

Partial Monte Carlo sampling for computer generated holograms

Alfonso Blesa¹  | Juan Antonio Magallón²  | Francisco José Serón² 

¹Department of Electronics and Communication Engineering, University of Zaragoza, Escuela Universitaria Politécnica, Teruel, Aragón, Spain

²Department of Computer Sciences, University of Zaragoza, Zaragoza, Aragón, Spain

Correspondence

Alfonso Blesa, Department of Electronics and Communication Engineering, University of Zaragoza, Escuela Universitaria Politécnica, Teruel, Aragón, ES 44003, Spain.

Email: ablesa@unizar.es

Funding information

Gobierno de Aragón, Grant/Award Number: T33-20D

CGH is considered as the key technology to integrate synthetic 3D imaging to get best photo-realistic rendering quality and optimum viewing experience. We present a partial Monte Carlo sampling algorithm that uses a random subset of rays for the calculation of the CGH, reducing the computational cost. The similarity of the images obtained from the CGH with respect to the original one is evaluated as a function of the image resolution and the percentage of rays used. The results obtained are presented for both simulation and experimental set-up.

KEYWORDS

CGH, holography, Monte Carlo sampling, ray tracing

1 | INTRODUCTION

As is well known, two essential requirements for future 3D displays will be the following: (1) to display an image of a scene/object equally magnified to its size in all three dimensions, and (2) to provide continuous parallax in all directions, so that viewers do not suffer from any accommodation and convergence conflict and there is no optimal viewing distance when viewing the 3D image.

At the moment there are two technologies that exist that can meet these requirements: light field and holography. They have a very similar optical geometry but their light fields in front of the display panel are not the same.¹ This difference is induced from the fact that the light field is formed by the overlapping between expanding beams of pixel cells for light field display but coherent addition of rays from each pixel with those from other pixel cell in holographic displays.

Holography is considered to be the natural 3D visualization technology. In essence, a hologram modifies an input wave (typically a plane beam) by modifying its amplitude and phase in such a way that the desired wavefront (a real or synthetic object) is reproduced.

Currently, there are two limitations that prevent progress in the generation of realistic synthetic hologram 3D images: The first is the SLM (spatial light modulator display) technology, its size and resolution and the possibility of modulating both amplitude and phase.² They cannot perform as expected due to the lack of technical supports from optical components and display panels. The most of these problems are caused by the finite pixel's sizes of the currently available displays and the resolving power of viewing zone forming optics.³ Another limitation is that current devices cannot

simultaneously represent the amplitude and phase modulation required by ideal holograms. Finally, the small size of these devices does not make them suitable for commercial applications.

But pending the arrival of a suitable technology, progress can be made in solving the problems related to the synthesis of computer generated holograms (CGH). Therefore, in some cases, it is necessary to simulate the process that generates a scene from the CGH. This process involves the CGH calculation and its subsequent reconstruction when it is illuminated by a reconstruction wave.

The second drawback is the computational cost of generating holograms. As in the case of graphics, the amplitude and phase pattern of the CGH can be obtained from a model by means of calculations.

There are different techniques to calculate CGH:⁴ Point cloud represent objects as a collection of discrete luminous points, Polygon methods are based on convolution calculus (e.g., the angular spectrum method), depth map encoding holograms and ray-based methods that, essentially, approximate the hologram by a discretized light field.⁵ It is possible improve its final quality but increase the use of computational resources (CPU/GPU usage time, memory ...) as they recalculate the hologram (modifying the phase or intensity patterns): as examples we can cite the use of neural computing,^{6,7} phase hologram optimization⁸ or realistic images with global illumination.⁹ These feedback techniques do not allow estimating the computation time of a hologram in a deterministic way.

Monte Carlo integration offer the most general solution to physically accurate lighting simulation.¹⁰ It is based on using a random subset of all the rays that should be traced to obtain a perfect image with no obvious loss in the final quality obtained. In this case, the relevant parameters are the optical path of the ray from the source to the CGH plane (including the corresponding bounces on the objects in the scene) and the intensity of the ray.

Given this scenario, the question arises: can the use of computational resources for CGH synthesis be reduced? The answer to this question proposed in this article is defined as partial Monte Carlo sampling (PMCS) for CGH. Another question is: How does the quality of the image obtained from a CGH change using PMCS-based computational techniques? Holography naturally reproduces three-dimensional scenes, but the commonly used image quality assessment functions consider point arrays (2D) from 3D scenes. Therefore, to answer this question we have restricted our CGH analysis to two-dimensional scenes obtained by synthetic or real photography of a three-dimensional world.

So, Present work evaluates the holograms calculated using PMCS techniques. The goal is to reduce the computational cost associated with CGH generation by controlling the quality of the image reconstructed from it, in terms of the similarity between the images obtained from the CGH and the original.

Section 2 describes the PMCS algorithm used. Section 3 describes the variables and procedures used to quantify the quality of the images. Section 4 shows the results of the images obtained in the simulations. Section 5 shows the results of the images obtained in the laboratory. Sections 6 and 7 analyse and compare the results obtained. The Annex shows some of the images generated for this work.

2 | CGH AND MONTE CARLO: PMCS

Generation of a CGH for a complex computer model of a three-dimensional scene (that simulates a real world-scene) involves two fundamental steps: (a) Selecting the sample points on the scene, that will act as light sources: even if the scene can have a continuous model, computer rendering of the scene needs to discretize it, so only a number of points in the scene will contribute to the generation of the hologram. (b) Propagating the light wavefronts produced on those sample points to the plane of the hologram, and accumulating them.

Both processes are related, so distribution of selected points can favor or disfavor different techniques of propagation. For example, if all selected points lie on a plane parallel to the hologram plane, some kind of plane wavefront propagation could be used (such as direct convolution, FFT convolution or angular spectrum methods). If the sample points follow any kind of geometric layout, each one should be managed as an independent point wave source and propagated using Kirchhoff or Fresnel formulae.

In either case, there is an additional problem: not all points selected in the scene contribute to all the plane in the hologram, as occlusions can exist. This occlusion problem can be solved by means of ray-tracing techniques: for each light source we must decide if it illuminates a given pixel in the hologram, throwing a ray from the point to the pixel. This kind of direct simulation is very inefficient (as it was already found for non-holographic ray tracing rendering techniques). So we must use an inverse method: first, for each pixel in the hologram, throw some rays into the scene, that will define the point sources that are visible from that pixel. Then, add the contribution of those sources to the hologram pixel, as independent point sources.

In order 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 (CGH plane). Information recorded on this section plane will allow later to reconstruct (recover) the full light wavefront, as originally generated by our scene.

This wavefront is the sum of all the light coming from our scene (see Figure 1), that is inherently continuous, but can be discretized as a number S of source points, considering each of them as a point light source which generates an spherical wave. The value of the wavefront at the hologram section will be calculated as the sum of all those spherical waves, evaluated at the hologram plane.

The hologram plane is also discretized (with a resolution of $W \times H$ pixels), so each spherical wave must contribute to every pixel. For each wave, we use the classical formula for the complex amplitude of an spherical wave, so the value for the wave at point in hologram is obtained as:

$$U(\vec{q}) = \sum_S U(\vec{p}_i) = U(\vec{p}_i) e^{j(\vec{k} \cdot \vec{r})} \quad (1)$$

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

For each point \vec{q} in hologram, we must accumulate contributions for all source points S . That accounts for a total of $S \times W \times H$ operations. Quality of the result depends on how precisely the scene is discretized, that is, how many points S are used to sample the scene. For large numbers of samples S this can become very time-consuming.

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 PMCS by choosing only some random subset R of the S 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:

$$U(\vec{q}) = \sum_R U(\vec{p}_i) = U(\vec{p}_i) e^{j(\vec{k} \cdot \vec{r})} \cdot w_i \quad (2)$$

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].

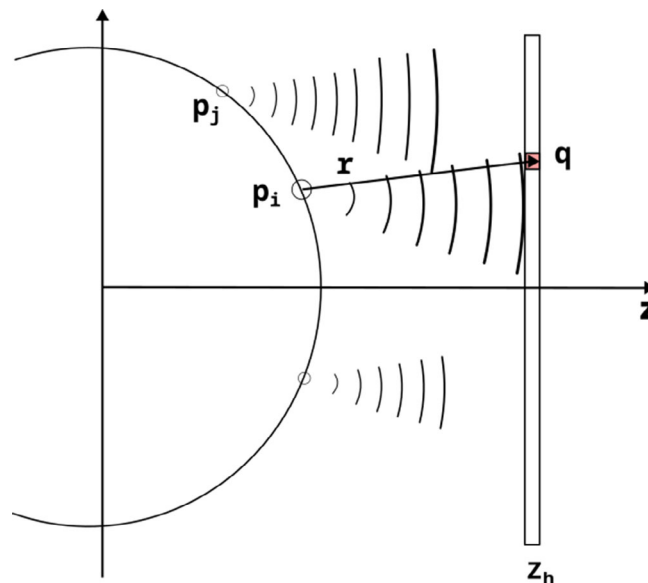


FIGURE 1 Simplified geometry of wave propagation from a generic object to hologram plane.

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:

$$U_z = F^{-1}(F(U_0).P(z)), \quad (3)$$

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

For now, the goal with this PMCS algorithm is to have a static scene (i.e., geometry and materials do not change over time) and test if it is possible to generate an image in real time. This requirement can be defined as a target FPS (frames per second) render rate. It ranges from 25 FPS for traditional cinema, 50 FPS for computer displayed images, and up to 100 or 200 FPS for a high-quality display. The higher the rate, the more faithfully movement is perceived. A detailed cost estimation analysis for CGH calculations are compiled in Reference 11. The main problem to achieve the desired results is the calculation time, which depends directly on the number of rays used to calculate the scene.

In this method, multiple rays are shot from each pixel of the synthetic 2D image obtained to calculate each pixel on the SLM. 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. Let's assume a flat synthetic scene of size 1024×1024 points of which we want to calculate its hologram to display on a screen of 1920×1080 pixels. Each of the object's points must send amplitude and phase information to all the pixels of the hologram. This amounts to a total of 2174×10^{12} rays.

Image pixels mesh are generated with 32, 64, 128, 256, 512, and 1024 points per side. 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, ($8\mu\text{m}$). High frequency information is lost when the image resolution decreases (see Table 2), which affects the details of the original image as will be seen in the images in following sections.

2.1 | PMCS algorithm

This section describes the application of Monte Carlo techniques to the calculation of the complex light wavefront generated by a two-dimensional scene under illumination. In order to analyse the effects of Monte Carlo sampling (which will be later detailed) we choose a simple scene that allow to perform some optimizations that do not affect the conclusions of the analysis.

The scene is a set of continuous surfaces in space generating a wavefront that travels to the eye of the observer (or the optic system of a recording setup). Our goal is to register that wavefront over a planar surface, our hologram (H). Following the Huygens principle, this continuous wavefront could be modeled at his origin (the surfaces on our scene) as a set of point sources, ideally infinite in number. We should just integrate over all those sources to get the value of the wavefront at each point of H. That integral is discretized, so in fact we use a set of samples over the scene, with their amplitude adjusted according to the surface area they represent.

To perform the integral over the *full* set of S_f discrete sources, we implement a Monte Carlo integration method. In order to calculate the complex wavefront value W at a point on the hologram (x_h, y_h) , the contribution $w_s(x_h, y_h)$ of each source wave s from out set of S_f on that point must be summed:

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

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

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

TABLE 1 Rays used to CGH calculus versus percentage of PCMS rays used (K) and sampled image size ($N \times N$ points).

K (%)	$N \times N$ (pixels mesh)					
	Distance between mesh points (units: CGH pixel)					
	32^2	64^2	128^2	256^2	512^2	1024^2
	27.34	13.67	6.83	3.42	1.71	0.85
5	106.10 ⁸	425.10 ⁸	170.10 ⁹	679.10 ⁹	272.10 ¹⁰	109.10 ¹¹
10	212.10 ⁸	849.10 ⁸	340.10 ⁹	136.10 ¹⁰	544.10 ¹⁰	217.10 ¹¹
15	319.10 ⁸	127.10 ⁹	510.10 ⁹	204.10 ¹⁰	815.10 ¹⁰	326.10 ¹¹
20	425.10 ⁸	170.10 ⁹	679.10 ⁹	272.10 ¹⁰	109.10 ¹¹	435.10 ¹¹
25	531.10 ⁸	212.10 ⁹	847.10 ¹⁰	34.10 ¹⁰	136.10 ¹¹	544.10 ¹¹
30	637.10 ⁸	255.10 ⁹	102.10 ¹⁰	408.10 ¹⁰	163.10 ¹¹	652.10 ¹¹
35	743.10 ⁸	297.10 ⁹	119.10 ¹⁰	476.10 ¹⁰	190.10 ¹¹	761.10 ¹¹
40	849.10 ⁸	340.10 ⁹	136.10 ¹⁰	544.10 ¹⁰	217.10 ¹¹	870.10 ¹¹
45	956.10 ⁸	382.10 ⁹	153.10 ¹⁰	612.10 ¹⁰	245.10 ¹¹	978.10 ¹¹
50	106.10 ⁹	425.10 ⁹	170.10 ¹⁰	679.10 ¹⁰	272.10 ¹¹	109.10 ¹²
55	117.10 ⁹	467.10 ⁹	187.10 ¹⁰	747.10 ¹⁰	299.10 ¹¹	120.10 ¹²
60	127.10 ⁹	510.10 ⁹	204.10 ¹⁰	815.10 ¹⁰	326.10 ¹¹	130.10 ¹²
65	138.10 ⁹	552.10 ⁹	221.10 ¹⁰	883.10 ¹⁰	353.10 ¹¹	141.10 ¹²
70	149.10 ⁹	595.10 ⁹	238.10 ¹⁰	951.10 ¹⁰	381.10 ¹¹	152.10 ¹²
75	159.10 ⁹	637.10 ⁹	255.10 ¹⁰	102.10 ¹¹	408.10 ¹¹	163.10 ¹²
80	170.10 ⁹	679.10 ⁹	272.10 ¹⁰	109.10 ¹¹	435.10 ¹¹	174.10 ¹²
85	180.10 ⁹	722.10 ⁹	289.10 ¹⁰	116.10 ¹¹	462.10 ¹¹	185.10 ¹²
90	191.10 ⁹	764.10 ⁹	306.10 ¹⁰	122.10 ¹¹	489.10 ¹¹	196.10 ¹²
95	202.10 ⁹	807.10 ⁹	323.10 ¹⁰	129.10 ¹¹	516.10 ¹¹	207.10 ¹²
100	212.10 ⁹	849.10 ⁹	340.10 ¹⁰	136.10 ¹¹	544.10 ¹¹	2174.10 ¹²

Note: CGH size: $W = 1920$, $H = 1080$ pixels.

TABLE 2 Loss of information when decreasing image resolution from Nyquist theorem.

$N \times N$ (mesh)	32^2	64^2	128^2	256^2	512^2	1024^2
Δx	256	128	64	32	16	8
f_N	$16 \times f_s$	$32 \times f_s$	$64 \times f_s$	$128 \times f_s$	$256 \times f_s$	$512 \times f_s$
Loss factor	0.03125	0.0625	0.125	0.25	0.5	1

Note: $X = 7$ mm: horizontal image size. $f_s = 1/X$: horizontal max object frequency. f_N : Nyquist frequency. $\Delta x = \frac{X}{N}$ (μm) CGH pixel: $8\mu\text{m}$.

Monte Carlo integration allows to reduce the number of sources used for the integration: choose a random subset S_m of samples from S_f , where each sample is chosen with a probability density $p(s)$, and calculate the integral as

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

The choice of the probability density $p(s)$ can help to enhance the accuracy of the calculation, if some *a priori* knowledge on the distribution of $w_s(x_h, y_h)$ is available (i.e., choose the samples which will contribute a higher value with higher

probability). In our case, where a totally general complex scene could be the source (so no clue about the wavefront can be used) the last resort for $p(s)$ is the uniform random distribution, so $p(s) = 1$.

In a complex scene the full set of S_f sampled sources in the scene will be determined *implicitly* from a rendering setup where both scene and point of view will define which points in the scene contribute to the hologram. For the tests in this article, as we focus on the effect of the Monte Carlo sampling, we use a simplified scene with just one surface (a square, possibly textured), and the full set of S_f sources is *explicitly* defined.

2.1.1 | Discretization of the scene

First step is to generate the S_f samples for the full discretized version of our surface. The process can be described by the following steps:

- Resolution to sample our square is chosen, let it be N .
- A uniform sampling grid of size $S_f = N \times N$ samples is layered on the square, generating points (u_i, v_j) on parametrical coordinates on the range $[0, 1] \times [0, 1]$.
- For each sampling point (u_i, v_j) , an spherical source is generated, with origin at the point in the square determined by (u_i, v_j) , and with amplitude A proportional to its area.
- In case the square should be textured, the amplitude A is modulated by a gray level extracted from the texture at coordinates (u_i, v_j) , via point sampling.

So, we construct the list of point sources that will contribute to the hologram. This process is not taken in account to compute the time cost, because is previous to CGH calculations.

2.1.2 | Monte Carlo samples selection

Next step is to select, for each pixel on the hologram, a random subset of S_m samples suitable for Monte Carlo calculations. For a continuous domain of integration, this process is usually done selecting integration points in the domain with one (or several) uniform random variables ξ .

For a discrete domain like ours, this process is not so simple. Choosing S_m samples from a full set of S_f using a random variable ξ can lead to selection of duplicate samples, distorting the Monte Carlo calculation. A method to avoid repetition of samples must be used. This is accomplished with a method called *shuffling*:

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

This process should be performed for *every pixel* on the hologram. Execution of the shuffling algorithm is expensive (it must be repeated billions of times), and can lead to some implementation problems when GPUs are used. To solve this problems, a slightly modified version can be used:

- Perform the shuffling only one time at the beginning of the process, giving a randomly sorted set of samples.
- For each pixel, choose a random starting point ξ inside the set of S_f samples.
- Use then next S_m samples from ξ position (possible wrapping around the set of S_f samples).

This process guarantees that a different uniform random set of samples is used for each hologram pixel, as the Monte Carlo algorithm requires.

2.1.3 | Hologram calculation

To compute the CGH, all values of the wavefront at the hologram plane must be calculated. The hologram can be represented as an array of $H \times W$ complex values. The full calculation of the hologram can be depicted with this algorithm:

```
// Define the full set of Sf samples
samples_f = ...
// For each pixel of the hologram
for i=1 to H
  for j=1 to W
    // Determine position
    xh = ...
    yh = ...
    // Traverse full set of sources
    hologram[i,j] = 0
    for s=1 to Sf
      hologram[i,j] += wave(samples_f[s],xh,yh)
    // Average
    hologram[i,j] = hologram[i,j]/Sf
    // Scale to integration volume
    hologram[i,j] = hologram[i,j]*Sf
```

where `wave()` is a function that calculates the contribution of an individual point source to the hologram. Using our Monte Carlo algorithm, this can be rewritten as:

```
// 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 random 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
```

Figure 2 shows the flowchart of this pseudocode.

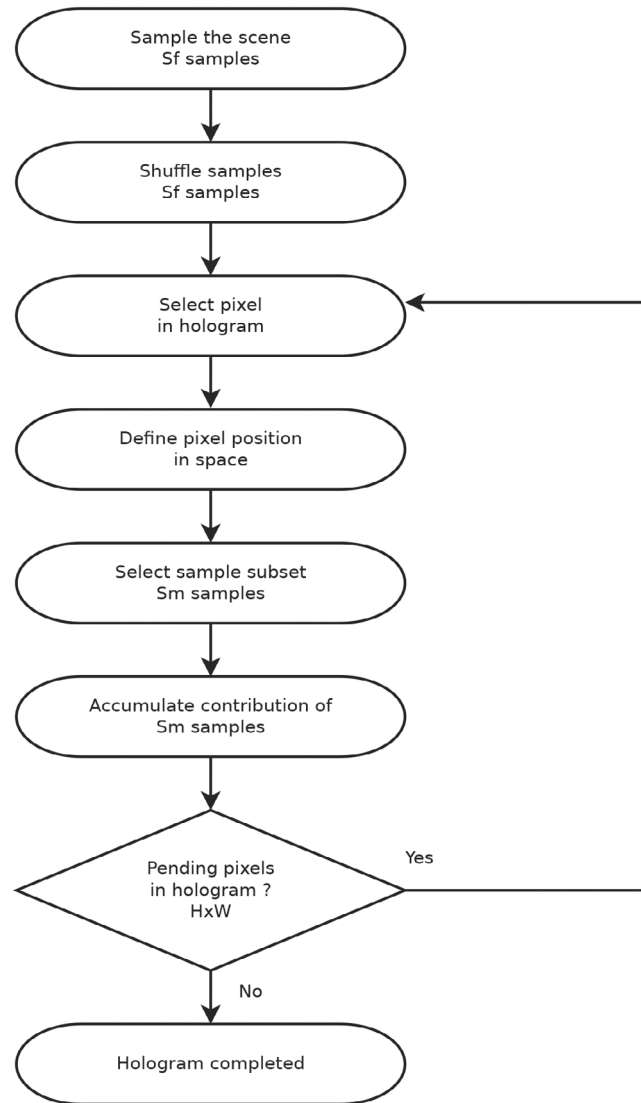


FIGURE 2 Monte Carlo algorithm flowchart.

3 | PMCS FOR CGH AND QUALITY MEASUREMENTS

In order to arrange the image quality measurements, we consider three variables related to CGH synthesis using PMCS:

1. The number of beams used on PMCS algorithm. *PMCS ratio*: it is based on using a random subset of all the rays that should be traced in order to obtain a perfect image with no obvious decrease in the final quality obtained.
2. The resolution considered for the scene. *PMCS resolution*: it is also interesting to analyse the behavior of the images as a function of the resolution used to discretize the object: the sampling period of an image, that is, the number of points at which it is encoded, also determines the computation time for the CGH.
3. CGH simulated versus CGH displayed in the lab: simulated CGHs have been modeled including amplitude and phase modulation. CGHs used in the laboratory use only phase modulation, due to the limitations of current SLMs.

We have not used any algorithm to improve the quality of the final image. The aim of the study is to know the efficiency and effectiveness of the proposed methods as a function of these variables when calculating a CGH by free space propagated ray tracing.

Two options has been considered as reference image when making comparisons between holograms:

1. The image obtained from the hologram calculated with all possible rays (100% of rays), that is, the image obtained from a “perfect” hologram.
2. The object that generates the hologram (the original image).

To compare an image I with image R (considered as reference image), some evaluation functions can be used. The first one is root-mean-square error (RMSE) (Equation 7), defined as. where I and R are the two images to compare, p and q are pixel coordinates, \bar{I} and \bar{R} are the mean values of I and R respectively, $I_{p,q}$ are the pixels p, q of the image to be compared with the reference image defined as $R_{p,q}$. RMSE used definition is slightly modified in order to compare series with different reference image.

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

Another well-known evaluation function is the correlation coefficient (CC) defined as

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

When $\text{RMSE} = 0$ means that both images are the same. The CC ranges from 0, which means that there is no correlation between images, to 1, which represents an exact match. We will use them in the following sections to compare the quality of the reconstructed image changes modifying PMCS parameters as resolution or ray ratio conditions.

4 | PMCS FOR CGH: SIMULATION

Several images have been used for the analysis, in this work we present the results of the image shown in Figure 3, since we have observed that results are generalizable. It is a calculator and has been chosen for the following characteristics: it is an object of 7 mm side, with fine details (e.g., the numbers) and with well-defined edges. Both, details and edges of the image (which are step functions in the intensity function) are strongly affected by the diffractive effects implicit in hologram generation.

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 for each series is from 5% to 100% in increments of $\Delta 5\%$ (see Table 1).

From this object the ideal hologram has been calculated with various resolutions points/side (see Table 3). Both the CGH plane and the reconstructed images are framed in a 15.36×8.64 mm field that coincides with the area of a commercial SLM (PLUTO-2.1 spatial light modulator) in order to compare the simulated holograms with those obtained by a real device.

The images obtained from the CGH are calculated by Fourier transform propagation of a plane wave perturbed by the CGH, from the hologram plane to the plane where the image is formed.

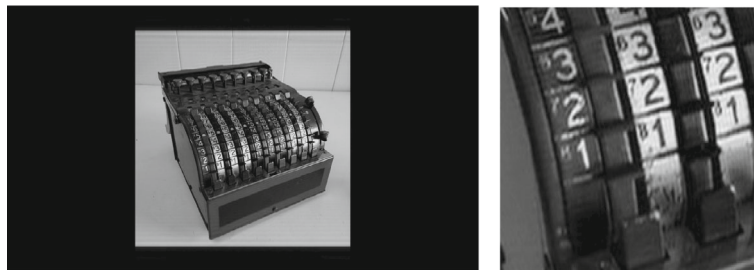


FIGURE 3 (Left) Initial image used for hologram synthesis, 7 mm side length, 1024×1024 mesh points. Image framed in the 15.36×8.64 mm field. (Right) Image detail.

TABLE 3 Graphics series of quality measurements.

Type	Simulation images		Lab images	
Ref.	PCMS	Object	PMCS	Object
RMSE	Figure 7	Figure 9	Figure 13	-
CC	Figure 8	Figure 10	Figure 14	Figure 15

Abbreviations: Object, original image; PMCS, image obtained from CGH calculated with 100% of rays; Ref., image used as reference.

Figure 4 shows both, the rays involved for the synthesis of each hologram (x axis) and the time required to calculate the CGH of the calculator image for all resolutions as a function of the number of rays used. The calculus have been made using a PC computer (Intel Xeon X5675 @ 3.07GHz, 20 GB RAM and GPU 2× GeForce GTX 660h). The range of rays used goes from 106×10^8 rays for the hologram of a square with 32 points of resolution per side and 5% of rays used to over 217×10^{12} s for an object with 1024 points of resolution per side and the total number of rays emitted for each one (see Table 1).

These time values can be significantly improved by using more modern hardware. According to official CUDA benchmark test, it can be divided by a factor of 100 using GeForce RTX 3090 Ti. This would imply that holograms with the resolution and percentage of beams used in PMCS could be synthesized in real time below the blue line shown in Figure 4. Figure 5 shows the simulation results from CGH with 100% rays used and all resolutions considered and Figure 6 shows details of them.

Following subsections show the results of the quality measurements for different series. Table 3 organizes the graphs according to the type of CGH obtained (simulated or laboratory) and the reference image used.

4.1 | 100% PMCS CGH as reference

RMSE results are shown in Figure 7. In these plots the reference image is the one obtained with a 100% ray-synthesized hologram. As an example, some images of series, with 128, 512, and 1024 points per side resolution are shown on Annex

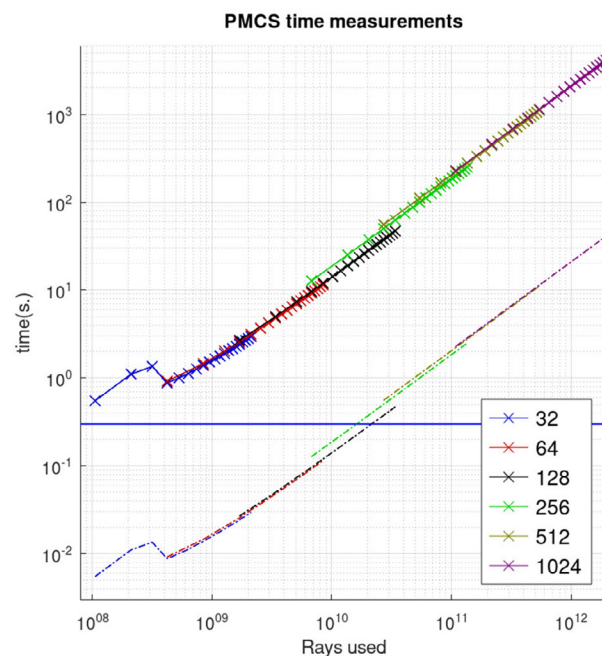


FIGURE 4 Time measurements: Rays send from the object to virtual SLM for all series (x axis) and CGH time cost versus rays used. Blue line shows the estimated limits for moving images (30 fps). Dashed lines are time cost estimations using a GPU GeForce RTX 3090 Ti.

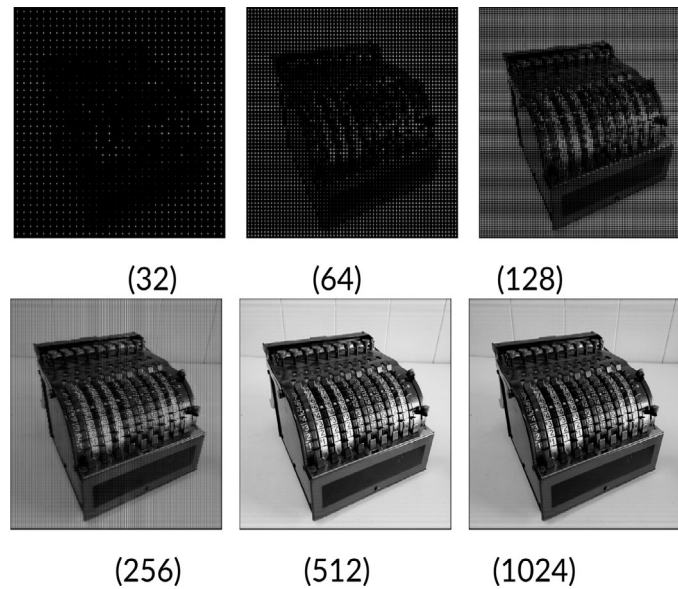


FIGURE 5 Reconstruction images obtained from 100% rays PMCS CGH under several object resolution (points per size).

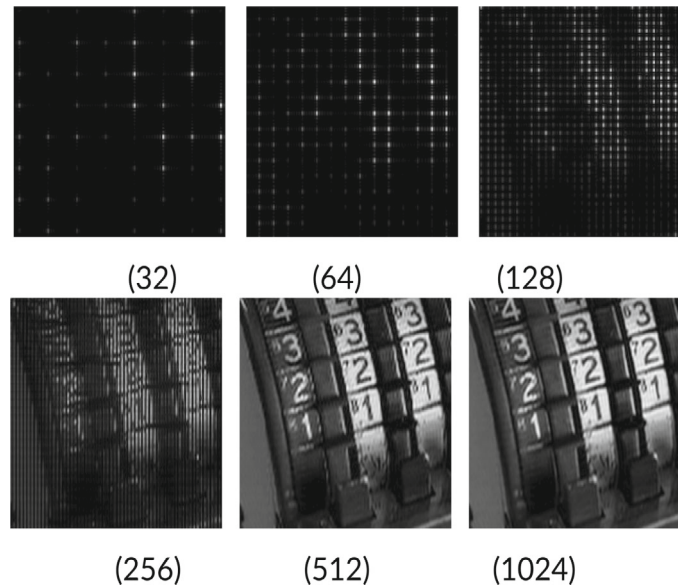


FIGURE 6 Detail of reconstruction images obtained from 100% rays PMCS CGH under several object resolution (N = points per size).

(see Tables A1, A3, and A5). Details are shown too (Tables A2, A4, and A6). For all series, RMSE evolution in function of PMCS rate indicates its improvement when the number of rays used per hologram increases.

Figure 8 shows the correlation coefficient for all series and images. Again, the reference image in each case is the image reconstructed from the hologram calculated with 100% rays. It should be noted the largely smooth behavior of the graph for all series, such that significantly decreasing the percentage of rays used has little effect on the image quality, as can be seen in the tables referred to.

4.2 | Original image as reference

Another option is to compare the images of each series with the original image. The results are shown in Figure 9 (RMSE) and Figure 10 (CC) and show analogies and differences with respect to the previous graphs: in each series, both indicators improve as the percentage of rays used increases.

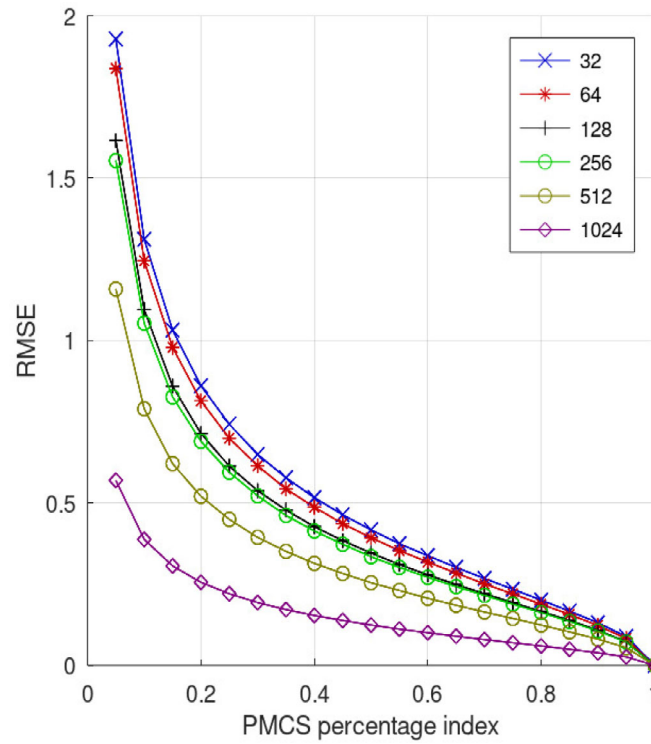


FIGURE 7 Simulation images: RMSE as a function of resolution (color) and percentage of rays used. The reference image is obtained from the hologram calculated with 100% PMCS.

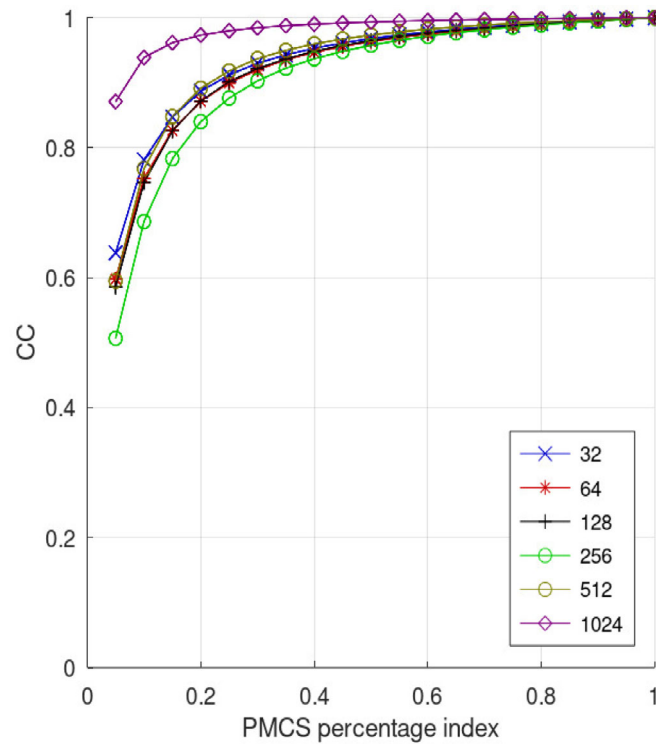


FIGURE 8 Simulation images: CC as a function of resolution (color) and percentage of rays used. The reference image is obtained from the hologram calculated with 100% PMCS.

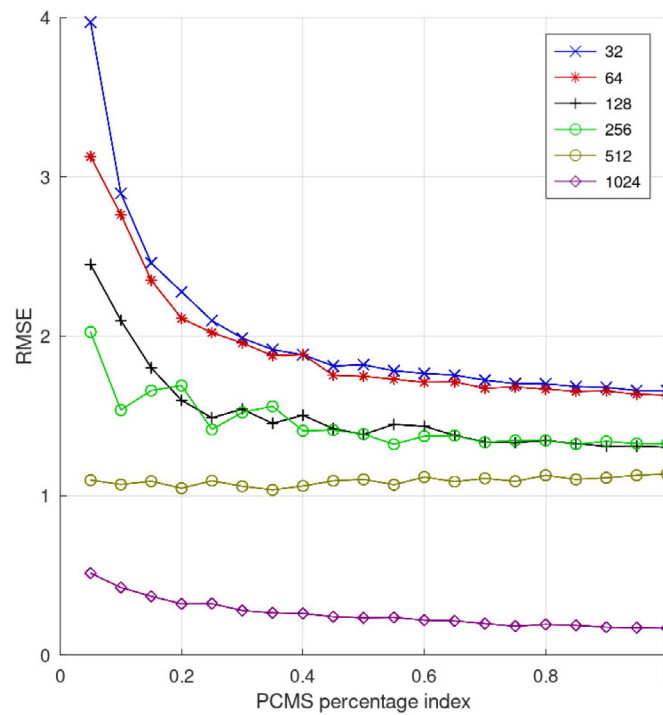


FIGURE 9 Simulation images: RMSE as a function of resolution (color) and percentage of rays used. The reference image is the original sampled at each resolution.

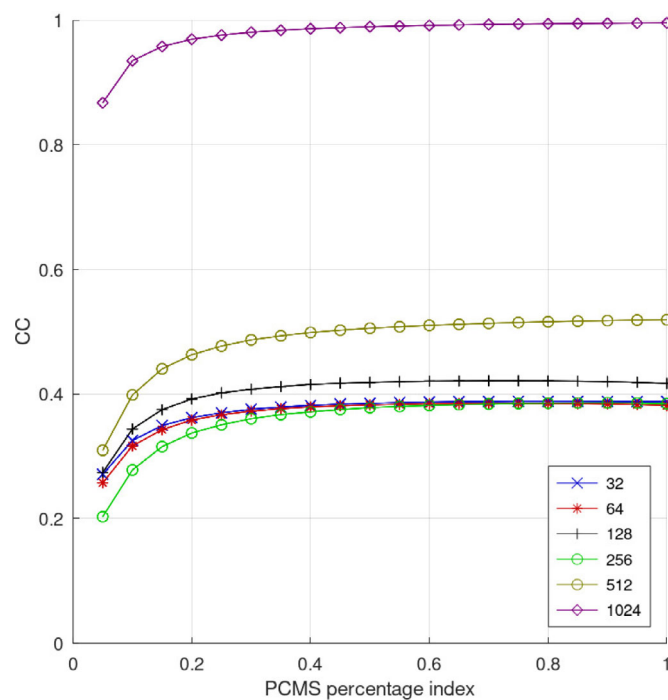


FIGURE 10 Simulation images: CC as a function of resolution (color) and percentage of rays used. The reference image is the original sampled at each resolution.

This improvement is barely noticeable from a relatively low percentage of rays used, around 30% according to the series (see also Tables A1, A3, and A5). It is again observed that the performance improves, in general terms, as the resolution used increases.

5 | PMCS PHASE CGH: EXPERIMENTAL RESULTS

To test the CGHs generated by PMCS in the laboratory, the previous analysis has been repeated with the images obtained from the previously calculated phase holograms, displaying them in an SLM with the same geometrical characteristics (PLUTO-2.1 spatial light modulator). The hologram has been calculated so that the object is in a plane where it can be captured directly on a CCD, without the need for additional optics (Figure 11).

Both simulation process and laboratory set-up for CGH register and images reconstruction have the same geometrical parameters (distance to the virtual-real SLM, pixel size, no lens, wavelength, etc.). This allows comparisons between the two measurements. The essential difference between the simulated and real holograms lies in the fact that the simulation allows modulation in amplitude and phase, whereas the SLMs available in the laboratory only allow modulation in phase.

To obtain the reconstructed image in the simulated holograms, we use the free-space propagation of a plane wave perturbed by the CGH (using the well-known Fourier transform formulation) and evaluate it at the distance that coincides with the distance used to record the CGH reconstruction in the laboratory (300 mm).

It should be noted that this SLM only modulates the phase of the incident wave, so no direct relationship can be established between the results obtained and the simulated results. In addition, the whole experimental set-up (actual SLM modulation, CCD non-linear response, etc.) involves the inclusion of an added transfer function which also affects the recorded image.

Figure 12 shows the holograms obtained with 100% beams for all resolutions analysed in this work. It shows not only the field of the image but also the field of the whole SLM. The details of the images are also shown in the center.

For images with low resolutions, diffraction orders appear. These are associated with the spatial frequencies generated by the periodic distribution of the points and worsen the image quality. Edge effects are also clearly observed in higher resolution images. In all cases the SLM size frame appears and is also shown in the images.

Following the criteria used for the simulated images, we show the series obtained for 128, 512, and 1024 points of resolution per image side and their central details (Tables A7, A9, A11 for complete image and Tables A8, A10, A12).

In the case of the series with low resolution we find that the diffraction orders added by the periodic distribution of the points decrease as the percentage of rays decreases when using PMCS, disappearing for very low percentages. On the

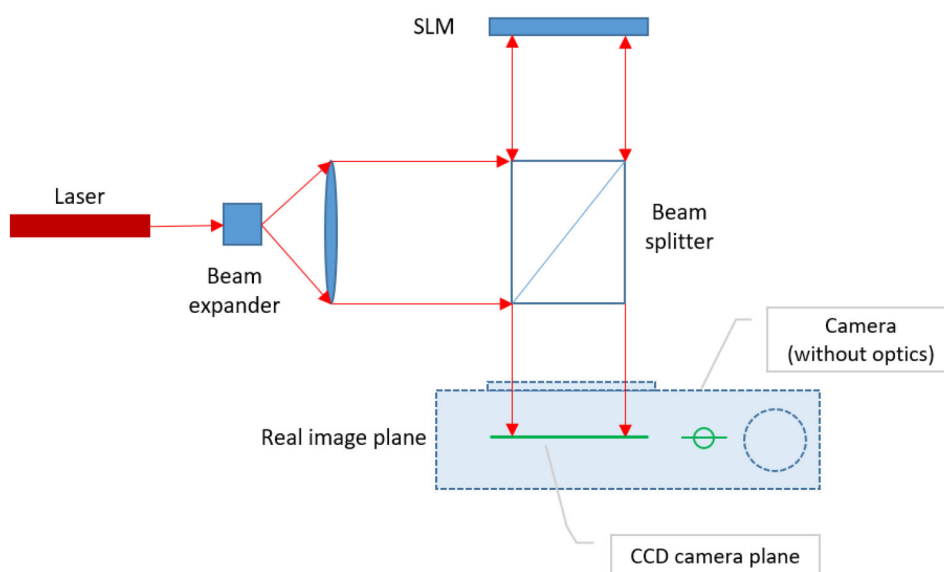


FIGURE 11 Experimental setup schema. SLM plane is located at 300 mm of object plane (and CCD camera plane).

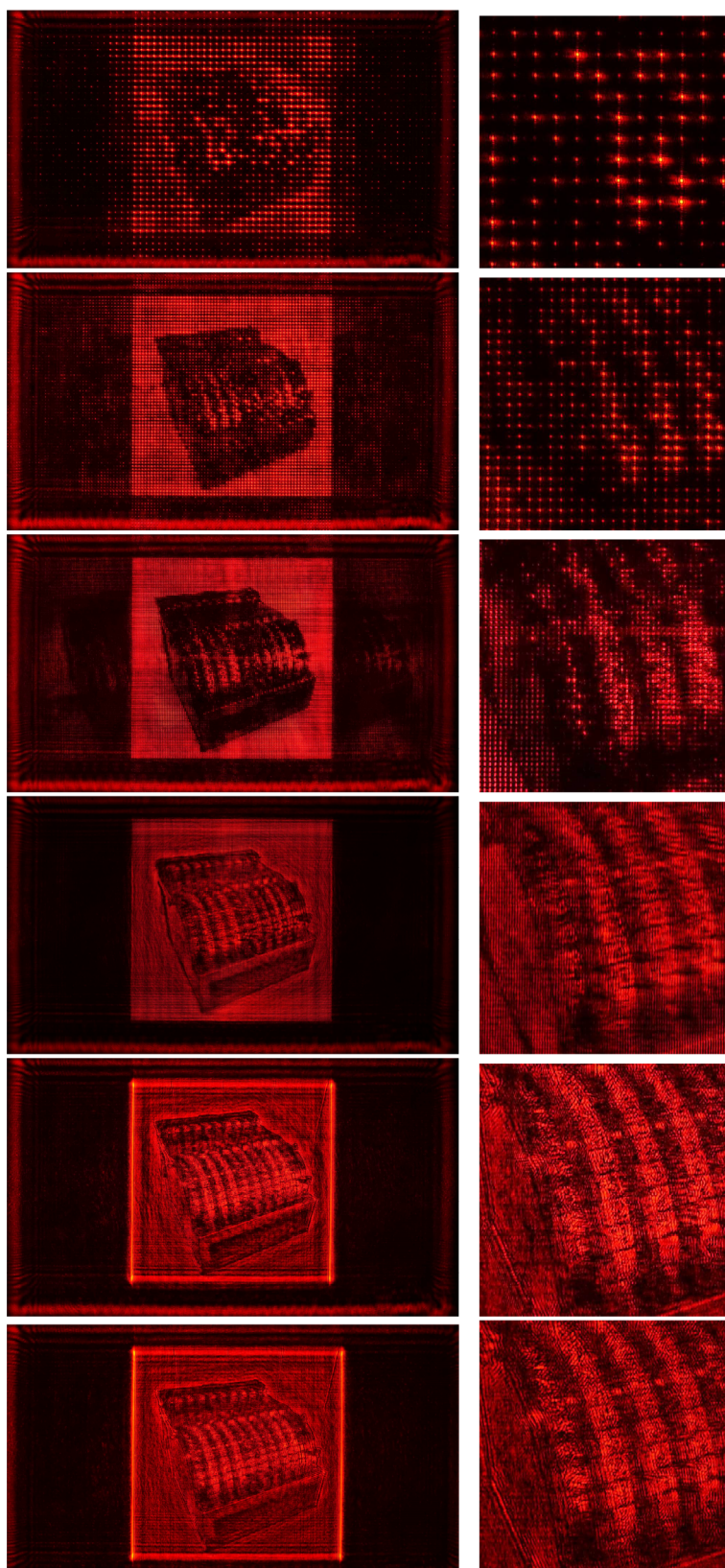


FIGURE 12 Experimental images obtained with 100% rays for all resolutions. Image used for hologram synthesis. 7 mm side. (Left) Image framed in the 15.36×8.64 mm field. (Right) Detail in the center of the image.

other hand, noise increases. For high resolution images (1024 dots/side) edge diffraction effects also decrease significantly as the percentage of rays decreases. Image noise also increases.

Figures 13 and 14 show the RMSE and CC for all calculated series. As in the case of the simulated measurements, the image calculated with 100% of the rays has been taken as the reference image.

Figure 15 shows the CC for the series using as reference the object image (Figure 3). Figures 16 and 17 show detail images of the three highest resolution series (1024, 512, and 256 dots per side) at various ray ratios.

6 | DISCUSSION RESULTS

Computer-generated graphics allow us to put ideas into our minds. In fact, an image is something in between an idea and an object, real or virtual. In this article, we focus on generating by simulation physically accurate holograms of scenes in which the phenomenologically correct interaction of light with matter is simulated. For this purpose, the holography phenomenon is modeled and the photorealistic rendering of the scene is performed.

But what makes a well-made image? What kind of images will allow useful information to be transferred accurately and efficiently from the computer to our minds?

Given the wide range of possible approaches offered by the world of computer graphics and the visualization devices available to us, the answer is far from be simple. In the context of this article, we are looking for generating by simulation holograms of rendered images whose final result is physically correct. However, in this objective binomial, we want to involve another aspect normally less considered, the one that involves the observer's brain, which is the one that ultimately perceives the scene. The reason for this approach is that the authors consider that both physical precision and subjective perception play a definitive role in what could be called a realistic image.

Series of CGHs of a flat object, parallel to the CGH plane, have been calculated using as variables the image resolution (points/side) and the percentage of rays used for the calculation of amplitude and phase. The reconstructed images resulting from illuminating these CGHs have been simulated and the images resulting from projecting the calculated phase onto a commercial SLM have been obtained in the laboratory.

The simulations and laboratory images obtained show that a reasonable quality can be obtained with fewer rays. In fact, some of the effects linked to diffraction effects (e.g., edge effects) are improved, avoiding the computational overhead required by other proposals already mentioned.

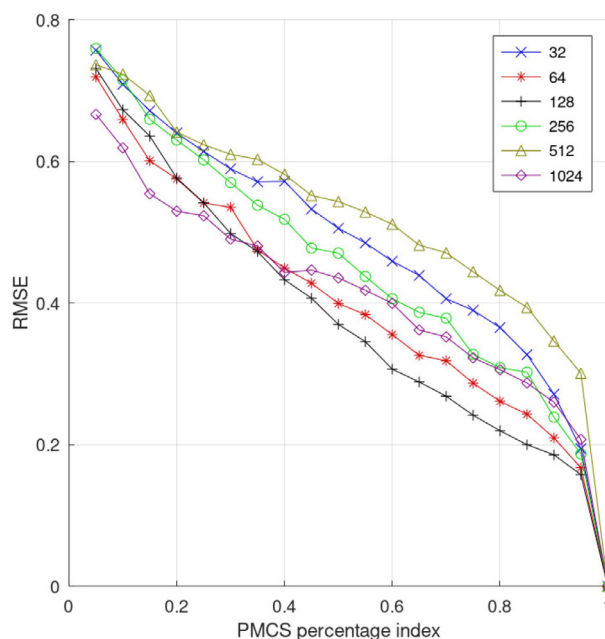


FIGURE 13 Lab. images: RMSE as a function of resolution (color) and percentage of rays used. The reference image is the original sampled at each resolution.

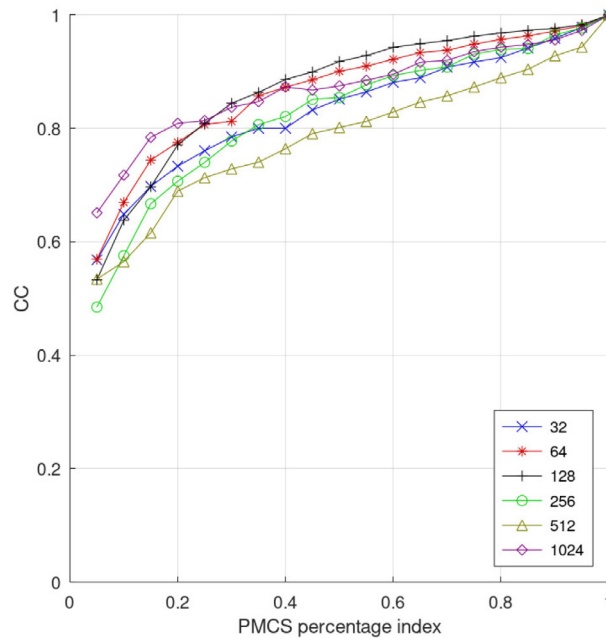


FIGURE 14 Lab. images: CC as a function of resolution (color) and percentage of rays used. The reference image is the original sampled at each resolution.

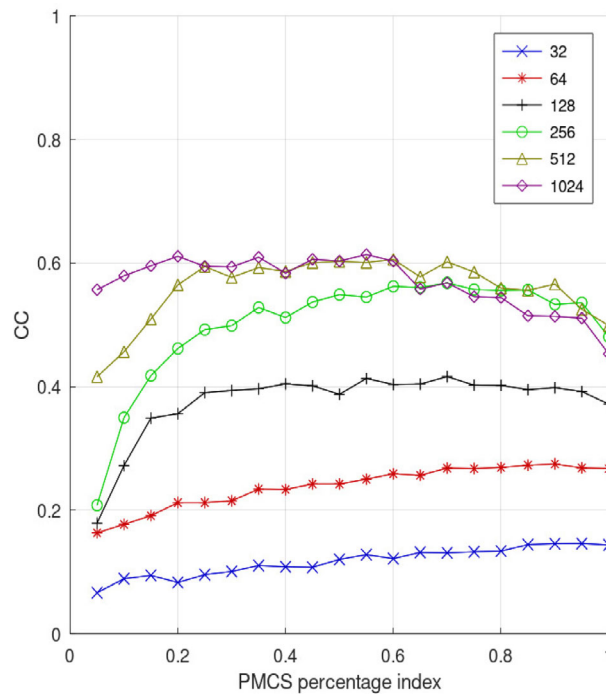


FIGURE 15 Lab. images: CC using as reference image the original image. Example images are shown on Figures 17 and 16.

Quality measurements (RMSE and CC) have been performed using as reference two images for each series: the image obtained from the CGH calculated from 100% of the rays emitted by the sampled object and the image of the real object. These measurements have been performed both for the simulated images and for the images obtained in the laboratory.

Reducing the percentage of rays does not significantly worsen the image quality when using amplitude and phase modulations (simulated images). For images obtained in the laboratory (phase-only CGH) the measurements indicate an improvement when comparing these images with respect to the original.

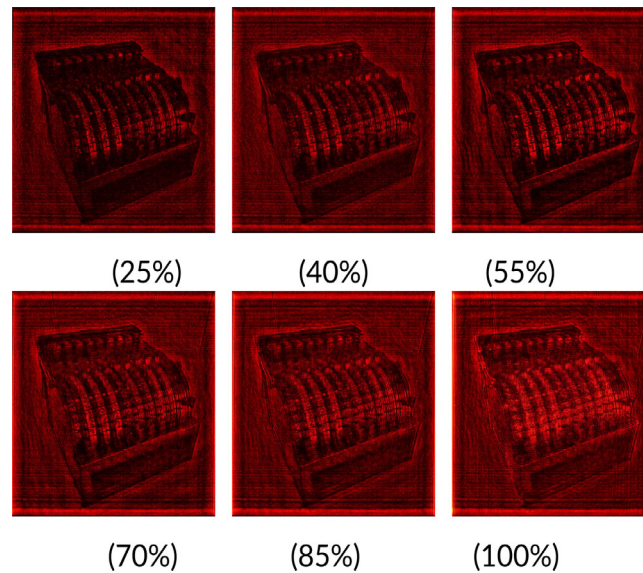


FIGURE 16 Examples of lab. calculator image of 1024 points-size image versus percentage of PMCS used.

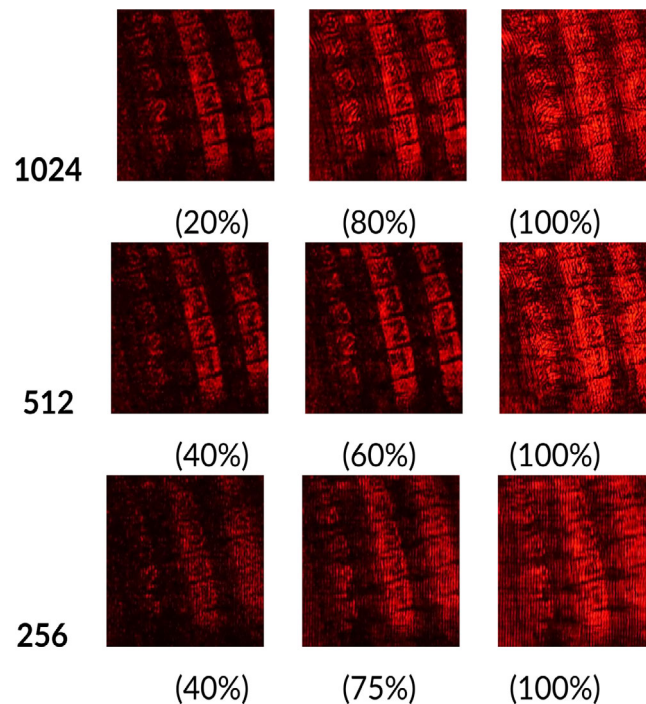


FIGURE 17 Examples of detail images related to the graphic in Figure 15. Rows: Image resolution (pixels/size). Columns: Percentage of PMCS used.

Tables show that noise dominates the image detail for low percentages of rays used in the PMCS algorithm. On the other hand, the diffraction effects of each point are evident for high values of ray percentage.

Previous work has shown that there is a threshold beyond which the observer does not perceive a significant improvement in the quality of the image he or she sees.^{12,13} The results of the previous section also point in this direction. In the case of the simulated images, the correlation coefficients obtained vary very little from a threshold of the percentage of rays (Figures 8 and 10). Figure 8 uses as a reference image the one obtained with the hologram using 100% rays. Data are

asymptotic to the CC value 1. When compared to the original image (Figure 10) the variation remains very small for most of the graph but the image resolution effects become evident as we decrease the number of dots per side of the image. The same is observed for RMSE (Figures 7 and 9)

Comparing experimental PMCS images with 100% PMCS reference images, RMSE and CC worsen for several reasons: SLM uses only phase modulation information, CCD camera sensor, optical setup ... also affects the total transfer function of the system. Nevertheless, they maintain a similar behavior to those obtained in the simulation. Figures 13 and 14 show these values.

The fundamental difference with simulated CGH is that we only get closer to the ideal image when increasing the resolution above the pixel size of the modeled SLM (see Figure 15). For lower resolutions, point diffraction effects become more and more evident the lower the resolution. Even in the case of holograms calculated with 100% rays there is a clear mismatch with original image.

But is this reference image the best possible? Graph on Figure 16 and tables on Annex show that the details of the calculator are best seen with a lower percentage of rays than 100%. It can be seen how the effects of diffraction on the one hand and noise on the other hand reduce the value of this parameter for high and low percentages respectively. This analysis is subject of future work.

7 | CONCLUSIONS

As in the case of computer graphics, the digital approach always generates a sampling of the world. But in addition, in order to calculate a CGH, phase-related calculations have to be added to the CG calculations, which forces the computational cost to increase. Even in this case, they are not perfect reconstructions of the initial objects: diffractive optics has inherent phenomena (such as edge effects, speckle, ghost images due to higher orders ...). This cost would increase even more if techniques are used to improve the quality of final images.

This work looks in the opposite direction: we describe an algorithm based on Monte Carlo techniques for CGH calculation, which we call PMCS, and series of CGHs have been calculated by varying the sampling resolution for the model mesh (pixels/side) and the percentage of rays used (see Table 1).

To analyse the quality of these series (Section 4), the RMSE and CC have been calculated taking as reference the CGH that uses all possible rays and the original image. These measurements have been performed for simulated holograms (including amplitude and phase modulation) and holograms in the laboratory (phase only).

Both RMSE and CC indicate that it is possible to decrease the percentage of rays used without significantly decreasing the image quality. This information is consistent with both simulated and laboratory holograms. It is evident that quality images increases with number rays used and image resolution.

In the case of laboratory holograms, it is observed that the appropriate reference image is the original, and the best CC values are obtained for intermediate ranges of ray ratios (see Figure 16). It is also in agreement with the observed images (Figure 17).

We establish a direct relationship between desired image quality and calculation time in function of the specific needs of each case. For example, in order to obtain real time in the generation of an image (below 30 FPS) and seeing the results obtained and estimated, it would be necessary in the best case of image quality a factor 2 of computing power with respect to a GeForce RTX 3090 Ti GPU (Figure 4).

Future work: For production applications based on Monte Carlo algorithm efficiency is of primary concern: image noise (variance) must be low at practical computations times. Our next goals will be sampling techniques that significantly improve rendering efficiency for image-plane sampling, maintaining the idea of reduced sampling, but adapting sampling distributions to the characteristics of the image pixels, and including 3D scenes.

AUTHOR CONTRIBUTIONS

Alfonso Blesa: conceptualization (equal); data curation (equal); formal analysis (equal); investigation (equal); methodology (equal); validation (equal); writing – original draft (equal); writing – review and editing (equal). **Juan Antonio Magallón:** conceptualization (equal); data curation (equal); investigation (equal); methodology (equal); software (equal); visualization (equal); writing – review and editing (equal). **Francisco José Serón:** conceptualization (equal); formal analysis (equal); investigation (equal); methodology (equal); supervision (equal); validation (equal); writing – review & editing (equal).

ACKNOWLEDGMENTS

Gobierno de Aragón, Research groups support program: Interactive Systems, Adaptivity, Autonomy and Cognition Group (cod. T33-20D).

CONFLICT OF INTEREST STATEMENT

Authors have no conflict of interest relevant to this article.

DATA AVAILABILITY STATEMENT

Data available on request from the authors.

ORCID

Alfonso Blesa  <https://orcid.org/0000-0001-9688-2260>

Juan Antonio Magallón  <https://orcid.org/0000-0002-3355-0055>

Francisco José Serón  <https://orcid.org/0000-0003-1683-4694>

REFERENCES

1. Son JY, Lee H, Lee BR, Byeon J, Park MC. Holographic and light field displays: what are the differences? Proceedings of the 2017 16th Workshop on Information Optics (WIO); 2017:1–2.
2. Yamaguchi M. Light-field and holographic three-dimensional displays. *J Opt Soc Am A*. 2016;33(12):2348–2364. <http://opg.optica.org/josaa/abstract.cfm?URI=josaa-33-12-2348>
3. Yaraş F, Kang H, Onural L. State of the art in holographic displays: a survey. *J Disp Technol*. 2010;6(10):443–454.
4. Blinder D, Ahar A, Bettens S, et al. Signal processing challenges for digital holographic video display systems. *Signal Process Image Commun*. 2018;2019(70):114–130. doi:10.1016/j.image.2018.09.014
5. Sahin E, Stoykova E, Mäkinen J, Gotchev A. Computer-generated holograms for 3D imaging: a survey. *ACM Comput Surv*. 2020;53(2):32.
6. Shi L, Li B, Kim C, Kellnhofer P, Matusik W. Towards real-time photorealistic 3D holography with deep neural networks. *Nature*. 2021;591(7849):234–239.
7. Peng Y, Choi S, Padmanaban N, Kim J, Wetzstein G. Neural holography. Proceedings of the ACM SIGGRAPH 2020 Emerging Technologies; 2020:2–3.
8. Chen L, Zhang H, Cao L, Jin G. Non-iterative phase hologram generation with optimized phase modulation. *Opt Express*. 2020;28(8):11380–11392.
9. Blinder D, Chlipala M, Kozacki T, Schelkens P. Photorealistic computer generated holography with global illumination and path tracing. *Opt Lett*. 2021;46(9):2188–2191.
10. Pharr M, Humphreys G. *Physically Based Rendering: From Theory to Implementation*. Elsevier Inc.; 2010.
11. Magallón J, Blesa A, Serón F. SLM simulation and MonteCarlo path tracing for computer-generated holograms. *SN Comput Sci*. 2021;2(3):1–16. doi:10.1007/s42979-021-00632-6
12. Navarro F, Castillo S, Serón FJ, Gutierrez D. Perceptual considerations for motion blur rendering. *ACM Trans Appl Percept*. 2011;8(3):20.
13. Reddy M. Perceptually optimized 3D graphics. *IEEE Comput Graph Appl*. 2001;21(5):68–75.

How to cite this article: Blesa A, Magallón JA, Serón FJ. Partial Monte Carlo sampling for computer generated holograms. *Engineering Reports*. 2023;e12673. doi: 10.1002/eng2.12673

ANNEX

Tables of simulated and lab images PMCS series, both complete and detail, for 1024², 512², and 128² mesh grid.

TABLE A1 Simulation calculator reconstruction for 1024 points/size CGH resolution versus PMCS percentage of rays used.

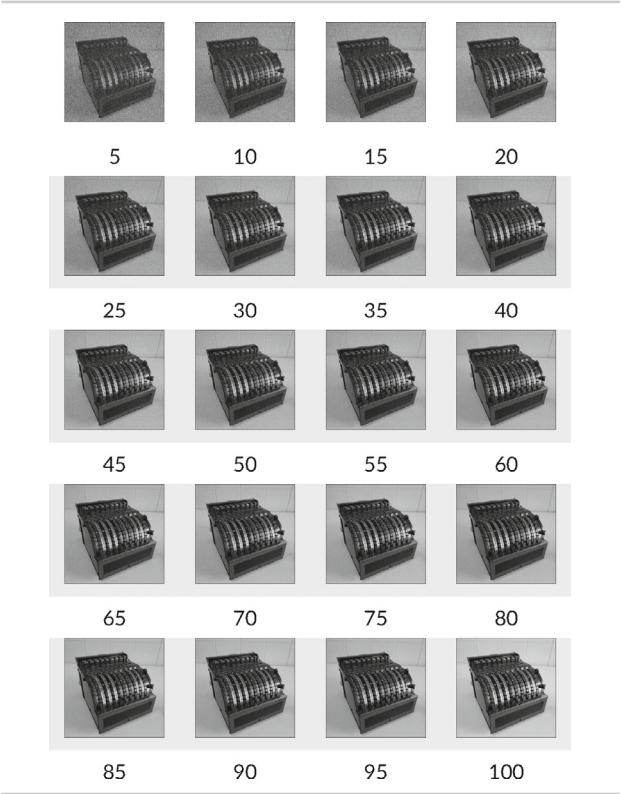
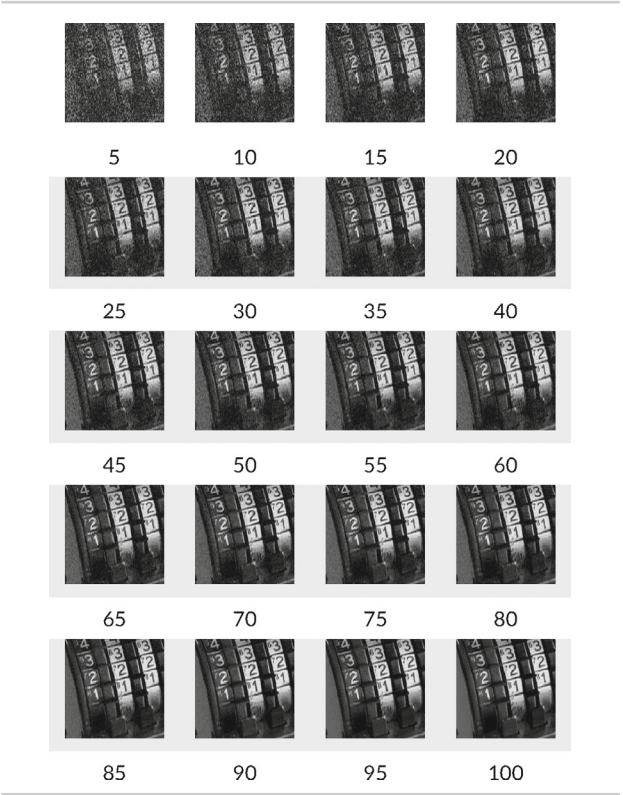


TABLE A2 Detail of calculator reconstruction for 1024 points/size CGH resolution.



Note: Image reconstruction versus PMCS percentage of rays used.

TABLE A3 Simulation calculator reconstruction for 512 points/size CGH resolution versus PMCS percentage of rays used.

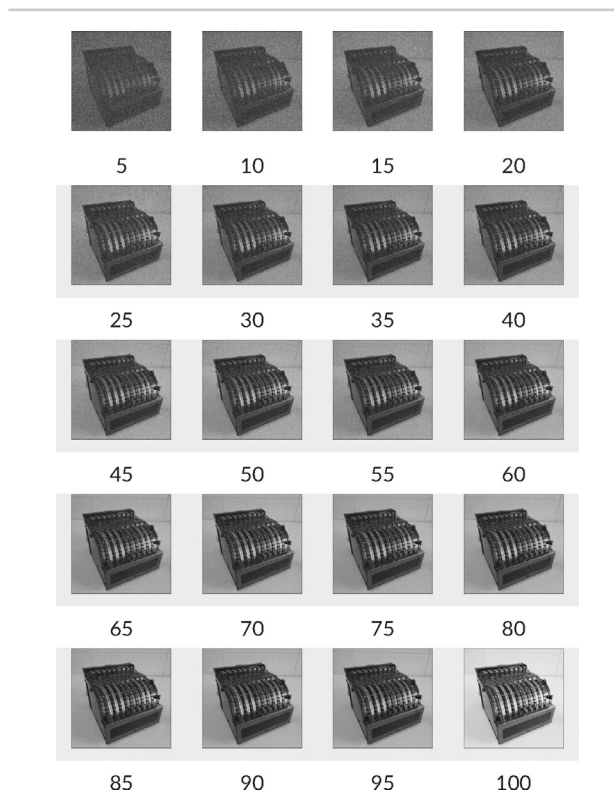
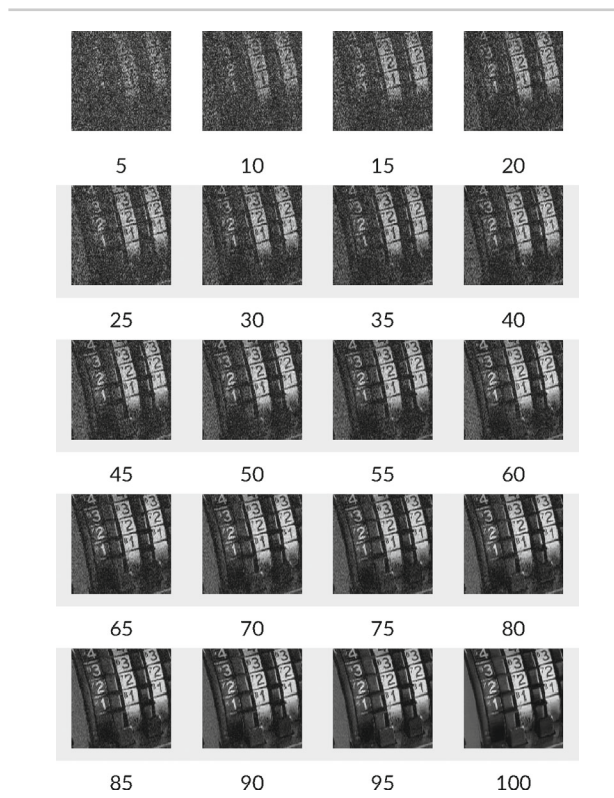


TABLE A4 Detail of calculator reconstruction for 512 points/size CGH resolution.



Note: Image reconstruction versus PMCS percentage of rays used.

TABLE A5 Simulation calculator reconstruction for 128 points/size CGH resolution versus PMCS percentage of rays used.

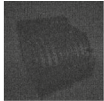
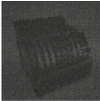
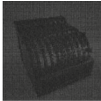
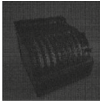
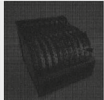
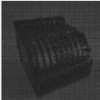

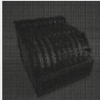
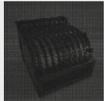
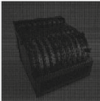
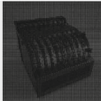
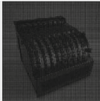
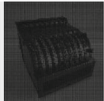
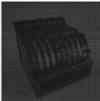
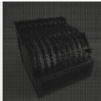
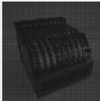
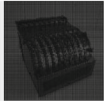
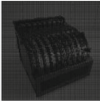
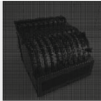
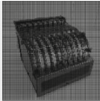




















			
5	10	15	20
			
25	30	35	40
			
45	50	55	60
			
65	70	75	80
			
85	90	95	100

TABLE A6 Detail of calculator reconstruction for 128 points/size CGH resolution.

			
5	10	15	20
			
25	30	35	40
			
45	50	55	60
			
65	70	75	80
			
85	90	95	100

Note: Image reconstruction versus PMCS percentage of rays used.

TABLE A7 Lab. calculator reconstruction for 1024 points/size CGH resolution versus PMCS percentage of rays used.

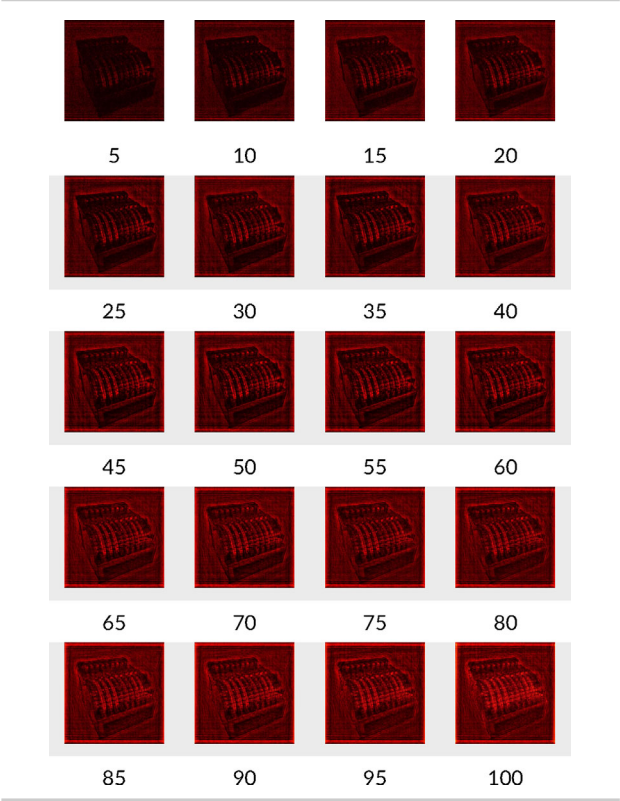


TABLE A8 Detail of calculator reconstruction for 1024 points/size CGH resolution versus PMCS percentage of rays used.

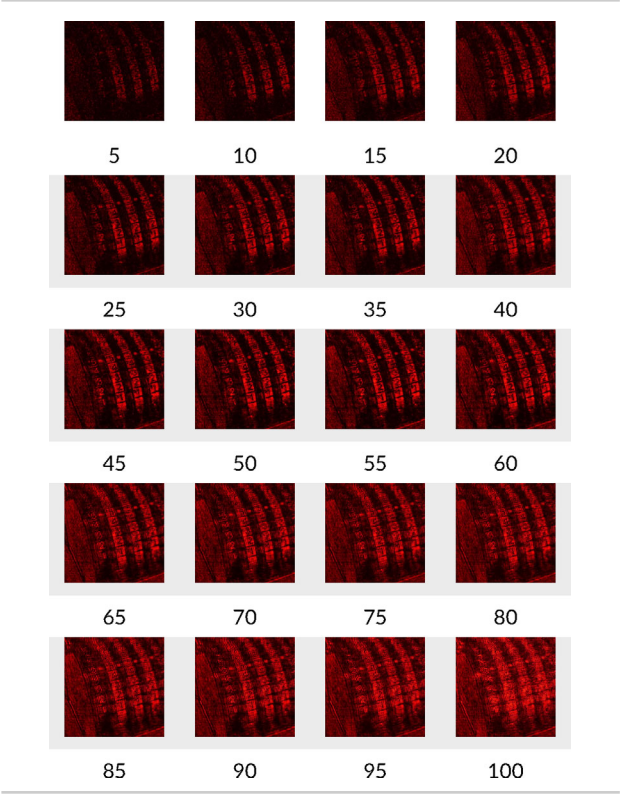


TABLE A9 Lab. calculator reconstruction for 512 points/size CGH resolution versus PMCS percentage of rays used.

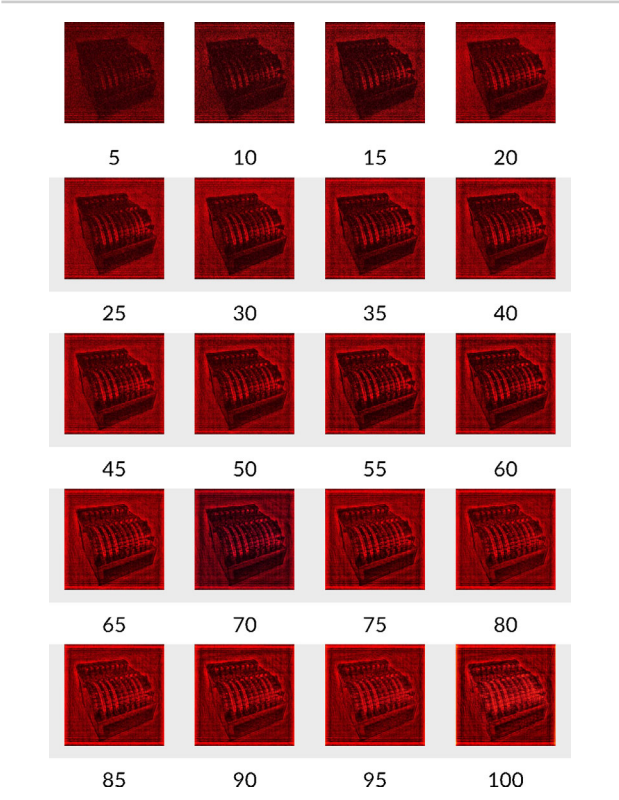


TABLE A10 Detail of calculator reconstruction for 512 points/size CGH resolution versus PMCS percentage of rays used.

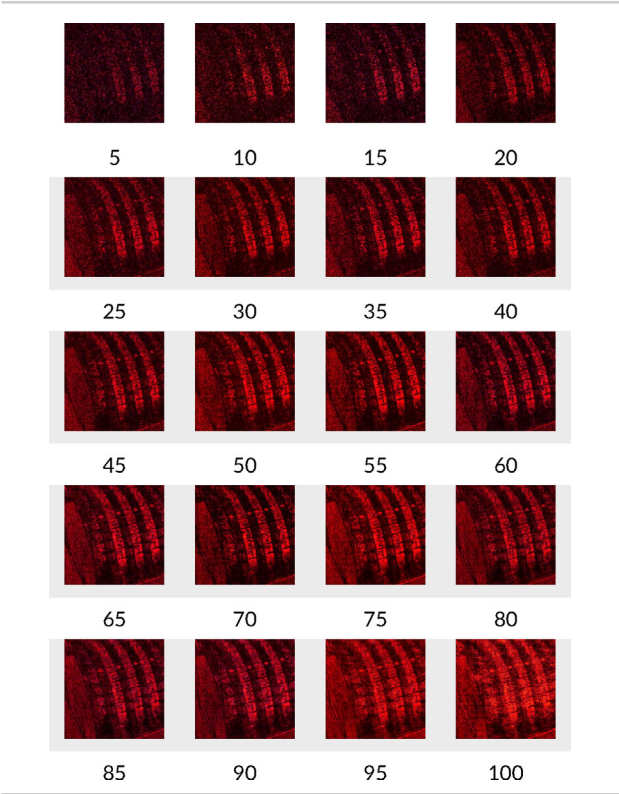


TABLE A11 Lab. calculator reconstruction for 128 points/size CGH resolution versus PMCS percentage of rays used.

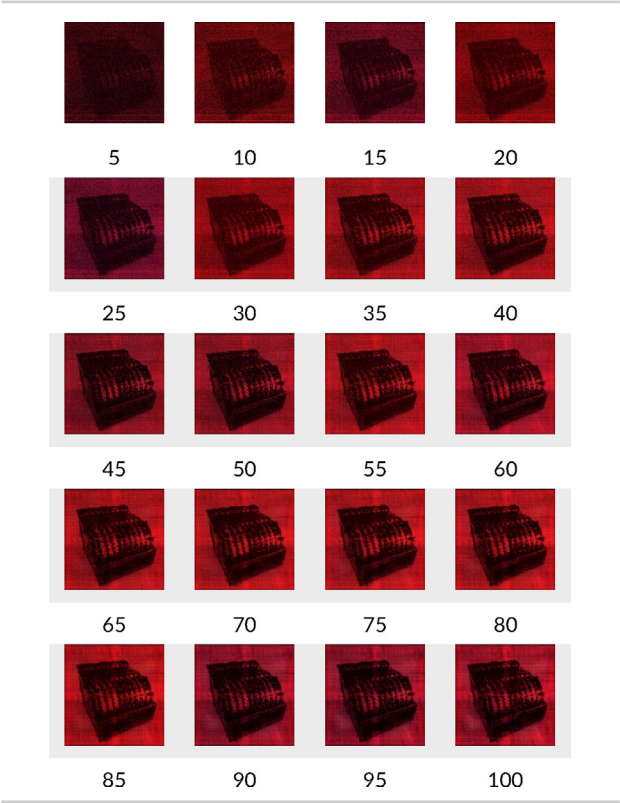
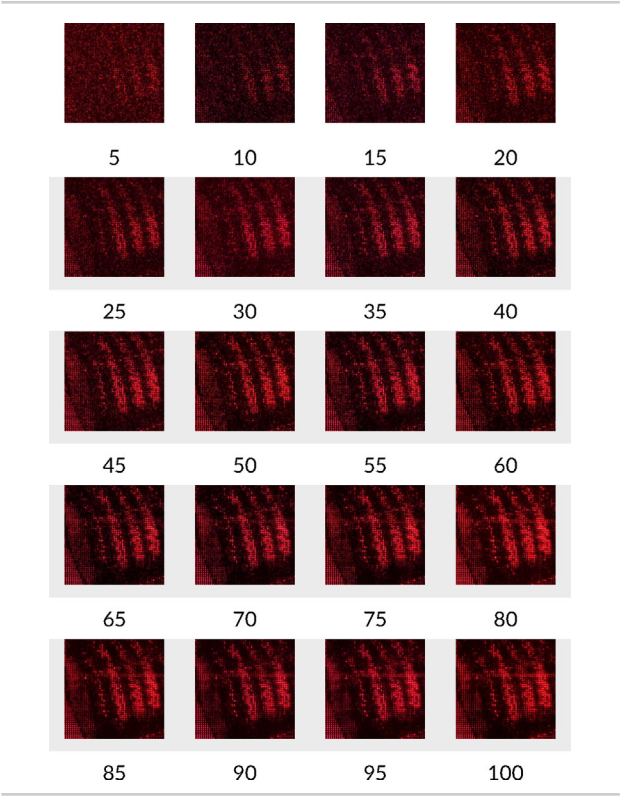


TABLE A12 Detail of calculator reconstruction for 128 points/size CGH resolution versus PMCS percentage of rays used.



AUTHOR BIOGRAPHIES



Dr. Alfonso Blesa holds his degree in Physics (1989) and a PhD in Physics (1996) from University of Zaragoza. He is currently lecturer of Electronic Technology at the University of Zaragoza. He teaches programmable electronic systems at different levels. He has been involved in research and innovation work related to diffractive optics, holography (both analogue and digital) and microprocessor-based systems, focusing on embedded systems for Industry 4.0.



Dr. Juan Antonio Magallón received his Industrial Engineering degree from the School of Engineering and Architecture (EINA), at University of Zaragoza in 1991, with specialization in Computer Science, Electronics and Control. He targeted his research on Computer Graphics and Simulation of Global Illumination and got his PhD on this subject on 2003 from UZ. Since then he focused on precise physical simulation of light transport in subjects like global illumination and real-time computer graphics, in parallel with visualization techniques for high quality imagery, and HPC hardware and software related to these fields, like parallel computing on both CPU and GPU.



Dr. Francisco José Serón has been formed in the Zaragoza University, received his Bachelor's Degree on Sciences in 1979 and completed his Doctorate in Sciences in 1984. He then joined the Applied Mathematics Department as Associated Professor and stayed there until transferring to the Computer Science Department in 1990, where he works currently as Full Professor. During this time, Dr. Francisco José Serón has done Research and Innovation work on fields such as Simulation of Natural Phenomena, Visualization and Computer Graphics techniques, Numerical Computation and Parallel Computing and Artificial Intelligence. He has published numerous papers in these areas. His research has been funded by regional, national and European Union government agencies and by private corporations including IBM, GME, CASA, INDAL, WWP ...