Simultaneous intermediate-view interpolation and multiplexing algorithm for a fast lenticular display

HanShin Lim
Seok-Hoon Kim
Yun-Gu Lee
Lee-Sup Kim
Jong Beom Ra, MEMBER SPIE
HyunWook Park
Korea Advanced Institute of Science and Technology (KAIST)
Department of Electrical Engineering
373-1 Guseong-dong, Yuseong-gu
Daejeon 305-701, Korea
E-mail: hwpark@ee.kaist.ac.kr

Abstract. Lenticular displays are a promising form of autostereoscopic technology and several products have recently been commercialized. For lenticular displays, typically several views of a scene are acquired from different viewpoints, or are generated by using depth information. If the depth information of a scene is known, then it is possible to easily generate several intermediate views simply by using the stereo image pair and depth information. The paper presents a simple method to correct the lenticular alignment error by compensating the correction coefficients to the viewpoint determination formula. For the realization of a fast lenticular display and removal of image distortion and artifacts, the proposed algorithm simultaneously performs intermediate floating-pointview interpolation and multiplexing on the scanline using the left-view and right-view images and depth information. Experimental results show that lenticular images having considerably reduced distortion and artifacts are generated by using the proposed algorithm. [DOI: 10.1117/1.2802511]

Subject terms: 3D display; computer graphics; lenticular display; multiplexed image; view interpolation.

Paper 070124R received Feb. 10, 2007; revised manuscript received May 31, 2007; accepted for publication Jun. 4, 2007; published online Nov. 29, 2007.

1 Introduction

Recently, a considerable amount of research has been directed toward the development of 3-D displays for augmenting reality and 3-D effects.[1,2] 3-D displays can be categorized into stereoscopic displays and autostereoscopic displays according to whether a viewer requires auxiliary devices or not. In stereoscopic displays, the viewer sees a 3-D image by the aid of some devices, e.g., polarized or time-multiplexing glasses. On the other hand, in the case of autostereoscopic displays, the viewer can see 3-D images without glasses or other devices. Since autostereoscopic displays do not require any auxiliary devices, they are the preferred candidate for future applications such as 3-D TV.[3]

Recently, the lenticular display has become the representative technology among the commercially introduced autostereoscopic displays. In lenticular displays, lenticules are placed on top of a liquid crystal display (LCD) in such a way that the different positions of the left and right eyes provide different images for each eye such that a 3-D effect can be obtained. Since multiple-view images should be stored in the LCD behind the lenticules, the image resolution of each view is reduced by a factor of the number of views.[4,5] A slanted lenticular display can reduce the Moiré patterns and balance the resolution in vertical and horizontal directions.[5] Figure 1 shows an example of the multiplexing process of a nine-view lenticular display. Each red, green, blue (RGB) position of a pixel in the image has its own viewpoint for the lenticular display. For example, if the viewpoint of the red component on a pixel in the multiplexed image is three, then only the red color of the pixel at the same coordinates in the third-view image should be assigned. If the viewpoint determination formula provides the wrong view number for each pixel, then the color values in pixels having misassigned view numbers can lead to image distortions such as boundary dislocations.[6]

The main goal of this paper is to propose a simultaneous multiplexing and view interpolation method based on the use of a stereo image pair and its depth image for a fast multiview lenticular display. Before introduction of the proposed method, the paper presents a simple lenticular misalignment correction method that involves compensation of some correction coefficients to the viewpoint determination formula. The proposed simultaneous multiplexing and view

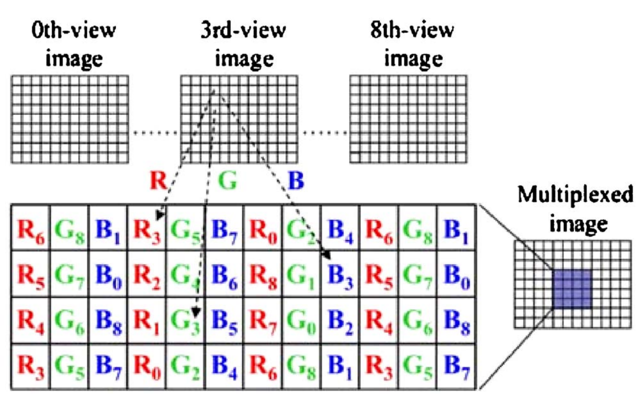

Fig. 1 Multiplexing process of a nine-view lenticular display.

0091-3286/2007/$25.00 © 2007 SPIE

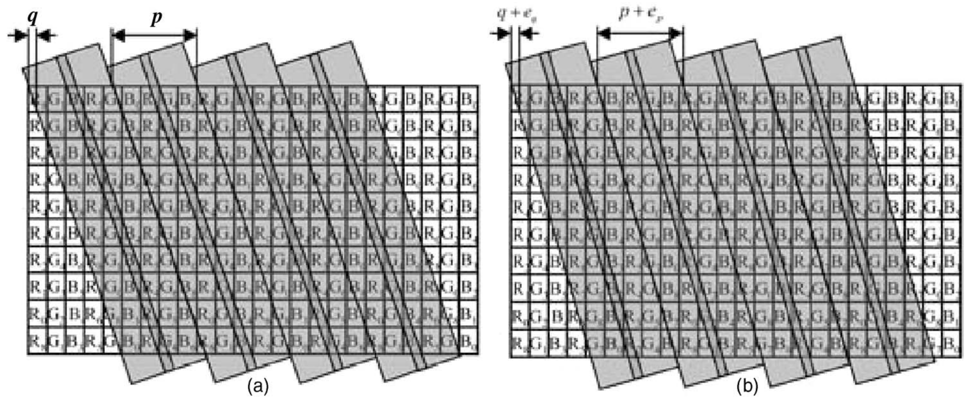

Fig. 2 Alignment of the lenticule: (a) in an ideal case; and (b) with rotational alignment error.[6]

interpolation method can support floating-point viewpoints that are obtained from the corrected viewpoint determination formula.

The paper is organized as follows. In Sec. 2, we introduce a simple lenticular misalignment correction method using the floating-point viewpoint determination formula. In Sec. 3, we describe a proposed multiplexing and view interpolation method for a fast multiview lenticular display where each subpixel has an integer viewpoint. In Sec. 4, the proposed multiplexing and view interpolation method is adopted to subpixels that have floating-point viewpoints generated from the corrected viewpoint determination formula. Section 5 presents experimental results, and we conclude the paper in Sec. 6.

2 Alignment Error Correction Using the Compensation Coefficients

The integer viewpoint determination formula for each pixel (u,v) in the multiplexed image, which can be used to assign the appropriate color value to the pixel, is described as follows:

$$n(u,v) = \left\lfloor \left(\frac{(u + u_{\text{off}} - 3v \tan \phi) \bmod N_u}{N_u} \right) \times N_{\text{tot}} + 0.5 \right\rfloor, \quad (1)$$

where $n(u,v)$ is the integer viewpoint corresponding to the pixel (u,v) in the multiplexed image; N_u is the number of views per lens measured along a single row of the LCD; N_{tot} is the total number of views; ϕ is the angle between vertical lines of the LCD and lenticule; $\lfloor x \rfloor$ is the largest integer value less than or equal to x; and the parameter u_{off} is the offset of an arbitrary horizontal shift of the lenticular lens array with respect to the LCD.[5]

Equation (1) can be rewritten to reduce the computation burden as follows:

$$n(u,v) = (au - bv + c) \bmod N_{\text{tot}}, \quad (2)$$

where $a = N_{\text{tot}}/N_u$, $b = 3N_{\text{tot}}/N_u \tan \phi$ and $c = N_{\text{tot}}/N_u x_{\text{off}}$, all of which can be precomputed, and $x \bmod y$ means "x modulo y." In general nine-view lenticular displays, the lenticule is aligned as depicted in Fig. 2(a). In this case, $a=2$ and $b=1$. However, if the lenticule is misaligned in terms of translation and rotation, the original viewpoint determination formula cannot assign the correct viewpoint to the subpixels.

These lenticular misalignment errors can be reduced by adding correction terms to Eq. (2). Equation (2) then becomes

$$n_c(u,v) = \{(a + a_{\text{cor}})u - (b + b_{\text{cor}})v + c + c_{\text{cor}}\} - \left\lfloor \frac{(a + a_{\text{cor}})u - (b + b_{\text{cor}})v + c + c_{\text{cor}}}{N_{\text{tot}}} \right\rfloor \times N_{\text{tot}}, \quad (3)$$

where a_{cor}, b_{cor}, and c_{cor} are the floating-point coefficients for correction of the lenticular alignment error.

Let p and q be the lenticular pitch and horizontal shift of a lenticule between successive rows, respectively, and e_p and e_q errors of p and q, respectively. In general nine-view lenticular displays, $p=4.5$ and $q=0.5$. In addition, let N_{ccv} and N_{cch} be the number of color changes along the horizontal and vertical directions, respectively, when we assign $R=255$ to the zeroth, third, and sixth views, $G=255$ to the first, fourth, and seventh views, $B=255$ to the second, fifth, and eighth views, and all the others set to zero. $|e_p|$ and $|e_q|$ can then be calculated as follows:[6]

$$|e_p| \approx \frac{p^2 N_{\text{cch}}}{3 N_{\text{tot}} N_w},$$

$$|e_q| \approx \frac{p N_{\text{ccv}}}{N_{\text{tot}} N_H}, \quad (4)$$

where N_H and N_w are the vertical and horizontal resolution of the LCD panel, respectively. The corrected coefficients $a' = a + a_{\text{cor}}$ and $b' = b + b_{\text{cor}}$ can then be acquired by the following simple calculations:

$$a' = a + a_{\text{cor}} = \frac{N_{\text{tot}}}{p + e_p}, \quad (5)$$

$$b' = b + b_{\text{cor}} = \frac{q + e_q}{q}. \quad (6)$$

3 The Proposed Intermediate View and Multiplexing Algorithm for a Lenticular Display

3.1 *Overall Procedure of the Method*

In this paper, we assume the lenticular display has nine horizontal viewing regions ($N_{\text{tot}}=9$). The proposed algorithm uses the left-most and right-most view images and depth information generated from computer graphics, or stereo images of which depth information is computed from the stereo images or acquired from particular devices. Several intermediate-view images are generated from the left-most and right-most view images by use of the view interpolation method.[7] The correspondences of the stereo image pairs are assumed to be on the scanline. The size of each view image is one-third for the vertical line and one-third for the horizontal line of the multiplexed image, where the multiplexed image has the same resolution as the LCD. We only deal with one subpixel component among the R, G, and B components, since the other two components can be processed in the same manner.

Figure 3 shows the overall procedure of the proposed algorithm. The mapping relation between pixels of the multiview images and the multiplexed image is calculated by using the depth information and the viewpoint determination formula. Color values of the corresponding left-view and right-view pixels are sequentially interpolated and mapped to the locations of the multiplexed image. A method to remove holes, which are generated mainly due to occlusion and the boundary of the image, is also processed on the scanline.

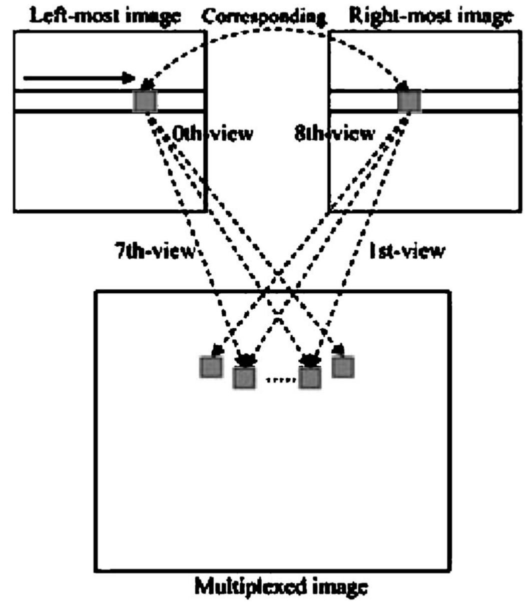

Fig. 3 Generation of a multiplexed image from the left-most and right-most images.

3.2 *Simultaneous View and Multiplexing*

We assume that each viewing region is ordered from zero, the left-most viewing region, to eight, the right-most viewing region. Let (u,v) be the multiplexed image coordinates and $n(u,v)$ an integer view point of (u,v) computed from Eq. (2). Let us define $(u',v')_n$ as a multiplexed image pixel that has the viewpoint of n among 3×3 adjacent pixels centered at (u',v'), as shown in Fig. 4, such that

$$(u',v') = \left(3\left(x - \left\lfloor \frac{n}{8}d(x,y) + 0.5\right\rfloor\right), 3y\right), \tag{7}$$

where (x,y) are the left-most view image coordinates and $d(x,y)$ the disparity value of (x,y) with a viewpoint of n.

Our purpose is to map the pixel from (x,y) in the n'th view image to (u,v) in the multiplexed image without generation of the view interpolated images. Let $I_n(x,y)$ be the color value of (x,y) in the n'th view image. $I_n(x-(n/8)d(x,y),y)$ is then calculated as follows:

$$I_n\left(x - \frac{n}{8}d(x,y), y\right) = \begin{cases} \frac{8-n}{8}I_l(x,y) + \frac{n}{8}I_r(x-d(x,y),y) & \text{if}\left|I_l(x,y) - I_r(x-d(x,y),y)\right| \leqslant \text{thresh} \\ I_l(x,y) & \text{otherwise} \end{cases}, \tag{8}$$

where I_l and I_r are the left-most and right-most view images, respectively. Finally, $I_n(x-(n/8)d(x,y),y)$ is mapped to $I(u',v')_n$, which denotes the color value of the multiplexed image pixel that has the viewpoint of n among 3×3 adjacent pixels centered at (u',v').

3.3 *Hole Filling*

Holes are usually generated near object boundaries, whose adjacent pixels have disparity differences. It is well known that handling of holes is an important process for obtaining high-quality multiview images.[8,9] To find occurrences of holes, we define the difference between the distances from the left-most view image to the n'th view image of $(x-1,y)$ and (x,y) as follows:

$$D_n(x,y) = \left\lfloor \frac{n}{8}d(x-1,y) + 0.5\right\rfloor - \left\lfloor \frac{n}{8}d(x,y) + 0.5\right\rfloor. \tag{9}$$

If $D_n(x,y)$ is negative, the disparity value of the previous pixel is shorter than that of the current pixel, thereby overlapping the pixels. The position mapped from the previous pixel is replaced by the current pixel. Otherwise, if $D_n(x,y)$ is positive, we can predict that holes can occur at

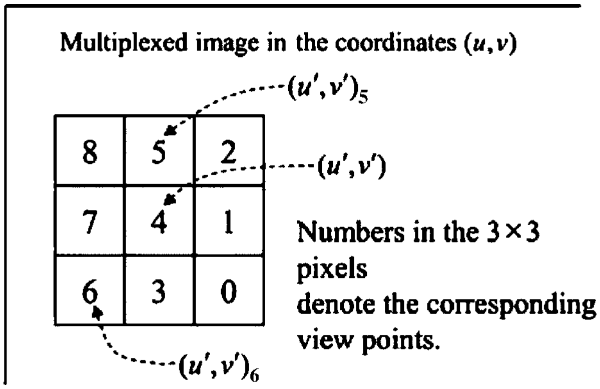

Fig. 4 An example of mapping from (x,y) to (u,v) for the integer viewpoint.

pixels from $(x-[n/8d(x-1,y)+0.5],y)$ to $(x-[n/8d(x,y)+0.5],y)$ in the n'th view image. The length of the hole in the n'th view image becomes $D_n(x,y)$. To remove the holes in the n'th view image, we assume that the pixels from $(x-D_n(x,y),y)$ to $(x-1,y)$ in the left-most view image have the same disparity value of $d(x,y)$. We then reperform the simultaneous view interpolation and multiplexing method described in Sec. 3.2 to the pixels from $(x-D_n(x,y),y)$ to $(x-1,y)$ in the left-most view image for filling the holes in the n'th view image. In this case, the holes are filled according to the following formula:

(i) If $|I_l(x,y)-I_r(x-d(x,y),y)|\leq$ thresh [from Eq. (8)]

$$I_n\left(x-\left\lfloor\frac{n}{8}d(x,y)+0.5\right\rfloor-k,y\right)=I_r(x-d(x,y)-k,y),\quad 1\leq k\leq D_n(x,y), \tag{10}$$

(ii) If $|I_l(x,y)-I_r(x-d(x,y),y)|>$ thresh

$$\begin{cases} I_n\left(x-\left\lfloor\frac{n}{8}d(x,y)+0.5\right\rfloor-k,y\right)=I_r(x-d(x,y)-k,y) & \text{if}\left|I_r(x-d(x,y)-k,y)-I_r(x-d(x,y),y)\right|\geq \text{thresh} \\ I_n\left(x-\left\lfloor\frac{n}{8}d(x,y)+0.5\right\rfloor-k,y\right)=I_l(x,y) & \text{otherwise}. \end{cases} \tag{11}$$

Finally, $I_n(x-[n/8d(x,y)+0.5]-k,y)$ is mapped to $I(u_h',v_h')_n$, where (u_h',v_h') is defined as follows:

$$(u_h',v_h')=\left(3\left(x-\left\lfloor\frac{n}{8}d(x,y)+0.5\right\rfloor-k\right),3y\right). \tag{12}$$

3.4 *Subpixel Interpolation*

The purpose of interpolation is to reduce degradation of 3-D image quality caused by discord between the pixel positions of the multiview images and the multiplexed image. If the corresponding point is not located in the correct position in the multiplexed image, it can be a major source of image distortion and degradation. Therefore, we reduce the inconsistency of pixel values in the n'th view image by applying an interpolation method. Figure 5 illustrates the general concept of the subpixel interpolation process.

4 The Proposed Floating-Point View Interpolation and Multiplexing Method

As described in Sec. 2, the lenticular misalignment error can be corrected by adding the correction coefficients to the viewpoint determination formula. However, if the formula is applied to the LCD subpixels, then each subpixel will not have an integer view number but rather a floating-point view number. If we round off the floating-point viewpoint to the nearest integer value, then the simultaneous view interpolation and multiplexing algorithm described in Sec. 3 can be directly applied. In this case, however, a slanted line pattern is generated in the 3-D lenticular image due to the abrupt change of viewpoints between the neighboring subpixels. Figure 6 is a picture of a lenticular image when the green subpixels of the rounded-off fourth-view image

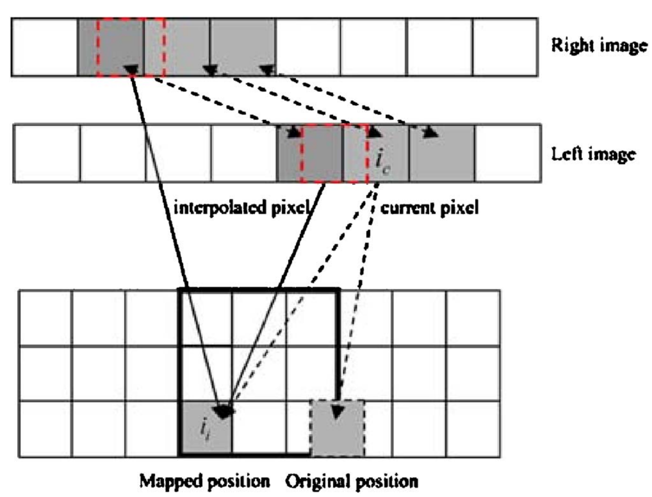

Fig. 5 General concept of the subpixel interpolation process.

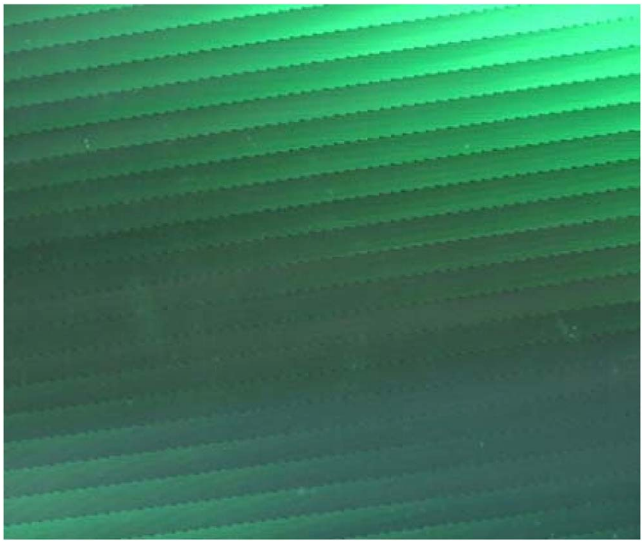

Fig. 6 Occurrence of slanted line pattern artifact due to application of Eq. (3) and integer view mapping.

were set to 255 and the others were set to zero. The image shows a slanted line pattern artifact. To remove the artifact and assign more accurate pixel values to the multiplexed image, the simultaneous view interpolation and multiplexing algorithm described in Sec. 3 are modified to assign correct pixel values to the subpixels that have floating-point viewpoints, as shown in Fig. 7. The floating-point view interpolation and multiplexing method is comprised of two steps, integer approximation and refinement, as follows.

- *Integer approximattion step*: Let $n_c(u,v)$ be the corrected viewpoint of (u,v) computed from Eq. (3) and define $(u',v')_{n_c}$ as the multiplexed image pixel that has the viewpoint of n_c such that $|n_c - n| < 0.5$, among 3×3 adjacent pixels centered at (u',v'), as shown in Fig. 8, such that

$$(u',v') \equiv \left(3\left(x - \left\lfloor \frac{n+0.5}{9} d(x,y) + 0.5 \right\rfloor\right), 3y\right). \tag{13}$$

First, (x,y) in the left-most view image is mapped to $(u',v')_{n_c}$. However, there are some errors such that $n \neq n_c$ and the actual mapping location of (x,y) may not be equal to $(u',v')_{n_c}$, i.e.,

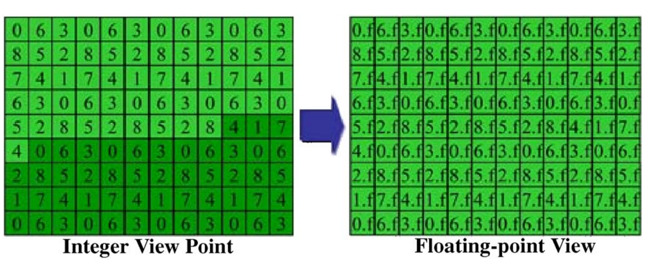

Fig. 7 Conversion from integer to floating-point viewpoint.

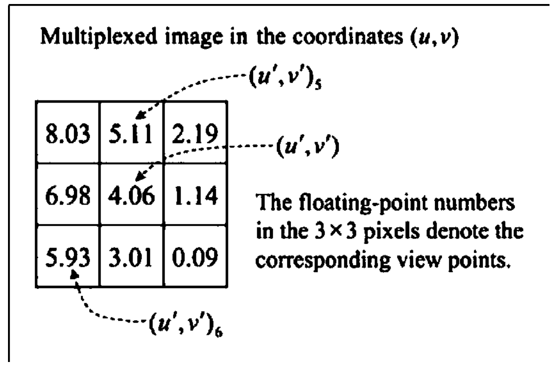

Fig. 8 An example of mapping from (x,y) to (u,v) for the floating-point viewpoint.

$$(u',v')_{n_c} \neq \left(3\left(x - \frac{n+0.5}{9} d(x,y)\right), 3y\right). \tag{14}$$

- *Refinement step*: In order to correct the above errors, we estimate the pixel point of (x_{n_c},y), which is the correct mapping position when assuming $d(x,y) = d(x_{n_c},y)$, as follows:

$$x_{n_c} = \frac{u'_{n_c}}{3} + \frac{n_c}{9} d(x,y), \tag{15}$$

where u'_{n_c} is the u-coordinate value of $(u',v')_{n_c}$.

However, the actual $d(x_{n_c},y)$ may not be equal to $d(x,y)$. If $|d(x,y) - d(x_{n_c},y)| \leqslant 1$, the corresponding point of $(u',v')_{n_c}$ should then be adjusted to $(\tilde{x}_{n_c},y)$ with the assumption of linear interpolation of the disparity between $d(x,y)$ and $d(x_{n_c},y)$, which satisfies the following two equations:

$$d(\tilde{x}_{n_c},y) = \frac{(\tilde{x}_{n_c} - x)d(x_{n_c},y) + (x_{n_c} - \tilde{x}_{n_c})d(x,y)}{x_{n_c} - x}, \tag{16}$$

$$\tilde{x}_{n_c} - \frac{n_c}{9} d(\tilde{x}_{n_c},y) = \frac{u'_{n_c}}{3}. \tag{17}$$

From Eqs. (16) and (17), $\tilde{x}_{n_c}$ can be derived as follows:

$$\tilde{x}_{n_c} = x + \frac{\frac{u'_{n_c}}{3} - x + \frac{n_c}{9} d(x,y)}{\left\{1 - \frac{n_c}{9(x_{n_c} - x)}(d(x_{n_c},y) - d(x,y))\right\}}. \tag{18}$$

Otherwise, if $|d(x,y) - d(x_{n_c},y)| > 1$, we assume $d(x_{n_c},y) = d(\tilde{x}_{n_c},y)$ and estimate $\tilde{x}_{n_c}$ as follows:

$$\tilde{x}_{n_c} = \frac{u'_{n_c}}{3} + \frac{n_c}{9} d(x_{n_c},y). \tag{19}$$

Let $I_{n_c}(x,y)$ be the color value of (x,y) in the n_c'th view image. $I_{n_c}(\tilde{x}_{n_c} - (n_c/9)d(\tilde{x}_{n_c},y),y)$ is then mapped to $I(u',v')_n$, which denotes the intensity of the multiplexed image pixel that has the viewpoint of n_c such that $|n_c-n|<0.5$, among 3×3 adjacent pixels centered at (u',v'). If $|d(x,y)-d(x_{n_c},y)| \leqslant 1$, then $I_{n_c}(\tilde{x}_{n_c}-(n_c/9)d(\tilde{x}_{n_c},y),y)$ is calculated as follows:

$$I_{n_c}\left(\tilde{x}_{n_c}-\frac{n_c}{9}d(\tilde{x}_{n_c},y),y\right)=\begin{cases}\frac{9-n_c}{9}I_l(\tilde{x}_{n_c},y)+\frac{n_c}{9}I_r(\tilde{x}_{n_c}-d(\tilde{x}_{n_c},y),y) & \text{if}|I_l(\tilde{x}_{n_c},y)-I_r(\tilde{x}_{n_c}-d(\tilde{x}_{n_c},y),y)|\leqslant \text{thresh}\\ I_l(\tilde{x}_{n_c},y) & \text{otherwise.}\end{cases} \quad (20)$$

Otherwise, if $|d(x,y)-d(x_{n_c},y)|>1$, $I(u',v')_n$ is determined in accordance with the prediction of overlaps or holes between $d(x,y)$ and $d(x_{n_c},y)$ as follows:

(i) If $(d(x,y)-d(x_{n_c},y)<-1$ and $x<x_{n_c})$ or $(d(x,y)-d(x_{n_c},y)>1$ and $x>x_{n_c})$, then overlaps can occur in the region. In this case, $I(u',v')_n$ is mapped using Eq. (20).

(ii) If $(d(x,y)-d(x_{n_c},y)>1$ and $x<x_{n_c})$ or $(d(x,y)-d(x_{n_c},y)<-1$ and $x>x_{n_c})$, then holes can occur in the region. If $d(x,y)-d(x_{n_c},y)>1$ and $x<x_{n_c}$, $I_r(\tilde{x}_{n_c}-d(x_{n_c},y),y)$ is mapped to $I(u',v')_n$. Otherwise, if $d(x,y)-d(x_{n_c},y)<-1$ and $x>x_{n_c}$ and $|d(\tilde{x}_{n_c},y)-d(x_{n_c},y)|<1$, then the assigned pixel value $I(u',v')_n$ in the multiplexed image is calculated from Eq. (20).

Also, holes can occur when

$$D_n(x,y)=\frac{n+0.5}{9}d(x-1,y)-\frac{n+0.5}{9}d(x,y) \quad (21)$$

is positive. In this case, the holes can be filled in the same manner as described in Sec. 3.3 by replacing the reperforming process with the method in Sec. 4.

5 Experimental Results

The proposed method was applied to stereo video sequences with depth information generated by computer graphics. The display device was a nine-view lenticular display system manufactured by Samsung SDI. A PC having Pentium-IV 2.6 GHz and 1 GB RAM was used in the experiment. We implemented the proposed algorithm using C/C++ and generated the test sequences by computer graphics based on OpenGL graphics API.[10,11] Figure 9 shows four selected images from the test sequence. In the experiments, the viewpoint of a pixel was determined by Eq. (2) for the original case and Eq. (3) for the corrected case, and the lenticular parameters were calculated using Eqs. (5) and (6), whose values were $a=1.999$ and $b=1.02$. We set $c=11\,520$ and thresh=15. The resolution of the multiplexed image is 1280×1024, which corresponds to the resolution of the display system.

Figure 10(a) is a picture of a test lenticular image generated from the integer viewpoint determination formula.[2] As shown in the circled regions, the object boundaries are dislocated due to incorrect assignment of viewpoints for each subpixel by using the integer viewpoint determination formula, in which lenticular misalignment errors were not corrected. In Fig. 10(b), the image was generated from the corrected viewpoint determination formula with an integer viewpoint that was rounded off from the floating-point viewpoints. In the picture, the dislocations are considerably reduced and the slanted line pattern artifact is clearly observed. Finally, in Fig. 10(c), the image was generated from the corrected viewpoint determination formula with the floating-point viewpoint. In the image, the dislocations are considerably reduced and the slanted line pattern artifact has been removed.

Finally, we verified that the simplified version of the proposed algorithm is suitable for implementation in a 3-D graphics processor for fast generation of lenticular images.[12]

6 Conclusion

In this paper, we proposed a simple correction method of a viewpoint determination formula for lenticular misalign-

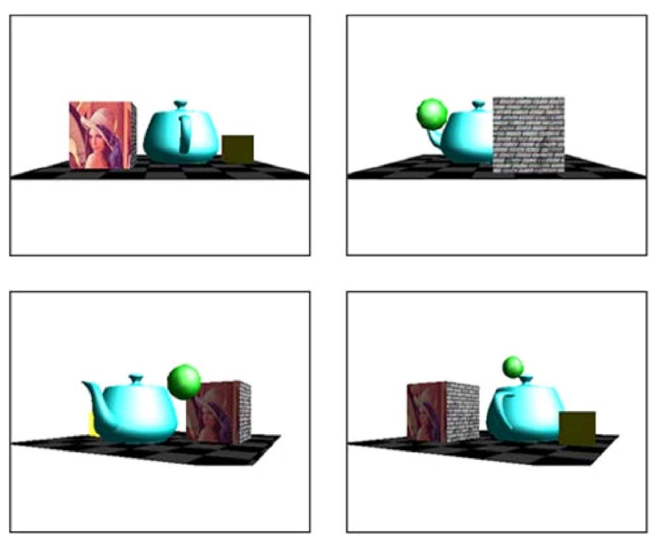

Fig. 9 Selected left-most images of the test sequence from OpenGL API.

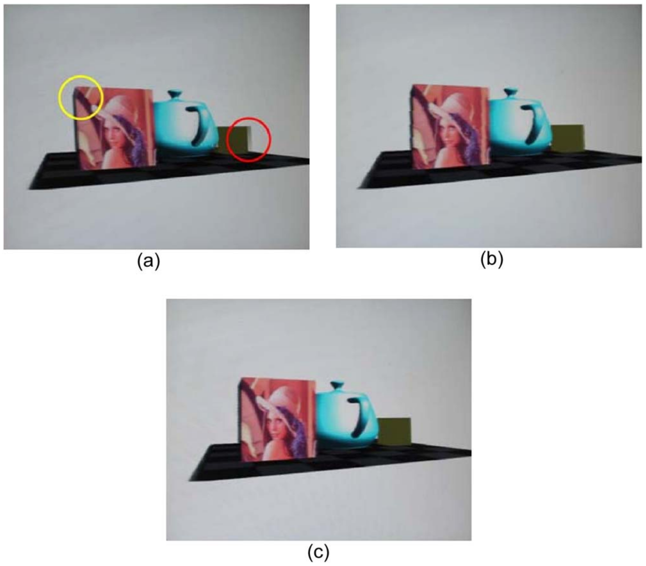

Fig. 10 Pictures from a lenticular display, whose multiplexed image is computed by: (a) the integer viewpoint determination formula; (b) the corrected viewpoint determination formula with the integer viewpoint; and (c) the corrected viewpoint determination formula with the floating-point viewpoint.

ment correction and a simultaneous floating-point view interpolation and multiplexing method for lenticular displays. The simultaneous view interpolation and multiplexing algorithm uses the left-most and right-most view images and their depth information. In the proposed method, each pixel has a floating-point viewpoint from the corrected viewpoint determination formula. We observed boundary dislocations from the integer viewpoint determination formula and slanted line pattern artifacts from the corrected viewpoint determination formula with the integer viewpoint, and found that both dislocations and artifacts are dramatically reduced by applying the corrected viewpoint determination formula with the floating-point viewpoint.

Since the proposed algorithm is simple and only uses small memory, the algorithm could be implemented in a 3-D graphics processor for real-time image generation for a lenticular display. We expect that the algorithm can be adopted for the development of 3-D graphics or content on lenticular devices.

Acknowledgment

This research was supported in part by the University IT Research Center Project.

References

1. N. Holliman, "3D display systems," Technical Report, Department of Computer Science, University of Durham (2002).
2. J. Y. Son, B. Javidi, and K. D. Kwack, "Methods for displaying three-dimensional images," *Proc. IEEE* **94**(3), 502–523 (2006).
3. L. M. J. Meesters, W. A. IJsselsteijn, and P. J. H. Seuntiëns, "A survey of perceptual evaluations and requirements of three-dimensional TV," *IEEE Trans. Circuits Syst. Video Technol.* **14**(3), 381–391 (2004).
4. C. V. Berkel, D. W. Parker, and A. R. Franklin, "Multiview 3D-LCD," in *Stereoscopic Displays Virtual Reality Systems*, *Proc. SPIE* 2653, 32–39 (1996).
5. C. V. Berkel, "Image preparation for 3D-LCD," in *Stereoscopic Displays Virtual Reality Systems*, *Proc. SPIE* **3639**, 84–91 (1999).
6. Y. G. Lee and J. B. Ra, "Reduction of the distortion due to non-ideal lens alignment in lenticular 3D displays," in *Stereoscopic Displays Virtual Reality Systems*, *Proc. SPIE* **5664**, 506–516 (2005).
7. E. Chen and L. Williams, "View interpolation for image synthesis," in *Proc. SIGGRAPH'93*, Anheim (August 1993).
8. D. Scharstein, *View Synthesis Using Stereo Vision*, LNCS, Vol. **1583**, Springer, New York (1999).
9. J. H. Park and H. W. Park, "Fast view interpolation of stereo images using image gradient and disparity triangulation," *Signal Process. Image Commun.* **18**, 401–416 (2003).
10. OpenGL Architecture Review Board, *The OpenGL Reference Manual: The Official Reference Document to OpenGL*, Version 1.4, Dave Shreiner, Ed., Addison-Wesley, Reading, Massachusetts (2004).
11. http://www.opengl.org/
12. S. H. Kim, J. S. Yoon, C. H. Yu, D. H. Kim, K. S. Chung, H. S. Lim, H. W. Park, and L. S. Kim, "A 36 fps SXGA 3D display processor with a programmable 3D graphics rendering engine," in *Proc. ISSCC*, San Francisco, CA (February 2007).

HanShin Lim received a BS degree in electrical engineering from Yonsei University, Seoul, Korea in 2004 and a MS degree in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, where he is currently pursuing a PhD degree in the Department of Electrical Engineering. His research interests include image processing, 3-D displays, and multiview image manipulation.

Seok-Hoon Kim received a BS degree in electrical engineering and computer science from KAIST, Daejeon in 2005, and also completed a MS degree there in 2007. He is currently working toward a PhD degree in electrical engineering and computer science at KAIST. His current interests are 3-D graphics hardware and 3-D displays.

Yun-Gu Lee was born in Gyeonggi Province, Korea, in 1978. He received BS, MS, and PhD degrees in electrical engineering and computer science from the Korea Advance Institute of Science and Technology (KAIST), Daejeon, in 2000, 2002, and 2006, respectively. In 2006, he joined the Digital Media R&D Center, Samsung Electronics Company, Ltd., Suwon, Korea, where he is a senior engineer. His research interests include digital image processing, video signal coding, and 3-D display systems.

Lee-Sup Kim received a BS degree in electronics engineering from Seoul National University, Seoul, Korea, in 1982, and MS and PhD degrees in electrical engineering from Stanford University, Stanford, CA, in 1986 and 1990, respectively. He was a Postdoctoral Fellow at Toshiba Corporation, Kawasaki, Japan, from 1990 to 1993, where he was involved in the design of the high-performance DSP and single-chip MPEG2 decoder. Since March 1993, he has been with the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea. In September 2002, he became a professor. During 1998, he was on the sabbatical leave with Chromatic Research and SandCraft Inc. in Silicon Valley. His research interests are 3-D graphics processing unit design and high-speed/low-power digital IC design.

Jong Beom Ra received his BS degree in electronic engineering in 1975 from Seoul National University, Korea, and his MS and PhD degrees in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, in 1977 and 1983, respectively. From 1983 to 1987, he was an associate research scientist with Columbia University, New York. Since 1987, he has been a professor with the Department of Electrical Engineering and Computer Science, KAIST, where he currently directs the Image Information Research Center. His research interests are digital image processing, video signal processing, 3-D visualization, 3-D display systems, and medical imaging, including magnetic resonance imaging. He is a member of SPIE and a senior member of the IEEE.

HyunWook Park is a professor of the department of electrical engineering at KAIST. He received a BS degree in electrical engineering from Seoul National University, Seoul, Korea in 1981 and MS and PhD degrees in electrical engineering from Korea Advanced Institute of Science and Technology (KAIST), Seoul, Korea in 1983 and 1988, respectively. He has been a professor in the electrical engineering department since 1993 and an adjunct professor in the biosystem department since 2003, at KAIST, Korea. He was a research associate at the University of Washington from 1989 to 1992 and was a senior executive researcher at Samsung Electronics Co. Ltd. from 1992 to 1993. His current research interests include image computing system, image compression, medical imaging, and multimedia systems.