MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

An Analytical Model for Synthesis Distortion Estimation in 3D Video

Fang, L.; Cheung, N-M; Tian, D.; Vetro, A.; Sun, H.; Au, O.C.

TR2013-100 October 2013

Abstract

We propose an analytical model to estimate the synthesized view quality in 3D video. The model relates errors in the depth images to the synthesis quality, taking into account texture image characteristics, texture image quality and the rendering process. Specifically, we decompose the synthesis distortion into texture-error induced distortion and depth-error induced distortion. We analyze the depth-error induced distortion using an approach combining frequency and spatial domain techniques. Experiment results with video sequences and coding/rendering tools used in MPEG 3DV activities show that our analytical model can accurately estimate the synthesis noise power. Thus, the model can be used to estimate the rendering quality for different system designs.

IEEE Transactions on Image Processing

An Analytical Model for Synthesis Distortion Estimation in 3D Video

Lu Fang, Ngai-Man Cheung, Dong Tian, Anthony Vetro, Huifang Sun and Oscar C. Au

***Abstract*—We propose an analytical model to estimate the synthesized view quality in 3D video. The model relates errors in the depth images to the synthesis quality, taking into account texture image characteristics, texture image quality and the rendering process. Specifically, we decompose the synthesis distortion into texture-error induced distortion and depth-error induced distortion. We analyze the depth-error induced distortion using an approach combining frequency and spatial domain techniques. Experiment results with video sequences and coding/rendering tools used in MPEG 3DV activities show that our analytical model can accurately estimate the synthesis noise power. Thus, the model can be used to estimate the rendering quality for different system designs.**

***Index Terms*—3D video, DIBR, depth map coding, rendering, view synthesis, power spectral density, gradient-based analysis**

I. Introduction

A. Motivation

3D video (3DV) has attracted much attention recently [1]–[5]. 3D datasets usually consist of multiple video sequences (texture data) captured by cameras at different positions, along with the associated depth images. The per-pixel depth information in the depth images allows synthesis of virtual views at user-chosen viewpoints via depth-image-based rendering (DIBR) [6] [7]. Depth information could be measured using some range imaging devices such as time-of-flight cameras. Alternatively, it could be estimated from the texture data using computer vision techniques.

In many 3DV applications, the quality of the synthesized view is imperative [8], [9]. The rendering quality, however, depends on several factors and complicated interactions among them. In particular, texture and depth images may contain errors due to imperfect sensing or lossy compression [10], [11], and it is not clear how these errors interact and affect the rendering quality. Unlike texture errors, which cause distortion in the luminance/chrominance level, depth errors cause *position errors* in synthesis [12], i.e., pixels are warped to slightly shifted positions during synthesis. The effect of depth errors is very subtle. For instance, the impact of depth errors would vary with the image contents, and images with less textures tend to be more resilient to the depth errors. The impact of depth errors also depends on the camera configuration as this affects the magnitudes of position errors. Along the rendering pipeline, depth errors are also transformed in different operations complicating the study of their effects.

An accurate analytical model to estimate the rendering quality is very valuable for the design of 3DV systems. As an example, the model may help understand under what conditions reducing the depth error would substantially improve the synthesis output. 3DV encoders can then use the information to decide when to allocate more bits to code the depth images. As another example, the model may be used to estimate how much improvement can be achieved by placing cameras closer together given other factors such as errors in the texture data.

Lu Fang is with University of Science and Technology of China (USTC) (fanglu@ustc.edu.cn). Ngai-Man Cheung is with the Singapore University of Technology and Design (SUTD), Singapore (ngaiman_cheung@sutd.edu.sg). Dong Tian, Anthony Vetro and Huifang Sun are with Mitsubishi Electric Research Laboratories, USA ({tian, avetro, hsun}@merl.com). Oscar C. Au is with Hong Kong University of Science and Technology (eeau@ust.hk).

B. Our contributions

In this work, we analyze how depth errors relate to the rendering quality, taking into account texture image characteristics, texture image quality and the rendering process. In particular, we propose a framework that decouples the effects due to errors in texture frames and depth maps to facilitate analysis. We propose to model the distortions due to depth map errors using an approach that combines frequency domain and spatial domain analysis. Frequency domain analysis provides a concise and compact representation to understand the synthesis distortions, while spatial domain analysis accounts for the spatial variant signals in the video frames. In our frequency domain analysis, we use power spectral density (PSD) [13]. This is inspired by earlier work which used PSD to study the effect of motion vector inaccuracy [14] and disparity inaccuracy [15]. However, while previous work applied PSD to analyze the efficiency of the motion/disparity compensated predictors in predictive coding, our work uses PSD to quantify the noise power in the rendering output of the synthesis pipeline. As will be clear, we focus on transformation/interaction of the texture/depth error in the synthesis pipeline. In addition, we analyze the synthesis distortion caused by depth errors in the spatial variant signals along strong edges with a spatial domain analysis. We decompose the local signals into gradient-based representations, and show that linear approximation of the signals (i.e., constant gradient approximation) can achieve accurate estimation with negligible computation. Specifically, our contributions are:

- We propose an analytical model to estimate the rendering quality given depth map errors, texture image characteristics (smooth or textural) and texture image quality as the inputs.

- We propose a model that combines frequency and spatial domain analysis to estimate the distortion due to depth map errors.
- In our proposed model, depth errors are used to compute the position errors, and the probability distribution of the position errors is in turn used to estimate, along with the texture image PSD, the synthesis noise power of the *spatial invariant* signals in the video frame[1].
- We also propose to decompose the *spatial variant* signals into gradient-based representations to facilitate analysis. The analysis results show that linear approximations of the spatial variant signals can lead to a computationally-efficient and yet accurate estimation.
- We verify our model with substantial experiments using video sequences and coding and rendering tools from the MPEG 3DV activities [16], [17].

C. Related work

Several algorithms have been proposed to estimate the rendering quality. Nguyen and Do [18] analyzed the rendering quality of image-based rendering (IBR) algorithms and used Taylor series expansion to derive the *upper bound* of the mean absolute error (MAE) in the synthesis output. They also quantified the effect of sample jitters caused by depth errors. On the contrary, our work analyzes the rendering quality with a combination of frequency and spatial domain techniques that are quite different from their work, and our model estimates the value (instead of upper bound) of the mean squared error (MSE). We also test the model with video sequences and coding/rendering tools used in MPEG 3DV activities [16], [17].

Liu et al. [19] proposed a distortion model to evaluate the synthesized view. Their work approximated errors due to depth map artifacts using a linear model of average magnitude of mean-squared disparity errors over an entire frame and a motion sensitivity factor computed from the energy density. This was motivated by earlier work of using linear distortion model to characterize the effect of motion warping error [20]. Our work is different in that we characterize the disparity errors using their distribution rather than their average, and use a different analysis technique to derive the distortion caused by the disparity error distribution. We also notice that spatial variant signals would cause non-negligible discrepancy (In our previous work [13], this necessitates compensating with a video sequence specific constant). Therefore, we propose to augment frequency-domain analysis with spatial-domain analysis of spatial-variant signals. Our analysis framework is also different and leads to a different formulation for synthesis error decoupling.

Yuan et al. [21] proposed a frequency approach to estimate synthesis distortion. Similar to Liu et al. [19], their work was motivated by the linear distortion model characterizing the effect of motion warping error [20]. In Yuan et al. [21], they derived a linear model that relates synthesized view distortion with the quantization steps of the texture and depth videos. Model parameters are estimated by synthesis of three virtual views using compressed texture/depth videos with different texture/depth quantization steps. Note that their model parameters are specific to particular virtual view positions, scene characteristics, coding algorithms and encoding options (since quantization step is used as input in their model). That is, new model parameters need to be estimated using synthesis when these variables change. The author has also extended the work to wiener filter design in [22]. The approach proposed by Wang et al. [7] is similar to Yuan et al., with focus on rate-distortion analysis of free viewpoint coding.

An autoregressive model was proposed by Kim et al. [23] to estimate the synthesis distortion at the block level and was shown to be effective for rate-distortion optimized mode selection. In their work, rendering distortion of a block is approximated by the local video signal variance and a first order autoregressive model for the correlation coefficients. A distortion model as a function of the view location was also proposed by Velisavljevic et al. [24] for bit allocation. Takahashi [25] proposed an optimized view interpolation scheme based on frequency domain analysis of depth map error. Some of his frequency analysis is similar to our previous work [13], but there is no accuracy comparison provided in his work. Our work is different as we take into account both distortions in the texture images and depth maps, and estimate the distortion due to depth map artifacts using probability distribution of position errors, PSD of texture, and linear approximation of local spatial-variant signals. The present work significantly extends our previous work [13] by augmenting the frequency domain analysis with a spatial domain analysis. The frequency domain analysis is also modified in order to accommodate the new approach. Elaborated experiment results and analysis are provided in this work.

The outline of the rest of the paper is as follows. Sections II-IV discuss our analytical model. Section V summarizes our proposed model. Section VI presents experiment results and Section VII concludes the paper.

II. System model

Figure 1 models the processing in a typical synthesis pipeline (See Table I for a summary of notations). Two reference texture frames captured by the left and right cameras (denoted by $X_l(m,n)$ and $X_r(m,n)$ respectively) along with their associated depth images (denoted by $D_l(m,n)$ and $D_r(m,n)$ respectively) are used to generate the synthesized frame $U(m,n)$ at a certain virtual camera position. First, in frame warping, pixels are copied from X_l to form an intermediate frame U_l, from position (m',n) to (m,n). We assume the cameras are rectified and arranged linearly, and there exists only horizontal disparity given by

$$m - m' = \frac{D_l(m',n)}{255}(d_{near} - d_{far}) + d_{far}, \qquad (1)$$

where $d_{near} = \frac{f \cdot b_l}{z_{near}}$, $d_{far} = \frac{f \cdot b_l}{z_{far}}$, f is the focal length, b_l is the distance between the left and virtual camera centers, and z_{near} and z_{far} are the nearest and farthest depth values.

Likewise, pixels are copied from X_r to form the intermediate frame U_r with horizontal disparity $m - m''$. Then, U_l and

[1]This is based on our conference paper work in [13].

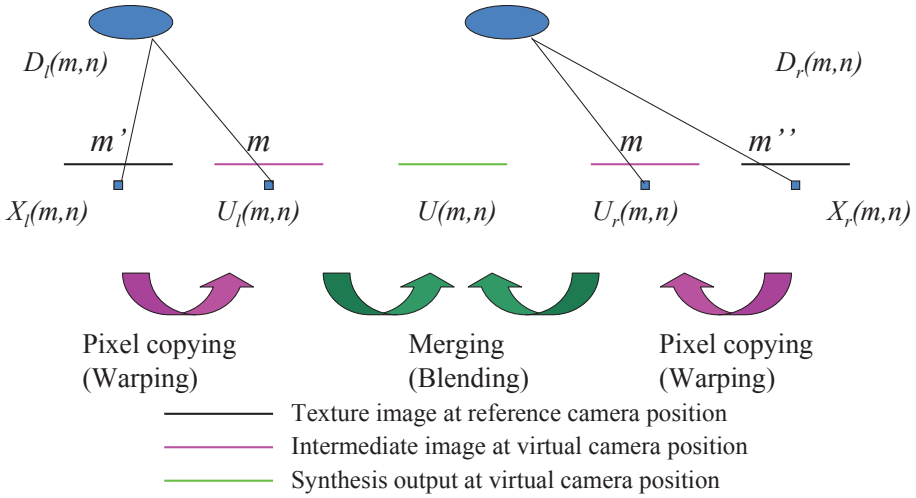


Fig. 1. Processing in the synthesis pipeline. Based on the depth maps, pixel of X_l at location (m', n) is warped to (m, n), while pixel of X_r at location (m'', n) is warped to (m, n). Horizontal disparity is $m - m'$ for the left reference, and $m - m''$ for the right reference.

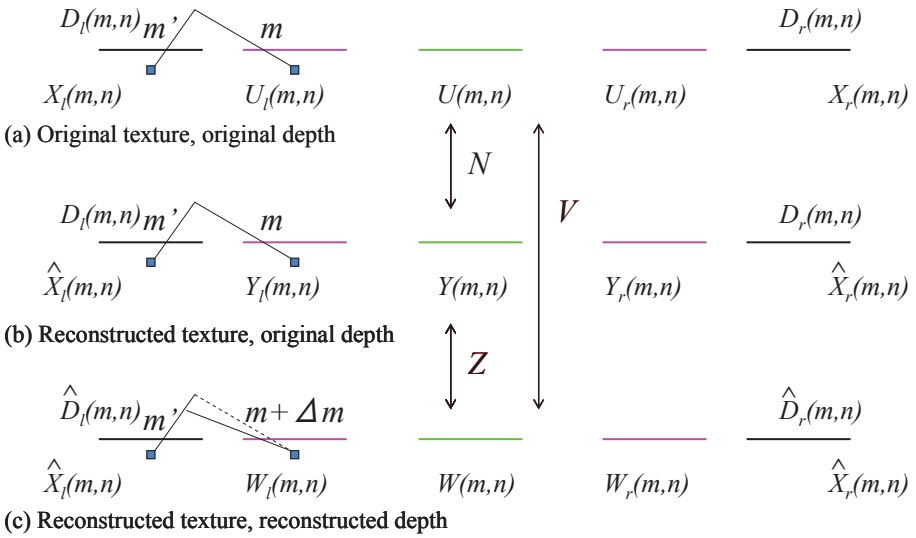


Fig. 2. Analysis of the rendering error. (a) Synthesis with the original texture and original depth images. (b) Synthesis with the reconstructed texture and original depth images. (c) Synthesis with the reconstructed texture and reconstructed depth images. N is the noise due to error in texture images. Z is the noise due to error in depth images. V is the overall synthesis noise.

U_r are merged (blended) to generate U. We assume merging by linear combination

$$U(m,n) = \alpha U_l(m,n) + (1-\alpha)U_r(m,n), \qquad (2)$$

where the weight α is determined by the distances between the virtual camera position and the left/right reference camera positions. Note that other merging techniques have been proposed, e.g., those that take into account the depth [26]. However, linear merging remains to be a popular practical technique and could be a good baseline technique. Note also that at some pixel locations, $U_l(m,n)$ or $U_r(m,n)$ or both may be missing due to position rounding error, disocclusion or outside of the field-of-view of the reference cameras. Nevertheless, if the distances between the reference/virtual cameras are small, such number of missing pixels is usually small, and they would not cause significant model discrepancy.

In practice, the texture and depth images are lossy encoded, and Figure 2 depicts our approach to analyze the effect of the coding errors in rendering. In particular, in Figure 2(c), the reconstructed texture/depth images $(\hat{X}_l, \hat{X}_r, \hat{D}_l, \hat{D}_r)$ are fed into the synthesis pipeline to produce the left/right intermediate images (W_l and W_r respectively), which are merged to generate the synthesis output W. Synthesis quality is usually measured, as in the ongoing MPEG 3DV activities, between the rendering outputs with the *original* texture/depth images and the *reconstructed* texture/depth images, i.e., between U in Figure 2(a) and W in Figure 2(c). We denote the synthesis noise by $V = U - W$, i.e., V is the noise in the synthesis output due to (coding) error in the texture/depth images.

To facilitate the analysis, we consider an intermediate step to model the synthesis noise. As shown in Figure 2(b), we consider the case when the *reconstructed* texture images and the *original* depth images are used in the synthesis to produce the output Y. Note that U and Y are different solely due to the fact that reconstructed texture frames $\hat{X}_l, \hat{X}_r$ are used in

TABLE I
NOTATIONS

Notation	Meaning
X_l, X_r	Original reference texture frames captured from the left and right cameras respectively
D_l, D_r	Original depth maps associated with the left and right images respectively
$\hat{X}_l$, $\hat{X}_r$	Reconstructed texture frames for the left and right cameras respectively
$\hat{D}_l$, $\hat{D}_r$	Reconstructed depth maps associated with the left and right images respectively
m, n	Horizontal and vertical coordinates in a texture frame or a depth map respectively
U_l, U_r	Intermediate frames at virtual position produced from warping X_l with depth map D_l, and X_r with depth map D_r respectively
U	Synthesized frame at virtual position using X_l, X_r, D_l, D_r
Y_l, Y_r	Intermediate frames at virtual position produced from warping $\hat{X}_l$ with depth map D_l, and $\hat{X}_r$ with depth map D_r respectively
Y	Synthesized frame at virtual position using $\hat{X}_l$, $\hat{X}_r$, D_l, D_r
W_l, W_r	Intermediate frames at virtual position produced from warping $\hat{X}_l$ with depth map $\hat{D}_l$, and $\hat{X}_r$ with depth map $\hat{D}_r$ respectively
W	Synthesized frame at virtual position using $\hat{X}_l$, $\hat{X}_r$, $\hat{D}_l$, $\hat{D}_r$
N	Synthesis noise induced by lossy coding of texture frames, $N = U - Y$
Z	Synthesis noise induced by lossy coding of depth maps, $Z = Y - W$
V	Overall synthesis noise induced by lossy coding of texture frames and depth maps
α	Weight in merging the intermediate frames
ρ_N	Correlation coefficient between $X_l - \hat{X}_l$ and $X_r - \hat{X}_r$
ρ_Z	Correlation coefficient between Z_l and Z_r

the synthesis instead of the original texture. Same depth maps D_l, D_r are used to produce U and Y. Thus, $N = U - Y$ is the noise component due to lossy coding of texture frames. On the other hand, $Z = Y - W$ is the additional distortion due to error in the depth images. Note that $V = N + Z$. Yuan et al. [21] [22] have performed detailed analysis on $E[NZ]$. Their analysis uses Taylor series expansion of several quantities. Under the assumption that quantization error in lossy coding can be modeled as zero mean white noise, they show that the expected value of several quantities are zero and $E[NZ]$ is approximately equal to zero. Therefore, we have

$$\begin{aligned} E[V^2] &= E[N^2] + E[Z^2] + 2E[NZ] \\ &= E[N^2] + E[Z^2]. \end{aligned} \tag{3}$$

(3) suggests that the synthesis noise power due to texture image coding ($E[N^2]$) and depth image coding ($E[Z^2]$) can be estimated separately. As will be seen, this simplifies the estimation of each components, and the total noise power can be approximated simply by summing the two components. We would like to emphasis that we introduce this intermediate step as shown in Figure 2(b) solely for the purpose of facilitating the analysis and this helps decouple the effect due to errors in texture frames and depth maps[2].

III. Estimate the Noise Power Due to Texture Coding

We proceed to discuss how to estimate the two component noise signals in (3). We first focus on the noise caused by lossy coding of texture image. Refer to Figures 2(a) and 2(b),

$$N(m,n) = U(m,n) - Y(m,n), \tag{4}$$

$$\begin{aligned} U(m,n) &= \alpha U_l(m,n) + (1-\alpha) U_r(m,n) \\ &= \alpha X_l(m',n) + (1-\alpha) X_r(m'',n), \end{aligned} \tag{5}$$

$$\begin{aligned} Y(m,n) &= \alpha Y_l(m,n) + (1-\alpha) Y_r(m,n) \\ &= \alpha \hat{X}_l(m',n) + (1-\alpha) \hat{X}_r(m'',n). \end{aligned} \tag{6}$$

Therefore,

$$\begin{aligned} N(m,n) &= \alpha \left(X_l(m',n) - \hat{X}_l(m',n) \right) \\ &+ (1-\alpha) \left(X_r(m'',n) - \hat{X}_r(m'',n) \right). \end{aligned} \tag{7}$$

In (5), pixel in X_l at location (m',n) is copied to the intermediate image U_l location (m,n) (See Figure 2(a)). Likewise, in (6), pixel in $\hat{X}_l$ at location (m',n) is copied to intermediate image Y_l location (m,n) (See Figure 2(b)). Importantly, pixels in X_l and $\hat{X}_l$ involved in computing $N(m,n)$ are *spatially collocated*, both from (m',n) (similarly for the right camera, both from (m'',n)). This is because in this first step (computing $N(m,n)$) the same (original) depth information is used in both (5) and (6) to calculate the disparity. The fact that pixels involved in computing $N(m,n)$ are collocated simplifies the estimation,

$$\begin{aligned} E\left[N^2\right] &= \alpha^2 E[(X_l - \hat{X}_l)^2] + (1-\alpha)^2 E[(X_r - \hat{X}_r)^2] \\ &+ 2\alpha(1-\alpha)\rho_N \sigma_{X_l - \hat{X}_l} \sigma_{X_r - \hat{X}_r}, \end{aligned} \tag{8}$$

where $X_l - \hat{X}_l$ and $X_r - \hat{X}_r$ are the texture coding noise signals, and ρ_N is the correlation coefficient between $X_l - \hat{X}_l$ and $X_r - \hat{X}_r$. ρ_N tends to be small, and depends on the quality of texture image coding. In particular, if the texture images are encoded at low quality, there would be considerable structural information remained in $X_l - \hat{X}_l$ and $X_r - \hat{X}_r$, and they would be more correlated. We trained a model to estimate

[2]Note that if we use $\hat{D}$ and X in the intermediate step (instead of D and $\hat{X}$ as in Figure 2(b)), then the difference between the intermediate step and the original one would be the distortion caused by depth error (Z), and the difference between the final step and intermediate step becomes the distortion caused by texture error (N). Equation (3) will still hold subject to the lossy coding assumption, and the estimation of N and Z can follow mostly the same steps. Thus, the use of $\hat{D}$ or $\hat{X}$ in the intermediate step will only have minimum effect on our analysis.

ρ_N (parameterized by the average of $E[(X_l - \hat{X}_l)^2]$ and $E[(X_r - \hat{X}_r)^2]$), and the same model is used in all sequences and coding conditions.

IV. Estimate the Noise Power Due to Depth Coding

We then focus on the rendering noise caused by error in the depth images. Refer to Figures 2(b) and 2(c),

$$\begin{aligned} Z(m,n) &= Y(m,n) - W(m,n), && (9)\\ Y(m,n) &= \alpha Y_l(m,n) + (1-\alpha) Y_r(m,n), && (10)\\ W(m,n) &= \alpha W_l(m,n) + (1-\alpha) W_r(m,n). && (11) \end{aligned}$$

Substitute (10) and (11) into (9), and with $Z_l = Y_l - W_l$, $Z_r = Y_r - W_r$, we have

$$Z(m,n) = \alpha Z_l(m,n) + (1-\alpha) Z_r(m,n), \quad (12)$$

$$\begin{aligned} E[Z^2] &= \alpha^2 E[Z_l^2] + (1-\alpha)^2 E[Z_r^2] \\ &+ 2\alpha(1-\alpha)\rho_Z \sigma_{Z_l}\sigma_{Z_r}. \end{aligned} \quad (13)$$

(13) suggests that the noise power due to depth error can be estimated from the error components Z_l, Z_r in the left/right cameras respectively. To estimate $E[Z_l^2]$ (and likewise $E[Z_r^2]$),

$$\begin{aligned} Z_l(m,n) &= Y_l(m,n) - W_l(m,n) \\ &= Y_l(m,n) - Y_l(m - \Delta m_l, n). \end{aligned} \quad (14)$$

Here the depth error causes a horizontal position error Δm_l.

We propose to estimate $E[Z_l^2]$ (and likewise $E[Z_r^2]$) using an approach that combines frequency domain analysis with PSD and spatial domain analysis with local gradient information. While PSD has been used in various distortion estimation problems in video signal processing, it assumes that the underlying image signals are spatial invariant (i.e., wide-sense stationary), which we found that in the current application this would cause significant estimation discrepancy [13]. Specifically, an image $X = \{X(m,n)\}$ is conventionally modeled as a random field (each $X(m,n)$ is a random variable). Spatial invariant assumption implies that

$$E[X(m,n)] = \mu(m,n) = \mu, \quad (15)$$

where μ is a constant, and

$$E[(X(i,j) - \mu(i,j))(X(m,n) - \mu(m,n))] = R_X(i-m, j-n), \quad (16)$$

where $R_X(\cdot,\cdot)$ is the autocovariance function. However, across strong texture edges $X(m,n)$ changes much more quickly than in the non-edge regions. Edge pixels exhibit significantly different correlation statistics compared with those in the non-edge regions (autocovariance function decreases significantly faster in edge pixels). We found that models that fail to account for these non-stationary characteristics would incur considerable estimation discrepancy in rendering quality estimation. It is because at regions where $X(m,n)$ changes rapidly (strong texture edges) pixel shifts would result in substantial rendering errors and they are not negligible.

We propose to partition the video frame signals into *Spatial Invariant (SI)* signals and *Spatial Variant (SV)* signals, and analyze these signals with frequency and spatial techniques respectively. Specifically, we start by analyzing the gradient map of texture image $X_l(X_r)$ and partition the video frame into spatial-invariant and spatial-variant regions. We consider pixels whose gradient magnitudes (computed from texture images) are lower than a predetermined gradient threshold as belonging to spatial invariant regions, and pixels whose gradient magnitudes are higher than the predetermined gradient threshold as belonging to spatial variant regions. To determine the gradient threshold automatically, we use the Otsu's algorithm [27] and apply it to the gradient magnitudes of a video frame (texture image). The Otsu's algorithm determines a threshold that minimizes the weighted sum of class variances. That is, the (weighted) sum of the variances of the gradient magnitudes within the spatial-invariant regions and the spatial-variant regions shall be minimized. As shown in Fig. 3, we take one frame (texture image) of the sequence "Kendo" as an example to demonstrate the thresholding of SI and SV regions via Otsu's algorithm. In the following two subsections, we will estimate $E[Z_l^2]$ (and likewise $E[Z_r^2]$) for the pixels belonging to SI and SV regions, which we denote them as $E[Z_{l,\mathrm{SI}}^2]$ and $E[Z_{l,\mathrm{SV}}^2]$ respectively. We determine $E[Z_l^2]$ as a combination of distortions,

$$E[Z_l^2] = E[Z_{l,\mathrm{SI}}^2] + E[Z_{l,\mathrm{SV}}^2], \quad (17)$$

where $E[Z_{l,\mathrm{SI}}^2]$, $E[Z_{l,\mathrm{SV}}^2]$ are normalized by the number of pixels in a video frame, as will be discussed. Note that most of the pixels change moderately and they are classified as belonging to SI regions by the Otsu's algorithm as shown in Fig. 3. Isolated regions of pixels with rapidly changing correlation statistics are classified as belonging to SV regions and they are analyzed individually with a spatial domain technique.

A. Error Model for Spatial Invariant Regions

All the SI regions in a video frame are characterized by a single parametric PSD model in the frequency-domain analysis. Specifically, from (14), the PSD of $Z_{l,SI}$ can be derived by

$$\Phi_{Z_{l,\mathrm{SI}}}(\omega_1,\omega_2) = 2(1-\cos(\Delta m_l \cdot \omega_1))\Phi_{Y_{l,\mathrm{SI}}}(\omega_1,\omega_2). \quad (18)$$

(18) is obtained by taking discrete-time Fourier transform of (14) and computing the squared modulus from both sides (See Appendix A). Since Δm_l is random, we take expectation in (18) w.r.t. the probability distribution of Δm_l and $p(\Delta m_l)$,

$$\begin{aligned} \Phi_{Z_{l,\mathrm{SI}}}(\omega_1,\omega_2) &= 2(1 - E[\cos(\Delta m_l \cdot \omega_1)])\Phi_{Y_{l,\mathrm{SI}}}(\omega_1,\omega_2) \\ &= 2(1 - \mathrm{Re}\{P(\omega_1)\})\Phi_{Y_{l,\mathrm{SI}}}(\omega_1,\omega_2), \end{aligned} \quad (19)$$

where $P(\omega_1)$ is the Fourier transform of $p(\Delta m_l)$, and $\mathrm{Re}\{P(\omega_1)\}$ denotes the real part of $P(\omega_1)$. Here we use the result of $E[\cos(\Delta m_l \cdot \omega_1)] = \mathrm{Re}\{P(\omega_1)\}$. This can be derived by starting with the identity, $\cos(\Delta m_l \cdot \omega_1) = (e^{j\Delta m_l \cdot \omega_1} + e^{-j\Delta m_l \cdot \omega_1})/2$, and then noticing that taking expectation of $e^{-j\Delta m_l \cdot \omega_1}$ w.r.t. distribution $p(\Delta m_l)$ is equivalent to applying Fourier transform $\mathscr{F}[.]$ to $p(\Delta m_l)$,

$$\begin{aligned} E_{p(\Delta m_l)}[e^{-j\Delta m_l \cdot \omega_1}] &= \int e^{-j\Delta m_l \cdot \omega_1} p(\Delta m_l) d\Delta m_l \\ &= \mathscr{F}[p(\Delta m_l)] \end{aligned} \quad (20)$$

(a)

(b)

Fig. 3. (a) Texture image of Kendo sequence (view 3) (b) Corresponding thresholding result using Otsu's algorithm (black: spatial invariant regions, white: spatial variant regions).

Details of the derivation can be found in [14]. Approximating $\Phi_{Y_{l,\mathrm{SI}}}$ by $\Phi_{\hat{X}_{l,\mathrm{SI}}}$, we finally obtain[3],

$$\Phi_{Z_{l,\mathrm{SI}}}(\omega_1,\omega_2) \approx 2(1-\mathrm{Re}\{P(\omega_1)\})\Phi_{\hat{X}_{l,SI}}(\omega_1,\omega_2). \quad (21)$$

We use (21) to compute the distortion. We approximate $\Phi_{\hat{X}_{l,SI}}(\omega_1,\omega_2)$ by a parametric model. We assume that the signals in SI regions follow an isotropic autocorrelation function,

$$R_{\hat{X}_{l,SI}}(\Delta m,\Delta n) = \exp(-\omega_0\sqrt{\Delta m^2+\Delta n^2}). \quad (22)$$

where $R_{\hat{X}_{l,SI}}(\cdot)$ denotes the autocorrelation function for the pixels in spatial-invariant regions of $\hat{X}_l$ [28]. Δm and Δn are the horizontal and vertical distances of two samples in spatial domain of $\hat{X}_{l,\mathrm{SI}}$, i.e., $(\Delta m,\Delta n) = (1,0)$ and $(\Delta m,\Delta n) = (0,1)$ represent the horizontally and vertically adjacent samples respectively. ω_0 is the correlation between adjacent pixels in SI. The PSD of the signal is [28],

$$\Phi_{\hat{X}_{l,SI}}(\omega_1,\omega_2) = \begin{cases} 2\pi\omega_0\left(\omega_0^2+\omega_1^2+\omega_2^2\right)^{-3/2}, & |\omega_1|\le\pi \text{ and } |\omega_2|\le\pi, \\ 0, & \text{otherwise.} \end{cases} \quad (23)$$

(23) is used for $\Phi_{\hat{X}_{l,SI}}(\omega_1,\omega_2)$ in (21) for distortion estimation. (21) suggests that the PSD of the error due to lossy coding of the (left) depth image is the product of the PSD of the texture signal and the envelop, $2(1-\mathrm{Re}\{P(\omega_1)\})$, which depends on the distribution $p(\Delta m_l)$. The distribution $p(\Delta m_l)$ for the left camera depends on the depth error and the camera set-up, and can be readily obtained from D_l, $\hat{D}_l$ and binning Δm_l (similarly for the right camera),

$$\Delta m_l(m,n) = k_l(D_l(m,n)-\hat{D}_l(m,n)), \quad (24)$$

where k_l is a spatially invariant constant depending only on the camera setup,

$$k_l = \frac{f\cdot b_l}{255}\left(\frac{1}{z_{near}}-\frac{1}{z_{far}}\right). \quad (25)$$

We integrate $\Phi_{Z_{l,\mathrm{SI}}}$ in (21) to estimate $E[Z^2_{l,\mathrm{SI}}]$. To illustrate how $\Phi_{Z_{l,\mathrm{SI}}}$ depends on $p(\Delta m_l)$, Figure 4 depicts the empirical $p(\Delta m_l)$ of a video frame from the sequence Kendo, the corresponding envelope $2(1-\mathrm{Re}\{P(\omega_1)\})$, the PSD of the texture signal, and the PSD of the rendering noise. As suggested in Fig. 4, error due to lossy depth coding mostly depends on the high frequencies of the texture signals. This agrees with the observation that lossy depth coding causes more rendering artifacts in scenes with a lot of textures (which have a lot of high frequencies), but less degradation in homogeneous scenes (which have primarily low frequencies).

B. Error Model for Spatial Variant Regions

1) Overview of our approach: To estimate the distortion due to depth errors in the spatial-variant (SV) regions, we process the frame row-by-row. For each row, we process one by one each SV region (a SV region consists of consecutive pixels classified as SV). For each SV region, we approximate the rendering distortion based on the average horizontal gradient and average position error of that region using a very simple equation. We will give details and justifications in the following sections.

2) Gradient-based analysis of SV regions: Let us denote a vector $\vec{S_L}$ as the pixel values of a SV region of extent (width or length) L in $Y_{l,\mathrm{SV}}$, $\vec{S'_L}$ as the one in $W_{l,\mathrm{SV}}$. Recall that due to the depth coding artifacts, there exists depth error for depth map, resulting in horizontal disparity error during texture image wrapping, i.e., $W_{l,\mathrm{SV}}(m,n) = Y_{l,\mathrm{SV}}(m-\Delta m_l,n)$, as shown in (14) and Figure 2.

Specifically, in SV regions, the sharp edge would magnify the effect of horizontal disparity error on the rendering distortion between $\vec{S_L}$ and $\vec{S'_L}$. To model the effect of both gradient value and depth error (horizontal disparity error) on the rendering result, we decompose $\vec{S_L}$ into L *gradient-based component-vectors*, such that

[3] Note that we propose to estimate the rendering error caused by the texture and depth errors *without* actually performing any warping / synthesis of the virtual view. In other words, Y_l is in fact unknown. Fortunately, the spectral content of $\hat{X}_l$ and Y_l would be similar as they represent the same scene at slightly different view angles. Therefore, PSD of $\hat{X}_l$, $\Phi_{\hat{X}_l}$, can be used to approximate PSD of Y_l, Φ_{Y_l}. This approximation is accurate provided that the view angle difference is small.

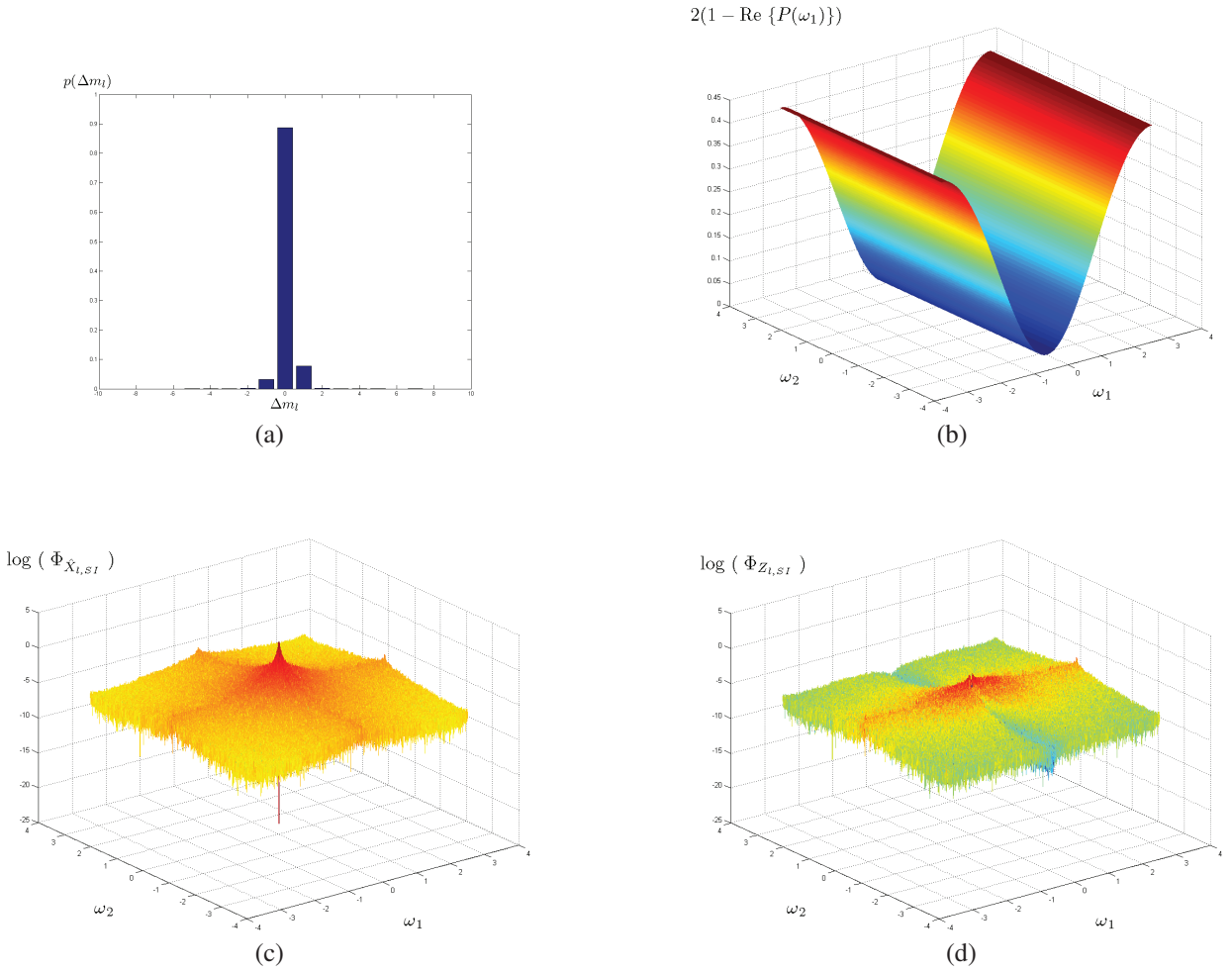


Fig. 4. Rendering noise due to depth errors as characterized by (21). (a) Empirical distribution of the position errors, $p(\Delta m_l)$. This is computed from the depth errors of a typical reconstructed depth image using (24). In particular, depth maps in the Kendo sequence are encoded using JMVC Encoder 8.3.1. Note that the distribution has non-zero probability mass from -10 to 10, although most of the probability mass is concentrated in the three peaks: $-1, 0, 1$. (b) The corresponding frequency envelope: $2(1-\text{Re}\{P(\omega_1)\})$. This is obtained by applying Fourier transform on (a) to obtain $P(\omega_1)$, and taking only the real part of $P(\omega_1)$. Note that the envelope attenuates low horizontal frequency signals. (c) PSD of the texture signals. This is obtained by applying discrete-time Fourier transform on a typical reconstructed video frame in the Kendo sequence, and PSD is computed as the squared magnitude of frequency representation. (d) PSD of the rendering noise due to errors in the depth images. This is obtained as the product of (b) and (c) following our deviation. A valley at the low horizontal frequency can be observed.

$$\vec{S}_L = \sum_{k=1}^{L} \vec{s}_k, \tag{26}$$

where $k = 1, 2, \cdots, L$ and $\vec{s}_k$ is the k^{th} gradient-based component-vector, given by

$$\vec{s}_k = g_k \vec{1}_k, \tag{27}$$

where g_k is the gradient value at the k^{th} spatial location in $\vec{S}_L$, and $\vec{1}_k$ is a vector with $k-1$ zeros followed by $L-k+1$ ones, i.e., $\vec{1}_k = [\underbrace{0, \cdots, 0}_{k-1}, \underbrace{1, \cdots, 1}_{L-k+1}]$. Fig. 5 depicts an example of the decomposition with $L = 4$.

Note that the motivation of the following discussion is to relate the rendering distortion into gradients and disparity/position errors, instead of computing them one pixel by one pixel. This simplifies estimation and captures the relationship between rendering distortion and gradients. Given (26) and (27), the squared error between $\vec{S}_L$ and $\vec{S}'_L$ is

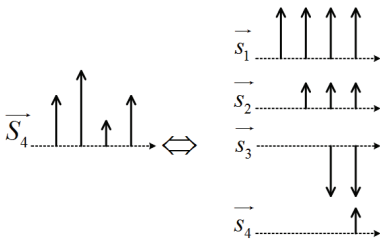


Fig. 5. Example of the decomposition of a SV region into gradient-based component-vectors. The extent of the SV region, L, is 4 in this example. Heights of the entries (arrows) in $\vec{S}_4$ are the *pixel values* in the SV region (figure on the left), whereas height of the non-zero entires in $\vec{s}_k$ is the *gradient value* at the k^{th} location in the SV region (figure on the right).

$$\begin{aligned} ||E_S||_2^2 &= ||\vec{S}_L - \vec{S}'_L||_2^2 \\ &= \sum_{i=1}^{L} \left(\vec{S}_L(i) - \vec{S}'_L(i) \right)^2 \\ &= \sum_{k=1}^{L} \sum_{i=1}^{L} (\vec{s}_k(i) - \vec{s}'_k(i))^2 + 2 \sum_{k=1}^{L-1} \sum_{l=k+1}^{L} \sum_{i=1}^{L} (\vec{s}_k(i) - \vec{s}'_k(i))(\vec{s}_l(i) - \vec{s}'_l(i)). \end{aligned} \tag{28}$$

The detailed derivation of (28) is shown in Appendix B. Let us denote $\vec{e_k} = \vec{s_k} - \vec{s'_k}$ as the *error vector* for the k^{th} gradient-based component-vector, (28) can be rewritten as

$$||E_S||_2^2 = \sum_{k=1}^{L} ||\vec{e_k}||_2^2 + 2\sum_{k=1}^{L-1}\sum_{l=k+1}^{L} \vec{e_k} \cdot \vec{e_l}. \tag{29}$$

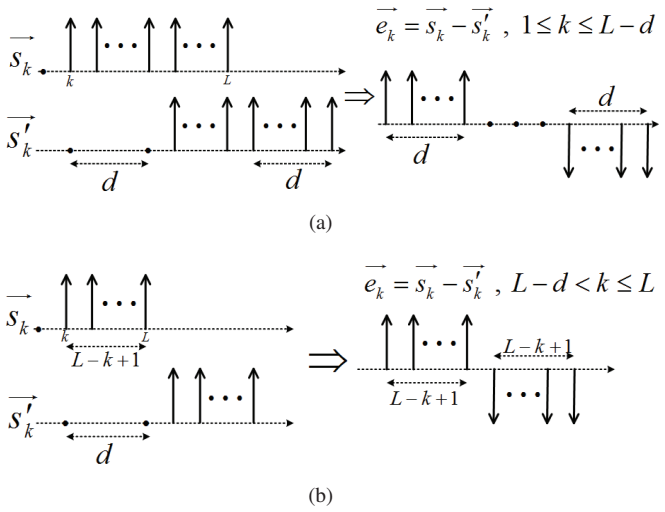


Fig. 6. (a) rendering error for the k^{th} gradient-based component-vector when $1 \le k \le L-d$ (b) rendering error for the k^{th} gradient-based component-vector when $L-d < k \le L$. In (a), supports of $\vec{s_k}$ and $\vec{s'_k}$ overlap. In (b), supports of $\vec{s_k}$ and $\vec{s'_k}$ are disjoint. Note that the non-zero entries in $\vec{s_k}$ and $\vec{s'_k}$ are the same, since they are the decompositions of $\vec{S_L}$ and $\vec{S'_L}$ respectively, and $\vec{S'_L}$ is the shifted counterpart of $\vec{S_L}$.

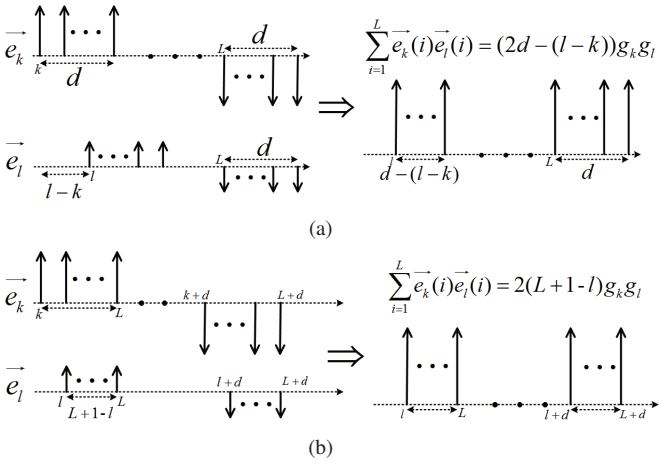


Fig. 7. (a) rendering error of $\vec{e_k} \cdot \vec{e_l}$ when $1 \le k \le L-d$ (b) rendering error of $\vec{e_k} \cdot \vec{e_l}$ when $L-d < k \le L$.

The first term of (29) is given by

$$\begin{aligned} ||\vec{e_k}||_2^2 &= \sum_{i=1}^{L} (\vec{s_k}(i) - \vec{s'_k}(i))^2 \\ &= \begin{cases} 2dg_k^2 & \text{for } k = 1, 2\cdots, L-d; \\ 2(L-k+1)g_k^2 & \text{for } k = L-d+1, \cdots, L, \end{cases} \end{aligned} \tag{30}$$

where d is the average position error for this spatial variant region. The two cases that $k = 1, 2\cdots, L-d$ and $k = L-d+1, \cdots, L$ are illustrated in Fig. 6(a) and 6(b) respectively. With a position error d, the supports of $\vec{s_k}$ and $\vec{s'_k}$ overlap for $k = 1, 2\cdots, L-d$ (Fig. 6(a)). On the other hand, the supports of $\vec{s_k}$ and $\vec{s'_k}$ are disjoint for $k = L-d+1, \cdots, L$ (Fig. 6(b)). These lead to different ways to calculate the error vector magnitude in (30).

Similarly, the second term of (29) can be derived from (30) and Fig. 6, as illustrated in (31) and Fig. 7 respectively,

$$\vec{e_k} \cdot \vec{e_l} = \begin{cases} (2d-(l-k))g_k g_l & \text{for } k = 1, 2\cdots, L-d; \\ 2(L+1-l)g_k g_l & \text{for } k = L-d+1, \cdots, L. \end{cases} \tag{31}$$

Substituting (30) and (31) into (29) and representing it in matrix form (see Appendix C), we have

$$||E_S||_2^2 = 2\sum_{i,j} (\mathbf{D} \circ \mathbf{G})_{ij} = 2\sum_{i,j} (\mathbf{D})_{ij} (\mathbf{G})_{ij}, \tag{32}$$

where "$\circ$" represents the *Hadamard product* or *element-wise multiplication* of two matrices. $\mathbf{G}$ is the matrix related to g (the gradient of SV regions in texture map), as shown in (50) of Appendix C). $\mathbf{D}$ is the matrix related to d (the depth-error-induced position error), as shown in (52) of Appendix C).

As noted in [21] and following the law of large numbers, when the number of samples is large, the average value of all the samples approximates their expectation. Therefore, $E[Z_{\text{SV}}^2]$ can be approximated by the MSE of the rendering distortions in SV regions,

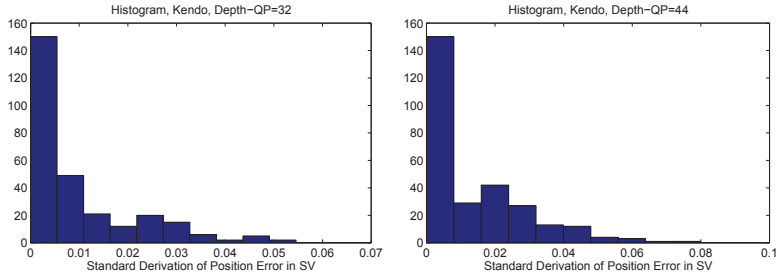
$$E[Z_{l,\text{SV}}^2] = \frac{1}{MN} \sum_{S \in \text{SV}} ||E_S||_2^2, \tag{33}$$

where $M \times N$ is the spatial dimension of a video frame. (33) can be used in (17) to estimate the overall distortion caused by depth errors.

Note that we use the average position error d of a SV region instead of per-pixel position errors to estimate the distortion, in order to simplify the computation. This can be justified by the fact that L (the extent of a SV region) is usually small, e.g., see Fig. 3(b). Furthermore, to illustrate that the average position error is a reasonable approximation to per-pixel position errors, we measure the Standard Derivation (SD) of the position errors in each of the Spatial Variant (SV) regions. Histograms of SD are shown in Figure 8. As shown in the figure, SD tends to be very small (less than 0.08; SD of many SV regions is almost zero), indicating that per-pixel position errors tend to be very close to the average position error of a SV region. This occurs because the change in position-error per unit change in depth-map-error can be computed via (24) (when $D_l - \hat{D}_l = 1$) and is very small (see Table II). This is true as long as the camera distance b_l is small (see (24)).

TABLE II
CHANGE IN POSITION-ERROR PER UNIT CHANGE IN DEPTH-MAP-ERROR FOLLOWING THE MPEG 3DV 2-VIEW TEST CASES [16]

Video Sequence	Change in Position-error
Kendo	0.094118
Ballons	0.094118
Poznan_Hall2	0.200000
Poznan_Street	0.154902



(a) (b)

Fig. 8. Histogram of the standard derivation (SD) of the position errors in Spatial Variant (SV) regions of the Kendo sequence: (a) Depth QP=32, (b) Depth QP=44. Note that SD tends to be very small, indicating that per-pixel position errors tend to be very close to the average position error of a SV region.

3) Simplified equation to compute the rendering distortions in SV regions: Here we discuss how to simplify (32) to compute $||E_S||_2^2$. A SV region consists of pixels around a sharp edge and the extent (width) of a SV region is usually small (e.g., see Fig. 3(b)). That is, L is very small for a typical $\vec{S_L}$. Thus, it is reasonable to use linear approximation to approximate the L pixel values in a SV region (we will argue that such approximation causes negligible discrepancy). Specifically, we approximate the gradient values (g_k, $k = 1, 2, \cdots, L$) in the spatial variant regions with the mean of all the g_k,

$$g_0 = \frac{1}{L}\sum_{k=1}^{L} g_k. \tag{34}$$

The gradient-based component-vectors $\vec{s_k}$ and the texture SV region $\vec{S_L}$ are approximated by $\vec{t_k}$ and $\vec{T_L}$ respectively,

$$\vec{t_k} = g_0\vec{1}_k, \tag{35}$$

and

$$\vec{T_L} = \sum_{k=1}^{L}\vec{t_k}. \tag{36}$$

The corresponding rendering distortion for $\vec{T_L}$, given an average horizontal disparity error d, is simply

$$\begin{aligned} ||E_T||_2^2 &= 2g_0^2\sum_{i,j}(\mathbf{D})_{ij} \\ &= \begin{cases} (-\frac{d^3}{3} + L^2 d + Ld + \frac{d}{3})g_0^2 & \text{for } d \le L; \\ L(L+1)g_0^2 & \text{otherwise.} \end{cases} \end{aligned} \tag{37}$$

We use (37) in lieu of (32) to estimate the rendering distortions in SV regions. Clearly, (37) requires negligible computation complexity.

4) Discrepancy analysis: Here we assess the discrepancy between using $\vec{S_L}$ and the approximation $\vec{T_L}$, i.e., discrepancy when average gradient g_0 is used in lieu of g_k, $k = 1, 2, \cdots, L$. Using g_k we obtain the distortion vector $E_S = \vec{S_L} - \vec{S'_L}$, while using approximation g_0 we obtain the distortion vector $E_T = \vec{T_L} - \vec{T'_L}$. We argue that the norm squared of the discrepancy, i.e., $||E_S - E_T||_2^2$, shall be much smaller than the estimation $||E_S||_2^2$.

$$\begin{aligned} ||E_S - E_T||_2^2 &= ||(\vec{S_L} - \vec{S'_L}) - (\vec{T_L} - \vec{T'_L})||_2^2 \\ &= ||(\vec{S_L} - \vec{T_L}) - (\vec{S'_L} - \vec{T'_L})||_2^2 \\ &= ||\vec{\Delta_L} - \vec{\Delta'_L}||_2^2, \end{aligned} \tag{38}$$

where

$$\vec{\Delta_L} = \vec{S_L} - \vec{T_L} = \sum_{k=1}^{L}(\vec{s_k} - \vec{t_k}) = \sum_{k=1}^{L}\delta_k\vec{1}_k, \tag{39}$$

and

$$\delta_k = g_k - g_0, \tag{40}$$

where δ_k is the deviation of the gradient g_k from its average g_0. Following (28), except that we replace $\vec{S_L}$ with $\vec{\Delta_L}$, and $\vec{S'_L}$ with $\vec{\Delta'_L}$, and the result in (32) except that the g_k is replaced by δ_k, we have

$$||E_S - E_T||_2^2 = 2\sum_{i,j}(\mathbf{D} \circ \mathbf{G}_\Delta)_{ij}, \tag{41}$$

where $\mathbf{G}_\Delta = [\delta_1, \delta_2, \cdots, \delta_L]^T[\delta_1, \delta_2, \cdots, \delta_L]$. Let $\delta_{\max} = \max_k\{|\delta_k|\} = \max_k\{|g_k - g_0|\}$ denote the maximum variation between g_k and the approximation g_0 (the average of g_k), and $g_{\min} = \min_k\{|g_k|\}$ as the minimal gradient value. Recall that in a SV region, all the gradient values are greater than some threshold. Therefore, the gradient value itself is in general much larger than the variation from the average, i.e., $|g_k| \gg |\delta_k|$. We have

$$\begin{aligned} ||E_S - E_T||_2^2 &= 2\sum_{i,j}(\mathbf{D} \circ \mathbf{G}_\Delta)_{ij} \\ &< 2\delta_{\max}^2\sum_{i,j}(\mathbf{D})_{ij} \\ &\ll 2g_{\min}^2\sum_{i,j}(\mathbf{D})_{ij} \\ &< ||E_S||_2^2 \end{aligned} \tag{42}$$

In other words, the norm squared of the discrepancy between E_S and the approximation E_T is much smaller than the norm squared of the rendering error itself (E_S). Therefore, the approximation of g_0 is a reasonable choice to reduce the computational complexity without causing much estimation discrepancy.

V. Model Summary

Here we summarize the modeling process, which estimates the noise power in the synthesis output from X_l, X_r, $\hat{X}_l$, $\hat{X}_r$, D_l, D_r, $\hat{D}_l$, $\hat{D}_r$ analytically. First, mean squared errors (MSEs) between X_l and $\hat{X}_l$, and between X_r and $\hat{X}_r$, are computed and used in (8) to determine $E[N^2]$. The Otsu's algorithm is then applied to the gradient maps of X_l and X_r to determine the spatial invariant and spatial variant regions. Correlation between adjacent pixels in SI regions is computed

to determine the parameter for the parametric PSD $\Phi_{\hat{X}_{l,SI}}$ (see (23)). This parametric PSD along with $P(\omega_1)$ (1-D FFT of $p(\Delta m_l)$) are used in (21) to compute the distortion spectrum $\Phi_{Z_{l,\mathrm{SI}}}(\omega_1,\omega_2)$ of the SI regions. We integrate $\Phi_{Z_{l,\mathrm{SI}}}(\omega_1,\omega_2)$ to obtain $E[Z^2_{l,\mathrm{SI}}]$. For the SV regions, we apply (37) to compute the individual distortions, and (33) to obtain $E[Z^2_{l,\mathrm{SV}}]$. $E[Z^2_{l,\mathrm{SI}}]$ and $E[Z^2_{l,\mathrm{SV}}]$ are summed to obtain $E[Z^2_l]$. $E[Z^2_r]$ can be estimated in a similar way. $E[Z^2_l]$ and $E[Z^2_r]$ are then combined to obtain $E[Z^2]$ following (13). Finally, $E[N^2]$ and $E[Z^2]$ are summed to obtain the overall synthesis distortion power $E[V^2]$ following (3).

VI. Experiments

We have performed experiments to verify the accuracy of the proposed models. Following the camera configurations in the MPEG 3DV 2-view test cases [16], two reference views were used to render a virtual view in-between. Both the texture and depth videos were encoded with JMVC Encoder 8.3.1. Each group-of-pictures consisted of an anchor frame and several hierarchically coded B frames. Inter-view prediction was also used in encoding. Quantization parameters (qp) were set to be 32, 36, 40 and 44 for both texture and depth image encoding. VSRS 3.5 [17] was used to synthesis the virtual view, with the merging method chosen to be averaging[4].

We also compare our model with Yuan's model [21] in the following experiments, regarding accuracy, complexity and convenience for applications. Yuan et al. proposed a linear model that relates synthesized view distortion with quantization steps of the texture/depth videos. Their model parameters are specific to video/scene characteristics and particular virtual view positions. To estimate the model parameters, view synthesis is required [21]. While they also have some analysis in frequency domain, their approach and our approach are very different. Both Yuan's model and our model are inspired by previous work on efficiency analysis of motion compensation in conventional video coding. However, Yuan's model takes the approach proposed by Taubman [20] [29] and we take the approach proposed by Girod [14]. The principles and mathematical details of them are very different, and consequently we also have very different model and analysis results from [21].

Figures 9, 10, 11 and 12 compare the empirical results and the model results using our model and Yuan's model [21] for video sequences Kendo (1024×768), Balloons (1024×768), PoznanHall2 (1920×1088) and Poznan-Street (1920×1088). The empirical results were measured from the rendering output of VSRS. Our model results were computed following the discussion in Sections II, III and IV. For Kendo, we also show the results for texture qp (color_qp) of 24 to highlight the effects of different texture image quality. As shown in the figures, our model can accurately estimate the rendering quality with different encoding conditions and situations. However, there is some gap between Yuan's estimation and the empirical data[5].

We further compare Yuan's model and our model regarding complexity and convenience for applications. In Yuan's model, model parameters are specific to video/scene characteristics and particular virtual view positions. In applications where the video contents/characteristics remain similar and the virtual view position is fixed and predefined, their model requires low complexity and is convenient for computing rough estimates of the rendering distortion. However, in practical applications where the video contents may change frequently, their model requires re-estimation of the model parameters, and this necessitates frequent re-synthesis of virtual views to calculate the distortion data points for computing the model parameters. On the contrary, our proposed model does not require any view synthesis, and is flexible regarding change in video contents and virtual view positions. Specifically, in our SI/SV signal analysis, we use gradient field of the texture image, which can be easily computed. We use a parametric PSD model which is parameterized by the correlation coefficients of the pixels. Also, estimating the distortion at a different virtual view position requires only updating the distribution of the position errors (in SI signals analysis) and the average position errors (in SV signals analysis). These updating operations require low complexity, as depth errors and position errors are related linearly with the camera setup factor k_l (24), and only k_l would change for a different virtual view position.

From the experiment results in Figures 9, 10, 11 and 12, it is suggested that, with lower quality texture images (e.g., color_qp $= 44$ in Fig. 9(e)), only small gains in the rendering output can be obtained with improving the quality of the depth images (reducing depth_qp). This is because with lower quality texture images the noise due to texture coding $E[N^2]$ dominates the overall synthesis noise power in (3), and reduction in $E[Z^2]$ has only a small impact. On the other hand, when the texture images have good quality (e.g., color_qp $= 24$ in Fig. 9(a)), large gains in the rendering quality can be obtained with improving the quality of the depth images (reducing depth_qp). Another observation is that in the case of high texture quality (small color_qp) and low depth quality (large depth_qp), e.g., Fig 9(a) with depth_qp $= 44$, the accuracy of the modeling results appears slightly worse compared to other cases. This is because with high quality texture images the noise due to depth coding $E[Z^2]$ dominates the overall synthesis noise power in (3), and a slight distortion in modeling of depth error ($E[Z^2]$) may have strong impact on the overall modeling accuracy.

To further illustrate the characteristics of rendering distortion caused by texture error ($E[N^2]$) and depth error ($E[Z^2]$), we plot $E[N^2]$ and $E[Z^2]$ respectively in Fig. 13. Due to

[4]In our experiment, the reference image for MSE / PSNR calculation is the synthesis output using uncompressed texture / uncompressed depth of adjacent views. This follows the experiment practice in MPEG 3DV. Alternatively, an existing view directly captured by camera can be used as reference for comparison. However, we noticed that empirical data using these two reference images (the one synthesized using uncompressed texture / uncompressed depth of adjacent views, and the one captured by camera at the same view angle) are rather different. This could be due to some small errors in camera calibration, and such differences in cameras have not been considered in view synthesis. Our current method uses characteristics of the adjacent views to estimate the synthesized view quality and has not considered such differences. Our future work will investigate extended models that consider other issues such as imperfect camera calibration in addition to view position difference.

[5]Note that Yuan's focus is on bit allocation and their work seems to suggest that bit allocation may tolerate some model inaccuracy.

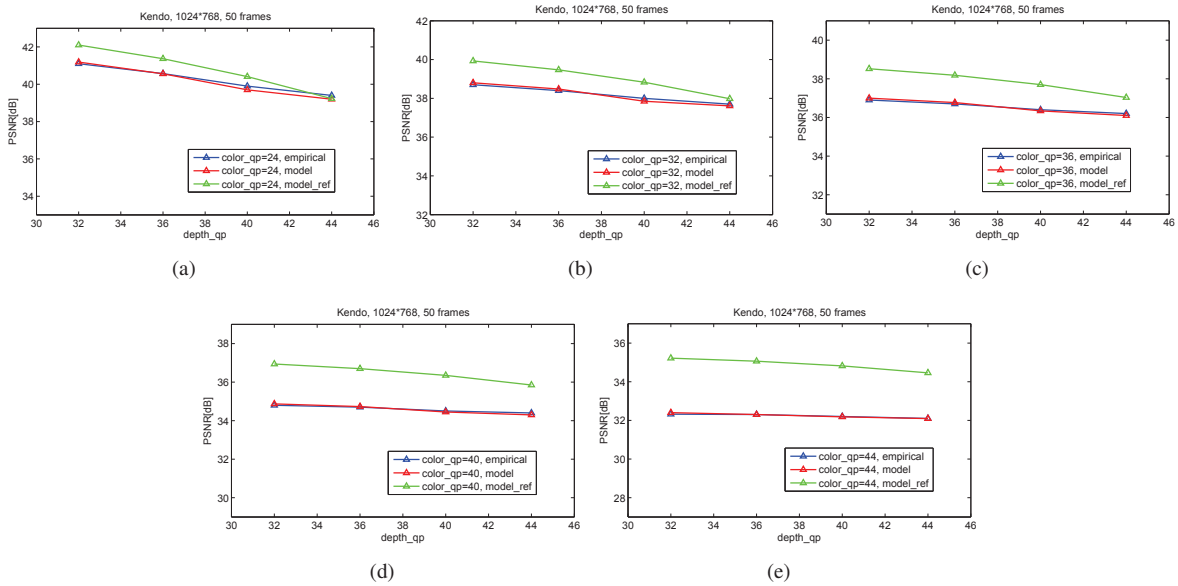


Fig. 9. Modeling result: Kendo. Results are shown for texture images encoded with different qp (color_qp). The horizontal axis represents the qp used in encoding the depth images (depth_qp), and the vertical axis represents the rendering quality. "empirical": synthesis quality using VSRS; "model": estimation using our model; "model_ref": estimation using Yuan's model.

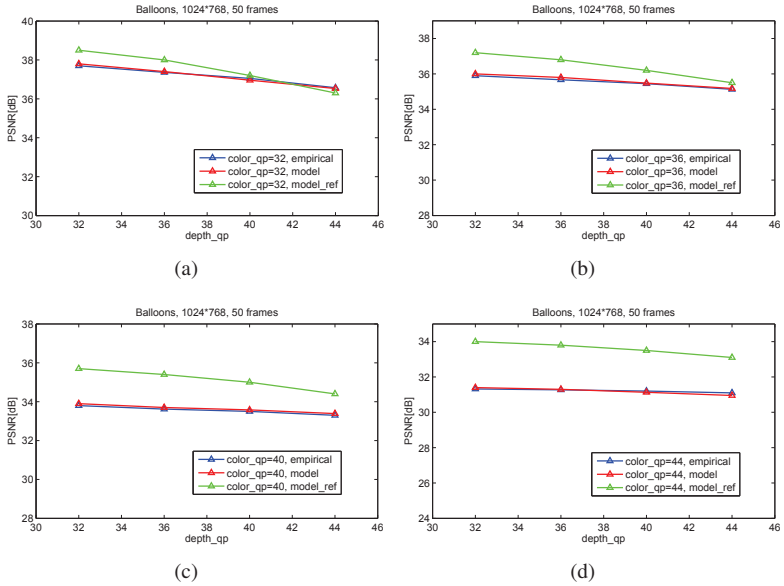


Fig. 10. Modeling result: Balloons. "empirical": synthesis quality using VSRS; "model": estimation using our model; "model_ref": estimation using Yuan's model.

limited space, we merely show the typical empirical results and the modeling results for sequence Kendo in the cases that texture qp $= 32$ and qp $= 44$. As we expect, the empirical and model results of $E[N^2]$ remain unchanged for different depth image quality, as depicted in Fig. 13(a) and (c). Another observation is that a large texture qp (color_qp $= 44$) causes large texture-error induced distortion $E[N^2]$ (MSE is over 34, see Fig. 13(c)), while depth-error induced distortion $E[Z^2]$ is relatively small (MSE is less than 6, see Fig. 13(d)). Such large $E[N^2]$ dominates the final rendering quality, resulting in a relatively small change in rendering quality at different depth map quality (see Fig. 9(e)). On the other hand, for a smaller color qp (color_qp $= 32$), the magnitude of the texture-error induced distortion (MSE is around 6) is comparable to the one caused by depth error. Therefore, the total rendering quality would be equally affected by both texture error and depth error, and we can observe noticeable variation in rendering quality at different depth map quality (see Fig. 9(b)).

We have also implemented the cubic distortion model proposed in [24] and comparison results are depicted in Figure 13. As depicted in Figure 13 (a), the cubic model in [24] introduces somewhat non-negligible inaccuracy during the estimation of texture-error-induced rendering distortion ($E[N^2]$). In estimating the depth-error-induced rendering distortion ($E[Z^2]$), the accuracy of the cubic model is close to our approach as shown in Figure 13 (b). One of the reasons is that

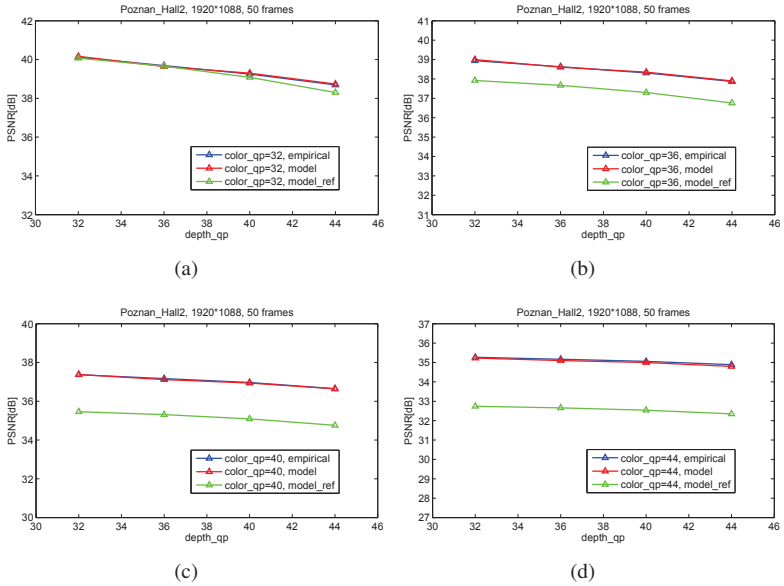


Fig. 11. Modeling result: PoznanHall2. "empirical": synthesis quality using VSRS; "model": estimation using our model; "model_ref": estimation using Yuan's model.

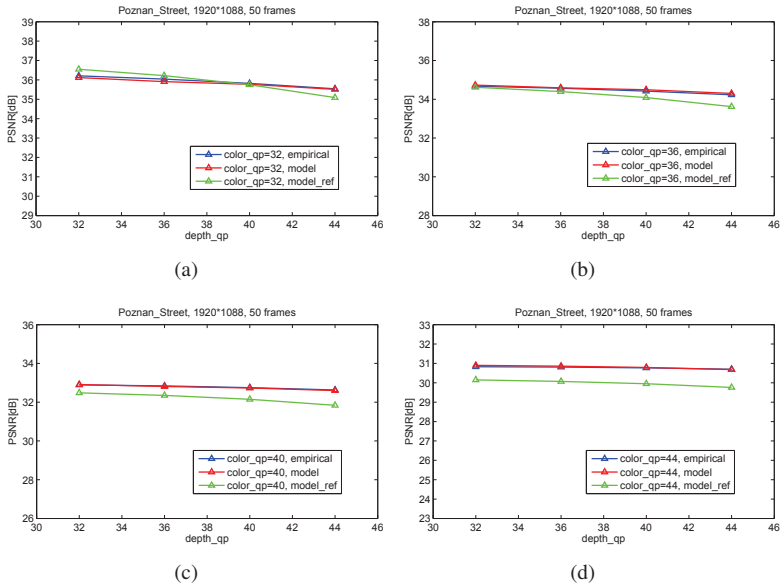
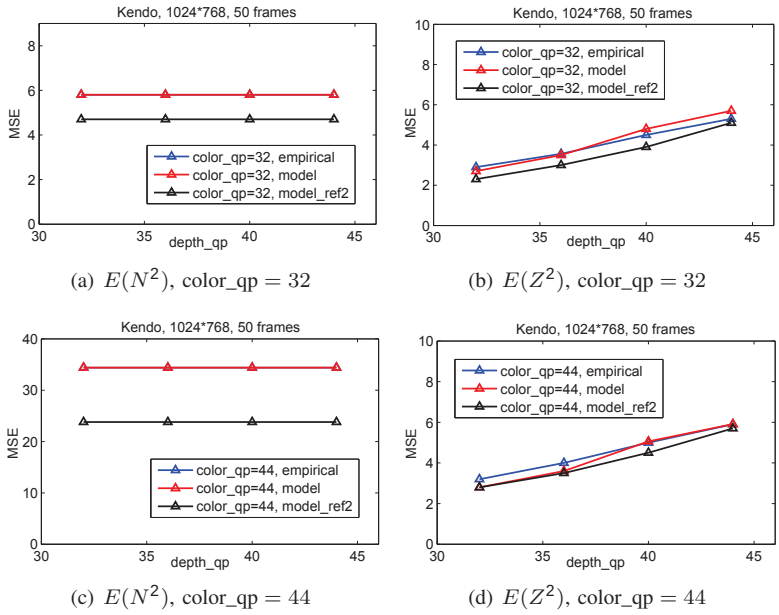


Fig. 12. Modeling result: Poznan Street. "empirical": synthesis quality using VSRS; "model": estimation using our model; "model_ref": estimation using Yuan's model.

the cubic model assumes independent zero-mean quantization error in the left and right reference texture images. In other words, ρ_N, the correlation coefficient between $X_l - \hat{X}_l$ and $X_r - \hat{X}_r$ in our approach, is assumed to be zero in their work. We found that the independence assumption may not be valid in some cases. In our work, we trained a model to characterize the correlation between the quantization errors, and this leads to better estimation accuracy. Examining Figure 13 (c), when texture-QP increases, the discrepancy between the empirical and the cubic modeling results for $E[N^2]$ becomes more pronounced. When texture images are encoded at lower quality (with higher texture-QP), there would be more structural information remained in $X_l - \hat{X}_l$ and $X_r - \hat{X}_r$, and the quantization errors tend to be more correlated. In this case, the effect caused by the assuming $\rho_N = 0$ will be more pronounced and thus the estimation of $E[N^2]$ tends to perform worse in the cubic model of [24].

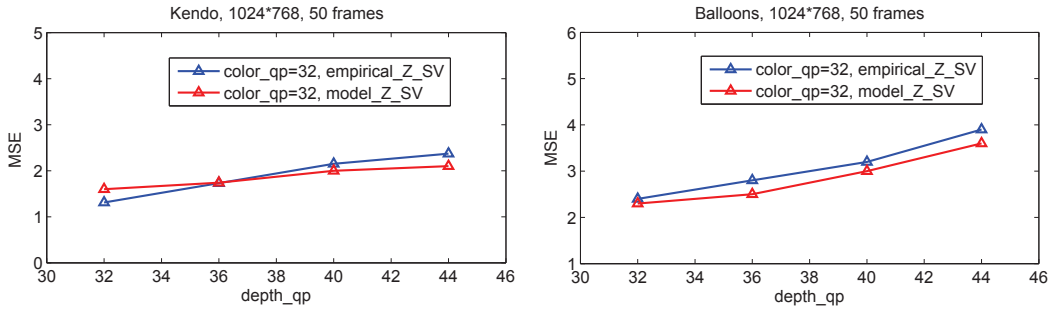
In order to simplify the estimation of rendering error, we approximate the per-pixel position errors using the average position error for a SV region of length L. Note that L is usually small. Here we estimate the specific portion of rendering error in SV regions using our model (i.e., with average position errors) and compare it with the corresponding empirical rendering error (i.e., with per-pixel position errors). As shown in Figure 14, the comparison suggests that the estimation of rendering distortion using average position errors approximates well the empirical rendering distortion using per-pixel position errors (we observed similar results in other



(a) $E(N^2)$, color_qp $= 32$ (b) $E(Z^2)$, color_qp $= 32$

(c) $E(N^2)$, color_qp $= 44$ (d) $E(Z^2)$, color_qp $= 44$

Fig. 13. Modeling result of Kendo: (a) rendering distortion caused by texture error ($E[N^2]$) when texture color_qp $= 32$ (b) rendering distortion caused by depth error ($E[Z^2]$) when color_qp $= 32$ (c) rendering distortion caused by texture error ($E[N^2]$) when color_qp $= 44$ (d) rendering distortion caused by depth error ($E[Z^2]$) when color_qp $= 44$. "empirical" represents empirical measurements, "model" represents modeling results based on our approach, "model_ref2" represents cubic modeling results using [24]. Note that in Figures (a) and (c) our results overlap with the empirical measurements.



(a) Kendo (b) Balloons

Fig. 14. Rendering distortion induced by depth errors (i.e., $E[Z^2]$) in Spatial Variant (SV) regions of the Kendo and Balloons sequences: the BLUE curve (empirical) represents the empirical MSE computed with per-pixel position errors; the RED curve (model) represents the estimated MSE using average position errors.

testing sequences). Therefore, the approximation using average position error in SV regions seems to be reasonable and effective.

VII. Conclusions

We have proposed an analytical model to estimate the synthesized view quality in 3D video. The model relates errors in the depth images to the rendering quality, taking into account texture image characteristics, texture image quality and the rendering process. We decoupled the estimation of the power of the synthesis distortion into two steps, one focusing on the texture error induced distortion, and the other focusing on the depth error induced distortion. We showed that the PSD of the rendering distortion due to depth coding is the product of the PSD of texture data and a frequency envelope depending on the probability distribution of the position errors. We also derived equations to estimate the rendering distortions in spatial variant regions along strong edges. Experiment results showed that the model can accurately estimate the synthesis noise power. The model can be used to predict the rendering quality for different system designs. The analysis can also help inform designs of coding, transmission and rendering systems for 3D video.

VIII. Acknowledgment

The authors would like to thank the Associate Editor Dr. Dimitrios Tzovaras and two anonymous reviewers for their valuable comments. This work was supported in part by Natural Science Foundation of China (NSFC) under contract No. 61303151.

Appendix A

Here we present a detailed derivation of (18). From (14),

$$\begin{aligned} Z_l(m,n) &= Y_l(m,n) - W_l(m,n) \\ &= Y_l(m,n) - Y_l(m-\Delta m_l, n). \end{aligned} \tag{43}$$

Apply discrete-time Fourier transform to (43), we obtain:

$$\mathcal{Z}_l(\omega_1,\omega_2) \quad = \quad \mathcal{Y}_l(\omega_1,\omega_2) - \mathcal{Y}_l(\omega_1,\omega_2)e^{-j\Delta m_l\cdot\omega_1}. \tag{44}$$

Taking the squared magnitude of both sides:

$$\begin{aligned} |\mathcal{Z}_l(\omega_1,\omega_2)|^2 &= |\mathcal{Y}_l(\omega_1,\omega_2) - \mathcal{Y}_l(\omega_1,\omega_2)e^{-j\Delta m_l\cdot\omega_1}|^2 \\ &= |\mathcal{Y}_l(\omega_1,\omega_2)|^2|1-e^{-j\Delta m_l\cdot\omega_1}|^2. \end{aligned} \tag{45}$$

The second term of RHS of (45) can be simplified as:

$$\begin{aligned} |1-e^{-j\Delta m_l\cdot\omega_1}|^2 &= |1-\cos(\Delta m_l\cdot\omega_1)+j\sin(\Delta m_l\cdot\omega_1)|^2 \\ &= 2-2\cos(\Delta m_l\cdot\omega_1). \end{aligned} \tag{46}$$

PSD is the square of the magnitude of the frequency representation. Therefore, from (45), we obtain:

$$\Phi_{Z_{l,\mathrm{SI}}}(\omega_1,\omega_2) \quad = \quad 2(1-\cos(\Delta m_l\cdot\omega_1))\Phi_{Y_{l,\mathrm{SI}}}(\omega_1,\omega_2) \tag{47}$$

APPENDIX B

Here we present a detailed derivation of (28). Recall that $\vec{S_L}$ can be represented by $\vec{S_L} = \sum_{k=1}^{L}\vec{s_k}$ from (26) and Fig. 5, where $k = 1, 2, \cdots, L$ and $\vec{s_k}$ is the k^{th} gradient-based component-vector, given by $\vec{s_k} = g_k\vec{1}_k$. Then $||E_S||_2^2$ can be computed by (48). Note that in lines 5 to 7 we expand the squared of summation term in line 4.

APPENDIX C

Here we present a detailed derivation of (32). Substituting (30) and (31) into (29), we have (49).

Note that line 1 is followed from the derivation results of Appendix B. In line 2, we partition the summation. In line 3, $||\vec{e_k}||_2^2$ and $\vec{e_k}\cdot\vec{e_l}$ are substituted by values following the analysis in Figures 6, 7 and (30), (31).

Examining (49), $||E_S||_2^2$ is computed by weighted summation of the basic terms (i.e., g_k^2 and g_kg_l) with different weight coefficients. In other words, we can define a matrix $\mathbf{G}$ to represent all the basic terms,

$$\mathbf{G} = [g_1, g_2, \cdots, g_L]^T[g_1, g_2, \cdots, g_L]. \tag{50}$$

Similarly, the weight coefficients can be denoted as another matrix $\mathbf{D}$ based on only the average position error d of a SV region. Then, $||E_S||_2^2$ will be represented by operations on the two matrices,

$$||E_S||_2^2 \quad = 2\sum_{i,j}(\mathbf{D}\circ\mathbf{G})_{ij} = 2\sum_{i,j}(\mathbf{D})_{ij}(\mathbf{G})_{ij}, \tag{51}$$

where $\mathbf{D}$ is given by (52).

REFERENCES

[1] A. Vetro, W. Matusik, H. Pfister, and J. Xin, "Coding approaches for end-to-end 3D TV systems," in *Proc. Picture Coding Symposium (PCS)*, 2004.

[2] T. Maugey, P. Frossard, and G. Cheung, "Consistent view synthesis in interactive multiview imaging," in *Proc. IEEE Int'l Conf. Image Processing (ICIP)*, 2012.

[3] Z. Zhang, R. Wang, C. Zhou, Y. Wang, and W. Gao, "A compact stereoscopic video representation for 3D video generation and coding," in *Proc. IEEE Data Compression Conference (DCC)*, 2012.

[4] D. Min, J. Lu, and M. N. Do, "Depth video enhancement based on weighted mode filtering," *IEEE Trans. Image Processing*, vol. 21, no. 3, pp. 1176–1190, 2012.

[5] D. Min, D. Kim, S. Yun, and K. Sohn, "2D/3D freeview video generation for 3DTV system," *Signal Processing: Image Communication*, vol. 24, no. 1-2, pp. 31–48, 2009.

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

[7] Q. Wang, X. Ji, Q. Dai, and N. Zhang, "Free viewpoint video coding with rate-distortion analysis," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 22, no. 6, pp. 875–889, June 2012.

[8] P. Merkle, Y. Morvan, A. Smolic, D. Farin, K. Mueller, P. H. N., and T. Wiegand, "The effects of multiview depth video compression on multiview rendering," *Signal Process.: Image Commun*, vol. 24, pp. 73–88, 2009.

[9] D. Tian, A. Vetro, and M. Brand, "A trellis-based approach for robust view synthesis," in *ICIP-Proc*, 2011.

[10] I. Daribo, C. Tillier, and B. Pesquet-Popescu, "Adaptive wavelet coding of the depth map for stereoscopic view synthesis," in *Proc. IEEE Int'l Workshop on Multimedia Signal Processing (MMSP)*, 2008.

[11] A. Sanchez, G. Shen, and A. Ortega, "Edge-preserving depth-map coding using tree-based wavelets," in *Proc. Asilomar Conf. Signals, Systems, and Computers*, 2009.

[12] W.-S. Kim, A. Ortega, P. Lai, D. Tian, and C. Gomila, "Depth map distortion analysis for view rendering and depth coding," in *Proc. IEEE Int'l Conf. Image Processing (ICIP)*, 2009.

[13] N.-M. Cheung, D. Tian, A. Vetro, and H. Sun, "On modeling the rendering error in 3D video," in *Proc. IEEE Int'l Conf. Image Processing (ICIP)*, 2012.

[14] B. Girod, "The efficiency of motion-compensating prediction for hybrid coding of video sequences," *IEEE J. Selected Areas in Communications*, vol. 5, no. 7, pp. 1140–1154, 1987.

[15] P. Ramanathan and B. Girod, "Rate-distortion analysis for light field coding and streaming," *Signal Process.: Image Commun*, vol. 21, pp. 462–475, 2006.

[16] MPEG Video and Requirement Group, "Call for proposals on 3D video coding technology," Tech. Rep., MPEG, 2011, MPEG N12036.

[17] MPEG, "View synthesis software manual release 3.5 (VSRS 3.5)," Tech. Rep., ISO/IEC JTC1/SC29/WG11 MPEG, 2009.

[18] H. T. Nguyen and M. N. Do, "Error analysis for image-based rendering with depth information," *IEEE Trans. Image Processing*, vol. 18, no. 4, pp. 703–716, 2009.

[19] 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," *Image Commun.*, vol. 24, no. 8, pp. 666–681, Sept. 2009.

[20] A. Secker and D. S. Taubman, "Highly scalable video compression with scalable motion coding," *IEEE Trans. Image Processing*, vol. 13, no. 8, pp. 1029–1041, 2004.

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

[22] H. Yuan, J. Liu, H. Xu, Z. Li, and W. Liu, "Coding distortion elimination of virtual view synthesis for 3-d video system: Theoretical analyses and implementation," *IEEE Trans. on Broadcasting*, vol. 58, no. 4, pp. 558–567, Dec. 2012.

[23] W.-S. Kim, A. Ortega, P. Lai, D. Tian, and C. Gomila, "Depth map coding with distortion estimation of rendered view," in *Proc. SPIE Visual Information Processing and Communication (VIPC)*, 2010.

[24] V. Velisavljevic, G. Cheung, and J. Chakareski, "Bit allocation for multiview image compression using cubic synthesized view distortion model," in *Proc. IEEE International Workshop on Hot Topics in 3D*, 2011.

[25] K. Takahashi, "Theoretical analysis of view interpolation with inaccurate depth information," *IEEE Trans. Image Processing*, vol. 21, no. 2, pp. 718–732, 2012.

[26] Z. Ni, D. Tian, S. Bhagavathy, J. Llach, and B. S. Manjunath, "Improving the quality of depth image based rendering for 3D video systems," in *Proc. IEEE Int'l Conf. Image Processing (ICIP)*, 2009.

[27] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. on System, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, Jan. 1979.

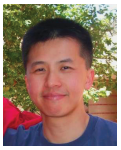
[28] J. B. O'Neal and T. R. Natarajan, "Coding isotropic images," *IEEE Trans. on Information Theory*, vol. 23, no. 6, pp. 697–707, Nov. 1977.

[29] R. Mathew and D. S. Taubman, "Quad-tree motion modeling with leaf merging," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 20, no. 10, pp. 1331–1345, Oct. 2010.

$$
\begin{aligned}
||E_S||_2^2 = & \ ||\vec{S_L} - \vec{S'_L}||_2^2 \\
= & \ \sum_{i=1}^{L} \left(\vec{S_L}(i) - \vec{S'_L}(i) \right)^2 \\
= & \ \sum_{i=1}^{L} \left(\sum_{k=1}^{L} \vec{s_k}(i) - \sum_{k=1}^{L} \vec{s'_k}(i) \right)^2 \\
= & \ \sum_{i=1}^{L} \left[\sum_{k=1}^{L} (\vec{s_k}(i) - \vec{s'_k}(i)) \right]^2 \\
= & \ \sum_{i=1}^{L} \left[(\vec{s_1}(i) - \vec{s'_1}(i)) + (\vec{s_2}(i) - \vec{s'_2}(i)) + \cdots + (\vec{s_L}(i) - \vec{s'_L}(i)) \right]^2 \\
= & \ \sum_{i=1}^{L} \Big[(\vec{s_1}(i) - \vec{s'_1}(i))^2 + (\vec{s_2}(i) - \vec{s'_2}(i))^2 \cdots + (\vec{s_L}(i) - \vec{s'_L}(i))^2 \\
+ & \ 2(\vec{s_1}(i) - \vec{s'_1}(i))(\vec{s_2}(i) - \vec{s'_2}(i)) + \cdots + 2(\vec{s_1}(i) - \vec{s'_1}(i))(\vec{s_L}(i) - \vec{s'_L}(i)) \\
+ & \ \cdots + 2(\vec{s_{L-1}}(i) - \vec{s'_{L-1}}(i))(\vec{s_L}(i) - \vec{s'_L}(i)) \Big] \\
= & \ \sum_{i=1}^{L} \left[\sum_{k=1}^{L} (\vec{s_k}(i) - \vec{s'_k}(i))^2 + 2 \sum_{k=1}^{L-1} \sum_{l=k+1}^{L} (\vec{s_k}(i) - \vec{s'_k}(i))(\vec{s_l}(i) - \vec{s'_l}(i)) \right] \\
= & \ \sum_{i=1}^{L} \sum_{k=1}^{L} (\vec{s_k}(i) - \vec{s'_k}(i))^2 + 2 \sum_{i=1}^{L} \sum_{k=1}^{L-1} \sum_{l=k+1}^{L} (\vec{s_k}(i) - \vec{s'_k}(i))(\vec{s_l}(i) - \vec{s'_l}(i)) \\
= & \ \sum_{k=1}^{L} \sum_{i=1}^{L} (\vec{s_k}(i) - \vec{s'_k}(i))^2 + 2 \sum_{k=1}^{L-1} \sum_{l=k+1}^{L} \sum_{i=1}^{L} (\vec{s_k}(i) - \vec{s'_k}(i))(\vec{s_l}(i) - \vec{s'_l}(i)).
\end{aligned}
\tag{48}
$$

$$
\begin{aligned}
||E_S||_2^2 = & \ \sum_{k=1}^{L} ||\vec{e_k}||_2^2 + 2 \sum_{k=1}^{L-1} \sum_{l=k+1}^{L} \vec{e_k} \cdot \vec{e_l} \\
= & \ \sum_{k=1}^{L-d} ||\vec{e_k}||_2^2 + \sum_{k=L-d+1}^{L} ||\vec{e_k}||_2^2 + 2 \sum_{k=1}^{L-d} \sum_{l=k+1}^{L} \vec{e_k} \cdot \vec{e_l} + 2 \sum_{k=L-d+1}^{L-1} \sum_{l=k+1}^{L} \vec{e_k} \cdot \vec{e_l} \\
= & \ 2 \sum_{k=1}^{L-d} d g_k^2 + 2 \sum_{k=L-d+1}^{L} (L-k+1) g_k^2 \\
& + 2 \sum_{k=1}^{L-d} \sum_{l=k+1}^{L} (2d - (l-k)) g_k g_l + 2 \sum_{k=L-d+1}^{L-1} \sum_{l=k+1}^{L} 2(L+1-l) g_k g_l
\end{aligned}
\tag{49}
$$

Lu Fang received the B.S. degree in 2007 from University of Science and Technology of China (USTC) and Ph.D. degree in 2011 from Hong Kong University of Science and Technology (HKUST). She used to visit Northwestern University under the support of Professor Aggelos K. Katsaggelos in 2010. From 2011 to 2012, she was the post-doc research fellow in HKUST and Singapore University of Technology and Design (SUTD) respectively. She is currently an Associate Professor in USTC, with research interests in Multimedia Processing, Image and Video Coding, Machine Learning etc.



Ngai-Man Cheung is an Assistant Professor with Singapore University of Technology and Design (SUTD). He received his Ph.D. degree in Electrical Engineering from University of Southern California (USC), Los Angeles, CA, in 2008. From 2009 to 2011, he was a postdoctoral researcher with the Image, Video and Multimedia Systems group at Stanford University, Stanford, CA. He has also held research positions with Texas Instruments Research Center Japan, Nokia Research Center, IBM T. J. Watson Research Center, HP Labs Japan, Hong Kong University of Science and Technology (HKUST) and Mitsubishi Electric Research Labs (MERL). His research interests are image and video processing, signal processing and multimedia communication.

$$\mathbf{D} = \begin{bmatrix} d & 2d-1 & 2d-2 & \cdots & d+1 & \cdots & d & d-1 & \cdots & 1 \\ 0 & d & 2d-1 & 2d-2 & \cdots & d+1 & \cdots & d-1 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & d & 2d-1 & 2d-3 & 2d-5 & \cdots & 3 & 1 \\ 0 & \cdots & 0 & 0 & d & 2d-2 & 2d-4 & \cdots & 4 & 2 \\ 0 & \cdots & 0 & 0 & 0 & d-1 & 2d-4 & \cdots & 4 & 2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 3 & 4 & 2 \\ 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}_{L \times L} . \quad (52)$$

Dong Tian received the Ph.D. degree at Beijing University of Technology in 2001, and the M.Eng. and B.Eng. degrees on automation from the University of Science and Technology of China (USTC) in 1998 and 1995, respectively. He is currently a Senior Principal Member Research Staff in the Multimedia Group of Mitsubishi Electric Research Laboratories (MERL) at Cambridge, MA. Prior to joining MERL, he worked with Thomson Corporate Research at Princeton, NJ for over 4 years, where he was devoted to AVC encoder optimization and 3D video coding/processing projects, especially to the standards of Multiview Video Coding (MVC) and later on 3D Video (3DV) within MPEG. Before that, he spent 4 years (from Jan 2002 to Dec 2005) at Tampere University of Technology in Finland as a Post-Doc researcher for a Nokia funded project and made contributions on the standardization of MPEG-4 AVC /H.264 and its application for mobile applications. He currently mainly conducts researches on image/video coding and processing. He is a member of IEEE.

Anthony Vetro (S'92-M'96-SM'04-F'11) received the B.S., M.S., and Ph.D. degrees in electrical engineering from Polytechnic University, Brooklyn, NY. He joined Mitsubishi Electric Research Labs, Cambridge, MA, in 1996, where he is currently a Group Manager responsible for research and standardization on video coding, as well as work on display processing, information security, sensing technologies, and speech/audio processing. He has published more than 150 papers in these areas. He has also been an active member of the ISO/IEC and ITU-T standardization committees on video coding for many years, where he has served as an ad-hoc group chair and editor for several projects and specifications. He was a key contributor to the Multiview Video Coding extension of the H.264/MPEG-4 AVC standard, and current serves as Head of the U.S. delegation to MPEG. Dr. Vetro is also active in various IEEE conferences, technical committees, and editorial boards. He currently serves as an Associate Editor for IEEE TRANSACTIONS ON IMAGE PROCESSING, and as a member of the Editorial Boards of IEEE MultiMedia and IEEE JOURNAL ON SELECTED TOPICS IN SIGNAL PROCESSING. He served as Chair of the Technical Committee on Multimedia Signal Processing of the IEEE Signal Processing Society and on the steering committees for ICME and the IEEE TRANSACTIONS ON MULTIMEDIA. He served as an Associate Editor for IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY (2010-2013) and IEEE Signal Processing Magazine (2006-2007) and, and later served as a member of the Editorial Board (2009-2011). He also served as a member of the Publications Committee of the IEEE TRANSACTIONS ON CONSUMER ELECTRONICS (2002-2008). He has also received several awards for his work on transcoding, including the 2003 IEEE Circuits and Systems CSVT Transactions Best Paper Award. He is a Fellow of IEEE.

Huifang Sun received the B.S. degree in Electrical Engineering from Harbin Engineering Institute (Harbin Engineering University now), Harbin, China in 1967, and the Ph.D. degree in Electrical Engineering from University of Ottawa, Ottawa, Canada. In 1986 he jointed Fairleigh Dickinson University, Teaneck, New Jersey, as an Assistant Professor and promoted to an Associate Professor in 1990. From 1990 to 1995 he was with the David Sarnoff Research Center (Sarnoff Corp), Princeton, New Jersey, as a member of technical staff and later promoted to Technology Leader of Digital Video Technology. He joined Mitsubishi Electric Research Laboratories (MERL), in 1995 as a Senior Principal Technical Staff and was promoted as Deputy Director in 1997, Vice President and MERL Fellow in 2003 and now as MERL Fellow. He holds 65 U.S. patents and has authored or co-authored 2 books as well as more than 150 journal and conference papers. For his contributions on HDTV development he obtained 1994 Sarnoff technical achievement award. He also obtained the best paper award of IEEE Transactions on Consumer Electronics in 1993, the best paper award of International Conference on Consumer Electronics in 1997 and the best paper award of IEEE Transaction on CSVT in 2003. He was an Associate Editor for IEEE Transaction on Circuits and Systems for Video Technology and the Chair of Visual Processing Technical Committee of IEEE Circuits and System Society. He is an IEEE Life Fellow.

Oscar C. Au Oscar C. Au received his B.A.Sc. from Univ. of Toronto in 1986, his M.A. and Ph.D. from Princeton Univ. in 1988 and 1991 respectively. After being a postdoctoral researcher in Princeton Univ. for one year, he joined the Hong Kong University of Science and Technology (HKUST) as an Assistant Professor in 1992. He is/was a Professor of the Dept. of Electronic and Computer Engineering, Director of Multimedia Technology Research Center (MTrec), and Director of the Computer Engineering (CPEG) Program in HKUST. His main research contributions are on video and image coding and processing, watermarking and light weight encryption, speech and audio processing. He has published 60+ technical journal papers, 350+ conference papers, and 70+ contributions to international standards. His fast motion estimation algorithms were accepted into the ISO/IEC 14496-7 MPEG-4 international video coding standard and the China AVS-M standard. His light-weight encryption and error resilience algorithms are accepted into the China AVS standard. He was Chair of Screen Content Coding AdHoc Group in the JCTVC for the ITU-T H.265 HEVC video coding standard. He has 20+ granted US patents and is applying for 70+ more on his signal processing techniques.

Dr. Au is a Fellow of IEEE, a Fellow of the Hong Kong Institute of Engineers (HKIE), and a Board 0f Governor member of the Asia Pacific Signal and Information Processing Association (APSIPA). He is/was Associate Editors of IEEE Trans. On Circuits and Systems for Video Technology (TCSVT), IEEE Trans. on Image Processing (TIP), and IEEE Trans. on Circuits and Systems, Part 1 (TCAS1). He is on the Editorial Boards of Journal of Visual Communication and Image Representation (JVCIR), Journal of Signal Processing Systems (JSPS), APSIPA Trans. On Signal and Information Processing (TSIP), Journal of Multimedia (JMM), and Journal of Franklin Institute (JFI). He is/was Chair of IEEE CAS Technical Committee on Multimedia Systems and Applications (MSATC), Chair of IEEE SPS TC on Multimedia Signal Processing (MMSP), and Chair of APSIPA TC on Image, Video and Multimedia (IVM). He is a member of CAS TC on Video Signal Processing and Communications (VSPC), CAS TC on Digital Signal Processing (DSP), SPS TC on Image, Video and Multidimensional Signal Processing (IVMSP), SPS TC on Information Forensics and Security (IFS), and ComSoc TC on Multimedia Communications (MMTC). He served on the Steering Committee of IEEE Trans. On Multimedia (TMM), and IEEE Int. Conf. of Multimedia and Expo (ICME). He also served on the organizing committee of IEEE Int. Symposium on Circuits and Systems (ISCAS) in 1997, IEEE Int. Conf. On Acoustics, Speech and Signal Processing (ICASSP) in 2003, the ISO/IEC MPEG 71st Meeting in 2005, Int. Conf. on Image Processing (ICIP) in 2010, and other conferences. He was General Chair of Pacific-Rim Conference on Multimedia (PCM) in 2007, IEEE Int. Conf. on Multimedia and Expo (ICME) in 2010 and the International Packet Video Workshop (PV) in 2010. He will be General Chair of APSIPA ASC 2015 and IEEE ICME 2017. He won best paper awards in SiPS 2007, PCM 2007 and MMSP 2012. He is an IEEE Distinguished Lecturer (DL) in 2009 and 2010, APSIPA DL in 2013 and 2014, and has been keynote speaker multiple times.