Journal of Imaging Science and Technology® 55(1): 010506–010506-15, 2011.

Efficient Generation of Holographic Video of Three-Dimensional Objects using Spatio-Temporal Redundancy of Three-Dimensional Video Imagery and Novel Look-Up Table Methods

Sun-Hee Kim and Hong-Chul Lee
Division of Information Management Engineering, Korea University, 5-1 Anam-dong, Sungbuk-gu, Seoul 136-701, Korea
E-mail: hclee@korea.ac.kr

Seung-Cheol Kim
3D Display Research Center, Department of Electronic Engineering, Kwangwoon University, 447-1 Wolgye-Dong, Nowon-Gu, Seoul 139-701, Korea

Abstract. *In this paper, a new method for efficient generation of video hologram for three-dimensional (3D) video is proposed by combined use of redundant data of 3D video and look-up table techniques. That is, 3D video is a collection of sequential 3D images having depth data as well as intensity and neighboring moving pictures in the 3D video differ slightly from each other. Therefore, a method for fast computation of computer generated holography (CGH) patterns for 3D video images is proposed by combined use of temporal redundancy and look-up table techniques. Furthermore, adjacent pixels of a 3D image have very similar values of intensity and depth, and some of them even have exactly the same values of them. In other words, a 3D image has a spatial redundancy in intensity and depth data. Therefore, a method for fast computation of CGH patterns for the 3D image taking into account of the spatial redundancy of the 3D image is proposed. To confirm the feasibility of the proposed method, some experiments with a 3D test object are carried out and the results are compared to those of the conventional methods in terms of a computational speed and a required memory size. © 2011 Society for Imaging Science and Technology.*
[DOI: 10.2352/J.ImagingSci.Technol.2011.55.1.010506]

INTRODUCTION

Recently, a lot of research has been actively carried out on three-dimensional (3D) imaging and display technology due to high interest throughout the world.[1–4] Among them, holographic technology has been particularly regarded as one of the more promising and attractive approaches for creating an authentic illusion of observing volumetric objects, because holographic technology can supply very high-quality object images and accurate depth cues viewed by the human eye without any special observation devices.[5–8]

However, recording holograms of real 3D objects in the usual optical holographic system demands wave interference between the two intense laser beams with a high degree of coherence in a dark room. Therefore, this system must be kept very stable since even 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 outdoor recording.

As a partial solution for these limitations of the conventional holographic system, a new approach, to so-called computer-generated holograms (CGH), has been proposed.[9] 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 electroholographic 3D display system can be constructed.

In this approach, a ray-tracing method was originally employed for calculating the contributions at the hologram plane from each object point source. That is, an object image to be generated could 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 may lie in the vicinity of the hologram, so that it would be more suitable for various display geometries. However, this approach involves computational complexity, since it requires one-by-one calculation of the fringe pattern for each image point in the hologram sample. Thus, real-time generation of the CGH pattern for a 3D image is not achievable.[10]

To overcome this problem, a look-up table (LUT) method has been presented by Lucente.[10] In this method, an object image to be generated is also approximated as a collection of self-luminous points of light as in 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

Received Feb. 17, 2010; accepted for publication Aug. 16, 2010; published online Dec. 27, 2010.
1062-3701/2011/55(1)/010506/15/$20.00.

LUT. Then, in the process of CGH generation, fringe patterns for each point consisting of the object image can be generated just by accessing the corresponding ones from the precalculated LUT, contrary to the ray-tracing method, where fringe patterns for all object points are directly calculated on a one-by-one basis. Therefore, a great increase in computational speed can be obtained with this LUT method. But the greatest drawback of this approach is the enormous memory size required by the LUT.[11]

Recently, a novel look-up table (N-LUT) method able to significantly reduce the number of precalculated interference patterns required for generation of digital holograms has been proposed.[12] 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 precalculated, 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 precalculated 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 precalculated and stored in the N-LUT, so that the size of the N-LUT can be dramatically reduced compared to the size of the conventional LUT method.

On the other hand, ordinary 3D video images have numerous similarities between sequential video frames, so that the computation time to generate digital holograms for them can be effectively reduced by removal of redundant data just like in the case of the conventional 2D video images.[12] 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 for the 3D video images, the volume of data to be calculated can be dramatically decreased, and as a result, calculation time for the 3D video holograms can also be significantly reduced.

Moreover, a 3D image consists of depth and intensity data, contrary to the 2D image which has only intensity information. As in 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.[13,14] In other words, a 3D image has a spatial redundancy in both 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 reduction of the number of 3D image points to be involved in generation of the CGH pattern.[15] In another words, the calculation time required for CGH generation of the 3D object can be dramatically shortened.

As mentioned above, ordinary 3D moving pictures have numerous similarities between video frames; the computation time to generate digital holograms from them can be effectively reduced by the removal of redundant image data just as in the case of the conventional 2D video images. This similarity between neighboring video frames is called here temporal redundancy. Moreover adjacent interframe pixels of a 3D video image have very similar values of intensity and depth and some of them even have the exactly same values. In other words, a 3D video image has also spatial redundancy in intensity and depth data. That is, 3D video images have spatially redundant and temporally redundant data. Thus, if we employ an image and video compression technique like MPEG for the generation of the 3D video holograms, the volume of 3D video data to be calculated can be dramatically decreased, and as a result, calculation time can also be significantly reduced.[12,15]

Accordingly, in this article, a new approach for efficient generation of video hologram for 3D video images by combined use of DPCM (differential pulse code modulation) as a video compression technique, run-length encoding (RLE) method as an image compression technique, and the novel look-up table (N-LUT) is proposed. In addition, some experiments with test 3D video images are carried out and the results are compared to those of the conventional methods in terms of the number of the object points and computation time. To confirm the feasibility of the proposed method, some experiments with a 3D test object were carried out, and the results are compared to those of the conventional methods in terms of a computation speed and required memory size.

REDUNDANT DATA IN 3D VIDEO OBJECT DATA

Temporal Redundancy in 3D Video Image

Two-dimensional video is a collection of the 2D sequential images having only intensity data. In most 2D video images, neighboring pictures differ very slightly from each other. They may contain some stationary areas that will not be changed at all between the consecutive video frames. In other words, there exist some similarities between the neighboring frames called temporal redundancy of the video images, so that video data to be transmitted at each frame can be significantly reduced by exploiting this temporal redundancy. Now, the obvious conclusion is to cancel out the common parts of the two consecutive video frames and transmit only the differences between each frame. Redundancy removal between the video frames can be performed with a simple communication modulation scheme such as differential pulse code modulation (DPCM). Accordingly, it is clear that the number of the CGH patterns to be calculated for 2D video images can be greatly reduced using this image compression technique.[12]

Contrary to the 2D video case, 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 pictures in 3D video also differ slightly from each other, but in the 3D video images, intensity and depth data are simultaneously changed. For example, Figure 1 shows the adjacent four frames of 3D video images composed of a fixed house and a moving car. In these four frames, only the car is moving and the other parts of the image are not changed. Thus,

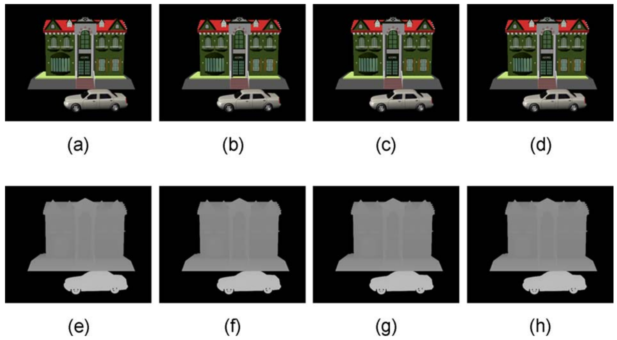

Figure 1. Adjacent four frames of a 3D video: (a)–(d) Intensity images, (e)–(h) Depth images. (a) Intensity of first frame. (b) Intensity of second frame. (c) Intensity of third frame. (d) Intensity of fourth frame. (e) Depth of first frame. (f) Depth of second frame. (g) Depth of third frame. (h) Depth of fourth frame

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 DPCM technique to both the intensity and depth data. As a result, the calculation time for generation of the 3D video images can be dramatically shortened by taking advantage of the temporally redundant data in 3D video image sequences.[12]

Spatial Redundancy of 3D Image

Spatial redundancy represents a statistical correlation between pixels within an image frame. Hence, it is also called interframe redundancy. It is well known that for most properly sampled TV (television) signals, the normalized autocorrelation coefficients along a row (or a column) with a one-pixel shift is very close to the maximum value of unity. 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 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.[13]

This spatial redundancy has been used as a concept for reduction of the amount of data to be sent in a conventional communication system. One of the data reduction methods based on this spatial redundancy concept is the run-length encoding (RLE) method.[14] 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 most useful for data sets that may contain many such runs.

On the other hand, 3D images are composed both of depth and intensity data, contrary to the 2D images, which only has intensity data. As in 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.

Spatial Redundancy in Interframe of 3D Video Image

As mentioned above, 3D video is a collection of sequential 3D images in which neighboring frames exhibit similarity. Therefore, the difference part could be extracted using the DPCM method between frames. At this time, the difference part is a 3D statistical image. Thus spatially redundant data could be extracted from this difference part using the RLE method. Therefore, if we use this temporally and spatially redundant data of 3D video images, a 3D video hologram could be efficiently generated. Accordingly, in this article, an efficient method for generating a video hologram using spatio-temporally redundant data in 3D video is proposed.

CONVENTIONAL APPROACHES FOR GENERATION OF CGHS

CGH Generation using Ray-Tracing and LUT Methods

CGH is a digital hologram generated by computing the interference pattern produced by the 3D object and the reference waves. So far, ray-tracing and LUT methods have mostly been used for generation of CGH patterns of the 3D object.

Basically, an object image to be generated can be approximated as a collection of self-luminous points of light, so that in the ray-tracing method fringe patterns for all object points are calculated by using the equations of optical diffraction and interference which are added up to obtain the whole interference pattern of an 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 is more suitable for various display geometries. However, it exhibits a computational complexity since one-by-one calculation of the fringe pattern for each image point in the hologram sample is required in this approach. Thus, real-time generation of the CGH pattern for a 3D image may not be achievable.[10]

To overcome this problem, a look-up table (LUT) method has been suggested by Lucente.[10] In this method, an object image to be generated is also approximated as a collection of self-luminous points of light as in 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 precalculated and stored in the LUT. Then, in the process of CGH generation, fringe patterns for each point consisting of the object image can be generated just by accessing the corresponding ones from the precalculated LUT, contrary to the ray-tracing method where fringe patterns for all object points are directly calculated on a one-by-one basis. Therefore, significant increase in computational speed can be obtained with this LUT method, but the method requires a massive memory space of for storing the LUT fringe patterns of all possible object image points.[11]

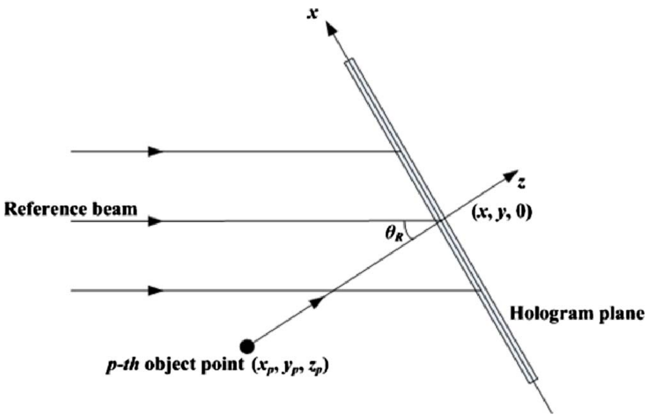

Figure 2. Computational model for generation of Fresnel hologram.

CGH Generation using N-LUT Method

Recently, to alleviate this problem of the conventional LUT method the N-LUT method was proposed.[11] In this method, the number of fringe patterns to be stored in the LUT can be dramatically reduced by employing a new concept of the principal fringe pattern (PFP). This strategy results in a vast reduction of the required memory size of the LUT.

Geometry for calculating the Fresnel hologram of an object image is shown in Figure 2. Here, the location coordinate of the *p-th* 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, a_p, φ_p, respectively. The CGH pattern is also assumed to be positioned at the plane $z=0$.[11]

Here we can treat a 3D object as a set of image planes discretely sliced along the z-direction, in which each image plane with a fixed depth is approximated as a collection of self-luminous object points of light. In the N-LUT method, only the fringe patterns of the center points on each image plane are precalculated, called principal fringe patterns (PFPs), and stored in the LUT. Therefore, the unit magnitude principal fringe pattern for the center object point $(0,0,z_p)$ on the image plane with a depth of z_p, $T(x,y;z_p)$ can be defined as Eq. (1).[16]

$$T(x,y;z_p) \equiv \frac{1}{r_p}\cos[kr_p + kx\sin\theta_R + \phi_p]. \qquad (1)$$

The wave number k is defined as $k=2\pi/\lambda$, in which λ is the free-space wavelength of the light and θ_R mean the incident angle of the reference beam. Here, the oblique distance r_p between the *p-th* object point of (x_p, y_p, z_p) and the point on the hologram plane of $(x,y,0)$ is given by Eq. (2).

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

Then, the fringe patterns for other object points on each image plane can be obtained by simply shifting this precalculated 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 precalculated PFPs of Eq. (1) as shown in Eq. (3),

$$I(x,y) = \sum_{p=1}^{N} a_p T(x-x_p, y-y_p; z_p). \qquad (3)$$

Where N is the number of object points. Equation (3) 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 principal fringe pattern and adding them, whereas in the N-LUT method, by multiplying the amplitudes of each object point to the corresponding PFPs precalculated 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 finally be generated.

Here, the phase of the object point φ_p is set to zero for all points in this approach. This phase information of the object point is invisible to the viewer and important only in the interactions among image points. If the discretization step of the object image is much larger than the spot size of the display system, no overlap occurs between the image points. In this case, as with the holographic display, restricting the phase values of the object points to the specific values does not limit the size, depth, or quality of the reconstructed object image. Therefore, the relative phases of the object points can arbitrarily be set to be zero.[10]

CGH Generation using Temporal Redundancy and N-LUT Method (TR-N-LUT Method)

Recently, a fast generation method of CGH patterns for the 3D video images by combined use of the DPCM (differential pulse code modulation) as a video compression technique and the novel look-up table (N-LUT) has been proposed.[12] 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 nth frame on the hologram plane can be obtained by superposition of the object wave fronts as shown in Eq. (4).

$$O_n(x,y) = \sum_{p=1}^{N_n} \frac{a_p}{r_p}\exp[j(kr_p + \phi_p)]. \qquad (4)$$

Where N_n represent the number of object points of 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 $\exp(j\omega t)$ is not included explicitly in Eq. (4).

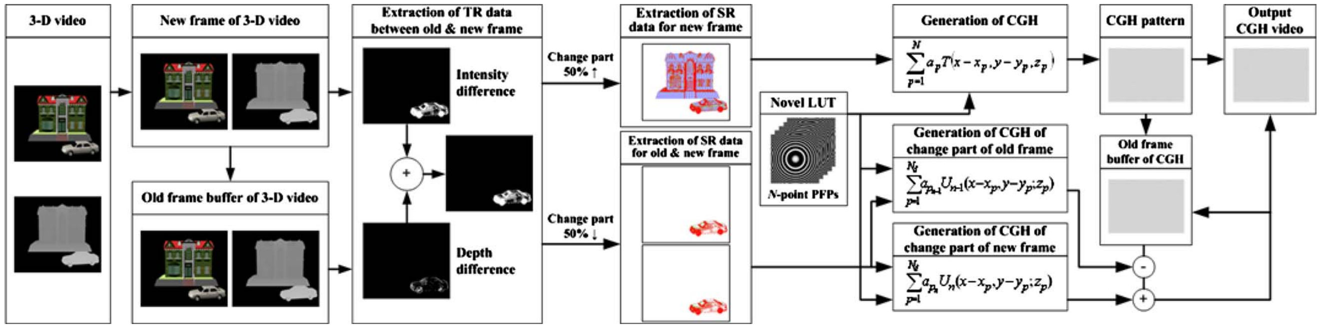

Figure 3. Block diagram of the proposed method for generation of video hologram for the 3D video

In this approach, the N-LUT method is used to calculate the hologram patterns.[11] 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 principal fringe patterns (PFPs) are stored, so that a unit 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. (1) as mentioned above.

In this method, first, 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 the 3D data for 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 vanished part of the first frame from the CGH pattern of the first frame and adding the CGH pattern for the new part of the second frame to the CGH pattern of the first frame as shown in Eq. (5).

$$I_n(x,y) = I_{n-1}(x,y) - \sum_{p=1}^{N_d} a_{p_{n-1}} U_{n-1}(x-x_p, y-y_p; z_p) + \sum_{p=1}^{N_d} a_{p_n} U_n(x-x_p, y-y_p; z_p) \tag{5}$$

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 as shown in Eq. (6).

$$U_n(x,y;z_p) = \begin{cases} T(x,y;z_p) & \text{for changed part} \\ 0 & \text{for unchanged part} \end{cases}. \tag{6}$$

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 may be repeated for all of the video frames.

CGH Generation using Spatial Redundancy and N-LUT Method (SR-N-LUT Method)

Recently, a method for fast generation of CGH patterns for 3D images by combined use of RLE (run-length encoding) as an image compression technique and the novel look-up table (N-LUT) is proposed.[15] In this method, spatially redundant data of a 3D image is analyzed using the RLE method and the data are re-grouped into the N-point redundancy map according to the number of the neighboring object points having the same 3D value. N-point PFPs corresponding to the N-point redundancy maps are calculated by shifting and adding the one-point PFP of the conventional N-LUT. Finally, the CGH pattern of the 3D object is calculated with these precalculated N-point PFPs.

Here, the one-point PFP is defined as Eq. (1) mentioned above in the conventional N-LUT.[11] And, the two-point PFP for two adjacent object points with unit magnitude and depth z_p can be expressed by Eq. (7).

$$T_2(x,y,z_p) \equiv T(x,y;z_p) + T(x-d,y;z_p), \tag{7}$$

where d is the discretization step separating adjacent points.[11] 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 Eq. (8).

$$T_n(x,y;z_p) \equiv \sum_{k=1}^{n} T[x-(k-1)d,y;z_p]. \tag{8}$$

Therefore, N adjacent object points can be displayed using the spatial redundancy map and the N-point PFP. That is, for the case of N adjacent object points having the same intensity and depth values, N calculation processes are needed in the conventional N-LUT method, whereas in this method only one calculation process is needed. Therefore, the calculation time for generation of the CGH pattern of the 3D object can be significantly reduced.

PROPOSED METHOD

Figure 3 shows an overall block-diagram of the proposed method to generate digital video holograms for the 3D moving pictures using spatio-temporal redundant data. The proposed method is largely consisted of four steps. First, intensity and depth data of the current frame of the 3D video are extracted and compared with those of the previous frame using DPCM method. Second, in case the difference of intensity and depth data between these two consecutive video frames is larger than 50%, spatially redundant components of the intensity and depth data in current frame are analyzed by the RLE method, and they are regrouped into the N-point redundancy map according to the number of the neighboring object points having the same 3D value. Third, the CGH pattern for the current video frame is calculated with precalculated N-point PFPs. Lastly, 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, in case the difference between two frames is smaller than 50%, spatially redundant components of the intensity and depth data in the previous and the current frame corresponding to the extracted temporal redundancy map are analyzed using the RLE method, and they are regrouped into the N-point redundancy map according to the number of the neighboring object points having the same 3D value. After that, 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.

Computation of 3D Video Holograms using Spatio-temporally Redundant Data and N-LUT Method

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, a_p, φ_p, respectively. Thus, the complex amplitude $O_n(x,y)$ of nth frame on the hologram plane can be given by Eq. (4) above.

In this paper, the TR-N-LUT method is used to calculate the hologram patterns between frames of 3D video.[12] In this method, temporally redundant data of 3D video image is analyzed using the DPCM method. The SR-N-LUT method is used to calculate the hologram patterns for these extracted temporal redundant data.[15] In this method, spatially redundant data of the 3D image are analyzed by using the RLE method and they are re-grouped into the N-point redundancy map according to the number of the neighboring object points having the same 3D value. N-point PFPs corresponding to the N-point redundancy maps can be generated by Eq. (8).

In the proposed method, first, the CGH pattern for the first frame of the 3D video images is generated with the SR-N-LUT method because all points of first frame are changing parts, and the calculated CGH pattern, as well as 3D data of the first frame, are 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 using SR-N-LUT method, and it 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 using SR-N-LUT method 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 vanished part of the first frame from the CGH pattern of the first frame and adding the CGH pattern for the new part of the second frame to the CGH pattern of the first frame as shown in Eq. (5). Moreover, $U_n(x,y;z_p)$ means the N-point PFPs of the nth frame as shown in Eq. (6). Then, the calculated CGH pattern of the second frame $I_2(x,y)$ is moved to the CGH video output as well as stored in the previous frame buffer of the CGH. These processes may be repeated for all of the video frames.

In this article, five types of 3D video images are used to confirm the proposed method for various cases. Each 3D video has a 50 frame images that are computationally generated, and each frame has a resolution of $640\times480\times256$ pixels. These 3D video images show the sequential front views of a 3D scene with a tank moving in front of a fixed house. Figure 4 shows five frames of the intensity and depth images of the each test 3D videos. In this paper, each of the 3D videos is designated "Scene I" to "Scene V," respectively. In Scene I, the front tank is moving from right to left as shown in Fig. 4(a). In Scene II, the front tank is moving forward from the rear as shown in Fig. 4(b). In "Scene III," the front tank is moving from front to back as shown in Fig. 4(c). In "Scene IV," the front tank is rotating as shown in Fig. 4(d). In "Scene V," the front tank is moving from right to left as shown in Fig. 4(e). In Figs. 4(a)–4(d) the camera is fixed but in Fig. 4(e) the camera is panning from left to right.

Some of the parts changing between the previous and current frames in intensity and depth data are shown in Figure 5. As can be seen, Figs. 5(a)–5(d) includes only some tank parts from the 3D image because only tank parts are changing. Fig. 5(e) shows big differences in the 3D scenes because the camera is moved rapidly in the horizontal direction; thus all parts of the scene are changing.

Figure 6 shows the selected spatial redundancy maps extracted from horizontal scanning of the first frame and the changing part of the test 3D video of Figs. 4 and 5 using the RLE method. It is noted that both horizontal and vertical scanning methods can be used for extraction of spatial redundancy maps from the test 3D object, but here in Fig. 6 the horizontal scanning method is optionally shown. 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 and blue colors mean that two or three adjacent object-points, respectively, have the same intensity and depth values. In addition, the white color means the object points correspond to the "don't care condition." Using this spatial redundancy map, CGH patterns can be also generated.

Reconstruction of 3D Video Holograms

Here in this article CGH patterns with 1600×1600 pixels are generated using the intensity and depth data of the test 3D videos of Fig. 4. Each pixel size of the CGH pattern is given by $10\times10\ \mu m^2$. Moreover, the horizontal and vertical discretization step is set to be 30 μm. Therefore, the

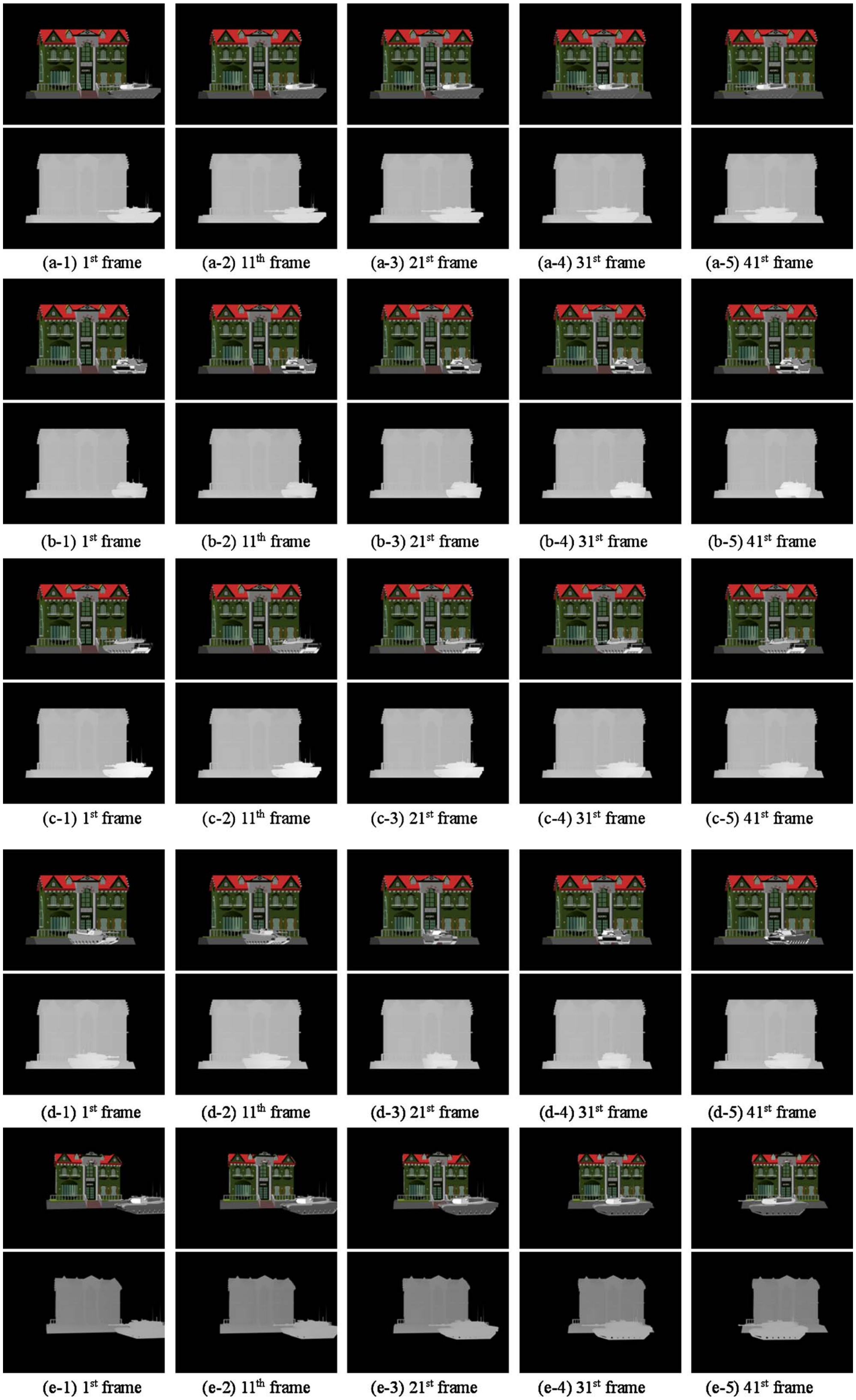

Figure 4. Fig. 4 Five frames of the intensity and depth images of the test 3D videos (a) Scene I (b) Scene II (c) Scene III (d) Scene IV (e) Scene V.

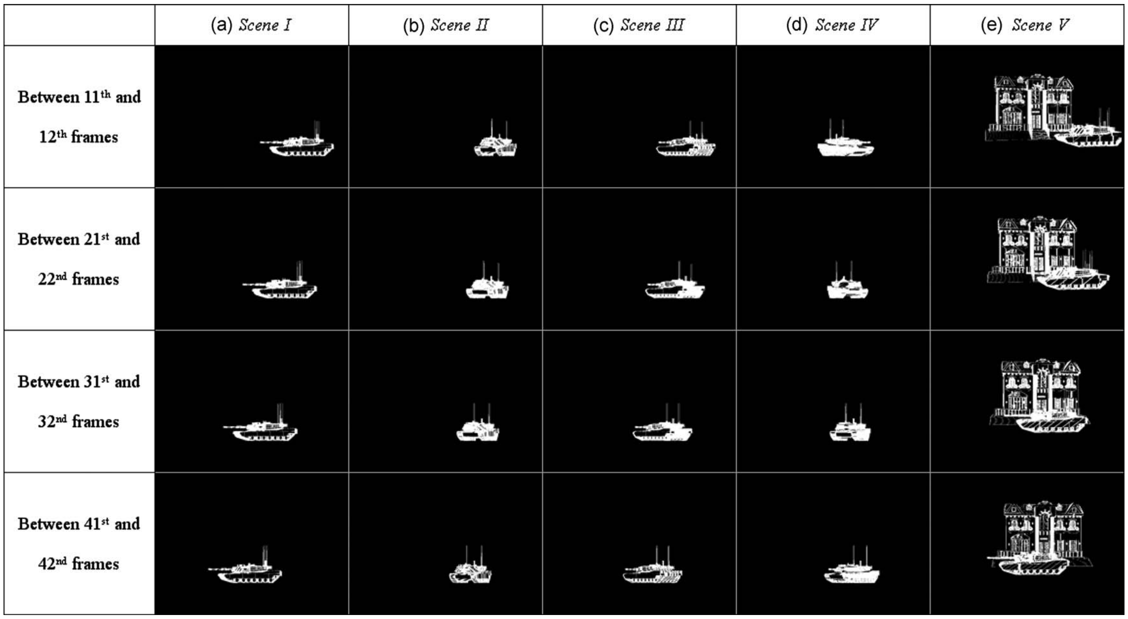

Figure 5. Changed parts of the intensity and depth data between the previous and current 3D video frames (a) Scene I (b) Scene II (c) Scene III (d) Scene IV (e) Scene V.

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 500×3 pixels$=1500$ pixels horizontally and 400×3 pixels$=1200$ pixels vertically. Thus, the total resolution of the PFP should become to be $3100(1600+1500) \times 2800(1600+1200)$ pixels.

These CGH patterns are digitally reconstructed, and these five sets of results are shown in Figure 7. As can be seen in Fig. 7, in all cases object images are found to be successfully reconstructed.

Number of Objects Points and Calculation Time for Each Method

Figure 8 shows the number of the calculated object points in generation of CGH patterns for each method. Figures 9 and 10 also show the comparison results for the calculation times needed for one object frame and one object point. Table I shows the number of calculated object points, calculation time for one frame, calculation time for one point, and total memory size required for each method. In the experiment, a PC system with an Intel Core^{TM}2 Quad operating at 2.5 GHz, a main memory of 4 GB and an operating system of Microsoft Windows XP as well as MATLAB 2009 are used.

As we can see in Table I, the average original numbers of calculated object points for each scene are 90 579, 88 851, 89 841, 88 517, 60 068, respectively. In Scene I, average calculated object points can be reduced by 15.6% using the conventional TR-NLUT method. By applying the proposed method, average number of calculated object points can be reduced by 12.4% and 11.5% for horizontal scanning and 12.9% and 12.2% for vertical scanning. That is, in Scene I horizontal scanning is more efficient than vertical scanning. In Scene II, average calculated object points can be reduced by 14.7% using the conventional TR-NLUT method. By applying the proposed method, the average number of calculated object points can be reduced by 11.6% and 10.6% for horizontal scanning and 12.2% and 11.4% for vertical scanning. That is, in Scene II horizontal scanning is more efficient than vertical scanning. In Scene III, average calculated object points can be reduced by 16.3% using the conventional TR-NLUT method. By applying the proposed method, the average number of calculated object points can be reduced by 12.5% and 11.4% for horizontal scanning and 13.4% and 12.6% for vertical scanning. That is, in Scene III horizontal scanning is more efficient than vertical scanning. In Scene IV, average calculated object points can be reduced by 18.8% using the conventional TR-NLUT method. By applying the proposed method, the average number of calculated object points can be reduced by 13.8% and 12.2% for horizontal scanning and 14.5% and 13.2% for vertical scanning. That is, in Scene IV horizontal scanning is more efficient than vertical scanning. In Scene V, the difference between most frames is greater than 50%. Therefore, if the difference between frames is larger than 50%, we use the conventional method to generate the CGH pattern. That is, the number of object points to be calculated in the proposed method is exactly the same as with the conventional N-LUT method. But, the difference of some frames is smaller than 50%, where we use the proposed method to generate the CGH pattern for these frames. Therefore, the average calculated object points can be reduced by 91.8% using the conventional TR-NLUT method. And by applying the proposed method, the average number of calculated object points can be reduced by 67.5% and 58.2% for horizontal scanning and 64.5% and 55.2% for vertical scanning. That is, in Scene V vertical scanning is more efficient than horizontal scanning. These results reveal that the average number of object points

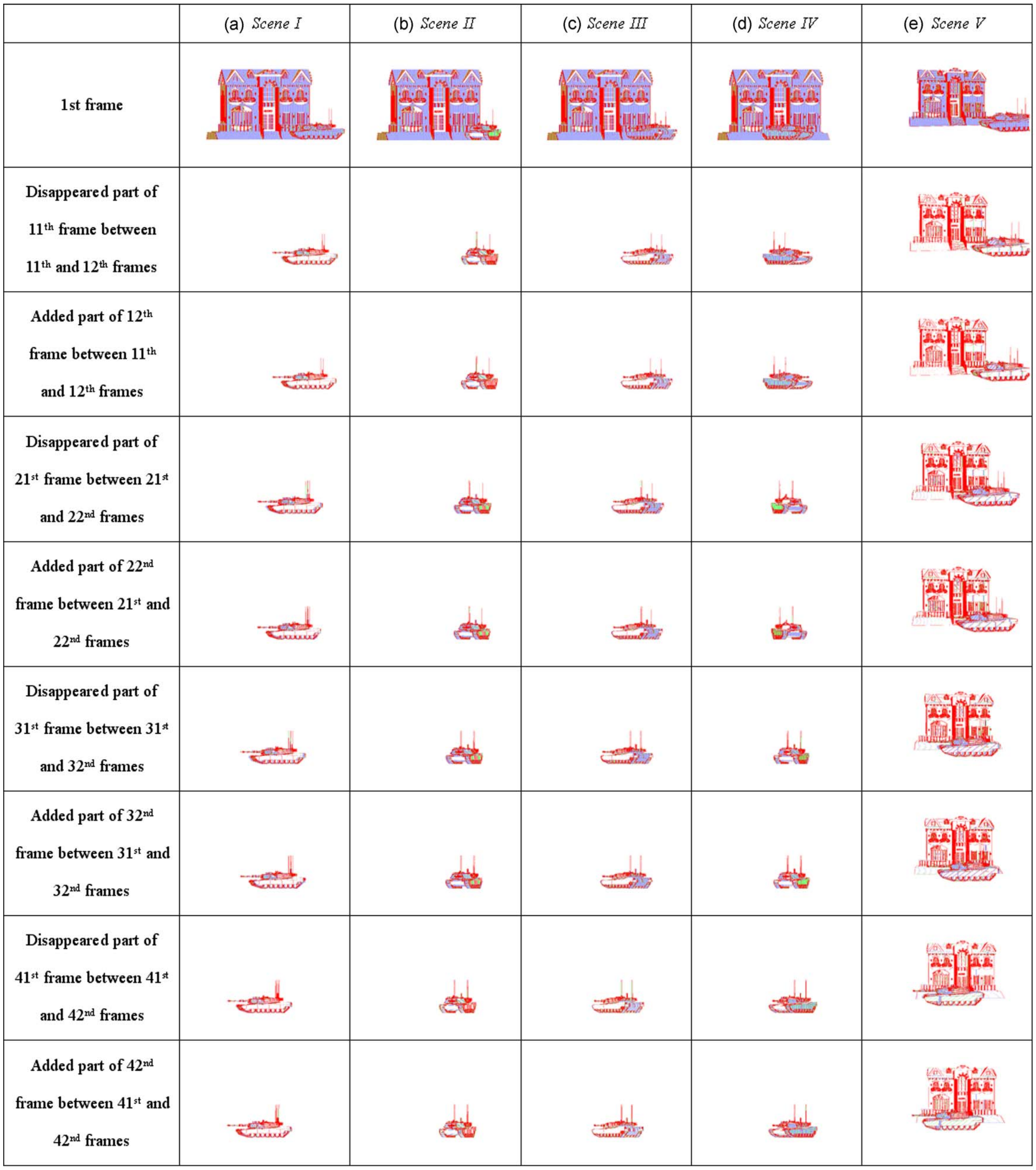

Figure 6. Three-point spatial redundancy maps extracted from horizontal scanning of the first frame and changing part of test 3D video (a) Scene I, (b) Scene II, (c) Scene III, (d) Scene IV, (e) Scene V.

to be calculated in these methods can be significantly reduced compared to the conventional N-LUT and TR-N-LUT method.

Average calculation times for one object point for horizontal scanning in Scene I are estimated to be 7.517, 1.167, 0.929, and 0.859 ms, respectively. That is, average calculation times are reduced by 15.5%, 12.4%, and 11.4% using the proposed method, respectively. For vertical scanning, average calculation times for one object point are estimated to be 0.971 and 0.912 ms, respectively. Thus, average calculation times are reduced by 12.9% and 12.1% using the proposed method, respectively. For Scene II, average calculation times for one object point for horizontal scanning are estimated to be 7.517, 1.106, 0.869, and 0.799 ms, respectively. That is, average calculation times are reduced by 14.7%, 11.6%, and 10.6%, respectively, using the proposed method. For vertical scanning, average calculation times for one object point are estimated to be 0.916 and 0.858 ms, respectively. Thus, average calculation times are reduced by 12.2% and 11.4%, respectively, using the proposed method. For Scene III, average calculation times for one object point for horizontal scanning are estimated to be 7.517, 1.224, 0.940, and 0.854 ms, respectively. That is, average calculation times are reduced by 16.3%, 12.5%, and 11.4%, respectively, using the proposed method. For vertical scanning, average calculation times for one object point are estimated to be 1.008 and 0.943 ms, respectively. Thus, average calculation times are reduced by 13.4% and 12.6%, respectively, using the pro-

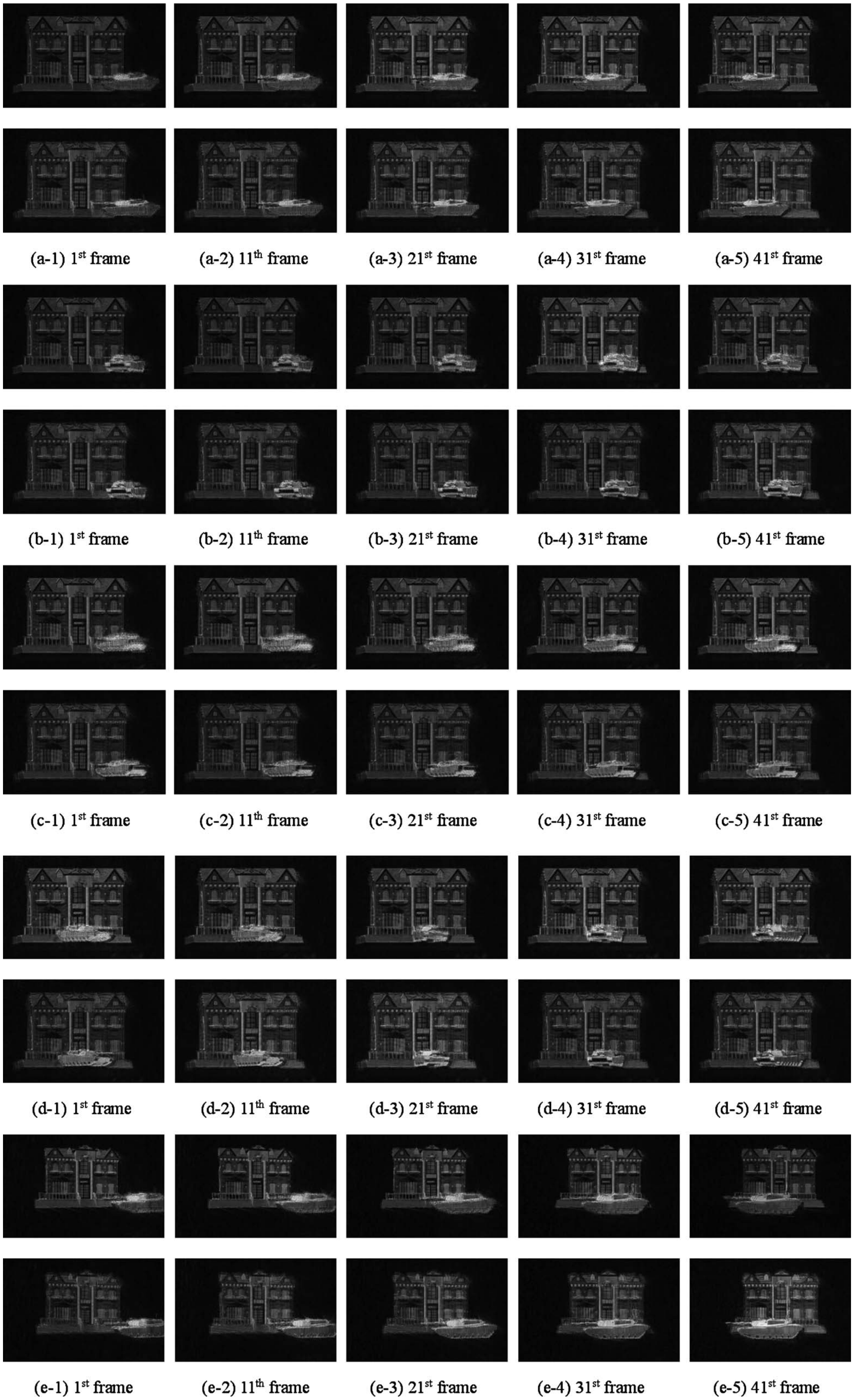

Figure 7. Computationally reconstructed video images for each method (a) Scene I, (b) Scene II, (c) Scene III, (d) Scene IV, (e) Scene V.

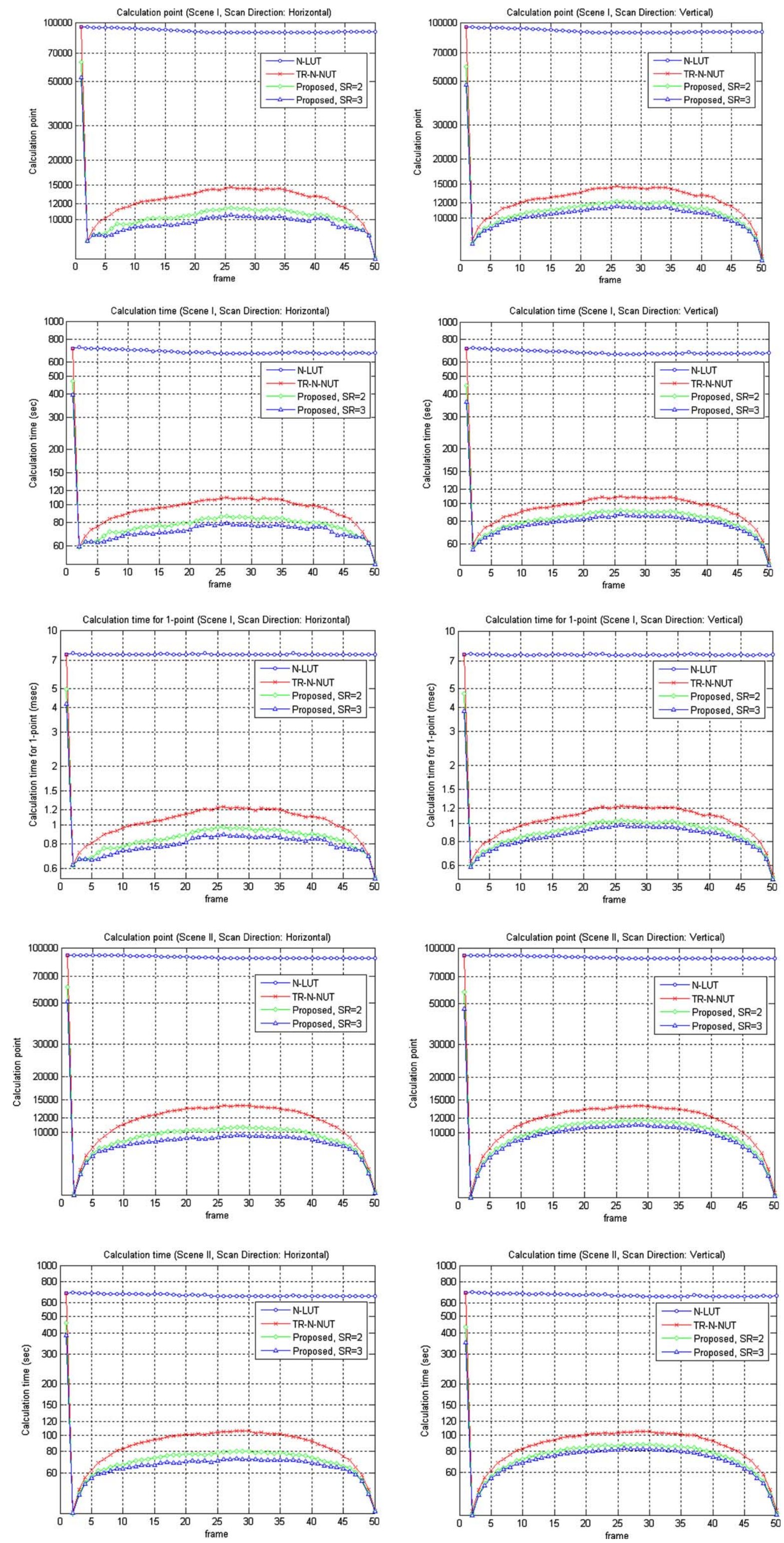

Figure 8. Comparison results of each method in terms of the number of the calculated object points

posed method. For Scene IV, average calculation times for one object point for horizontal scanning are estimated to be 7.517, 1.409, 1.033, and 0.918 ms, respectively. That is, average calculation times are reduced by 18.7%, 13.7%, and 12.2%, respectively, using the proposed method. For vertical scanning, average calculation times for one object point are estimated to be 1.093 and 0.994 ms, respectively. Thus, average calculation times are reduced by 14.5% and 13.2%, respectively, using the proposed method. For Scene V, average calculation times for one object point for horizontal scanning are estimated to be 7.517, 6.910, 5.077, and 4.371 ms, respectively. That is, average calculation times are reduced by 91.9%, 67.5%, and 58.1%, respectively, using the proposed method. For vertical scanning, average calculation

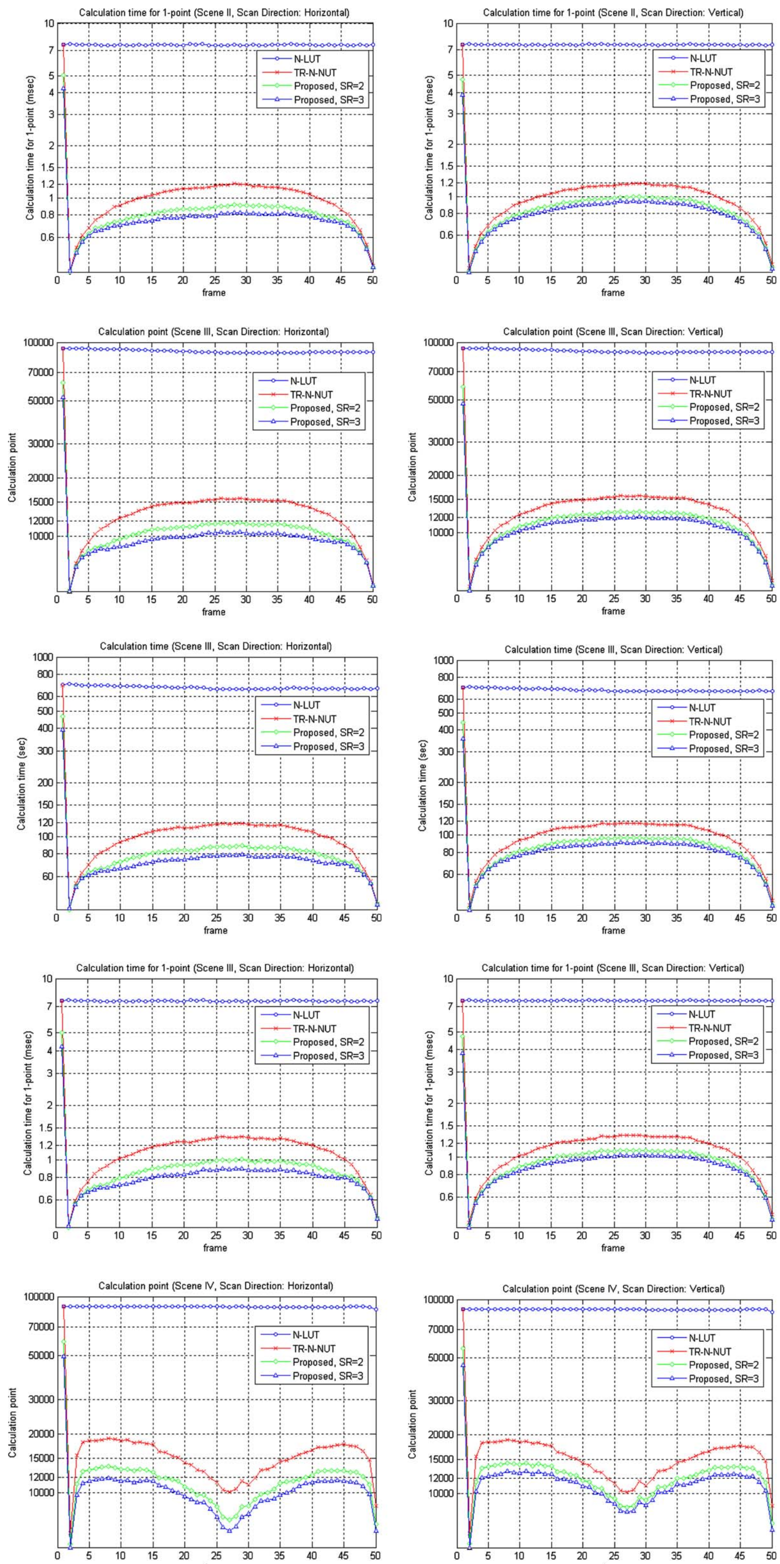

Figure 9. Comparison results of each method in terms of the calculation times for one frame.

times for one object point are estimated to be 4.851 and 4.146 ms, respectively. Thus, average calculation times are reduced by 64.5% and 55.2%, respectively, using the proposed method. That is, the calculation time is reduced for Scene V in spite of the camera moving rapidly. These results also show that the average calculation times for one object point in these methods can be considerably reduced compared to the conventional N-LUT and TR-N-LUT method.

Accordingly, we conclude here that for the normal 3D video cases involving a slow image change, the number of object points to be calculated and the calculation time required for generation of 3D hologram patterns can be sig-

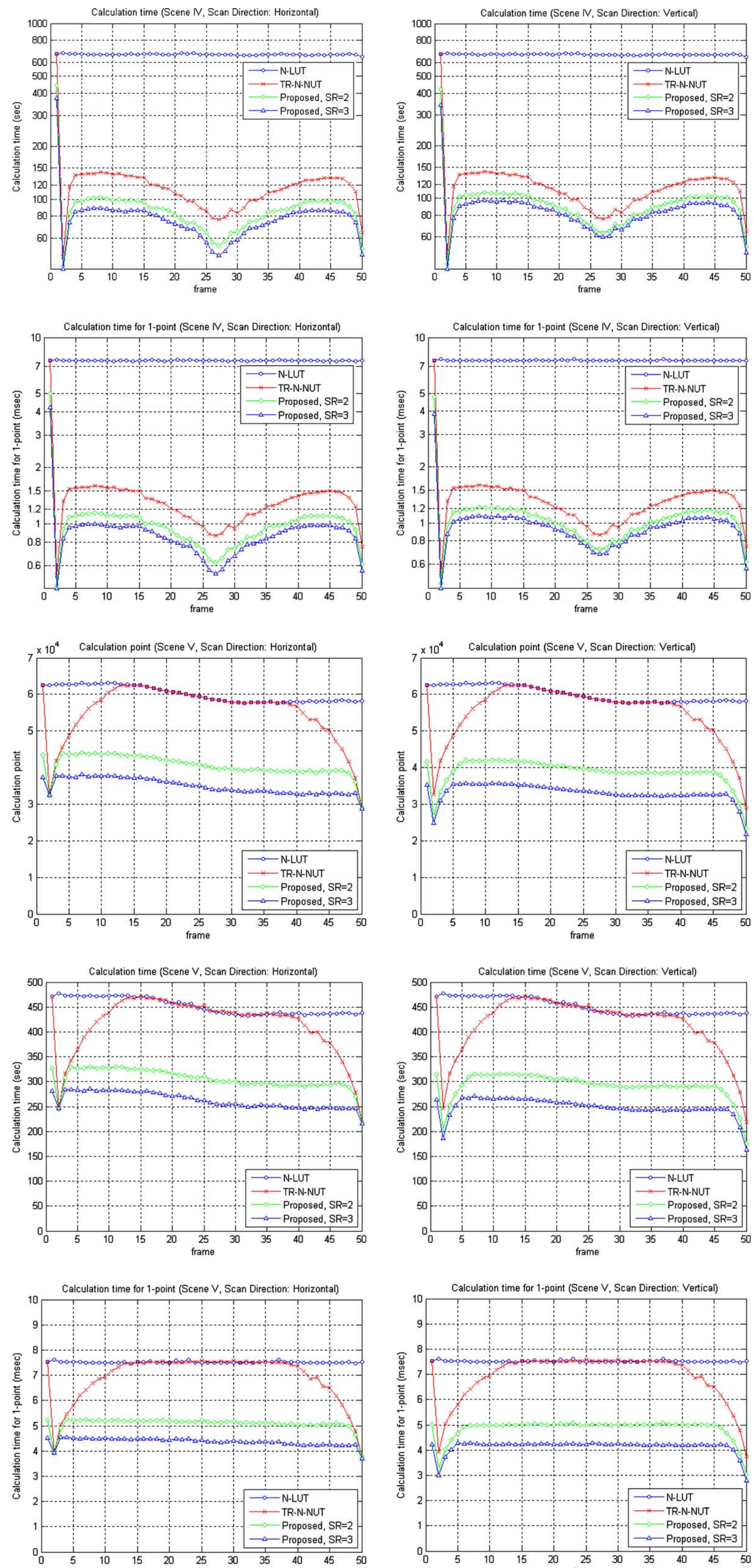

Figure 10. Comparison results of each method in terms of the calculation times for one point.

Table I. Calculated point, calculation time for 1-frame, calculation time for 1-point and total memory size for each method

		Conventional method		Proposed method	
		N-LUT	TR-N-LUT	Two-point	Three-point
Average number of calculated point	Scene I(H)	90579(100%)	14100(15.6%)	11211(12.4%)	10372(11.5%)
	Scene I(V)			11718(12.9%)	11024(12.2%)
	Scene II(H)	88851(100%)	13096(14.7%)	10279(11.6%)	9456(10.6%)
	Scene II(V)			10837(12.2%)	10168(11.4%)
	Scene III(H)	89841(100%)	14649(16.3%)	11244(12.5%)	10219(11.4%)
	Scene III(V)			12055(13.4%)	11298(12.6%)
	Scene IV(H)	88517(100%)	16598(18.8%)	12173(13.8%)	10815(12.2%)
	Scene IV(V)			12876(14.5%)	11726(13.2%)
	Scene V(H)	60068(100%)	55172(91.8%)	40571(67.5%)	34953(58.2%)
	Scene V(V)			38760(64.5%)	33171(55.2%)
Average computation time for one frame (s)	Scene I(H)	680.878(100%)	106.04(15.6%)	84.316(12.4%)	77.975(11.5%)
	Scene I(V)			88.099(12.9%)	82.745(12.2%)
	Scene II(H)	667.883(100%)	98.492(14.7%)	77.307(11.6%)	71.088(10.6%)
	Scene II(V)			81.472(12.2%)	76.319(11.4%)
	Scene III(H)	675.330(100%)	110.168(16.3%)	84.56(12.5%)	76.826(11.4%)
	Scene III(V)			90.627(13.4%)	84.806(12.6%)
	Scene IV(H)	665.367(100%)	124.788(18.8%)	91.52(13.8%)	81.298(12.2%)
	Scene IV(V)			96.79(14.5%)	88.013(13.2%)
	Scene V(H)	451.531(100%)	414.877(91.9%)	305.082(67.6%)	262.732(58.2%)
	Scene V(V)			291.328(64.5%)	249.055(55.2%)
Average computation time for one object point (ms)	Scene I(H)	7.517(100%)	1.167(15.5%)	0.929(12.4%)	0.859(11.4%)
	Scene I(V)			0.971(12.9%)	0.912(12.1%)
	Scene II(H)	7.517(100%)	1.106(14.7%)	0.869(11.6%)	0.799(10.6%)
	Scene II(V)			0.916(12.2%)	0.858(11.4%)
	Scene III(H)	7.517(100%)	1.224(16.3%)	0.940(12.5%)	0.854(11.4%)
	Scene III(V)			1.008(13.4%)	0.943(12.6%)
	Scene IV(H)	7.517(100%)	1.409(18.7%)	1.033(13.7%)	0.918(12.2%)
	Scene IV(V)			1.093(14.5%)	0.994(13.2%)
	Scene V(H)	7.517(100%)	6.910(91.9%)	5.077(67.5%)	4.371(58.1%)
	Scene V(V)			4.851(64.5%)	4.146(55.2%)

nificantly reduced by combined use of temporally redundant data and spatially redundant data of 3D video, and N-LUT techniques. It can be also concluded here that for the special 3D video cases having a fast image change, the number of object points to be calculated and the calculation time required for generation of 3D hologram patterns can be significantly reduced by combined use of temporally redundant data and spatially redundant data of 3D video and N-LUT techniques.

CONCLUSIONS

In this article, a new method for fast computation of CGH patterns for 3D video images has been proposed using temporally redundant data in interframe of 3D video images, spatially redundant data in intraframe of 3D video images and N-LUT techniques. By using the DPCM algorithm, temporally redundant interframe data in the 3D video images were removed; using the RLE algorithm spatially redundant intraframe data in the 3D video images were removed. Then CGH patterns for these compressed video images were generated with the N-LUT technique. Good experimental results with 3D test moving pictures finally confirmed the feasibility of the proposed method for fast generation of CGH patterns for 3D videos.

ACKNOWLEDGMENTS

This work was supported by the Second Brain Korea 21 Project.

REFERENCES

[1] K. Iizuka, "Welcome to the Wonderful World of 3D: Introduction, Principles and History", Opt. Photonics News **17**, 42 (2006).
[2] S.-C. Kim and E.-S. Kim, "A new liquid crystal display-based polarized stereoscopic projection method with improved light efficiency", Opt. Commun. **249**, 51 (2005).
[3] S.-C. Kim and E.-S. Kim, "A novel configuration of LCD projectors for efficient orthogonal polarization of two projected views", Opt. Commun. **266**, 55 (2006).
[4] S.-C. Kim, P. Sukhbat, and E.-S. Kim, "Generation of three-dimensional integral images from a holographic pattern of 3D objects", Appl. Opt. **47**, 3901 (2008).
[5] C. J. Kuo and M. H. Tsai, *Three-Dimensional Holographic Imaging* (John Wiley and Sons, New York, 2002).
[6] U. Schnars and W. Jueptner, *Digital Holography-Digital Hologram Recording, Numerical Reconstruction, and Related Techniques* (Springer, Berlin, 2004).
[7] T.-C. Poon, *Digital Holography and Three-dimensional Display* (Springer, Berlin, 2007).
[8] H. Kang, T. Yamaguchi, H. Yoshikawa, S.-C. Kim, and E.-S. Kim, "Acceleration method of computing a compensated phase-added stereogram on a graphic processing unit", Appl. Opt. **47**, 5784 (2008).
[9] A. W. Lohmann and D. P. Paris, "Binary Fraunhofer holograms generated by computer", Appl. Opt. **6**, 1739 (1967).
[10] M. Lucente, "Interactive computation of holograms using a look-up table", J. Electron. Imaging **2**, 28 (1993).
[11] S.-C. Kim and E.-S. Kim, "Effective generation of digital holograms of 3D objects using a novel look-up table method", Appl. Opt. **47**, D55 (2008).
[12] S.-C. Kim, J.-H. Yoon, and E.-S. Kim, "Fast generation of 3D video holograms by combined use of data compression and look-up table techniques", Appl. Opt. **47**, 5986 (2008).
[13] R. C. Dorf, *Electrical Engineering Handbook*, 2nd ed. (CRC Press, Boca Raton, FL, 1997).
[14] J. Higgins, *Introduction to SNG and ENG Microwave* (Butterworth-Heinemann, United Kingdom, 2004).
[15] S.-C. Kim and E.-S. Kim, "Fast computation of hologram patterns of a 3D object using run-length encoding and novel look-up table methods", Appl. Opt. **48**, 1030 (2009).
[16] P. Hariharan, *Optical Holography; Principles, Techniques, and Applications*, Cambridge Studies in Modern Optics (Cambridge University Press, Cambridge, England, 1996).