Hole-filling Based on Disparity Map for DIBR

Ran Liu[1,2,3], Hui Xie[1], Fengchun Tian[1], Yingjian Wu[4], Guoqin Tai[1], Yingchun Tan[1], Weimin Tan[1], Bole li[1], Hengxin Chen[3] and Liang Ge[2,3]

[1] College of Communication Engineering, Chongqing University,
Chongqing 400044, China
[2] Chongqing Key Laboratory of Software Theory & Technology,
Chongqing 400044, China
[3] College of Computer Science, Chongqing University,
Chongqing 400044, China
[4] Homwee Technology Co., Ltd, Changhong Group,
Chengdu 610031, China
[e-mail: {ran.liu_cqu, hui.xie_cqu}@qq.com, fengchuntian@cqu.edu.cn, taiguoqin@163.com, {yingchuntan_cqu, weimintan_cqu}@126.com, bl_li@foxmail.com, gelinag@cqu.edu.cn, chenhengxin@cqu.edu.cn, wuyj@changhong.com]
*Corresponding author: Ran Liu

Received June 23, 2012; revised August 26, 2012; revised September 28, 2012; accepted October 2, 2012; published October 29, 2012X, 2012

Abstract

Due to sharp depth transition, big holes may be found in the novel view that is synthesized by depth-image-based rendering (DIBR). A hole-filling method based on disparity map is proposed. One important aspect of the method is that the disparity map of destination image is used for hole-filling, instead of the depth image of reference image. Firstly, the big hole detection based on disparity map is conducted, and the start point and the end point of the hole are recorded. Then foreground pixels and background pixels are distinguished for hole-dilating according to disparity map, so that areas with matching errors can be determined and eliminated. In addition, parallaxes of pixels in the area with holes and matching errors are changed to new values. Finally, holes are filled with background pixels from reference image according to these new parallaxes. Experimental results show that the quality of the new view after hole-filling is quite well; and geometric distortions are avoided in destination image, in contrast to the virtual view generated by depth-smoothing methods and image inpainting methods. Moreover, this method is easy for hardware implementation.

Keywords: 3D TV, depth-image-based rendering, hole filling, 3D image warping, disparity map

This work was jointly supported by the National Natural Science Foundation of China (Grant No. 61071190, 61274131, 61070246, 61171158), the National Science Foundation for Young Scholars of China (Grant No. 61201347, 61202346, 61003246), the Natural Science Foundation Project of CQ CSTC (Grant No. cstc2012jjA40011), and the Fundamental Research Funds for the Central Universities (Grant No. CDJZR10180013, CDJXS12160002).

http://dx.doi.org/10.3837/tiis.2012.10.013

1. Introduction

The recent popularity of 3D films has prompted the developing boom of 3D TV. Many new 3D TV technologies and solutions have been proposed [1][2]. Among these new developments, *depth-image-based rending* (DIBR) technique, which can generate *arbitrary views* from *reference image* and its associated *depth image* as well as reduce the *transmission bandwidth*, has been widely accepted for 3D TV system [1][3][4][5].

However, due to the sharp depth transition in depth image, big holes will appear in the generated virtual view (*destination image*) after DIBR [6][7]. How to fill these holes is a main problem for the DIBR system. Many methods for hole-filling have been proposed [3][4][5][7][8][9][10][11]. These methods can be classified into three types according to their technical features:

(1) Preprocessing depth image

In this method, depth image is smoothed by smoothing filter to reduce the size of holes. Fehn *et al*. adopted *Gaussian filter* to smooth the whole depth map before *3D image warping* [12]. Although the size of holes are decreased with this method, *geometric distortions* also appear. After that, many filters such as *edge dependent depth filter*, *asymmetric Gaussian filter*, and *Directional Gaussian filter* (DGF), were proposed to reduce the *vertical geometric distortion* after smoothing [3][5][6][9]. In addition, people also gave out solutions to the hardware implementation of these complex smoothing operations.

(2) Using multiple videos

In this method, two or more *video streams* are transmitted to provide the viewer with exact quality of 3D-effect [1][3][10]. Due to the variation of the illumination conditions of each video stream, *color correction* should be performed on the filled big hole to avoid inconsistency.

(3) Filling holes in the generated destination image

Common solutions for the post-processing of hole-filling are background or neighborhood *pixel interpolation*, these methods are efficient for small size holes. Complex algorithms such as *image inpainting* are also proposed in recent years [7][8][11].

Generally, preprocessing of depth image would degrade the quality of destination image as well as introduce geometric distortions; transmitting multiple videos may result in large bandwidth requirement and encounter the problem in real time processing; and image inpainting is highly complex in most cases. In this paper, a new hole-filling method based on disparity map is proposed. It is a post-processing method. As depth image is substituted by disparity map, and *matching error elimination* is also considered, this method can achieve reasonably good result under the premise of not degrading the quality of *non-hole area* in destination image. Moreover, it is relatively easy for hardware implementation.

The remaining portions of this paper are organized as follows. In Section 2, causes of the holes and matching errors are discussed in detail. Section 3 is devoted to describing the hole-filling method based on disparity map. In Section 4, the performance of the method is evaluated. Conclusions can be found in Section 5.

2. Holes and Matching Errors

2.1 Holes

The reason that big holes may appear in destination image can be explained by the 3D image warping equations [13], which are described below

$$\begin{cases} u_{des} - u_{ref} = n \cdot \dfrac{r}{4096} \cdot (D_{zps} - D(u_{ref}, v_{ref})), \\ v_{des} = v_{ref} \end{cases} \tag{1}$$

where (u_{ref}, v_{ref}) and (u_{des}, v_{des}) are matching points in reference image and destination image respectively; n is the number of virtual view when multi views are generated using DIBR for an *auto-stereoscopic* display, if destination image is left view, $n < 0$, otherwise $n > 0$; r represents scale factor; *Dzps* is the depth of *ZPS plane* [12][14], and $D(u_{ref}, v_{ref})$ is the depth value of point (u_{ref},v_{ref}) in depth image. Note that the equations above adapts to the DIBR system with *shift-sensor camera setup* [14].

Suppose point **F** lies on the left of point **B** and they are neighborhoods in reference image, the *horizontal shifts* of **F'** and **B'** relative to **F** and **B** are d_f and d_b respectively, which is shown in **Fig. 1**.

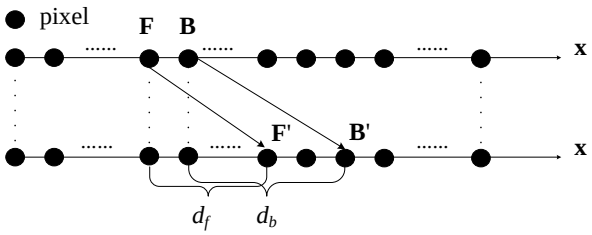


Fig. 1. Illustration of horizontal shifts.

Let D_{wf} and D_{wb} be the depth values of point **F** and point **B**, then according to (1) we have

$$d_f - d_b = -n \cdot \frac{r}{4096} \cdot \left[D_{wf} - D_{wb} \right] \tag{2}$$

If destination image is assigned to be left view, then $n < 0$. Thus from (2), there are three situations:

a) if D_{wf} equals D_{wb}, then horizontal shifts of the two adjacent points are the same and the point **F'** will appear on the left of **B'**;

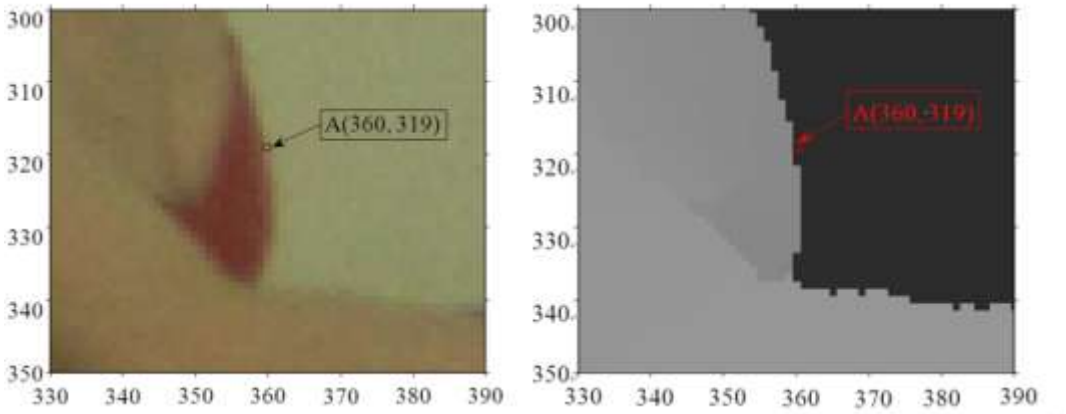
b) if D_{wf} is greater than D_{wb}, then **F'** will appear on the right of **B'**, or they are the same point;

c) if Dw_f is smaller than D_{wb} (hence **B'** is an *foreground pixel*), the shift of **F** will be smaller than that of **B**, so holes will appear between **F'** and **B'**, i.e. holes appear on the left of *foreground pixels*. In other words, pixels on the right of holes must be foreground pixels.

Similarly, pixels on the left of holes must be foreground pixels on condition that destination image is right view.

2.2 Matching Errors

Objects contours are often inaccurate in depth image [5, 9]. Generally, foreground objects contours in reference image are larger than that in the associated depth image. **Fig. 2** shows the images copied from "Ballet" sequence. As it can be seen from **Fig. 2**, point **A** with coordinates (360, 319) lies on the foreground object contour in reference image. However, the point **A'** with the same coordinates in depth image indicates a background pixel.



(a) (b)

Fig. 2. Illustration of the inaccuracy of depth value at edges of foreground objects. **(a)** reference image; **(b)** the associated depth image.

As foreground objects contours in reference image are inconsistent with that in depth image, it is inevitable that 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. We call this phenomenon *matching errors* [15], as shown in **Fig. 3**. If it is not eliminated, the result after hole-filling will be like **Fig. 3 (c)** and **(d)**.

(a) (b)

(c) (d)

Fig. 3. Illustration of matching errors. **(a)** the generated right view after 3D image warping; **(b)** matching errors caused by inaccuracy of depth value, labeled with rectangle; **(c)** hole-filling by the proposed method without eliminating matching errors($l = 0$); **(d)** Enlargement of image **(c)**.

Three methods can be used to eliminate matching errors.

(1) Smooth depth image, so as to extend the foreground object contour.
(2) Do not warp background pixels at the edges of high discontinuities [10].
(3) Expand edges of holes where matching errors exist after 3D image warping.

It is difficult to avoid over-smoothing which may result in obvious distortions in destination image with the first method. For the second approach, it is not easy to label background pixels at edges of high discontinuities accurately. Hence we prefer the third method for matching error elimination for our hole-filling method.

3. Description of Hole-filling Method

To fill holes as well as correct matching errors, we propose a method in which disparity map is used.

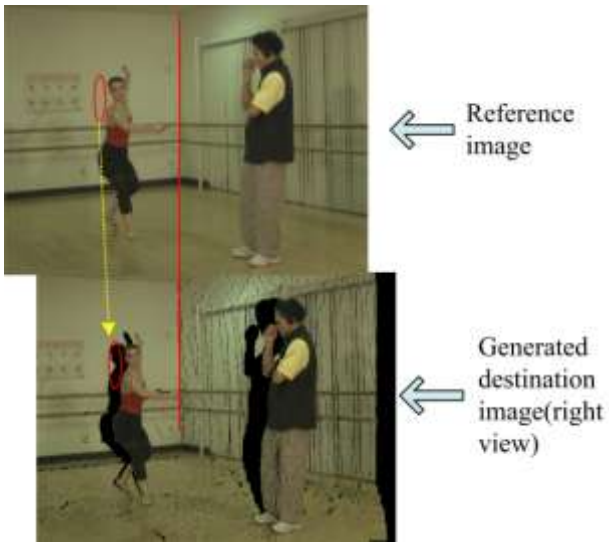


Fig. 4. Hole-filling with background pixels from reference image. Images are aligned for comparison.

The basic idea of the proposed method is that holes are filled by copying the related pixels from reference image according to disparity map **M**, which is associated with destination image. The value of each point in **M** is defined by $u_{des} - u_{ref}$(parallax); and if the point is a hole, parallax of this point is set to –128 (the horizontal parallax almost cannot achieve at -128 in order to avoid visual fatigue [13]). In fact, **M** represent horizontal shifts from reference image to destination image. To fill big holes, corresponding background pixels in reference image are shifted by the same displacement as their neighboring foreground pixels do. Consequently, holes are filled by these background pixels from reference image, as depicted in **Fig. 4**.The proposed method consists of three steps: (i) detecting holes, (ii) dilating big holes, and (iii) filling holes, as shown in **Fig. 5.** In the following, these three steps will be addressed in detail.

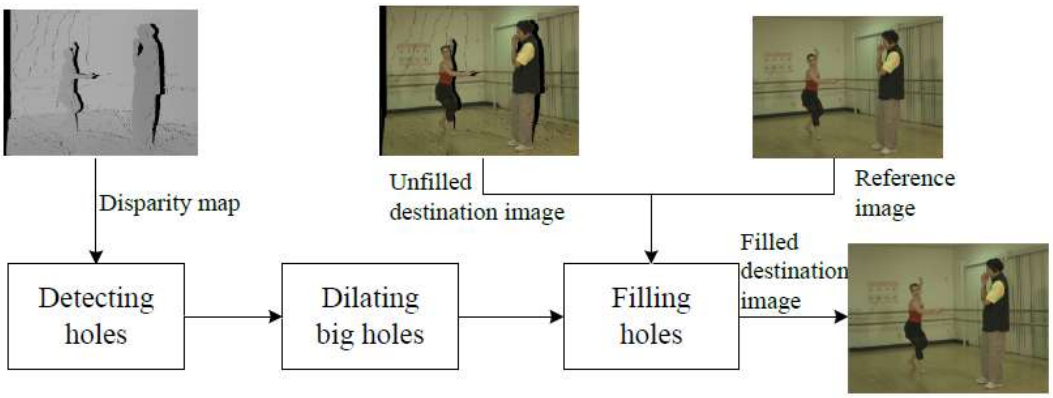


Fig. 5. Flowchart of the proposed method, take right view for example.

***Step 1**: Detecting holes*

The procedure of holes detection can be described by **Fig. 6**. The figure illustrates how the block "Detecting holes" in **Fig. 5** works by a practical example. To detect holes, disparity map **M** is scanned from left to right, line by line. As mentioned above, each pixel in destination image has a corresponding parallax value in disparity map at the same position (point). If the parallax of a point equals -128, then the point is a *hole-point*; else it is a *non-hole point*. The start and the end points of the hole are recorded during the scanning process. If the number of *hole-points* in a hole is greater than the pre-set threshold *len_bighole*, the hole is labeled as a "big hole", and go to ***Step 2***; else record the *parallax-value* of the right *non-hole point* which is nearest to the hole, then go to ***Step 3***.

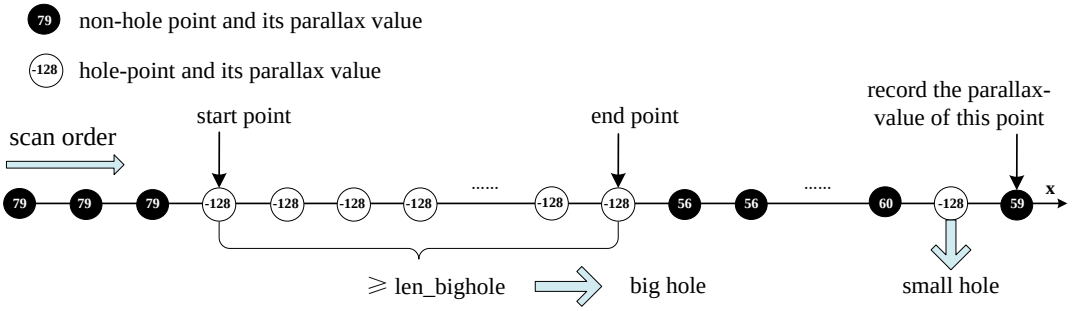


Fig. 6. Illustration of holes detection processing.

Step 2: *Dilating big holes according to* **M**

The purpose of this step is to eliminate matching errors. What's different from S. Zinger's method [10] is that we expand edges of holes where matching errors appear after 3D image warping. Our method can remove errors without removing correct texture pixels. All these processes are performed only on **M**. The flowchart can be found in **Fig. 7**.

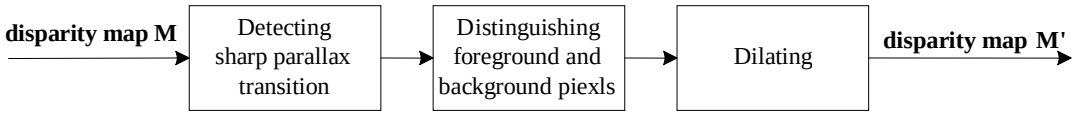


Fig. 7. Flowchart of dilating big holes.

As big holes will appear on the left side of foreground objects for left view, and right side for right view, the processing of left view is different from that of right view. They are discussed respectively.

- *Processing left view*

Suppose the destination image is a left view. From (2), we know that parallax-values of foreground pixels are bigger than that of background ones in left view. First of all, we check non-hole points on the right border of the big hole from left to right in **M**, and sharp transitions are detected, as displayed in **Fig. 8**.

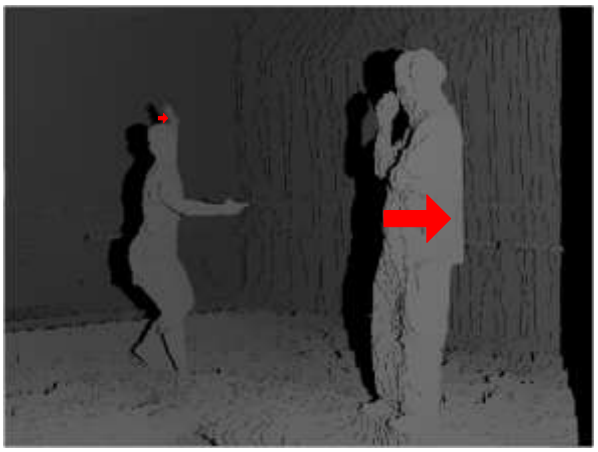


Fig. 8. An example of distinguishing foreground pixels from background pixels according to the disparity map of left view.

Then, we distinguish foreground from background pixels according the sharp transitions. There are three cases: If parallax-values of two neighboring non-hole points change sharply from smaller to larger in the checking process for the first time, then we can tell that background pixels lie at the right border of the hole, and record the larger parallax-value d; else, if parallax-values change from larger to smaller, or no sharp change appears all the time, foreground pixels must be on the right border of the hole, and the parallax-value d of the first non-hole point on the right border of the hole is recorded.

Finally, edges of the hole are dilated. If background pixels lie at the right border of the hole, both sides need to be dilated; else only the left edge is dilated. The hole-dilating operation is as

simple as only changing the parallax-value of the non-hole point to -128 (*hole-point*) in **M**. When dilating is finished, upload the start point and the end point of the hole and go to ***step 3***.

Fig. 9 accounts for the dilation processing of left view. Let non-hole point $\mathbf{I}_l(u + 1, v)$ be the foreground pixel, and non-hole point $\mathbf{I}_l(u - num, v)$ ($num \geq 0$) be the background pixel. There are *num* hole-points between point $\mathbf{I}_l(u + 1, v)$ and point $\mathbf{I}_l(u - num, v)$. Suppose the number of points to be dilated is l, then all the parallax-values of these l points in **M** should be changed to -128. Consequently, the size of the hole is extended to $num + l$. In fact, we only upload the start point and end point. After ***Step 3***, $num + l$ background pixels should be copied to fill the hole from the reference image. As it can be seen from **Fig. 9**, the start and end point are $\mathbf{I}_{ref}(u - num - l - d + 1, v)$ and $\mathbf{I}_{ref}(u - d, v)$ in reference image, respectively, where d is the record parallax-value of point $\mathbf{M}(u + 1, v)$.

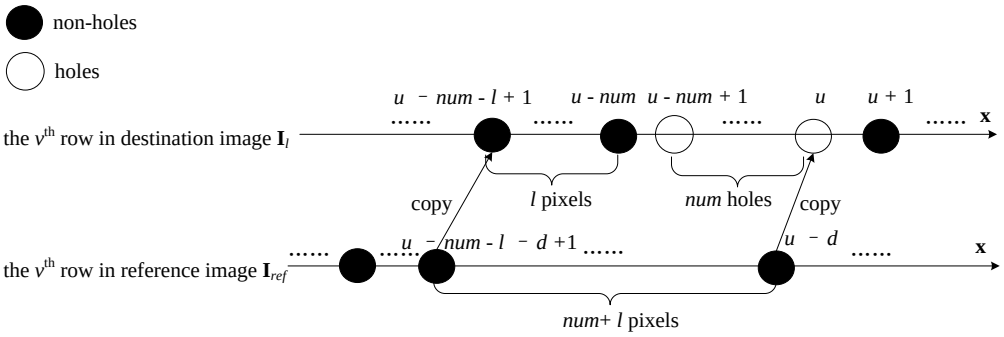


Fig. 9. Illustration of the dilation processing (left view).

➢ *Processing right view*

Similar to the processing of left view, we need to distinguish the foreground pixels from background pixels which lie at left side of the big hole at first. What are different from the processing of left view are the scan order and rules for distinguishing foreground and background pixels.

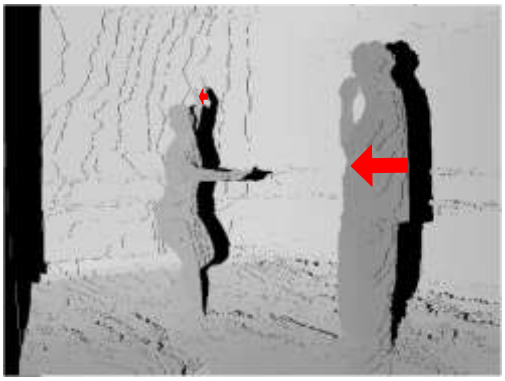


Fig. 10. An example of distinguishing foreground pixels from background pixels according to disparity map of the right view.

We should detect the changes of parallax-value of non-hole points on the left border of the big hole from right to left, as shown in **Fig. 10**. As parallax-values of foreground pixels are smaller than that of background ones in right view (it can be deduced from (2)), when parallax-values of two neighboring non-hole points change sharply from larger to smaller, we can tell that background pixels locate at the left border of the hole, and record the smaller parallax-value d. Else, if parallax-values change from smaller to larger, or no sharp change appears all the time, foreground pixels must be on the left border of the hole, and the parallax-value d of the first non-hole point on the left border of the hole is recorded.

Note that holes lie at edges of destination image are not dilated and the record parallax-value d is set to zero.

Step 3: *filling holes with corresponding background pixels from reference image*

With the start point, the end point of a big hole and the parallax-value d gained from above steps, corresponding background pixels can be copied from reference image to destination image. Examples are shown in **Fig. 4** and **Fig. 9**.

4. Experiments and Discussions

In this paper, "Ballet", "Breakdancer" and other sequences [16] are used for the experiment. Note that "Ballet" and "Breakdancer" sequences captured by Camera 4 (reference images) along with its depth image are used for generating stereo pairs. In order to generate novel views, parameters n, r and D_{zps} in (1) are set to 1, 819.2 and 255, respectively in experiments.

First of all, we generate novel views which contain big holes for our method for testing. **Fig. 11** shows reference images and the generated destination images without pre-processing depth image and hole-filling. Without considering the big holes at the edges of destination images, big holes appear on the left of the foreground object for the novel left-views (**Fig. 11 (b)**, **(e)**, **(h)**, **(k)**), while on the right of the foreground object for right-views (**Fig. 11 (c)**, **(f)**, **(i)**, **(l)**). Note that after 3D image warping, disparity map is also produced.

(a) (b) (c)

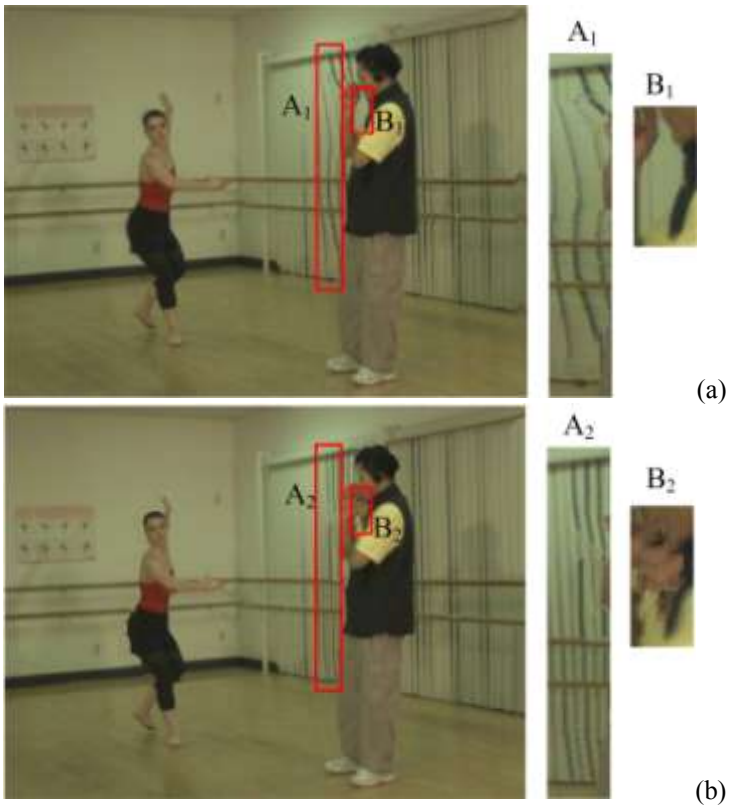
(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

Fig. 11. Reference images and destination images synthesized by reference images and its depth images. From left to right, the images are reference image, novel left view and novel right view, respectively. **(a)**, **(b)** and **(c)** are views for "Ballet" sequence; **(d)**, **(e)** and **(f)** are views for "Breakdancer" sequence; **(g)**, **(h)** and **(i)** are views for "Flower" sequence; **(j)**, **(k)** and **(l)** are views for "Load" sequence.

Then parameter l is set to 3, and we perform the proposed hole-filling method on images shown in **Fig. 11**. Results of "Ballet" sequence can be found in **Fig. 12 (c)** and **(f)**, and matching errors are greatly reduced in contrast to **Fig. 3 (c)**.



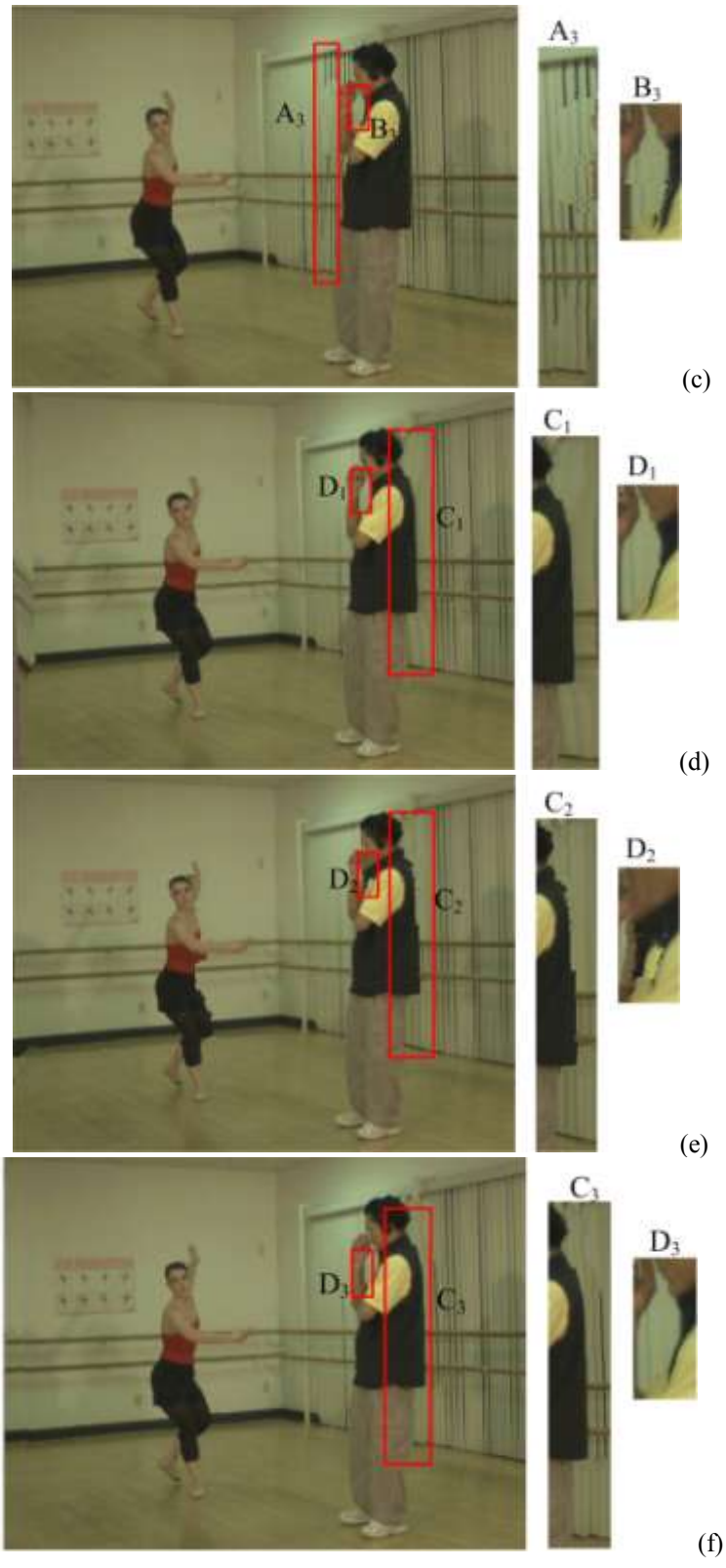
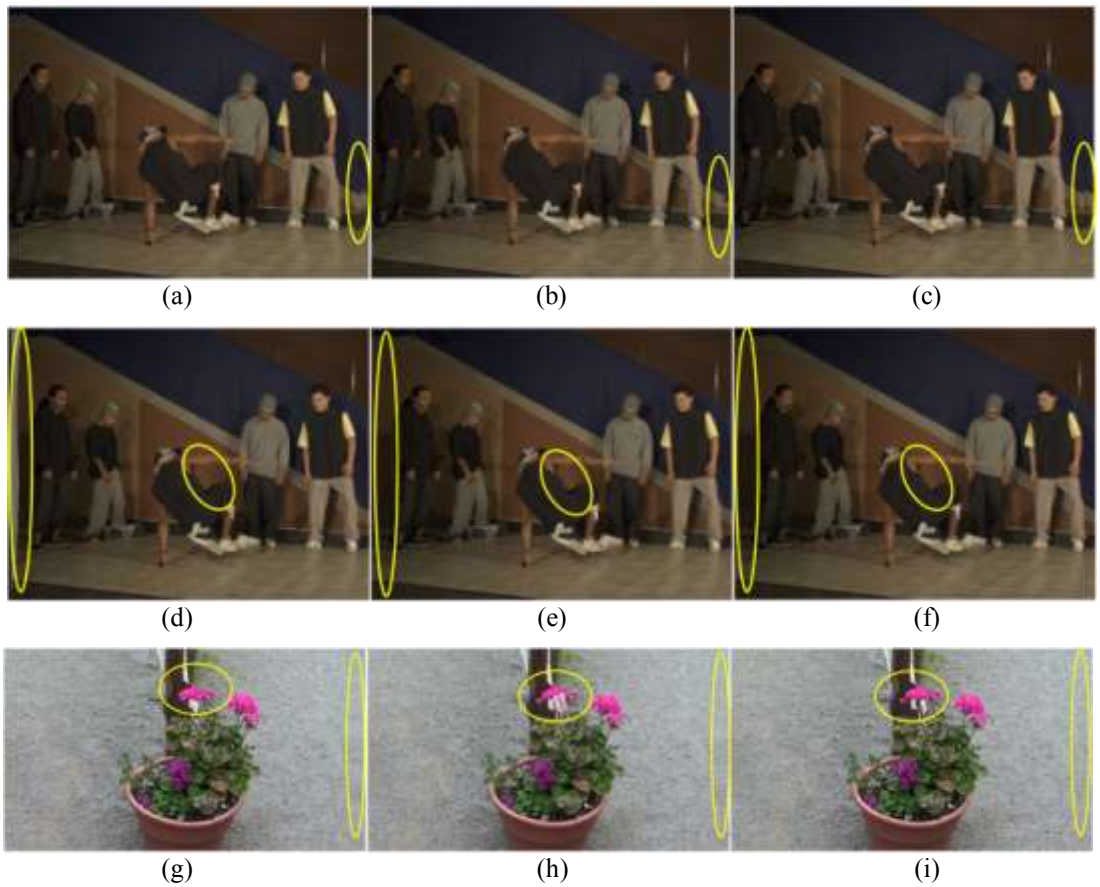

A3
B3
A3
B3
(c)
C1
D1
D1
C1
(d)
C2
D2
D2
C2
(e)
C3
D3
D3
C3
(f)

Fig. 12. Comparison of DGF method, Gautier's inpainting method and the proposed method when the texture of background pixels is abundant. **(a)** and **(d)** are left view and right view generated by DGF method, **(b)** and **(e)** are left view and right view generated by Gautier's inpainting method, **(c)** and **(f)** are left view and right view generated by the proposed method. The mask size of DGF method is 31×11, and the standard deviation σ_v and σ_h are 6 and 2 respectively. The window size of Gautier's inpainting method is 13×11.

Fig. 12 (a) and **(d)** show the images generated by DGF method which smoothes the depth image by applying Directional Gaussian filter guided by edge direction in the hole-flag area iteratively [9]. **Fig. 12 (b)** and **(e)** show the images generated by Gautier's inpainting method [11] which fills holes with depth-based image inpainting method. **Fig. 12 (c)** and **(f)** show the images generated by the proposed method. As it can be seen from **Fig. 12**, these three methods can fill all holes regardless of big or small. With respect to **Fig. 12 (a)** and **(d)**, obvious distortions occur in the background near the filled big holes, as show in A_1 and C_1. In **Fig. 12 (b)** and **(e)**, error textures are found to fill holes with Gautier's inpainting method, especially where both edges of holes are foreground pixels or a little background pixels, this phenomenon can be found in B_2 and D_2. **Fig. 12 (c)** and **(f)** split the difference of image quality generated by DGF method and Gautier's inpainting method. In addition, the proposed method works well in the case of simple or flat textures. Performance comparison of the three methods in the case of simple or flat textures are shown in **Fig. 13**.



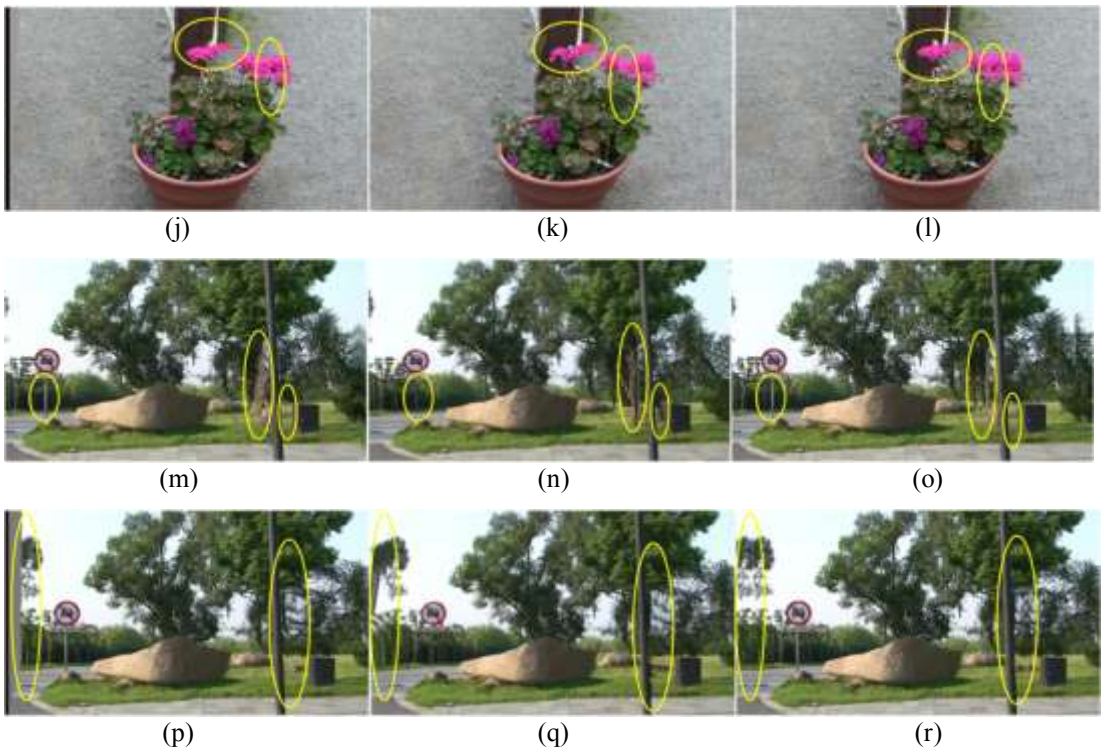
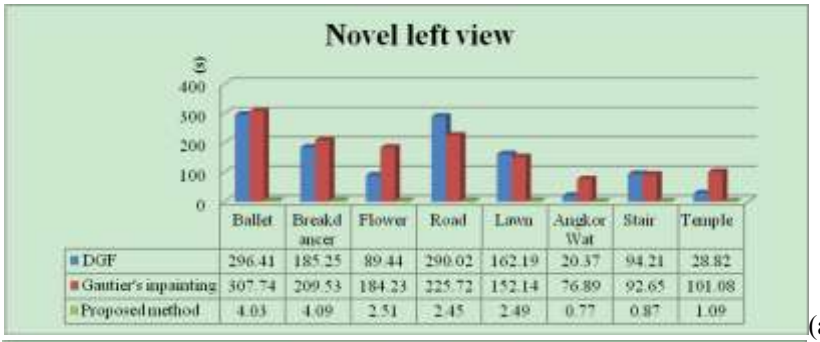


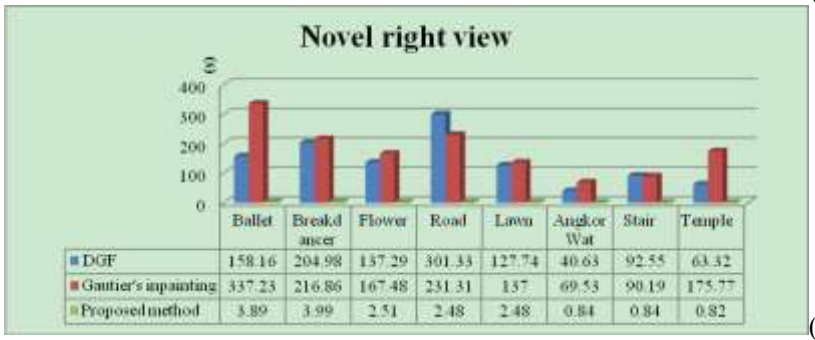
Fig. 13. Performance comparison of three methods in the case of simple or flat textures. Novel views generated by DGF method, Gautier's inpainting method and the proposed method are listed form left to right. **(a)**, **(b)** and **(c)** are novel left views for "Breakdancer" sequence; **(d)**, **(e)** and **(f)** are novel right views for "Breakdancer" sequence; **(g)**, **(h)** and **(i)** are novel left views for "Flower" sequence; **(j)**, **(k)** and **(l)** are novel right views for "Flower" sequence; **(m)**, **(n)** and **(o)** are novel left views for "Road" sequence; **(p)**, **(q)** and **(r)** are novel right views for "Road" sequence.

As can be seen from **Fig. 13**, DGF method and Gautier's inpainting method have their respective advantages and shortcomings. On the whole, the proposed method can gain better 3D image quality. Note that the proposed method is based on shift-sensor camera setup, which is quite different from the other methods. In a DIBR system with shift-sensor camera setup, there are only horizontal parallaxes contained in the generated stereo pair (see equations in (1)). It is not appropriate to measure the image quality obtained with these methods by objective criteria such as PSNR and SSIM. Hence subjective evaluations are performed above.

Furthermore, in DGF method the number of iteration T = max (width of holes) / min (σ_v, σ_h) [9], which indicates that the lager the hole is, the stronger the smoothing is or the more iteration times are needed, and it takes much more time to generate a novel view. As for Gautier's inpainting method, it needs much time to find matching patch, and the *running time* has much to do with the size of patch and holes. On the contrary, the *time complexity* of the proposed method has nothing to do with the size of holes, which results in less running time especially in the case of large holes. A comparison of time complexity between these three methods can be found in **Fig. 14**.



(a)



(b)

Fig. 14 Comparison of running time of DGF method, Gautier's inpainting method and the proposed method. **(a)** Running time when generating left views. **(b)** Running time when generating right views. Note that "running time" here indicates the average time of generating a novel view, and the first 10 frames of each sequence are used here to synthesize novel views for speed evaluation.

5. Conclusion

This paper presents a hole-filling method based on disparity map. Hole-filling is applied after 3D image warping in this method, and it required only one video stream with its depth images. One important aspect of the method is that the disparity map converted from depth image is used during the hole-filling. With disparity map, foreground pixels and background pixels are distinguished; and matching errors are obviously reduced or completely removed by dilating edges of big holes. Furthermore, holes are filled by copying corresponding pixels from reference image instead of neighboring pixels in destination image. In comparison with depth-smoothing method and inpainting method, the proposed method is simpler and much easier to implement on hardware. In addition, as depth-smoothing is avoided in this method, it ensures the "authenticity" of non-hole area of destination image; and geometric distortions are reduced or disappear in destination image. Experimental results show that the proposed method can achieve good 3D image quality with lower time complexity compared to previous work.

References

[1] A. Smolic, K. Mueller, P. Merkle, A. Vetro, "Development of a new MPEG standard for advanced 3D video applications," in *6th International Symposium on Image and Signal Processing and Analysi*, pp. 400-407, Sep 2009. Article (CrossRef Link).

[2] R. Liu, Q. Zhu, X. Xu, L. Zhi, H. Xie, J. Yang, X. Zhang, "Stereo effect of image converted from planar,".*Information Sciences*, vol. 178, no. 8, pp. 2079-2090, Apr 2008. Article (CrossRef Link).

[3] L. Pei-Jun, Effendi, "Nongeometric Distortion Smoothing Approach for Depth Map Preprocessing," *IEEE Transactions on Multimedia*, vol. 13, no. 2, pp. 246-254, Apr 2011. Article (CrossRef Link).
[4] X. Yang, J. Liu, J. Sun, X. Li, W. Liu, Y. Gao, "DIBR based view synthesis for free-viewpoint television," in *5th 3DTV Conference*, pp. 1-4, May 2011. Article (CrossRef Link).
[5] L.-H. Wang, X.-J. Huang, M Xi, D-X Li, M Zhang, "An asymmetric edge adaptive filter for depth generation and hole filling in 3DTV*," IEEE Transactions on Broadcasting*, vol. 56, no. 3, pp. 425-431, Sep 2010. Article (CrossRef Link).
[6] P.-J. Lee, Effendi, "Adaptive edge-oriented depth image smoothing approach for depth image based rendering," in *2010 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting,* pp. 1-5, Mar 2010. Article (CrossRef Link).
[7] I. Daribo, B. Pesquet-Popescu, "Depth-aided image inpainting for novel view synthesis," in *2010 IEEE International Workshop on Multimedia Signal Processing (MMSP)*, pp. 167-170, Oct 2010. Article (CrossRef Link).
[8] L. Azzari, F. Battisti, A. Gotchev, M. Carli, K. Egiazarian, "A modified non-local mean inpainting technique for occlusion filling in depth-image-based rendering," in *Stereoscopic Displays and Applications XXII*, Jan 2011. Article (CrossRef Link).
[9] H. Ying-Rung, T. Yu-Cheng, C. Tian-Sheuan, "Stereoscopic images generation with directional Gaussian filter," in *Proceedings of 2010 IEEE International Symposium on Circuits and Systems* (*ISCAS*), pp. 2650-2653, May 30-June 2, 2010. Article (CrossRef Link).
[10] S. Zinger, L. Do, P. H. N. De With, "Free-viewpoint depth image based rendering,". *Journal of Visual Communication and Image Representation*, vol. 21, no. 5-6, pp. 533-541, Jul 2010. Article (CrossRef Link).
[11] J. Gautier, O. Le Meur, C. Guillemot, "Depth-based image completion for view synthesis," in *5th 3DTV Conferenc* , pp. 1-4, May 2011. Article (CrossRef Link).
[12] C. Fehn, "Depth-image-based rendering (DIBR), compression and transmission for a new approach on 3D-TV," in *Stereoscopic Displays and Virtual Reality Systems XI*, pp. 93-104, Jan 2004. Article (CrossRef Link).
[13] R. Liu, Y.-C. Tan, F.-C. Tian, H Xie, G.-Q. Tai, W.-M. Tan, J.-L. Liu, X.-Y. Xu, C. Kadri, N. Abakah, "Visual Fatigue Reduction Based on Depth Adjustment for DIBR System," *Ksii Transactions on Internet and Information Systems*, vol. 6, no. 4, pp. 1171-1187, 2012.
Article (CrossRef Link).
[14] R. Liu, H. Xie, G.-Q. Tai, Y.-C. Tan, R.-L. Guo, W.-Y. Luo, X.-Y. Xu, J.-L. Liu, "Depth adjustment for depth-image-based rendering in 3D TV system,". *Journal of Information and Computational Science*, vol. 8, no. 16, pp. 4233-4240, 2011. Article (CrossRef Link).
[15] L Zhang, G.-Q. Tai, R. Liu, H. Xie, X.-Y. Xu, "Correction algorithm of matching error for DIBR.," *Huanan Ligong Daxue Xuebao/Journal of South China University of Technology* (*Natural Science*), vol. 39, no. 12, pp. 51-55, 2011. Article (CrossRef Link).
[16] G. Zhang, J. Jia, T.-T. Wong, H. Bao, "Consistent depth maps recovery from a video sequence," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 6, pp. 974-988,2009. Article (CrossRef Link).

Ran Liu received the B.E., M.E., and D.E. degrees in Computer Science from Chongqing University, Chongqing, China, in 2001, 2004, and 2007, respectively. He worked as post doctoral researcher in Homwee Technology Co., Ltd, Chengdu, China from 2008 to 2010. He is now a senior engineer in Homwee Technology Co., Ltd. He is also a teacher in the College of Communication Engineering and the College of Computer Science, Chongqing University, China. His research interests include 3D TV, virtual reality and computer vision.

Hui Xie received her Bachelor's degree at the College of Physics, Hunan University of Science and Technology, Hunan, China, in 2010. Now, she is pursuing her Master's degree at the College of Communication Engineering, Chongqing University, Chongqing, China. Her research interests are 3D TV and image processing.

Fengchun Tian received the B.E., M.E., and D.E. degrees in radio engineering, biomedical instruments and engineering, theoretical electric engineering from Chongqing University, Chongqing, P.R. China, in 1984, 1986, and 1996, respectively. Since 1984, he has been working in Chongqing University as a teacher. Since 2007, he is also an adjunct professor in the University of Guelph, Canada. His current research interests are image processing (including optical image processing and video), biomedical and bioinformatics and modern signal processing technology.

Yingjian Wu received the Master's degree and Doctor's degrees in physics from Beijing University and Institute of Physics Chinese Academy of Sciences in 1984 and 1988, respectively. Since 1984, he has been working in Xi'an Jiaotong University as a teacher. Since 2005, he is a senior engineer in Changhong Group. His current research interests are image processing and information technology.

Guoqin Tai is currently pursuing his Master's degree at the College of Communication Engineering, Chongqing University, Chongqing, China. He received the Bachelor's degree at the College of Information Engineering, Qingdao Agriculture University, Shandong, China, in 2009. His research interests include 3D TV, virtual reality and computer vision.

Yingchun Tan is currently pursuing her Master's degree in Communication Engineering, Chongqing University, Chongqing, China. She got her Bachelor's degree in Communication Engineering from Chongqing Uniersity, Chongqing, China, in 2008. Her research interests include 3D TV, digital signal processing.

Weimin Tan is currently pursuing his Master's degree at the College of Communication Engineering, Chongqing University, Chongqing, China. He received the Bachelor's degree at the College of Physic and Electronic, Hunan Institute of Science and Technology, Hunan, China, in 2008. His research interests include 3D TV, image processing.

Bole Li is currently pursuing his Master's degree at the College of Communication Engineering, Chongqing University, Chongqing China. He received the Bachelor's degree at the College of Electronic Information, Hangzhou Dianzi University, Zhejiang, China, in 2012. His research interests include 3D TV, image processing.

Hengxin Chen received his M.E. and D.E. degrees in Computer Science from Chongqing University, Chongqing, China, in 2005 and 2010, respectively. He is currently working at the college of computer science, Chongqing University. His research interests include pattern recognition, image processing, software engineering, and etc.

Liang Ge received his M.E. and D.E. degrees in Computer Science from Chongqing University, Chongqing, China, in 2004 and 2009, respectively. He is currently working at the college of computer science, Chongqing University. His research interests include image registration, image stereo matching, and three-dimensional modeling.