

## On Modeling the Rendering Error in 3D Video

Cheung, N-M; Tian, D.; Vetro, A.; Sun, H.

TR2012-069 September 2012

### Abstract

We propose an analytical model to estimate the rendering 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, the camera configuration and the rendering process. Specifically, we derive position (disparity) errors from the depth errors, and the probability distribution of the position errors is used to calculate the power spectral density of the rendering errors. Experiment results with video sequences and coding/ rendering tools used in MPEG 3DV activities show that the model can accurately estimate the synthesis noise up to a constant offset. Thus, the model can be used to estimate the change in rendering quality for different system designs.

*IEEE International Conference on Image Processing (ICIP)*

This work may not be copied or reproduced in whole or in part for any commercial purpose. Permission to copy in whole or in part without payment of fee is granted for nonprofit educational and research purposes provided that all such whole or partial copies include the following: a notice that such copying is by permission of Mitsubishi Electric Research Laboratories, Inc.; an acknowledgment of the authors and individual contributions to the work; and all applicable portions of the copyright notice. Copying, reproduction, or republishing for any other purpose shall require a license with payment of fee to Mitsubishi Electric Research Laboratories, Inc. All rights reserved.



# ON MODELING THE RENDERING ERROR IN 3D VIDEO

Ngai-Man Cheung, Dong Tian, Anthony Vetro, Huifang Sun

Mitsubishi Electric Research Laboratories

## ABSTRACT

We propose an analytical model to estimate the rendering 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, the camera configuration and the rendering process. Specifically, we derive position (disparity) errors from the depth errors, and the probability distribution of the position errors is used to calculate the power spectral density of the rendering errors. Experiment results with video sequences and coding/rendering tools used in MPEG 3DV activities show that the model can accurately estimate the synthesis noise up to a constant offset. Thus, the model can be used to estimate the change in rendering quality for different system designs.

**Index Terms**— 3D video, DIBR, depth map coding, rendering, power spectral density

## 1. INTRODUCTION

3D video (3DV) has attracted much attention recently. 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) [1]. 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 [2, 3]. The rendering quality, however, depends on several factors and complicated interactions between them. In particular, texture and depth images may contain errors due to imperfect sensing or lossy compression [4, 5], 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 [6], and the effect is more 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 error in the texture data.

In this work, we analyze how depth errors relate to the rendering quality, taking into account texture image characteristics, texture image quality, the camera configuration and the rendering process. In particular, depth errors are used to compute the position errors, and the probability distribution of the position errors is in turn used to estimate the synthesis noise power at the frame level. We use power spectral density (PSD) to analyze the impact of depth errors. This is inspired by previous work which used PSD to study the effect of motion vector inaccuracy [7, 8] and disparity inaccuracy [9]. 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, although some of our discussions and experiments focus on texture/depth error due to predictive coding, we make no assumption on how texture/depth was distorted, and we focus on transformation/interaction of the texture/depth error in the synthesis pipeline.

Regarding previous work, Nguyen and Do [10] is similar to our work in objective, but the approaches are different. They 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. On the other hand, we use PSD to estimate the *value* of the mean squared error (MSE) and test the model with video sequences and coding/rendering tools used in MPEG 3DV activities [11, 12]. An autoregressive model was proposed by Kim et al. [13] to estimate the synthesis distortion at the block level and was shown to be effective for rate-distortion optimized mode selection. A distortion model as a function of the view location was also proposed by Velisavljevic et al. [14] for bit allocation.

The outline of the rest of the paper is as follows. Section 2 discusses our proposed model. Section 3 presents experiment results and Section 4 concludes the paper.

## 2. SYSTEM MODEL

Figure 1 models the processing in a typical synthesis pipeline. 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  $U_i$ , from position  $(m', n)$  to  $(m, n)$ . We assume the cameras are rectified and arranged linearly, and there exists only horizontal disparity given by

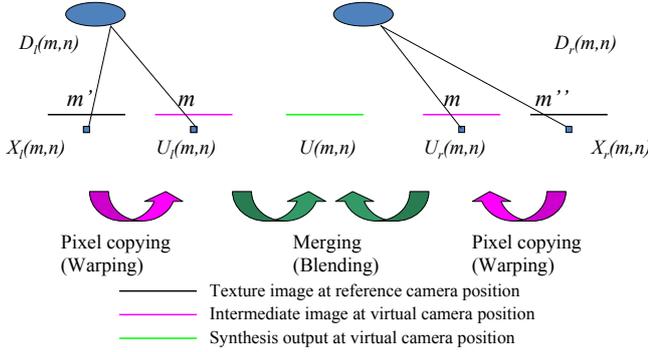
$$m - m' = \frac{D_l(m', n)}{255} (d_{near} - d_{far}) + d_{far} \quad (1)$$

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

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

Here the weight  $\alpha$  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 [15]. However, linear merging remains to be a popular practical technique and could be a good approximation to other advanced merging techniques. 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.



**Fig. 1.** Processing in the synthesis pipeline. Horizontal disparity is  $m - m'$  for the left reference, and  $m - m''$  for the right reference.

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<sup>1</sup>. 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.

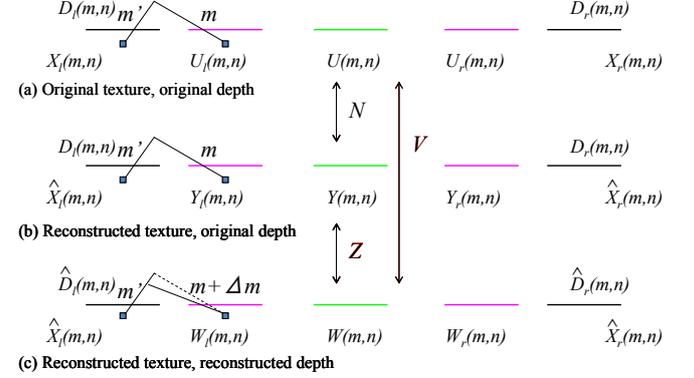
To facilitate the analysis, we consider an intermediate step to model the synthesis noise. As shown in Figure 2.b, we consider the case when the *reconstructed* texture images and the *original* depth images are used in the synthesis to produce the output  $Y$ . Note that  $U$  and  $Y$  are different solely due to the fact that reconstructed texture frames  $\hat{X}_l, \hat{X}_r$  are used in the synthesis instead of the original texture. 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  $V = N + Z$ . Assuming  $N, Z$  are uncorrelated and  $E[N] = 0$ ,

$$E[V^2] = E[N^2] + E[Z^2] + 2E[NZ] \quad (3)$$

$$= E[N^2] + E[Z^2] \quad (4)$$

<sup>1</sup>To ease the discussion, we assume the distortions in the texture/depth images are due to coding. However, the analysis is applicable to other texture/depth errors, e.g., sensing errors.

(4) 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 component, and the total noise power can be approximated simply by summing the two components.



**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.

## 2.1. Estimate the noise power due to texture coding

We proceed to discuss how to estimate the two component noise signals in (4). We first focus on the noise caused by lossy coding of texture image. Refer to Figures 2.a and 2.b,

$$N(m, n) = U(m, n) - Y(m, n) \quad (5)$$

$$\begin{aligned} U(m, n) &= \alpha U_l(m, n) + (1 - \alpha) U_r(m, n) \\ &= \alpha X_l(m', n) + (1 - \alpha) X_r(m'', n) \end{aligned} \quad (6)$$

$$\begin{aligned} Y(m, n) &= \alpha Y_l(m, n) + (1 - \alpha) Y_r(m, n) \\ &= \alpha \hat{X}_l(m', n) + (1 - \alpha) \hat{X}_r(m'', n) \end{aligned} \quad (7)$$

Therefore,

$$\begin{aligned} N(m, n) &= \alpha(X_l(m', n) - \hat{X}_l(m', n)) + \\ &\quad (1 - \alpha)(X_r(m'', n) - \hat{X}_r(m'', n)) \end{aligned} \quad (8)$$

In (6), pixel in  $X_l$  at location  $(m', n)$  is copied to the intermediate image  $U_l$  location  $(m, n)$ . Likewise, in (7), pixel in  $\hat{X}_l$  at location  $(m', n)$  is copied to intermediate image  $Y_l$  location  $(m, n)$ . Importantly, pixels in  $X_l$  and  $\hat{X}_l$  involved in computing  $N(m, n)$  are *spatially collocated*, at  $(m', n)$  (similarly for the right camera), as we choose to decouple the estimation into two steps, and in the first step the same (original) depth information is used in both (6) and (7) to calculate the disparity. The fact that pixels involved in computing  $N(m, n)$  are collocated simplifies the estimation:

$$\begin{aligned} E[N^2] &= \alpha^2 E[(X_l - \hat{X}_l)^2] + \\ &\quad (1 - \alpha)^2 E[(X_r - \hat{X}_r)^2] + \\ &\quad 2\alpha(1 - \alpha)\rho_N \sigma_{X_l - \hat{X}_l} \sigma_{X_r - \hat{X}_r} \end{aligned} \quad (9)$$

Here  $X_l - \hat{X}_l$  and  $X_r - \hat{X}_r$  are the texture coding noise signals, and  $\rho_N$  is the correlation coefficient between  $X_l - \hat{X}_l$  and  $X_r - \hat{X}_r$ .  $\rho_N$  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  $\rho_N$  (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.

## 2.2. 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,

$$Z(m, n) = Y(m, n) - W(m, n) \quad (10)$$

$$Y(m, n) = \alpha Y_l(m, n) + (1 - \alpha) Y_r(m, n) \quad (11)$$

$$W(m, n) = \alpha W_l(m, n) + (1 - \alpha) W_r(m, n) \quad (12)$$

Substitute (11) and (12) into (10), and with  $Z_l = Y_l - W_l$ ,  $Z_r = Y_r - W_r$ , we have

$$Z(m, n) = \alpha Z_l(m, n) + (1 - \alpha) Z_r(m, n) \quad (13)$$

$$E[Z^2] = \alpha^2 E[Z_l^2] + (1 - \alpha)^2 E[Z_r^2] + 2\alpha(1 - \alpha)\rho_Z\sigma_{Z_l}\sigma_{Z_r} \quad (14)$$

(14) 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  $E[Z_r^2]$ ),

$$Z_l(m, n) = Y_l(m, n) - W_l(m, n) \quad (15)$$

$$= Y_l(m, n) - Y_l(m - \Delta m_l, n) \quad (16)$$

Here the depth error causes a horizontal position error  $\Delta m_l$ . From (16), the PSD of  $Z_l$  can be derived:

$$\Phi_{Z_l}(\omega_1, \omega_2) = 2(1 - \cos(\Delta m_l \cdot \omega_1))\Phi_{Y_l}(\omega_1, \omega_2) \quad (17)$$

Since  $\Delta m_l$  is random, we take expectation in (17) w.r.t. the probability distribution of  $\Delta m_l$ ,  $p(\Delta m_l)$ :

$$\Phi_{Z_l}(\omega_1, \omega_2) = 2(1 - E[\cos(\Delta m_l \cdot \omega_1)])\Phi_{Y_l}(\omega_1, \omega_2) \quad (18)$$

$$= 2(1 - \text{Re}\{P(\omega_1)\})\Phi_{Y_l}(\omega_1, \omega_2) \quad (19)$$

where  $P(\omega_1)$  is the Fourier transform of  $p(\Delta m_l)$ , and  $\text{Re}\{P(\omega_1)\}$  denotes the real part of  $P(\omega_1)$ . (19) can be derived by using the identity:  $\cos(\Delta m_l \cdot \omega_1) = (e^{j\Delta m_l \cdot \omega_1} + e^{-j\Delta m_l \cdot \omega_1})/2$ . Approximating  $\Phi_{Y_l}$  by  $\Phi_{\hat{X}_l}$ , we obtain:

$$\Phi_{Z_l}(\omega_1, \omega_2) \approx 2(1 - \text{Re}\{P(\omega_1)\})\Phi_{\hat{X}_l}(\omega_1, \omega_2) \quad (20)$$

(20) 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:  $2(1 - \text{Re}\{P(\omega_1)\})$ , 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  $\Delta m_l$  (similarly for the right camera):

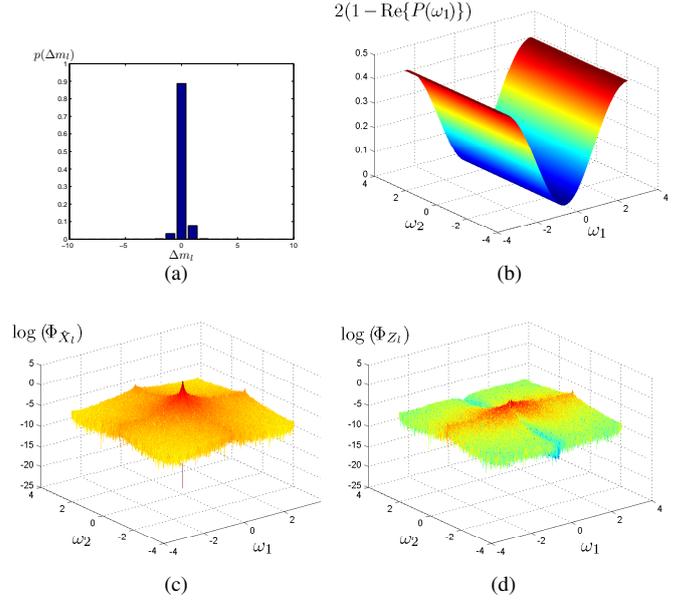
$$\Delta m_l(m, n) = k_l(D_l(m, n) - \hat{D}_l(m, n)) \quad (21)$$

Here  $k_l$  is a spatially invariant constant depending only on the camera setup:

$$k_l = \frac{f \cdot b_l}{255} \left( \frac{1}{z_{near}} - \frac{1}{z_{far}} \right) \quad (22)$$

We integrate  $\Phi_{Z_l}$  in (20) to estimate  $E[Z_l^2]$ .

To illustrate how  $\Phi_{Z_l}$  depends on  $p(\Delta m_l)$ , Figure 3 depicts the empirical  $p(\Delta m_l)$  of a video frame from the sequence Kendo, the corresponding envelope  $2(1 - \text{Re}\{P(\omega_1)\})$ , the PSD of the texture signal, and the PSD of the rendering noise. As suggested in Figure 3, 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).



**Fig. 3.** Rendering noise due to depth errors. (a) Empirical distribution of the position errors,  $p(\Delta m_l)$ . This is computed from depth errors. (b) The corresponding frequency envelope:  $2(1 - \text{Re}\{P(\omega_1)\})$ . Note that the envelope attenuates low horizontal frequency signals. (c) PSD of the texture signals. (d) PSD of the rendering noise due to errors in the depth images. This is the product of (b) and (c). A valley at the low horizontal frequency can be observed.

## 2.3. 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 (9) to determine  $E[N^2]$ . Fast Fourier transform (FFT) of  $\hat{X}_l$  is used to compute  $\Phi_{\hat{X}_l}$  in (20), and  $P(\omega_1)$  is determined from  $p(\Delta m_l)$ , which in turn is estimated from  $D_l$  and  $\hat{D}_l$  and binning  $\Delta m_l$  following (21).  $E[Z_l^2]$  can then be estimated by integrating  $\Phi_{Z_l}$ .  $E[Z_r^2]$  can be estimated in a similar way.  $E[Z_l^2]$  and  $E[Z_r^2]$  are then used in (14) to estimate the noise power due to depth coding,  $E[Z^2]$ . In addition, we found that  $\rho_Z$ , the correlation coefficient between  $Z_l$  and  $Z_r$ , would depend on the variances of  $\Delta m_l$  and  $\Delta m_r$ . In particular, the correlation decreases as the variances of the position error increase. We trained a model to estimate  $\rho_Z$  (parameterized by the average of the variances of  $\Delta m_l$  and  $\Delta m_r$ ), and the same model is used for all sequences and conditions. Finally,  $E[N^2]$  and  $E[Z^2]$  are summed to estimate the noise power in the synthesis output following (4).

Note that some approximation of  $\Phi_{\hat{X}_l}$  and  $\Phi_{\hat{X}_r}$  can be used, e.g., by modeling the texture signals as spatial random fields with isotropic autocorrelation functions [16]. In addition, depth errors can be modeled as Gaussian or Laplacian distributed random variables with variances  $E[(D_l - \hat{D}_l)^2]$  and  $E[(D_r - \hat{D}_r)^2]$ , and  $p(\Delta m_l)$  and  $p(\Delta m_r)$  can be derived following (21).

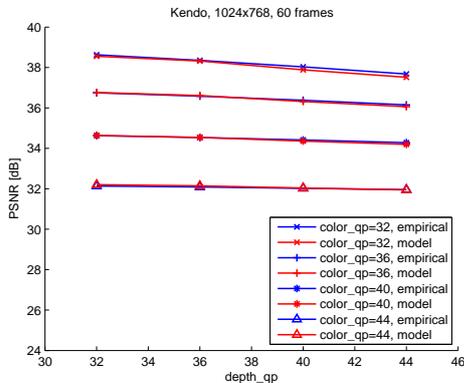
## 3. EXPERIMENTS

We have performed experiments to verify the accuracy of the proposed models. Following the camera configurations in the MPEG 3DV 2-view test cases [11], 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 were 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 were used to synthesis the virtual view, with the merging method chosen to be averaging.

We found that our proposed model requires a constant adjustment to be accurate. The constant adjustment is the same for all coding conditions, but different for different sequences. We conjecture that this is due to the fact that the distribution of the depth error is not entirely random in transform coding. In particular, errors tend to occur more frequently along the edges of the depth images. When these depth edge errors coincide with some strong texture edges, the resulting rendering errors could be substantial and become non-negligible biases to the overall synthesis noise. Such biases tend to be video sequence specific, as they depend on how often depth edges collocate with strong texture edges. While we are still investigating a model for these biases, our current model is sufficient to evaluate the *change* in quality with different encoding conditions and situations, and in many practical applications this is sufficient.

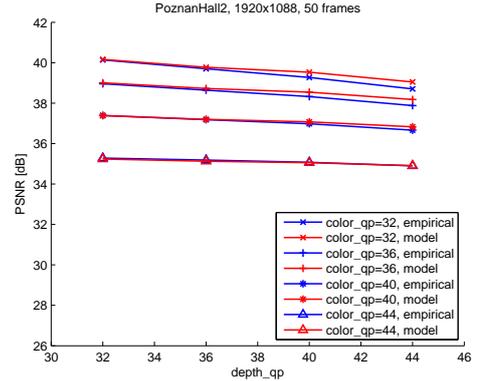
Figures 4 and 5 compare the empirical results and the model results for sequences Kendo ( $1024 \times 768$ ) and PoznanHall2 ( $1920 \times 1088$ ). The empirical results were measured from the rendering output of VSRS. The model results were computed following the discussion in Section 2. A sequence specific constant has also been added to the estimated synthesis noise power (and the same constant for all coding conditions). As shown in the figures, the model can accurately estimate the synthesis quality after the constant adjustment. The results also suggest that, with lower quality texture images (e.g.,  $\text{color\_qp} = 44$ ), only small gains in the rendering output can be obtained with improving the quality of the depth images (reducing  $\text{depth\_qp}$ ). This is because with lower quality texture images the noise due to texture coding  $E[N^2]$  dominates the overall synthesis noise power in (4), and reduction in  $E[Z^2]$  has only a small impact.



**Fig. 4.** Modeling result: Kendo. Results are shown for texture images encoded with different qp ( $\text{color\_qp}$ ). The horizontal axis represents the qp used in encoding the depth images ( $\text{depth\_qp}$ ), and the vertical axis represents the rendering quality. Empirical PSNR were obtained using the actual rendered videos from VSRS instead of model.

#### 4. CONCLUSIONS

We have proposed an analytical model to estimate the rendering 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, the camera configuration and the rendering process. We decoupled the estimation of the power of the synthesis noise into two steps, one focusing on the error due to texture coding, and the other focusing on the error due to depth coding.



**Fig. 5.** Modeling result: PoznanHall2.

We showed that the PSD of the rendering errors 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. Experiment results showed that the model can accurately estimate the synthesis noise up to a constant offset. Thus, the model can be used to predict the change in rendering quality for different system designs.

#### 5. REFERENCES

- [1] Christoph 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.
- [2] P. Merkle, Y. Morvan, A. Smolic, D. Farin, K. Mueller, P. H. N. de With, and T. Wiegand, "The effects of multiview depth video compression on multiview rendering," *Signal Process.: Image Commun.*, vol. 24, pp. 73–88, 2009.
- [3] D. Tian, A. Vetro, and M. Brand, "A trellis-based approach for robust view synthesis," in *Proc. IEEE Int'l Conf. Image Processing (ICIP)*, 2011.
- [4] 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.
- [5] 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.
- [6] 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.
- [7] 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.
- [8] 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.
- [9] P. Ramanathan and B. Girod, "Rate-distortion analysis for light field coding and streaming," *Signal Process.: Image Commun.*, vol. 21, pp. 462–475, 2006.
- [10] 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.
- [11] MPEG Video and Requirement group, "Call for proposals on 3D video coding technology," Tech. Rep., MPEG, 2011, MPEG N12036.
- [12] M. Tanimoto, T. Fujii, and K. Suzuki, "View synthesis algorithm in view synthesis reference software 2.0 (VSRS2.0)," Tech. Rep., MPEG, 2009, MPEG M16090.
- [13] 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 (VIPIC)*, 2010.
- [14] Vladan Velisavljevic, Gene Cheung, and Jacob Chakareski, "Bit allocation for multiview image compression using cubic synthesized view distortion model," in *Proc. IEEE International Workshop on Hot Topics in 3D*, 2011.
- [15] Zefeng Ni, Dong Tian, Sitaram Bhagavathy, Joan 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.
- [16] Anil K. Jain, *Fundamentals of Digital Image Processing*, Prentice Hall, 1988.