

VIEW AND RATE SCALABLE MULTIVIEW IMAGE CODING WITH DEPTH-IMAGE-BASED RENDERING

Vladan Velisavljević¹, Vladimir Stanković², Jacob Chakareski³, Gene Cheung⁴

¹Deutsche Telekom Laboratories, Berlin, Germany, ²University of Strathclyde, Glasgow, UK

³Ecole Polytechnique Fédérale de Lausanne, Switzerland, ⁴National Institute of Informatics, Tokyo, Japan

ABSTRACT

“Texture plus depth” refers to the format where a sender encodes both texture and depth maps at multiple camera-captured viewpoints. Having received such a representation, the decoder can synthesize novel intermediate view images via depth-image-based rendering (DIBR), using as anchors the texture and depth maps of the two closest captured viewpoints. Ideally then, one would optimally allocate available source coding bits among the encoded texture and depth maps, such that the synthesized view distortion is minimized. However, in many practical application scenarios the precise rate constraint may either: i) be unknown at encoding time, or ii) it can take on multiple values for clients of heterogeneous connectivities. In this paper, we propose a flexible codec and an associated bit allocation strategy to address both of these scenarios. In particular, we first present an edge-adaptive wavelet multiview image codec capable of producing a scalable bitstream from which proper subsets can be extracted and decoded at different bit-rates. Given our scalable codec, we then propose a rate allocation algorithm that performs one of the following two actions. The algorithm will either incrementally increase the number of bits for encoding texture or depth maps of already encoded viewpoints, or it will introduce into the scalable representation new texture or depth maps of previously uncoded captured viewpoints. The incremental choice of either refining an existing view or introducing a new one is carried out one layer at a time, such that the associated rate-distortion tradeoff is locally optimized. By employing our novel bit allocation strategy the proposed coder outperforms the state-of-the-art H.264/SVC codec as well as the same wavelet-based coder when armed with a simple suboptimal bit allocation with the same rate allocated to each map, in all coding scenarios studied in our experiments. Furthermore, our coder can achieve an arbitrarily fine granularity of encoding bit rates, while providing the additional functionality of view embedded encoding, unlike the other related coders that we examined.

Index Terms— Multiview imaging, depth-image-based rendering, bit allocation, view and rate scalable encoding

The work of J. Chakareski has been supported by the Swiss National Science Foundation under Ambizione grant PZ00P2-126416.

1. INTRODUCTION

The advances in camera technology have enabled capturing images from an array of consumer-level cameras at a reasonable cost. If in addition to texture maps (typical RGB or grayscale images), *depth maps* (comprising per-pixel distances between captured objects in the scene of interest and the camera) at the same captured view locations are also available¹, then both texture and depth maps can be encoded at the sender into one bitstream, commonly called the “texture plus depth” format [3]. Having received such a representation, the decoder can then synthesize novel intermediate view images via depth-image-based rendering (DIBR) techniques [4], e.g., 3D warping [5], which employ the texture and depth maps of the two closest encoded view locations as anchors. The ability for the viewers to select and generate images at any desired viewpoint within a specified range for observation is a great leap forward in interactive viewing experience, enabling a host of novel applications, e.g., free viewpoint TV (FTV) [6].

Transmission resources are generally limited in the present day networks. Therefore, compression of the multi-view texture and depth map content associated with a scene is necessary. Ideally, one would optimally allocate the available encoding bit budget among texture and depth maps of captured viewpoints, such that the expected synthesized distortion is minimized. However, in many practical application scenarios, the precise rate constraint may be unknown at encoding time or can take on multiple values for clients with heterogeneous connectivities and device capabilities. The question that the present paper addresses is how a good “texture plus depth” representation can be derived, given either uncertainty or multiplicity of rate constraints at encoding time, as explained above.

In particular, we first present an edge-adaptive wavelet multiview image codec that is capable of producing a view and rate scalable bitstream, out of which proper subsets can be extracted and decoded at different bit-rates. At each step of its operation, the codec simultaneously considers the possibility of either enhancing the quality of already encoded views or introducing a new viewpoint into the scalable representation.

¹Depth maps can be captured directly using time-of-flight cameras [1], or estimated using depth-estimation algorithms [2].

The specific choice between refinement of an already encoded texture or depth map, for one of the encoded views, or compression of the texture or depth map of a newly selected view, is done based on the rate-distortion efficiencies of the two operations. Constructing a scalable representation in such a way allows us to incrementally add encoding bits to the compressed content, while maintaining high quality synthesized view distortion. We can control the scalability level of the compressed bitstream that our codec produces in response to the specific deployment scenario. For instance, the codec can enable decoding at very fine-grain bit-rates, which is necessary for the case when the transmission rate is unknown ahead of time. Equally important, the codec can also create a somewhat coarser, layered, representation of the compressed content that is more suitable for the case when there is a finite set of transmission rate constraints that needs to be met. We show via simulation experiments that our coder with the novel optimized rate allocation can achieve a superior rate-distortion performance to those of the same coder with a simple uniform rate allocation and the state-of-the-art H.264/SVC scalable coder [7]. Furthermore, unlike H.264/SVC, our coder can provide an arbitrarily fine granularity of the encoded bit rate, while simultaneously providing view scalability.

The paper is organized as follows. We first discuss related work in Section 2. Then, we overview the edge-adaptive wavelet codec that we use for multiview image coding in Section 3. Subsequently, we formulate our bit allocation problem for the two scenarios under consideration in Section 4. Finally, we present our experimental results and provide some concluding remarks in Sections 5 and 6, respectively.

2. RELATED WORK

In general, many different representations of a static scene are possible for image-based rendering of any viewpoint at the receiver, e.g., layered depth images [8], light fields [9], lumbigraph [10], and view-dependent texture mapping (VDTM) [11]. See [12, 4] for surveys of representations in the literature. In contrast, “texture + depth” format [3]—the focus of this paper—has one texture and depth map at each captured viewpoint, where each depth map is a 2D representation of the 3D surface associated with the static scene. Image sequences encoded in the “texture + depth” format can enable the decoder to synthesize novel intermediate views via depth-image-based rendering (DIBR) techniques such as 3D warping [5].

“Texture + depth” format has several desirable properties. First, depth maps can be obtained relatively easily, either via stereo-matching algorithms [2], or directly using time-of-flight cameras [1]. Second, depth maps can better handle scenery with multiple objects compared to mesh-based model that requires dense image sampling around a single object. Finally, “texture + depth” format is more adaptable to dynamic scenes where objects change positions and shapes over time.

The present paper extends our previous work [13] on bit allocation among texture and depth maps for DIBR. [13] as-

sumed that a single rate constraint is known a priori, so that the encoder can optimally select texture and depth maps of captured viewpoints for encoding at appropriate rates to minimize synthesized view distortions. In this paper, we assume either: (1) the singular rate constraint is not known at encoding time, or (2) there are multiple rate constraints given heterogeneous clients have different transmission rate requirements. This new problem is more challenging than one in [13] and calls naturally for a scalable coding approach.

A bitstream is called scalable or layered if it can be truncated in such a way that the resulting sub-streams are still decodable, providing lower reconstruction quality or resolution than the original stream. A layered bitstream starts with the most important layer (base layer), and continues with a set of progressively less important layers (enhancement layers).

H.264 Scalable Video Coding (SVC) [7] is a recent extension of the H.264/AVC standard that provides efficient scalability functionalities with competitive video quality. An SVC encoded stream has a layered structure consisting of an AVC base layer (for compatibility with AVC), and one or more enhancement layers that provide temporal, spatial, and quality scalability, or any combination of these. Quality or signal-to-noise ratio (SNR) scalability, the focus of our paper, enables the use of a single stream to describe video content at different fidelity levels. In this way, the receivers that only receive a part of the stream can still reconstruct the content, though at lower quality. The more enhancement layers the receiver decodes the higher the reconstruction video quality. State-of-the-art wavelet-based scalable video coders (see [14] and references therein) that use motion-compensated temporal filtering usually provide better quality scalability features than SVC but suffer from performance loss. However, a recent JPEG2000-compatible scalable wavelet-based codec of [15] provides results close to that of H.264.

Scalability has been used for multiview video coding (MVC). Extensions of single-view SVC to MVC is presented in [16, 17]. Building on single-view wavelet-based scalable video coding, in [18, 19, 20], scalable multiview wavelet-based video coders are proposed that outperform simulcast coding with AVC and SVC. In [21], SVC and MVC are combined for joint texture+depth coding, where each view is coded as a two-layer representation, with the texture forming the base layer and the depth the enhancement layer, coded using the coarse granular scalability (CGS) of SVC.

3. MULTIVIEW IMAGE CODER AND RENDERING

In this section, we first review the concepts of shape-adaptive wavelet transforms (SA-WT) and depth image-based rendering for virtual view synthesis. Then, we explain how these concepts are combined in our scalable multiview image coder.

3.1. Shape-adaptive wavelet image codec

The SA-WT has been originally proposed in [22] to efficiently process irregular shapes of objects in images, where the wavelet filtering has been adapted to object boundaries to

avoid filtering across edges and generating high magnitude wavelet coefficients. The SA-WT has been modified in [23] to allow for adaptation of the wavelet filtering to the shapes defined by open contours.

For encoding texture and depth images, we use the modified version of the SA-WT from [23] followed by the wavelet-based image coder Set Partitioning in Hierarchical Trees (SPIHT) [24]. The coder is applied to each image separately, while the output bit rate is controlled for optimizing the overall RD performance.

3.2. Depth image-based rendering

In a common DIBR setup, a virtual view is synthesized using information captured at two anchor (*reference*) viewpoints. The captured information consists of a texture map and an associated per-pixel depth map that is used to determine the distance (*depth*) between the camera and the scene's 3D surface. A virtual view in between the two anchor views is synthesized by warping the captured texture maps to the new virtual view location, where the corresponding disparity shift is computed from the captured depth maps. To prevent overwriting the foreground with background pixels in the synthesized view, a depth buffer is maintained such that, given more pixels projected to the same pixel coordinate at the virtual view, only the closest pixel (with the minimal depth) is retained.

Denote the two anchor views as *left*, v_l , and *right*, v_r , and the associated captured texture and depth maps as t_l , d_l , t_r and d_r . Warping the two texture maps, t_l and t_r , to the virtual viewpoint v results in two projections, $t'_{l \rightarrow v}$ and $t'_{r \rightarrow v}$, that are not perfectly identical in realistic circumstances due to occlusion or rounding of the captured depth values. Following the results of [25], these two projections are blended using the following equation:

$$t_v(i) = \begin{cases} (1-x)t'_{l \rightarrow v}(i) + xt'_{r \rightarrow v}(i) & t'_{l \rightarrow v}(i), t'_{r \rightarrow v}(i) \neq 0, \\ t'_{r \rightarrow v}(i) & t'_{l \rightarrow v}(i) = 0, \\ t'_{l \rightarrow v}(i) & t'_{r \rightarrow v}(i) = 0, \\ 0 & t'_{l \rightarrow v}(i) = t'_{r \rightarrow v}(i) = 0 \end{cases}, \quad (1)$$

where i is the pixel coordinate, x is the distance between v and v_l and $t'_{l \rightarrow v}(i) = 0$ or $t'_{r \rightarrow v}(i) = 0$ means that the respective pixel value is unavailable. The remaining zeros obtained by the bottom line in (1) are filled in a post-processing step of inpainting or interpolation.

3.3. Scalable multiview image codec

Our proposed scalable multiview image coder generates coding layers. At each coding layer, the coder encodes a selected subset of texture and depth maps from the captured views. For each selected viewpoint, the SA-WT followed by SPIHT is used to compress the difference between the original captured uncompressed image and the prediction at the same viewpoint, which is either (i) the previously quantized version of the same image, or (ii) a synthesized image obtained via DIBR using the closest left and right previously encoded views as anchors. Thus, in case (i), the encoder refines an

already compressed view using the best predictor, whereas in case (ii), the encoder starts encoding a new captured view.

Given the possibility to control the encoding bit rate of each image and the freedom to select the views that are encoded at each layer with respect to their available predicted instances, the coder chooses the optimal strategy in order to optimize the RD performance. The optimization techniques depend on the specific coding scenarios and criterion functions, as explained next.

4. FORMULATION

4.1. System Overview

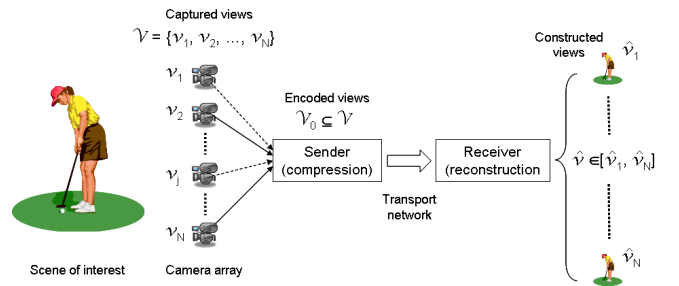


Fig. 1. Overview of a multiview imaging system. A set \mathcal{V} of N images are captured by a 1D camera array from different viewpoints. A subset \mathcal{V}_0 of views is encoded and transmitted. The receiver reconstructs view $\hat{v} \in [\hat{v}_1, \hat{v}_N]$ selected by the client. Solid lines from Camera array to Sender denote the views in \mathcal{V}_0 .

We first outline the multiview imaging system that our bit allocation algorithms target. A set of capturing cameras in a 1D array, at viewpoints $\mathcal{V} = \{v_1, \dots, v_N\}$, $v_i < v_{i+1}$ ², sense texture and depth maps at their respective locations, synchronously. A multiview image encoder encodes the captured texture and depth maps at selected viewpoints $\mathcal{V}_0 \subseteq \mathcal{V}$. Each view $v_{j_m} \in \mathcal{V}_0$, for $m = 1, \dots, |\mathcal{V}_0|$, is allocated encoding rates R_{t,j_m} and R_{d,j_m} , respectively, for its texture and depth maps. The compressed content corresponding to \mathcal{V}_0 is then transmitted to interested client(s). Fig. 1 provides an illustration of the multiview image communication system described above.

Upon receipt of the encoded bitstream, a client can select any view $v \in [v_1, v_N]$ for reconstruction and display, where the chosen view v can be at any continuous-coordinate location in between the left-most and right-most views v_1 and v_N . Because the encoder does not know a priori which viewpoint the client will choose to view, the objective is to minimize the

²We denote view v_1 as the left-most and view v_N as the right-most even though the viewpoint indices do not have necessarily to grow in the rightward direction.

aggregate distortion at all allowed viewpoints, i.e.,

$$D(\mathcal{V}_0) = \int_{v=v_1}^{v_N} d_s(v) dv \quad (2)$$

where $d_s(v)$ is the reconstructed view distortion at viewpoint v . If v is actually one of the coded viewpoints, i.e., $v \in \mathcal{V}_0$, then $d_s(v)$ is the distortion between the encoded and captured images at view v . On the other hand, if v is a *virtual view*, then the image at v is synthesized via DIBR using as anchors the texture and depth maps of the two closest encoded left and right viewpoints, v_l and v_r , respectively. The distortion $d_s(v)$ for $v_l \leq v \leq v_r$ then corresponds to the difference between the images reconstructed at view v using respectively the original texture and depth maps of views v_l and v_r or their compressed versions. Following the conclusion from [26], we model $d_s(v)$ as a cubic polynomial of viewpoint location v .

4.2. Rate Constraints

The total bit rate transmitted from the encoder to a client is equal to the sum of the rates allocated to all encoded texture and depth maps, i.e., $R = \sum_{m=1}^{|\mathcal{V}_0|} (R_{t,j_m} + R_{d,j_m})$. Now, the underlying transport network can impose multiple rate constraints on R . In particular, we consider the following two prospective scenarios. In the first one, we assume that the eventual transmission rate is unknown at encoding time. Therefore, a scalable bitstream, whose subsets can be decoded at very fine granularity in order to adapt to any possible transmission rate, is desirable. We refer to this setup as the *rateless constrained problem*. In the second scenario we consider, we assume that there are K monotonically increasing rate constraints r_1, r_2, \dots, r_K known at encoding time at which the encoded content needs to be served to heterogeneous clients. We denote this setting the *multiple constrained problem*. We address both of these problems by designing an efficient view and rate scalable representation of the encoded content that is described next.

4.3. Rateless Constrained Problem

For the rateless constrained problem, at encoding time we only know the possible range of the eventual transmission rate r , but not its actual value, i.e., we assume that $r \in [r_{\min}, r_{\max}]$. Hence, a scalable bitstream must be optimized for all possible rates between r_{\min} and r_{\max} .

More specifically, in the rateless constrained problem, using the scalable codec described in Section 3, we allocate bits into layers to construct layered scalable stream $\Phi = \{\phi_1, \dots, \phi_L\}$, where layer l , ϕ_l , has rate $R_\Phi(l)$ and layers 1 up to l collectively induce distortion $D_\Phi(l)$. Note that each layers 1 up to l will in general include texture and depth maps of different subset of captured views \mathcal{V} . The goal is to find layered stream Φ , whose subset of layers $\mathcal{L}(r) \leq L$ has coding rate no larger than r , and induces the smallest

possible resulting distortion $D_\Phi(\mathcal{L}(r))$, for all possible r . Mathematically, we write:

$$\min_{\Phi} \sum_{r=r_{\min}}^{r_{\max}} D_\Phi(\mathcal{L}(r))$$

$$\mathcal{L}(r) = \max_{l=1, \dots, L} \{l\}, \quad \text{s.t.} \quad \sum_{k=1}^l R_\Phi(k) \leq r \quad (3)$$

where $\mathcal{L}(r)$ is the largest number of layers l such that the total rate of the first l layers, $\sum_{k=1}^l R_\Phi(k)$, do not exceed rate budget r .

To solve (3), we design the following allocation strategy.

1. **Initialization:** Set $\mathcal{V}_0 = \{v_1, v_N\}$, $R_{t,1} = R_{d,1} = 0$, $R_{t,N} = R_{d,N} = 0$, and $l = 1$. Let ΔR be a given constant.
2. **Action - refinement:** Given a viewpoint $v \in \mathcal{V}_0$, compute and record the overall distortion in (2) resulting from each of the following two actions: (i) Refine the encoded texture map of v by encoding the difference between the original captured image and its previously quantized version, using ΔR additional bits; (ii) Refine the corresponding encoded depth map of v in the same way.
3. **Action - inserting new viewpoint:** For each viewpoint location $v \in \mathcal{V} \setminus \mathcal{V}_0$, synthesize the corresponding virtual view using DIBR, where as reference views (anchors) we employ either (a) the original texture and depth maps at the left v_l and right v_r viewpoints in \mathcal{V}_0 closest to v or (b) the previously quantized versions of the texture and depth maps associated with v_l and v_r . Then, compute and record the overall distortion $D(v \cup \mathcal{V}_0)$ using (2), for the following two cases:
 - (i) Encode the difference between the two synthesized versions of the texture map for view v , as described above, using ΔR bits; (ii) Encode the difference between the two synthesized versions of the corresponding depth map for v , again using ΔR bits.
4. **Choosing optimal action:** Choose the smallest distortion value from the $2|\mathcal{V}|$ values recorded in Steps 2 and 3 above. If the minimum is achieved by refinement (Step 2), for a given view $v \in \mathcal{V}_0$, then increase by ΔR the corresponding rate $R_{t,v}$ or $R_{d,v}$ already allocated to the texture and depth map information of view v , respectively. That is, either update $R_{t,v} = R_{t,v} + \Delta R$ or $R_{d,v} = R_{d,v} + \Delta R$. Otherwise (the minimum is achieved in Step 3), introduce the corresponding minimum achieving new viewpoint $v \in \mathcal{V} \setminus \mathcal{V}_0$ into the set of already encoded views, i.e., $\mathcal{V}_0 = v \cup \mathcal{V}_0$. Finally, set either

$(R_{t,v}, R_{d,v}) = (\Delta R, 0)$ or $(R_{t,v}, R_{d,v}) = (0, \Delta R)$, depending on whether option (a) or option (b), respectively, in Step 3 achieved the minimum value.

5. **Next layer:** Increase l : $l = l + 1$. If $l \leq \mathcal{L}(r)$, go back to Step 2.

4.4. Multiple Rate Constrained Problem

The multiple rate constrained problem can be formulated similarly to the rateless constrained problem (3), with the exception that the rate constraint r is known to be in a sparse set r_1, \dots, r_K . We can hence write the rate allocation problem as follows:

$$\min_{\Phi} \sum_{k=1}^K D_{\Phi}(\mathcal{L}(r_k)) \quad (4)$$

where $\mathcal{L}(r)$ is defined in (3).

In this scenario, we consider that the bit budget $r_j - r_{j-1}$ available for encoding the j -th layer, for $j = 1, \dots, K$ and assuming $r_0 = 0$, is not constant, i.e., ΔR , as in Section 4.3. In addition, we assume that $r_j - r_{j-1}$ can be significantly larger in value than the constant rate increment between individual layers employed in Section 4.3. Therefore, to allow for a finer bit allocation, we divide each layer budget into M sub-layers. In particular, the allocation strategy from Section 4.3 is repeated with fixed rate increments per sub-layer defined as $\Delta R_l^{(j)} = (r_j - r_{j-1})/M$ for all possible sequences of actions in steps 2 and 3. The sequence that results in the minimal distortion computed by (4) is chosen as optimal. Finally, we include the selected optimal sequence of actions into the scalable representation of the content by appending the corresponding refinement bitstream to the already compressed layers $1, \dots, n - 1$.

5. PERFORMANCE EVALUATION

5.1. Experimental setup

To compare the performance of our proposal to that of related state-of-the-art algorithms (e.g., H.264/SVC [7]), we use the Middlebury [27] data set `ROCKS2` with 7 texture and depth maps captured with a 1D camera setup at the resolution of 1110×1276 pixels. Due to constraints of the H.264/SVC JSVM9.8 reference software [28], the images were cropped to size 1024×1024 pixels. In addition, to reduce the computational complexity, only 4 viewpoints are used (viewpoints 2 to 5 from the original data set).

In the proposed system, the SA-WT is applied to the data set and the transform coefficients are encoded using SPIHT in two different encoding scenarios, as explained in Section 4. The virtual view synthesis distortion in between the encoded viewpoints is computed as an MSE using the cubic model (similarly to [26]) and the average synthesis distortion is used for performance evaluation.

5.2. Performance metric

To objectively evaluate scalable bitstreams encoded by different codecs and bit allocation algorithms, we define metrics for the rateless constraint problem and the multiple rate constraint problem, as follows. Let $D_{\Phi}(L(r))$ denote the aggregate distortion over all possible viewpoints $v \in [v_1, v_N]$, as defined in (2), given that up to $L(r)$ layers of the scalable bitstream Φ are used. Specifically, $L(r)$ denotes the maximum number of layers of bitstream Φ such that the aggregate data rate of these layers does not exceed r . In the rateless constraint problem, r can be any integer value between r_{\min} and r_{\max} . Hence, the metric characterizing the performance of a scalable bitstream Φ can be written as:

$$\sum_{r=r_{\min}}^{r_{\max}} D_{\Phi}(L(r)), \quad (5)$$

$$\text{for } L(r) = \max\{l\} \text{ s.t. } \sum_{k=1}^l R_k \leq r,$$

where R_k denotes the encoding rate of layer k in bitstream Φ .

On the other hand, in the multiple rate constraint problem, the transmission rate can only take on a finite set of values r_1, \dots, r_K . Therefore, the performance metric for a scalable bitstream Φ can be written in this case as

$$\sum_{k=1}^K D_{\Phi}(L(r_k)) \quad (6)$$

In the following, we employ the two metrics above to evaluate the performance of the coding algorithms we examine, for the two respective scenarios studied in this paper.

5.3. Rateless operation: Experiments

In the rateless scenario (Section 4.3), the rate granularity step for our bit allocation algorithm between two consecutive encoding layers is fixed to 0.01bpp^3 . We compare in Table 1 the performance metric of our proposed codec with optimal bit allocation (denoted as `optimal`) with three other alternatives. The first scheme uses the same proposed codec together with a simple bit allocation (`simple`), where all captured texture and depth maps are processed equally and the rate increments are allocated uniformly across all compressed maps.

The second and third schemes under examination use H.264/SVC to respectively encode four and two (leftmost and rightmost) captured views, for all layers, and are henceforth denoted as `SVC4` and `SVC2` in Table 1. Note that the SVC coder cannot achieve such a fine granularity of the encoding rate, and therefore only a few layers could be generated. For both `SVC4` and `SVC2`, we concatenated the selected views and encoded independently the corresponding texture and

³Note that the bit rates are expressed in terms of bits-per-pixel (bpp) computed as the total number of bits divided by the image resolution (1024×1024 in this case), even though the total number of pixels including all encoded texture and depth maps is equal to a multiple of the resolution.

Table 1. MSE of competing codecs & bit allocation schemes

Algorithm	MSE for rateless	MSE for multiple
optimal	90.08	78.12
simple	104.07	84.59
SVC4	117.10	103.38
SVC2	122.42	108.51

depth images into two CGS layers, using the same QP parameters for both texture and depth maps. In total, eight texture and ten depth layers are then generated by medium granular scalability (MGS) via bit extraction from the enhancement layer.

In the left column of Table 1, we show the aggregate distortion associated with each of the four schemes under examination, for the rateless scenario. In particular, we employed the metric (5) to compute the overall MSE associated with the scalable bitstream produced by each coding scheme, for the rate range $[r_{\min}, r_{\max}] = [0.01\text{bpp}, 1.5\text{bpp}]$. It can be seen from Table 1 that the proposed coding scheme and optimal allocation outperforms the other three techniques, providing a 13% reduction in MSE over `simple` as well as 20% and 26% reductions in MSE relative to `SVC4` and `SVC2`, respectively.

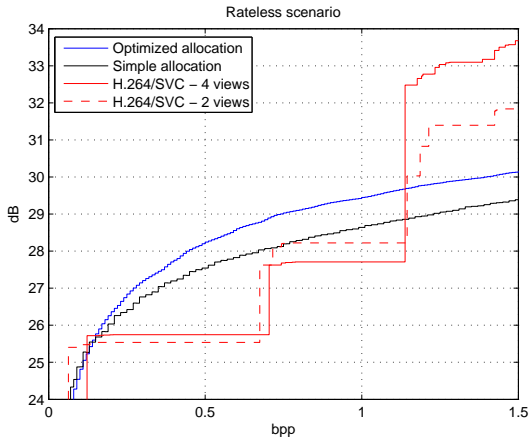


Fig. 2. Performance comparison of our coder with optimized and simple rate allocation (shown in blue and black, respectively) and H.264/SVC (red) in the rateless scenario.

In addition, we graphed in Fig. 2 the performances of the four schemes in terms of the average Peak-Signal-to-Noise-Ratio (PSNR) per viewpoint, as a function of the rate $r \in [r_{\min}, r_{\max}]$. It can be seen from Fig. 2 that our optimization technique `optimal` outperforms the other three schemes for the majority of rate values r . For instance, at $r = 0.5\text{bpp}$, `optimal` provides an improvement of over 2dB in average view quality over the two SVC schemes. Similarly, at $r = 1\text{bpp}$, `optimal` again outperforms the SVC schemes with a gain of 1.5 dB.

Finally, the optimal bit rate allocation, as computed by our optimization, is shown in Fig. 3. Specifically, the vertical bars in Fig. 3 illustrate the rate allocated to each of the captured texture and depth maps for several layers of the compressed

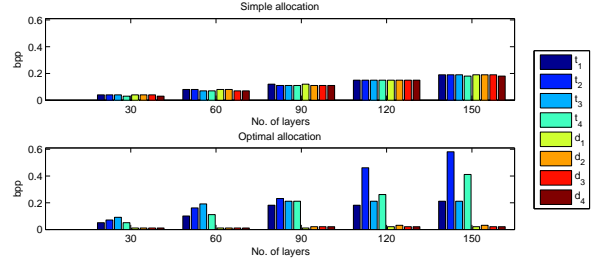


Fig. 3. Rate allocation to each of the captured texture and depth maps at several layers in the rateless scenario.

scalable bitstream.

5.4. Multi-Rate constrained operation: Experiments

In the multiple constraint scenario, we apply an exhaustive search to compute the best sequence of rate allocations, using $S = 4$, as explained in Section 4.4. The rates are constrained to the finite set $r \in \{0.06, 0.10, 0.15, 0.67, 0.74, 1.14, 1.19, 1.23, 1.36, 1.43, 1.47, 1.52, 1.55, 1.62, 1.65, 1.69, 1.71\}\text{bpp}$. Note that these rate constraints are obtained as the resulting rates from the SVC coder and, even though our coder can achieve any rates in between these values, we impose the same constraints for the sake of fair performance comparison. In Table 1, we also computed the aggregate MSE performances of the four schemes under comparison for this scenario, using the metric provided in (6). It can be seen from Table 1 that our proposed technique `optimal` again exhibits the lowest MSE distortion relative to the other three schemes.

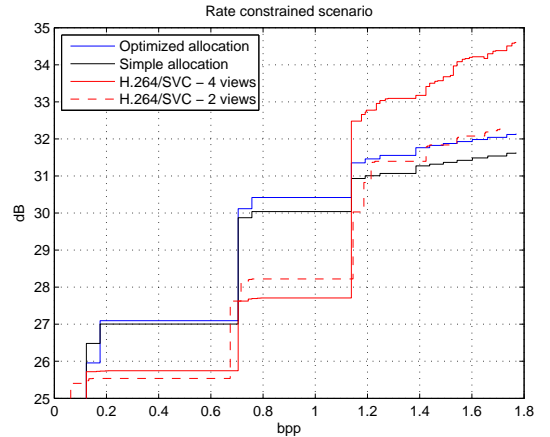


Fig. 4. Performance comparison of our coder with optimized and simple rate allocation (blue and black, respectively) and H.264/SVC (red) in the rate constrained scenario.

Next, in Fig. 4 we show the average PSNR performances of the four schemes, as a function of the available rate $r \in \{r_1, \dots, r_K\}$ computed in a similar manner as those shown in Fig. 2. It can be seen from Fig. 4 that also for this scenario `optimal` outperforms the other coding systems, except at very high rates, where the SVC coder has a comparable performance. Note again that H.264/SVC is only capable

of providing a very coarse granularity featuring large rate increments between consecutive layers, whereas our coder can adaptively choose any rates within the given range. However, in the comparison in Fig. 4, for the reason of comparison fairness, we match the constrained rates to those produced by the SVC coder.

Finally, the cumulative rate allocation across the different texture and depth maps, as computed by our optimization algorithm for this scenario, is shown in Fig. 5 after the 5-th, 9-th, and 14-th layer.

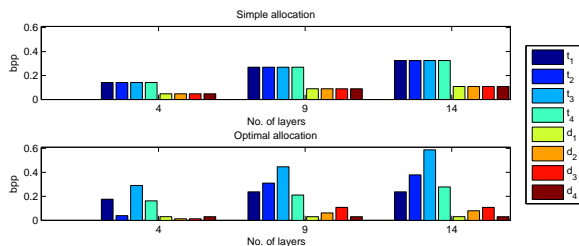


Fig. 5. Optimal rate allocation to each of the captured texture and depth maps after the 5-th, 9-th, and 14-th layer in the rate constrained scenario.

Note that the exhaustive search with S sub-layers in the multiple constraint scenario can result in high computational complexity of the entire scalable coding algorithm. Still, we focused in the present paper on formulating the problem under investigation, deriving an optimization framework to solve it, and on investigating the resulting performance via simulation experiments. We leave as future work the study of further improvements of our framework with respect to complexity and rate-distortion performance.

6. CONCLUSIONS

”Texture plus depth” refers to the format where texture and depth maps of multiple captured viewpoints are encoded at the encoder. In this paper, we address the problem of finding the best “texture plus depth” representation of a static scene for two network scenarios: i) the rateless constrained case where the transmission rate is not known at encoding time, and, ii) the multiple constrained case where the transmission rate takes on multiple values for heterogeneous clients with different network connectivities. In response, we first proposed a rate-scalable edge-adaptive wavelet multiview image codec to code the texture and depth maps of multiple captured viewpoints. We then define our objective of interest to be the synthesized view distortion at all possible intermediate views—each intermediate view is synthesized using coded texture and depth maps of the closest left and right views via depth-image-based rendering (DIBR). We posed the bit allocation problem for each scenario as the problem of constructing a single optimal scalable bitstream, where the bitstream’s distortion objective is evaluated at multiple rate points corresponding to different possible rate constraints. Our experimental results showed that the proposed scalable codec and bit allocation strategy outperform H.264/SVC, in all experi-

mental settings under consideration.

7. REFERENCES

- [1] S. Gokturk, H. Yalcin, and C. Bamji, “A time-of-flight depth sensor—system description, issues and solutions,” in *Conference on Computer Vision and Pattern Recognition Workshop (CVPRW)*, Washington, DC, June 2004.
- [2] M. Tanimoto, T. Fujii, and K. Suzuki, “Multi-view depth map of Rena and Akko & Kayo,” ISO/IEC JTC1/SC29/WG11 MPEG Document M14888, Oct. 2007.
- [3] P. Merkle, A. Smolic, K. Mueller, and T. Wiegand, “Multi-view video plus depth representation and coding,” in *IEEE International Conference on Image Processing*, San Antonio, TX, October 2007.
- [4] H.-Y. Shum, S.-C. Chan, and S. B. Kang, *Image-Based Rendering*, Springer, 2007.
- [5] Y. Morvan, D. Farin, and P. H. N. de With, “Multiview depth-image compression using an extended H.264 encoder,” in *Advanced Concepts for Intelligent Vision Systems, Lecture Notes in Computer Sciences*, 2007, vol. 4678, pp. 675–686.
- [6] T. Fujii and M. Tanimoto, “Free viewpoint TV system based on ray-space representation,” in *Proceedings of SPIE*, 2002, vol. 4864, p. 175.
- [7] H. Schwarz, D. Marpe, and T. Wiegand, “Overview of the scalable video coding extension of the H.264/AVC standard,” in *IEEE Transactions on Circuits and Systems for Video Technology*, September 2007, vol. 17, no.9, pp. 1103–1120.
- [8] J. Shade, S. Gortler, L. He, and R. Szeliski, “Layered depth images,” in *ACM SIGGRAPH*, New York, NY, September 1998.
- [9] M. Levoy and P. Hanrahan, “light field rendering,” in *ACM SIGGRAPH*, New Orleans, LA, August 1996.
- [10] S. Gortler, R. Grzeszczuk, R. Szeliski, and M. Cohen, “The lumigraph,” in *ACM SIGGRAPH*, New Orleans, LA, August 1996.
- [11] P. Debevec, C. Taylor, and J. Malik, “Modeling and rendering architecture from photographs,” in *ACM SIGGRAPH*, New Orleans, LA, August 1996.
- [12] H.-Y. Shum, S. B. Kang, and S.-C. Chan, “Survey of image-based representations and compression techniques,” in *IEEE Transactions on Circuits and Systems for Video Technology*, November 2003, vol. 13, no.11, pp. 1020–1037.
- [13] G. Cheung and V. Velisavljević, “Efficient bit allocation for multiview image coding & view synthesis,” in *IEEE International Conference on Image Processing*, Hong Kong, September 2010.
- [14] P. Eder, D. Engel, and A. Uhl, “Jpeg2000-based scalable video coding with mctf,” Universität Salzburg Technical Report 2007-04, October 2007.
- [15] T. André, M. Cagnazzo, M. Antonini, and M. Barlaud, “JPEG2000-compatible scalable scheme for wavelet-based video coding,” *EURASIP J. Image and Video Proc.*, vol. 2007, no. 1, pp. 1–11, 2007.
- [16] N. Ozbek and A. Murat Tekalp, “Scalable multi-view video coding for interactive 3d tv,” in *IEEE International Conference on Multimedia and Expo*, Toronto, Ontario, Canada, July 2006.
- [17] A. Droese, C. Clemens, and T. Sikora, “Single-view scalable video coding to multi-view based on h.264/avc,” in *IEEE International Conference on Image Processing*, Atlanta, GA, October 2006.
- [18] W. Yang, Y. Lu, F. Wu, J. Cai, K.N. Ngan, and S. Li, “4-D wavelet-based multiview video coding,” *IEEE Trans. Circuits Sys. Video Techn.*, vol. 16, no. 11, pp. 1385–1396, 2006.
- [19] J.-G. Garbas, U. Fecker, and A. Kaup, “Wavelet-based multi-view video coding with full scalability and illumination compensation,” in *The 15th ACM Int. Conf. MULTIMEDIA '07*, Augsburg, Bavaria, Germany, September 2007.

- [20] J.-G. Garbas, B. Pesquet-Popescu, M. Trocan, and A. Kaup, "Wavelet-based multi-view video coding with joint best basis wavelet packets," in *IEEE International Conference on Image Processing*, San Diego, CA, October 2008.
- [21] J. Zhang, M.M. Hannuksela, and L. Houqiang, "Joint multiview video plus depth," in *IEEE International Conference on Image Processing*, Hong Kong, China, September 2010.
- [22] S. Li and W. Li, "Shape-adaptive discrete wavelet transforms for arbitrarily shaped visual object coding," *IEEE Trans. on Circuits and Systems for Video Tech.*, vol. 10, pp. 725–743, 2000.
- [23] M. Maitre, Y. Shinagawa, and M.N. Do, "Wavelet-based joint estimation and encoding of depth-image-based representations for free-viewpoint rendering," in *IEEE Transactions on Image Processing*, June 2008, vol. 17, no.6, pp. 946–957.
- [24] A. Said and W. A. Pearlman, "A new, fast, and efficient image codec based on set partitioning in hierarchical trees," *IEEE Trans. on Circuits and Systems for Video Tech.*, vol. 6, pp. 243–250, 1996.
- [25] P. Merkle, Y. Morvan, A. Smolic, D. Farin, K. Muller, P.H.N. de With, and T. Wiegand, "The effects of multiview depth video compression on multiview rendering," *Signal Proc.: Image Comm.*, vol. 24, no. 1–2, pp. 73–88, 2009.
- [26] G. Cheung, V. Velisavljević, and A. Ortega, "On dependent bit allocation for multiview image coding & view synthesis," in *IEEE Transactions on Image Processing*, submitted in November 2010.
- [27] "2006 stereo datasets," <http://vision.middlebury.edu/stereo/data/scenes2006/>.
- [28] J. Reichel, H. Schwarz, and M. Wien, "Joint scalable video model 9 (jsvm 9)," Joint Video Team, Doc. JVT-V202, January 2007.