2009 11th IEEE International Symposium on Multimedia

A Synthesis-Quality-Oriented Depth Refinement Scheme for MPEG Free Viewpoint Television (FTV)

Chun-Chi Chen[1], Yi-Wen Chen, Fu-Yao Yang and Wen-Hsiao Peng[2]
Department of Computer Science, National Chiao Tung University
1001 Ta-Hsueh Rd., HsinChu 30010, Taiwan
E-mail:{[1]cheerchen.cs96g@g2.nctu.edu.tw, [2]w.peng@ieee.org}

Abstract

This paper addresses the problem of refining depth information from the received reference and depth images within the MPEG FTV framework. An analytical model is first developed to approximate the per-pixel synthesis distortion (caused by depth-image compression) as a function of depth-error variances, intensity variations, ground-truth depth and virtual camera locations. We then follow the model to detect unreliable depth pixels by inspecting intensity gradients and to refine their values with a candidate-based block disparity search. Additional side information is transmitted to make both operations robust against compression effects. Experimental results show that our scheme offers an average PSNR improvement of 1.2 dB over MPEG FTV and consistently outperforms the state-of-the-art methods. Moreover, it can remove synthesis artifacts to a great extent, producing a result that is very close in appearance to the ground-truth view image.

1 Introduction

Technology evolution in the capture and display of 3D videos will soon extend visual sensation from 2D to 3D while allowing unrestricted spatiotemporal scene navigation. In general, offering a 3D depth impression of a real-world scene requires two separate images captured from properly arranged viewing positions. To enable scene navigation, a multi-view video may have to be acquired through a dense camera set-up. However, due to the complexity involved in acquisition, storage and transmission, it is unlikely to have a large number of camera inputs. An efficient 3D data format is thus needed to allow the generation of intermediate views from a sparse sampling of the observed scene.

For this, the MPEG committee has recently defined a "multi-view video plus depth" data format [1], which specifies a way of multiplexing the coded representations of a multi-view video and its associated per-pixel depth information. With explicit scene geometry, an arbitrary virtual view can be generated at the receiver side by means of the so-called depth-image-based rendering (DIBR) [2][3][4][5], requiring only a small number of view images for scene navigation. Since depth images must be conveyed together with the corresponding view images, both types of scene representations are compressed, based mostly on H.264/AVC, for an efficient use of network bandwidth.

Although block-based hybrid coding is equally applicable to depth-image compression, it causes undesirable synthesis artifacts. This is because depth images represent scene geometry information, the characteristics of which are very different from those of intensity data. It was shown in [6] that visually imperceptible depth errors can still have a profound effect on synthesis quality.

A few approaches have been proposed to alleviate synthesis artifacts caused by depth-image compression. In [7] Tanimoto et al. found that the magnitude of synthesis errors is linearly proportional to the distance between the virtual and reference cameras. They proposed to compensate the synthesis errors in a virtual view by estimating their magnitudes from the errors observed in a nearby reference view. Sung et al. [8], on the other hand, made use of the Lambertian condition to refine depth images. The process involves using the similarity between the depth (and intensity) values of corresponding pixels to detect unreliable depth pixels and then refining their values through a group-based disparity search. Because both schemes rely entirely on the decoded information for intensity correction or depth refinement, their performance is greatly influenced by compression effects.

In this paper, we propose a synthesis-quality-oriented depth refinement scheme. Rather than trying

978-0-7695-3890-7/09 $26.00 © 2009 IEEE
DOI 10.1109/ISM.2009.68

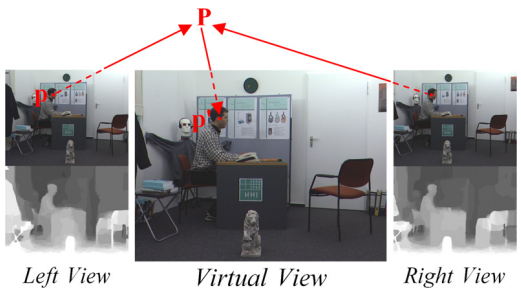


Figure 1. View synthesis based on multi-view video plus depth.

to minimize depth errors, our scheme, as implied by its name, intends to detect and refine only those depth pixels that are highly sensitive to errors. An analytical model was derived to measure how sensitive a depth pixel is to its error in terms of synthesis distortions. The model was also used as a guide for detection and refinement. In order for the two operations to be able to adapt to statistical changes due to compression effects, the settings of their control parameters are first determined at the sender side by evaluating the performance as perceived by the receiver over the range of all possible choices, and then sent to the receiver as the side information. Although extra bits are required for signaling, the overhead is negligible and justified by the significant improvement in synthesis quality. Experimental results show that the proposed scheme has an average PSNR gain of 1.2dB over MPEG FTV and consistently outperforms the state-of-the-art methods.

This paper is organized as follows: Section 2 contains a brief review of DIBR. Section 3 introduces an analytical model for characterizing synthesis distortions caused by depth-image compression. Section 4 describes our proposed synthesis-quality-oriented depth refinement scheme. Section 5 compares the proposed scheme with the state-of-the-art approaches in terms of synthesis quality. The paper is concluded with a summary of our observations.

2 Depth-Image-based Rendering

Depth-image-based rendering (DIBR) is a view generation method that renders virtual views of a scene from a known reference image and its associated per-pixel depth information. Often referred to as 3D image warping, the process involves first reprojecting the reference image into the 3D space utilizing its depth information, followed by the projection of the reconstructed scene onto the image plane of a virtual view camera. The warping defines a vector-valued function Ψ that takes pixel coordinates $\mathbf{p} = [x, y]^T$ in the reference view as input and returns the corresponding coordinates $\mathbf{p}' = [x', y']^T$ in the virtual view as output:

$$\Psi : \begin{bmatrix} \mathbf{p} \\ 1 \end{bmatrix} \mapsto \begin{bmatrix} \mathbf{p}' \\ 1 \end{bmatrix} = \mathbf{A}'\mathbf{R}\mathbf{A}^{-1} \begin{bmatrix} \mathbf{p} \\ 1 \end{bmatrix} + \frac{1}{Z_p}\mathbf{A}'\mathbf{T}, \quad (1)$$

where the rotation and translation matrices, $\mathbf{R}$ and $\mathbf{T}$, specify the relative position of the virtual camera; $\mathbf{A}'$ and $\mathbf{A}$ indicate respectively the intrinsic parameters of the virtual and reference cameras; and Z_p is the depth value associated with $\mathbf{p}$. In the above, we have tacitly assumed parallel camera configuration. The reader is referred to [2] for details. For brevity we use $\Psi(\mathbf{p}; Z_p)$ to denote the warping of the pixel $\mathbf{p}$.

Eq. (1) establishes a depth dependent relation between the pixel coordinates of corresponding points in an image pair. According to the equation, an arbitrary virtual view can in principle be generated, provided that the depth value Z_p is known for every pixel $\mathbf{p}$ in the reference image and that camera parameters are available. In practice, however, the viewpoint navigation is constrained by disocclusion problems: "holes" appear in synthesized images if areas occluded in the reference view become visible in a virtual view. Such artifacts become most obvious when the virtual view is very far away from its reference.

To reduce the effects of disocclusion, the MPEG committee has recently proposed a "multi-view video plus depth" data format that enables the generation of a virtual view to make use of more than one reference view. Figure 1 shows a classic illustration of the view synthesis based on such data format. In the example, each pixel in the virtual view is formed by a weighted sum of its corresponding points in the two reference views, and depending on the disocclusion level, the weight vector can vary from one pixel to another. To find the corresponding points, depth images must be transmitted along with their video signals. Due to the enormous amount of data involved, both view and depth images must be compressed prior to transmission. The influence of depth-image compression on synthesis quality is the subject of the next section.

3 Per-Pixel Synthesis Distortion Model

In this section, we introduce an analytical model for characterizing synthesis distortions caused by depth-image compression. The model is explained with reference to Figure 2, which illustrates an example of disparity-compensated interpolation based on an impaired depth representation. In the figure, I_T denotes

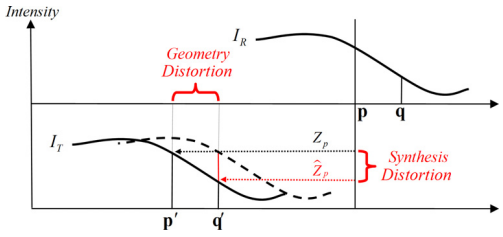


Figure 2. Disparity-compensated interpolation using an impaired depth representation.

a virtual view image generated from the reference image I_R utilizing its ground-truth, per-pixel depth information. As mentioned previously, the warping Ψ establishes a relation between the intensity values of the reference and virtual images: $\mathbf{p}' = \Psi(\mathbf{p};Z_p)$ and $I_T(\mathbf{p}') = I_R(\mathbf{p})$. To simplify our discussion, we assume that the reference image I_R is without coding errors. The more general case can be analyzed along the same lines of derivations that follow.

To examine the influence of depth-image compression on synthesis quality, we approximate the coding effects of depth images by an additive noise model, i.e., $\widehat{Z}_p = Z_p + n_p$. Through the warping function Ψ, the depth error n_p causes the projection of the pixel $\mathbf{p}$ to move from $\mathbf{p}' = \Psi(\mathbf{p};Z_p)$ to $\mathbf{q}' = \Psi(\mathbf{p};\widehat{Z}_p)$**;** the effect is known as *geometry distortion*. It then follows that $I_R(\mathbf{p})$ is substituted for $I_T(\mathbf{q}')$ as the intensity value of the pixel $\mathbf{q}'$; the squared difference indicates the synthesis distortion contributed by n_p:

$$\begin{aligned}\xi_p &\triangleq \left(I_R(\mathbf{p}) - I_T(\mathbf{q}')\right)^2 = \left(I_R(\mathbf{p}) - I_R(\mathbf{q})\right)^2 \\ &\approx \left(I_R(\mathbf{p}) - I_R(\mathbf{p}) - \nabla I_R(\mathbf{p})\cdot(\mathbf{q}-\mathbf{p})\right)^2 \\ &= \left(-\nabla I_R(\mathbf{p})\cdot(\mathbf{q}-\mathbf{p})\right)^2, \end{aligned} \tag{2}$$

where $\mathbf{q}'$ is inversely projected to I_R by the inverse mapping function $\Psi^{-1}(\mathbf{q}';Z_q)$ and a Taylor's series expansion is used to approximate $I_R(\mathbf{q})$. Recognizing that $\mathbf{q}' = \Psi(\mathbf{q};Z_q) = \Psi(\mathbf{p};Z_p + n_p)$, we solve for the vector difference $(\mathbf{q}-\mathbf{p})$ as

$$\mathbf{q}-\mathbf{p} = \frac{-n_p}{Z_q(Z_p+n_p)}\mathbf{c},$$

where $\mathbf{c} = \begin{bmatrix}\mathbf{I}_2 & \mathbf{0}_{2\times 1}\end{bmatrix}\mathbf{A}\mathbf{R}^{-1}\mathbf{T}$ is a vector depending solely on camera parameters. Substituting this result into Eq. (2) then yields

$$\xi_p \approx \left(\frac{n_p}{Z_q(Z_p+n_p)}\nabla I_R(\mathbf{p})\cdot\mathbf{c}\right)^2. \tag{3}$$

Now let us consider parallel camera configuration, with which the vector $\mathbf{c}$ has the form of $[c,0]^T$ where $|c|$ is proportional to the distance between the reference and virtual cameras. Then it is obvious that

$$\xi_p \approx \left(\frac{n_p}{Z_q(Z_p+n_p)}\right)^2 \times g_x^2(\mathbf{p}) \times c^2, \tag{4}$$

where $g_x(\mathbf{p})$ denotes the x component of the gradient $\nabla I_R(\mathbf{p}) = [g_x(\mathbf{p}), g_y(\mathbf{p})]^T$ computed at $\mathbf{p}$. To obtain the expected per-pixel synthesis distortion conditioned on ground-truth depth values, we take conditional expectations of both sides and expand $(n_p/Z_q(Z_p+n_p))^2$ into its Taylor series in n_p:

$$\begin{aligned} & E\{\xi_p|Z_p,Z_q\} \\ \approx\ & E\left\{\left(\frac{n_p}{Z_q(Z_p+n_p)}\right)^2|Z_p,Z_q\right\}\times m_g^{(2)}(\mathbf{p})\times c^2 \\ =\ & \frac{1}{Z_q^2}\times\left(\frac{E\{n_p^2\}}{Z_p^2}-2\frac{E\{n_p^3\}}{Z_p^3}+3\frac{E\{n_p^4\}}{Z_p^4}-\ldots\right) \\ & \times m_g^{(2)}(\mathbf{p})\times c^2 \\ =\ & \frac{1}{Z_q^2}\times\left(\frac{\sigma_n^2(\mathbf{p})}{Z_p^2}+9\frac{\sigma_n^4(\mathbf{p})}{Z_p^4}+75\frac{\sigma_n^6(\mathbf{p})}{Z_p^6}+\ldots\right) \\ & \times m_g^{(2)}(\mathbf{p})\times c^2 \\ \approx\ & \frac{1}{Z_q^2}\times\frac{\sigma_n^2(\mathbf{p})}{Z_p^2}\times m_g^{(2)}(\mathbf{p})\times c^2, \end{aligned} \tag{5}$$

where $m_g^{(2)}(\mathbf{p}) = E\{g_x^2(\mathbf{p})\}$ can be viewed as a measure of how rapidly the intensity changes along the horizontal direction at $\mathbf{p}$, and $\sigma_n^2(\mathbf{p})$ indicates the corresponding depth-error variance. In the above, n_p is assumed to be independent of $g_x(\mathbf{p})$ and to obey the normal distribution, i.e., $n_p \sim N(0,\sigma_n^2(\mathbf{p}))$. The last approximation in Eq. (5) is justified because Z_p is usually much greater than $\sigma_n(\mathbf{p})$.

Eq. (5) provides a non-stationary model for the expected per-pixel synthesis distortion, which suggests that the depth error for different pixels should have different contributions to the overall synthesis distortions. From the equation, the distortion caused by n_p is determined by several factors measured at $\mathbf{p}$: the depth-error variance, the intensity variation, the (ground-truth) depth value**,** as well as the position of the virtual camera. Further insight into the combined effects of these factors is gained by looking at Figure 3, which displays the ratio of Z_p to $\sigma_n(\mathbf{p})$ as a function of $E\{\xi_p|Z_p,Z_q\}$, under various settings of Z_p, Z_q, and $m_g^{(2)}(\mathbf{p})$ simulating smoothly- or rapidly-changing depth/intensity fields. In the experiment, $\sigma_n(\mathbf{p})$ was varied to identify the highest level of error variance at which the specified distortion is achieved. The result is then used to compute $Z_p/\sigma_n(\mathbf{p})$. Intuitively, the ratio, which we call *depth-error sensitivity*, characterizes

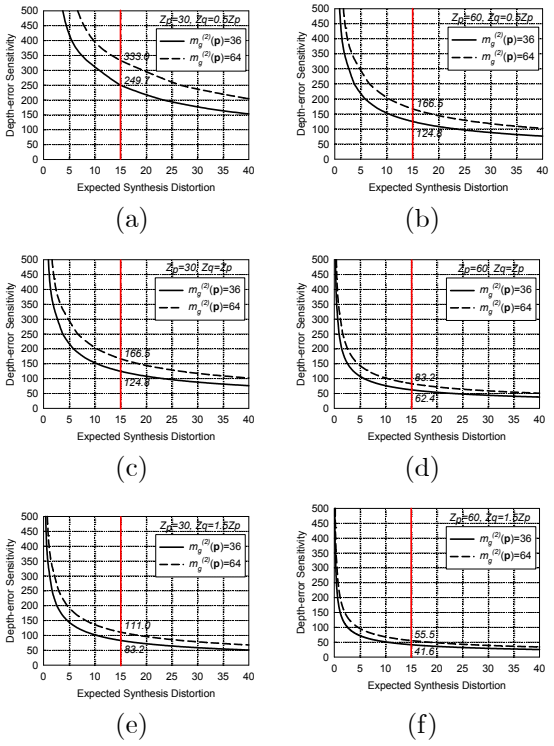


Figure 3. Measuring the depth-error sensitivity under various settings of Z_p, Z_q and $m_g^{(2)}(\mathbf{p})$.

how sensitive a pixel is to its depth error in terms of the extent of synthesis distortions. A higher ratio (sensitivity) implies that a small error in depth can lead to a considerable distortion.

From the figure, several important observations can be made:

1. Compare the curves produced with different settings of $m_g^{(2)}(\mathbf{p})$. The larger the value of $m_g^{(2)}(\mathbf{p})$, the more sensitive the pixel $\mathbf{p}$ is to its depth error; namely, when depth errors happen in areas with vertical edges or fine texture details, their effects on synthesis quality are more apparent. This observation is also corroborated by [7].

2. Compare parts (a)(c)(e) with parts (b)(d)(f). When a pixel corresponds to a farther clipping plane, it exhibits a lower depth-error sensitivity. In this case, the pixel has a larger depth value Z_p and according to Eq. (1), the resulting geometry distortion is less significant.

3. Compare part (e) with parts (a)(c) (or (f) with (b)(d)). When a pixel $\mathbf{p}$ is ill-warped to $\mathbf{q}'$, the resulting synthesis error is less observable if Z_q is much greater than Z_p (and hence $\widehat{Z}_p$). The result can be explained using the example shown in Figure 4, where $\mathbf{q}_1$ and $\mathbf{q}_2$ denote respectively the inverse projections of $\mathbf{q}'$ for the two extreme cases: $Z_{q1} \gg \widehat{Z}_p$ and $Z_{q2} \ll \widehat{Z}_p$. Since $Z_{q1} \gg \widehat{Z}_p \approx Z_p \gg Z_{q2}$, the artifact is more noticeable when a depth error causes warping to substitute a background pixel for a foreground pixel, which explains the less significant change in intensity when $Z_q \gg Z_p$.

4. Observe the reciprocal relation between $\sigma_n^2(\mathbf{p})/Z_p^2$ and c^2 in Eq. (5). It suggests that when a pixel $\mathbf{p}$ is warped to a virtual view that is farther away from the reference view, it is more sensitive to depth errors.

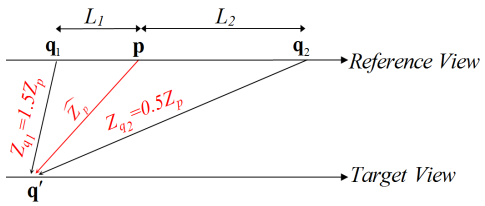


Figure 4. A geometrical interpretation of the effect of Z_q on depth-error sensitivity.

These observations remain valid for other camera configurations, except that the effects of the intensity variation and camera arrangement must jointly be considered by evaluating $E\{(\nabla I_R(\mathbf{p}) \cdot \mathbf{c})^2\}$.

4 Algorithm Details

The framework of MPEG FTV [9] views the transmitted depth images as deterministically specifying the depth information for the reference images. The compression effects of depth images were neglected during the rendering of virtual views. As seen from the analysis in §3, depth errors can cause disturbing synthesis artifacts, especially at areas with sharp edges or fine texture details. To tackle the problem, we propose to regard both the received view and depth images as sources of information about the ground-truth depth of the scene, and provide ways to detect and refine unreliable depth values.

4.1 System Architecture

To allow for an easier understanding of our algorithm, Figure 5 depicts the system block diagram with

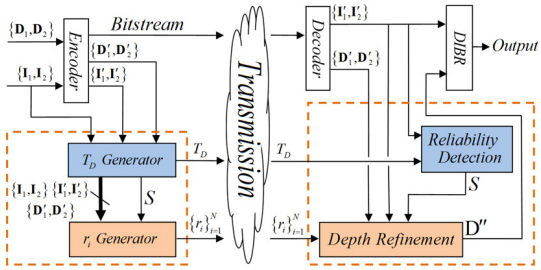


Figure 5. System Block Diagram.

a highlight on the data communicated between functional blocks. As shown, for an economic use of network bandwidth, both reference images $\{I_1, I_2\}$ and their respective per-pixel depth information $\{D_1, D_2\}$ are compressed prior to transmission. These data are decoded and reconstructed at the receiver side before they are used for the creation of virtual views. The "prime" symbols in the figure differentiate the coded view and depth images from their original sources.

Recognizing that depth-image compression may give rise to depth errors, we introduce a depth refinement mechanism at the receiver side. The objective is to improve synthesis quality by refining the depth values for those pixels (which we call *unreliable pixels*) being highly sensitive to depth errors. The process consists of two sequentially operated steps: (1) the detection of unreliable pixels and (2) the refinement of their depth values, both need to access the coded view and depth images. To make their performance robust against compression effects, additional control parameters are transmitted to the receiver as the side information, with their settings being determined at the sender side by evaluating the detection and refinement quality as perceived by the receiver over the range of all possible choices. The details are elaborated in the subsequent sections.

4.2 Reliability Detection

The detection process at the receiver side aims to discover unreliable pixels–i.e., those that are highly sensitive to depth errors and hence require higher fidelity for their depth values in order to minimize rendering errors. From the theoretical analysis in §3, a pixel is likely to be unreliable if it locates in a region with large intensity variation, or if it represents a pixel in a near clipping plane. Although both facts can jointly be utilized to form detection criteria, we consider only the use of intensity variation because view images are generally better compressed than their depth representations, making the intensity information more reliable for decision-making.

To quantify intensity variation, we adopt the Gaussian derivative operator to compute gradient for all the pixels in view images. A pixel $\mathbf{p}$ is considered to be unreliable and its depth value deserves refining if the magnitude $\|\nabla I'_R(\mathbf{p})\|$ of its gradient exceeds a given level T_D[1]. According to Observation #1 in §3, such a pixel is highly sensitive to depth errors, hence requiring higher precision for its depth value. Apparently, the value of T_D plays a pivotal role in determining the detection accuracy. With non-stationary signal statistics, we propose to adapt T_D on a frame-by-frame basis. This is realized by transmitting its value as the frame-level side information.

In determining the value of T_D for a particular frame, we wish to strike a good balance between the hit and false-alarm rates. The best setting of T_D, denoted by T_D^*, should have the subset of pixels $\mathcal{S}(T_D^*) = \{\mathbf{p} : \|\nabla I'_R(\mathbf{p})\| > T_D^*\}$ contain as many unreliable pixels as possible while keeping the number of reliable ones to be minimal. To find T_D^*, we first associate each plausible choice of T_D and the corresponding set of pixels $\mathcal{S}(T_D)$ with a matching score that weights the hit rate against the false-alarm rate:

$$J(T_D) = \sum_{\mathbf{p}\in\mathcal{S}(T_D)} \left(\mathbf{1}_{\mathcal{S}}(\mathbf{p})\xi_p - (1-\mathbf{1}_{\mathcal{S}}(\mathbf{p}))\pi\right),$$

where $\mathbf{1}_{\mathcal{S}} : \mathbf{p} \in \mathcal{S} \rightarrow \{0, 1\}$ is an indicator function defined as

$$\mathbf{1}_{\mathcal{S}}(\mathbf{p}) = \begin{cases} 1 & \text{if } \xi_p \geq \delta \\ 0 & \text{if } \xi_p < \delta \end{cases}.$$

Then we choose, among all possible choices, the one that yields the highest matching score, i.e., $T_D^* = \arg\max_{T_D} J(T_D)$. The approach can be interpreted as to evaluate, at the sender side, the detection quality as perceived by the receiver.

In the course of computing the matching score, it is necessary to decide whether a hit or false alarm occurs. This is accomplished by evaluating the per-pixel synthesis distortion ξ_p at the sender side with I_1 and I_2 (or in the reverse order) being used in place of I_R and I_T, respectively (cf. Eq. (2)). Specifically, if ξ_p is greater than or equal to a threshold δ, indicating that the depth associated with the pixel $\mathbf{p}$ may be unreliable, a hit is identified; otherwise, a false alarm is signaled. Ideally, the δ should be set to zero according to the Lambertian condition; however, in practice a non-zero value was used to compensate camera noises and illumination difference between view images. The

[1] With parallel camera configuration, only the x component of the gradient is computed and compared with T_D (cf. Eq. (4)). Also, I'_R represents a coded reference image.

settings of δ and π that yield the best synthesis quality (in terms of PSNR) are searched exhaustively at the sender side. Note that they need not be transmitted to the receiver.

4.3 Depth Refinement

After we discover all the unreliable pixels, our next step is to refine their depth values. Because depth refinement is performed by the receiver, its operation must be made computationally simple and efficient. For this reason, we adopt a candidate-based disparity estimation scheme to derive depth from the received view images. As in most block-based algorithms, a constant disparity is searched for each block of pixels (of size 7×7), centered on an unreliable pixel $\mathbf{p}$, by minimizing the error between the two view images after disparity compensation. However, unlike their techniques, which usually require examining a large number of disparities, ours restricts the search to only those disparities that correspond to an integer depth value in the interval of $[\widehat{Z}_p - R_p, \widehat{Z}_p + R_p]$. On one hand, this constraint is an expediency out of complexity considerations, and on the other hand, it prevents the simple block-based search from getting an improper disparity.

Although reducing the number of search candidates helps to simplify the disparity search, the issues are how to determine a proper value of R_p for each unreliable pixel and how to signal the information efficiently. As described previously, the value of R_p determines the maximum modification of $\widehat{Z}_p$ that can be caused by depth refinement–i.e., it controls the strength of refinement. It was found in our analysis that the depth error sensitivity of a pixel is related to its ground-truth depth value, implying that the adaptation of R_p should refer to the value of $\widehat{Z}_p$ (which is an approximation of Z_p). For a trade-off between quality and overhead, we divide the set $\mathcal{S}(T_D^*)$ into N disjoint subsets $s_i(T_D^*), 1 \leq i \leq N$, each of which is assigned a refinement search range r_i. A uniform quantizer that operates on the received depth $\widehat{Z}_p$ is used to categorize the unreliable pixels in $\mathcal{S}(T_D^*)$ into one of the N subsets. After that, the best settings of $\{r_i\}_{i=1}^N$ are searched exhaustively at the sender side and transmitted to the receiver as the side information.

Figure 6 shows a sample result of our refinement process. Observe that depth compression introduces blocking artifacts on the decoded depth image (see Figure 6 (b)(e)). With depth refinement, we can remove the artifacts largely (see parts (c) and (f) of Figure 6); note the clarity of object boundaries that simply are not visible in the decoded depth image. Interestingly, the refinement can even recover some details that are removed by the enforcement of depth smoothing (compare parts (a)(d) and (c)(f) of Figure 6).

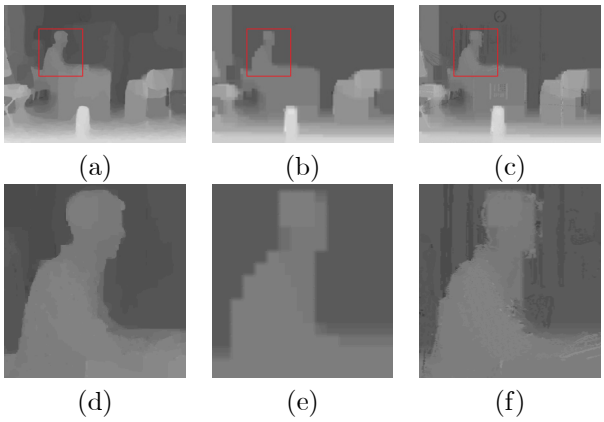


Figure 6. A sample result of the proposed depth refinement algorithm: (a)(d) the original depth image, (b)(e) the decoded depth image, and (c)(f) the refined depth image.

5 Experiments

Simulation was carried out to demonstrate the performance of the proposed scheme, and the results were compared with that of [7] and [8]. All the refinement schemes were implemented with the MPEG committee software VSRS 2.1. All experiments used DERS 2.0 to generate depth images and JMVC 3.0.1 to encode multi-view videos and their depth. The average PSNR of synthesized images was computed based on the first 100 frames of each test sequence. Particularly, in implementing the method described in [7], we employed the magnitude of synthesis errors rather than manually generated edge maps to distinguish pixels of different categories. For a fair comparison, all the threshold values used in [7] and [8] were determined by optimizing the quality of synthesized images.

Figure 7 compares the PSNR of various schemes when the depth QP is varied from 22 to 44. The curves associated with MPEG FTV were produced without depth refinement. To see the effects of reference quality, parts (a) and (b) show the results generated utilizing high-quality references (QP=22), whereas parts (c) and (d) are their low-quality counterparts (QP=31). It can be seen that all three schemes outperform MPEG FTV in all test sequences, and as expected, the improvement is the greatest when depth images are coarsely quantized. Moreover, ours has the highest gain of all the schemes–an average PSNR improvement of 1.2dB over MPEG-FTV. The results are

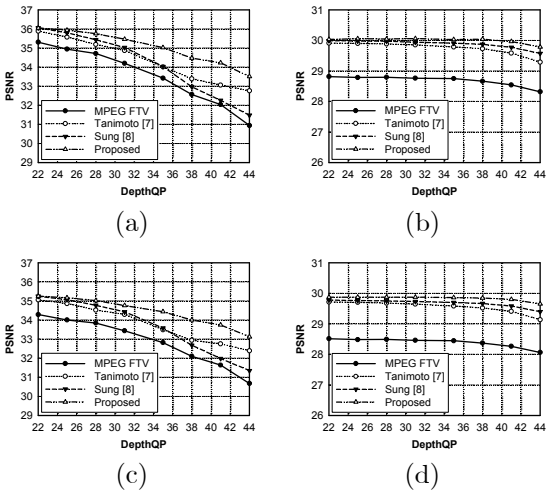


Figure 7. PSNR of synthesized images as a function of the depth and reference QP. The reference view images are coded with QP=22 (a)(b) and QP=31 (c)(d).

consistent with different test conditions.

Figure 8 further compares the subjective quality of synthesized images. Part (a) illustrates what can happen if incorrect depth information is used for view synthesis. Parts (b) through (d) show the results obtained by correcting depth with one of the three schemes just described (i.e., [7], [8], and ours). As can be seen, "ghost effects" appear around object boundaries if the depth is not refined; in comparison, the visual results with depth refinement are considerably improved. Our scheme even produces a result that is very close in appearance to the ground-truth view image. The reason behind the superior performance can be explained with Figure 9, which makes visible the unreliable pixels detected by the three schemes. As expected, our scheme tends to correct more depth pixels locating in areas with fine texture details or vertical edges–namely, those that will crucially affect synthesis quality.

6 Conclusion

To alleviate the coding effects of depth images, we proposed in this paper a synthesis-quality-oriented depth refinement scheme. The approach is characterized by the unique consideration of attempting to refine only those depth pixels that are likely to cause noticeable synthesis artifacts. In the course, we developed an analytical model to establish criteria for reliability detection and to form guidelines for depth refinement. Since both operate on the decoded information, additional side information is transmitted to make them robust against compression effects. Experimental results show that our scheme has the highest PSNR gain of all the state-of-the-art methods. It also produces a result that is visually similar to the ground-truth image. Better performance is expected with the incorporation of more sophisticated disparity search. Besides, the analytical model can find its application in developing depth compression algorithms.

References

[1] C. Fehn, R. Barre, and R. S. Pastoor, "Interactive 3-DTV: Concepts and Key Technologies," *Proceedings of the IEEE*, vol. 94, pp. 524–538, March 2006.

[2] C. Fehn, "A 3D-TV Approach Using Depth-Image-Based Rendering (DIBR)," *Proceedings of Visualization, Imaging, and Image Processing*, September 2003.

[3] C. L. Zitnick, S. B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski, "High-Quality Video View Interpolation Using a Layered Representation," *ACM Transactions on Graphics*, vol. 23, pp. 600–608, August 2004.

[4] A. Smolic, K. Muller, K. Dix, P. Merkle, P. Kauff, and T. Wiegand, "Intermediate View Interpolation based on Multiview Video plus Depth for Advanced 3D Video Systems," *IEEE Int'l Conf. on Image Processing*, October 2008.

[5] E. Cooke, P. Kauff, and T. Sikora, "Multi-view Synthesis: A Novel View Creation Approach for Free Viewpoint Video," *Signal Processing: Image Communication*, vol. 21, pp. 476–492, July 2006.

[6] P. Merkle, A. Smolic, K. Muller, and T. Wiegand, "Multi-view Video plus Depth Representation and Coding," *IEEE Int'l Conf. on Image Processing*, October 2007.

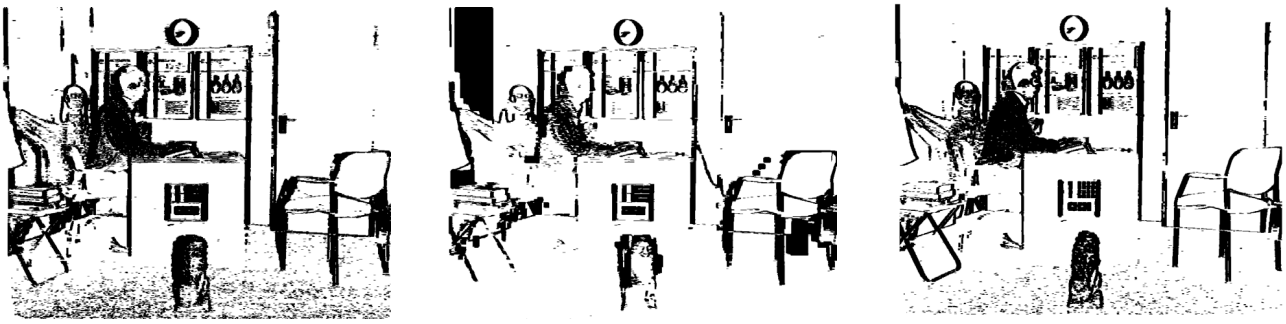
[7] M. Tanimoto, T. Fujii, M. P. Tehrani, M. Wildeboer, and H. Furihata, "Error Cancellation in Free-viewpoint Image Generation for FTV," *ISO/IEC JTC1/SC29/WG11, MPEG09/M16607*, April 2009.

[8] J. Sung, Y. J. Jeon, J. H. Lim, and B. M. Jeon, "Improving View Synthesis Results based on Depth Quality Measure," *ISO/IEC JTC1/SC29/WG11, MPEG09/M16417*, April 2009.

[9] "Applications and Requirements on 3D Video Coding," *ISO/IEC JTC1/SC29/WG11, MPEG09/N10570*, April 2009.

(a) (b) (c) (d)

Figure 8. Subjective quality comparison of synthesized images: (a) MPEG FTV (without depth refinement), (b) Tanimoto [7], (c) Sung [8] and (d) the proposed scheme. The depth QP is set to 44.



(a) (b) (c)

Figure 9. Pixels whose depth values are judged unreliable: (a) Tanimoto [7] (category 2), (b) Sung [8] and (c) the proposed scheme.