Multimedia Tools and Applications
https://doi.org/10.1007/s11042-024-19991-2

A Monte Carlo-based approach to reduce the cost of generating synthetic holograms for 3D computer graphics scenes

Francisco Serón[1] · Alfonso Blesa[2]

Received: 20 September 2023 / Revised: 26 June 2024 / Accepted: 30 July 2024

Abstract
Similar to computer graphics, the digital approach of holography (Computer Genarated Holograms) inherently samples the world. Calculating CGHs needs additional phase-related calculations, elevating computational demands.These calculations can make the development time excessively long or, in the worst case, unaffordable. Our approach, Partial Monte Carlo Sampling (PMCS), calculates CGHs via Monte Carlo techniques. By varying sampling resolution and ray percentages, reasonable quality is achieved, even with fewer rays. Quality analysis employing RMSE and CC reveals that reducing ray percentages does not severely compromise image quality. Both simulated and laboratory holograms display consistent trends, showcasing improved quality with increased rays and resolution. A direct relationship is established between image quality and computational time, addressing specific needs.

Keywords Computer Generated Hologram -CGH- · Monte Carlo sampling · 3D Computer Graphics · Ray tracing

1 Introduction

Traditionally, graphics scenes have been represented through 3D geometry, texture and appearance (local and global reflectance characteristics) and all representations- share the property that their treatment of light is motivated through ray optics and that rendering involves projection and rasterization, or ray tracing.

Alfonso Blesa contributed equally to this work.

✉ Alfonso Blesa
ablesa@unizar.es

Francisco Serón
seron@unizar.es

[1] Department of Informatics, Universidad de Zaragoza, 50018 EINA. Campus Río Ebro, Zaragoza, Zaragoza, Spain

[2] Department of Electronics, Universidad de Zaragoza, 44003 EUPT, Teruel, Teruel, Spain

Published online: 13 August 2024

In recent years, has been an increased interest in holographic displays [1], which reconstruct the 3D object or scene wave front directly from the underlying holographic representation. Holograms are elegant structures capturing the phase and amplitude of a scene wave front as seen from all possible views through a window of given aperture.

On the other hand, holograms represent visual scene appearance in the most elegant way, containing any possible view from a continuous viewport region without aliasing. The holograms main characteristics are (via waves) refocusing, aliasing free, scene independent sampling, phase information for depth encoding, recording without optical elements, compression or combination with geometrical representations (synthetic scenes).

In many ways, holograms are complementary to light fields (see Table 1 from [2]). The major difference to light filed is its intrinsic wave optics handling antialiasing implicitly and its ability to reproduce object depth through phase info, thus, holograms overcome some of the inherent limitations of image-based methods including defocus and compositing with conventional graphics scenes.

Holography is mathematically considerably more demanding than geometric optics. Recent computer technology makes it possible to synthesize holograms on a desktop computer, but even on a state-of-the-art computer it takes several hours or as long as few days to create a full-parallax hologram by ray tracing methods.

To better understand an approximation to the computational load requirements of CGHs (Computer Generated Holograms), let's assume a discretized scene with a resolution of HDTV (1902 × 1080 pixels). A SLM (Spatial Light Modulator) is a device that allows CGH to be displayed when properly illuminated. With current technology we can assume a pixel size of 8 μm and to have a sufficiently large viewing window (e.g. 27 inches monitors in 16 × 9 format implies 65,8 cm wide and 33,6 cm high) we need about $3, 45 \times 10^9$ pixels. Since each pixel in the scene must send information to each pixel in the SLM, we need approximately $3, 15 \times 10^{15}$ times the computational effort of a single ray. Spatial or temporal multiplexing that can generate color requires three times as much.

In spite of some current limitations, such computational cost and sampling rates, the rapid development of computer power and data storage makes holographic representation, image generation and display technology of greatest potential for the future of computer graphics and interactive visual representation. Holograms can be computer generated form synthetic data and rendered either on a conventional displays as [3–5]. Last work present a full framework for a holography-inspired graphics pipeline, which allows to generate, process and render holograms from a synthetic and real 3D objects. Its structure include the thoretical bases of Propagation (Scalar wave representation, Wavefront propagation, Angular spectrum, Discrete

Table 1 CGH Series

Screen size	16 × 16	32 × 132	64 × 64	128 × 128	256 × 256
PCMS ratio					
10	7464960	29859840	119439360	477757440	1911029760
20	14929920	59719680	238878720	955514880	3822059520
40	29859840	119439360	477757440	1911029760	7644119040
60	44789760	179159040	716636160	2866544640	11466178560
80	59719680	238878720	955514880	3822059520	15288238080
100	74649600	298598400	1194393600	4777574400	19110297600

Rays used for CGH Series. PCMS percentage vs. size screen. SLM size 540 × 540 pixels propagation), the recording process and the reconstruction one (image generation). The final image generation include simulation of an optical system (e.g. camera), multi-wavelength rendering of coloured holograms, depth evaluation for compositing of multiple objects and image enhancement (e.g. speckle noise reduction).

There are different techniques to compute CGH [6]: Point clouds, Polygonal methods that are based on convolution computation, Depth map that encodes holograms or Ray-based methods that essentially approximate the hologram by a discretized light field [7].

It is possible to improve its final quality but increase the use of computational resources as they recalculate the hologram (by modifying the phase or intensity patterns): as examples we can cite the use of neural computation [8, 9], optimization of phase holograms [10] or realistic images with global illumination [11]. These feedback techniques do not allow estimating the computational time of a hologram deterministically.

In order to reconstruct 3D images at a given camera position, the original wavefront has to be reconstructed from the hologram and propagated through space. To this end, fast, discrete approximation methods based on Fourier theory and angular spectrum are used. So, the reconstruction image process are not the bottleneck to get the final results.

Monte Carlo integration provides a comprehensive solution for achieving physically accurate lighting simulation [12]. This method revolves around employing a randomly selected subset of all rays that necessitate tracing to attain an image of near-perfect quality, minimizing any perceptible loss. In this context, crucial factors encompass the optical path traversed by the ray from its source to the CGH plane, encompassing pertinent bounces on scene objects, and the intensity of the ray.

In this context, a pertinent query arises: Can the computational expense associated with CGH synthesis be curtailed? Addressing this query, this paper proposes the concept of Partial Monte Carlo Sampling (PMCS) for CGH computation. Another pivotal inquiry follows: What is the effect of employing PMCS-based computational techniques on the image quality derived from CGH?

Our CGH allows to generate, process and render holograms from a synthetic 3D objects. Thus, the present study focuses on the evaluation of holograms generated using PMCS methodologies. The main objective is to reduce the computational burden associated with CGH generation and, at the same time, to determine the quality of the reconstructed image compared to the image obtained from a CGH using all possible rays. For completeness of the work presented, this study is performed for images obtained by simulation and in the laboratory.

Section 2 describes the point cloud selection process. Section 3 describes the PMCS algorithm used. Section 4 shows the main results. Section 5 describes the variables and procedures used to quantify the quality of the images. Sections 6 and 7 analyse and compare the results obtained.

2 CGH generation

In this work we assume a static scene illuminated by one or more monochromatic sources in which occlusions may appear. The scene is defined by means of a ray tracer with a Phong type illumination model with bouncing [13], which solves the problem of occlusions. The scene is discretized by a screen of $N \times M$ size (see Figs. 1 and 2a). In classical ray tracers, the camera is placed at a specific coordinate.

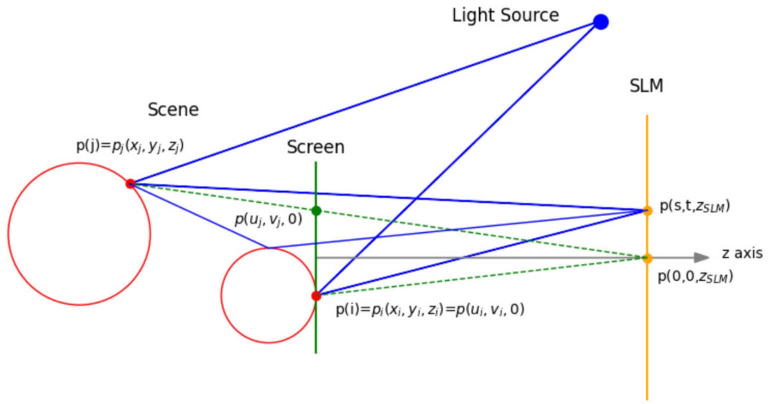
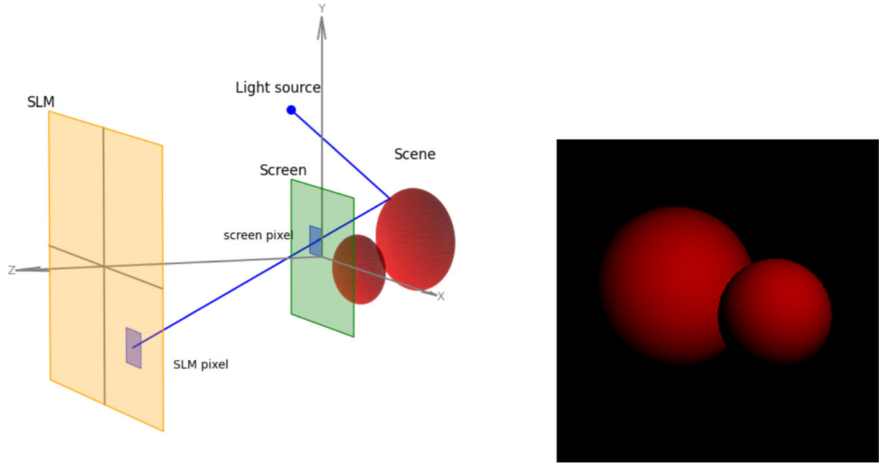


Fig. 1 Simplified geometry of wave propagation from a generic object to hologram plane. $p(u_k, v_k, 0)$ are the pixel screens to define the cloud points in scene $p_k(x_k, y_k, z_k)$ (via dashed lines). Solid lines are the calculated light path. Only central SLM pixel is shown for simplicity

In our case the SLM is an array of $W \times H$ pixels, and we can consider each pixel as a camera receiving information from the scene: SLM pixels are distributed by the coordinates (s,t) and the scene is discretized by the resolution defined by the screen (with u,v coordinates). Figure 1 shows how a ray originating at pixel (s,t) is incident on point p(j) with a direction defined by the positions (s,t) and (u,v) of both screens (dashed green lines). Each SLM pixel can define a set of up to $N \times M$ points in the scene. The same pixel p(s,t) of the SLM receives bounced rays from point p(j) or rays coming from other points in the scene (e.g. p(i)) (solid blue lines).

If we place the camera at centre of the SLM plane, the rendered scene observed through the screen is shown in Fig. 2b. In this figure we have resolved the image with a resolution of 256×256 equispaced coordinates for the screen. That is, the camera receives rays defined by this resolution and distributed on the screen.



(a) 3D Simplified geometry. See Fig. 1 for details

(b) Synthetic image of the selected scene

Fig. 2 Basics for Ray tracing process

Moreover, in classical ray tracing, it is sufficient for each ray to give information about the intensity it carries to construct the 3D scene. In the case of holography, it is necessary to know the amplitude and phase (the path travelled by each ray) to correctly sum the contributions of each ray in the camera. To design a CGH, the following must be taken into account: Each SLM pixel can be considered as a camera position so that, initially, each SLM pixel sees a different set of points in the scene. This approach is not adequate since the interferential effect requires that the information summed for each SLM pixel be consistent with those of the others.

Then a set of S_f samples must be generated over the scene that adequately covers the scene and is the discretized version of the surfaces. This process can be described with the following steps:

- Identifying a significant subset of SLM pixels that adequately cover the scene: In our case it will be the centre pixel and the corner pixels (5 SLM pixels).
- Each pixel generates a uniform network of samples of size $N \times M$, therefore there will be 5 meshes.
- S_f is the set of points on the scene obtained from these 5 meshes.
- Each pixel of the SLM is considered a camera position for all the S_f obtained from the scene.

So, we construct the list of points sources that will contribute to the hologram. This process is not taken in account to compute the time cost, due to it is previous to CGH calculations, it is calculated just one time and it is negligible compared with ray tracing cost computing.

The scene we analyze in this paper is the one shown in Fig. 2b. They are two spheres of radii 1mm and 1.5 mm and are separated on the z-axis by a distance of 12 mm between centers. The window size is 4 mm per side. The spheres are illuminated by a point source located at the spatial coordinates (5,5,10) mm. CGH can be framed in a 15.36 x 8.64 mm field that coincides with the area of a commercial SLM (PLUTO-2.1 Spatial Light Modulator: phase modulator with 8 μm pixel size) in order to compare the simulated holograms with those obtained by a real device. The resolution shown in the image is 256 pixels per side. The SLM is located 200 mm away (on the z-axis) from the closest position of the scene (see Fig. 2a) and the optical behaviour of the spheres corresponds to that of a diffuse material. It can be occlusions for several SLM pixels: not all chosen points should contribute to all SLM plane and the well known ray tracing occlusion techniques must be used.

CGH generation involves a high computational cost. In this sense, and always depending on the application to be developed, an appropriate balance must be sought between factors such as image quality, CGH size and image resolution. It must also be taken into account that CGHs are closely linked to diffractive optics, so unwanted effects such as (e.g. as typical edge patterns) may appear. These effects are strongly conditioned by the periodic structure used in the discretisation process of a scene.

In Fig. 3 we can see some of these effects: CGHs with very low resolution (a) (32 pixels per side) are fast to compute but do not give the expected image quality. As the resolution increases, “artefacts” are observed due to the diffractive behaviour of light when confronted with a periodic structure. For this, it is necessary to slightly perturb the impact positions on the screen, which significantly improves the result obtained. This effect appears in the images in the top row: (b) is the reconstruction of a CGH with 128 pixels of resolution per side to discretize the scene. (c) uses 256 pixels of resolution. In both cases the SLM resolution is 580 pixels. To minimize this effect the SLM area can be increased: (d) is equivalent to (c) but with an SLM of 1080 pixels per side. As mentioned above, it is appropriate to slightly perturb the ray hit positions on the screen to avoid this effect: Thus, the bottom row images

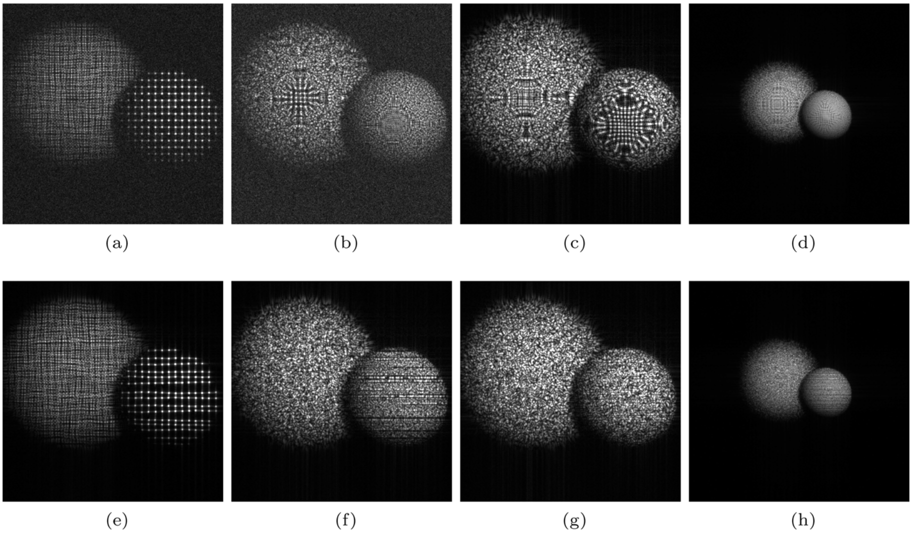


Fig. 3 Images to evaluate CGH synthesis issues (a) low resolution CGH (32 screen resolution), (b,c) diffractive artefacts due to periodic sampling (128 and 256 screen resolution), (d) Same as (c) but increasing the SLM area. (e,f,g,h) the same but using a non-periodic sampling to avoid the unwanted effects. The plane of focus is in front of the small sphere and the rear sphere is out of focus

(e), (f), (g) and (h) are the simulations obtained to their top row equivalents. It is clearly seen that the effects tned to disappear.

It is also observed that a minimum number of pixels in the SLM is necessary to properly store the scene information. Thus, by increasing from 540 to 1080 pixels per side, the result also improves significantly, decreasing the interferential speckle effects.

All these effects have been evaluated both in simulation and in the laboratory, with consistent results in both cases. Figure 4 shows that: Upper row are simulated CGH reconstructions for several screen resolutions (32, 64, 128 and 256 pixels per side) and bottom row are the same for lab images with the same resolution. In summary: to improve the quality of the image obtained from a CGH, the computation times increase significantly. The objective of this work is to evaluate Monte Carlo techniques to reduce them.

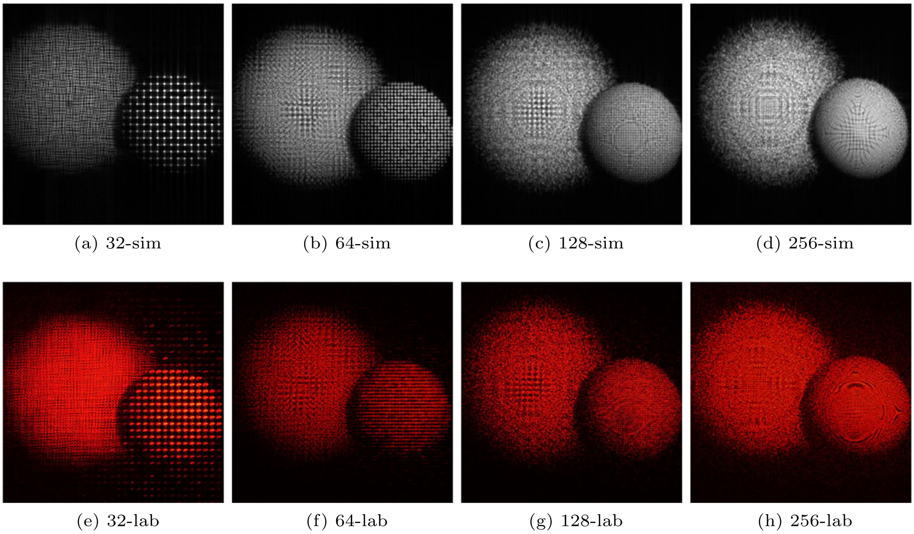
3 CGH and Monte Carlo: PMCS

The next step is propagating the light wavefronts produced on those sample points to the plane of the hologram, and accumulating them and if the sample points follow any kind of geometric layout, each one should be managed as an independent point wave source and propagated using Kirchoff or Fresnel formulae.

To generate a hologram for a 3D scene, we must record a section of the light wavefront generated by the scene on a given plane (SLM plane). This plane is discretized with $W \times H$ pixels, and each S_f scene pixels generate an spherical wave that contribute to all SLM pixels.

$$U(\vec{q}) = \sum_{S} U(\vec{p}_i) = U(\vec{p}_i) e^{j(\vec{k}.\vec{r})} \tag{1}$$

with $\vec{r} = \vec{q} - \vec{p}$, and S the set of samples over the scene.



(a) 32-sim (b) 64-sim (c) 128-sim (d) 256-sim

(e) 32-lab (f) 64-lab (g) 128-lab (h) 256-lab

Fig. 4 Lab. vs. Simulation issues at several scene resolutions

What we are really computing, for each pixel in the hologram, is the integral of the light arriving from the scene, and we sample the scene to compute the integral. Initially we use a brute force sampling method. In CG it has been proven that Monte Carlo methods are efficient to compute this kind of integrals, so we can apply them here. We define a Partial Monte Carlo Sampling (PMCS) by choosing only some random set or subset R of the S_f samples on the scene to compute the value at each pixel on the hologram. For this method to be correct, the subset must obey a known probability density function (PDF) (which defines the weight of each sample), and be different for each pixel on the hologram. Our calculation now becomes:

$$U(\vec{q}) = \sum_{R} U(\vec{p}_i) = U(\vec{p}_i)e^{j(\vec{k}.\vec{r})}.w_i \tag{2}$$

with $\vec{r} = \vec{q} - \vec{p}_i$, R the random set of samples chosen for point q, and w_i the weight of sample i.

Once the hologram is constructed, the wavefront at any other plane parallel to it can be calculated, simulating the wavefront propagation. This process of propagation between two parallel planes can be accomplished by Fourier optics, in particular with the Angular Spectrum method [ref]. This method only requires a couple Fourier transforms to propagate the wave, so it can be implemented very efficiently via FFT. If wave section U_0 (complex amplitude) is known at a $z = 0$ coordinate, the value U_z at another z position can be calculated as:

$$U_z = \mathcal{F}^{-1}(\mathcal{F}(U_o).P(z)) \tag{3}$$

where $P(z)$ is a propagation function that depends only on the distance z, and $\mathcal{F}$, $\mathcal{F}^{-1}$ are the direct and inverse Fourier transform operators.

In this method, multiple rays are shot from each pixel of the synthetic 3D scene obtained to calculate the amplitude on each SLM pixel. So, if we start from a Synthetic Image (SI) with a number of pixels $N \times M$, and the SLM has a number of pixels $W \times H$, the theoretical number of rays that must be launched from each pixel of the synthetic image would be $W \times H$ and the total rays needed to generate the hologram would be $(W \times H) \times (N \times M)$. As we increase the resolution of the images, the computational cost increases

Figure 5 shows the comparison between a synthetic scene with $N = M = 512$ pixels and the CGH result obtained (both simulated and in the laboratory) when $W = H = 1080$ pixels. This CGH involves the calculation of the order of 3.05×10^{11} individual ray paths, which is a considerable computational effort. To reduce it, two options are possible: reduce the scene resolution or choose randomly the rays used for the CGH calculation.

For the firts option, image pixels mesh are generated with 32, 64, 128 and 256 points per side. Using a SLM of 540 × 540 pixels, rays used for CGH are shown on Table 1. Same table shows the distance between image points for the different series using as measurement unit the simulated pixel size, $(8\mu m.)$. High frequency information is lost when the image resolution decreases (see Fig. 3), which affects the details of the original image as will be seen in the images in following sections.

But this work is focused in the second option: Choosing a randomly rays subset using Monte Carlo techniques, as we explain below..

3.1 PMCS algorithm

This section delineates the utilization of Monte Carlo techniques for the computational assessment of the intricate light wavefront emanating from a 3D scene when subjected to illumination. We have deliberately opted for a simplified scene that allows for the integration of targeted optimizations, without compromising the foundational insights derived from our subsequent analysis.

To undertake the integration across the complete set of discrete sources, S_f, we adopt a Monte Carlo integration approach. This methodology allows us to compute the value of the complex wavefront, W, at a given point on the hologram (x_h, y_h). The contribution $w_s(x_h, y_h)$ from each source wave s within our set of S_f is cumulatively summed at that specific point:

$$W(x_h, y_h) = \sum_{s=1}^{S_f} w_s(x_h, y_h) \tag{4}$$

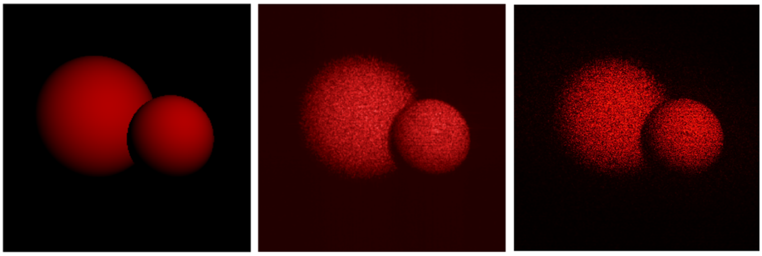

(a)Scene (b)CGH simulation. (c)CGH lab.

Fig. 5 Initial image versus CGH reconstruction, both simulated and captured from the propagation from the SLM in the laboratory. Wave propagation from SLM to object plane: -200 mm; screen size: 4mm. SLM: 1080 pixels per side, screen pixel 512 pixels. 5b and 5c are focusing on small sphere and scene is lighting with red light

To apply Monte Carlo methods, we need to rewrite the sum as the product of a volume and an average value, like

$$W(x_h, y_h) = S_f \left(\frac{1}{S_f} \sum_{s=1}^{S_f} w_s(x_h, y_h) \right) \tag{5}$$

Monte Carlo integration enables a reduction in the number of sources employed for integration. By selecting a randomized subset S_m of samples from S_f, wherein each sample is chosen according to a probability density $p(s)$, the integral can be computed as:

$$W(x_h, y_h) = S_f \left(\frac{1}{S_m} \sum_{s=1}^{S_m} \frac{w_s(x_h, y_h)}{p(s)} \right) \tag{6}$$

The selection of the probability density function $p(s)$ can significantly enhance calculation accuracy, especially if there exists some prior understanding of the distribution of $w_s(x_h, y_h)$. For instance, by favouring samples that are more likely to contribute higher values (based on a higher probability), the accuracy can be improved. In cases like ours, where a fully comprehensive complex scene serves as the source (and hence no insight into the wavefront is available), the most straightforward choice for $p(s)$ is a uniform random distribution, hence $p(s) = 1$.

3.1.1 Monte Carlo samples selection

For every pixel encompassing the hologram, a random subset of S_m samples that are suitable for Montecarlo computations. In a continuous integration realm, this process often involves selecting integration points within the domain through one (or several) uniformly distributed random variables ξ.

However, in the context of a discrete domain such as ours, this process becomes less straightforward. Opting for S_m samples from a comprehensive collection of S_f by employing a random variable ξ might inadvertently lead to duplicated selections, thereby distorting the integrity of the Montecarlo calculation. A strategy must be employed to prevent the recurrence of identical samples. This objective is achieved by employing a method known as *shuffling*:

- Sort the set of S_f samples in a random order using the *shuffle* algorithm. For each sample i in the set:
  - Choose another random position j within the set, starting from i to the end.
  - Swap samples i and j.
- Select the first S_m samples from the shuffled set of all S_f.

This process needs to be carried out for *every pixel* on the hologram. However, executing the shuffling algorithm can be computationally expensive due to its repetitive nature (it needs to be repeated billions of times). Additionally, there might be implementation challenges when using GPUs. To address these concerns, a slightly modified version of the algorithm can be employed:

- Perform the shuffling process only once at the outset, creating a set of samples sorted in a random order.
- For each pixel, select an initial random position ξ within the set of S_f samples.
- Utilize the subsequent S_m samples starting from the position ξ, with the possibility of wrapping around the set of S_f samples if necessary.

This adaptation ensures the utilization of a distinct uniform random sample set for each pixel on the hologram, a crucial requirement for the Monte Carlo algorithm.

3.1.2 Hologram calculation

To compute the CGH, it is essential to determine the values of the wavefront across the hologram plane. The hologram can be conveniently represented as an array of complex values with dimensions $H \times W$. The complete hologram calculation can be outlined as follows:

- Employing the Monte Carlo algorithm described earlier, a Monte Carlo approach is taken.
- Instead of the full calculation, a summation over a subset of the Monte Carlo samples is performed.

By employing the Monte Carlo algorithm, the hologram calculation process is transformed into a more efficient summation over a selected subset of Monte Carlo samples.

```
// Define the full set of Sf samples
samples_f = ...
// Shuffle the full set of Sf samples
samples_s = shuffle(samples_f)
// For each pixel of the hologram
for i=1 to H
        for j=1 to W
                xh = ...
                yh = ...
                // Select ramdom subset of Sm sources
                // from the shuffled sample set
                r = uniform_random_var(1,Sf);
                samples_m = samples_s[r ... r+Sm]
                // Traverse random subset of sources
                hologram[i,j] = 0
                for s=1 to Sm
                        hologram[i,j] +=
                                wave(samples_m[s],xh,yh)
                // Average
                hologram[i,j] = hologram[i,j]/Sm
                // Scale to integration volume
                hologram[i,j] = hologram[i,j]*Sf
```

4 PMCS for CGH: main results

Several scenes have been used for the analysis, in this work we present the results of the scene shown in Fig. 2b, described on Section 2, since we have observed that results are generalizable. They are two spheres framed on a square (4 mm size). It is under Phong illumination and with well-defined edges. Spheres are placed to get occlusions effects through the screen. We suppose up 3 bounds for ray.

CGHs are obtained by ray tracing, using PMCS calculations. The CGHs obtained take in account both amplitude and phase modulation. The range of rays used are shown on Table 1.

The PMCS algorithm has been implemented on a Linux PC with i9 CPU (8 cores and 16 threads). The code is parallelized using 12 threads and the time required to obtain the CGH is shown in Fig. 6, taking as reference for each series the time of the CGH using all the rays. The results in this figure show that the algorithm establishes a linear relationship between the time cost and the percentage of beams used (which we define as the PCMS rays ratio) and, therefore, it is possible to predict the time cost for any other CGH.

Measurements shown that it is possible to calculate the time cost for any other CGH. Under this conditions, the time cost for each ray is $1,35 \times 10^{-6}s$. Figure 7 shows the evolution of the scene when changing the percentage of rays used for CGH recording. A screen of 256 resolution per side has been used, although not all rays hit the objects. The simulated SLM has a resolution of 540 pixels per side, each pixel has a size of 8 um. The sphere is illuminated with a 632.8 nm beam. Although not explicitly shown in the paper, series of 32, 64 and 128 pixel resolution have been performed. As with the images shown in Figs. 4 and 7, the results obtained in the laboratory are consistent with the simulations.

5 PMCS for CGH and quality measurements

To organize the assessments of image quality, we address three variables pertinent to Computer-Generated Hologram (CGH) synthesis employing PMCS:

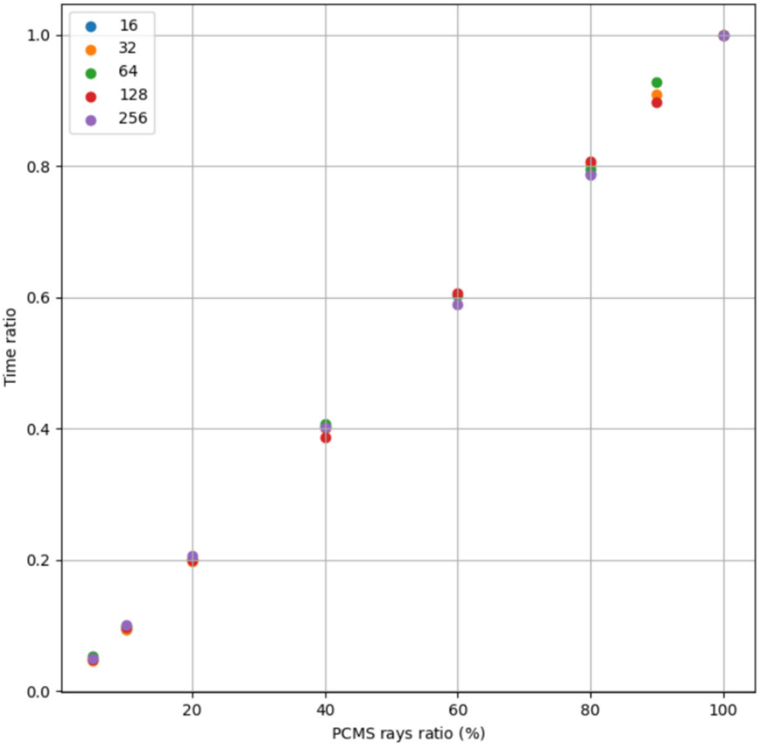
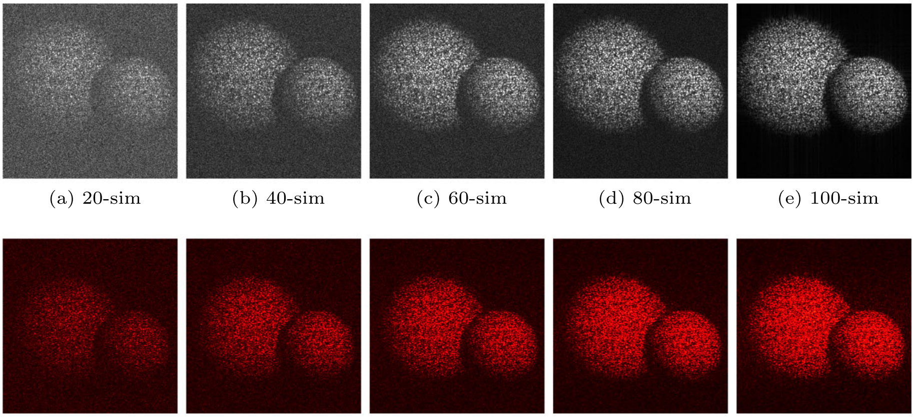


Fig. 6 Time cost for CGH Series (all measurements in s.). PCMS percentage vs. size screen. SLM size 540 × 540



(a) 20-sim (b) 40-sim (c) 60-sim (d) 80-sim (e) 100-sim

(f) 20-lab (g) 40-lab (h) 60-lab (i) 80-lab (j) 100-lab

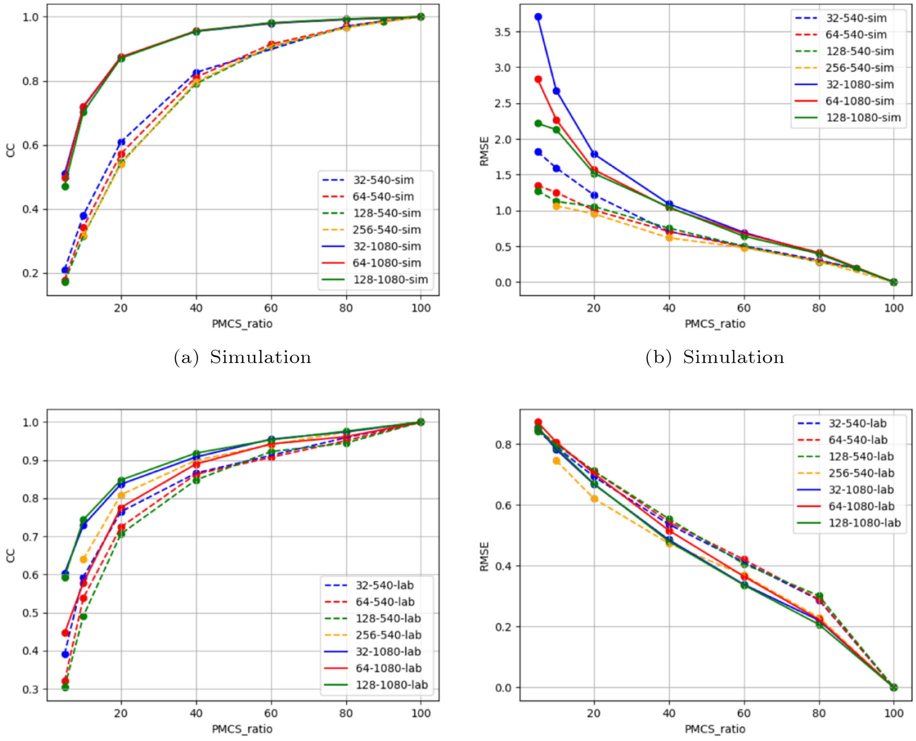
Fig. 7 PMCS series. Wave propagation from SLM to object plane: -200 mm; screen size: 4mm. SLM: 540 pixels per side, screen pixel 256 pixels. Top line: Simulation CGH images for serveral PMCS ratio. Bottom line: Lab. CGH images. Focusing plane is on small sphere front. The number in the figure indicates the percentage of rays used

1. The percentage of rays used in the PMCS algorithm. This is referred to as the *PMCS ratio*. This parameter involves employing a randomized subset of all the rays essential for generating an ideally pristine image, while still achieving a satisfactory level of final quality. This parameter will be represented on the abscissa axis on Fig. 8 graphics.
2. The chosen resolution for the scene, termed the *PMCS resolution*. It is also insightful to explore the impact of image behaviour in relation to the resolution utilized for discretizing the object. Notably, the sampling interval of an image the number of points at which it is digitally encoded also governs the computational time required for CGH. This parameter will be represented by the colours of the curves. This parameter will be represented on the abscissa axis of the same figures.
3. Comparison between simulated CGH and the CGH displayed in the laboratory environment: Simulated CGHs encompass both amplitude and phase modulation. Conversely, CGHs employed in the laboratory are restricted to phase modulation alone, owing to the prevailing limitations of available Spatial Light Modulators (SLMs). This parameter is identified by the style of lines in the graphs: solid lines correspond to images obtained in the laboratory, dashed lines correspond to images obtained by simulation.

These variables provide a comprehensive framework for assessing the quality and efficacy of CGH synthesis through the PMCS technique. No enhancement algorithms have been applied in our study to improve the final image quality. The primary objective of this research is to comprehensively assess the efficiency and effectiveness of the proposed methodologies across various variables. These variables are key components in the calculation of a Computer-Generated Hologram (CGH) using ray tracing through free space propagation.

In the context of comparisons between holograms, the reference image has been established as the image obtained from the hologram calculated using all rays, representing a “perfect” hologram (100% ray utilization).

To facilitate the comparison between an image I and a reference image R, several evaluation metrics can be employed. One of these is the root-mean-square error (RMSE), expressed as shown in (7), where I and R denote the two images being compared, p and q represent



(a) Simulation

(b) Simulation

(c) Lab

(d) Lab

Fig. 8 CC and RMSE for simulation and lab measurements

pixel coordinates, $\bar{I}$ and $\bar{R}$ denote the mean values of images I and R, respectively, and $I_{p,q}$ signifies the pixels at p, q in the image under comparison with the reference image $R_{p,q}$. Notably, the RMSE definition has been slightly adjusted to allow for the comparison of series with distinct reference images.

$$RMSE = \sqrt{\frac{\sum_{N,M}^{p,q}(I_{p,q} - R_{p,q})^2}{\sum_{N,M}^{p,q}(R_{p,q})^2}} \tag{7}$$

Another well-known evaluation function is the Correlation Coefficient (CC) defined as

$$CC = \frac{\sum_{N,M}^{p,q}(I_{p,q} - \bar{I})(R_{p,q} - \bar{R})}{\sqrt{(\sum_{N,M}^{p,q}(I_{p,q} - \bar{I})^2)(\sum_{N,M}^{p,q}(R_{p,q} - \bar{R})^2)}} \tag{8}$$

An RMSE value of 0 indicates that the two images are identical. Conversely, the Correlation Coefficient (CC) ranges from 0 to 1, where 0 signifies a lack of correlation between the images, while 1 implies a precise match between them. In subsequent sections, we will employ these metrics to assess alterations in the quality of the reconstructed image resulting from modifications in PMCS parameters such as resolution or ray ratio conditions.

6 Discussion

Computer-generated graphics enable the transmission of ideas into our minds, bridging the gap between concepts and objects, be they real or virtual. This paper focuses on generating physically accurate holograms through simulation, replicating the intricate interplay between light and matter. The holography phenomenon is modelled, and scenes are rendered with photorealism.

The quest for well-crafted images raises questions: What defines image quality, ensuring efficient and accurate information transfer from computers to our perception?

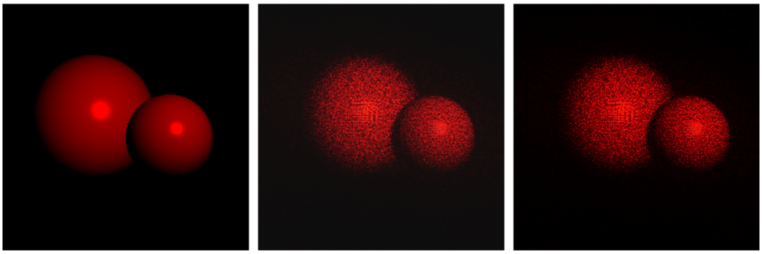
Amidst the diverse approaches within computer graphics and visualization tools, finding answers is intricate. Here, our focus is on simulating holograms using various image resolutions (points/side) and ray percentages to calculate amplitude and phase. Reconstructed images and real-world projections onto commercial SLMs corroborate the potential of achieving reasonable quality with fewer rays. This alleviates certain diffraction-related effects, surpassing the computational load of alternate proposals.

Quality measurements (RMSE and CC) reference two images per series: the CGH derived from 100% rays and the image of the actual object. Both simulated and laboratory images undergo assessment, revealing that image quality can remain satisfactory even with fewer rays. Noise dominates image detail for low ray percentages in the PMCS algorithm. Conversely, high ray percentages expose the diffraction effects of each point.

Prior research indicates that beyond a threshold, observers perceive minimal image quality improvement [14, 15]. Our results align with this observation. For simulated images, correlation coefficients remain relatively constant after a ray percentage threshold (Fig. 8a) when CGH pixels increase. This is related to the fact that the higher the number of pixels in the CGH, the more scene information is stored. Figure 8b also shows this effect, although in this case, the RMSE is worse when the size of the CGH increases (especially at low PMCS values) due to the increased number of elements to be added, quantifying the noise effects observed in these cases.

Other factors must be taken into account to correctly interpret the experimental results for CC (Fig. 8c) and RMSE (Fig. 8d) of the PMCS images, such as the transfer function introduced by the phase modulation of the SLM or the CCD camera sensor in the measurement process. They are not the subject of this study but nevertheless the curves maintain the behaviour observed in the simulations (Fig. 8a and b).

There are other factors that can affect the behaviour of these measurements, such as the overall brightness level of the image or the energy carried by each beam into the CGH. This



(a) Scene (b) CGH simulation. (c) CGH lab.

Fig. 9 Scene under diffuse and specular illumination. SLM: 1080 pixels per side, screen pixel 256 pixels. 9b and 9c are focusing on small sphere and all scenes are lighting with red light

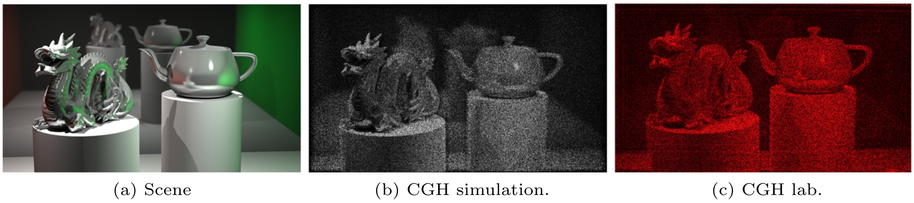

(a) Scene (b) CGH simulation. (c) CGH lab.

Fig. 10 Complex scene. SLM and screen: 1920 x 1080 pixels. Scene illuminated with four point sources. Diffuse and specular lighting. 10b and 10c are focusing on dragon head and it is used monochromatic light to recover the scene from CGH

analysis is beyond the scope of the present work and must be taken into account if a more complete model is to be obtained. All these results are also found when we use specular and ambient lighting for the scene (see Fig. 9).

We have attached a more complex 3D scene (Fig. 10) in terms of the defining elements of the geometry and the lighting model used [12]. This scene has been calculated using the same code but significantly increasing the number of threads. Figure 10 shows the synthetic scene (10a), simulated (10b) and obtained in the laboratory (10c), and it is verified that the scalability potential of PMCS remains linear.

7 Conclusions

In this work we present the concept of Partial Monte Carlo Sampling (PMCS) to generate and obtain Computer Graphics Holograms (CGH) with a lower computational cost, being able to select the quality of the obtained results measured by the Root Mean Square Error (RMSE) and the Correlation Coefficient (CC) of the final propagated CGH images. The algorithm has been tested with various scene resolutions and illumination types, demonstrating its linear behaviour with respect to the workload for each CGH, so that the computation time can be predicted according to the specific characteristics of each hologram. Scene point selection resolves the occurrence of unwanted interference patterns by perturbing the initial propagation directions calculated for each scene.

All the results obtained by simulation of a Spatial Light Modulator (SLM) (with amplitude and phase modulation) have been verified in the laboratory with an state of the art SLM (phase only), obtaining very good agreement.

The paper describes in detail the point cloud selection process, the PMCS algorithm used, and shows the main results. It shows some of the images generated, the variables and procedures used to quantify the quality of the images, and analyses and compares the results obtained.

Future work should include the influence of global illumination on the quality measurements and the formalization of a minimum subset of SLM points to obtain a full parallax effect.

It has been verified that the final quality depends on speckle and other effects related to the diffractive behavior of light. In this work, only the direction of each ray has been randomly perturbed to eliminate the diffractive artefacts that appear.

The results obtained in Fig. 10 clearly show that the effects related to speckle condition the quality of the final scene obtained and need to be corrected. Studying the effect of PMCS on speckle behavior is a line of work to continue in future studies.

Given the linear behavior of the method, solving dynamic scenes does not affect the algorithm's behavior and will depend on the available computing capacity. PMCS saves computation time while maintaining the selected quality of the obtained images. So, PMCS method can work with any type of 3D scenes, lighting models, and number of sources.

Acknowledgements The authors would like to thank Prof. Dr. Juan Magallón (Department of Informatics at the University of Zaragoza) for his contribution in the design and synthesis of Fig. 10a.

Funding Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature.

Availability of data and materials Both simulated and laboratory images are available on request.

Declarations

Conflicts of interest The authors declare that they have no conflict of interest.

References

1. Slinger C, Cameron C, Stanley M (2005) Computer-generated holography as a generic display technology. Comput 38(8):46–53. https://doi.org/10.1109/MC.2005.260
2. Ziegler G, Bucheli S, Ahrenberg L, Magnor M, Gross M (2007) A bidirectional light field - hologram transform. Proc. Eurographics. 26:435–446
3. Lucente M, Galyean TA (1995) Rendering interactive holographic images. In: Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques. SIGGRAPH '95, pp 387-394. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/218380.218490. https://doi.org/10.1145/218380.218490
4. Matsushima K (2005) Computer-generated holograms for three-dimensional surface objects with shade and texture. Appl. Opt. 44(22):4607–4614. https://doi.org/10.1364/AO.44.004607
5. Ziegler R, Kaufmann P, Gross M (2007) A framework for holographic scene representation and image synthesis. Trans Vis Comput Graph IEEE Trans. 13(2):403–415. https://doi.org/10.1109/TVCG.2007.27
6. Blinder D, Ahar A, Bettens S, Birnbaum T, Symeonidou A, Ottevaere H, Schretter C (September 2018) Schelkens P (2019) Signal processing challenges for digital holographic video display systems. Signal Process Image Commun. 70:114–130. https://doi.org/10.1016/j.image.2018.09.014
7. Sahin E, Stoykova E, Mäkinen J, Gotchev A (2020) Computer-Generated Holograms for 3D Imaging: A Survey. ACM Comput Surv. 53(2). https://doi.org/10.1145/3378444
8. Shi L, Li B, Kim C, Kellnhofer P, Matusik W (2021) Towards real-time photorealistic 3D holography with deep neural networks. Nat 591(7849):234–239. https://doi.org/10.1038/s41586-020-03152-0
9. Peng Y, Choi S, Padmanaban N, Kim J, Wetzstein G (2020) Neural Holography. ACM SIGGRAPH 2020 Emerging Technologies. SIGGRAPH 2020:2–3. https://doi.org/10.1145/3388534.3407295
10. Chen L, Zhang H, Cao L, Jin G (2020) Non-iterative phase hologram generation with optimized phase modulation. Opt Express 28(8):11380. https://doi.org/10.1364/oe.391518
11. Blinder D, Chlipala M, Kozacki T, Schelkens P (2021) Photorealistic computer generated holography with global illumination and path tracing. Opt Lett 46(9):2188. https://doi.org/10.1364/ol.422159
12. Pharr M, Humphreys G (2010) Physically Based Rendering: From Theory To Implementation. pp 1-1171. Elsevier Inc., ???. https://doi.org/10.1016/C2009-0-30446-8
13. Phong BT (1975) Illumination for computer generated pictures. Comput Sci Commun ACM 18(6):311–317. https://doi.org/10.1145/360825.360839
14. Navarro F, Castillo S, Serón FJ, Gutierrez D (2011) Perceptual considerations for motion blur rendering. ACM Trans Appl Percept 8(3). https://doi.org/10.1145/2010325.2010330

15. Reddy M (2001) Perceptually optimized 3D graphics. IEEE Comput Graph Appl 21(5):68–75. https://doi.org/10.1109/38.946633