Attention-Weighted Texture and Depth Bit-Allocation in General-Geometry Free-Viewpoint Television

Camilo Dorea, *Member, IEEE,* and Ricardo L. de Queiroz, *Senior Member, IEEE*

***Abstract*—In a free-viewpoint television network, each viewer chooses its point of view from which to watch a scene. We use the concept of total observed distortion, wherein we aim to minimize the distortion of the view observed by the viewers as opposed to the distortion of each camera, to develop an optimized bit-rate allocation for each camera. Our attention-weighted approach effectively gives more bits to the cameras which are more watched. The more concentrated the viewer distribution, the larger the bit-rate savings, for a given total observed distortion, compared to the uniform rate allocation. We analyze and model the distortion of a synthesized view as a function of the distortions (both in texture and/or depth) of the nearby cameras. Based on such models, we develop optimal rate-allocation methods for texture images, considering a uniform bit allocation for depth, and for both texture and depth simultaneously. Simulation results are shown, demonstrating not only the correctness of the optimized solution, but also measuring its improvement against uniform rate allocation for a few viewer distributions.**

***Index Terms*—Free-viewpoint television, rate allocation, attention weighting.**

I. Introduction

In Free-Viewpoint Television (FTV) [1] a viewer can observe a remote three-dimensional (3D) scene by freely changing its viewpoint. This media can offer a continuum of selectable viewing positions, broadcast to a possibly very large audience. The scene, however, is generally captured by a finite number of cameras.

We envision a broadcast FTV system wherein N cameras are placed in an event, for example in a sports arena, and there are M viewers watching the program. Each camera generates texture video imagery and depth maps [2] as well. In a sense, each camera is both a texture (or color) camera and a depth map camera, and all captured information is broadcast over a network (e.g., Internet or a private one) to the viewers, as depicted in Fig. 1. Each viewer can then freely choose its own viewpoint. If there are no cameras which coincide with the viewer selected position, the service should synthesize the required view. We adopt the common depth-image based rendering (DIBR) approach [3], [4] to view synthesis. In the envisioned system, synthesis should occur anywhere in the network, probably closer to the end user, or within the end user's terminal. There is a provision for a slower feedback channel, regularly informing the encoder side at which position each viewer is watching the scene.

All cameras are then compressed with a given quality (bit-rate) and all camera streams are potentially broadcast to everyone in the viewer network. The question we ask ourselves and address here is how to allocate bits among all the camera compressors. When we refer to bit allocation among cameras, we mean the bit allocation of their respective compressors. The uniform allocation is the one in which all cameras are allocated the same distortion (or same bit-rate). We want to allocate more bits to the cameras that are more watched, i.e., either directly watched or involved in synthesis of requested viewpoints. In an extreme, if no viewer is watching or using a given camera, there may be no need to transmit it at all. This approach, in the FTV context, is called attention-weighted bit allocation. Our development is guided by the minimization of distortions of views observed by all end users, referred to as the total observed distortion.

Previous work on attention-weighted allocation of bit-rate across multiple views was initiated in [5]. The authors consider a multiview video coding scenario but do not address view synthesis. In [6], the authors propose attention-weighted rate allocation for texture in an FTV system. The proposal is developed for a regularly-spaced one-dimensional camera arrangement. In [7], rate allocation for texture was extended to general, possibly two-dimensional, camera arrays. However, these works do not consider an optimal allocation for depth maps. In [6] depth maps were uniformly coded (i.e., with constant quantization parameter) across all cameras whereas in [7] depth maps were assumed to be available in lossless form. In specific coding conditions [8], texture can comprise the greater portion of the overall bitstream, while the rate overhead of the depth video encodings may be kept relatively low. Further reduction of depth overhead may be achieved with dedicated, although non-standardized, coding techniques [9], [10] and certain FTV architectures [1] can altogether avoid depth trans-

C. Dorea and R. L. de Queiroz are with the Computer Science Department, University of Brasilia, DF, Brazil (emails: camilodorea@unb.br and queiroz@ieee.org).

This work has been partially supported by CNPq under grant 308150/20147 and by CAPES under grant BEX7000/14-6.

Manuscript received; revised.

mission by relegating to the end user the task of depth estimation. Nevertheless, in a general FTV broadcast system, depth maps can account for a substantial part of network traffic.

We consider depth maps, along with texture images, and model effects of their distortions upon observed view distortion. In particular, we consider effects of asymmetric distortions arising from non-uniform bit allocations. Based on such models, we develop an attention-weighted rate allocation technique for texture images and propose a joint technique for both texture and depth cameras. The procedures are cast within a general-geometry framework, allowing multiple cameras at arbitrary positions, as envisioned for the FTV system. System performance is demonstrated for a few viewer distributions on publicly available data sets containing regularly-spaced, one- and two-dimensional camera arrangements. We only consider multicasting the N video streams compressed independently with, for example, H.264/AVC [11], rather than using multiview plus depth extensions, such as 3D-AVC [12], since the independence of the views would facilitate the transport of only a few camera streams over sub-networks. Furthermore, multiview extensions such as 3D-AVC require a single, centralized coder which may be unfeasible when dealing with a large number of dispersed cameras. We compare our results to those of 3D-AVC. Other compression alternatives, such as HEVC [13], are also viable within our framework.

We have intentionally left network and routing issues out of this work. For example, three issues will be addressed elsewhere to preserve the length and focus of this text. Firstly, is how and where, within a possibly heterogeneous network, to synthesize views, in order to single-stream the synthetic view to the subscriber. Secondly, sub-networks may mirror the overall rate-allocation problem, so that the N camera streams could potentially be transcoded to fit the attention distribution of the sub-network. In an extreme, sub-networks who do not require all the camera viewpoints need not receive all the camera streams. Thirdly, scheduling can be weighted by viewer attention as well.

A few works (e.g., [14]–[17]) address rate allocation in the context of multiview video. Nevertheless the distortion considered was the video content distortion and not the total observed distortion that we propose. The work in [5] also addresses attention-weighted bit allocation, although without considering view synthesis, and its motivation is closer to ours here. The traditional bit-rate allocation approaches consider camera distortions as independent. We, however, consider the observed view distortion and this allows us to differentiate importance among cameras prior to rate allocation. Traditional rate allocation, within our perspective, is equivalent to assuming a uniform distribution of viewers, for example, a single viewer at each camera.

Our work is based on the ideas in [6], however, that work was inspired by others on free-viewpoint video such as [18] and [19]. In these single-user free-viewpoint approaches, one tries to send just a few (ideally two) viewpoints. However, as the viewer changes viewpoint, discontinuity may be observed. They use head-tracking to evaluate viewpoint and to predict the next one, in order to smooth the transitions and to increase the quality of experience. Another attention-driven approach, but also for a single consumer, was described in [20]. In it, they propose a proprietary video coder based on known image and video coding tools, combined with a scalable multiview video approach. As with [18] and [19], in [20] one tries to anticipate the single viewer viewpoint selection. In another very relevant work [21], a cubic distortion model for estimating the distortion of synthetic views in a one-dimensional camera arrangement was proposed. We propose alternative models, i.e., two different linear models to relate texture and depth distortions to the distortion of the synthetic view. Our models consider asymmetric distortions occurring in texture or depth, as well as their relative importance towards synthesis. Unlike [21], our models are applied in data sets with lower quality depth maps. Asymmetric coding, in the context of 3D video, has been subject of assessment in [22], [23]. In a very recent paper [24], in some sense concurrent to the present work, the authors also study the case of multiple viewers. They base their work on the proprietary coders in [20] and extend the concept to make a source-channel rate allocation problem for multiple viewers. As they state the problem quite differently from [6], they resort to a full search over a large space to find their solutions. Our approach and solution differs entirely in many aspects as they consider joint source and channel allocation and employ the view synthesis distortion models from [21].

Of paramount importance, in this development, is to properly model the distortion of a synthesized view using texture images and/or depth maps of asymmetric quality, and we do that in Sec. II. With the distortion model and borrowing from known rate-distortion optimization techniques, we develop an optimized rate-allocation algorithm for texture cameras in Sec. III and for both texture and depth in Sec. IV. Experimental results are presented in Sec. V, followed by the conclusions of the present work.

II. A view synthesis distortion model

As mentioned, we adopt the common DIBR [3] approach to synthesize free-viewpoint images. We label viewpoints, captured by any of N cameras, as $\vec{c}_j$. Each captured viewpoint is associated to texture (or color) and depth image [2] components, whose pixel values are denoted $p_j(x,y)$ and $d_j(x,y)$, respectively. Adjacent captured images are used to synthesize a virtual texture image at viewpoint $\vec{v}_m$ through the weighted blending [25] of projected pixels $p_{m|j}(x',y')$ such that

$$p_m(x',y') = \sum_{j\in\Psi_m} \beta_{mj}\; p_{m|j}(x',y') \tag{1}$$

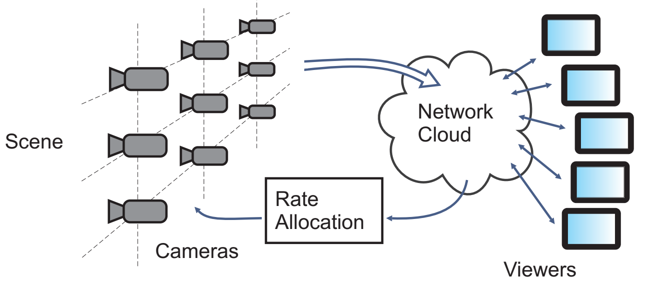


Fig. 1. General architecture for broadcast FTV using cloud services. Viewer attention feedback is used to determine optimal rate allocation among cameras.

where Ψ_m is the set of indices of cameras used in synthesis at virtual viewpoint $\vec{v}_m$, and β_{mj} are the camera blending weights which guarantee that, when available, projections from cameras closer to synthesis position are given greater relevance.

Depth information $d_j(x,y)$ is appropriately scaled into disparity values and used in determining correspondences between pixels of viewpoints $\vec{c}_j$ and $\vec{v}_m$ such that

$$\begin{aligned} p_{m|j}(x',y') = p_j\Big(& x + k g_x(\vec{v}_m,\vec{c}_j) d_j(x,y), \\ & y + k g_y(\vec{v}_m,\vec{c}_j) d_j(x,y) \Big) . \end{aligned} \tag{2}$$

The k component of the depth scaling factor represents intrinsic camera parameters, such as focal length and depth normalization constants [25]. Distances between viewpoints also take part in depth to disparity scaling. They are represented in their x- and y-components, respectively, by distance measures $g_x(\vec{v}_m,\vec{c}_j)$ and $g_y(\vec{v}_m,\vec{c}_j)$. Note that common camera configurations are often confined to one-dimensional (collinear or arc) setups in which adjacent cameras are to the left and right of synthesis position and associated disparity values are restricted to horizontal shifts along the x-axis. Our general-geometry framework does not impose limits on the number of associated cameras nor their positioning.

The blending weights assume the influence of a given camera to be inversely related to its distance from the desired viewpoint. As weights, $0 \le \beta_{mj} \le 1$ and, for a given synthesis position, $\sum_j \beta_{mj} = 1$. In order to define β_{mj} we use the following auxiliary weight:

$$\alpha_{mj} = \begin{cases} 1 & \text{if viewpoint at } \vec{v}_m \text{ uses camera at } \vec{c}_j \\ & \text{for synthesis,} \\ 0 & \text{else} \end{cases} \tag{3}$$

and an absolute distance measure between viewpoints: $g(\vec{v}_m,\vec{c}_j) = g(\vec{c}_j,\vec{v}_m)$. The blending weight used to ponder pixels projected from camera at viewpoint $\vec{c}_j$ is thus given by

$$\beta_{mj} = \frac{1/g(\vec{v}_m,\vec{c}_j)}{\sum_{k=0}^{N-1} \alpha_{mk}/g(\vec{v}_m,\vec{c}_k)} \tag{4}$$

for $g(\vec{v}_m,\vec{c}_k) \neq 0$. The closer $\vec{c}_j$ is to $\vec{v}_m$, the larger β_{mj}, until the point that the views coincide. In this case, $g(\vec{v}_m,\vec{c}_j) = 0$ and $\beta_{mj} = 1$, by definition, with all others $\beta_{mk} = 0$ for $k \neq j$. In one-dimensional setups with cameras to the left and right of the synthesized view, blending weights for each of the two cameras may be reduced to the form $(1-h)$ and h, respectively, where h is the distance from synthesis position to the left camera normalized by the camera baseline distance.

A. Distance-based distortion model for texture

Consider texture error ϵ (e.g., coding artifact) in a captured texture image at viewpoint $\vec{c}_j$. The erred pixel is given by

$$\tilde{p}_j(x,y) = p_j(x,y) + \epsilon_j(x,y) . \tag{5}$$

Projection of $\tilde{p}_j(x,y)$ onto a virtual viewpoint through (2) entails projections of both $p_j(x,y)$ and $\epsilon_j(x,y)$ such that the synthesized pixel becomes

$$\begin{aligned} \tilde{p}_m(x',y') &= \sum_{j\in\Psi_m} \beta_{mj}\, \tilde{p}_{m|j}(x',y') \\ &= \sum_{j\in\Psi_m} \beta_{mj} p_{m|j}(x',y') + \sum_{j\in\Psi_m} \beta_{mj}\epsilon_{m|j}(x',y'). \end{aligned} \tag{6}$$

In other words, texture distortion within a captured image is also subject to weighted blending. As such, given texture images from multiple cameras, one expects distortions within images closer to the virtual viewpoint to have a greater impact on synthesis.

This distance-dependent effect of texture distortion on resulting synthesis may be observed in Fig. 2. Cameras 3 (left) and 5 (right) of the *Poznan Street* data set [26] were used to synthesize views at various intermediate positions. Distortion was imposed on the captured texture images through compression with various combinations of quantization parameters (QP). (Further simulation details may be found in Sec. V.) First, observe that an increase in distortion of both left and right texture images (e.g., from both at QP=22 to both at QP=37) produces an increase in distortion (lower PSNR) across all synthesized intermediate views. Consider now an asymmetric distortion profile with greater distortion in the left texture image (QP=37) relative to right (QP=22). As synthesis position distances itself from the left camera's influence (position 3 towards 5), there is a clear increase in quality and PSNR as exemplified in Fig. 2. As expected, when the quality asymmetry is reversed, the image quality drops as synthesis occurs from left to right.

B. Distortion model for depth

Differently from texture distortions, the effects of depth distortions are only indirectly observed upon synthesized views. An error in depth pixel value of the form $\hat{d}_j(x,y) = d_j(x,y) + \varepsilon_j(x,y)$ leads to disparity error, i.e., a

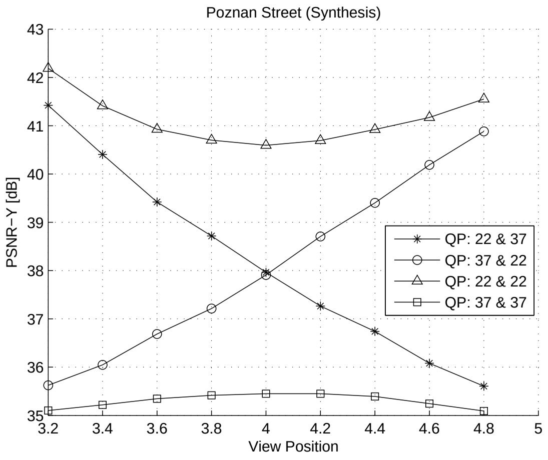


Fig. 2. PSNR of views synthesized from symmetrically and asymmetrically compressed texture images (cameras 3 and 5), using all depth maps compressed with QP=22, at various intermediate positions. PSNR is relative to synthesis from uncompressed images.

displacement in the position of projected pixels used in view synthesis, such that

$$\hat{p}_m(x',y') = \sum_{j\in\Psi_m} \beta_{mj} p_j\Big[x + kg_x(\vec{v}_m,\vec{c}_j)(d_j(x,y)+\varepsilon_j(x,y)), \\ y + kg_y(\vec{v}_m,\vec{c}_j)(d_j(x,y)+\varepsilon_j(x,y))\Big] . \quad (7)$$

In general, larger disparity errors cause larger distortion in synthesized view as spatial correlation of projections is degraded by displacements. Effects of depth distortion on synthesis are also subject to factors such as scene content and camera parameters. Synthesis distortion has been modeled in [27] as a linear function of disparity error, subject to a constant scaling factor s. Under this model, (7) may be expressed as a sum of a projected pixels term and a scaled disparity error term

$$\hat{p}_m(x,y) = \sum_{j\in\Psi_m}\beta_{mj}p_{m|j} + \sum_{j\in\Psi_m}\beta_{mj}s[kg(\vec{v}_m,\vec{c}_j)\varepsilon_j(x,y)] . \quad (8)$$

Note that the scaled disparity error term is also subject to weighted blending. Here, the depth error $\varepsilon_j(x,y)$ is multiplied by $g(\vec{v}_m,\vec{c}_j)$ and by β_{mj}, which in turn has been defined as inversely proportional to $g(\vec{v}_m,\vec{c}_j)$ in (4). We assume projections from multiple cameras are generally available for synthesis and that these are subject to weighted blending. Under this condition, unlike texture errors, the effects of depth error upon synthesis are in essence independent of the distance between viewpoints. This property is illustrated in Fig. 3. Views are synthesized at numerous intermediate positions for the *Poznan Street* data set as captured depth maps are subject to various compression combinations. The imposition of symmetric distortions in both left and right depth maps (e.g., from both at QP=2 to both at QP=8) produces a clear drop in PSNR across all synthesized views. Furthermore, asymmetric distortion conditions among left and right depth maps (e.g., QP=2 and QP=8 for the left and right cameras, or QP=8 and QP=2 for the left and right cameras) lead to approximately equal synthesis distortions regardless of the proximity or distance of the view from the camera. This is observed with depth maps of greater or lesser distortion.

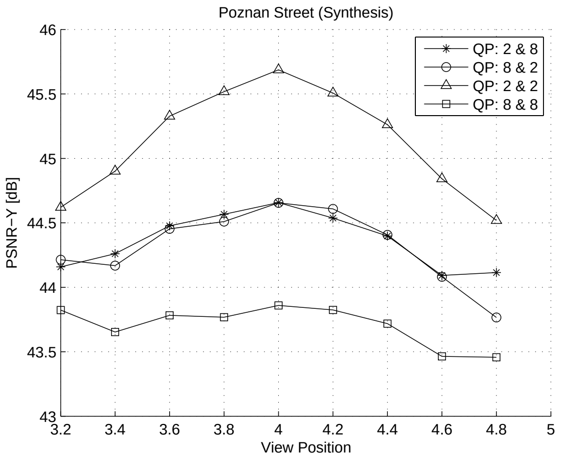


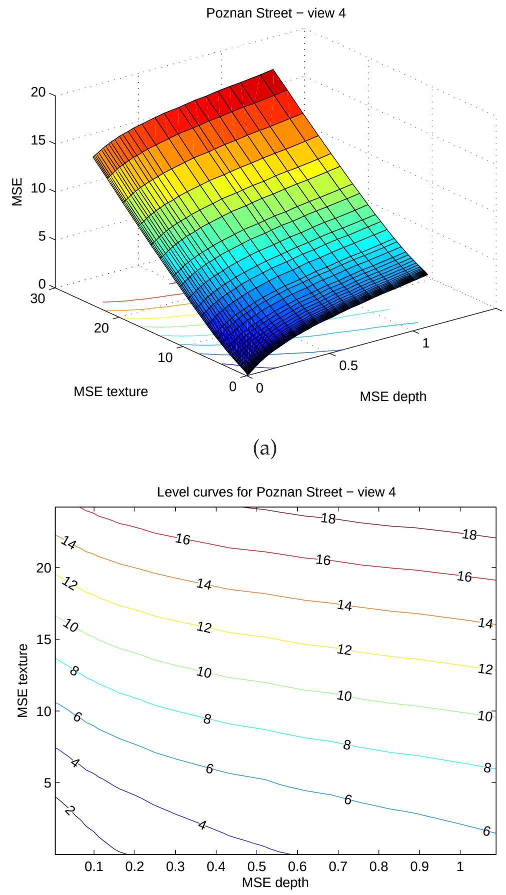
Fig. 3. PSNR of views synthesized from symmetrically and asymmetrically compressed depth maps (cameras 3 and 5), using all texture images compressed with QP=22, at various intermediate positions. PSNR is relative to synthesis from uncompressed images.

C. Texture and depth distortion ratio

As expressed by (6) and (8), synthesis distortion is directly proportional to texture distortion and to depth distortion within captured images. Nevertheless, distortions in texture and depth contribute in different degrees towards synthesis, depending on scene content and camera parameters. The relative importance of each of these components, or texture-to-depth ratio η, towards view synthesis can be empirically determined. We illustrate the procedure with synthesis results from the *Poznan Street* sequence. Fig. 4(a) presents a surface plot of synthesized view distortion for a range of texture and depth distortions. In this case, view 4 is synthesized from compressed images of cameras 3 and 5. For a given synthesis distortion, the relative importance of texture and depth distortions may be estimated through the inclination of the corresponding level curve upon the distortion surface. The curve is modeled by linear regression, with least squares-fitting, and its inclination is represented by the resulting line's slope. Texture-to-depth ratio is assumed as the average slope of the level curves within the tested range. The average slope of the curves depicted in Fig. 4(b) suggests, in this case, that $\eta = 8.6$. Once a texture-to-depth ratio has been determined, contributions of each component may be combined towards establishing synthesis distortion.

III. General Rate Allocation for Texture

Texture from each camera video, say the n-th camera, is compressed and transmitted using, for example,



(a)

(b)

Fig. 4. (a) MSE of synthesis of *Poznan Street* (view 4) from coded images (cameras 3 and 5) with respect to synthesis from original images for various texture and depth distortions and (b) associated level curves. Left and right texture (and depth) pairs are coded with same QP within the range 1-37 (and 1-27). Distortion of coded texture/depth pairs in terms of overall MSE relative to original texture/depth image.

H.264/AVC [28] with a given QP, spending an encoding bit-rate R_n^t and achieving a distortion D_n^t. Each viewer observes a synthetic image, synthesized from the video of cameras operating near the selected virtual viewpoint. For each virtual viewpoint, from a total of M viewers, we can assign a distortion δ_m which tells us the distortion incurred by the view when the camera texture videos are compressed.

We argue that the $\{D_n^t\}$ are not directly relevant and what is of real importance are the observed distortions $\{\delta_m\}$ which are experienced by the viewers, who are the actual consumers in the process. In this context, we want to minimize the overall rate

$$R = \sum_{n=0}^{N-1} R_n^t \tag{9}$$

while also minimizing the total observed distortion

$$D = \sum_{m=0}^{M-1} \delta_m \ . \tag{10}$$

The optimization may be achieved by minimizing the cost function $J = R + \lambda D$, where λ is a Lagrangian multiplier to control the overall rate-distortion trade-off [29], [30]. For that, the mechanisms we have at our disposal are the selection of the QP for each camera compressor, which controls the individual $R_n^t \times D_n^t$ trade-off. Note that we ignored the rate used for the transmission of the M feedback channels for two reasons. Firstly, they are not dependent on the compression and, secondly, the feedback transmission rate of viewer position can be kept very low, having reduced impact.

It is natural that for synthesizing a given viewpoint $\vec{v}_m$ we would use a few cameras, for example two in a linear arrangement: those immediately to the left and right of the requested viewpoint. Additionally employing other cameras, at greater distances, provides only marginal contributions to synthesis quality [31]. Assuming proportionality between observed view distortion and distortions in the captured texture views, the former is approximated as a linear combination of the latter such that

$$\delta_m = \sum_{j \in \Psi_m} w_j \ D_j^t \tag{11}$$

where w_j are respective weights ($0 \leq w_j \leq 1$ and $\sum w_j = 1$). Moreover, as noted in the observations pertaining to (6), the influence of texture distortion from a given camera is distance-based, in which its contribution to observed distortion is inversely proportional to its distance from the desired viewpoint. Thus, we express w_j in terms of α_{mj} and β_{mj} and distortions of synthesized views as

$$\delta_m = \sum_{n=0}^{N-1} \alpha_{mn} \ \beta_{mn} \ D_n^t \tag{12}$$

and

$$D = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \alpha_{mn} \ \beta_{mn} \ D_n^t \ . \tag{13}$$

If we define

$$\phi_n = \sum_{m=0}^{M-1} \alpha_{mn} \ \beta_{mn} \ , \tag{14}$$

then

$$D = \sum_{n=0}^{N-1} \phi_n \ D_n^t = \sum_{n=0}^{N-1} \tilde{D}_n \ . \tag{15}$$

One can view the above equation as a sum of adjusted distortion measures $\{\tilde{D}_n\}$ for each camera, which take into account all the viewers. The cost function we seek to minimize is now

$$J = R + \lambda D = \sum_{n=0}^{N-1} (R_n^t + \lambda \tilde{D}_n). \tag{16}$$

Optimal bit allocation is found by adjusting the QP, at each camera compressor, in order to minimize

$$J_n^t = R_n^t + \lambda \phi_n D_n^t. \tag{17}$$

A. Special case of regularly-spaced camera arrangements

In the special case of regularly-spaced cameras in one dimension (in a line or arc), we can normalize distances to the interval between cameras and measure all distances from the first camera, such that viewpoints $\{\vec{v}_i\}$ and $\{\vec{c}_i\}$ can be described as scalars indicating the position in the path of cameras and viewpoints. Then, $0 \leq v_k, c_k \leq N-1$, $g(c_k, c_{k-1}) = 1$, and $c_k = k$. Let us separate integer and fractional parts of a viewpoint as $s_k = \lfloor v_k \rfloor$ and $q_k = v_k - s_k$. Hence, v_k is synthesized from cameras at s_k and $s_k + 1$, such that $g(c_{s_k}, v_k) = q_k$ and $g(c_{s_k+1}, v_k) = 1 - q_k$.

The distortion associated with the k-th viewpoint is simply

$$\delta_k = (1 - q_k) D_{s_k} + q_k D_{s_k+1}, \tag{18}$$

such that $\alpha_{ij} = 0$ except for

$$\alpha_{i,s_i} = \alpha_{i,s_i+1} = 1 \ . \tag{19}$$

Also,

$$\beta_{i,s_i} = 1 - q_i \quad \text{and} \quad \beta_{i,s_i+1} = q_i \tag{20}$$

which is the linear-arrangement case described in [6].

B. Extreme cases

There are two extreme cases. Firstly, the proposed bit allocation would not improve the total observed distortion in the case of uniform distribution of viewers, i.e., where selected viewpoint distribution is analogous within every inter-camera interval. This would imply that all cameras (apart from the peripheral) receive exactly the same influence from viewers and, thus, have equal weights for their distortion. Therefore, each camera would have equally important distortion and the uniform bit allocation should be optimal. Secondly, in the other extreme, if there is only one camera being watched, no other camera is transmitted and we may re-allocate all their bit budgets to the camera being watched. This would result in the largest gain in optimizing the bit allocation for the cameras.

IV. Rate Allocation for Texture and Depth

The free-viewpoint video transmission of texture images may be accompanied by the transmission of corresponding depth maps. In this case, each depth map video is also subject to compression using, for example, H.264/AVC [28] with a given QP, spending an encoding bit-rate R_n^d and achieving a distortion D_n^d. Our problem now consists of minimizing overall rate of captured texture and depth views

$$\mathcal{R} = \sum_{n=0}^{N-1} (R_n^t + R_n^d) \tag{21}$$

subject to a total observed distortion

$$\mathcal{D} = \sum_{m=0}^{M-1} \Delta_m \tag{22}$$

where Δ_m is the observed view distortion, experienced by the viewers, resulting from the compression of both texture and depth maps used in view synthesis.

Our model of observed view distortion is proportional to both the distortions of the captured texture views and those of the captured depth views such that

$$\Delta_m = \sum_{j \in \Psi_m} \left(w_j \ D_j^t + \eta \ \omega_j \ D_j^d \right) \tag{23}$$

where η is the texture-to-depth ratio, which weighs the relative importance of texture distortion with respect to that of depth distortion towards synthesis, and ω_j are the captured depth distortion weights ($0 \leq \omega_j \leq 1$ and $\sum \omega_j = 1$).

Unlike texture distortion, we assume the influence of depth distortion of each camera upon the synthesis at $\vec{v}_m$ to be equal, regardless of distances between camera and $\vec{v}_m$, as discussed in Sec. II-B. Thus, ω_j may be expressed in terms of $||\Psi_m||$ (the number of cameras in Ψ_m) and auxiliary variable α_{mj}, while w_j is still a distance-based weight involving β_{mj}. The observed view distortion is then given by

$$\Delta_m = \sum_{n=0}^{N-1} \left(\alpha_{mn} \ \beta_{mn} \ D_n^t \ + \eta \ \alpha_{mn} \ \frac{1}{||\Psi_m||} \ D_n^d \right) \tag{24}$$

and the total observed distortion is

$$\mathcal{D} = \sum_{m=0}^{M-1} \left(\sum_{n=0}^{N-1} \left(\alpha_{mn} \ \beta_{mn} \ D_n^t \ + \eta \ \alpha_{mn} \ \frac{1}{||\Psi_m||} \ D_n^d \right) \right). \tag{25}$$

Using (14) and defining

$$\varphi_n = \eta \sum_{m=0}^{M-1} \frac{\alpha_{mn}}{||\Psi_m||} \ , \tag{26}$$

we have the total observed distortion

$$\mathcal{D} = \sum_{n=0}^{N-1} \left(\phi_n \ D_n^t + \varphi_n \ D_n^d \right) = \sum_{n=0}^{N-1} \tilde{\mathcal{D}}_n \tag{27}$$

as a function of adjusted distortion measures for each camera $\{\tilde{\mathcal{D}}_n\}$ which take into account all viewers as well as a appropriately weighted distortion contributions from both texture and depth cameras.

Optimal bit allocation is thus achieved by minimizing the cost function

$$\mathcal{J} = \mathcal{R} + \lambda\mathcal{D} = \sum_{n=0}^{N-1}(R_n^t + R_n^d + \lambda\tilde{\mathcal{D}}_n) \tag{28}$$

for a given Lagrangian multiplier λ responsible for the rate-distortion trade-off. Texture- and depth-dependent components of the cost function are treated independently and, at each camera compressor, bit allocation is found by adjusting QP for texture cameras in order to minimize

$$J_n^t = R_n^t + \lambda\phi_n D_n^t \tag{29}$$

and adjusting QP for depth cameras to minimize

$$J_n^d = R_n^d + \lambda\varphi_n D_n^d \ . \tag{30}$$

V. Experimental Results

We simulated the proposed optimization algorithms, comparing them to uniform rate allocation, for several examples. We have chosen publicly available data sets containing large numbers of cameras, thus affording greater freedom in viewpoint selection. Motivated by the broadcast nature of the envisioned application, we consider large numbers of viewers. We present here a few illustrative cases. Three different viewer attention distributions were used. In each one, viewer position is randomly chosen according to a given distribution and, for each viewer position, an appropriate set of cameras is selected for the desired view synthesis. Cameras, therefore, may be watched (or referenced) by different numbers of viewers. The camera usage statistics, for given instantiations of viewer distribution, are represented in the form of histograms.

Two distributions are based on the *Pantomime* multiview sequence (1280×960-pixels resolution) data set [32] with even numbered cameras in the range 20-58. This data set corresponds to a one-dimensional regularly-spaced camera arrangement. In the first case, we simulated a bimodal Gaussian distribution of 200 and 300 viewers centered at cameras 29 and 49 with standard deviations 5 and 4, respectively. Fig. 5 depicts the corresponding camera usage histogram. In the second case, we simulated a sharper Laplacian distribution of 400 viewpoints with mean 37 and standard deviation 3, as shown in Fig. 6.

The third distribution is based on the *Akko & Kayo* multiview sequence (640×480-pixels resolution) data set [33]. We selected the cameras originally labeled 27-29, 47-49 and 67-69 with accompanying depth maps to form a two-dimensional (2D) distribution. Each group of cameras is regularly distributed across one of three rows with 5 cm of horizontal and 20 cm of vertical spacing among them. A total of 400 viewpoints are randomly spread according to a Gaussian distribution centered at coordinates (3.75, 15) cm from the origin, set at camera 27, and standard deviation of (2, 7.5) cm in horizontal and vertical directions. Synthesis is allowed at arbitrary positions along lines between horizontally, vertically or diagonally adjacent cameras. An instantiation of viewer attention distribution among cameras is shown in Fig. 7.

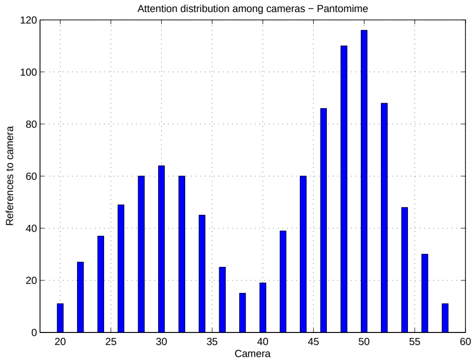


Fig. 5. Bimodal Gaussian distribution of viewer attention over 1D camera setup for *Pantomime*.

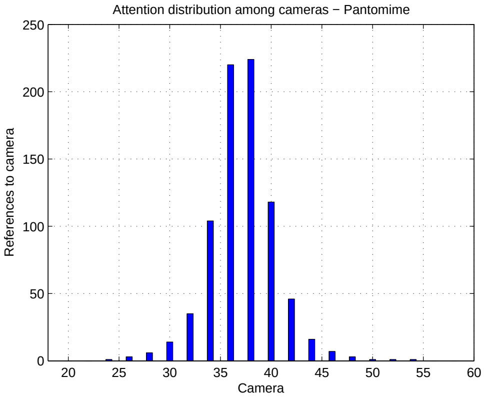


Fig. 6. Laplacian distribution of viewer attention over 1D camera setup for *Pantomime*.

In our proposed algorithms, each texture camera and each depth camera are compressed independently with H.264/AVC JM Reference Software v17.2 [34]. Free-viewpoint images are formed through DIBR with MPEG View Synthesis Reference Software v3.5 [35]. Distortion is measured in terms of the MSE between viewpoints synthesized from compressed and uncompressed adjacent camera views. Total observed distortion is taken as the MSE across all M viewpoints and reported in terms of PSNR. Overall bit-rate considers the sum of all N texture camera rates and, when encoded, all N depth camera rates as well. Bit-rate is reported in terms of bits per pixel per camera (bpc) for the first frame of each view. Comparisons of our proposals are drawn against the *Uniform Allocation* in which all cameras are subject to the same distortion (or same bit-rate) and employ equal QPs. In this case, texture QPs are selected from the range $\{17, 22, 27, 32, 37\}$ and depth QPs, when encoded, from the range $\{2, 3, 4, 5, 6\}$. The depth QP range was chosen to secure proportionality between depth distortions and observed view distortion for the tested

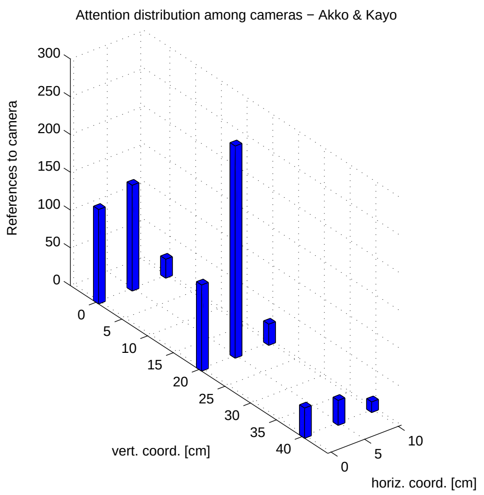


Fig. 7. Gaussian distribution of viewer attention over 2D camera array for *Akko & Kayo*.

TABLE I
Average PSNR gains and bit-rate savings [37] of *Texture Model* relative to *Uniform Allocation* for various viewer distributions.

Distribution	**Avg. PSNR gain**	**Avg. rate savings**
Bimodal	0.48 dB	9.2 %
Laplacian	2.41 dB	39.3 %
2D Gaussian	1.54 dB	24.5 %

sequences. Note that for data sets with low quality depth maps, large depth QP ranges may include particular quantization levels which can violate our proportionality assumption. In specific cases, certain larger QP value (such as suggested in [36]) were observed to improve synthesis distortion with respect to lower QP values by contributing to depth noise removal.

We first examine the attention-weighted rate allocation for texture, termed the *Texture Model*, presented in Sec. III. In it, depth maps are assumed available at the decoder. The maps are applied in uncompressed form towards DIBR and are not considered in overall bit-rate calculations. Results are summarized in Table I. With the bimodal Gaussian instantiation, the *Texture Model* achieves average PSNR gain of 0.48 dB and bit-rate savings of 9.2% relative to the *Uniform Allocation* case, as shown in Fig. 8. Of all tested viewer distributions, the sharper Laplacian instantiation produces the largest average PSNR gains and bit-rate savings, 2.41 dB and 39.3%, respectively, as pictured in Fig. 9. For the 2D distribution case, overall rate and total observed distortions are presented in Fig. 10. The *Texture Model* achieves an average PSNR gain of 1.54 dB and bit-rate savings of 24.5% over *Uniform Allocation*.

We then carried out tests for rate allocation of both texture and depth, referred to as *Texture plus Depth Model*, developed in Sec. IV. In these comparisons, both the *Uniform Allocation* and the *Texture Model* employ uniformly coded depth maps (same QP for all cameras).

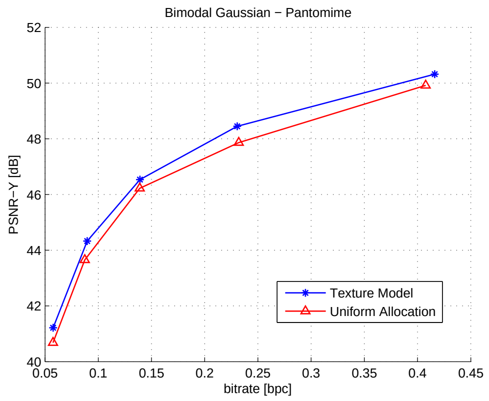


Fig. 8. Overall R-D performance comparison between *Texture Model* and *Uniform Allocation* for *Pantomime* with bimodal Gaussian viewer distribution.

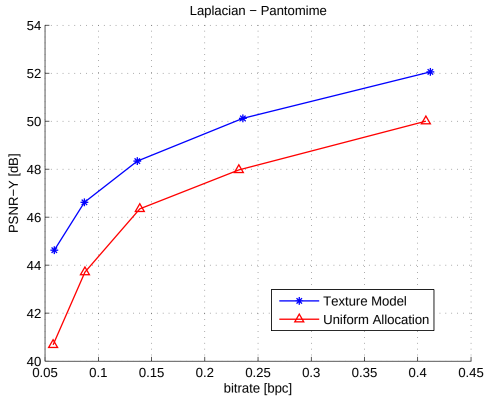


Fig. 9. Overall R-D performance comparison between *Texture Model* and *Uniform Allocation* for *Pantomime* with Laplacian viewer distribution.

Depth map bit-rate is considered as part of the overall bit-rate. For the distributions using the *Pantomime* data set we used $\eta = 1000$, while for *Akko & Kayo* we used $\eta = 200$. Each value was determined from analysis of a synthesis distortion surface for a single view, as described in Sec. II-C. Results for the bimodal Gaussian distribution are shown in Fig. 11. The *Texture plus Depth Model* achieves significant gains over *Uniform Allocation* as well as the *Texture Model*, yielding average 24.7% bit-rate savings and gains of 2.11 dB over the *Uniform* case, as summarized in Tables II and III. Note that the *Texture Model* is similar to the allocations proposed in [6] and [7] where depth cameras employ a uniform QP. The Laplacian distribution yields the largest gains of the proposed approach and the results are shown in Fig. 12. Average PSNR gains of 3.18 dB and 28.7% bit-rate savings were achieved for this distribution. More modest gains for the *Texture plus Depth Model* over both *Uniform Allocation* and *Texture Model* for the 2D distribution can be observed in Fig. 13. This may be attributable to a flatter viewer distribution and the large variability of η in a camera setup where vertical baselines are significantly

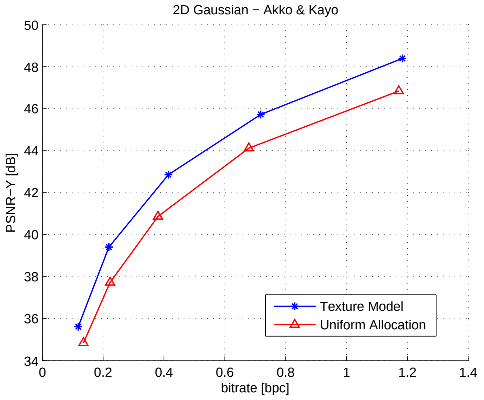


Fig. 10. Overall R-D performance comparison between *Texture Model* and *Uniform Allocation* for *Akko & Kayo* with Gaussian viewer distribution.

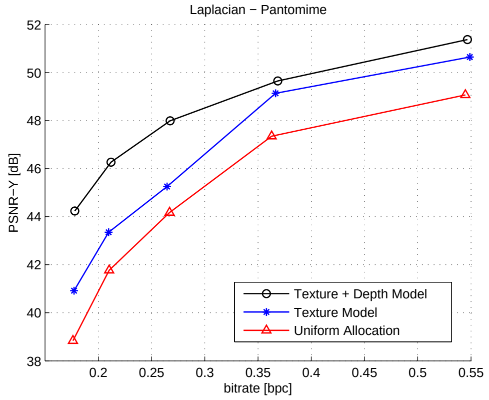


Fig. 12. Overall R-D performance comparison between *Texture plus Depth Model*, *Texture Model* and *Uniform Allocation* for *Pantomime* with Laplacian viewer distribution.

larger than horizontal ones.

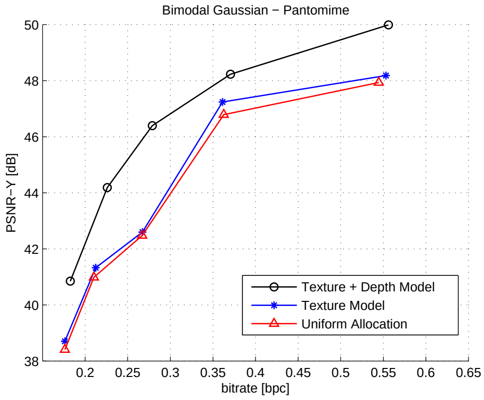


Fig. 11. Overall R-D performance comparison between *Texture plus Depth Model*, *Texture Model* and *Uniform Allocation* for *Pantomime* with bimodal Gaussian viewer distribution.

TABLE II

Average PSNR gains of *Texture Model* and *Texture plus Depth Model* relative to *Uniform Allocation*, using compressed depth maps, for various viewer distributions.

Distribution	Texture Model	Texture + Depth
Bimodal	0.33 dB	2.11 dB
Laplacian	1.58 dB	3.18 dB
2D Gaussian	1.16 dB	1.64 dB

Visual results are shown in Fig. 14 for *Pantomime* under the Laplacian distribution. Detail crops of synthesized images from the central portion (view 37) of the attention profile, see Fig. 6, are presented along with those from the periphery (view 27) which receive less attention. The largest subjective quality differences may be observed between views 37 and 27, in Figs. 14 (d) and (c), respectively, synthesized from attention-weighted allocation, under the *Texture plus Depth Model*. Quality differences among synthesized images resulting from *Uniform Allocation*, in Figs. 14 (a) and (b), are less noticeable. For views that are more watched, attention-weighted allocation produces synthesis results of superior quality when compared to the *Uniform Allocation*, as seen in Figs. 14 (d) and (b). Whereas for views receiving less attention, attention-weighted allocation is inferior to *Uniform Allocation*, as seen in Figs. 14 (c) and (a).

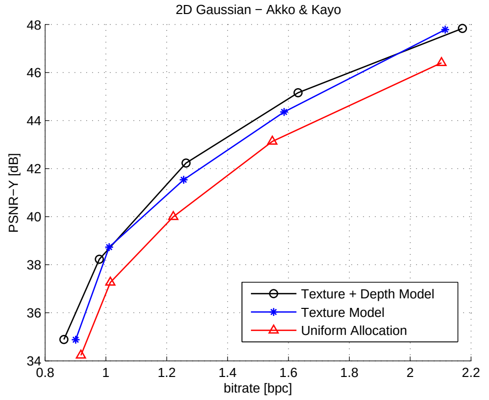


Fig. 13. Overall R-D performance comparison between *Texture plus Depth Model*, *Texture Model* and *Uniform Allocation* for *Akko & Kayo* with Gaussian viewer distribution.

Following, we compare system performance, under varying viewer distribution concentrations, against 3D-AVC [12]. 3D-AVC jointly codes multiview texture and depth streams, exploring depth-based coding tools, to achieve reported bit-rate savings of up to 35% over MVC [38]. In applications with large numbers of cameras, however, deployment of 3D-AVC will require a single coder capable of centralizing the multiple camera streams. Furthermore, due to inter-view dependencies, all streams must be made available at decoding, regardless of viewer attention. We use the 3DV-ATM Reference Software v14.0 [39]. For each row of our *Akko & Kayo* set up, we apply 3D-AVC across the 3 cameras of texture and depth, as recommended in [40]. Results are depicted in Fig. 15. 3D-AVC presents average bit-rate savings of 45% with respect to our proposed rate allocation using

TABLE III
AVERAGE BIT-RATE SAVINGS OF *Texture Model* AND *Texture plus Depth Model* RELATIVE TO *Uniform Allocation*, USING COMPRESSED DEPTH MAPS, FOR VARIOUS VIEWER DISTRIBUTIONS.

Distribution	Texture Model	Texture + Depth
Bimodal	4.9 %	24.7 %
Laplacian	19.2 %	28.7 %
2D Gaussian	7.9 %	10.5 %

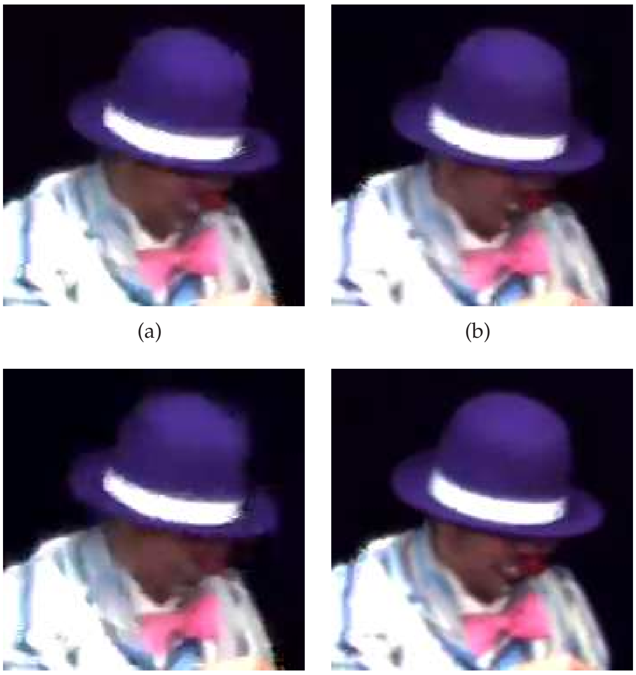


Fig. 14. Visual comparison of detail crops for *Pantomime* under Laplacian attention distribution (see Fig. 6). Synthesis results for (a) view 27 and (b) view 37 from a *Uniform Allocation* and (c) view 27 and (d) view 37 from attention-weighted compression with *Texture plus Depth Model*. Overall bit-rate is 0.17 bpc.

the *Texture plus Depth Model* under a uniform viewer attention distribution (labeled σ = infinity). As viewer attention is concentrated under Gaussian distributions with decreasing standard deviations ($\sigma = 2.5$, 1.0 and 0.5 cm, in both horizontal and vertical directions), significant gains in total observed distortion (PSNR) and overall bit-rate are achieved by our proposal, surpassing the 3D-AVC performance.

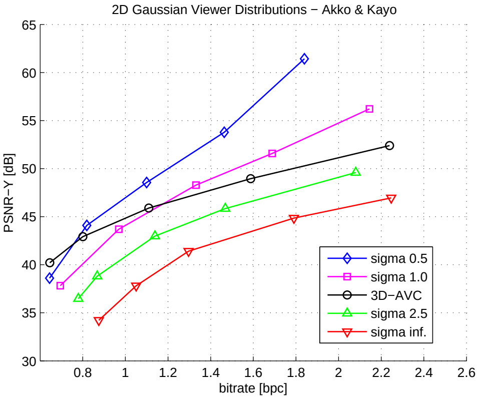


Fig. 15. Overall R-D performance comparison between *Texture plus Depth Model* for *Akko & Kayo* under Gaussian viewer distributions with varying standard deviations, uniform viewer distribution (σ = infinity) and 3D-AVC [12].

VI. CONCLUSIONS

We proposed an attention-weighted rate-allocation technique for texture images and depth maps to minimize the total observed distortion within a FTV broadcast system. We model the effects of both texture and depth camera distortions upon observed view distortion as well as their relative impact. A joint optimization framework is presented for attention-weighted rate allocation among both texture and depth cameras. Results show significant gains of attention-weighted rate allocation of texture relative to uniform allocation. The joint optimized allocation of texture and depth was shown to outperform both the uniform and the texture-only rate allocation. For concentrated viewer attention distributions, our proposal can also outperform 3D-AVC coding of the multiple texture and depth streams.

Future work may focus on many other aspects of FTV, such as attention-weighted networking and transcoding among sub-networks. For applications with large numbers of cameras, 3D-AVC or other multiview coding extensions may be incorporated and applied across camera sub-sets, each of which attention-weighted in response to viewer demands. Another aspect of further study is system adaptation to dynamic scenarios with changing scene contents. Methods for preventing, or attenuating, possible image quality reduction experienced by viewers as camera rate allocation varies asymmetrically over time may be investigated. Of particular interest are the estimation and update policies for the texture-to-depth ratio, through the efficient monitoring of distortion surfaces, such as Fig.4(a), in response to changes in scene contents.

REFERENCES

[1] M. Tanimoto, M. P. Tehrani, T. Fujii, and T. Yendo, "Free-viewpoint TV," *IEEE Signal Processing Magazine*, vol. 28, no. 1, January 2011.

[2] C. L. Zitnick, S. B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski, "High-quality video view interpolation using a layered representation," in *ACM SIGGRAPH'04*, 2004.

[3] C. Fehn, "Depth-image-based rendering (DIBR), compression, and transmission for a new appr0ach on 3D-TV," *Proc. SPIE 5291, Stereoscopic Displays and Virtual Reality Systems XI*, vol. 93, May 2004.

[4] P. Kauff, N. Atzpadin, C. Fehn, M. Muller, O. Scheer, A. Smolic, and R. Tanger, "Depth map creation and image-based rendering for advanced 3DTV services providing interoperability and scalability," *Signal Processing: Image Communication*, vol. 22, no. 2, February 2007.

[5] A. Fiandrotti, J. Chakareski, and P. Frossard, "Popularity-aware rate allocation in multiview video," in *Visual Communications and Image Processing*, 2010.

[6] T. Scandaroli, R. L. de Queiroz, and D. Florencio, "Attention-weighted rate allocation in free-viewpoint television," *IEEE Signal Processing Letters*, vol. 20, no. 4, April 2013.

[7] C. Dorea and R. L. de Queiroz, “General rate-allocation in free-viewpoint television,” in *IEEE Int. Conf. on Image Processing*, Paris, France, October 2014.

[8] A. Vetro, S. Yea, and A. Smolic, “Towards a 3D video format for autostereoscopic dispalys,” in *Applications of Digital Image Processing XXXI, Proceedings of the SPIE*, 2008, vol. 7073.

[9] Y. Morvan, D. Farin, and P. H. N. de With, “Depth-image compression based on R-D optimized quadtree decomposition for the transmission of multiview images,” in *IEEE Int. Conf. on Image Processing*, September 2007.

[10] K. Y. Kim, G. H. Park, and D. Y. Suh, “Bit-plane-based lossless depth-map coding,” *Optical Engineering*, vol. 49, no. 6, 2010.

[11] *ITU-T Recommendation and International Standard of Joint Video Specification*, ITU-T Rec H.264/ISO/IEC 14496-10 AVC, January 2012.

[12] M. M. Hannuksela, D. Rusanovsky, W. Su, L. Chen, R. Li, P. Aflaki, D. Lan, M. Joachimiak, H. Li, and M. Gabbouj, “Multiview-video-plus-depth coding based on the advanced video coding standard,” *IEEE Transactions on Image Processing*, vol. 22, no. 9, 2013.

[13] *High efficiency video coding*, ITU-T Recommendation H.265, October 2014.

[14] Y. Liu, Q. Huang, S. Ma, D. Zhao, and W. Gao, “Joint video/depth rate allocation for 3D video coding based on view synthesis distortion model,” *Signal Processing: Image Communication*, vol. 24, no. 8, 2009.

[15] Y. Liu, Q. Huang, S. Ma, D. Zhao, W. Gao, S. Ci, and H. Tang, “A novel rate control technique for multiview video plus depth based 3D video coding,” *IEEE Transactions on Broadcasting*, vol. 57, no. 2, 2011.

[16] E. Bosc, V. Jantet, M. Pressigout, L. Morin, and C. Guillemot, “Bit-rate allocation for multi-view video plus depth,” in *3DTV Conference: The True Vision-Capture, Transmission and Display of 3D Video (3DTV-CON)*, 2011.

[17] N. Ozbek, A. M. Tekalp, and E. T. Tunali, “Rate allocation between views in scalable stereo video coding using an objective stereo video quality measure,” in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2007.

[18] E. Kurutepe, M. R. Civanlar, and A. M. Tekalp, “Client-driven selective streaming of multiview video for interactive 3DTV,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 11, November 2007.

[19] C. Zhang and D. Florencio, “Joint tracking and multiview video compression,” in *Proceedings of the SPIE*, July 2010, vol. 7744.

[20] J. Chakareski, V. Velisavljevic, and V. Stankovic, “User-action-driven view and rate scalable multiview video coding,” *IEEE Transactions on Image Processing*, vol. 22, no. 9, September 2013.

[21] G. Cheung, V. Velisavljevic, and A. Ortega, “On dependent bit allocation for multiview image coding with depth-image-based rendering,” *IEEE Transactions on Image Processing*, vol. 20, no. 11, November 2011.

[22] G. Saygili, C. G. Gurler, and A. M. Tekalp, “Evaluation of asymmetric stereo video coding and rate scaling for adaptive 3D video streaming,” *IEEE Transactions on Broadcasting*, vol. 57, no. 2, June 2011.

[23] P. Aflaki, M. M. Hannuksela, and M. Gabbouj, “Subjective quality assessment of asymmetric stereo-scopic 3D video,” *Signal, Image and Video Processing*, vol. 9, no. 2, February 2015.

[24] J. Chakareski, V. Velisavljevic, and V. Stankovic, “View-popularity-driven joint source and channel coding of view and rate scalable multi-view video,” *IEEE Journal of Selected Topics in Signal Processing*, April 2015.

[25] D. Tian, P.-L. Lai, P. Lopez, and C. Gomila, “View synthesis techniques for 3D video,” in *Applications of Digital Image Processing XXXII, Proceedings of the SPIE*, 2009, vol. 7443 (2009).

[26] M. Domanski, T. Grajek, K. Klimaszewski, M. Kurc, O. Stankiewicz, J. Stankowski, and K. Wegner, “Poznan multiview video test sequences and camera parameters,” in *ISO/IEC JTC1/SC29/WG11 MPEG 2009/M17050*, Xian, China, October 2009.

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

[28] *ITU-T Recommendation and International Standard of Joint Video Specification*, ITU-T Rec H.264/ISO/IEC 14496-10 AVC, March 2005.

[29] Y. Shoham and A. Gersho, “Efficient bit allocation for an arbitrary set of quantizers,” *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 36, no. 9, September 1988.

[30] G. J. Sullivan and T. Wiegand, “Rate-distortion optimization for video compression,” *IEEE Signal Processing Magazine*, vol. 15, no. 6, November 1998.

[31] J. Chakareski, “Transmission policy selection for multi-view content delivery over bandwidth constrained channels,” *IEEE Transactions on Image Processing*, vol. 23, no. 2, February 2014.

[32] M. Tanimoto, M. Fujii, T. Senoh, T. Aoki, and Y. Sugihara, “Test sequences with different camera arrangements for call for proposals on multiview video coding,” in *ISO/IEC JTC1/SC29/WG11 MPEG 2005/M12338*, Poznan, Poland, July 2005.

[33] M. Tanimoto, M. Fujii, and K. Fukushiuma, “1D parallel test sequences for MPEG-FTV,” in *ISO/IEC JTC1/SC29/WG11 MPEG 2008/M15378*, Archamps, France, April 2008.

[34] “JM H.264 reference software v17.2,” in *http://iphome.hhi.de/suehring/tml/*.

[35] M. Tanimoto, M. Fujii, T. Suzuki, K. Fukushima, and N. Mori, “Reference softwares for depth estimation and view synthesis,” in *ISO/IEC JTC1/SC29/WG11 MPEG 2008/M15377*, Archamps, France, April 2008.

[36] A. Rusanovsky, K. Muller, and A. Vetro, “Common test conditions of 3DV core experiments,” in *ITU-T SG16/WP36 ISO/IEC JTC1/SC29/WG11 JCT2-A1100*, Stockholm, Sweden, July 2012.

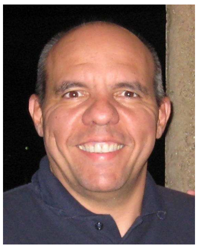
[37] G. Bjontegaard, “Calculation of average PSNR differences between RD-curves,” in *ITU-T SG16/Q6 VCEG-M33*, Austin, TX, USA, March 2001.

[38] A. Vetro, T. Wiegand, and G. J. Sullivan, “Overview of the stereo and multiview video coding extensions of the H.264/MPEG-4 AVC standard,” *Proceedings of the IEEE*, vol. 99, no. 4, 2011.

[39] “3DV-ATM reference software v14.0,” in *http://mpeg3dv.nokiaresearch.com/*.

[40] “Call for proposals on 3D video coding technology,” in *doc. N12036.doc, MPEG*, March 2011.

Camilo Dorea received the B.S. degree from the University of Brasilia, Brazil, in 1997, the M.S. degree from the University of Maryland at College Park, USA, in 1999, both in electrical engineering, and the Ph.D. degree in telecommunications from the Technical University of Catalonia (UPC), Barcelona, Spain, in 2007. From 2007 to 2008 he was with Thomson Corporate Research at Princeton, NJ, USA. In 2009 he joined the Department of Computer Science at the University of Brasilia where he is currently Assistant Professor. His research interests include video segmentation and analysis, video coding, and multiview and 3D processing.



Ricardo L. de Queiroz received the Engineer degree from Universidade de Brasilia , Brazil, in 1987, the M.Sc. degree from Universidade Estadual de Campinas, Brazil, in 1990, and the Ph.D. degree from the University of Texas at Arlington , in 1994, all in Electrical Engineering.

In 1990–1991, he was with the DSP research group at Universidade de Brasilia, as a research associate. He joined Xerox Corp. in 1994, where he was a member of the research staff until 2002. In 2000–2001 he was also an Adjunct Faculty at the Rochester Institute of Technology. He joined the Electrical Engineering Department at Universidade de Brasilia in 2003. In 2010, he became a Full Professor at the Computer Science Department at Universidade de Brasilia.

Dr. de Queiroz has published over 150 articles in journals and conferences and contributed chapters to books as well. He also holds 46 issued patents. He is an elected member of the IEEE Signal Processing Society's Multimedia Signal Processing (MMSP) Technical Committee and a former member of the Image, Video and Multidimensional Signal Processing (IVMSP) Technical Committee. He is a past editor for the EURASIP Journal on Image and Video Processing, IEEE Signal Processing Letters, IEEE Transactions on Image Processing, and IEEE Transactions on Circuits and Systems for Video Technology. He has been appointed an IEEE Signal Processing Society Distinguished Lecturer for the 2011–2012 term.

Dr. de Queiroz has been actively involved with the Rochester chapter of the IEEE Signal Processing Society, where he served as Chair and organized the Western New York Image Processing Workshop since its inception until 2001. He is now helping organizing IEEE SPS Chapters in Brazil and just founded the Brasilia IEEE SPS Chapter. He was the General Chair of ISCAS'2011, and MMSP'2009, and is the General Chair of SBrT'2012. He was also part of the organizing committee of ICIP'2002. His research interests include image and video compression, multirate signal processing, and color imaging. Dr. de Queiroz is a Senior Member of IEEE, a member of the Brazilian Telecommunications Society and of the Brazilian Society of Television Engineers. Telecommunications Society and of the Brazilian Society of Television Engineers.