Ann. Telecommun.
DOI 10.1007/s12243-013-0375-6

Rate-distortion analysis of multiview coding in a DIBR framework

Boshra Rajaei · Thomas Maugey · Hamid-Reza Pourreza · Pascal Frossard

Received: 15 October 2012 / Accepted: 21 May 2013

Abstract Depth image-based rendering techniques for multiview applications have been recently introduced for efficient view generation at arbitrary camera positions. The rate control in an encoder has thus to consider both texture and depth data. However, due to different structures of depth and texture data and their different roles on the rendered views, the allocation of the available bit budget between them requires a careful analysis. Information loss due to texture coding affects the value of pixels in synthesized views, while errors in depth information lead to a shift in objects or to unexpected patterns at their boundaries.In this paper, we address the problem of efficient bit allocation between texture and depth data of multiview sequences.We adopt a rate-distortion framework based on a simplified model of depth and texture images, which preserves the main features of depth and texture images. Unlike most recent solutions, our method avoids rendering at encoding time for distortion estimation so that the encoding complexity stays low. In addition to this, our model is independent of the underlying inpainting method that is used at the decoder for filling holes in the synthetic views. Extensive experiments validate our theoretical results and confirm the efficiency of our rate allocation strategy.

Keywords Depth image-based rendering · Multiview video coding · Rate allocation · Rate-distortion analysis

B. Rajaei (✉) · H.-R. Pourreza
Ferdowsi University of Mashhad, Mashhad, Iran
e-mail: b_rajaee@sadjad.ac.ir

H.-R. Pourreza
e-mail: hpourreza@um.ac.ir

B. Rajaei
Sadjad Institute of Higher Education, Mashhad, Iran

T. Maugey · P. Frossard
Signal Processing Laboratory (LTS4), École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland

T. Maugey
e-mail: thomas.maugey@epfl.ch

P. Frossard
e-mail: pascal.frossard@epfl.ch

1 Introduction

Multiview coding is a research field that has witnessed many technological revolutions in the recent years. One of them is the significant improvement in the capabilities of camera sensors. Nowadays, high-quality camera sensors that capture color and depth information are easily accessible [1]. Obviously this brings important modifications in the data that the 3D transmission systems have to process. A few years ago, transmission systems used disparity estimation to improve the compression performance [2, 3]. Nowadays, 3D systems rather employ depth information to augment compression performance or to improve the quality of experience by increasing the number of views that could be displayed at the receiver side [4, 5]. This is possible using depth image-based rendering (DIBR) techniques [6, 7] that project one reference image onto virtual views using depth as geometrical information. Figure 1 shows the overall structure of a DIBR multiview coder that is considered in this paper. It includes the following steps. First, the captured views along with their corresponding depth maps are coded at the bit rates assigned by a rate allocation method. Then, the coded information is transmitted to the decoder. Finally, the reference views are decoded, and virtual views are synthesized using the depth information at the decoder.

Published online: 19 June 2013

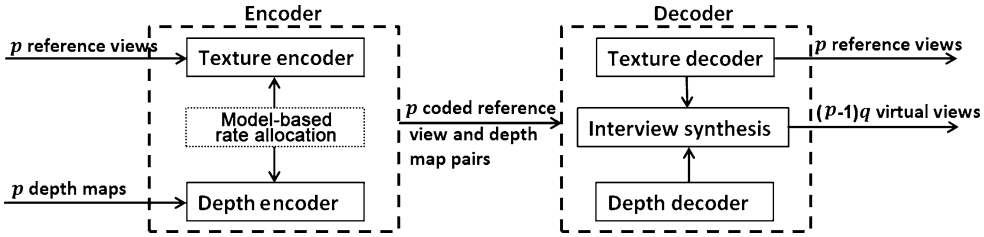

Fig. 1 A DIBR multiview system with p reference cameras and q equally spaced virtual views between each two reference views

View synthesis consists of two parts, namely the projection into the virtual view location using the closest reference views, and inpainting for filling the holes [8, 9] or pixels that remain undetermined after projection.

DIBR techniques offer new exciting possibilities but also impose new challenges. One of these important questions relies on the effect of depth compression on the view synthesis performance [10]; in particular, for a given bit budget R, what is the best allocation between depth and texture data? In other words, how can we distribute the total bit rate between color and geometrical information in order to maximize the rendering quality? It is important to note that the quality of the rendered view is of interest here, and not the distortion of depth images [10, 11]. This renders the problem of rate allocation particularly challenging.

In this paper, we propose a novel rate-distortion (RD) model to solve the rate allocation problem in DIBR systems with arbitrary number of reference and virtual views and without rendering at the encoder side. Inspired by [12–14], we first simplify different aspects of a multiview coder and keep only its main features. In particular, we make simple models for depth and texture coders, camera setup, and under observation scene. Then, we introduce a RD framework, where a RD function is used for optimizing the rate allocation in multiview coding. An important property of our allocation method is that we do not consider any specific inpainting step for virtual view synthesis at the decoder. There are two reasons for this choice: first, we want to design an allocation strategy that is independent of the actual inpainting method; second, we focus on the effect of view projections, which is mostly related to the geometry of the scene. To this aim, our RD analysis and later in experiments, distortion calculations are performed over nonoccluded regions. Experimental results show that our model-based rate allocation method is efficient for different system configurations. The approach proposed in this paper has low complexity and simultaneously provides a distortion that is not far from optimum. In particular, it outperforms a priori rate allocation strategies that are commonly used in practice.

The rate allocation problem has been the topic of many researches in the past few years. Allocating a fixed percentage of a total budget to the texture and depth data is probably the simplest allocation policy in the DIBR coding methods [15–17]. More efficient methods have, however, been proposed recently, and we discuss them in more details below.

First the current multiview coding (MVC) profile of H.264/AVC [3, 18, 19] uses the distortion of depth maps to distribute the available bit budget between texture and depth images. A group of papers try to improve MVC by taking into account depth properties. In [20], the authors suggest a preprocessing step based on an adaptive local median filter to enhance spatial, temporal, and inter-view correlations between depth maps and, consequently, to improve the performance of MVC. The work in [21] skips some depth blocks in the coding using the correlation between reference views and, hence, reduces the required bit budgets for coding depth maps. Other methods try to estimate the distortion of virtual views at the encoder side and replace it with the depth map distortion in the mode decision step of MVC [18]. In [22], the authors provide an upper bound for virtual view distortion that is related to the depth and texture errors and the gradients of the original reference views. Another upper bound for synthetic view distortion is proposed using the assumption of access to the original intermediate views at the encoder [23]. In [24], the authors calculate the translation error induced by depth coding and then try to estimate the rendered view distortion from the texture data. In a similar approach, the work in [25] models the distortion at each pixel of a virtual view, including the pixels in occluded regions. These methods only try to improve the current MVC profile. Without modeling the RD behavior, however, they cannot be used as general solutions for the rate allocation problem.

Beside improving the current MVC allocation policy, other papers build a complete RD model to solve the rate allocation problem and distribute a bit budget between texture and depth data in a DIBR multiview coder [26–30]. For example, assuming independency between depth and texture errors, the work in [26] proposes a RD function to find the optimal allocation in a video system with one reference and one virtual view. A region-based approach for estimating the distortion at virtual views is proposed in [28]. The allocation scheme is an iterative algorithm that needs to render one virtual view at every iteration for parameter initialization. This is very costly in terms of computational complexity. Along the same line of research, we also notice

the rate allocation and view selection method proposed in [29]. In this work, the authors first provide a cubic distortion model for synthetic views; they estimate the model coefficients by rendering at least one intermediate view between each reference camera views. Then, using this distortion model, a RD function is formulated, and a modified search algorithm is executed to simplify the rate allocation. Finally, a RD function is provided for a layer-based depth coder in [30]. The main drawbacks in the above allocation schemes reside in the rendering of at least one virtual view at encoding time and in the construction of RD functions that are view-dependent. Rendering at encoder side dramatically increases the computational complexity of the coder and is therefore not acceptable for real-time applications. In addition for view rendering at arbitrary camera positions, multiview systems require rate allocation strategies that are independent of actual reference and virtual views and their exact positions.

The organization of this paper is as follows. The next section clarifies the notations, camera and scene models, and RD framework that are used in Section 3 for calculation of our allocation model. Section 4 addresses a few optimization issues for determining the best rate allocation. Finally, Section 5 includes the details of our experimental settings and comparisons to other allocation strategies.

2 Framework and model

In this section, we define a few preliminary concepts that are used in our rate-distortion study. Our main focus is the problem of distributing the encoding bit rate between several reference views and the corresponding depth maps in a DIBR multiview system, such that the distortion over all reference and rendered views at the decoder is minimized. In particular, we are interested in constructing a RD function for rate allocation without explicit view synthesis at the encoder. We first construct a RD model for a typical wavelet-based texture coder and a simple quantization-based depth map coder, along with a simple model of scene.

Below, we present some general notations and the wavelet framework. Then we describe our RD analysis framework, our model of the scene and of the camera.

2.1 Notation

Let $\phi : \mathbb{R} \to \mathbb{R}$ and $\psi : \mathbb{R} \to \mathbb{R}$ be the univariate scaling and wavelet functions of an orthonormal wavelet transform, respectively [31]. The shifted and scaled forms of these functions are denoted by $\psi_{s,n}(t) = 2^{s/2}\psi(2^s t - n)$ and $\phi_{s,n}(t) = 2^{s/2}\phi(2^s t - n)$, where $s, n \in \mathbb{Z}$ are, respectively, the scaling and shifting parameters, and $\mathbb{Z}$ is the set of integer numbers. The most standard construction of two-dimensional wavelets relies on a separable design that uses $\Psi^1_{s,n_1,n_2}(t_1,t_2) = \phi_{s,n_1}(t_1)\psi_{s,n_2}(t_2)$, $\Psi^2_{s,n_1,n_2}(t_1,t_2) = \psi_{s,n_1}(t_1)\phi_{s,n_2}(t_2)$, and $\Psi^3_{s,n_1,n_2}(t_1,t_2) = \psi_{s,n_1}(t_1)\psi_{s,n_2}(t_2)$ as the bases. It is proved in [31] that separable wavelets provide an orthonormal basis for $L_2(\mathbb{R}^2)$. Therefore, any function $f \in L_2(\mathbb{R}^2)$ can be written as

$$f(t_1,t_2) = \sum_{s,n_1,n_2} \sum_{i=1}^{3} C^i_{s,n_1,n_2} \Psi^i_{s,n_1,n_2}(t_1,t_2),$$

where, for every $s, n_1, n_2 \in \mathbb{Z}$,

$$C^i_{s,n_1,n_2} = \left\langle f, \Psi^i_{s,n_1,n_2} \right\rangle, \ i = 1,2,3.$$

Practically, the wavelet transform defines a scale s_0 as the coarsest scale. If we call $C^i_{s,n_1,n_2}, s > s_0$ as the high-frequency bands, at s_0, we only have one low-frequency band $\langle f, \Phi_{s_0,n_1,n_2}\rangle$, where $\Phi_{s_0,n_1,n_2}(t_1,t_2) = \phi_{s_0,n_1}(t_1)\phi_{s_0,n_2}(t_2)$.

2.2 Scene and camera configuration model

We use a very simple model of scene in our analysis, and we consider foreground objects with arbitrary shapes and flat surfaces on a flat background.[1] Additionally, even though a real scene is three-dimensional, our model is a collection of 2D images as we consider projections of the 3D scene into cameras' 2D coordinates.

Let $\mathcal{H}^Q(\Omega)$ be the space of 2D functions, $f : \mathbb{R}^2 \to \mathbb{R}$, on the interval $[0,1]^2 \subset \mathbb{R}^2$, where Q is the number of foreground objects and $\Omega = \{\Omega_i, i = 0, \ldots, Q-1\}$ denotes the foreground objects. We define $f \in \mathcal{H}^Q(\Omega)$ as follows:

$$f(t_1,t_2) = \begin{cases} 1, & \text{if } \exists i : (t_1,t_2) \in \Omega_i \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

Our RD analysis is first performed on $\mathcal{H}^1(\Omega)$ where $\Omega = \{\Omega_0\}$. The extension to multiple foreground objects follows naturally. For the sake of clarity, we skip the superscript notation and represent this class by $\mathcal{H}(\Omega)$. Figure 2 shows a sample function from $\mathcal{H}(\Omega)$. This figure shows one arbitrarily shape foreground object on a flat background as it is projected into a 2D camera plane.

In addition to our simple scene model, we describe now our camera configuration model. Let us denote as $\mathcal{B}^p_q(\mathcal{P})$ a configuration with p reference cameras and q equally spaced intermediate views between each two consecutive reference views. Then, $\mathcal{P}$ is the set of intrinsic and extrinsic parameters for reference and virtual cameras. It is defined as $\mathcal{P} = \{(A_i, R_i, T_i) : i = 0, \ldots, p-1\} \cup \{(A'_j, R'_j, T'_j) :$

[1] The extension of our analysis to the scenes with C^α regular surfaces are straightforward.

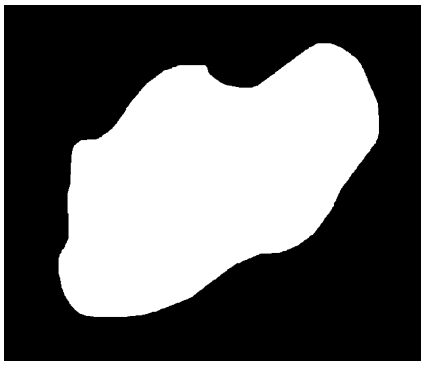

Fig. 2 A sample function in $\mathcal{H}^1(\Omega)$

$j = 0, \ldots, (p-1)q - 1\}$, where A_i and R_i are, respectively, the intrinsic and rotation matrices of the ith reference camera, and T_i is its corresponding translation vector. The similar parameters for virtual cameras are given by A'_j, R'_j, and T'_j. Figure 1 shows a multiview system that corresponds to a $\mathcal{B}_q^p(\mathcal{P})$ configuration. In this paper, we consider that a texture image and a depth map are coded and are sent to the decoder for each reference view. In our camera configuration $\mathcal{B}_q^p(\mathcal{P})$, we have p pairs of texture images and depth maps to be coded. The number of coded views is given by system design criteria or RD constraints [29].

2.3 Rate-distortion framework

Let us define three classes of signals $\mathcal{T} \subset L_2(\mathbb{R}^2)$, $\mathcal{V} \subset L_2(\mathbb{R}^2)$ and $\mathcal{D} \subset L_2(\mathbb{R}^2)$ as reference images, virtual views, and depth maps, respectively. Then, define $\mathcal{F}$ as the class of all

$$f = \{(t_i, d_i) : t_i \in \mathcal{T}, d_i \in \mathcal{D}, i = 0, \ldots, p-1\}$$

and similarly, $\mathcal{G}$ as the class of all

$$g = \left\{\left(t_i, v_j\right) : t_i \in \mathcal{T}, v_j \in \mathcal{V}, i = 0, \ldots, p-1, \quad j = 0, \ldots, (p-1)\,q - 1\right\}.$$

Here, $\mathcal{F}$ represents all the coded data, and $\mathcal{G}$ indicates the set of all reference and virtual views that are reconstructed at the decoder.

A typical multiview coding strategy consists of at least three building blocks, namely encoder, decoder, and renderingalgorithm. Consider a texture encoding scheme $\mathcal{E}_\mathcal{T} : \mathcal{T} \to \{1, 2, \ldots, 2^{R_\mathcal{T}}\}$ and similarly, a depth encoding scheme $\mathcal{E}_\mathcal{D} : \mathcal{D} \to \{1, 2, \ldots, 2^{R_\mathcal{D}}\}$, where $R_\mathcal{T} = \sum_{i=0}^{p-1} R_{t_i}$ and $R_\mathcal{D} = \sum_{i=0}^{p-1} R_{d_i}$ are the total number of bits allocated to the texture and depth information, respectively. This represents a total rate $R = R_\mathcal{T} + R_\mathcal{D}$ bit at the encoder. Correspondingly, we call the texture and depth decoders as$\Gamma_\mathcal{T} : \{1, 2, \ldots, 2^{R_\mathcal{T}}\} \to \mathcal{T}$ and $\Gamma_\mathcal{D} : \{1, 2, \ldots, 2^{R_\mathcal{D}}\} \to \mathcal{D}$. Finally, we denote the rendering scheme as $\Upsilon : \mathcal{F} \to \mathcal{G}$. Each rendering scheme has two parts: first, the projection into intermediate view using a few close reference views and their associated depth maps and second, filling the holes that are not covered by any of these reference views. In this paper, we are using only the two closest reference views for rendering. Furthermore, we assume in our theoretical analysis that we have no hole in the reconstructed images. Thus, rendering becomes a simple projection of the closer reference views on an intermediate view using depth information. As we explained in Section 1, the main reason behind this decision is designing a rate allocation method that is independent of underlying inpainting method.

Let us denote the decoded data as $\hat{f} = \Gamma_R(\mathcal{E}_R(f))$. The distortion[2] in the rendered version of the data, $\hat{g} = \Upsilon(\hat{f})$, and the original version, $g = \Upsilon(f)$, is given by

$$D\left(g, \hat{g}\right) = \sum_{i=0}^{p-1} \|t_i - \hat{t}_i\|_2 + \sum_{j=0}^{(p-1)q-1} \|v_j - \hat{v}_j\|_2. \tag{2}$$

We finally define the distortion of the coding scheme as the distortion of the encoding algorithm in the least favorable case, i.e.,

$$D_{\mathcal{E},\Gamma,\Upsilon}(R) = \sup_{g \in \mathcal{G}} D\left(g, \hat{g}\right). \tag{3}$$

When the encoding, decoding, and rendering strategies are clear from the context, we use a simpler notation, $D(R)$, and call it the RD function.

3 Theoretical analysis

In this section, we propose a RD function based on our simple model of scenes $\mathcal{H}^Q(\Omega)$. We first consider a simple camera configuration $\mathcal{B}_1^1(\mathcal{P})$ with only one reference view and one virtual view. Then we extend analysis to more virtual views with camera configuration $\mathcal{B}_q^1(\mathcal{P})$ and to more reference views with configuration $\mathcal{B}_q^p(\mathcal{P})$. For each class of functions, the RD analysis is built in the wavelet domain where the distortion is the distance between the original and coded wavelet coefficients. The distortion in wavelet domain is equal to the distortion in the signal domain when wavelets form an orthonormal basis, and the wavelet representation of our virtual and reference views simplifies the RD analysis. Assuming that coding has negligible effect on the average signal value, then we can ignore the distortion in the lowest frequency band. Therefore, in the following analysis, we only focus on the distortion of coefficients of high frequency bands. In all the proofs, we assume that the wavelets have a finite support of length ℓ and that their first moments are equal to zero.

[2] In this paper, we consider the ℓ_2 distortion. However, extensions to other error norms are straightforward.

Theorem 1 *The coding scheme that uses wavelet-based texture coder and uniform quantization depth coder, achieves the following RD function on scene configuration* $\mathcal{H}^1(\Omega)$ *and camera setup:* $\mathcal{B}_1^1(\mathcal{P})$*:*

$$D(R_t, R_d) \sim O\left(2\mu\sigma^2 2^{\alpha R_t} + K\frac{\Delta Z}{Z[Z2^{\beta R_d} + \Delta Z]}\right),$$

where R_t *and* R_d *are the texture, and depth bit rates,* $K = A'R'|T - T'|$ *depends on camera parameters,* $\Delta Z = Z_{max} - Z_{min}$*,* Z_{max}*, and* Z_{min} *are the maximum and minimum depth values in the scene,* Z *is the foreground object depth value,* σ^2 *is the reference frame variance, and* μ*,* α*, and* β *are positive constants.*

Proof For the camera configuration $\mathcal{B}_1^1(\mathcal{P})$, we have $g = \{(t_0, v_0)\}$ with one reference view and one virtual view. In all the proofs, we consider for the sake of simplicity that there is no occluded region. Inspired by [32], we consider the same quantization level for each wavelet coefficient. This suboptimal choice of quantization will only affect constant factors of the RD function and will not change the final upper bound equation. In addition, we consider a quantization-based coder for depth map coding that simply splits depth image into uniform square areas; for each square, the average depth is quantized and coded. Therefore, if we assign b bits for coding each wavelet coefficient in the reference frame and b' bits for coding each depth value, there will be three sources of distortion after decoding and rendering at the decoder side.

First, at every scale s, the number of nonzero wavelet coefficients is $3 \times d\Omega\ell 2^s$, where $d\Omega$ is the boundary length of Ω in v_0, and the factor 3 is related to the three wavelet bands. Using the definitions of Section 2.1, the magnitude of coefficients at scale s of a standard wavelet decomposition is bounded by

$$\begin{aligned}&\left|C^1_{s,n_1,n_2}\right| \leq \\ &\int_{t_0}^{t_0+\ell 2^{-s}} \int_{t_0'}^{t_0'+\ell 2^{-s}} \left|f(t_1,t_2)\right| \left|\Psi^1_{s,n_1,n_2}(t_1,t_2)\right| dt_1 dt_2 \leq \\ &2^s \int_{t_0}^{t_0+\ell 2^{-s}} \int_{t_0'}^{t_0'+\ell 2^{-s}} \left|\phi\left(2^s t_1 - n\right)\psi\left(2^s t_2 - n\right)\right| dt_1 dt_2 \leq \\ &2^{-s}. \end{aligned} \tag{4}$$

We have similar results in case of $|C^2_{s,n_1,n_2}|$ and $|C^3_{s,n_1,n_2}|$. By assigning b bits for coding each coefficient, all the coefficients at scale s with $2^{-s} < 2^{-b-1}$ will be mapped to zero. Therefore, the first source of coding distortion D_1 is

$$D_1 = 3\ell d\Omega \sum_{s=b+2}^{\infty} 2^s \times \left(2^{-s}\right)^2 = c_1 2^{-b} \tag{5}$$

where $c_1 = 12\ell d\Omega$. Note that a factor of 2 is added here because the error due to skipping small wavelet coefficients similarly affects the distortion in both t_0 and v_0.

Then, the depth map quantization also introduces distortion as it leads to shifts in foreground objects. Recall that we are calculating distortion in the wavelet domain. Consider s_1 as the largest scale with wavelet support length that is smaller than the amount of shift in the foreground object. Nonzero wavelet coefficients at scales larger than or equal to s_1 suffer from position changes due to depth coding. Assume that Δ_0 is the maximum position error in v_0 with a b' bits quantization-based depth coder. Then we have $\ell 2^{-s_1-1} < \Delta_0 < \ell 2^{-s_1}$. Hence, our second source of error, D_2, is

$$D_2 = 2 \times 3\ell d\Omega \sum_{s=s_1+1}^{b+1} 2^s \left(2^{-s}\right)^2 = c_1\left(2^{-s_1} - 2^{-b-1}\right). \tag{6}$$

Here, the factor 2 is due to the shift of significant coefficients and to the distortion at its main and shifted location.

Finally, additional distortion is generated by quantization of nonzero coefficients. Using the definitions of b and s_1 for the reference frame t_0, we have large coefficients quantization error at $s \leq b+1$, while for the virtual view v_0, this happens at $s \leq s_1$. Thus, according to Eq. 2, for this third source of distortion, we have

$$\begin{aligned} D_3 &= 3\ell d\Omega\left[\sum_{s=1}^{b+1} 2^j \left(2^{-b-1}\right)^2 + \sum_{s=1}^{s_1} 2^s \left(2^{-b-1}\right)^2\right] \\ &= c_1\left(2^{-b} + 2^{s_1}2^{-2b}\right). \end{aligned} \tag{7}$$

Using Eqs. 5, 6, and 7 and our additive distortion model at Eq. 2, the total distortion is

$$D = c_1\left[2^{-b} + 2^{-s_1} - 2^{-b-1} + 2^{-b} + 2^{s_1}2^{-2b}\right]. \tag{8}$$

From the definitions of s_1 and Δ_0, we have $s_1 \leq b$ and $s_1 \geq \log \Delta_0^{-1} - 1$. Therefore, we can simplify the above equation and estimate the distortion as

$$D = O\left(2^b + \Delta_0\right).$$

The first term only depends on texture coding errors and the second term on depth quantization. We replace the texture coding term with a simple distortion model $\mu\sigma^2 2^{-\alpha R}$ [33],

where μ and α are model parameters, σ^2 is the source variance, and R is the target bit rate. Using the formulation of maximum shift error Δ_0, in [24], for the depth distortion term, we finally have

$$D(R_t, R_d) = O\left(2\mu\sigma^2 2^{-\alpha R_t} + A'R'|T - T'|\frac{Z_{\max}-Z_{\min}}{Z\left[Z2^{\beta R_d}+Z_{\max}-Z_{\min}\right]}\right) \quad (9)$$

where β is another model parameter that depends on depth coding method, and Z is the foreground object depth value. □

We now extend the above analysis to more complex camera configurations. We first consider q virtual views in a $\mathcal{B}_q^1(\mathcal{P})$ configuration.

Theorem 2 *The coding scheme that uses wavelet-based texture coder and a uniform quantization depth coder achieves the following RD function on scene configuration $\mathcal{H}^1(\Omega)$ and camera setup $\mathcal{B}_q^1(\mathcal{P})$:*

$$D(R_t, R_d) \sim O\left((q+1)\mu\sigma^2 2^{\alpha R_t} + \sum_{j=0}^{q-1} K_j \frac{\Delta Z}{Z[Z2^{\beta R_d} + \Delta Z]}\right),$$

where R_T and R_D are the texture and depth coding rates; $K_j = A'_j R'_j |T - T'_j|$, for $j = 0, \ldots, q-1$ depends on camera parameters; $\Delta Z = Z_{max} - Z_{min}$, Z_{max}, and Z_{min} are the maximum and minimum depth values in the scene; Z is the foreground object depth value; σ^2 is the reference frame variance; and μ, α, and β are positive constants.

Proof With q virtual cameras and aggregating the virtual view distortions in the three sources of distortion in the proof of Theorem 1, we have

$$D_1 = c_1(q+1)2^{-b}, \quad (10)$$

$$D_2 = 2 \times 3\ell d\Omega \sum_{j=0}^{q-1} \sum_{s=s_j+1}^{b+1} 2^s \left(2^{-s}\right)^2 = c_1 \left(\sum_{j=0}^{q-1} 2^{-s_j} - q2^{-b-1}\right) \quad (11)$$

and

$$D_3 = c_1 \left(2^{-b} + 2^{-2b} \sum_{j=0}^{q-1} 2^{s_j}\right). \quad (12)$$

We also have $s_j \leq b$ and $s_j \geq \log \Delta_j^{-1} - 1$ for $j = 0 \ldots q-1$; thus, using Eq. 2, we have

$$D = O\left((q+1)2^b + \sum_{j=0}^{q-1} \Delta_j\right).$$

The RD function is then obtained by following exactly the same replacements as in the proof of Theorem 1. □

Finally, we extend the analysis to configurations with more reference views. We assume that we have equally spaced reference cameras and virtual views, and that the number of intermediate views is identical between every two consecutive reference cameras. A weighted interpolation strategy using the two closest reference views is employed for synthesis at each virtual view point. The weights are related to the distances between the virtual view and the corresponding right and left reference views similarly to that in [22]. Theorem 3 provides the RD function in a general camera configuration with p reference views and $(p-1)q$ virtual views.

Theorem 3 *The coding scheme that uses wavelet-based texture coder and a uniform quantization depth coder achieves the following RD function on scene configuration $\mathcal{H}^1(\Omega)$ and camera setup $\mathcal{B}_q^p(\mathcal{P})$:*

$$D\left(R_{t_0}, \ldots, R_{t_{p-1}}, R_{d_0}, \ldots, R_{d_{p-1}}\right) \sim O\left(\sum_{i=0}^{p-1} \mu\sigma_i^2 2^{\alpha R_{t_i}} + \sum_{j=0}^{(p-1)q-1} \left(\frac{d_{j,r}}{d}\right)^2 \times \left[\mu\sigma_l^2 2^{\alpha R_{t_l}} + K_{j,l}\frac{\Delta Z}{Z\left[Z2^{\beta R_{d_l}} + \Delta Z\right]}\right] + \left(\frac{d_{j,l}}{d}\right)^2 \times \left[\mu\sigma_r^2 2^{\alpha R_{t_r}} + K_{j,r}\frac{\Delta Z}{Z\left[Z2^{\beta R_{d_r}} + \Delta Z\right]}\right]\right),$$

where R_{t_i} and R_{d_i} are the texture and depth coding rates for the ith reference view; $\Delta Z = Z_{max} - Z_{min}$, Z_{max}, and Z_{min} are the maximum and minimum depth values in the scene; Z is the foreground object depth value; σ_i^2 is variance of the ith reference view; and μ, α, and β are positive constants. Also, d indicates the distance between each two reference views, and $d_{j,l}$ and $d_{j,r}$ are the distances between jth virtual view and its left and right reference camera views. Similarly, we have $K_{j,l} = A'_j R'_j |T_l - T'_j|$ and $K_{j,r} = A'_j R'_j |T_r - T'_j|$ that depend on the camera parameters.

Proof First, using Theorem 2, we can write the distortion of a reference view, r, and the q virtual views on its left as

$$D\left(R_{t_r}, R_{d_r}\right) = O\left(\mu\sigma_r^2 2^{\alpha R_{t_r}} + \sum_{j=0}^{q-1}\left[\mu\sigma_r^2 2^{\alpha R_{t_r}} + K_{j,r}\frac{\Delta Z}{Z\left[Z2^{\beta R_{d_r}} + \Delta Z\right]}\right]\right) \tag{13}$$

Clearly, the first and second terms define the distortion of the reference and virtual views, respectively. By adding another reference view, l, and using a weighted average of the two closest reference views for synthesizing the virtual views, we have

$$\begin{aligned} D\left(R_{t_r}, R_{t_l}, R_{d_r}, R_{d_l}\right) = \\ O\Bigg(\mu\sigma_r^2 2^{\alpha R_{t_r}} + \mu\sigma_l^2 2^{\alpha R_{t_l}} + \\ \sum_{j=0}^{q-1}\left(\frac{d_{j,r}}{d}\right)^2\left[\mu\sigma_l^2 2^{\alpha R_{t_l}} + K_{j,l}\frac{\Delta Z}{Z\left[Z2^{\beta R_{d_l}} + \Delta Z\right]}\right] + \\ \left(\frac{d_{j,l}}{d}\right)^2\left[\mu\sigma_r^2 2^{\alpha R_{t_r}} + K_{j,r}\frac{\Delta Z}{Z\left[Z2^{\beta R_{d_r}} + \Delta Z\right]}\right]\Bigg) \end{aligned} \tag{14}$$

where d indicates the distance between the two reference cameras, and $d_{j,l}$ and $d_{j,r}$ are the distances between the jth virtual view and its left and right reference camera views. Our weights are simply related to the distance between virtual view and its neighbor reference views. Finally, summing up the terms of Eq. 14 for all reference views leads to the distortion in Theorem 3. □

The above RD analysis is performed on $\mathcal{H}^1(\Omega)$. However, the extension to multiple foreground objects is straightforward by setting $Z = Z_{\min}$.

4 RD optimization

In this section, we show how the analysis in Section 3 can be used for optimizing the rate allocation in multiview coding. Using Theorem 3, the rate allocation problem turns into the following convex nonlinear multivariable optimization problem with linear contraints:

$$\begin{aligned} &arg\min_{\overrightarrow{R}_t, \overrightarrow{R}_d} g_t\left(\overrightarrow{R}_t\right) + g_d\left(\overrightarrow{R}_d\right) \\ &\text{such that } \left\|\overrightarrow{R}_t + \overrightarrow{R}_d\right\|_1 \le R \end{aligned} \tag{15}$$

where

$$g_t\left(\overrightarrow{R}_t\right) = \sum_{i=0}^{p-1}(q+1)\mu\sigma_i^2 2^{\alpha R_{t_i}},$$

$$g_d\left(\overrightarrow{R}_d\right) = \sum_{j=0}^{(p-1)q-1}\left[\left(\frac{d_{j,l}}{d}\right)^2 K_{j,r}\frac{\Delta Z}{Z_{\min}\left[Z_{\min}2^{\beta R_{d_r}} + \Delta Z\right]} + \left(\frac{d_{j,r}}{d}\right)^2 K_{j,l}\frac{\Delta Z}{Z_{\min}\left[Z_{\min}2^{\beta R_{d_l}} + \Delta Z\right]}\right]$$

and R is the total target bit rate. The convexity proof is straightforward since the above optimization problem is the sum of terms in the form $a2^{-bx}$, which are convex. Therefore, it can be solved efficiently using classical convex optimization tools. Note that the above optimization problem is for the general camera configuration $\mathcal{B}_q^p(\mathcal{P})$. The rate allocation for simpler configurations is straightforward by replacing the objective functions with terms from Theorem 1 and 2. We can finally note that the rate allocation strategy is only based on encoder data.

The last issue that we have to address is the choice of the model parameters. There are three parameters—μ, α, and β in Eq. 15—that we estimate using the following offline method. Using the first texture and depth images, we estimate the model parameters by solving the following regression problem

$$\left[\mu^\star, \alpha^\star, \beta^\star\right] = \arg\min_{\mu,\alpha,\beta}\sum_{k=0}^{n-1}\left|D\left(R_k\right) - D^*\left(R_k\right)\right| \tag{16}$$

where n is the number of points in the regression problem; it is further discussed in the next section. $D(R_k)$ is the distortion obtained by our rate allocation strategy in Eq. 15 with target bit rate R_k, and $D^*(R_k)$ is the best possible allocation obtained by a full search method at the same bit rate.

5 Experimental results

In the previous sections, we have studied the bit allocation problem on simple scenes and have extracted a model for estimating RD function of a DIBR multiview coder with wavelet-based texture coding and a quantization-based depth coding. This section studies the RD behavior and the accuracy of the proposed model on real scenes where JPEG2000 is used for coding depth and reference images.

We use 12 datasets as it is shown in Fig. 3. Here, *Ballet* and *Breakdancers* datasets are from Interactive Visual Group of Microsoft Research [34], and the others are selected from Middlebury stereo datasets [35, 36]. In our simulations, gray-scale versions of the images in these datasets are used. Each view in the *Ballet* and *Breakdancers* datasets contains 100 temporally consecutive frames, and all the numerical results in this section are the average of the frames with temporal indices 0, 49, and 99. The camera intrinsic and extrinsic parameters $\mathcal{P}$, and the scene parameters $Z_{\min}$ and $Z_{\max}$, are set to the values given by datasets. In cases where the parameters are changed to study the model under some special aspects, we mention the parameter values explicitly.

In an offline stage using Eq. 16, we adjust the parameters μ, α, and β in Eq. 15 at four regression points, i.e., $n = 4$, for each dataset. Note that in all tests, the parameter estimation is performed over frames that are not included in the performance evaluation. For instance, in the case of *Ballet* and *Breakdancers*, the parameters are calculated using a random frame that is different from frames with indices 0, 49, or 99. The parameter values are fixed for the different camera configurations. In the following, we study the RD model of Eq. 15 for rate allocation in different camera configurations, namely $\mathcal{B}_1^1(\mathcal{P})$, $\mathcal{B}_6^1(\mathcal{P})$, and $\mathcal{B}_3^2(\mathcal{P})$. As a comparison criterion, we use the optimal allocation that is obtained by rendering all the intermediate views and searching the whole RD space for the allocation with minimal distortion.

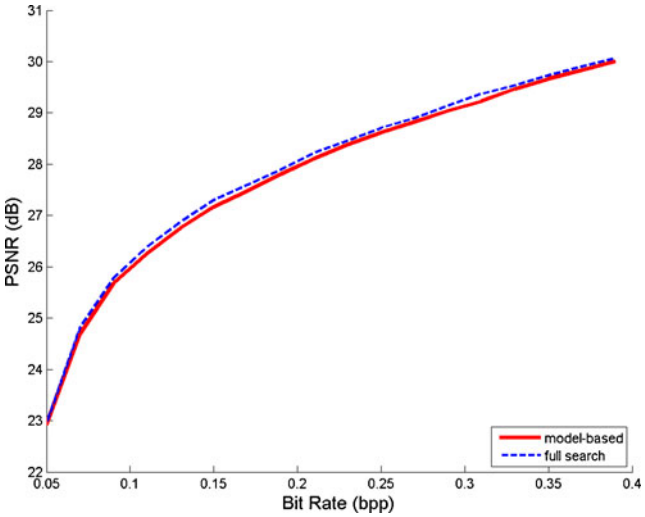

Fig. 4 Comparison of the coding performance for $\mathcal{B}_1^1(\mathcal{P})$ using the proposed allocation method and the best allocation in terms of PSNR at rates ranging from 0.05 to 0.4 bpp. The performance has been averaged over our 12 datasets

Fig. 3 Test datasets (*from top-left to bottom right*): *Aloe*, *Art*, *Baby*, *Ballet*, *Bowling*, *Breakdancers*, *Cloth*, *Cones*, *Midd*, *Rocks* and *Wood*

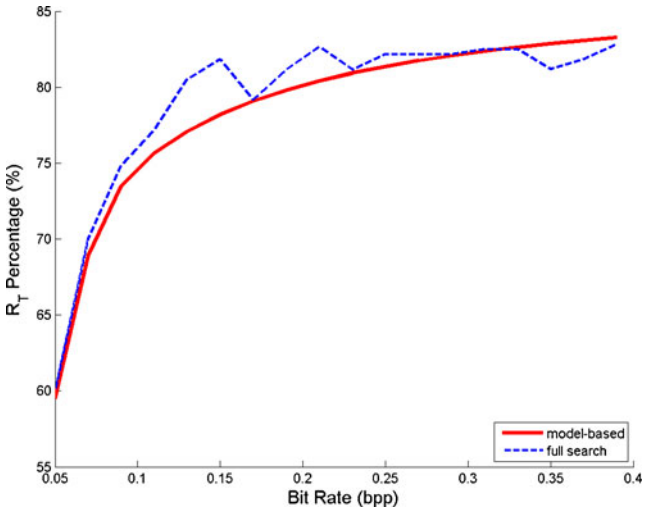

Fig. 5 Rate allocation results of $\mathcal{B}_1^1(\mathcal{P})$ using our proposed method and the optimal allocation in terms of R_t percentage of total rates, ranging from 0.05 to 0.4 bpp. The results have been averaged over 12 datasets

Finally, as we want to keep our model independent of any special strategy for filling occluded regions, all occluded regions are ignored in the distortion and peak signal-to-noise ratio (PSNR) calculations.

5.1 $\mathcal{B}_1^1(\mathcal{P})$ configuration

We start with the $\mathcal{B}_1^1(\mathcal{P})$ camera setup which is a simple configuration with one reference view and only one virtual view. We use the cameras 0 and 1 of the datasets as reference and target cameras, respectively. All camera-related parameters in Eq. 15 are set accordingly.

A RD surface is first generated offline for the desired bit rate range to generate the distortion benchmark values. In our study, R_t and R_d are set between 0.02 and 0.4 bpp with 0.02-bpp steps. It means that R_t and R_d axes are discretized into 20 values. Since we are coding only one reference view and one depth map, this range of bit rate is pretty reasonable. The RD surface is generated by actual coding of the texture and depth images at each (R_t, R_d) pair and by calculating the distortion after decoding and synthesis.

Then, for each target bit rate, R, the optimal rate allocation is calculated by cutting the above surface with a plane $R_t + R_d = R$ and minimizing the distortion. If the minimum point occurs between grid points (because we have a discretized surface), a bicubic interpolation is used to estimate the optimal allocation. Here, R is set between 0.05 and 0.4 bpp with a 0.02-bpp step. Figure 4 provides compression performance of DIBR coder in terms of the PSNR averaged over all datasets. The estimated curve is generated by solving the optimization problem provided in Eq. 15 with the proposed RD model. The average and maximum difference between the model-based and optimal curves are 0.09 and 0.30 dB, respectively. Figure 5 shows the R_t percentage of the best and model-based allocations versus the bit rate where the percentage has been averaged over the 12 test datasets. Clearly, our model-based allocation follows closely the best allocation.

We study now the performance of a priori fixed rate allocations, which are commonly adopted in practice. We consider R_t relative to the total budget fixed at 80 % as the common a priori allocation [15–17]. Table 1 shows the average PSNR loss compared to the best allocation for our 12 test datasets. We compare the performance of the rate allocation estimated with our RD model, and we show that our allocation is always better. Figure 5 further shows that, using a model-based allocation instead of a priori allocation is more important at low bit rates (on average less than 0.15 bpp). This is the reason why we have significant differences between average and maximum PSNR loss in fixed allocation results. In our proposed allocation, the results are close to optimal in all datasets as the model adapts to

Table 1 Performance penalty in $\mathcal{B}_1^1(\mathcal{P})$

Dataset	*Aloe*	*Art*	*Baby*	*Ballet*	*Bowling*	*Break*	*Cloth*	*Cones*	*Lamp*	*Midd*	*Rocks*	*Wood*	Overall
$R_t = 80$ %													
Avg	0.12	0.12	0.15	0.21	0.22	0.10	0.13	0.14	0.07	0.08	0.13	0.28	0.15
Max	0.50	0.84	1.17	1.12	0.51	0.91	1.18	1.03	0.31	0.24	0.44	1.90	0.85
DMDA													
Avg	0.65	0.69	0.75	0.97	1.09	1.44	1.05	0.67	0.84	1.37	1.09	1.05	0.97
Max	0.80	1.00	1.06	1.30	1.60	1.80	1.27	0.95	1.35	1.94	1.50	1.67	1.35
Our model													
Avg	0.06	0.07	0.06	0.14	0.17	0.10	0.06	0.08	0.14	0.06	0.09	0.11	0.09
Max	0.29	0.22	0.30	0.34	0.33	0.25	0.21	0.24	0.28	0.17	0.48	0.49	0.30

Comparison between the proposed model, a priori allocation policy, and depth map distortion-based allocation in terms of average and maximum differences to the best achievable PSNR at total rates ranging from 0.05 to 0.4 bpp

the scene content. In datasets with highly textured regions or close-to-camera objects, like *Wood* and *Ballet*, we have more significant benefits with our adaptive allocation. In the cases of *Wood* and *Ballet*, the model-based approach performs better than fixed allocation by 1.41 and 0.78 dB, respectively. The last column of shows the average benefit of our model compared to a fixed rate allocation with 80 % of rate in texture coding. In addition to fixed allocation, we provide the results of a rate allocation strategy similar to H.264/AVC coder [18]. In this coder rate, allocation is performed directly based on the depth map distortion. We call this allocation method depth map distortion-based allocation (DMDA). Table 1 shows the significant improvement of using our model in contrast to DMDA, which is expected due to the indirect effect of depth map distortion on the final view quality.

Finally, to study the performance of our proposed model on various frames of one sequence, Table 2 provides PSNR loss results of frames 10 to 90 from the *Ballet* dataset. Frame 0 is used for parameter estimation. The overall gain of using our model in contrast to fixed allocation reaches 0.4 dB.

5.2 $\mathcal{B}_q^1(\mathcal{P})$ configuration

In this section, we study the allocation problem for camera configurations with multiple virtual views. The camera 4 of the *Ballet* and *Breakdancers* datasets is used as the reference camera, and six virtual cameras separated by 1 cm are considered, three at each side of the reference camera. At each side, the parameters of the virtual cameras are set according to cameras 3 and 5 in the dataset, respectively. For the other datasets, the settings are the same except that we are using the parameters of the first stereo camera in all cases.

The optimal allocation process is obtained similarly to Section 5.1. The optimal RD surface is generated offline, for R_t and R_d rates between 0.05 and 0.4 bpp with 0.02-bpp steps. Then, at each total bit rate R, the best allocation is calculated using interpolation over this RD surface. The model-based allocation is the result of solving Eq. 15 for $\mathcal{B}_6^1(\mathcal{P})$. The reported distortion is the average distortion over all six virtual views and the reference view and also, in the case of the *Ballet* and *Breakdancers* datasets, over the three representative frames in each set, i.e., frames 0, 49, and 99.

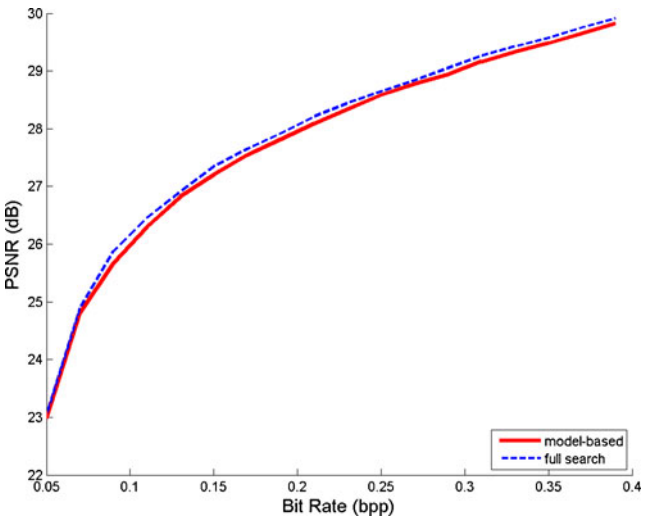

Fig. 6 Comparison of the coding performance for $\mathcal{B}_6^1(\mathcal{P})$ using the proposed allocation method and the best allocation in terms of PSNR at rates, ranging from 0.05 to 0.4 bpp. The results have been averaged over 12 datasets

Figure 6 represents the performance in terms of PSNR with respect to target bit rate, R, where R varies between 0.05 and 0.4 bpp. The two curves correspond to the best allocation and the model-based allocation averaged over all 12 test datasets. The average and maximum amount of loss due using our model is 0.11 and 0.34 dB, respectively. The corresponding performance penalties are 0.17 and 0.91 dB for the R_t percentage at 80 % and 0.88 and 1.27 dB for the rate allocation based on depth map distortion. Although the average of the fixed allocation is close to our model, it has large variances, which is mainly due to inefficient allocation at low bit rates. Figure 7 clarifies this claim by presenting the best and the model-based allocation in terms of percentage of the total rate allocated to R_t, for different values of

Table 2 Performance penalty in $\mathcal{B}_1^1(\mathcal{P})$

Frame number	10	20	30	40	50	60	70	80	90	Overall
$R_t = 80$ %										
Avg	0.26	0.19	0.27	0.26	0.24	0.21	0.20	0.25	0.22	0.23
Max	0.93	0.84	0.74	1.00	0.93	0.72	0.59	0.97	0.99	0.86
Our model										
Avg	0.22	0.13	0.18	0.12	0.17	0.17	0.19	0.19	0.21	0.18
Max	0.49	0.31	0.55	0.38	0.45	0.49	0.43	0.47	0.51	0.46

Comparison between the proposed model and a priori allocation policy in terms of average and maximum differences to the best achievable PSNR at total rates ranging from 0.05 to 0.4 bpp over frames 10 to 90 of the *Ballet* dataset

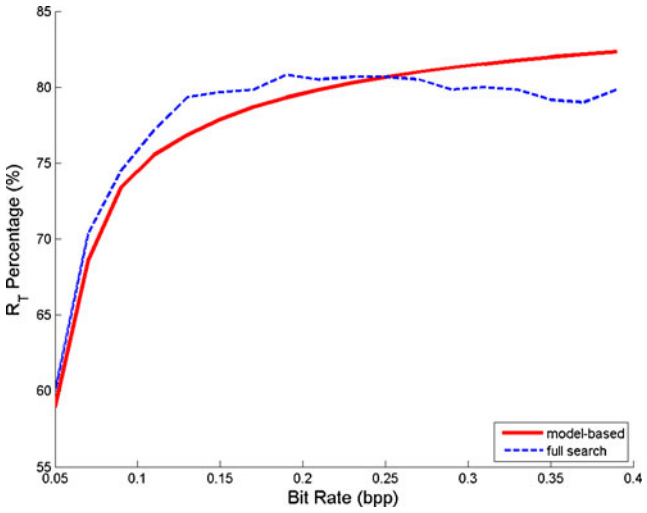

Fig. 7 Rate allocation results of $\mathcal{B}_6^1(\mathcal{P})$ using our proposed method and the optimal allocation in terms of R_t percentage of total rates, ranging from 0.05 to 0.4 bpp. The results have been averaged over 12 datasets

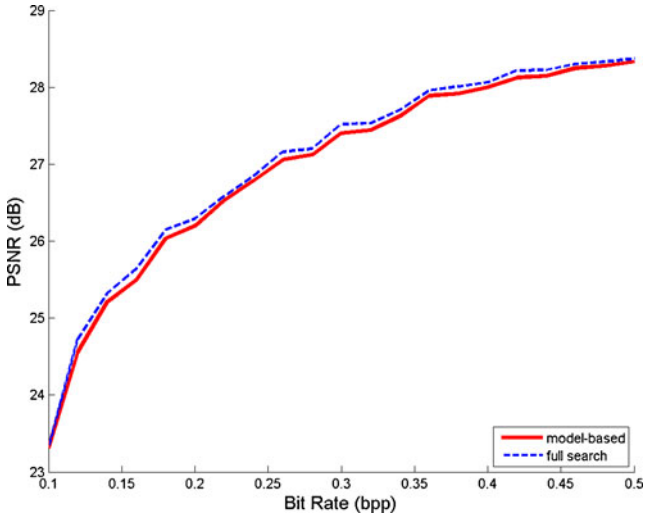

Fig. 8 Comparison of the coding performance for $\mathcal{B}_3^2(\mathcal{P})$ using the proposed allocation method and the best allocation in terms of PSNR at rates, ranging from 0.1 to 0.5 bpp. The results have been averaged over 12 datasets

R. As it is shown in this figure, for a large portion of bit rates, approximately higher than 0.1 bpp, the optimal allocation is around 80 % which is the reason both schemes have similar PSNR loss. But at lower bit rates, the adaptive allocation plays a more significant role and, due to this different behavior, we have 0.91 dB in PSNR loss for the fixed allocation. Clearly, our model again performs very close to the optimal allocation at all bit rates. This yields to improvements over a priori rate allocations as given in Table 3 in case of $\mathcal{B}_6^1(\mathcal{P})$. Similar to $\mathcal{B}_1^1(\mathcal{P})$ configuration, the benefit of the model-based allocation in contrast to fixed allocation is more significant in textured images, like *Wood*, or datasets with close-to-camera objects, like *Ballet*. In these two cases, our proposed method outperforms the fixed allocation by up to 1.48 and 1.41 dB, respectively. Also, for all datasets, we have significant improvement over DMDA, which uses depth map distortion for rate allocation.

5.3 $\mathcal{B}_q^p(\mathcal{P})$ configuration

We now consider the most general configuration, $\mathcal{B}_q^p(\mathcal{P})$, with two reference cameras ($p = 2$) and three equally spaced virtual views between them ($q = 3$). For the *Ballet* and *Breakdancers* datasets, the cameras 4 and 5 are considered as the two reference views, and A'_j and R'_j, $j = 1, 2, 3$, for virtual views are set as the average of intrinsic and rotation matrices of our reference cameras. For the other ten datasets, the settings are set according to the provided stereo cameras. Each virtual view v_j is generated in two steps. If π is the position of v_j, then each of the reference views

Table 3 Performance penalty in $\mathcal{B}_6^1(\mathcal{P})$

Dataset	*Aloe*	*Art*	*Baby*	*Ballet*	*Bowling*	*Break*	*Cloth*	*Cones*	*Lamp*	*Midd*	*Rocks*	*Wood*	Overall
$R_t = 80$ %													
Avg	0.11	0.15	0.15	0.40	0.27	0.08	0.14	0.12	0.11	0.05	0.13	0.29	0.17
Max	0.51	1.13	1.14	1.83	0.46	0.78	1.07	1.02	0.35	0.17	0.42	2.03	0.91
DMDA													
Avg	0.67	0.58	0.77	0.57	0.82	1.49	0.95	0.67	0.77	1.30	1.07	0.85	0.88
Max	0.84	1.12	1.10	0.77	1.33	1.83	1.23	0.95	1.25	1.70	1.60	1.57	1.27
Our model													
Avg	0.09	0.09	0.05	0.24	0.11	0.11	0.11	0.07	0.14	0.04	0.08	0.14	0.11
Max	0.36	0.36	0.24	0.42	0.30	0.59	0.27	0.21	0.23	0.13	0.42	0.55	0.34

Comparison between the proposed model, a priori allocation policy, and depth map distortion based allocation in terms of average and maximum differences to the best achievable PSNR at total rates ranging from 0.05 to 0.4 bpp

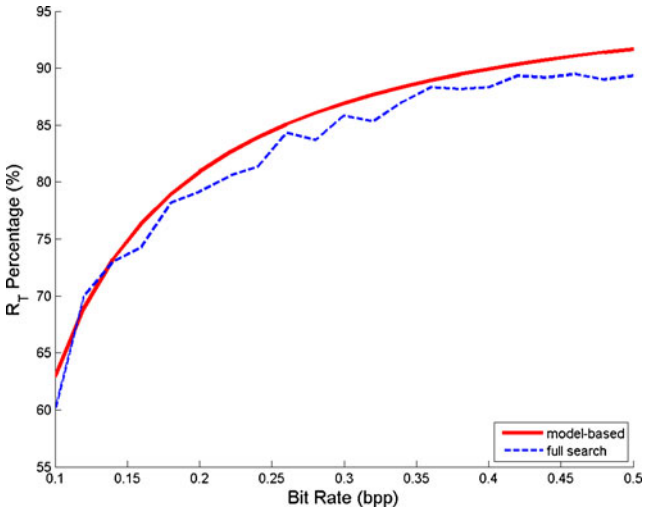

Fig. 9 Rate allocation results of $\mathcal{B}_3^2(\mathcal{P})$ using our proposed method and the optimal allocation in terms of R_t percentage of total rates, ranging from 0.1 to 0.5 bpp. The results have been averaged over our 12 datasets

are projected into π using depth map information. This step produces $v_{j,r}$ and $v_{j,l}$ as projection results from the right and left cameras, respectively. Next, we construct the synthetic view as

$$v_j = \frac{d_{j,l}}{d} v_{j,r} + \frac{d_{j,r}}{d} v_{j,l} \tag{17}$$

where d is the distance between two reference cameras, while $d_{j,l}$ and $d_{j,r}$ are the distances between v_j and the left and right reference cameras, respectively.

The allocation problem in this case consists of distributing the available bit budget between two reference views and two depth maps. For comparison purposes, we calculate a RD hypersurface of the best allocation with R_{t_1}, R_{t_2}, R_{d_1}, and R_{d_2} ranging from 0.05 to 0.5 bpp with 0.02 steps. Then for each target bit rate R, the best allocation is the minimum of the resulting curve from cutting this hypersurface with the hyperplane $R_{t_1} + R_{t_2} + R_{d_1} + R_{d_2} = R$.

Figure 8 compares the best allocation and the model-based allocation in Eq. 15 as an average over all 12 test datasets of this study, for target bit rates ranging from 0.1 to 0.5 bpp. Our allocation model only yields a 0.09-dB loss in average and a maximum loss of 0.34 dB compared to the optimal allocation. Figure 9 shows the best and estimated allocations in terms of the percentage of the texture bits $(R_{t_1} + R_{t_2})$ relatively to the total bit rate. The advantage of using our model over the commonly used strategy of a priori rate allocation is shown in Table 4. In the a priori allocation, the bit rate assigned to each pair of reference view and depth map is equal. For instance, in $\mathcal{B}_3^2(\mathcal{P})$, if the total bit rate is 0.4 bpp for the a priori allocation of 80 %, $R_{t_1} = R_{t_2} = 0.16$ and $R_{d_1} = R_{d_2} = 0.04$ bpp. Our model performs better than the a priori allocation by up to 0.64 dB, which is due to adaptivity to content and setup. From Tables 1 to 4, we can conclude that the best performance of an a priori allocation strategy depends on the number of reference and virtual views and on the scene content. However, our model-based allocation works well in all cases and gives the opportunity to determine the number of virtual views only at decoder side. Also, the nonoptimality of using depth map distortion in rate allocation is proved experimentally in this setting, too. Our model outperforms the DMDA method by 1.60 dB on average and up to 1.81 dB at maximum.

6 Conclusion

We have addressed the problem of RD analysis of multi-view coding in a depth image-based rendering context. In particular, we have shown that the distortion in the reconstruction of camera and virtual views at decoder is driven by the coding artifacts in both the reference images and

Table 4 Performance penalty in $\mathcal{B}_3^2(\mathcal{P})$

Dataset	*Aloe*	*Art*	*Baby*	*Ballet*	*Bowling*	*Break*	*Cloth*	*Cones*	*Lamp*	*Midd*	*Rocks*	*Wood*	Overall
$R_t = 80$ %													
Avg	0.43	0.19	0.51	0.33	1.29	0.17	0.42	0.49	1.35	0.50	0.28	0.62	0.55
Max	0.72	0.36	0.96	0.68	2.16	0.47	0.87	0.77	2.27	0.87	0.52	1.13	0.98
DMDA													
Avg	1.33	1.40	1.24	1.30	2.50	1.95	1.61	1.37	2.69	1.89	1.44	1.59	1.69
Max	1.66	1.69	1.77	1.66	3.50	2.20	1.90	1.74	3.35	2.31	1.71	2.25	2.15
Our model													
Avg	0.02	0.10	0.19	0.11	0.02	0.14	0.19	0.11	0.03	0.05	0.08	0.03	0.09
Max	0.20	0.37	1.11	0.35	0.10	0.37	0.41	0.33	0.11	0.21	0.21	0.29	0.34

Comparison between the proposed model, a priori allocation policy, and depth map distortion based allocation in terms of average and maximum differences to the best achievable PSNR at total rates ranging from 0.1 to 0.5 bpp

the depth maps. We have proposed a simple yet accurate model of the RD characteristics for simple scenes and different camera configurations. We have used our novel model for deriving effective allocation of bit rate between reference and depth images. One of the interesting features of our algorithm, beyond its simplicity, consists in avoiding the need for view synthesis at the encoder, contrary to what is generally used in state-of-the-art solutions. We finally demonstrate in extensive experiments that our simple model stays valid to complex multiview scenes with arbitrary numbers of reference and virtual views. It leads to an effective allocation of bit rate with close-to-optimal quality under various rate constraints. In particular, our rate allocation outperforms common strategies based on a priori rate allocation, since it is adaptive to the scene content. Finally, we plan to extend our analysis to multiview video encoding where motion compensation poses nontrivial challenges in rate allocation algorithms due to additional coding dependencies.

Acknowledgments This work has been partially supported by Iran Ministry of Science, Research and Technology and the Swiss National Science Foundation under grant 200021_126894.

References

1. Zhang Z (2012) Microsoft kinetic sensor and its effect. IEEE Multimed 19:4–10
2. Merkle P, Smolic A, Muller K, Wiegand T (2007) Efficient prediction structures for multiview video coding. IEEE Trans Circ Syst Video Technol 17(11):1461–1473
3. Vetro A, Wiegand T, Sullivan G (2011) Overview of the stereo and multiview video coding extensions of the H.264/MPEG-4 AVC standards. Proc IEEE 99(4):626–642
4. Müller K, Merkle P, Wiegand T (2011) 3D video representation using depth maps. Proc IEEE 99(4):643–656
5. Tian D, Lai P, Lopez P, Gomila C (2009) View synthesis techniques for 3D video, SPIE Optical Engineering + Applications. In: International society for optics and photonics, pp 74430T–74430T
6. Fehn C (2004) Depth-image-based rendering (DIBR), compression and transmission for a new approach on 3D-TV. Proc SPIE Stereosc Image Process Render 5291:93–104
7. Shao F, Jiang GY, Yu M, Zhang Y (2011) Object-based depth image-based rendering for a three-dimensional video system by color-correction optimization. Opt Eng 50:047006–047006-10
8. Oh K-J, Yea S, Ho Y-S (2009) Hole filling method using depth based in-painting for view synthesis in free viewpoint television and 3-D video. In: Proceedings of picture coding symposium. Chicago
9. Cheng C-M, Lin S-J, Lai S-H, Yang J-C (2008) Improved novel view synthesis from depth image with large baseline. In: Proceedings of international conference on pattern recognition. Tampa
10. Merkle P, Morvan Y, Smolic A, Farin D, Muller K, de With PHN, Wiegand T (2009) The effects of multiview depth video compression on multiview rendering. Signal Processing: Image Commun 24:73–88
11. Maitre M, Do MN (2010) Depth and depth-color coding using shape-adaptive wavelets. J Vis Commun Image Repr 21:513–522
12. Donoho DL (1999) Wedgelets: nearly minimax estimation of edges. Ann Stat 27:859–897
13. Le Pennec E, Mallat S (2005) Sparse geometrical image approximation with bandelets. IEEE Trans Image Process 14(4):423–438
14. Maleki A, Rajaei B, Pourreza HR (2012) Rate-distortion analysis of directional wavelets. IEEE Trans Image Process 21(2):588–600
15. Sanchez A, Shen G, Ortega A (2009) Edge-preserving depth-map coding using graph-based wavelets. In: Proceedings asilomar conference signals, systems computers. Los Angeles, pp 578–582
16. Daribo I, Tillier C, Pesquet-Popescu B (2008) Adaptive wavelet coding of the depth map for stereoscopic view synthesis. In: IEEE proceedings of international workshop on multimedia signal processing. Paris, pp 413–417
17. Milani S, Zanuttigh P, Zamarin M, Forchhammer S (2011) Efficient depth map compression exploiting segmented color data. In: Proceedings of international conference on multimedia and expo. pp 1–6
18. ITU-T and ISO/IEC JTC 1 Advanced video coding for generic audiovisual services TU-T recommendation. ITU Recommendation H.264 and ISO/IEC 14496-10 (MPEG-4 AVC), Version 1: May 2003; Version 2: May 2004; Version 3: Mar. 2005 (including FRExt extension); Version 4: Sep. 2005; Version 5 and Version 6: Jun. 2006; Version 7: Apr. 2007; Version 8: Jul. 2007 (including SVC extension); Version 9: Jul. 2009 (including MVC extension), Switzerland
19. ISO/IEC JTC1/SC29/WG11 (2008) Text of ISO/IEC 14496-10:200X/ FDAM 1 multiview video coding. Doc. N9978. ISO/IEC, Hannover
20. Ekmekcioglu E, Velisavljevic V, Worrall ST (2011) Content adaptive enhancement of multi-view depth maps for free viewpoint video. IEEE J Selected Topics Sig Proc 5(2):352–361
21. Lee JY, Wey H-C, Park D-S (2011) A fast and efficient multi-view depth image coding method based on temporal and inter-view correlations of texture images. IEEE Trans Circ Syst Video Technol 21(12):1859–1868
22. Liu Y, Ma S, Huang Q, Zha D, Gao W, Zhang N (2009) Compression-induced rendering distortion analysis for texture/depth rate allocation in 3d video compression. In: Proceedings data compression conference. Beijing, pp 352–361
23. Nguyen HT, Do MN (2009) Error analysis for image-based rendering with depth information. IEEE Image Proc 18(4):703–716
24. Kim W-S, Ortega A, Lai P, Tian D, Gomila C (2010) Depth map coding with distortion estimation of rendered view. Proc SPIE Visual Inf Proc Commun 7543:75430B–75430B-10
25. Oh BT, Lee J, Park D-S (2011) Depth map coding based on synthesized view distortion function. IEEE J Selected Topics Sig Proc 5(7):1344–1352
26. Davidoiu V, Maugey T, Pesquet-Popescu B, Frossard P (2011) Rate distortion analysis in a disparity compensated scheme. In: Proceedings IEEE international conference acoustics, speech, and signal processing. Paris, pp 857–860
27. Maitre M, Do MN (2008) Joint encoding of the depth image based representation using shape-adaptive wavelets. In: Proceedings IEEE international conference image processing (ICIP). Urbana, pp 1768–1771
28. Wang Q, Ji X, Dai Q, Zhang N (2012) Free viewpoint video coding with rate-distortion analysis. IEEE Trans Circ Syst Video Technol 22(6):875–889
29. Cheung G, Velisavljevic V, Ortega A (2011) On dependent bit allocation for multiview image coding with depth-image-based rendering. IEEE Trans Image Proc 20(11):3179–3194
30. Gelman A, Dragotti PL, Velisavljevic V (2012) Multiview image coding using depth layers and an optimized bit allocation. IEEE Trans Image Proc 21(9):4092–4105

31. Mallat S (1997) A wavelet tour of signal processing. Academic, San Diego
32. Prandoni P, Vetterli M (1999) Approximation and compression of piecewise smooth functions. Phil Trans Royal Soc London 357(1760):2573–2591
33. Cover TM, Thomas JA (2006) Elements of information theory (telecommunications and signal processing). Wiley, New York
34. Anonymous (2004) Sequence microsoft ballet and breakdancers. http://research.microsoft.com/en-us/um/people/sbkang/3dvideodownload
35. Anonymous (2005) Middlebury stereo dataset. http://vision.middlebury.edu/stereo/data/scenes2005
36. Anonymous (2006) Middlebury stereo dataset. http://vision.middlebury.edu/stereo/data/scenes2006/