International Journal of Hybrid Information Technology
Vol. 9, No. 5 (2016), pp. 145-164
http://dx.doi.org/10.14257/ijhit.2016.9.5.12

Hole-filling Based on Disparity Map and Inpainting for Depth-Image-Based Rendering

Ran Liu[1,2,3], Zekun Deng[1], Lin Yi[2], Zhenwei Huang[1], Donghua Cao[3], Miao Xu[3] and Ruishuang Jia[3]

[1]*College of Communication Engineering, Chongqing University, Chongqing 400044, China*
[2]*Chongqing Key Laboratory of Translational Research for Cancer Metastasis and Individualized Treatment, Chongqing Cancer Institute, Chongqing 400030, China*
[3]*Key Laboratory of Dependable Service Computing in Cyber Physical Society of the Ministry of Education, Chongqing University, Chongqing 400044, China*
linyi.cqch@qq.com

Abstract

Due to the changes in viewpoint, holes may appear in the novel view synthesized by depth-image-based rendering (DIBR). A hole-filling method combining disparity-map-based hole-filling and inpainting is proposed. The method first eliminates matching errors in destination image according to the disparity map. Then, holes are classified into different types according to their sizes. Finally, a disparity-map-based approach and an improved exemplar-based inpainting algorithm are used to fill different types of holes according to the type of hole. Experimental results show that the artifacts at the edge of foreground objects can be reduced in synthesized view since the matching errors are eliminated before hole-filling. In addition, the proposed method can achieve more natural and satisfactory results in filled areas in comparison with the disparity-map-based approach and Gautier's inpainting algorithm, though it may result in higher time complexity.

Keyword*: Depth-image-based Rendering, Hole-filling, Holes, Disparity Map, Inpainting*

1. Introduction

In *free viewpoint television* (FTV) or *auto-stereoscopic display* system, *holes* may appear in novel views (called *destination images* or *virtual views*) synthesized by *depth-image-based rendering* (DIBR) engine, which may result in heavy image quality degradation [1-5]. The main cause of holes is the changes in viewpoint [6, 7], and larger baseline usually involves larger holes [8]. How to fill these holes, especially the big holes in synthesized views, is a key problem to DIBR engine.

Many methods for hole-filling are proposed [9, 10]. Generally, these methods can be classified into two types according to their processing features [9]:

(1) Preprocessing of depth map

In this method, *depth map* is smoothed by smoothing filter before the destination image generated. As the depth information is smoothed, *sharp depth transitions* (*discontinuities*) in depth map are reduced [9, 11]. As a result, the size of holes may be decreased in novel views.

A common problem with this method is that it may introduce *geometric distortions* to the destination image [9, 11-13]. People have tried many different smoothing methods to

ISSN: 1738-9968 IJHIT

deal with geometric distortion, for example: *asymmetric smoothing* [13], *edge dependent depth filtering* [14], *directional Gaussian filtering* (DGF) [15], *adaptive smoothing* [16], and *optimized adaptive filtering* [17]. Although these smoothing methods do reduce the geometric distortions, it is quite difficult for them to eliminate the geometric distortions completely.

(2) Post-processing of destination image

In this method, holes are filled with pixels (*non-hole point*) sampled from the *reference image* or destination image [12]. *Image inpainting* is the typical algorithm used by post-processing method. Many inpainting algorithms for hole-filling are proposed in recent years. Started from Criminisi's work, Daribo *et al*. proposed an inpainting algorithm in which depth information is applied to reduce the *artifacts* [18, 19]. However, artifacts can still be found at the edge of foreground objects. In order to further reduce these artifacts, Gautier *et al*. improved the method of *priority computation* in Daribo' algorithm by using 3D *tensor* and *directional background propagation* [8]. However, this method cannot provide an ideal result for the holes at the border of the destination image. Different from Gautier's method, Wu *et al*. [20] and Wang *et al*. [21] put forward an idea of hole-filling which processes foreground and background respectively. Wu *et al*. adopted the *watershed algorithm* to distinguish foreground from background. However, it may lead to *over-segmentation* and require much time to merge generated images. Wang *et al*. used a *background estimation* method to estimate the static background, which contributes to higher filling speed in comparison with classical inpainting method. However, this method needs multi frames to estimate background, and this limits its applications.

Different from these post-processing methods based on image inpainting, our previous work proposed a hole-filling approach based on disparity map [9]. One important aspect of the approach is that holes are filled by copying corresponding pixels from the reference image instead of destination image according to the disparity map converted from a depth map. No depth-smoothing or inpainting methods are needed. Hence, geometric distortions almost disappear in synthesized view, and it is quite simpler than inpainting method. However, when the baseline is too large or the virtual view is located beyond the field of view of real cameras [22], the image quality of the virtual view degrades.

To overcome this issue, we have made a careful analysis of the characteristics of the holes generated by DIBR. There are two major characteristics of the holes in the novel view:

(1) Most of the holes are small even when the baseline is quite large. A simple hole-filling method is enough for image quality here.

(2) The size of holes between foreground and background is large, and most of these holes should be filled with background pixels [4].

Using the characteristics mentioned above, this paper extends our previous work in a number of ways. First, matching errors in the destination image are eliminated according to the disparity map. Then, holes are classified into different types according to their sizes. Finally, a *disparity-map-based* approach [9] and an improved *exemplar-based inpainting* algorithm are used to fill different types of holes. Note that the proposed method is based on shift-sensor camera setup [9], for which only the case that the *virtual viewpoint* shifts horizontally is considered when synthesizing novel views at the receiver side [23]. In addition, the input 3D video for our method consists of only regular 2D color video (i.e. reference images) and an accompanying depth map sequence with the same spatial-temporal resolution excluding *camera parameters* [5], which is used widely in current 3D display system. What's more, only current frame is used for hole-filling in our method.

The remaining portions of this paper are organized as follows. In Sec. 2, the proposed method is discussed in detail. Sec. 1 is devoted to evaluating the performance of the proposed method. Conclusions are shown in Sec. 2.

2. Proposed Method

This paper puts forward a novel method that can classify the holes generated by DIBR into different types automatically and fill different types of holes with different algorithms: As for small or medium holes, they can be filled by the disparity-map-based approach with good image quality rather than the time-consuming inpainting algorithm; for the rest big holes, if both sides of a hole are foreground pixels, then it is also filled by the disparity-map-based approach, otherwise it is filled by an improved exemplar-based inpainting algorithm. Furthermore, we add a module for matching error elimination before hole classification, since there are often matching errors in destination image due to the inaccuracy of the depth map [15, 24]. These matching errors have a negative impact on the image quality of the holes repaired by inpainting, and thus they must be removed.

An overview of the proposed method is illustrated in Figure 1. The inputs are the reference image and an accompanying depth map, and the output is the destination image without holes. As shown in Figure 1, there are four main modules in this method: matching error elimination, hole classification, disparity-map-based hole-filling, and inpainting. The details of each module are addressed in the following sections.

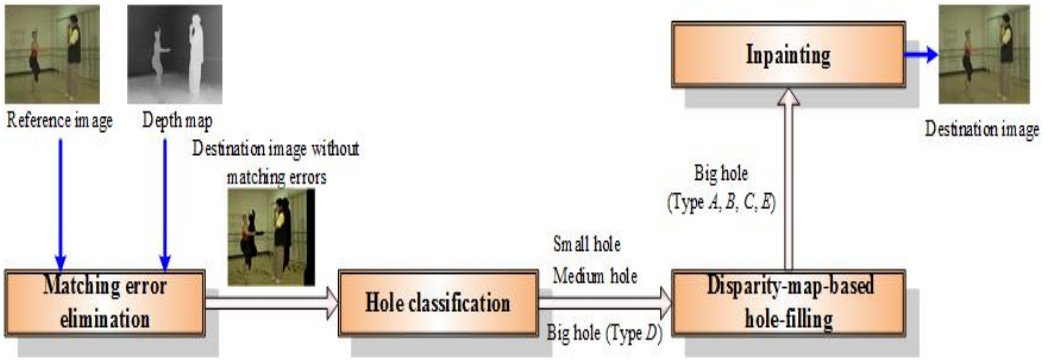


Figure 1. Flowchart of the Proposed Hole-filling Method for DIBR

2.1. Matching Error Elimination

Object contours in depth map are often inconsistent with that in the reference image. Usually, foreground object contours in the reference image are larger than that in the associated depth map. It is inevitable that the shifts of some foreground pixels are the same as that of their neighboring background pixels after 3D image warping. As a result, the boundary of the big hole adjoining the background mixes with some foreground pixels, as shown in Figure 2. Hence the so-called matching errors appear [9].

In this module, we use the algorithm described in Ref. [24] to remove the matching errors. In addition, a median filter is used to eliminate the singularities after 3D image warping [25].

(a) (b)

Figure 2. Illustration of Matching Errors: (a) Synthesized Destination Image after 3D Image Warping; (b) Enlargement of Image

2.2. Hole Classification

This module classifies the holes into different types and thereafter different hole-filling processes can be performed depending on these types.

As is shown in Figure 3, holes are classified into three main types according to their sizes: *big hole*, *medium hole*, and *small hole*. In addition, *big hole* can be further classified into five subtypes: Type *A*, *B*, *C*, *D* and *E*. The algorithms used to fill different types/subtypes of holes are also presented in the figure. The details of hole classification are addressed below.

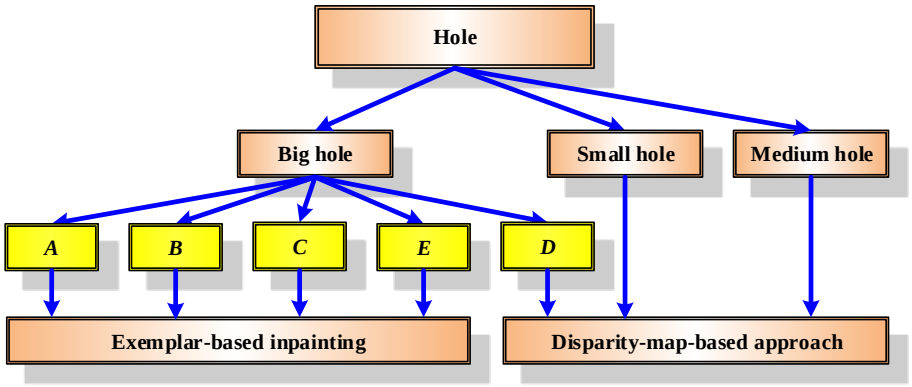


Figure 3. Types/Subtypes of Holes

There are two key steps in hole classification.

In the first step, holes are classified into the above mentioned three types according to the length of the hole *num*. This procedure is similar to the procedure of *hole detection* proposed in Ref. [9]. In fact, the "medium hole" here is the "big hole" in Ref. [9].

Suppose *len_hole*1 and *len_hole*2 are two thresholds, holes are classified by the rules below in the first step:

(1) If $num < len\_hole1$, then it is labeled "small hole";

(2) If $len\_hole1 \leq num < len\_hole2$, then it is labeled "medium hole";

(3) If $num \geq len\_hole2$, then it is labeled "big hole".

Note that the threshold values of *len_hole*1 and *len_hole*2 have important influences on the hole-filling results. If an improper value of *len_hole*1 is selected, some of small

holes may be incorrectly filled with foreground pixels. What's more, an oversize *len_hole*2 may result in a negative effect because the disparity-map-based approach is not suitable for big holes, and an undersize one may lead to some of medium holes not being filled. *len_hole*1 = 3 and *len_hole*2 = 20 are empirically recommended for various videos to obtain average image quality according to our tests.

As for the "big hole", most of them can be effectively filled by inpainting algorithm, but there is an exception: When both sides of a big hole are foreground, seldom or no background pixels can be used for hole-filling. As a result, the disparity-map-based approach outperforms the inpainting algorithm according to our tests. So it is necessary to further classify "big hole" into subtypes for different hole-filling processes.

In the second step, "big holes" are further classified into five subtypes:

(1) If both sides of a hole are background, then it is labeled "Type *A*";

(2) If the of a hole is background and the right side is foreground, then it is labeled "Type *B*";

(3) If the left side of a hole is foreground and the right side is background, then it is labeled "Type *C*";

(4) If both sides of a hole are foreground, then it is labeled "Type *D*";

(5) Otherwise, the hole must be a marginal hole, we labeled it "Type *E*".

An example of subtypes of holes can be found in Figure 4.

Figure 4. Illustration of Subtypes of Big Holes. Each Subtype of Big Hole is Labeled with Subtype Name and Marked by Yellow Circles

As can be seen from Figure 4, how to distinguish foreground from background is a key problem in the second step. We use the method proposed in Ref. [9] for *foreground-background discrimination*. This method distinguishes foreground pixels from background ones according to the *sharp parallex transition*. As an example, the discrimination of hole Type *D* is presented below:

(1) Foreground pixel discrimination

The purpose of this step is to distinguish foreground pixels from background ones at the edges of a hole based on disparity map.

Take the synthesized left view as an example. Suppose:

d_m: *Parallax difference* between the first pixel on the right border and the first pixel on the left border of a hole in disparity map **M**;

d_{m1}: Parallax difference between the other pixel on the right border and the first pixel on the right border of the hole;

d_{m2}: Parallax difference between the other pixel on the right border and the first pixel on the left border of the hole;

sharp_th: Threshold for the detection of *sharp parallax transition*.

Then, three situations may occur for a specific hole:

(a) If $d_m \geq sharp\_th$, then the first pixel on the right border of the hole is a foreground pixel. Record this pixel.

(b) If $d_m \leq -sharp\_th$, then the foreground pixels are on the left border of the hole. In this case, there must also be foreground pixels near the right border of the hole. Scan the pixels on the right side of the hole from left to right. If $d_{m1} \geq sharp\_th$, then it means we find a sharp parallex transition, the pixel with larger parallax-value is a foreground pixel. Record this pixel.

(c) If $-sharp\_th < d_m < sharp\_th$, then we need to scan the pixels on the right side of the hole from left to right to find sharp parallax transition. If $d_{m2} \geq sharp\_th$, then the pixel with larger parallax-value is a foreground pixel, record this pixel. If $d_{m2} \leq -sharp\_th$, then the first pixel on the right border of the hole is a foreground pixel. Record this pixel. If $-sharp\_th < d_{m2} < sharp\_th$, then it indicates there are no sharp parallax transitions and all pixels are foreground pixels on the right border of the hole. Also, the first pixel on the right border is recorded.

The threshold *sharp_th* has important influences on the discrimination results. Too large or too small a *sharp_th* may result in incorrect foreground pixel discrimination. As the length of a hole *num* is less than or equal to its parallax difference d_m [9], and d_m has a close relation with *sharp_th*, we can choose a proper value of *sharp_th* according to *num*. Eq. (1) is empirically recommended for various videos to obtain the average accuracy of the discrimination results according to our test.

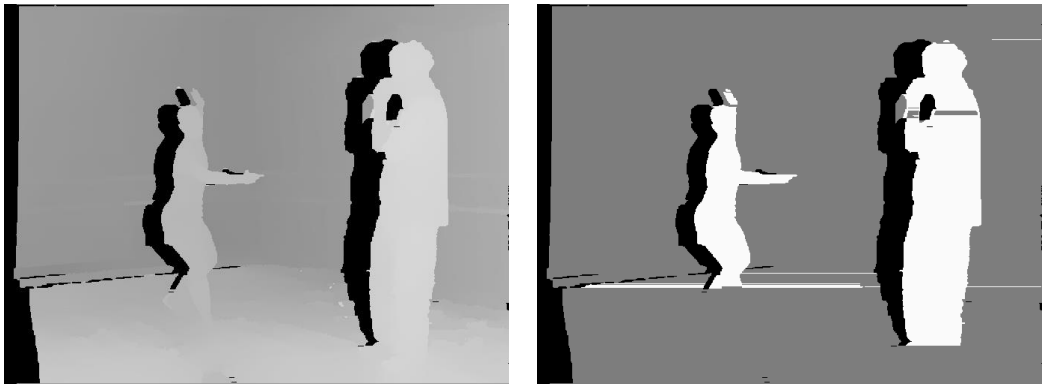
$$\begin{cases} sharp\_th = num & num \leq 2 \\ sharp\_th = num - 2 & 2 < num \leq 6 \\ sharp\_th = num / 2 & 6 < num \end{cases} \quad (1)$$

The processing of right view is similar to that of left view. The only difference between them is the scan order. The scan order for right view is opposite to that for left view.

(2) Foreground pixel marking

The purpose of this step is to mark the foreground pixels at the edges of a hole. First, a *0-1 matrix* **L** is created based on disparity map **M**. Each point in **M** has a corresponding value in 0-1 matrix at the same position (point). If a point in **M** is a *hole point* (parallax equals -128), then its corresponding value in **L** is set to 0; otherwise, it is set to 1.

Second, the values of the foreground pixels in **L** are changed from 1 to 2 by the following algorithm (take the left view as an example): Take the recorded pixel in step (1) as a start point, scan the pixels on the right side of this pixel in **M** from left to right. If the parallax difference d_{m3} between the start point and the current pixel satisfies $-sharp\_th \leq d_{m3} \leq sharp\_th$, then change the corresponding value of the current pixel in **L** to 2. If $d_{m3} < -sharp\_th$ or $d_{m3} > sharp\_th$, then the current pixel is a background pixel, the algorithm will stop marking and start to look for the next big hole. Repeat the algorithm until all the pixels are scanned. Consequently, all foreground pixels on both sides of big holes will be marked by 2, and the 0-1 matrix **L** will be changed to a *marking map* **L'**. An example of foreground pixel marking can be found in Figure 5.



(a) (b)

Figure 5. Disparity Map and Its Marking Map: (a) Disparity Map; (b) Marking Map Obtained from (a)

As shown in Figure 5 (b), only foreground pixels nearest to holes (in *horizontal direction*) are marked by the foreground-pixel-marking algorithm above. Note that part of the dancer's arm is not marked because her head is nearer to the hole than the unmarked part of her arm.

In addition, there are some big holes between the wall and the floor due to the inaccuracy of depth map (see Figure 5 (a)). They might result in incorrect foreground pixel marking. Fortunately, most sizes of these holes are not large in vertical direction; hence, they can be filled with the parallaxes of their neighboring pixels in disparity map before foreground pixel marking. Figure 5 (b) shows the final marking map obtained from (a).

(3) Hole Type *D* discrimination

Scan the marking map from left to right, line by line. If both sides of a hole (marked with value 0) are foreground (marked with value 2), then label the hole "Type *D*".

2.3. Disparity-Map-Based Hole-Filling

This module is used to fill "small hole", "medium hole", and hole "Type *D*" using the disparity-map-based approach.

The basic idea of the disparity-map-based approach is that holes are filled by copying the related pixels from the reference image according to the disparity map, which is associated with the destination image. There are three main steps in this approach:

(1) Hole detection based on disparity map;

(2) Big hole dilation. In this step, intrusive matching errors are eliminated;

(3) Hole filling. Holes are filled with background pixels from the reference image according to the disparity map.

The detailed descriptions of this approach can be seen in Ref. [9].

As for our proposed method, matching errors have been removed in module "Matching error elimination". Therefore, the step "big hole dilation" can be ignored here.

2.4. Inpainting

As shown in Figure 3, this module is used to fill the holes Type *A*, *B*, *C*, and *E* using an improved exemplar-based inpainting algorithm. Different from Criminisi's method [26], this module only uses background pixels to fill the holes so as to reduce the artifacts along foreground objects.

As can be seen from Figure 6, three submodules are contained in this module: *Patch selection*, *patch matching* [27], and *patch filling*.

Figure 6. Flowchart of the Improved Exemplar-based Inpainting Algorithm

For the convenience of the algorithm description, we adopt the notations shown in Figure 7 [26]. The description of these notations can be found in Table 1.

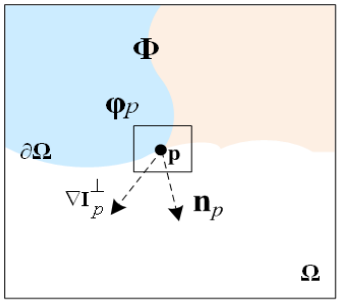


Figure 7. Notations Used for Inpainting Algorithm Description

Table 1. Notation Description

Ω	*Hole region* [10]
Φ	*Non-hole region* [10]
$\partial\Omega$	Hole contour/edge
$\mathbf{n}_p$	A unit vector orthogonal to $\partial\Omega$
φ_p	Patch of central point **p**
$\nabla \boldsymbol{I}_p^{\perp}$	Isophote direction in location **p**

We will discuss each of these submodules in detail in the following sections.

2.4.1. Patch Selection

The purpose of patch selection is to select the *optimal patch* [22] to be filled from candidate patches along the *fill front* [28]. There are three steps in the selection process.

Step 1: Hole contour marking

As shown in Figure 8, in this step, the disparity map **M** is scanned from left to right, line by line. If there are hole points in the *8-neighborhood-pixels set* of a non-hole point, then the hole point is labeled "*edge pixel*" (and it will be marked with "1", otherwise it will be marked with "0"). Hence we get a *hole contour map* corresponding to **M**.

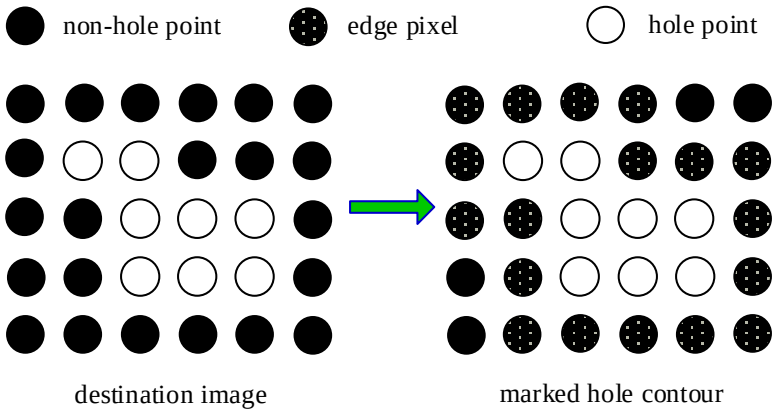


Figure 8. Illustration of Hole Contour Marking

Step 2: Priority computation

The priority $P(p)$ of an edge pixel p can be calculated by

$$P(p) = C(p) \cdot (r + s \cdot D(p) + t \cdot E(p)) \tag{2}$$

Where $C(p)$ is the *confidence term* which indicates the number of the background pixels in the patch $\boldsymbol{\varphi}_p$; $D(p)$ is the *data term* which gives special priority to the *isophote direction*; $E(p)$ is the *depth term*; and r, s, t are coefficients ($s >= r >= t$).

a). Confidence term $C(p)$

The confidence term $C(p)$ can be calculated by Eq. (3). Note that the foreground pixels are marked with 0, which ensures the *propagation direction* (from background pixels to foreground pixels) of texture and reflects the idea of background pixels being preferentially used for filling.

$$C(p) = \frac{\sum_{q \in \boldsymbol{\Phi} \cap \boldsymbol{\varphi}_p} C(q)}{area(\boldsymbol{\varphi}_p)}, \quad D(p) = \mathbf{n}_p * (\nabla \mathbf{I}_p)^{\perp}, \tag{3}$$

b). Data term $D(p)$

Expression of data term is shown in Eq. (3). In order to decrease the false boundary, an improved Sobel operator is proposed, whose horizontal mask $\mathbf{S}_x$ and vertical mask $\mathbf{S}_y$ are defined as follows:

$$\boldsymbol{S}_x = \begin{bmatrix} -e & 0 & e \\ -f & 0 & f \\ -g & 0 & g \end{bmatrix} \quad \boldsymbol{S}_y = \begin{bmatrix} -e & -f & -g \\ 0 & 0 & 0 \\ e & f & g \end{bmatrix} \tag{4}$$

where e, f, g are coefficients set to 1, 2, and 1, respectively if there are non-hole points under the mask. If there are *hole points*, the coefficients at the corresponding position are set to 0.

c). Depth term

The depth term is computed by

$$DE(p) = \sum_{q \in \boldsymbol{\Phi} \cap \boldsymbol{\varphi}_p} d_{\max} - D(q) \Big/ d_{\max} * area(\boldsymbol{\varphi}_p)', \tag{5}$$

where $d_{\max}$ is the maximal *depth value* in the depth map.

To ensure the texture propagates from background to foreground, the hole contour map (Figure 8) is scanned in accordance with the priorities shown in Figure 9.

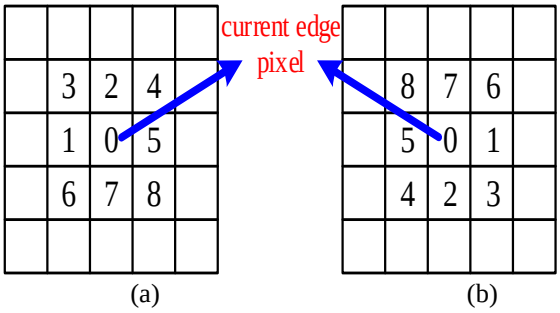


Figure 9. Illustration of Priority Level of the 8-Neighborhood-pixels Set. (a) Left View; (b) Right View

In Figure 9, "0" indicates current edge pixel, "1" indicates the highest priority level, and "8" indicates the lowest priority level. As we know from step 1, the edge pixel in hole contour map is marked with "1". The scanning is started from an edge pixel. It is considered to be the current edge pixel which is labeled with "0" priority level. The order of seeking the next edge pixel depends on the priority level. Take the left view as an example. First, we need to check the pixel with "1" priority level. If it is an edge pixel, then it is the pixel we are looking for; else, check other pixels based on their priority level until an edge pixel is found. Then, we calculate the priority of the found edge pixel using Eq. (2) and continue to search for the next edge pixel.

Step 3: Patch to be filled selection

After priority computation, the patch $\boldsymbol{\varphi}_p$ with the highest priority is selected to be filled. But there is a special case. If the priorities of all pixels equal 0, then the patch contains the pixel with maximum confidence is selected.

2.4.2. Patch Matching

The *matching patch* $\boldsymbol{\varphi}_q$ of patch $\boldsymbol{\varphi}_p$ can be found using Eq. (6). Depth information is introduced in this equation in order to find the best matching patch.

$$\begin{aligned} \boldsymbol{\varphi}_q &= \arg(\min(d(\boldsymbol{\varphi}_p, \boldsymbol{\varphi}_q))) \\ d &= SSD_{RGB}(\boldsymbol{\varphi}_p, \boldsymbol{\varphi}_q) + 3 * SSD_D(\boldsymbol{\varphi}_p, \boldsymbol{\varphi}_q) \end{aligned}, \tag{6}$$

Our method is similar to the patch matching method presented in Ref. [29] except that SSD (sum of absolute difference) is used instead of SAD. This is because there will be fewer patches with the same minimum value if SSD is applied. If there are multiple patches with the same minimum value, the nearest patch is chosen. Note that the *search range* is determined by the *K-nearest neighbor method* (K = 5).

Note that the SSD of a candidate/matching patch $\boldsymbol{\varphi}_q$ is set to infinity when there are holes in patch $\boldsymbol{\varphi}_q$ or no background pixels in patch $\boldsymbol{\varphi}_q$ can be used to fill patch $\boldsymbol{\varphi}_p$. Moreover, there may be artifacts at the edge of foreground objects due to the influence of foreground pixels, which can be seen in Figure 10. To reduce these artifacts, only background pixels of the patch $\boldsymbol{\varphi}_p$ are considered when calculating SSD.

Figure 10. Illustration of Artifacts at the Edge of Foreground Object

2.4.3. Patch Filling

If a hole point in patch $\boldsymbol{\varphi}_p$ correspond to a background pixel at the same position in patch $\boldsymbol{\varphi}_q$, then the background Pixel is used to fill the hole point; else the hole point is not filled. The confidence term of the filled point is updated by that of the central point of patch $\boldsymbol{\varphi}_p$. In addition, the hole contour map is also updated.

Repeat operations described in Sec. 2.4.1, Sec. 2.4.2, and Sec. 2.4.3 until all holes are filled, and finally we'll get a destination image without holes.

3. Evaluations and Discussions

In this paper, "Ballet", "Breakdancers" [19] and other six sequences are used for evaluations (subjective evaluation and objective evaluation). Subjective evaluation is mainly conducted by comparing the visual quality of the image filled by different algorithms. Meanwhile, objective evaluation focuses on contrasting the time complexity of these methods. In order to generate novel views, the 3D image warping equation in Ref. [12, 23] is applied in the experiment. In addition, large baseline is applied to produce large holes. The parameters n and r in 3D image warping equation are set to be different according to the resolution of the test sequence ($D_{zps} = 0$). The parameters for warping, the size of patches used in inpainting algorithm, and the widths of the biggest hole in the left view and right view (expressed in pixels) are described in detail in Table 2.

Table 2. Parameter Settings for Experiment

Test sequences	Resolution	Frame Number	*n*	*r*	Size of patch	Widths
Ballet	768×1024	0-9	±4	819.2	31×31	128, 128
Breakdancers	768×1024	0-9	±4	819.2	19×19	131, 128
Flower	540×960	30-39	±3	819.2	25×25	114, 92
Road	540×960	0-9	±3	819.2	39×39	124, 127
Lawn	540×960	0-9	±3	819.2	33×33	127, 126
Angkorwat	352×576	0-9	±3	819.2	29×29	64, 59
Stair	352×576	0-9	±3	409.6	35×35	73, 72
Temple	352×576	0-9	±3	409.6	35×35	52, 59

3.1. Subjective Evaluation

3.1.1. Evaluation of Matching Error Correction Algorithm

As we know, the accumulation of matching errors is a shortcoming in the performance of the classical inpainting algorithm. Figure 11 illustrates the results with and without matching error correction when the proposed method is carried out.

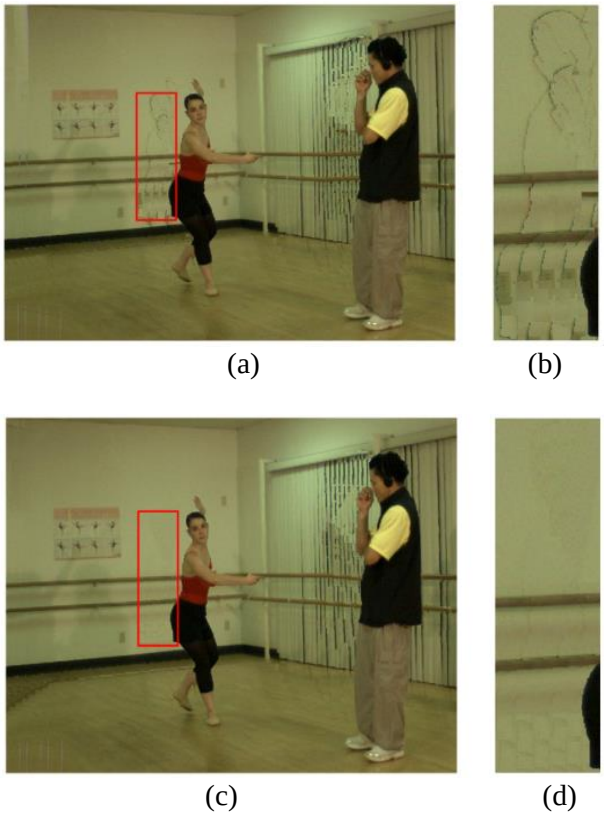
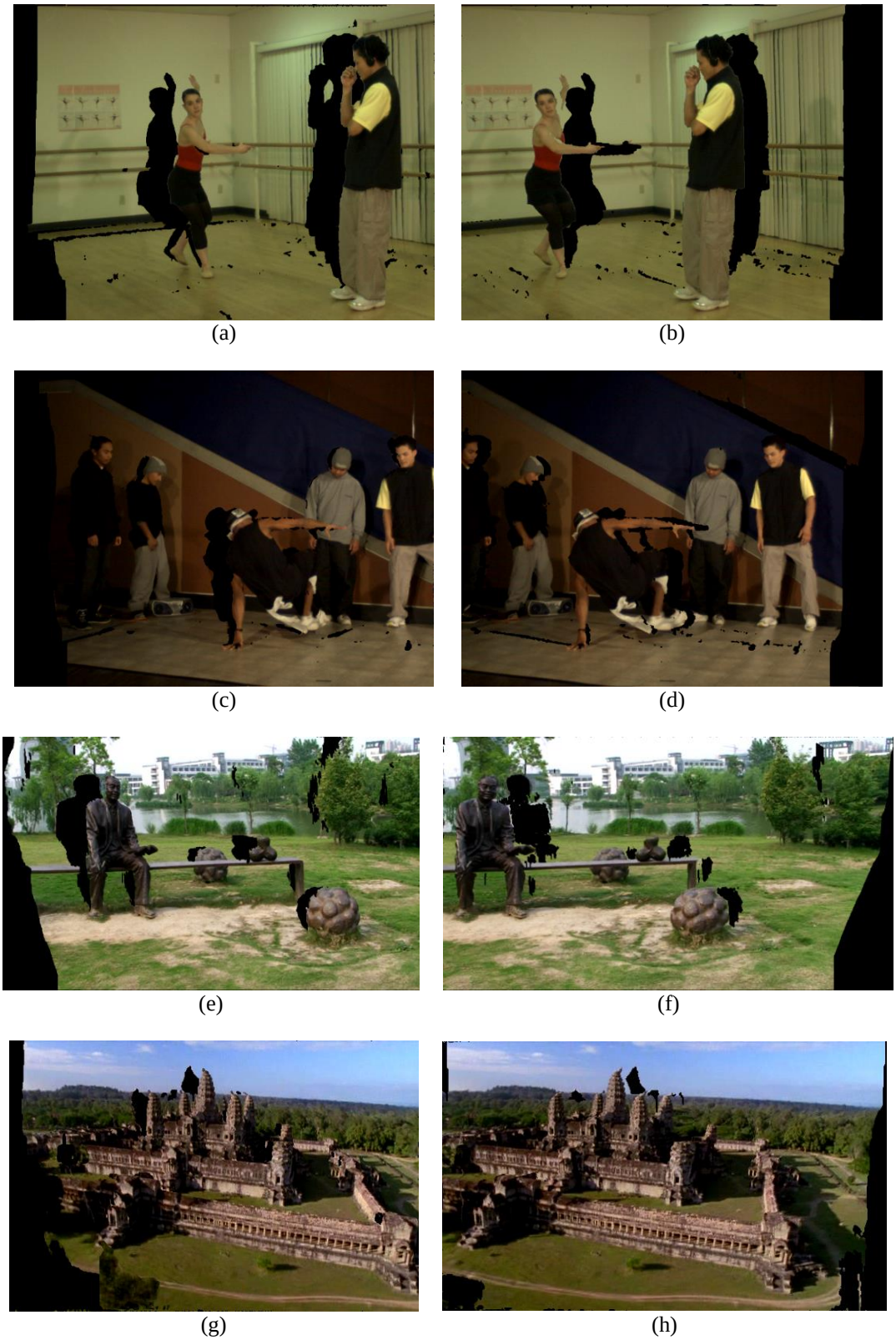


Figure 11. Comparison of Generated Views with and without Matching Error Correction. (a) Generated View Without Matching Error Correction; (b) Enlargement of Image (a); (c) Generated View with Matching Error Correction; (d) Enlargement of Image (c)

From Figure 11 (b) and (d), we can see that after removing the matching errors, a more natural hole-filled image can be generated by the proposed method. Using the matching patch to fill holes depending on the information of the damaged areas is the main idea of inpainting. If there are matching errors at the edge of big holes, matching errors are more likely to be used to fill the big holes for many times, which leads to the accumulation of matching errors. So it is necessary to remove the matching errors before the implementation of the inpainting algorithm, and by this way we may get a perfect repaired image.

Figure 12 shows more novel views generated from different sequences with matching error correction.



(a) (b)

(c) (d)

(e) (f)

(g) (h)

Figure 12. Destination Images after Matching Error Correction

The test sequences from top to down in Figure 12 are “Ballet”, “Breakdancers”, “Lawn”, and “Angkorwat”. Images from left to right are left views and right views, respectively.

3.1.2. Comparison of Different Algorithms

Figure 13 shows the performance comparison of disparity-map-based approach [9], Gautier's inpainting algorithm, and the proposed method. The main differences of images generated by these three algorithms are marked with red rectangles in Figure 13. Background textures of the test sequences are complex.

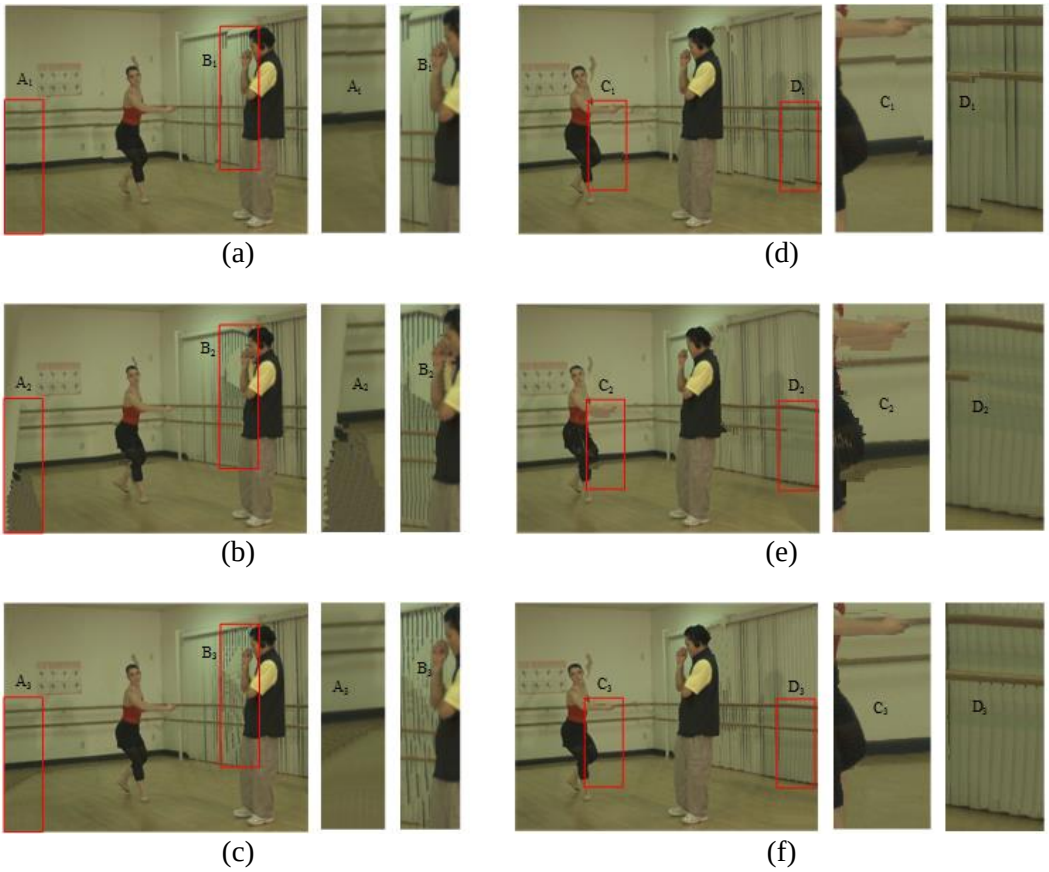


Figure 13. Performance Comparison of the Three Algorithms for Complex Background Textures. (a) Disparity-map-based Approach (Left View); (b) Gautier's Inpainting Algorithm (Left View); (c) Proposed Method (Left View); (d) Disparity-map-based Approach (Right View); (e) Gautier's Inpainting Algorithm (Right View); (f) Proposed Method (Right View)

There is an obvious "fracture" in Figure 13 (a) and (d). These views are generated by disparity-map-based approach, which fills the holes with pixels simply copied from the reference image. Fractures might appear when the hole is big due to the algorithms regardless of texture.

As for Gautier's Inpainting Algorithm, the directions of texture propagation specified in the algorithm may not propagate along with the *texture direction*, hence artifacts produced, as shown in A2 and D2 in Figure 13. Furthermore, there are artifacts at the edge of foreground objects, which can be found in B2 and C2 in Figure 13 (b) and (c), respectively.

Novel views generated by the proposed method are shown in Figure 13 (c) and (f). We can see that better image quality of views can be achieved from these images.

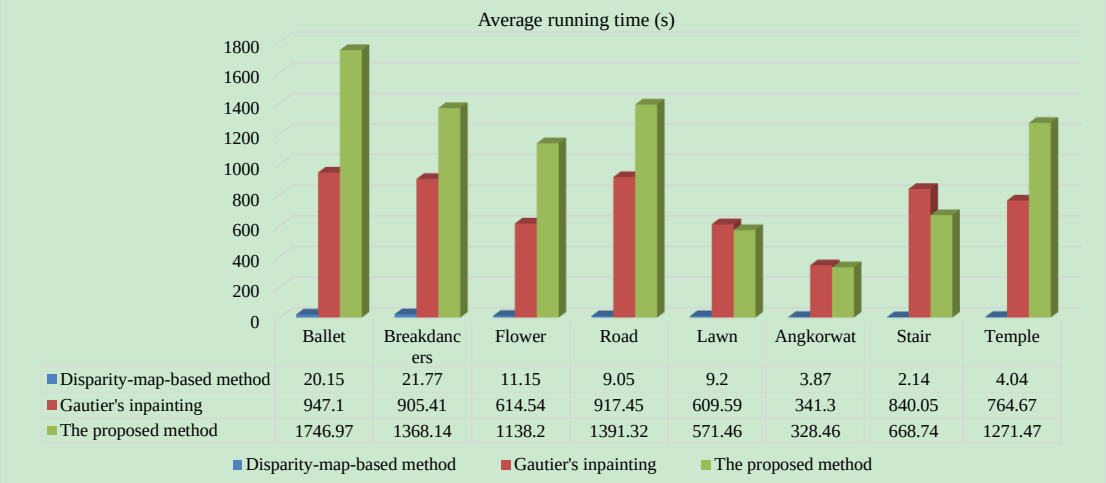
Figure 14. Performance Comparison In The Case Of Simple or Flat Textures

Figure 14 shows performance comparison of these three algorithms in the case of simple or flat textures. Novel views generated by disparity-map-based approach, Gautier's inpainting algorithm, and the proposed method are listed from left to right. (a), (b), and (c) are novel left views generated from "Breakdancers" sequence; (d), (e), and (f) are novel right views generated from "Breakdancers" sequence; (g), (h), and (i) are novel left views generated from "Lawn" sequence; (j), (k), and (l) are novel right views generated from "Lawn" sequence; (m), (n), and (o) are novel left views generated from "Angkorwat" sequence; (p), (q), and (r) are novel right views generated from "Angkorwat" sequence.

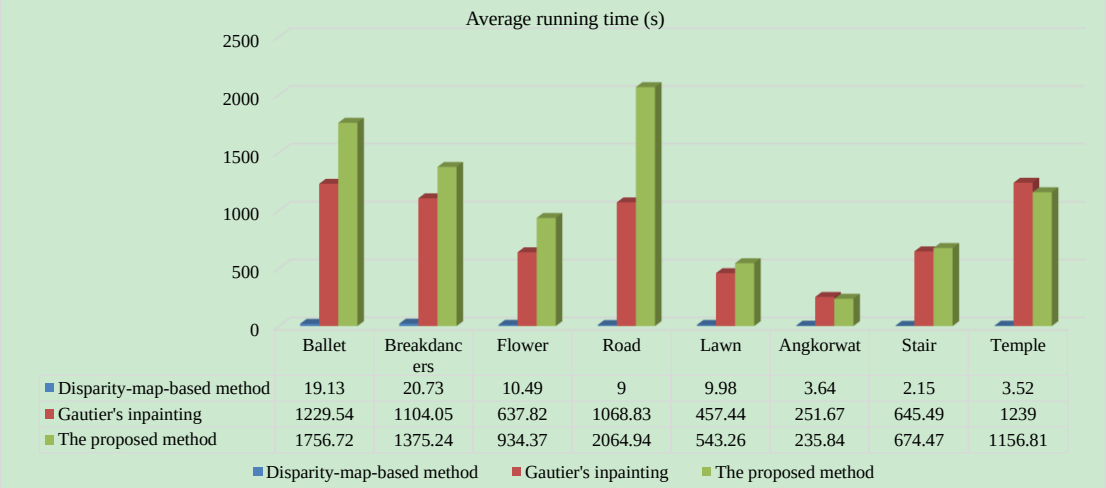
As can be seen from Figure 14, Obvious artifacts appear in the views generated with disparity-map-based approach. Gautier's inpainting algorithm cannot remove all the matching errors, which can be seen in Figure 14 (b), and its performance degrades when filling the holes at the border of the destination image. Moreover, there are some artifacts at the edge of foreground objects. In comparison with these two algorithms, the proposed method can synthesize more natural images, though there may still be some undesired artifacts left in some cases.

3.2. Objective Evaluation

In a DIBR system with shift-senor camera setup, there are only horizontal parallaxes contained in the generated stereo pair. Therefore, it is not appropriate to evaluate the performance of these three methods by objective criteria such as PSNR and SSIM. Hence, objective evaluations are performed by comparing the time complexity. The evaluation results of time complexity of disparity-map-based approach, Gautier's inpainting algorithm and the proposed method are shown in Figure 15.



(a) Average Running Time when Generating Left Views



(b) Average Running Time when Generating Right Views

Figure 15. Comparison of Time Complexity

Note that the "average running time" in Figure 15 indicates the average time of synthesizing one destination image, and the first 10 frames of each sequence are used for evaluation. As can be seen from Figure 15, the disparity-map-based approach has the lowest time complexity and Gautier's inpainting algorithm performs better than the proposed method. This is because the proposed method needs to distinguish foreground from background and its priority computation is more complex than that of Gautier's.

However, it is exceptional when novel views are generated from "Angkorwat" sequence, as shown in Figure 15. In this case, most holes in the destination image are small and they are filled by the disparity-map-based approach rather than the time-consuming inpainting algorithm in our proposed method. As a result, the average running time of our proposed method is less than Gautier's inpainting algorithm. We can conclude that the time complexity of our method will be reduced in the case of filling the small holes.

4. Conclusions

The disparity-map-based approach proposed by Ref. [9] can be used to fill small holes, and the big holes with no background pixels on both sides of them (hole Type *D*) with quite good image quality. But for other types of big hole (hole Type *A*, *B*, *C*, *E*), its performance degrades. In contrast to the disparity-map-based approach, the inpainting algorithm can fill all kinds of holes. However, it is time-consuming and easy to produce artifacts. In this paper, a novel hole-filling method combining the disparity-map-based approach and inpainting algorithm is presented. The proposed method can effectively avoid "fracture" phenomena and other artifacts at the edge of foreground objects in comparison with the disparity-map-based approach and Gautier's inpainting algorithm. Besides, it can also ensure the "authenticity" of non-hole area of destination image without depth map smoothing. Experimental results show that the proposed method can synthesize more natural images in the case of long baseline in comparison with the disparity-map-based approach and Gautier's inpainting algorithm regardless of the average running time.

Acknowledgements

This work was jointly supported by the Medical Research Project of Chongqing Municipal Commission of Health and Family Planning (No. 2013-2-124), the 2013 Innovative Team Construction Project of Chongqing Universities (No. KJTD201331),

the National Natural Science Foundation of China (No. 61201347), Fundamental Research Funds for the Central Universities (No. CDJZR13185502), and the Graduate Student Research Innovation Project of Chongqing (No. CYS14021).

References

[1] P. J. Lee and Effendi, "Adaptive Edge-Oriented Depth Image Smoothing Approach For Depth Image Based Rendering", Proceedings of the 2010 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting, Shanghai, China, **(2010)**, pp. 1-5.
[2] K. Luo, D.X. Li, Y. M. Feng and M. Zhang, "Depth-Aided Inpainting for Disocclusion Restoration of Multi-View Images Using Depth-Image-Based Rendering", Journal Of Zhejiang University-Science A, vol. 10, no. 12, **(2009)**, pp. 1738-1749.
[3] K. J. Oh, S. Yea and Y. S. Ho, "Hole Filling Method Using Depth Based in-Painting for View Synthesis in Free Viewpoint Television and 3-D Video", Proceedings of the 2009 Picture Coding Symposium, Chicago, USA**, (2009)**, pp. 1-4.
[4] S. M. Muddala, M. Sjostrom and R. Olsson, "Depth-Based Inpainting for Disocclusion Filling", Proceedings of the 3DTV-Conference: The True Vision - Capture, Transmission and Display of 3D Video, Budapest, Hungary, **(2014)**, pp. 1-4.
[5] R. Liu, W. Tan, Y. Wu, Y. Tan, B. Li, H. Xie, G. Tai and X. Xu, "Deinterlacing of Depth-Image-Based Three-Dimensional Video For A Depth-Image-Based Rendering System", Journal of Electronic Imaging, vol. 22, no. 3, **(2013)**.
[6] P. J. Lee and Effendi, "Nongeometric Distortion Smoothing Approach for Depth Map Preprocessing", IEEE Transactions on Multimedia, vol. 13, no. 2, **(2011)**, pp. 246-254.
[7] S. M. Seitz, C. R. Dyer, "View Morphing", Proceedings of the ACM SIGGRAPH Conference on Computer Graphics, New Orleans, USA, **(1996)**, pp. 21-30.
[8] J. Gautier, O. Le Meur and C. Guillemot, "Depth-Based Image Completion For View Synthesis", Proceedings of the 3DTV-Conference: The True Vision - Capture, Transmission and Display of 3D Video, Antalya, Turkey, **(2011)**, pp. 1-4.
[9] R. Liu, H. Xie, F. Tian, Y. Wu, G. Tai, Y. Tan, W. Tan, H. Chen and L. Ge, "Hole-filling Based on Disparity Map for DIBR", KSII Transactions on Internet and Information Systems. vol. 6, no. 10, **(2012)**, pp. 2663-2678.
[10] X. Xu, L. M. Po, C. H. Cheung, L. Feng, K. H. Ng, K. W. Cheung, "Depth-aided Exemplar-based Hole Filling for DIBR View Synthesis", Proceedings of the 2013 IEEE International Symposium on Circuits and Systems, Beijing, China, **(2013)**, pp. 2840-2843.
[11] L. Pei-Jun and Effendi, "Non Geometric Distortion Smoothing Approach For Depth Map Preprocessing", IEEE Transactions On Multimedia, vol. 13, no. 2, **(2011)**, pp. 246-254.
[12] R. Liu, Y. Tan, F. Tian, H. Xie, G. Tai, W. Tan, J. Liu, X. Xu, C. Kadri and N. Abakah, "Visual Fatigue Reduction Based on Depth Adjustment for DIBR System", KSII Transactions on Internet and Information Systems, vol. 6, no. 4, **(2012)**, pp. 1171-1187.
[13] L. Zhang and W.J. Tam, "Stereoscopic Image Generation Based on Depth Images for 3D TV", IEEE Transactions on Broadcasting, vol. 51, no. 2, **(2005)**, pp. 191-199.
[14] W. Y. Chen, Y. L. Chang, S. F. Lin, L. F. Ding and L. G. Chen, "Efficient Depth Image Based Rendering with Edge Dependent Depth Filter and Interpolation", Proceedings of the IEEE International Conference on Multimedia and Expo, Amsterdam, Netherland **(2005)**, pp. 1314-1317.
[15] Y. R. Horng, Y. C, Tseng and T. S. Chang, "Stereoscopic Images Generation With Directional Gaussian Filter", Proceedings of the 2010 IEEE International Symposium on Circuits and Systems, Paris, France, **(2010)**, pp. 2650-2653.
[16] Y. K. Park, K. Jung, Y. Oh, S. Lee, J. K. Kim, G. Lee, H. Lee, K. Yun, N. Hur and J. Kim, "Depth-Image-Based Rendering for 3DTV Service Over T-DMB", Signal Processing: Image Communication, vol. 24, no 1-2, **(2009)**, pp. 122-136.
[17] M. Koppel, M. Ben Makhlouf and P. Ndjiki-Nya, "Optimized Adaptive Depth Map Filtering", Proceedings of the 20th IEEE International Conference on Image Processing (ICIP), Melbourne, Australia, **(2013)**, pp. 1356-1360.
[18] K. M. Chang and T. C. Lin, "Parallax-Guided Disocclusion Inpainting For 3D View Synthesis", Proceedings of the 2012 IEEE International Conference on Consumer Electronics, Las Vegas, USA **, (2012)**, pp. 398-399.
[19] I. Daribo and B. Pesquet-Popescu, "Depth-Aided Image Inpainting for Novel View Synthesis", Proceedings of the 2010 IEEE International Workshop on Multimedia Signal Processing, Saint Malo, France, **(2010)**, pp. 167-170.
[20] H. Wu, J. Feng, H. Zhang and Q. Lv, "A Virtual View Synthesis Algorithm Based on Image Inpainting", Proceedings of the third International Conference on Networking and Distributed Computing, Hangzhou, China, **(2012)**, pp. 153-156.

[21] K. Wang, P. An, H. Cheng, H. Li and Z. Zhang, "A New Method of DIBR Based on Background Inpainting", Proceedings of the 9th International Forum on Digital TV and Wireless Multimedia Communication, Shanghai, China **, (2012)**, pp. 478-484.
[22] S. Choi, B. Ham and K. Sohn, "Space-Time Hole Filling With Random Walks in View Extrapolation for 3D Video", IEEE Transactions on Image Processing, **(2013)**, vol. 22, no. 6, pp. 2429-2441.
[23] R. Liu, H. Xie, G. Q. Tai, Y. C. Tan, R. L. Guo, W. Y. Luo, X. Y. Xu and J. L. Liu, "Depth Adjustment for Depth-Image-Based Rendering in 3D TV System", Journal of Information and Computational Science, vol. 8, no. 16, **(2011)**, pp. 4233-4240.
[24] S. Zinger, L. Do and P. H. N. De With, "Free-Viewpoint Depth Image Based Rendering", Journal of Visual Communication and Image Representation, vol. 21, no. 5-6, **(2010)**, pp. 533-541.
[25] R. Liu, G. Tai, H. Xie, Y. Tan, R. Guo, W. Luo and X. Xu, "Matching Error Correction for Depth-Image-Based Rendering Based on Disparity Map Filtering", Journal of Computational Information Systems, vol. 8, no. 3, **(2012)**, pp. 9713-9720.
[26] A. Criminisi, P. Perez and K. Toyama, "Object Removal By Exemplar-Based Inpainting", Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Madison, USA, **(2003)**, vol. 2, pp. 721-728.
[27] K. M. Chang, T. C. Lin and Y. M. Huang, "Parallax-Guided Disocclusion Inpainting for 3D View Synthesis", Proceedings of the 2012 IEEE International Conference on Consumer Electronics, Las Vegas, USA,**(2012)**. pp. 398-399.
[28] A. Criminisi, P. Perez and K. Toyama, "Region Filling and Object Removal by Exemplar-Based Image Inpainting", IEEE Transactions on Image Processing, vol. 13, no. 9, **(2004)**, pp. 1200-1212.
[29] K. Luo, D. Lu, Y. Feng and M. Zhang, "Arbitrary View Generation Based on DIBR and Depth-Aided Image Inpainting", Journal of Image and Graphics, vol. 15, no. 3, **(2010)**, pp. 443-449.

Authors

Ran Liu received his BE, ME, and DE degrees in computer science from Chongqing University, Chongqing, China, in 2001, 2004, and 2007, respectively. He worked as a postdoctoral researcher for Homwee Technology Co., Ltd., Chengdu, China, from 2008 to 2010. He is now an associate professor at the College of Communication Engineering and the College of Computer Science, Chongqing University, China. His research interests include 3D imaging, medical image processing, and medical imaging.

Zekun Deng is currently pursuing her master's degree at the College of Communication Engineering, Chongqing University, Chongqing, China. She received her bachelor's degree from Chongqing University, Chongqing, China, in 2014. Her research interests include 3D TV and medical image processing.

Lin Yi received her bachelor's degree from Chongqing Medical University, Chongqing, China, in 2004. Her research interests include medical image processing and medical imaging.

Zhenwei Huang is currently pursuing his master's degree at the College of Communication Engineering, Chongqing University, Chongqing, China. He received his bachelor's degree from Chongqing University, Chongqing, China, in 2012. His research interests include 3D TV and stereo image processing.

Donghua Cao is currently pursuing her master's degree at the Key Laboratory of Dependable Service Computing in Cyber Physical Society of the Ministry of Education, Chongqing University, Chongqing, China. She received her bachelor's degree from Qufu Normal University, Shandong, China, in 2013. Her research interests include 3D TV and stereo image processing

Miao Xu is currently pursuing her master's degree at the Key Laboratory of Dependable Service Computing in Cyber Physical Society of the Ministry of Education, Chongqing University, Chongqing, China. She received her bachelor's degree from Shanxi Normal University, Shanxi, China, in 2014. Her research interests include 3D TV and stereo image processing.

Ruishuang Jia is currently pursuing her master's degree at the Key Laboratory of Dependable Service Computing in Cyber Physical Society of the Ministry of Education, Chongqing University, Chongqing, China. She received her bachelor's degree from Northeast University at Qinhuangdao, Hebei, China, in 2014. Her research interests include 3D TV and IC design**.**