This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2019.2919198, IEEE Transactions on Image Processing

Occlusion-Aware Depth Map Coding Optimization Using Allowable Depth Map Distortions

Pan Gao and Aljosa Smolic

***Abstract*—In depth map coding, rate-distortion optimization for those pixels that will cause occlusion in view synthesis is a rather challenging task, since the synthesis distortion estimation is complicated by the warping competition and the occlusion order can be easily changed by the adopted optimization strategy. In this paper, an efficient depth map coding approach using allowable depth map distortions is proposed for occlusion-inducing pixels. Firstly, we derive the range of allowable depth level change for both zero disparity error case and non-zero disparity error case with theoretic and geometrical proofs. Then, we formulate the problem of optimally selecting the depth distortion within allowable depth distortion range with the objective to minimize the overall synthesis distortion involved in the occlusion. The unicity and occlusion order invariance properties of allowable depth distortion range is demonstrated. Finally, we propose a dynamic programming based algorithm to locate the optimal depth distortion for each pixel. Simulation results illustrate the performance improvement of the proposed algorithm over other state-of-the-art depth map coding optimization schemes.**

***Index Terms*—Depth map coding, allowable depth distortions, occlusion, rate-distortion optimization.**

I. Introduction

With the growing demand for a more immersive visual experience, texture-plus-depth has become a popular data representation, where a decoder can synthesize a virtual view at any desired viewpoint using transmitted texture and depth map via depth-image-based rendering (DIBR) [1]. As the texture can be effectively compressed using conventional 2-D video techniques, how to efficiently compress depth map has attracted considerable attention recently [2]. Since depth map is used to provide geometry information for view synthesis instead of being viewed by end users, it is acknowledged that the synthesized view distortion should be employed as quality measurement for 3-D video coders [3]. Therefore, it is of importance to explore the the relationship between the view synthesis distortion and depth distortion and design corresponding efficient coding strategies.

Generally speaking, the relationship between the depth distortion and view synthesis distortion can be broadly classified into two categories, i.e., one-to-one mapping and many-to-one mapping. In one-to-one mapping, there are a large number of view synthesis distortion models that have been proposed, such as the linear model [4], autoregressive model [5], multiplicative model [6], regional linear model [7], cubic model [8], quintic model [9], *etc*. In contrast, there are few works focused on the investigation of many-to-one relation. Specifically, many-to-one mapping here refers to that multiple depth distortions are mapped to an identical disparity error, which is due primarily to significantly less number of disparity levels compared to the number of available depth levels in practice. The representative works in the second category are the depth no-synthesis-error (D-NOSE) model in [10] and the allowable depth distortion model in [11]. Generally speaking, all the proposed synthesis distortion and allowable depth distortion models can be employed in depth coding to improve coding performance of 3-D video to some extent. However, almost all of them do not consider the effect of occlusion, which is a common phenomenon in view synthesis. Though, in [6], [8] and [9], the effect of occlusion on the prediction of view synthesis distortion is considered, they are tackled in non-allowable depth distortion controlled depth coding optimization. Furthermore, these algorithms implicitly or explicitly skip the synthesized pixels that are occluded, and only calculate the synthetic distortion of the winning pixel as the overall distortion of the occlusion. Consequently, the estimated distortion may be inappropriate for occlusion-specific depth optimization, since there are still some distortions occurred by the process of shifting the occluded pixels from original positions to destination positions, although the associated texture distortions are zero intuitively.

In this paper, we consider the occlusion effect on depth map coding optimization. We model the occlusion distortion as the summation of the view synthesis distortions of a group of pixels that are involved in the occlusion, which can better serve as optimization criterion for each depth pixel. We consider the inherent distortion dependency between pixels established in the warping competition in occlusion. In general, occlusion-specific depth map coding optimization is a lot more challenging since the occlusion order can be significantly changed by the adopted optimized scheme. To this end, we introduce allowable depth errors into depth coding optimization in occlusion scenario, which can potentially improve the coding performance by properly adjusting the depth distortions without alteration of occlusion order[1]. A formal definition of allowable depth error range is explicitly given, where we consider the intervals of allowable depth level variation for both zero disparity error and non-zero disparity error cases as in [11]. However, different from [11],

P. Gao is with College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China. He was with School of Computer Science and Statistics, Trinity College Dublin, Dublin 2, Ireland. A. Smolic is with School of Computer Science and Statistics, Trinity College Dublin, Dublin 2, Ireland (gaopa@scss.tcd.ie, smolica@scss.tcd.ie).

A preliminary study of this work has been presented at ICIP 2018, Athens, Greece, and is available online [12]. This work has emanated from research conducted with the financial support of Science Foundation Ireland (SFI) under the Grant Number 15/RP/2776.

[1]In a rate-distortion framework, the selection of depth error within allowable depth error range will not change the view synthesis distortion for each pixel. However, the overall view synthesis distortion for occlusion may differ for different depth distortions due to the distortion dependency between pixels. Further, the bit rate will change as the depth distortion changes. The potential coding gain that can be achieved by exploiting the allowable depth distortion redundancy is demonstrated in [11].

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2019.2919198, IEEE Transactions on Image Processing

we theoretically derive and prove a new range of allowable depth level change that exists for the non-zero disparity error-inducing depth error. Finally, we formulate the optimization problem as optimally selecting the possible allowable depth error from the derived range for each pixel such that the overall distortion involved in the occlusion is minimized. We propose a dynamic programming solution to efficiently find the vector of allowable depth level changes.

A preliminary study of this work is presented in [12], in which rate-distortion optimization of depth map coding in occlusion is formulated using traditional-model derived allowable depth distortions. In this paper, we extend [12] from two aspects. Firstly, we give a detailed theoretical proof of the new allowable depth distortion range for non-zero disparity error case, which is revealed to be a generic form of interval of allowable depth distortion for any depth level with any amount of depth distortion. Secondly, The occlusion order invariance property of allowable depth distortion range is justified for non-zero and zero disparity error cases, which explains why the proposed algorithm can improve the coding performance by adjusting the depth distortions without alteration of the occlusion order.

The rest of this paper is organized as follows. We first give a review of the related work on depth coding in Section II, and investigate the allowable depth distortions that inherently exist for each potential depth error of each pixel in Section III. Next, in Section IV, with the aim of improving coding efficiency while keeping the occlusion order unchanged, we develop a rate-distortion optimal occlusion-aware depth map coding scheme based upon allowable depth distortions. Experimental results are discussed in Section V. Finally, Section VI concludes this paper.

II. Related Work

A. Depth Map Coding Using Its Characteristics

Depth map compression is important for compact texture plus depth representation of a 3D scene, and various approaches have been proposed recently in this area. By using both temporal and inter-view correlations between the previously encoded texture images, a multi-view depth image coding method is proposed in [13], where some blocks of the depth image are skipped without encoding for the purpose of reducing the encoder complexity. Similarly, to better remove the depth coding artifacts, Liu *et al.* [14] proposed a trilateral filter and a sparse dyadic coding mode for depth compression by properly referencing structural information from corresponding texture. To reduce the loss of boundary information, Kang *et al.* [15] proposed an adaptive geometry based intra prediction scheme, in which a set of partitioned intra prediction modes are defined as the alternatives of conventional intra prediction. For a better representation of edges in depth map, 3D-HEVC also includes new intra prediction modes, i.e., depth modelling modes (DMMs), where two types of partitions are defined, namely Wedgelets and Contours [16]. As mode decision between existing intra modes and DMMs results in a huge computational burden to the encoder, the authors of [17] proposed to use the most probable mode as the indicator to simplify the mode decision process. In order to further enhance the performance, an edge-based intramode selection scheme is proposed for depth coding [18]. In this work, the relationship between the Hadamard coefficients of adjacent depth levels is firstly investigated, and then, unnecessary DMMs are selectively omitted in the mode decision based on the edge classification results. Based on intra coding information from spatio-temporal, inter-component and inter-view neighbouring coding units, an intra-mode selection scheme is proposed to accelerate depth coding procedure [19]. In order to improve the existing algorithm in the presence of depth edges that cannot be efficiently predicted by the directional modes, a constrained depth modelling mode (CDMM) is proposed in [20] based on explicit edge representation. The principle idea of CDMM is to impose constraints on block partitioning and block partitioning slope, and combine with the flexible partitioning scheme to improve the flexibility to approximate depth edges. In [21], to promote sparsity in the transform domain for depth map coding gain, the authors proposed a unified graph based transform and transform domain sparsification optimization framework. By using surface-smooth prior of depth map, a multi-resolution approach is proposed for depth map compression using graph-based transform [22], [23]. The key idea of this approach is to encode the depth map in different resolutions: encode edges in original high resolution to preserve sharpness, and encode smooth surfaces in low-pass-filtered and down-sampled low resolution to save coding bits.

B. View Synthesis Oriented Depth Map Coding

As mentioned in Section I, there are many view synthesis-oriented depth map coding and optimization algorithms in the literature, along with a number of depth-distortion-related synthesis distortion models. In [4], Kim *et al.* proposed a linear model of the depth error to approximate the synthesized view distortion over the entire frame. Based on local video characteristics, an autoregressive model was developed in [5] to characterize the rendered view distortion at the block level, which is then used in mode selection for compression. By assuming a linear interpolation in rendering, the view synthesis distortion is approximated as the product of the depth error and the gradient value of the texture pixel [6], and in the case of occlusion, the gradient is updated by excluding the to-be-occluded neighbouring pixels. In [24], Yuan *et al.* derived a polynomial synthesis distortion model for bit allocation between texture and depth based on Taylor series expansion. Taking into account texture image characteristics, Fang *et al.* [25] proposed to estimate the rendering distortion using an approach that combines frequency domain analysis with power spectral density and spatial domain analysis with local gradient. Similarly, a regional linear model with different slopes was developed to estimate the geometry-error-induced synthesis distortions for different depth regions classified by the smoothness of the associated texture [7]. To analyze how the virtual view position affects the synthesis distortion, Velisavljevic *et al.* [8] developed a cubic synthesized view distortion model as a function of the view's location for multiview image compression. By further analyzing different possible overlapping/disocclusion scenarios and the effects due

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2019.2919198, IEEE Transactions on Image Processing

to different extents of depth error, Fang *et al.* [9] extended the cubic model to an analytic quintic model, which mathematically combines the quadratic/biquadratic models for view synthesis distortion and the linear models for the probability under different defined scenarios. Similar to [8], the linearity of the proportion of the occluded pixels with the virtual view distance is used in distortion derivation. In [26], Yang *et al.* analyzed the respective monotonicity properties of the view synthesis distortion related with texture error and depth error. [27] and [28] considered channel errors possibly occurred in the texture and depth, and modeled the statistical relationship between the end-to-end view synthesis distortion and expected depth distortion. In all the algorithms discussed above, the relationship between the view synthesis distortion and depth distortion is modeled as one-to-one mapping function, and [6], [8] and [9] consider the occlusion effect in distortion modeling mainly through excluding the distortion of the occluded pixels from the overall distortion.

To characterize the many-to-one relationship, Zhao *et al.* [10] pioneered the D-NOSE model, within which all the depth levels correspond to the same disparity level due to the disparity rounding operation. The D-NOSE model is used to smooth depth map, the feasibility of which is then illustrated by employing the smoothed depth map for compression. Based on the texture adaption, a low complexity view synthesis optimization algorithm is proposed in [29], in which the real re-rendering required by the direct computation of view synthesis distortion change is skipped if there is no change in the synthesized view with the distorted depth map. In [30], for the purpose of estimation of view synthesis distortion in the presence of transmission error, a bipartite probabilistic graphical model is proposed to represent the relationship between a reference depth map and its warped virtual depth map, where the probability mass function of the depth level of the to-be-synthesized pixel is derived. To reduce the complexity of the graphical model for view synthesis distortion estimation in [30], a depth-bin-based graphical model is developed in [31], where the set of all the depth values with the same rounded disparity is defined as a depth bin, and then the per-pixel depth value probability distribution is converted into depth bin probability distribution. However, in all these algorithms, only one kind of many-to-one mapping relationship, i.e., multiple possible depth distortions correspond to the zero disparity error in view synthesis, is considered. When performing joint optimization of filtering and encoding, these algorithms often become ineffective as the compression error usually leads to a non-zero disparity error.

By considering that there also exists an allowable range for the depth error that leads to one non-zero disparity error, the authors of [11] proposed a complete allowable depth distortion model, which is then employed in the procedures of rate-distortion optimized mode decision and motion estimation in multiview depth map coding. However, in this work, the interval of allowable depth level change for non-zero disparity error case is simply modeled as the zero-disparity-error allowable depth error range of the original depth level scaled by adding the associated depth error to its lower and upper bounds, which may be inaccurate. This is because that, this interval is heuristically obtained from the original depth level, which cannot guarantee that any depth level change within it generates zero disparity error with respect to the modified depth level. In other words, the depth errors in the defined interval may generate different non-zero disparity errors. Further, the occlusion that multiple pixels are warped to the same destination in view synthesis is not handled in [11]. In this paper, we derive the allowable depth error range in the non-zero disparity error case with both mathematical and geometrical proofs, and then incorporate the derived allowable depth errors into depth map coding optimization in occlusion scenario.

III. Allowable Depth Distortion Range

In this section, we firstly give a complete definition of allowable depth error range in view synthesis. Then, we derive the lower and upper bounds for the defined range depending on whether the associated depth errors generate geometrical errors. We also demonstrate the existence and unicity of the proposed allowable depth distortion range.

In the view synthesis, a depth level of a pixel may render the pixel to be mapped to a position in the warped view that is usually different from its location in the original view, which results in disparity. Further, distortion in depth level may render that the pixel is mapped to another different position in the warped view. Therefore, the depth distortion of a pixel may generate disparity error in the warping. However, in practice, several depth errors correspond to the same disparity error and thus depth distortion within a certain range will induce the same view synthesis distortion in the pixel mapping. In light of this, we first give a formal definition of allowable depth distortion range as follows.

Definition 1. *For a pixel i and its original depth level v_i, assume that ΔV_i is a depth level error range for the possible depth level change Δv_i of this pixel. If ΔV_i can guarantee that any two depth level changes in it lead to the same disparity error in view synthesis, we call it allowable depth error range for pixel i.*

With this definition, we describe how to derive the allowable depth error range for a given depth pixel. In what follows, we firstly review the zero-disparity-error interval of allowable depth distortion, i.e., the D-NOSE model in [10]. Then, we evolve to the derivation of the proposed allowable depth distortion model.

A. The D-NOSE Model

Assume that the physical depth value of i is z_i. The disparity between the two positions of pixel i in the reference view and warped view can be expressed as

$$d_i = \frac{fl}{z_i} \tag{1}$$

where f and l are the focal length of the camera and the baseline distance of the two views, respectively.

In 3-D video, a non-linear quantization function is usually employed to convert the continuous depth values to discrete

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2019.2919198, IEEE Transactions on Image Processing

depth levels, the expression of which is [32]

$$v_i = \left\lfloor 255 \cdot \frac{Z_{\text{near}}}{z_i} \cdot \frac{Z_{\text{far}} - z_i}{Z_{\text{far}} - Z_{\text{near}}} + 0.5 \right\rfloor \tag{2}$$

where Z_{near} and Z_{far} denote the nearest and farthest depth values of the scene. With (1) and (2), the disparity of i can be obtained from the associated depth level v_i

$$d_i = fl(C_1 \cdot v_i + C_2) \tag{3}$$

where $C_1 = 1/255(1/Z_{\text{near}} - 1/Z_{\text{far}})$ and $C_2 = 1/Z_{\text{far}}$. In the following, we use $D(v)$ to represent the general disparity function of depth level v for simplicity.

Generally, the initially mapped pixel position needs to be rounded for determining the exact warped pixel position in the synthetic picture. Without loss of generality, the disparity rounding function with $1/N$ sub-pixel sampling precision is assumed, i.e.,

$$R(D(v_i)) = \frac{\lceil (D(v_i) - o)N \rceil}{N} \tag{4}$$

where o represents the offset error and determines the decision level of the rounding process. The value of o is $0 < o \leq 1/N$. Assume now pixel i has a depth level change of Δv_i caused by compression, and thus the resulting depth level $v_i + \Delta v_i$ corresponds to another disparity for pixel i. Therefore, the disparity error Δd_i due to Δv_i can be written as

$$\Delta d_i = R(D(v_i + \Delta v_i)) - R(D(v_i)) \tag{5}$$

In (5), if $\lceil (D(v_i + \Delta v_i) - o)\, N \rceil = \lceil (D(v_i) - o)\, N \rceil$, then $\Delta d_i = 0$ holds. Due to limited number of disparity levels, there exists a range of depth level change that makes Δd_i equal to zero. Based on (4), the sufficient condition for depth level v_i of no disparity difference introduced in pixel mapping is that $v_i + \Delta v_i$ corresponds to the same rounded disparity as what v_i generates. In light of this, we have

$$-\left(\frac{1}{N} - o\right) \leq D(v_i + \Delta v_i) - R(D(v_i)) < o \tag{6}$$

After some substitutions and rearrangements for (6), we can get the allowable depth level variation range ΔV_i for depth level v_i for the case when the associated Δd_i is zero, which is shown at the bottom of this page. For simplicity, we use Δv_i^- and Δv_i^+ to represent the lower bound and upper bound of Δv_i. In this case, the zero-disparity-error interval of allowable depth level change can be defined as $\Delta v_i \in \Delta V_i = [\Delta v_i^-, \Delta v_i^+]$, and $\Delta v_i^- < 0$ and $\Delta v_i^+ > 0$. It should be emphasized that ΔV_i for zero disparity error case is originally observed and modeled in [10]. However, in this paper, we focus on modeling of ΔV_i for non-zero disparity error case in view rendering as to be shown, which is ignored in [10] and not fully and rigorously investigated in [11].

B. The Proposed Allowable Depth Distortion Model

When the associated Δd_i is not zero, there also exists an allowable depth error range. However, as $\Delta d_i \neq 0$, (6) does not hold, and thus the D-NOSE model is not applicable. In the following, we theoretically derive the allowable depth distortion interval in the non-zero disparity error case from the D-NOSE model. For ease of understanding, we also provide a geometrical illustration for the proposed allowable depth error interval.

1) Derivation of the Proposed Allowable Depth Distortion Model: The solution of the proposed allowable depth distortion model is detailed in the following lemma.

Lemma 1. *Assume that a particular depth level change of pixel i is Δv_i^k ($\Delta v_i^k \notin [\Delta v_i^-, \Delta v_i^+]$), which leads to a non-zero Δd_i in pixel mapping. Thus, the zero-disparity-error range of allowable depth level variation (i.e., $[(v_i + \Delta v_i^k)^-, (v_i + \Delta v_i^k)^+]$) for depth level $v_i + \Delta v_i^k$ provides the range of allowable depth error around Δv_i^k with respect to v_i, which can be represented as $\Delta v_i \in [\Delta v_i^k + (v_i + \Delta v_i^k)^-, \Delta v_i^k + (v_i + \Delta v_i^k)^+]$.*

Proof: In the non-zero disparity error case of v_i, as assumed, the particular depth error Δv_i^k results in a disparity error Δd_i. Based on (5), Δd_i can be calculated as $\Delta d_i = R(D(v_i + \Delta v_i^k)) - R(D(v_i))$. With the definition 1, the allowable depth error range for v_i is the one within which any depth level change Δv_i generates the same Δd_i.

When v_i is changed to $v_i + \Delta v_i^k$, the allowable depth error range $[(v_i + \Delta v_i^k)^-, (v_i + \Delta v_i^k)^+]$ for depth level $v_i + \Delta v_i^k$ for the zero-disparity-error case can be calculated using the above derivation. In this interval, any depth level change selected will not induce disparity difference with the disparity induced by $v_i + \Delta v_i^k$, i.e., $\forall \Delta\tilde{v}_i \in [(v_i + \Delta v_i^k)^-, (v_i + \Delta v_i^k)^+] : R(D(v_i + \Delta v_i^k + \Delta\tilde{v}_i)) = R(D(v_i + \Delta v_i^k))$, where $\Delta\tilde{v}_i$ here represents the possible depth level change with respect to $v_i + \Delta v_i^k$. Thus, we have $\forall \Delta\tilde{v}_i \in [(v_i + \Delta v_i^k)^-, (v_i + \Delta v_i^k)^+] : R(D(v_i + \Delta v_i^k + \Delta\tilde{v}_i)) - R(D(v_i)) = \Delta d_i$. In other words, when $\Delta\tilde{v}_i \in [(v_i + \Delta v_i^k)^-, (v_i + \Delta v_i^k)^+]$, the depth level change Δv_i with respect to v_i that is equal to $\Delta v_i^k + \Delta\tilde{v}_i$ generates the identical disparity error Δd_i. Therefore, $[\Delta v_i^k + (v_i + \Delta v_i^k)^-, \Delta v_i^k + (v_i + \Delta v_i^k)^+]$ is indeed the allowable depth error range around Δv_i^k with respect to v_i when Δv_i^k introduces a non-zero disparity error in view synthesis. ■

2) Geometric Illustration of the Proposed Allowable Depth Error Interval: In order to better understand the derivation of Lemma 1, we give a geometry illustration of the intervals of allowable depth distortion in zero and non-zero disparity error cases in view synthesis in Fig. 1, where, for convenience, depth level v_i corresponds to disparity level d_i, and depth level $v_i + \Delta v_i^k$ corresponds to another disparity level d_i'. In other words, the depth error Δv_i^k generates a non-zero disparity error represented by Δd_i. In Fig. 1, the interval for which the depth levels generate the same disparity with v_i is represented as $[v_i + \Delta v_i^-, v_i + \Delta v_i^+]$, and the interval for which

$$\left\lceil \frac{R(D(v_i)) - 1/N + o - flC_2}{flC_1} \right\rceil - v_i \leq \Delta v_i \leq \left\lfloor \frac{R(D(v_i)) + o - flC_2}{flC_1} \right\rfloor - v_i \tag{7}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2019.2919198, IEEE Transactions on Image Processing

the depth level generate the same disparity with $v_i + \Delta v_i^k$ is represented as $[a, b]$, which means that all the depth levels in $[a, b]$ generate the disparity error Δd_i with respect to v_i. $[a, b]$ is thus the allowable depth error range to be found in the non-zero disparity error case for depth level v_i with the depth error Δv_i^k. Similar to $[\Delta v_i^-, \Delta v_i^+]$ defined for v_i, denoted by $[(v_i+\Delta v_i^k)^-, (v_i+\Delta v_i^k)^+]$ the depth error interval for $v_i+\Delta v_i^k$ in depth no-synthesis-error case, it is thus easy to derive $[a, b]$ as $[v_i + \Delta v_i^k + (v_i + \Delta v_i^k)^-, v_i + \Delta v_i^k + (v_i + \Delta v_i^k)^+]$.

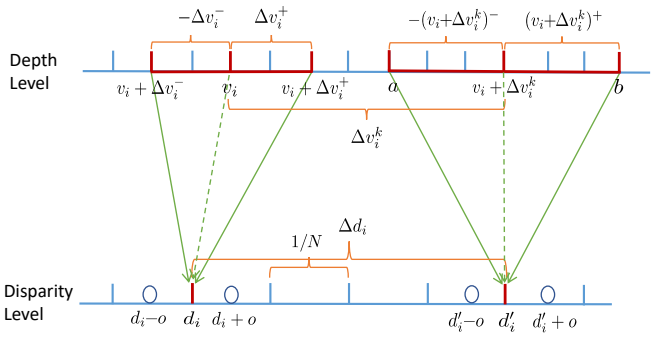


Fig. 1. Illustration of intervals of allowable depth distortion in zero and non-zero disparity error cases. In the disparity level line, the major tick mark represents $1/N$ accuracy sub-pixel sampling position, and the circle represents the rounding bound with an offset error o. The depth level v_i corresponds to disparity d_i, and the depth level $v_i + \Delta v_i^k$ generates another disparity d_i'. The depth no-synthesis error model for v_i is denoted by $[\Delta v_i^-, \Delta v_i^+]$, while the allowable depth error range for $v_i + \Delta v_i^k$ in zero disparity error case is shown with $[(v_i + \Delta v_i^k)^-, (v_i + \Delta v_i^k)^+]$. It should be noted here that $[(v_i + \Delta v_i^k)^-, (v_i + \Delta v_i^k)^+] \neq [\Delta v_i^-, \Delta v_i^+]$.

With the proof of Lemma 1, when Δv_i^k introduces a zero disparity error, i.e., $\Delta d_i = 0$, we can still have allowable depth error range equal to $\Delta v_i \in [\Delta v_i^k + (v_i + \Delta v_i^k)^-, \Delta v_i^k + (v_i + \Delta v_i^k)^+]$ for v_i, which means $[\Delta v_i^k + (v_i + \Delta v_i^k)^-, \Delta v_i^k + (v_i + \Delta v_i^k)^+]$ is a *generic form* of interval of allowable depth error for v_i with depth error Δv_i^k. In addition, in the case of $\Delta d_i = 0$, $\Delta v_i^k + (v_i + \Delta v_i^k)^- = \Delta v_i^-$ and $\Delta v_i^k + (v_i + \Delta v_i^k)^+ = \Delta v_i^+$ hold, which indicates that the depth no-synthesis error range for quantized depth level $v_i + \Delta v_i^k$ can be calculated using the corresponding range for initial depth level v_i in the form of $[\Delta v_i^- - \Delta v_i^k, \Delta v_i^+ - \Delta v_i^k]$.

With the Lemma 1 and (7), we can calculate the allowable depth error range for each possible depth level change of each depth pixel in view synthesis. It is also clear that $\Delta V_i = [\Delta v_i^k + (v_i + \Delta v_i^k)^-, \Delta v_i^k + (v_i + \Delta v_i^k)^+]$ is the one and only allowable depth error range around Δv_i^k with respect to v_i in both the zero and non-zero disparity error cases. After the derivation of the allowable depth error range, we start to formulate the problem of optimally selecting the depth level change from the allowable depth error range for each pixel such that the overall distortion involved in the occlusion process is minimized.

IV. Occlusion-Aware Depth Map Coding Optimization

In this section, we incorporate the allowable depth distortions into occlusion-aware depth map coding optimization. Firstly, we formulate an optimization problem of how to properly selects the depth distortion from the derived allowable range for each pixel in occlusion in a rate-distortion sense. Then, we propose a feasible dynamic programming scheme to find the optimal solution. Finally, the implementation detail and complexity are given.

A. Problem Formulation

When an occlusion occurs after view synthesis using the reconstructed texture and depth, there are multiple texture pixels in the reference view that are mapped to a same position in the virtual view, and the texture pixel with the largest quantized depth level is chosen as the final synthesized pixel. Let N be the total number of pixels involved in this occlusion. Assume now that the pixel j is the final winning pixel, which means that the depth levels of all the previous pixels are smaller than the depth level of the winning pixel, i.e., $v_i+\Delta v_i < v_j+\Delta v_j : i \in [1, j-1]$, and the depth levels of all the subsequent pixels are smaller than or equal to that of the winning pixel, i.e., $v_i+\Delta v_i \leq v_j+\Delta v_j : i \in [j+1, N]$. Recall that Δv_j (or Δv_i) represents the reconstructed depth error, which, for performance optimization, can be adjusted based on allowable depth errors derived in Section III. Let $P(\Delta v_j)$ be the probability of the depth pixel j choosing the possible depth change Δv_j from allowable depth error range ΔV_j, and $\bar{d}_j(\Delta v_j)$ be the associated view synthesis distortion caused by the particular depth error Δv_j. $P(\Delta v_i)$ with subscript $v_i + \Delta v_i < v_j + \Delta v_j$ represents the probability of the depth pixel i choosing the depth error Δv_i from its allowable depth distortion model, with the condition that Δv_i must satisfy $v_i + \Delta v_i < v_j + \Delta v_j$, i.e., the changed depth level for pixel i is smaller than the changed depth level for the winning pixel j. Therefore, the final view synthesis distortion induced by choosing Δv_j for pixel j can be represented as

$$\begin{aligned}\bar{D}(\Delta v_j) &= P(\Delta v_j) \times \left(\prod_{i=1}^{j-1} \underset{v_i+\Delta v_i < v_j+\Delta v_j}{P} (\Delta v_i) \right) \\ &\quad \times \left(\prod_{i=j+1}^{N} \underset{v_i+\Delta v_i \leq v_j+\Delta v_j}{P} (\Delta v_i) \right) \times \bar{d}_j(\Delta v_j)\end{aligned} \tag{8}$$

In (8), the final view synthesis distortion for the winning pixel j is measured by multiplying the Δv_j-induced $\bar{d}_j(\Delta v_j)$ with its probability for choosing Δv_j and the depth error selection probability for each other depth pixel that is involved in occlusion. $\bar{d}_j(\Delta v_j)$ can be estimated from the rendering position error induced by Δv_i using a mathematical model as done in [11], or directly measured by using the re-rendering-based algorithm in [33]. In this work, for simplicity and complexity consideration, we choose the method proposed in [11] for distortion estimation. That is, we firstly calculate the rendering position error from the depth error Δv_i using (5), and then, the view synthesis error is estimated from the rendering position error using a linear model. In addition, as can be observed from (8), the selection of the depth change of other pixels could affect the choice of that of the current pixel, and thus the depth error selection of the current pixel should be jointly considered with other pixels. In the following, considering the correlation between the pixels

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2019.2919198, IEEE Transactions on Image Processing

involved in the warping competition process, we formulate a view synthesis distortion criterion for all the pixels. For ease of derivation, we rearrange the pixels in a depth-level-monotonic-increasing order, with the winning pixel being the last one. Let $\mathbf{\Delta v} = [\Delta v_1, \Delta v_2, \cdots, \Delta v_N]$ be a vector representing the depth level changes assigned to the N pixels. The total view synthesis cost caused by the selection of the depth level change within the allowable depth error range for all the pixels under a given total bit rate R_c can be formulated as

$$\begin{aligned} J(\mathbf{\Delta v}) = & \sum_{z=1}^{N} \left\{ \left(\prod_{j=1}^{z} P(\Delta v_j) \right) \times \bar{d}_z(\Delta v_z) \right\} \\ & + \lambda \sum_{z=1}^{N} R_z(\Delta v_z) \\ \text{s.t.} \quad & P(\Delta v_j) : v_{j-1} + \Delta v_{j-1} \leq v_j + \Delta v_j, \ \forall j = 2, \ldots, N. \end{aligned} \tag{9}$$

where $R_z(\Delta v_z)$ denotes the bit rate for pixel z when the pixel value v_z is changed to $v_z + \Delta v_z$, which can be obtained by entropy encoding this pixel with distortion Δv_z. λ is called the Lagrangian multiplier. In (9), we sum the view synthesis distortions of all the depth pixels in the occlusion, and the distortion of each pixel is measured by multiplying the depth error-induced synthesis distortion $\bar{d}_z(\Delta v_z)$ with its depth error selection probability and the depth error selection probabilities for all the preceding pixels. Then, the depth coding optimization using allowable depth distortions for the occlusion scenario is to find $(\mathbf{\Delta v})^*$ which minimizes $J(\mathbf{\Delta v})$, i.e.,

$$(\mathbf{\Delta v})^* = \arg\min_{\mathbf{\Delta v}} J(\mathbf{\Delta v}) \tag{10}$$

If $(\mathbf{\Delta v})^*$ leads to that $\sum_{z=1}^{N} R_z(\Delta v_z)$ happens to R_c, then $(\mathbf{\Delta v})^*$ is also an optimal solution to the constrained problem of minimizing the total view synthesis distortion subject to a given maximum bit rate of R_c. It is well-known that when λ sweeps from zero to infinity, the solution to (10) traces out the convex hull of the rate distortion curve, which is a nonincreasing function. Hence, the bisection method in [34] can be used to find the optimal λ.

Therefore, the task at hand is how to find the optimal solution to the problem expressed in (10). Since (10) is evaluated for any possible combination of the allowable depth level changes of all the pixels, one possible way to accomplish this task is by exhaustive search. By enumerating all possible $\mathbf{\Delta v}$ and substituting into (10), an optimal vector $(\mathbf{\Delta v})^*$ can be determined. That being said, suppose there are V depth level changes on average available for each competing pixel, the resulting search space will be V^N, which, in general, is computationally intense. Especially, for some video sequences that have a large number of available depth level variations, the complexity of the exhaust search is prohibitive.

B. Dynamic Programming Solution

In this section, a dynamic programming optimization scheme is proposed, which aims to reduce the computational complexity in solving (10). The basic principle is to break down (9) into simpler sub-problems in a recursive manner, and then recursively find the optimal solutions to the sub-problems. It consists of creating a trellis in terms of states and stages to represent all the viable allowable depth distortions in an occlusion. Each path in the trellis has an associated cost, and the shortest path is defined as the optimal solution. To proceed the development of the dynamic programming solution, the following lemma is needed.

Lemma 2. *For an arbitrary pixel i in the occlusion process, suppose its depth value v_i has a change of Δv_i^k after compression, and Δv_i^k will lead to a non-zero disparity error in the view synthesis. Denote by $[(v_i + \Delta v_i^k)^-, (v_i + \Delta v_i^k)^+]$ the allowable depth level range for $v_i + \Delta v_i^k$ that will not induce any disparity difference in the view synthesis. If the depth level change Δv_i of each pixel is selected from the corresponding set $[\Delta v_i^k + (v_i + \Delta v_i^k)^-, \Delta v_i^k + (v_i + \Delta v_i^k)^+]$, then the occlusion order in which all the depth values $v_i + \Delta v_i^k$ proceed will not be influenced.*

Proof: We assume that two different pixels i and j are involved in this warping competition, and the depth values of them are v_i and v_j, respectively. After compression, v_i changes to v_i' with the depth error being $\Delta v_i^k = v_i' - v_i$. The Δv_i^k then leads to a non-zero disparity difference $\Delta d_i = R(D(v_i')) - R(D(v_i))$. Similarly, the depth error Δv_j^k of pixel v_j can also induce a disparity error Δd_j during view synthesis, where Δd_j could be zero or non-zero. Finally, the compressed depth values v_i' and v_j' render these two pixels mapping to the same position in the virtual view. We assume that pixel i occludes j, which means $v_i' > v_j'$.

If the depth level change Δv_i is selected from the set $\Delta V_i = [\Delta v_i^k + (v_i + \Delta v_i^k)^-, \Delta v_i^k + (v_i + \Delta v_i^k)^+]$, which makes v_i changing to v_i'', as demonstrated in Section III, $R(D(v_i'')) = R(D(v_i'))$ holds. Likewise, any varied depth level $v_j'' \in [v_j + \Delta v_j^k + (\Delta v_j^k + v_j)^-, v_j + \Delta v_j^k + (\Delta v_j^k + v_j)^+]$ satisfies $R(D(v_j'')) = R(D(v_j'))$ (or, if $v_j' \in [v_j + \Delta v_j^-, v_j + \Delta v_j^+]$, any varied depth level $v_j'' \in [v_j + \Delta v_j^-, v_j + \Delta v_j^+]$ satisfies $R(D(v_j'')) = R(D(v_j')) = R(D(v_j))$). Since the disparity function is strictly increasing and $v_i' > v_j'$, $D(v_i') > D(v_j')$ holds, and thus $R(D(v_i')) \geq R(D(v_j'))$ due to the non-decreasing properties of the rounding function. On the other hand, as i and j are two different pixels and they are mapped to the same position generating occlusion, we have $R(D(v_i')) \neq R(D(v_j'))$. Therefore, we can only have $R(D(v_i')) > R(D(v_j'))$. Since $R(D(v_i')) > R(D(v_j'))$, we have $R(D(v_i'')) > R(D(v_j''))$, and thus $v_i'' > v_j''$, which thus preserves the occlusion order of $v_i' > v_j'$.

■

A corollary of this lemma is that, when, for a pixel i, the compression error Δv_i^k leads to a zero disparity error during view synthesis, the depth level change Δv_i being selected from the set $[\Delta v_i^-, \Delta v_i^+]$ will not alter the occlusion order in which the depth value $v_i + \Delta v_i^k$ (or v_i) proceeds. It is worth pointing out that the occlusion order invariance property allows us to optimize depth map coding using allowable depth errors in a rate-distortion optimal manner.

Based on Lemma 2, the constraint of monotonic increase of depth levels for $P(\Delta v_j)$ can be dropped for convenience,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2019.2919198, IEEE Transactions on Image Processing

which leads to (9) as

$$J(\mathbf{\Delta v}) = \sum_{z=1}^{N} \left\{ \left(\prod_{j=1}^{z} P(\Delta v_z) \right) \times \bar{d}_z(\Delta v_z) + \lambda R_z(\Delta v_z) \right\} \tag{11}$$

Based on (11), for the purpose of establishing forward dynamic programming, we define the cost function at state Δv_k of stage k as follows

$$g_k(\Delta v_k) = P(\Delta v_k)\bar{d}_k(\Delta v_k) + \lambda R_k(\Delta v_k) \tag{12}$$

where stage k corresponds to the kth depth pixel in the occlusion ($1 \leq k \leq N$), and each state Δv_k at stage k represents a possible depth error used in the k pixel, which is selected from the set of available allowable depth level variations of stage k, i.e., ΔV_k.

Then, when $k = 1$, the cost-to-go function can be written as

$$J_1(\Delta v_1) = g_1(\Delta v_1) \tag{13}$$

and when, $2 \leq k \leq N$, the cost-to-go function at stage k, state Δv_k is shown in (14) at the bottom of this page, where

$$\Delta v_j^* = \begin{cases} \underset{\Delta v_{k-2} \in \Delta V_{k-2}}{\arg\min} \; J_{k-1}(\Delta v_{k-1}), & j = k-2 \\ \underset{\Delta v_j \in \Delta V_j}{\arg\min} \; J_{j+1}(\Delta v_{j+1}^*), & 1 \leq j \leq k-3 \end{cases} \tag{15}$$

For a given λ, the proposed dynamic programming solution is described as follows. It proceeds from the first pixel to the winning pixel in the occlusion. For each state of stage k, it evaluates all the paths that lead to the state from any admissible state in the previous stage $k-1$, and store the one that produces the minimum Lagrangian cost. At the last stage, the minimal cost associated with the state becomes the optimal cost, and the path that leads to this state from the first stage given by (15) determines the optimal pixel level change vector $\mathbf{\Delta v}$ for the occlusion-inducing depth pixels. Assuming that each depth pixel takes an average of V allowable depth level changes. Since the proposed dynamic programming solution considers the dependency only from the immediately preceding stage for the current stage, the time complexity is $O(V^2 \times N)$, which is significantly smaller than that of the exhaustive search, i.e., $O(V^N)$. The proposed occlusion-aware depth map coding optimization method is embedded into view synthesis optimization (VSO) in HTM, and the implementation details are summarized in Algorithm 1.

Algorithm 1 Proposed occlusion-aware depth map coding optimization

Input: The reconstructed depth error vector $[\Delta v_1, \Delta v_2, \cdots, \Delta v_N]$
Determination of occlusion-inducing depth pixels
Calculate the allowable depth error range ΔV_z for each pixel z, $z \in [1, N]$
for ($z \in [1, N]$ and $\Delta v_z \in \Delta V_z$) **do**
    Estimate the synthesis distortion $\bar{d}_z(\Delta v_z)$ for a Δv_z
    Estimate the QP_z for the Δv_z using the linear model in [7]
    Entropy encode the pixel z with QP_z, and obtain the bit rate $R_z(\Delta v_z)$
end for
Formulate the total view synthesis cost $J(\mathbf{\Delta v})$ for occlusion via (9) and (10)
Determination of the λ in (9) by the bisection algorithm
procedure DYNAMIC PROGRAMMING SOLUTION
    Initialization:
    For ($k = 1$ and $\Delta v_1 \in \Delta V_1$), calculate the $J_1(\Delta v_1)$ via (13)
    Recursion:
    for ($k = 2, \cdots, N$ and $\Delta v_k \in \Delta V_k$) **do**
        Define a pointer $i(\Delta v_k)$ to track the optimal path
        Perform $i(\Delta v_k) = \arg\min J_k(\Delta v_k)$ using (14)
    end for
    Termination:
    Perform $i(\Delta v_{N+1}) = \arg\min J_{N+1}(\Delta v_{N+1})$ via (14) with $g_{N+1}(\Delta v_{N+1}) = 0$
    Backtracking:
    $k = N$
    $\Delta v_k^* = i(\Delta v_{k+1})$
    while $(k \geq 1)$ **do**
        $\Delta v_{k-1}^* = i(\Delta v_k^*)$
        $k = k - 1$
    end while
end procedure
Output: The optimal depth error vector $[\Delta v_1^*, \Delta v_2^*, \cdots, \Delta v_N^*]$ and the associated $[QP_1^*, QP_2^*, \cdots, QP_N^*]$

V. EXPERIMENTAL RESULTS

We implement the proposed depth map optimization algorithm on the 3D-HEVC reference software HTM-16.0 [35], and the VSRS-1D-Fast view synthesis software included in HTM 16.0 is used to render the intermediate virtual views. The standard multi-view 3-D video sequences "BookArrival" [36], "Ballet" and "Breakdancers" [37], "Lovebird1" [38], "Newspaper" [39], "GT_Fly" and "Undo_Dancer" [40] are chosen for our simulations. Note that the first five sequences are camera-captured multiview sequences with imperfect depth map values, while the rest sequences are computer-generated multiview sequences with ground truth depth maps. For each sequence, each view is encoded with a group of pictures (GOP) size of 8 frames, and the intra period is 24. In 3-D video coding, the encoder uses variable block-size motion and disparity estimation, with a search range of 64 pels. VSO and "UseEstimatedVSD" are enabled. A variety

$$J_k(\Delta v_k) = \min_{\Delta v_{k-1} \in \Delta V_{k-1}} \left\{ J_{k-1}(\Delta v_{k-1}) + \left(\prod_{j=1}^{k-2} P(\Delta v_j^*) \right) P(\Delta v_{k-1})[g_k(\Delta v_k) - \lambda R_k(\Delta v_k)] + \lambda R_k(\Delta v_k) \right\} \tag{14}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2019.2919198, IEEE Transactions on Image Processing

of QP combinations for texture and depth are considered: (15;24), (20;29), (25;34), (30;39), (35;42), and (45;48). In view rendering, the virtual views are generated with half-pel precision and symmetric rounding. The final warped pixel is blended from all warped images using a linear weighting function. The "BlendUseDistWeight" is enabled, which means weight blending depending on the view distance. The specific descriptions of the used MVD test sequence properties are listed in Table I. It should be noted that the experimental setup is configured in accordance with the Common Test Condition of the Joint Collaborative Team for 3DV [41]. The depth error in compression is assumed to follow the Gaussian distribution.

TABLE I

Specific description of MVD test sequences and some important simulation parameters.

Test sequence	Input views	Synthesized view	Frame rate	Resolution	Spacing (cm)
Lovebird1	6-8	"7"	16.7	1024 × 768	3.5
BookArrival	8-10	"9"	30	1024 × 768	6.5
Newspaper	4-6	"5"	30	1024 × 768	5
Breakdancers	1-5	"3"	30	1024 × 768	20
Ballet	4-8	"6"	30	1024 × 768	20
GT_Fly	5-9	"6"	25	1920 × 1088	Synthetic
Undo_Dancer	3-6	"4"	25	1920 × 1088	Synthetic

A. Evaluation of the Proposed Allowable Depth Distortion Model

In order to confirm the validity of the proposed allowable depth distortion model, in this section, we compare the percentages of the number of the correctly estimated allowable depth distortions with the proposed allowable depth distortion model (referred to as ADD_P), the allowable depth distortion model developed in [11] (referred to as ADD), and the D_NOSE model proposed in [10]. In the experiment, we firstly compress the video sequence using different QP pairs to get various depth errors for each depth pixel, and then, for each certain compression depth error, we estimate the allowable depth distortions using these three models. For the ground-truth allowable depth distortions, we firstly change the original depth level of each pixel to all other 255 possible depth levels, and then calculate the disparity error caused by each depth level change. If the disparity error is equal to that induced by the compression depth error[2], we consider that the corresponding depth level change is a ground-truth allowable depth distortion. With the ground-truth allowable depth distortions for each depth error, we can thus calculate the percentages of the number of correctly estimated allowable depth distortions by these three models. It should be noted that the percentage for each model is obtained by counting the estimated allowable depth errors for all the compression depth errors for all the pixels in a image.

Fig. 2 illustrates the estimation accuracy of the proposed model at various rendering precisions, i.e., integer-pel, half-pel, quarter-pel, corresponding to $N = 1, 2, 4$ in the rounding function, respectively. In this figure, the blue bars represent the percentages of the correctly estimated allowable depth distortions for zero-disparity error case in the overall ground-truth allowable depth distortions, while the yellow bars denote the percentages of the correctly estimated allowable depth distortions for non-zero-disparity error case. Each test model corresponds to a stacked bar. As can be observed, the ADD_P can correctly estimate all the allowable depth errors at various rendering precisions, which includes the allowable depth errors for the zero disparity-error inducing depth pixels and the non-zero disparity error inducing depth pixels. As for the depth pixels that induce zero disparity-errors in rendering, D-NOSE and ADD have the same amount of the percentage of the number of correctly estimated allowable depth distortions as that of ADD_P. However, among the depth pixels inducing non-zero disparity errors, D-NOSE is not applicable as it only models the depth-no-synthesis error range for 3-D warping. ADD can correctly estimate only around 28% $\sim$ 40% allowable depth distortions, which is significantly less than that of ADD_P. The reason can be explained as follows. As shown in ADD in [11], for a pixel i with depth level of v_i, when the compressed depth error Δv_i^k generates a non-zero disparity error, the interval of allowable depth error around Δv_i^k is modeled as $[\Delta v_i^k + \Delta v_i^-, \Delta v_i^k + \Delta v_i^+]$, where $[\Delta v_i^-, \Delta v_i^+]$ is the depth no-synthesis-error range for initial depth level v_i. Based on the theoretical proof of Lemma 1 for allowable depth error range, we know that the allowable depth error range around Δv_i^k should be $[\Delta v_i^k + (v_i + \Delta v_i^k)^-, \Delta v_i^k + (v_i + \Delta v_i^k)^+]$. Therefore, ADD can only correctly estimate the allowable depth distortions that are in the overlapping range of $[\Delta v_i^k + \Delta v_i^-, \Delta v_i^k + \Delta v_i^+]$ and $[\Delta v_i^k + (v_i + \Delta v_i^k)^-, \Delta v_i^k + (v_i + \Delta v_i^k)^+]$.

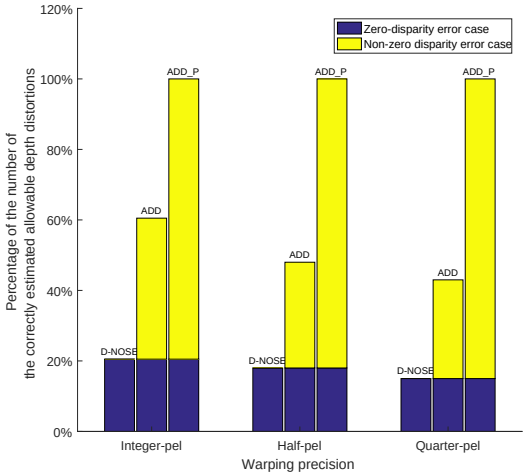


Fig. 2. Percentage of the number of correctly estimated allowable depth distortions by the D_NOSE, ADD, and ADD_P models at various warping precisions.

B. Performance of the Proposed Occlusion-Aware Depth Map Coding Optimization

For performance evaluation, Bjøntegaard Delta PSNR (BDPSNR) [42] and Bjøntegaard Delta Bit Rate (BDBR) [42]

[2]The disparity error caused by a certain depth compression error can be zero or non-zero.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2019.2919198, IEEE Transactions on Image Processing

are used for objective video quality assessment, which are measured by the total bit rate of the texture and depth along with the average peak signal-to-noise ratio (PSNR) of the synthesized views. The PSNR of intermediate view is computed between the virtual view images synthesized by the uncompressed texture and depth images and the decoded texture and depth images. In the comparison, the original 3D-HTM without allowable depth distortion consideration is used as the anchor. For reference baselines, as depth map coding scheme in [10] is a lossless solution and [11] addresses the mode decision/motion estimation in depth coding, for fair comparison, we choose the D-NOSE model and ADD model as the competing models, and incorporate them into a traditional depth coding optimization scheme. The traditional depth coding optimization method here means that each depth pixel is optimized with the allowable depth distortions without consideration of the effect of the occlusion in view rendering, i.e., selecting the allowable depth distortion from the defined allowable depth distortion model to minimize the associated Lagrangian cost as done in generic rate-distortion optimization in video coding. More specifically, in D-NOSE based depth coding (D-NOSE_DC), we use the D-NOSE model only for optimization of the pixels that induce zero disparity errors, while the the non-zero disparity error inducing depth pixels are not optimized due to the inapplicability of the D-NOSE model. In ADD based depth coding (ADD_DC), we optimize all the depth pixels using the ADD model. For the proposed occlusion-aware depth map coding optimization in this work, we divide it into two schemes for the convenience of evaluation of the performance step-by-step. The first scheme is the proposed ADD_P model combined with the traditional depth coding optimization method, similar to the algorithmic formulation of D-NOSE_DC and ADD_DC, which is referred to as ADD_P_DC. The other one is the proposed ADD_P model combined with occlusion-aware depth coding optimization, denoted by ADD_P_ODC. It should be noted that, in ADD_P_ODC, the depth pixels that are not involved in occlusion are optimized in an independent manner, i.e., each pixel selects the allowable depth distortion within its own range to minimize the view synthesis cost. In addition, the depth distortion adjustment can be implemented by slightly modifying the initial QP values.

The results of comparison of BDBR and BDPSNR for various test methods with different QP pairs of texture and depth map are tabulated in Table II. As can be observed, when the QP pair (25;34) is used, ADD_P_DC achieves an average BDBR gain of 14.2%, which is higher than BDBR gain of 11.8% brought by ADD_DC and 6.8% bit rate reduction of D-NOSE_DC. In terms of BDPSNR, ADD_P_DC achieves 0.94 dB PSNR gain on average, which is also the highest among all the comparative approaches that use traditional depth map coding optimization method. With the increase of QPs for texture and depth, it is noted that the performance gain of ADD_P_DC also increases, while the average coding gains of both D-NOSE_DC and ADD_DC tend to be reduced. For some specific sequences, D-NOSE_DC is even inferior to the anchor. This phenomenon is expected. As mentioned earlier, in D-NOSE_DC, the D-NOSE model only considers the allowable depth distortions for zero-disparity-error case, which renders that D-NOSE_DC cannot optimize the depth pixels that induce non-zero disparity errors in view synthesis. As QP gets larger, more depth pixels will generate the non-zero disparity errors due to larger compression errors, for which the performance of D-NOSE_DC is thus restricted. Although ADD_DC accounts for allowable depth distortions for both zero disparity error and non-zero disparity error cases, the associated allowable depth distortion range is heuristically determined. As illustrated in the subsection V-A, the ADD in [11] can only correctly estimate a limited portion of allowable depth distortions for the depth pixels resulting in non-zero disparity errors, which thus adversely affects the optimization performance, especially in the case of having a larger number of non-zero-disparity-error inducing depth pixels.

When comparing the coding results of ADD_P_DC and ADD_P_ODC, we can see that additional performance gains are obtained for all the test QP pairs and video sequences. For instance, at the QP pair of (25;34), with respect to ADD_P_DC, ADD_P_ODC yields 3.8% more average bit rate reduction, or further improves the PSNR gain by an average of 0.38 dB. This clearly demonstrates the advantage of considering the pixel dependences caused by occlusion in depth map coding optimization.

C. Evaluation of the Effect of the Occlusion on the Depth Coding Optimization

To further examine the effect of the occlusion on the optimization performance, we test the above algorithms by using different proportions of occluded pixels in the synthesized view rendered by compressed texture and depth map. As has been demonstrated in [8] and [9], the number of occluded pixels in the synthesized view is linearly proportional to the distance between the reference view and synthesized view. Inspired by this, we manually vary the distance of the virtual view with respect to the reference view to generate different proportions of occluded pixels. Specifically, in this test, those virtual views at the same positions as those in Table I are synthesized, but the reference views with larger distance relative to the virtual view are selected as input views for each sequence. The specific selected reference views and the related results are shown in Table III. As can be observed, when the relative distance increases, the performances of D-NOSE_DC and ADD_DC are significantly degraded, and the performance gaps between the reference algorithms and ADD_P_ODC become larger. This is mostly due to the fact that, both D-NOSE_DC and ADD_DC do not optimize the pixels that are involved in occlusion considering the influence of warping competition on the synthesis distortion, which, in fact, can contribute to the bit rate required to be transmitted and overall view synthesis distortion. In contrast, ADD_P_ODC minimizes the overall view synthesis distortion of occlusion by optimally allocating the allowable depth distortions (or bits) between the associated pixels. We also notice that the performance of ADD_P_ODC almost remains unchanged or even degrades with the increase of portion of occluded pixels. This result is somewhat surprising as ADD_P_ODC should

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2019.2919198, IEEE Transactions on Image Processing

TABLE II.
Comparison of BDBR and BDPSNR for various test methods under different QP pairs of texture and depth map. Negative values of BDBR indicate the bit rat savings compared to the anchor, while negative values of BDPSNR represent PSNR loss compared to the anchor.

QP pair	Test sequence	BDBR (%)				BDPSNR (dB)			
		D-NOSE_DC	ADD_DC	ADD_P_DC	ADD_P_ODC	D-NOSE_DC	ADD_DC	ADD_P_DC	ADD_P_ODC
(15;24)	Lovebird1	–7.6	–11.6	–12.0	–13.8	0.48	0.67	0.69	0.81
	BookArrival	–7.9	–13.7	–13.9	–15.6	0.54	0.79	0.84	1.07
	Newspaper	–8.8	–12.6	–13.8	–16.4	0.56	0.91	0.92	0.96
	GT_Fly	–9.0	–13.7	–14.5	–16.6	0.67	0.96	0.98	1.06
	Undo_Dancer	–9.7	–13.8	–14.7	–16.5	0.65	1.02	1.06	1.12
	Ballet	–9.1	–12.1	–12.9	–15.3	0.61	0.90	0.91	1.09
	BreakDancers	–9.5	–12.6	–12.9	–15.9	0.62	0.90	0.96	1.07
	Average	**–8.8**	**–12.8**	**–13.5**	**–15.7**	**0.59**	**0.88**	**0.91**	**1.03**
(20;29)	Lovebird1	–6.4	–11.4	–12.1	–14.5	0.45	0.65	0.70	0.91
	BookArrival	–7.1	–13.6	–14.0	–16.7	0.48	0.76	0.86	1.21
	Newspaper	–8.0	–12.1	–14.1	–17.3	0.49	0.90	0.93	1.03
	GT_Fly	–8.1	–13.2	–14.8	–17.5	0.52	0.94	0.99	1.20
	Undo_Dancer	–8.4	–12.1	–15.1	–17.2	0.50	1.00	1.09	1.24
	Ballet	–7.2	–11.9	–13.0	–16.6	0.42	0.89	0.92	1.21
	Breakdancers	–8.1	–12.4	–13.1	–17.1	0.43	0.86	0.97	1.19
	Average	**–7.6**	**–12.3**	**–13.7**	**–16.7**	**0.47**	**0.86**	**0.92**	**1.14**
(25;34)	Lovebird1	–5.3	–10.5	–12.4	–15.6	0.34	0.63	0.72	0.98
	BookArrival	–6.4	–13.2	–14.3	–17.8	0.35	0.71	0.89	1.28
	Newspaper	–6.8	–11.4	–14.6	–18.9	0.38	0.88	0.94	1.12
	GT_Fly	–7.6	–12.6	–15.3	–19.5	0.41	0.91	1.01	1.25
	Undo_Dancer	–7.4	–11.8	–15.6	–18.3	0.39	0.96	1.12	1.36
	Ballet	–6.8	–11.3	–13.3	–17.9	0.32	0.87	0.93	1.28
	Breakdancers	–7.7	–12.0	–13.6	–18.2	0.31	0.82	1.00	1.31
	Average	**–6.8**	**–11.8**	**–14.2**	**–18.0**	**0.36**	**0.82**	**0.94**	**1.22**
(30;39)	Lovebird1	–3.5	–8.4	–13.6	–18.1	0.21	0.38	0.87	1.21
	BookArrival	–2.3	–6.3	–15.3	–22.6	0.18	0.56	0.77	1.36
	Newspaper	–4.6	–7.6	–14.8	–20.8	0.24	0.49	0.98	1.41
	GT_Fly	–5.2	–8.9	–15.8	–22.6	0.15	0.47	1.10	1.44
	Undo_Dancer	–4.3	–9.1	–16.3	–23.6	0.24	0.58	1.13	1.49
	Ballet	–3.7	–7	–14.9	–22.7	0.27	0.52	0.92	1.47
	BreakDancers	–4.1	–7.9	–14.8	–21.9	0.26	0.57	1.03	1.44
	Average	**–4.0**	**–7.9**	**–15.1**	**–21.8**	**0.22**	**0.51**	**0.97**	**1.40**
(35;42)	Lovebird1	–1.6	–6.3	–14.7	–19.2	0.12	0.21	0.91	1.25
	BookArrival	2.6	–4.2	–16.8	–24.6	–0.13	0.25	0.88	1.55
	Newspaper	1.2	–3.9	–16.8	–23.9	–0.05	0.26	1.10	1.51
	GT_Fly	–2.8	–5.1	–17.4	–25.3	0.09	0.31	1.12	1.67
	Undo_Dancer	–2.4	–5.8	–18.9	–26.8	0.11	0.48	1.17	1.58
	Ballet	–2.3	–5.9	–16.9	–23.8	0.22	0.47	1.09	1.55
	Breakdancers	–2.7	–6.8	–17.4	-23.2	0.20	0.45	1.08	1.48
	Average	**–1.1**	**–5.4**	**–17.0**	**–23.8**	**0.08**	**0.35**	**1.05**	**1.51**
(45;48)	Lovebird1	–1.4	–5.7	–16.3	–21.1	0.10	0.18	0.96	1.38
	BookArrival	2.8	–3.3	–18.9	–25.7	–0.17	0.23	0.93	1.70
	Newspaper	2.0	–3.6	–18.7	–24.8	-0.10	0.22	1.37	1.65
	GT_Fly	–2.5	–4.1	–19.1	–26.1	0.07	0.27	1.40	1.78
	Undo_Dancer	–2.0	–4.0	–20.4	–27.4	0.09	0.42	1.31	1.70
	Ballet	–1.9	–3.2	–19.1	–24.7	0.18	0.40	1.29	1.62
	Breakdancers	–2.1	–4.7	–19.9	–24.2	0.17	0.39	1.32	1.58
	Average	**–0.7**	**–4.0**	**–19.0**	**–24.8**	**0.05**	**0.30**	**1.22**	**1.63**

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2019.2919198, IEEE Transactions on Image Processing

TABLE III.
BDBR and BDPSNR comparison at QP pair of (25;34) for a variety of sequences in the case when the distance between the reference view and virtual view gets larger. In this test, the virtual views at the same positions as those in Table I are synthesized, but the reference views with larger distance relative to the virtual view are selected as input views for each sequence.

Test sequence	Input views	BDBR (%)			BDPSNR (dB)		
		D-NOSE_DC	ADD_DC	ADD_P_ODC	D-NOSE_DC	ADD_DC	ADD_P_ODC
Lovebird1	5-9	–3.5	–8.6	–15.8	0.26	0.44	1.01
	4-10	–2.8	–6.3	–16.1	0.21	0.39	1.05
BookArrival	7-11	–4.9	–10.4	–17.2	0.31	0.52	1.26
	6-12	–3.2	–7.2	–18.1	0.23	0.44	1.32
Newspaper	3-7	–5.1	–9.3	–19.1	0.32	0.61	1.16
	2-8	–4.5	–7.0	–18.7	0.26	0.54	1.20
GT_Fly	4-10	–6.2	–10.3	–19.8	0.34	0.65	1.31
	3-11	–5.3	–7.1	–20.1	0.30	0.54	1.29
Undo_Dancer	2-7	–5.8	–8.6	–18.6	0.32	0.58	1.35
	1-8	–4.2	–6.5	–18.9	0.26	0.33	1.38

perform better when more occluded pixels are available to be optimized. To understand that, we emphasize that, with the increase of the distance between the reference and virtual views, the number of available allowable depth distortions in the range for each pixel can be decreased based on (7) and Lemma 1, which thus limits the performance gain that can be achieved by allowable depth distortion control, and in turn counter-acts the overall improvement of ADD_P_ODC.

D. Subjective Quality Evaluation

For subjective quality evaluation, Fig. 3 shows the synthesized views for the Undo_Dancer and Newspaper sequences using encoding schemes of D-NOSE_DC, ADD_DC, and ADD_P_ODC at the QP pair of (25;34). For better comparison, we also include the synthesized view rendered by the anchor-scheme-coded texture and depth as benchmark. As can be observed, all the test algorithms can improve the rendered view quality visually compared to the anchor, and ADD_P_ODC achieves the largest subjective quality gain. Especially, from the zoomed-in images shown in Figs. 3 (e)-(h) and (i)-(l), we can also observe that ADD_P_ODC generally yields better synthesis quality around the area of object edges where occlusion frequently occurs. Similar visual improvement on the synthesized views can be found by comparing the results of the Newspaper sequence in Figs. 3 (q)-(t). The reason for the visual improvement of ADD_P_ODC over other two algorithms can also be attributed to the optimization of occlusion-inducing depth pixels and the derivation of a more accurate allowable depth level change range for each possible depth error. For occlusion-inducing depth pixels, although the texture pixels are occluded in the synthesized view, as mentioned earlier, the depth errors of depth pixels still result in geometry errors in the rendered view, which may significantly affect the visual synthesis quality. ADD_P_ODC jointly optimizes all the pixels involved in the occlusion in a rate-distortion sense, which can make the resulting overall geometry errors minimal, thus preserving the sharp edges of the objects. On the other hand, for non-occlusion-inducing depth pixels, all the test algorithms rely on allowable depth distortion control. Although this kind of optimization method keeps the resulting geometry error unchanged, it can reduce the bit rate required by depth coding by properly selecting the larger depth distortion, which lead to that more bit rate can be allocated to texture pixels to enhance the visual synthesis quality. Therefore, the more accurate the allowable depth error range is, the better visual synthesis quality will be. The test of applying the proposed ADD_P_ODC algorithm on other sequences yields similar visual results. Finally, we give the subjective visual results of the to-be-occluded area in the reference views. For the Undo_Dancer sequence, the associated areas in the captured views 3 and 6 that will be occluded in the virtual view 4 are illustrated in Fig. 4 (a) and (b), respectively, where the non-black areas represent the to-be-occluded pixels in the reference views. It should be noted that, due to unequal baseline distances from the reference view to the virtual view used in warping the virtual view of the Undo_Dancer sequence, the proportions of the occluded pixels in the left and right views are not equal. Similarly, the associated to-be-occluded areas in the left and right reference views of the Newspaper sequence are presented in Fig. 4 (c) and (d). Compared to the Undo_Dancer sequence, "Newspaper" generally has more pixels that will result in occlusion, since it captures nearer scene and has larger baseline distance between the virtual and captured views.

E. Encoding Complexity Comparison

In this section, we quantitatively evaluate the time complexity of the proposed depth map coding optimization solution. We run all the test algorithms on a Windows 10 based desktop equipped with Intel Core i7 8700 CPU and 16GB Dual Channel DDR4 RAM. Fig. 5 shows the encoding time for a GOP among different depth map coding optimization methods. As can be observed, with respect to the original 3D-HTM reference software, D-NOSE_DC increases the encoding time by around 63%, ADD_DC consumes 124% more time complexity, and ADD_P_DC increases the time by 173%. The reason for the time increments of these algorithms is due to

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2019.2919198, IEEE Transactions on Image Processing

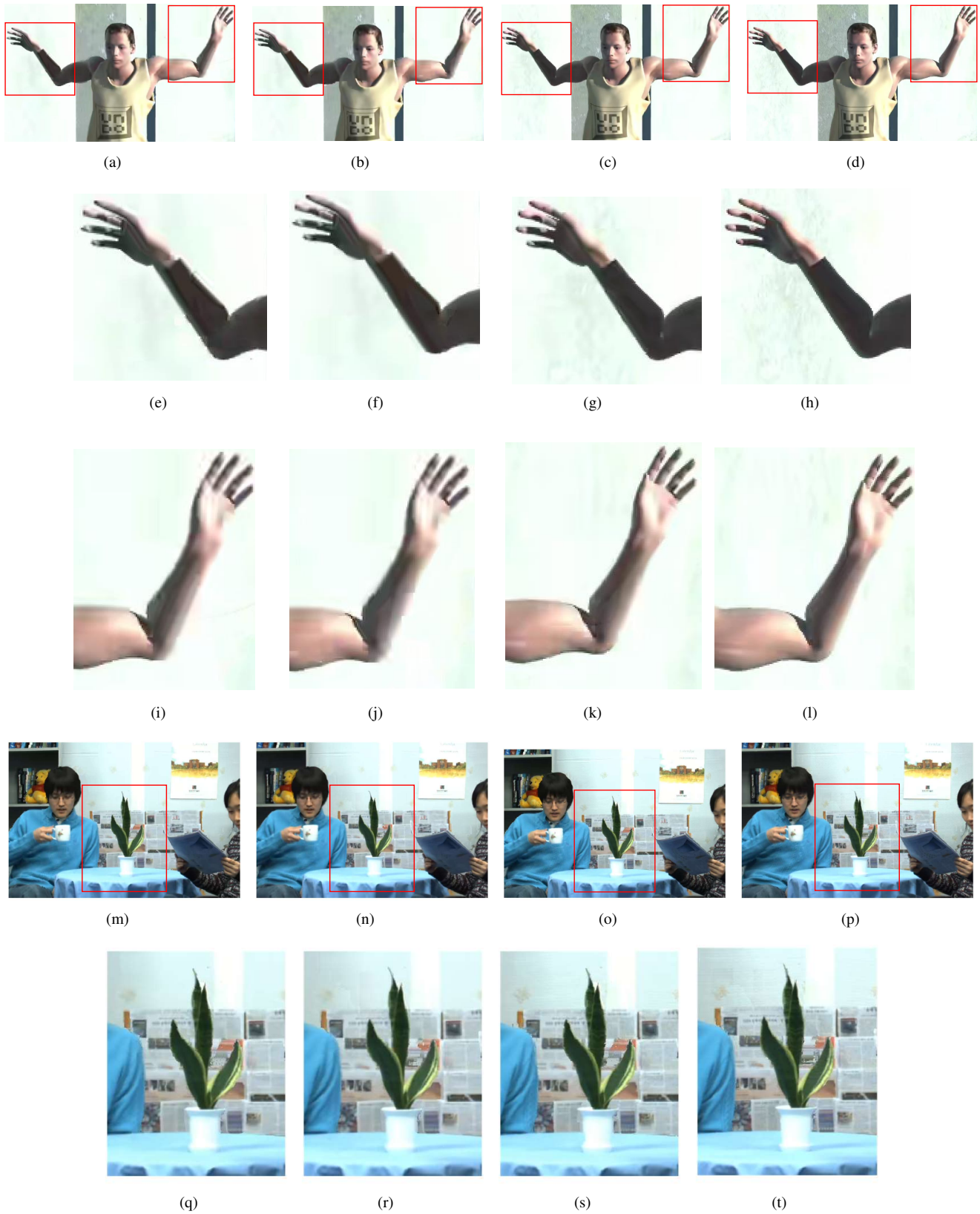
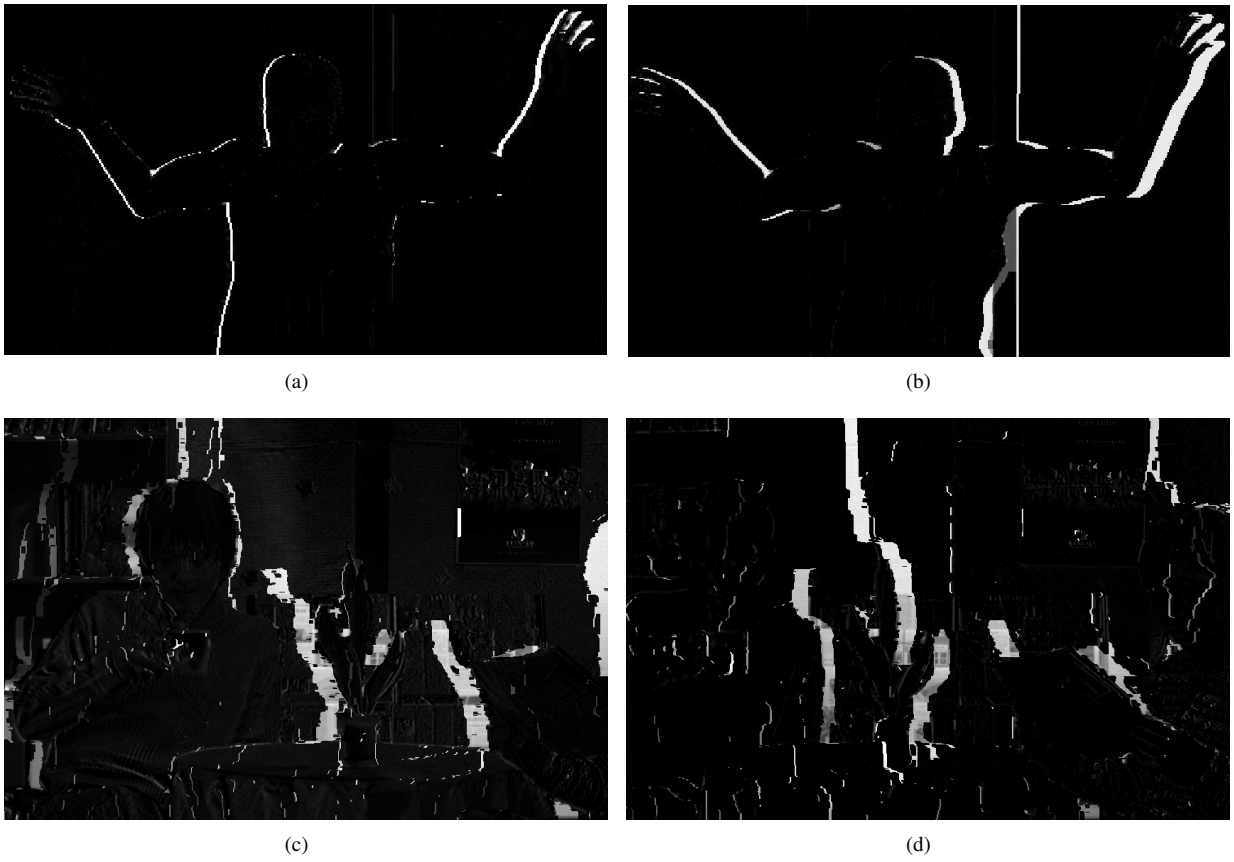


Fig. 3. Visual comparison of the synthesized views. (a)-(d) are the synthesized views for Undo_Dancer, with zoomed-in images in (e)-(l), while (m)-(p) are the synthesized images for BookArrival, with zoomed-in images in (q)-(t). The zoomed-in images are displayed in the same order their synthesized images appear. (a) Synthesized view rendered with anchor-coded depth images. (b) Synthesized view rendered with D-NOSE_DC-optimized depth images. (c) Synthesized view rendered with ADD_DC-optimized depth images. (d) Synthesized view rendered with ADD_P_ODC-optimized depth images. (e)-(h): Zoomed-in images of the highlighted areas on the left of images in the first row. (i)-(l): Zoomed-in images of the highlighted areas on the right of images in the first row. (m) Synthesized view with anchor method. (n) Synthesized view with D-NOSE_DC method. (o) Synthesized view with ADD_DC method. (p) Synthesized view with ADD_P_ODC method. (q)-(t): Zoomed-in images of the highlighted areas in the images in the fourth row.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2019.2919198, IEEE Transactions on Image Processing



(a) (b) (c) (d)

Fig. 4. Illustration of the regions in the reference view that are occluded in the virtual view, where the non-black areas represent the to-be-occluded pixels. (a) The to-be-occluded regions in the left view 3 of the Undo_Dancer sequence. (b) The to-be-occluded regions in the right view 6 of the Undo_Dancer sequence. (c) The to-be-occluded regions in the left view 4 of the Newspaper sequence. (d) The to-be-occluded regions in the right view 6 of the Newspaper sequence.

the calculation of the allowable depth errors for each pixel and the multiple pass encoding involved in computation of the bit rates of the allowable depth errors for each depth pixel. The more the allowable depth errors the optimization algorithm can obtain, the more time the encoder needs for selecting the optimal one. Therefore, due to higher estimation accuracy of ADD_P, ADD_P_DC generally needs the most computational complexity compared to D-NOSE_DC and ADD_DC. When comparing the time complexity between ADD_P_DC and ADD_P_ODC, we can find that the time increment of ADD_P_ODC over ADD_P_DC is around 8%. This time complexity increase can be attributed to the computation for determining the occluded pixels, the calculation of the total synthesis distortion involved in occlusion, and the dynamic programming solution. In the time-complexity-constrained 3-D video coding scenario, the complexity of the proposed depth map coding optimization solution can be reduced by employing the bit rate model to estimate the bit rate instead of multiple pass entropy encoding.

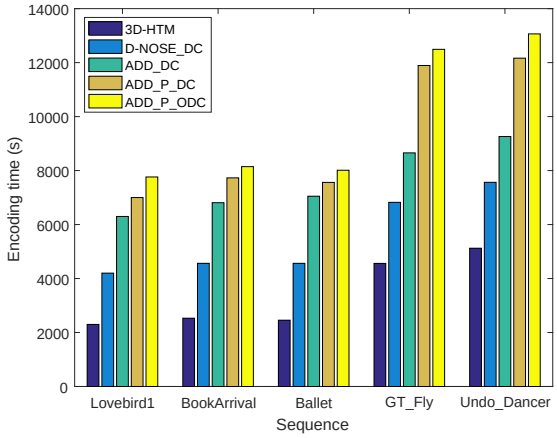


Fig. 5. Encoding time comparison between the test algorithms, i.e., 3D-HTM, D-NOSE_DC, ADD_DC, ADD_P_DC and ADD_P_ODC.

VI. Conclusion

In this paper, we address the problem of how to jointly optimize the occlusion-inducing pixels in a rate-distortion sense in depth map coding. Firstly, we generalize the allowable depth error range in zero disparity error case to non-zero disparity error case with theoretical proof. Then, toward the goal of improving the coding efficiency for the occlusion-inducing pixels while maintaining the occlusion order, we present an algorithm to select the depth distortion from the allowable depth distortion range for all the pixels involved in the occlusion. We formulate this algorithm in such a way that the overall view synthesis distortion subject to the given

bit rate is minimized. We explicitly consider the distortion dependency in occlusion due to warping competition. Finally, a dynamic programming based approach is proposed to facilitate the determination of the optimal depth distortion for each pixel. Experimental results demonstrate that the proposed algorithm considerably outperforms the state-of-the-art allowable depth distortion control algorithms, especially in the case when there is a large number of non-zero-disparity-error inducing depth pixels or occluded pixels.

REFERENCES

[1] K. Müller, P. Merkle, and T. Wiegand, "3-D video representation using depth maps," *Proceedings of the IEEE*, vol. 99, no. 4, pp. 643-656, Apr. 2011.

[2] K. Müller *et al.*, "3D high efficiency video coding for multi-view video and depth data," *IEEE Trans. Image Process.,* vol. 22, no. 9, pp. 3366–3378, Sep. 2013.

[3] B. T. Oh and K.-J. Oh, "View synthesis distortion estimation for AVC- and HEVC-compatible 3-D video coding," *IEEE Trans. Circuits Syst. Video Technol.,* vol. 24, no. 6, pp. 1006–1015, Jun. 2014.

[4] 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. (ICIP)*, Nov. 2009, pp. 721-724.

[5] W.-S. Kim, A. Ortega, P. Lai, and D. Tian, "Depth map coding optimization using rendered view distortion for 3D video coding," *IEEE Trans. Image Process.,* vol. 24, no. 11, pp. 3534–3545, Nov. 2015.

[6] B. T. Oh, J. Lee, and D.-S. Park, "Depth map coding based on synthesized view distortion function," *IEEE Journal of Selected Topics in Signal Process.*, vol. 5, no. 7, pp. 1344-1352, Nov. 2011.

[7] Y. Zhang, S. Kwong, L. Xu, S. Hu, G. Jiang, C.-C. J. Kuo, "Regional bit allocation and rate distortion optimization for multiview depth video coding with view synthesis distortion model," *IEEE Trans. Image Process.,* vol. 22, no. 9, pp. 3497–3512, Sep. 2013.

[8] V. Velisavljevic, G. Cheung, and J. Chakareski, "Bit allocation for multiview image compression using cubic synthesized view distortion model," in *Proc. IEEE ICME*, Jul. 2011, pp. 1-6.

[9] L. Fang, Y. Xiang, N.-M. Cheung, and F. Wu, "Estimation of virtual view synthesis distortion toward virtual view position," *IEEE Trans. Image Process.,* vol. 25, no. 5, pp. 1961–1976, May 2016.

[10] Y. Zhao, C. Zhu, Z. Chen, and L. Yu, "Depth no-synthesis-error model for view synthesis in 3-D video," *IEEE Trans. Image Process.*, vol. 20, no. 8, pp. 2221-2228, Aug. 2011.

[11] Y. Zhang, S. Kwong, S. Hu, and C.-C. J. Kuo, "Efficient multiview depth coding optimization based on allowable depth distortion in view synthesis," *IEEE Trans. Image Process.,* vol. 23, no. 11, pp. 4879-4892, Nov. 2014.

[12] P. Gao, C. Ozcinar, and A. Smolic, "Optimization of occlusion-inducing depth pixels in 3-D video coding," in *Proc. IEEE Int. Conf. Image Process. (ICIP),* Oct. 2018, pp. 3608–3612.

[13] J. Y. Lee, H.-C. Wey, and D.-S. Park, "A fast and efficient multi-view depth image coding method based on temporal and inter-view correlations of texture images," *IEEE Trans. Circuits Syst. Video Technol.,* vol. 21, no. 12, pp. 1859–1868, Dec. 2011.

[14] S. Liu, P. Lai, D. Tian, and C. W. Chen, "New depth coding techniques with utilization of corresponding video," *IEEE Trans. Broadcasting,* vol. 57, no. 2, pp. 551–561, Jun. 2011.

[15] M.-K. Kang and Y.-S. Ho, "Depth video coding using adaptive geometry based intra prediction for 3-D video systems," *IEEE Trans. Multimedia*, vol. 14, no. 1, pp. 121–128, Feb. 2012.

[16] G. Tech, Y. Chen, K. Muller, J.-R. Ohm, A. Vetro, and Y.-K. Wang, "Overview of the multiview and 3D extensions of high efficiency video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 1, pp. 35–49, Jan. 2016.

[17] Z. Gu, J. Zheng, N. Ling, and P. Zhang, "Fast depth modeling mode selection for 3D HEVC depth intra coding," in *Proc. IEEE Int. Conf. Multimedia Expo Workshops (ICMEW)*, Jul. 2013, pp. 1-4.

[18] C.-S. Park, "Edge-based intramode selection for depth-map coding in 3D-HEVC," *IEEE Trans. Image Process.,* vol. 24, no. 1, pp. 155–162, Jan. 2015.

[19] L. Shen, K. Li, G. Feng, P. An, and Z. Liu, "Efficient intra mode selection for depth map coding utilizing spatiotemporal, inter-component and inter-view correlations in 3D-HEVC," *IEEE Trans. Image Process.,* vol. 27, no. 9, pp. 4195–4206, Sep. 2018.

[20] L. F. R. Lucas, K. Wegner, N. M. M. Rodrigues, C. L. Pagliari, E. A. B. Da Silva, and S. M. M. De Faria, "Intra predictive depth map coding using flexible block partitioning," *IEEE Trans. Image Process.,* vol. 24, no. 11, pp. 4055–4068, Nov. 2015.

[21] G. Cheung, W.-S. Kim, A. Ortega, J. Ishida, and A. Kubota, "Depth map coding using graph based transform and transform domain sparsification," in *Proc. IEEE Int. Workshop Multimedia Signal Process. (MMSP), Oct. 2011, pp. 1–6.*

[22] W. Hu, G. Cheung, X. Li, O. Au, "Depth map compression using multi-resolution graph-based transform for depth-image-based rendering," in *Proc. IEEE Int. Conf. Image Process. (ICIP),* Sep. 2012, pp. 1297–1300.

[23] W. Hu, G. Cheung, A. Ortega, O. Au, "Multiresolution graph Fourier transform for compression of piecewise smooth images," *IEEE Trans. Image Process.,* vol. 24, no. 1, pp. 419–433, Jan. 2015.

[24] H. Yuan, Y. Chang, J. Huo, F. Yang, and Z. Lu, "Model-based joint bit allocation between texture videos and depth maps for 3-D video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 4, pp. 485-497, Apr. 2011.

[25] L. Fang, N.-M. Cheung, D. Tian, A. Vetro, H. Sun, and O. C. Au, "An analytical model for synthesis distoriton estimation in 3D video," *IEEE Trans. Image Process.,* vol. 23, no. 1, pp. 185-199, Jan. 2014.

[26] M. Yang, N. Zheng, C. Zhu, and F. Wang, "A novel method of minimizing view synthesis distortion based on its non-monotonicity in 3D video," *IEEE Trans. Image Process.,* vol. 26, no. 11, pp. 5122–5137, Nov. 2017.

[27] B. Macchiavello, C. Dorea, E. M. Hung, G. Cheung, and W. Tan, "Loss-resilient coding of texture and depth for free-viewpoint video conferencing," *IEEE Trans. Multimedia*, vol. 16, no. 3, pp. 711-725, Apr. 2014.

[28] P. Gao, Q. Peng, and W. Xiang, "Analysis of packet-loss-induced distortion in view synthesis prediction-based 3D video coding," *IEEE Trans. Image Process.,* vol. 26, no. 6, pp. 2781–2796, Jun. 2017.

[29] S. Ma, S. Wang, and W. Gao, "Low complexiy adaptive view synthesis optimzation in HEVC based 3D video coding," *IEEE Trans. Multimedia,* vol. 16, no. 1, pp. 266–271, Jan. 2014.

[30] D. Zhang and J. Liang, "View synthesis distortion estimation with a graphical model and recursive calculation of probability distribution," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 5, pp. 827-840, May 2015.

[31] J. Jin, J. Liang, Y. Zhao, C. Lin, C. Yao, and A. Wang, "A depth-bin-based graphical model for fast view synthesis distortion estimation," *IEEE Trans. Circuits Syst. Video Technol.,* published online, Jun. 2018, DOI: 10.1109/TCSVT.2018.2844743.

[32] C. Fehn, "Depth-image-based rendering (DIBR), compression and transmission for a new approach on 3-D-TV," in *Proc. 11th SPIE Stereoscopic Displays Virtual Reality Syst.*, Jan. 2004, pp. 93-104.

[33] G. Tech, K. Muller, H. Schwarz, and T. Wiegand, "Partial depth image based re-rendering for synthesized view distortion computation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 6, pp. 1273–1287, Jun. 2018.

[34] A. D. Abreu, G. Cheung, P. Frossard, and F. Pereira, "Optimal Lagrange multipliers for dependent rate allocation in video coding," in *CoRR, vol. /1603.06123*, Mar. 2016. [Online]. Available: https://arxiv.org/abs/1603.06123

[35] 3D-HTM reference software version 16.0. Accessed: Mar. 2016. [Online]. Available: https://hevc.hhi.fraunhofer.de/svn/svn_3DVCSoftware/tags/HTM-16.0/

[36] "HHI Test Material for 3D Video," Document M15413, ISO/IEC JTC1/SC29/WG11, Archamps, France, May 2008.

[37] MSR 3D Video Dataset. Accessed: Mar. 2015. [Online]. Available: https://www.microsoft.com/en-us/download/details.aspx?id=52358

[38] "Contribution for 3D Video Test Material of Outdoor Scene," Document 15371, ISO/IEC JTC1/SC29/WG11, Apr. 2008.

[39] "3DV Sequences of ETRI and GIST," Electron. Telecommun. Res. Institute and Gwangju Inst. Sci. Technol., Daejon, Korea, Apr. 2008. Accessed: Apr. 2013. [Online]. Available: ftp://203.253.130.48

[40] Nokia. Multi-View Test Sequence Download Page. Accessed: Apr. 2013. [Online]. Available: ftp://mpeg3dv.research.nokia.com

[41] ITU-T SG 16 WP 3, ISO/IEC JTC 1/SC 29/WG 11, "Common test conditions of 3DV core experiments," document JCT3V-G1100, San Jose, CA, USA, 2014.

[42] G. Bjøntegaard, *Calculation of Average PSNR Differences Between RD Curves*, document VCEG-M33, ITU-T SG16/Q6 (VCEG), Austin, TX, Apr. 2001.