Computational approaches for fast generation of digital 3D video holograms

Invited Paper

Seung-Cheol Kim and Eun-Soo Kim*

3D Display Research Center, Department of Electronic Engineering, Kwangwoon University,
Wolgye-Dong, Nowon-Gu, Seoul 139-701, Korea
**E-mail: eskim@kw.ac.kr*

Received July 17, 2009

Several approaches for fast generation of digital holograms of a three-dimensional (3D) object have been discussed. Among them, the novel look-up table (N-LUT) method is analyzed to dramatically reduce the number of pre-calculated fringe patterns required for computation of digital holograms of a 3D object by employing a new concept of principal fringe patterns, so that problems of computational complexity and huge memory size of the conventional ray-tracing and look-up table methods have been considerably alleviated. Meanwhile, as the 3D video images have a lot of temporally or spatially redundant data in their inter- and intra-frames, computation time of the 3D video holograms could be also reduced just by removing these redundant data. Thus, a couple of computational methods for generation of 3D video holograms by combined use of the N-LUT method and data compression algorithms are also presented and discussed. Some experimental results finally reveal that by using this approach a great reduction of computation time of 3D video holograms could be achieved.

OCIS codes: 090.1760, 090.5694, 050.1940, 999.9999 (look-up table).
doi: 10.3788/COL20090712.1083.

1 Introduction

Recently, a lot of research works have been actively done on the three-dimensional (3D) imaging and display technology due to its high interests throughout the world[1−7]. Among them, the holographic technology has been particularly regarded as one of the promising and attractive approaches for creating the most authentic illusion of observing volumetric objects. It is because the holographic technology can supply very high-quality object images and accurate depth cues viewed by human eyes without any special observation devices[8−11].

However, recording holograms of real 3D objects in the optical holographic system may demand wave interference between the two intense laser beams with a high degree of coherence between them in a dark room. Therefore, this system must be kept to be very stable since even a very slight movement can destroy the interference fringes, in which both intensity and phase information of the 3D objects are contained. These requirements, together with the development and printing processes, have prevented conventional hologram recorders from becoming widely used in the outdoor recording.

As a partial solution for these limitations of the conventional holographic system, a new approach, so-called computer-generated hologram (CGH), has been suggested[12]. A CGH is a digital hologram generated by computing the interference pattern produced by the object and the reference waves. Using this CGH pattern, an electro-holographic 3D display system can be constructed[13].

In this approach, a ray-tracing method has been originally employed for calculating the contributions at the hologram plane from each object point source. That is, an object image to be generated can be approximated as a collection of self-luminous points of light, therefore the fringe patterns for all object points are calculated with the ray-tracing method and added up to obtain the whole interference pattern of the object image.

This method can produce arbitrary 3D images including image-plane holograms, in which images might lie in the vicinity of the hologram, so that it might be more suitable for various display geometries. However, this approach shows a computation complexity, since it requires one by one calculation of the fringe pattern per image point per hologram sample. Thus, real-time generation of the CGH pattern for a 3D image could not be achievable[14].

To overcome this problem, a look-up table (LUT) method has been presented by Lucente[14]. In this method, an object image to be generated is also approximated as a collection of self-luminous points of light likewise the case of the ray-tracing method, but all fringe patterns corresponding to point source contributions from each of the possible locations in image volume are pre-calculated and stored in the LUT. Then, in the process of CGH generation, fringe patterns for each point in the object image can be generated just by accessing the corresponding ones from the pre-calculated LUT, which is contrary to the ray-tracing method, where fringe patterns for all object points are directly calculated on the one by one basis. Therefore, a great increase in computation speed can be obtained with this LUT method. But the greatest drawback of this approach is the enormous memory size of the LUT[15].

Recently, a novel look-up table (N-LUT) method to be able to significantly reduce the number of pre-calculated interference patterns required for generation of digital holograms has been proposed[15]. In this N-LUT method, a 3D object is approximated as a set of discretely sliced image planes having different depths, and only the fringe pattern of the object point centered on each image plane is pre-calculated, which is called here a principal fringe pattern (PFP). Fringe patterns for other object points on that image plane can be obtained by simply shifting this pre-calculated PFP to those points depending on the location values displaced from the center. That is, in this method, only the PFPs for object points centered on each image plane are pre-calculated and stored in the N-LUT, so that the size of the N-LUT could be dramatically reduced compared with the size of the conventional LUT method.

On the other hand, ordinary 3D video images have numerous similarities between the video frames, so that the computation time to generate digital holograms for them can be effectively reduced by the removal of redundant data just like in the case of the conventional two-dimensional (2D) video images. This similarity between neighboring video frames is called here a temporal redundancy. Thus, if we employ a video compression technique in the hologram generation process of the 3D video images, 3D video data to be calculated can be dramatically decreased, and as a result, calculation time of the 3D video holograms can also be significantly reduced.

Moreover, 3D image consists of depth and intensity data, contrary to the 2D image which has only the intensity information. Just like the case of the 2D image, adjacent pixels of a 3D image have very similar values of intensity and depth and some of them even have the exactly same values. In other words, a 3D image has a spatial redundancy in intensity and depth data. This spatial redundancy can be represented with the run-length encoding (RLE) method, which has been used for data reduction of the conventional 2D images. That is, by applying this conventional RLE method to the 3D object, its spatially redundant intensity and depth data can be removed, which results in the reduction of 3D image points to be involved in generation of the CGH pattern. In other words, the calculation time required for CGH generation of the 3D object can be dramatically shortened.

Accordingly, in this paper, several approaches for fast generation of CGH patterns for 3D video images by using the N-LUT method, and the combined use of the data compression techniques and the N-LUT method are reviewed and the results are discussed.

2. Approaches for generation of CGH patterns

2.1 Ray-tracing method

In the classical ray-tracing method, an object image to be computationally generated can be approximated as a collection of self-luminous points of light. And then, fringe patterns for all object points are calculated by using the equations of optical diffraction and interference, and added up to obtain the whole interference pattern of the object image.

Figure 1 shows the geometry for calculating the Fresnel hologram of an object by using the ray-tracing method. Here, the location coordinate of the pth object point is specified by (x_p, y_p, z_p) and each object point is assumed to have an associated real-valued magnitude and phase of a_p, ϕ_p, respectively.

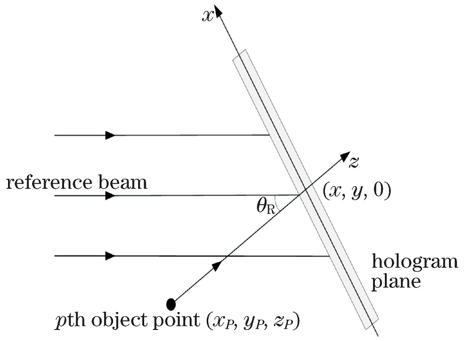


Fig. 1. Computational model for generation of Fresnel hologram.

Here, the CGH is also assumed to be positioned at the plane of $z = 0^{[16]}$. Then, the complex amplitude $O(x, y)$ on the hologram plane can be obtained by superposition of the object wavefronts as

$$O(x, y) = \sum_{p=1}^{N} \frac{a_p}{r_p}\exp[j(kr_p + \phi_p)], \tag{1}$$

where N represents the number of object points. The wave number k is defined as k=$2\pi/\lambda$, in which λ is the free-space wavelength of the light. It must be noted that the factor exp(jωt) is not included explicitly in Eq. (1). Here, the oblique distance r_p between the pth object point and the point $(x, y, 0)$ on the hologram plane is given by

$$r_p = \sqrt{(x - x_p)^2 + (y - y_p)^2 + z_p^2}. \tag{2}$$

Also, the complex amplitude of the collimated reference beam $R(x, y)$ is represented by

$$R(x, y) = a_{\rm R}\exp[{\rm j}(kx\sin\,\theta_{\rm R})], \tag{3}$$

where $a_{\rm R}$ and $\theta_{\rm R}$ mean the real-valued amplitude and the incident angle of the reference beam, respectively.

The total fringe intensity on the hologram plane obtained from the interference pattern between the object beam $O(x, y)$ and the reference beam $R(x, y)$ is given by

$$\begin{aligned} I(x,y) &= |R(x,y) + O(x,y)|^2 \\ &= |R(x,y)|^2 + |O(x,y)|^2 \\ &\quad + 2|R(x,y)||O(x,y)| \\ &\quad \cos[kr_p + kx\sin\theta_R + \phi_p]. \end{aligned} \tag{4}$$

In Eq. (4), the first and second terms represent the reference and the object self-interference beam intensity, respectively. Only the third term means the interference pattern between the object beam and the reference beam, in which holographic information of the object image to be generated is contained, so that $I(x, y)$ of Eq. (4) can be substituted into

$$I(x, y) = 2\sum_{p=1}^{N} \frac{a_p}{r_p}\cos(kr_p + kx\sin\,\theta_{\rm R} + \phi_p). \tag{5}$$

Equation (5) reveals that the fringe patterns for all object points must be directly computed on the one by one basis in the classical ray-tracing method. Therefore, as the number of object points increases, the computation time required to calculate the CGH pattern for that object sharply increases. This makes the method so difficult to implement the digital hologram of a 3D object in the practical fields.

2.2 LUT method

Another approach for solving the computation complexity of the classical ray-tracing method in generation of CGH patterns of an object is a LUT method proposed by Lucente[14]. In this approach, all contributions to $I(x, y)$ from an image point of unity magnitude for each possible value of (x_p, y_p, z_p) in the image volume are computed in advance, and the results are stored in the LUT.

An efficient LUT must include all of the spatially dependent fringe patterns shown in Eq. (5), so that the unity-magnitude fringe pattern for an object point (x_p, y_p, z_p), $T(x, y\ ; x_p, y_p, z_p)$ can be defined by

$$T(x, y; x_p, y_p, z_p) \equiv \frac{1}{r_p}\cos\left[kr_p + kx \sin\theta_\mathrm{R} + \phi_p\right]. \qquad (6)$$

Once the fringe patterns for all object points of Eq. (6) are computed and stored in the LUT, this LUT can be used for generating the CGH patterns of arbitrary object images. Instead of directly computing the fringe patterns of the object points at each time we need, as shown in Eq. (5) in the classical ray-tracing method, each object point (x_p, y_p, z_p) is merely mapped into the corresponding pre-calculated fringe pattern in the LUT method. Thus, the finally obtained hologram pattern for the object $I(x, y)$ in the LUT method can be expressed in terms of the pre-calculated fringe patterns of Eq. (6) as

$$I(x, y) = \sum_{p=1}^{N} a_p T\left(x, y; x_p, y_p, z_p\right). \qquad (7)$$

In the LUT method, a CGH pattern of the object image can be generated just by bringing back the corresponding fringe patterns to each object point from the LUT and adding them together. That is, there is no need to directly calculate the fringe patterns of the object points just like in the case of the ray-tracing method mentioned above. Only two operations of multiplication and addition are needed in the LUT method. Therefore, this LUT method can drastically improve the calculation speed of the CGH pattern for an object image. This method, however, has a big burden to require a massive memory space of the LUT for storing fringe patterns of all possible object image points[15].

2.3 N-LUT method

Another approach, the so-called N-LUT method, was recently proposed which can dramatically reduce the size of the conventional LUT while keeping the conventional LUT's advantage of fast computational speed[15]. This method consists of three steps: construction of the N-LUT, generation of the CGH pattern for an object using the N-LUT, and reconstruction of the object image. Firstly, the N-LUT is constructed with only the fringe patterns of the object points centered on each depth-dependently sliced image planes of the 3D object, which is called PFPs. Then the CGH patterns are generated using these PFPs provided by the N-LUT, and from which 3D object images can be finally reconstructed.

In the N-LUT method, the N-LUT contains only the fringe patterns of the object points with unity magnitudes located at each center of the depth-dependent image planes of the object. Each PFP can be regarded as the Fresnel zone pattern (FZP) computed at each depth. Therefore, the unity magnitude PFP for the center object point $(0, 0, z_p)$ on the image plane having a depth of z_p, $T(x, y; z_p)$ can be defined as

$$T(x, y; z_p) \equiv \frac{1}{r_p}\cos\left[kr_p + kx \sin\theta_\mathrm{R} + \phi_p\right], \qquad (8)$$

where r_p is the same as that of Eq. (2).

Then, the fringe patterns for other object points on each image plane can be obtained by simply shifting this pre-calculated PFP according to the displaced location values from the center to those points and adding them together. Fringe patterns for all object points located on each image plane can be generated by adding the shifted versions of the PFP. Therefore, the final CGH pattern for an object volume can be obtained by overlapping all PFPs generated on each depth-dependent image plane. The CGH pattern for the object $I(x, y)$ in the N-LUT method can be expressed in terms of the shifted versions of pre-calculated PFPs of Eq. (8) as

$$I(x, y) = \sum_{p=1}^{N} a_p T\left(x - x_p, y - y_p; z_p\right). \qquad (9)$$

Equation (9) shows that the CGH pattern of an object can be obtained just by shifting the PFPs depending on the displaced values of image points from the reference points on each image plane and adding up all together. That is, in the conventional LUT method, the CGH pattern can be generated by multiplying the amplitudes of each object point to the corresponding PFP and adding them, whereas in N-LUT method by multiplying the amplitudes of each object point to the corresponding PFPs pre-calculated for each depth plane and shifting them depending on the displaced values of the object points in the x, y directions and adding them together, the CGH pattern for the object can be finally generated.

3. Generation of CGH patterns by considering data redundancy of 3D video images

3.1 Temporal redundancy-based CGH generation of 3D video images

3.1.1 Temporal redundancy of 3D video images

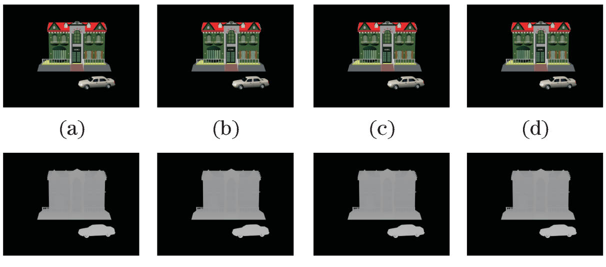


Fig. 2. Adjacent four frames of a 3D video. (a)−(d) Intensity images; (e)−(h) depth images.

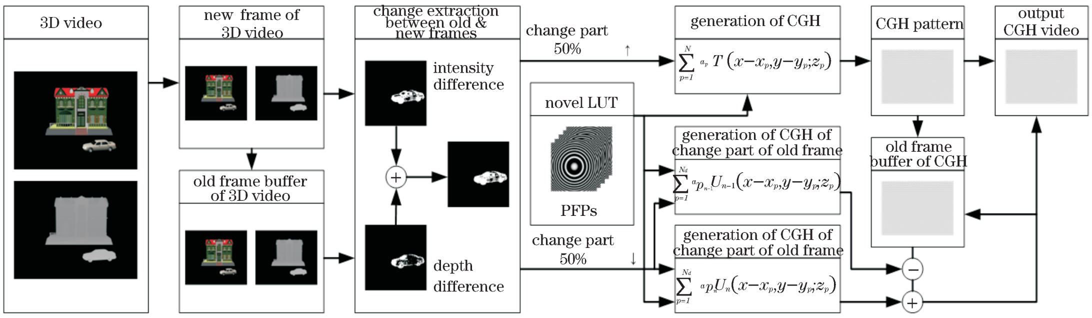

3D video
new frame of 3D video
old frame buffer of 3D video
change extraction between old & new frames
intensity difference
depth difference
change part 50% ↑
change part 50% ↓
novel LUT
PFPs
generation of CGH
$\sum_{p=1}^{N} a_p T(x-x_p, y-y_p; z_p)$
generation of CGH of change part of old frame
$\sum_{p=1}^{N_d} a_{p_{n-1}} U_{n-1}(x-x_p, y-y_p; z_p)$
generation of CGH of change part of old frame
$\sum_{p=1}^{N_d} a_{p_n} U_n(x-x_p, y-y_p; z_p)$
CGH pattern
old frame buffer of CGH
output CGH video
−
+

Fig. 3. Block diagram of the temporal redundancy-based 3D CGH generation method.

Contrary to the 2D video, 3D video is a collection of sequential 3D images having depth data as well as intensity. Just like the case of the 2D video images, neighboring moving pictures in the 3D video also differ slightly from each other, but in the 3D video images, intensity and depth data are simultaneously changed.

For example, Fig. 2 shows four adjacent frames of the 3D video images composed of a fixed house and a moving car. In these four frames, only the car is moving and the other part of the image is not changed. Thus, there exist slight changes in intensity and depth between the consecutive frames. In case of the 3D video images, temporal redundancy in both of the intensity and depth images must be considered for effective reduction of the 3D video data.

Now, redundant data between the 3D video frames can also be removed by applying the differential pulse code modulation (DPCM) technique to each of the intensity and depth data. As a result, the calculation time for generation of the 3D video images can be dramatically shortened.

3.1.2 Fast generation of CGH patterns by considering the temporal redundancy

Figure 3 shows an overall block-diagram of the method to generate digital video holograms for the 3D moving pictures by using the N-LUT method and the temporal redundancy property of 3D video images[17].

This method largely consists of three steps. In the first step, intensity and depth data of the current frame of the 3D video are extracted and compared with those of the previous frame. In the second step, the CGH pattern for the current video frame is calculated with the conventional N-LUT method in the case that the difference of intensity and depth data between these two consecutive video frames is larger than 50%. In the third step, the calculated CGH pattern is transmitted to the CGH video output as well as stored in the previous frame buffer of the CGH.

On the contrary, if the difference between two frames is smaller than 50% in the second step, CGH patterns for changed parts in both of the previous and current frames are calculated with the N-LUT method. Then, the CGH pattern for the previous frame is subtracted from that of the current frame and the result is added to the CGH pattern of the previous frame. Finally, the calculated CGH pattern is transmitted to the CGH video output as well as stored in the previous frame buffer of the CGH.

3.1.3 Computation of 3D video holograms

Basically, a 3D object image to be generated can be approximated as a collection of self-luminous points of light. Here, the horizontal, vertical, and depth location of an object point is specified as x_p, y_p, and z_p, respectively, and each point has an associated real-valued magnitude and phase of a_p, ϕ_p, respectively. Thus, the complex amplitude $O_n(x, y)$ of the nth frame on the hologram plane can be obtained by superposition of the object wavefronts as

$$O_n(x,y) = \sum_{p=1}^{N_n} \frac{a_p}{r_p} \exp[\mathrm{j}(kr_p + \phi_p)], \qquad (10)$$

where N_n represent the number of object points of the nth frame and k means the wave number defined as $k = 2\pi/\lambda$, in which λ is the free-space wavelength of the light, and r_p is the same parameter defined in Eq. (2). It must be noted here that the factor $\exp(\mathrm{j}\omega t)$ is not included explicitly in Eq. (10).

In this approach, the N-LUT method is used to calculate the hologram patterns[15]. In the N-LUT method, only the fringe patterns of the object points with unity magnitudes located at each center of the depth-dependent image planes of the object, so-called PFPs are stored, so that a unity-magnitude PFP for the center object point $(0, 0, z_p)$ on the image plane having a depth of z_p, $T(x, y; z_p)$ can be given by Eq. (8) mentioned above.

In this method, firstly, the CGH pattern for the first frame of the 3D video images is generated with the N-LUT method, and the calculated CGH pattern as well as 3D data of the first frame is stored in the buffer. In the next step, the difference in 3D data between the first and second frames is extracted, and the CGH pattern for the changed part of the first frame is generated and is subtracted from the CGH pattern of the first frame stored in the CGH pattern buffer. Subsequently, the CGH pattern for the changed part of the second frame is generated and added to the subtracted CGH. That is, the CGH pattern of the second frame can be generated by subtracting the CGH pattern for the disappearing part of the first frame from the CGH pattern of the first frame and adding the CGH pattern for the appearing part of the second frame to the CGH pattern of the first frame as

$$\begin{aligned} I_n(x,y) = & \; I_{n-1}(x,y) \\ & -\sum_{p=1}^{N_d} a_{p_{n-1}} U_{n-1}(x-x_p, y-y_p; z_p) \\ & +\sum_{p=1}^{N_d} a_{p_n} U_n(x-x_p, y-y_p; z_p), \end{aligned} \qquad (11)$$

where I_n is the CGH pattern for the nth frame, N_d is the number of different image points in 3D data between the nth frame and the $(n-1)$th frame. Moreover, $U_n(x, y; z_p)$ means the PFPs of the nth frame:

$$U_n(x,y;z_p) = \begin{cases} T(x,y;z_p) & \text{for changed part} \\ 0 & \text{for unchanged part} \end{cases} . \quad (12)$$

Then, the calculated CGH pattern of the second frame $I_n(x, y)$ is moved to the CGH video output as well as stored in the previous frame buffer of the CGH. These processes might be repeated for all of the video frames.

3.2 Spatial redundancy-based CGH generation of 3D video images

3.2.1 Spatial redundancy of 3D images

Spatial redundancy represents a statistical correlation between pixels within an image frame. Hence, it is also called inter-frame redundancy. It is well known that for most properly sampled television (TV) signals, the normalized autocorrelation coefficients along a row (or a column) with a one-pixel shift are very close to the maximum value 1. That is, intensity values of the pixels along a row (or a column) have a very high autocorrelation (close to the maximum autocorrelation) with those of pixels along the same row (or the same column) but are shifted by a pixel. This does not come as a surprise because most of the intensity values may change continuously from pixel to pixel within an image frame except for the edge region[18].

This spatial redundancy mentioned above has been used as a concept of reduction of the data amount to be sent in the conventional communication system. One of the data reduction methods basing on this spatial redundancy concept is the RLE method[19]. RLE is a very simple form of data compression in which runs of data (that is, sequences in which the same data value occurs in many consecutive data elements) are stored as a single data value and count, rather than as the original run. This might be the most useful on data that may contain many such runs[20].

On the other hand, 3D images are composed of both depth and intensity data, contrary to the 2D images which only have intensity data. Just like the case of the 2D image, adjacent pixels of a 3D image have very similar values of intensity and depth, so that a 3D image also has a spatial redundancy both in intensity and depth data.

Here the spatial redundancy of a 3D image represents the statistical correlation between the pixels within a 3D image frame. That is, intensity and depth values of the pixels along a row (or a column) also have a very high autocorrelation with those of pixels along the same row (or the same column) just like the case of the 2D image.

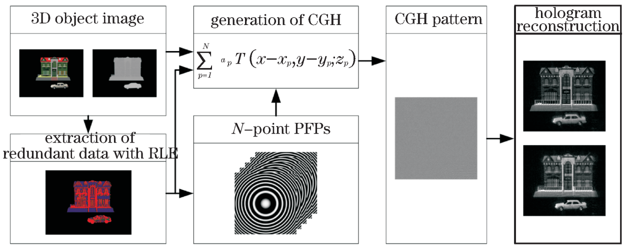
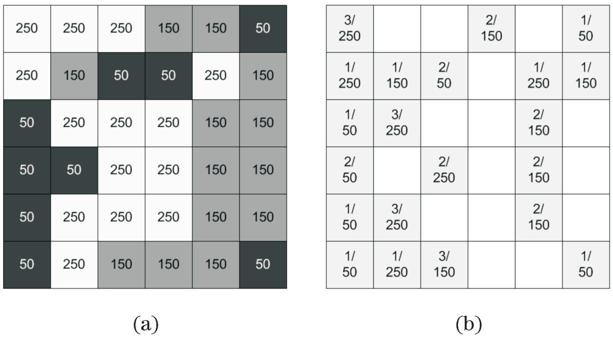

3D object image
generation of CGH
$\sum_{p=1}^{N} a_p T(x-x_p, y-y_p; z_p)$
CGH pattern
hologram reconstruction
extraction of redundant data with RLE
N–point PFPs

Fig. 4. Block diagram of a spatial redundancy-based 3D CGH generation method.



250	250	250	150	150	50
250	150	50	50	250	150
50	250	250	250	150	150
50	50	250	250	150	150
50	250	250	250	150	150
50	250	150	150	150	50

(a)

3/ 250			2/ 150		1/ 50
1/ 250	1/ 150	2/ 50		1/ 250	1/ 150
1/ 50	3/ 250			2/ 150	
2/ 50		2/ 250		2/ 150	
1/ 50	3/ 250			2/ 150	
1/ 50	1/ 250	3/ 150			1/ 50

(b)

Fig. 5. Spatial redundancy of the 3D input image. (a) Gray scales of the test image; (b) spatial redundancy map.

Accordingly, by applying the RLE method mentioned above to both of intensity and depth data of the 3D image, 3D image data can be significantly reduced, which results in a decrease of the calculating time required for the generation of 3D holograms.

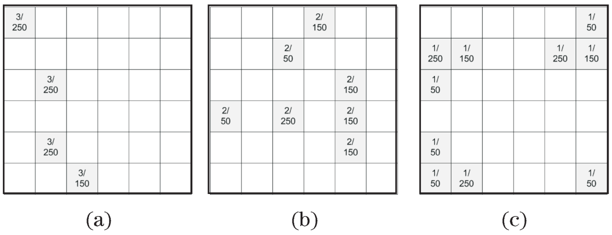
3.2.2 Fast generation of CGH patterns considering the spatial redundancy

Figure 4 shows an overall block-diagram of the proposed method to fast generate CGHs for a 3D object[21]. The proposed method largely consists of four steps. Firstly, the spatial redundancy of the intensity and depth data of the 3D object is analyzed by using the RLE method and the results are re-grouped into the N-point redundancy map according to the number of the neighboring object points having the same 3D value. Secondly, N-point PFPs corresponding to the N-point redundancy maps are calculated by shifting and adding the 1-point PFP of the conventional N-LUT. Thirdly, the CGH pattern of the 3D object is calculated with these pre-calculated N-point PFPs. Fourthly, the 3D object image is reconstructed from the generated CGH pattern.

3.2.3 Extraction of redundant data from a 3D object

In general, adjacent pixels of the 3D object image have nearly the same brightness, color, and depth values. That is, there can be a spatial redundancy both in intensity and depth data of the input 3D image. For example, Fig. 5(a) shows the intensity distribution of an object image having a resolution of 6×6 pixels. Here it is assumed that the object image is located in the same depth plane for convenience.

On the first horizontal line of Fig. 5(a), we can find that some adjacent pixels have the same intensity values. That is, by using the RLE method, these redundant data can be represented by three ‘250’, two ‘150’, and one ‘50’. Figure 5(b) shows a map of the spatial redundancy, so-called 3-point redundancy map extracted from the object image of Fig. 5(a) with the RLE method, in which ‘3/250’ means that there exist three adjacent image pixels having the same gray value of ‘250’ in the corresponding row.



(a)

3/ 250					
	3/ 250				
	3/ 250				
		3/ 150			

(b)

			2/ 150		
		2/ 50			
				2/ 150	
2/ 50		2/ 250		2/ 150	
				2/ 150	

(c)

					1/ 50
1/ 250	1/ 150			1/ 250	1/ 150
1/ 50					
1/ 50					
1/ 50	1/ 250				1/ 50

Fig. 6. Separated spatial redundancy maps of Fig. 5(b). (a) 1-point redundancy map; (b) 2-point redundancy map; (c) 3-point redundancy map.

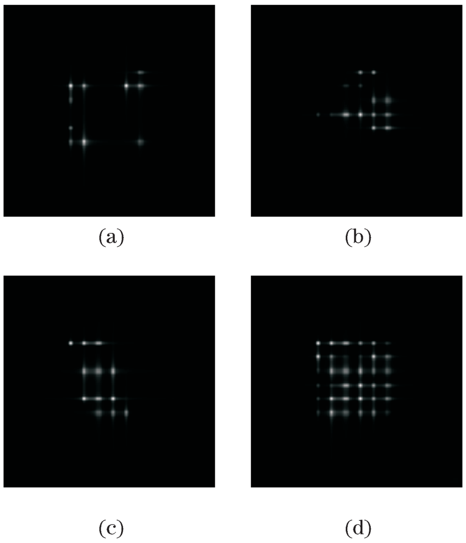

(a) (b) (c) (d)

Fig. 7. Reconstructed object images from the CGH patterns. (a) 1-point image; (b) 2-point image; (c) 3-point image; (d) all point image.

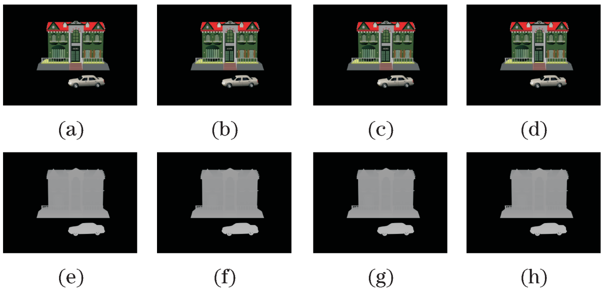

(a) (b) (c) (d)
(e) (f) (g) (h)

Fig. 8. Four frames of the intensity and depth images for the test 3D videos. (a)−(d) Intensity images (31st−34th frames); (e)−(h) depth images (31st−34th frames).

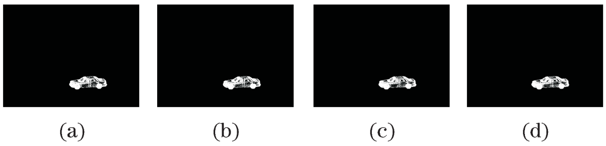

(a) (b) (c) (d)

Fig. 9. Changed parts of the intensity and depth data between the previous and current 3D video frames (31st−34th frames).

As can be seen in Fig. 5(b), there are four and seven cases that three and two adjacent pixels have the same intensity and depth values, respectively. In addition, in ten cases, adjacent pixels do not have the same intensity and depth values.

Therefore, only 21(= 4+7+10) calculation processes will be needed in this method for the generation of the CGH pattern instead of 36(= 6×6) calculation processes normally needed in the conventional approach. That is, in this approach, 15 calculation processes in CGH generation can be reduced. There exist 15 empty spaces called ‘don’t care condition’ in Fig. 5(b), in which the ‘don’t care condition’ means no need of hologram calculation.

3.2.4 Generation of *N*-point PFPs

Here, the 1-point PFP is defined as Eq. (8) mentioned above in the conventional N-LUT[15]. The two-point PFP for two adjacent object points with unity magnitude and depth of z_p can be expressed by

$$T_2(x, y, z_p) \equiv T(x, y; z_p) + T(x - d, y; z_p), \qquad (13)$$

where d is a discretization step of adjacent points[15]. Likewise, the N-point PFP for N adjacent object points with unity magnitude and depth of z_p, $T_n(x,\ y;\ z_p)$ can be generally expressed by

$$T_n(x, y; z_p) \equiv \sum_{k=1}^{n} T(x - (k-1)d, y; z_p). \qquad (14)$$

Therefore, N adjacent object points can be displayed by using the spatial redundancy map and the N-point PFP. That is, for the case of N adjacent object points

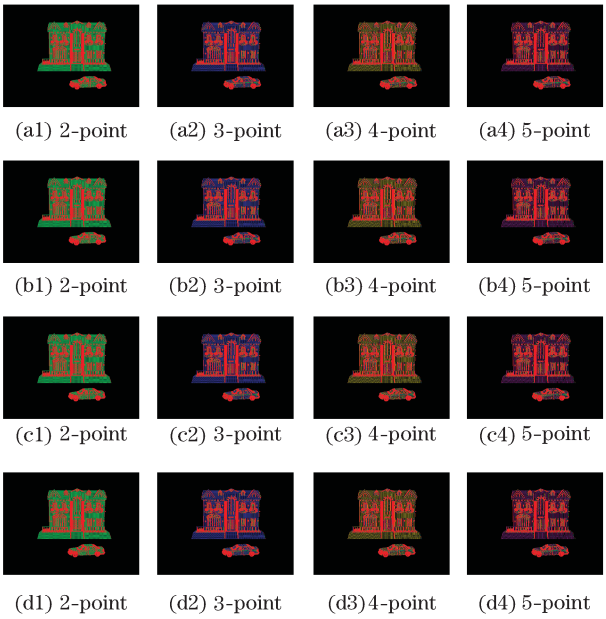

(a1) 2-point (a2) 3-point (a3) 4-point (a4) 5-point
(b1) 2-point (b2) 3-point (b3) 4-point (b4) 5-point
(c1) 2-point (c2) 3-point (c3) 4-point (c4) 5-point
(d1) 2-point (d2) 3-point (d3)4-point (d4) 5-point

Fig. 10. Spatial redundancy maps extracted from horizontal scanning of the test 3D video object. (a) 31st frame; (b) 32nd frame; (c) 33rd frame; (d) 34th frame.

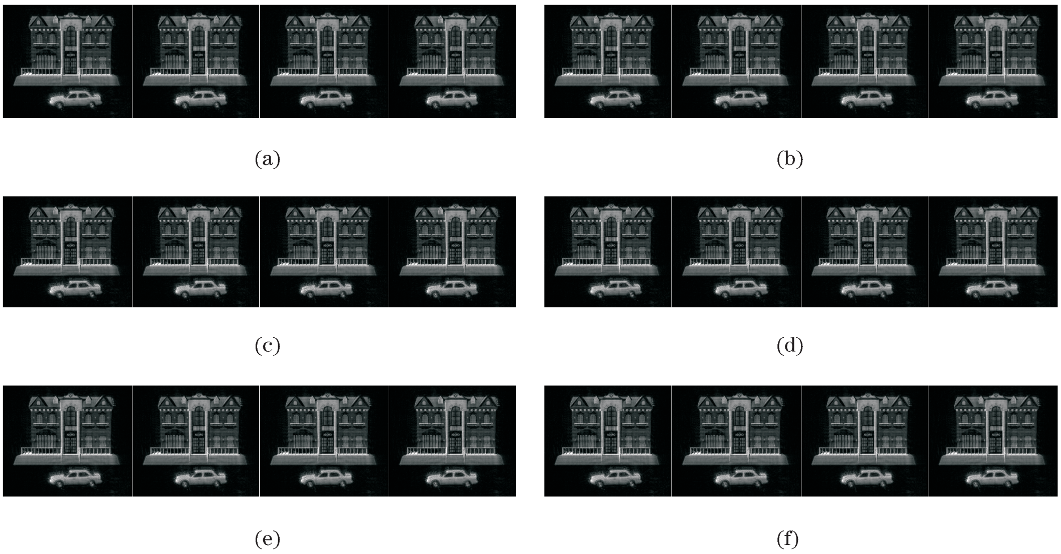


Fig. 11. Reconstructed images for each method. (a) N-LUT method; (b) temporal redundancy; (c) spatial redundancy (2-point); (d) spatial redundancy (3-point); (e) spatial redundancy (4-point); (f) spatial redundancy (5-point).

having the same intensity and depth values, N calculation processes are needed in the conventional N-LUT method, whereas in the proposed method only one calculation process is needed. Therefore, the calculation time for the generation of the CGH pattern of the 3D object can be significantly reduced.

3.2.5 Calculation of CGH patterns

Figure 6 shows three redundancy maps for one, two, and three neighboring points having the same intensity and depth values, in which each map is separated from the original object points of Fig. 5(b). That is, in the case of Fig. 6(a) there are no adjacent object points having the same intensity and depth values. On the other hand, Figs. 6(b) and (c) show the spatial redundancy maps for two and three adjacent pixels with the same intensity and depth values along the horizontal direction. The object images reconstructed from the CGH patterns are shown in Fig. 7.

4. Performance analysis of the CGH generation methods

To comparatively evaluate the performances of several CGH generation methods mentioned above, three types of experiments have been performed. For the experiment, 100 frames of 3D video images are computationally generated as a test 3D video, in which each frame has a resolution of 640 × 480 pixels, as shown in Fig. 8. The CGH pattern is assumed to have a resolution of 1200 × 1200 pixels, in which each pixel size is given by 10 × 10 (μm). Moreover, the horizontal and vertical discretization steps are set to be 30 μm. Therefore, the amount of the pixel shift in this method is given by 3 pixels. To fully display the fringe patterns for the first and end image points located on each image plane, the PFP must be shifted by 640 × 3 pixels = 1920 pixels horizontally and 480 × 3 pixels = 1440 pixels vertically. Thus, the total resolution of the PFP should become to be 3120 (1200+1920) × 2640 (1200+1440) pixels. Now, with these test 3D video images, the PFPs for the center image points located on each plane can be calculated.

Some changing parts between the previous and current frames are shown in Fig. 9 for the case of CGH generation method basing on the temporal redundancy. Figure 9 shows that a car might be moved from the right to the left. Using this temporal redundancy map, CGH patterns can be generated.

Figure 10 also shows the spatial redundancy

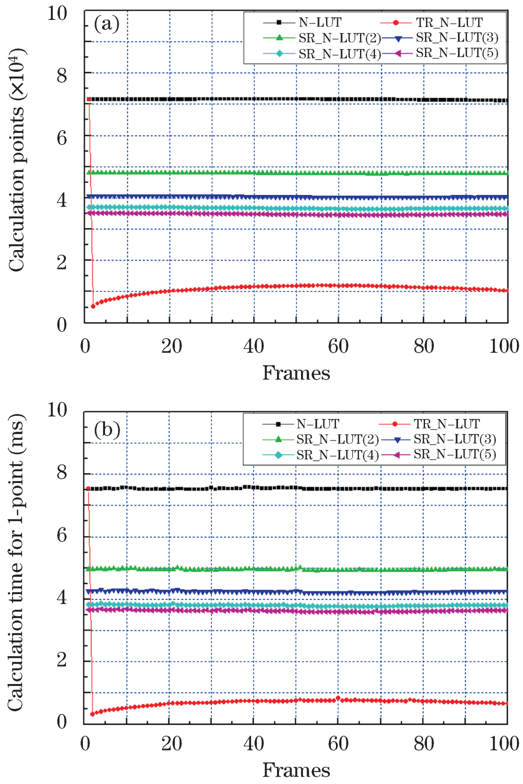


Fig. 12. Comparison results of each method in terms of (a) the number of calculated object points and (b) calculation time for one point. TR and SR represent temporal redudancy and spatial redundancy, respectively.

Table 1. Calculated Points, Calculation Time for 1-Point and Total Memory Size for Each Method

	N-LUT	Temporal Redundancy	Spatial Redundancy			
			2-Point	3-Point	4-Point	5-Point
Average Number of Calculated Point	71486 (100%)	11350 (15.88%)	47929 (67.05%)	40374 (56.48%)	36751 (51.41%)	34799 (48.68%)
Average Computation Time for One Object Point (ms)	7.534 (100%)	0.752 (9.98%)	4.950 (65.70%)	4.240 (56.28%)	3.805 (50.51%)	3.629 (48.17%)
Total Memory Size (GB)	1.96 (100%)	1.96 (100%)	3.93 (200%)	5.89 (300%)	7.86 (400%)	9.82 (500%)

maps extracted from the horizontal scanning of the test object of Fig. 8 by using the RLE method for the case of CGH generation method basing on the spatial redundancy. It is noted here that both of horizontal and vertical scanning methods can be used for extraction of spatial redundancy maps from the test 3D object, but in this experiment the horizontal scanning method is optionally employed.

In the extracted redundancy maps, the red color means that there are no adjacent object points having the same intensity and depth values, while the green, blue, yellow, and purple colors mean that there are two, three, four, and five adjacent objectpoints having the same intensity and depth values, respectively. In addition, the black color means the object points of ‘do not care condition’. Using this spatial redundancy map, CGH patterns can be also generated.

Then, six sets of object images computationally reconstructed from the CGH patterns are shown in Fig. 11. As can be seen in Fig. 11, object images are found to be successfully reconstructed in all cases.

Figure 12(a) shows the number of the calculated object points in generation of CGH patterns for each method. Figure 12(b) shows the comparison results on the calculation times needed for one object point. Table 1 shows the number of the calculated object points, calculation time for 1-point, and total memory size for each method. As we can see in Table 1, the average numbers of calculated object points for the considered methods are estimated to be 71486, 11350, 47929, 40374, 36751, and 34799, respectively. That is, calculated object points can be reduced by 15.88% by using the temporal redundancy, and 67.05%, 56.48%, 51.41%, and 48.68% by using the spatial redundancy, respectively.

These results reveal that the average number of object points to be calculated in these methods can be significantly reduced compared with the conventional N-LUT method. Hence, the values of average calculation time for one object point for these methods are estimated to be 7.534, 0.752, 4.950, 4.240, 3.805, and 3.629 ms, respectively. That is, the average calculation time is reduced by 9.98% by using the temporal redundancy and 65.70%, 56.28%, 50.51%, and 48.17% by using the spatial redundancy, respectively. These results also show that the average calculation time for one object point in these methods can be considerably reduced compared with the conventional N-LUT method.

As also can be seen in Table 1, a total memory size required for storing all fringe patterns of the 3D image volume of 3120 × 2640 × 256 pixels in the conventional N-LUT method is calculated to be 1.96 GB, in which image data for one fringe pattern are assumed to be 7.85 MB (= 3120 × 2640 × 8 bit). In the CGH generation method, using the temporal redundancy might require the same memory size of the conventional N-LUT method because the 1-point PFPs are used. But, in the CGH generation method using the spatial redundancy, the total memory size is calculated to be enlarged by 2, 3, 4, and 5 times compare with the N-LUT method because the 2-, 3-, 4-, and 5-point PFPs are simultaneously used.

These experimental results finally confirm that by the combined use of the N-LUT and data compression methods in the process of generating 3D video holograms, a great reduction of computation time could be obtained, so that this computational approach might be applied for real-time generation of digital holograms of 3D objects in the real world.

5. Conclusions

As a new approach for fast generation of digital holograms of a 3D object, the N-LUT method and its modified versions have been discussed. The N-LUT method is found to significantly alleviate the problems of computational complexity and vast memory size of the conventional ray-tracing and LUT methods. Moreover, by employing the data compression techniques together with the N-LUT method, the computational time of the 3D video holograms could be even more shortened. Some experimental results finally confirm the feasibility of this method in the practical applications of the 3D CGHs.

This work was supported by the MKE (Ministry of Knowledge Economy), Korea, under the ITRC (Information Technology Research Center) support program supervised by the NIPA (National IT Industry Promotion Agency) (NIPA-2009-C1090-0902-0018).

References

1. K. Iizuka, Opt. Photon. News **17,** (7) 42 (2006).
2. S.-C. Kim and E.-S. Kim, Opt. Commun. **249,** 51 (2005).
3. S.-C. Kim and E.-S. Kim, Opt. Commun. **266,** 55 (2006).
4. S.-C. Kim, P. Sukhbat, and E.-S. Kim, Appl. Opt. **47,** 3901 (2008).
5. S.-C. Kim and E.-S. Kim, 3D Res. **1,** 010101 (2009).
6. Y. Kim, K. Hong, and B. Lee, 3D Res. **1,** 010102 (2009).
7. J. Rosen, B. Katz, and G. Brooker, 3D Res. **1,** 010103 (2009).
8. C. J. Kuo and M. H. Tsai, *Three-Dimensional Holo-*

graphic Imaging (Wiley, New York, 2002).

9. U. Schnars and W. Jueptner, *Digital Holography—Digital Hologram Recording, Numerical Reconstruction, and Related Techniques* (Springer, Berlin, 2004).
10. T.-C. Poon, *Digital Holography and Three-Dimensional Display* (Springer, Berlin, 2007).
11. H. Kang, T. Yamaguchi, H. Yoshikawa, S.-C. Kim, and E.-S. Kim, Appl. Opt. **47,** 5784 (2008).
12. A. W. Lohmann and D. P. Paris, Appl. Opt. **6,** 1739 (1967).
13. Y. Li, D. Abookasis, and J. Rosen, Appl. Opt. **40,** 2864 (2001).
14. M. Lucente, J. Electron. Imag. **2,** 28 (1993).
15. S.-C. Kim and E.-S. Kim, Appl. Opt. **47,** D55 (2008).
16. P. Hariharan, *Optical Holography: Principles, Techniques, and Applications* (Cambridge University Press, Cambridge, 1996).
17. S.-C. Kim, J.-H. Yoon, and E.-S. Kim, Appl. Opt. **47,** 5986 (2008).
18. R. C. Dorf, *Electrical Engineering Handbook* (2nd edn.) (CRC Press, Boca Raton, 1997).
19. J. Higgins, *Introduction to SNG and ENG Microwave* (Butterworth-Heinemann, Oxford, 2004).
20. K. N. Ngan, C. W. Yap, and K. T. Tan, *Video Coding for Wireless Communication Systems* (Marcel Dekker, New York, Ltd, 2001).
21. S.-C. Kim and E.-S. Kim, Appl. Opt. **48,** 1030 (2009).