



SLM Simulation and MonteCarlo Path Tracing for Computer-Generated Holograms

Juan Magallón¹ · Alfonso Blesa² · Francisco Serón¹

Received: 1 February 2021 / Accepted: 2 April 2021 / Published online: 23 April 2021
© The Author(s) 2021

Abstract

Computer holography is a growing research field that must pay attention to two main issues concerning computing effort: the visualization of a 3D virtual scene with photo-realistic quality and the bottleneck related to hologram digitization and visualization limits. This work shows a computational approach based on a Monte Carlo path-tracing algorithm, which accounts for both geometrical and physical phenomena involved in hologram generation, and, therefore, makes a feasible estimation of computing time costs. As these holograms also require yet unavailable visualization devices, their behavior needs to be simulated by computer techniques.

Keywords Computer-generated hologram · 3D · SLM

Introduction

Photo-realistic rendering is one of the primary goals in Computer Graphics. Much progress has been made in graphics over the years in a bid to attain this goal, with significant advancement in 3D model acquisition and representation, such as measurement and representation of object surface properties (BRDF, sub-surface scattering), or modeling of natural objects and phenomena, such as plants, water, fog, smoke, snow, fire or rainbows. The cause is also aided by more sophisticated graphics hardware offering faster rendering, accurate shading via programmable vertex and pixel engines, larger caches and memory footprints and floating-point pixel formats. In other words, a variety of well-established approaches and systems are available for rendering models. See the surveys on physically based rendering [20], global illumination methods [3], and photon mapping [9].

Despite all the advances in traditional computer graphics areas, competing with the complexity of images of real scenes is still challenging. This difficulty in achieving photorealism with conventional 3D and model-based graphics has led researchers to take a kind of “shortcut” and work directly with real images. This approach is called Image-Based Modeling and Rendering (IBMR).

IBMR techniques have received considerable attention as a robust alternative to traditional geometry-based techniques for image synthesis. These use images rather than geometry as the main primitives for rendering different views of a scene. There are also techniques mixing both geometry and image-based methods, each to a different level. Previous surveys related to image-based rendering have suggested characterizing techniques based on how image-centric or geometry-centric they are. This has resulted in the image-versus-geometry-based continuum of rendering techniques or IBMR continuum [11, 17].

Figure 1 classifies the various rendering techniques into three categories, namely, rendering with no geometry, rendering with implicit geometry, and rendering with explicit geometry. These categories, rather than as absolute discrete, should actually be viewed as a continuum, given that some techniques defy strict categorization.

In this paper, we have extended the IBMR continuum by introducing computer-generated holography (CGH) [23] as the key technology to integrate synthetic 3D imaging into holography (see Fig. 1), and attain both the best

✉ Alfonso Blesa
ablesa@unizar.es

Juan Magallón
magallon@unizar.es

Francisco Serón
seron@unizar.es

¹ Universidad de Zaragoza, EINA, Campus Río Ebro, 50010 Zaragoza, Spain

² Universidad de Zaragoza, EUPT, 44003 Teruel, Spain

Fig. 1 IBMR continuum: rendering types for synthetic images (from [22]). Solid lines are related with established techniques and dashed lines show the approximation in this work

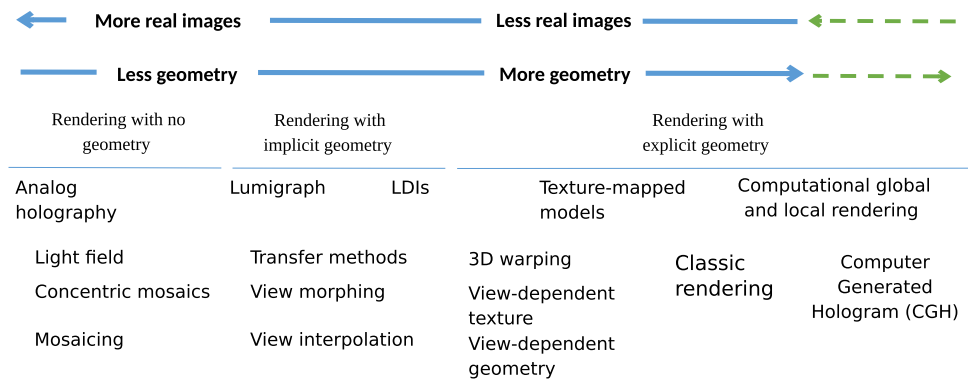


photo-realistic rendering quality and optimum viewing experience. This extension is based on the behavior of the human visual system and novel spatial image display technologies that will become mainstream at some stage.

To attain a photo-realistic effect in an image, geometrical (i.e., distances, angles, occluded and hidden surfaces) and physical (absorption, reflection, refraction or diffraction, polarization, etc.) behaviors are the main concerns. However, synthesized scenes must be presented in a display favoring their spatial perception by the viewer, so they can truly perceive a full three-dimensional sensation when looking at the scenes. Although this task has been approached from different perspectives over the years (single-image display, stereoscopic display, holography), each requiring different scene representations (single or dual radiance maps and interference maps, among others), and using different calculation methods (such as ray tracing, path tracing, point-source wave propagation or Fourier wave propagation), none has fulfilled the requirements of realistic image appearance and suitable image display for real 3D perception. Traditional rendering techniques use complex geometric and appearance models, but rendering results are only plain images or stereo pairs at most. Holography, on the other hand, delivers a powerful light field display, but the traditional sources of holographic images are merely analog captures or extremely simple synthetic models, usually due to unbearable algorithm complexity and cost.

This work presents a Monte Carlo based formalism to include the complex geometrical and physical behavior of a scene in the synthesis of the light wave front generated by an object, which can serve to generate an interference map suitable for display using holographic means. This technique pushes the boundaries of technology in two main aspects: the huge computing power needed to simulate the scene, and holographic display devices, usually referred to as SLMs (spatial light modulators), with enough quality (with respect to bandwidth and resolution) to display these high-quality holograms. Looking forward to what the future can offer in terms of SLM technology advances, we would also like to predict the level of quality that could be reached with these

new display devices, and therefore, a simulator for upcoming SLM display devices is also needed. This simulator can be verified using it to model current SLMs and then comparing results with real devices. This paper also presents an estimation of computing time for rendering and visualization to predict the power needs for dynamic holographic applications.

Consequently, the pros and cons of stereoscopic display versus holographic display, taking the human visual system into account, and the basic holographic principles are addressed in "Fundamentals of 3D Imagery Display" section. "Holographic Monte Carlo Path Tracing (HMCPT)" section contains a description of the holographic Monte Carlo path-tracing method and an estimation of the time involved in these calculations. Sections "Image Reconstruction from Synthetic Digital Holograms" and "Results Comparison" deals comparisons between simulated and real 3D scenes get from SLM. Finally, the conclusions of this research are summarized in "Conclusions and Future Work" section. The calculations for both hologram design and cost estimations are compiled in Appendix A.

Fundamentals of 3D Imagery Display

The two main methods used to display imagery in a way that offers a kind of 3D feel are stereoscopic displays and holographic displays. The principles of both are fundamentally different.

Stereoscopic displays have been on the market for several years. As a result, their image quality, for example resolution and color reproduction, is good, even if they have not yet succeeded in breaking through as a mass-market product. Holographic displays, however, are still at the lab or prototype stage.

Concerning 3D imagery, currently available display devices can all be referred to as spatial light modulators (SLMs). They can modify the intensity (and, consequently, color, when applied on a spectral level) and/or phase of an incident light beam. SLMs are implemented using several

technologies, such as a liquid-crystal display (LCD), a liquid crystal on silicon (LCOS) display or a micro-mirror display (MMD); all are electrically addressable SLMs (or EASLMs). There exist also analog media, known as optically addressable spatial light modulators (OASLMs) or acoustic-optical modulators.

The Human Visual System: Stereoscopic Versus Holographic Display

Stereoscopic displays provide two different images of the same object, one for the observer's left eye and one for the right eye. These images are displayed either simultaneously or sequentially, in other words, by either spatial or temporal multiplexing. Each image is displayed on an SLM, which, in most cases, is an LCD. Each eye directly perceives the image as it is displayed on the SLM. Spatial or temporal beam-splitting techniques make each eye see only its associated image. Both images should be generated depending on eye position, and only offer a kind of realistic 3D feel if viewed from these specific positions. Consequently, if the eye position changes, both images have to be generated again from scratch.

Holographic displays can reproduce the full scalar light field (or light wave front) coming from the scene. Only after the display has generated this light wave front, does the observer choose the point of view and reproduce both eyes' images. Ideally, a holographic object reconstruction perfectly mimics a real object. Rather than the display device itself, observers perceive the 3D object reconstructed in space by the hologram. When viewing a hologram, observers can change their point of view (within specific limits that will be defined later) onto the wave front and see a different view of the object.

In both proposals, the human visual system uses two depth cues characterizing the difference in spatial vision in a stereoscopic display and in a holographic display: convergence of the eyes and accommodation, in other words, focus of the eye lens. Figure 2 demonstrates these depth cues. For normal viewing, an object (the sphere) is seen by both eyes. The eyes converge toward the object with a convergence angle α . The human vision system merges the two images seen by the eyes and deduces depth information. Convergence and accommodation are both depth cues. The eye lens focuses on the object, thereby optimizing the perceived contrast. Both depth cues provide depth information. The normal viewing situation also applies to holographic displays, as they mimic a real existing object by reconstructing the light wave front that would be generated by a real object.

Stereoscopic displays inherently fail to provide conclusive depth information. Autostereograms and stereoscopic displays provide two images with different perspective views, as indicated by the two small circles on the display plane.

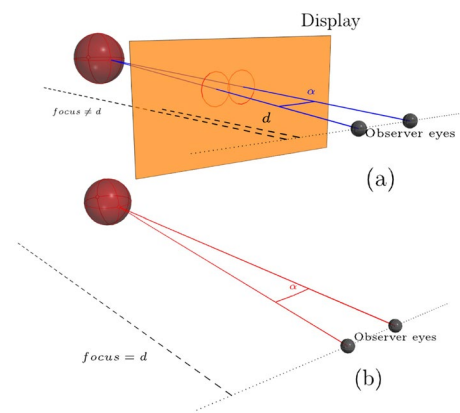


Fig. 2 Stereographic viewing versus natural viewing: **a** autostereogram, **b** natural viewing

The eyes have the correct convergence angle, as if they were looking at the real object. However, the eye lenses focus on the image plane, as both images are displayed on the display plane. Therefore, there is an inherent mismatch between depth information from convergence and from accommodation. Furthermore, stereoscopic displays are only an approximation to natural viewing, as each image seen by the eyes is flat. All depth layers of an object that spread in space appear at the same depth plane on the display. The informational difference between convergence and focus may lead to vision problems such as eyestrain and fatigue if they are too large, and, if the stereoscopic display is watched for a long time, usability of stereoscopic displays decreases.

Conversely, holographic displays mimic natural eye behavior from the point of view of convergence and accommodation, which can predict them as future devices when the actual technological restrictions are overcome.

Stereoscopic displays are designed for (theoretically) just one observer; others would see the scene with some kind of fault, such as distortion, double image, etc. In contrast, the 3D scene shown by a holographic display can be (theoretically again) observed from multiple points of view, limited only by the field of view defined by display device technology.

Basis of Holography

Holography is based on the light diffraction theory. A hologram can be defined as a combination of gratings and, in turn, a grating is defined in Born and Wolf [1] as “any arrangement which imposes on an incident wave a periodic variation of amplitude or phase, or both”; that is, holographic images are based on light diffraction. In the ideal case, light diffracted at the hologram plane with the diffraction pattern encoded on it shows the same wave front as the light that would be generated by a real object.

Holography dates from 1947, from the work by Denis Gabor. It is a well-established discipline and, in essence, consists in saving a diffraction pattern in a suitable recording medium (e.g., a photosensitive plate). When this medium is adequately illuminated, a wave front similar to that generated by an object is recovered. The procedure to achieve this result is based on the registration and subsequent reconstruction of the object wave front by interference with a reference wave.

Figure 3 shows the registration and operation mode of a hologram schematically: In the registration stage, a real object (e.g., the spheres) is illuminated by some kind of light source that is spatially and temporally coherent with a reference wave $\mathcal{R}(\mathbf{r})$ (see Fig. 3a), which, in turn, illuminates registration media at the hologram plane. Therefore, it is possible to know the amplitude and phase generated for all points of the scene ($\mathcal{U}_i(\mathbf{r})$, $\mathcal{U}_j(\mathbf{r})$,...) in all hologram plane points p_i .

In the reconstruction stage, all wave fronts of an object point $\mathcal{U}_i(\mathbf{r})$ are recovered from a reconstruction wave $\mathcal{R}(\mathbf{r})$ similar to that used in the registration stage. Figure 3b shows how the hologram works: using $\mathcal{R}(\mathbf{r})$ as illumination wave, phase and/or intensity variations introduced by a hologram give the object wave.

Math for Hologram Recording

Generating a hologram requires encoding both the amplitude and phase of the original wave front across a plane of observation into intensity variations, which is converted to transparency or phase variation levels at the time of registration. In scalar wave optics, a light wave is defined as scalar amplitude [1]:

$$u(\mathbf{r}) = U(\mathbf{r}) e^{i(\mathbf{k} \cdot \mathbf{r} - \omega \cdot t + \varphi_0)} \quad (1)$$

If we separate the time and space oscillating terms:

$$u(\mathbf{r}) = U(\mathbf{r}) e^{i(\mathbf{k} \cdot \mathbf{r} + \varphi_0)} e^{i(-\omega \cdot t)} \quad (2)$$

We can define a complex amplitude

$$\mathcal{U}(\mathbf{r}) = U(\mathbf{r}) e^{i(\mathbf{k} \cdot \mathbf{r} + \varphi_0)} = U(\mathbf{r}) e^{i\varphi(\mathbf{r})} \quad (3)$$

with module and phase, so

$$u(\mathbf{r}) = \mathcal{U}(\mathbf{r}) e^{i(-\omega \cdot t)} \quad (4)$$

As the time part will be consistent across all the scene, we can just disregard it, and use only the complex amplitude (or phasor) for the hologram calculations.

The light coming from a scene can be modeled as a set of spherical waves $\mathcal{U}_i(\mathbf{r})$ originating at each point of the object. Then, the complex amplitude of this spherical wave on a point \mathbf{p} the hologram plane is:

$$\mathcal{U}_i(\mathbf{p}) = U_i(\mathbf{p}) e^{i\varphi_i(\mathbf{p})} \quad (5)$$

where phase $\varphi_i(\mathbf{p})$ on the SLM plane for a wave $\mathcal{U}_i(\mathbf{r})$ travelling from object point \mathbf{o}_i can be defined as

$$\varphi_i(\mathbf{p}) = \frac{2\pi}{\lambda} \|\mathbf{p} - \mathbf{o}_i\| + \varphi_{i,0} \quad (6)$$

where offset $0 \leq \varphi_{i,0} \leq 2\pi$ is an initial condition for each point of the scene.

The *object wave* $\mathcal{U}(\mathbf{p})$ is the superposition of all the elementary waves ($\mathcal{U}_i(\mathbf{p})$,..., $\mathcal{U}_j(\mathbf{p})$) originating on all points in our scene:

$$\mathcal{U}(\mathbf{p}) = \sum_i \mathcal{U}_i(\mathbf{p}) = U(\mathbf{p}) e^{i\varphi(\mathbf{p})} \quad (7)$$

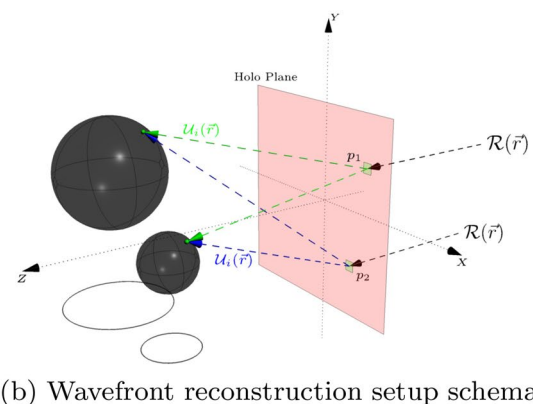
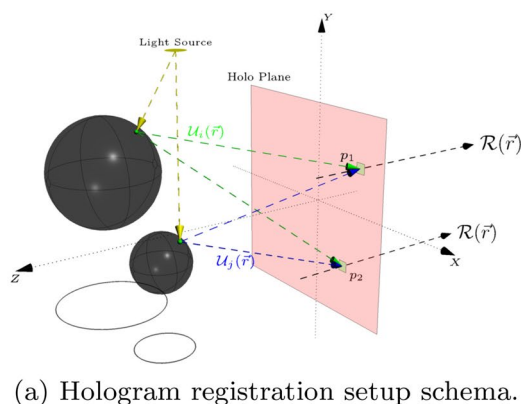


Fig. 3 Basic schema for digital holography registration and operation. For clarity reasons, the direction of reference wave $\mathcal{R}(\mathbf{r})$ is opposite to the direction of object waves and only two plane coordinates and two object coordinates are shown. Scheme **a** shows the main ele-

ments: holo plane, light source, object waves ($\mathcal{U}_i(\mathbf{r})$, $\mathcal{U}_j(\mathbf{r})$,...) and reference wave ($\mathcal{R}(\mathbf{r})$), used in hologram registration procedure. In scheme **b** the hologram is illuminated with a reference beam ($\mathcal{R}(\mathbf{r})$) and recreates the 3D scene

If the hologram plane is illuminated with both $\mathcal{U}(\mathbf{p})$ and a reference wave $\mathcal{R}(\mathbf{p}) = R(\mathbf{p}) e^{i\varphi_R(\mathbf{p})}$, the total complex wave $\mathcal{H}(\mathbf{p})$ on each point on the HP plane is the superposition (interference) of both. The resulting complex wave after interference at the HP plane can be calculated as:

$$\mathcal{H}(\mathbf{p}) = \mathcal{U}(\mathbf{p}) + \mathcal{R}(\mathbf{p}) = \sum_i \mathcal{U}_i(\mathbf{p}) + \mathcal{R}(\mathbf{p}) \quad (8)$$

and the hologram can be defined as the interference pattern of the object light wave $\mathcal{U}(\mathbf{p})$ with the reference wave $\mathcal{R}(\mathbf{p})$. From now on, and for the sake of clarity, all references to positional parameters \mathbf{r} or \mathbf{p} will be omitted in the formulas.

When \mathcal{U} is added with a reference wave \mathcal{R} , total intensity at hologram plane is calculated with the following well known expression:

$$\begin{aligned} I &= \mathcal{H} \times \mathcal{H}^* \\ &= (\mathcal{U} + \mathcal{R}) \times (\mathcal{U} + \mathcal{R})^* \\ &= (\mathcal{U} \times \mathcal{U}^*) + (\mathcal{R} \times \mathcal{R}^*) + (\mathcal{U} \times \mathcal{R}^*) + (\mathcal{U}^* \times \mathcal{R}) \\ &= I_U + I_R + |\mathcal{U}| |\mathcal{R}| \cos(\varphi - \varphi_R) \end{aligned} \quad (9)$$

When a plate of photosensitive material is placed at the HP plane, it behaves with a linear response to the energy carried by the incident waves, so it is possible to record $I(\mathbf{p})$ getting a transmittance function $t(\mathbf{p})$ proportional to it.

$$t \propto I = I_U + I_R + |\mathcal{U}| |\mathcal{R}| \cos(\varphi - \varphi_R) \quad (10)$$

Then, the hologram (understood as a photosensitive plate that stores the transmittance function $t(\mathbf{p})$) encodes combined information of both the amplitude and the phase of the object wave \mathcal{U} . More specifically, this plate is much more sensitive to variations in phase difference $\varphi - \varphi_R$, than to absolute value of intensity itself.

When this hologram is later illuminated with a *reconstruction* wave $\mathcal{R}' \simeq$ similar to the reference one \mathcal{R} (see Fig. 3b), the light wavefront exiting from the hologram is composed by four terms:

$$\begin{aligned} U_{out} &= t \mathcal{R}' \\ &\propto I_U \mathcal{R}' + I_R \mathcal{R}' + (\mathcal{R}^* \mathcal{R}') \mathcal{U} + (\mathcal{R} \mathcal{R}') \mathcal{U}^* \end{aligned} \quad (11)$$

First two terms are related with registered intensities I_U and I_R . There are another two terms associated to real image \mathcal{U} and conjugate image \mathcal{U}^* , respectively [5].

It is usually assumed, without loss of generality, that both $\mathcal{R}(\mathbf{r})$ and $\mathcal{R}'(\mathbf{r})$ are plane waves, with amplitude normalized to 1 and perpendicular to hologram plane, so phase is the same on every point, and can even be set arbitrarily equal to 0. Under these assumptions, registration and reconstruction waves becomes just $\mathcal{R}(\mathbf{p}) = 1$ and $\mathcal{R}'(\mathbf{p}) = 1$. Then, the expression for the wave exiting the hologram plane becomes:

$$U_{out} = k + \mathcal{U} + \mathcal{U}^* \quad (12)$$

Even in this scenario, there are practical issues to solve [13]: removing the I_U , I_R and conjugate images terms or reducing speckle noise are those that can be address via computing effort.

Math for Hologram Synthesis

The primary goal for hologram recording is to obtain the phase (and/or intensity) pattern of a light wave front over a plane and save it on a registration medium, which will be used later to modify an illumination wave with well-defined propagation parameters $\mathcal{R}(\mathbf{r})$ wave in Fig 3) to re-obtain the original wave front $\mathcal{U}(\mathbf{r})$. The reference illumination wave plays a decisive role in this process. The entire analog holography process is based on interference to obtain a wave with an intensity that is proportional to phase shifts. For this reason, if we only use a monochromatic laser source with constant propagation vector \mathbf{k} , perpendicular to the SLM plane, the mathematical expressions are then the simplest among all the configurations.

If this entire process is moved to the digital domain, we do not need any interference tricks to obtain information about phase shifts of waves on the SLM plane as they can just be numerically evaluated. Consequently, there is no need for interference with a reference wave of any kind, just a numerical model of object waves $\mathcal{U}_i(\mathbf{r})$, which should be evaluated at the SLM plane as $\mathcal{U}_i(\mathbf{p})$, and added together to obtain the full wave front $\mathcal{H}(\mathbf{p})$ reaching the SLM plane:

$$\mathcal{H}(\mathbf{p}) = \mathcal{U}(\mathbf{p}) = \sum_i \mathcal{U}_i(\mathbf{p}) = H(\mathbf{p}) e^{i\varphi(\mathbf{p})} \quad (13)$$

It is worth noting that no reference wave is used in this computerized version of hologram registration. The numerical values for amplitude $H(\mathbf{p})$ and/or phase $\varphi(\mathbf{p})$ can be loaded on a digital SLM from $\mathcal{H}(\mathbf{p})$ to be illuminated with a coherent light source, and our wave front retrieved.

Several approaches can define object wave sources $\mathcal{U}_i(\mathbf{r})$ for virtual 3D scenarios and propagate and add them to the hologram plane. They are based on point sets, primitive meshes and Fourier transform methods. There are also efforts to extend the usual computer graphics tools to address issues of CGH tasks, such as ray tracing to obtain correct occlusion [25], the scattering-transmission function [15] or random-phase method to speckle noise reduction [21]. These approaches can work both in object and in image/SLM space:

Object Space

Select which elements in the scene generate individual wave fronts and propagate them to the hologram plane.

These methods can use different elements to generate waves:

- Point sets: select individual points all over the scene, define a spherical wave starting at each one, and propagate them to every SLM pixel.
- Complex primitives: mesh the scene using some kind of primitives, for which defining the light wave front they emit is simple enough, and propagate them. This is currently only feasible for triangles.
- Fourier methods: slice the object space in planes that are parallel to the HP, which allow using Fourier transform propagation methods for waves.

Image/SLM Space

For each pixel \mathbf{p} on the SLM, obtain all the points in the scene contributing to interference at $\mathcal{U}(\mathbf{p})$. Although this is usually done by simple ray casting, it can be enhanced with path-tracing methods to improve quality and realism.

The final aim of this paper is to use a full Monte Carlo path-tracing algorithm to generate realistic images that can be visualized with holographic methods. However, to achieve this goal, a full infrastructure for simulation and visualization must be validated first. In this preliminary work, we use a point-set approximation to define complex object wave fronts, generate holograms and visualize them. This synthetic hologram generation and visualization pipeline is validated with some examples by comparing the simulation and real visualization of the models on a physical SLM. We also propose extending these algorithms to full Monte Carlo path-tracing generated holograms in "[Holographic Monte Carlo Path Tracing \(HMCPT\)](#)" section.

Limits of Holographic Display

As has been noted before, holograms are usually treated as the superposition of planar gratings and the limit of their spatial frequency is given by pixel size: a grating whose elements are separated by distance P diffracts a normally incident beam of light of wavelength λ into a set of beams, at angles θ_n given by

$$\sin(\theta_n) = \frac{n\lambda}{P}, \quad n = 0, \pm 1, \pm 2, \pm 3, \dots \quad (14)$$

This is known as the grating equation. The finer the grating spacing (P), the greater the angular separation of the diffracted beams. The order zero corresponds to $n = 0$, the first orders to $n = +1$ and $n = -1$, and so on.

The detailed structure of the repeating pattern determines the form of the individual diffracted beams and their relative intensity, while the grating spacing always determines the angles of the diffracted beams. For digital media (e.g.,

displays, TV, etc.), the minimum hologram period is $P = 2p$, where p is the pixel size.

The physical parameters of the human visual system, such as field of view, spatial resolution or visible spectrum, among others, are limited. Therefore, the spatial bandwidth of a holographic display device may also be limited based on the corresponding limitations of the human visual system. Studies of the human visual system show that the field of view, $2\phi_v$, in which the retina has the maximum resolution, is between 2 and 5 degrees. However, the resolution of the perceived image, derived from the intensity on the retina, degrades for higher viewing angles. In any case, a low-resolution image may still extend to 100–140 degrees of viewing angle, intruding what is known as peripheral vision.

Semiconductor technologies are currently involved in a race to improve SLM devices: state of the art for commercial ones offers a pixel pitch of about $5 \mu\text{m}$ and an SLM size of about 2 cm. The diffraction angle is under 10° and drops when control of the diffraction efficiency is mandatory (as each period needs more pixels). These data are far from being suitable for commercial holographic displays. For example, [2] concludes that a $100 \text{ mm} \times 100 \text{ mm}$ hologram and a diffraction angle of $15^\circ \times 15^\circ$ would require a pixel pitch of $1.18 \mu\text{m} \times 1.18 \mu\text{m}$.

In addition, for a digital hologram whose size is L_{\max} and whose pixel period is Δp , the required number of pixels along one axis is $L_{\max}/\Delta p$. All these fundamental relationships directly affect cost computing, defining the bandwidth to record a specific hologram. Previous works show that using three wavelengths and 30 fps implies 3.6×10^{11} pixels per second. These estimations are made once the holograms are already calculated and do not include the time cost to synthesize the scene.

To understand real bandwidth needs, it should be noted that an analog reflection hologram (used in classical holography to display 3D scenes) can have diffraction patterns with a spatial frequency of 3000 l/mm (lines per millimeter) (even more); this clearly implies sub-micron pixel size (at least $0.3 \mu\text{m}$). Since current visualization devices have limited spatial frequency, imposed by pixel size, any efforts made to achieve realistic scenes could be compromised by this limitation.

The synthesis of scenes based on ray tracing, such as the one described in the previous section, allows simple scenes to be generated directly but as the scene contains more information—more points—problems related to the SLM resolution and dynamic range appear, which affects image quality.

Simulation is a tool that allows us to predict the behavior of these devices in advance. The computation problem is, in this case, twofold. First, from a synthetic 3D scene, the hologram presented to the SLM must be known. In a second step, the 3D scene can be visualized by modeling the behavior of the SLM, which, in the first approximation, is related

to pixel size and the modulation of intensity and phase that it introduces into the illumination beam.

The results of this simulation can be contrasted with the behavior of real SLMs. From there we can foresee the behavior of SLMs that are not currently on the market due to the aforementioned manufacturing limitations.

Holographic Monte Carlo Path Tracing (HMCPT)

Path tracing is a computer graphics Monte Carlo method for rendering images of 3D scenes so that global illumination is faithful to reality. Fundamentally, the algorithm then performs the integration of all the illuminance arriving at a single point on the surface of an object. Next, the illuminance is reduced by a surface reflectance function (BRDF) or a surface transmittance function (BSDF) to determine how much of it will go towards the camera on the viewpoint. This integration procedure is repeated for every pixel in the output image. When combined with physically accurate models of surfaces, accurate models of real light sources and optically correct cameras, path tracing can produce still images that are indistinguishable from photographs.

Path tracing naturally simulates several effects, such as soft shadows (sampling the solid angle of the light sources), depths of field (sampling the camera aperture), motion blur (sampling the shutter time), indirect lighting and interreflection (sampling path traversal) or dispersion (sampling in wavelength). By working with the electromagnetic field, polarization, interference and diffraction effects can be simulated, and by sampling the area associated with the image pixel, aliasing phenomena can be reduced.

Since path tracing is accurate and unbiased, it is used to generate reference images when testing the quality of other rendering algorithms. Obtaining high-quality images from this method requires tracing many rays to avoid noisy artifacts.

The light traveling to the observer may come from different physical effects, which are usually classified as reflection and transmission. There are different types of reflection and transmission functions, and each algorithm used to solve the global illumination problem can accurately capture only some of them, or, in other words, follow different kinds of paths for light propagation.

To describe combinations of paths an algorithm can simulate, it is common to code each kind of interaction with a character K :

- E step where light reaches to the eye
- D diffuse reflection or transmission
- G glossy reflection or transmission
- S specular reflection or refraction

L light emission on a source

and use regular expressions to describe the paths each algorithm can follow:

- K^+ one or more of K events
- K^* zero or more of K events
- $K^?$ zero or one K event
- $(K | K')$ a K or K' event

Particular rendering techniques may be characterized by the paths that they consider:

- ED^*L radiosity
- $E(D | G)L$ ray casting
- $E[S^*](D | G)L$ recursive ray tracing

Standard MCPT

In traditional MCPT, rays are shot from the eye through pixels on the image plane and paths are created through the scene. The illumination reflected to the eye is computed at each intersection point between the path and the scene. A path is cut off after a random number of bouncing steps, so that the light propagation is stopped. This light is stored at a pixel on the image plane. Obtaining a precise estimation of the light arriving at a certain pixel requires evaluating multiple paths for each of them.

Basic schema for standard MCPT is depicted in Fig. 4a, where “light source” is the simulated illumination source and the observer will see the 3D scenario through the image window.

With the encoding depicted above, path tracing can be described as

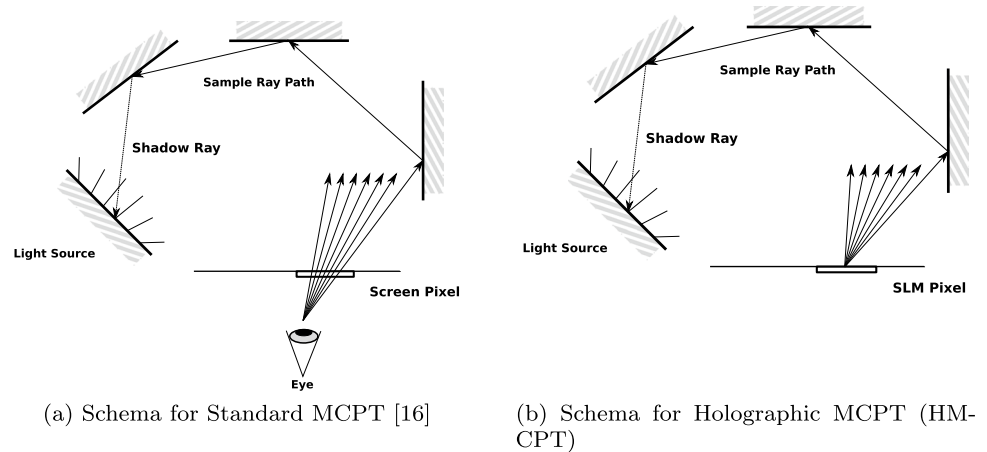
$$E[(D | G)S]^*(D | G) \quad (15)$$

For each pixel on the image plane, paths are defined starting from the eye and passing through the pixel. On each scattering event, a reflected or refracted ray direction is chosen depending on the BDSF of the material, thus performing correct Monte Carlo integration.

Holographic MCPT

In our Holographic Monte Carlo Path Tracing (HMCPT), a path is also cut off after a random number of bouncing steps, but the light coming from points in the scene needs to be propagated to every pixel at the SLM registration media, because we must manage light as a scalar wave, not just as a single reflected ray. This can be implemented either by modeling the emitted light as a spherical wave front or by

Fig. 4 Schemas for MCPT (a). For HMCPT case (b) each SLM pixel acts as an observer



shooting multiple rays for each pixel on the SLM. Our basic schema is depicted in Fig. 4b, where the SLM will register the appearance of the scene as a whole wave front. Later, when a set of multiple observers look at the SLM under the right illumination conditions, they will be able to see the real 3D scenario, each from a different point of view (or eye position). The SLM stores the scene appearance for a whole range of points of view (under the limits defined in "Limits of Holographic Display" section).

In this method, multiple rays are also shot to calculate each pixel on the SLM, but they do not depend on eye position (for hologram calculation there is no eye). Instead they should cover all solid angles from where light can reach the pixel on the SLM or every point-wave source that is part of the scene.

The Holographic Monte Carlo Path Tracing (HMCPT) Algorithm

Our goal with this HMCPT algorithm is to have a static scene (i.e., geometry and materials do not change over time) and be able to generate a sequence of images in real time as the viewer changes position.

This real-time requirement can be defined as our target FPS (frames per second) render rate. This FPS rate ranges from 25 FPS for traditional cinema, 50FPS 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.

Since we suppose (for the moment) that rendering a frame is independent from other frames, this FPS merely serves to define the maximum render time we have for each frame, or, from an opposite viewpoint, if we estimate the computer power needed to render a frame, we can easily scale it to what we need for each second of animation.

As stated above, we have a static scene, extended with an acceleration structure, that is, any kind of spatial index that serves the purpose of reducing the time to intersect a ray

with the whole scene. The time to intersect a ray with the scene is critical, depends on the acceleration structure and can be estimated for each structure. We show an example for a uniform spatial subdivision scheme.

The Holographic Monte Carlo Path-Tracing (HMCPT) flow can be described succinctly with a simple procedure: bounce a ray into the scene, accumulate the loss of light intensity on each bounce (defined by the surface material properties), and, when the ray finally hits a light, multiply light intensity for the accumulated loss factor to obtain what finally arrives at the pixel.

From a more formal standpoint, the algorithm can be detailed by the following steps (see Fig. 4b):

For each pixel on the SLM

For each sample of illumination desired on the pixel

- Step 0: Initialize accumulated reflectance to 0.
- Step 1: Choose a sampling direction over the hemisphere on the pixel and define the ray to shoot into the scene as first step in the path.
- Step 2: Repeat until ray stops bouncing
 1. Step 2.1: Trace ray into the scene, finding intersection point with the geometric model.
 2. Step 2.2: Eval the BRDF at point of intersection, and accumulate it.
 3. Step 2.3: Decide whether to stop or scatter the ray based on BRDF.
 4. Step 2.4: If ray scatters, define scattering direction depending on the BRDF and go to Step 2.1. If it does not, bounce ends and exit the loop to Step 3.
- Step 3: Trace a shadow ray to light source to check if it is not occluded.
- Step 4: If it is not occluded, calculate direct light contribution times accumulated reflectance.
- Step 5: Define the wave $\mathcal{U}(\mathbf{r})$ with origin in the first intersection point of the path and direct light from Step

4 as amplitude (Eq. 5), propagate it to the pixel and accumulate it with Eq. 8.

Total Time in HMCPT

The total time of the HMCPT T_{HMCPT} process will be the sum of the time spent in the various steps in the algorithm above. If we suppose that all pixels (in number P) are calculated with the same number of samples (let it be S), the total time for rendering one frame can be estimated as:

$$T_{HMCPT} = (P \times S) \times (T_1 + T_2 + T_3 + T_4 + T_5) \quad (16)$$

(where each time T_i is the time needed to complete step i), and for a full second of animation at F frames per second:

$$T_{HMCPT,F} = (F \times P \times S) \times (T_1 + T_2 + T_3 + T_4 + T_5) \quad (17)$$

In other words, all the paths must be generated (time T_1 for Step 1) and for all paths we must check whether the light source is occluded (time T_3 for Step 3) at the end and accumulate the light source contribution (times T_4 and T_5). Estimating time T_2 for Step 2 is slightly harder, as the number of iterations depends on the random decision of whether the ray bounces or not. For the rest of the discussion we will focus on the time needed to render just one of our independent frames.

Time T_2 is a function of the number of bounces a path makes on the scene, also known as the *path length*. This number can vary for different paths, but we can make a statistical estimation of its average length for the full hologram rendering.

If we define ρ as the average probability that a ray bounces on a surface, it will depend on average scene reflectivity and transmittance. Such a number can be easily estimated from material properties.

Let us imagine that we shoot all rays from the SLM at the same time. At the first step, we launch $N = P \times S$ rays from the pixels on the SLM. If all the rays bounced the same number of times (k), the total number of rays N_B that we must bounce and trace into the scene would be

$$\begin{aligned} N_B &= N + k \times N \\ &= (1 + k) \times N \\ &= l \times N \end{aligned} \quad (18)$$

with $l = 1 + k$ as the *path length*. As rays will only bounce randomly or not bounce at all depending on ρ , paths have not the same length. After the first bounce, only ρN rays will remain “alive” on average. After the second bounce, only $\rho \times \rho N = \rho^2 N$ rays will stay. After k bounces, the number of alive rays can be determined as $\rho^k N$. Therefore, the total number of rays we should trace to evaluate all paths in the SLM calculation will be:

$$\begin{aligned} N_B &= N + \rho N + \rho^2 N + \dots + \rho^k N \\ &= (1 + \rho + \rho^2 + \dots + \rho^k) \times N \\ &= \frac{(1 - \rho^{k+1})}{1 - \rho} \times N \end{aligned} \quad (19)$$

As the number of times that a path can bounce in the scene is theoretically infinite, the estimated number of rays when k goes to infinite is:

$$N_B = \frac{1}{1 - \rho} \times N \quad (20)$$

We can define $l_{avg} = \frac{1}{1 - \rho}$ as the average length of the paths, so

$$N_B = l_{avg} \times N \quad (21)$$

The total time for HMCPT can be then estimated as

$$\begin{aligned} T_{HMCPT} &= N \times T_1 + N_B \times T_2 + N \times T_3 \\ &= N \times T_1 + (l_{avg} \times N) \times T_2 + N \times T_3 \\ &= N \times (T_1 + l_{avg} \times T_2 + T_3) \end{aligned} \quad (22)$$

Times related to evaluation and sampling of the BRDF ($T_1, T_{2.2}, T_{2.3}$ and $T_{2.4}$) are usually very short compared to the time to trace a ray into the scene, which is the most computer-resource consuming task in any algorithm using ray tracing as a sampling method. If we call this trace time T_t and neglect all other times in our equation, so $T_{2.1} = T_t$ and $T_3 = T_t$, total time can be reduced to

$$\begin{aligned} T_{HMCPT} &= N \times (l_{avg} \times T_{2.1} + T_3) \\ &= N \times (1 + l_{avg}) \times T_t \end{aligned} \quad (23)$$

Time T_t is dependent on the kind of acceleration structure used to speed up the intersection of rays and scene geometry (voxel grids, BSP, trees, etc.). An estimation of this cost can be seen in Appendix A.

Image Reconstruction from Synthetic Digital Holograms

This section contains details of both the synthesis of digital holograms and image reconstruction from them. Image reconstruction can be handled both in a laboratory with a physical SLM and on a computer by means of a numerical simulation. Simulation in CGH implies visualizing how it will look when shown on a real SLM device. The goal is to generate a synthetic hologram (computer calculated, not registered) and compare the visualization from a physical SLM device and the simulation obtained from our pipeline.

Hologram Synthesis

First step is hologram synthesis. Based on the theoretical method exposed in "[Math for Hologram Synthesis](#)" section, specially in Eq. (13), light wavefronts are calculated for different 3D scenes, and its interference pattern is calculated on the SLM plane.

As the interest in preliminary steps focuses merely on the correctness of the simulation, a point mesh approximation is used as a first test case (some points in the scene are selected as wave sources), and models are chosen in such a way that the need for an occlusion check (HSR, hidden surface removal, as it is commonly known in computer graphics) is simple enough or even unneeded. More complex scenes will be tested later.

The final result of the synthesis step is the interference map at the SLM plane, which is mainly a rectangular array of complex numbers (phasors). Depending on the capabilities of the physical SLM used, the information for module and/or phase for this map is stored, usually as gray-level images on conventional standard image files, such as JPEG or PNG. These images can be later fed to the SLM to reconstruct the scene wave front.

Physical SLM Setup

Our physical setup includes a HeNe laser, standard optics to expand and collimate the light, a beam splitter to illuminate the SLM (Pluto by Holoeye™), a polarizer and a camera without a lens. The laboratory setup can be observed in Fig. 5.

The absence of a lens in the camera allows capturing the wave front exiting from the SLM directly, without the interference caused by any other optic equipment. The sensor plane in the camera intersects the wave generated by the SLM. The focus can be set on a specific plane so that the geometry out of this plane will look out of focus.

The SLM is loaded with the data stored in the previously described interference map and illuminated with the coherent light source (laser), and its wave front is registered on

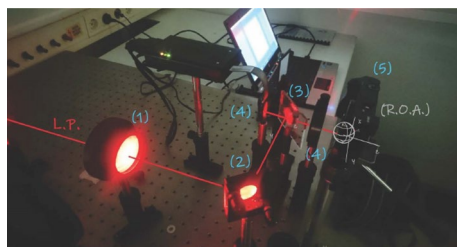


Fig. 5 Basic experimental setup for hologram visualization. L.P: light path, (1) collimating lens, (2) first surface mirror, (3) beam splitter, (4) SLM/Hologram plane, (5) CCD camera (R.I.A., Real Image Area)

the camera sensor and stored as an image file. This step makes it possible to verify the correctness of the hologram generation procedure.

Simulated SLM setup

In the simulated SLM workflow, the wave coming from the hologram must be propagated to a virtual camera sensor. Without any loss of generality, we can set our virtual sensor parallel to the plane of the SLM (as in the physical SLM setup), so propagation is just a convolution with the Kirchhoff diffraction kernel. This convolution can be carried out by means of Fourier transforms. The steps are as follows:

- Calculate the outgoing light wavefront at the SLM plane
- Transform both wavefront and diffraction kernel to the Fourier space
- Multiply in Fourier space
- Transform back to physical space

Although both transforms and multiplications can be time-consuming tasks, use of FFT and optimizations for multi-core CPUs and GPUs can bring times to acceptable levels. The simulation result is also an image, which can be compared to the one captured by the camera in the physical SLM workflow.

Results Comparison

Some examples are used to verify the correctness of the simulation pipeline. They range from simple to more complex models to check how the system behaves with respect to the common holography pitfalls described in previous sections. The results of the physical SLM setup reveal issues such as diffraction of SLM borders, speckle and higher order images, among others.

Figure 6 shows a wireframe cube. It is used as a simple enough scene to check geometry sizes, effect of camera sensor position on focus, and SLM behavior with a number of wave sources that is not so high. Even in this first image, out-of-focus geometry is seen correctly, showing a depth-of-field effect, not due to aperture or focus of the camera optical system of (there is none, as described in 5), but to the propagation of the wave front.

A wireframe sphere is shown in Fig. 7. It depicts higher point density but is still arranged on a wireframe fashion. Out-of-focus lines are even more evident at the back of the sphere.

Figure 8 shows a monochromatic solid quad that is slightly tilted with respect to the camera. The quad is rendered with different sampling resolutions (16 and 256 points per side) in order to test aliasing problems when

Fig. 6 Images of a synthetic hologram of a wireframe cube visualized on a physical SLM versus a simulated SLM

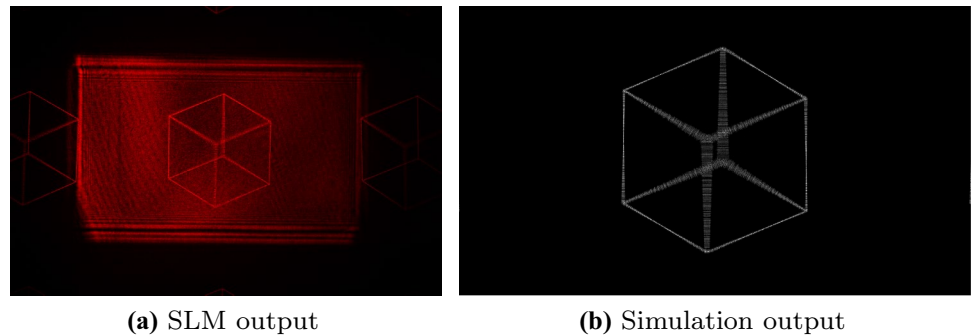


Fig. 7 Images of a synthetic hologram of a wireframe sphere visualized on a physical SLM versus a simulated SLM

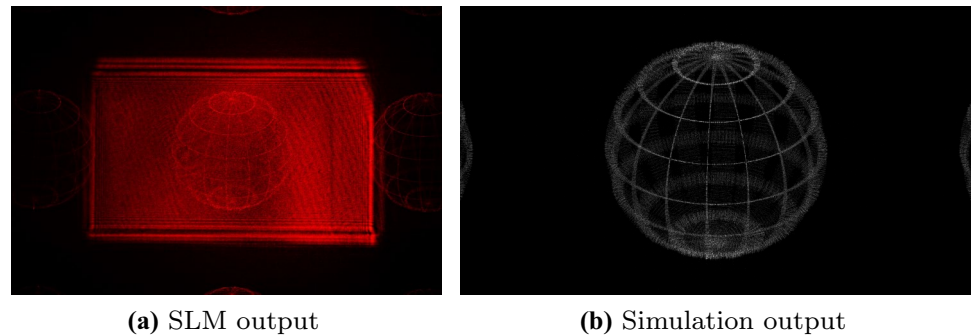
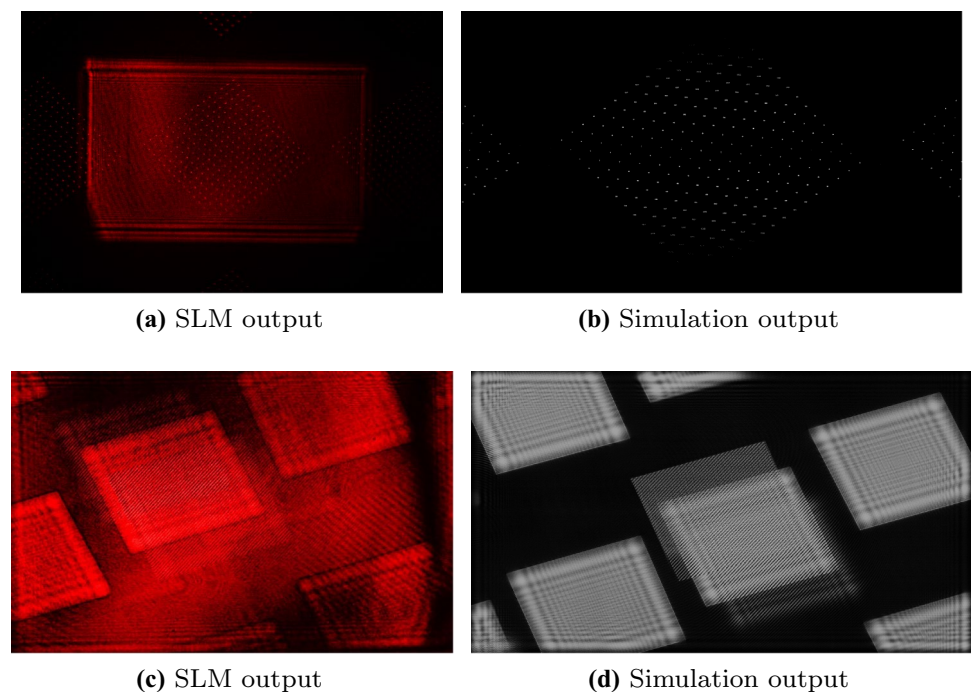


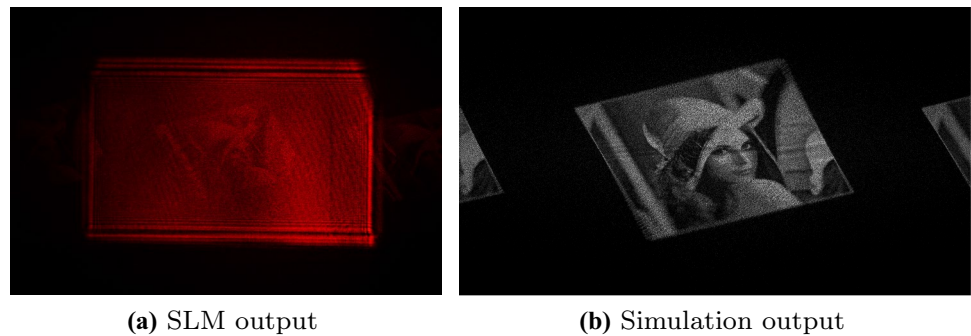
Fig. 8 Images of a synthetic hologram of a solid quad, meshed with 16×16 points (upper row) and 256×256 points (lower row), visualized on a physical SLM versus a simulated SLM. Main diagonal of **a** and **b** images intersects with CCD camera plane and can be observed on focused points. Figure **c** and **d** shows the limits of SLM resolution: There are artifacts on both real and simulated images



point density becomes so high that it reaches SLM limits. Out-of-focus areas can also be observed at the top and bottom corners. In both the physical and the simulated SLMs, the effect of the artifacts appearing due to the limitations imposed by the SLM resolution is clearly observed.

Figure 9 shows a textured quad that is slightly tilted with respect to the camera. The quad is rendered with a high sampling resolution (128 points per side) to test the effect of high spatial frequency in scene illumination information. Again, some issues related to bandwidth and diffraction efficiency appear in the real image. We conclude that, to

Fig. 9 Images of a synthetic hologram of a textured quad visualized on a real SLM versus the simulation to show the limits of the real SLM. Figure **a** shows the limits of real SLM. Lenna image can be recovered but it appears with a lower contrast due to SLM modulation calibration and other effects, such as speckle



introduce more powerful image synthesis techniques (such as HMCPT), current SLMs cannot reproduce the information involved on a high resolution (both in space and in color) hologram.

Towards HMCPT Based Holograms: A First Approximation

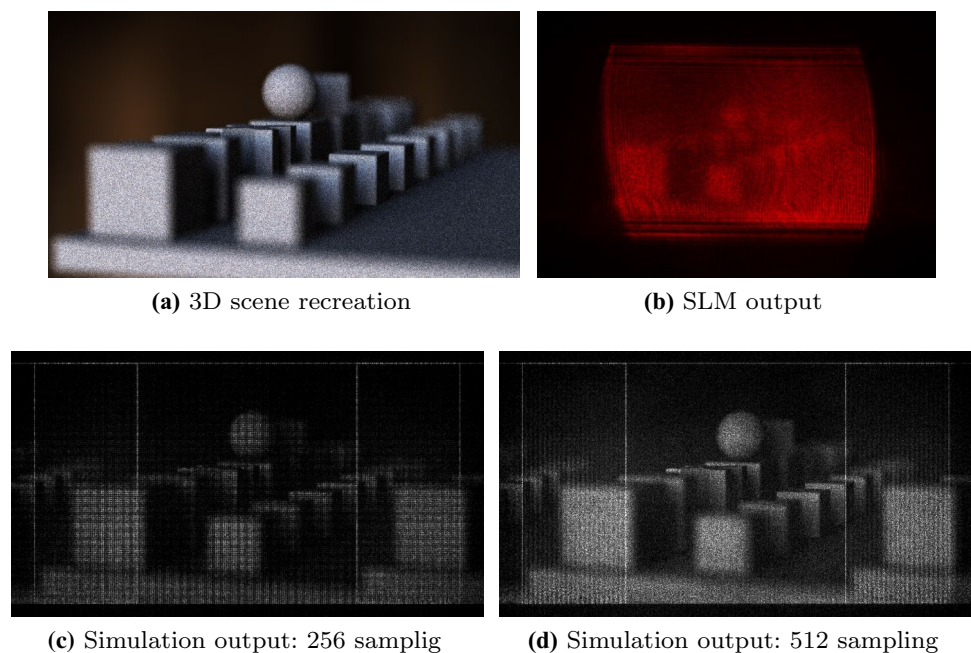
Figure 10 shows a synthetic scene (a model of a simple chess game board) calculated using HMCPT methods, and the simulation results for two virtual SLMs. Figure 10a is a scene recreation: chess under simple illumination conditions. This image is generated using a preliminary version of the HMCPT formalism described in section "[Holographic Monte Carlo Path Tracing \(HMCPT\)](#)". The use of this final HMCPT image may result in a hologram that can be displayed or simulated via the standard ray-tracing methods described in section "[Fundamentals of 3D Imagery Display](#)", which have already been applied to other previous figures.

Figure 10b shows the SLM output using the basic setup in Fig. 5. There are several issues related to the limits of our SLM characteristics (dynamic range and spatial bandwidth) and others that are inherent in coherent light (such as speckle). We should note that there is not an exhaustive analysis of diffraction efficiency yet; this work is focused only on image formation.

Figures 10c and d show the simulation results. There is overlapping between orders due to scene size (which can be observed in the SLM image as well), but they help to understand the behavior of complex synthesized holograms.

We propose a first step to try to understand, and implement accordingly, the operation of a virtual SLM. To do this, we build a simulable mathematical model of the system, which collects its essential parts, involving variables, parameters, formulas, relationships, diagrams, etc. In this way, we can predict its behavior, which is currently impossible for certain types of problems and resolutions such as the one presented in this paper.

Fig. 10 SLM and simulation behavior for a complex scene rendered with a basic HMCPT algorithm: **a** Recreation of a chess game (including a short depth of field) to show the scene to be calculated. **b** SLM output showing several issues: limited dynamic range and spatial bandwidth, speckle, or lack of calibration of physical setup (**c** and **d**). Simulated images for scene using 256 (**c**) or 512 (**d**) resolution sampling over the scene. In both the real and simulated images, overlapping between diffraction orders can be observed



Our goal is also to use configurable virtual SLMs, such as the one proposed, to perform *a computational approach to achieve photo-realistic computer-generated holograms* using Monte Carlo path-tracing methods. It is a first approximation to generate realistic scenes and it allows us to estimate the computing effort they involve. In our approach, we have decoupled the two systems by applying MCPT to generate an image and then used that image to generate the synthetic hologram and its visualization through the virtual SLM.

Table 1 shows the computer effort needed for hologram synthesis and simulation of the example scenes in this work. We show the number of *point sources* (waves) used for each figure. Since each interferogram pixel must be calculated using all the wave sources, the number of point-wave sources used should be conceptually equivalent to the number of rays per pixel shot in conventional image synthesis. To understand the computation time, we must think that a hologram with 4000 sources is “equivalent” to an image rendered using 4000 rays per pixel. The most complex model shown (Fig. 10d)) based on HMCPT uses about 500,000 waves (or rays) per pixel. CPU and GPU synthesis times are related to the cost of propagating all the sources to the HP. Finally, column *SIM* shows the time needed to propagate the wave from SLM to the (virtual) CCD plane image, which is more or less the same for all tests, because all the images use the same FFT size (a full-hd 1920×1080 resolution image, padded to a double resolution 3840×2160 size to avoid cyclic convolution).

Conclusions and Future Work

In digital holography, and for simple scenes (dots, lines, etc.), computing needs increase significantly, starting with more demanding calculations than in synthetic images. These calculations arise from the need to know the interference pattern in a hologram recording plane (i.e., the diffraction pattern that an SLM must present to a reconstruction beam).

Table 1 Images simulation time cost (ms)

Figure	Sources	CPU	GPU	SIM
Fig. 6b	1536	40853	1838	1073
Fig. 7b	15840	394336	16157	1075
Fig. 8d	65536	1589932	66802	1104
Fig. 9b	65536	1636757	67202	1082
Fig. 10c	116480	–	135398	1073
Fig. 10d	465920	–	496128	1077

Sources: number of scene points. CPU: synthesis time using CPU (Core i7-4790 @ 3.60GHz). GPU synthesis time using GPU (nVidia GTX 1060). SIM scene simulation time (wave propagation from SLM plane to image plane)

As in synthetic images, in which photorealism plays an increasingly relevant role, applications requiring computer-based holograms demand scenes that are indistinguishable from reality (holo-realistic if we mimic the conventional photo-realistic term applied to traditional high-quality rendering). Again, the computational effort increases significantly, and strategies must be established to optimize this computational cost to obtain the desired image quality.

This work focuses on how a synthetic scene illuminates the SLM plane, proposing an algorithm based on holographic Monte Carlo path-tracing techniques (HMCPT). Thus, each beam originating from one single laser source of illumination travels through the scene and interacts with several surfaces until it reaches the SLM plane. Amplitude and phase information must be carried and the BRDF function must be defined for each surface.

In particular, this work makes the following contributions are the following:

- It extends the IBR continuum: Fig. 1 shows rendering types for synthetic images (from [21]): Solid lines are related to established techniques and dashed lines show the approximation in this work, including computer-generated holograms
- The principal reasons for this effort are: (a) holographic displays mimic natural eye behavior from the point of view of convergence and accommodation, which allow us to predict them as future devices when the current technological and manufacture limits are overcome (b) holography allows several observers to see the same 3D scene, placed on different points of view, without changing the information on the SLM display. Restrictions will be saved.
- It recapitulates basic holography principles, the limits of holographic display, and presents an example of actual SLM technology, and the parameters needed for future display.
- It presents our holographic Monte Carlo path tracing and: (a) theoretical total computation time for a static frame, one order of diffraction and one laser source (see Formula 23). The total computational cost, T_{HMCPT} , considers the cost as the sum of several terms related to the cost of tracing rays into the scene, and the summation of wave front complex amplitudes in the SLM. (b) An estimation of the time based on the average length of the paths, see Formula 19. And, (c) the time for an example for an acceleration structure used to speedup the intersection of rays, see Appendix A.
- SLM state of the art has been evaluated to display complex scenes. Bandwidth limits have been observed, both for real and simulation images. The use of simulators of 3D scenes that cannot yet be displayed has been proposed.

- Following the successive refinement methodology, the simulated SLMs have been observed to behave sufficiently similar to the real system, as has been demonstrated in the results that we have already obtained and shown.
- Our goal has been to incorporate sufficient image realism elements as well as simplicity of development and calculation time. To do this, we demonstrate that our virtual SLM serves as a close approximation to the real SLM system and that it can incorporate most image characteristic aspects, and we have developed a calculation system for generating the “real image”.

The total computational cost, T_{HMCPT} , considers the cost as the sum of several terms related to the cost of tracing rays into the scene, and the summation of wave front complex amplitudes on the SLM.

Future Work: Inherent Issues of Computer Holography

Diffraction optics has its own rules and object wave front quality depends on the following holographic display key parameters that affect the computation cost that would be included in total time cost.

Simulators for synthesized 3D scenes: Designing and displaying scenes with photo-realistic quality, for example complex illumination patterns, involves devices that are not yet on the market. Simulation techniques are useful tools for evaluating them. We propose combining, in a single simulator, the virtual SLM and the realistic image generation model based on the MCPT technique, whose level of complexity for system programming is orders of magnitude higher than our current approach.

Diffraction efficiency: It is mandatory in 3D display applications for the main diffracted order to obtain all the input energy. Scalar theory predicts an ideal diffraction efficiency η for continuous variation of the phase profile ($\Delta\Phi$), in other words, continuous variation of the refraction index (Δn); however, in an SLM there are discrete phase levels, and η depends on the number of phase levels M . For 8 phase levels, the diffraction efficiency is about 95%, whereas for 2 phase levels (binary gratings) it is only 40.5% [8]. Therefore, in a first approximation, 64 subpixels (82 pixels) would be needed for every pixel considered in the digital hologram, increasing the calculus cost by a factor of almost 100.

Full colour holography: The use of several light sources multiplexed to obtain the adequate color effect implies the calculus of several holograms (one per light source) and multiplying the computation time by in the same factor.

Multiple illumination sources: As real scenes have several illumination sources, obtaining holorealistic synthetic scenes requires including them in the calculus. Again, multiplexing techniques are the right way, but they increase the time computing cost again.

Speckle: Speckle has a statistical distribution in size and intensity and it is associated with highly coherent light sources, both spatially and temporally. Consequently, speckle removal techniques essentially involve applying low-pass filters (e.g., median) with image enhancement or using multiple references waves to superpose various speckle field distributions [10]. Again, we must pay a cost (in terms of computing time) to remove this effect.

Inherent issues due to digitalization: The digitization of holograms introduces its own characteristics, which limits the quality of the scene formed. In fact, the use of scanning techniques (required for digital modeling of a synthetic scene) necessarily involves defining a finite pixel size. Reproduction by a hologram (e.g., SLM) also means defining the digital device pixel size of this device and the separation between adjacent pixels. Finally, we must also consider that the size of the SLM also influences the final quality [12].

Appendix A: Cost Analysis of Ray Casting Algorithm

Given a geometrical scene S defined by K number of objects o_i or primitives, the technique of ray casting is performed using a uniform spatial subdivision acceleration scheme. Being C the set of subdivision cells c_i , the general expression for the time cost to trace N rays into the scene, as is showed in [24] is

$$T_S = T_b + N \times T_t \quad (24)$$

where T_b is the time to build the acceleration structure, N is the number of rays and T_t is the time to trace one ray into the structure, or *cost function*.

The cost function can be defined as

$$T_t = T_0 + \sum_{i=1}^{N_C} p(c_i)T_s + \sum_{i=1}^{N_C} p(c_i)\vartheta(c_i)T_i \quad (25)$$

On the right term, T_0 is the time to initialize a ray, N_C is the number of cells that are visited in average to find the intersection, $\sum_{i=1}^{N_C} p(c_i)T_s$ is the cost of stepping through cells and $\sum_{i=1}^{N_C} p(c_i)\vartheta(c_i)T_i$ is the cost of explicit intersection test.

The average number of cells N_C needed to traverse to find an intersection point, depends on several parameters of the

structure. As an example, [24] shows that in average, time T_i can be estimated as

$$E \langle T_i \rangle = T_0 + \left[1 + \frac{3L_r N_C^{\frac{1}{3}}}{2V(\nu)^{\frac{1}{3}}} \right] \left[T_s + \frac{\kappa}{N_C} T_i \left[K + \frac{3D(\theta)(N_C)^{\frac{1}{3}}}{V^{\frac{1}{3}}(\nu)} + \frac{3SA(\theta)(N_C)^{\frac{2}{3}}}{4V^{\frac{2}{3}}(\nu)} + \frac{3V(\theta)(N_C)}{4V(\nu)} \right] \right] \quad (26)$$

Table 2 Symbols and terms of ray casting

N	Number of rays
T_b	Preprocessing time to build efficiency structure
T_0	Time to initialize a ray
T_s	Time used for ray to step from one cell to another
T_i	Time used to check if a ray intersects an object

Table 3 Mathematical symbols

$E \langle x \rangle$	Expected value of x
$p(c_i)$	Probability that a random ray will strike cell i
$\Theta(c_i)$	Occupancy: number of objects which intersect cell i

Table 4 Symbols and terms of geometric description of a scene

S	The set of all objects or scene
K	The number of objects in S
$V(A)$	The volume of a point set A
ν	The bounding volume of a point set A
C	Set of subdivision cells
N_C	The number of cells in C
e_D	Total diameter
e_{SA}	Total surface area
e_V	Total volume
e_L	Average ray length scale
$D(A)$	Average length of A 's projection onto a random plane
$L(A)$	Length of a 1-D point set A
$SA(A)$	Surface area of a 3-D point set A
c_i	Cell i in C
o_i	Object i in O
L	Side length of a cubical cell
L_r	Average length of a ray
κ	Collision factor between the two distributions of objects and rays

with all parameters depending on features of the speedup structure.

The full set of parameters involved in this calculations are detailed in Tables 2, 3 and 4.

Acknowledgements This work has been partly financed by Diputación General de Aragón (Interactive Systems, Adaptivity, Autonomy and Cognition, cod. T33-20D).

Declarations

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Born M, Wolf E. Principles of optics. 7th ed. Cambridge: Cambridge University Press; 1999.
2. Chen JS, Chu D. Realization of real-time interactive 3d image holographic display (invited). Appl Opt. 2016;55(3):A127–34.
3. Dutre P, Bala K, Bekaert P, Shirley P. Advanced global illumination. Wellesley, MA: AK Peters Ltd. CRC Press; 2006.
4. Gabor D. A new microscopic principle. Nature. 1948;161(4098):777–8.
5. Goodman JW. Introduction to Fourier optics. Englewood, CO: W. H. Freeman & Co Ltd; 2005. ISBN: 9780974707723.
6. Hahn J, Kim H, Lim Y, Park G, Lee B. Wide viewing angle dynamic holographic stereogram with a curved array of spatial light modulators. Opt Express. 2008;16(16):12372–86.
7. Hecht E. Optics. Boston: Pearson education: Addison-Wesley; 2002.
8. Herzig H. Micro-optics: elements, systems and applications. Abingdon: Taylor & Francis; 1997.
9. Jensen HW. Realistic image synthesis using photon mapping. Natick: A. K. Peters Ltd; 2001.
10. Jiang H, Zhao J, Di J. Digital color holographic recording and reconstruction using synthetic aperture and multiple reference waves. Opt Commun. 2012;285(13–14):3046–9. <https://doi.org/10.1016/j.optcom.2012.02.076>.

11. Kang SB. Survey of image-based rendering techniques. *Proc SPIE*. 1998;3641:2–16. <https://doi.org/10.1117/12.333774>.
12. Kelly DP, Hennelly BM, Pandey N, Naughton TJ, Rhodes WT. Resolution limits in practical digital holographic systems. *Opt Eng*. 2009;48(9):095801–095801 13.
13. Kelly DP, Monaghan DS, Pandey N, et al. Digital holographic capture and optoelectronic reconstruction for 3d displays. *Int J Digit Multimed Broadcast*. 2010. <https://doi.org/10.1155/2010/759323>.
14. Kogelnik H. Coupled wave theory for thick hologram gratings. *Bell Syst Tech J*. 1969;48(9):2909–47.
15. Kurihara T, Takaki Y. Shading of a computer-generated hologram by zone plate modulation. *Opt Express*. 2012;20(4):3529–40.
16. Lafortune EP, Willems YD. Reducing the number of shadow rays in bidirectional path tracing. *Proc Winter School Comput Graph Vis*. 1995;95:384–92.
17. Lengyel J. The convergence of graphics and vision. *Computer*. 1998;31(7):46–53. <https://doi.org/10.1109/2.689676>.
18. Li X, Ren H, Chen X, Liu J, Li Q, Li C, Xue G, Jia J, Cao L, Sahu A, Hu B, Wang Y, Jin G, Gu M. Athermally photoreduced graphene oxides for three-dimensional holographic images. *Nat Commun*. 2015;6:6984. <https://doi.org/10.1038/ncomms7984>.
19. Noda H, Namioka T, Seya M. Ray tracing through holographic gratings. *J Opt Soc Am*. 1974;64(8):1037–42.
20. Pharr M, Humphreys G. Physically based rendering: from theory to implementation. Burlington: Morgan Kaufmann; 2004.
21. Shi CX, Shin CW, Jeong JR, Kim N. Acceleration method for computer generated hologram using random-sampled wavefront recording plane. *Imaging Appl Opt*. 2016;20(2):6–8.
22. Shum HY, Chan SC, Kang SB. Image-based rendering. US: Springer; 2007.
23. Slinger C, Cameron C, Stanley M. Computer-generated holography as a generic display technology. *Computer*. 2005;38(8):46–53.
24. Walter B, Shirley P. Cost analysis of a ray tracing algorithm. 1997. <http://www.graphics.cornell.edu/~bjw/mca.pdf>.
25. Zhang H, Collings N, Chen J, Crossland B, Chu D, Xie J. Full parallax three-dimensional display with occlusion effect using computer generated hologram. *Opt Eng*. 2011;50(7):074003–074003 5.
26. Zhang H, Tan Q, Jin G. Full parallax three-dimensional computer generated hologram with occlusion effect using ray casting technique. *J Phys Conf Ser* 415(1), 012048 (2013). <http://stacks.iop.org/1742-6596/415/i=1/a=012048>.
27. Zhao J, Jiang H, Di J. Recording and reconstruction of a color holographic image by using digital lensless Fourier transform holography. *Opt Express*. 2008;16(4):2514–9. <https://doi.org/10.1364/OE.16.002514>.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.