



**HAL**  
open science

# Fast generation of complex modulation video holograms using temporal redundancy compression and hybrid point-source/wave-field approaches

Antonin Gilles, Patrick Gioia, Rémi Cozot, Luce Morin

## ► To cite this version:

Antonin Gilles, Patrick Gioia, Rémi Cozot, Luce Morin. Fast generation of complex modulation video holograms using temporal redundancy compression and hybrid point-source/wave-field approaches. SPIE Optics + Photonics 2015, Aug 2015, San Diego, United States. <10.1117/12.2187045>. <hal-01229472v2>

**HAL Id: hal-01229472**

**<https://hal.science/hal-01229472v2>**

Submitted on 13 Dec 2016

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

# Fast generation of complex modulation video holograms using temporal redundancy compression and hybrid point-source/wave-field approaches

Antonin Gilles<sup>1\*</sup>

Patrick Gioia<sup>1,2</sup>

Rémi Cozot<sup>1,3</sup>

Luce Morin<sup>1,4</sup>

<sup>1</sup> IRT b<>com  
Cesson-Sévigné  
France

<sup>2</sup> Orange Labs  
Rennes  
France

<sup>3</sup> University of Rennes 1  
Rennes  
France

<sup>4</sup> INSA Rennes  
Rennes  
France

## Abstract

The hybrid point-source/wave-field method is a newly proposed approach for Computer-Generated Hologram (CGH) calculation, based on the slicing of the scene into several depth layers parallel to the hologram plane. The complex wave scattered by each depth layer is then computed using either a wave-field or a point-source approach according to a threshold criterion on the number of points within the layer. Finally, the complex waves scattered by all the depth layers are summed up in order to obtain the final CGH. Although outperforming both point-source and wave-field methods without producing any visible artifact, this approach has not yet been used for animated holograms, and the possible exploitation of temporal redundancies has not been studied. In this paper, we propose a fast computation of video holograms by taking into account those redundancies. Our algorithm consists of three steps. First, intensity and depth data of the current 3D video frame are extracted and compared with those of the previous frame in order to remove temporally redundant data. Then the CGH pattern for this compressed frame is generated using the hybrid point-source/wave-field approach. The resulting CGH pattern is finally transmitted to the video output and stored in the previous frame buffer. Experimental results reveal that our proposed method is able to produce video holograms at interactive rates without producing any visible artifact.

**Keywords :** Computer-Generated Hologram, Color holography, Video holography, Three-dimensional imaging

## 1 INTRODUCTION

Holography is often considered as the most promising 3D visualization technology, since it can provide the most realistic and natural three-dimensional illusion to the naked eye. Indeed, it provides complete human depth cues without the need for special viewing devices and without causing eye-strain [1]. Holograms can be generated by computer calculation techniques. These techniques calculate Computer-Generated Holograms (CGH) of synthetic or existing scenes by simulating the propagation of light scattered by the scene towards the hologram plane. Thanks to its attractive visualization properties, CGH may have application in the field of videoconferencing or telepresence systems. However, because of the huge calculation burden, real-time CGH computation is still very challenging. In this paper, we propose a new method which aims at reducing the CGH calculation time for 3D video in order to get closer to real-time computation.

CGH computation techniques usually sample 3D scenes by a set of primitives and calculate light propagation as the sum of complex light waves scattered by each primitive. Two approaches for CGH computation are commonly used: the point-source approach, which samples scenes by a collection of points, and the wave-field approach, which samples scenes by a collection of planar segments.

In the point-source approach, 3D scenes are populated by a collection of self-luminous points. The complex waves scattered by the points are computed using the monochromatic spherical light wave equation and summed up together in order to get the wave scattered by the entire scene, as shown in Figure 1a. This approach presents the advantage of not imposing any restriction on the scene geometry. However, its computational complexity is very high since it requires one calculation per point of the scene per pixel of the hologram. Moreover, to produce shapes that appear solid and continuous, the scene needs to be sampled at very high densities, making the CGH computation prohibitively slow. In order to reduce the computational complexity, several methods have been proposed, including geometric symmetry [2], difference and recurrence formulas [3, 4], image holograms [5, 6],

---

\*This work has been achieved within the Institute of Research and Technology b<>com, dedicated to digital technologies. It has been funded by the French government through the National Research Agency (ANR) Investment referenced ANR-A0-AIRT-07. Authors can be reached at {antonin.gilles, patrick.gioia, remi.cozot, luce.morin}@b-com.com.

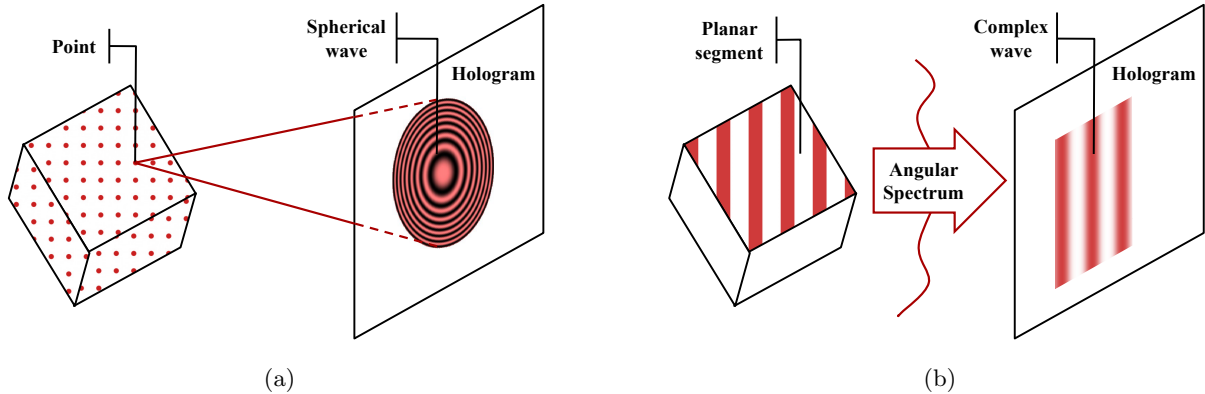


Figure 1: Two approaches are commonly used for CGH computation: (a) the point-source approach, which samples 3D scenes by a collection of self-luminous points and calculate light propagation as the sum of spherical waves scattered by each point, and (b) the wave-field approach, which samples 3D scenes by a collection of self-luminous planar segments and calculate light propagation using diffraction formulas.

wave-front recording planes [7, 8, 9], look-up tables [10, 11], interframe and interline redundancy reduction [12, 13], using GPU hardware [14, 15], and special purpose hardware [16, 17].

In the wave-field approach, 3D scenes are populated by a collection of self-luminous planar segments. The complex waves scattered by the segments are computed using diffraction formulas such as the Angular Spectrum [18] and summed up together in order to get the wave scattered by the entire scene, as shown in Figure 1b. The Angular Spectrum can be calculated using the Fast Fourier Transform (FFT) algorithm twice. The computation of the complex wave scattered by each segment is therefore more time-consuming than the calculation of the spherical light wave scattered by a single point. However, complex waves scattered by scene points located within a single planar segment are calculated all at once using the Angular Spectrum. Therefore, this approach is more efficient than the point-source approach when objects in a scene consist of large planar segments containing many points. However, when the scene geometry contains complex shapes, a large number of small planar segments containing only one or a few points are needed to sample it, making the wave-field approach less efficient than the point-source approach. In order to reduce the CGH calculation time, several methods have been proposed, including the use of analytic expression of the angular spectrum [19, 20, 21, 22, 23] and color-space conversion [24, 25].

These two approaches complement one another: while the point-source approach is very efficient when the scene contains complex shapes, such as trees or human bodies, the wave-field approach is more efficient when objects in the scene consist of large planar surfaces, such as roads or buildings. However, most real 3D scenes contain both complex shapes and planar surfaces. As a consequence, these two approaches are rarely fully efficient for computing holograms of real scenes in their entirety. In order to overcome these limitations, a new hybrid point-source/wave-field approach has recently been proposed [26]. In this approach, the scene is sliced into several depth layers parallel to the hologram plane. The complex wave scattered by each depth layer is then computed using either a wave-field or a point-source approach according to a threshold criterion on the number of points within the layer. Finally, the complex waves scattered by all the depth layers are summed up in order to obtain the final CGH. Although outperforming both point-source and wave-field methods without producing any visible artifact, this method requires the CGH to be entirely recomputed for each video frame, increasing the number of redundant calculations.

Indeed, there exist only slight changes between consecutive frames in 3D video, and the computation time of video holograms can be reduced by removing these temporal redundancies [27, 28, 29, 30]. In [27], the author proposed an incremental method for making localized changes to a CGH in order to rapidly modify the scene image it reconstructs. At each simulation time step, the method tracks scene changes and updates the CGH patterns of affected scene points only. In [28], the authors proposed to remove temporal redundancies in the input 3D video images using the differential pulse code modulation (DPCM) algorithm. Then, for each video frame, the CGH pattern of the changed part of the scene only is computed and added to the CGH of the previous frame in order to get the final CGH for the current frame. In [29], the authors proposed a CGH method based on motion compensation technique. For each video frame, motion vectors of 3D moving objects are extracted and used in order to update the CGH pattern of the previous frame. In [30], the authors proposed a MPEG-based method for CGH computation. In this method, the input 3D video frames are grouped into sets of four, in which the first and remaining three frames become the reference (RF) and general frames (GFs). Then, each video frame is divided into blocks, from which motion vectors between the RF and the current frame are estimated. These motion vectors are then used in order to update the CGH pattern of the RF.

Since only affected regions of the CGH are updated at each video frame, these methods allow the computational burden to be dramatically reduced. However, all of these proposed methods are based on the point-source approach, and the possible exploitation of temporal redundancies using the hybrid point-source/wave-field approach has never been studied. Accordingly, in this paper, we propose a fast method for the computation of video holograms based on the hybrid approach by taking into account these temporal redundancies. Section 2 presents our method and Section 3 gives experimental results.

## 2 PROPOSED METHOD

### 2.1 Overview

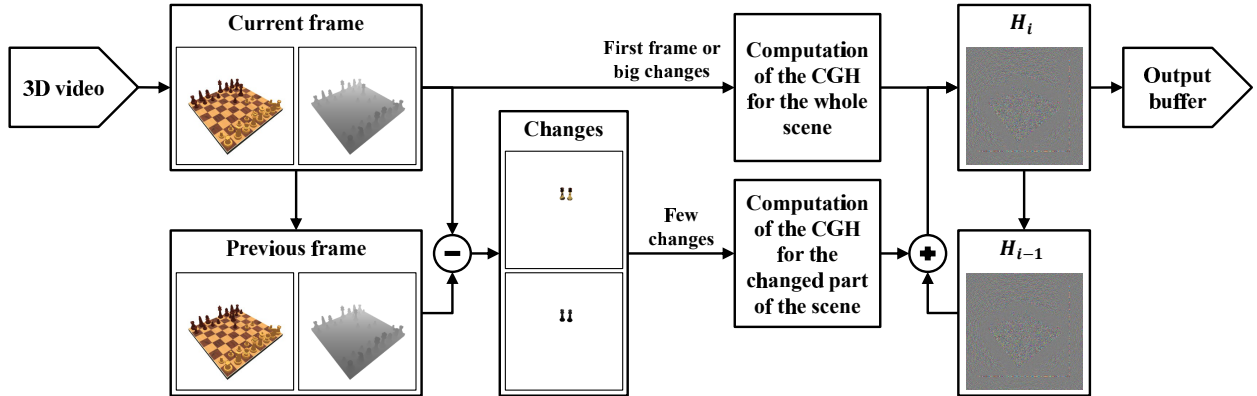


Figure 2: Overview of the proposed method.

Figure 2 shows the overall block-diagram of the proposed method. The input of the method is a 3D video consisting of views and depth maps pairs of resolution  $M_x \times M_y$ . For each video frame  $i$ , a CGH  $H_i$  of resolution  $N_x \times N_y$  with pixel pitch  $p$  is computed.

The CGH computation is done differently for the first video frame and for the following frames. For the first frame,  $H_0$  is computed from the view and depth data using the hybrid point-source/wave-field approach. For the following frames, the view and depth data are first compared to those of the previous frame in order to track the differences. A criterion, defined in Section 2.6, determines whether these differences constitute an important part of the whole scene or not. When the changed part of the scene is important,  $H_i$  is computed from the view and depth data of the current frame  $i$  using the hybrid point-source/wave-field approach. Otherwise, if only a little part of the scene has changed, the temporal redundancies can be removed in order to reduce the computational burden. Therefore, the method computes the CGH of the changed part of the scene only and sums it with the CGH of previous frame  $H_{i-1}$  in order to get  $H_i$ . Finally,  $H_i$  is transmitted to the CGH video output and stored in the previous frame buffer.

### 2.2 Scene geometry

Figure 3 shows the scene geometry and coordinate system used by the proposed method. The coordinate system is defined by  $(x, y, z)$  so that the hologram lies on the  $(x, y, 0)$  plane. For each video frame  $i$ , a 3D point cloud is extracted from the view and depth data, where each point is given an amplitude proportional to its corresponding pixel value in the view image. Since each depth map is encoded as an 8-bits gray level image, the extracted 3D point cloud is naturally sliced as a set of  $N_z = 256$  depth layers parallel to the hologram plane and regularly located between  $z_{\min}$  and  $z_{\max}$ .

The depth layers are classified into two categories: the wave-field layers and the point-source layers. This classification is performed depending on the number of scene points within each layer: if the number of scene points  $M_{i,d}$  within layer  $d$  exceeds a threshold value  $M_{d,\max}$  (selection criterion which will be determined in Section 2.5), this layer is considered as a wave-field layer, otherwise it is considered as a point-source layer. Moreover, each depth layer  $d$  operates as a surface source of light which emits a complex wave  $o_{i,d}$  given by

$$o_{i,d}(x, y) = a_{i,d}(x, y) \exp [j\phi_d(x, y)], \quad (1)$$

where  $a_{i,d}(x, y)$  is the amplitude of the  $(x, y)$  point within layer  $d$  for frame  $i$ , and  $\phi_d(x, y)$  is its phase, set to a random value in order to render a diffusive scene.

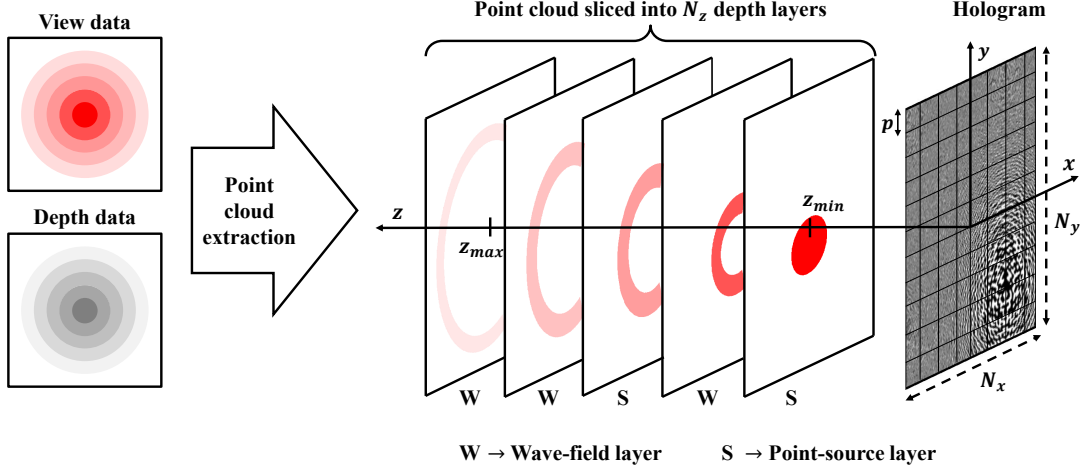


Figure 3: Scene geometry and coordinate system used by the proposed method.

Light scattered by each depth layer  $d$  is numerically propagated towards the hologram plane using either a point-source or a wave-field approach depending on which category layer  $d$  belongs to. Then, the complex waves scattered by all the depth layers are summed up in the hologram plane in order to obtain the final CGH  $H_i$ , according to

$$\begin{aligned}
 H_i(x, y) &= \mathcal{H} \left\{ \{o_{i,d}\}_{0 \leq d < N_z} \right\} = \sum_{\substack{d=0 \\ M_{i,d} > M_{d,\max}}}^{N_z-1} \mathcal{P}_{z_d}^w \{o_{i,d}\}(x, y) + \sum_{\substack{d=0 \\ M_{i,d} \leq M_{d,\max}}}^{N_z-1} \mathcal{P}_{z_d}^s \{o_{i,d}\}(x, y) \\
 &= H_i^w(x, y) + H_i^s(x, y),
 \end{aligned} \tag{2}$$

where operator  $\mathcal{H}$  stands for the numerical CGH computation using the hybrid approach, and operators  $\mathcal{P}_{z_d}^w$  and  $\mathcal{P}_{z_d}^s$  stand for the numerical propagation of light from layer  $d$  at depth  $z_d$  to the hologram plane using the wave-field and the point-source approaches, respectively.

### 2.3 Wave-field layers

When the number of points  $M_{i,d}$  within layer  $d$  exceeds  $M_{d,\max}$ , light scattered by this layer is numerically propagated towards the hologram plane using a wave-field approach. To this end, we use the Angular Spectrum [18], which expresses light propagation between two parallel planes separated by a distance  $z_d$  by

$$\mathcal{P}_{z_d}^w \{o_{i,d}\}(x, y) = \mathcal{F}^{-1} \left\{ \mathcal{F} \{o_{i,d}\} e^{-j2\pi z_d \sqrt{\lambda^{-2} - f_x^2 - f_y^2}} \right\}(x, y), \tag{3}$$

where  $\lambda$  is the wavelength of light,  $f_x$  and  $f_y$  are the spatial frequencies, and  $\mathcal{F}$  is the forward Fourier Transform.

In order to avoid one FFT per layer and therefore to speed-up the computation, the algorithm sums up the complex waves scattered by each wave-field layer directly in the frequency domain, and then inverse Fourier transforms the result to get  $H_i^w$ , as proposed in [19]:

$$\hat{H}_i^w(f_x, f_y) = \sum_{\substack{d=0 \\ M_{i,d} > M_{d,\max}}}^{N_z-1} \mathcal{F} \{o_{i,d}\} e^{-j2\pi z_d \sqrt{\lambda^{-2} - f_x^2 - f_y^2}} \tag{4}$$

$$H_i^w(x, y) = \mathcal{F}^{-1} \left\{ \hat{H}_i^w \right\}(x, y). \tag{5}$$

### 2.4 Point-source layers

When the number of points  $M_{i,d}$  within layer  $d$  is less than  $M_{d,\max}$ , light scattered by this layer is numerically propagated towards the hologram plane using a point-source approach. Scene points located within the layer are therefore considered as isolated light sources, and light propagation is computed as the sum of complex waves scattered by each point. The complex wave scattered by a point  $k$  located at coordinates  $(x_k, y_k)$  within layer  $d$  is given by the Angular Spectrum [18] as

$$w_k(x, y) = o_{i,d}(x_k, y_k) \mathcal{F}^{-1} \left\{ e^{-j2\pi z_d \sqrt{\lambda^{-2} - f_x^2 - f_y^2}} \right\} \otimes \delta(x - x_k, y - y_k), \tag{6}$$

where  $o_{i,d}(x_k, y_k)$  is the complex amplitude of the point and  $\otimes$  is the convolution operator.

Since convolving a function with a position-shifted Dirac delta shifts it by the same amount, knowing the inverse Fourier transform term in Eq. (6) beforehand allows us to compute  $w_k$  simply by scaling this term with  $o_{i,d}(x_k, y_k)$ , followed by a shifting operation. In order to speed up the computation, we use a pre-calculated LUT  $T$ , as proposed in [11].  $T$  is pre-computed as

$$T(x, y, z) = \mathcal{F}^{-1} \left\{ e^{-j2\pi z \sqrt{\lambda^{-2} - f_x^2 - f_y^2}} \right\} h(x, y, z), \quad (7)$$

$h$  being a window function used to restrict the region of contribution of a given point, equal to one within the region of contribution of the point and zero elsewhere. This function limits the spatial frequencies of the complex wave to avoid aliasing in the CGH.

According to the Nyquist Sampling Theorem, the maximum spatial frequency that can be represented with a sampling pitch  $p$  is  $f_{\max} = (2p)^{-1}$ . The relation between the maximum spatial frequency  $f_{\max}$  and the maximum diffraction angle  $\theta$  is given by the grating equation [18] as  $\sin(\theta) = \lambda f_{\max}$ . Therefore, the region of contribution of a point at depth  $z$  is given by its maximum radius  $R_{\max}(z)$  by

$$R_{\max}(z) = z \tan(\theta) = z \tan \left( \arcsin \left( \frac{\lambda}{2p} \right) \right), \quad (8)$$

as shown in Figure 4. The window function  $h$  can thus be defined as

$$h(x, y, z) = \begin{cases} 1 & \text{if } \sqrt{x^2 + y^2} < R_{\max}(z) \\ 0 & \text{otherwise.} \end{cases} \quad (9)$$

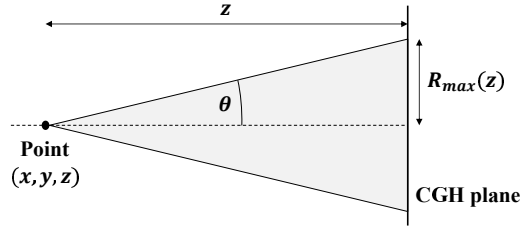


Figure 4: Region of contribution of a given point source.

Thanks to its geometrical symmetry and to the window function  $h$  defined by Eq. (9), the LUT doesn't have to be computed for every  $(x, y)$  values. Instead, and in order to limit its number of samples,  $T$  is pre-computed only within the upper quarter of the square circumscribing the disk defined by  $h$ . Therefore, the number of samples  $N_{T,z_d}$  of the LUT for depth  $z_d$  is given by

$$N_{T,z_d} = \left( \frac{R_{\max}(z_d)}{p} \right)^2 = \left[ \frac{z_d}{p} \tan \left( \arcsin \left( \frac{\lambda}{2p} \right) \right) \right]^2. \quad (10)$$

Then, the complex wave scattered by point-source layer  $d$  in the hologram plane can be obtained by simply addressing this pre-calculated LUT, as

$$\mathcal{P}_{z_d}^s \{o_{i,d}\} (x, y) = \sum_{k=0}^{M_{i,d}-1} o_{i,d}(x_k, y_k) T(|x - x_k|, |y - y_k|, z_d). \quad (11)$$

The complex wave scattered by all the point-source layers in the hologram plane is therefore given by

$$H_i^s(x, y) = \sum_{\substack{d=0 \\ M_{i,d} \leq M_{d,\max}}}^{N_z-1} \sum_{k=0}^{M_{i,d}-1} o_{i,d}(x_k, y_k) T(|x - x_k|, |y - y_k|, z_d). \quad (12)$$

## 2.5 Layers classification

The first step for implementing the proposed method is determining the value of  $M_{d,\max}$ . We call  $t_w$  the time needed to compute the complex wave scattered by a layer at depth  $z_d$  with  $M_{i,d}$  luminous points using operator  $\mathcal{P}_{z_d}^w$ , and  $t_s$  the time needed to compute it using operator  $\mathcal{P}_{z_d}^s$ . Execution times  $t_w$  and  $t_s$  are obviously dependent on the implementation of  $\mathcal{P}_{z_d}^w$  and  $\mathcal{P}_{z_d}^s$ , respectively.

Since the computation of  $\mathcal{P}_{z_d}^w$  involves one complex multiplication per hologram pixel and a Fourier transform,  $t_w$  is dependent on the number of hologram pixels  $N = N_x \times N_y$ . In our implementation, the Fourier transform of Eq. (3) is performed on the GPU using the CUDA cuFFT library, so  $t_w$  is dependent on the implementation of this library. In order to find the expression of  $t_w$ , we measured the CGH computation time of one thousand random scenes containing only one wave-field layer for various hologram resolutions.  $t_w$  is experimentally found to be linearly dependent on the number of hologram pixels  $N$ , such that

$$t_w(N) = kN. \quad (13)$$

On the other hand, the computation of  $\mathcal{P}_{z_d}^s$  involves one complex multiplication per sample of the LUT per point within the layer, so  $t_s$  is dependent on the number of samples  $N_{T,z_d}$  of the LUT for depth  $z_d$  and on the number of points  $M_{i,d}$  within the layer. In order to find the expression of  $t_s$ , we measured the CGH computation time of one thousand random scenes containing only one point-source layer for various number of scene points and various number of samples of the LUT.  $t_s$  is experimentally found to be expressed by

$$t_s(M_{i,d}, N_{T,z_d}) = \left[ a(bN_{T,z_d} + c)^{\frac{1}{2}} + dN_{T,z_d} + e \right] M_{i,d}. \quad (14)$$

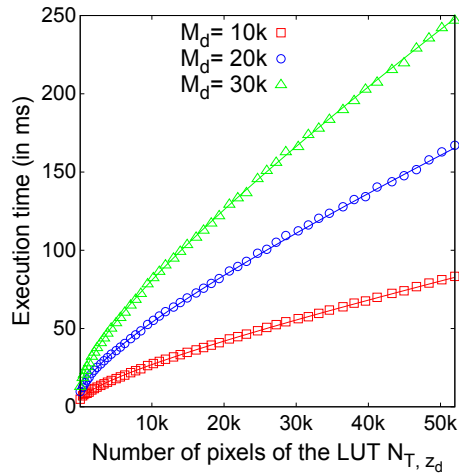


Figure 5: Time needed to compute the complex wave scattered by a point-source layer depending on the number of samples of the LUT and on the number of scene points.

Figure 5 shows the graph of  $t_s$  depending on the number of samples  $N_{T,z_d}$  of the LUT for three different values of  $M_{i,d}$ . As shown in this figure,  $t_s$  increases quickly with small values of  $N_{T,z_d}$  and then increases more slowly for higher values of  $N_{T,z_d}$ . This is due to the fact that when the number of samples  $N_{T,z_d}$  of the LUT is large, there is more chances for the regions of contribution of the different scene points to overlap in the hologram plane.

We find the numerical values for the coefficients  $a$ ,  $b$ ,  $c$ ,  $d$ ,  $e$  and  $k$  in Eq. (13) and Eq. (14) using the Gnuplot implementation of the nonlinear least-squares Levenberg-Marquardt algorithm [31]:

$$\begin{cases} a = 2,69.10^{-5} & b = 0,34 & c = 1,11 \\ d = 8,49.10^{-7} & e = 2,76.10^{-4} & k = 5,54.10^{-7} \end{cases} \quad (15)$$

In order to maximize the efficiency of the proposed method,  $M_{d,\max}$  must be set such that

$$t_s(M_{d,\max}, N_{T,z_d}) = t_w(N) \quad (16)$$

$$\Leftrightarrow M_{d,\max} = \frac{kN}{\left[ a(bN_{T,z_d} + c)^{\frac{1}{2}} + dN_{T,z_d} + e \right]}. \quad (17)$$

## 2.6 Incremental update

Similarly to 2D video, consecutive frames in the 3D video differ slightly from each other. This means that most of the scene points extracted from view and depth data remain the same from one frame to another. As a consequence, the CGH of the scene does not need to be entirely recomputed for each frame. Instead, the proposed method first tracks the scene changes between two consecutive frames in order to remove the temporal redundancies. Then, if only a little part of the scene has changed, the method computes the CGH of the changed

part of the scene only and sums it with the CGH of the previous frame  $H_{i-1}$  in order to get  $H_i$ . This allows the CGH computational burden to be dramatically reduced.

In order to track the differences between two consecutive video frames  $i-1$  and  $i$ , the method computes the complex wave  $o'_{i,d}$  scattered by the appearing and disappearing scene points within each depth layer  $d$ . We have

$$\begin{aligned} o'_{i,d}(x, y) &= o_{i,d}(x, y) - o_{i-1,d}(x, y) \\ &= (a_{i,d}(x, y) - a_{i-1,d}(x, y)) \exp[j\phi_d(x, y)], \end{aligned} \quad (18)$$

where  $a_{i,d}(x, y)$  and  $a_{i-1,d}(x, y)$  are the amplitude of the  $(x, y)$  point within layer  $d$  for frames  $i$  and  $i-1$ , respectively. The CGH of the changed parts of the scene is therefore given by

$$H'_i(x, y) = \mathcal{H} \left\{ \{o'_{i,d}\}_{0 \leq d < N_z} \right\}. \quad (19)$$

The CGH  $H_i$  can therefore be computed as

$$\begin{aligned} H_i(x, y) &= \mathcal{H} \left\{ \{o'_{i,d} + o_{i-1,d}\}_{0 \leq d < N_z} \right\} \\ &= \mathcal{H} \left\{ \{o'_{i,d}\}_{0 \leq d < N_z} \right\} + \mathcal{H} \left\{ \{o_{i-1,d}\}_{0 \leq d < N_z} \right\} \\ &= H'_i(x, y) + H_{i-1}(x, y), \end{aligned} \quad (20)$$

where  $\mathcal{H} \left\{ \{o'_{i,d} + o_{i-1,d}\}_{0 \leq d < N_z} \right\} = \mathcal{H} \left\{ \{o'_{i,d}\}_{0 \leq d < N_z} \right\} + \mathcal{H} \left\{ \{o_{i-1,d}\}_{0 \leq d < N_z} \right\}$ , according to the definition of operator  $\mathcal{H}$ .

Since there exist only slight changes in view and depth data between two consecutive frames of the 3D video, the number  $M'_{i,d}$  of appearing and disappearing points between frames  $i-1$  and  $i$  may be far less than the total number of scene points  $M_{i,d}$  for frame  $i$ . Therefore, the computation of the CGH for the changed parts of the scene  $H'_i$  may be faster than the computation of the CGH for the whole scene  $H_i$ . The CGH computation time  $t_i$  for frame  $i$  using Eq. (2) is given by

$$t_i = \sum_{\substack{d=0 \\ M_{i,d} > M_{d,\max}}}^{N_z-1} t_w(N) + \sum_{\substack{d=0 \\ M_{i,d} \leq M_{d,\max}}}^{N_z-1} t_s(M_{i,d}, N_{T,z_d}). \quad (21)$$

We call  $t_i$  and  $t'_i$  the computation times of  $H_i$  and  $H'_i$  using Eq. (2), respectively. The computation of  $H_i$  using Eq. (23) is therefore faster than using Eq. (2) if

$$\begin{aligned} & t'_i < t_i \\ \sum_{\substack{d=0 \\ M'_{i,d} > M_{d,\max}}}^{N_z-1} M_{d,\max} + \sum_{\substack{d=0 \\ M'_{i,d} \leq M_{d,\max}}}^{N_z-1} M'_{i,d} & < \sum_{\substack{d=0 \\ M_{i,d} > M_{d,\max}}}^{N_z-1} M_{d,\max} + \sum_{\substack{d=0 \\ M_{i,d} \leq M_{d,\max}}}^{N_z-1} M_{i,d}. \end{aligned} \quad (22)$$

For each video frame  $i$ , the CGH  $H_i$  is therefore computed using the recurrence formula

$$H_i(x, y) = \begin{cases} \mathcal{H} \left\{ \{o_{i,d}\}_{0 \leq d < N_z} \right\} & \text{if } i = 0 \text{ or } t'_i \geq t_i \\ \mathcal{H} \left\{ \{o'_{i,d}\}_{0 \leq d < N_z} \right\} + H_{i-1}(x, y) & \text{otherwise.} \end{cases} \quad (23)$$

### 3 EXPERIMENTAL RESULTS AND DISCUSSION

The proposed method was implemented in C++/CUDA on a PC system employing an Intel Core i7-4930K CPU operating at 3.40 GHz, a main memory of 16 GB and an operating system of Microsoft Windows 8 as well as three GPUs NVIDIA GeForce GTX 780Ti.

#### 3.1 Input 3D video

For the experiments, we generated with Unity an input 3D video containing 375 views and depth map pairs of resolution  $512 \times 512$ . From each view and depth map pair, a 3D point cloud is extracted, where each point is given an amplitude proportional to its corresponding pixel value in the view image. The extracted 3D point cloud is considered to be located between  $z_{\min} = 0$  and  $z_{\max} = 2\text{cm}$  in front of the CGH plane. Finally, the CGH to be computed has a resolution of  $2048 \times 2048$  with a sampling pitch  $p = 8, 1\mu\text{m}$ .

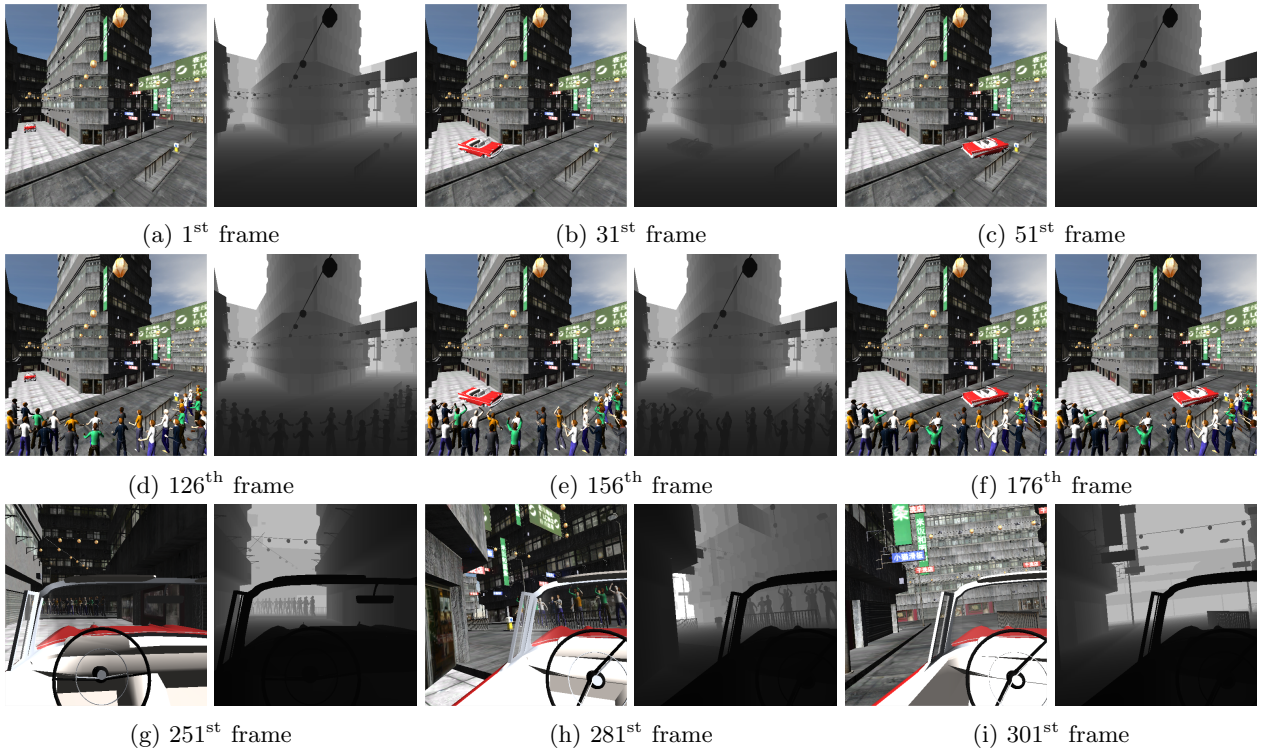


Figure 6: View and depth maps pairs for nine different frames of the input 3D video.

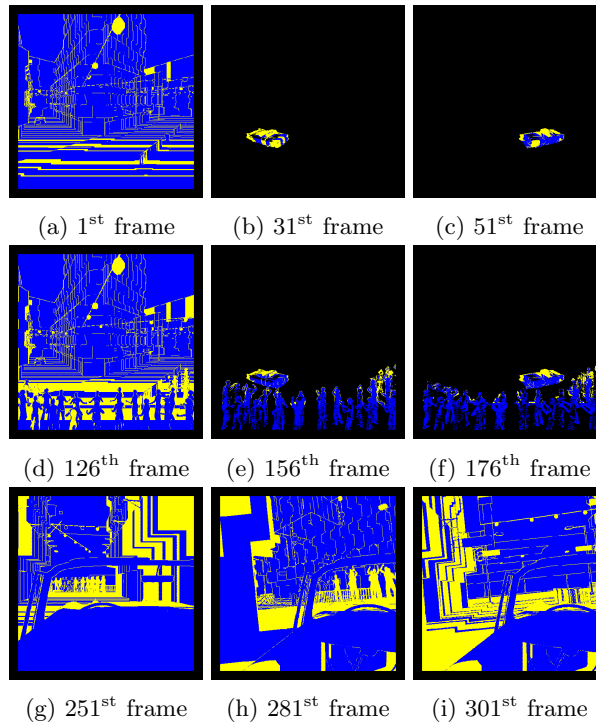


Figure 7: Scene points whose complex waves are updated at each frame. The scene points whose complex waves are computed by the proposed method using the wave-field approach are shown in blue. The scene points whose complex waves are computed using the point-source approach are shown in yellow.

Figure 6 shows the view and depth maps pairs for nine different frames of the video, which is composed of three parts. The first part (1<sup>st</sup> to 125<sup>th</sup> frames) shows a car turning around a building in a street. The second part (126<sup>th</sup> to 250<sup>th</sup> frames) shows the same scene with some cheering people. Finally, the third part (251<sup>st</sup> to 375<sup>th</sup> frames) shows the same scene from the driver's viewpoint. In the first and second parts of the video the camera is fixed, whereas it is moving with the car in the third part of the video. As a consequence, while consecutive frames in the first and second parts of the video differ slightly from each other, there exist large

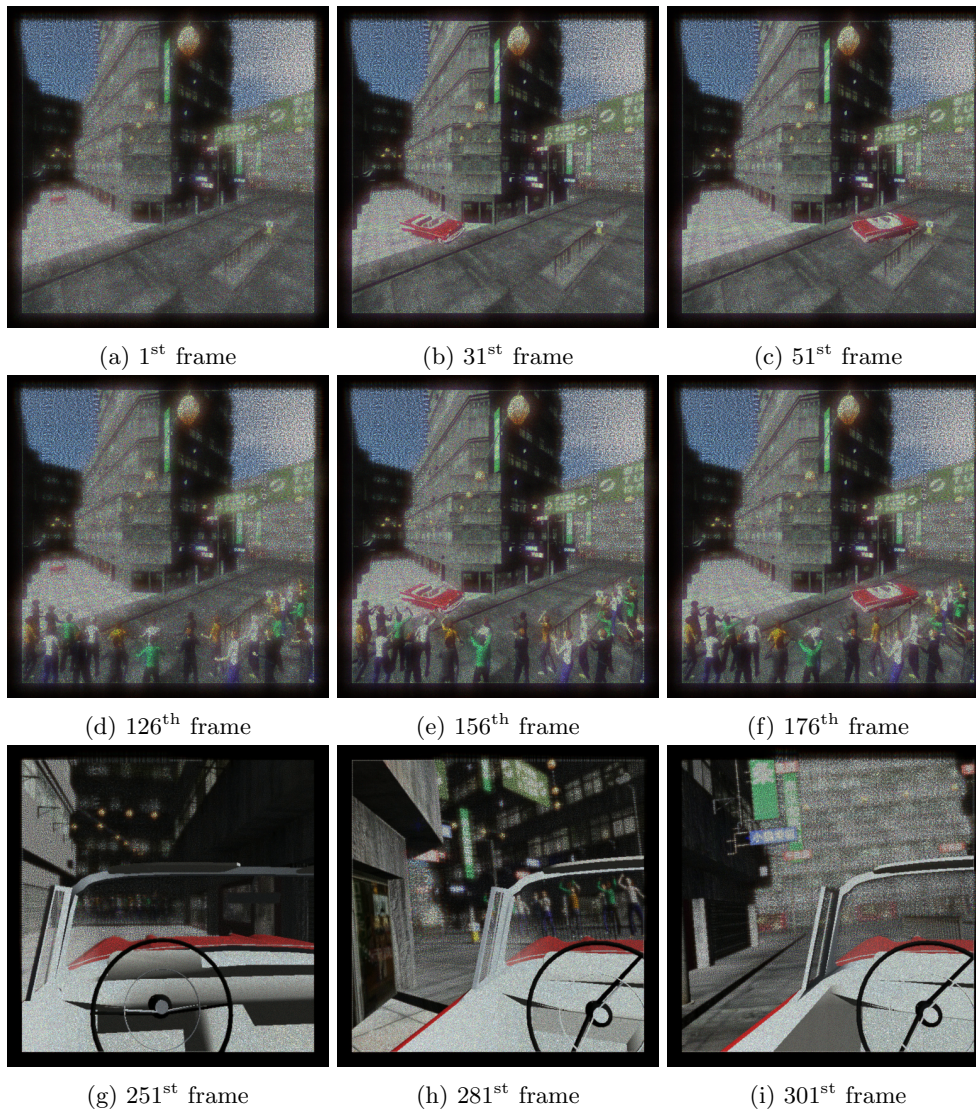


Figure 8: Scene images numerically reconstructed from the CGH patterns of nine different frames.

changes between consecutive frames in the third part of the video.

### 3.2 Reconstructed scene images

Figure 7 shows the scene points whose complex waves are updated at each frame. The scene points whose complex waves are computed using the wave-field approach are shown in blue, whereas those whose complex waves are computed using the point-source approach are shown in yellow. As seen in this figure, since there exist slight changes between consecutive frames in the first and second parts of the video, the proposed method updates the complex waves scattered by only a few scene points at each frame. On the other hand, since a large part of the scene changes between consecutive frames in the third part of the video, the proposed method recomputes the complex wave scattered by the whole scene at each frame. Moreover, due to the sudden changes of the scene between the first and second parts of the video, the proposed method also recomputes the complex wave scattered by the whole scene for the 126<sup>th</sup> frame.

Figure 8 shows the scene images numerically reconstructed from the CGH patterns of nine different frames. For the first and second parts of the video, the reconstructed images are focused on the facade of the building, and for the third part of the video, the reconstructed images are focused on the steering wheel. As shown in this figure, the proposed method does not produce any visible artifact, even at the boundaries between the two categories of scene points. In order to evaluate the objective quality of the reconstructed images compared to the hybrid method proposed in [26], we used the Peak Signal-to-Noise Ratio (PSNR). The PSNR of the reconstructed images were found to be higher than 50dB for each video frame. These experimental results show that the CGH pattern is accurately updated at each video frame and that the proposed method does not reduce the quality of the reconstructed scene images compared to the hybrid method.

### 3.3 Performance analysis

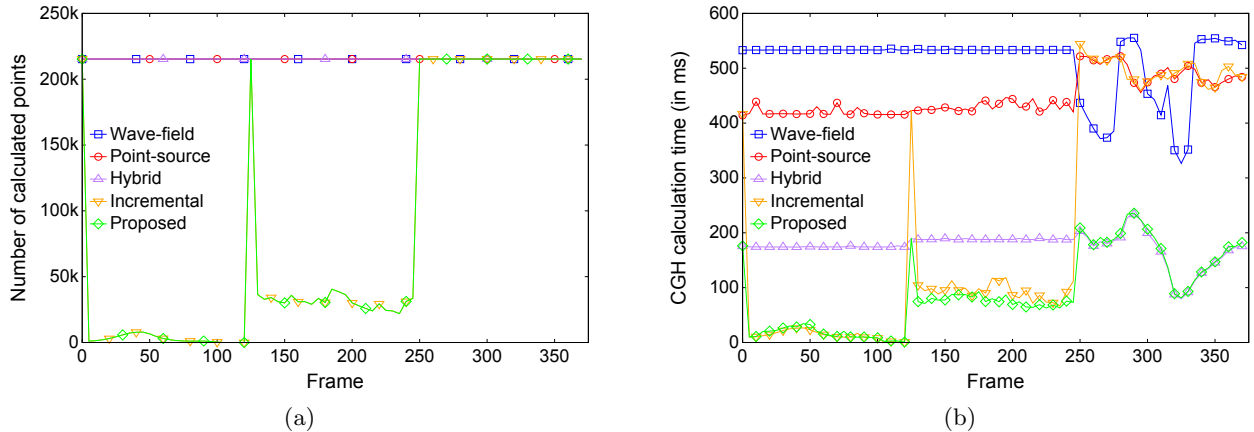


Figure 9: Performance comparison between the wave-field method (in blue), the point-source method (in red), the hybrid method (in purple), the incremental method (in orange), and the proposed method (in green): (a) number of points whose complex waves are computed at each frame, (b) CGH computation time per frame.

In order to evaluate the performance of the proposed method in terms of computation time, we compare it with GPU implementations of four other methods: (1) the method proposed in [32], based on a wave-field approach, (2) the method proposed in [11], based on a point-source approach, (3) the method proposed in [28], based on a point-source approach and incremental update of the CGH at each frame, and (4) the method proposed in [26], based on a hybrid approach. In the following, these methods are called wave-field method, point-source method, incremental method and hybrid method, respectively. We adapted the first three methods in order to produce colorful complex modulation CGH.

Figure 9a shows the number of scene points whose complex waves are computed at each frame for all the compared methods. Since the wave-field, point-source and hybrid methods do not take into account temporal redundancies between consecutive frames in the 3D video, the number of calculated scene points remains the same for each video frame. By contrast, the incremental and proposed methods take into account those redundancies in order to reduce the number of calculated scene points. As a consequence, since consecutive frames in the first and second parts of the video differ slightly from each other, the number of calculated points is much lower for the incremental and proposed methods than for the wave-field, point-source and hybrid methods. Due to the cheering people appearing in the second part of the video, the number of calculated scene points is higher for these frames than for the first part of the video. Finally, since there exist large changes between consecutive frames in the third part of the video, the number of calculated scene points is exactly the same for all the compared methods.

Figure 9b shows the CGH computation time per frame for all the compared methods. The computational complexity of the wave-field method is dependent on the number of depth layers containing scene points. By contrast, the computational complexity of the point-source method is dependent on the number of scene points and on their location within the scene. Since consecutive frames in the first and second parts of the video differ slightly from each other, the number of non-empty depth layers and the location of scene points are almost the same from one frame to another. As a consequence, the CGH computation times of the wave-field and point-source methods remain almost constant for the first and second parts of the video, as shown in Figure 9b. On the contrary, since there exist large changes between consecutive frames in the third part of the video, the number of non-empty depth layers and the location of scene points change from one frame to another. Therefore, the CGH computation times of the wave-field and point-source methods also change from one frame to another.

Since it combines the wave-field and point-source approaches, the computational complexity of the hybrid method is dependent on the number of depth layers containing scene points, on the number of scene points and on their location within the scene. As a consequence, its CGH computation time varies in the same way as those of the wave-field and point-source methods, as shown in Figure 9b. However, by combining these two approaches, the hybrid method takes advantages from both of them and is therefore always more efficient than the wave-field and point-source methods.

In the first and second parts of the video, the number of calculated points of the incremental and proposed methods is much lower than that of the wave-field, point-source and hybrid methods. As a consequence, the incremental and proposed methods take much less time to calculate the CGH patterns than the wave-field, point-source and hybrid methods, as shown in Figure 9b. While the CGH computation times of the incremental and proposed methods are almost the same for the first part of the video, the proposed method is more efficient

than the incremental method for the second part of the video. This is due to the fact that when the number of calculated scene points increases, the efficiency of the hybrid approach also increases [26]. Finally, in the third part of the video, the number of calculated scene points is exactly the same for all the compared methods, and the CGH computation times of the incremental and proposed methods reach those of the point-source and hybrid methods, respectively. As a consequence, the proposed method is much more efficient than the incremental method for the third part of the video.

	Video part	Hybrid method	Incremental method	Proposed method
<b>Number of calculated points</b>	<i>Part 1</i>	215293 (100%)	4282 (1,99%)	4282 (1,99%)
	<i>Part 2</i>	215293 (100%)	32537 (15,1%)	32537 (15,1%)
	<i>Part 3</i>	215296 (100%)	215296 (100%)	215296 (100%)
<b>CGH computation time (in ms)</b>	<i>Part 1</i>	174,46 (100%)	16,69 (9,57%)	16,60 (9,52%)
	<i>Part 2</i>	188,02 (100%)	96,15 (51,1%)	75,37 (40,1%)
	<i>Part 3</i>	164,57 (100%)	492,80 (299%)	168,07 (102%)

Table 1: Comparison of the average number of calculated scene points and the average CGH computation time per frame between the hybrid, the incremental and the proposed methods.

Table 1 shows the average number of calculated scene points and the average CGH computation time per frame for the hybrid, incremental and proposed methods. As shown in this table, for the first part of the video, the average number of calculated scene points per frame is equal to 215293 for the hybrid method and to 4282 for the incremental and proposed methods. The average number of calculated scene points has therefore been reduced by 98,01% using the incremental and proposed methods. As a consequence, the average CGH computation time per frame is equal to 174,46ms, 16,69ms and 16,60ms for the hybrid, incremental and proposed methods, respectively. The CGH computation time per frame has therefore been reduced by 90,43% and 90,48% using the incremental and proposed methods, respectively.

Similarly, for the second part of the video, the average number of calculated scene points per frame is equal to 215293 for the hybrid method and to 32537 for the incremental and proposed methods. The average number of calculated scene points has therefore been reduced by 84,9% using the incremental and proposed methods. As a consequence, the average CGH computation time per frame is equal to 188,02ms, 96,15ms and 75,37ms for the hybrid, incremental and proposed methods, respectively. The CGH computation time per frame has therefore been reduced by 48,9% and 59,9% using the incremental and proposed methods, respectively. This confirms that the proposed method is more efficient than the incremental method for the second part of the video.

Finally, for the third part of the video, the average number of calculated scene points per frame is equal to 215296 for the hybrid, incremental and proposed methods. As a consequence, the average CGH computation time per frame is equal to 164,57ms, 492,80ms and 168,07ms for the hybrid, incremental and proposed methods, respectively. The proposed method is therefore much more efficient than the incremental method when there exist important changes between consecutive frames of the input 3D video. The CGH computation time per frame has been increased by 2% using the proposed method compared to the hybrid method. This is due to the additional pre-processing stage used in order to track the differences from one frame to another.

These experimental results show that for the 3D video cases in which consecutive frames differ slightly from each other, the number of scene points to be calculated and the CGH computation time can be significantly reduced using the proposed method. Additionally, for the 3D video cases in which there exist large changes between consecutive frames, the proposed method is still more efficient than the methods based on point-source or wave-field approaches.

## 4 CONCLUSION

In this paper, we proposed a fast Computer-Generated Hologram (CGH) computation method based on a hybrid point-source/wave-field approach and temporal redundancy compression. Our algorithm consists of three steps. First, intensity and depth data of the current 3D video frame are extracted and compared with those of the previous frame in order to remove temporally redundant data. Then, the CGH pattern for this compressed frame is generated using the hybrid point-source/wave-field approach. The resulting CGH pattern is finally transmitted to the video output and stored in the previous frame buffer. Experimental results with three kinds of input 3D videos revealed that our method is able to produce colorful video holograms at interactive rates without producing any visible artifact.

## References

- [1] Ulf Schnars and Werner Jüptner. *Digital Holography: Digital Hologram Recording, Numerical Reconstruction, and Related Techniques*. Springer Science & Business Media, December 2005.
- [2] J. L. Juárez-Pérez, A. Olivares-Pérez, and L. R. Berriel-Valdos. Nonredundant calculations for creating digital Fresnel holograms. *Applied Optics*, 36(29):7437–7443, October 1997.
- [3] Hiroshi Yoshikawa, Susumu Iwase, and Tadashi Oneda. Fast computation of Fresnel holograms employing difference. In *Practical Holography XIV and Holographic Materials VI*, volume 3956, pages 48–55, May 2000.
- [4] Kyoji Matsushima and Masahiro Takai. Recurrence Formulas for Fast Creation of Synthetic Three-Dimensional Holograms. *Applied Optics*, 39(35):6587–6594, December 2000.
- [5] Takeshi Yamaguchi, Gen Okabe, and Hiroshi Yoshikawa. Real-time image plane full-color and full-parallax holographic video display system. *Optical Engineering*, 46(12):125801–125801–8, 2007.
- [6] Hiroshi Yoshikawa, Takeshi Yamaguchi, and Ryo Kitayama. Real-Time Generation of Full Color Image Hologram with Compact Distance Look-up Table. In *Advances in Imaging*, OSA Technical Digest (CD), page DWC4. Optical Society of America, April 2009.
- [7] Tomoyoshi Shimobaba, Nobuyuki Masuda, and Tomoyoshi Ito. Simple and fast calculation algorithm for computer-generated hologram with wavefront recording plane. *Optics Letters*, 34(20):3133–3135, October 2009.
- [8] Tomoyoshi Shimobaba, Hirotaka Nakayama, Nobuyuki Masuda, and Tomoyoshi Ito. Rapid calculation algorithm of Fresnel computer-generated-hologram using look-up table and wavefront-recording plane methods for three-dimensional display. *Optics Express*, 18(19):19504–19509, September 2010.
- [9] Jiantong Weng, Tomoyoshi Shimobaba, Naohisa Okada, Hirotaka Nakayama, Minoru Oikawa, Nobuyuki Masuda, and Tomoyoshi Ito. Generation of real-time large computer generated hologram using wavefront recording method. *Optics Express*, 20(4):4018–4023, February 2012.
- [10] Mark E. Lucente. Interactive computation of holograms using a look-up table. *Journal of Electronic Imaging*, 2(1):28–34, January 1993.
- [11] Seung-Cheol Kim and Eun-Soo Kim. Effective generation of digital holograms of three-dimensional objects using a novel look-up table method. *Applied Optics*, 47(19):D55–D62, July 2008.
- [12] Seung-Cheol Kim and Eun-Soo Kim. Fast computation of hologram patterns of a 3d object using run-length encoding and novel look-up table methods. *Applied Optics*, 48(6):1030–1041, February 2009.
- [13] Seung-Cheol Kim, Woo-Young Choe, and Eun-Soo Kim. Accelerated computation of hologram patterns by use of interline redundancy of 3-D object images. *Optical Engineering*, 50(9):091305–091305–10, 2011.
- [14] Lukas Ahrenberg, Philip Benzie, Marcus Magnor, and John Watson. Computer generated holography using parallel commodity graphics hardware. *Optics express*, 14(17):7636–7641, August 2006.
- [15] Nobuyuki Masuda, Tomoyoshi Ito, Takashi Tanaka, Atsushi Shiraki, and Takashige Sugie. Computer generated holography using a graphics processing unit. *Optics Express*, 14(2):603–608, January 2006.
- [16] Tomoyoshi Ito, Nobuyuki Masuda, Kotaro Yoshimura, Atsushi Shiraki, Tomoyoshi Shimobaba, and Takashige Sugie. Special-purpose computer HORN-5 for a real-time electroholography. *Optics Express*, 13(6):1923–1932, March 2005.
- [17] Yasuyuki Ichihashi, Hirotaka Nakayama, Tomoyoshi Ito, Nobuyuki Masuda, Tomoyoshi Shimobaba, Atsushi Shiraki, and Takashige Sugie. HORN-6 special-purpose clustered computing system for electroholography. *Optics Express*, 17(16):13895–13903, August 2009.
- [18] Joseph W. Goodman. *Introduction to Fourier Optics*. Roberts and Company Publishers, Englewood, Colo, 3rd edition, 2005.
- [19] Lukas Ahrenberg, Philip Benzie, Marcus Magnor, and John Watson. Computer generated holograms from three dimensional meshes using an analytic light transport model. *Applied Optics*, 47(10):1567–1574, April 2008.

- [20] Hwi Kim, Joonku Hahn, and Byoung-ho Lee. Mathematical modeling of triangle-mesh-modeled three-dimensional surface objects for digital holography. *Applied Optics*, 47(19):D117–D127, July 2008.
- [21] Hironobu Sakata and Yuji Sakamoto. Fast computation method for a Fresnel hologram using three-dimensional affine transformations in real space. *Applied Optics*, 48(34):H212–H221, December 2009.
- [22] Yuanzhi Liu, Jianwen Dong, Yiyang Pu, Bingchu Chen, Hexiang He, Zilan Deng, and Hezhou Wang. A fast analytical algorithm for generating CGH of 3d scene. In *Practical Holography XXIV: Materials and Applications*, volume 7619, pages 76190N–76190N–8, 2010.
- [23] Yuanzhi Liu, Jianwen Dong, Yiyang Pu, Bingchu Chen, Hexiang He, and Hezhou Wang. High-speed full analytical holographic computations for true-life scenes. *Optics Express*, 18(4):3345–3351, February 2010.
- [24] Tomoyoshi Shimobaba, Takashi Kakue, and Tomoyoshi Ito. Acceleration of color computer-generated hologram from three-dimensional scenes with texture and depth information. In *Three-Dimensional Imaging, Visualization, and Display 2014*, volume 9117, pages 91170B–91170B–8, 2014.
- [25] Tomoyoshi Shimobaba, Yuki Nagahama, Takashi Kakue, Naoki Takada, Naohisa Okada, Yutaka Endo, Ryuji Hirayama, Daisuke Hiyama, and Tomoyoshi Ito. Calculation reduction method for color digital holography and computer-generated hologram using color space conversion. *Optical Engineering*, 53(2):024108–024108, 2014.
- [26] Antonin Gilles, Patrick Gioia, Rémi Cozot, and Luce Morin. Complex modulation Computer-Generated Hologram by a fast hybrid Point-source/Wave-field approach. In *International Conference on Image Processing*, Quebec City, September 2015. IEEE.
- [27] Wendy Plesniak. Incremental update of computer-generated holograms. *Applied Optics*, 42(6):1560–1571, June 2003.
- [28] Seung-Cheol Kim, Jung-Hoon Yoon, and Eun-Soo Kim. Fast generation of three-dimensional video holograms by combined use of data compression and lookup table techniques. *Applied Optics*, 47(32):5986–5995, November 2008.
- [29] Seung-Cheol Kim, Xiao-Bin Dong, Min-Woo Kwon, and Eun-Soo Kim. Fast generation of video holograms of three-dimensional moving objects using a motion compensation-based novel look-up table. *Optics Express*, 21(9):11568–11584, May 2013.
- [30] Xiao-Bin Dong, Seung-Cheol Kim, and Eun-Soo Kim. MPEG-based novel look-up table for rapid generation of video holograms of fast-moving three-dimensional objects. *Optics Express*, 22(7):8047, April 2014.
- [31] D. Marquardt. An Algorithm for Least-Squares Estimation of Nonlinear Parameters. *Journal of the Society for Industrial and Applied Mathematics*, 11(2):431–441, June 1963.
- [32] Muharrem Bayraktar and Meriç Özcan. Method to calculate the far field of three-dimensional objects for computer-generated holography. *Applied Optics*, 49(24):4647–4654, August 2010.