# Attention-Weighted Rate Allocation in Free-Viewpoint Television

Thacio Scandarolli, *Student Member, IEEE*, Ricardo L. de Queiroz, *Senior Member, IEEE*, and Dinei A. Florencio, *Senior Member, IEEE*

***Abstract*—An architecture for free-viewpoint broadcast television transmission is proposed where all the views are transmitted at potentially different qualities and watched by a large number of viewers. The quality (or bit-rate) of each view is controlled by the distribution of viewpoints chosen by the viewers. For example, if most viewers are watching synthetic views in between views $n$ and $n+1$, those views are allocated more transmission bits than views that are scarcely watched. We developed an attention-weighted bit-rate-allocation method that is optimal in the total observer distortion sense. The optimality of the method relies on knowing the viewpoint probability distribution at every moment. Simulation results show that overall transmission rate can be reduced for the same total observed distortion.**

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

## I. Introduction

FREE-VIEWPOINT VIDEO (FVV) entails a transmission of video wherein the decoder has the freedom to chose from which viewpoint to observe the represented scene [1]–[4]. We are here concerned with free-viewpoint television (FVTV) where there is one transmitter broadcasting video and we would like each of the many receivers to be able to choose its own viewpoint to watch the video. It is not feasible, however, to directly acquire and transmit a continuum of views around the scene. Thus, the generally adopted solution involves capturing the scene from a (possible large but finite) number of cameras, and estimating any desired viewpoint in between cameras, a process normally referred to as *view synthesis*. Although less computationally intensive methods for view synthesis do exist [5], high quality synthesis requires computationally intensive operations [6]. In particular, high quality view synthesis requires knowledge of scene depth or range. Even with the recent progress in *time of flight* and *projected pattern* depth cameras, sensor technology is not yet mature to provide relatively inexpensive, reliable and safe range measuring devices to use in outdoor activities. Hence, depth estimation usually relies on visual information from the cameras, matching regions from one view to another, estimating camera positions, etc. Of course, whenever depth signals are directly acquired, this is used to refine and denoise the depth estimates obtained by depth sensors. All those operations demand intense computation, making view synthesis and depth estimation a major computation bottleneck in the system. Thus, although other solutions for view synthesis exist, and would work similarly well with the proposed method, in our simulation we assume depth-based view synthesis is used, and that depth maps are transmitted along with the camera views. Since there are many more viewers than cameras, in order to simplify view synthesis at each decoder, it is reasonable to move to the encoder as much as possible of the repeated operations (i.e., those operations that need to be performed by every decoder). We greatly simplify the view synthesis procedure at the decoder by deriving the depth maps at the encoder side and transmitting them along with the video. In effect, we exchange decoder complexity, in numerous receivers, for encoder complexity at a few cameras and bandwidth to transmit the depth maps.

An additional bottleneck that influences the overall system design is the bandwidth requirements. Since each viewer has the freedom to chose its arbitrary viewpoint, at the edge of the network, as many different views as existing viewers will be required. If we were to transmit all camera and depth signals to all receivers, we would multiply even further the required bandwidth. Instead, we make use of a receiver agent.

Our approach to FVTV is introduced in Fig. 1. Color imagery is captured, which, along with their estimated depth maps, are streamed to the many decoders. View synthesis for each receiver and potentially depth map estimation may take place within the network. An encoder agent may collect the multiview video (from all cameras), estimate depth maps and stream all data in multicast. Each receiver agent may be responsible to carry the desired view synthesis delivering a single-video stream with the proper view to the display device, which can very well be a low-computation device such as a tablet. In cases where stereo video is desired, the receiver agent would synthesize both right and left views. By broadcasting the depth maps, we avoid the need to recompute depth maps many times. By placing a receiver agent as close as possible to the display device, we save computation at that device, and reduce bandwidth requirements and response time when a user changes his or her viewpoint.

We want to improve the video transmission in this scenario by allocating more bits to viewpoints which receive more attention. Note that we target the case of actively chosen viewpoint, unlike the approaches in [7], [8], which are targeted at motion parallax based viewpoint [9], [10]. As such, we assume a feedback

Manuscript received December 20, 2012; accepted January 28, 2013. Date of publication February 12, 2013; date of current version February 26, 2013. This work was supported by CNPq. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Shantanu D. Rane.

T. Scandarolli was with Universidade de Brasilia. He is now with the University of Southern California, Los Angeles, CA 90089 USA (e-mail: thacio@image.unb.br).

R. L. de Queiroz is with the Computer Science Department, Universidade de Brasilia, Brasilia, DF, Brazil (e-mail: queiroz@ieee.org).

D. A. Florencio is with Microsoft Research, Redmond, WA 98052-7329 USA (e-mail: dinei@microsoft.com).

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

Digital Object Identifier 10.1109/LSP.2013.2246863

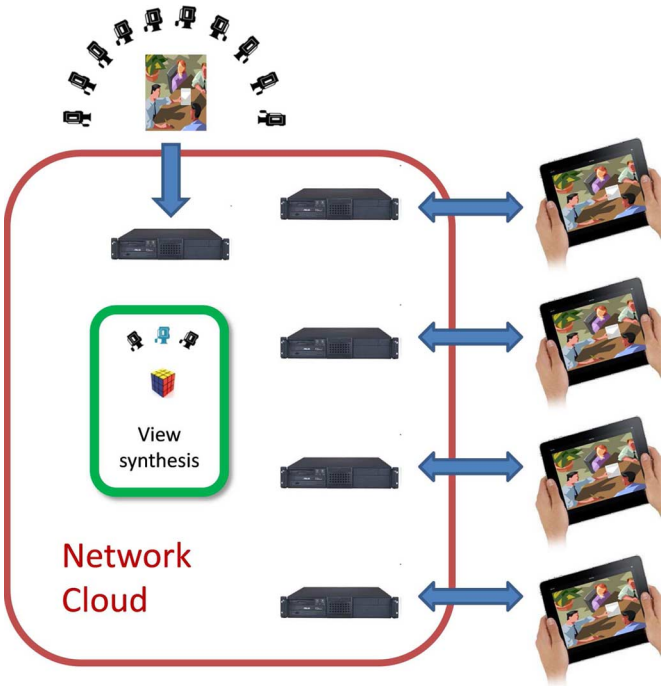


Fig. 1. General architecture for broadcast FVTV using cloud services. The demanding operations of view synthesis and depth estimation can be carried somewhere within the network. For that we add encoding and decoding cloud agents to do the hard work, allowing for simple FVTV displays.

channel, albeit slow, which allows for the encoder to precisely know where the viewpoints are. We then develop an optimized bit allocation scheme. If no feedback channel is available, the FVTV system would then resort to a default value based on expected distribution, or even to uniform bit-rate (or distortion) allocation for all cameras.

## II. Attention-Weighted Rate Allocation

Assume a linear arrangement of $N$ cameras. The path of all viewpoints traverses all cameras, which are sequentially numbered. The viewpoint $p$ of the $n$-th camera is at $p = n$. An active viewpoint is a viewpoint being watched (potentially synthesized) by a viewer. There are $M$ viewers, each watching at a viewpoint $v_i, 1 \leq v_i \leq N$. Viewpoint positions are linearly distributed in between camera viewpoints. If the view synthesis is for a position $x\%$ of the way from camera $n$ to $n+1$, then the active viewpoint is at $p = n + x/100$. We also assume $M \gg N$ and that there is a feedback channel, perhaps much slower than the forward channel, in which each decoder (or its network agent) can inform the encoder about its current viewpoint. Not only we assume viewpoints lying on a line segment between two cameras, but we also assume a one-dimensional arrangement in a sense that these two cameras are indeed the closest ones to any viewpoint in between them. Hence, view synthesis in any segment only includes the views from the two nearest cameras, i.e., the synthesized view $v_i$ depends only on camera views $\alpha_i = \text{floor}(v_i)$ and $\alpha_i + 1$, so that $\alpha_i \leq v_i < \alpha_i + 1$. Note that this is only a simplification, and not a constraint of the proposed method. More specifically, we chose this linear arrangement because it is representative of many practical applications, and serves to show the results in a simplified and direct form. View synthesis from other points is possible, but quality typically degrades quickly, as occluded regions cannot be properly predicted. Note, however, that the proposed methodology could be used even on these situations, by properly accounting for the participation of each camera in generating the corresponding viewpoints.

The $k$-th view is encoded using say H.264/AVC [11] with quantizer parameter $QP_k$ yielding a bit-rate $\rho_k$ and distortion $\delta_k$. While total rate for transmitting all camera views is

$$R = \sum_{i=1}^{N} \rho_i, \tag{1}$$

we do not want to assume the total distortion $D$ as $\sum_{i=1}^{N} \delta_i$ because these views are, ultimately, just an intermediate representation of the data, and some views will be used more often than others in synthesizing the requested viewpoints. Thus, we define the overall distortion as the total observed distortion (TOD) which is the sum of the distortion $D_i$ observed by each viewer, i.e.,

$$D = \sum_{i=1}^{M} D_i, \tag{2}$$

Although the exact pixels used from each view depend on the depth information, the synthesized view is ultimately, a linear combination of pixels from the two neighboring cameras. Thus, we model the distortion on a viewpoint as proportional to the distance to the camera view. Let $\beta_i = v_i - \alpha_i$, then

$$D_k = (1 - \beta_k)\delta_{\alpha_k} + \beta_k \delta_{\alpha_k + 1}, \tag{3}$$

so that

$$D = \sum_{k=1}^{M} (1 - \beta_k)\delta_{\alpha_k} + \beta_k \delta_{\alpha_k + 1}. \tag{4}$$

Assume the set of active viewpoints to be sorted in non-decreasing order, i.e., $v_{i+1} \geq v_i$. We can then, break the set of $\{v_i\}$ into $N-1$ groups corresponding to each segment. Let $\Omega_n$ be the set of all indexes of the active viewpoints in the $n$-th segment, i.e., $\{v_i | \alpha_n \leq v_i < \alpha_n + 1\}$. Defining the breakpoint index for each segment as $t_n$ then

$$\Omega_n = \{t_{n-1} + 1, t_{n-1} + 2, \ldots, t_n\} \tag{5}$$

i.e., if $t_{n-1} < i \leq t_n$ then $\alpha_n \leq v_i < \alpha_n + 1$ for $n = 1, \ldots, N-1$ and $i = 1, \ldots, M$. Note that $t_0 = 0$ and the first element of $\Omega_1$ should be 1 if there is any active viewpoint in the segment in between cameras 1 and 2. If there are active viewpoints on the last camera ($v_i = N$), we can define $\Omega_N = \{t_{N-1} + 1, \ldots, M\}$ where there are as many entries in $\Omega_N$ as viewers watching (active viewpoints) the $N$-th camera. Hence,

$$\begin{aligned} D &= \sum_{n=1}^{N} \sum_{k \in \Omega_n} D_n \\ &= \sum_{n=1}^{N-1} \sum_{k \in \Omega_n} [(1 - \beta_k)\delta_n + \beta_k \delta_{n+1}] + \sum_{k \in \Omega_N} \delta_N \\ &= \sum_{k \in \Omega_1} (1 - \beta_k)\delta_1 + \sum_{n=2}^{N-1} \sum_{k \in \Omega_n} (1 - \beta_k)\delta_n \\ &\quad + \sum_{n=2}^{N-1} \sum_{k \in \Omega_{n-1}} \beta_k \delta_n + \sum_{k \in \Omega_{N-1}} \beta_k \delta_N + \sum_{k \in \Omega_N} \delta_N. \end{aligned} \tag{6}$$

Let

$$\gamma_1 = \sum_{k\in\Omega_1}(1-\beta_k), \tag{7}$$

$$\gamma_n = \sum_{k\in\Omega_n}(1-\beta_k) + \sum_{k\in\Omega_{n-1}}\beta_k \tag{8}$$

for $n = 2,\ldots,N-1$, and

$$\gamma_N = \sum_{k\in\Omega_{N-1}}\beta_k + \sum_{k\in\Omega_N} 1. \tag{9}$$

Then,

$$D = \sum_{n=1}^{N}\delta_n\gamma_n = \sum_{n=1}^{N}\delta'_n \tag{10}$$

is the TOD expression, where the $\gamma_n$ are the attention weights for each given view/camera. Furthermore, if $\omega_n$ is the number of elements in $\Omega_n$ and if $s_n = \sum_{k\in\Omega_n}\beta_k$ then

$$\gamma_1 = \omega_1 - s_1 \tag{11}$$

$$\gamma_n = \omega_n - s_n + s_{n-1} \tag{12}$$

$$\gamma_N = \omega_N + s_{N-1}. \tag{13}$$

Since $R = \sum_{i=1}^{N}\rho_i$ and $D = \sum_{i=1}^{N}\delta'_i$, for each camera view we have to allocate $\rho_i$ and $\delta'_i = \delta_i\gamma_i$. With those linear relations, we know that optimal allocation occurs when we operate at the point that minimizes $R + \lambda D$ for all nodes, i.e., for each view we seek the $QP$ that minimizes

$$J_i = \rho_i + \lambda\delta_i\gamma_i. \tag{14}$$

As long as we keep the same $\lambda$ for all views we operate at a globally optimal point in a TOD sense. Obviously, $\lambda$ controls the rate vs. quality trade-off.

## III. Simulation Results

In order to evaluate the technique we simulated a FVTV system with $M = 400$ viewers and $N = 10$ cameras. All 10 camera views are transmitted along with their respective depth maps encoded with H.264/AVC. We carried tests using views 35 through 53, in steps of two, of popular multiview sequences Pantomime and Champagne. For depth estimation and view synthesis we used ISO/IEC reference software [12].

The 400 active viewpoints were randomly spread according to a Gaussian distribution centered at the middle view ($v_i = 5.5$, i.e., in between cameras 5 and 6) with a 2.2 standard deviation. A instantiation of a viewpoint distribution within such a statistical model is shown in Fig. 2. Distortion $\{D_i\}$ for each active viewpoint $v_i$ was computed between the synthetic-view versions with and without compressing the camera views at $\alpha_i$ and $\alpha_i + 1$. In one test, we used the same quantizer parameter ($QP$) for all camera views ($QP_k = QP'$) and another for the depth maps ($QP_{\text{depth}}$). This is the "uniform $QP$" test and is the trivial way of encoding all the views. For the RD-optimized attention-weighted allocation, with the viewer distribution one can calculate the weights $\gamma_i$. Adjusting $\lambda$ from $-1$ to $-29$ one can then calculate optimized $QP$s for each of the camera views. All depth maps were compressed with H.264/AVC using the same $QP$. RD-curves comparing the uniform and the attention-weighted allocation are shown in Fig. 3 for sequence Champagne and in Fig. 4 for sequence Pantomime.

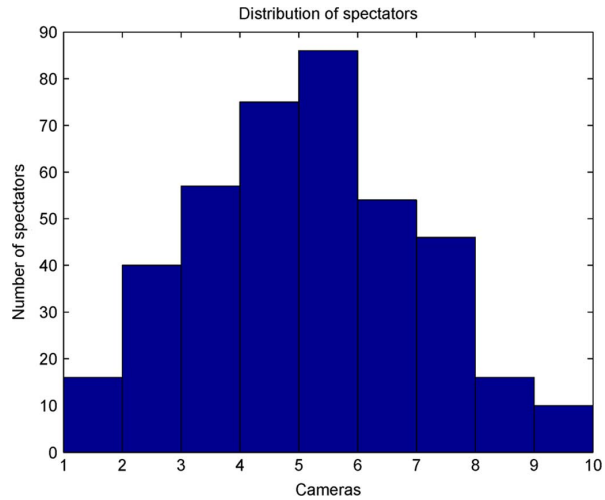


Fig. 2. Viewpoint distribution for the 10 camera multiview system used for tests.

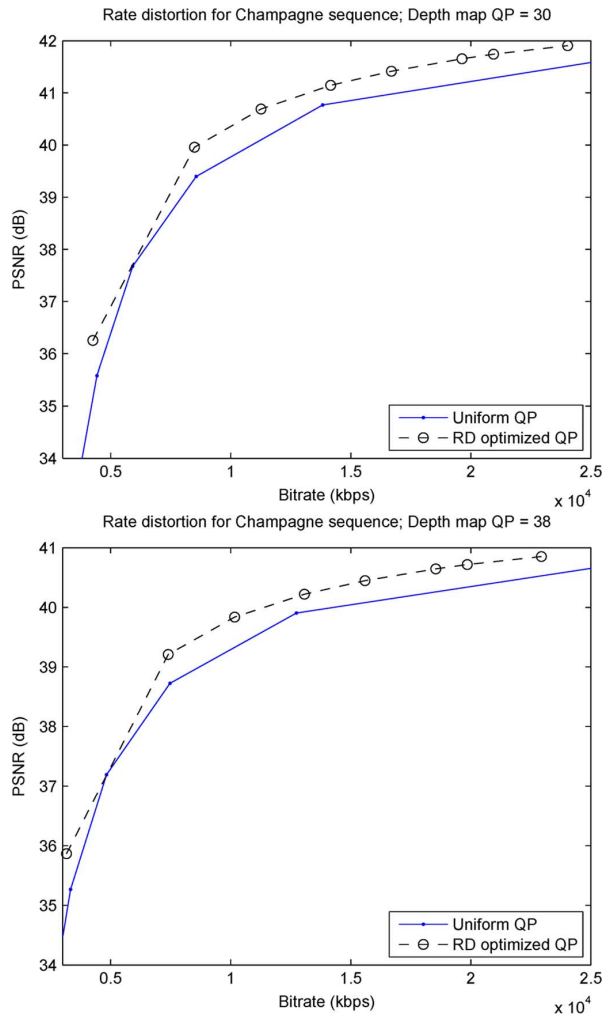


Fig. 3. Rate-distortion plot for compressing $N = 10$ views of sequence Champagne in a FVTV system being watched by $M = 400$ viewers, following the viewer distribution in Fig. 2.

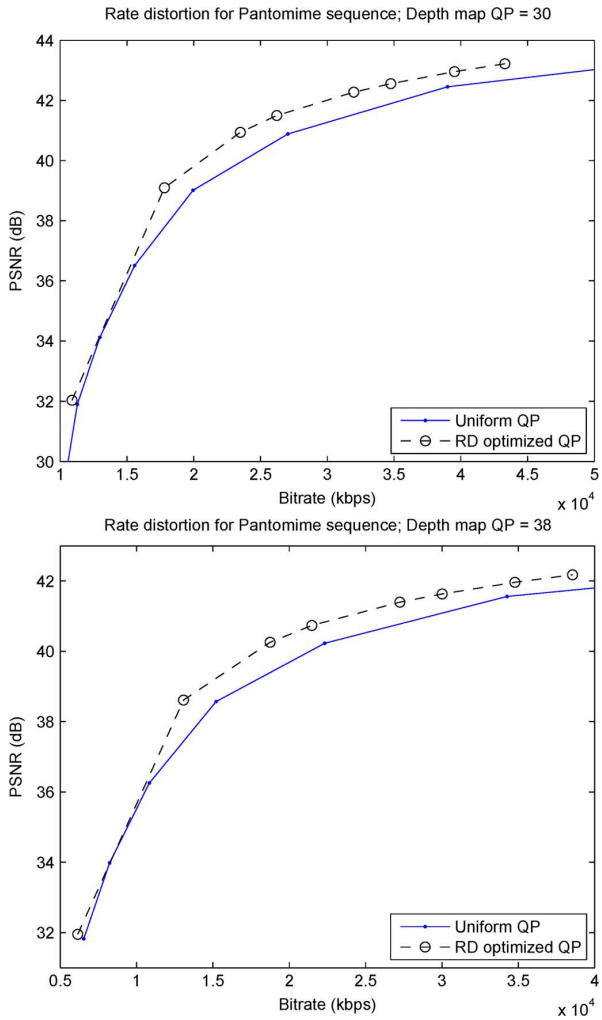


Fig. 4. Rate-distortion plot for compressing $N = 10$ views of sequence Pantomime in a FVTV system being watched by $M = 400$ viewers, following the viewer distribution in Fig. 2.

In the RD plots, rate is the sum of all $N$ rates of the camera views and of the depth maps, while distortion is the TOD of all $M$ viewers.

Results indicate sizable gains in optimizing rate allocation based on viewer attention. More uniform viewing distributions would lead to smaller gains, while more eccentric distributions would lead to larger gains.

## IV. Conclusion

This letter discusses two contributions. First, attention weighted rate allocation in FVTV is proposed. In effect we propose to give more bits to cameras in viewpoint regions with higher audience and attention. In an extreme, if nobody is watching from any viewpoint that requires a certain camera view, there is no need to encode that camera view at all, so that we can devote the saved bits to enhance the quality of camera views being used more often. The method requires a slower feedback channel to inform the encoder about the viewpoint locations.

Second, in the search for a means to allocate bit-rates $(QP_k)$ for the different camera views according to their audience, we developed a method that leads to optimal camera view rate allocation considering total observed distortion and a linear model for the distortion of synthesized views. Such a linear model was proven to be efficient given the positive results obtained in our simulations.

Simulation results computing the synthesized views at 400 receivers show that the method yields substantial gains in rate-distortion performance considering the effective observed distortion.

Further enhancements could include accommodating a depth-map bit allocation scheme [13], as well as a more detailed handling of non-linear camera arrangements.

## References

[1] M. Tanimoto, M. P. Tehrani, T. Fujii, and T. Yendo, "Free-viewpoint TV," *IEEE Signal Process. Mag.*, vol. 28, no. 1, pp. 67–76, Jan. 2011.

[2] A. Smolic *et al.*, "3D video and free viewpoint video—Technologies, applications and MPEG standards," in *Proc. IEEE Int. Conf. Multimedia and Expo, ICME*, Toronto, ON, Canada, Jul. 2006, pp. 2161–2164.

[3] H. Kimata, M. Kitahara, K. Kamikura, and Y. Yashima, "Multi-view video coding using reference picture selection for free-viewpoint video communication," in *Proc. Picture Coding Symp.*, San Francisco, CA, USA, 2004.

[4] A. Kubota *et al.*, "Multiview imaging and 3DTV: Special issue overview and introduction," *IEEE Signal Process. Mag.*, vol. 24, no. 11, pp. 10–21, Nov. 2007.

[5] H. Kimata *et al.*, "Real-time MVC viewer for free viewpoint navigation," in *Proc. IEEE Int. Conf. Multimedia and Expo, ICME*, 2008, pp. 1437–1440.

[6] S. C. Chan, H.-Y. Shum, and K.-T. Ng, "Image-based rendering and synthesis," *IEEE Signal Process. Mag.*, vol. 24, no. 6, Jun. 2007.

[7] V. Testoni, D. Florencio, and M. Costa, "Optimizing QPs for multiview image coding for free viewpoint video," in *Proc. Simpósio Brasileiro de Telecomunicações*, Blumenau, Brazil, Sep. 2009.

[8] D. Florencio and C. Zhang, "Multiview video compression and streaming based on predicted viewer position," in *Proc. IEEE Int. Conf. Acoust. Speech an Signal Proc. ICASSP*, Taipei, Taiwan, 2009, pp. 657–660.

[9] C. Zhang, Z. Yin, and D. Florencio, "Improving depth perception with motion parallax and its application in teleconferencing," in *Proc. IEEE Int. Workshop Multimedia Signal Proc. MMSP*, Rio de Janeiro, Brazil, 2009, pp. 1–6.

[10] C. Zhang, D. Florencio, and Z. Zhang, "Improving immersive experiences in telecommunication with motion parallax," *IEEE Signal Process. Mag.*, vol. 28, no. 1, Jan. 2011.

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

[12] "Reference softwares for depth estimation and view synthesis," Apr. 2008, ISO/IEC JTC1/SC29/WG11, Doc. M15377.

[13] V. Velisavljevic, G. Cheung, and J. Chakareski, "Bit allocation for multi-view image compression using cubic synthesized view distortion model," in *Proc. IEEE Int. Conf. Multimedia and Expo ICME*, Barcelona, Spain, 2011.