Information Sciences 269 (2014) 159–175

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Reversible data hiding for depth maps using the depth no-synthesis-error model

Kuo-Liang Chung [a,1], Wei-Jen Yang [a,*], Wei-Ning Yang [b]

[a] Department of Computer Science and Information Engineering, National Taiwan University of Science and Technology, No. 43 Section 4, Keelung Road, Taipei, Taiwan 10672, ROC
[b] Department of Information Management, National Taiwan University of Science and Technology, No. 43 Section 4, Keelung Road, Taipei, Taiwan 10672, ROC

ARTICLE INFO

Article history:
Received 1 April 2013
Received in revised form 29 November 2013
Accepted 29 December 2013
Available online 7 January 2014

Keywords:
3D video
Depth map
D-NOSE model
Difference expansion
Histogram modification
Reversible data hiding

ABSTRACT

When embedding hidden data in 3D images, conventional reversible data hiding methods, which are designed for 2D gray and color images, can be applied to color and depth maps. However, directly applying these methods to depth maps may cause synthesis errors and lead to visual artifacts in the rendered virtual views. Two novel reversible data hiding methods based on the depth no-synthesis-error (D-NOSE) model are proposed to embed hidden data in the depth maps of 3D images. The proposed methods can preserve the quality of the rendered virtual view and achieve substantially higher embedding capacity. Experimental results show that the proposed methods achieve better performance than the existing state-of-the-art methods in terms of both embedding capacity and the quality of the rendered virtual views.

1. Introduction

Data hiding has become a promising technique for embedding secret data in an image. Conventional data hiding methods were specifically developed for gray and color images [6,11,12,17,19,21,34–36]. However, conventional data hiding usually induces irreversible distortions and hence leads to a disagreeable effect after extracting hidden data from the marked images. For reversibility, Barton [5] first developed a reversible data hiding technique, which recovers the original image after extracting the hidden data. Most reversible data hiding methods emphasize maximizing the embedding capacity while preserving the quality of the marked images. According to the format of the input images, the developed reversible data hiding methods can be classified into two categories: compression domain-based methods [7–10] and spatial domain-based methods [1,15,16,20,22,24,27–31,33,37]. The former embed the hidden data in the compressed images and the latter in the pixels of the original images. The spatial domain-based methods usually achieve higher embedding capacity and better quality of the marked images.

Compression domain-based methods embed the hidden data in the compressed information, such as discrete cosine transformation coefficients [8,10], vector quantization indices [7], and common bitmaps [9]. Since the hidden data are embedded in the compressed images, the compression domain-based methods usually produce lower embedding capacity and worse quality of the marked images. The spatial domain-based methods, which embed hidden data in the pixels by using

* Corresponding author.
E-mail addresses: klchung01@gmail.com (K.-L. Chung), wjyang@mail.ntust.edu.tw (W.-J. Yang).
[1] Supported by the National Science Council of the ROC under Contracts NSC 99-2221-E-011-078-MY3 and NSC 101-2221-E-011-139-MY3.

0020-0255/$ - see front matter © 2014 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.ins.2013.12.049

the distribution of pixel values, can be divided into three categories: the difference expansion-based category, the histogram modification-based one, and the difference expansion- and histogram modification-based one.

The difference expansion method, first proposed by Tian [31], embeds the hidden data by expanding the difference in the gray values of two adjacent pixels, which are the high-frequency coefficients generated by integer-Haar-wavelet-transform [23]. Alattar [1] generalized Tian's method by embedding the hidden data in the high-frequency coefficients which are generated by any integer transform on gray values of successive pixels. Sachnev et al. [27] improved the embedding capacity of Alattar's method by embedding the hidden data in 2D quad pixels and further simplifying the location map. Kamstra and Heijmans [15] improved Tian's method by using the low-frequency coefficients to select the suitable expanding pixels, leading to substantial reduction in the distortion of the marked images. Kim et al. [16] improved Kamstra and Heijmans' method by developing a novel difference expansion transform with a simplified location map. Since the distortions of the marked images generated by the difference expansion methods depend on the difference magnitude, difference expansion methods may suffer from poor quality of the marked images. Furthermore, difference expansion methods usually involve more computations in generating the difference values by integer transforms and selecting the suitable expanding pixels.

The histogram modification method, first proposed by Ni et al. [24], embeds the hidden data by modifying the histogram of the image via shifting the bins between the maximal peak and the neighboring valley. Ni et al.'s method involves lower computational complexity and the theoretical lower bound on the quality of the marked image can be easily derived. Since higher peaks of the histogram imply larger embedding capacity, histogram modification methods can be improved by raising the peaks of the histogram. Lin et al. [20] achieved this using the difference image and hence increased the embedding capacity of Ni et al.'s method. Tai et al. [29] further improved Lin et al.'s method by removing the requirement of sending the multiple peak points to the extraction side. An et al. [2] applied the clustering scheme in the wavelet domain to raise the peaks of the histogram. In addition, An et al. [3] developed a novel statistical quantity histogram shifting and clustering-based method to enhance the visual quality and embedding capacity. Gao et al. [14] proposed a generalized statistical quantity histogram to achieve better performance for different kinds of images and for different capacity requirements. To solve the salt-and-pepper noise problem and achieve the tradeoff between invisibility and robustness, An et al. [4] proposed a content-adaptive reliable robust lossless data embedding method. Since only the peak bins are used to embed hidden data, the histogram modification methods usually suffer from limited embedding capacity.

Thodi and Rodriguez [30] integrated difference expansion and histogram modification by embedding, using difference expansion, the hidden data in the pixels with small prediction errors, avoided the problem of ambiguity by applying the histogram modification to the pixels with large prediction errors. This integrated method, usually referred to as the prediction error-based method, can result in higher embedding capacity and better quality of the marked images. Sachnev et al.'s integrated method [28] reduced the prediction errors to increase the embedding capacity by averaging the gray values of the four neighboring pixels in a rhombus shape. Luo et al.'s integrated method [22] reduced the prediction errors to increase the embedding capacity by using the interpolation-based prediction scheme. Li et al. [18] developed an integrated method which applies the adaptive difference expansion technique to increase the embedding capacity. Yang et al. [37] further developed an integrated reversible data hiding method based on the spectral-spatial correlation in the color difference domain [26] for color filter array mosaic images. Nowadays, prediction error-based methods have become the dominant methods in reversible data hiding fields because of their high embedding capacity and the good quality of the marked images.

Due to the advent of advanced technology, three-dimensional (3D) image representation systems have received considerable attention in the electronics market, and therefore equipping 3D images with a secret data embedding capability has become an important issue. Traditional 3D images are constructed by the multi-view representation which mimics the perception of human eyes. However, multi-view representation involves large storage and transmission requirements, and does not allow individual viewers to adjust the disparity range of stereoscopic videos based on their preference for the intensity of the 3D perception. To alleviate these problems, instead of multi-view representation, 3D image systems often exploit a classical 2D color image plus a depth map [38] to provide viewers with the immersive perception of a 3D scene. The view synthesis technique utilizes the color image and the depth map to generate a virtual view, which is then combined with the color image to construct the 3D image. The most well-known view synthesis technique is the depth-image-based rendering (DIBR) [13] and the generated virtual view is called the rendered virtual view. The 3D image with the format consisting of a color image plus a depth map involves low storage and transmission requirements, and allows individual viewers to adjust the disparity for comfortable 3D perception. When embedding hidden data in a 3D image with the format consisting of a color image plus a depth map, in addition to directly applying the conventional reversible data hiding methods mentioned above to the three color channels of the color image, it is natural to consider embedding hidden data in the depth map for maximizing the embedding capacity. However, directly applying conventional reversible data hiding methods to the depth maps may cause synthesis errors from changing the depth values, leading to visual artifacts in the rendered virtual view. This motivates us to develop, for the depth map, a reversible data hiding method which can achieve high embedding capacity without causing any synthesis errors after embedding.

In this paper, we propose two novel reversible data hiding methods, based on the difference expansion and histogram modification, respectively, for depth maps where the depth no-synthesis-error (D-NOSE) model [39] is used to avoid synthesis errors. The proposed two methods based on the D-NOSE model assure that each marked depth value falls within the allowable range, determined by the D-NOSE model, leading to no synthesis errors in the rendered virtual view. Thus, the rendered virtual view generated from the marked depth map, after embedding, is identical to that from the original depth

map. That is, embedding hidden data does not affect the quality of the rendered virtual view. Furthermore, due to the inherent local homogeneity characteristic of depth maps, the histogram of a depth map is sparse and the prediction errors are usually small, implying that the proposed embedding methods, using the histogram modification and difference expansion, can achieve high embedding capacity. Experimental results on typical 3D images with a format consisting of a color image plus a depth map demonstrate that the proposed methods achieve better performance in terms of both objective and subjective measures when compared to three conventional state-of-the-art methods [24,28,30].

The rest of this paper is organized as follows. In Section 2, a brief introduction to the DIBR view synthesis technique and the D-NOSE model is given. Section 3 presents the proposed two reversible data hiding methods for depth maps. In Section 4, we report the experimental results to demonstrate the superiority of the proposed data hiding methods. Section 5 addresses some concluding remarks.

2. The DIBR view synthesis technique and the D-NOSE model

In Section 2.1, we first introduce the DIBR view synthesis technique, and then a brief introduction to the D-NOSE model is given in Section 2.2.

2.1. DIBR view synthesis technique

The DIBR view synthesis technique [13] is used for view synthesis in 3D representation systems. The DIBR consists of two key steps, namely, warping and hole-filling. The warping step generates a left or right rendered virtual view according to the color image and depth map in the original view. Due to the fact that warping is not a one-to-one mapping, hole-filling is then used to polish the rendered virtual view. Since our research emphasizes the influence of data hiding on warping for the depth map, we focus on the warping technique in this sub-section.

The warping procedure transforms a pixel with coordinate vector $(x,y)^T$ in the color image of the original view to the pixel with coordinate vector $(x',y')^T$ in the rendered virtual view, where the superscript T denotes the transpose of a vector. Since we only consider the left or right rendered virtual view, we focus on the horizontal disparity $d_x = x' - x$ between the original and rendered virtual views.

The perspective projection equation, which is shown in Eq. (1), describes how to transform a world coordinate vector $(X,Y,Z)^T$ of a 3D space point to the camera coordinate vector $(x,y,z)^T$. The color and the corresponding depth value z of a 3D space point are recorded, respectively, in the point with coordinate $(x,y)^T$ in the 2D color image and depth map. Thus, the color image and depth map form the real camera coordinate system. The perspective projection equation can be expressed as

$$z\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = A\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \tag{1}$$

where

$$A = \begin{bmatrix} f & \tau & o_x \\ 0 & \eta f & o_y \\ 0 & 0 & 1 \end{bmatrix}$$

denotes the intrinsic camera parameter matrix with f denoting the focal length, $(o_x,o_y)^T$ the principal-point position, and parameters η and τ modeling the aspect ratio and skew of pixels. In practice, when employing modern digital cameras, it can be safely assumed that pixels are square ($\eta = 1$) and non-skewed ($\tau = 0$).

For human eyes to perceive 3D images, a virtual view with horizontal disparity represented in the camera coordinate system is required. The camera coordinate vector $(x',y',z')^T$ in the virtual view with horizontal disparity can be expressed by

$$z'\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = AR\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} + At \tag{2}$$

where the rotation matrix R equals the identity matrix, i.e. $R = I$, under the horizontal disparity assumption, and $t = [\ell, 0, 0]^T$ represents the translation vector with ℓ (<0) being the baseline length between the left real camera and the right virtual camera.

Substituting Eq. (1) into Eq. (2) gives the classical affine disparity equation, which defines the depth-dependent relation between corresponding points with coordinate vectors $(x,y,z)^T$ and $(x',y',z')^T$ in the real camera coordinate system and virtual camera coordinate system, respectively, yielding

$$z'\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = z\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} + At. \tag{3}$$

By simple algebra, we can rewrite the affine disparity equation as

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} x + d_x \\ y \\ z \end{pmatrix}, \tag{4}$$

where $d_x = f\ell/z$ denotes the associated horizontal disparity, implying that, for each pixel in the real and virtual camera coordinate systems, there only exists horizontal disparity which depends on the depth value and the pixel position in the real camera coordinate system. In order to save storage space, we only store the color and depth maps of the original view; we then render the virtual view according to Eq. (4). Note that the depth map in the original view is used for generating the rendered virtual view. Only the color maps in the original and rendered virtual views are required for human eyes to perceive a 3D image.

According to Eq. (1), the depth value z in the original view is the same as the depth value Z in the world coordinate system. So as to save on storage, the depth value z in the original view is usually converted to a quantized depth value z_q by the quantization function used in MPEG-3DV [32]

$$z_q = Q(z) = \left\lfloor 255 \times \frac{Z_{near}}{z} \times \frac{Z_{far} - z}{Z_{far} - Z_{near}} + 0.5 \right\rfloor \tag{5}$$

where Z_{near} and Z_{far} are the nearest and farthest real depth values in the original view and $\lfloor x \rfloor$ is the greatest integer less than or equal to x. Values 0 and 255 of the quantized depth value correspond, respectively, to the farthest pixel and the nearest pixel for the camera. To generate the rendered virtual view according to Eq. (4), the quantized depth value z_q needs to be dequantized by

$$z = Q^{-1}(z_q) = \frac{1}{\frac{z_q}{255}\left(\frac{1}{Z_{near}} - \frac{1}{Z_{far}}\right) + \frac{1}{Z_{far}}} \tag{6}$$

to calculate the horizontal disparity d_x $(= f/Q^{-1}(z_q))$. Since the horizontal disparity is restricted to being an integer for digital images, based on the D-NOSE model [36], there exist different quantized depth values corresponding to the same horizontal disparity.

2.2. The D-NOSE model

When performing warping, the quantized depth value z_q is first dequantized to $Q^{-1}(z_q)$ by Eq. (6). The dequantized depth value $Q^{-1}(z_q)$ is then used for calculating the horizontal disparity $d_x = f\ell/Q^{-1}(z_q)$. The λ-rounding $(0 < \lambda \leqslant 1)$ with an integer-precision equation for horizontal disparity for d_x can be expressed as

$$\tilde{d}_x(z_q) = \lceil d_x - \lambda \rceil = \left\lceil \frac{f\ell}{Q^{-1}(z_q)} - \lambda \right\rceil, \tag{7}$$

which is called the rounded horizontal disparity, where $\lceil x \rceil$ denotes the smallest integer greater than or equal to x.

Since the rounded disparity $\tilde{d}_x(z_q)$ is not a one-to-one transformation from the quantized depth value z_q, for a fixed quantized depth value z_q^0, there exist different quantized depth values corresponding to the same rounded disparity $\tilde{d}_x(z_q^0)$. That is, for each quantized depth value z_q, there exists a set $\{v|\tilde{d}_x(v) = \tilde{d}_x(z_q)\}$ within which varying the quantized depth value yields the common rounded horizontal disparity. Since $\tilde{d}_x(z_q)$ is a monotonically non-decreasing function in z_q, for each quantized depth value z_q, there exists a unique range $[\ell(z_q), u(z_q)]$ such that

$$\ell(z_q) = \min\{v|\tilde{d}_x(v) = \tilde{d}_x(z_q)\}$$
$$u(z_q) = \max\{v|\tilde{d}_x(v) = \tilde{d}_x(z_q)\}.$$

The D-NOSE model states that, for each quantized depth value z_q, varying the quantized depth value within $[\ell(z_q), u(z_q)]$ yields the same rounded horizontal disparity and hence does not result in any synthesis error in the rendered virtual view. Note that the D-NOSE model delivers the same allowable range for each quantized depth value falling within the allowable range.

3. Proposed reversible data hiding method for depth Maps

In this section, we propose two reversible data hiding methods, based on difference expansion and histogram modification, respectively, by applying the D-NOSE model to depth maps. For convenience, the two reversible data hiding methods are, respectively, called the D-NOSE-and-difference expansion-based (DNDE-based) and the D-NOSE-and-histogram modification-based (DNHM-based) reversible data hiding methods. The key characteristics of the proposed methods are twofold: to avoid synthesis errors in the DIBR stage and to maximize the embedding capacity over the allowable range provided by the D-NOSE model.

3.1. Proposed DNDE-based reversible data hiding method

In this sub-section, we present the proposed DNDE-based reversible data hiding method. We first describe the proposed hidden data embedding and extraction strategies, and then the overhead embedding and extraction strategies. Finally, the whole procedure of the proposed method is given.

3.1.1. Hidden data embedding and extraction strategies

For describing the hidden data embedding and extraction strategies of the proposed method based on the difference expansion where prediction errors are used as differences, each pixel in the depth map is marked as ▲ or ○ and may belong, respectively, to set Ω_1 or Ω_2 as depicted in Fig. 1. These two sets are disjoint and the quantized depth value of each pixel in Ω_1 is predicted using the quantized depth values of the neighboring pixels in Ω_2 and vice versa.

Let $z_q(i,j)$ denote the quantized depth value of the pixel located at position (i,j) in the depth map. When predicting the quantized depth value of the pixel located at position (i,j), in addition to using the depth information of the neighboring pixels, the effect of the edge information is also considered. The predicted quantized depth value of the pixel located at position (i,j) can be expressed as

$$\widehat{z}_q(i,j)=\begin{cases}\left\lfloor\frac{z_q(i-1,j)+z_q(i+1,j)}{2}\right\rfloor, & \text{if } \frac{|z_q(i-1,j)-z_q(i+1,j)|}{|z_q(i,j-1)-z_q(i,j+1)|}<\alpha\\ \left\lfloor\frac{z_q(i,j-1)+z_q(i,j+1)}{2}\right\rfloor, & \text{if } \frac{|z_q(i,j-1)-z_q(i,j+1)|}{|z_q(i-1,j)-z_q(i+1,j)|}<\alpha\\ \left\lfloor\frac{z_q(i-1,j)+z_q(i+1,j)+z_q(i,j-1)+z_q(i,j+1)}{4}\right\rfloor, & \text{otherwise},\end{cases} \tag{8}$$

where $\lfloor x\rfloor$ denotes the rounded value of x and the determination of the parameter α is discussed in A. Then, the prediction error can be expressed as

$$e(i,j)=z_q(i,j)-\widehat{z}_q(i,j). \tag{9}$$

To assure that no synthesis errors appear in the rendered virtual view, the depth value $z'_q(i,j)$ of the marked pixel located at position (i,j) needs to fall within the allowable range $[\ell(z_q(i,j)),u(z_q(i,j))]$ obtained from the D-NOSE model. Since a large portion of the depth map is often homogeneous, most prediction errors assume the value 0, as shown in Fig. 2. For increasing the embedding capacity, in addition to the prediction errors with value 0, the proposed method selects the prediction errors with value -1 or 1, depending on which has the higher frequency. For convenience, we only consider the case of embedding hidden data using the prediction errors with values 0 and -1. To maximize the embedding capacity, the number of embedding bits for the pixel located at position (i,j) is determined by

$$n(i,j)=\begin{cases}\left\lfloor\log_2\left(\widehat{z}_q(i,j)-\ell(z_q(i,j))+1\right)\right\rfloor-1, & \text{if } e(i,j)=-1\\ \left\lfloor\log_2\left(u(z_q(i,j))-\widehat{z}_q(i,j)+1\right)\right\rfloor, & \text{if } e(i,j)=0\\ 0, & \text{otherwise}.\end{cases} \tag{10}$$

When embedding hidden data by the difference expansion, modifying the prediction error $e(i,j)$ according to the following rule:

$$e'(i,j)=\begin{cases}2^{n(i,j)}\times e(i,j)-h(i,j)=-2^{n(i,j)}-h(i,j), & \text{if } e(i,j)=-1\\ 2^{n(i,j)}\times e(i,j)+h(i,j)=h(i,j), & \text{if } e(i,j)=0\end{cases} \tag{11}$$

with hidden data $h(i,j)\in\{0,1,\ldots,2^{n(i,j)}-1\}$, which ensures that the quantized depth value of the marked pixel

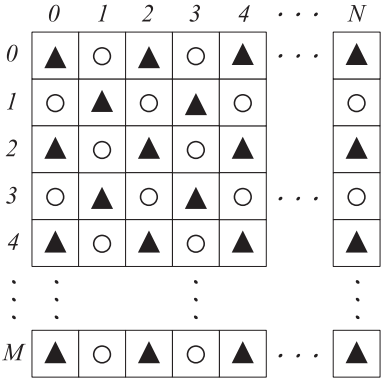

Fig. 1. Pixels ▲ $\in\Omega_1$ and ○ $\in\Omega_2$.

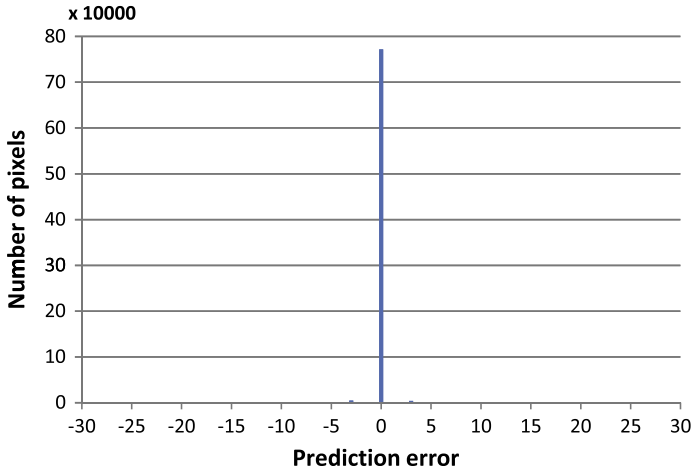

Fig. 2. Histogram of prediction errors in 3D Kendo image.

$$z'_q(i,j) = \widehat{z}_q(i,j) + e'(i,j), \tag{12}$$

falls within the allowable range $[\ell(z_q(i,j)), u(z_q(i,j))]$.

In order to be able to extract the hidden data, we need to record if each pixel embeds the hidden data. When embedding hidden data using prediction errors with values 0 and -1, denote by M the location map which records if the pixel located at position (i,j) can be used to embed the hidden data with

$$M(i,j) = \begin{cases} 0, & \text{if } e(i,j) \in \{0,-1\} \\ 1, & \text{otherwise.} \end{cases} \tag{13}$$

Since the proportions of 0 and 1 in the location map are usually very different, the storage size of the location map can be substantially reduced by compressing the location map using arithmetic coding [25]. The ratio of the storage size of the compressed location map over the available embedding capacity for nine 3D test images with the format consisting of a color image plus a depth map are given in Table 1. It is clear that the ratios are less than 10%. Small ratios indicate that the overhead incurred by using the location map is not significant.

After embedding the hidden data in the pixels in Ω_1, we can similarly embed the hidden data in the pixels in Ω_2 based on the neighboring marked pixels in Ω_1. Thus, when extracting the hidden data, we first extract from the pixels in Ω_2 and then from the pixels in Ω_1.

Since the pixel with $M(i,j) = 0$ can be used for embedding the hidden data, we only consider the extraction from the pixel with $M(i,j) = 0$. Given the quantized depth value $z'_q(i,j)$ of the marked pixel with $M(i,j) = 0$, we describe in detail how to extract the hidden data $h(i,j)$ and recover the original quantized depth value $z_q(i,j)$ on the extraction side. We first compute the predicted depth value $\widehat{z}_q(i,j)$, by Eq. (12), for each pixel in Ω_2 based on the depth values of the neighboring marked pixels in Ω_1. Then, we can obtain the modified prediction error $e'(i,j) = z'_q(i,j) - \widehat{z}_q(i,j)$ whose sign is used to determine the original prediction error $e(i,j)$ according to Eq. (11).

Associated with the allowable range $[\ell(z_q(i,j)), u(z_q(i,j))]$ by the D-NOSE model, $e(i,j)$ can be used to determine the number of hidden bits $n(i,j)$ by Eq. (10). Finally, the hidden data $h(i,j)$ can be extracted by

$$h(i,j) = e'(i,j) \bmod 2^{n(i,j)}. \tag{14}$$

Table 1
Ratio of the storage size of compressed location map over the available embedding capacity for nine 3D test images.

3D image	Size of compressed location map (bit)	Available embedding capacity (bit)	Ratio (%)
Mobile	43,384	520,817	8.329
Kendo	84,229	1,100,918	7.651
Balloons	81,342	1,100,894	7.389
Newspaper	65,152	774,151	8.416
Breakdancer	17,510	586,937	2.983
Ballet	15,893	542,682	2.929
Hall	6914	900,930	0.767
Undodancer	160,999	1,861,069	8.650
Street	88,028	898,448	9.798

Once the hidden data $h(i,j)$ is extracted, the original quantized depth value of the pixel can be recovered by

$$z_q(i,j) = \widehat{z}_q(i,j) + e(i,j). \tag{15}$$

After recovering the quantized depth values of pixels in Ω_2, we can similarly extract the hidden data and recover the quantized depth values of pixels in Ω_1. In the next sub-section, we address the embedding and extraction strategies for the overheads including the compressed location map, the hidden-data size, and the overhead size.

3.1.2. Overhead embedding and extraction strategies

When using the proposed method to embed hidden data, extra overheads should be addressed. In addition to the hidden data, extra overheads including the compressed location map of size μ, the hidden-data size ρ and the overhead size $\tau = \mu + \rho$ should be embedded in the depth map. Conventional reversible data hiding methods usually embed the overheads in the first $\mathbb{H}$ ($\geqslant \tau$) pixels by LSB replacement [31], and then use an extra correction bit stream to record the LSB values that are replaced by the overheads. However, overhead embedding in the depth map based on the LSB replacement may cause synthesis errors. To avoid such errors, we propose a novel overhead embedding strategy based on the same D-NOSE model. Since we should preserve the first $\mathbb{H}$ pixels of a depth map to embed the overheads, the embedding for hidden data starts from the $(\mathbb{H}+\mathbb{1})$-st pixel.

When embedding the overheads, first divide the allowable range $[\ell(z_q(i,j)), u(z_q(i,j))]$, corresponding to the quantized depth value $z_q(i,j)$, into $\lfloor \frac{u(z_q(i,j))-\ell(z_q(i,j))+1}{2} \rfloor$ disjoint subranges, denoted by $[\ell_\gamma(z_q(i,j)), u_\gamma(z_q(i,j))]$ for $\gamma = 1, 2, \ldots, \lfloor \frac{u(z_q(i,j))-\ell(z_q(i,j))+1}{2} \rfloor$. Then, when $z_q(i,j) \in [\ell_\gamma(z_q(i,j)), u_\gamma(z_q(i,j))]$, to assure that the marked quantized depth value $z'_q(i,j)$ falls within the allowable range $[\ell(z_q(i,j)), u(z_q(i,j))]$, embed the overheads such that

$$z'_q(i,j) = \begin{cases} \ell_\gamma(z_q(i,j)), & \text{if } o(i,j) = 0 \\ u_\gamma(z_q(i,j)), & \text{if } o(i,j) = 1, \end{cases} \tag{16}$$

where $o(i,j)$ denotes the overhead bit embedded in $z_q(i,j)$. Note that the pixel with $z_q(i,j) = u(z_q(i,j))$ and $u(z_q(i,j)) - \ell(z_q(i,j))$ occurring as even will not be used for embedding the overheads.

To recover the original quantized depth value at the extraction side, a correction bit is required to record if the quantized depth value equals the upper or lower bound of a specific subrange. Denote by $c(i,j)$ the correction bit corresponding to the quantized depth value $z_q(i,j)$ falling within the specific subrange $[\ell_\gamma(z_q(i,j)), u_\gamma(z_q(i,j))]$ such that

$$c(i,j) = \begin{cases} 0, & \text{if } z_q(i,j) = \ell_\gamma(z_q(i,j)) \\ 1, & \text{if } z_q(i,j) = u_\gamma(z_q(i,j)). \end{cases} \tag{17}$$

On the extraction side, to extract the hidden data, we first extract the overheads. Given the quantized depth value $z'_q(i,j)$ of the marked depth map, since the D-NOSE model delivers the common allowable range for each quantized depth value falling within the range, we can determine the allowable range $[\ell(z_q(i,j)), u(z_q(i,j))]$ according to $z'_q(i,j)$. Then, by Eq. (16), we can locate the specific subrange $[\ell_\gamma(z_q(i,j)), u_\gamma(z_q(i,j))]$ in which $z'_q(i,j)$ falls, and finally extract the overhead $o(i,j) = 0$ if $z'_q(i,j) = \ell_\gamma(z_q(i,j))$ and $o(i,j) = 1$, otherwise. As for recovering the original quantized depth value, given the marked depth value $z'_q(i,j)$ and the correction bit $c(i,j)$, the original quantized depth value can be recovered by

$$z_q(i,j) = \begin{cases} \ell_\gamma(z_q(i,j)), & \text{if } c(i,j) = 0 \\ u_\gamma(z_q(i,j)), & \text{if } c(i,j) = 1. \end{cases} \tag{18}$$

3.1.3. The whole procedure of the proposed DNDE-based method

We now describe all of the steps for the proposed DNDE-based reversible data hiding method. The embedding process is first applied on Ω_1 and then Ω_2 while the extraction process is first applied on Ω_2 and then Ω_1. Note that the number $\mathbb{H}$ ($\geqslant \tau$) of pixels used to embed overheads is given on both the embedding and extraction sides.

Given the depth map and the data to be embedded, four steps in the embedding process are followed to generate the marked depth map.

Step 1: Construct the location map M according to Eq. (13), and compress M by arithmetic coding.
Step 2: Figure out the overheads including the compressed location map, the hidden-data size μ, and the overhead size τ.
Step 3: Embed the overheads in the first $\mathbb{H}$ pixels according to the overhead embedding strategy described in Section 3.1.2, and obtain the correction bits by Eq. (17).
Step 4: Starting from the $(\mathbb{H}+\mathbb{1})$-st pixel, embed the correction bits and the hidden data according to the hidden data embedding strategy described in Section 3.1.1.

Given the marked depth map, four steps in the extraction process are followed to extract the hidden data and recover the original depth map.

Step 1: From the first $\mathbb{H}$ pixels in the marked depth map, extract the overheads according to the overhead extraction strategy described in Section 3.1.2.
Step 2: Decompress the compressed location map to obtain the location map M.
Step 3: Starting from the $(\mathbb{H}+\mathbb{1})$-st pixel, extract the correction bit stream and the hidden data by Eq. (14), and then recover the original quantized depth values by Eq. (15) according to the hidden data extraction strategy described in Section 3.1.1.
Step 4: Based on the correction bit stream, recover the depth values of the first $\mathbb{H}$ pixels by Eq. (18).

For clarity, the block diagram of the proposed DNDE-based method for depth maps is shown in Fig. 3.

3.2. Proposed DNHM-based reversible data hiding method

In contrast to the histogram of a gray image, the histogram of a depth map is usually sparse, as shown in Fig. 4. Based on this characteristic, the capacity can be substantially increased when using the histogram modification for the depth map. Furthermore, to maximize the embedding capacity without causing any synthesis error, we propose a reversible data hiding method which embeds the hidden data by modifying the histogram and in the meantime ensures that the quantized depth value of each marked pixel falls within the allowable range determined by the D-NOSE model. First, we describe the hidden data embedding and extraction strategies, and then give the whole procedure of the proposed method. The overhead embedding strategy used in the proposed DNHM-based method is the same as that in the proposed DNDE-based method.

3.2.1. Hidden data embedding and extraction strategies

When using the histogram modification to embed the hidden data, the peaks of the histogram of the quantized depth values are used for embedding the hidden data. A peak in the histogram is the depth value whose frequency is higher than those of the adjacent depth values. For the discreteness characteristics of the quantized depth values, the histogram of quantized depth values often has more and higher peaks than the histogram of gray values, implying that modifying the histogram of quantized depth values can achieve higher embedding capacity.

Denote by p_k the kth peak in the histogram of quantized depth values and $[\ell(p_k), u(p_k)]$ the corresponding allowable range determined by the D-NOSE model. Consider embedding the hidden data by shifting the histogram to the right such that the number of hidden bits for each pixel depends on the size $u(p_k) - p_k$; to increase the number of hidden bits for each pixel without causing any synthesis error, we can shift the depth value p_k to $\ell(p_k)$ to achieve the maximum embedding capacity. After shifting the depth value p_k to $\ell(p_k)$, the number of the hidden bits can be determined by

$$n(p_k) = \lfloor \log_2(u(p_k) - \ell(p_k) + 1) \rfloor. \tag{19}$$

Note that the above embedding strategy for each depth value can be applied to shift the depth value p_k to $u(p_k)$, which yields the same embedding capacity. Let $z'_q(p_k)$ denote the quantized depth value of the marked pixel whose original quantized depth value is p_k. Embedding the hidden data by shifting $\ell(p_k)$, we have

$$z'_q(p_k) = \ell(p_k) + h(p_k) \tag{20}$$

with hidden data $h(p_k) \in \{0, 1, \ldots, 2^{n(p_k)} - 1\}$.

When there exists more than one peak in the allowable range, divide the allowable range into non-overlapping subranges such that each subrange includes one peak. Then, shift the depth value of each peak to the lower bound of the corresponding subrange to achieve the maximum embedding capacity. For illustration, we describe how to divide the allowable range into

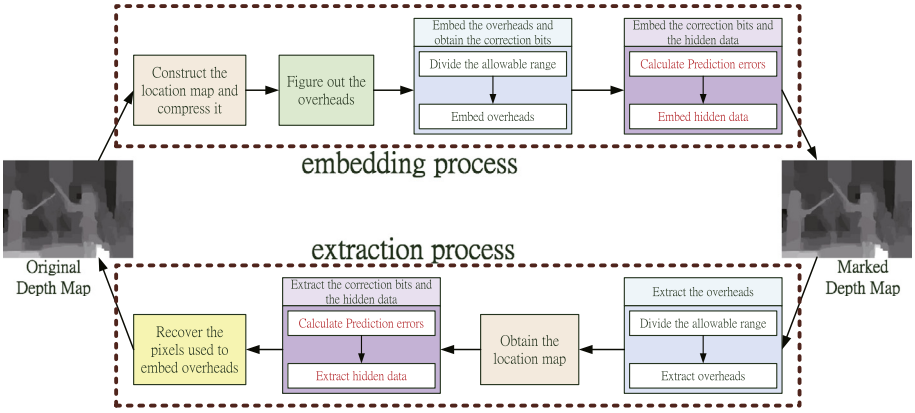

Fig. 3. The block diagram of the proposed DNDE-based method.

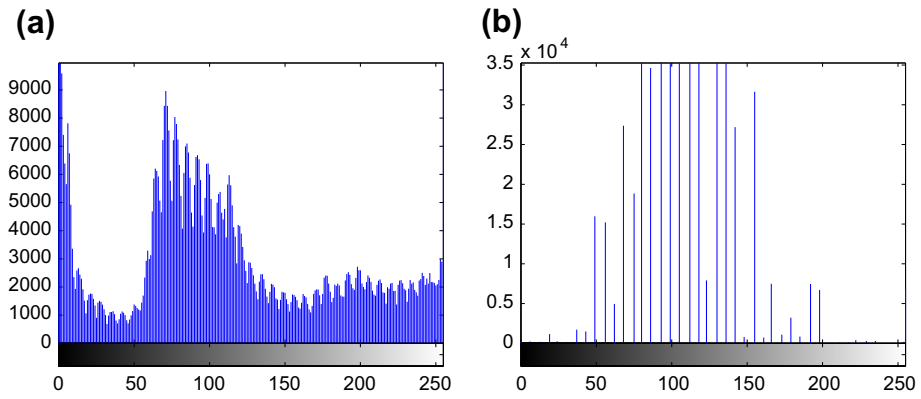

Fig. 4. Histograms of (a) gray image and (b) depth map for Kendo image.

two subranges when there exist two peaks in the allowable range. Considering two peaks p_{k_1} and p_{k_2} ($p_{k_1} < p_{k_2}$) within the allowable range $[\ell(p_k), u(p_k)]$, to maximize the embedding capacity, the threshold T to divide the allowable range is determined by

$$T = \arg \max_{p_{k_1} \leqslant T^* \leqslant p_{k_2}} \left\{ N(p_{k_1}) \lfloor \log_2(T^* - \ell(p_k) + 1) \rfloor + N(p_{k_2}) \lfloor \log_2(u(p_k) - T^*) \rfloor \right\} \tag{21}$$

where $N(p_{k_1})$ and $N(p_{k_2})$, respectively, denote the number of pixels with the depth values p_{k_1} and p_{k_2}. Consequently, the two divided subranges are $[\ell(p_k), T]$ and $[T+1, u(p_k)]$. The above divided approach can be extended to deal with those cases in which there exist more than two peaks in the allowable range.

To avoid the ambiguous problem on the extraction side, we need to record the positions of the pixels with peak depth values using a location map. Let $z_q(i,j)$ denote the quantized depth value of the pixel located at position (i,j). Then, the location map can be expressed as

$$M(i,j) = \begin{cases} 0, & \text{if } z_q(i,j) \text{ equals some peak depth value,} \\ 1, & \text{otherwise.} \end{cases} \tag{22}$$

For the inherent local homogeneity characteristic of the depth map, the proportions of 0 and 1 in the location map are usually very different, indicating that we can reduce the size of the location map by arithmetic coding [25]. The size of the compressed location map is usually small. The ratios of the size of the compressed location map to the available embedding capacity for nine 3D test images with the format consisting of a color image plus a depth map are given in Table 2. All ratios reported are less than 15%. Small ratios indicate that the overhead incurred by using the location map is insignificant. It is interesting to notice that the sizes of the compressed location maps for the 3D test images Mobile, Kendo, Balloons, Newspaper, and Hall are much smaller than those for the 3D test images Breakdancer, Ballet, Undodancer, and Street since the histograms of the first five images are much sparser than those of the other four images. In other words, almost all the pixels in the first five images can be used to embed the hidden data, leading to good compression performance of the location map. As for the other four images, some histogram bins of depth values are continuous, resulting in more 1s in the location maps and the degradation of the compression performance of the location maps.

Given the location map, peak depth values, and the marked depth map, we can extract the hidden data and recover the original depth map completely as follows. For extraction and recovery, since only the peaks are used for embedding the hidden data, we only tackle the marked depth value $z'_q(i,j)$ with $M(i,j) = 0$ and identify the allowable range $[\ell(z_q(i,j)), u(z_q(i,j))]$ that covers $z'_q(i,j)$, and hence the corresponding peak depth value is p_k. Then, we can extract the hidden data $h(i,j)$ by

Table 2
Ratio of the size of compressed location map over the available embedding capacity for nine 3D test images.

3D image	Size of compressed location map (bit)	Available embedding capacity (bit)	Ratio (%)
Mobile	35,457	538,040	6.584
Kendo	220	1,669,328	0.013
Balloons	530	1,648,691	0.032
Newspaper	13	1,372,890	0.001
Breakdancer	63,706	492,195	12.943
Ballet	55,778	380,272	14.667
Hall	30	1,137,951	0.002
Undodancer	111,370	810,982	13.372
Street	106,938	855,319	12.502

$$h(i,j) = z'_q(i,j) - \ell(z_q(i,j)) \tag{23}$$

and recover the original quantized depth value simply by replacing the marked depth value by p_k. For the case that there are two peaks in the allowable range, we can extract the hidden data from the left peak by Eq. (23) and the hidden data from the right peak by $h(i,j) = z'_q(i,j) - (T+1)$.

Similar to the proposed DNDE-based method, some extra overheads should be addressed when embedding the hidden data by the proposed DNHM-based method. In addition to the hidden data, extra overheads including the compressed location map of size μ, all the peaks of size σ, the hidden-data size ρ, and the overhead size $\tau = \mu + \sigma + \rho$ should be embedded in the depth map. We use the same overhead embedding and extraction strategies based on the D-NOSE model described in Section 3.1.2 to tackle the overheads.

3.2.2. The whole procedure of the proposed DNHM-based method

We now describe the whole procedure of the proposed DNHM-based reversible data hiding method. Note that the number $\mathbb{H}$ ($\geqslant \tau$) of pixels used to embed overheads is known on both the embedding and extraction sides.

Given the depth map and the hidden data to be embedded, four steps in the embedding process are followed to generate the marked depth map.

Step 1: Construct the location map M according to Eq. (22), and compress M by arithmetic coding.
Step 2: Figure out the overheads including the compressed location map, all the peak values p_k's, the hidden-data size μ, and the overhead size τ.
Step 3: Embed the overheads in the first $\mathbb{H}$ pixels according to the overhead embedding strategy described in Section 3.1.2, and obtain the correction bits by Eq. (17).
Step 4: Starting from the $(\mathbb{H}+\mathbb{1})$-st pixel, embed the correction bits and the hidden data according to the hidden data embedding strategy described in Section 3.2.1.

Given the marked depth map, four steps in the extraction process are followed to extract the hidden data and recover the original depth map.

Step 1: From the first $\mathbb{H}$ pixels in the marked depth map, extract the overheads according to the overhead extraction strategy described in Section 3.1.2.
Step 2: Decompress the compressed location map to obtain the location map M.
Step 3: Starting from the $(\mathbb{H}+\mathbb{1})$-st pixel, extract the hidden data and correction bits by Eq. (23), and then recover the original quantized depth value by replacing the marked depth value by the corresponding peak.
Step 4: Based on the correction bits, recover the depth values of the first $\mathbb{H}$ pixels by Eq. (18).

For clarity, the block diagram of the proposed DNHM-based method for depth maps is shown in Fig. 5.

4. Experimental results

We compared the proposed DNDE-based and DNHM-based reversible data hiding methods with three state-of-the-art conventional reversible data hiding methods [24,28,30]. The experiments were conducted on nine 3D test images with a format consisting of a color image plus a depth map shown in Fig. 6. The color image and depth map of Fig. 6(a) have a size of 720×540, those of Figs. 6(b)–(f) have a size of 1024×768, and those of Figs. 6(g)–(i) have a size of 1088×1092. All the

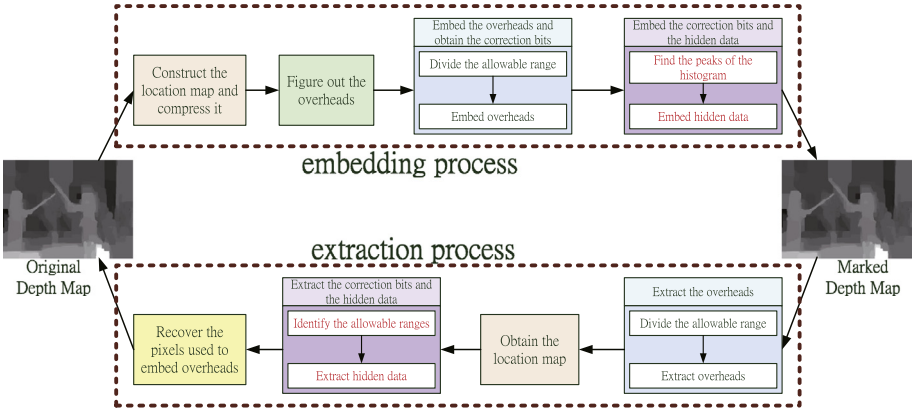

Fig. 5. Block diagram of the proposed DNHM-based method.

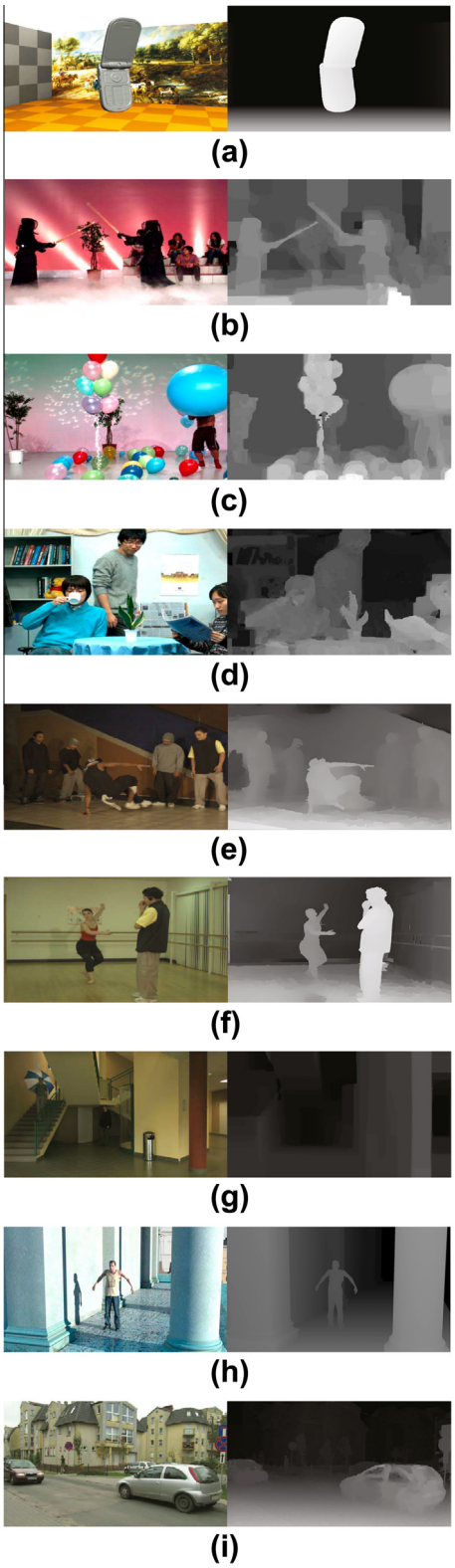

Fig. 6. Color images plus depth maps for (a) Mobile, (b) Kendo, (c) Balloons, (d) Newspaper, (e) Breakdancer, (f) Ballet, (g) Hall, (h) Undodancer, and (i) Street.

concerned reversible data hiding methods were first applied to the depth map of the original view to generate the marked depth map and then the color image, and the generated marked depth map were utilized to yield the rendered virtual view by the DIBR warping process. All the experiments were implemented on an IBM compatible computer with Intel Core i7-960 CPU 3.2 GHz, 24 GB RAM, and Microsoft Windows 7 64-bit operating system. The program development environment was Borland C++ Builder 6.0, and the hidden data were generated by the function *rand*() in the C++ language.

Table 3
CPSNR for different values of Capacity under 3D Mobile image.

Method	Capacity (bpp)						
	0.1	0.3	0.5	0.7	0.9	1.3	1.4
Ni et al.'s [24]	30.948	30.948	30.129	–	–	–	–
Thodi and Rodriguez's [30]	53.225	33.693	31.029	28.947	26.942	–	–
Sachnev et al.'s [28]	41.520	33.898	28.164	25.949	24.871	–	–
Proposed DNDE-based	∞	∞	∞	∞	∞	∞	–
Proposed DNHM-based	∞	∞	∞	∞	∞	∞	∞

Table 4
CPSNR for different values of Capacity under 3D Kendo image.

Method	Capacity (bpp)						
	0.1	0.3	0.5	0.7	0.9	1.3	2.1
Ni et al.'s [24]	32.872	19.867	19.852	19.540	19.298	–	–
Thodi and Rodriguez's [30]	57.735	39.024	33.903	23.159	21.830	–	–
Sachnev et al.'s [28]	∞	37.376	35.252	21.833	20.455	–	–
Proposed DNDE-based	∞	∞	∞	∞	∞	∞	–
Proposed DNHM-based	∞	∞	∞	∞	∞	∞	∞

Table 5
CPSNR for different values of Capacity under 3D Balloons image.

Method	Capacity (bpp)						
	0.1	0.3	0.5	0.7	0.9	1.4	2.1
Ni et al.'s [24]	21.867	14.573	14.147	14.136	13.669	–	–
Thodi and Rodriguez's [30]	41.191	31.577	29.700	25.745	23.925	–	–
Sachnev et al.'s [28]	31.576	26.121	25.066	16.898	14.792	–	–
Proposed DNDE-based	∞	∞	∞	∞	∞	∞	–
Proposed DNHM-based	∞	∞	∞	∞	∞	∞	∞

Table 6
CPSNR for different values of Capacity under 3D Newspaper image.

Method	Capacity (bpp)					
	0.1	0.3	0.5	0.7	0.9	1.7
Ni et al.'s [24]	∞	31.385	23.411	20.605	20.341	–
Thodi and Rodriguez's [30]	31.816	28.710	26.888	24.145	21.227	–
Sachnev et al.'s [28]	33.094	25.137	21.169	19.484	17.761	–
Proposed DNDE-based	∞	∞	∞	∞	∞	–
Proposed DNHM-based	∞	∞	∞	∞	∞	∞

Table 7
CPSNR for different values of Capacity under 3D Breakdancer image.

Method	Capacity (bpp)								
	0.05	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8
Ni et al.'s [24]	27.212	–	–	–	–	–	–	–	–
Thodi and Rodriguez's [30]	38.476	35.332	29.736	28.095	26.297	26.457	25.787	25.350	24.319
Sachnev et al.'s [28]	39.289	34.807	30.004	27.939	25.451	24.732	23.748	22.977	22.309
Proposed DNDE-based	∞	∞	∞	∞	∞	∞	∞	∞	–
Proposed DNHM-based	∞	∞	∞	∞	∞	∞	–	–	–

The proposed reversible data hiding methods are first compared with three previous conventional reversible data hiding methods in terms of two objective measures, the color peak signal-to-noise ratio (CPSNR) and the Capacity. The CPSNR measures the quality of the rendered virtual view generated from the marked depth map whereas the Capacity, measured in bits per pixel (bpp), represents the number of embedded hidden bits. The CPSNR of a generated rendered virtual view of size $X \times Y$ can be expressed as

$$\mathrm{CPSNR} = 10\log_{10}\frac{255^2}{\frac{1}{3XY}\sum_{i=0}^{X-1}\sum_{j=0}^{Y-1}\sum_{K\in\{R,G,B\}}[m_K(i,j)-m'_K(i,j)]^2} \tag{24}$$

where $m_K(i,j)$ and $m'_K(i,j)$ denote, respectively, the color values of channel K of the pixel at position (i,j) in the rendered virtual views generated from the original and the marked depth maps. Higher values of the CPSNR indicate better quality of the generated rendered virtual view. The Capacity of a marked depth map of size $X \times Y$ is defined as

$$\mathrm{Capacity} = \frac{\#\{\text{embedded hidden bits}\}}{XY}, \tag{25}$$

where $\#\{\text{embedded hidden bits}\}$ denotes the number of the embedded hidden bits. Large values of the Capacity indicate higher embedding capacity.

Table 8
CPSNR for different values of Capacity under 3D Ballet image.

Method	Capacity (bpp)							
	0.05	0.1	0.2	0.3	0.4	0.5	0.6	0.7
Ni et al.'s [24]	23.302	–	–	–	–	–	–	–
Thodi and Rodriguez's [30]	27.782	24.838	22.017	20.917	20.127	19.108	18.695	18.338
Sachnev et al.'s [28]	25.980	23.132	20.552	19.546	18.629	17.976	16.881	16.361
Proposed DNDE-based	∞	∞	∞	∞	∞	∞	∞	∞
Proposed DNHM-based	∞	∞	∞	∞	∞	–	–	–

Table 9
CPSNR for different values of Capacity under 3D Hall image.

Method	Capacity (bpp)								
	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Ni et al.'s [24]	∞	26.901	24.596	24.596	24.596	21.573	21.471	21.397	–
Thodi and Rodriguez's [30]	27.254	25.653	24.962	24.582	23.879	21.494	20.862	20.286	19.895
Sachnev et al.'s [28]	25.237	23.859	23.110	21.630	20.641	19.686	18.890	18.327	17.982
Proposed DNDE-based	∞	∞	∞	∞	∞	∞	∞	–	–
Proposed DNHM-based	∞	∞	∞	∞	∞	∞	∞	∞	∞

Table 10
CPSNR for different values of Capacity under 3D Undodancer image.

Method	Capacity (bpp)								
	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.9	1.3
Ni et al.'s [24]	24.866	23.364	–	–	–	–	–	–	–
Thodi and Rodriguez's [30]	30.583	27.543	25.462	24.138	23.266	22.332	21.673	20.624	–
Sachnev et al.'s [28]	27.253	24.389	22.470	20.597	19.974	19.263	18.709	17.621	–
Proposed DNDE-based	∞	∞	∞	∞	∞	∞	∞	∞	∞
Proposed DNHM-based	∞	∞	∞	∞	∞	–	–	–	–

Table 11
CPSNR for different values of Capacity under 3D Street image.

Method	Capacity (bpp)							
	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8
Ni et al.'s [24]	22.496	–	–	–	–	–	–	–
Thodi and Rodriguez's [30]	25.079	22.205	20.472	19.420	18.616	17.595	17.112	16.697
Sachnev et al.'s [28]	24.683	21.618	20.036	18.721	17.921	16.998	16.451	15.900
Proposed DNDE-based	∞	∞	∞	∞	∞	∞	∞	–
Proposed DNHM-based	∞	∞	∞	∞	∞	∞	–	–

Fig. 7. Rendered virtual views for 3D Newspaper image generated from (a) original depth map and from marked depth maps (with Capacity = 0.9 bpp) by (b) Ni et al.'s method, (c) Thodi and Rodriguez's method, (d) Sachnev et al.'s method, (e) the proposed DNDE-based method, and (f) the proposed DNHM-based method.

The comparisons of CPSNR with different values of Capacity for nine 3D test images are given in Tables 3 through 11 where the symbol "–" denotes that the corresponding method is infeasible in the test image with the specified value of Capacity. It is clear that the proposed methods which achieve infinitive CPSNR outperform the three conventional methods since no synthesis errors are incurred in the rendered virtual views generated from the color maps and the corresponding marked depth maps using the proposed methods. Furthermore, since the proposed methods maximize the embedding capacity over the allowable range derived from the D-NOSE model, the proposed methods usually achieve higher, sometimes more than 1 bpp, embedding capacity than the three conventional methods. The proposed DNHM-based method can achieve much higher, more than 2 bpp, embedding capacity in the 3D test images Kendo and Balloons whose depth maps have sparse histograms. For the 3D test images Breakdancer, Ballet, Undodancer, and Street, where the histograms have continuous bins, both Ni et al.'s method and the proposed DNHM-based method have poor performance, as expected.

When comparing the proposed DNDE-based with the DNHM-based methods, Tables 3–11 show that the maximum embedding capacity of the DNHM-based method is more sensitive to the test image than the DNDE-based method. For the depth map with a sparse histogram, the DNHM-based method usually achieves much higher embedding capacity. Thus,

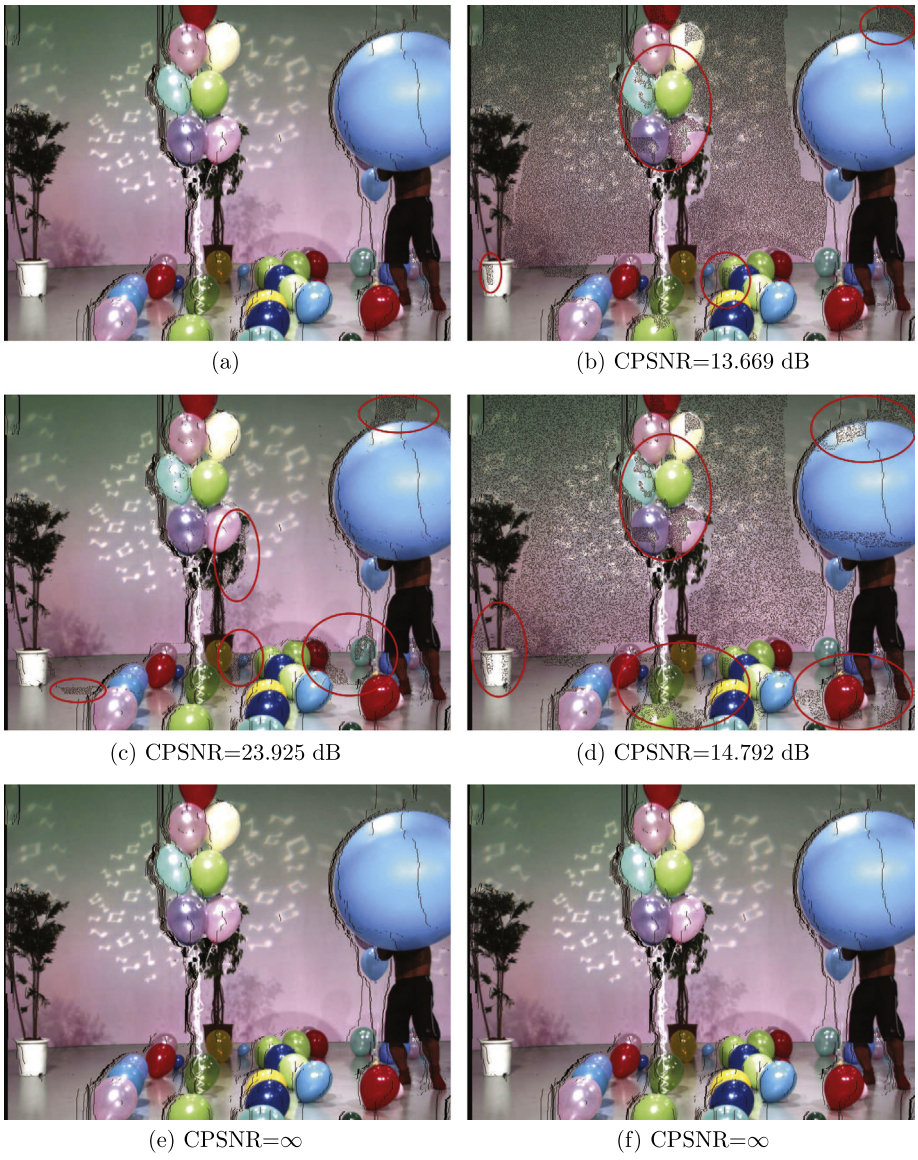

Fig. 8. Rendered virtual views for 3D Balloons image generated from (a) original depth map and from marked depth maps (with Capacity = 0.9 bpp) by (b) Ni et al.'s method, (c) Thodi and Rodriguez's method, (d) Sachnev et al.'s method, (e) the proposed DNDE-based method, and (f) the proposed DNHM-based method.

if the depth map has a sparse histogram, the proposed DNHM-based method is appropriate for embedding hidden data; otherwise, the proposed DNDE-based method should be used.

In addition to the objective metrics, the subjective visual evaluation was used to demonstrate the effectiveness of the proposed reversible data hiding methods. After embedding hidden data in the depth map, for the three conventional methods, the change of depth values may cause synthesis errors in the rendered virtual view, leading to degradation in the visual perception. For subjective visual comparisons, the rendered virtual views of the 3D Newspaper and the Balloon images are generated from the original and marked depth maps by the three conventional and the proposed reversible data hiding methods, respectively. As shown in Figs. 7 and 8, the rendered virtual views generated from the marked depth maps by the proposed DNDE-based and DNHM-based methods are exactly the same as those generated from the original depth maps since no synthesis errors appear in the rendered virtual views, whereas some artifacts are observed and circled by the red lines in the rendered virtual views generated from the marked depth maps by the three conventional methods.

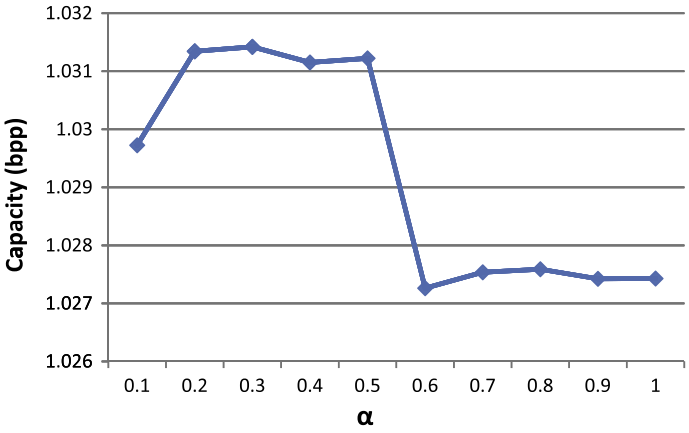

Fig. A.1. The maximum embedding capacity against different values of α for the proposed DNDE-based method.

5. Conclusions

We have developed two novel reversible data hiding methods, based on the D-NOSE model, for depth maps. The hiding flexibility provided by the D-NOSE model is exploited to avoid synthesis errors and thus the rendered virtual view generated from the marked depth map is identical to that generated from the original depth map. The proposed methods can achieve substantially higher embedding capacity due to the inherent local homogeneity characteristic of the depth maps. Experimental results on nine typical 3D images with a format consisting of a color image plus a depth map demonstrate that the proposed reversible data hiding methods achieve better performance in terms of both objective and subjective measures than the three state-of-the-art conventional methods [24,28,30].

Acknowledgments

The authors appreciate the programming help of Ms. Li-Min Huang and the proofreading help of Ms. Catherine Harrington.

Appendix A. Determination of the parameter α

The maximum embedding capacity of the proposed DNDE-based method depends on the selected parameter α. The average embedding capacity over nine test images against different values of α of the proposed DNDE-based method is shown in Fig. A.1. It is clear that the average embedding capacity of the proposed DNDE-based method is not sensitive to the selected α, indicating that $\alpha = \frac{1}{2}$ can be used to achieve high embedding capacity and avoid floating point computation.

References

[1] A.M. Alattar, Reversible watermark using the difference expansion of a generalized integer transform, IEEE Trans. Image Process. 13 (2004) 1147–1156.
[2] L. An, X. Gao, X. Li, D. Tao, C. Deng, J. Li, Robust reversible watermarking via clustering and enhanced pixel-wise masking, IEEE Trans. Image Process. 21 (2012) 3598–3611.
[3] L. An, X. Gao, Y. Yuan, D. Tao, Robust lossless data hiding using clustering and statistical quantity histogram, Neurocomputing 77 (2012) 1–11.
[4] L. An, X. Gao, Y. Yuan, D. Tao, C. Deng, F. Ji, Content-adaptive reliable robust lossless data embedding, Neurocomputing 79 (2012) 1–11.
[5] J.M. Barton, Method and Apparatus for Embedding Authentication Information Within Digital Data, US Patent# 5 646 997, 1997.
[6] C.C. Chang, P.Y. Tsai, M.H. Lin, SVD-based digital image watermarking scheme, Pattern Recogn. Lett. 26 (2005) 1577–1586.
[7] C.C. Chang, C.Y. Lin, Reversible steganographic method using SMVQ approach based on declustering, Inf. Sci. 177 (2007) 1796–1805.
[8] C.C. Chang, C.C. Lin, C.S. Tseng, W.L. Tai, Reversible hiding in DCT-based compressed images, Inf. Sci. 177 (2007) 2768–2786.
[9] C.C. Chang, C.Y. Lin, Y.H. Fan, Lossless data hiding for color images based on block truncation coding, Pattern Recogn. 41 (2008) 2347–2357.
[10] C.C. Chang, T.D. Kieu, W.C. Wu, A lossless data embedding technique by joint neighboring coding, Pattern Recogn. 42 (2009) 1597–1603.
[11] K.L. Chung, C.H. Shen, L.C. Chang, A novel SVD- and VQ-based image hiding scheme, Pattern Recogn. Lett. 22 (2001) 1051–1058.
[12] K.L. Chung, W.N. Yang, Y.H. Huang, S.T. Wu, Y.C. Hsu, On SVD-based watermarking algorithm, Appl. Math. Comput. 188 (2007) 54–57.
[13] C. Fehn, Depth-image-based rendering (DIBR), compression, and transmission for a new approach on 3D-TV, in: Proc. of SPIE Stereoscopic Displays and Virtual Reality Systems XI, San Jose, CA, USA, 2004, pp. 93–104.
[14] X. Gao, L. An, Y. Yuan, D. Tao, X. Li, Lossless data embedding using generalized statistical quantity histogram, IEEE Trans. Circ. Syst. Video Technol. 21 (2011) 1061–1070.
[15] L.H.J. Kamstra, A.M. Heijmans, Reversible data embedding into images using wavelet techniques and sorting, IEEE Trans. Image Process. 14 (2005) 2082–2090.
[16] H.J. Kim, V. Sachnev, Y.Q. Shi, J. Nam, H.G. Choo, A novel difference expansion transform for reversible data embedding, IEEE Trans. Inf. Forensics Secur. 3 (2008) 456–465.
[17] I.S. Lee, W.H. Tsai, Data hiding in grayscale images by dynamic programming based on a human visual model, Pattern Recogn. 42 (2009) 1604–1611.

[18] X. Li, B. Yang, T. Zeng, Efficient reversible watermarking based on adaptive prediction-error expansion and pixel selection, IEEE Trans. Image Process. 20 (2011) 3524–3533.
[19] P.L. Lin, C.K. Hsieh, P.W. Huang, A hierarchical digital watermarking method for image tamper detection and recovery, Pattern Recogn. 38 (2005) 2519–2529.
[20] C.C. Lin, W.L. Tai, C.C. Chang, Multilevel reversible data hiding based on histogram modification of difference images, Pattern Recogn. 41 (2008) 3582–3591.
[21] R. Liu, T. Tan, An SVD-based watermarking scheme for protecting rightful ownership, IEEE Trans. Multimedia 4 (2002) 121–128.
[22] L. Luo, Z. Chen, M. Chen, X. Zeng, Z. Xiong, Reversible image watermarking using interpolation technique, IEEE Trans. Inf. Forensics Secur. 5 (2010) 187–193.
[23] S. Mallat, A Wavelet Tour of Signal Processing, 2nd ed., Academic Press, Orlando, FL, 1999.
[24] Y.Q. Ni, N.A. Shi, W. Su, Reversible data hiding, IEEE Trans. Circ. Syst. Video Technol. 16 (2006) 354–362.
[25] E. Parzen, Modern Probability Theory and Its Applications, John Wiley and Sons, New York, 1960.
[26] S.C. Pei, I.K. Tam, Effective color interpolation in CCD color filter arrays using signal correlation, IEEE Trans. Circ. Syst. Video Technol. 13 (2003) 503–513.
[27] V. Sachnev, H.J. Kim, S. Xiang, J. Nam, An improved reversible difference expansion watermarking algorithm, in: The 6th International Workshop on Digital Watermarking (IWDW 2007), Guangzhou, China, 2007, pp. 254–263.
[28] V. Sachnev, H.J. Kim, J. Nam, S. Suresh, Y.Q. Shi, Reversible watermarking algorithm using sorting and prediction, IEEE Trans. Circ. Syst. Video Technol. 19 (2009) 989–999.
[29] W.L. Tai, C.M. Yeh, C.C. Chang, Reversible data hiding based on histogram modification of pixel differences, IEEE Trans. Circ. Syst. Video Technol. 19 (2009) 906–910.
[30] D.M. Thodi, J.J. Rodriguez, Expansion embedding techniques for reversible watermarking, IEEE Trans. Image Process. 16 (2007) 721–730.
[31] J. Tian, Reversible data embedding using a difference expansion, IEEE Trans. Circ. Syst. Video Technol. 13 (2003) 890–896.
[32] D. Tian, P.L. Lai, P. Lopez, C. Gomila, View synthesis techniques for 3D video, in: Proc. of the SPIE, vol. 7443, 2009, pp. 74430T-1–74430T-11.
[33] C.L. Tsai, H.F. Chiang, K.C. Fan, C.D. Chung, Reversible data hiding and lossless reconstruction of binary images using pairwise logical computation mechanism, Pattern Recogn. 38 (2005) 1993–2006.
[34] C.H. Tzeng, Z.F. Yang, W.H. Tsai, Adaptive data hiding in palette images by color ordering and mapping with security protection, IEEE Trans. Commun. 52 (2004) 791–800.
[35] R.Z. Wang, C.F. Lin, J.C. Lin, Image hiding by optimal LSB substitution and genetic algorithm, Pattern Recogn. 34 (2001) 671–683.
[36] S. Xiang, H.J. Kim, J. Huang, Invariant image watermarking based on statistical features in the low-frequency domain, IEEE Trans. Circ. Syst. Video Technol. 18 (2008) 777–790.
[37] W.J. Yang, K.L. Chung, H.Y.M. Liao, Efficient reversible data hiding for color filter array images, Inf. Sci. 190 (2012) 208–226.
[38] L. Zhang, W.J. Tam, Stereoscopic image generation based on depth images for 3D TV, IEEE Trans. Broadcasting 51 (2005) 191–199.
[39] Y. Zhao, C. Zhu, Z. Chen, L. Yu, Depth no-synthesis-error model for view synthesis in 3-D video, IEEE Trans. Image Process. 20 (2011) 2221–2228.