Depth Map Coding Using Histogram-Based Segmentation and Depth Range Updating

Chunyu Lin[1], Yao Zhao[1], Jimin Xiao[2] and Tammam Tillo[2]
Institute of Information Science, Beijing Jiaotong University
Beijing Key Laboratory of Advanced Information Science and Network
Beijing, China 100044
[e-mail: {cylin, yzhao}@bjtu.edu.cn]
[2] Department of Electrical and Electronic Engineering, Xian Jiaotong-Liverpool University
Suzhou, China, 215123
[e-mail: {Jimin.Xiao, tammam.tillo}@xjtlu.edu.cn]
*Corresponding author: Chunyu Lin

Received August 8, 2014; revised November 24, 2014; accepted February 28, 2015; published March 31, 2015

Abstract

In texture-plus-depth format, depth map compression is an important task. Different from normal texture images, depth maps have less texture information, while contain many homogeneous regions separated by sharp edges. This feature will be employed to form an efficient depth map coding scheme in this paper. Firstly, the histogram of the depth map will be analyzed to find an appropriate threshold that segments the depth map into the foreground and background regions, allowing the edge between these two kinds of regions to be obtained. Secondly, the two regions will be encoded through rate distortion optimization with a shape adaptive wavelet transform, while the edges are lossless encoded with JBIG2. Finally, a depth-updating algorithm based on the threshold and the depth range is applied to enhance the quality of the decoded depth maps. Experimental results demonstrate the effective performance on both the depth map quality and the synthesized view quality.

Keywords: Depth map coding, 3D coding, histogram-based segmentation

This work was supported in part by 973 program ((NO.2012CB316400), by National Natural Science Foundation of China (no.61402034, no.61210006 and no.61202240), supported by Beijing Natural Science Foundation(4154082) and SRFDP (20130009120038).

http://dx.doi.org/10.3837/tiis.2015.03.016 **ISSN : 1976-7277**

1. Introduction

Among 3D visual data representation, the texture-plus-depth format is very promising due to its suitability for data compression [1] and versatility for different applications. For example, with texture-plus-depth format, a decoder can synthesize the intermediate viewpoints via depth image based rendering (DIBR) [2]. However, to get a better synthesized view, multiple texture images and depth maps of different viewpoints should be stored and transmitted; however, these activities increase storage costs and bandwidth requirements. To solve this problem, many depth map compression schemes have been proposed. The straightforward approach for depth map coding is simply applying a conventional image/video coding algorithm such as JPEG or H.264/AVC. However, since the depth map and its corresponding texture image have different characteristics, this approach cannot achieve the expected performance. Generally, a depth map is an 8-bit gray scale image, in which each pixel records the distance from the camera to the scene. In a depth map, the pixels in the same object have relatively similar depth value, while the edges between different objects look sharp. In regard to the synthesized view, the same object region is less sensitive to depth change, while the edge is very sensitive. Hence, various depth encoding methods try to preserve the edge.

In [3], the smooth surfaces and sharp edges are encoded in different resolutions by graph-based transform (GBT). In [4], edge-aware intra prediction is proposed to code a block with valid prediction or with edge information. In [5], the edge information is estimated by a canny detector; the other regions are encoded by shape adaptive wavelet coding. This kind of encoder achieves significant performance because there is no filtering process across boundaries. However, even though the edge estimation is not very complex, it is not always accurate. Most importantly, the detected edges might not be connected together to form close regions, something that is not easy for the following shape adaptive wavelet coding. In [6], depth maps are modeled as smooth regions using piecewise-linear functions and sharp edges by a straight line.

In [7], each block is approximated with one palette and one object shape map, in which the palette consists of two representative depth values for foreground object and background object in the target block. In [8], two non-rectangular regions are represented with wedgelet and contour mode. Since this kind of introduced modes is implemented on high efficiency video coding (HEVC), each coding block will try these two modes and higher computations are introduced. In addition, wedgelet partition mode is restricted to linear approximation but no detailed information is provided on how to generate the contour partition. In [9], the wedgelet and contour partition are further specified with a depth block containing a more complex separation between both partitions. The contour can then be derived from the corresponding texture block. Furthermore, view synthesis optimization (VSO) is employed. In [10], model-based intra-coding mode is proposed using a depth lookup table in which depth maps can be segmented by a straight line or segmented with texture prediction. These three schemes are block-based schemes, which are proposed in 3D high efficiency video coding (3D-HEVC). By including the depth model coding (DMC) and depth modeling modes (DMM), a relatively high computation is introduced.

In this paper, a simple yet effective depth map coding scheme is proposed by utilizing histogram-based segmentation and depth range updating. The contribution of the paper can be concluded as following. By employing the feature of the depth map characterized by piecewise smooth regions with sharp edges; the scheme firstly employs a histogram based

segmentation algorithm to separate the foreground and background regions. A shape adaptive wavelet encoding scheme is separately implemented on the two regions. Since the two regions have different depth ranges, further 3D depth sensation editing could be provided. For example, increasing the depth values in the range of the foreground regions could enhance the depth sense. In addition, an unequal bit rate allocation could be provided if the distortions of the foreground and background have different effects on our eyes. With the two depth ranges, a depth range updating scheme is proposed to correct the possible compression errors. Finally, the synthesized distortion model is introduced to optimize the synthesized performance.

The rest of the paper is organized as follows. The proposed scheme is discussed in Section 2, and the experimental results and analysis are provided in Section 3. Finally, Section 4 concludes the paper.

2. Proposed scheme

For a typical image/video codec, such as JPEG or H.264/AVC, high frequency is sacrificed for low frequency as a tradeoff for rate distortion sense. This kind of codec is not suitable when edge information is important for view rendering because the edge information generally produces high frequency information. Therefore, an efficient depth map coding scheme is proposed in **Fig. 1**.

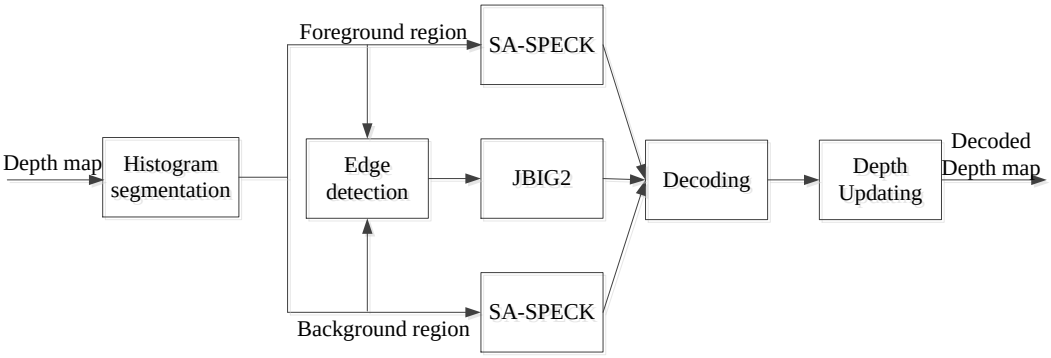


Fig. 1. The diagram of the proposed scheme.

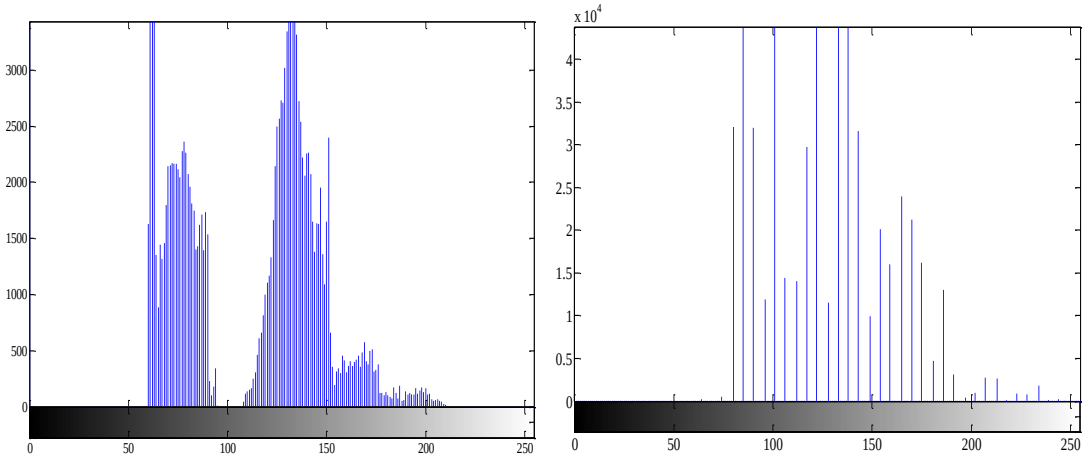
Firstly, the depth map is separated into foreground and background regions using histogram-based segmentation. The two separated regions are smooth and they are separately wavelet transformed. Less or no high frequency coefficients will be generated on the edge between the two separated regions, thus the edge between the foreground and background will not be depressed. Here, the foreground and background regions are transformed with shape adaptive wavelet (SA) and encoded with Set Partition Embedded bloCK (SPECK) [11]. Secondly, the edges between the foreground and background regions could be easily obtained from the two separated regions; the edges are encoded with the Joint Bi-level Image Experts Group (JBIG2) standard [12]. Finally, at the decoder end, the decoded depth map will be updated with its segmented threshold and its original depth range. The details are described below.

2.1 Histogram-based segmentation

In a depth map, edges are generated because different objects vary in distance to the camera. Meanwhile, pixels in one object generally have a similar depth value. Therefore, in one

histogram, each peak value roughly corresponds to one main depth plane, and a simple histogram-based segmentation algorithm can divide the depth map into the foreground and background regions. For example, **Fig. 2** gives the histograms of *Teddy* and *Balloons*. For the histogram of *Teddy* in **Fig. 2(a)**, the "valley" value between the two main peaks will be selected as the threshold. The foreground and background regions are then separated accordingly. Obviously, the case with two main peaks is easy to separate; however, some images may contain more than two peaks (for example, in **Fig. 2(b)**, there are three main peaks). In this case, each valley value will be tested to select the best value as the threshold. The entire histogram-based segmentation algorithm is shown in **Procedure** 1. If there are only two main peaks in the histogram, the middle valley value will be selected as the threshold. For histograms with more than two peaks, the rate distortion (RD) for each case will be calculated to test each valley between any two neighboring peaks. The valley value with the minimum RD will then be used as the threshold. In our case, SPECK encoding can provide a fixed bit rate encoding process. With the fixed bit rate, different distortion values may be obtained through assorted segmented thresholds. Hence, the threshold that provides the minimum distortion is chosen. Notice, the distortion in this example is the rendering distortion instead of the classical encoding distortion, which will be specified in Section 2.2.

Since the probability for a depth histogram with many equivalent peaks is not high, the required computation is not intensive. Actually, to simplify the RD approximation algorithm, Haar wavelet transform can be used and the entropy of the quantized wavelet coefficients can be calculated as the bit rate.



(a) (b)

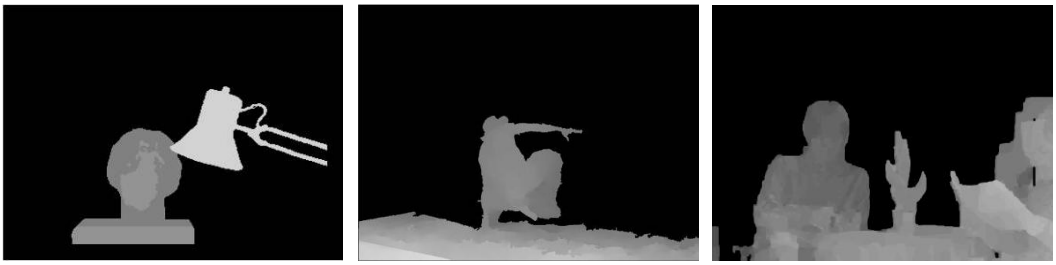
Fig. 2. The histogram of depth map; (a) *Teddy*; (b) *Balloons*

Procedure 1: Histogram-based segmentation
Given I is the input depth map
Given w, h are the width and height of I, respectively
Get the histogram, H, of the depth map, I
Get the number of unique depth values, represented as r Get the main peaks for H that is larger than $wh=r$, denoted as m
if (m==2)
Set the middle valley value as the threshold
else

Find the optimal valley from the main peaks by RD optimization
set the determined value as the threshold
end if
Segment the depth map using the obtained threshold

From the histogram in **Fig. 2(b)**, it can be noted that the depth value is very sparse for *Balloons*. Generally, the histogram of an 8-bit texture image will occupy the whole x-axis range. Certain depth values, alternatively, will not appear in the histogram of depth maps, which is especially true for an indoor depth map because there are frequently limited distance layers in a typical depth map. This feature will be used for depth range updating in Section. 2.3.

Fig. 3 shows examples of the segmented foreground regions with depth maps for *Tsukuba*, *BreakDancer* and *Newspaper*. For the three depth maps, *Tsukuba* is a ground truth disparity image, *Breakdancer* is a relatively accurate estimated image, while *Newspaper* has relatively more errors. It can be seen that the segmented results are quite satisfying. After obtaining the two segmented regions, the edge information is acquired by any simple edge detection scheme. Firstly, the foreground and background regions are set as 255 and 0, respectively, to form a binary image. With such a binary image, an edge detection algorithm can work well with no isolated edge points or lines. In our case, Sobel edge detection algorithm [13] is used.



(a) (b) (c)

Fig. 3. The segmented results by the proposed scheme; (a) *Tsukuba*; (b) *BreakDancer; (c) Newspaper*.

2.2 Encoding

Shape adaptive wavelet transform has successfully been applied in [7] [14] in which the edge line will not be filtered to avoid large wavelet coefficients. Here, the shape adaptive wavelet transform is utilized for the segmented regions. 'Shape adaptive' means the signal being processed could be in any shape. Since the foreground and background regions are wavelet transformed separately, there is no filtering process on the edge between the two regions; therefore, no large wavelet coefficients are generated. This process saves the bit rate cost on the high wavelet coefficients; however, the edge information will not be depressed so as to provide a good synthesized performance. To better illustrate the shape adaptive wavelet coding over traditional wavelet transform (i.e., non adaptive transform), one simplified example on a 1D signal is shown below. $x(n)$ is a simple signal with edge represented by the transition between $x(3)$ and $x(4)$.

$$x(n) = \{1, 2, 3, 4, 100, 101, 102, 100\} \quad (1)$$

If Cohen-Daubechies-Feauveau (CDF) 5/3 wavelet filters [15] are employed, x is decomposed as a low subband x_L and as a high subband x_H

$$\begin{cases} x_L(n) = \{18.5616, -12.5511, 124.6276, 161.3971\} \\ x_H(n) = \{0.0000, 33.5876, -0.0000, -34.2947\} \end{cases} \tag{2}$$

On the other hand, for the shape adaptive wavelet transform, the signal is firstly separated into two sub-signals at the edge

$$\begin{cases} x1(n) = \{1, 2, 3, 4\} \\ x2(n) = \{100, 101, 102, 100\} \end{cases} \tag{3}$$

Then, the two sub-signals are decomposed as follows:

$$\begin{cases} x1_L(n) = \{2.1213, 4.9497\} \\ x1_H(n) = \{0.0000, -1.4142\} \\ x2_L(n) = \{141.0678, 143.8962\} \\ x2_H(n) = \{-0.0000, 0.7071\} \end{cases} \tag{4}$$

By comparing Formulas (2) and (4), it can be seen that no larger high frequency coefficients are generated during the shape adaptive wavelet transform. Since the two sub-signals are encoded independently, there is less bit rate cost on high frequency while the edge information is still preserved. Most importantly, the preserving of edge information can assure a good virtual rendering performance. After wavelet transform, the SPECK [11] image-coding scheme is applied. Compared with the Set Partitioning in Hierarchical Tree (SPIHT) [16] scheme, SPECK only employs the intra-correlations of wavelet coefficients. Nevertheless, the performance of SPECK is comparable to that of SPIHT.

The separated foreground and background regions have different characteristics and different effects on the synthesizing performance; therefore, the bit rate of these two regions should be appropriately allocated to minimize the total distortion. Depth maps are only supplemental data and will not be displayed in 3D devices. So, instead of depth distortion, the synthesized distortion is employed here. In the proposed scheme, the depth map coding is the focus, thus the texture image will not be compressed here. According to the depth image based rendering (DIBR) process, the distortion of the depth map results in position error in the synthesized view. For the parallel camera setting, the position error ΔP can be represented as a horizontal translation [17],

$$\Delta P = \alpha \cdot \delta_x \left(\frac{1}{Z_{near}} - \frac{1}{Z_{far}}\right)\frac{1}{255}\Delta M = k \cdot \Delta M \tag{5}$$

where α is the focal length, δ_x represents the horizontal distance between two cameras, Z_{near} and Z_{far} denote the nearest and farthest depth values, respectively. k is used to reflect the linear relationship between the depth map error ΔM and the horizontal translation, ΔP.

Assuming only the horizontal synthesizing view is considered while the occluded regions are ignored [18], the synthesized distortion D_s can be approximated as

$$\begin{aligned} D_s &= \sum |V - \tilde{V}| = \sum |w(C, M) - w(C, \tilde{M})| \\ &\approx \sum |C(x) - C(x - \Delta P)|^2 \\ &\approx \sum \frac{1}{2}\Delta P(|C(x) - C(x+1)| + |C(x) - C(x-1)|) \end{aligned} \tag{6}$$

where V denotes the synthesized view generated with the original texture image and the original depth map, $\tilde{V}$ represents the synthesized view generated with the original texture image and the compressed depth map, w is the warping function that maps the texture color

value C with a depth map value M into synthesized view. $\tilde{M}$ is the decoded version of M, while $C(x-1)$ and $C(x+1)$ are the adjacent color values without considering the occlusion case. Using formula (5), ΔP can be represented with a depth error of $k\Delta M$. Hence, formula (6) indicates that the synthesized distortion will largely depend on the depth error and the local characteristics of texture information. Generally, the color value between the foreground and background regions is quite different; therefore, even small distortions on the edge of the depth map will lead to large synthesized distortion. Through the proposed scheme, the edge information is preserved with a lower bit rate.

The depth coding errors of the foreground and background regions are represented as ΔM_f and ΔM_b, which are functions of the bit rate for foreground and background, R_f and R_b, respectively. For the fixed total bit rate R, the rest rate allocation that minimizes the total synthesized distortion can be concluded as

$$\begin{cases} minimize \quad (k(\Delta M_f(R_f)(|C_f(x)-C_f(x+1)|+|C_f(x)-C_f(x-1)|) \\ \qquad +\Delta M_b(R_b)(|C_b(x')-C_b(x'+1)|+|C_b(x')-C_b(x'-1)|))) \\ \qquad s.t. \quad R_f+R_b=R \end{cases} \tag{7}$$

where x and x' are any two positions in the foreground and background, respectively. By tuning the bit rates R_f and R_b, different foreground and background distortion is obtained; the combination that minimizes the synthesized distortion is selected. For example, if the foreground region is more sensitive to the synthesized distortion while its corresponding color region is not smooth, more bits should be assigned to it. As a general case, the bit rate of the foreground and background are tuned to be equal. In formula (7), the bit rate of the edge regions is not included in R because the edge regions are lossless encoded by JBIG2.

It should be noted that just the foreground and background regions are segmented in this paper. To achieve further better results, more regions could be generated for depth maps composed of many main objects at different distances. The cost, in this case, is that more corresponding edge information needs to be encoded, which increases the bit rate on the edges. Hence, there is some tradeoff between the number of segmented regions and the compression performance, study of which will be further work.

2.3. Depth range updating

As mentioned in section 2.1, the histogram is generally very sparse. For a typical, 8-bit depth map, the histogram of the depth map does not cover the whole x-axis range and the unique depth values only comprise a small portion of the whole range. These unique depth values for the input depth map are stored in the bit stream so that the reconstructed values can be updated in the decoder. Moreover, the foreground and background regions have variable and non-overlapping ranges, which could be exploited at the decoder side to update some depth values if these values are out of their depth range. For example, the compression error that causes the foreground pixels to appear as a background could easily be compensated by using **Procedure** 2, something that is very important for view synthesized performance. In **Procedure** 2, the minimum and maximum depth value of the input depth maps are obtained and represented as *min* and *max*, respectively. With segmented threshold *th*, the foreground and background regions are processed separately. Using foreground region *f* as an example, if a decoded pixel value *p* is out of range, that is [*th*, *max*], the pixel value could be updated into the range. In addition, if a decoded pixel value *p* is in the range but no corresponding original depth value is present, the decoded value will be forced to become an original depth value, *n.*

Here n is selected according to the following formula

$$\underset{n}{argmin}\{n \mid n \in unique\ of\ [th,max] : \left(0.25 \sum_{p' \in N_4(p)} |p'-n| + |p-n| \right)\} \tag{8}$$

where $N_4(p)$ is used to represent the 4-neighborhood of pixel p, that is the pixels to the left, right, above and below pixel p. Because the depth values are sparse, not all of the values in [*min max*] can be spanned. Furthermore, the number of unique values for a depth map is not larger than *max*-*min*+1. If the reconstructed depth value p does not belong to the unique values, it can be reassigned a close value so that the decoded mean square error (MSE) is smaller than before.

This updating algorithm can correct some compression errors to improve the performance of decoded depth maps. Most importantly, the out of bounds error usually causes the reversing of the foreground and background, thus this error generates large synthesized distortion. The updating algorithm can improve the synthesized performance, as well, based on this reversal. The same principle is used to update the depth values in the background region.

Procedure 2: Depth range updating algorithm for the foreground

Input f the decoded foreground region

Given [th max] the depth range of foreground region

for each pixel value p in f

  if ($p < th$)

    set p as th

  else if ($p > max$)

    set p as max

  else

    find a depth value n in [th max] that is close to p numerically

    if($p \neq n$ and the 4-neighborhood of p is close to n numerically)

      set p as n

    end if

  end if

end for

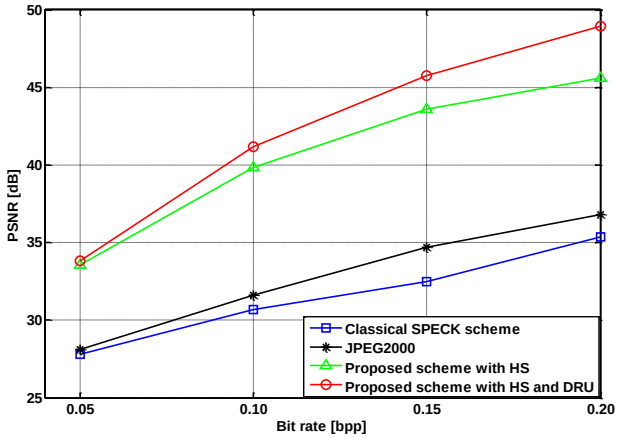
3. Experimental Results and Analysis

In this section, the experimental results for four different types of depth maps are provided. The first type of map is the ground truth depth map, *Tsukuba* and *Teddy* from Middlebury dataset [19]. The depth regions with missing data in *Teddy* are in-painted. The second type of depth map is composed of the first frames in *Kendo*, *Balloons* and *Newspaper* and is an estimated depth map adopted by MPEG-3DV. The third type of map is also an estimated depth map; it is the first frame in *BreakDancer* from Microsoft [20]. The last type also comes from MPEG-3DV and is the first frame in *UndoDancer*; however, this map is a computer graphic (CG) depth map.

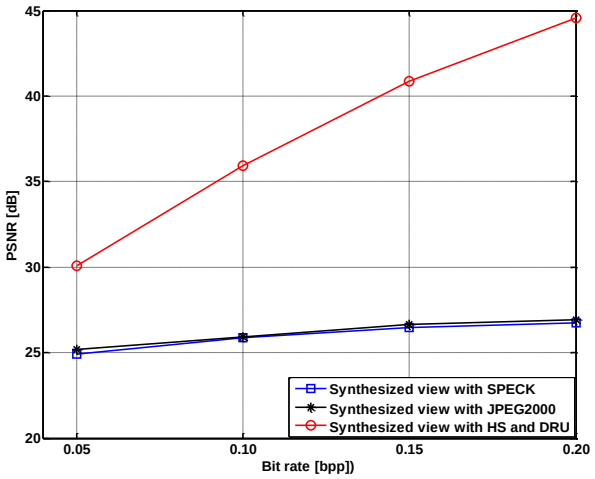
Daubechies 9/7 wavelet transform is employed in the encoding, and the decomposition level depends on the resolution of depth maps. For the first type of depth map with a relatively small resolution, 6-level decomposition is used. For the second and third type depth maps with a resolution of 1024×768, 8-level decomposition is applied. Finally, 9-level decomposition is adopted for *UndoDancer* with a resolution of 1920×1088.

The experiment is conducted on both the proposed scheme with histogram-based segmentation (HS) alone and the proposed scheme with HS and depth range updating (DRU). For comparison, the results of a classical SPECK scheme are also provided. The depth map coding results and the synthesizing results are presented in Figures (a) and (b), respectively. For the depth map of *Tuskuba* in **Fig. 4(a)**, it can be seen that there is up to a 10 dB gain with HS; furthermore, an additional 2 dB gain can be achieved with the depth range updating (DRU). The gain is much larger for the synthesizing performance as up to 18 dB can be achieved, as shown in **Fig. 4(b)**. For the depth of *Balloons* in **Fig. 7(a)**, more than 1 dB gain can be obtained with HS, while up to a 2 dB gain can be reached with the DRU. For *Kendo* in **Fig. 8(a),** a similar gain as that of *Balloons* can be achieved with both HS and DRU. These depth maps are very sparse; hence, except for using HS, the DRU achieves extra gain. The gain of the synthesizing performance is also larger, which can be seen from **Fig. 7(b)** and **Fig. 8(b)**. For *Teddy* in **Fig. 5(a)**, the result of algorithm using piecewise linear functions (PLF) with coefficients prediction [6] is also presented, while the synthesized result of algorithm using multi-resolution graph based transform(M-GBT)[3] is provided in **Fig. 5(b).** It can be seen that up to a 4 dB gain is reached with HS. However, HS and DRU only achieve an approximate 0.1 dB gain because the depth values in *Teddy* are not very sparse, as seen from **Fig. 2(b)**. However, gains can still be made close to 5 dB in the synthesizing performance shown in **Fig. 5(b)**. This case also applies to *UndoDancer* and *BreakDancer* in **Fig. 6** and **Fig. 9**, respectively. For *UndoDancer* in **Fig. 6**, the bit rate cannot exceed 0.15 by using the JPEG2000 software, Jasper [21], so not all bit rates are spanned. **Fig. 10 (a)** presents the results of *Newspaper*. It can be seen that relative smaller gains are achieved by the proposed scheme because there are more edges inside. However, there remains more than 2 dB gain in the synthesizing performance. In conclusion, the proposed depth map coding schemes are more efficient, compared to classical schemes.

Because the threshold is obtained by the rate distortion optimization, the complexity of the proposed scheme is relatively higher than that of the classical scheme. The introduced computation depends on the content of the depth map. Generally, 2-5 times the complexity of the classical SPECK scheme is required. As JPEG2000 has a complexity of 2-4 times of SPECK [11, 22], the computation of the proposed scheme is still acceptable.

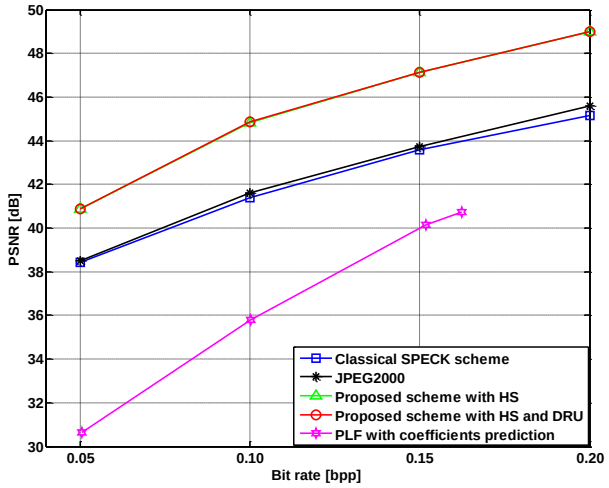


(a)

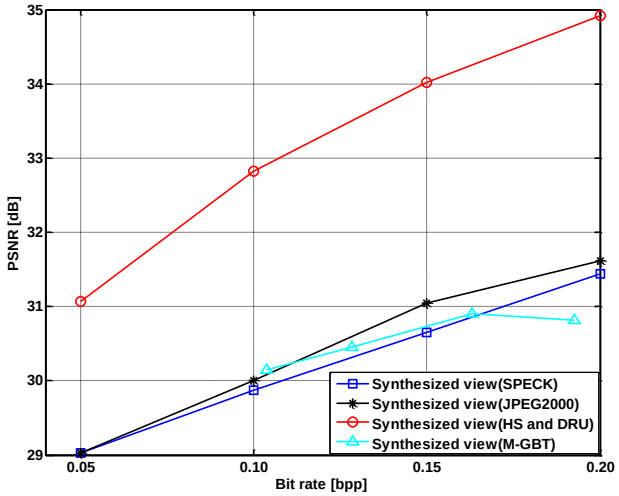


(b)

Fig. 4. The results of *Tsukuba*. (a) Depth map; (b) Synthesized view.

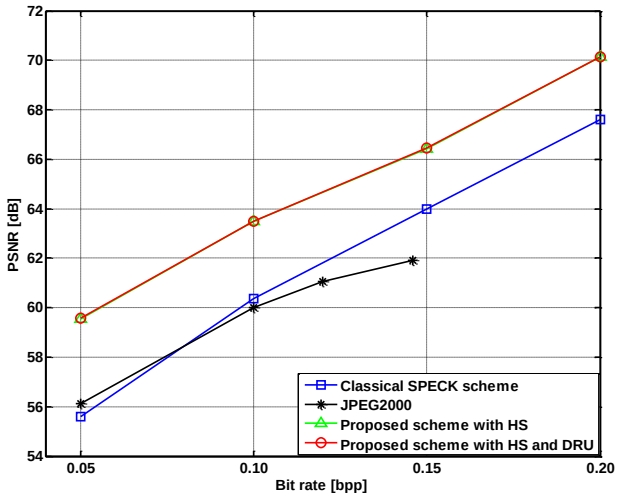


(a)

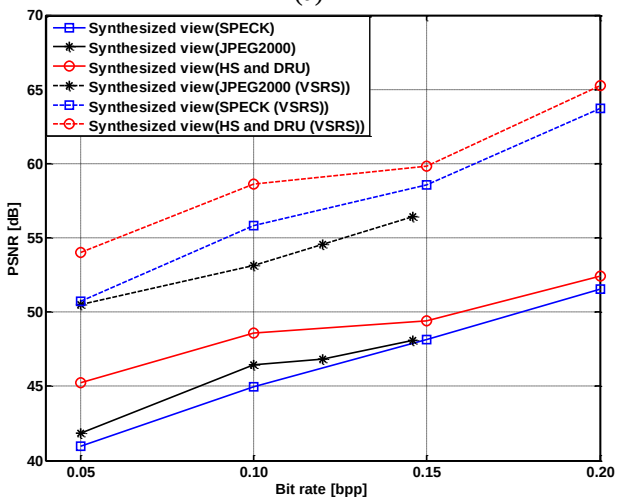


(b)

Fig. 5. The results of *Teddy*. (a) Depth map; (b) Synthesized view.

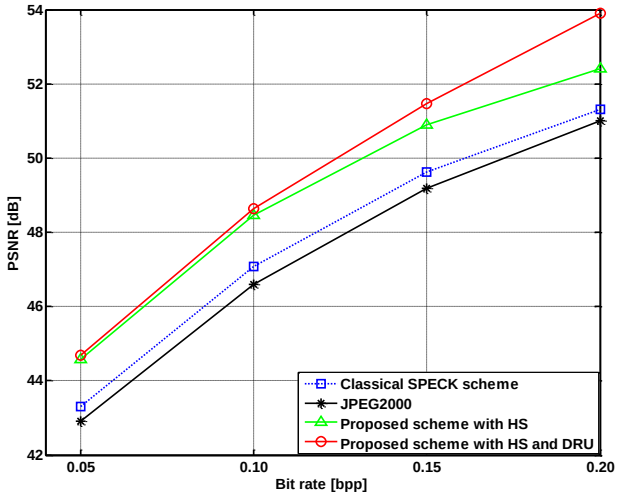


(a)

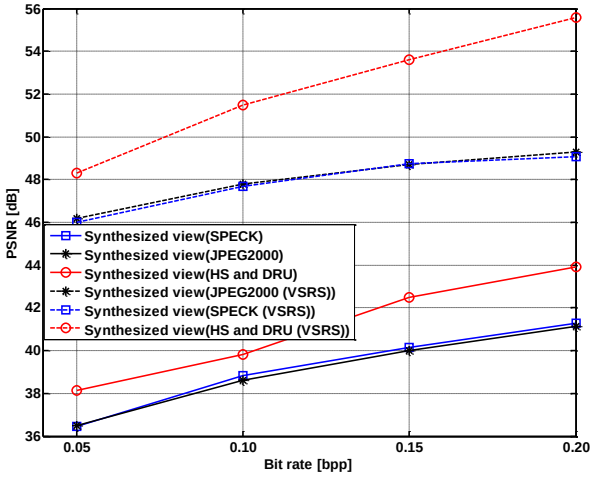


(b)

Fig. 6. The results of *UndoDancer*. (a) Depth map; (b) Synthesized view.

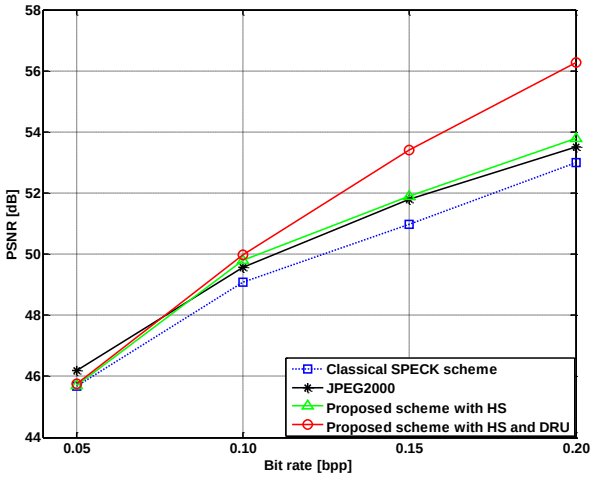


(a)

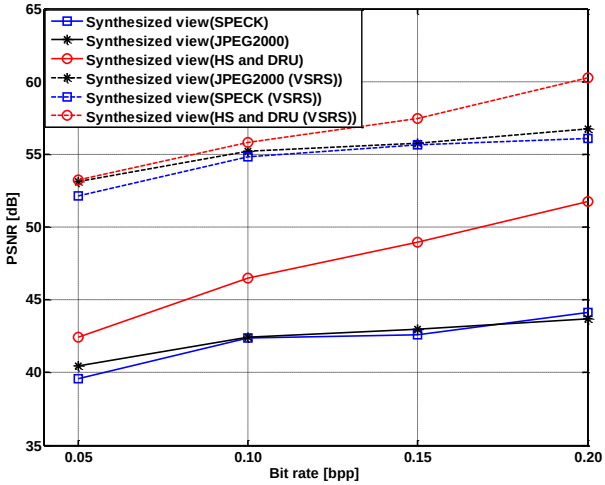


(b)

Fig. 7. The results of *Balloons*. (a) Depth map; (b) Synthesized view.

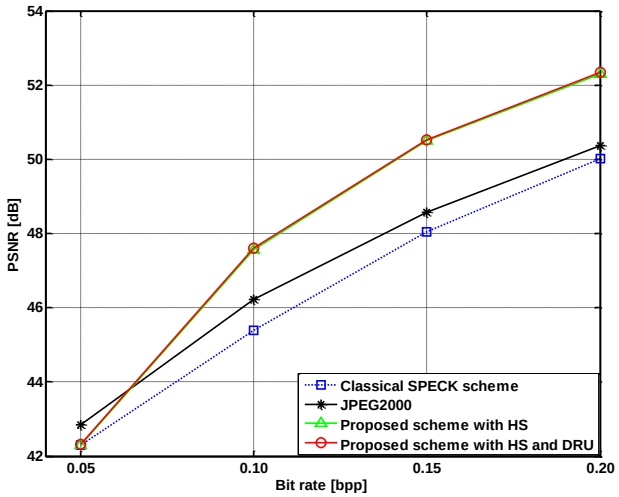


(a)

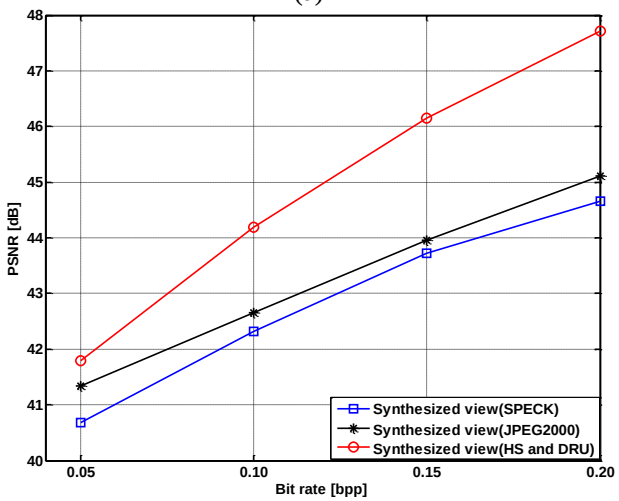


(b)

Fig. 8. The results of *Kendo*. (a) Depth map; (b) Synthesized view.

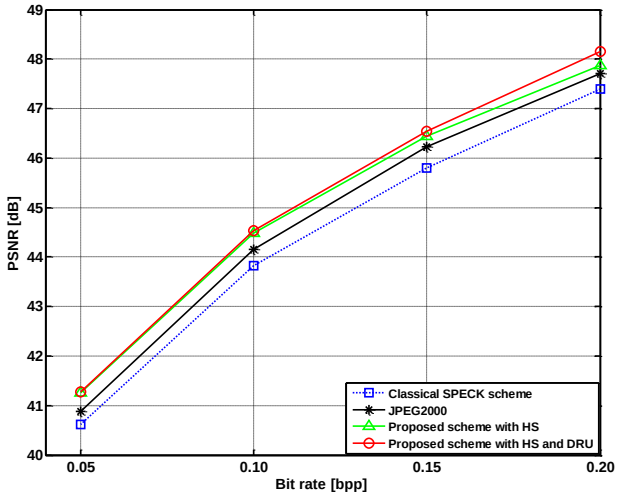


(a)

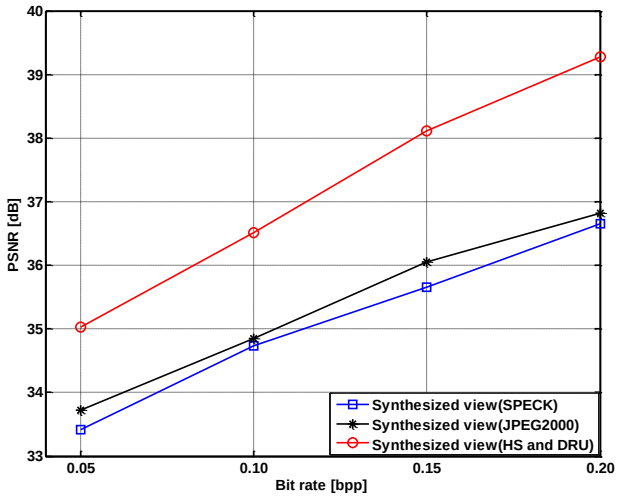


(b)

Fig. 9. The results of *BreakDancer*. (a) Depth map; (b) Synthesized view.



(a)



(b)

Fig. 10. The results of *Newspaper*. (a) Depth map; (b) Synthesized view.

The depth map is not displayed on the 3D screen, it is used in view synthesizing instead. Hence, from **Fig. 4 (b)** to **Fig. 10 (b)**, the results of the synthesized view with the classical SPECK and the proposed scheme are also included. The synthesized view renders by a simple warping process using the coded depth map and the original texture image [2] [23]. The results show that the proposed scheme is much better than the classical scheme with more than 3 dB gain. To demonstrate the robustness of the algorithm, another kind of warping algorithm, View Synthesis Reference Software (VSRS) 3.5 [24] [25], is used to render the intermediate view. For VSRS, the left and right view and the corresponding depth maps are required, in addition to the camera parameters. Using three public configuration files for the sequences of *UndoDancer*, *Balloons* and *Kendo*, **Fig. 6 (b)** to **Fig. 8 (b)** include the VSRS results. It can be seen that the synthesized results with the VSRS are much better than that of a simple warping algorithm. On one hand, the better results occur because the VSRS employs two views and two depth maps. On the other hand, one depth map is compressed while the second depth map remains untouched during rendering. From the results, this study concludes that all of the results with the two synthesis algorithms demonstrate similar trend and gain, which shows the efficiency and robustness of the proposed algorithm.

In this case, it is worth mentioning that the gain of the proposed scheme is smaller on a low bit rate because the edge information takes a higher percentage. For clarity, the bit rate of the edge and the percentage it takes is shown in **Table 1.**

Table 1. The total bit rate of the edge and the percentage it takes in

Sequence	edge rate	Edge rate percentage (0.05,0.10,0.15,0.20)			
Tsukuba	0.0181	36.21%	18.11%	12.07%	9.05%
Teddy	0.0293	58.60%	29.30%	19.53%	14.65%
UndoDancer	0.0134	26.80%	13.40%	8.93%	6.70%
Balloons	0.0019	3.88%	1.94%	1.29%	0.97%
Kendo	0.0111	22.20%	11.10%	7.40%	5.55%
BreakDancer	0.0177	35.40%	17.70%	11.80%	8.85%
Newspaper	0.1141	22.82%	11.41%	7.61%	5.71%

To present an average gain achieved by the proposed scheme, the BD rate [26] gain as compared to SPECK scheme is shown in the following tables. **Table 2** illustrates the BD gain on depth maps alone. It can be seen that a larger gain is achieved for the sparse depth maps, such as in Tsukuba. For the depth maps containing noises and holes, a relatively smaller gain is obtained.

Table 2. BD gains (Bjøntegaard delta) for depth sequences

Sequence	HS scheme (BD-YPSNR)	HS and DRU scheme (BD-YPSNR)
Tsukuba	5.170483	13.89135
Teddy	3.356747	3.36648
UndoDancer	3.163195	3.180728
Balloons	1.318387	1.691971
Kendo	0.604719	1.157352
BreakDancer	1.869406	1.904356
Newspaper	0.625167	0.710332

Table 3 shows the BD gain on the synthesized view. For the rendering algorithm with a single texture image plus a depth map, the gains are very significant, especially for the sparse depth maps. Not all the results are presented with VSRS algorithm because of limited public configuration files. Notice, for the depth maps with more noises such as *Newspaper*, even though the gain is smaller for the depth map alone, the gain of the synthesized performances remains at 1.88 dB. This is because the proposed scheme is successful at preserving the edge information, while the edge information is important for the view synthesizing. Because of employing two textures in addition to two VSRS depth images, the gains are relatively small. In addition, the results do not present the same trend as that of a single color plus single depth map case because of filtering and blending introduced into the VSRS. However, there is still a clear advantage to proposed scheme. Consequently, the whole performance testifies the efficiency of the proposed scheme.

Table 3. BD gains (Bjøntegaard delta) for synthesized views

Sequence	1 texture + 1 depth (BD-YPSNR)	2 texture + 2 depth (BD-YPSNR)
Tsukuba	10.532567	not available
Teddy	2.88635	not available
UndoDancer	3.289217	2.674933
Balloons	2.164233	4.008
Kendo	4.457617	1.15378
BreakDancer	1.9361	not available
Newspaper	1.8851	not available

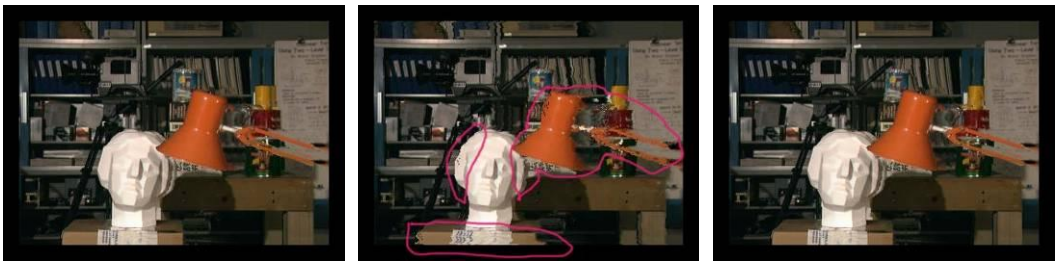
Fig. 11 and **Fig. 12** provide the subjective comparison of the synthesized results obtained by the classical and proposed schemes. From the difference between the classical scheme and proposed scheme, it can be found that the synthesized images with the proposed algorithm look clearer and smoother along object boundaries at the same bit-rate level. Therefore, the proposed scheme is also superior on the subjective performance. For more results on the other synthesized images, please refer to the website[1].

[1] https://www.dropbox.com/sh/o2n5jonqzl1yp5n/n4rnrcFQYO

4. Conclusions

By analyzing the characteristic of depth maps, an efficient depth map coding scheme based on histogram-based segmentation and depth range updating is proposed in this paper. The proposed histogram based segmentation scheme effectively separates depth maps into the foreground and background regions. The two regions are then separately encoded with rate distortion optimization. Compared with the classical coding scheme, 1-10 dB gain can be obtained. At the decoding end, the depth range updating scheme can correct some compression distortions by checking the original depth range and the threshold for segmentation. Hence, up to 4 dB gain can be achieved for sparse depth maps. The results of the synthesized view also testify the efficiency of the proposed scheme, with more than 3 dB gain on most of the depth maps.

With the segmented regions, 3D depth sensing could be edited by separately tuning the depth values in the two regions, which is one benefit of the proposed scheme. The unequal bit rate allocation can be implemented if the region of interest driven on the depth map is provided. However, only two regions are separated in this paper even though better performance and editing can be obtained.



(a) (b) (c)

Fig. 11. Synthesized view of *Tsukuba* (0.05 bpp): (a) Original; (b) Classical scheme; (c) Proposed HS and DRU scheme

(a) (b) (c)

Fig. 12. Synthesized view of *Teddy* (0.05 bpp): (a) Original; (b) Classical scheme; (c) Proposed HS and DRU scheme

References

[1] P. Merkle, A. Smolic, K. Muller, and T. Wiegand, "Multi-view video plus depth representation and coding," in *Proc. of IEEE International Conference on Image Processing (ICIP)*, pp. 201-204, 2007. Article (CrossRef Link)

[2] C. Fehn, "Depth-image-based rendering (DIBR), compression, and transmission for a new approach on 3D-TV," *in Proc. of SPIE 5291*, pp. 93-104, 2004. Article (CrossRef Link)

[3] W. Hu, G. Cheung, X. Li, and O. Au, "Depth map compression using multi-resolution graph-based transform for depth-image-based rendering," in *Proc. of IEEE International Conference on Image Processing (ICIP)*, pp. 1297-1300, 2012. Article (CrossRef Link)

[4] G. Shen, W.-S. Kim, A. Ortega, J. Lee, and H. Wey, "Edge-aware intra prediction for depth-map coding," in *Proc. of IEEE International Conference on Image Processing (ICIP)*, pp. 3393-3396, 2010. Article (CrossRef Link)

[5] M. Maitre and M. N. Do, "Depth and depth-color coding using shape-adaptive wavelets," *J. Vis. Comun. Image Represent.*, vol. 21, no. 5-6. pp. 513-522, 2010. Article (CrossRef Link)

[6] Morvan Y., Farin, D. de With P.H.N., "Depth-image compression based on an R-D optimized quadtree decomposition for the transmission of multiview images," in *Proc. of IEEE International Conference on Image Processing (ICIP)*, pp. 105-108, 2007. Article (CrossRef Link)

[7] Shinya Shimizu, Hideaki Kimata, Shiori Sugimoto and Norihiko Matsuura, "Block-adaptive palette-based prediction for depth map coding," in *Proc. of IEEE International Conference on Image Processing (ICIP)*, pp. 117-120, 2011. Article (CrossRef Link)

[8] P. Merkle, C. Bartnik, K. Muller, D. Marpe, and T. Wiegand, "3D video: Depth coding based on inter-component prediction of block partitions," in *Proc. of Picture Coding Symposium (PCS)*, pp. 149-152, 2012. Article (CrossRef Link)

[9] K. Muller, P. Merkle, G. Tech, and T. Wiegand, "3D video coding with depth modeling modes and view synthesis optimization," in *Proc. of Asia-Pacific Signal Information Processing Association Annual Summit and Conference*, pp. 1-4, 2012. Article (CrossRef Link)

[10] F. Jager, M. Wien, and P. Kosse, "Model-based intra coding for depth maps in 3D video using a depth lookup table," in *Proc. of 3DTV-Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON)*, pp. 1-4, 2012. Article (CrossRef Link)

[11] W. Pearlman, A. Islam, N. Nagaraj, and A. Said, "Efficient, low-complexity image coding with a set-partitioning embedded block coder," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 11, pp. 1219-1235, 2004. Article (CrossRef Link)

[12] F. Ono, W. Rucklidge, R. Arps, and C. Constantinescu, "JBIG2-the ultimate bi-level image coding standard," in *Proc. International Conference on Image Processing (ICIP)*, pp. 140-143, 2000. Article (CrossRef Link)

[13] R. Gonzalez and R. Woods, Digital image processing (2nd edition), Prentice-Hall Inc., 2002.

[14] S. Li and W. Li, "Shape-adaptive discrete wavelet transforms for arbitrarily shaped visual object coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 10, no. 5, pp. 725-743, 2000. Article (CrossRef Link)

[15] D. I. Cohen, A. and J.-C. Feauveau, "Biorthogonal bases of compactly supported wavelets," *Communications on Pure and Applied Mathematics*, vol. 45, no. 5, pp. 485-560, 1992. Article (CrossRef Link)

[16] A. Said and W. Pearlman, "A new, fast, and efficient image codec based on set partitioning in hierarchical trees," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, no. 3, pp. 243-250, 1996. Article (CrossRef Link)

[17] W. S. Kim, A. Ortega, P. Lai, D. Tian, and C. Gomila, "Depth map distortion analysis for view rendering and depth coding," in *Proc. of IEEE International Conference on Image Processing (ICIP)*, pp. 721-724, 2009. Article (CrossRef Link)

[18] B. T. Oh, J. Lee, and D.-S. Park, "Depth map coding based on synthesized view distortion function," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 7, pp. 1344-1352, 2011. Article (CrossRef Link)

[19] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo

correspondence algorithms," *Int. J. Comput. Vision*, vol. 47, no. 1-3, pp. 7-42, 2002. Article (CrossRef Link)
[20] C. L. Zitnick, S. B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski, "High-quality video view interpolation using a layered representation," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 600-608, 2004. Article (CrossRef Link)
[21] M. Adams and F. Kossentini, "Jasper: a software-based JPEG-2000 codec implementation," in *Proc. of International Conference on Image Processing (ICIP)*, pp. 53-56, 2000. Article (CrossRef Link)
[22] Oliver, J.; Malumbres, M.P., "Low-Complexity Multiresolution Image Compression Using Wavelet Lower Trees," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 11, pp. 1437-1444, 2006. Article (CrossRef Link)
[23] L. Zhang and W. J. Tam, "Stereoscopic image generation based on depth images for 3D TV," *IEEE Transactions on Broadcasting*, vol. 51, no. 2, pp. 191-199, 2005. Article (CrossRef Link)
[24] C. Lee and Y. S. Ho, "View synthesis tools for 3D video," *ISO/IEC JTC1/SC29/WG11 MPEG2008/M15851*, 2008.
[25] I. JTC1/SC29/WG11, "View synthesis algorithm in view synthesis reference software 2.0 (VSRS2.0)," *Doc. M16090*, 2009.
[26] G. Bjntegaard, "Improvements of the BD-PSNR model," *ITU-T SG16 Q.6 Documen*t, VCEG-AI11, 2008.

Chunyu Lin was born in LiaoNing Province, China. He works as a lecturer in Beijing Jiaotong University. He obtained his doctor degree in Beijing Jiaotong University in 2011. From 2009 to 2010, he was a visiting researcher at the ICT group of Delft University of Technology, Netherlands. From 2011 to 2012, He was a postdoc in Gent University, Belgium. His research interests are in the areas of image/video compression and robust transmission, stereo matching and 3D video coding.

Yao Zhao received the BS degree from Fuzhou University, China, in 1989, and the ME degree from Southeast University, Nanjing, China, in 1992, both from the Radio Engineering Department, and the PhD degree from the Institute of Information Science, Beijing Jiaotong University (BJTU), China, in 1996. He became an associate professor at BJTU in 1998 and became a professor in 2001. From 2001 to 2002, he was a senior research fellow with the Information and Communication Theory Group, Faculty of Information Technology and Systems, Delft University of Technology, Delft, The Netherlands. He is currently the director of the Institute of Information Science, BJTU. His current research interests include image/video coding, digital watermarking and forensics, and video analysis and understanding. He serves on the editorial boards of several international journals, including as associate editors of IEEE Transactions on Cybernetics, IEEE Signal Processing Letters, and an area editor of Signal Processing: Image Communication (Elsevier), etc. He was named a distinguished young scholar by the National Science Foundation of China in 2010, and was elected as a Chang Jiang Scholar of Ministry of Education of China in 2013. He is a senior member of the IEEE.

Jimin Xiao received the B.S. and M.E. degrees in telecommunication engineering from Nanjing University of Posts and Telecommunications, Nanjing, China, in 2004 and 2007, respectively, and the dual Ph.D. degree in electrical engineering and electronics from University of Liverpool, Liverpool, U.K., in 2013. He served as a Visiting Researcher at Nanyang Technological University, Singapore. From Nov. 2013 to Nov. 2014, he was a senior researcher in the Department of Signal Processing, Tampere University of Technology, Finland, and external researcher in Nokia Research Center, Tampere, Finland. Currently, he was a lecturer in Xi'an Jiaotong–Liverpool University. His research interests are in the areas of video streaming, image and video compression, and multiview video coding.

Tammam Tillo received eived the Dipl.Ing. degree in electrical engineering from Damascus University, Damascus, Syria, in 1994 and the Ph.D. degree in electronics and communication engineering from Politecnico di Torino, Turin, Italy, in 2005. He was a Visiting Researcher with École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland, in 2004 and was a Post-Doctoral Researcher with the Image Processing Laboratory, Politecnico di Torino, from 2005 to 2008. For a few months, he was an Invited Research Professor with Digital Media Laboratory, Sungkyunkwan University, Seoul, Korea, before joining Xi'an Jiaotong–Liverpool University (XJTLU), Suzhou, China, in 2008. He was promoted to Full Professor in 2012. His research interests include robust transmission of multimedia data, image and video compression, and hyperspectral image compression.