complexity, however. In this paper, we instead focus on the case when Δ is small, so that synthesizing all required intermediate views at the encoder and encoding them require too many bits and hence is not a viable option. By excluding this possibility, the search strategy presented here is much simpler than that in [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 the 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.

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 distortions 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 $d^s_{v_i,v_j}(v)$ at intermediate viewpoints v, $v_i < v < v_j$, i.e.,

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

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

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

[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 the 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.

of $d^s_{v_i,v_j}$ at the sender requires the DIBR view synthesis of many images at many v values. Furthermore, $d^s_{v_i,v_j}$ 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 the 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 the left coded view v_i and the right coded view v_j. For simplicity of derivation, we assume that $v_i = 0$ 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 [38]. 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 the depth-map distortion at view 0. In other words, $g_0(v)$ is the *offset* in the number of (horizontal) pixels away from the true corresponding pixel coordinate (k, y) in the left texture map, due to the left depth-map distortion, resulting in the erroneous pixel coordinate $(k + g_0(v), y)$ instead. In [29], it is shown that $g_0(v)$ linearly grows with the view location v, i.e., $g_0(v) = b_0 v$, $b_0 > 0$.

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

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

where $w_0(k)$ is a zero-mean Gaussian variable with variance σ_0^2. One can argue that the 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 the view synthesis. The expectation of the resulting squared error is

$$\begin{aligned} d_0^s(g) &= E\left[\left|X_0(k+g) - X_0(k)\right|^2\right] \\ &= E\Big[\,\big|\rho^g X_0(k) + \rho^{g-1} w_0(k) + \rho^{g-2} w_0(k+1) \\ &\qquad + \cdots + w_0(k+g-1) - X_0(k)\big|^2\,\Big] \\ &= E\left[\left|(\rho^g - 1)X_0(k) + \sum_{t=1}^{g} \rho^{g-t} w_0(k+t-1)\right|^2\right] \\ &= (\rho^g - 1)^2 E\left[X_0(k)^2\right] + \sigma_0^2 \sum_{t=1}^{g} \rho^{2(g-t)} \leq (g+1)\sigma_0^2 \end{aligned}$$

where $E[X_0(k)^2] = R_0(0) = \sigma_0^2$ is 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, i.e., $d_0^s(g_0(v)) = d_0^s(v) = (b_0 v + 1)\sigma_0^2$. 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 a typical DIBR view synthesis, pixel $Y(v)$ in the 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 weights $(1-v)$ and (v) linearly depend 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 the 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 $d^s_{0,1}(v) = |\hat{Y}(v) - Y(v)|^2$ in the synthesized pixel due to the distortion in the left and right depth maps can be derived as follows:

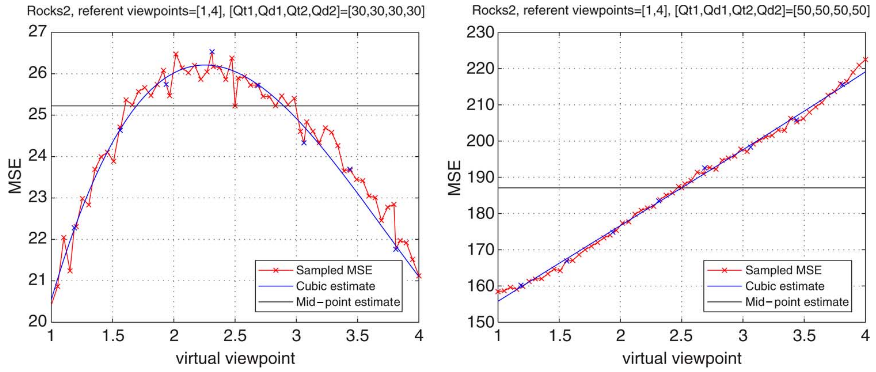
$$\begin{aligned} d^s_{0,1}(v) &= E\left[\left|\hat{Y}(v) - Y(v)\right|^2\right] \\ &= E\Big[\,\left|(1-v)X_0\left(k + g_0(v)\right) + (v)X_1\left(l + g_1(v)\right)\right. \\ &\qquad \left. -(1-v)X_0(k) - (v)X_1(l)\right|^2\,\Big] \\ &= E\Big[\,\left|(1-v)\left(X_0\left(k + g_0(v)\right) - X_0(k)\right)\right.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

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 is equal to zero, and we have the following 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 that the maximum distortion occurs at mid-



(a) (b)

Fig. 2. Synthesized distortion is plotted against viewpoint location for different quantization levels for the Rocks2 sequence [41]. (Blue) Cubic distortion model, (black) midpoint, and actual synthesized distortion at 0.05 view spacing are shown. Synthesized MSE versus viewpoint for (a) $\mathrm{QP} = 30$. and (b) $\mathrm{QP} = 50$.

point $v = 1/2$. We can hence conclude the following: if distortions in the left and right depth maps are not severe, then DIBR will be performed using the corresponding pixels in the left and right texture maps of the same object for the majority of pixels in the synthesized view, and the resulting distortion is quadratic. This is what was experimentally observed in [39] as well. If distortions in the left and right depth maps are severe enough that DIBR erroneously uses pixels of different objects for interpolation for the majority of pixels in the synthesized view, then the distortion becomes cubic.

Note that, in addition to (4), there are secondary nonlinear effects on the synthesized distortion $d^s_{v_i,v_j}(v)$ due to the following: 1) occlusion of different spatial regions with respect to viewpoint v determined by complex scene geometry; 2) pixel coordinate rounding operations used in the view synthesis (i.e., a 3-D-warped point is usually displayed at the nearest integer pixel location in the synthesized view); and 3) statistical discrepancies in texture maps, as previously discussed. We consider these effects secondary and instead focus 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 the Cubic Distortion Model

Although we have concluded that the appropriate distortion model as a function of the intermediate view v is a cubic function, we still need to find coefficients c_i's that characterize the 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, i.e., synthesize a small number of images at intermediate views v_k between v_i and v_j and calculate corresponding distortions $d^s_{v_i,v_j}(v_k)$, so that, using samples (v_k, d^s_k), 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 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

$$\underbrace{\begin{bmatrix} 1 & v_1 & v_1^2 & v_1^3 \\ 1 & v_2 & v_2^2 & v_2^3 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & v_S & v_S^2 & v_S^3 \end{bmatrix}}_{\mathbf{V}} \underbrace{\begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix}}_{\mathbf{c}} = \underbrace{\begin{bmatrix} d^s_1 \\ d^s_2 \\ \vdots \\ d^s_S \end{bmatrix}}_{\mathbf{d}^s} + \underbrace{\begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_S \end{bmatrix}}_{\mathbf{e}}. \tag{6}$$

By optimal, we mean coefficients c_i that lead to the smallest squared errors $\mathbf{e}$ possible. Using linear regression [40], optimal c_i values can be simply calculated as

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

where $\mathbf{V}^+$ is the Moore–Penrose pseudoinverse 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, i.e., $\tilde{d}^s_{v_i,v_j}$, 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 a 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 [40] 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 the computation, in the second sampling method, we only sample

[5] The 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, particularly when the capturing cameras are physically very close to each other.

at the midpoint $(v_i + v_j)/2$ 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.,

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

As previously discussed, if distortions in the left and right depth maps are small, then we expect a quadratic function with the peak at the midpoint, and this midpoint sampling method captures the maximum distortion. If distortions in the left and right depth maps are very large, then this midpoint 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 in general 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.

IV. Formulation

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 1-D-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 the quantization level q_{j_k} and p_{j_k}, respectively. q_{j_k} and p_{j_k} take on discrete values from the 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 that inter-view differential coding is used for coded views as done in [22]. That means there exists dependence 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, i.e., frames 3 and 4 in Fig. 3, is coded as a P-frame using the previous coded view j_{k-1} as predictor for disparity compensation. Each uncoded or intermediate view depends on two neighboring coded views.

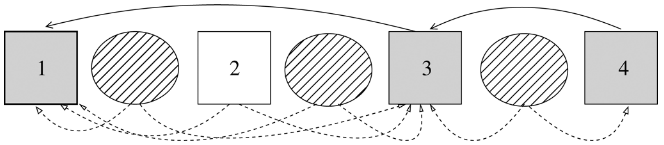


Fig. 3. Example of multiview image sequence. (Solid arrows) Coding dependences among (gray) coded views. (Dotted arrows) View-synthesis dependences between (patterned) an intermediate view and (gray) two neighboring coded views and between (white) an uncoded view and (gray) two neighboring coded views. 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.

A. Signal Distortion

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

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

which states that distortion $d^c_{j_1}$ of the starting viewpoint j_1 (coded as an I-frame) depends only on its own texture quantization level q_{j_1}, whereas 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 [32] 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 dependences to $q_{j_{k-1}}$ only is a good first-order approximation, as done in previous works such as in [42].

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 $\mathbf{p} = [p_{j_1}, \ldots, p_{j_K}]$, i.e.,

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

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 other words, the 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, i.e., R^c and R^s, respectively, as follows:

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

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

Equation (12) states that the encoding rate for the texture map of a coded view, i.e., $r^c_{j_k}$, 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, i.e., $r^s_{j_k}$, depends on both the texture- and depth-map quantization levels q_{j_k} and p_{j_k}, respectively, and its predictor's texture- and depth-map levels $q_{j_{k-1}}$ and $p_{j_{k-1}}$, respectively. Note that, although we encode depth maps independently from texture maps in the experiments in 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 [43]. Our formulation is sufficiently general to include the case when depth maps are differentially coded using texture maps as predictors.

C. RD Optimization

Given the aforementioned formulation, the optimization we are interested in is to find the coded-view indexes $\mathcal{J} \subseteq \mathcal{N}$ and the associated texture and depth quantization vectors $\mathbf{q}$ and $\mathbf{p}$, respectively, such that the Lagrangian objective L_λ is minimized for a given Lagrangian multiplier $\lambda \geq 0$, i.e.,

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

For clarity of later presentation, we will, in addition, define the *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 the 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-map 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 demonstrate how an optimal solution to (14) corresponds to the SP in a specially constructed 3-D 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 the 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 and Viterbi Algorithms

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

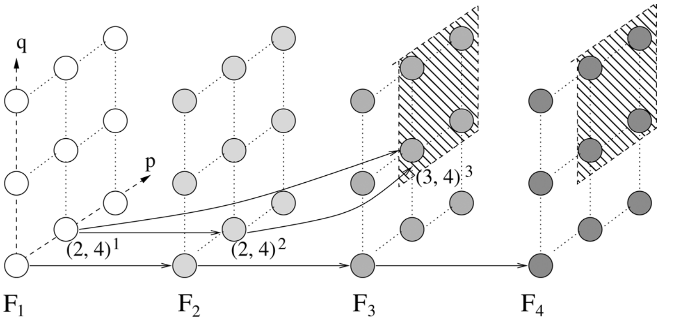


Fig. 4. Optimization of 3-D Trellis.

We can construct a trellis, e.g., the one corresponding to the earlier example is shown in Fig. 4, for the selection of coded-view indexes $\mathcal{J}$ and 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 $\phi_{v_1}(q_{v_1}, p_{v_1})$ for different level pairs $(q_{v_1}, p_{v_1})^{v_1}$. Each directed edge from 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 the 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 state $(q_{v_1}, p_{v_1})^{v_1}$ in the first plane to state $(q_{v_n}, p_{v_n})^{v_n}$ in a further-away plane of the 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 that the captured views 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 leftmost plane to any state in the rightmost plane, found using VA, corresponds to the optimal solution to (14). However, the number of states and edges in the trellis alone are prohibitively large, i.e., $O(|\mathcal{Q}||\mathcal{P}|N)$ 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. Toward that goal, we first discuss monotonicity assumptions next.

B. Monotonicity in the Predictor's Quantization Level

Motivated by a similar empirical observation in [32], we show here the *monotonicity in the predictor's quantization level* for both Lagrangian $\phi_{j_k,j_{k-1}}$ of the coded view j_k and the 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 the coded view j_k and the synthesized view distortion $d^s_{j_k,j_{k+1}}$ are monotonically nondecreasing functions of the predictor's quantization levels, i.e.,

$$\begin{aligned}
&\phi_{j_k,j_{k-1}}(q_{j_k}, p_{j_k}, q_{j_{k-1}}, p_{j_{k-1}}) \\
&\quad \leq \phi_{j_k,j_{k-1}}\left(q_{j_k}, p_{j_k}, q^+_{j_{k-1}}, p_{j_{k-1}}\right) \\
&\phi_{j_k,j_{k-1}}(q_{j_k}, p_{j_k}, q_{j_{k-1}}, p_{j_{k-1}}) \\
&\quad \leq \phi_{j_k,j_{k-1}}\left(q_{j_k}, p_{j_k}, q_{j_{k-1}}, p^+_{j_{k-1}}\right) \qquad (15)
\end{aligned}$$

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

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

In other words, (15) states that, if the predictor view j_{k-1} uses a coarser quantization level in the 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 the monotonicity of the synthesized view distortion $d^s_{j_k,j_{k+1}}$. 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 the Predictor's Distance

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

> The Lagrangian term $\phi_{\zeta,\xi}(q_\zeta, p_\zeta, q_\xi, p_\xi)$ for the coded view ζ, given the predictor view ξ, and the synthesized view distortion $d^s_{\zeta,\xi}(q_\zeta, p_\zeta, q_\xi, p_\xi)(v)$ for the intermediate view v, given closest left and right coded views ζ and ξ, respectively, are monotonically nondecreasing functions of the predictor's distance, i.e.,
>
> $$\phi_{\zeta,\xi}(q_\zeta, p_\zeta, q_\xi, p_\xi) \leq \phi_{\zeta,\xi^-}(q_\zeta, p_\zeta, q_\xi, p_\xi) \tag{17}$$
>
> $$\begin{aligned} d^s_{\zeta,\xi}(q_\zeta, p_\zeta, q_\xi, p_\xi)(v) &\leq d^s_{\zeta,\xi^+}(q_\zeta, p_\zeta, q_\xi, p_\xi)(v) \\ d^s_{\zeta,\xi}(q_\zeta, p_\zeta, q_\xi, p_\xi)(v) &\leq d^s_{\zeta^-,\xi}(q_\zeta, p_\zeta, q_\xi, p_\xi)(v) \end{aligned} \tag{18}$$
>
> 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 $\phi_{\zeta,\xi}(q_\zeta, p_\zeta, q_\xi, p_\xi)$. Equation (18) states that, for the 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 the predictor frame at view v_i and the 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 been also shown valid in [44] using the Markov-random-field prior model, and we will verify it empirically in Section VI. We note that, while the monotonicity in the predictor's quantization level has been extensively used [32], [33], [36], we are the first in the literature to exploit monotonicity in the 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 subpath (minimum Lagrangian cost subpath) from any states of the first view to state $(q_{v_n}, p_{v_n})^{v_n}$ of the captured view v_n. The first lemma eliminates *suboptimal states* $(q_{v_n}, p_{v_n})^{v_n}$, given computed $\Phi_{v_n}(q_{v_n}, p_{v_n})$ values, using the monotonicity in the predictor's quantization level.

Lemma 1: For a given texture-map quantization level p_{v_n}, if, at the state plane of the 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 subpaths 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 the subpath cost to state $(q^+_{v_n}, p_{v_n})$ with a coarse texture quantization level $q^+_{v_n}$ is already larger than the subpath cost to state $(q^*_{v_n}, p_{v_n})$ with a fine texture quantization level $q^*_{v_n}$, then state $(q^+_{v_n}, p_{v_n})$ is globally suboptimal. A simple proof is provided in the Appendix.

Lemma 1 also holds true for the 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 $(q_{v_n}, p^+_{v_n})^{v_n}$, $\forall p^+_{v_n} > p^*_{v_n}$, are globally suboptimal and can be pruned.

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

Lemma 2: Given a start state $(q_{v_n}, p_{v_n})^{v_n}$ of the captured view v_n, the end state $(q_\xi, p_\xi)^\xi$ of the captured view ξ and the in-between captured view v_{n+1}, $v_n < v_{n+1} < \xi$, if the cost of the 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 the 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 the captured view v_n, the traversing state $(q_{v_n}, p_{v_n})^{v_{n+1}}$ of the captured view v_{n+1} with same quantization levels is cheap in Lagrangian cost compared with a lower bound cost of the skipping captured view v_{n+1}, en route to destination state $(q_\xi, p_\xi)^\xi$, then the skipping captured view v_{n+1} using edge $(q_{v_n}, p_{v_n})^{v_n} \rightarrow (q_\xi, p_\xi)^\xi$ is suboptimal. 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 indexes larger than ξ, cannot also belong to the SP. The reason is that the synthesized distortion $d^s_{v_n,\xi}(v)$ of the intermediate view v using the coded views v_n and ξ as predictors is surely no larger than $d^s_{v_n,\xi^+}(v)$ using the coded view v_n and the further-away coded view ξ^+ with same or coarser quantization levels. Hence, the said condition must also hold for $(q_{\xi^+}, p_{\xi^+})^{\xi^+}$, 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

We now describe a bit allocation algorithm, shown in Algorithm 1, exploiting the lemmas derived in the 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 the trellis, for each captured view v_n, using computed subpaths to states $(q_{v_n}, p_{v_n})^{v_n}$ with subpath Lagrangian costs[6] $\Phi_{v_n}(q_{v_n}, p_{v_n})$, we first eliminate states with larger Lagrangian costs Φ_{v_n} and coarser texture quantization levels $q^+_{v_n}$ than a minimum state $(q^*_{v_n}, p_{v_n})$, given p_{v_n}. The same procedure is applied for the depth quantization levels $p^+_{v_n}$, given a fixed q_{v_n}. These suboptimal states are eliminated due to Lemma 1.

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 the 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 $(q^+_{v_n}, p_{v_n})^{v_n}$, $q^+_{v_n} > q^*_{v_n}$.
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 $(q_{v_n}, p^+_{v_n})^{v_n}$, $p^+_{v_n} > p^*_{v_n}$.
4: For each survived state $(q_{v_n}, p_{v_n})^{v_n}$ of view $v_n$, evaluate forward subpaths to states $(q_{v_{n+1}}, p_{v_{n+1}})^{v_{n+1}}$ of the 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 the neighboring captured view $v_{n+1}$, evaluate subpaths forward, i.e.,
6: $\zeta \leftarrow$ the 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 a nonzero vector, increment $\zeta$ to the next neighboring captured view, and go to step 7.
15: If $n < N$, increment $n$, and repeat steps 2 to 14.
```

In step 4, for each survived state $(q_{v_n}, p_{v_n})^{v_n}$ of view v_n, we *evaluate* all forward subpaths to states $(q_{v_{n+1}}, p_{v_{n+1}})^{v_{n+1}}$ 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 subpath to $(q_{v_{n+1}}, p_{v_{n+1}})^{v_{n+1}}$ to date, i.e., $\Phi_{v_{n+1}}(q_{v_{n+1}}, p_{v_{n+1}})$, 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 accordingly updated.

In step 5, for each survived state $(q_{v_n}, p_{v_n})^{v_n}$, we next evaluate feasible edges to states $(q_\zeta, p_\zeta)^\zeta$ of the captured views ζ, $\zeta > v_{n+1}$. Feasible edges are the 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 the monotonicity in the predictor's quantization level and distance. We then evaluate the quality of our estimate of the intermediate synthesized view distortion using our proposed cubic distortion model. Finally, we show the effectiveness of our proposed bit allocation strategy.

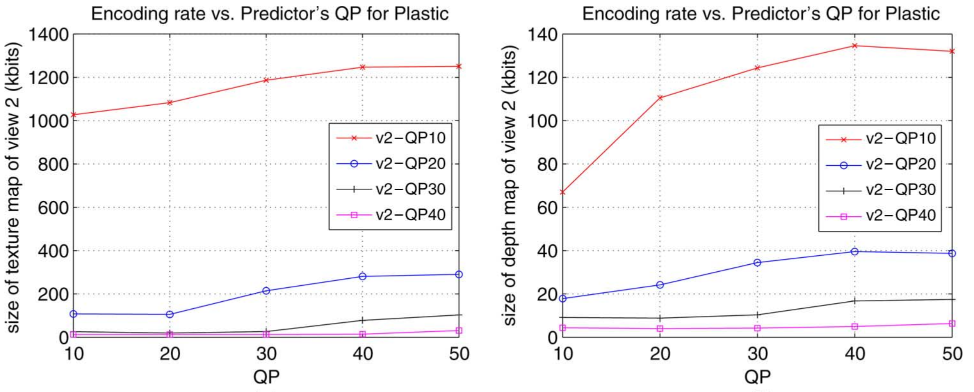
For test data sets, we used four Middlebury multiview image sequences [41], i.e., `Plastic`, `Lampshade1`, `Rocks2`, and `Bowling2`, of sizes 1270 × 1110, 1300 × 1110, 1276 × 1110, and 1330 × 1110, respectively. We assumed that the captured camera views were $\{1, 2, 3, 4, 5\}$ and the desired constructed view spacing Δ at the decoder was 0.05. For all our experiments, we used H.264 JM16.2 [45] video codec to encode texture and depth maps (texture and depth maps were independently encoded 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 the DIBR virtual view synthesis at the decoder, we used a simple algorithm presented in [39]. 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 to the source anchor view. At the synthesized view pixel coordinates, when 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. When pixels are unavailable from both of the anchor views, holes are filled in a postprocessing in-painting or interpolation step.

A. Validation of Monotonicity Assumptions

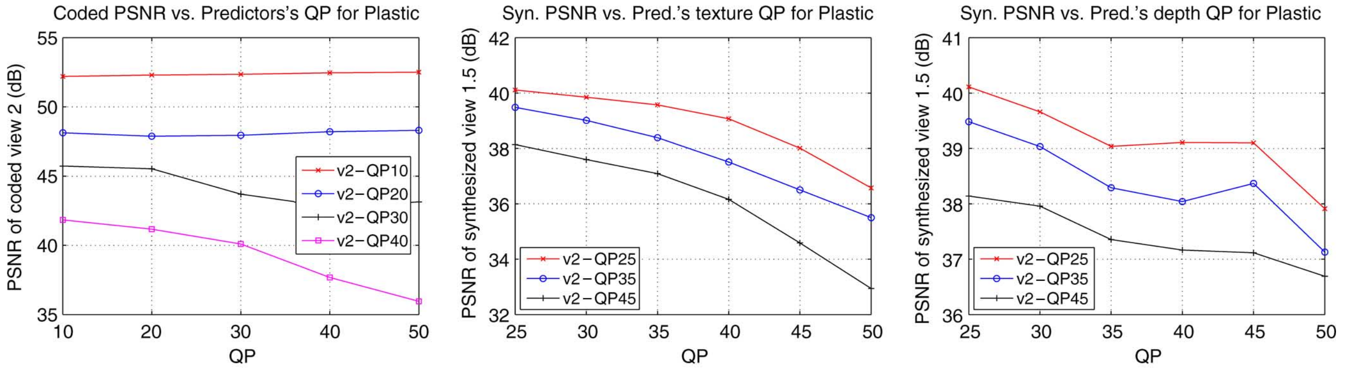
We first provide empirical evidence to show that the assumption of the monotonicity in the predictor's quantization level and distance are indeed valid. Using the `Plastic` sequence, we first plotted the texture-map coding rate of captured view 2, using

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



(a) (b)

Fig. 5. Encoding rate of the texture and depth maps of coded view 2 are plotted against the quantization level of predictor view 1 for the `Plastic` sequence. Each curve is generated using constant quantization level for view 2. (a) Texture-map coding rate versus the predictor's QP. (b) Depth-map coding rate versus the predictor's QP.



(a) (b) (c)

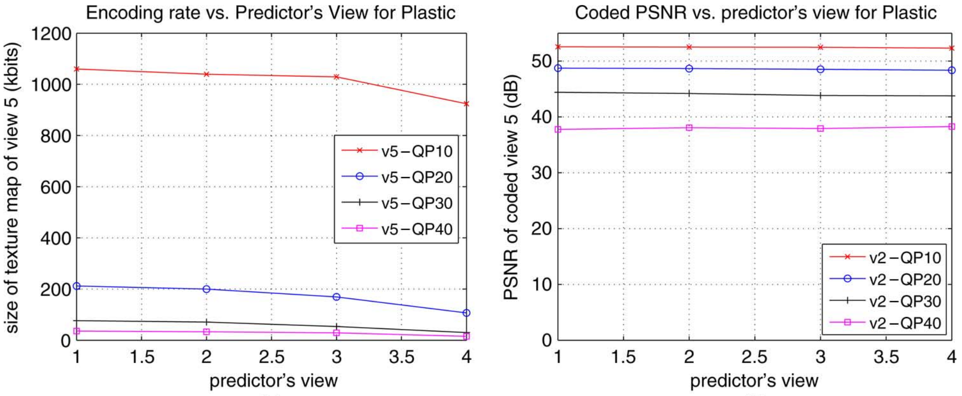
Fig. 6. Visual quality of coded view 2 and synthesized view 1.5 are plotted against the quantization levels of predictor view 1 for the `Plastic` sequence. Each curve is generated using constant quantization level(s) for coded view 2. (a) Coded PSNR versus predictor's QP. (b) Synthesized PSNR versus predictor's texture QP. (c) Synthesized PSNR versus predictor's depth QP.

captured view 1 as the predictor, as function of the quantization level of view 1 (quantization level of view 2 was kept constant for each curve). In Fig. 5(a), we see that, for all curves, the texture-map coding rate of view 2 increased as the quantization level of view 1 became larger (coarser). In Fig. 5(b), we see the same trend for the depth-map coding rate of view 2 as a function of the quantization level of predictor view 1. 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 level), more bits must be spent.

We also plotted the peak signal-to-noise ratio (PSNR; a common objective measure for image quality) of coded view 2 as a function of the quantization level of predictor view 1 in Fig. 6(a). We see that, for all curves, the PSNR either remained roughly constant or decreased (distortion increased) as the quantization level of view 1 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 deteriorates. Since the Lagrangian cost is a weighted sum of the distortion and the coding rate, given empirical evidence showing that the distortion and the coding rate increase as a function of the predictor's quantization level, we can conclude that our assumption of the Lagrangian-cost monotonicity of the predictor's quantization level [see (15)] is shown to be valid.

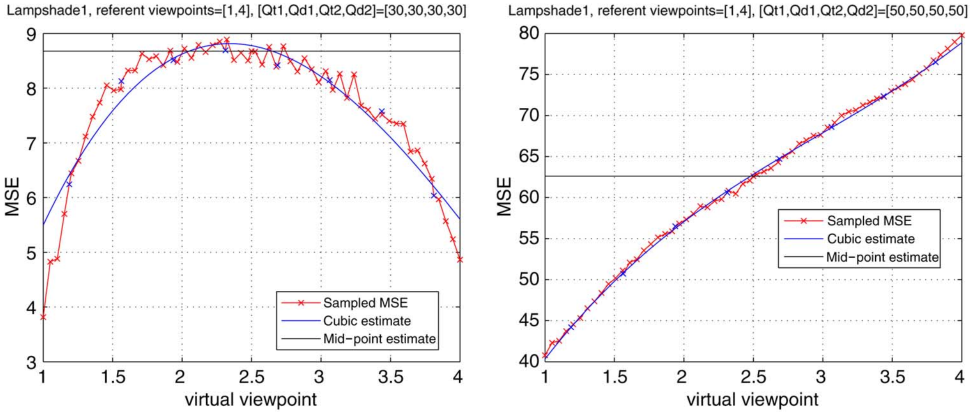
We also plotted the PSNR of synthesized view 1.5 as a function of the *texture-map* quantization level of predictor view 1 in Fig. 6(b) and as a function of *depth-map* quantization level of predictor view 1 in Fig. 6(c). (Quantization levels of the other map of view 1 and the texture and depth maps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

To validate our assumption of the monotonicity of the predictor's distance, we first plotted the texture-map coding rate of view 5 as a function of the predictor's view in Fig. 7(a). (Quantization levels of the texture maps of the predictor and view 5 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. Although not shown, the depth-map coding



(a) (b)

Fig. 7. Texture-map encoding rate and visual quality of coded view 5 are plotted against the predictor's view for the `Plastic` sequence. Each curve is generated using constant quantization level(s) for all coded views. (a) Texture-map coding rate versus predictor's view. (b) Coded view PSNR versus predictor's view.



(a) (b)

Fig. 8. Synthesized distortion is plotted against viewpoint location for different quantization levels for the `Lampshade1` sequence. (Blue) Cubic distortion model, (black) midpoint, and actual synthesized distortion at 0.05 view spacing are shown. Synthesized MSE versus viewpoint for (a) $\mathrm{QP} = 30$. and (b) $\mathrm{QP} = 50$.

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 a function of the predictor's view. As discussed earlier, intuitively, the quality of the predictee (view 5) is mostly controlled by its quantization level; thus, 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 assumption of the Lagrangian cost monotonicity of the predictor's distance (17).

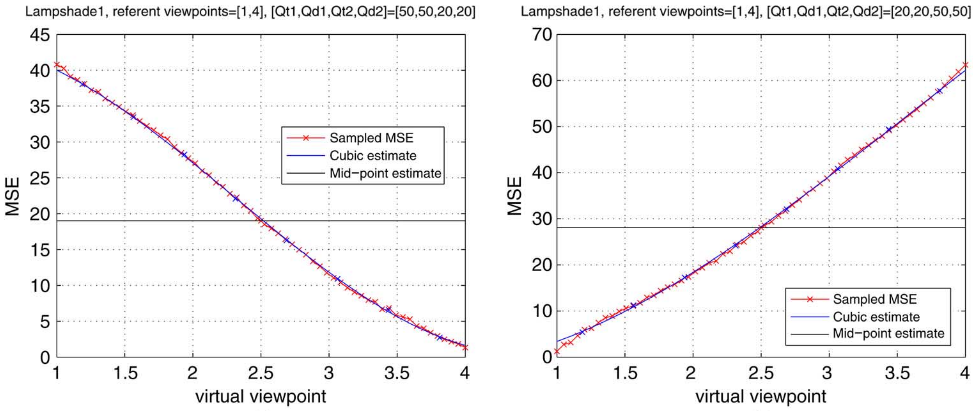
B. Accuracy of the Cubic Distortion Model

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 a function of the viewpoint location in Fig. 8(a) and (b) for two different sets of quantization levels, i.e., $\mathrm{QP} = 30$ and $\mathrm{QP} = 50$ in Fig. 8(a) and (b), respectively. The actual computed MSE of the synthesized view, as compared with the "clean" synthesized view when interpolated using uncompressed texture and depth maps of two nearest captured views, is shown in red. The constructed cubic distortion model is shown in blue. We first observe that there was a non-negligible noise term $n(v)$ in the measured MSE due to secondary effects such as occlusion, rounding, etc. Second, we visually see that, for both plots, our proposed distortion model did track this trend of synthesized distortion as a function of the 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), the quantization level for the left view was set coarser than the right, whereas in Fig. 9(b), the 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 the measured MSE accurately, showing the accuracy of our model.

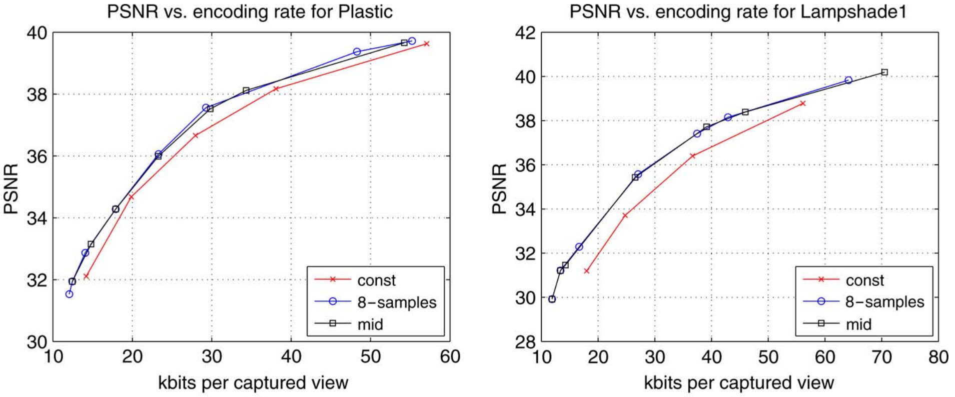
C. Comparing the RD Performance of Bit Allocation Strategies

We tested the performance of our proposed bit allocation strategy using both sampling methods discussed in Section III-B, i.e., S samples to construct the cubic model ($8-$`samples`) and a single midpoint sample (`mid`) to bound the 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



(a) (b)

Fig. 9. Synthesized distortion is plotted against viewpoint location for different quantization levels for the Lampshade1 sequence. (Blue) Cubic distortion model, (black) midpoint, and actual synthesized distortion at 0.05 view spacing are shown. (a) Synthesized MSE versus viewpoint for $QP_1 > QP_4$ and (b) $QP_1 < QP_4$.



(a) (b)

Fig. 10. Performance comparison between optimal and constant-QP coded view and quantization level selection schemes. (a) Plastic. (b) Lampshade1.

assigns a constant quantization level to all texture and depth maps of the coded views.

In Fig. 10, we see the performance of the bit allocation strategies for Plastic and Lampshade1, shown as the PSNR versus the bitrate per captured view (including both texture and depth maps). First, we see that both 8 − samples and mid have better RD performance than const over all bitrate regions, i.e., by up to 0.80 and 1.51 dB for Plastic and Lampshade1, respectively. This shows that the correct selection of quantization levels per frame is important. Second, as the bitrate decreased, 8–samples and mid selected fewer captured views for coding and instead relied on the decoder's view synthesis of captured views (four leftmost points in Plastic and three leftmost points in Lampshade1 of 8 − samples represented selections of uncoded views). This is also the region where 8 − samples and mid outperformed const the most; hence, the selection of captured views for coding is also important for the 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, as compared with 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 3-D trellis. We found that the computation savings ranged from 80% to 99%, with the maximum saving occurring at the rightmost RD point.

In Fig. 11, we see the RD performance of the competing bit allocation schemes for sequences Rocks2 and Wood1. We see that the general trend is similar to the earlier two sequences, i.e., the performance gain of our bit allocation strategies 8–samples and mid over constant-QP scheme const is more pronounced at a low bitrate, when captured views are skipped. The maximum gain in PSNR for these two sequences are 1.05 and 0.95 dB, 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) shows the number of captured views selected by mid for encoding as a function of the encoding bitrate for the image sequences wood1 and lampshade1. We observe that, at lower bitrate region, a fewer number of views were selected for encoding. This is intuitive since a fewer number of encoded views leads to smaller bit expenditure in general. This is also the region where mid outperformed const the most. This shows that, when bitrate is more of a concern than synthesized view quality, selecting the right subset of captured frames for encoding is very important for good RD performance.

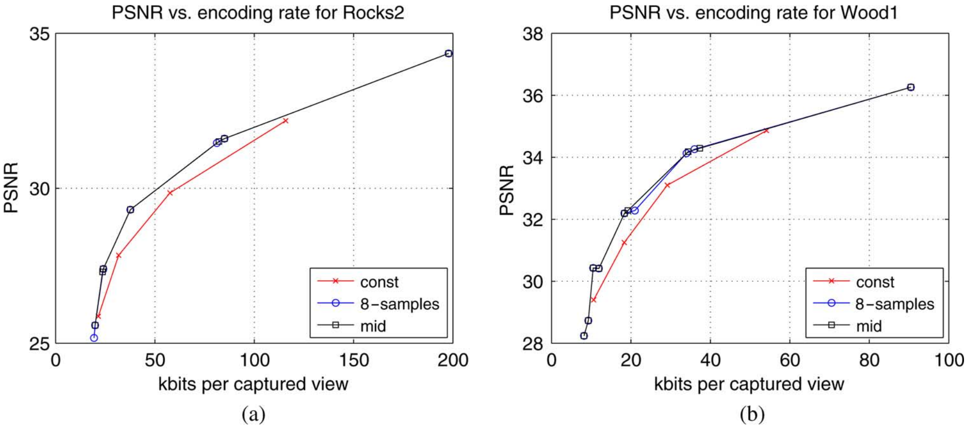


Fig. 11. Performance comparison between optimal and constant-QP coded view and quantization level selection schemes. (a) `Rocks2`. (b) `Wood1`.

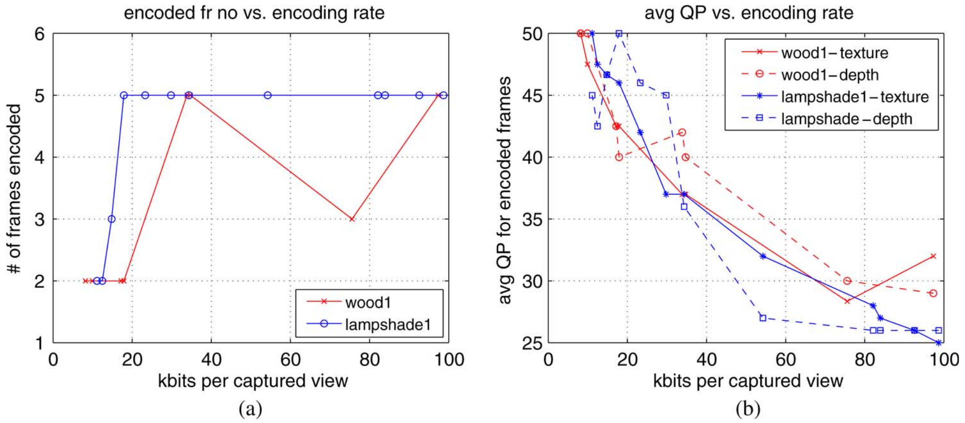


Fig. 12. Number of captured frames selected for encoding and average QP for selected encoded frames as a function of the encoded bitrate for `wood1` and `lampshade1`. (a) Number of encoded frames versus encoding rate. (b) Average QP versus encoding rate.

In Fig. 12(b), we plotted the average QP of the selected encoded views in solutions generated by `mid` as a function of the bitrate for `wood1` and `lampshade1`. We see that, as the 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

Toward the goal of finding a compact multiview image representation, i.e., the one that takes advantage of both the efficient texture- and depth-map coding tools at the encoder and the view-synthesis tool using DIBR at the decoder, in this paper, we have presented an algorithm to select captured views for encoding and quantization levels of the corresponding texture and depth maps in an RD optimal manner. We have first derived a cubic distortion model that models the synthesized view distortion between two coded views. We have then shown that, using the monotonicity in the predictor's quantization level and distance, the search complexity can be drastically reduced without loss of optimality. Experiments have shown that our selection scheme outperformed a heuristic scheme by up to 1.5 dB 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 the shortest subpath up to state $(q^+_{v_n}, p_{v_n})^{v_n}$, $q^+_{v_n} > q^*_{v_n}$, is a part of an end-to-end SP. That means that the captured view v_n is a coded view; let $j_k = v_n$. If we replace the subpath to $(q^+_{j_k}, p_{j_k})^{j_k}$ with the subpath to $(q^*_{j_k}, p_{j_k})^{j_k}$, synthesized intermediate views to the right of j_k and the coded view j_{k+1} that depend on the texture map of view j_k 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 the monotonicity in the predictor's quantization level [see (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 the subpath to $(q^+_{j_k}, p_{j_k})^{j_k}$ with the subpath 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 the traversing state

$(q_{v_n}, p_{v_n})^{v_{n+1}}$, considering intermediate synthesized views v, $v_n < v < v_{n+1}$, and the captured view v_{n+1} is smaller than not traversing it by assumption. Moreover, the Lagrangian cost of the coded view ξ and the distortion of synthesized views to the right of view v_n that was predicted from view v_n will not increase, predicting view v_{n+1} instead with same quantization levels due to the monotonicity of the predictor's distance [see (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, Jan. 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 *Proc. IEEE Int. Conf. Multimedia Expo*, Toronto, ON, Canada, Jul. 2006, pp. 437–440.

[3] MPEG—Technologies—Introduction to multiview video coding Jan. 2008, iSO/IEC JTC 1/SC 29/WG 11 N9580.

[4] T. Chen, "Adaptive temporal interpolation using bidirectional motion estimation and compensation," in *Proc. IEEE Int. Conf. Image Process.*, Rochester, NY, Sep. 2002, pp. II-313–II-316.

[5] J. Zhai, K. Yu, J. Li, and S. Li, "A low complexity motion compensated frame interpolation method," in *Proc. IEEE Int. Symp. Circuits Syst.*, Kobe, Japan, May 2005, pp. 4927–4930.

[6] P. Merkle, A. Smolic, K. Mueller, and T. Wiegand, "Multi-view video plus depth representation and coding," in *Proc. IEEE Int. Conf. Image Process.*, San Antonio, TX, Oct. 2007, pp. I-201–I-204.

[7] P. Kauff, N. Atzpadin, C. Fehn, M. Mller, O. Schreer, A. Smolic, and R. Tanger, "Depth map creation and image-based rendering for advanced 3DTV services providing interoperability and scalability," *Signal Process. Image Commun.*, vol. 22, no. 2, pp. 217–234, Feb. 2007, Special Issue on three-dimensional video and television.

[8] S. Gokturk, H. Yalcin, and C. Bamji, "A time-of-flight depth sensor—System description, issues and solutions," in *Proc. Conf. CVPRW*, Washington, DC, Jun. 2004, p. 35.

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

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

[11] G. Cheung and V. Velisavljević, "Efficient bit allocation for multiview image coding & view synthesis," in *Proc. IEEE Int. Workshop Multimedia Signal Process.*, St. Malo, France, Oct. 2010.

[12] J. Shade, S. Gortler, L. He, and R. Szeliski, "Layered depth images," in *Proc. SIGGRAPH*, New York, Sep. 1998, pp. 231–242.

[13] M. Levoy and P. Hanrahan, "Light field rendering," in *Proc. SIGGRAPH*, New Orleans, LA, Aug. 1996, pp. 31–42.

[14] S. Gortler, R. Grzeszczuk, R. Szeliski, and M. Cohen, "The lumigraph," in *Proc. SIGGRAPH*, New Orleans, LA, Aug. 1996, pp. 43–54.

[15] P. Debevec, C. Taylor, and J. Malik, "Modeling and rendering architecture from photographs," in *Proc. SIGGRAPH*, New Orleans, LA, Aug. 1996, pp. 11–20.

[16] H.-Y. Shum, S. B. Kang, and S.-C. Chan, "Survey of image-based representations and compression techniques," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 11, pp. 1020–1037, Nov. 2003.

[17] M. Magnor, P. Ramanathan, and B. Girod, "Multi-view coding for image-based rendering using 3-D scene geometry," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 11, pp. 1092–1106, Nov. 2003.

[18] S. Seitz and C. Dyer, "View morphing," in *Proc. SIGGRAPH*, New Orleans, LA, Aug. 1996, pp. 21–30.

[19] W. Mark, L. McMillan, and G. Bishop, "Post-rendering 3D warping," in *Proc. Symp. Interactive 3D Graph.*, New York, Apr. 1997, p. 17.

[20] 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.

[21] M. Flierl, A. Mavlankar, and B. Girod, "Motion and disparity compensated coding for multiview video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 11, pp. 1474–1484, Nov. 2007.

[22] P. Merkle, A. Smolic, K. Muller, and T. Wiegand, "Efficient prediction structures for multiview video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 11, pp. 1461–1473, Nov. 2007.

[23] T.-Y. Chung, I.-L. Jung, K. Song, and C.-S. Kim, "Multi-view video coding with view interpolation prediction for 2D camera arrays," *J. Vis. Commun. Image Represent.*, vol. 21, no. 5/6, pp. 474–486, Jul. 2010.

[24] 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 *Proc. IEEE Int. Conf. Image Process.*, San Antonio, TX, Sep. 2007, pp. V-105–V-108.

[25] M. Maitre, Y. Shinagawa, and M. Do, "Wavelet-based joint estimation and encoding of depth-image-based representations for free-viewpoint rendering," *IEEE Trans. Image Process.*, vol. 17, no. 6, pp. 946–957, Jun. 2008.

[26] 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 *Proc. IEEE Int. Conf. Image Process.*, San Diego, CA, Oct. 2008, pp. 2448–2451.

[27] J.-R. Ohm, E. Izquierdo, and K. Muller, "Systems for disparity-based multiple-view interpolation," in *Proc. IEEE Int. Symp. Circuits Syst.*, Monterey, CA, 1998, vol. 5, pp. 502–505.

[28] 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," *Signal Process. Image Commun.*, vol. 24, no. 8, pp. 666–681, Sep. 2009.

[29] W.-S. Kim, A. Ortega, P. Lai, D. Tian, and C. Gomila, "Depth map distortion analysis for view rendering and depth coding," in *Proc. IEEE Int. Conf. Image Process.*, Cairo, Egypt, Nov. 2009, pp. 721–724.

[30] W.-S. Kim, A. Ortega, P. Lai, D. Tian, and C. Gomila, "Depth map coding with distortion estimation of rendered view," in *Proc. SPIE Visual Inf. Process. Commun.*, San Jose, CA, Jan. 2010, pp. 75 430B-1–75 430B-10.

[31] Y. Shoham and A. Gersho, "Efficient bit allocation for an arbitrary set of quantizers," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 36, no. 9, pp. 1445–1453, Sep. 1988.

[32] K. Ramchandran, A. Ortega, and M. Vetterli, "Bit allocation for dependent quantization with applications to multiresolution and MPEG video coders," *IEEE Trans. Image Process.*, vol. 3, no. 5, pp. 533–545, Sep. 1994.

[33] J.-H. Kim, J. Garcia, and A. Ortega, "Dependent bit allocation in multiview video coding," in *Proc. IEEE Int. Conf. Image Process.*, Genoa, Italy, Sep. 2005, pp. II-293–II-296.

[34] H. Song and C.-C. J. Kuo, "Rate control for low-bit-rate video via variable-encoding frame rates," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 4, pp. 512–521, Apr. 2001.

[35] G. Liebl, M. Kalman, and B. Girod, "Variable frame skipping scheme based on estimated quality of non-coded frames at decoder for real-time block-based video coding," in *Proc. IEEE Int. Conf. Multimedia Expo.*, Taipei, Taiwan, Jul. 2004, pp. 1127–1130.

[36] S. Liu and C.-C. J. Kuo, "Joint temporal-spatial bit allocation for video coding with dependency," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 1, pp. 15–26, Jan. 2005.

[37] G. Cheung and V. Velisavljević, "Efficient bit allocation for multiview image coding & view synthesis," in *Proc. IEEE Int. Conf. Image Process.*, Hong Kong, Sep. 2010, pp. 2613–2616.

[38] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision.*. Cambridge, U.K.: Cambridge Univ. Press, 2003.

[39] 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," *Signal Process. Image Commun.*, vol. 24, no. 1/2, pp. 73–88, Jan. 2009.

[40] C. Bishop, *Pattern Recognition and Machine Learning.*. New York: Springer-Verlag, 2006.

[41] 2006 Stereo Datasets [Online]. Available: http://vision.middlebury.edu/stereo/data/scenes2006/

[42] G. Cheung, W.-T. Tan, and C. Chan, "Reference frame optimization for multiple-path video streaming with complexity scaling," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 6, pp. 649–662, Jun. 2007.

[43] I. Daribo, C. Tillier, and B. Pesquet-Popescu, "Motion vector sharing and bit-rate allocation for 3D video-plus-depth coding," *EURASIP: Special Issue on 3DTV in Journal on Advances in Signal Processing*, vol. 2009, pp. 258 920-1–258 920-13, Jan. 2009.

[44] W. Li and B. Li, "Virtual view synthesis with heuristic spatial motion," in *Proc. IEEE Int. Conf. Image Process.*, San Diego, CA, Oct. 2008, pp. 1508–1511.

[45] T. Wiegand, G. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Jul. 2003.

Gene Cheung (M'00–SM'07) received the B.S. degree in electrical engineering from Cornell University, Ithaca, NY, in 1995, and the M.S. and Ph.D. degrees in electrical engineering and computer science from the University of California, Berkeley, in 1998 and 2000, respectively.

From 2000 to 2009, he was a Senior Researcher with Hewlett-Packard Laboratories Japan, Tokyo, Japan. He is currently an Assistant Professor with the National Institute of Informatics, Tokyo. He has published over 15 international journals and 70 conference publications. His research interests include media representation and network transport, single-/multiple-view video coding and streaming, and immersive communication and interaction.

Dr. Cheung has been serving as the Associate Editor of the IEEE TRANSACTIONS ON MULTIMEDIA since 2007 and as the Associate Editor of the Digital Signal Processing Applications Column in the IEEE Signal Processing Magazine since 2011. He has also served as the Area Chair in the IEEE International Conference on Image Processing 2010 and the Technical Program Cochair of the International Packet Video Workshop 2010. He serves as the Track Cochair for the Multimedia Signal Processing track in the IEEE International Conference on Multimedia and Expo 2011. He was a corecipient of the Top 10% Paper Award in the IEEE International Workshop on Multimedia Signal Processing 2009.

Vladan Velisavljević (M'06) received the B.Sc. and M.Sc. (Magister) degrees from the University of Belgrade, Belgrade, Serbia, in 1998 and 2000, respectively, and the Master and Ph.D. degrees from the Ecole Polytecnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland, in 2001 and 2005, respectively.

From 1999 to 2000, he was a Member of the academic staff with the University of Belgrade. In 2000, he was a Teaching and Research Assistant with the Audiovisual Communications Laboratory, EPFL, where he worked on the Ph.D. degree in the field of image processing. In 2003, he was a Visiting Student with the Imperial College London, London, U.K. Since 2006, he has been a Senior Research Scientist with Deutsche Telekom Laboratories, Berlin, Germany. His research interests include image, video and multiview-video compression and processing, wavelet theory, multiresolution signal processing, and distributed image/video processing.

Dr. Velisavljević is currently a member of the IEEE Communications Society Review Board for Multimedia Communications Technical Committee.

Antonio Ortega (S'91–M'95–SM'00–F'07) received the Telecommunications Engineering degree in 1989 from the Universidad Politecnica de Madrid, Madrid, Spain, and the Ph.D. degree in electrical engineering in 1994 from Columbia University, New York, NY, where he was supported by a Fulbright scholarship.

Since 1994, he has been with the Department of Electrical Engineering Systems, University of Southern California (USC),Los Angeles, where he is currently a Professor. He was a Director of the Signal and Image Processing Institute and is currently the Associate Chair of Electrical Engineering Systems with the USC. His work with the USC has been or is being funded by agencies such as the National Science Foundation, the National Aeronautics and Space Administration, and the DOE, as well as a number of companies. Over 25 Ph.D. students have completed their Ph.D. thesis work under his supervision at the USC, and his work has led to over 250 publications in international conferences and journals. His research interests are in the areas of multimedia compression, communications, and signal analysis. His recent work is focusing on distributed compression, multiview coding, error tolerant compression, wavelet-based signal analysis, and information representation in wireless sensor networks.

Dr. Ortega is a member of the Association for Computing Machinery. He has been the Chair of the Image and Multidimensional Signal Processing Technical Committee and a member of the Board of Governors of the IEEE Signal Processing Society (2002). He has been Technical Program Cochair of the IEEE International Conference on Image Processing 2008, the IEEE International Workshop on Multimedia Signal Processing 1998, and the IEEE International Conference on Multimedia and Expo 2002. He has been the Associate Editor for the IEEE TRANSACTIONS ON IMAGE PROCESSING, the IEEE SIGNAL PROCESSING LETTERS, and the EURASIP Journal on Advances in Signal Processing. He received the National Science Foundation CAREER award, the 1997 IEEE Communications Society Leonard G. Abraham Prize Paper Award, the IEEE Signal Processing Society 1999 Magazine Award, and the 2006 EURASIP Journal of Advances in Signal Processing Best Paper Award.