Západočeská universita v Plzni
Fakulta aplikovaných věd

URYCLENÍ VÝPOČTU ČÍSLICOVÉHO HOLOGRAMU

Ing. Ivo Hanák

disertační práce
k získání titulu doktor
v oboru Inženýrská informatika

Školitel: Prof. Ing. Václav Skala, CSc.
Katedra: Katedra informatiky a výpočetní techniky

Plzeň 2009

University of West Bohemia
Faculty of Applied Sciences

ACCELERATING DIGITAL HOLOGRAM GENERATION

Ing. Ivo Hanak

dissertation thesis
submitted in partial fulfilment of the requirements
for a degree of Doctor of Philosophy
in Computer Science and Engineering

Supervisor: Prof. Ing. Vaclav Skala, CSc.

Department of Computer Science and Engineering

Pilsen 2009

Declaration

I present a dissertation thesis to be reviewed and defended. The thesis was created at the end of doctoral study on Faculty of Applied sciences, University of West Bohemia. I hereby declare that I have created this work alone if not noted otherwise and that all used references are properly cited in a list at the end of this thesis.

In Pilsen, November 30th, 2009 .....

Ing. Ivo Hanak

Abstrakt

Svět kolem sebe vidíme pomocí odraženého světla. Hologram toto světlo dokáže zaznamenat a zreprodukovat, přičemž reprodukce není rozlišitelná od skutečnosti. Díky existenci numerického modelu můžeme využít této vlastnosti hologramu pro zobrazení virtuální scény.

Výpočet hologramu je však náročný proces a proto se zabýváme akcelerací výpočtu hologramu velkých velikostí. Jedním z možných přístupů je, podobně jako v počítačové grafice, snížit detail, který dokážeme zaznamenat. Předložená metoda proto převádí scénu z povrchového vyjádření na vyjádření s použitím unifikovaného elementu. Zatímco pro zpracování elementu využívá propagace úhlového spektra, pro řešení viditelnosti používá vrhání paprsků. Tedy metoda kombinuje dva odlišné přístupy k výpočtu hologramu. Ukazujeme, že tato kombinace je nejen možná, ale také poskytuje prostor pro další urychlení. Výsledek sice není absolutně nejrychlejší v obecném případě, ale ukazujeme, že i tento omezený rozsah je dostatečně volný pro velké hologramy běžných scén.

Kromě návrhu nové metody a její akcelerace se pak zabýváme akcelerací existujícího přístupu založeného na použití paprsků. Předkládáme úpravu, která umožňuje efektivní využití grafické karty či programovatelného technického vybavení.

klíčová slova: číslicová holografie, výpočet hologramu, propagace světla

Abstract

We see the real world through reflected light. The hologram is a recording of such light and it offers reproduction that is not distinguishable from the reality. Since we known a numerical approximation of the hologram, we can display virtual scene through it.

However, hologram calculation is a time extensive process. Therefore, we accelerate calculation of large holograms intended for viewing purposes. Being inspired by the computer graphics, we reduce the detail of the scene by converting it to uniform elements. The method combines two different trends of hologram generation. We apply propagation of the angular spectrum to process the elements but we also apply ray-casting to solve visibility. We show that this combination is possible and it offers opportunities for further acceleration. The resulting method is not the fastest in a general case but we show that it is satisfactory for usual scenes.

Besides the new method, we also discuss acceleration of another existing approach that uses strictly ray-casting. We present an adjustment, that can facilitate efficiently the graphical processing unit and that is compatible with programmable hardware.

keywords: digital holography, hologram generation, light propagation

Acknowledgement

I would like to thank many of those who I met during my studies. I would like to thank my supervisor, prof. Václav Skala for introducing me to the 3DTV project, which is acknowledged below, and for his support. Also, I would like to thank Martin Janda and Petr Lobaz for a smooth collaboration and many fruitful discussions, Dr. Libor Váša for his valuable help with documenting reconstructions, Dr. Pavel Zemčík and Dr. Adam Herout for their insights on application of a programmable hardware and for providing me a support while I was finishing this thesis. Thank you, prof. Levent Onural, prof. Vencezslav Kovachev, prof. Rositza Iljeva, the rest of the team at the İSYAM laboratory for allowing me to discover the secrets of holography. Thank you all of you.

This work has been supported by the project 3DTV PF6-2003-IST-2, Network of Excellence, No: 511568 and by the project Centre of Computer Graphics, National Network of Fundamental Research Centers, MSMT Czech Rep., No: LC 06008. The access to the METACentrum computing facilities provided under the research intent MSM6383917201 is acknowledged.

This work is dedicated to my creepy brain that allowed me this opportunity.

Contents

Chapter 1

Introduction

Holography is a research area that focuses on a by-product of light interaction with small obstacles. First observed by Dr. Denis Gabor, it allows both to capture light and to reproduce it subsequently. We can numerically approximate this process but it is computationally demanding. Therefore, we propose a new method that address primarily this issue. We show that this new method is fast and it can be accelerated efficiently. Also, we show that we can adjust an existing method to be compatible with hardware and thus we can reduce the computational time.

Holography assumes that light is an electro-magnetic wave that interacts with small obstacles. By appropriate application of this interaction we can modulate light and hence we can recreate light that was captured. The recording, which is being created under special light conditions, contains complete visual information that can be detected by a human visual system. As a consequence, holography is the only tool that is able to create an impression almost indistinguishable from the reality. While creating the photograph we assume a viewer and as a consequence we capture only a single view that was seen by the camera. Unlike that, while creating a hologram a viewer is excluded and therefore the hologram captures everything. This is the most important feature of holography. Usually, we refer to the recording of light as the hologram and we refer to the process of reproduction as the reconstruction.

This work intends to optimise the calculation of a hologram from a virtual scene. We calculate a structure that we can converted to a hologram using a simple calculation. For us, the hologram serves as a multi-viewer 3D display and the image is formed on a retina of a viewer. Since this is similar to the reality, we see the holography as a next step in evolution of the computer graphics.

The major issue is a long computation time caused by both a high computational complexity and a high resolution of the hologram. In the computer graphics we assume usually only one viewer but in holography every sample is a viewer. Besides that, a discrete hologram requires a sampling step that is approximately $10\times$ smaller than a sampling step of a picture displayed on a contemporary LCD. As a consequence, using a the most primitive algorithm without an acceleration means to calculate a square of 2×2 mm in an order of tens of hours. This is not acceptable and therefore we address it in this work.

All methods that calculate holograms follow a single equation and they try to implement it efficiently. Analysing the methods, we recognised three possible trends. Every trend complements each other in terms of advantages and disadvantages and it is a subject of research for a long time. However, combination of trends is not discussed often. Since two of

these three trends process a hologram in the same form, there is a high probability that we can combine them. Thus, we focus on this in this work.

We presume that we can combine two different trends that are well known together and we can reduce the computation time through that. Besides that, we draw an analogy between the LCD and the hologram. Even tough the LCD pixel is large, the viewer is not significantly disturbed by it when viewing a planar image and hence the spatial detail of the virtual scene can be reduced as well. These two ideas are the major contribution of this work. We do not strive for real-time generation of holograms, we aim to reduce the calculation time of the 2×2 mm square to the order of minutes before applying any hardware-based acceleration.

We propose a method that is efficient under given conditions. We define these conditions and we show that these condition are not restrictive. Hence, we show that we can reach lower computational times for usual scenes. We, however, do not present any contribution to the theory of light interaction. Neither we modify nor we develop a new description of light behaviour because this work should be in a scope of the computer graphics. Exploring the proposed method, we discuss a possibility of acceleration through a special hardware such as the graphical processing unit (GPU) or programmable hardware. We, however, do not experiment with them due to time constraints. Also, we do not address physical reconstruction of a hologram. When necessary, we apply an ad-hoc approach that might introduce additional noise. Thus, if we recognise the original scene by observing the reconstruction, we may threat the reconstruction as valid.

Besides the major contribution we include results of our attempts on accelerating another method. We collaborated with Martin Janda on adjustments of his method that uses ray-casting. We reorganised his algorithm such that it became compatible with GPU and programmable hardware. Doing these adjustments, we collected valuable experience about an appropriate structure of a hologram generation method and we applied this experience in the further development of our proposed method. Despite that, our proposed method differs significantly from his method.

This work assumes basic knowledge of common approaches in the computer graphics. It should be emphasised that we do not assume inherently a viewer in the scene. This is very important for comprehending of all relative positions of scene components. In this work, we begin with a brief overview of a holography background. We try to reduce detail of any information that is not vital for the proposed method. We continue with an overview of methods used for hologram generation. We try to cluster them through common features and we focus on major advantages and disadvantages. Then, we present the proposed method including its acceleration. After that we add a brief chapter on acceleration of the method proposed by Martin Janda. Finally, we conclude with a summary.

Chapter 2

Holography

Holography deals with the light in a form of a wave and with interaction of the light in an environment. This chapter contains an introduction into a part of holography that is applied in subsequent chapters. First, we introduce a mathematical model of wave that is applicable for light. We use these definitions for description of light interaction with an obstacle. At the end, we describe briefly a process of hologram recording. We use this process to numerically create hologram for optical reconstructions. Since this chapter serves only the purpose of following chapters, it does not explore the area in detail. For more details, which are out of the scope of this work, refer to [Goo05] or [BW05].

2.1 Wave

Light can be described through waves or through rays. The ray is an approximation used by the ray optics and it is appropriate for describing a light behaviour on human scale. Yet, the ray fails to express an interaction between light and microscopic obstacles. On the other hand, while the wave used by the wave optics is unnecessarily accurate for larger scales, it allows both a simple and an accurate description of interaction with microscopic obstacles. Since holography explores the interaction of light on the microscopic scale, it relays on the wave optics.

The light is an electro-magnetic disturbance that can be modelled by the Maxwell equations [Goo05]. The Maxwell equations relates magnetic and electric field vectors. If we assume certain attributes of the environment, both the electric field and the magnetic field follows the same scalar wave equation

$$\nabla^2 u(\mathbf{p}, t) - \frac{n^2}{c^2} \frac{\partial^2 u(\mathbf{p}, t)}{\partial t^2} = 0, \tag{2.1}$$

where $u(\mathbf{p}, t)$ is the disturbance at the position $\mathbf{p}$ and the time t examined in a medium with refractive index n.[1] The disturbance travels through the medium at a speed of c/n, where c is a speed of the light. In the further text, let us denote the disturbance as a wave.

The optical frequency ω or a wavelength $\lambda = \frac{c}{n\omega}$ is an attribute of a wave. Since light can be decomposed to individual wavelengths, we shall consider only a monochromatic wave

[1] The expression Eq. (2.1) is valid if the wave travels in dialectic medium that is a linear system, properties of medium does not depend on polarisation (an isotropic medium), permittivity is constant (a homogeneous medium), permittivity (a nondispersive medium), and magnetic permeability equals to vacuum permeability (nonmagnetic medium).

if not noted otherwise. The monochromatic wave is

$$\begin{aligned} u(\mathbf{p},t) &= a(\mathbf{p})\cos[\phi(\mathbf{p})-2\pi\omega t], \\ &= \Re\{u(\mathbf{p})\exp(-\mathrm{j}2\pi\omega t)\}, \end{aligned} \tag{2.2}$$

where $\Re\{\}$ denotes real part of a complex number, $a(\mathbf{p})$ is an amplitude and $\phi(\mathbf{p})$ is a phase of the wave at the position $\mathbf{p}$ and

$$u(\mathbf{p}) = a(\mathbf{p})\exp[\mathrm{j}\phi(\mathbf{p})] \tag{2.3}$$

is known **the complex amplitude**. Since we examine only monochromatic waves, ωt is constant for all waves. Furthermore, we can assume that the space contains sources that emit the same disturbance all the time and they did it a long time before we began to examine the space. As a consequence, the complex amplitude Eq. (2.3) serves as an adequate description of the wave and thus it is used for this purpose in the following text. We denote a volume with known complex amplitudes $u(\mathbf{p})$ as **the optical field**.

Since Eq. (2.2) describes the wave, it has to satisfy Eq. (2.1). A substituting Eq. (2.2) in Eq. (2.1) yields a time-independent equation

$$(\nabla^2 + k^2)u(\mathbf{p}) = 0, \tag{2.4}$$

where $k = \frac{2\pi}{\lambda}$ is known as **the wavenumber**. The equation Eq. (2.4) is known as **the Helmholtz equation** and a complex amplitude of a wave has to satisfy the condition in order to describe a valid disturbance.

The optical field fills the whole space and every disturbance is spread in the whole field, i.e., the disturbance is distributed into the whole volume. Thus, we can reconstruct partial information about the source just by knowing a subset of the field, e.g., using known field values at a surface, we can estimate field values outside the surface. Furthermore, since the time is omitted, a subset of a volume contains waves emitted at the different time. Hence, we refer to the process of estimating values as **a wave propagation**.

Another important feature of the light is its optical power. Actually, this is the only attribute of the field that we can detect directly using our eyes or a detector such as CCD. **The optical intensity** is an energy that crosses a unit area perpendicular to the energy flow during a unit of a time. If the time period is short enough [Har96], we can approximate the intensity I of the wave $u(\mathbf{p})$ with

$$I = u(\mathbf{p})u^*(\mathbf{p}) = |u(\mathbf{p})|^2. \tag{2.5}$$

Since we omit the time in our considerations, we use expression Eq. (2.5) to calculate intensity of an optical field at the given location.

2.1.1 Interference and Coherence

Interaction of two or more waves is known as **interference**. Intensity of the interference plays an important role in the holography because it is employed during hologram recording process. Since the scalar wave equation Eq. (2.1) assumes a linear medium and all waves are monochromatic, interference of two waves $u_1(\mathbf{p})$ and $u_2(\mathbf{p})$ yields a wave $u(\mathbf{p}) = u_1(\mathbf{p}) + u_2(\mathbf{p})$. Following [Har96], intensity of such a wave at the location $\mathbf{p}$ is

$$\begin{aligned} I(\mathbf{p}) &= |u_1(\mathbf{p}) + u_2(\mathbf{p})|^2, \\ &= |u_1(\mathbf{p})|^2 + |u_2(\mathbf{p})|^2 + \frac{1}{2}u_1(\mathbf{p})u_2^*(\mathbf{p}) + \frac{1}{2}u_1^*(\mathbf{p})u_2(\mathbf{p}), \\ &= I_1(\mathbf{p}) + I_2(\mathbf{p}) + 2\left[I_1(\mathbf{p})I_2(\mathbf{p})\right]^{1/2}\cos\left[\varphi_1(\mathbf{p}) - \varphi_2(\mathbf{p})\right], \end{aligned} \tag{2.6}$$

where $\varphi_1(\mathbf{p})$ and $\varphi_2(\mathbf{p})$ are phases of the waves $u_1(\mathbf{p})$ and $u_2(\mathbf{p})$ respectively. On a surface, the intensity from Eq. (2.6) forms a structure that is known as **an interference pattern** or, as one may call it, a hologram.

Coherence of light is also essential for holography because it influences the visibility of the interference pattern that can be detected by a recording device [Har96]. Visibility of the interference pattern from Eq. (2.6) at the given position $\mathbf{p}$ is

$$V = \frac{2(I_1 I_2)^{1/2}}{I_1 + I_2} \cos(\psi), \tag{2.7}$$

where I_1 and I_2 are intensities of the first wave and the second wave respectively. The angle ψ is an angle between electrical vectors of both waves, i.e., it is a polarisation.[2] For clarity of further that, let us assume that electrical vectors of all considered waves are parallel. Notice that the expression Eq. (2.7) says that only the third term of expression Eq. (2.6) is desirable, the rest acts as a background noise. This fact is exploited later by a technique known as the bipolar intensity.

If light is incoherent, the interference pattern is not visible enough. In an ideal case, coherent light is monochromatic light emitted by a point light source (PLS) for an infinite time period. In reality, however, a light source is quasi-monochromatic and it has a finite size.[3] Coherence expresses how much is a given light source close to the ideal one. Coherence can be expressed analytically and it serves as a multiplicative coefficient in Eq. (2.7). Nevertheless, for purpose of numerical simulations, coherence is not essential and therefore we omit any further details. For more information on the subject, refer to [Har96, BW05].

2.1.2 Elementary Waves

In the previous sections, we discuss attributes of both waves and sources. In this section we focus on a form of waves. The Helmholtz equation Eq. (2.4) is a condition that has to be satisfied by a function $u(\mathbf{p})$ that describes a wave. Solutions that fulfil the condition are known as elementary waves. In this work, we used the planar wave and the spherical wave. Any optical field can be expressed as a weighted superposition of given elementary waves. This feature is exploited by the wave propagation.

Elementary waves differ by the shape of their wavefront. **The wavefront** is an iso-surface defined by the phase $\varphi(\mathbf{p}) = 0$ of the wave $u(\mathbf{p})$. The **planar wave** is a wave whose wavefronts are infinite planes [Goo05, Gra03, Kra04]. The optical field of the planar wave is

$$\begin{aligned} u(\mathbf{p}) &= a \exp(\mathrm{j}\varphi) \exp(\mathrm{j}\mathbf{k} \cdot \mathbf{p}), \\ &= a \exp(\mathrm{j}\phi) \exp[\mathrm{j}(x_\mathbf{k} x_\mathbf{p} + y_\mathbf{k} y_\mathbf{p} + z_\mathbf{k} z_\mathbf{p})], \end{aligned} \tag{2.8}$$

where a is an amplitude and ϕ is a phase of the planar wave at the origin, i.e., at $(0, 0, 0)$. We denote the vector $\mathbf{k} = (x_\mathbf{k}, y_\mathbf{k}, z_\mathbf{k})$ as **the wavevector**. If the wave is emitted by XY-plane, the wavevector is a direction of wave propagation. The length of the wavevector is the

[2] Notice that visibility drops to zero if $\psi = \pi/2$. Hence, waves of polarised light do not create a visible interference pattern if polarisation directions are perpendicular to each other.

[3] Quasi-monochromatic light contains a narrow range of optical frequencies instead of single optical frequency ω. Thus, the light emitted by the source consist of various wavelengths and wavefronts of various shapes. The wavefront is an iso-surface defined by the phase $\varphi(\mathbf{p}) = 0$ of the wave $u(\mathbf{p})$. An interference pattern of such light consist of multiple overlapping patterns and thus it might become undistinguishable from the background

wavenumber k, i.e., $|\mathbf{k}| = k$. Intensity on the planar wave at any location is constant and it equals to $I = |a|^2$. Planar waves are important for propagation of waves that is discussed in Sec. 2.2.2.

The **spherical wave** is a wave generated by a point light source (PLS). Wavefronts of the spherical wave are concentric spherical surfaces centred at the location of PLS. The optical field emitted by PLS at origin $(0, 0, 0)$ is

$$u(\mathbf{p}) = \frac{a \exp(\mathrm{j}\varphi)}{|\mathbf{p}|} \exp(\mathrm{j}k|\mathbf{p}|), \tag{2.9}$$

where $p = |\mathbf{p}|$ is a distance from PLS defined by the amplitude a and by the phase φ. The fraction $\frac{a}{|\mathbf{p}| \exp(\mathrm{j}\varphi)}$ prevents growth of wavefront energy because increasing the distance increases the area of the wavefront. Hence, the amplitude of an unit of area has to decrease in order to keep overall energy constant. Since the spherical surface grows quadratically and the intensity is $I = |a|^2$, the amplitude has to be divided just by the distance.

PLS generates the spherical wave. If we examine only a window that has finite extent, we can observe either a planar wave far away from the source as depicted in Fig. 2.1. In other words, if the distance is large enough in comparison to extents in both the X-axis and the Y-axis, we can approximate the spherical wave by the planar one or by the paraboloidal one. This is exploited by the Fraunhofer approximation and by the Fresnel approximation discussed in Sec. 2.2.2.

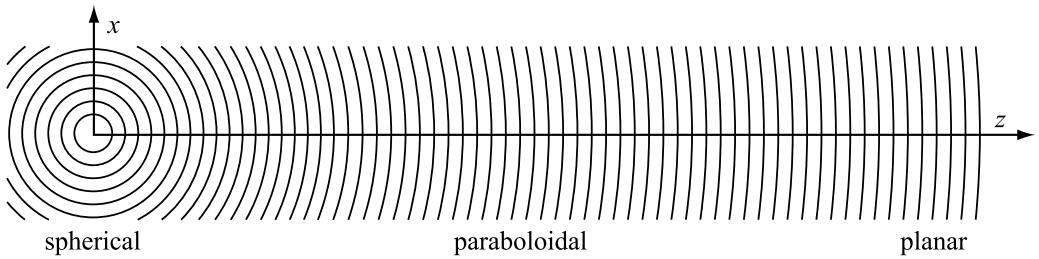


Figure 2.1: Relation between the spherical wave and the planar wave as seen though a small window.

2.2 Diffraction and Propagation

Diffraction and propagation describe the same physical phenomenon of light interaction with an environment on a microscopic level. Interaction of the light with a free space is denoted as a propagation while interaction with obstacles is denoted as diffraction. The diffraction is exploited by hologram reconstruction and both interactions are exploited by hologram recording. For that reason, this section contains formulations that are fundamental for following chapters.

2.2.1 Diffraction

Diffraction is a result of an interaction between light and obstacles at microscopic levels. The interaction is a complicated process but in a case of a screen with an opening we can solve it analytically. Even though this is a special case, it is widely used for approximating the interaction in other cases as well.

Diffraction can be described by two formulations that are widely applied in numerical simulations: the Kirchhoff formulation with its Fresnel-Kirchhoff diffraction formula and the Rayleigh-Sommerfeld formulation with its Rayleigh-Sommerfeld diffraction formulae [Goo05, BW05, LBL02]. Both formulations are based the scalar wave theory and both formulations describe interaction of light with an aperture. The aperture is an opening in an infinite screen. The formulations allow to calculate an optical field starting from a distance of a few wavelengths behind the screen from known optical field in front of the aperture. Since a diffraction formula is essential for almost every method, we shall give a brief overview of its derivation. Readers that are not interested in finer details might skip this section.

The first step toward the Kirchhoff formulation is to express a field value at a point in the volume using a field value at a point on enclosing surface. If there are two continuous functions $u(\mathbf{p})$ and $g(\mathbf{p})$ in the volume V, relation between the boundary S' and the volume V follows the Green's theorem and it is

$$\iiint_V (u\nabla^2 g - g\nabla^2 u)\,\mathrm{d}V = \iint_S' \left(u\frac{\partial g}{\partial \mathbf{n}} - g\frac{\partial u}{\partial \mathbf{n}}\right)\,\mathrm{d}S., \tag{2.10}$$

Let us assume that the function $g(\mathbf{p}_1)$ is a wave generated by PLS of the amplitude $a = 1$ and the phase $\varphi = 0$ located at $\mathbf{p}_0$, i.e., $g(\mathbf{p}_1) = \frac{\exp(\mathrm{j}kr_{01})}{r_{01}}$. As a consequence, $S' = S_\epsilon + S$ as depicted in Fig. 2.2. This prevents discontinuity from the expression Eq. (2.9). Since both functions u and g represent a wave, they have to satisfy the Helmholtz equation Eq. (2.4) at the same time. Thus, we can apply Eq. (2.10) to yield

$$-\iint_{S_\epsilon} \left(u\frac{\partial g}{\partial \mathbf{n}} - g\frac{\partial u}{\partial \mathbf{n}}\right)\,\mathrm{d}S = \iint_S \left(u\frac{\partial g}{\partial \mathbf{n}} - g\frac{\partial u}{\partial \mathbf{n}}\right)\,\mathrm{d}S \tag{2.11}$$

Since the left side of Eq. (2.11) equals to $4\pi u(\mathbf{p}_0)$ [Goo05], the expression Eq. (2.11) becomes

$$u(\mathbf{p}_0) = \frac{1}{4\pi}\iint_S \left(u\frac{\partial g}{\partial \mathbf{n}} - g\frac{\partial u}{\partial \mathbf{n}}\right)\,\mathrm{d}S. \tag{2.12}$$

The expression Eq. (2.12) is known as **the Helmholtz-Kirchhoff integral theorem**.

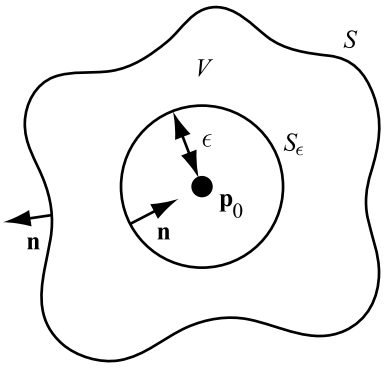


Figure 2.2: A configuration assumed by the Helmholtz-Kirchhoff integral theorem. [Goo05]

The next step adds a screen and solves the optical field behind it. Adding screen splits the surface S into the part S_1 that is located immediately after the screen and the hemispherical cap S_2 that is depicted in Fig. 2.3. The contribution of each part to the final solution is solved independently following Eq. (2.12). Increasing the radius r, the contribution of the hemispherical cap S_2 becomes neglectable [Goo05].[4] Hence, we can reduce the integration domain of Fig. 2.2 to the planar part S_1.

[4]This is a consequence of the Sommerfeld radiation condition $\lim_{r\to\infty} r\left(\frac{\partial u}{\partial \mathbf{n}} - \mathrm{j}ku\right) = 0$ that is satisfied by the diverging spherical wave and that guarantees only outgoing waves on the surface S_2.

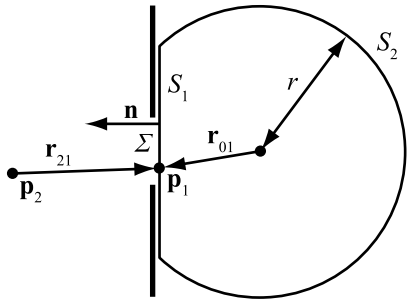


Figure 2.3: The screen and the volume S used for derivation of the Kirchhoff formulation. [Goo05]

For purpose of further simplification, the function u is restricted by **the Kirchhoff boundary condition** that limits the function u. The field values u are not influenced by the screen across the surface Σ in Fig. 2.3. Outside the surface Σ, both the function u and its derivation $\frac{\partial u}{\partial \mathbf{n}}$ are zero. In a general case such a function u does not exist but if the aperture is much larger than the wavelength, it serves as an acceptable approximation [Goo05]. The boundary condition reduces the integration domain in Eq. (2.12) to the surface Σ at the opening.

The final step towards the Fresnel-Kirchhoff diffraction formula is to express the optical field on the surface Σ. The function g describes an optical field generated by PLS at $\mathbf{p}_0$ on the surface Σ, i.e., $g(\mathbf{p}_1) = \frac{1}{r_{01}} \exp(\mathrm{j}kr_{01})$ where $r_{01} = |\mathbf{r}_{01}|$. Since the wavenumber $k \gg \frac{1}{r_{01}}$, derivate of the function g becomes

$$\frac{\partial g(\mathbf{p}_1)}{\partial \mathbf{n}} = \left(\mathrm{j}k - \frac{1}{r_{01}}\right) \frac{\exp(\mathrm{j}kr_{01})}{r_{01}} \hat{\mathbf{n}} \cdot \hat{\mathbf{r}}_{01}, \tag{2.13}$$

$$\approx \mathrm{j}k \frac{\exp(\mathrm{j}kr_{01})}{r_{01}} \hat{\mathbf{n}} \cdot \hat{\mathbf{r}}_{01}. \tag{2.14}$$

Furthermore, let us assume that PLS located at $\mathbf{p}_2$ in front of the aperture as depicted in Fig. 2.3. An optical field value at the aperture is $u(\mathbf{p}_1) = a_2 \exp(\mathrm{j}\varphi_2) \frac{\exp(\mathrm{j}kr_{21})}{r_{21}}$ where $r_{21} = |\mathbf{r}_{21}|$. Since $k \gg \frac{1}{r_{01}}$, the derivate $\frac{\partial u(\mathbf{p}_2)}{\partial \mathbf{n}}$ is similar to Eq. (2.14). Combining it with Eq. (2.12), it yields

$$u(\mathbf{p}_0) = \frac{a_2 \exp(\mathrm{j}\varphi_2)}{\mathrm{j}\lambda} \iint_\Sigma \frac{\exp[\mathrm{j}k(r_{21} + r_{01})]}{r_{21} r_{01}} \frac{\hat{\mathbf{n}} \cdot \hat{\mathbf{r}}_{01} - \hat{\mathbf{n}} \cdot \hat{\mathbf{r}}_{21}}{2} \,\mathrm{d}s. \tag{2.15}$$

The expression Eq. (2.15) is known as **the Fresnel-Kirchhoff diffraction formula**. Notice that from the viewpoint of the source and the observation point, the Fresnel-Kirchhoff diffraction formula is symmetrical. PLS located at $\mathbf{p}_2$ yields the same result at $\mathbf{p}_0$ as the same PLS located at $\mathbf{p}_0$ and observed from $\mathbf{p}_2$. This effect is known as **the reciprocity theorem of Helmholtz**. The formula in Eq. (2.15) is valid only for a single PLS in front of the aperture. We can expand it towards multiple PLS in front of the aperture by summing results of individual PLS.

The Kirchhoff formulation contains internal inconsistencies due to Kirchhoff boundary condition. The condition presumes that both the function and its normal derivate are zero behind the screen. This implies that the field behind the aperture should be zero as well [Goo05]. However, it is in contradiction to physical experiments and even to results calculated by Eq. (2.15). Nevertheless, if the opening is much large than the wavelength and the

observation point is further away from the aperture at same time, the results corresponds to physical experiments and thus the Kirchhoff formulation is widely used in practice.

The inconsistency of the Kirchhoff formulation is addressed by the Rayleigh-Sommerfeld formulation. The formulation eliminates the inconsistency by redefining of the function g such that either the function g or is derivate $\frac{\partial g}{\partial \mathbf{n}}$ is zero over surface S_1. This removes necessity to enforce zero on both the function u and its derivate $\frac{\partial u}{\partial \mathbf{n}}$ at the same time while restricting the integration domain to the planar part Σ.

The function g that fulfils the new definition is constructed as an interference of two PLS on a plane: PLS located at $\mathbf{p}_0$ and a mirrored PLS located at $\mathbf{p}'_0$ as depicted in Fig. 2.4.[5] The amplitude of both sources is one, the phase may differ. There are two setups that allow fulfilling the definition: either both sources emits in phase or in an opposite phase, i.e., their phases differs by a half of a period.[6] Each setup forms one solution.

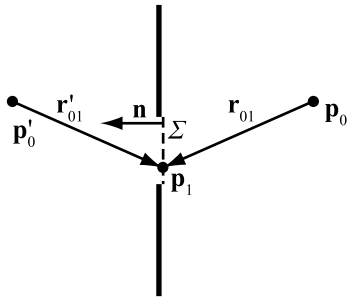


Figure 2.4: A screen and a symmetrical PLS used by a definition of the function g. The setup is used to derive the Rayleigh-Sommerfeld formulation. [Goo05]

Now, let us express the Rayleigh-Sommerfeld solution. Similar to the Kirchhoff solution, we define one PLS in front of the aperture. The PLS generates a field $u(\mathbf{p}_1) = a_2 \exp(\mathrm{j}\varphi_2)\frac{\exp(\mathrm{j}kr_{21})}{r_{21}}$ at the aperture Σ. Using this, **the Rayleigh-Sommerfeld diffraction formula** is

$$\begin{aligned} u_{\mathrm{I}}(\mathbf{p}_0) &= \frac{1}{\mathrm{j}\lambda}\iint_{\Sigma} u(\mathbf{p}_1)\frac{\exp(\mathrm{j}kr_{01})}{r_{01}}\,\hat{\mathbf{n}}\cdot\hat{\mathbf{r}}_{01}\,\mathrm{d}s, \\ &= \frac{a_2\exp(\mathrm{j}\varphi_2)}{\mathrm{j}\lambda}\iint_{\Sigma}\frac{\exp[\mathrm{j}k(r_{21}+r_{01})]}{r_{21}r_{01}}\,\hat{\mathbf{n}}\cdot\hat{\mathbf{r}}_{01}\,\mathrm{d}s \end{aligned} \tag{2.16}$$

and the second solution is

$$u_{\mathrm{II}}(\mathbf{p}_0) = -\frac{a_2\exp(\mathrm{j}\varphi_2)}{\mathrm{j}\lambda}\iint_{\Sigma}\frac{\exp[\mathrm{j}k(r_{21}+r_{01})]}{r_{21}r_{01}}\,\hat{\mathbf{n}}\cdot\hat{\mathbf{r}}_{21}\,\mathrm{d}s. \tag{2.17}$$

The first and second RayLeigh-Sommerfeld solution resembles the Kirchhoff-Fresnel diffraction formula Eq. (2.15). The only difference are both the sign and the last cosine-based component. It can be shown that the Kirchhoff formulation is an average of both the first and the second Rayleigh-Sommerfeld formulation [Goo05]. Both the Kirchhoff and the Rayleigh-Sommerfeld formulations are almost identical for small angles and larger distance but they differ at small distances from the aperture. Since we use the Rayleigh-Sommerfeld formulation most of the time, more detailed discussion is our of a scope of this work. For more details, refer to [Goo05].

[5] This is rather an important difference from the Kirchhoff formulation. The Rayleigh-Sommerfeld formulation assumes that the screen is a plane while Kirchhoff does not.

[6] Actually, the latter causes that the field generated by the second PLS at $\mathbf{p}'_0$ is subtracted from the field generated by PLS at $\mathbf{p}_0$.

2.2.2 Propagation

While the diffraction describes an interaction of light with obstacles, propagation describes interaction of light with a free space. Propagation allow us to calculate an optical field values at a part of the space from known optical field values.[7] The propagation is exploited for calculating of the optical field generated by a scene and for reconstruction purposes. Thus, it is appropriate to give a brief overview of the phenomena.

For a single point source described in Sec. 2.1.2, the propagation equals to shift of the phase according to a shift of the observation location. The sign of the change can be arbitrary but it has to be consistent for all computations. For purpose of clarity, this work employs the principle described in [Goo05]. The time dependent component $\exp(-\mathrm{j}\omega t)$ of the wavefunction Eq. (2.2) causes the phase to decrease as time advances. Mentioned in Sec. 2.1, we assume sources. As a consequence, the waves emitted later are closer to the source. Thus, moving further from the source increases the phase, i.e., the sign of the change is positive.

Propagation of the wave follows the **Huygens-Fresnel principle** [Goo05]. It is an improved Huygens principle. The original one defines the new wavefront as an envelope of spherical wave sources generated on the previous primary wavefronts. Such definition, however, causes back-waves that are physically unjustified. Nevertheless, the Huygens-Fresnel principle states that every point on the primary wavefront is a secondary PLS, which emits spherical waves, and the result is superposition of these secondary PLS [Har05, Wei]. The Huygens-Fresnel principle is confirmed by both the Kirchhoff diffraction formula and the Rayleigh-Sommerfeld diffraction formulae. As mentioned in [LBL02], we can rewrite the Rayleigh-Sommerfeld diffraction formula defined from the expression Eq. (2.16) as

$$u(\mathbf{p}_0) = \frac{1}{\mathrm{j}\lambda} \iint_{\Sigma} u'(\mathbf{p}_1) \frac{\exp(\mathrm{j}kr_{01})}{r_{01}} \cos\theta \,\mathrm{d}s, \tag{2.18}$$

where $u'(\mathbf{p}_1) = a(\mathbf{p}_1)\exp[\mathrm{j}\varphi(\mathbf{p}_1)]$ represents PLS located at $\mathbf{p}_1$ within the aperture Σ, $r_{01} = |\mathbf{r}_{01}|$, and $\cos\theta = \hat{\mathbf{n}} \cdot \hat{\mathbf{r}}_{01}$. Since the expression Eq. (2.18) is based on the Rayleigh-Sommerfeld solution, all considered locations $\mathbf{p}_1$ have to be on a single plane. For purpose of simplification, the plane is the plane $\kappa_\xi : z = \xi$, which is parallel to the plane $\kappa : z = 0$, and thus $\cos\theta = z_{\mathbf{r}_{01}}/r_{01}$.

Furthermore, we can rewrite the expression Eq. (2.18) as a convolution

$$u(\mathbf{p_0}) = \frac{1}{\mathrm{j}\lambda} \iint_{\Sigma} u'(\mathbf{p}_1) h(\mathbf{p}_0, \mathbf{p}_1) \,\mathrm{d}s, \tag{2.19}$$

where h is an impulse response function $h(\mathbf{p_0}, \mathbf{p_1}) = \frac{\exp(\mathrm{j}kr_{01})}{r_{01}} \frac{z_{\mathbf{r}_{01}}}{r_{01}}$. The form in Eq. (2.19) is appropriate for calculating if all samples $u(\mathbf{p}_0)$ are located the plane $\kappa : z = 0$. In such case, Eq. (2.19) can be solved by applying the Fourier transform [Goo05].

2.2.3 The Angular Spectrum

According to Sec. 2.2.2, the optical field can be calculated as a superposition of spherical waves. Besides the spherical wave that is an elementary wave, a planar wave can be used for

[7] Actually, the term "propagation" might be misleading. As it was stated at a beginning of Sec. 2.1, we omit the time. As a consequence, the configuration of disturbances in the optical field is stable and it fills the whole free space. Under such circumstances, the propagation allow us to calculate the missing information without any obstacles and any additional sources.

similar purpose. Since such a use is exploited by numerical reconstructions and it is used by numerous approaches, we discuss it in this section.

As it is shown in [EO06, Onu07], the optical field can be expressed as a superposition of planar waves following

$$u(\mathbf{p}) = \iint_{-\infty}^{+\infty} \mathcal{U}(x_\mathbf{k}, y_\mathbf{k}) \exp(\mathrm{j}\mathbf{k}\cdot\mathbf{p})\,\mathrm{d}x_\mathbf{k}\,\mathrm{d}y_\mathbf{k}, \tag{2.20}$$

where $\mathbf{k} = (x_\mathbf{k}, y_\mathbf{k}, z_\mathbf{k})$ is the wavevector, $z_\mathbf{k} = [k^2 - x_\mathbf{k}^2 - y_\mathbf{k}^2]^{1/2}$, and $\mathcal{U}(x_\mathbf{k}, y_\mathbf{k})$ is known as **the angular spectrum** [Lal68, Goo05, EO06]. The meaning of $\mathcal{U}$ can be derived from a special case when all samples $u(\mathbf{p})$ are located on a plane $\kappa : z = 0$. In that case, the expression Eq. (2.20) becomes

$$\begin{aligned} u(\mathbf{p}) &= \iint_{-\infty}^{+\infty} \mathcal{U}(x_\mathbf{k}, y_\mathbf{k}) \exp[\mathrm{j}(x_\mathbf{k}x_\mathbf{p} + y_\mathbf{k}y_\mathbf{p})]\,\mathrm{d}x_\mathbf{k}\,\mathrm{d}y_\mathbf{k}, \\ &= 4\pi^2 \iint_{-\infty}^{+\infty} \mathcal{U}(2\pi x_f, 2\pi y_f) \exp[\mathrm{j}2\pi(x_f x_\mathbf{p} + y_f y_\mathbf{p})]\,\mathrm{d}x_f\,\mathrm{d}y_f. \end{aligned} \tag{2.21}$$

The equation Eq. (2.21) resembles inverse Fourier transform and thus $\mathcal{U}$ is proportional to Fourier transform of the optical field U at plane κ.

According to Eq. (2.20), every frequency in the angular spectrum corresponds to a planar wave propagating in a given direction. Relation between the direction of propagation and the frequency can be seen from a diffraction of light on a cosine grating [Goo05]. For purpose of explanation let us now consider only the 2D case.

In a 2D space, the cosine grating is spatially limited amplitude-modulating structure on a line $\kappa' : z = 0$ defined by a attenuation function

$$t_c(x) = \left[\frac{1}{2} + \frac{m}{2}\cos(2\pi f_c x)\right] \mathrm{rect}\left(\frac{x}{2w}\right) \tag{2.22}$$

depicted in Fig. 2.5(a). If a planar wave that propagates in a direction perpendicular to the line κ' hits the grating, it is diffracted. Since the planar wave is modulated by t_c, the angular spectrum of optical field immediately behind the grating is a Fourier transform of the expression Eq. (2.22), i.e.,

$$\mathcal{U}(f_x) = \left[\frac{1}{2}\delta(f_x) + \frac{m}{2}\delta(f_x + f_c) + \frac{m}{2}\delta(f_x - f_c)\right] \star \mathrm{sinc}(2wf_x), \tag{2.23}$$

where $\star$ denotes convolution. As it is shown in [Goo05]. the intensity $|\mathcal{U}|^2$ is proportional to an intensity of the optical field generated by the grating at a line $\kappa'_{z'} : z = z'$ where z' is large. The intensity at distance z' forms a pattern depicted in Fig. 2.5(b).

The central peak from Fig. 2.5(b) is original wave influenced by sides of the grating, side peaks are the wave diffracted by the cosine function from Eq. (2.22) and influenced by the sides of the grating at the same time. Individual side peaks are interpreted as original wave deflected from the Z-axis. The deflection is described by relation known as the **diffraction condition**:

$$\sin\theta = \frac{\lambda}{\Lambda}, \tag{2.24}$$

where $\Lambda = \frac{1}{f_c}$ is a spatial frequency and θ is an angle of deflection away from the Z-axis. If we use a grating that modules both the amplitude and the phase instead of Eq. (2.22), both

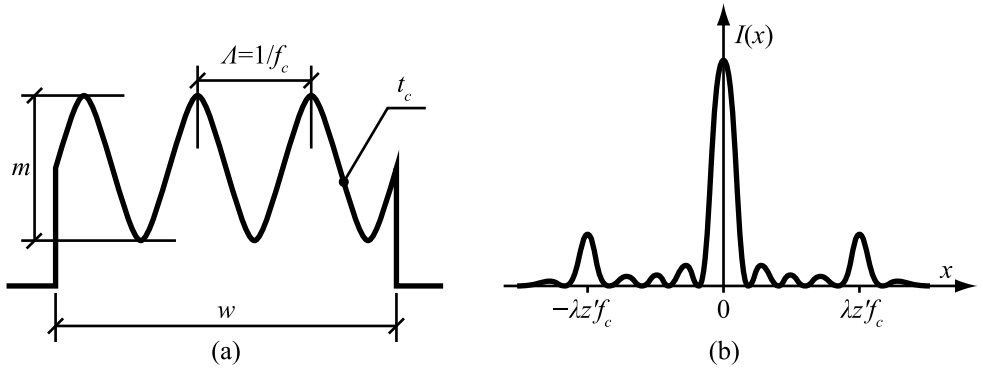


Figure 2.5: (a) The amplitude-modulating function t_c of a cosine grating and (b) an effect of the cosine grating on a planar wave observed from larger distance. [Goo05]

the central peak and one side peak from Fig. 2.5(b) disappear. This is caused by the fact that the Fourier transform of the amplitude-phase grating is not symmetric as in the case of the amplitude grating, which is defined by a real-valued function t_c. Hence, the expression Eq. (2.24) represents a relation between the frequency and the direction of propagation, i.e., $x_{\mathbf{k}} = \frac{2\pi}{\lambda}\sin(\theta)$. Extension to 3D space follows linear theorem of the Fourier transform and therefore the additional axis is handled separately.

Propagation of the angular spectrum exploits relation between expression Eq. (2.21) and the Fourier transform. It is an alternative to Eq. (2.19). Let us now define a plane κ_z that is parallel with plane κ and the shortest distance between the plane κ and the plane κ_z is z. The equation Eq. (2.21) represents a sample at the plane κ. If the equation Eq. (2.20) is considered only for samples at the plane κ_z, the only difference between Eq. (2.21) and Eq. (2.20) is a the phase shift $\exp(\mathrm{j}z_{\mathbf{k}}z)$. Hence, the propagation between the plane κ and the plane κ_z can be expressed as

$$\mathcal{U}_z(x_{\mathbf{k}}, y_{\mathbf{k}}) = \mathcal{U}(x_{\mathbf{k}}, y_{\mathbf{k}}) \exp(\mathrm{j}z_{\mathbf{k}}z), \tag{2.25}$$

where $z_{\mathbf{k}} = (k^2 - x_{\mathbf{k}}^2 - y_{\mathbf{k}}^2)^{1/2}$ and $\mathcal{U}$ and $\mathcal{U}_z$ are angular spectrums at planes κ and κ_z respectively. If $z_{\mathbf{k}}^2 \leq 0$, the expression Eq. (2.25) is still valid but the phase shift becomes an attenuation factor $\exp(-|z_{\mathbf{k}}^2|^{1/2}z)$ instead. Waves for which $z_{\mathbf{k}}^2 \leq 0$ are known as **the evanescent waves** and they are usually zeroed inherently while calculating the expression Eq. (2.25) because they are undetectable at a distance of only a few wavelengths away from the plane κ.

Propagation of the angular spectrum can be further enhanced towards handling of tilted planes [LF88, TB93, YAC02, EO06]. The tilt is achieved by rotating the wave vector $\mathbf{k}$. The rotation is applied in a form of a 3×3 matrix $\mathbf{R}$ and the transformed wavevector $\mathbf{k}'$ is $\mathbf{k}' = \mathbf{R}\mathbf{k}$. Application of the transformed wavevector to Eq. (2.25) yields

$$\mathcal{U}_z(x'_{\mathbf{k}}, y'_{\mathbf{k}}) = \mathcal{U}(x'_{\mathbf{k}}, y'_{\mathbf{k}}) \exp(\mathrm{j}z'_{\mathbf{k}}z) J(z_{\mathbf{k}}, z'_{\mathbf{k}}), \tag{2.26}$$

where $\mathbf{k}' = (x'_{\mathbf{k}}, y'_{\mathbf{k}}, z'_{\mathbf{k}})$ and $J(z_{\mathbf{k}}, z'_{\mathbf{k}})$ is Jacobian correction factor [EO06]. The transformation of the wavevector $\mathbf{k}$ by the matrix $\mathbf{R}$ equals to a shift of spatially limited hemispherical surface over a hemisphere that is defined by all possible wavevectors excluding wavevectors of the evanescent waves as depicted in Fig. 2.6. The radius of the hemisphere is the wavenumber.

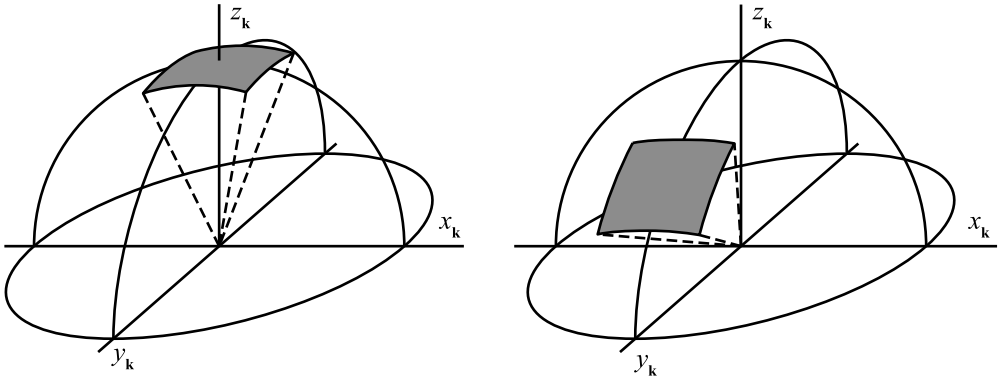


Figure 2.6: (a) All possible wavevectors with $z_{\mathbf{k}}^2 > 0$ form a hemisphere in the angular spectrum. (b) A rotation of the spectrum rotates a hemisphere. The greyed part of the hemisphere in both (a) and (b) represents the same subset of wavevectors.

2.2.4 Approximations

In section Sec. 2.2.2, we presented an expression that allows to calculate a wave distribution in a free space. The expression is accurate enough starting from distances of few wavelengths from the source. Nevertheless, for larger distances the optical field can be approximated by even simpler expressions. Since these expressions are used by various methods mentioned in this work, it is appropriate to describe them as well.

The Huygens-Fresnel principle described by Eq. (2.18) can be used to handle a case of two parallel planar surfaces as depicted in Fig. 2.7. The distance $r_{01} = |\mathbf{r}_{01}|$ is calculated as $r_{01} = [z_{\mathbf{p}_0}^2 + (x_{\mathbf{p}_0} - x_{\mathbf{p}_1})^2 + (y_{\mathbf{p}_0} - y_{\mathbf{p}_1})^2]^{1/2}$. If $z_{\mathbf{p}_0}^2 \gg (x_{\mathbf{p}_0} - x_{\mathbf{p}_1})^2 + (y_{\mathbf{p}_0} - y_{\mathbf{p}_1})^2$, we can replace square root function by the Maclaurin series [Goo05]. Such an approximation is known as the Fresnel approximation.

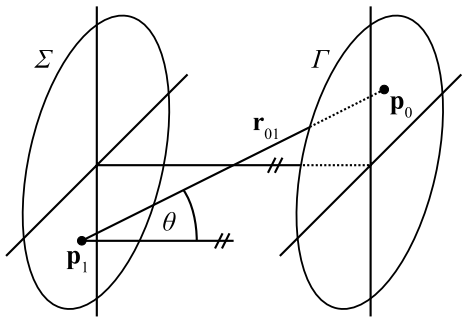


Figure 2.7: A setup used by the Fresnel/Fraunhofer approximation. [Goo05]

The Fresnel approximation exploits that $(1+b^2)^{1/2}$ can be expressed by the Maclaurin series[8], i.e.,

$$(1 + b^2)^{1/2} = 1 + \frac{1}{2}b^2 + \frac{1}{8}b^4 + \dots. \tag{2.27}$$

[8]The Maclaurin series of $(1 + b^2)^{1/2}$ is also known as the binomial expansion.

If $|b| \ll 1$, we can omit higher members of the series in Eq. (2.27). For simplification reasons, it is beneficial to use only the first two members of Eq. (2.27) yielding

$$r_{01} \approx z_{\mathbf{p}_0} + \frac{1}{2}\frac{(x_{\mathbf{p}_0} - x_{\mathbf{p}_1})^2 + (y_{\mathbf{p}_0} - y_{\mathbf{p}_1})^2}{z_{\mathbf{p}_0}}. \tag{2.28}$$

The expression Eq. (2.28) approximates the phase shift in Eq. (2.18). The distance r_{01}, which modifies the amplitude in Eq. (2.18), is approximated only by the first member of the expansion Eq. (2.28) because the optical field is very sensitive to an error in a phase but less sensitive to an error in the amplitude [MNF+02].[9] Applying the approximation reduces Eq. (2.18) to

$$u(\mathbf{p}_0) \approx \frac{\exp(\mathrm{j}kz_{\mathbf{p}_0})}{\mathrm{j}\lambda z_{\mathbf{p}_0}} \iint_{\Sigma} u'(\mathbf{p}_1) \exp\left[\frac{(x_{\mathbf{p}_0} - x_{\mathbf{p}_1})^2 + (y_{\mathbf{p}_0} - y_{\mathbf{p}_1})^2}{z_{\mathbf{p}_0}}\right] \mathrm{d}x_{\mathbf{p}_1}\,\mathrm{d}y_{\mathbf{p}_1}, \tag{2.29}$$

where $\mathbf{p}_0 = (x_{\mathbf{p}_0}, y_{\mathbf{p}_0}, z_{\mathbf{p}_0})$ and $\mathbf{p}_1 = (x_{\mathbf{p}_1}, y_{\mathbf{p}_1}, 0)$. The equation Eq. (2.29) is known as the **Fresnel diffraction integral** and with a proper reordering it takes a form that resembles the Fourier transform, i.e.,

$$\begin{aligned} u(\mathbf{p}_0) \quad &\approx \quad \frac{\exp(\mathrm{j}kz_{\mathbf{p}_0})}{\mathrm{j}\lambda z_{\mathbf{p}_0}} \exp\left(\mathrm{j}k\frac{x_{\mathbf{p}_0}^2 + y_{\mathbf{p}_0}^2}{2z_{\mathbf{p}_0}}\right) \\ &\quad \times \iint_{\Sigma} \bar{u}(x_{\mathbf{p}_1}, y_{\mathbf{p}_1}) \exp\left[-\mathrm{j}2\pi(x_{\mathbf{p}_0}x_{\mathbf{p}_1} + y_{\mathbf{p}_0}y_{\mathbf{p}_1})\frac{1}{\lambda z_{\mathbf{p}_0}}\right] \mathrm{d}x_{\mathbf{p}_1}\,\mathrm{d}y_{\mathbf{p}_1}, \end{aligned} \tag{2.30}$$

where $\bar{u}(x_{\mathbf{p}_1}, y_{\mathbf{p}_1}) = u'(\mathbf{p}_1)\exp(\mathrm{j}k\frac{x_{\mathbf{p}_1}^2 + y_{\mathbf{p}_1}^2}{2z_{\mathbf{p}_0}})$ is an optical field at the planar surface Σ multiplied by **the chirp function**.

The Fresnel approximation uses Eq. (2.27) that limits the validity of results. Since the optical field is more sensitive to error in the phase [MNF+02, Goo05], error of approximation is estimated as the third component of Eq. (2.27), i.e., the first component omitted from calculation of the phase shift in Eq. (2.18). The Fresnel approximation leads to a valid result only if the error is much less than 2π rad, i.e.,

$$z_{\mathbf{p}_0}^3 \gg \frac{k}{8}[(x_{\mathbf{p}_0} - x_{\mathbf{p}_1})^2 + (y_{\mathbf{p}_0} - y_{\mathbf{p}_1})^2]^2 \tag{2.31}$$

for all combinations of $\mathbf{p}_0 \in \Gamma$ and $\mathbf{p}_1 \in \Sigma$. If the condition Eq. (2.31) is fulfilled for the planar surface Σ and the planar surface Γ in Fig. 2.7, the planar surface Γ is said to be in **the near field** or **the Fresnel region** from the viewpoint of the planar surface Σ.

Now, let us increase the distance between the surface Γ and the surface Σ such that $z_{\mathbf{p}_0} \gg \frac{k}{2}(x_{\mathbf{p}_1}^2 + y_{\mathbf{p}_1}^2)$. Then, the chirp function in Eq. (2.30) becomes 1.0 and the expression Eq. (2.30) is reduced to

$$\begin{aligned} u(\mathbf{p}_0) \quad &\approx \quad \frac{\exp(\mathrm{j}kz_{\mathbf{p}_0})}{\mathrm{j}\lambda z_{\mathbf{p}_0}} \exp\left(\mathrm{j}k\frac{x_{\mathbf{p}_0}^2 + y_{\mathbf{p}_0}^2}{2z_{\mathbf{p}_0}}\right) \\ &\quad \times \iint_{\Sigma} u'(\mathbf{p}_1) \exp\left[-\mathrm{j}2\pi(x_{\mathbf{p}_0}x_{\mathbf{p}_1} + y_{\mathbf{p}_0}y_{\mathbf{p}_1})\frac{1}{\lambda z_{\mathbf{p}_0}}\right] \mathrm{d}x_{\mathbf{p}_1}\,\mathrm{d}y_{\mathbf{p}_1}. \end{aligned} \tag{2.32}$$

The expression Eq. (2.32) is known as **the Fraunhofer approximation** and the planar surface Γ is said to be in **the far field** or **the Fraunhofer region**.

[9] In fact, even if the amplitude set to 1.0 in all samples, the reconstruction is possible. The intensity of such reconstruction differs from the intensity of a reconstruction from the unmodified optical field.

The distance enforced by the Fraunhofer approximation is much greater than the distance required of Fresnel approximation, e.g., for a circular aperture of a radius 0.01 m and the same observation region the minimum distance required by the Fraunhofer approximation is 247.37 m while the Fresnel approximation requires only 0.23 m. Therefore, the Fraunhofer approximation is used for analysing the features of imaging system rather than hologram generation.

2.2.5 Lens

Methods presented in sections Sec. 2.2.2 and Sec. 2.2.3 allow to calculate a propagation of the wave in a free space. If an obstacle is put into the free space, the light will be diffracted. With a proper optical attributes of the obstacle we can control the propagated light. One of such an obstacle is a lens. Since the lens is mentioned in the following chapters, this sections gives a brief overview on a special version of a lens that is widely applied in numerical simulations.

A wave that propagates through an optically dense material is delayed in comparison to the same wave propagating in the vacuum. A lens is an optically dense material that is homogeneous and that has a certain geometry controlling the effect of the delay. One of the most frequently simulated lens is a thin lens known from the ray-based optics. **The thin lens** denotes a lens which the ray enters and exits at approximately the same location. Thus, the thin lens causes only a phase shift. Any other modification of wave propagation due to a different optical density can be neglected.

In the following text, a lens is centred around the origin and it is located in the plane $\kappa : z = 0$. The effect of a thin lens on an incoming wave has a form of a multiplicative factor $t_l(\mathbf{p}_1)$, where $\mathbf{p}_1 = (x_{\mathbf{p}_1}, y_{\mathbf{p}_1}, 0)$. Thus, optical field values immediately behind the lens is $u_l(\mathbf{p}_1) = u(\mathbf{p}_1)t_l(\mathbf{p}_1)$, where $u(\mathbf{p}_1)$ is the incoming wave, i.e., optical field values at the plane $\kappa : z = 0$. Attenuation due to reflection and due to losses inside the lens is omitted. According to [Goo05, SJ05], the phase shift due to a thin lens depicted in Fig. 2.8 is

$$\begin{aligned} t_l(\mathbf{p}_1) &= \exp[\mathrm{j}kn\Delta(\mathbf{p}_1)] \exp\{\mathrm{j}k[\Delta_0 - \Delta(\mathbf{p}_1)]\} \\ &= \exp(\mathrm{j}k\Delta_0) \exp[\mathrm{j}k(n-1)\Delta(\mathbf{p}_1)], \end{aligned} \tag{2.33}$$

where $\Delta(\mathbf{p}_1)$ is a lens thickness function, Δ_0 is a maximum thickness of the lens, and n is a refraction index of the lens.

The thickness function from Eq. (2.33) controls the phase delay. Following Fig. 2.8, the thickness function is

$$\begin{aligned} \Delta(\mathbf{p}_1) &= \Delta_1(\mathbf{p}_1) + \Delta_2(\mathbf{p}_1) \\ &= \Delta_0 - r_1 \left[1 - \left(1 - \frac{x_{\mathbf{p}_1}^2 + y_{\mathbf{p}_1}^2}{r_1^2} \right)^{1/2} \right] + r_2 \left[1 - \left(1 - \frac{x_{\mathbf{p}_1}^2 + y_{\mathbf{p}_1}^2}{r_2^2} \right)^{1/2} \right]. \end{aligned} \tag{2.34}$$

If the spatial extent of the lens in both the X-axis and the Y-axis is small in comparison to radii r_1 and r_2, we can approximate Eq. (2.34) by the binomial series Eq. (2.27), i.e., $[1 - \frac{1}{r_1^2}(x^2 + y^2)]^{1/2} \approx 1 - \frac{1}{2r_1^2}(x^2 + y^2)$. Applying the approximation to Eq. (2.34), the phase

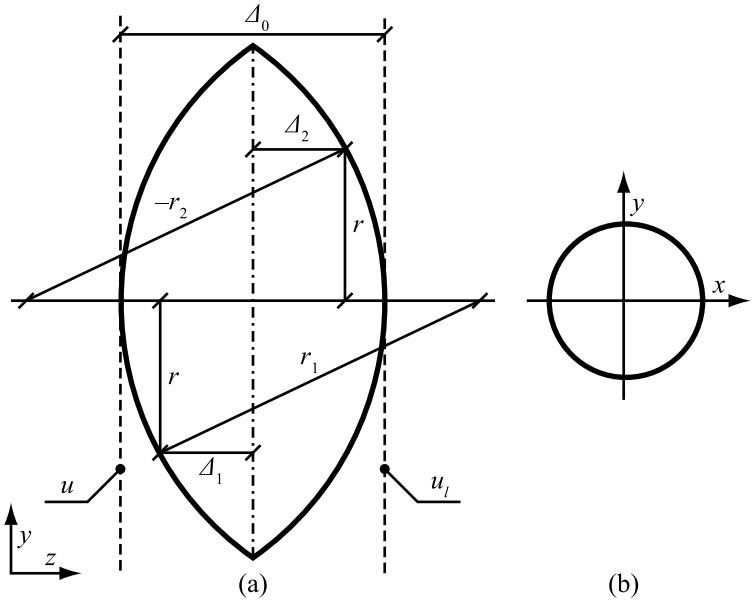


Figure 2.8: (a) A side view of the lens and (b) a front view of the lens. Resulting optical field values $u_l(\mathbf{p}_1)$ are on the right side of the lens. The input optical field values are on the left side. Notice that the radius r_2 is negative because the waves is assumed to travel from left to right. The radius r is $r = (x^2_{\mathbf{p}_1} + y^2_{\mathbf{p}_1})^{1/2}$. [Goo05, SJ05]

shift Eq. (2.33) becomes

$$\begin{aligned} t_l(\mathbf{p}_1) &= \exp(\mathrm{j}kn\Delta_0)\exp\left[-\mathrm{j}k(n-1)\frac{x^2_{\mathbf{p}_1}+y^2_{\mathbf{p}_1}}{2}\left(\frac{1}{r_1}-\frac{1}{r_2}\right)\right], \\ &= \exp(\mathrm{j}kn\Delta_0)\exp\left(-\mathrm{j}k\frac{x^2_{\mathbf{p}_1}+y^2_{\mathbf{p}_1}}{2f}\right), \end{aligned} \tag{2.35}$$

where $\frac{1}{f} \equiv (n-1)(\frac{1}{r_1}-\frac{1}{r_2})$ is **the lens maker equation** and f is a focal distance.[10] Since the multiplicative factor $\exp(\mathrm{j}kn\Delta_0)$ from Eq. (2.35) is constant for the lens, it is usually omitted from calculations because it causes just a constant phase shift.

The lens described by Eq. (2.35) causes an aberration in phase that can be neglected if the intensity I is the desired output because $I \propto |U|^2$. The aberration can be corrected by a multiplicative factor $p(\mathbf{p}_0)$ to a correct optical field $u_(\mathbf{p}_0)$. In such a case, the lens is applied as $u_(\mathbf{p}_0) = p(\mathbf{p}_0)\mathcal{P}\{u'(\mathbf{p}_1)t_l(\mathbf{p}_0)\}$ where $u'(\mathbf{p}_1)$ is an optical field value immediately in front of the lens and the propagation operator $\mathcal{P}\{\}$ is propagation of light in a free space. The correction factor is derived following the fact that any image at the distance $2f$ in front of lens is projected into almost the same image at the distance $2f$ behind the lens. The only difference is inversion of both the X-axis and the Y-axis. Hence, the multiplicate factor is determined by comparison of an optical field propagated though lens with an optical field propagated without lens.[11] As described in [SJ05], using the Fresnel approximation as the

[10] Notice that this causes a plane wave which direction of propagation perpendicular to the plane κ to focus to a single point at the distance f behind the lens.

[11] This means that in the first case we propagate at the distance $4f$ in a free space and in the second case, we propagate through lens.

propagation operation $\mathcal{P}\{\}$, the factor $p(\mathbf{p}_0)$ is

$$p(\mathbf{p}_0) = \exp\left(-\mathrm{j}k\frac{x_{\mathbf{p}_0}^2 + y_{\mathbf{p}_0}^2}{2f}\right). \tag{2.36}$$

A thin lens described by Eq. (2.35) has a wide application in the holography. It can be used to estimate an effect of the optical field on a human viewer because human visual system contains a lens. Among others, it is employed by a special case of holograms known as Fourier holograms that is mentioned briefly in Sec. 2.3.3.

2.3 Holograms

A significant part of holography deals with holograms. A hologram is a recording of the optical field that can be replayed. It was discovered by Dr. D. Gabor as a tool for microscopy [Gab49]. [12] Among others, this discovery allows to verify calculated optical fields through optical experiments, i.e., we can display content of the optical field. Therefore, this sections gives a brief overview of the principle and the basic hologram types mentioned in the following chapters. For more details, refer to [Har96].

The hologram is recorded using interference between an unknown wave and a known wave. We denote the known wave as the reference wave. The unknown wave is usually light reflected off or transmitted though a scene. The resulting interference pattern modifies opacity of a photosensitive material. If such a material is developed and put as an obstacle to the same reference wave, a consecutive diffraction causes a reconstruction of the original unknown wave and a viewer is able to see the scene. We denote the process of hologram replaying as **the hologram reconstruction**.

Let us now give a mathematical model of the process. Interference of the reference wave $u_r(\mathbf{p}_1)$ and the optical field O from the scene forms an interference pattern on a surface. At a given location, the interference pattern has intensity

$$\begin{aligned} I(\mathbf{p}_1) &= |u_r(\mathbf{p}_1) + o(\mathbf{p}_1)|^2 \\ &= |u_r(\mathbf{p}_1)|^2 + |o(\mathbf{p}_1)|^2 + u_r^*(\mathbf{p}_1)o(\mathbf{p}_1) + u_r(\mathbf{p}_1)o^*(\mathbf{p}_1), \end{aligned} \tag{2.37}$$

where $o(\mathbf{p}_1)$ is a sample of the optical field O at the point $\mathbf{p}_1$ and $o^*(\mathbf{p}_1)$ denotes a complex conjugate of a sample $o(\mathbf{p}_1)$. Even though Eq. (2.37) works for any kind of surface and any kind of reference wave, usually only a planar surface and a planar reference wave is used. Thus, for purpose of this work, the interference pattern is observed on a spatially limited recording plate (the hologram) in the plane $\kappa : z = 0$, i.e., $\mathbf{p}_1 \in \kappa$, .

Recording of the intensity $I(\mathbf{p}_1)$ to a photosensitive material followed by a developing forms an obstacle that modulates amplitude by the multiplicative factor $t_A(\mathbf{p}_1)$ that is proportional to intensity from Eq. (2.37).[13] The difference between $t_A(\mathbf{p}_1)$ and $I(\mathbf{p}_1)$ is due to physical properties of the material and thus it can be neglected for purpose of the explanation, i.e., $t_A(\mathbf{p}_1) \approx I(\mathbf{p}_1)$. Also, similar to the thin lens case discussed in Sec. 2.2.5, we assume that light is not influenced by a different optical properties of the material.

[12] If the hologram is recording by light with the wavelength λ_1 and it is replayed using light with a shorter wavelength $\lambda_2 = \xi\lambda_1$, the recorded setup is scaled uniformly in the spatial domain. This is a consequence of $\xi < 0$ applied to Eq. (2.18).

[13] Modifying the developing process, it is possible to create a phase modulating hologram that allows reconstruction too but with better efficiency, i.e., less energy of the reference wave is lost due to interaction with the material of the hologram.

In our case, the hologram modulates the amplitude of the transmitted wave by the multiplicative factor $t_A(\mathbf{p}_1)$. When such a hologram is illuminated by the reconstruction wave $u'_r(\mathbf{p}_1)$, the optical field immediately behind the hologram is $u'(\mathbf{p}_1) = t_A(\mathbf{p}_1)u'_r(\mathbf{p}_1)$. The reconstruction wave is proportional to the reference wave $u_r(\mathbf{p}_1)$ and for purpose of the simplicity let $u'_r(\mathbf{p}_1) = u_r(\mathbf{p}_1)$. As a consequence, following Eq. (2.37) an optical field sample u' located $\mathbf{p}_1$ immediately behind the hologram is

$$u' = |u_r|u_r + |o|u_r + ou_r^* u_r + u_r o^* u_r. \tag{2.38}$$

The first term of Eq. (2.38) is an undisturbed reference wave and it is known as the DC term. The second term of Eq. (2.38) is caused by a self-interference of the optical field O and is known as a halo. Both the first and the second component are unwanted and disturbing. The recorded optical field is encoded in the last two components.

The third term of Eq. (2.38) is proportional to original waves $o(\mathbf{p}_1)$ of the optical field O. When the reconstructed optical field is observed, the third term forms a view on the scene as if the scene was present, i.e., it is **the virtual image** of the scene. The fourth term contains modified copy of waves from the optical field O as well. The exact interpretation of the fourth term depends on an actual type of hologram. In general, the fourth term behaves as there would be another mirrored version of the original scene. In the first invented kind of hologram discussed in Sec. 2.3.1 the fourth term behaves as there would be a copy of the scene in front of the hologram and therefore the fourth term causes a **the real image** of the scene. An example of real and virtual image is given in Fig. 2.9.

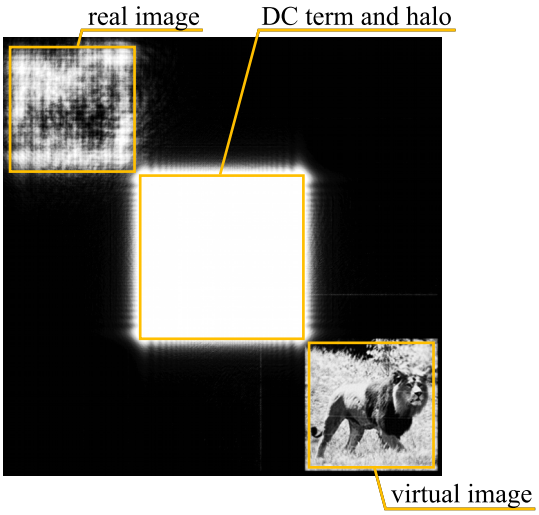


Figure 2.9: An intensity of a propagated optical field. The optical field was formed just after an amplitude-modulating hologram. In this case, the hologram is a recording of sources on a plane with various intensity. The optical field is propagated to a location of the virtual image. At the same time, the real image is out of focus. Notice that this examples present an off-axis hologram.

2.3.1 In-line Hologram

In-line hologram is the oldest and the most simple setup.[14] The setup requires that a hologram on the plane κ, a recorded object and the source of the reference wave are aligned

[14]The in-line hologram is also known as the Gabor hologram.

in a line as depicted in Fig. 2.10. The reference wave is a planar wave with a wavevector $\mathbf{k} = (0, 0, k)$. Since the reference wave is transmitted through the object, the object has to be mostly transparent and almost planar in an ideal case, e.g., a wire-frame or a transparency.

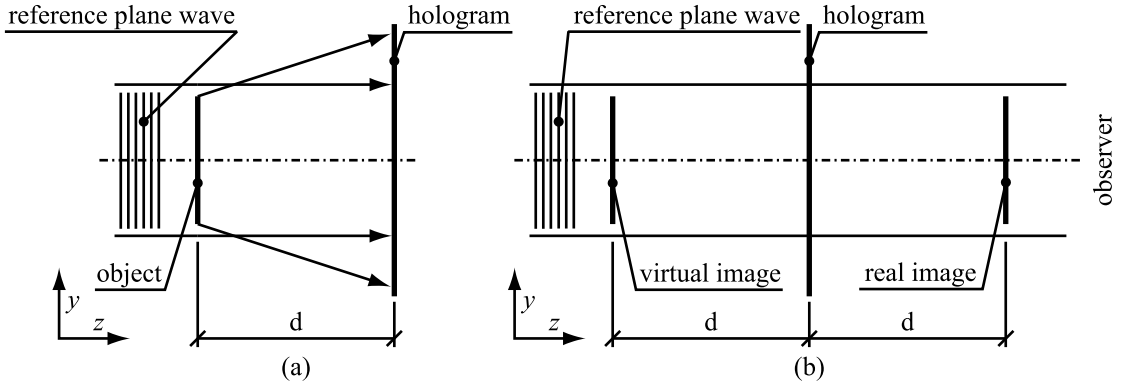


Figure 2.10: (a) An in-line hologram setup for recording and (b) a in-line hologram reconstruction. The distance of the object to the hologram is d. [Har96]

During the hologram reconstruction, the in-line hologram forms a real and a virtual image that overlap each other when observed. As depicted in Fig. 2.10(b) real image is symmetrically located on the opposite side of the hologram. The depth of the real image is inverted. Since all terms in Eq. (2.38) overlap when the hologram is observed, the in-line hologram is applicable only to a high-contrast objects, very small objects, or a sparse field of particles and therefore it is not usually considered for displaying purposes.

2.3.2 Off-axis Hologram

The off-axis hologram offers a solution to the overlapping problem of the in-line hologram.[15] It assumes a setup that is almost similar to the in-line hologram with exception of the reference wave that hits the hologram plane κ under a different angle as shown in Fig. 2.11(a), i.e., the wavevector $\mathbf{k} \neq (0, 0, k)$. Unlike the in-line hologram, the objects can be a 3D-shape without restriction because the optical field generated by the scene is created by light reflecting off the scene.[16]

Since the incidence angle of the reference wave in Fig. 2.11 is different from zero, the terms of Eq. (2.38) are partially separated during reconstruction by a different direction of propagation. As it is shown in Fig. 2.11(b), staring from a particular distance the virtual image does not overlap any other term as illustrated with Fig. 2.9. The higher the incidence angle, the better the separation. However, increasing the incidence angle increases the maximum frequency in the interference pattern at the same time. This is a drawback of the off-axis hologram because it requires materials able to record high frequency patterns while retaining the contrast. Nevertheless, the off-axis hologram allows to record and reconstruct reflected light from 3D-object and therefore it is suitable for displaying purposes.

[15]The off-axis hologram is also known as the Leith-Upatnieks hologram.

[16]To satisfy physical limitations imposed by coherence, the light illuminating the scene is generated by a beam splitter from the reference wave. At the recording plane the difference of the path between these two split parts of the light has be less than the coherence length mentioned in Sec. 2.1.1, otherwise the interference pattern will not be visible. This means that the coherence length limits the maximum depth that can be recorded in the hologram.

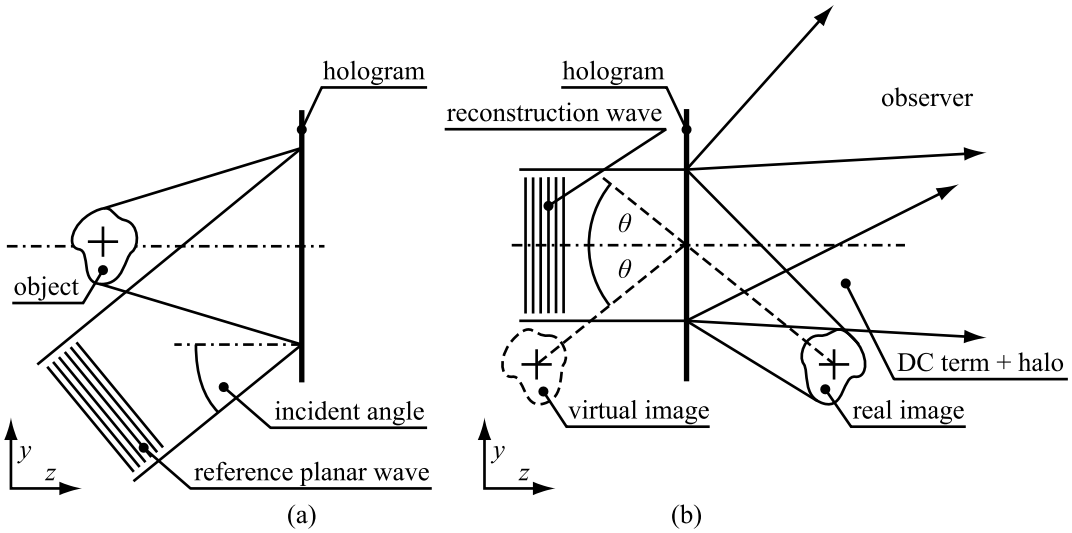


Figure 2.11: (a) An off-axis hologram recording setup and (b) a reconstruction. The angle θ represents an incidence angle of the planar reference wave from the hologram. [Har96]

2.3.3 Fourier Hologram

Unlike previously mentioned holograms, **the Fourier hologram** uses a lens for recording purpose. Through the lens, a Fourier transform of the optical field is create and through interference it is recorded as depicted in Fig. 2.12(a).[17] The object is assumed to be a transparency because it is usually illuminated front the back.

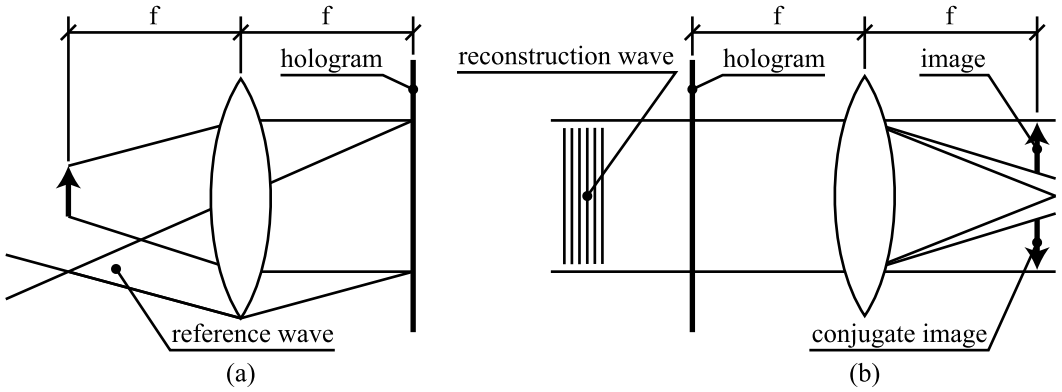


Figure 2.12: (a) A Fourier hologram recording setup and (b) a reconstruction. The distance f is a focal distance of the lens. [Har96]

During the reconstruction, the hologram is illuminated by a planar wave. The reconstructed optical field passes through a lens as depicted in Fig. 2.12(b) and at the back focal plane it forms the real and the virtual image. Both images are at the focus and they are symmetric as illustrated with Fig. 2.13. Optically created Fourier hologram are not used for displaying purposes because they assume transparent object that are almost planar. Nevertheless, Fourier holograms are used in numerical simulations and in that case they allow recording of 3D scenes.

[17] A lens transforms an optical field such that an optical field at the back focal plane is a Fourier transform of the optical field in the front focal plane and vice versa [Goo05].

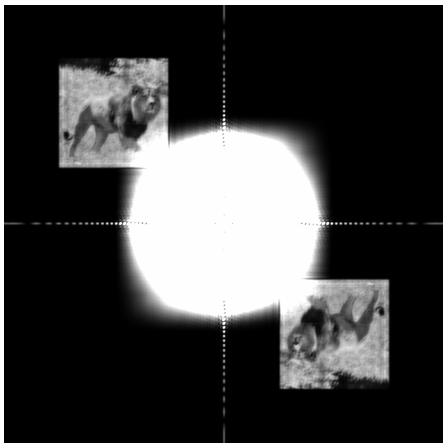


Figure 2.13: A numerical simulation of reconstruction from the Fourier hologram.

Chapter 3

Digital Holography

Digital holography is a discrete counterpart of optical holography. It deals with numerical simulations of light behaviour. It employs models described in the previous chapter and it designs their new interpretations. This chapter gives an overview of existing solutions, i.e., the previous work. Since this work is aimed on hologram generation, the overview focuses on the subject as well. This chapters shall help the viewer to clarify the design decisions in the method proposed in the next chapter.

Since digital holography is able to numerically simulate interaction of light with the environment, it is able to fully cooperate with optical holography, i.e., digital holography may post-process data recorded during physical experiments or it may provide data for physical experiments. All numerical simulations runs in a discrete environment of a computer and follows discrete versions of expression mentioned in Chap. 2. From a viewpoint of this work, digital holography solves hologram generation, numerical reconstruction of the hologram, information retrieval from the hologram, compression of the hologram, and hologram reproduction as depicted in Fig. 3.1.

Hologram generation deals with calculation of a hologram or an optical field from a virtual scene. In this work, calculated holograms are intended for viewing purposes. Since hologram generation is the aim of this work, it is discussed in greater detail later in this chapter. Following text gives a brief overview of other subgroups.

The most technologically demanding area is hologram reproduction. The aim of hologram reproduction is to introduce digitally simulated data to physical experiments, i.e., replaying of the calculated hologram. Existing solutions exploits various printing techniques to create static holograms including hologram binarisation [BL66, Hua71, Har96] as a pre-step to printing using laser printers [Mac97], CD/DVD burners [SMU04], and custom printers [MKM06]. Dynamically changing holograms can be replayed using spatial light modulators (SLM) [Goo05, SCS05, KIO$^+$06] that allow controlled modification of light on a very small scales. Since this work aims on hologram generation, some of available solutions for hologram reproduction were used for purposes of optical verification.

The results of optical verification proves functionality of the numerically calculated hologram. However, hologram reproduction may be expensive or slow and therefore a numerical reconstruction can be used to detect a hologram that does not work. Usually, a hologram is located on a plane $\kappa : z = 0$ and numerical reconstruction applies formulations from Sec. 2.2.3 and Sec. 2.2.4 to find an optical field generated by the hologram on the plane $\kappa_\xi : z = \xi$. For viewing purposes, the intensity is extracted from the optical field. Usually, no lens is applied

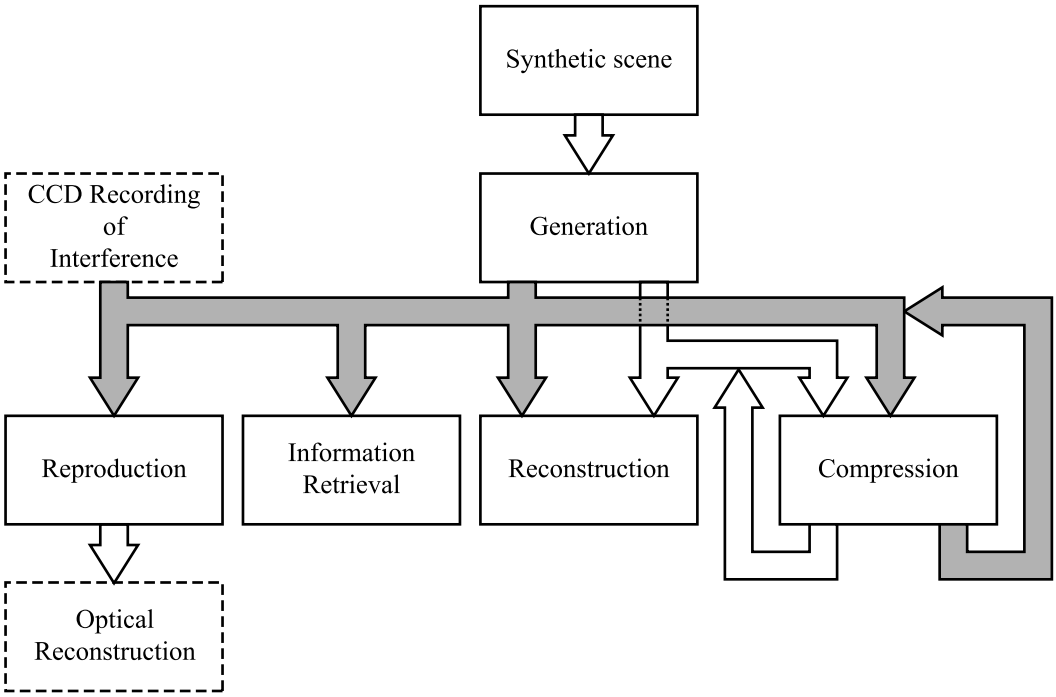


Figure 3.1: Digital holography areas and their relation in a context of this work. The dashed boxes belong to optical holography. Lines between boxes represents data transfer: white lines represent an optical field, greyed lines represent an interference pattern, i.e., a hologram.

in the process and therefore if the input is a hologram of a virtual scene, the plane κ_ξ has to intersect the scene. Sources of waves at the intersection are denoted as being in the focus. Even though such result is not equal to an image formed on a retina of an eye because it lacks both a lens and an aperture, it is acceptable for verification purposes[1].

With increasing size of a hologram or an optical field, the amount of required data is increased too. While for storage purposes this might not be crucial, the amount of data is a problem for a data transfer between devices [Luc96]. For that reason, hologram compression aims to decrease the size of hologram representation while avoiding degradation of the hologram. The simplest solution may rely on compression schemes intended for image compression [NFJT02]. More sophisticated solutions may incorporate the compression as a part of the hologram generation method [Luc94, Luc96]. This allows finer control of compression-based noise. The compression of holograms is outside the scope of this work and therefore we shall not discussed it any further.

An optical field encoded in a hologram contains information that can be used to estimate attributes of the source or obstacles between the source and the hologram. Information retrieval methods try to extract information from the hologram captured by the CCD cameraSec. 2.3. An original optical field is the basic content that can be extracted from a hologram [YZ97, LBU04]. If the scene is similar to a sparse set of particles, detection of particle locations is possible [BLCLO00]. However, in a general case, it is very complicated, or even impossible, to make an estimation of an obstacle shape that caused given interfer-

[1]If the hologram or an optical field forms expected intensity pattern on a plane κ_ξ, the method that generated the hologram is considered correct.

ence pattern. Since the task of the information extraction is beyond scope of this work, this paragraph is presented only for completeness of the list.[2].

3.1 Principles of hologram generation

The aim of this work is to generate holograms for viewing purposes. This section presents basic methods in greater detail and defines an input and an output of hologram generation. The purpose of this section is to provide an overview of the most common methods to the reader. Acceleration techniques are discussed in the next section.

Hologram generation deals with calculating discrete samples u_{mn} of an optical field U or discrete samples h_{mn} of a hologram H. As shown in Sec. 2.3, optical field U can be reduced to the hologram H and thus we shall consider the optical field as the default output of a method.[3] Furthermore, let us assume that samples are located on the plane $\kappa : z = 0$ and they are organised into a rectangular and uniform grid. A distance between the samples along the X-axis and the Y-axis is D_x and D_y respectively. As a consequence, the location of a sample is $\mathbf{u}_{mn} = (mD_x, nD_y, 0)$. For purpose of simplification, let the optical field U be represented by $N \times N$ samples.

The optical field U that is sampled at the plane κ is generated by a virtual scene. The virtual scene consists of objects. Every object is defined by the surface.[4] The scene is organised along the negative Z-axis and its orthogonal projection onto the plane κ does not overlap the maximum extent of the hologram given by the number of samples and the sampling step.[5]

The surface of objects is described by a triangular mesh with a complex texture $A(\mathbf{s}) = a(\mathbf{s}) \exp[\mathrm{j}\varphi(\mathbf{s})]$ where $\mathbf{s}$ is a point on the surface, $a(\mathbf{s})$ is an amplitude, and $\varphi(\mathbf{s})$ is a phase. The texture contains a result of interaction between a light in the scene and the surface. Such a surface is considered as a self-luminous surface, i.e., it is the source of waves and it is not influenced by self-interference. The texture is an input because hologram generation focuses only on calculation of a hologram and it does not try to simulate interaction between light and the scene. Besides that, the phase $\varphi(\mathbf{s})$ shall not be constant because a self-luminous surface with such a texture is not able to form a viewable hologram [LHJ68]. The effect of such a constant phase is depicted in Fig. 3.2.

Hologram generation follows discrete version of formulations from Sec. 2.2.1. Individual methods are usually based on a discrete Rayleigh-Sommerfeld formulation that describe an optical field in a free space behind the planar aperture. The formulation assumes that the optical field is generated by a single point light source (PLS) in front of the aperture or an optical field in the aperture. The aperture is the obstacle, PLS in front of the aperture is a point on the surface, and the observing point is the a calculated sample.

Hologram generation method address two problems: a propagation of a wave in a free space between obstacles and influence of obstacles on the wave. The influence of obstacles is commonly denoted as a visibility. Solutions of the problems applied by various methods

[2] Hrome, tuto část snad píši počtvrté zcela znovu.

[3] Backward conversion of a hologram into an optical field is possible but it is beyond the scope of the work.

[4] Since both translucency or transparency, which might require the knowledge of volume, are not considered in majority of cases, the surface is sufficient for a description of the object.

[5] This is in agreement with the sign notation mentioned in Sec. 2.2.2. Nevertheless, even the scene that is organised along the positive Z-axis can create a working hologram.

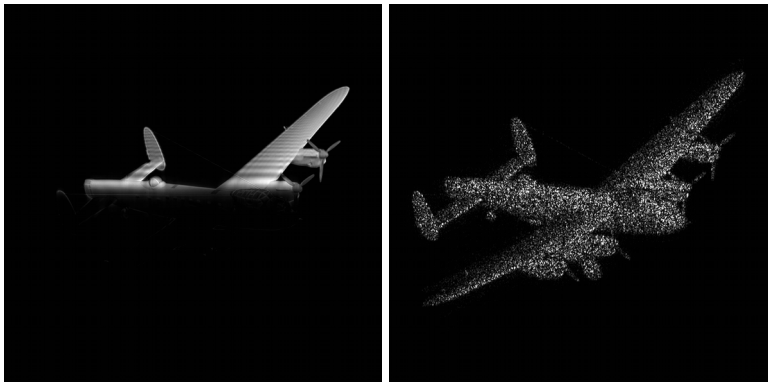


Figure 3.2: A numerical reconstruction from a partially covered optical field generated by a scene with (a) a constant phase on the surface and (b) a pseudo-random phase on the surface. Lower half of the optical field was zeroed covered prior the reconstruction. Notice that this is similar to applying an aperture, e.g., a pinhole of a camera or a pupil of an eye.

clusters the methods into trends: geometry-based methods, wave-base methods, and view-based methods. All three groups are discussed in following subsections.

3.1.1 Geometry-based methods

A common attribute of geometry-based methods is a use of a ray for calculation of a contribution to the optical field. This is similar to the computer graphics but in this case the ray carries a phase besides intensity. The phases of contributions in neighbouring samples have to correspond each other for a successful reconstruction. As a consequence, the scene cannot be sampled on a complete random basis. The visibility is solved by a ray-casting [Wat00] as illustrated with Fig. 3.3. Even though using of ray-casting for a visibility solution means ignoring diffraction on obstacles, it creates a working hologram and therefore it is considered as an acceptable approximation [Und97].

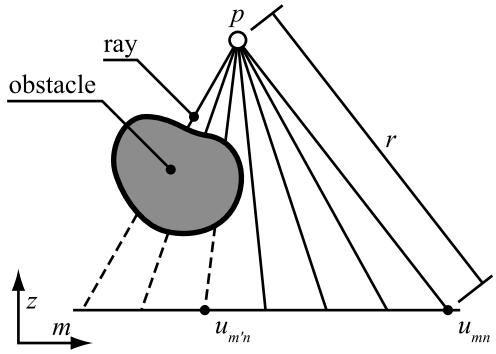


Figure 3.3: Visibility solution of PLS p using a geometry-based method. While the sample $u_{m'n}$ does not obtain any contribution from PLS due to the obstacle, the contribution to sample u_{mn} is proportional to $\frac{\exp(\mathrm{j}kr)}{r}\frac{z}{r}$.

In order to force the correspondence between phases, the surface is usually converted to different representations, e.g., a cloud of PLS. When a cloud of PLS is used, the computation complexity is $\mathcal{O}(PN^2)$ where P is number of PLS. For an acceptable visual quality the

number of points has to be high, i.e., $P \sim N^2$. As a consequence, the worst computational complexity is beyond $\mathcal{O}(N^4)$. Since a selection of a proper representation is a exploited by various acceleration techniques, it is further discussed in Sec. 3.2.

Geometry-based methods use a simple calculation and allows various acceleration techniques including hardware-based solutions as discussed in Sec. 3.2. In majority of cases gained acceleration is linear only. If the task is properly limited, smaller holograms can be generated in a real-time manner [Luc94, IMY+05]. The drawback of geometry-based methods is an amount of data such as PLS that has to be processed, i.e., a data bandwidth is high. This is the major limitation of the geometry-based methods. Since geometry-based methods show similarity to computer graphics methods, they are partially exploited by this work.

3.1.2 Wave-based methods

Wave-based methods rely on a propagation of the wave. Especially, a propagation of the angular spectrum, which is discussed in Sec. 2.2.3, is used because it allows to calculate the optical field much faster. The major problem of wave-based methods is visibility because the propagation in angular spectrum operates in frequency domain while visibility has to be solved in the spatial domain in a general case though masking some optical field samples. This forces frequent switching between domains and it leads to a usual computation complexity of $\mathcal{O}(TN^2 \log_2 N)$ where T is number of propagation-visibility pairs that have to be calculated, e.g., it is number of planar patches in the scene.

The angular spectrum of the optical field is obtained by FFT. Since FFT assumes periodicity, the result of propagation behaves as if the input consisted of periodically repeated patch of optical field samples as illustrated with Fig. 3.4. Also, FFT assumes a uniformly sampled signal. This becomes a problem when the optical field is rotated. The rotation described in Sec. 2.2.3 causes a non-linear deformation of the spectrum and the spectrum has to be resampled prior application of the inverse FFT [EO06]. This degrades the signal and increases an amount of noise.

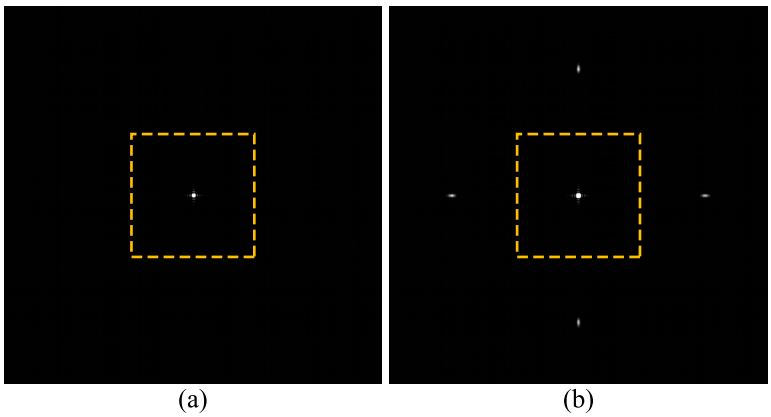


Figure 3.4: A numerical reconstruction of an optical field with added frame of zeros. Optical field was calculated using (a) analytical expression for PLS and (b) propagation of the angular spectrum. Notice the repeating of the pattern due to FFT in (b). The dashed line shows a size of the original optical field.

As shown in Sec. 2.2.3, wave propagation is the most straightforward for two parallel planes. This fact is exploited by **layered holograms** [Loh78]. The scene is sliced by planes

parallel with the plane κ and waves are propagated from slice to slice towards the plane κ. Each slice serves as an amplitude-phase modulator. The intersection of the surface with the slice is a patch of sources and it is added to the optical field samples. Te area corresponding to the object inside is a mask that zeroes optical field samples inside the object as illustrated with Fig. 3.5(a). The drawback of the method is a high number of slices for a proper approximation of the surface. Howover, at the same time the distance between slices has to be long enough for propagation to become something else than a plain phase shift due to the diffraction condition Eq. (2.24).

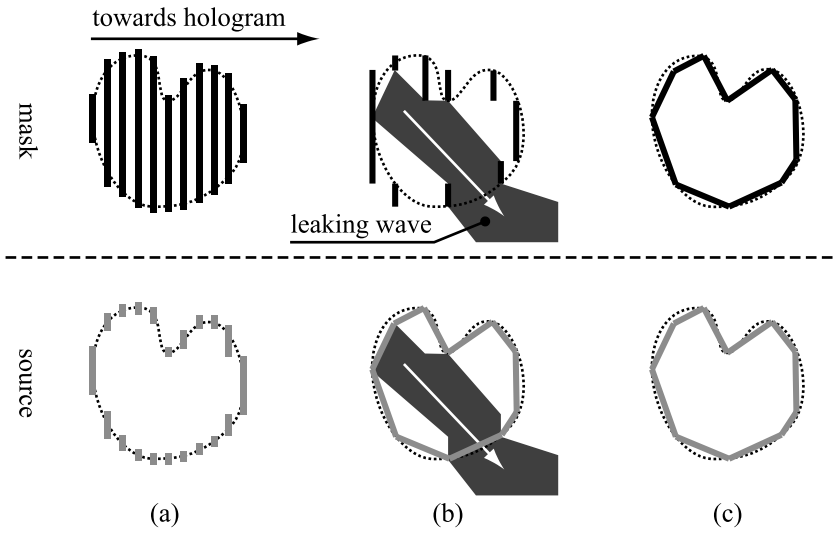


Figure 3.5: A visibility solution for wave-based methods is based on set of masks and corresponding patches of sources. (a) Layered holograms uses parallel planes to slice the scene, (b) the silhouette approximation combines parallel masks with titled sources but suffers from wave leaking, and (c) an exact solution requires frequent rotation of the optical field.

Reduction of slices is possible through a better approximation of the surface, i.e., using titled patches. If the distance between samples is large enough such that the maximum frequency $f_{\max} = \frac{1}{2D_x}$ is less than $\frac{1}{\lambda}$, the mask for a titled planar patch can be approximated by a parallel patch. This is known as **the silhouette approximation** [MK04] and it removes necessity to rotate the arriving optical field because only a newly added patch of sources is rotated as illustrated with Fig. 3.5(b). Applied mask is an orthogonal projection of the patch with sources. While the solution reduces the number of necessary slices it heightens an influence of leaked waves. **The wave leaking** denotes a situation when a wave originating from completely occluded patch is not masked and reaches the hologram as depicted in Fig. 3.5(b).

The problem of leaking waves is solved through a titled mask [Mat05]. An incoming optical field is rotated to become parallel with the source planar patch, it is masked and propagated to the next plane as illustrated with Fig. 3.5(c). Originally, the mask is binary but it can be enhanced towards a complex multiplicate factor that may have attributes of a lens or another passive optical element [ZCG08].[6] The visibility has to be solved by a

[6]Actually, the only difference between [ZCG08] and the original work of [Mat05] is the mask that is not binary but uses a complex number instead. It is rather strange that such a trivial modification was considered as a contribution worth of mentioning in a full paper on the EG conference.

Painter's algorithm [Wat00]. The algorithm is modified to consider the range in which the source can influence the final optical field.[7]

The method does not require any conversion of the input scene because both the source and the mask can be a triangle from the triangular mesh. The angular spectrum of the triangle is obtained by rasterization of the triangle into a regular grid [Mat05]. Such a solution allows any variation of both a phase and an amplitude. Another solution is to decompose the triangle using a scheme similar to the Sierpiński triangle [KHL08]. Such a scheme decomposes the original triangle to a group of smaller triangles the consist of two types: a triangle of the same shape of the original triangle and an up-side version of the triangle. Both smaller triangles have a constant phase and a constant amplitude. An angular spectrum of such triangles can be described by an analytical function [ABMW08, KHL08]. The solution allows only a limited variation of both a phase and a amplitude over the surface of the triangle in comparison to a solution that uses sampling. Notice that a hologram calculated from an object completely without a phase variation is not viewable by a human viewer [LHJ68].

Wave-based methods are able to calculate optical field of a virtual scene quickly if the visibility is ignored. If the visibility is considered a frequent execution of FFT occurs in a general case because visibility is efficiently solved in the spatial domain while the propagation is efficiently solved in the frequency domain. Also, wave-based methods suffers from limitations of FFT, i.e., an assumption of periodicity and a requirement of a regular grid that has be forced by resampling of the angular spectrum if necessary. While silhouette approximation resamples only a newly added angular spectrum while keeping the calculated optical field intact, the full solution resamples the calculated optical field over and over. This increases the amount of noise. Besides that, the angular propagation applied together with the Painter's algorithm limits possibilities of acceleration by technical means such as parallel computation.

3.1.3 View-based methods

View-based methods are not in a scope of this work and they are presented for completeness of the list only. Thus, the presentation is only brief. View-based methods exploit the mechanism of the Fourier hologram discussed in Sec. 2.3.3. The difference between individual methods in an interpretation how is the optical field in the back focal plane influenced by PLS in front of the lens. Nevertheless, all methods use multiple orthogonal views of the scene. The visibility is solved by standard means of the computer graphics.

A solution presented in [LAe01, AR03] shows that views can be generated by a tilting camera, if maximum tilt angles are kept small. Other solution presented in [SIY04] shows that the same effect can be achieved by a camera that looks at a point in the scene and rotates around the Z-axis at the same time. Nevertheless, before further method-dependant processing, the view is transformed by FFT. Then, it is processed and added to the resulting optical field. This makes the computational complexity of view-based methods approximately $\mathcal{O}(VN^2 \log_2 N)$, where V is a number of used views.

View-based methods are able to calculate optical field without interference and coherence. The computational complexity does not depend on a content of the scene because orthogonal views are the only inputs. Furthermore, view-based method might allow capturing real-life

[7] The range depends on the maximum frequency $f_{\max}$ that can be recorded in the discrete optical field. If $f_{\max}$ is applied to the diffraction condition Eq. (2.24), it gives a maximum deflection angle and as consequence a range that is influenced by a sample or by a patch.

scenes if the views are kept close to an orthogonal projection [SIY04]. Nevertheless, view-based methods are sensitive to rounding errors of the rasteriser if the views are calculated instead of recorded. Also, the number of required views might be high. Even though multiple views can be interpolated from existing ones [KSR07], still each view has to be processed separately, i.e., amount of data that has to be processed is still the same.

3.2 Acceleration of hologram generation

Previous section described the most common principles of hologram generation. Since calculation time is a major interest of this work, acceleration techniques are discussed separately in this section. This section contains an overview of possible acceleration techniques used by a wide range of methods.

The task of hologram generation follows the definition from the beginning of Sec. 3.1. The goal is to calculate discrete samples u_{mn} of an optical field U on a plane $\kappa : z = 0$ from a scene described by a triangular mesh with an applied texture. Similar to Sec. 3.1 the scene is assumed to be organised along the Z-axis and orthogonal projection of the scene onto the plane κ does not overlap a bounding rectangle of the samples.

In its principle, hologram generation calculates unknown target samples of the optical field from a known source samples of the same optical field. Usually, both sets of samples are located on a surface. If there is no obstacle between surfaces and the content stored in source samples has to be preserved, a number of target samples has to be greater or equal to a number of source samples. Since every source sample contributes to every target sample, the worst computational complexity for N^2 target samples is above $\mathcal{O}(N^4)$. The goal of acceleration is to find a coefficient χ such that the computational time driven by the computational complexity

$$\mathcal{O}\left(\frac{1}{\chi}N^4\right). \tag{3.1}$$

is reduced.

Since the work aims on holograms for viewing purposes, N in Eq. (3.1) is a larger number.[8] Therefore, even a linear acceleration means a significant reduction of the computation time. Thus, it is acceptable to deal with hardware compatibility or parallel/distributed computing. We compared principles of published method and we clustered them into groups of acceleration techniques based on simplification of the scene, simplification of the signal, and approximation including a special case. Usually, various acceleration techniques can be combined easily and the computation time reduction is cumulative in such a case.

3.2.1 Approximations and special cases

A technique that promises a reduction of calculation time is a replacement of an operation with fast yet less accurate formulation. In the digital holography, the most complicated operation is a square root that is usually applied for calculation of a distance $r = (x^2 +$

[8]This assumption is based on the diffraction condition Eq. (2.24) and the sampling step size. As it is shown in Sec. 3.1.2, the sampling step size limits the range of target samples that are affected by a given source sample. In order to calculate a proper hologram, the source sample should to be able to contribute to all target samples, i.e., the sampling step has to be as small as possible. As a consequence, the number of samples is high.

$y^2 + z^2)^{1/2}$ in Eq. (2.16). The square root can be approximated the first two members of the binomial series Eq. (2.27). The result is a function

$$r \approx z + \frac{x^2 + y^2}{2z}, \tag{3.2}$$

where the first component is omitted because it is constant for a constant z or it is possible to set z as the integer multiple of the wavelength. The function Eq. (3.2) can be calculated directly [NSM+05] or by an iterative process using a differential scheme [IMY+05, YIO00]. A single step of the iterative process for a set-up depicted in Fig. 3.6 is

$$\begin{aligned} r_{i+1} &= r_i + \Delta r_i, \\ \Delta r_{i+1} &= \Delta r_i + \Delta\Delta r, \end{aligned} \tag{3.3}$$

where $\Delta r_i = \frac{\partial r}{\partial x} = \frac{2xD_x + D_x^2}{2z}$, $r_0 = z$, and $\Delta\Delta r = \frac{\partial^2 r}{\partial x^2} = \frac{D_x^2}{z}$ is constant. The differential scheme allows to use additional members of the binomial series from Eq. (2.27) easily [YIO00]. Beside the calculation of the distance, the approximation can be applied for calculation of the wave vector component $z_{\mathbf{k}}$ [MK04]. Since $z_{\mathbf{k}} = (k^2 - x_{\mathbf{k}}^2 - y_{\mathbf{k}}^2)^{1/2}$ the approximation leads to a result similar to Eq. (3.2). In this case, however, the approximation is applied to obtain a separable function rather than to provide a speed-up.

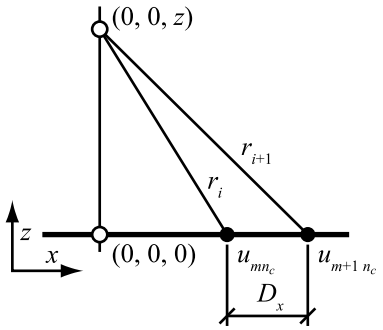


Figure 3.6: A setup used by a diffraction scheme from Eq. (3.3). A sample at $(0, 0, z_0)$ contributes to samples u_{mn_c} and $u_{m+1\,n_c}$, where $n_c = \text{const.}$. [YIO00]

Besides the removal of the square root, the approximation allows to show spatial relation between optical fields generated by sources in various depths. Let us demonstrate it using two PLS. The first PLS s is at $\mathbf{s} = (0, 0, z)$, the second PLS s_0 is at $\mathbf{s}_0 = (0, 0, z_0)$, where $z = z_0 + c\lambda$ where c is an integer. Intensity of both PLS equals to one. Phases of both PLS are the same and thus it can be omitted from calculation because the scene contains only these two PLS. Following Eq. (2.29), the contribution of PLS s to the sample $u(x, y)$ of the optical field on the plane $\kappa : z = 0$ is $u_s(x, y) = \frac{1}{z}\exp(\mathrm{j}kz + \mathrm{j}k\frac{x^2+y^2}{2z})$ and contribution of the sample s_0 is $u_{s_0}(x, y) = \frac{1}{z_0}\exp(\mathrm{j}kz_0 + \mathrm{j}k\frac{x^2+y^2}{2z_0})$. Since $z = z_0 + c\lambda$, components $\exp(\mathrm{j}kz)$ and $\exp(\mathrm{j}kz_0)$ can be omitted because $\exp(\mathrm{j}kz) = \exp(\mathrm{j}kz_0)$. As a result, it is possible to state that

$$u_s(x, y) = \sigma' u_{s_0}(x, y), \tag{3.4}$$

where $\sigma' = \frac{z_0}{z}\exp\left[\mathrm{j}k\left(\frac{z_0}{z} - 1\right)\right]$.

If $c\lambda \ll z_0$, the factor σ' might become neglectable because $\frac{z_0}{z} \approx 1$. However, in such a case equality is lost because $z \neq z_0$. Now, let us introduce coordinates $x' = x(\frac{z_0}{z})^{1/2}$ and $y' = y(\frac{z_0}{z})^{1/2}$ in to the left size of Eq. (3.4). After that, equality is restored even though factor $\sigma' \approx 1$ is omitted. The coordinates x' and y' can be interpreted as scaling of the optical field.

Hence, optical field generated by PLS s is a scaled version of the optical field generated by PLS s_0. The scale factor is

$$\sigma = \left(\frac{z_0}{z}\right)^{1/2}, \tag{3.5}$$

e.g., an optical field on the plane $\kappa : z = 0$ of PLS at $(0, 0, 4z_0)$ is approximately an optical field of PLS at $(0, 0, z_0)$ that is scaled twice as illustrated with Fig. 3.7. This fact is used by methods the rely on precalculated optical fields [BFJ+90, RBD+99, PM03]. Among others, Eq. (3.5) shows that a linear scaling of the optical field on the plane $\kappa : z = 0$ causes quadratic scaling of the distance. As a consequence, Eq. (3.5) is applicable if $|z - z_0|$ is small.

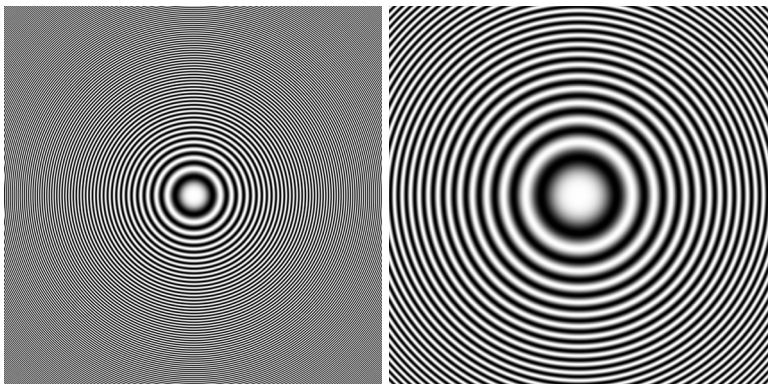


Figure 3.7: A real component of an optical field generated by PLS at (a) a distance z_0 and (b) a distance $4z_0$. Notice that additional circles in (a) are a product of an alias.

Further decrease in calculation time is possible by directly calculating a hologram instead of an optical field. A hologram is product intensity of interference between optical field samples $o(\mathbf{p}_1)$ generated by the scene and optical field samples $u_r(\mathbf{p}_1)$ known as the reference wave. As shown in Eq. (2.37), intensity of an interference pattern at the point $\mathbf{p}_1$ is $I(\mathbf{p}_1) = |u_r(\mathbf{p}_1)|^2 + |o(\mathbf{p}_1)|^2 + u_r^*(\mathbf{p}_1)o(\mathbf{p}_1) + u_r(\mathbf{p}_1)o^*(\mathbf{p}_1)$ where the most important is the third term and the fourth term because they cause a reconstruction of optical field samples $o(\mathbf{p}_1)$ and influence a visibility of an intensity pattern.[9] Therefore, it is beneficial to use only the third term and the fourth term and approximate the intensity $I(\mathbf{p}_1)$ as

$$\begin{aligned} I_{\mathrm{B}}(\mathbf{p}_1) &= u_r^*(\mathbf{p}_1)o(\mathbf{p}_1) + u_r(\mathbf{p}_1)o^*(\mathbf{p}_1), \\ &= 2|u_r(\mathbf{p}_1)||o(\mathbf{p}_1)| \cos\left[\varphi_o(\mathbf{p}_1) - \varphi_r(\mathbf{p}_1)\right], \end{aligned} \tag{3.6}$$

where $\varphi_o = \arg\{o(\mathbf{p}_1)\}$, and $\varphi_r = \arg\{u_r(\mathbf{p}_1)\}$. Since the resulting intensity can be negative, intensity I_{B} defined by Eq. (3.6) is also known as **the bipolar intensity** [Luc92]. Calculating only the bipolar intensity reduces to computation time to almost a half because it uses real numbers instead of complex arithmetic. The resulting hologram, however, is tailored only for a particular reference wave.

The computation time can be efficiently reduced by decreasing the number of contribution collected by the hologram. An example of such reduction is a setup that considers only one dimension instead of two. The effect of such reduction is a loss of parallax in the omitted dimension, i.e., in an extreme case, the viewer sees the same image from locations that

[9] The first term, aka. DC term, from Eq. (2.37) can be omitted because it modifies only the lightness of the result. The second term, aka. self-interference term, causes only disturbing artifacts and therefore it is desirable to omit it as well.

varies in the omitted dimension. Since human eyes are organised horizontally, it is desirable to preserve the horizontal dimension [Luc92, Luc94]. The resulting hologram is known as horizontal parallax only hologram, i.e., **the HPO hologram**. It is, in fact, a set of 1D holograms, each calculated independently on the others. This reduces the basic computational complexity of HPO holograms to $O(N^3)$. Nevertheless, while numerical reconstruction of such hologram is simple, optical reconstruction requires a special reconstruction setup [LAe01] or a specialised output device [LG95, Luc97] in order to prevent spreading of light in the vertical direction.

3.2.2 Simplification of a scene

In the previous section, we presented techniques that exploit approximations and special cases to reduce the computation time. Other possibility for acceleration is to manipulate the scene. Therefore, in this section we present techniques that focus on a scene and employs different description of scene content to reduce the computation time. Objects in the scene are described by a triangular mesh. Even though the mesh is already a set of primitives instead of a smooth continuous surface, it can be simplified even further. The goal is to find a representation using a primitive whose optical field can be calculated easily. Following the goal, the surface can be represented as a cloud of PLS, a wireframe (a cloud of line segments), and a cloud of triangles. A brief description of representations follows.

Cloud of point light sources

Hologram generation from a single PLS is fast because it is calculated through rays and its computational complexity is $\mathcal{O}(N^2)$. When the number of PLS is increased, the calculation time is increased linearly. Unfortunately, in order to represent a solid surface, the amount of PLS is high, comparable with N^2.[10] Despite that disadvantage, there is a large number of methods that accelerate hologram generation from a cloud of PLS. The reason is the simplicity of the optical field generated by PLS and a wide range of acceleration possibilities.

PLS emits a spherical wave described by Eq. (2.9). A sample of the final optical field on the plane $\kappa : z = 0$ can be estimated by a discrete Rayleigh-Sommerfeld solution

$$u_{mn} = \sum_i \nu_{mni} \frac{s_i}{r_{mni}} \exp(\mathrm{j}kr_{mni}) \frac{z_{\mathbf{s}_i}}{r_{mni}}, \tag{3.7}$$

where $r_{mni} = [(mD_x - x_{\mathbf{s}_i})^2 + (nD_y - y_{\mathbf{s}_i})^2 + z_{\mathbf{s}_i}]^{1/2}$ is a distance between PLS and the sample u_{mn}, s_i is a complex amplitude of PLS located at $\mathbf{s}_i$, and $\nu_{mni} \in \{0, 1\}$ is a result of a visibility check. The visibility check is zero if PLS s_i is not directly visible from sample u_{mn}, otherwise it is equal to one. An optical field of a single PLS is symmetrical. This fact can be exploited to save the calculation time by calculating only a $\frac{1}{8}$ of the optical field[JOPBV97] and by distributing calculated values to remaining samples. This speeds up the computation approximately four times.

Another possibility is to use tables of precalculated components of Eq. (3.7) such as a sine table, a cosine table, and a distance table. While both the sine and the cosine tables are 1D and small, the distance table is 5D in a general case or 4D for a scene containing only a single layer of PLS. [11] In either case, the table is too large to be contained in the operational

[10] Actually, if the goal is to compute a plane which size is equal to a spatially limited hologram, the number of considered PLS is N^2. This represents the extreme case.

[11] A 5D index consists of two indices of a target sample, three indices of a source PLS.

memory. Yet, when the Fresnel approximation is applied, the table can be reduced to a set of two 2D tables with a known maximum index [NSM+05] and these tables fit the memory. The resulting computation is about two times faster than a full evaluation of Eq. (3.7). Nevertheless, a side effect of the approximation is quantisation of PLS distribution.

More efficient solution is possible through HPO holograms and the bipolar intensity [Luc92]. For a given index pair (m, n), applying the bipolar intensity Eq. (3.6) to evaluation of the optical field Eq. (3.7) yields

$$I_{\mathrm{B}} = \sum_i \left(\Re\{s_i\}\mathrm{T_c}[\Delta_i, \bar{z}_{\mathbf{s}_i}] + \Im\{s_i\}\mathrm{T_s}[\Delta_i, \bar{z}_{\mathbf{s}_i}]\right),$$

where $\Delta_i = m - \lfloor \frac{1}{D} x_{\mathbf{s}_i} \rfloor$, $D = \frac{\pi}{x_{\mathbf{k}}}$, $\mathbf{k} = (x_{\mathbf{k}}, z_{\mathbf{k}})$ is a wavevector of the reference wave, and $\bar{z}_{\mathbf{s}_i}$ is a quantised Z-axis coordinate $z_{\mathbf{s}_i}$. The quantisation of the Z-axis coordinate can be non-linear and can depend on ability of the human visual system to distinguish between distances [Luc94].

Now, let us express the tables $\mathrm{T_c}$ and $\mathrm{T_s}$. Applying the quantisation step D, $x_{\mathbf{k}} \lfloor \frac{1}{D} x_{\mathbf{s}_i} \rfloor$ becomes equal to $c2\pi$, $c \in \mathbb{Z}$. As a consequence, phase $\varphi_r(mD_x)$ of the reference wave becomes $\varphi_r(mD_x) = x_{\mathbf{k}} mD_x - c2\pi = \varphi_r(\Delta_i)$. This allows to express the table $\mathrm{T_c}$ as $\mathrm{T_c}[\Delta, z] = \frac{1}{r}\cos(kr - x_{\mathbf{k}_r}\Delta)$ where r is a distance between the sample and PLS. The table $\mathrm{T_s}$ follows the same consideration. The only difference between the tables is that the table $\mathrm{T_s}$ uses the sine function instead of the cosine function. Despite the fact that this simplifies the calculation significantly, none of purely table-based techniques is able to remove the necessity to access and process every sample of the optical field. Since the calculation is simple, the time spend on data transfers between the tables and the optical field influences the resulting calculation time significantly.

Since the optical field calculated from a cloud of PLS is superposition as shown in Eq. (3.7), it is possible to exploit a hardware for accelerating purposes. The calculated optical field can be divided into tiles and each tile is processed separately in a parallel or a distributed environment [NSM+05]. The texture mapping ability of graphical processing unit (GPU) [RBD+99, PM03] or programmable components of GPU [MIT+06, ABMW06] can speed up the computation as well. Thanks to massive parallelism of GPU, an optical field of 2×10^6 samples can be generated with interactive rate if the number of PLS is kept small, e.g., 10^3 PLS [ABMW06].[12] More significant speedup can be achieved by applying a pipeline architecture implemented on a programmable hardware (FPGA) [IMY+05]. The solution uses the Fresnel approximation in a form of a differential scheme and it omits individual intensities of PLS. The solution is able to calculate an optical field of 1.5×10^6 samples from a cloud of 10^4 PLS in the real-time rate.[13]

Hardware-based acceleration techniques provide a speedup that is almost linear and the same time the the computation time depends linearly on a number of PLS. Since number of PLS increases dramatically with increasing complexity of a scene, the techniques cannot be used alone and they have to be combined with other techniques to became applicable. Also, none of them includes visibility solution. If they are capable of applying visibility, the visibility has to be pre-computed before the calculation begins. Besides that, in some cases the techniques limit variability of PLS in the cloud, e.g., all PLS has to have the same intensity [IMY+05] or the same phase [ABMW06] or both [RBD+99, PM03, MIT+06]. This

[12] Interactive rate means a framerate equal or greater than 1 frame per second.

[13] The real-time rate means a framerate equal or greater than 25 frames per second.

renders majority of available hardware-based acceleration techniques rather useless except for a quick and low-resolution previews.[14]

Let us now discuss generation of a cloud of PLS. The cloud can be generated from a triangular mesh using a ray-casting [Und97]. Visibility is calculated by a ray-casting using the original triangular mesh as in Sec. 3.1.1. Unlike a standard ray-casting that usually considers a low number of viewers, on a hologram each sample is a viewer that gathers rays from the scene. This significantly increases the computation time. In order to improve efficiency, it is possible to apply a triangle culling in a limited manner. For that purpose, a triangle is tested whether it is culled for a viewer located in a corner of a rectangular envelope of optical field samples.[15] A triangle that is culled from all four corners is considered completely invisible. If applicable, this step is performed on a 2D slice of the scene [Und97]. Another option is to approximate visibility through a lower resolution [ZCG08]. Visibility is examined from selected samples and obtained information is shared among neighbours. In an extreme case, visibility can be solved only for an orthogonal projection of the scene onto the hologram plane $\kappa : z = 0$ [KYY08]. This approximation is considered valid when a diagonal of the rectangular envelope is much smaller than a distance of the nearest PLS to the hologram plane κ.[16]

Cloud of line segments

The major drawback of techniques that rely on a cloud of PLS is a number of PLS. If a solid surface is processed, the number of PLS is high. In order to reduce the resulting computational time, it is appropriate to calculate at once as many PLS as possible. This can be achieved by using a line instead of PLS. A single line replaces many PLS and an optical field of a line can be approximated by a function.

Similar to the previous case, the samples of an optical field are located on the plane $\kappa : z = 0$. Since a rotation of the source about the Z-axis equals to a rotation of the optical field about the Z-axis, only a line $l : z = y \tan\gamma$, where the angle $\gamma \in [0, 2\pi)$, has to be considered. Let us assume that all point of the line l emits light with the same phase and the same intensity. When the Fresnel approximation is applied [FLB86], an optical field of the line l is

$$u_{mn} = \exp\left[\mathrm{j}k\frac{(nD_y)^2}{2z_\mathrm{l}}\right], \tag{3.8}$$

where z_l is an orthogonal distance between the line and the plane κ as depicted in Fig. 3.8. First component of the Fresnel approximation is omitted from Eq. (3.8) because it is constant.

An optical field U_s of a line segment l_s can be approximated by a rectangular part of an optical field U_l generated by an infinite line as illustrated with Fig. 3.9. A height of the rectangle equals to a length a of an orthogonal projection of the line segment l_s into the plane κ. A width of the rectangle is limited only by the size of the optical field U_s. This approximation causes a blur of endpoints in the reconstruction. If the length a is large, the blur is not disturbing and can be ignored. However, a smaller length a such that $a \sim D_y c$

[14] As illustrated with Fig. 3.2, ability to control the phase almost arbitrarily is necessary for a hologram viewable by a human viewer.

[15] The test compares a normal vector of a triangle with a vector connecting triangle vertices and the viewer. If the comparison fails for all three vertices, the triangle is considered as culled for the viewer located at the corner.

[16] In fact, this case is valid for almost all SLM currently available and authors seems to enjoy this technical limitation.

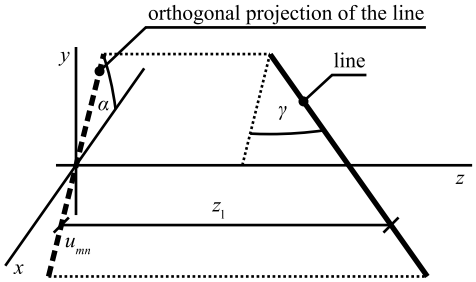


Figure 3.8: A part of an infinite line. An optical field of a line where $\alpha \neq 0$ can be calculated by rotating an optical field of a line where $\alpha = 0$. [FLB86]

where $c < 10^1$ causes a blur that is much stronger than the reconstructed line segment and the line segment is not visible in the reconstruction.

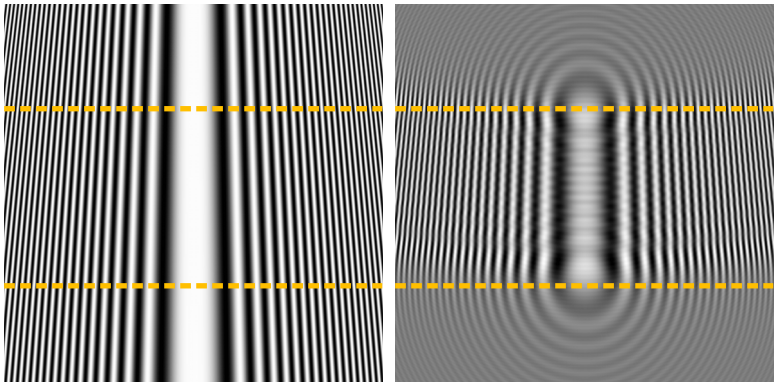


Figure 3.9: (a) A real part of an optical field generated by an infinite line and (b) a real part of an optical field generated by a line segment. The dashed rectangle specifies an area that is considered by the approximation.

Similar to the line segment, an optical field of a simple parametric curve depicted in Fig. 3.10 can be approximated by a function [BFJ+90]. In cylindrical coordinate system a curve on a cylinder of radius b is defined as

$$\rho = b, \ \beta = t, \ z = f(t),$$

where $t \in [0, 2\pi)$ is the parameter. If z fulfils requirements for the Fresnel approximation and function $f(t)$ is slowly varying, the optical field generated by the curve can be approximated by

$$u_{mn} \approx \exp\left\{ \mathrm{j}k \frac{[(m^2 D_x^2 + n^2 D_y^2)^{1/2} - b]^2}{2f(\beta)} \right\}, \ \beta = \arctan \frac{nD_y}{mD_x} \tag{3.9}$$

Similar to the line segment, the amplitude and the phase is constant over the curve. As it is shown in [BFJ+90], points of the curve that are in focus are reconstructed much brighter than the background and thus the reconstruction is successful.

Methods using a line segment are faster then methods using PLS because a single line segment is equal to a multiple PLS. Similar to PLS, it is possible to decrease computation time by exploiting GPU [RBD+99]. Nevertheless, unlike PLS, a line segment has a complicated visibility and an efficient hidden-line removal algorithm that would consider multiple viewers

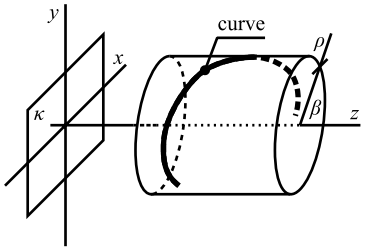


Figure 3.10: A curve on a surface of a cylinder and the plane κ. [BFJ+90]

at the same time does not currently exist. Therefore, the visibility is either ignored completely or it is faked using a visibility of an orthogonal projection of the scene. Also, necessity of a constant amplitude and a constant phase over a line segment harms visual quality of the reconstruction.[17] For that reasons, usage of line segments is suitable only for generating of quick previews.

Cloud of triangles

A number of at once processed PLS can be further increased through a surface, i.e., a triangle. The major drawback of such a solution is a complicated visibility and a complicated optical field. Even a triangle with a constant phase and a constant amplitude generates an optical field that is much more complex than an optical field of PLS. The visibility is usually not solved at all or it is approximated either by a triangle culling [ABMW08] or by using an estimation based on a ray-casting [KHL08]. A proper solution of the visibility requires a masking in a spatial domain. This slows down the computation as described in Sec. 3.1.2.

In a general case an optical field of a triangle cannot be expressed by a function. This can be overcome through a table of pre-calculated optical fields. As a consequence the resulting optical field is a weighted sum of appropriate table entries. Before summing, a retrieved entry is rotated about the Z-axis and translated to the final location in the XY-plane. The sum can be accelerated by GPU [KDS99, KDS01].[18] However, the number of indices of the table is the major concern of the approach because a triangles has much more parameters than a line segment or a point. Variability of triangles can be reduced by keeping the size and the shape of all triangles the same and by use of a constant shading, i.e., the amplitude over the triangle is constant. Still, the table has to be accessed by three indices: an angle of rotation about the X-axis, an angle of rotation about the Y-axis, and an orthogonal distance of the triangle to the plane $\kappa : z = 0$ as illustrated with Fig. 3.11. Each entry in the table is an optical field of a triangle. Therefore, the method requires large amount of memory and it is useless without efficient compression of the field.

If both the phase and the amplitude of a triangle are constant, the angular spectrum of such a triangle can be described by a function [ABMW08, KHL08]. The angular spectrum is expressed on a plane τ that is defined by the triangle. Rotation, translation, and propagation of the angular spectrum is handled by wave-based approaches mentioned in Sec. 3.1.2. The

[17] In fact, a constant phase over the line leads to an optical field that might not be suitable for the human observer because such line is an equivalent to a thin slit (i.e., a thin rectangular opening in an opaque screen) lit by a plane wave. While the observer is able to recognise the distance of edges from her/him, the depth of the centre is undetectable because the viewer sees the source, i.e., PLS in infinity.

[18] Papers [KDS99, KDS01] shows that results can be recycled easily without almost any additional effort if you have enough time. At the end, it is only a score that matters in the academic world.

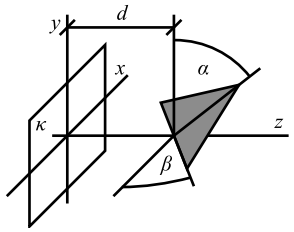


Figure 3.11: A triangle in a table is accessed through three indices: a rotation angle α about the X-axis, a rotation angle β about the X-axis, and an orthogonal distance d between the triangle and the plane κ.

triangle in the plane τ is defined as an amplitude coefficient

$$a_T(x_\tau, y_\tau) = \begin{cases} 1, & \text{inside the triangle } T \\ 0, & \text{outside the triangle } T, \end{cases}$$

where (x_τ, y_τ) is a coordinate in a local coordinate system on the plane τ.

Let us assume two triangles T and $\bar{T}$. If both triangles are related through an affine transformation, an angular spectrum $\mathcal{A}_T$ of the triangle T can be expressed using an angular spectrum $\mathcal{A}_{\bar{T}}$ of the triangle $\bar{T}$ [ABMW08]. Now, let us assume that the triangle $\bar{T}$ is a right triangle depicted in Fig. 3.12(a). Its angular spectrum $\mathcal{A}_{\bar{T}}$ can be derived analytically [ABMW08] and it is

$$\begin{aligned} \mathcal{A}_{\bar{T}}(u,v) &= \int_0^1 \int_0^x \exp[-\mathrm{j}2\pi(xu+yv)]\,\mathrm{d}y\,\mathrm{d}x \\ &= \begin{cases} \frac{1}{2}, & u=0, v=0 \\ \frac{1-\exp(-\mathrm{j}2\pi v)}{(2\pi v)^2} - \frac{\mathrm{j}}{2\pi v}, & u=0, v\neq 0 \\ \frac{\exp(-\mathrm{j}2\pi u)-1}{(2\pi u)^2} - \frac{\mathrm{j}\exp(-\mathrm{j}2\pi u)}{(2\pi u)^2}, & u\neq 0, v=0 \\ \frac{1-\exp(-\mathrm{j}2\pi v)}{(2\pi v)^2} + \frac{\mathrm{j}}{2\pi v}, & u=-v, v\neq 0 \\ \frac{\exp(-\mathrm{j}2\pi u)-1}{(2\pi)^2 uv} - \frac{1-\exp[-\mathrm{j}2\pi(u+v)]}{(2\pi)^2 v(u+v)}. & \text{otherwise} \end{cases} \end{aligned}$$

Yet, it is not the only triangle that can be expressed analytically. Another option is to use a rotation of a general triangle in the plane ρ [KHL08]. If the general triangle is rotated as illustrated with Fig. 3.12(b), its angular spectrum is

$$\mathcal{A}_{\bar{T}}(u,v) = \begin{cases} \frac{(a+b)c}{2}, & u=0, v=0 \\ \left(\frac{a+b}{c}\right)\exp(-\mathrm{j}2\pi vc)\left[-\frac{1+(\mathrm{j}2\pi vc-1)\exp(\mathrm{j}2\pi vc)}{(2\pi v)^2}\right], & u=0, v\neq 0 \\ \left(\frac{-\mathrm{j}c}{2\pi u}\right)\left\{\exp[\mathrm{j}\pi(3ua+vc)]\mathrm{sinc}(ua+vc) - \frac{\mathrm{sinc}(ub-vc)}{\exp[\mathrm{j}\pi(ub+vc)]}\right\}, & \text{otherwise} \end{cases} \quad (3.10)$$

where a, b, and c are defined by the rotated triangle, see Fig. 3.12(b).

Notice that in either case, the triangle has a constant phase over the surface. Arbitrary phase distribution on a triangle can be handled by dividing the original triangle into elementary triangles [KHL08] as illustrated with Fig. 3.13. The elementary triangle has the same shape as the original triangle but it is smaller. The division scheme requires two elementary triangle: an elementary triangle and its copy turned 180° about the Z-axis. Elementary triangles have a constant phase and a constant amplitude along the surface and therefore an

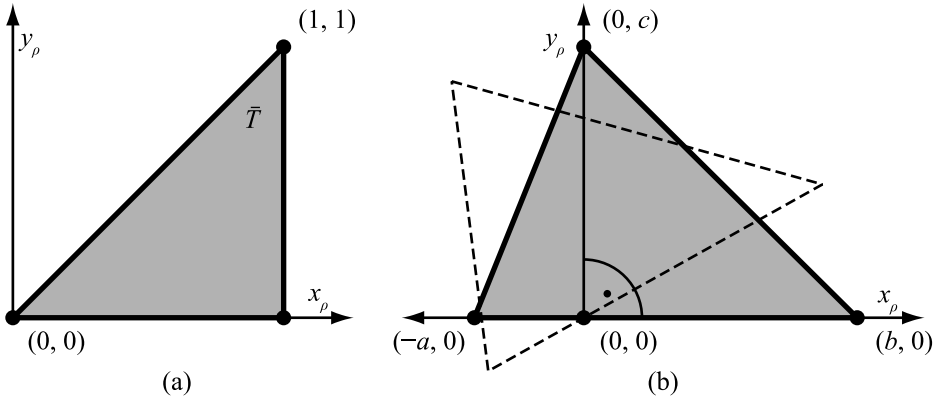


Figure 3.12: Triangle shapes that allow to express the angular spectrum analytically. An angular spectrum of a general triangle can be derived from (a) a special case triangle $\bar{T}$ [ABMW08] or (b) can be expressed directly using a properly rotated triangle [KHL08]. Dashed line in (b) shows a general triangle prior rotation.

optical field of the elementary triangles can be expressed using Eq. (3.10).[19] A combination of elementary triangles, each of a different phase, creates a desired phase distribution. Increasing fineness of the division increases variability of the phase and as a consequence it increases a viewing angle of the original tringle. Nevertheless, reaching the phase variation that is available in wave-based methods is not possible without a significant increase of the computation time due to a far too fine division scheme.

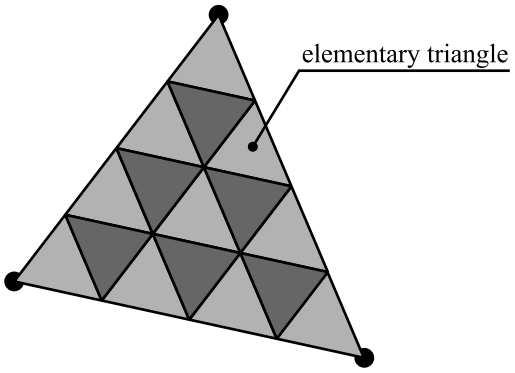


Figure 3.13: A triangle divided into elementary triangles. The division scheme requires two versions of the elementary triangle that are marked by lighter and darker gray. [KHL08]

A triangle is more complicated structure than a line segment and PLS. This implies complicated optical field that cannot be expressed analytically in general case. A table of triangles is not efficient because it limits possible configurations of triangles and it has a large memory footprint. An analytically expressed angular spectrum of a triangle with a constant phase and a constant amplitude is questionable because a shape with a constant phase is almost undetectable by the human viewer [LHJ68]. Even though combination of triangles might create a phase variation [KHL08], the number of such triangles has to be high in order to become comparable to basic wave-based methods described in Sec. 3.1.2. On the other hand, an analytically expressed angular spectrum allows to calculate a rotated angular

[19] Actually, only an angular spectrum of the first elementary triangle is necessary. The other angular spectrum is calculated by a rotation since the second elementary triangle is the first one rotated 180° about the Z-axis.

spectrum both accurately and without periodicity enforced by the discrete Fourier transform. This makes the analytical expression a tool that improves accuracy rather than performance.

3.2.3 Simplification of an optical field

Besides a scene it is possible to find a different encoding of an optical field to reduce the calculation time. Usually, an optical field is represented as a system of spatially limited functions with weights. In some cases, the encoding forces a simplification of the scene as well. As a consequence of a different encoding, each element in the scene requires a lower number of write operations and thus it is calculated faster.

The method described in [Luc92, Luc93, Luc94, Ple03] uses a different representation of a hologram to reduce cost of both storage and transportation. The representation was developed to reduce an data amount required for displaying the hologram using a MIT holovideo system [Luc97]. **The MIT holovideo** shares a similar design concept as a common CRT tube used by TV sets. Similar to the CRT tube, the device uses a time sharing to present the whole hologram. In a single time slot, it shoots a ray into a direction that corresponds to the time slot. The ray hits a digitally driven diffractive element. As a consequence, it is split and diffracted at the same time and resulting rays continue towards the viewer.

The input of the method is a cloud of PLS. The output suits the design of the device and therefore it is a hologram decomposed into independent diffractive elements knowns as hogels. **A hogel** is a 1D hologram [Luc92] calculated using the bipolar intensity from Eq. (3.6). Each PLS leads to an intersection of multiple rays as depicted in Fig. 3.14 and this is can be detected by the viewer.

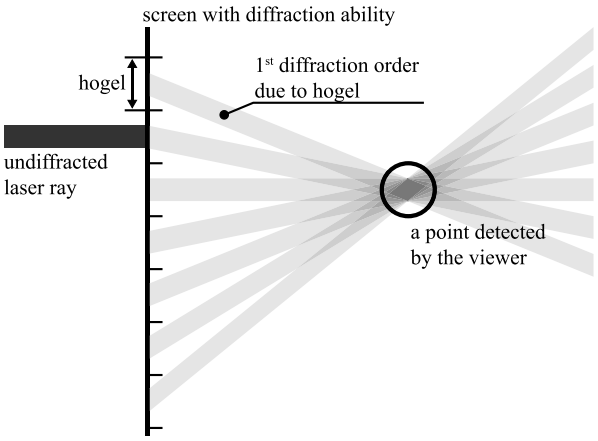


Figure 3.14: Relation between hogels and PLS. [Luc94]

Furthermore, the method assumes a quantised space. As a consequence a location of each PLS is given by a vector of indices (o, p, q). This allows to construct a table B_{mnopq} that converts PLS location indices (o, p, q) to an update of a hogel h_{mn}. This leads to an algorithm described by 1.

A disadvantage of the algorithm 1 is the rank of the table B_{mnopq}. Nevertheless, the rank can be reduced by restricting the scene and the hologram [Luc93]. The first reduction occurs if only HPO holograms are considered. This allows to remove the Y-axis coordinate from the table thus to reduce the table B_{mnopq} to a table B_{moq}. Since the hogel is calculated using the bipolar intensity and the reference wave is planar, only a relative position of the hogel and

Algorithm 1 The core algorithm of the method presented in [Luc92].

Let hogel h_{mn} be zero
for all PLS p from a cloud of PLS **do**
  Let o be a rounded X-axis coordinate $x_p^{\mathbf{x}}$ of a point p
  Let p be a rounded Y-axis coordinate $y_p^{\mathbf{x}}$ of a point p
  Let q be a rounded Z-axis coordinate $z_p^{\mathbf{x}}$ of a point p
  $h_{mn} = h_{mn} + a_p B_{mnopq}$
end for

PLS is relevant. This reduces the table B_{moq} to the table $B_{\Delta_m q}$, where $\Delta_m = |m - o|$. The resulting table $B_{\Delta_m q}$ does not depend on absolute position m within the n-th row and it is reused by all hogels.

Furthermore, a hogel can be encoded as a hogel vector to reduce storage and calculation requirements [LG95, Luc96]. **The hogel vector** $\bar{\mathbf{h}}_{mn}$ is a vector of weights such that a hogel $h_{mn} = \sum_i \bar{h}^i_{mn} \mathbf{b}_i$, where $\mathbf{b}_i$ is a precomputed diffraction structure known as a basic fringe. **The basic fringe** is a hogel that diffracts a ray to a selected range of directions.

Since a number of basic fringes is lower than the total number of all possible directions, a representation that is based on hogel vectors reduces a size of the hologram and the computation time. As a consequence, the method is able to generate, transfer, and display a single hologram of 10^5 PLS in circa 10^2 s without any hardware acceleration. Thus, a hardware-aided solution may allow a real-time rendering of such a cloud. Nevertheless, the drawback of the representation is a blur of the reconstructions [Luc96] caused by a low number of basic fringes.

Besides that, the method allows incremental updates [Ple03] that increase an interactivity of the display. The generating engine is aware of currently displayed cloud of PLS. When a change occurs, the engine computes an update 1D hologram from a current version of modified PLS. Then, it subtracts the update hologram from the current hologram, i.e., it removes modified PLS. After that it calculates a modifying hologram from a modified PLS and adds the modifying hologram to the current hologram. The efficiency of the approach depends on a number of affected hologram rows and therefore the overall speedup might not be significant for all cases.

A full-parallax solution utilizing the simplification of an optical field is possible. It benefits form a properties of neighbourhood of an optical field sample [LBU04, KYY08]. In a small neighbourhood of a sample u_{mn}, the distance r_{mni} [Eq. (3.7)] between the sample u_{mn} and PLS s_i can be approximated by a linear function. The linearity in a close neighbourhood means that a wavefront of a spherical wave emitted by PLS can be approximated by a set of planar waves as illustrated with Fig. 3.15. Since the angular spectrum of a plane wave consist of a single non-zero frequency, only a single frequency of the neighbourhood angular spectrum has to be updated instead of updating all samples of the neighbourhood in the spatial domain. A phase of a contribution to the frequency is kr_{mni}, where k is a wavenumber. After all contributions to the neighbourhood are collected, the optical field of the neighbourhood is obtained by applying the Fourier transform.

The approximation blurs PLS in all directions equally. Also, it limits the scene spatially because the size of the neighbourhood is inversely proportional to the minimal acceptable distance between PLS and the hologram. If FFT is used to convert the angular spectrum into the optical field, the desired frequency is rounded to the nearest available frequency.

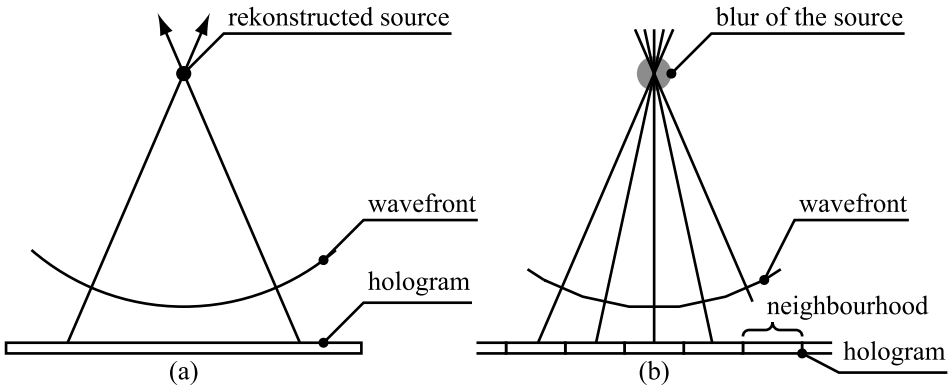


Figure 3.15: (a) A wavefront generated by PLS and (b) a wavefront approximated by a linear function. Notice that (b) is a almost ideal wavefront that lacks all disturbances due to discontinuities between neighbourhoods. [KYY08]

This causes additional blur. Nevertheless, thanks to that the method significantly reduces the computation time of a hologram.

3.3 Summary

In the previous section, we gave a brief description of methods and acceleration approaches. Before we proceed to the major contribution of this work, let us summarise features of methods and approach from a perspective of this work.

3.3.1 Methods

Every method that generates a hologram has its benefits and drawbacks The wave-based methods are generally fast because they use a propagation in an angular spectrum that is implemented through FFT. However, they are not able to solve a visibility efficiently. If speed is preferred to accuracy, the visibility can be solved using an orthogonal projection. Scene elements that are not visible, are simply removed. Nevertheless, this works only for smaller scenes that are further away from the hologram. In a general case, the solution of visibility is done in a spatial domain and therefore FFT has to be executed twice per an element of a scene.

Besides that, the wave-based methods requires that the scene is composed of planar elements, i.e, they are not able to handle curvy surfaces. Also, acceleration in a parallel or a discrete environment is not efficient because only a single propagation can be parallelised, the rest of the algorithm is sequential. Furthermore, the whole optical field has to fit into the memory for efficient computing of FFT, otherwise the computation time is significantly increased due to frequent exchanging of data between the memory and the external memory.

Contrary to that, the methods based on a cloud of PLS operates strictly in the spatial domain and therefore they are able to use a ray-tracing, which is a well developed technique, for solving of the visibility. Unlike the wave-based methods, the PLS-based methods are able to handle curvy surface. Their performance can be boosted easily using a parallel or a distributed environment. In such a case, the gained speedup is almost linear. Also, they can be implemented in a custom hardware that boosts their performance even further. Despite

that, these methods are much slower than the wave-based methods in general case. This is caused by a high number of PLS they have to process in order to obtain a comparable reconstruction.[20]

The view-based methods differs from previous ones. They use multiple views of the scene without a depth information. Hence they have potential to create holograms of a real-world scenes. However, the number of view is high and therefore these methods have similar performance as the PLS-based methods.

Individual groups of methods are well developed. Almost every aspect of methods has been already discussed in papers. As a consequence there is a low probability of a new significant contribution to the field. However, a combination of these method groups has not been discussed properly yet. Therefore, we examine this is area the following chapter.

3.3.2 Acceleration

Acceleration approaches that we discussed in the previous section can be applied almost to any method with only a few restriction. When the vertical parallax is omitted, the computational time is decreased significantly. However the resulting HPO hologram requires a complicated reconstruction setup and it cannot be converted to a hologram that is compatible with a full parallax hologram reconstruction setup. On the other hand, the bipolar intensity almost halves the computation time but the resulting hologram is tailored for a given reconstruction wave. Since both acceleration approaches can be applied without a significant reformulation of a given method, they are not considered primarily in this work.

Similar to the previously mentioned approaches, the simplification of the scene is simple and quite efficient if applied properly. The most accurate replacement of the scene is a properly generated cloud of PLS because such approach allows to handle almost any surface and a single PLS is processed quickly. However, the result is a dense cloud of PLS. On the other hand, if a triangle is used instead of PLS, the number of triangles is low. Unfortunately, the triangle is far too complex to processed quickly. Therefore, we try to search for a primitive that combines features of both a triangle and PLS in this work.

A simplification of an optical field is also an efficient tool for acceleration. A special solution based on a set of basic function has already been widely discussed in the papers. A solution that uses a linear approximation of a function has already been discussed too. Thus, there could be a low probability of a new signification contribution and therefore we do not focus on employing a simplification of the optical field in this work.

[20] This is especially true when the scene contains solid surfaces.

Chapter 4

Detail Driven Generation

This chapter contains a description of a method that is the major contribution of this thesis. The method shows that a combination between different principles is possible and that such a combination yields both a working and a fast method to hologram generation.

First, we describe the basic method. This will show the principles that are used. Then, we describe accelerations of the basic method. This will show that the method has a potential to be accelerated algorithmically. And finally, we present enhancements of the method. This will show that the method can be expanded and therefore it does not represent a dead-end direction.

4.1 The Basic Method

In this section we present principles used by the proposed method. First, we present a mechanism that we applied. Then, we show results calculated with the proposed method and we compare the proposed method to other relevant methods.

The major problem that complicates the solution is occlusion. It is not possible to omit occlusion because it is crucial to visual impression of the hologram content. Using a PLS-based approach, the occlusion can easily be solved though ray-casting [Und97]. The drawback, however, is that the scene contains a large number of PLS especially when a solid surface is to be encoded. On the other hand, a number of scene elements is low when a FFT-based approach is used. The FFT-based approaches works mostly with the angular spectrum, i.e., most of operations are done in the frequency domain. Unfortunately, the occlusion has to be solved in the spatial domain and this complicates the method and slows it significantly down. We noticed this fact and as a consequence we designed a method that combines both a PLS-based approach and a FFT-based approach.

We started our exploration from PLS. PLS generates spherical wave defined in Eq. (2.9) and occlusion can be approximated by rays [Luc94, Und97], i.e., a geometrical shadow of an occluder is taken into account. Even though the geometrical shadow is in contradiction to the physical experiments, the resulting approximation works. A propagation of the angular spectrum, which is used by FFT-based methods, requires planes, in ideal case parallel ones. PLS can be location within such a plane and therefore it is possible to use the propagation of the angular spectrum to calculate the optical field generated by PLS. Application of the geometrical shadow then becomes a piece-wise multiplication of a bit map (i.e., the shadow) with the calculated optical field values.

In the continuous domain, the above mentioned construction is valid since the angular spectrum is an exact description of an optical field [BW05]. In the discrete environment, however, the angular spectrum does not represent the optical field appropriately due to the discrete Fourier transform (DFT). DFT assumes periodicity of the input signal by default. As a consequence, the resulting optical field of PLS differs from the optical field calculated using Eq. (2.9), i.e., the resulting optical field is rather an estimation.

Let us now try to apply the shadow to the estimated optical field. We would like to know whether PLS can be reconstructed even in such a case and whether the shadow will behave similarly to a case that uses Eq. (2.9). The first question is easy. The shadow divides the field to two: the occluded part and the visible part. Since the sum of both fields is the resulting field, we can propagate the both fields separately and sum the result. And considering the fact that in the hologram (and hence the optical field) information about the source is distributed to almost every sample, PLS can be reconstruction even if one part is only considered.[1] Hence, we can reconstruct PLS using only the visible part of the optical field.

The second question is more complicated. In order to find an argument supporting our goal, we did numerical experiments with two optical fields of a single PLS. We calculated the first field using the spherical wave Eq. (2.9) and the second field using the propagation of the angular spectrum. In both cases we assumed a sampling step 0.5 μm, a resolution of $2,048 \times 2,048$ samples and a PLS distance of 1.2 mm.[2] We assumed a planar occluder at the distance of 0.6 mm. We applied the shadow of the occluder to both fields and reconstructed both fields at the occluder distance.

The reconstructions depicted in Fig. 4.1(a) and Fig. 4.1(b) were similar. The only difference were numerous copies in Fig. 4.1(b) superimposed on each other. We assumed that these copies were caused by periodicity of FFT applied for the propagation. We verified this assumption by two experiments. In the first experiment we padded the samples by a zero frame to obtain a grid of $4,096 \times 4,096$ samples. In the second experiment, we just increased the resolution to $4,096 \times 4,096$ samples and calculate the field using the propagation. We reconstructed both fields and observe a region that corresponds to the original size.

The results of the first experiment shows that the copies are easily recognisable because they further away from each other as depicted in Fig. 4.1(c). This is caused by the additional frame of zeros. In the case of the second experiment, which is depicted in Fig. 4.1(d), the copies are missing completely. This, however, is caused by the fact that PLS is too close to contribute to a neighbouring copy due to the diffraction condition Eq. (2.24). Therefore, the neighbouring copies are not superimposed into the reconstruction as they do in Fig. 4.1(b).

Even though increasing of the resolution at the generation time led to a lack of the copies, we decided to avoid it because it is not practical. Since DFT accesses all values of the optical field, the values have to fit into the memory otherwise it will not be efficient, i.e., the memory footprint is significantly increased when PLS is moved further beyond the safe distance. On the other hand, the first case implies that the geometrical shadow will not disturb the region of interest when the optical reconstruction is done. During the optical reconstruction, the frame, which we enlarged in the simulation, can be considered infinite, i.e., the disturbing copies will be out of region of interest. Therefore, we decided to use this approach.

[1] This assumes that either part are large enough. Obviously, it is not possible to reconstruct anything from a single sample.

[2] PLS at that distance can be captured in the optical field without a risk of aliasing due to inappropriately low sampling rate, i.e., it is beyond the safe distance given by the diffraction condition from Eq. (2.24).

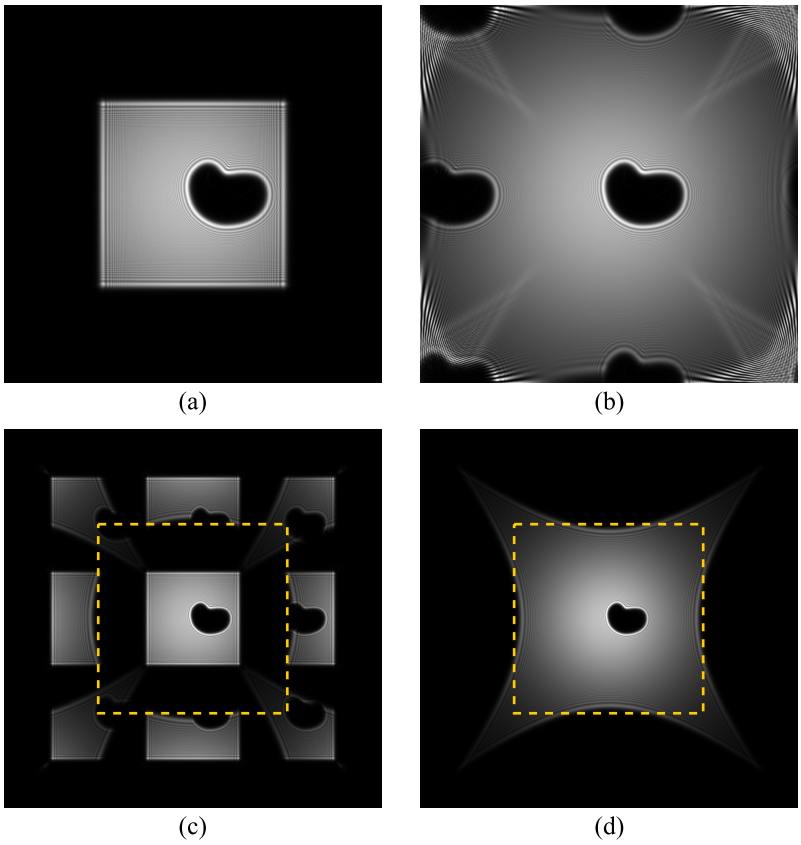


Figure 4.1: An effect of a geometrical shadow on PLS located at the distance 1.2 mm reconstructed numerically at the distance of 0.6 mm. (a) The input field was calculated using the spherical wave. (b) The other input was calculated using propagation of the angular spectrum and (c) the same input padded with a frame of zeros before reconstruction. (d) The input calculated using propagation of the angular spectrum and a larger resolution. The dashed rectangle in (c, d) corresponds to the resolution used in (a, b).

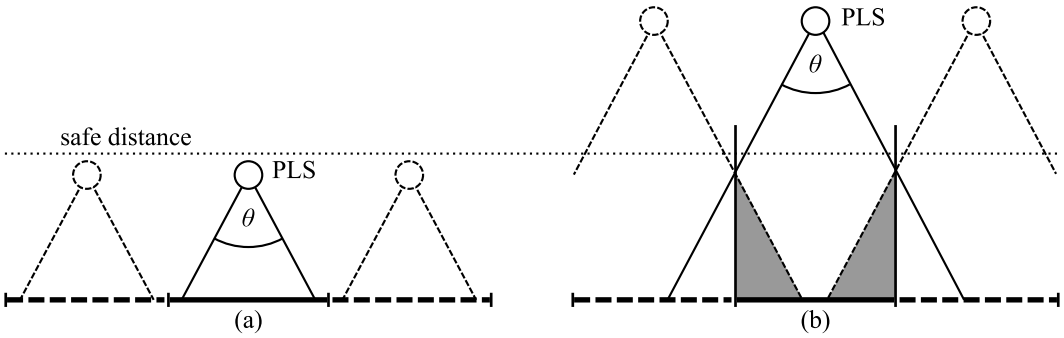


Figure 4.2: An influence of periodicity assumed by DFT on either PLS (a) closer than the safe distance and (b) beyond the safe distance. The angle theta is given by Eq. (2.24). The dashed lines indicates a side-effect due to the periodicity, the gray area in (b) shows overlapping of contributions.

Based on the text above, we can state that the propagation in the angular spectrum can be used as a replacement for evaluation of Eq. (2.9) even in the discrete environment.

Nevertheless, as it was stated above, PLS is too small to be used efficiently. Therefore, we use a planar patch instead of PLS and apply the geometrical shadow to the propagated patch.

This, however, is not a valid solution in a general case. A patch can be decomposed to a rectangular grid of PLS. Since each PLS has different spatial location, each PLS will use a slightly different shadow. If the same shadow is used for all PLS, a blur will appear. The larger the patch, the more blur will be present until the gap created by the geometrical shadow disappears. We validated this assumption by an experiment that considers a patch of a given size and a shadow of an occluder in a shape of a disc as illustrated with Fig. 4.3(a). We assumed that the shadow was calculated from the centre of the disc. We propagate the patch, applied the disc-like shadow and propagated a halfway back. In this case we used a resolution $1,024 \times 1,024$ samples and a sampling step 0.5 μm. The orthogonal distance of the patch to the optical field plane $\kappa : z = 0$ was 4.0 mm.

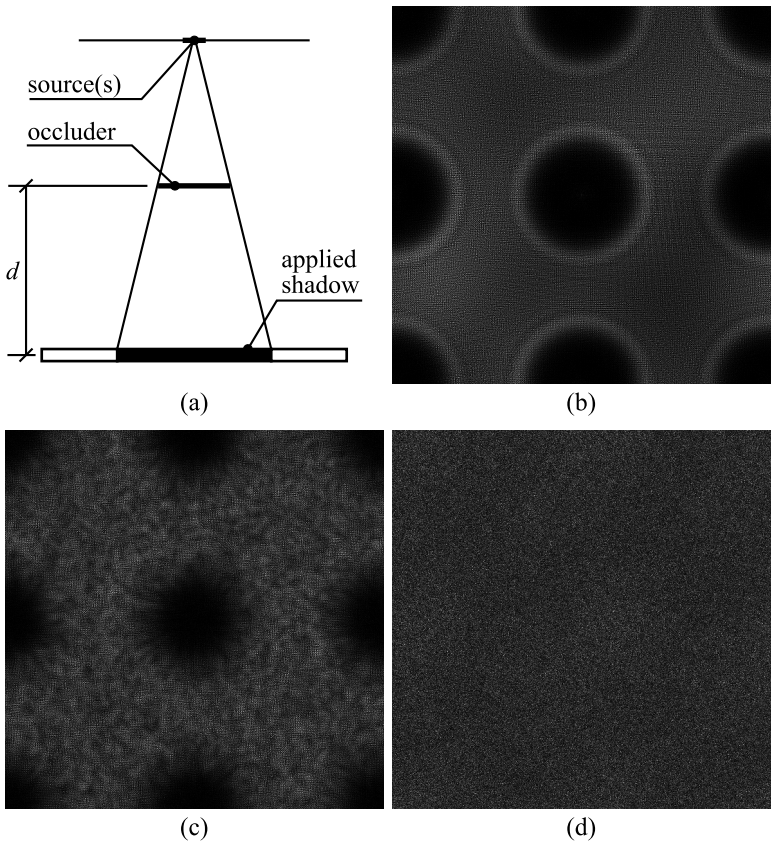


Figure 4.3: (a) The used setup and numerical reconstructions at the distance d. The source is a patch of (b) 16×16 PLS, (c) 256×256 PLS, and (d) $1,024 \times 1,024$ PLS.

As it can be seen in Fig. 4.3(b) a sufficiently small patch does not cause almost any blur unlike a case of a large patch, viz. Fig. 4.3(d). In such a case, the gap is almost lost in the blur. This means that we can use patches instead of PLS in our method but these patches has to be small enough.

Since now we know that we can combine PLS-based visibility approximation with propagation of the angular spectrum, we can specify our method more in a greater detail. Our method will calculate an approximation of a discrete optical field generated by the virtual scene. We assume that the optical field is intended for a human viewer. This implies that qualities of the surface and the physical size of the optical field. We define the discrete optical

field values U on the plane $\kappa : z = 0$ at points $\mathbf{u}_{mn}$ organised to a rectangular and uniform grid. A value at the point $\mathbf{u}_{mn}$ is u_{mn}. If not noted otherwise, we shall assume that the grid is spatially limited and contains $N \times M$ points. A pitch between points along the X-axis and the Y-axis is D_x and D_y respectively. Though the most of the text $N = M$ and $D_x = D_y$.

The virtual scene, which is the input of the method, consists of surfaces that are approximated by triangular meshes, i.e., it is fully compatible with scenes used in computer graphics. The mesh can be unclosed but it does not contain artifacts such as an edge shared by more than two triangles. Also, the scene does not contain intersecting meshes. The surface is solid, opaque, diffuse, and self-luminous, i.e., interaction of surfaces with each other is not considered.[3] All normals of the surface point consistently outward the objects. All triangles are located in a subspace defined by the plane κ and the positive Z-axis. No triangle touches the plane κ.

Let us now discuss occlusion. Even though it is possible to simulate an accurate solution of occlusion, our method approximates it. The reason is that such an accurate solution, which is illustrated in Fig. 4.4(a), is both unnecessarily slow and unnecessarily accurate.[4] If we omit the diffraction, we obtain an approximation that uses geometrical shadows and ray-casting as illustrated in Fig. 4.4(b) [Und97, JHO08]. Such an approximation does not have almost any significant impact on the visual quality of the reconstruction. However, it captures details that are comparable to a length of the sampling step. This is still too accurate if we consider a human viewer that is not able to perceive a detail in order of micrometers [Luc94]. Therefore, we can approximate the occlusion even further by undersampling it, i.e., the occlusion is solved in a much lower resolution that the optical field.

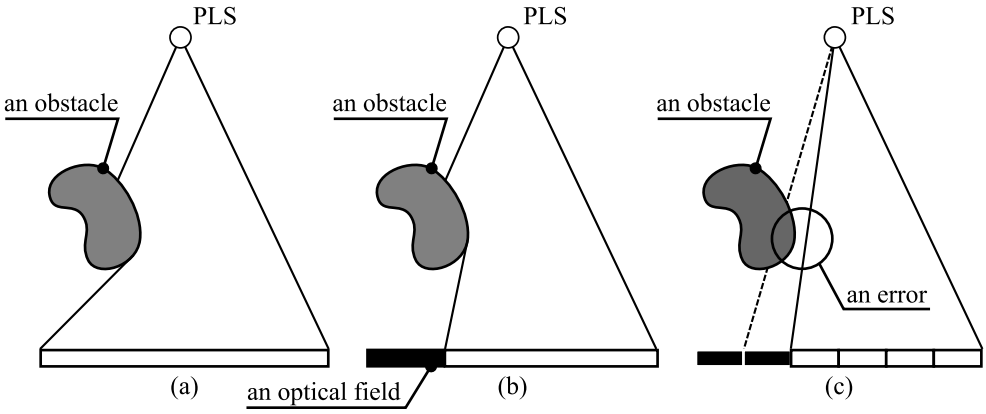


Figure 4.4: Occlusion of PLS due to an obstacle with a diffuse surface solved (a) without any approximation, (b) using a geometrical shadow, (c) using approximation applied in the detail driven method. Lighter parts of the optical field receive the contribution from PLS, darker parts are not influence by PLS. Notice that the example considers only the area of the optical field, waves that may reach the area outside the optical field are not considered.

For that purpose we cluster points of the grid to a coarser grid. We denote the grid as **the visibility grid**. The result of the visilibity/occlusion test is shared among all samples u_{mn} that belong to the cell g_{lo} of the visibility grid, where $l \in [-L/2, L/2-1]$, $o \in [-O/2, O/2-1]$, $L \propto M$, i.e., L is proportional to M, and $O \propto N$. Since no interpolation is applied to the

[3]This simplification is used by almost all authors and it is known as the source model. The accurate solution is known as the field model and it is much more computationally extensive. Despite inaccuracy of the source model, the resulting hologram works.

[4]An accurate solution requires a simulation of diffraction on the surface of the occluder.

result, there is an error as illustrated with Fig. 4.4(c). This error can be interpreted as a deformation of the obstacle and it can be neglected if the size of the patch is small, see Fig. 4.3.

The visibility approximation, however, leads only to a minor acceleration. Most of the time is spend on calculating the optical field generated by a element of the scene. This is caused by both a high number of samples that has to be calculated and a long computational time required by basic mathematical functions such as a square root, a sine and a cosine that are evaluated for each sample. We cannot accelerate the generation by reducing the number of samples because the number of samples has a direct impact on ability to capture the scene [KYY08]. Nevertheless, we can decrease the number of elements in the scene and improve the efficiency of their processing.

Since the scene consists of triangular meshes, the most efficient scene elements are triangles. However, each triangle has a different shape, a different size, a different rotation and a different phase/amplitude variation. These features complicates the computation since a rotation leads to resampling of the spectrum [Mat05, TB93, EO06] that inherently introduces a noise into the optical field. A solution might be to try to analytically express the angular spectrum of a triangle. This, however, is not possible due to phase variation that is required by a diffuse surface. Despite that fact that a mesh can be resampled to contain triangles of the same shape and the amplitude of a single triangle can be constant [5], we decided to ignore triangles and search for something in a halfway between a triangle and PLS.

In our method, we look at how a common 2D pictures are digitally stored. Instead of creating a high-definition vector description, a picture is composed of rectangular shapes, i.e., pixels. Even though, this limits the detail and might be memory inefficient, it can be processed easily. Following that, we replaced a general mesh with a cloud of patches. Every patch is parallel with the plane κ and it has a constant intensity as the pixel has. Due to a random variation of the phase, the patch cannot be expressed analytically. However, the numerical processing does not introduce any additional noise since there is no rotation. The size of the patch defines the smallest detail of the scene. In the following text, we assume that the size of a patch is comparable with a pixel size of contemporary LCD, i.e., 0.22 mm. We choose this size because even though the pixel can be recognised easily, it does not disturb viewer significantly when viewing images or playing games.

We specified that our method uses a cloud of patches that replaces meshes in the scene. Let us further limit the cloud. Following the analogy of a pixel-based image, a general location of the patch within the cloud is unnecessary. Therefore, we align patches to the visibility grid and we define that the size of a patch is equal to the size of a visibility grid cell. Furthermore, we limit the spatial extent of the cloud. Sice we plan to use the propagation of the angular spectrum, we shall use DFT. DFT assumes periodicity and therefore the patch has to be located completely inside a subspace defined by an axis aligned bounding box (AABB box) of grid points $\mathbf{u}_{mn}$ and the positive Z-axis.

Let us now summarise the proposed method. Our method replaces a mesh by a cloud of patches. Each patch is aligned to the cell g_{lo}. Since there can be many patches aligned to the cell g_{lo}, we denote d-th patch aligned to the cell as e^d_{lo}. The patch, then, is specified by an amplitude $a_{e^d_{lo}}$, an orthogonal distance $z_{e^d_{lo}}$ to the plane $\kappa : z = 0$ and a phase variation on the surface. A patch is sampled by $E \times E$ points and as a consequence the visibility grid contains $L \times O$ cells where $M = EL$ and $N = EO$. The output of our method is an optical field. The algorithm that describes our method is presented in Alg. 2. And we describe the

[5] This means flat shading [Wat00].

basic building blocks in steps 1, 3, 5–7 in the following subsections. Besides that we present results calculated by our method and we compare our method to others.

Algorithm 2 The core algorithm of the detail driven method.

```
1: Create patches
2: for all patches do
3:     Calculate an approximation of visibility.
4:     if the patch is visible then
5:         Calculate the optical field of the patch without visibility.
6:         Apply the visibility to the optical field of the patch.
7:         Add the optical field of the patch to the final optical field.
8:     end if
9: end for
```

4.1.1 Patch Generation

In this subsection we describe the process of patch generation from a triangular mesh, i.e., it is the step 1 from Alg. 2. We focus only on patch generation, other structures that are generated through the processing of the mesh are discussed in corresponding subsections.

In our method, we replace the mesh with patches aligned to a visibility grid, i.e., the mesh is resampled to patches. We use a ray-casting for this purpose [Wat00]. All rays are parallel and we shoot one ray par a cell. The ray is shot from the centre of a cell and it generates intersections h_i every time it intersects with the mesh. We evaluate a normal $\mathbf{n}_{h_i}$ of the surface at the intersection h_i and if the normal $\mathbf{n}_{h_i}$ points outwards the plane κ, we create a new patch, i.e., $d_{h_i} < 0$, where

$$d_{h_i} = \mathbf{n}_\kappa \cdot \mathbf{n}_{h_i}, \tag{4.1}$$

$\mathbf{n}_\kappa$ is a normal of the plane κ and $\cdot$ is a dot product.

We calculate an intersection using standard means of computer graphics [Wat00]. These are, however, singularities: a triangle that is perpendicular to the plane κ, an edge of a triangle and a vertex of a triangle. When a ray hits a triangle that is perpendicular to the plane κ, it generates almost an infinite number of intersections. In our case, the ray generates only two intersections: one at the entry and other on exit. Even though in both cases $d_{h_i} = 0$, we consider that $d_{h_i} > 0$ at the entry and $d_{h_i} < 0$ at the exit.

When a ray hits an edge that is shared by two triangles, it results to two intersections. Since all these intersections are calculated using the same equation and the same parameters, the distance z_{h_i} of the intersection h_i along the Z-axis equals each other and we can pick almost randomly one intersection and drop the other one. We apply this solution even in the case when d_{h_i} differs for any of triangles involved, i.e., the ray hits a silhouette of the mesh. Such an approach may significantly disturb the visual appearance only if it results to a large number of either scattered patches or missing patches. Since the patch is small, we assume that a group of neighbouring triangles that share the same result of Eq. (4.1) is large enough to prevent it. The rest of singularities such as intersecting meshes or an edge shared by more than two triangles are not considered because we assume that the scene does not contain such meshes. We use the approach described above to solve the intersection of a ray and a vertex too.

We use ray-casting to create patches. In our case, however, a general ray-casting in a 3D space is unnecessarily complex. Since many origins of rays share the same Y-axis coordinate, we can create a horizontal slice of the mesh and we solve the ray-casting on a 2D plane. The slice is an intersection of the scene with the plane $\rho_\xi : y = \xi$ and slices can be calculate using an efficient iterative algorithm [JHS07] because all origins of rays are uniformly distributed along the Y-axis.

Now, since we know intersections h_i that are valid in our case, we create a patch at each of those. We align the centre of the new patch to a location of the intersection. Other possibilities, which might be more sophisticated, such as selected a maximum or minimum from a close neighbourhood of intersection are not considered. Again, we assume the patch is too small so the this fine-tuning of the patch location will make only a little difference.

The amplitude a_{h_i} of the patch h_i is calculated using the cosine law [Pho75] because we assume a diffuse surface. In fact, we can use the complete Phong's lighting model [Pho75] or any other model to calculate the amplitude. In this case it is just matter of visual appearance and it has no influence on the ability of the hologram to recreate captured optical field. Even though the patch has a constant amplitude over its surface, it does not harm the visual appearance because, again, the patch is small.

On the other hand, the phase variation over the surface influences functionality of a hologram. We assume that the patch is self-luminous and hence the phase defined energy distribution.[6] A constant phase, which is inappropriately used by some authors [KHL08, ABMW08], allows an analytic expression of the angular spectrum [Goo05] but the result is not suitable for a human viewer [LHJ68]. In fact, a constant phase result to a set of apertures exposed to a plane wave. As a consequence, a human viewer is not able to detect the surface, which is, in fact, not existing, and sees only edges, which have low intensity.

We use a diffuse surface and thus the phase variation should causes the energy to be distributed uniformly to all directions available. Such an ideal case, however, does not have a solution except an ad-hoc iterative process [Luc94, WB89]. Since this problem is out of the scope of this work, we decided to use an approximation. We consider the phase as a pseudorandom function. As the result the energy is distributed to almost every direction even though the distribution is not uniform. This causes a speckle noise to appear on the surface of recorded objects. **The speckle noise** resembles tiny dots of high intensity whose configuration looks differently from every location, i.e., the object in the reconstruction seems to be glittering. Since this artifact does not deny the reconstruction of the surface, we does not address it and we ignore it.

The final algorithm that calculates the patches is described in 3. The input of the algorithm is a triangular mesh. The output is a cloud of patches whose centres are aligned with centres of visibility grid cells.

4.1.2 The Visibility Test

In this section we present a description of the visibility test that is used by our method. We define an auxiliary structure and we specify an algorithm that uses the structure to evaluate visibility of a patch and a cell.

[6] A patch that is self-luminous means that it emits the energy on its own and at the same time it is not influenced by other emitters.

Algorithm 3 The algorithm of patch generation.

```
1: Create the slice S_o, o = -O/2 of the scene using the plane ρ_η, η = -((O-1)/2) ED_y.
2: for all o, o ∈ [-O/2, O/2 - 1] do
3:     for all l, l ∈ [-L/2, L/2 - 1] do
4:         Shoot a ray r from (lD_x, 0) in a direction (0, 1) in the slice.
5:         Create a set H of intersections of the ray r and the slice S_o.
6:         Remove all intersections h_i where d_{h_i} > 0.
7:         Sort the set H according to the distance z_{h_i}.
8:         Let d = 0.
9:         for all i, h_i ∈ H do
10:            if z_{h_{i-1}} < z_{h_i} then
11:                Create a patch e^d_{lo}.
12:                Increment d.
13:            end if
14:        end for
15:    end for
16:    Create the next slice S_{o+1} using the slice S_o.
17: end for
```

Similar to PLS-based method, our method uses ray-casting to calculate the visibility test, i.e., it shots a ray from a patch towards a cell. If the ray intersects the mesh, the patch is considered occluded and does not contribute to the cell. Yet, use of the mesh for the test is unnecessarily accurate in our case because we approximate the visibility of the patch by a single ray. Therefore we propose a replacement of the mesh for the purpose of the visibility test.

We resample the mesh by pillars. A pillar is a cuboid segment of a space defined by the cell g_{lo} and the positive Z-axis. It contains a part of the volume defined by a mesh. Since we assume multiple meshes and concave shapes, we denote d-th pillar that corresponds to the cell g_{lo} as the pillar p^d_{lo}. A pillar is closely related to the intersections used in Sec. 4.1.1. The intersections define locations of both the front cap and the back cap of the cuboid segment along the Z-axis and the front cap shares the location with the patch. The pillars are generated during generation of patches. In fact, we modify Alg. 3 such that patches are generated after pillars were created. We create a patch e^d_{lo} at a front cap of each pillar p^d_{lo}.

Let us now describe generating of pillars. Following Sec. 4.1.1, we shoot rays into the scene but unlike it, we process each mesh of the scene separately first, i.e., for the mesh χ we calculate the set H_χ of intersections. Then, using Eq. (4.1), we distinguish front caps and back caps. We assume that a normal points outward the volume of an object. If $d_{h_i} \leq 0$, the intersection h_i will become a front cap. The rest of the intersections will become back caps. We sort the set H_χ according to the distances z_{h_i} of intersections ascendantly. If multiple intersections have the same Z-axis coordinate as a result of singularity, we assume that front caps are closer then back caps. After sorting, we expect that caps of both the same type and the same distance z_{h_i} are ordered randomly.

In an ideal case, the ordered set H_χ contains front caps succeeded by back caps, i.e., a sequence 'front-back'. Such a sequence defines a pillar. In a general case, however, the set H_χ contains invalid sequences of caps: 'front-front' and 'back-back'. This is a consequence of an unclosed mesh or a singularity, which we already discussed. We solve the 'front-front' case by adding a new back cap to the first front cap of the sequence. The new back cap is

created at the location of the front cap and it is shifted along the Z-axis by z_ε. The result is a valid sequence 'front-back-front'. The 'back-back' sequence is solved similarly. We add a new front cap, which is locate at first back cap of the sequence, and we shift it along the Z-axis by $-z_\varepsilon$. This leads to a valid sequence 'back-front-back'. All other attributes of these newly added caps are copied from caps that lead to their creation.

Since the major case of the invalid sequence is an unclosed mesh, we think of the constant z_ε as a thickness of the mesh surface. The mesh is infinitely thin in the ideal case and we exploit an analogical situation in the discrete environment. According to the diffraction condition from Eq. (2.24), the sampling step D_x and D_y defines a pyramid inside which we can obtain a contribution from PLS. If the sample is outside the pyramid, it will not obtain any contribution from PLS. Hence, a grid of samples close enough to PLS might receive a contribution only to a single sample, i.e., a propagation of PLS becomes phase shifting. Let z_ε by such a distance. As a consequence, if we had replaced both caps that are shifted from each other by z_ε for patches, both patches would have been in focus at the same distance. Thus, such a pillar is infinitely thin.

After the set H_χ contains pairs 'front-back', we create a set of pillars and we merge all sets of pillars to a single set P and we sort the set P according to Z-axis location of front caps ascendantly. The set P may contain overlapping pillars because we merged multiple sets together. Since we shall use the pillars to create patches, we have to merge overlapping pillars as illustrated in Fig. 4.5. We do that by iterating through the set P and applying rules depicted in Fig. 4.5. If two caps of the same type are in the same distance, we pick one of them randomly as we did in Sec. 4.1.1. The maximum number of iteration is equal to a size of the initial set P and in such a case the set P contains a single pillar at the end.

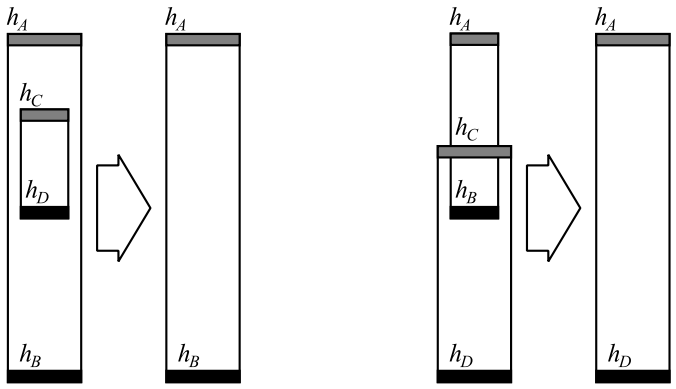


Figure 4.5: Invalid configurations of pillars and corresponding fixes. The black rectangles and gray rectangles denotes front caps and back caps respectively.

The resulting set P now contains mutually exclusive pillars p^d_{lo} and we can now create patches. The algorithm is summarised in Alg. 4 and it is able to solve all singularities that were mentioned in Sec. 4.1.1. The worst one is intersection of the ray r with a triangle that is perpendicular to the plane κ. As depicted in Fig. 4.6 the result is a single pillar.

The visibility test shots a ray from the centre of the patch to the centre of the cell. Since we replaced the scene with pillars, we can now exploit the regular organization the visibility grid. As a consequence, we are able to exclude pillars that can never be intersected by a ray from the visibility test. An orthogonal projection of ray into the plane κ is a line the connect two cells. If we use the visibility grid as a 2D raster as depicted in Fig. 4.7, only cells intersected by the line may contain intersected pillars.

Algorithm 4 The algorithm of patch generation including generation of pillars.

Create a slice S_o, $o = -\frac{O}{2}$ of the scene using the plane ρ_ξ, $\xi = -\frac{O-1}{2}ED_y$.
for all o, $o \in [-O/2, O/2-1]$ **do**
  for all l, $l \in [-L/2, L/2-1]$ **do**
    Shoot a ray r from $(lD_x, 0)$ in a direction $(0, 1)$ in the slice.
    Let the set P of pillars be empty.
    for all polygons of the slice S_o that belongs to a mesh χ **do**
      Create a set H_χ of intersections of the ray r and the polygons.
      Mark intersection either as a front cap or a back cap according to d_{h_i}.
      Sort the set H_χ according to the distance z_{h_i}.
      Inspect the set for both *front-front* and *back-back* sequences and fix them.
      Add pillars created from the set H_χ to the set P.
    end for
    Sort the set P according to the distance of front caps.
    Fix the set P following Fig. 4.5.
    Let $d = 0$.
    for all j, $p_j \in P$ **do**
      Let the pillar p_j became a pillar p^d_{lo}.
      Create a patch e^d_{lo} at the front cap of the pillar p^d_{lo}.
      Increment d.
    end for
  end for
  Create a new slice S_{o+1} using the slice S_o.
end for

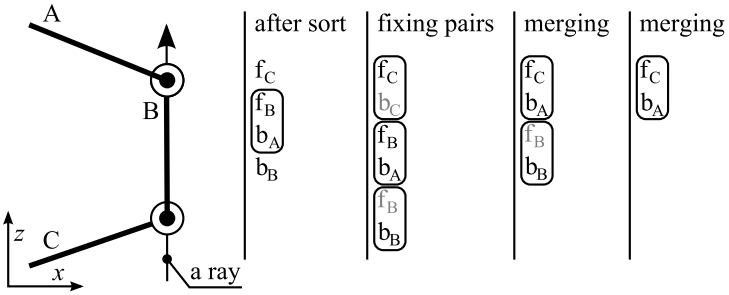


Figure 4.6: An example of a singular case solved by merging of pillars. The processed slice contains an edge from the triangle B that is perpendicular to the plane κ and the ray r hits the triangle B. Just for purpose of this example, a front cap and a back cap that were generated from the triangle B are denoted as f_B and b_B respectively. Greyed caps are added to fix missing member of a pair.

Finding the intersected cells is a problem that resembles closely a few problems solved by the computer graphics. The algorithm has to be able to find all intersected cells. This request is not compatible with a high-performance Bresenham's algorithm [FVDFH96] for 2D line resterization because the algorithm is not able to find all intersected cells, it finds only a subset. Besides that, the requested algorithm has to consider the fact that the number of parallel rays is low because the distance of patches is arbitrary. This request is not compatible with the shear-warp factorization algorithm [LL94] that is designed for a direct rendering of data stored in a uniform grid. Therefore, we decided to use the DDA algorithm [FTI86] that is designed to obtain all intersected cells.

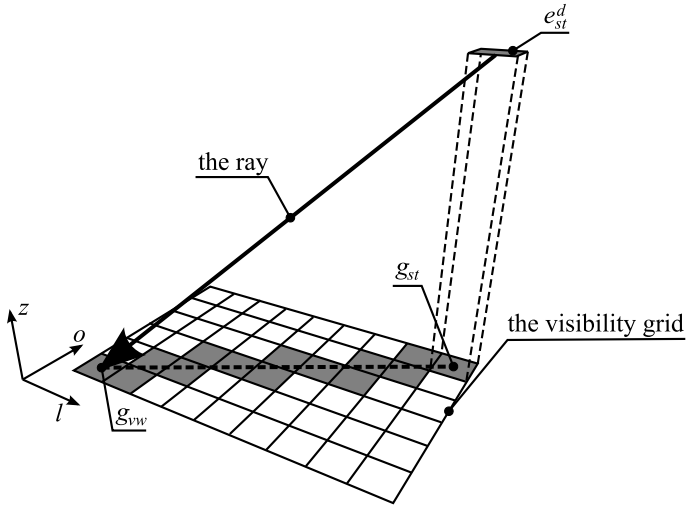


Figure 4.7: A visibility test between the patch e^d_{st} and the cell g_{vw}.

In order to explain the process, let g_{vw} and e^d_{st} be the cell and the patch respectively that we want to test. Using the DDA algorithm we find all possible index pairs (a, b) that identify the intersected cells as illustrated in Fig. 4.7. Since each pillar corresponds to a cell, the DDA algorithm identifies potentially intersected pillars as well. Therefore, we can extend the DDA algorithm to test all pillars p^d_{ab}. We calculate a depth at which the ray enters the cell and a depth at which the ray lefts the cell. This gives us a depth interval that is compared to all pillars p^d_{ab} corresponding to the cell g_{ab}. If the depth intervals are not mutually disjunctive, ray intersects a pillar and, consequently, the cell g_{vw} and the patch corresponding to the front facing cap of the pillar p^d_{qt} are mutually invisible.

The calculation of the depth interval corresponding to the cell g_{ab} exploits the linearity of the depth variation along the ray. As a result, for each cell g_{ab} we can evaluate the depth at the point where the projection of the ray enters the cell g_{ab} and the depth where the projection of the ray leaves the cell. Those two depth values determine the desired interval.

4.1.3 Computation of the Optical Field

In the previous section, we described the visibility test. In this section we shall apply it. This section closes the description of the proposed method. It contains a mathematical expressions used through the calculation. The section, however, is not necessary for understanding of the method and therefore it can be skipped.

In Sec. 4.1.2 we described the visibility test that evaluated a visibility of a patch e^d_{st} when viewed from a cell g_{vw}. We apply the test to calculate a visibility map of the patch. The visibility map T is a binary map. Each member t_{lo} of the map specifies wether the patch can be seen from the cell g_{lo}, i.e., the map is a coarse version of the geometrical shadow that is applied to the optical field of the patch e^d_{st}. Calculating the visibility map, we test whether the patch is visible from any cell. If not, we skip it.

We calculate the optical field U_{st} of the patch e^d_{st} using a propagation of the angular spectrum, i.e.,

$$U_{st} = \mathrm{FFT}^{-1}\left[\mathrm{FFT}\left(V\right) \otimes \mathcal{H}_{stz}\right], \tag{4.2}$$

where $\otimes$ denotes a piece-wise multiplication, V are optical fields values of a patch, z is a distance between the patch e^d_{st} and the plane κ and $\mathcal{H}_{stz}$ is a propagation operator.

The optical field values $V = [v_{mn}]$ are valid for a patch whose centre is located over the centre of the field. This means that

$$v_{mn} = \operatorname{rect}\left(\frac{m}{E} - \frac{1}{2}\right) \operatorname{rect}\left(\frac{n}{E} - \frac{1}{2}\right) \zeta_{mn}, \tag{4.3}$$

where ζ_{mn} defines optical field values of the patch. Since we defined that the surface is diffusive, we use $\zeta_{mn} = \exp[\mathrm{j}2\pi\phi_{mn}]$ where ϕ_{mn} is a pseudorandom function. The operator $\mathcal{H}_{stz} = [h_{\eta\psi}(s,t,z)]$ combines a phase shift due to the distance with a phase shift due to a spatial shift in the plane of the patch. The latter is a shift theorem [Goo05]. Let η and ψ be indices of frequencies. Then,

$$h_{\eta\psi}(s,t,z) = \underbrace{\exp\left(-\mathrm{j}2\pi\eta s\frac{E}{M} - \mathrm{j}2\pi\psi t\frac{E}{N}\right)}_{\text{a shift in the plane}} \underbrace{\exp\left\{\mathrm{j}2\pi z\left[\frac{1}{\lambda^2} - \left(\frac{\eta}{X}\right)^2 - \left(\frac{\psi}{Y}\right)^2\right]^{1/2}\right\}}_{\text{propagation of the angular spectrum, see Eq. (2.25)}}, \tag{4.4}$$

where $X = MD_x$ and $Y = ND_y$ are width and height of a rectangle that encloses points at which the optical field is sampled.

Since we now know that the patch e^d_{st} is visible and we estimated its optical field values U_{st}, we can calculate a contribution u'_{mn} to the final optical field value u_{mn} as

$$u'_{mn} = a_e \hat{u}_{mn} t_{\lfloor \frac{m}{E} \rfloor \lfloor \frac{n}{E} \rfloor}, \tag{4.5}$$

where $\hat{u}_{mn}$ is a value of the optical field values U_{st} and a_e is an amplitude of the patch e^d_{st}. Evaluated, we add the contribution u'_{mn} to the corresponding value of the resulting optical field.

In this section we gave a mathematical description of calculations used while the processing of a patch. This closes the description of the method. In the following section we shall present results calculated with the proposed method and compare the method to other relevant methods.

4.1.4 Results

Since we described principle of our method, we can now present results obtained. The results are presented in two steps. First, we present result showing that our method calculate working optical fields. Then, we present time measurements. The latter is contained in the next section.

In this section we present evidence to show that our method calculates working optical fields. Following a principle that is used by all other methods, we verify the content of the calculated optical field by reconstruction. By reconstructing the optical field we refer to calculation of optical field values from known ones. Following ability of detectors to measure only intensity, the result of the reconstruction is usual a 2D image of intensity. Using the reconstruction, we are able to explore the content encoded in the optical field. And if we are able to reconstruct encoded sources, we shall consider the result valid.

In the ideal case, the reconstruction is done optically, i.e., by creating a hologram and reconstructing it using a (quasi-)coherent light [Har96]. This, however, is too complicated

and it requires expensive equipment but it provides the most reliable result. The optical reconstruction can be simulated and the results of the simulation are similar to results captured by CCD. A result depicted in Fig. 4.8(a) was capture by CCD without any additional optics. In this particular case, we used a small LCD as SLM. Since there is no additional optics that might cause a deformation, we composed the final image from multiple projections. In order to compare the measured result we simulated it numerically propagation of the angular spectrum and we obtained a similar result that is depicted in Fig. 4.8(b). In both cases we used an optical field values calculated by a method based on ray-casting [JHO08] and the phase on the surface was constant. Based on result, we consider the numerical reconstruction similar to the optical one.

(a) (b) (c)

Figure 4.8: (a) An optical reconstruction of a virtual scene with a constant phase on its surface. (b) A numerical reconstruction of the same scene and (c) a numerical reconstruction using an aperture. Notice the missing part of the object due to the pinhole. The dashed circle shows boundary of the aperture.

The goal of our method is to calculate an optical field that can be viewed by a human viewer and that does not require projecting into a flat screen or any other similar medium. Therefore, the optical field has to have features of an optically captured hologram, i.e.,

1. It has to contain multiple views on the scene. This allows every eye to see a different image and thus reconstruct the depth.
2. It has to alow focusing on various depths. This allows a natural viewing of the scene.
3. Obscuring by a screen with opening should limit only the range of views or sharpness of the reconstruction. As a consequence, we can select different views [NM08].
4. Applying a lens should cause a deformation similar to the perspective one.

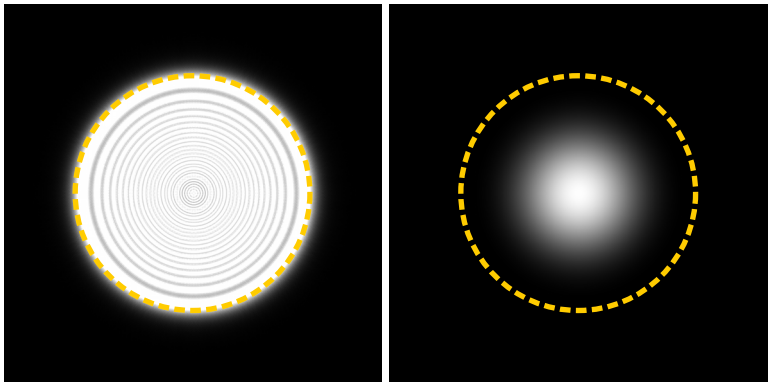
Almost all papers use just propagation of the angular spectrum between two planes. This, however, validates only the first and the second feature. It cannot validate the third and the fourth feature. Especially, the third feature is crucial for the human viewer [LHJ68]. While the numerical reconstruction depicted in Fig. 4.8(b) proves that the optical field contains the scene, it does not show that by applying a pin-hole, part of the scene disappears as depicted in Fig. 4.8(c). This fact is widely ignored by many publication on computer generated holograms.

Following the goal of our method, we numerically simulate the human eye. The human eye consist of a lens, a pinhole and a projection surface, i.e., the retina. In our case, we use

a planar surface as the projection plane. The pinhole is an aperture with a small circular opening. Since every sharp edge generates a strong response as illustrated with Fig. 4.9(a), we decide to define the pinhole as

$$a_{mn} = \begin{cases} \frac{1}{2} - \frac{1}{2}\cos\left\{\pi \frac{[(mD_x)^2+(nD_y)^2]^{1/2}}{d}\right\} & \text{if } (mD_x)^2 + (nD_y)^2 \leq d^2, \\ 0 & \text{otherwise,} \end{cases} \tag{4.6}$$

where d is a radius of the pinhole. The effect of such a definition is illustrated with Fig. 4.9(b). Since the reconstruction is not a major concern of this work, we did not experiment with other aperture definitions that might be better.



(a) (b)

Figure 4.9: Intensity of a plane wave propagated through either (a) a pinhole with sharp edges or (b) an amplitude modulator defined by Eq. (4.6). The dashed circle shows an area where the pinhole function is not zero.

Similar to the ideal thin lens [Wal95, Goo05], we consider a lens a phase shifting medium [ZCG08]. We, however, do not use any approximation from Eq. (2.35). Instead of this, we define a lens such that a point at a distance f_A in front of the lens is projected by the lens to the distance f_B behind the lens [Lob08]. The phase shift of such a lens is

$$\phi^l_{mn} = -\frac{1}{\lambda}\{[(mD_x)^2 + (nD_y)^2 + f_A^2]^{1/2} + [(mD_x)^2 + (nD_y)^2 + f_B^2]^{1/2}\}. \tag{4.7}$$

As depicted in Fig. 4.10 the distance f_A controls what is being focused and the distance f_B controls the deformation. Such a lens does not apply any approximation and has similar features as does the thin lens from Sec. 2.2.5.

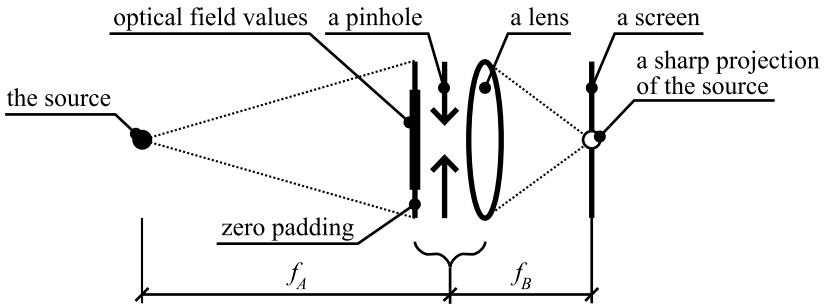


Figure 4.10: A setup used by the numerical reconstruction. The lens, the aperture and the plane with optical field values are located at the same distance from the screen.

For purpose of the demonstration, we create a few scenes. Each scene serves different purposes. The scene "Bunny" is based on the Standford Bunny dataset [SU94]. The scene show ability of the method to handle concave surfaces and surfaces almost perpendicular to the plane $\kappa : z = 0$. The scene "Chess" contains multiple small objects and it is used to demonstrate effect of our method on details. The scene "Plane" consists of a single plane with a texture. This scene the basic scene and it demonstrates validity of our method. The scenes "Primitives" and "Primitives2" contain six objects. In the first case, objects are distributed at significantly different depths and the scene is dedicated to test the perspective and the visibility. In the second case, the last object is much larger and the depth range is smaller than in the fist case. The scene is used to test the visibility. The last scene "StillLifeBunny" allows us to demonstrate the visual quality of the reconstruction because it contains multiple objects, detailed textures and precalculated shadows. Parameters of the scenes including orthogonal projections are presented in Chap. C.

In most of cases, we calculate two sets of optical fields: one intended for the numerical reconstruction, the other for optical reconstruction. Each of them differs in parameters. In both cases, however, we use a wavelength 635 nm, i.e., a red light.

In the case of optical reconstruction we calculate a fairly large optical field: $6,144 \times 6,144$ samples and we use the sampling step $D_x = D_y = 7.0$ μm. Using these parameters, we choose the size of the patch similar to a pixel size of contemporary LCD monitors, i.e., 0.22 mm. This gives us that a patch consists of 32×32 samples. We use this resolution of a patch for the numerical reconstruction case too. Since such a larger sampling step is useful only for numerical reconstruction, we denote such scenes with a symbol ‡in superscript, e.g., "Primitives‡". The meaning of such symbols is summarised in Tab. C.2.

Considering a hologram that is an real-valued amplitude modulator the range of viewing angles is 2.6° according to Eq. (2.24) and according to the fact that the angular spectrum of the modulator is symmetric. After evaluating the optical field, we calculate on off-axis hologram by adding a plane wave. Since we print holograms using a binary device, we use a threshold to create a binary hologram. The threshold is selected such that the amount of white and black pixels is approximately the same. This allows us to preserve as many structures in the hologram as possible.

We print the hologram using an image setter and illuminate it with a quasi-coherent light source.[7] We use a high luminous LED of 640 nm and we filter it with a pinhole to create an expanded beam of quasi-parallel rays. The images are captured using a regular camera with optics and therefore they are reliable in terms of visual impression on the human viewer.

In the case of the numerical reconstruction, we assume an optical field with following parameters: $4,096 \times 4,096$ samples, the sampling step $D_x = D_y = 0.5$ μm. Considering optical field values that are complex numbers the range of viewing angles is 78.8°. This is enough for testing perspective and visibility. During reconstruction, we pad the optical field with a frame of zeros so that we process optical fields of $12,228 \times 12,228$ samples.[8] If not noted otherwise, only $2,048 \times 2,048$ samples at the centre of the reconstruction image are presented. Since the lens inverts both the X-axis and the Y-axis, we inverted them back in the presented images. The distance f_B is 2.0 mm and the diameter of the pinhole is 0.5 mm.

The reconstructions will contain multiple copies of the object that do not overlap each other. This is a side effect of the periodicity assumed by FFT. Since we use FFT to generate the field, we cannot avoid these copies. If necessary, we can pad the optical field at the

[7] A usual image setter has 3600 DPI, i.e., it is able to create a 7.0 μm binary dot that is well defined.

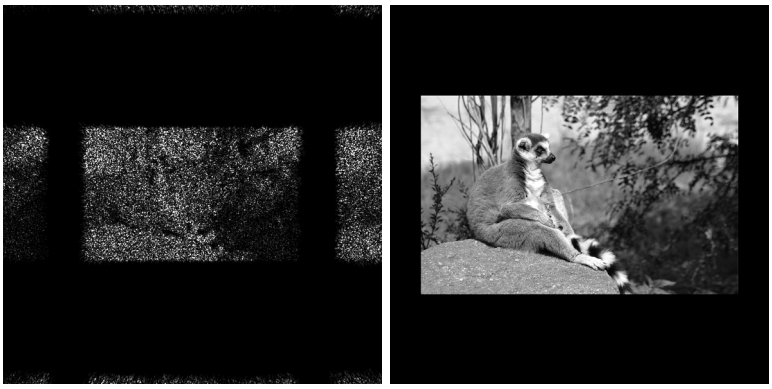
[8] $3 \times 4,096 = 12,228$

generation time so the copies will be further away from each other. Also, the copies might overlap each other due to perspective introduced by the lens. Since we solve visibility only for a limited range of viewpoints, the copies may contain holes.

We tested our method using the scene and the parameters described above. First, we tested the scene "Plane". In such a case, we can show that our method is fully functional. Since every cells sees every patch, our method follows the Babinet's principle [BW05], which describes a behavior of an optical field disturbed by two planar screens. Both screens contain openings that are disjunctive and that fill the whole plane when added up. Let us have a source that emits waves and thus forms an optical field values U on the plane κ. If the first screen is placed between the source and the plane κ, the optical field values U are disturbed and optical field values U_1 are detected on the plane κ instead. Similarly, when the second screen is used, optical field values U_2 are detected on the plane κ. The relation between the fields is $U = U_1 + U_2$. This is described by the Babinet's principle.

Let us now consider a planar screen that is parallel to the plane κ. An original source of waves is behind the screen and thus we can consider the screen to be a source of waves. The waves emitted by the screen form optical field values U on the plane κ. Now, we make a half of the screen black, i.e., opaque. As a consequence, we detect disturbed optical field values U_1 on the plane κ. When the second half is made black instead of the first one, we detect optical field values U_2. According to the Babinet's principle $U_1 + U_2$ gives us an undisturbed optical field U. Furthermore, we apply the principle on the field U_1 to decompose it into two optical field values. Such a recursive application of the principle can be repeated until used screen contains only a single opening of a size equal to the patch. Even in that case we are able to reconstruct the original values U from values generated due to screens. This shows that our method works for the scene consisting of a plane parallel to the plane κ.

This is verified by a successful numerical reconstruction depicted at Fig. 4.11(a). Despite that the reconstruction is damaged by a speckle noise, the original texture, which is depicted in Fig. 4.11(b), is recognisable.



(a) (b)

Figure 4.11: (a) An optical reconstruction of the scene "Plane" focused at 6.0 mm and (b) the original texture. The texture is courtesy of Libor Váša. Used with his permission.

Next, we calculated optical fields of scenes "Primitives" and "Primitives2" and we reconstructed them numerically. In order to obtain different views, we shifted the grid of calculated samples in the plane κ by 0.5 mm horizontally just after padding with zeros. The reconstructions in Fig. 4.12 show that the optical field works as expected: both the perspective and the visibility are correct and we are able to focus at different depths.

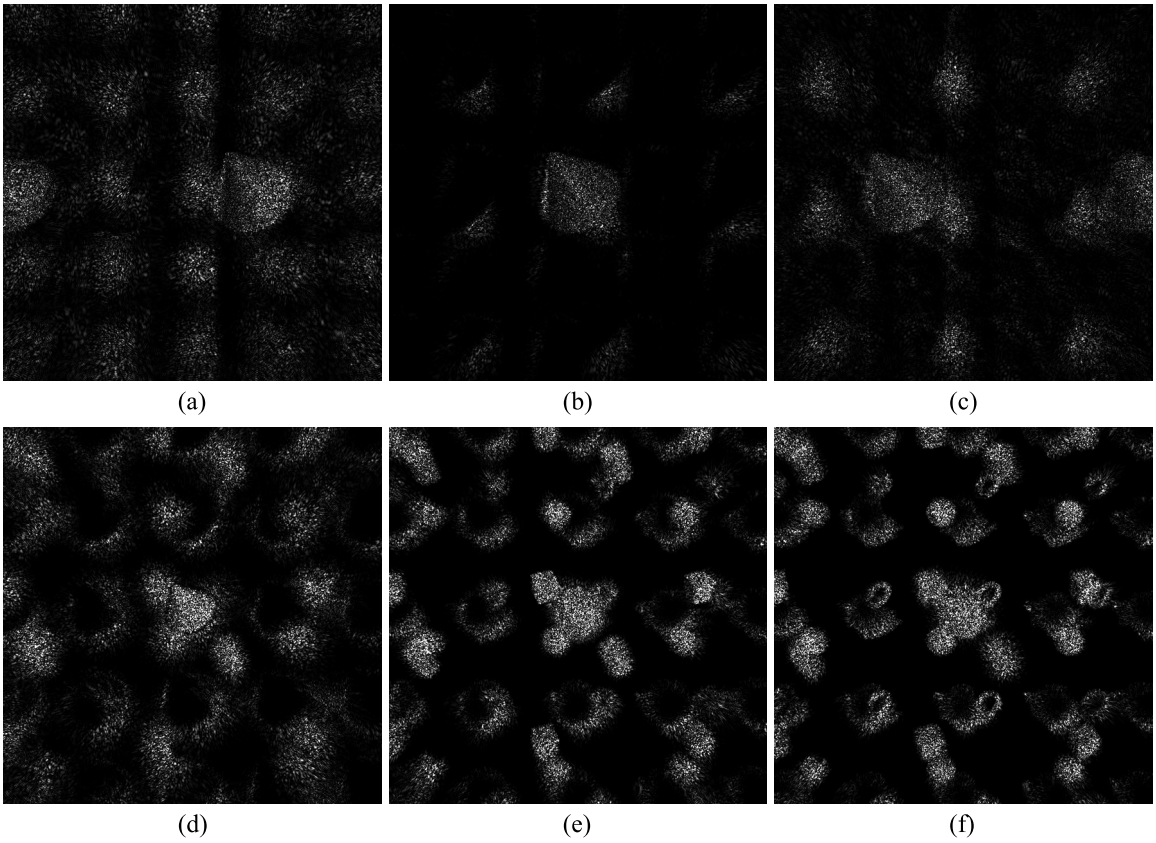


Figure 4.12: (a-c) A numerical reconstruction of the scene "Primitives" focused at the cone. The optical field was shifted horizontally by (a) -0.5 mm, (b) 0.0 mm, (c) $+0.5$ mm. (d-f) A numerical reconstruction of the scene "Primitives2" focused at different objects (different depths): (d) the cone (6.0 mm), (e) the cube (10.0 mm) and (f) the torus (14.0 mm).

We tested results of our method optically using the scene "Primitives[‡]". As depicted at Fig. 4.13 the reconstructions were successful. Objects are recognisable and changing of camera position changes relative position of objects while the visibility stays correct as expected. Thus, our method work for optical reconstruction too.

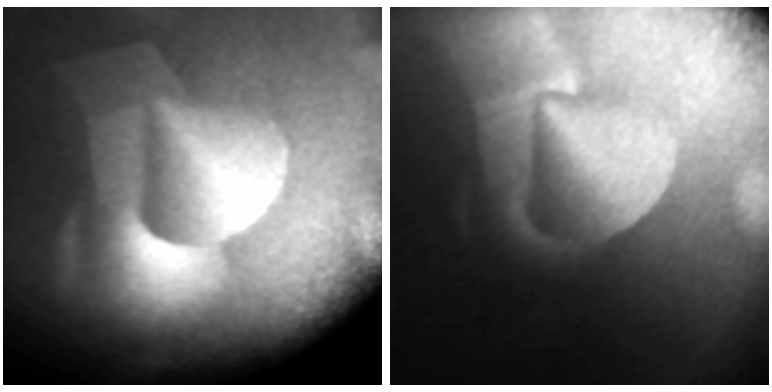


Figure 4.13: An optical reconstruction of the scene "Primitives[‡]" focused at the cone. Notice that visibility is solved correctly when viewer changes its location.

Before creating the hologram, we calculated a numerical reconstruction using just the propagation of the angular spectrum [Fig. 4.14(a)] and we observed artifacts at the edge of patches [Fig. 4.14(b)]. In this case artifacts forms vertical lines. These artifacts are overlapping energy contribution of patches. The nearer the reconstruction distance is to the distance of a patch, the more energy is gathered close to the patch. As as consequence, the artifact is recognisable at patches that are almost at the focus and it is much smaller than the patch. Despite that we were able to recognise the artifacts using the propagation in the angular spectrum, we are not able to recognise them on optical reconstruction [Fig. 4.14(c)]. Therefore, we assume that these artifacts will not disturb the viewer unless the patch is too large. Since the large patch case is suitable only for previewing purposes, we do not need to handle these artifacts.

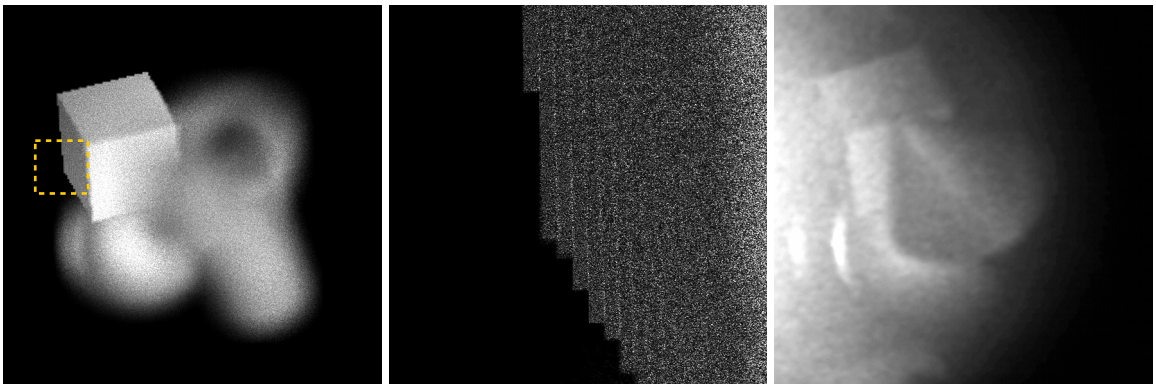


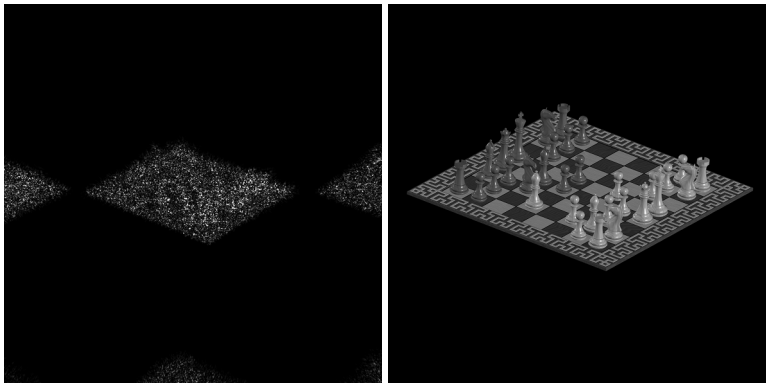
Figure 4.14: (a) A numerical reconstruction of the scene "Primitives‡" focused at the cube (0.5 m). The reconstruction uses a propagation in the angular spectrum without a pinhole and a lens. (b) An enlargement of the dashed rectangle showing artifacts in a form of vertical lines. (c) An optical reconstruction focused on the cube. Notice that the artifacts has no influence on the optical reconstruction.

As the next, we tested an influence of a patch size. The result, which is depicted at Fig. 4.15, indicates that details smaller than the patch size are lost completely. If the scene contains small objects, the scene may not be reconstructed successfully. On the other hand, if the scene contains larger objects, change of the patch size, which is illustrated by Fig. 4.16, changes mostly just the coarseness of the scene as expected. Notice that tiny details formed by a small patch [Fig. 4.16(a)] are lost in the speckle noise.

Finally, we demonstrate that our method can generate an optical field that provides a depth impression. For that purpose we calculated an optical field of the scene "StillLifeBunny‡". Shifting the optical field values in the plane κ, we created two reconstruction and combined them to a anaglyphic image presented at Fig. 4.17. If the viewer uses properly coloured glasses, a impression of a depth is recreated. This shows that our method can provide 3D image of the scene.

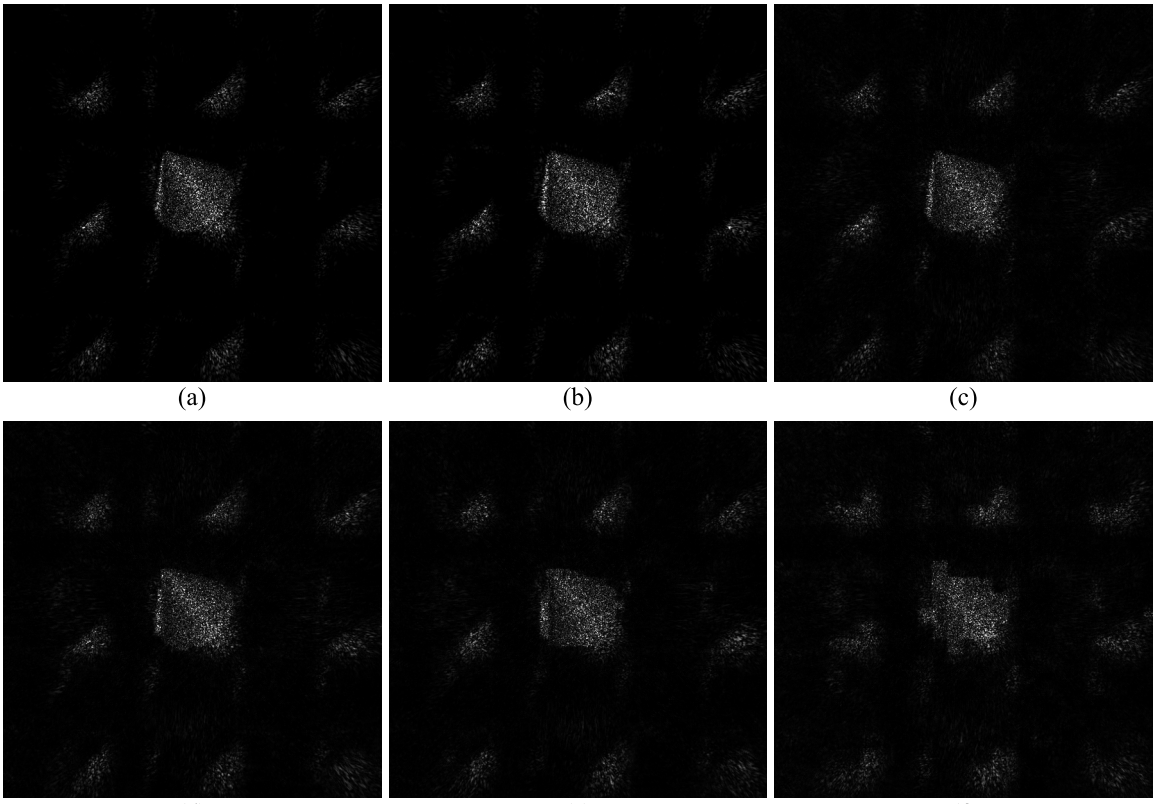
4.1.5 Comparison

In the previous section we presented results calculated by the detail driven method. We showed that results work for the optical reconstruction. In this section we shall compare our method to others. This section will show that our method is more efficient than strictly PLS-based methods and strictly wave-based methods.



(a) (b)

Figure 4.15: (a) A numerical reconstruction of the scene "Chess" focused at the closes corner of the chessboard (6.0 mm). (b) An orthogonal projection of the scene.



(a) (b) (c)

(d) (e) (f)

Figure 4.16: An numerical reconstruction of the scene "Primitives" focused at the cone (6.0 mm). The optical field was calculated using a patch size: (a) 8×8 samples, (b) 16×16 samples, (c) 32×32 samples, (d) 64×64 samples, (e) 128×128 samples and (f) 256×256 samples. Notice the coarseness of cone edges when the size of the patch increases.

Before we proceed to the comparison, let us first discuss criteria of the comparison. Since we generate hologram intended for viewing purposes, we compare the visual quality. For that purpose we choose a geometry-based method that uses a cloud of PLS (a PLS-based method) because there is no DFT-forced periodicity, the angular spectrum is not deformed

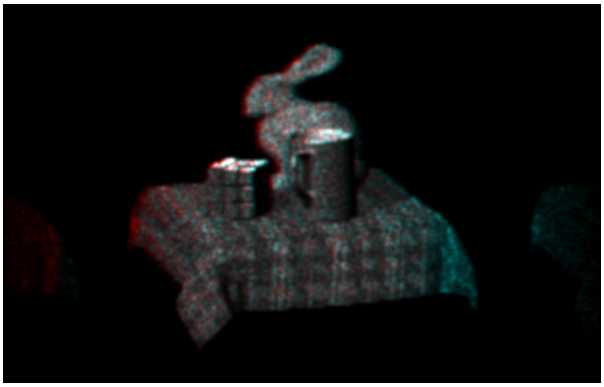


Figure 4.17: Numerical reconstructions of the scene "StillLifeBunny‡" focused at the corner of the table (0.4 m). The reconstructions are combined to anaglyphic image. A red filter is assumed to be on the left eye.

by resampling as in the case of wave-based methods and there are no wave-leaks described in Sec. 3.1.2.

Instead of using a general PLS-based method, we choose to use a ray-based method with a constant angular step (the AngRay method) [JHO08]. The method evaluates a visibility without under-sampling and it is able to capture a solid diffuse surface. At the same time, it supports GPU and therefore it can provide comparable results in reasonable time.

We decided that we shall not examine optical field values and we shall use numerical reconstructions instead because both methods define a phase variation of the diffuse surface slightly different and therefore a significant difference in phase does not imply a significantly malformed reconstruction.[9] Furthermore, we did not apply any metrics such mean square error or peak signal-to-noise ratio on the reconstructed image due to speckle noise. When viewed, speckle noise are tiny dots on the surface of the object. The pattern of these dots differs significantly for every viewing location. Since the viewer is not able to view the hologram perfectly still and the human eye integrates intensity over time, the reconstruction seems to be sparkling and perfectly sharp at the same time. However, through numerical evaluation, we obtain only a perfectly still view. Such a view differs from the real visual impression and thus a result of a numerical evaluation might be misleading. Therefore, we decided to compare results visually.

We calculated an optical field of $4,096 \times 4,096$ samples and reconstructed it using both the setup and the parameters from Sec. 4.1.4. Since we have already shown a loss of tiny details in the scene "Chess" in Sec. 4.1.4, we used the scene "Primitives" that contains multiple medium sized objects and the scene "Bunny" that contains a single object. As it can be seen in Fig. 4.18, larger objects are almost not affected by the loss of detail and the visibility approximation applied by our method does affect the result significantly too.

Since our method aims to reduce the computational time, we evaluated it as well. Following the principle of our method, we presume that it is faster only under certain conditions. Therefore, we derive analytically the computational complexity and based on that we present a lower estimation of boundary parameters. We verify all these them through measurements.

[9] Besides that, we lack equipment to capture a hologram of a real object and use it for comparison. If we had had such equipment, we would have used it instead of results of the PLS-based method.

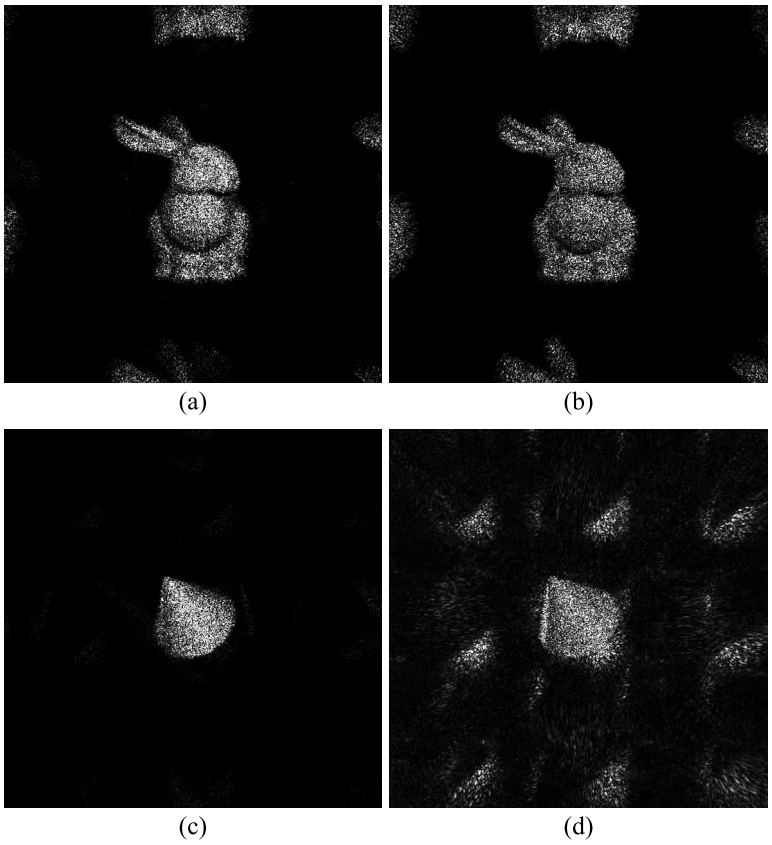


Figure 4.18: (a, c) A numerical reconstruction from optical fields calculated using the AngRay method [JHO08]. (b, d) A numerical reconstruction from optical fields calculate using the proposed methods. Notice the slight deformation of bunny's ear in (b) and a distortion of the cone in (d). Both artifacts are caused by use of patches.

Let us express formally the computational complexity. For simplicity let us assume that the optical field values are calculated at $N \times N$ sampling points. The number of visibility grid cells along a single axis is $C = \lfloor \frac{N}{E} \rfloor$. For a single patch, the estimated number of operations is a sum of the number of pillars that are examined during the visibility test and the number of steps required for FFT. The computational complexity of 2D FFT is $O(N^2 \log N)$. The number of pillars examined during DDA traversal can be expressed as follows. If the cell g_{00} is the starting one and the cell g_{vw} is the ending one, the number of examined cells is $v+w$ at maximum. An exception is a case where $v = w$. In such a case we assume the ray traverses from one cell to the next one through the common corner and the number of examined cells is v. We can, therefore, express the total number of examined cells when the visibility of the patch e^d_{00} corresponding with the cell g_{00} is solved as an arithmetic series. Computing the sum of the arithmetic series and assuming that each cell contains K pillars in average, the total computational complexity of processing one patch is

$$\mathcal{O}\left[N^2 \log_2 N + K\left(C^3 - \frac{3C^2 - C}{2}\right)\right]. \tag{4.8}$$

The expression in Eq. (4.8) is valid for other cells as well so we can now express the total complexity for the whole hologram. As $K \ll C$, we consider $KC^n \approx C^n$, $n \in \mathbb{N}$ in Eq. (4.8).

Since $C = \lfloor \frac{N}{E} \rfloor$, Eq. (4.8) becomes

$$\mathcal{O}\left(N^2 \log_2 N + \frac{N^3}{E^3} - \frac{3}{2}\frac{N^2}{E^2} + \frac{1}{2}\frac{N}{E}\right) \approx \mathcal{O}\left(N^2 \log_2 N + \frac{N^3}{E^3}\right). \tag{4.9}$$

There are $KC^2 \approx C^2$ patches that have to be processed. Therefore, for a larger N the computational complexity of the complete generation process is approximately

$$\mathcal{O}\left[N^4 \left(\frac{\log_2 N}{E^2} + \frac{N}{E^5}\right)\right]. \tag{4.10}$$

Notice that expression Eq. (4.10) shows a significant feature of our method: the efficiency of the method depends on a size of a patch. If the patch consists only of a single sample, our method will become a PLS-based method.[10] Thus it is desirable to have the patch as large as possible. At the same time, however, increasing the patch size increases coarseness of scene (Fig. 4.16). Therefore, the patch has to be chosen as balance between the visual quality and the computational time.

We can estimate patch size using an known ability of the human visual system to recognise details [Luc94]. In our case, however, we decided to use a much simpler solution: a pixel size of an contemporary LCD. Using such pixels, we can view an image without any significant degradation and hence the patch of similar size will not degrade the visual impression too. Among others, thanks to both perspective deformation and the fact that we assume the virtual scene fully behind the hologram, the patch will be smaller than that. As it was already noted in Sec. 4.1.4, we assume a 0.22 μm pixel.

As the next step, we verified the derived computational complexity by a measurement. For that purpose we measured computation times for the scene "Primitives[‡]" using various resolutions of the optical field and various resolutions of the patch. In all measurements, we used a sampling step of 7.0 μm. The method was implemented using the C++ language and the FFTW library [FJ]. All times were measured on a PC with Intel Xeon 3.2 GHz. The measured times were compared to times predicted by the computational complexity.

In the first set of measurements we kept the resolution of the patch equal to 32×32 values and we calculated optical fields at various resolutions. The pitch between the values was scaled proportionally to the resolution of the field so that the optical field size was always 43×43 mm. This preserved the ratio of a number of pillars to a number of cells almost constant and therefore we were able to compare easily the measured times. Results of the measurements are provided in Fig. 4.19(a). In the second set of measurements, we kept the resolution of the optical field equal to $6,144 \times 6,144$ values and used different resolutions of the patch. Unlike the first set, we kept the pitch between points $\mathbf{u}_{mn}$ constant. This again preserved the ratio of a number of pillars to a number of cells. Results of the measurements are provided in Fig. 4.19(b).

In order to compare the measured and predicted times, we multiplied the result of Eq. (4.10) by a constant σ. This is a valid operation because the expression in Eq. (4.10) is a computational complexity where multiplicative constants are neglected. The constant σ modifies the result of the expression in Eq. (4.10) so that the predicted time p_c becomes

[10] Actually, a PLS-based method with an incredibly inefficient calculation of PLS optical field due to a propagation of the angular spectrum.

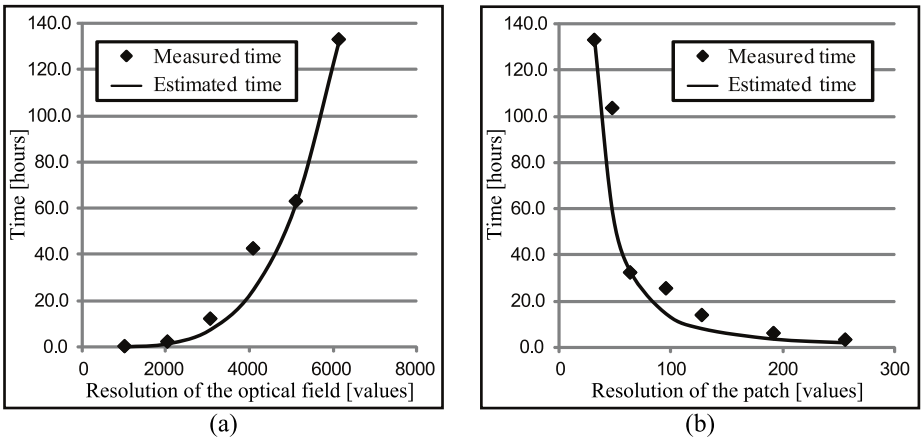


Figure 4.19: Time measurements of our method for the scene "Primitives‡". (a) The patch resolution is constant and both the sampling step and the resolution of the optical field differ. (b) Both the sampling step and the resolution are constant and the patch resolution differs.

equal to a calibration measurement t_c. The calibration measurement t_c is measured using an optical field of $6,144 \times 6,144$ values with a patch of 32×32 values. Therefore, the predicted time p_p corresponding to the measured time t_p is $p_p = \sigma s_p$, where $\sigma = \frac{t_c}{s_c}$. The estimations s_c and s_p are calculated according to the computational complexity expression Eq. (4.10) with parameters corresponding to the measurements t_c and t_p respectively. The estimated times presented in Fig. 4.19 correspond with measured times and we can consider the expression in Eq. (4.10) as a valid estimation of the computational complexity of our method.

Since we verified the complexity expression Eq. (4.10) by measurements, we can use it to establish performance relation with both PLS-based methods and wave-based method. Let us first discuss PLS-base methods. A computational complexity of PLS-based methods is $\mathcal{O}(PN^2)$ where P denotes the total number of PLS. This complexity does not include a visibility solution. The expression in Eq. (4.10) shows that our method has a lower complexity if

$$P > N^2 \left(\frac{\log_2 N}{E^2} + \frac{N}{E^5} \right). \tag{4.11}$$

If the cloud of PLS represents a solid surface then $P \approx N^2$ for common scenes and therefore the total complexity becomes $\mathcal{O}(N^4)$. Under such assumptions, our method is faster if $(\frac{1}{E^2}\log_2 N + \frac{N}{E^5}) < 1$ and for larger N if $N < E^5$.

To prove validity of the expression Eq. (4.11), we measured two computational times: a computation time t of our method and a computation time t_{PLS} of a single PLS using the spherical wave expression Eq. (2.9). The sampling step was 7.0 μm. We used the scene "Primitives" and a patch of 32×32 samples. The measurement was done on PC Intel Xeon 3.2 GHz. Then, we calculated a number n_{PLS} of PLS that could had been calculated during time used by our method, i.e., $n_{\mathrm{PLS}} = \lceil t/t_{\mathrm{PLS}} \rceil$. Then, we evaluated the expression Eq. (4.11) that defines efficiency boundary for our method using above given parameters and we compare them to measured results.

As it is shown in the table Tab. 4.1, the estimation using the expression Eq. (4.11) was higher than a number of PLS estimated from measured times and therefore the expression is valid. Furthermore, the scene "Primitives‡" consist of 972 triangles. This means that PLS-based method is able to use only 20 PLS per a triangle in the case of the largest optical

field. This is far from creating an impression of a solid surface. Hence, our method is faster than any PLS-based method.

Table 4.1: A comparison between an estimated number P of PLS and measured number n_{PLS} of PLS for a different resolution of the optical field. The measured number of PLS was estimated from the computation time t of the scene "Primitives2[‡]" using our method.

Resolution	t [s]	P	n_{PLS}
$1,024 \times 1,024$	226.5	10,272	687
$2,048 \times 2,048$	4,493.7	45,312	3,431
$4,096 \times 4,096$	42,749.3	198,656	7,602
$6,144 \times 6,144$	244,948.6	470,844	19,186

Let us now discuss wave-based methods. We choose to use a triangle-based generator [Mat05] because it handles both the visibility and a diffuse surface as our method does. Other methods do not offer similar abilities, i.e., they are not efficient [Loh78], the diffuse surface is either limited [KHL08] or not supported [ABMW08] or they do not solve the visibility [ABMW08]. Therefore, we excluded them from the comparison.

The selected wave-based method yields a computational complexity of approximately $O(TN^2 \log_2 N)$ where T denotes a total number of planar surfaces. In our case we consider triangles because it is compatible with a representation used in the computer graphics. Unlike PLS-based methods, the complexity includes a visibility solution. According to the expression in Eq. (4.10), our method has lower complexity if

$$T > \frac{N^2}{E^5}\left(E^3 + \frac{N}{\log_2 N}\right). \tag{4.12}$$

Similar to the case of the PLS-based method, we verified the expression Eq. (4.12) by measurement. Due to time restrictions, we implemented the selected method only partially. In its original form, the selected method does following operations for each triangle:

1. Rotate the angular spectrum.
2. Convert the angular spectrum to the spatial domain.
3. Raster the triangle.
4. Calculate the angular spectrum from the representation in the spatial domain.
5. Apply the propagation operator by piece-wise multiplication with the angular spectrum.

We implemented steps 2), 4) and 5) because we presumed that they are the most time expensive ones. As a consequence, the real computation time will be higher. Since all these steps together form a propagation of the angular spectrum, we used the same implementation as does our method.

Using the partial implementation and configurations from the case of the PLS-based methods, we measured the lower estimation of a time t_{TRI} per a single triangle and we estimated a number n_{TRI} of triangles as $n_{\mathrm{TRI}} = \lceil t/t_{\mathrm{TRI}} \rceil$, where t is a computational time

using our method. As it is shown in the table Tab. 4.2, the expression Eq. (4.12) gave us a number of triangle that is higher than a number estimated from measured times and therefore the expression is valid. Among others, the table Tab. 4.2 shows that our method is not well suited for scenes that contains a low number of triangles.

Table 4.2: A comparison between an estimated number T of triangles and measured number n_{TRI} of triangles for a different resolution of the optical field. The measured number of triangles was estimated from the computation time t of the scene "Primitives2‡" using our method.

Resolution	t [s]	T	n_{TRI}
$1,024 \times 1,024$	226.5	1,027	278
$2,048 \times 2,048$	4,493.7	4,119	1,360
$4,096 \times 4,096$	42,749.3	16,555	3,184
$6,144 \times 6,144$	244,948.6	37,413	5,751

In this section we showed that our method provides a visual quality similar to the more accurate PLS-based method. We derived computation complexity and we validated it through measurements. We compared performance of our method to others and we shown that our method is faster even though this is true under certain conditions. With an appropriate size of the patch, PLS-based methods are not able to create an impression of a solid surface during the same calculation time.

Our method is not suitable for scenes that contain a low number of triangles because wave-based methods can calculate them faster. On the other hand, our method can process scenes that contain a high number of triangles without any preprocessing such as triangle decimation in shorter times than the wave-based methods. Furthermore, our method does not require resampling of the angular spectrum because it does not rotate a patch. Hence, it does not introduce additional noise. Besides that, our method is able to control both the visual quality and the computation time though the patch size: a large patch can be used for previews while smaller patch for final production of a hologram.

This section closes the evaluation of the principle. In the following sections we shall discuss acceleration of the method by various approximations.

4.2 Accelerations

In the previous section we described the basic algorithm used by our method. The combination of both the PLS-based principle and the wave-based principle proved to be efficient. Despite this success we explored the method further and revealed that the method can be accelerated even further by modifications of the algorithm.

In this section we propose modifications that accelerate the algorithm. We discuss the impact of the modifications on the visual quality of the result and we measure time improvements using the scenes introduced in Sec. 4.1.4. We prove validity of modifications by reconstructions.[11]

[11] Actually, we follow a rule used by all other authors: if the recorded object is reconstructed properly, the method works and therefore it is valid.

Since we aim to decrease the computational time, we have to first analyse a computational time distribution among steps of the algorithm Alg. 2. Therefore, we calculated optical fields of $4,096 \times 4,096$ samples with the sampling step of 0.5 μm and a patch resolution of 32×32 samples. We used PC Intel Xeon 3.2 GHz to measure computation times of algorithm steps: creation of pillars (step 1), calculation of the visibility map (step 3), calculation of an optical field generated by a patch (step 5) and application of the visibility map (steps 6 and 7). The results are summarised in the table Tab. 4.3.

Table 4.3: Computation time of $4,096 \times 4,096$ samples and its distribution throughout steps of the algorithm Alg. 2 in percentage of the total time.

Scene	Time [hours]	Patch creation	Calculation of the vis. map	Opt. field of a patch	Application of the vis. map
"Bunny"	11.86	7.0×10^{-4} %	0.3 %	97.7 %	2.0 %
"Chess"	11.29	4.6×10^{-4} %	0.3 %	97.9 %	1.8 %
"Plane"	11.46	0.3×10^{-4} %	0.8 %	93.4 %	5.8 %
"Primitives"	12.58	0.6×10^{-4} %	0.7 %	95.5 %	3.8 %
"Primitives2"	9.21	1.2×10^{-4} %	0.6 %	96.1 %	3.3 %
"StillLifeBunny"	6.83	17.1×10^{-4} %	0.6 %	95.7 %	3.7 %

As it is presented in the table Tab. 4.3, the computation time of both the pillar creation and calculation of the visibility map is almost neglectable. Despite that application of the visibility map is very low, we presume that it may become significant if we reduce the most time extensive part: calculation of an optical field generated by a patch. Following the results, we shall focus on acceleration of the optical field calculation. Since the ratio of the calculation to the whole computational time does not almost depend on the scene, we chose the scene "Primitives" and use it for all time measurements.

4.2.1 Library and Grouping

Our method uses patches that are aligned to a visibility grid. However, patches have both a random variation of the phase and an arbitrary location along the Z-axis. Following the approach that we applied though designing the algorithm, these qualities of a patch might offer an opportunity for acceleration. In this section we propose an approach that reduces arbitrariness of qualities mentioned above and it decreases the computational time through that. We show that the reduction of arbitrariness does not degrade the visual quality of reconstructions and we present time measurements.

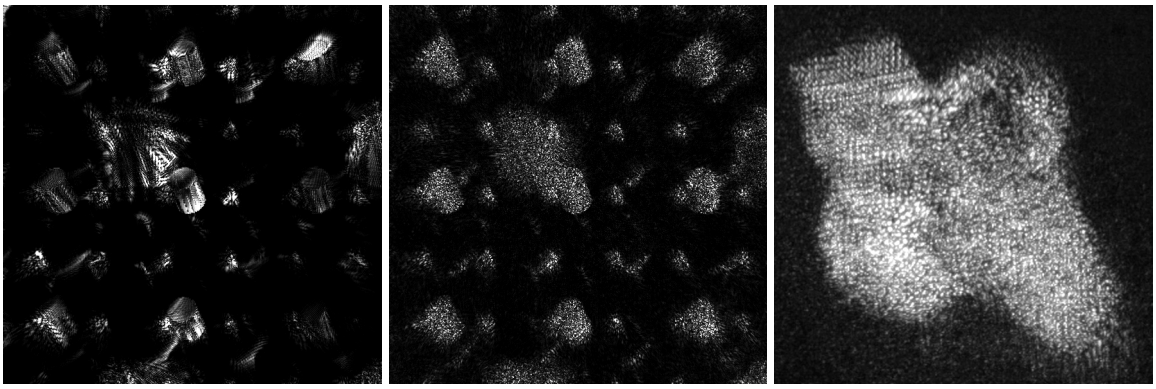
Let us first address the a random phase function. As it was discussed in Sec. 4.1.1, the phase has to be an arbitrary function that is a random function in our case. Nevertheless, following measurements in the table Tab. 4.3, the most time extensive part of the algorithm is calculation of the optical field generated by a patch. Since we use propagation of the angular spectrum, the part consist of a forward FFT, calculation of a phase shift, and an inverse FFT. From these three operations, both the forward and the inverse FFT have the highest computational complexity, i.e., $\mathcal{O}(N^2 \log N)$. Thus, if we had removed a single FFT, we would have been able to reduce the computation time to up to a half.

As a next step, we refine our assumption about the acceleration ratio. We measured time of individual operations and we found out that both FFT represent only about 50 % of the computation time and the rest is devoted to calculation of the phase shift. Despite that we focused on FFT because we assumed that optimization of the phase shift is rather a low-level adjustment. We addressed it later in Sec. 4.2.4. As a consequence, we adjusted our expectation about the time reduction to a range [66%, 75%].

According to the sequence used by the propagation, we can replace the forward FFT. In order to replace it, we have known the angular spectrum of a patch. If the patch was defined with a constant phase, we would be able to express the spectrum analytically. However, in our case we use an random phase and an analytical expression of the angular spectrum contains a convolution. Based on this, we choose to use a single precalculated angular spectrum for all patches instead of FFT, i.e., we defined **the spectrum library** that contains a single spectrum.

We calculated an optical field of the scene "Primitives" using a single precalculated angular spectrum and we tested it numerically using a setup depicted in Fig. 4.10. We focused the cylinder at 12.0 mm. We chose the cylinder because it is partially hidden behind the cone. In order to see the whole cylinder, we shifted the hologram by a vector (0.3 mm, 0.3 mm). We present only a centre of the reconstruction that contains $2,048 \times 2,048$ samples, the rest is clipped. If not noticed otherwise, this setup is used by other reconstructions as well.

The numerical reconstruction depicted in Fig. 4.20(a) seemed to be successful but it differed from the version calculated without the library [Fig. 4.20(b)]. The result contained the expected shape of the scene but unlike the calculation without the library, the surface was disturbed by a an almost regular pattern of dots. We presumed that this regularity might prevent the human viewer from focusing the surface.



(a) (b) (c)

Figure 4.20: Numerical reconstructions from optical fields calculated (a) using a library of a single spectrum and (b) not using a library compared with (c) an optical reconstruction.

In order to verify the presumption, we tested the optical field using the optical reconstruction as shown in Fig. 4.20(c). The optical reconstruction verified our assumption. The viewer was neither able to focus on the surface nor recognise the shape, the view was disturbed by a regular pattern. Since the artifact did not appeared when the library was not use, we assumed that the pattern is caused by interference between patches because patches are aligned to a regular grid and waves generated by patches interfere with each other. If each patch has a different random pattern, the interference is too weak to disturb. Therefore, we propose to increase the number of spectrums in the library.

The method uses diffusive patches. An ideal diffusive patch should illuminate evenly a plane ρ that is parallel to the patch and that is in the Fraunhofer region [Goo05], i.e., far away from the patch. At such a distance, the intensity of all samples at the plane should be constant because the patch is diffusive. A scene build from diffusive patches should yield similar result that is disturbed only by visibility solution as shown in Fig. Fig. 4.21(b). If we use just a single spectrum, the distribution shown in Fig. Fig. 4.21(c) differs significantly. In both cases, however, the average intensity is the same.

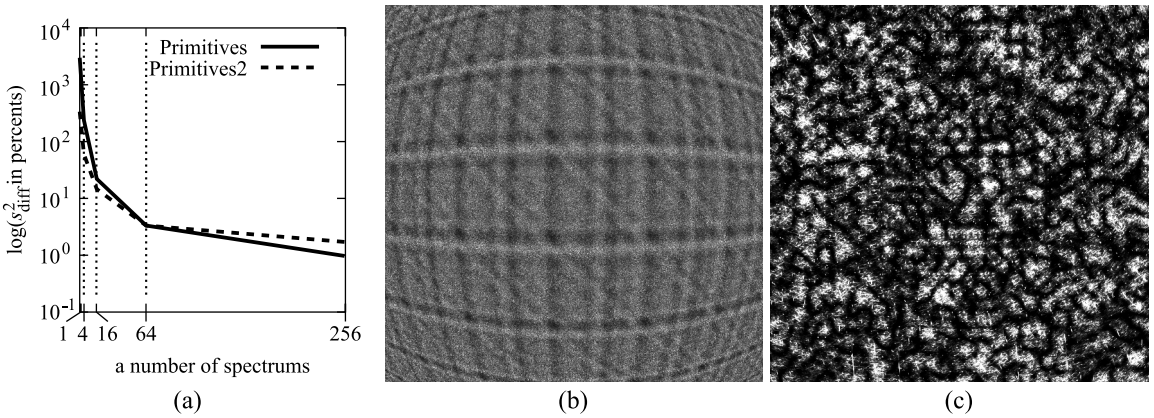


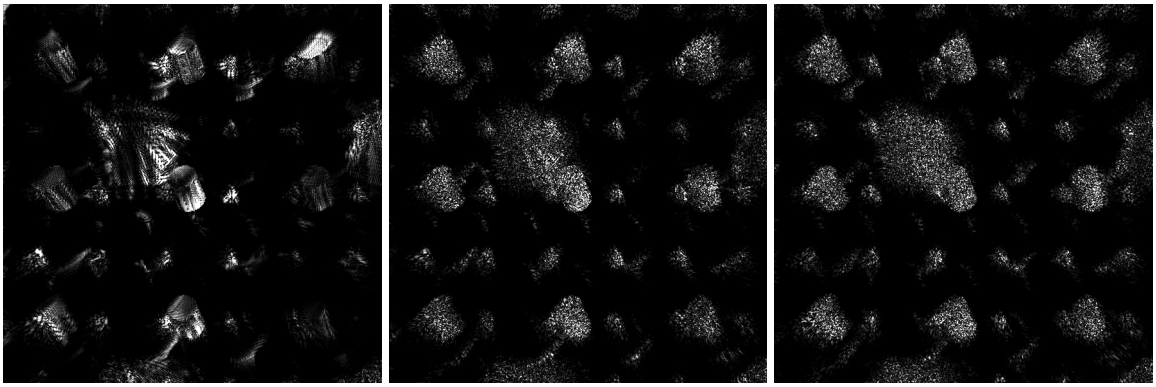
Figure 4.21: (a) A graph showing convergence of sample variance and intensity of samples at the Fraunhoffer region for a optical field calculated (b) without the library and (c) with a library of a single spectrum. The graph shows a difference s^2_{diff} of a sample variance from a sample variance of a case calculated without the library. A scene in (b, c) is "Primitives".

Hence, we estimated the necessary number patches by comparing intensity of samples in the Fraunhofer region [Goo05]. However, since the method uses a patch that is not an ideal diffusive patch, the intensity is disturbed by a high-frequency noise. Therefore, we applied a low-pass filter to attenuate the noise before comparison.

Since the result should be a constant function in the ideal case, we used the sample variance s^2 to evaluate the difference. The sample variance is $s^2 = \frac{1}{N^2}\sum_m \sum_n (I_{mn} - \bar{I})^2$ where $N \times N$ is a number of samples, $I_{mn} = |u_{mn}|^2$ is intensity of a sample u_{mn} and $\bar{I}$ is the average intensity of all samples. In all experiments, the average $\bar{I}$ depended on a used scene. The number of spectrums in the library did not affect it. The result depicted in Fig. Fig. 4.21(a) shows that for all considered scenes the sample variance s^2 converges to the value calculated without the library of spectra. Following the graph, we chose to use 64 spectra because after that convergence of the sample variance s^2 slows down.

We verified the decision by an experiment. We calculated optical fields of the scene "Primitives" and reconstructed them numerically. As it is shown in Fig. 4.22, increasing the number of spectrum in the library weakens the artifact. And as expected, if the library consists of 64 spectrums, the result [Fig. 4.22(c)] is almost indistinguishable from the optical field calculated without the library [Fig. 4.20(c)]. A lower number of spectrums illustrated with Fig. 4.22(a, b) leads to irregularity of intensity at the bottom of the cylinder.

While we increased a number of spectrums in the library, we adjusted slightly the algorithm Alg. 2. The spectrum has to be stored as a whole because it does exhibit neither periodicity nor symmetry. As a consequence, a naïve implementation is highly memory extensive. Therefore, we assign a patch an index of a library spectrum and sort patches



(a) (b) (c)

Figure 4.22: Numerical reconstructions from optical fields calculated (a) using a library of a single spectrum, (b) using a library of 8 spectrums, and (c) using a library of 64 spectrums. Notice the intensity variation at the bottom of the cylinder in (a) and (b).

according to the index prior their processing. As a consequence, we can generated the library of spectrums on the fly and store only a single item from the library.

We reduced significantly the execution of the forward FFT. Even if the library contains 64 samples, it is still less than 0.8 % of patches generated for the scene "Primitives". This means that the execution time of the forward FFT cease to influence the total execution time. In experiments, we measured calculation time and compared it to the computation time using the basic version of our method. The library of one spectrum and library of 64 spectrums reduced the calculation time to 76.1 % and 74.0 % respectively, i.e., in both cases the reduction agrees with our expectations.

The approach described above addresses the phase. Let us now address arbitrariness of the Z-axis location. In its original form, the patch can by located anywhere along the positive Z-axis. However, the viewer does not have a zero depth of field, i.e., the viewer focuses a range of depths not just one. Therefore we can quantise the Z-axis and if two patches use the same index of the spectrum library, we may apply a spatial shift to the second one instead of a full propagation. This will exchange an execution of two FFT plus evaluation of the phase shift for a memory move.

Considering the proposed optimisation, we avoided a fixed and uniform quantisation of the Z-axis. Instead of that we defined a maximum group depth and we grouped patches such that the depth range of the group is less than the maximum group depth. Hence, we denote this acceleration approach as **the grouping**. The larger the maximum group depth is, the more and more patches join the group. At the same time, however, it introduces holes to the surface or it degrades a surface with a low variation over the Z-axis to a plane. Therefore, we propose a criteria for maximum group depth selection that is ideal and compare it to other solution that is ad-hoc and uses a larger group depth.

Let us now assume that the viewer is able to distinguish any depth, i.e., the viewer employs only a propagation in a free space. We have to find such a range of depths that cannot be distinguished by the viewer. In the continuous environment, the range contains only a single depth. In the discrete environment, we can find a range where propagation equals to phase shift as depicted in Fig. 4.23. It is a side-effect of a limited sampling step. Since the propagation becomes a phase shift, the viewer cannot tell whether current optical

field values are optical field values of the patch or they are modified by propagation.[12] As a consequence, the visual impression does not degrade.

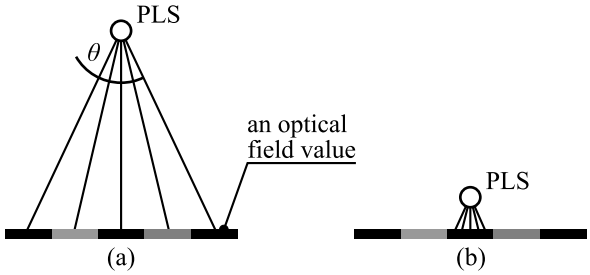


Figure 4.23: (a) PLS contributing to all optical field values and (b) PLS that is too close to contribute to any value but the only the middle. If had been contributing to all samples, it would have created aliasing. The angle θ denotes a maximum angle that does not lead to an aliasing.

We tested the proposed maximum group depth by calculation an amount of patches that shifted spatially instead of propagation. We used the scene "Primitives[‡]" and a library of 64 spectrums. Using the optical field of $6,144 \times 6,144$ samples and a sampling step of 7.0 μm, almost 63.3 % of $16,964$ patches were shifted spatially instead of propagation. However, when we decreased the sampling step to 0.5 μm and resolution of the optical field to $4,096 \times 4,096$, the ratio changed to 7.6 % of $8,000$ patches. The latter was calculated using a smaller optical field. Despite that, we assume the ratio is strictly dependant on the sampling step. Therefore the proposed maximum group depth is not suitable for smaller sampling steps.

Another option is to choose a larger maximum group depth. Since our method considers the side of a patch as the smallest possible detail, we can use the detail size for this purpose as well. This is an ad-hoc solution and we use it to illustrate an influence of a larger group depth on both performance and the visual impression. Using the sampling step of 0.5 μm, the ratio increased to 44.1 %. As expected, this is better then in a case of the maximum group depth based on a sampling step and we can expect lower computational times.

In order to introduce the grouping to the algorithm, we modify sorting used by the library of spectrums. We add a secondary key: the depth along the Z-axis, i.e., if both compared patches uses the same spectrum, they will be ordered according to the Z-axis coordinate. As a consequence, we can create groups on the fly and we do not need any additional memory.

Using the both maximum group depths, we calculated $4,096 \times 4,096$ optical field values generated by the scene "Primitives". The sampling step was 0.5 μm and in the reconstruction we focused 6.0 mm, i.e., the cone. The results are presented in Fig. 4.24. Even though the maximum group depth based on the patch size is ten times larger than the maximum group depth based on the propagation distance, the reconstructions does not show any significant disturbances.

Results in Fig. 4.24 might lead us to a conclusion that we can safely use larger group. In such a case, however, we have to estimate the boundary size of a group. A patch with a random phase variation is a complicated object and therefore we have to examine it numerically. Unfortunately, due to the speckle noise, we are not be able to identify presence of holes accurately enough. Therefore, we shall use a maximum group depth based on the sampling step size. Even though it might seem that this is significantly less efficient, in Sec. 4.2.5 we show that the overall performance is almost the same.

[12] Optical field values at the patch are not know to the viewer.

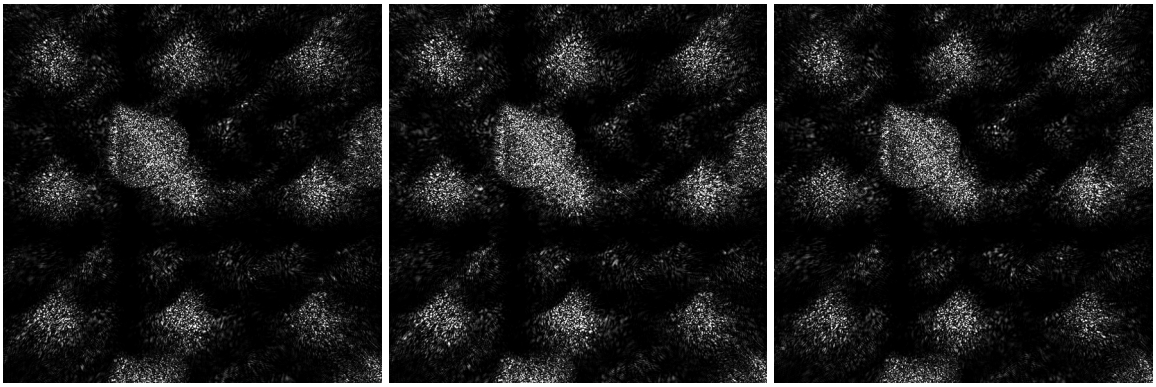


Figure 4.24: Numerical reconstructions from optical fields calculated using a library of 64 samples and (a) grouping disabled, (b) grouping using a maximum group depth along which a propagation equals to a phase shift of optical field values, and (c) grouping using a maximum group depth that is equal to the patch size, i.e., ten times larger than in the case (b).

Besides the results, we also measured times and compared them to the basic version of the method. The results are presented in Fig. 4.25 and they were measured using PC Intel Xeon 3.2 GHz. An inverted relation between the times in the 7.0 μm case is caused by an inverted relation between maximum group depths. According to our measurements, using of the grouping reduces the calculation time and it correlates all our assumptions.

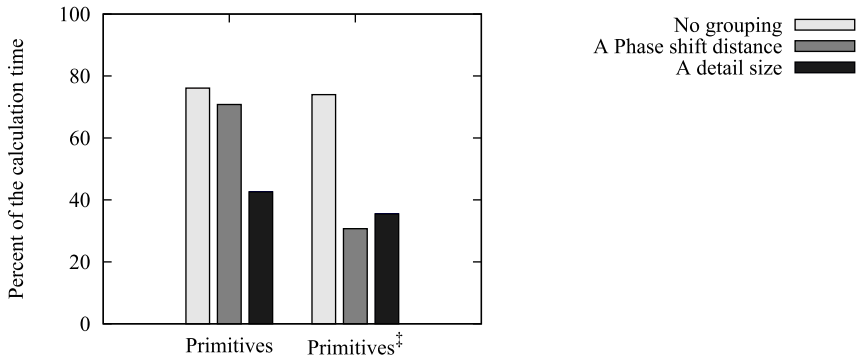


Figure 4.25: A comparison of basic version with the library spectrum version. The library contains 64 spectrums. All times are relative to the computational time of the basic version.

We have addressed a variation in the phase over the patch and reduced it using a library. At the same time, we did not enforce any restriction on the phase, i.e., we can still have use any function to express the phase variation on the surface of a patch. Also, we introduced a quantisation along the Z-axis that further improves efficiency of the spectrum library.

4.2.2 The Visibility by the Frequency Masking

In the previous section we addressed the fact that all patches has the same shape. This led us to pre-calculated angular spectrums and as a consequence we reduced the number of forward FFT executions. In this section we shall further extend the idea. We convert the calculation to the frequency domain and we show that we can approximate the visibility in that domain.

A patch is a part of the plane κ_z. Following the definitions from Sec. 4.1.3, the optical field values at the plane κ_z are $V = M \otimes U$, where U are samples of an optical field of a diffuse and infinite plane and M is a mask that represents spatial limitation of a patch, and $\otimes$ denotes piece-wise multiplication. The angular spectrum of values V is $\mathcal{V} = \mathcal{M} \star \mathcal{U}$, where $\mathcal{M} = \mathcal{F}\{M\}, \mathcal{U} = \mathcal{F}\{U\}$, $\star$ denotes convolution, and $\mathcal{F}\{\}$ denotes Fourier transform. Since we assume that U is based on a random function, we cannot predict anything about its spectrum $\mathcal{U}$. On the other hand, we defined the amplitude mask M as the rect function and therefore its spectrum $\mathcal{M}$ is the sinc function. The unpleasant feature of the sinc function is that amplitude of the function converges to zero but never reaches it. As a consequence, a single frequency in the spectrum $\mathcal{V}$ obtains contribution from all other frequencies due to convolution and thus the original spectrum $\mathcal{U}$ is significantly blurred. Hence, it seems that we cannot manipulate with the spectrum.

Now, let us discuss the effect of a patch. A patch contributes to all cells and hence we may consider the patch as a superposition of contributions to each single cell. Let us now look at the relation between a patch and a single cell g. We shall discuss it in 2D case, i.e. considering only the X-axis and the Z-axis. Extension towards a full 3D case does not require any reformulation. In a 2D case, the goal of the patch is to send energy towards the cell g while limiting influence of a neighbourhood of the cell g. Optical field values G at the cell are

$$G = (U \otimes M) \star H_z, \tag{4.13}$$

where H_z is a spatial domain version of the propagation kernel from Eq. (4.4).[13] The angular spectrum $\mathcal{G}$ of the cell g is $\mathcal{G} = (\mathcal{U} \star \mathcal{M}) \otimes \mathcal{H}_z$. Let us assume the optical field values U contain only a single frequency f_g that represents a plane wave propagating towards the cell as illustrated in Fig. 4.26. Thus, $\mathcal{U} = [\nu(f)]$, $\nu(f) = \delta(f - f_g)A_{f_g}$. Due to the convolution in the expression Eq. (4.13), the angular spectrum $\mathcal{G}$ contains the angular spectrum $\mathcal{M}$ whose central frequency is shifted to the frequency f_g and whose phases are shifted according to the kernel $\mathcal{H}_z$.

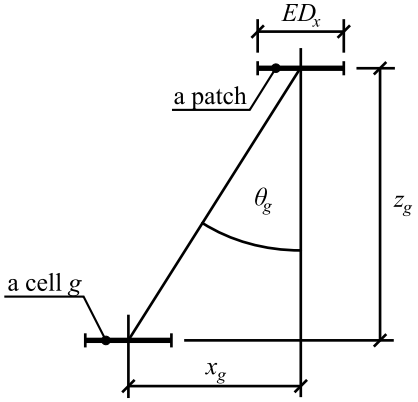


Figure 4.26: A patch and a cell including distanced used by the presented acceleration approach.

The spectrum $\mathcal{M}$ distributes the frequency f_g to the whole spectrum $\mathcal{G}$, i.e., it blurs the spectrum $\mathcal{U}$. For our purpose we would prefer absence of the blur. This, however, would prevent us from getting a predictable shape of a patch in the spatial domain. Nevertheless,

[13] We can use the kernel Eq. (4.4) directly or we can strip down the shift theorem component. In this particular discussion we assume that the shift theorem component is removed. Nevertheless, this assumption is not essential for a success of the discussion.

we can reduce the blur using the windowed-sinc filters [Smi97] instead of the rect function. A spectrum of the filter is finite and as a consequence only a limited number of frequencies are combined together, i.e., the frequency f_g is distributed only to a limited neighbourhood.

A patch, however, contributes to every cell and it forms optical field values G_c at the plane κ. If there is an obstacle that prevents contribution to a cell, resulting optical field values becomes $\mathcal{G}_c - \mathcal{G}$. This solution is accurate but also complex because it requires knowledge of $\mathcal{G}$. Subtraction removes some energy from the spectrum because we the spectrum $\mathcal{G}$ was contained in the spectrum $\mathcal{G}_c$. If spectrums $\mathcal{G}$ had not overlap spectrums of neighbouring cells, it would have been removed completely. We can use this fact and propose an approximation. Since the spectrum $\mathcal{M}$ is limited and a magnitude $|\mathcal{M}|^2$ has a peak over the frequency f_q, there is a high probability the frequency f_g is attenuated significantly.

We follow this mechanism. However, instead of subtracting the blurred frequency f_g from the spectrum $\mathcal{G}_c$, we remove it including a close neighbourhood. As a consequence, we damage contributions of cells surrounding the cell g. We, however, do not damage all contributions in a general case because the spectrum $\mathcal{M}$ is limited and $\mathcal{G}_c$ is a superposition of shifted spectrums $\mathcal{M}$.

Now, let us look at the geometrical shadow the we use to approximate the visibility as discussed in Sec. 4.1. Since the size of a patch is the smallest detail, we can expect that all possible occluders will be much larger than a single patch. Thus, the shadow will influence many neighbouring cells. Therefore, there is high probability that we shall remove the damaged cells later too. At the end, some cells are removed completely, some only partially, some are damaged and some are intact. If the shadow influences most of cells, the patch will be lost in background noise, i.e., it cannot be reconstructed. At the same time this means that such a patch is occluded most of the time anyway and therefore its loss does not harm the visual quality significantly. Hence, we may apply the approximation. Since we mask some frequencies, we denoted this acceleration technique as **the frequency masking**.

The relation between a patch and the cell is reciprocal. Thus, we can assume that the patch was created as product of interaction between cells and its spectrum is a superposition of spectrums blurred by the mask $\mathcal{M}$. Since the number of cells is limited, the spectrum of the patch is limited too. In order to create such a spectrum, we modify the definition of a patch from Eq. (4.3) such that optical field values $V = [v'_{mn}]$ are

$$v'_{mn} = \mathrm{B}\left(\frac{m}{E-1} - 1\right) \mathrm{sinc}\left[4\left(\frac{m}{E-1} - \frac{1}{2}\right)\right] \mathrm{B}\left(\frac{n}{E-1} - 1\right) \mathrm{sinc}\left[4\left(\frac{n}{E-1} - \frac{1}{2}\right)\right] v_{mn}, \tag{4.14}$$

where v_{mn} is an optical field value from Eq. (4.3), $\mathrm{sinc}(t) = \frac{\sin(\pi t)}{\pi t}$, and $B(t) = 0.42 - 0.50\cos(2\pi t) + 0.08\cos(4\pi t)$ is the Blackman window [Smi97]. While this assures that the frequency spectrum of a patch is no longer infinite, it reduces unevenly the energy radiated by PLS of the patch in the spatial domain. The edges of the patch become darker, the centre stays bright. This might influence the visual impression. Nevertheless, if the patches are small enough, the viewer will not recognise a matrix of larger dots. We assume that s/he will detect just a darker surface.

In order to apply the approach, we have identify the index m_{f_g} of the frequency f_g and a neighbourhood that has to be removed. First, we discuss a solution on a plane defined by the X-axis and the Z-axis and then we expand it towards a 3D space. The X-axis component $x_{\mathbf{k}}$ of the wavevector $\mathbf{k}$ that corresponds to the frequency f_g is $x_{\mathbf{k}} = 2\pi f_g$. Since $x_{\mathbf{k}} = \frac{2\pi}{\lambda}\sin\theta_g$

as illustrated with Fig. 4.26 and $f_g = \frac{1}{D_x}\frac{m_{f_g}}{M}$, the index of the frequency is

$$m_{f_g} = M\frac{D_x}{\lambda}\sin\theta_g. \tag{4.15}$$

Let us discuss a simplified case. We designed our method to calculate larger holograms intended for printing. In our first experiment we print a hologram using a cheap technique that enforces a larger sampling step of 7.0 μm. Thus, let us first discuss the this case. If the sampling step is large enough, $\sin\theta \approx \tan\theta$, where θ is the maximum deflection angle defined by Eq. (2.24). Hence, following the illustration in Fig. 4.26, the expression Eq. (4.15) becomes

$$m_{f_g} \approx M\frac{D_x}{\lambda}\frac{x_g}{z_g} \tag{4.16}$$

We estimate the range of neighbourhood indices following the fact that edges of the cell g are located at $x_g \pm \frac{E}{2}D_x$ along the X-axis. Since we can assume that the distance between a patch and a cell along the Z-axis is large in this case, we use just a centre of a cell. As a consequence, we shall remove all frequencies from a range $[m_{f_g} - D_{f_g}, m_{f_g} + D_{f_g}]$, where $D_{f_g} = \frac{E}{2}\frac{M}{z_e}\frac{D_x^2}{\lambda}$ and z_e is an orthogonal distance between the patch and the plane κ.

Using the expression Eq. (4.16), the neighbourhood indices are distributed evenly for a single patch. If united, ranges of frequencies corresponding neighbouring cells create a compact range without gaps. This compact range may not include all frequencies in the spectrum $\mathcal{U}_c$ and thus it may leave some remnants of the patch even though the patch is fully occluded. This happens in a case of patches that are further away from the plane κ. We zero these frequencies because these frequencies allow a periodical copy of a patch to influence the calculated optical field. As a consequence, if the patch is occluded completely, its contribution $\mathcal{U}_c$ to the final optical field will be zero.

If, however, we consider a smaller sampling step, the solution is different.[14] We begin with the expression Eq. (4.15). Since $\sin\theta < \tan\theta$ for $\theta \geq 0$, we have to evaluate it exactly and therefore

$$m_{f_g} = M\frac{D_x}{\lambda}\frac{x_g}{r_g}. \tag{4.17}$$

As a consequence, frequency indices of neighbourhood centres are not distributed evenly and sizes of frequency ranges are not constant. We calculate a corresponding range using range of angles upon which is the patch seen from the cell as illustrated in Fig. 4.27. The resulting ranges overlap each other slightly and they form a compact range without gaps if united as illustrated with Fig. 4.28. Also, frequencies outside this compact range are zeroed. Nevertheless, since we assume that occluded cells form compact neighbourhoods, we may examine a patch just from a centre of a cell. This shrinks ranges such that they touch each other without a gap. Except cells at the edge of the compact neighbourhood, this removes the same range of frequencies. Hence, we use this approximation.

Now, let us expand frequency ranges to a 3D space. In a simplified case described by Eq. (4.16), the problem is separable and the shape of a single range is a rectangle.[15] However, in a general case described by Eq. (4.17), the distance r_g depends on both a relative location of a cell along both the X-axis and the Y-axis and thus the problem is not separable. Shape of a range is a slightly curved and skewed rectangle as illustrated with Fig. 4.28. Nevertheless,

[14] The smaller sampling step allows us to capture objects that are closer to the plane κ. Therefore, one can assume that the smaller sampling step is more desirable.

[15] Actually, if we assume equal sampling steps along both axes, the shape of a range is a square.

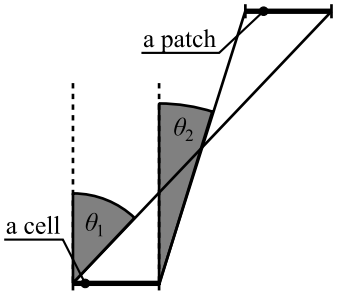


Figure 4.27: Angles used for an estimation of frequency ranges. If we had ignored the diffraction at the edge of the cell, only the plane waves from a range $[\theta_1, \theta_2]$ would have hit the patch.

we can approximate these rectangles using a closed polygon of four edges of a diamond-like shape.

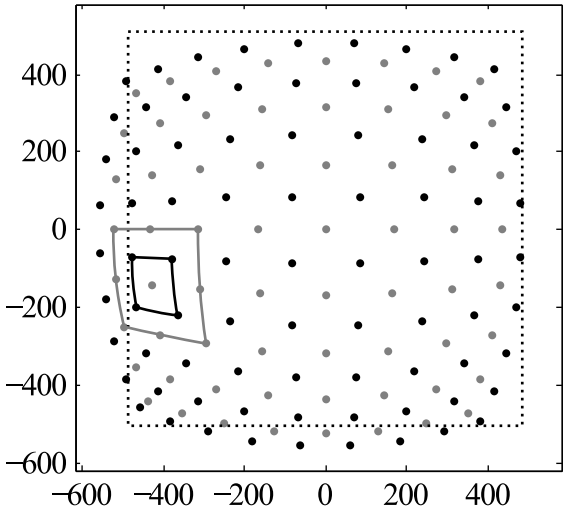


Figure 4.28: Corners of original frequency ranges (gray dots) and corners of used frequency ranges (black dots) including corresponding ranges (solid rectangles). In this case a patch is closer than a distance given by the diffraction condition Eq. (2.24) and therefore some ranges may overlap an edge of the angular spectrum (a dotted rectangle).

In order to introduce the method to the existing algorithm, we have to adjust step 5 and steps 6–7 of the algorithm Alg. 2. The steps 6–7 apply the visibility map and we enhanced steps 6–7 to calculate a frequency mask from frequency ranges. The step 5 calculates the optical field values of a patch. We use the library of spectrums and thus the forward FFT has to be executed only once per each spectrum in the library. Since the number of spectrums is much lower than the number of patches, we can neglect execution time of the forward FFT. Furthermore, the frequency masking operates directly in the angular spectrum and as a consequence the inverse FFT has to be executed only once per optical field. Thus, an execution time of FFT is neglectable if the frequency masking is used.

We tested the frequency masking using various scenes and two setups: one using a larger sampling step and one using a smaller sampling step. Since the smaller sampling step is a more general case, we focused on it. In the reconstruction, we sought for surface artifacts and visibility artifacts.

We tested the larger sampling step case using the sampling step of 7.0 μm, resolution of $6,144 \times 6,144$ samples, a patch size of 32×32 samples, and a library of 32 spectrums.[16] We calculated optical fields of the scene "Primitives2‡" using the frequency masking and using the spectrum library. While looking for artifacts, we shifted slightly the optical field in the XY-plane and reconstructed it using a lens.[17] The results shown in Fig. 4.29 represents a $1,500 \times 1,500$ samples from the centre.

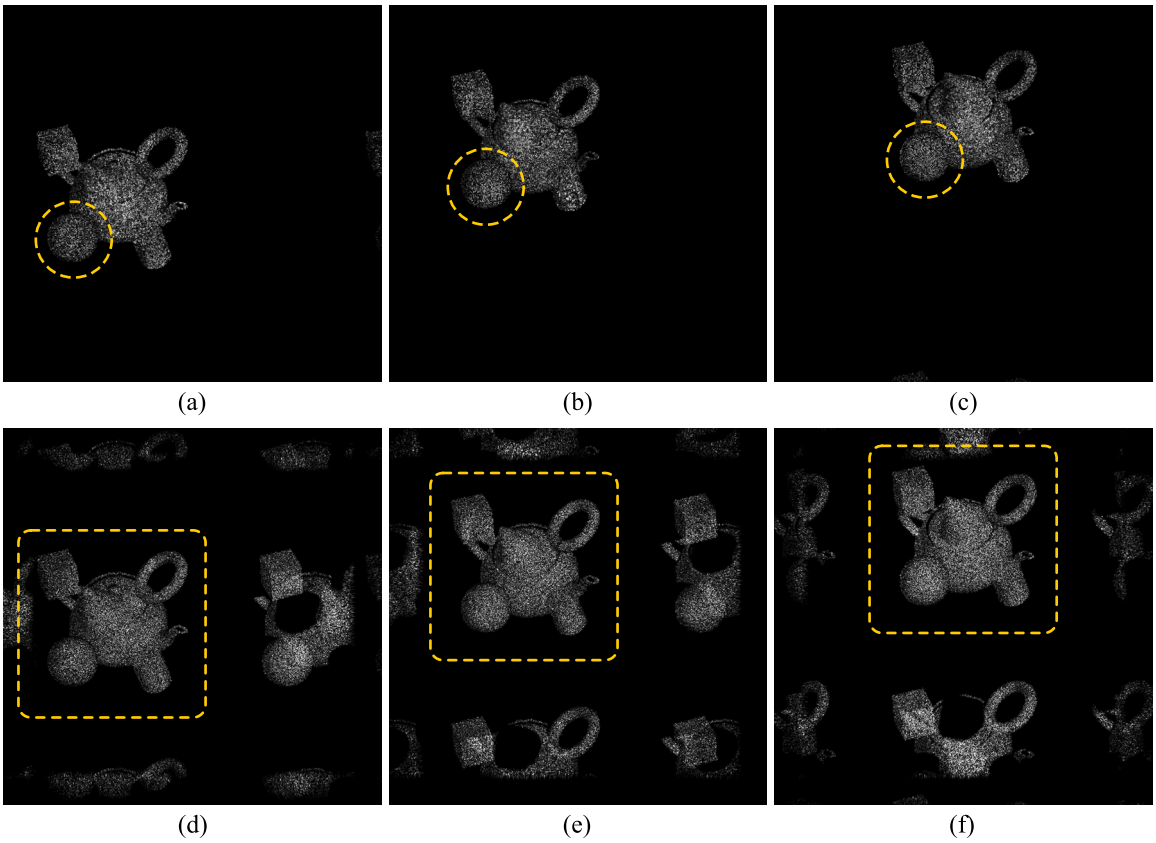


Figure 4.29: Numerical reconstructions of the scene "Primitives2‡" using an optical field shifted (a,d) by (16.1 mm, 0.0), (b,e) by (10.8 mm, 10.8 mm), and (c,f) by (0.0, 16.1 mm). The optical fields in (a–b) were calculated using the frequency masking, the optical fields in (d–f) were calculated using the spectrum library. All reconstruction focus the sphere at 0.725 m. Dashed parts shows either (a–c) the object that is in focus or (d–f) the requested reconstruction.

The results shown in Fig. 4.29(a-c) are similar to the results calculated using only the spectrum library depicted in Fig. 4.29(d-f). There is no visible overlapping or no obvious loss of a patch. There is, however, disturbance on the surface. The surface seems to composed of dots. Despite that the surface is not lost in blur or scattered and therefore we can assume that the viewer will be able to detect the surface.

Also, unlike the spectrum library, the frequency masking loads leads to less noticeable copies as shown in Fig. 4.29(d-f). This is caused by the fact that the frequency masking attempts to remove all frequencies that might allow a neighbouring copy to contribute to

[16] Notice, that in such a case, the patch size is similar to the size of a contemporary LCD, i.e., 0.22 mm.

[17] Since the sampling step was large, we used an pinhole with a radius of 7.0 mm and the distance between the lens and the projection plane was 0.1 m.

the optical field. Therefore, while the neighbouring copy still exists, it might not receive a contribution from the optical field during reconstruction.

We achieved similar results when we use a smaller sampling step. In such a case, we used the scene "Primitives2" and a setup from Sec. 4.2.1.[18] Unlike the larger sampling step, we show a detail of $1,024 \times 1,024$ samples from the centre of the reconstructed image. Numerical reconstruction are shown in Fig. 4.30.

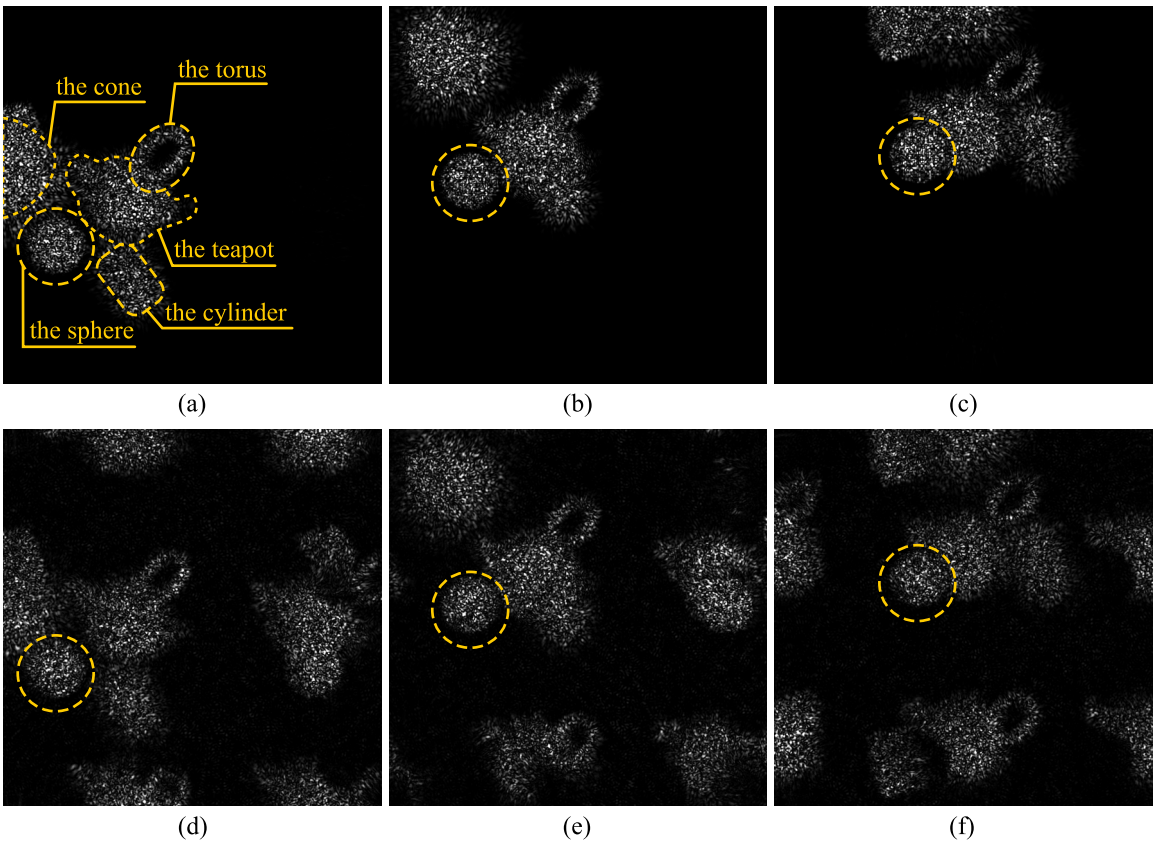
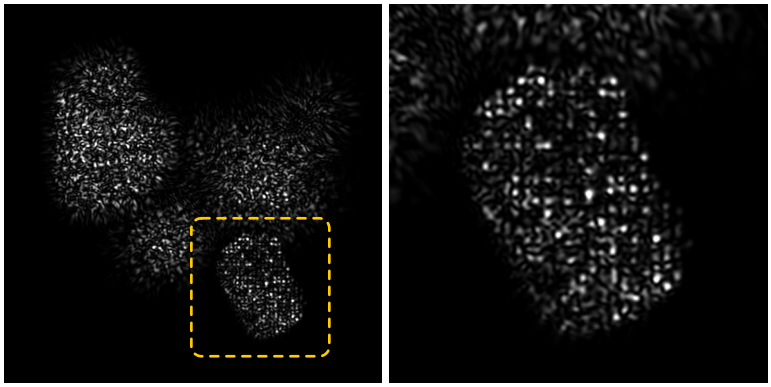


Figure 4.30: Numerical reconstructions of the scene "Primitives2" using an optical field shifted (a,d) by (0.7 mm, 0.0), (b,e) by (0.5 mm, 0.5 mm), and (c,f) by (0.0, 0.7 mm). The optical fields in (a-b) were calculated using the frequency masking, the optical fields in (d-f) were calculated without the frequency masking. All reconstruction focus the sphere at 12.0 mm.

We calculated optical fields using the frequency masking and using the spectrum library and we reconstructed these fields being shifted in the XY-plane. Results depicted in Fig. 4.30(a-c) are very similar to results calculated using the spectrum library alone. The only difference is the surface that seems to have a dot-like structure when enlarged as illustrated with Fig. 4.31. The difference between Fig. 4.29 and Fig. 4.30 is caused by a smaller sampling step in the latter, a different aperture size and a different f_B.[19] Since there are no artifacts except the multiple copies that are very weak, the frequency masking is acceptable approximation.

[18]The setup was: a sampling step of 0.5 μm, a resolution of $4,096 \times 4,096$ samples, a patch size 32×32 samples, a library of 64 spectrums. The radius of the pinhole was 0.5 mm and the distance between the lens and the projection plane was 2.0 mm.

[19]Both the value of f_B and the aperture size cannot be made too small due to a sampling step of 7.0 μm that is significantly larger than 0.5 μm.



(a) (b)

Figure 4.31: (a) A numerical reconstruction of the scene "Primitives2" using an optical field calculated using the frequency masking and (b) a enlarged detail of the reconstruction. Notice the structure the is present in the detail. The reconstruction focuses the cylinder at 8.0 mm.

Following the successful reconstruction, we did an experiment using a different definition of M. We replaced the proposed definition of the patch described by Eq. (4.14) by the original definition described by Eq. (4.3). We calculate optical fields of various scene and reconstructed them. To our surprise, the results, which are shown in Fig. 4.32, did not contain any significant visibility artifacts. Also, the illusion of the surface was not disturbed. The results resembled reconstructions of optical field calculated using the spectrum library alone. This means, the we might be able to use the original definition of the patch together with the frequency masking. Despite that, we decided to use the new and safe definition of M in the rest of the work.

Since we have shown that the frequency masking can provide working optical fields without significantly disturbing artifacts, we can now measure the acceleration we achieved. We used the setup of smaller sampling step and, as usually, we measured computation times using PC Intel Xeon 3.2 GHz. We compared the measured times to computation times of the basic method.

The measurements presented in Fig. 4.33 show an expected fact that there is a reduction of a computation time that is better than in the case of the spectrum library presented in Fig. 4.25. Also, measurements done with the grouping enabled correlate with results from Sec. 4.2.1, i.e., the larger the group, the better the reduction. Despite that we reduce the computation time by removing almost all FFT from the computation, the reduction did not meet our expectations. As it was already discussed in Sec. 4.2.1, this is caused by a lengthy calculation of the propagation kernel described by Eq. (4.4). We address this issue in Sec. 4.2.4.

By applying the frequency masking, we excluded almost all FFT execution from calculation of the optical field. This reduces the computation time. At the same time, it might cause a change in a distribution of the computation time among three major step of the algorithm Alg. 2: calculation of the visibility map (step 3), calculation of an optical field generated by a patch (step 5) and application of the visibility map (steps 6 and 7). Therefore we checked time distribution between these components. Being illustrated with Fig. 4.34, the computation of the optical field is still the dominant step. The only change happens when the grouping is enabled and the size of the group is large. In that case, a larger amount

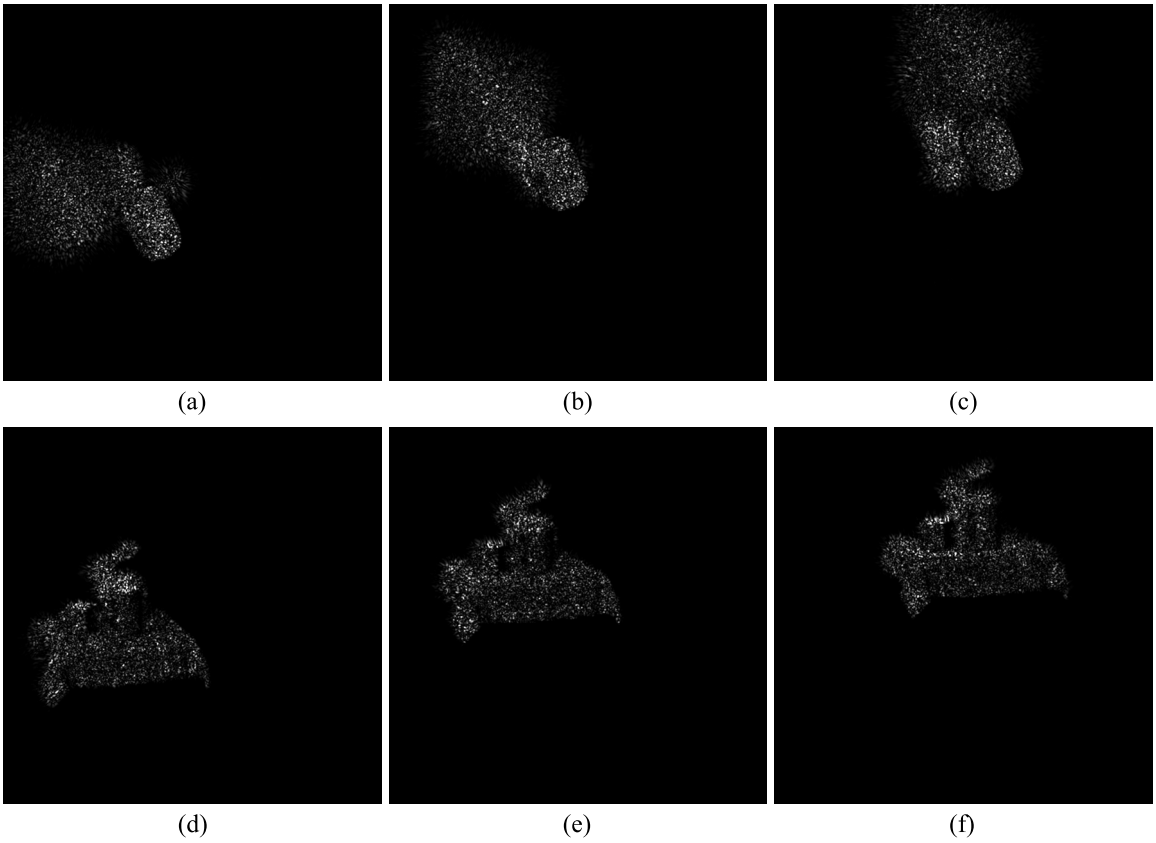


Figure 4.32: (a-c) Numerical reconstructions of the scene "Primitives" focused at the cylinder (12.0 mm) and (d-f) numerical reconstructions of the scene "StillLifeBunny" focused at the corner of the table (9.0 mm). The optical fields were shifted (a,d) by (0.7 mm, 0.0), (b,e) by (0.5 mm, 0.5 mm), and (c,f) by (0.0, 0.7 mm). All fields were calculated using the frequency masking and a patch defined by the expression Eq. (4.3) instead of the expression Eq. (4.14). The image is an area of $1,300 \times 1,300$ samples.

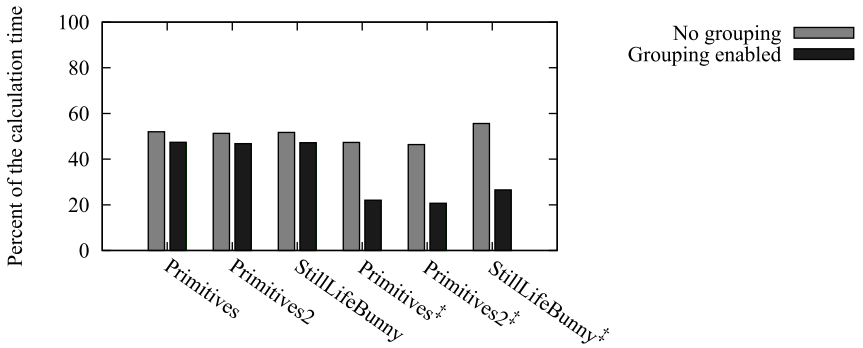


Figure 4.33: A computation times of various scenes using the frequency masking with or without grouping enabled. All times are relative to a computation time of the same scene using the basic version of the algorithm. The maximum group size equals to a distance for which propagation reduces to a phase shift.

of patches skips the step 5. Still, the optical field calculation is the most dominant step. Therefore, we have to still focus further on acceleration of the optical field calculation.

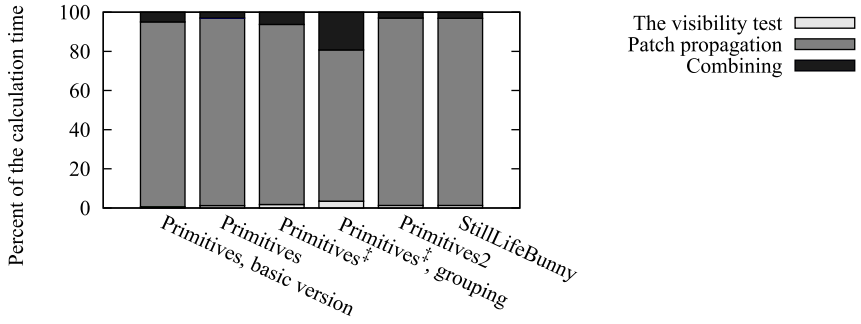


Figure 4.34: A distribution of computational time between steps of Alg. 2. If not noted otherwise, all results we measured using the frequency masking.

Besides the acceleration, the proposed frequency masking has another feature that greatly improves usability of the method. The original solution executes FFT frequently. 2D-FFT accesses every item in the input matrix and therefore the input matrix has to fit into computer memory.[20] This limits the maximum size of the processed optical field. Since the frequency masking does not require a transition to the spatial domain, we can process the angular spectrum by tiles. This is based on an assumption that there are significantly fewer spectrums in the library than the total number of patches. In such a case, the method can afford expensive execution of FFT using the external memory. As a consequence we can add a large frame of zeros about the patch and limit the influence of periodicity.

In this section we presented an acceleration approach that further extends the idea of the library. We have shown that it can be combined with the grouping and we derived that it can calculate optical field that are larger than available memory without an extensive data transfer between the external memory and operating memory. However, this approach did not reduce the dominance of the propagation step over the rest of the steps. Therefore, we focus on the propagation further in the next sections.

4.2.3 Adaptive Sampling

In the previous sections we addressed acceleration of the propagation by removing individual operations. We focused on reduction of FFT executions. In this section we address still propagation but we focus on reduction of data that has to be processed. We show that by applying a feature of the optical field we can significantly reduce computation times without unnecessary degradation of the visual quality.

Let us describe the principle. Following the diffraction condition Eq. (2.24), the sampling step defines the maximum frequency that can be captured by the field and as a consequence it defines the maximum deflection angle. Thus, we can define a distance at which PLS can contribute to all calculated samples of the field at the plane $\kappa : z = 0$ without a risk of an alias. If PLS is closer than that, it will not contribute to all samples at the plane κ and we shall not able to see the PLS from all available viewpoints.

[20]It is true, that 2D-FFT can be executed using 1D-FFT. 1D-FFT is executed over each row. Then, the matrix is transposed and 1D-FFT is executed again. While the first and the third step can be done efficiently using an external memory, the second step cannot because it requires frequent seek operations. Even though there are various schemes, 2D-FFT requires still to access to all items of the matrix almost simultaneously.

Reciprocally, we can calculate a maximum sampling step D' that allows to capture PLS at the distance z' without aliasing as $D' = \frac{\lambda}{2\sin\theta'}$, where

$$\theta' = \tan^{-1}\frac{\Delta}{z'} \tag{4.18}$$

is a deflection angle and $\Delta = \max(MD_x, ND_y)$. Thus, if $D' > \max(D_x, D_y)$, we may calculate a lower number of optical field values and resample the them later to fit the original sampling steps D_x and D_y. This reduces the amount of data that has to be processed but it adds resampling at the same time. Nevertheless, based on results of previous sections, we presume that the resampling will cost less than propagation. Since the sampling step D' depends on a distance z' of a patch, we denote this acceleration approach as **the adaptive sampling**.

Let us now assume that $D_x = D_y$ in the following text, i.e., we shall used the sampling step size $D = D_x$ instead. The size D' of the sampling step can be arbitrary but this will introduce resampling issues. Therefore, we set the sampling step D' to be an integer multiple of both the sampling step D_x and the sampling step D_y.[21] Furthermore, we decided to avoid any other filtering in order to prevent discontinuities that lead to additional copies as shown in [Onu07]. Therefore, we apply a convolution of the samples with the sinc function because this approach does not cause additional noise. The convolution is done in the frequency domain. FFT, which is used for this purpose, is most efficient if a number of processed samples is two to power of some integer. Considering this and assuming that the resulting grid contains two to power of some integer samples, we defined all possible sampling step sizes as

$$D_\omega = 2^\omega D, \tag{4.19}$$

where ω, $\omega \in \mathbb{Z}$ is **the zone number**.

This splits the orthogonal distance from the plane κ to zones. A patch at the distance z' from the plane κ belongs to a zone

$$\omega = \left\lfloor \log_2 \frac{\lambda}{2D}\frac{z}{\Delta} \right\rfloor, \tag{4.20}$$

where $\Delta = \max(MD_x, ND_y)$, i.e., $z' \in [z_\omega,\ z_{\omega+1})$, where z_ω is calculated using the expression Eq. (4.18) and the sampling step D_ω.[22] If $z' < z_0$, we assume that it belongs to the zone $\omega = 0$. This simplifies the resampling even further as discussed below. Since this work is actually a proof of a concept, we simplify the implementation as much as possible. Therefore, we impose a limit on a maximum zone number ω such that the sampling step size $D_\omega \leq ED$.

In order to calculate the zone, we calculate the sampling step D' using the expression Eq. (4.18) and we apply the sampling step to the expression Eq. (4.19). By default, we assumed that $\Delta = D\max\{M, N\}$ in the expression Eq. (4.18). This is not necessary. Due to a difference deflection angle, zone boundary distances z_ω of PLS located over the centre might are different from distances z_ω of PLS aligned to the edge of the visibility grid. As shown in Fig. 4.35, ranges of neighbouring zones overlap and thus we can us higher zones for a given patch. Therefore, we redefine the distance $\Delta = \max_n\{X_n\}$ where X_n is a distance between the patch orthogonally projected into the plane κ and a corresponding edge of the visibility grid as depicted in Fig. 4.36.

[21] Even though this condition seems to be a little bit tricky in a general case, in our experiment we usually assume that $D_x = D_y$ due to the printing device.

[22] We can capture PLS using the sampling step D_ω if the distance z' between PLS and the plane κ is $z' \in [z_\omega,\ z_{\omega+1})$.

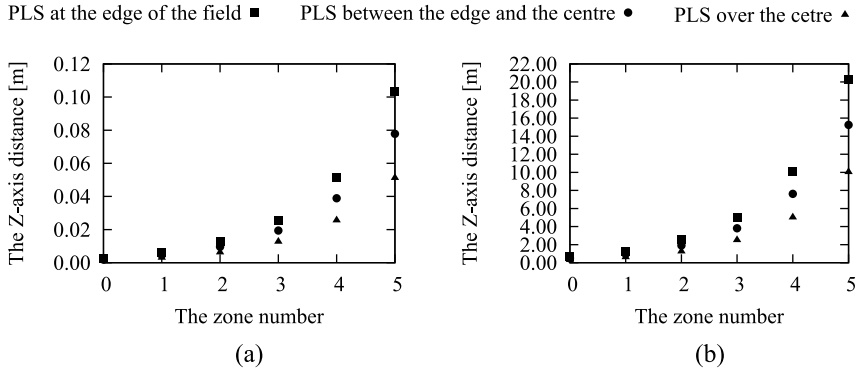


Figure 4.35: A minimum distance for zones calculated using an optical field of $4,096 \times 4,096$ samples and a sampling step D of either (a) 0.5 μm or (b) 7.0 μm. In both cases, the wavelength is 635 nm.

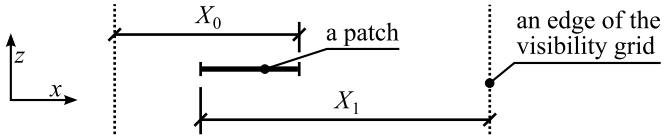


Figure 4.36: A calculation of distances X_0 and X_1 along the X-axis. Distances X_2 and X_3 are calculated similarly along the Y-axis.

Each patch might require a different sampling step. Hence, in order to combine samples of optical fields in grids of various sampling steps, it is necessary to resample them to match the resulting grid of samples. Thanks to the definition of D_ω in the expression Eq. (4.19), all samples in the spectrum of a coarser grid of samples overlaps samples in the spectrum of the resulting grid. Thus, in order to resample through the sinc function, samples of the spectrum of the coarser grid are padded with zeros and converted back to the spatial domain. If done improperly, this may change energy emitted by the processed patch. Therefore we examine the effects of the used sampling scheme on the energy of a patch and we introduce correction coefficients in following paragraphs. For simplification, let us assume that the processed patch belongs to the zone ω and we denote this patch as the under-sampled one.

First, let us examine a plane that is immediately behind a patch because energy emitted by the patch that arrives at this plane is the same as the energy that arrives at any other plane. The arriving energy due to an under-sampled patch is $e_\omega = \int_{P_\omega} |u(\mathbf{x})|^2 d\mathbf{x} \approx D_\omega^2 \sum_m \sum_n |u_{mn}|^2$, where P_ω is an area of the patch, $u(\mathbf{x})$ is a value of an optical field at a point at the plane and u_{mn} is a value of a sample of the coarser grid at the same plane. Since the plane is immediately behind the patch, the value u_{mn} is

$$|u_{mn}|^2 = \begin{cases} I_\omega, & \text{if the sample is inside the patch} \\ 0, & \text{otherwise.} \end{cases} \tag{4.21}$$

Since the intensity I_ω is a multiplicative factor and it is constant for all samples in Eq. (4.21), it can be removed and applied after propagation, i.e., we shall replace I_ω by 1 in Eq. (4.21) for purpose of the following text. Hence, the energy of the patch is $e_\omega = D_\omega^2 (\frac{E}{2^\omega})^2$. Assuming a normalized DFT, the Parseval's theorem [Smi97] states that $\sum_m \sum_n |u_{mn}|^2 = \sum_m \sum_n |f_{mn}|^2 = (\frac{E}{2^\omega})^2$, where f_{mn} is a sample of the spectrum. Since the propagation modifies just the phase, the energy is the same at any distance from the patch. By adding zeros

during the resampling, the energy of the patch becomes $e' = D^2(\frac{E}{2^\omega})^2 \neq e_\omega$. Thus, every sample f_{mn} of the spectrum has to be multiplied by 2^ω before the resampling.

Since $D_\omega \geq D$, the angle θ_ω defined by the diffraction condition Eq. (2.24) applied with the sampling step D_ω is $\theta_\omega \leq \theta$. Hence, the under-sampled patch spreads its energy over a smaller area. As a consequence, if both the original and the under-sampled patch contribute with the same energy, by applying an aperture such as the pupil of an eye, the under-sampled patch becomes brighter. Since this is not acceptable, the energy of the under-sampled patch has to be reduced.

Following Eq. (4.21), energy emitted by a patch is IE^2D^2, where $I = |u_{mn}|^2$ is constant. In the case of the patch, energy that arrives at the plane κ is $IE^2D^2 = I'a$, where a is area of a base of the pyramid at the plane κ as illustrated in Fig. 4.37. In the case of the under-sampled patch, the energy that arrives at the plane κ is the same and it is $I'_\omega a_\omega$. A hologram, however, captures only a fraction of bases of both pyramids. As a consequence, the under-sampled version is reconstructed with higher amount of energy, i.e., it appears brighter. In order to avoid such an unwanted artifact, intensity I_ω of a sample of the under-sampled patch has to be

$$I_\omega = I\frac{a_\omega}{a}, \tag{4.22}$$

where $a = (z\sin\theta + ED)^2$ and $a_\omega = (z\sin\theta_\omega + ED)^2$ are the areas of the pyramidal bases. As a consequence, the amplitude of a sample of the under-sampled patch is $|u_{mn}| = (I\frac{a_\omega}{a})^{1/2}$.

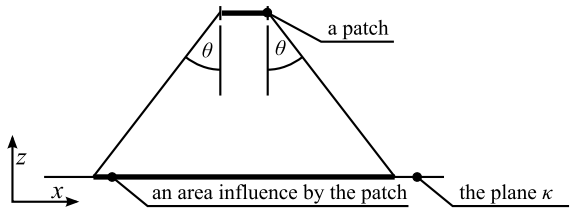


Figure 4.37: A pyramid defined by a patch and a diffraction condition.

The adaptive sampling is compatible with the library of spectra. However, the library of spectra contains spectra of fields sampled with a finer sampling step D. Thanks to Eq. (4.19), samples of a spectrum corresponding to the sampling step D_ω overlap samples of a spectrum corresponding to the sampling step D. Hence, the requested spectrum can be created by zeroing samples that corresponds to higher frequencies. Such an approach, however, fails to comply with Eq. (4.22). After zeroing the frequencies, energy emitted by the patch becomes $\hat{e}_\omega = (D\omega)^2\hat{I}_\omega$, where $\hat{I}_\omega = \sum_m\sum_n |f_{mn}|^2$ is a sum of non-zero samples of the spectrum, instead of $e_\omega = I_\omega E^2D^2$ that is requested energy of the under-sampled patch. Hence, in order to satisfy Eq. (4.22), an amplitude of every sample has to be multiplied by a coefficient c_ω, where $c_\omega^2 = \frac{e_\omega}{\hat{e}_\omega}$. Applying the assumption that intensity $|u_{mn}|^2$ of a sample of the patch is either 0 or 1 if sample is inside the patch, amplitude of every sample of the spectrum of the under-sampled patch has to be multiplied by a coefficient

$$c_\omega = \frac{E}{2^\omega}\left(\frac{a_\omega}{a}\frac{1}{\hat{I}_\omega}\right)^{1/2}. \tag{4.23}$$

In order to implement the adaptive sampling, the algorithm described in Alg. 2 has to be modified only slightly. Both the visibility test and the visibility application stays almost the same. We add a resampling sub-step to the visibility application step in the algorithm Alg. 2. Also, we modify the patch creation step of the algorithm Alg. 2 by calculating the

zone number ω for each patch. For that purpose we use the expression Eq. (4.18). Even though the expression can be simplified in a case of a large sampling step, we did not apply it because calculation of the zone number is executed only once per patch.[23]

The resampling sub-step samples an optical field of a patch using the sampling step D. This allows to sum fields calculated with various D_ω together. In order to avoid long computation times, we decided to sacrifice additional memory. We accumulate separately optical fields generated using a given sampling step D_ω and resample them all at once after all patches are processed. Since $D_\omega = 2^\omega D$, this approach increases memory consumption by one third of a size of an optical field.

Let us now test the proposed acceleration approach. We calculate $4,096 \times 4,096$ optical field values using a patch size of 32×32 samples and a the sampling step of 0.5 μm. If not noted otherwise, this setup is used by all other tests.

First, we test whether the adaptive zones will have impact on calculation time for testing scenes. As it is shown in Fig. 4.38 most of the scene spans over multiple zones and therefore we can expect decrease of the calculation time. Notice that most of scenes are further such that they do not occupy the zone $\omega = 0$, i.e., these scenes could be processed using larger sampling step by default. Therefore, we scaled some scenes and shifted them along the Z-axis so that the span of the scenes includes the zone 0. If not noted otherwise, we distinguish these scene using a symbol $*$ in a superscript, .e.g., the scene "Primitives*" is an adjusted version of the scene "Primitives". As it is shown in Fig. 4.38, these scene occupy the zone 0 and hence they allow us to demonstrate an impact of the adaptive sampling.

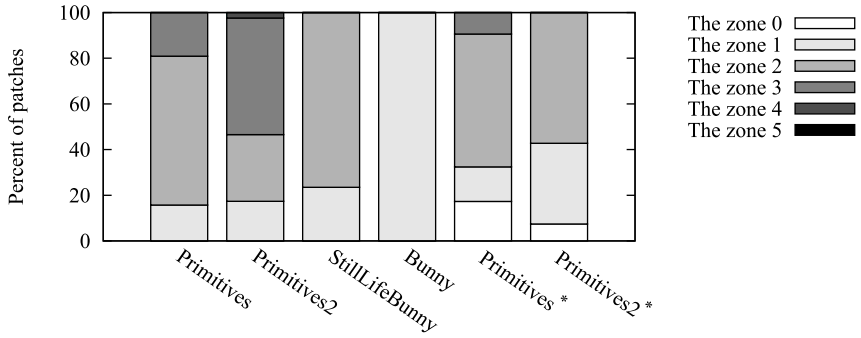


Figure 4.38: A distribution of patches into zones for various scenes.

Next, we verify an impact on the visual quality. We used scenes "Primitives" and "Primitives2" because these two scenes spans over the highest number of zones as shown in Fig. 4.38. We compared numerical reconstruction of optical fields calculated both with and without the adaptive sampling. The calculation applies a library of 64 spectrums without the frequency masking. Since we focus the cube that is obscured by other objects, we shift the field by $(-0.2 \text{ mm}, -0.2 \text{ mm})$ before the reconstruction. The result, which is presented in Fig. 4.39 shows no obvious degradation. The object in focus can be recognised and there is no disturbing intensity artifact. The most significant difference, which is visible in Fig. 4.39, is a lack of some copies. This is a side-effect of the increased sampling step. When the step is increased, the hologram cannot diffract the light towards the copy as it would have been possible if a smaller sampling step had been used.

[23] A sampling step size that is large enough is a step size such that $\sin\theta \approx \tan\theta$, where θ is the deflection angle.

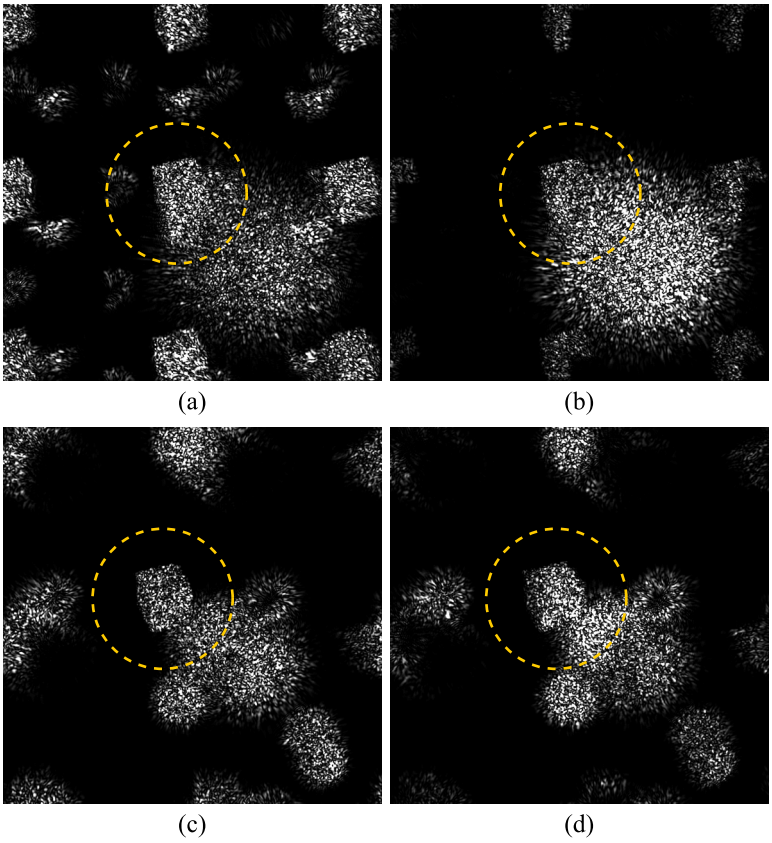


Figure 4.39: Numerical reconstruction of either (a, b) the scene "Primitives" or (c, d) the scene "Primitives2". The optical fields were calculated either (a, c) using the basic version or (b, d) using the adaptive sampling. Dashed circle emphasise the object in focus. The images represent an area of $1,024 \times 1,024$ samples from the centre.

Finally, we tested the impact of the acceleration approach on the calculation time. For that purpose we used the same scenes, i.e., the scenes "Primitives*" and "Primitives2*". For completeness, we include original versions of these scene. The results presented in Fig. 4.40 are relative to the calculation time of the basic version. Following results shown in Fig. 4.38, reduction of the computation time depends on distribution of patches into zones, i.e., the more patches in higher zones, the more significant is the reduction. Nevertheless, even if some patches occupy the zone 0, the reduction is significant.

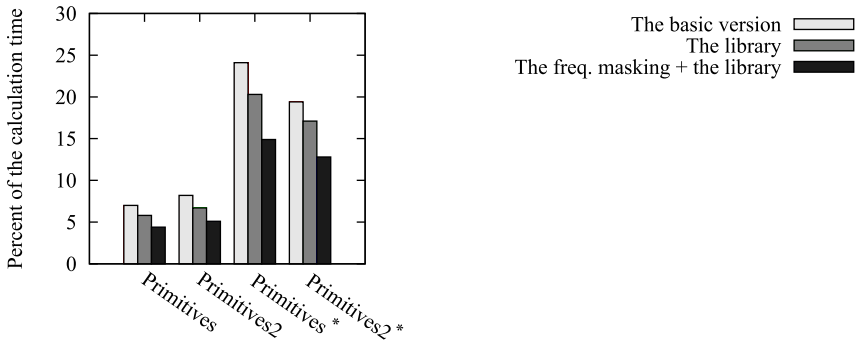


Figure 4.40: A computation times of various scenes using the adaptive sampling. All times are relative to a computation time of the same scene using the basic version of the algorithm.

Since we apply a different sampling step size in every zone, we can define a different maximum group size without slipping to an ad-hoc approach as discussed in Sec. 4.2.1. Using the spectrum library with the adaptive sampling, we measured influence of the grouping on the computation time. The results in Fig. 4.41 shows that the grouping influence has increased. The results correlates distribution of patches into zones depicted in Fig. 4.38. The more patches in higher zones, the better efficiency of the grouping. Despite that the ad-hoc approach is more efficient, in Sec. 4.2.5 we shall show that the efficiency is almost similar the grouping with a maximum depth size depending on a sampling step size.

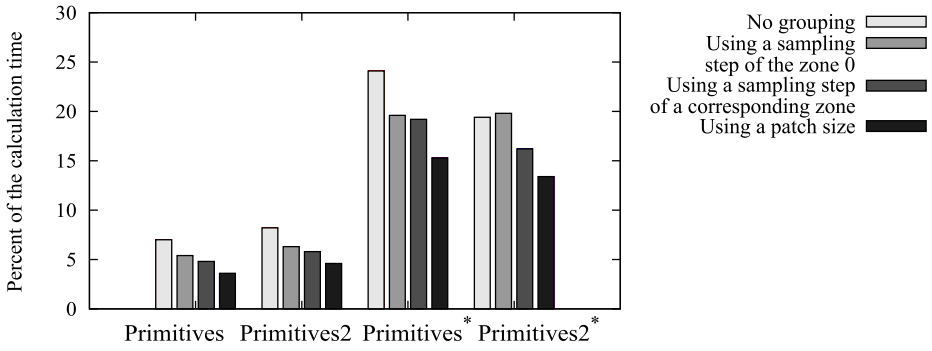


Figure 4.41: A computation times of various scenes using the adaptive sampling and grouping with various maximum group depths. All times are relative to a computation time of the same scene using the basic version of the algorithm.

Since we reduced the number of samples, we measured whether we still have to focus on propagation. According to our measurements shown in Fig. 4.42, the propagation is still the largest fraction of the calculation time. As a consequence, we shall focus on the propagation step even further.

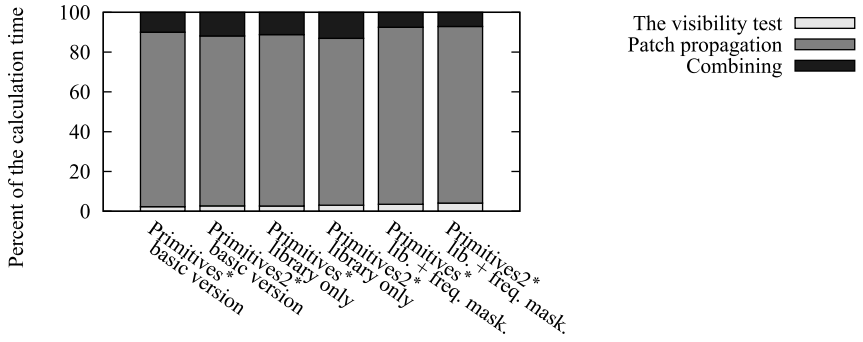


Figure 4.42: A distribution of the calculation time between steps of the algorithm Alg. 2 for various scenes and various applied acceleration approaches. All measurements uses the adaptive sampling. The basic sampling size is 0.5 μm. The measurements were done one PC Intel Xeon 3.2 GHz.

In this section we presented an acceleration approach that uses an adaptive sampling step calculated from a distance of the path. Even though we limited all possible sizes of the adaptive step, we still get significant reduction of calculation times. Furthermore, the approach is compatible with all other acceleration approaches and thus it can be applied together with them.

4.2.4 Low-level Optimisation

In the previous sections we identified the propagation step of the algorithm Alg. 2 as the most time consuming one and we address this issue. In this section we focus the calculating of the expression Eq. (4.4). We modify the calculation to use a single-precision float point data type (floats) and we show that we can use only 32-bit integers to evaluate. Such a modification is crucial for an efficient usage of hardware acceleration through a programmable hardware (FPGA) [IMY+05], streaming SIMD extensions (SSE) [Int07], and graphical processing unit (GPU) [NVI08].

In Sec. 4.2.1 we discussed the fact that application of the phase shift coefficient defined by Eq. (4.4) costs almost the same computation time as it does the rest of the propagation step, i.e., two FFT. The phase shift is plain piece-wise matrix multiplication, which is much simpler than FFT, but it uses a double-precision floating point data type (doubles) for accuracy reasons and it executes functions such a square root, a cosine, and a sine. This slows down significantly evaluation of the expression Eq. (4.4).

First, we address the issue by replacing the sine and the cosine function with two tables. Due to performance reasons, we do not assume any interpolation during extracting of a value from the tables. Following consideration about an acceptable phase error in the Fresnel approximation [Goo05], we set the length of the table N_{sincos} to $N_{\text{sincos}} \geq 2^9$. As a consequence, the quantisation error is approximately $0.7° \ll 1$ rad. Since the sine function is just a shifted cosine function, we refer to both tables as the sine/cosine table.

Next, we adjust the phase shift $2\pi\phi_z$ caused by the distance z to use floats. Following the expression Eq. (4.4), the phase shift is a multiplication $2\pi\phi_z = z_{\mathbf{k}}z$, where $z_{\mathbf{k}} \in [0, \frac{2\pi}{\lambda}]$ is a component of a wavevector $\mathbf{k}$. Let us discuss a direct application of floats. Since the wavelength $\lambda \approx 10^{-7}$ and the phase shift is used as an argument of a complex number, only a fractional part of $\frac{1}{2\pi}z_{\mathbf{k}}z$ is actually needed. The integer part, however, occupies a larger number of bits. In our case the distance $z \approx 10^{-1}$ and thus the integer requires approximately 20 bits. Since the mantissa of floats has 24 bits [IEE85], a maximum rounding error is 2^{-4}. Interpreted as an argument of the cosine function, it equals to an error of up to 11° that much close to 1 rad that the quantisation error discussed above.

Despite that the error seems to be high, we examined it by an experiment. For that purpose, we reformulate the phase shift $z_{\mathbf{k}}z$ to a form $2\pi z_\lambda t_z$, where $z_\lambda = \frac{z}{\lambda}$. Since we replaced both the sine function and the cosine function with tables, we can drop 2π. Besides that, we pre-calculate t_z into a table.

Using the above specified scheme, we propagated a patch. Since we wanted to examine the accuracy, we chose to use a short wavelength of 471 nm. This wavelength corresponds to a blue color and it might be taken as the shortest wavelength used for color holograms. Also, we put the patch at the distance of 0.8 m that can be considered a boundary distance for viewing purposes. As a consequence $z_\lambda = 1.7 \times 10^6 \approx 2^{21}$. Besides that, we aligned the patch to the corner of the boundary rectangle of optical field samples. This eliminates influence of the shift theorem. Following parameters used in this work, we set the sampling step to 0.5 μm, intensity at every sample of the patch to 1.0 and we used a random phase variation of the patch.

We propagated the calculated optical field values back to the patch and we examined intensity because it is the only directly measurable quality of the optical field. We did not apply any lens or aperture because they were not necessary. As depicted in Fig. 4.43(b), the patch is modulated a light noise but the shape is not deformed. Since we know the

exact result of the reconstruction, we can calculate a mean square error (MSE) to express numerically influence of the error. MSE is MSE $= \frac{1}{MN}\sum_m\sum_n(|u_{mn}|^2 - |v_{mn}|^2)^2$, where v_{mn} is a optical field value calculated using doubles and u_{mn} is calculated using floats. In this case, MSE $= 3.7 \times 10^{-7}$, i.e., it is very low. Thus, as long as the patch is closer than 0.8 m to the hologram plane κ, it seems that we can use floats.

(a) (b)

Figure 4.43: A enlarged detail of a numerical reconstruction. The reconstruction was calculated from an optical field of a patch calculated (a) using a full evaluation of the expression Eq. (4.4) with doubles and (b) using tables with floats.

This success is caused by a range of t_z. Since, $t_z \in [0.7, 1.0] \approx 2^{-1}$, the exponent of the floating point number is constant with a single exception at $t_z = 1.0$. As a consequence, the rounding error of t_z is similar for all frequencies and we can interpret it as a slight variation of the distance z in fractions of the wavelength, i.e., the patch is only slightly modulated by noise.

Even though our experiments showed only a slight noise modulation, we proposed an adjustment that improves accuracy. The accuracy issue is a consequence of a large integer part in expression $\phi_z = z_\lambda t_z$. While designing the sine and the cosine table, we declared a minimum table length $N_{\text{sincos}} \geq 2^{-9}$. Thus, we have to assure that the rounding error of the phase ϕ_z is equal or less than $\frac{1}{N_{\text{sincos}}}$, i.e., the representation of the phase ϕ_z has to contain at least χ bits in the fractional part where $2^\chi = N_{\text{sincos}}$.[24] Then, if $z_\lambda \leq 2^{X-\chi}$ where X is a bit length of mantissa, we may calculate safely ϕ using only the multiplication. However, this might limitation that is too strict. Let us demonstrate it using the shortest considered wavelength of $\lambda = 471$ nm, $\chi = 9$ as the minimal bit length of the sine/cosine table, and $X = 24$ for floats. In order to calculate an optical field using this setup we have assure that every patch is closer than $z \leq 15.4$ mm. Such a distance is too short for some scenes used in this work.

Therefore, we propose a redefinition of the distance z_λ as $z_\lambda = \dot{z}_\lambda + \bar{z}_\lambda \times 2^{X-\chi}$, where $\dot{z}_\lambda < 2^{X-\chi}$ and $\bar{z}_\lambda$ is integer. When multiplied by t_z, $\bar{z}_\lambda \times 2^{X-\chi} t_z$ shifts the decimal point of t_z by $X - \chi$ bits right by default. Since $\bar{z}_\lambda$ is an integer and we need only the fractional part of ϕ_z at the same time, the integer part of $2^{X-\chi}t_z$ can be dropped. As a consequence, $\phi_z = \dot{z}_\lambda t_z + \bar{z}_\lambda \bar{t}_z$, where $\bar{t}_z = \text{frac}(t_z 2^{X-\chi})$. Theoretically, the only limitation of the scheme is that $z_\lambda < 2^{2(X-\chi)}$, i.e., for the setup $\lambda = 471$ nm, $X = 24$, and $\chi = 9$ this means $z < 505.7$ m. This is far beyond the considered distance for viewing purposes.

[24] We can assume this because z_λ is constant and $t_z \in [0, 1]$.

We tested the proposed solution using a propagation of a patch at 0.8 m and we obtained MSE of 2.0×10^{-12} that is much lower than in the case without split z_λ. Thus, we succeed to improve the accuracy. The only price that we have to pay for that is additional multiplication and one additional table of $\bar{t}_z$. The measured impact on the calculation is discussed later in this section.

Now, let us examine the shift theorem part of the expression Eq. (4.4). The theorem is separable in variables x and y and therefore we shall discuss only the X-axis case. Since we apply the result of the theorem as an index to the sine/cosine table, the shift theorem is $\eta s \frac{E}{M} N_{\text{sincos}}$, where η is a frequency index along the X-axis, s is a coordinate of a cell g_{st} and N_{sincos} is a length of the sine/cosine table. Since we aim to calculate large optical fields of $M \times N$ samples, $M > 2^9$ and we can assume that $N_{\text{sincos}} = \max\{M, N\}$. As a consequence, the theorem becomes $\eta s E$, where both the index η and the coordinate s are integers. Since we need just χ lower bits of $\eta s E$, we can calculate it using 32-bit integers. The only limitation of the scheme is that M has to fit into a 32-bit integer, i.e., the edge of a hologram is limited to 10^{10} m for a sampling step of 0.5 μm. This is more than sufficient for purposes of this work.

We have shown that the shift theorem can be evaluated in 32-bit integers. Furthermore, we can show that the phase shift ϕ_z can be evaluated using 32-bit integers too. Since we need just the fractional part of ϕ_z, multiplication $z_\lambda t_z$ can be done in 32-bit integers if and only if both the fractional part of t_z and fractional part z_λ fit altogether in 32-bits. Under such condition, the overflow influences only the integer part of the result that we drop anyway. Since z_λ is a distance, a rounding error of z_λ is just a shift by a fraction of a wavelength along the Z-axis and thus we can assign z_λ much shorter fractional part.

We test the proposed solution using the same setup as in the previous cases, i.e., a patch at a distance of 0.8 m. We assigned 28 bits to a fractional part of t_z and 3 bits to a fractional part of z_λ. The resulting MSE of 5.8×10^{-8} is better then in a version using floats but it is worse than a version that uses a slit up z_λ. Hence, the improvement is caused by additional bits available for representing t_z. The limitation of the scheme is that the distance z_λ has to fit into 32-bit integer including the fractional part. If we left 3 bits for the fractional part and we use a wavelength of 471 nm, $z < 126.4$ m. Such a limitation is far beyond considered distanced for viewing purposes.

In order to test the proposed solution we used the version with the highest MSE, i.e., the version with a compact z_λ and floats. We used two setups. In both setups we use a wavelength of 635 nm. First, we took a small sampling step of 0.5 μm and a scene "Primitives" whose furthest object is at 30.0 mm. In such a case, the distance z is only slightly larger than the distance that allows a direct use of multiplication. Next, we took a larger sampling step of 7.0 μm and a scene "Primitives2‡" whose furthest objects is at 0.8 m. In such a case, the distance z_λ is large and a difference between t_z for neighbouring frequency indices is in lower bits since $t_z \in [0.99, 1.00]$, i.e., the accuracy of t_z will influence the result significantly.

We reconstructed the fields using a lens and a pinhole and we compared them visually to reconstruction from fields that were calculated using doubles. In both setups we focused the furthest object.[25] The results calculated from the scene "Primitives" show no visible difference. Unlike that, results calculated from the scene "Primitives2‡" depicted in Fig. 4.44 contain loss of details around the rim of the teapot lid. The detail in Fig. 4.44(c) shows

[25]For a sampling step of 0.5 μm, we used a pinhole with a radius of 0.5 mm and we shifted the optical field in the XY-plane such that the focused object (the torus) was not obscured significantly. For a sampling step of 7.0 μm, we did not use a pinhole and similar to a previous case, we shifted slightly the field in the XY-plane.

additional noise. Besides that, the surface of the teapot is not disturbed and there are no intensity artifacts as illustrated in Fig. 4.44 (b). Nevertheless, since the size of the gap is comparable to the size of the patch, we can neglect the error. Thus, it seems that the use of floats and tables can lead to recognisable results even for larger scenes. However, as we show below, the performance impact of slit z_λ can be neglected.

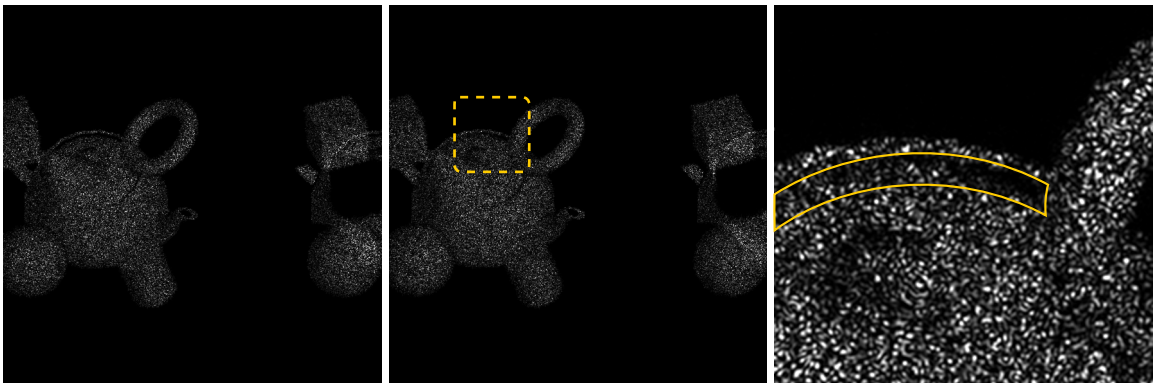


Figure 4.44: Numerical reconstructions of the scene "Primitives2‡" that is focused at the teapot (0.8 m). Numerical reconstructions uses optical fields that were calculated either (a) using doubles for evaluation of the expression Eq. (4.4) or (b, c) using floats and tables. (c) A detail, which is emphasised by a dashed rectangle in (b), shows additional noise in the gap at the rim. The shape in (c) bounds the gap that is clearly visible in (a). The result represent an area of $1,024 \times 1,024$ samples from the centre.

As shown in Fig. 4.45, the improvement of the computation time is approximately 2/3 of the calculation time using doubles.[26] Even if we used the version with the split distance z_λ, we obtained similar reduction of the calculation time. Hence, the split z_λ can be used without a significant negative effect on performance. Since we reduced the calculation time of the propagation step, we measured distribution of times between steps of the algorithm Alg. 2. As shown in Fig. 4.46, the ratios did not change and therefore there is still no need to accelerate other steps than the propagation one.

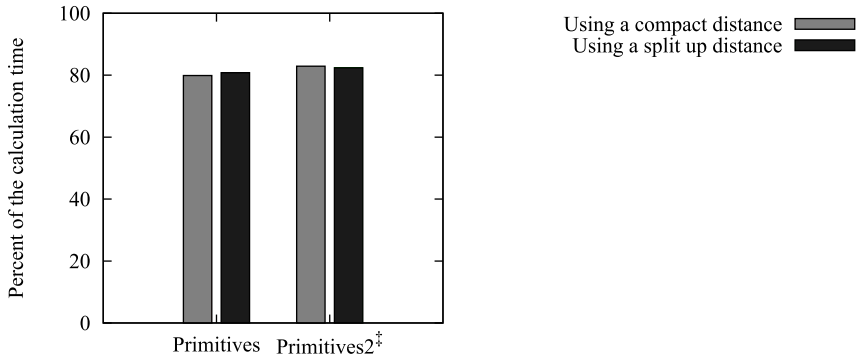


Figure 4.45: A comparison of calculation times. All times are relative to a corresponding calculation time using doubles.

In this section we presented acceleration through using of a shorted data type. We showed that the evaluation of Eq. (4.4) can be done using 32-bit integers or floats. This allows us to

[26] For measurements, we used the same PC as in the previous cases, i.e., PC Intel Xeon 3.2 GHz.

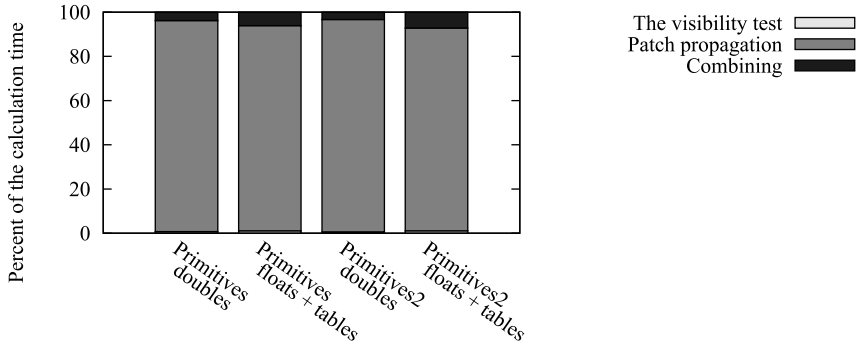


Figure 4.46: A distribution of calculation time into steps of the algorithm Alg. 2.

use hardware acceleration means such as FPGA, SSE, or GPU. The presented acceleration can be combined with other approaches to reduce the computation time even further. This fact is discussed in a following section.

4.2.5 Summary

In the previous section we presented approaches that decreases computation time. Some approaches are more efficient, some less. In this section we show that these approaches can be combined and their effect on the computation time is cumulative. This section closes the discussion about considered acceleration approaches of the detail driven method.

Following measurements from the table Tab. 4.3, we focused on acceleration of the propagation step. We measured reduction achieved by proposed approaches by calculating optical fields of $4,096 \times 4,096$ samples. For that purpose we used a sampling step of 0.5 μm and scenes: "Primitives", "Primitives2", "Bunny", "StillLifeBunny", "Primitives*", and "Primitives2*".[27] We chose to use a patch resolution of 32×32 samples because if we had used a sampling step of 7.0 μm, the size of such a patch would have been equal to a pixel size of a contemporary 17" LCD display.

Similar to previous sections, we expressed the measured times as a fractions of corresponding times of the basic version. The table Tab. 4.4 summarised worst-case acceleration for all discussed approaches. Since some approaches can be combined, we presume that maximum reduction of the calculation time is 12 % without the frequency masking and 8 % with the frequency masking. We verified the presumptions by an experiment.

We verified the presumption calculating optical fields of $4,096 \times 4,096$ samples from the scene "Primitives*". Since we presumed a different reduction of the calculation time for a case with the frequency masking and without it, we measured two sets of results on PC Intel Xeon 3.2 GHz. For each measurement we enabled an additional acceleration approach and we present the measured times in Fig. 4.47. Thus, by applying all acceleration approaches we were able to reduce the time to 9.8 % with the frequency masking and 15.6 % without it. This correlates with the presumption and thus we can state that individual approaches does not influence each other significantly.

Since we aimed the propagation step most of the time, we verified whether it is not necessary to accelerate other step as well. As showed in Fig. 4.48 the propagation step is

[27] Following the section Sec. 4.2.3, we denote scenes that were shifted and scaled in order to occupy the zone 0 by a symbol '$*$' in superscript.

Table 4.4: The worst-case reduction of a computation time using proposed accelerations. All values are relative to a computation time of the basic version and rounded up. If a library is applied, it will contain 64 spectrums.

The acceleration approach	Reduction
The library	76 %
The frequency masking with the library	52 %
The adaptive sampling	24 %
with the grouping	19 %
Propagation using floats and tables	82 %

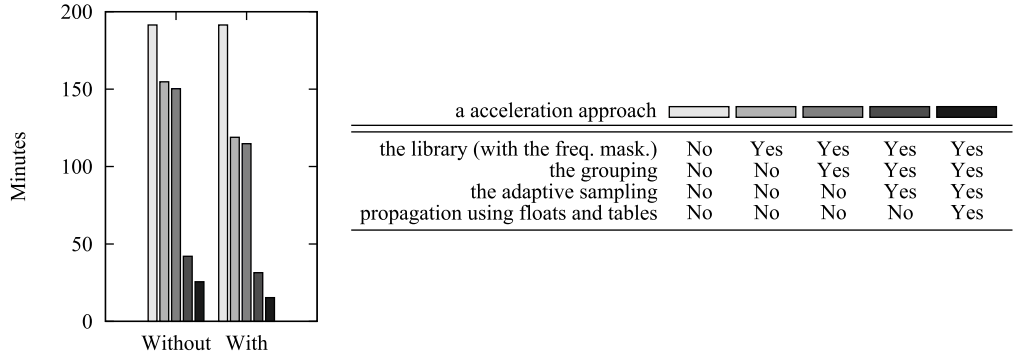


Figure 4.47: Calculation times of the scene "Primitives*" measured with and without the frequency masking and other acceleration approaches. The library consists of 64 spectrums and all times were measured using PC Intel Xeon 3.2 GHz.

still the most time consuming one. In the case of the scenes "Primitives", "Primitives2", and "StillLifeBunny" the propagation step is reduced much more but it is just a side-effect of the minimum distance in the scene. As discussed in Sec. 4.2.3 and illustrated with Fig. 4.36, no patch of these scenes occupies the zone 0. Therefore, they cannot be used a representative case and thus it is meaningful to address the propagation part in the future work.

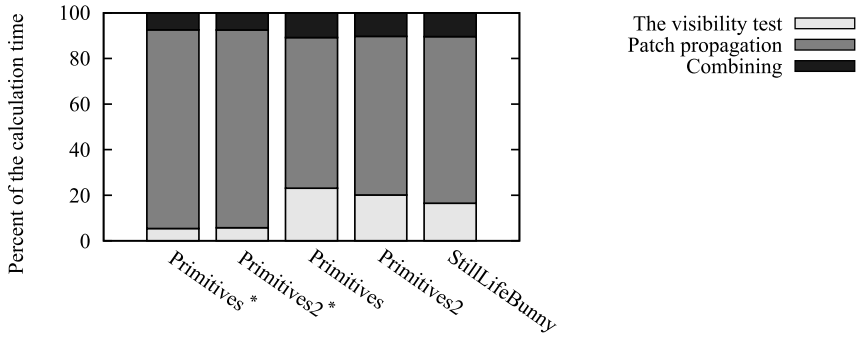


Figure 4.48: A distribution of the calculation time between steps of the algorithm Alg. 2 for various scenes. All measurements use the frequency masking with applied acceleration approaches such as the grouping, the adaptive sampling, and propagation using floats and tables. The basic sampling size is 0.5 μm. The calculation were done one PC Intel Xeon 3.2 GHz.

By applying all acceleration approaches, we reduced the computation time significantly as illustrated with Fig. 4.49. The scenes "Primitives", "Primitives2", and "StillLifeBunny" are accelerated more than the rest because they do not occupy the zone 0 as it was discussed in Sec. 4.2.3. Furthermore, the results in Fig. 4.49 shows that the final impact of the add-hoc maximum group depth differs from impact of the maximum group depth based on the sampling step size by less than a percent. This is caused by the fact that the grouping affects only propagation of a patch, the rest of steps has to be executed without any reduction. Since we reduced computation time spend on propagation, we reduced the impact of the grouping.

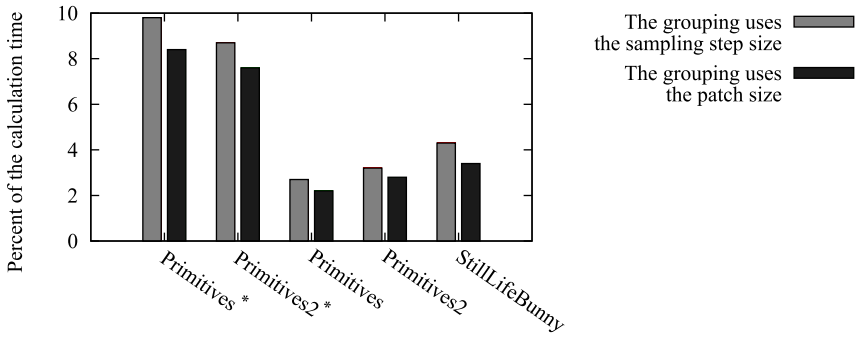


Figure 4.49: Computation times of various scenes using both the grouping with different maximum group sizes and the frequency masking with applied acceleration approaches. All times are relative to a computation time of the same scene using the basic version of the algorithm. The calculation were done on PC Intel Xeon 3.2 GHz

Besides the reduction of the time we have to address the visual quality of reconstructions. For that purpose we calculated optical fields of scenes "Primitives*" and "Primitives2*" with the frequency masking as the most efficient approach. Since the latter scene contains small objects, which might be significantly disturbed by the frequency masking, we also included the scene "Primitives2". We reconstructed the optical field using lens[28] and we compared the reconstructions visually with the results without the acceleration approaches enabled. In all reconstructions we focused the cube. As shown in Fig. 4.50, no shape of object in focus exhibits any deformations. Also, we do not observed any significant overlapping or any significant hole caused by visibility error. With an exception of Fig. 4.50(b) there are no significant intensity artifacts. The intensity artifacts in Fig. 4.50(b) are caused by the fact that the object is too small for the frequency masking and these artifacts are not present using the enlarged scene in Fig. 4.50(c).

Throughout the sections about the acceleration, we did not address directly use of brute force acceleration approaches through distributed computing, CPU streaming SIMD extensions, and graphical processing unit (GPU) because it is an implementation issue. The structure of the algorithm Alg. 2 allows independent processing of patches. In fact, patches can be processed in any order and we utilise this fact to implement efficiently the proposed acceleration approaches. This feature is similar to the ray-base solution [JHO08] in which we have shown that distributed environment leads to almost linear reduction of the calculation time.[29]

[28]The radius of the pinhole was 0.5 mm and the distance between the lens and the projection plane was 3.0 mm.

[29]In another words, by implementing the approach on a cluster of computers we would have verified an obvious fact. As a consequence we would have wasted valuable time on doing stuff that has only a little, if any, potential for scientific publication.

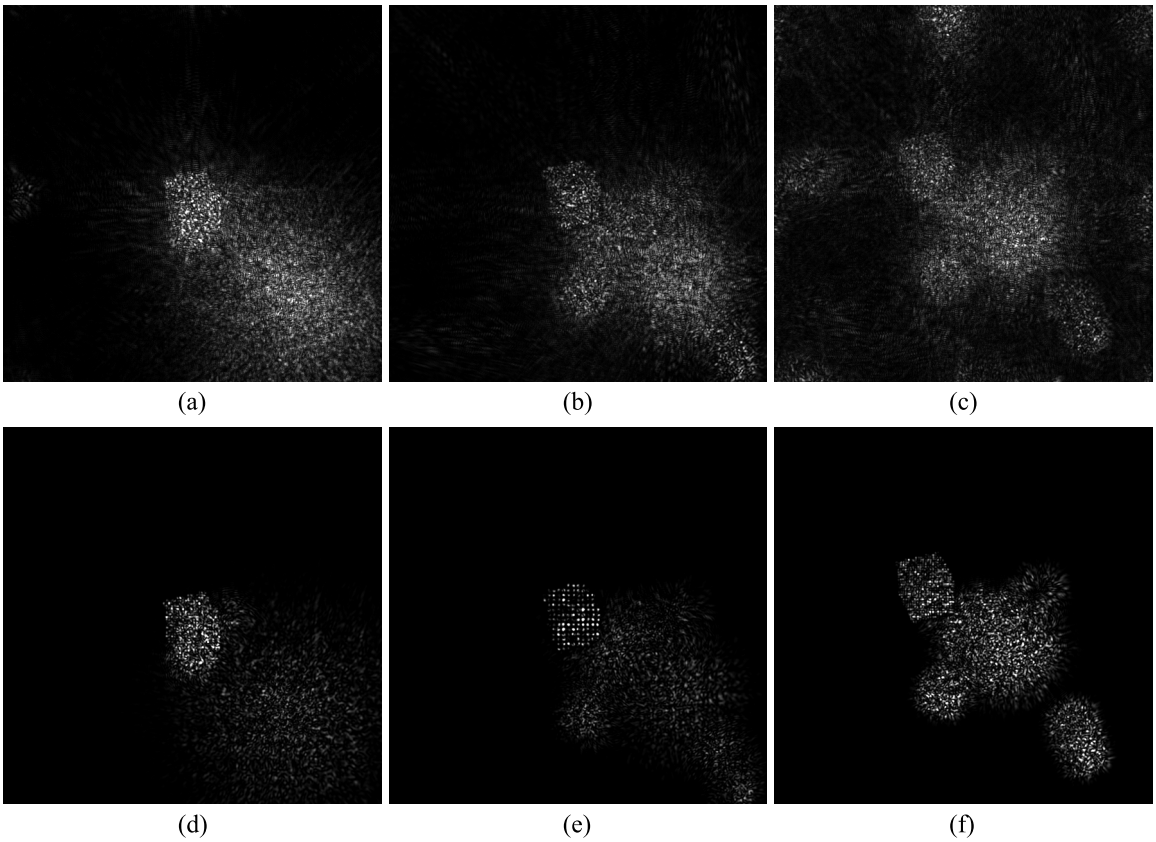


Figure 4.50: Numerical reconstructions of optical fields calculated from scenes: (a,d) Primitives*, (b,e) Primitives2*, and (c,f) Primitives2. The optical fields were calculated using either (a–c) the basic (non-accelerated) version or (d–f) the frequency masking with all acceleration approaches enabled. In all cases, the optical fields were shifted by $(-0.2\text{ mm}, -0.2\text{ mm})$ and we focused the cube at (a,d) 9.0 mm, (b,e) 5.0 mm, (c,f) 10.0 mm. The image represents a square of $1,024 \times 1,024$ samples containing all objects of the scene.

In this section we showed that we can combine the acceleration approaches freely and that their effect are cumulative. Even though the resulting calculation is approximately just $10\times$ faster than the original algorithm, the reduction is significant in absolute numbers. This closes the discussion about the acceleration of the detail drive method. Other possible means such as a programmable hardware should be considered as a future work.

4.3 Pillar Sidewalls

The presented method decomposes the virtual scene to patches and we process them separately. While this allows efficient acceleration, it also causes the major weakness. We process only patches and use parallel and orthogonal rays to create them. As a consequence, we are not able to capture properly large planes that are almost perpendicular to the hologram plane. In this section, we address this weakness. We propose to use additional patches and we show that such an approach is able to significantly reduce this weakness of our method.

The method decomposes the scene into pillars and it assigns a patch to each front cap of a pillar. However, this ignores sidewalls of pillars. As a consequence, a patch that is long enough and that is at sides of an object will prevent light from passing trough but it will not emit light. In this text, we refer to this artifact as the black hole. While this would not be a problem in a case of a single standalone pillar, in a case of a group of such pillar, the resulting effect might be disturbing for a viewer. Thus, we have to add an emitter to a sidewall of a pillar.

We examined two approaches: using sidewall segments and using auxiliary patches. The sidewall segment approach is the most straightforward one. We introduce an emitter in a shape of a sidewall for each sidewall. While this solution seems to be perfect, it has two issues: calculation of an optical field generated by a sidewall and estimation of sidewall visibility. The latter is caused by a size of a sidewall. We expect that pillars causing dark holes are long and thus we cannot approximate the visibility of a whole sidewall by a single boolean value. Therefore, we have to split a sidewall to sidewall segments and we estimate the visibility of each segment independently. In order to minimise overlapping due the visibility estimation, we limit the size of a sidewall segment to a size of a patch. Hence, there is a high probability that we will generate a lot of sidewall segments per a pillar.

The issue of optical field is caused by the spatial structure of the sidewall segment. A sidewall segment is a part of a plane perpendicular to the plane κ. Despite that, we cannot just rotate the angular spectrum as it is proposed by the wave-based methods described in Sec. 3.1.2. This is caused by the fact that FFT assumes periodicity, i.e., the segment is repeated infinitely on the plane perpendicular to the plane κ, at which we evaluate the optical field. This effect cannot be avoided, it can be only reduced by padding the segment by zeros. Even in this case, copies will be still present. Thus, we have to use a cloud of PLS and a geometry-based method that are very slow. This will increase calculation time. Nevertheless, we can significantly reduce its influence by pre-calculating optical field of sidewall segments and reuse them as a part of the spectrum library described in Sec. 4.2.1.

Facilitating the spectrum library, we have to have additional data files that depends on resolution of the optical field and the sampling step. Besides that, we presume that a regular scene will require a large number of sidewall segments because a size of a sidewall segment is similar to a patch size and depth of a scene is much greater than the patch size. This might harm the performance and thus we searched for another solution that is able to adaptively distribute additional structures and that does not need any other shape than a patch. Since

the patch is parallel with the plane κ, it is both faster and less prone to error to calculate optical field of a patch than it would have been if the patch had been a rotated plane as it was discussed in Sec. 2.2.3.

Let us examine the purpose of an emitter. A structure used as an emitter has to assure that any ray that is shot from an optical field sample towards the scene will hit the structure prior entering a volume of an object. Observing this, we decided to distribute patches along the pillar that is at the edge of an object and use these patches as emitters. We denote these additional patches as the auxiliary patches.

The distance between auxiliary patches is given by rays shot from edges of a bounding rectangle that encloses optical field samples. No other ray that hits the auxiliary pillar have a larger angle from the normal of the plane κ. Illustrated with Fig. 4.51, the distance D_n between two consecutive auxiliary patches e_n and e_{n+1} that are generated inside a pillar p^d_{lo} is

$$\begin{aligned} D_{n+1} &= x_c \tan \alpha_{n+1}, \\ \alpha_{n+1} &= \frac{1}{z}(z^d_{lo} + z_n), \\ z_{n+1} &= z_n + D_{n+1}, \end{aligned} \tag{4.24}$$

where the initial distance z_0 is $z_0 = 0$, z^d_{lo} is an orthogonal distance between the patch e^d_{lo} and the plane κ, $x_c = \min\{D_x, D_y\}E$ is the minimal size of a patch, and $x = \max\{l+L/2, o+O/2\}$ is the maximum distance to the edge of the bounding rectangle of the visibility grid containing $L \times O$ cells. Notice that our definition of the distance D_n assures that the auxiliary patches are sparser with increasing distance from the plane κ.

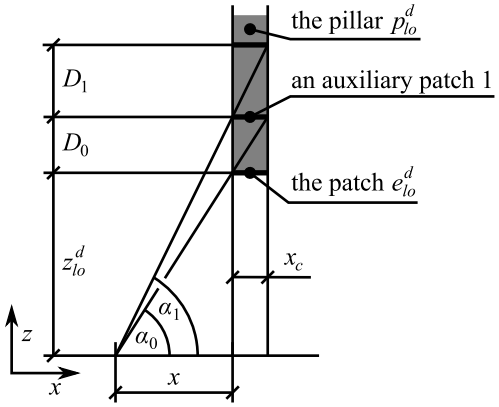


Figure 4.51: Distribution of auxiliary patches inside the pillar p^d_{lo}.

Since an auxiliary patch e_n inside the pillar p^d_{lo} is similar to a regular patch, we use almost the same approach to estimate the visibility. The only difference is that we remove temporarily the pillar p^d_{lo} prior the evaluation. As illustrated with Fig. 4.52, we can estimate the visibility accurately only for a the cell g_{lo}. The rest of cells lays in a half-shadow created by the preceding auxiliary patch e_{n-1}, where $e_0 = e^d_{lo}$. Since we do not support a partial visibility in the method, we decided to follow the visibility test of the method, i.e., an auxiliary patch will be fully visible if the centre of the auxiliary patch sees the centre of a cell. However, introduces overlapping artifacts that we shall discuss later.

We generate auxiliary patches just after we create pillars. After calculating a location of an auxiliary patch inside the pillar, we test whether it is not surrounded by other pillars in

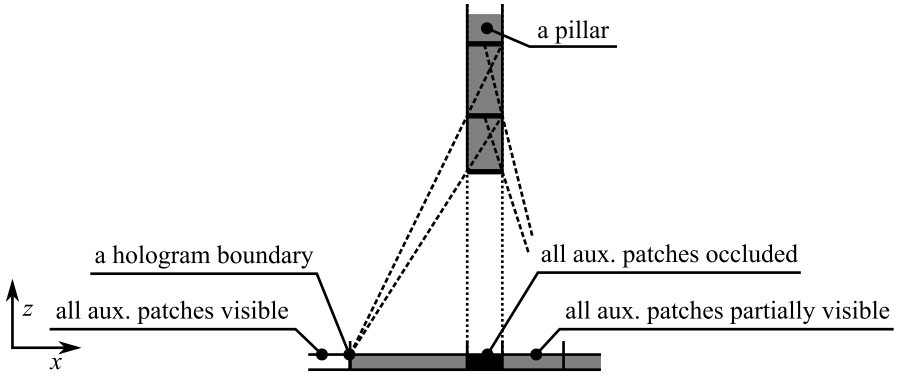


Figure 4.52: Partially occluded auxiliary patches. Decreasing the angle between a direction of a ray and the normal of the plane κ, the dashed lines show the first ray that hits an auxiliary patch from the edge of the hologram.

the 4-neighbourhood.[30] If it is not surrounded, we estimate intensity of the patch and set its phase variation, otherwise we drop the auxiliary pillar.

Similar to regular rays, we estimate intensity of auxiliary patches through ray-casting. We shoot rays from a centre of a patch along the X-axis and the Y-axis and at the intersection point with the object, we calculate intensity as illustrated with Fig. 4.53. In general case, we have to shoot four rays per an axis per a patch. Following the 4-neighbourhood test, we shoot only two rays usually. If we retrieve multiple results due to multiple rays in multiple axes, we select an intersection that is closest to centre of the ray. This might be similar to a random-like picking of an intersection. However, we can afford it because our methods ignores details and intensity variations that are smaller than a patch.

Furthermore, since we shoot rays along axes and origins of rays is at centre of patches, we can search for intersection in 2D. Let us now examine only rays along the X-axis. We slice the scene by the plane $\mu_o : x = (o + \frac{1}{2})ED_y$ and use the slice to calculate intersection between the scene and rays generated from auxiliary patches inside pillars p^d_{lo}, for all d and for all $l \in [-\frac{L}{2}, \frac{L}{2} - 1]$. The approach for Y-axis rays is similar. Since the slice are equidistant, we employ a fast iterative slicer [JHS06].

We assume a diffusive surface. Thus, we use a random phase variation on an auxiliary patch. Nevertheless, a random phase variation is not as appropriate for this purpose. A random phase causes that a patch emits lights in almost every direction. Applied to auxiliary patches, we might experience overlapping of auxiliary patches in reconstructions. Hence, we have to modulate the phase variation of a patch. This is a problem of digital diffusers. Since digital diffusers are out of scope of this work, we shall not address the issue here.

Since an auxiliary patch is handled similar to a regular patch, we can expect increase of computational time. The increase will depend on amount of auxiliary patches. For that purpose, we checked a number of additional auxiliary patches for various testing scenes. As it can be seen in the table Tab. 4.5, most of the scene has a low number of auxiliary patches. The only exception is the scene "Bunny", where the majority of auxiliary patches fill the base of the bunny. Thus, for testing of influence on a visual quality, we shall use only the scene "Bunny".

[30] We do not expect a chessboard like structure created by pillar. If, by any occasions, such a structure appear, the visibility test will prevent a hidden auxiliary patch to influence the result.

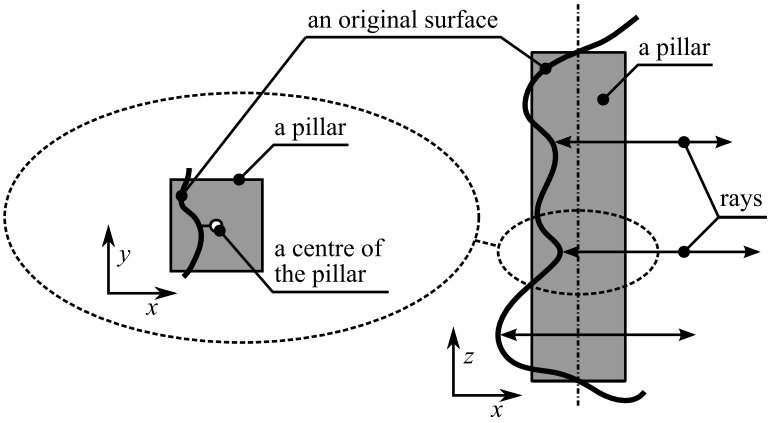


Figure 4.53: Obtaining intensity of an auxiliary patch from known location of the auxiliary path inside a pillar.

Table 4.5: Increase of number of patches due to additional auxiliary patches and a maximum number of auxiliary patches per a pillar. All scenes fit an optical field of $4,096 \times 4,096$ samples. Scene are ordered according to the increase of the auxiliary pillars.

Scene	*Original patches*	*Auxiliary patches*	*Max. per a pillar*
Bunny	4067	45.8 %	18
StillLifeBunny	4291	16.1 %	15
Primitives2	5738	15.1 %	6
Primitives	7999	6.6 %	10
Chess	3873	0.1 %	3

Besides the scene "Bunny" we included a special scene that contains a cube perpendicular on the plane κ. This scene is rather artificial but it represents the most inappropriate object, i.e., the worst case. Furthermore, we distributed some smaller objects around the cube to test visibility of auxiliary patches. In this text, we denote this scene as the scene "Cubes".

In order to show the perpendicular parts without being disturbed by copies, we scaled the scenes such that their orthogonal projections onto the plane κ fit to a rectangle defined by a grid of $1,024 \times 1,024$ samples. We denote such scaled scenes with a symbol "†" in a superscript. We calculated na optical field of $4,096 \times 4,096$ samples using the sampling step of 0.5 μm and reconstructed them numerically using a lens.[31] Reconstructing the scene "Cubes†", we shifted the lens to $(-0.3 \text{ mm}, 0.3 \text{ mm})$ and we focus at the second object, i.e., at 2.0 mm. This allowed one of objects to occlude some of auxiliary patches and thus we could test the visibility. Reconstructing the scene "Bunny†", we shifted the lens to $(0, 0.3 \text{ mm})$ and we focus at 1.8 mm. This allowed us to see the base of the bunny. Reconstructions are presented in Fig. 4.54.

The reconstruction in Fig. 4.54(b) shows that sidewalls are present unlike the case when auxiliary patches are disabled as depicted in Fig. 4.54(a). Also, the object, which is a tiny cube, in focus is not disturbed by auxiliary patches, i.e., visibility works. Similarly, the base

[31] In order to accentuate the perspective and reduced the circle of confusion, we shorten the distance from the lens to the projection plane to 1.0 mm and we use a smaller aperture with a radius of 0.25 mm respectively. These settings were applied only to scenes "Bunny†" and "Cubes†".

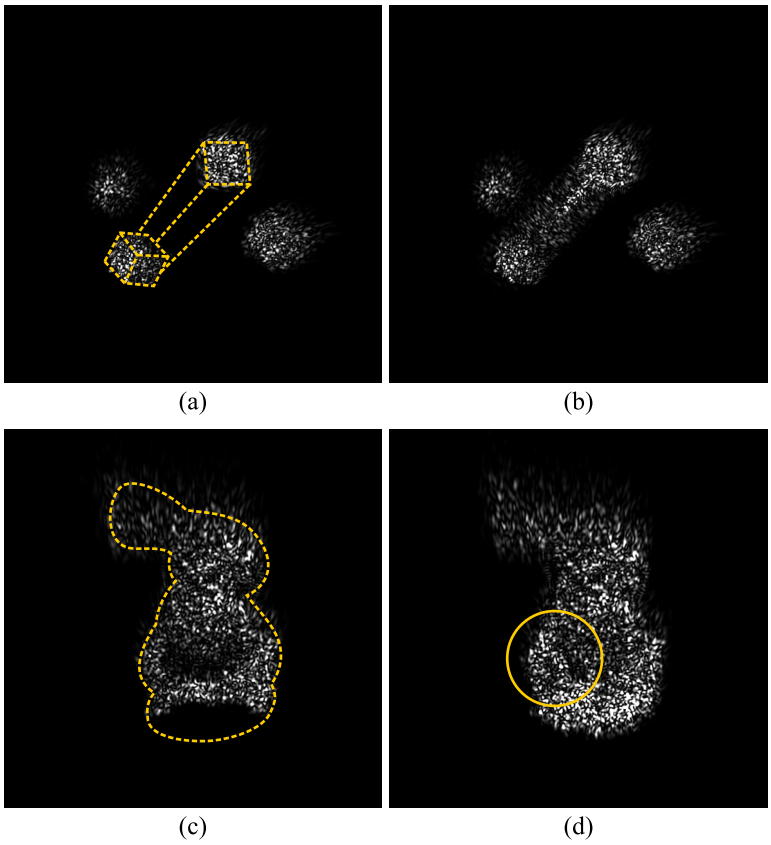


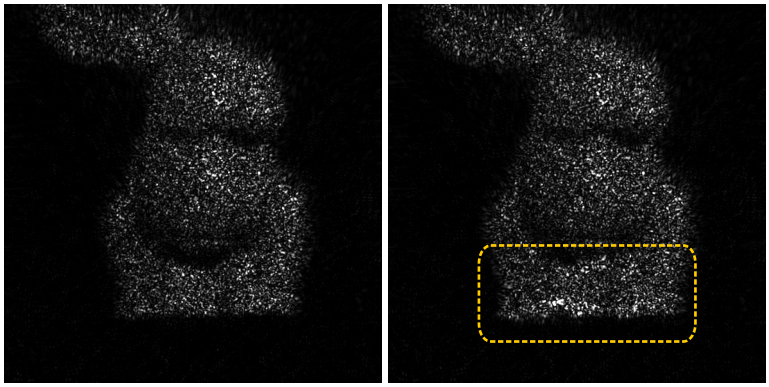
Figure 4.54: Numerical reconstruction of optical fields calculated from scenes (a,b) "Cubes†" and (c,d) "Bunny†". Calculating, auxiliary patches were either (a,c) disabled or (b,d) enabled. The dashed line shows outlines of objects, the solid line shows an example of overlapping. The image shows (a,b) $, 1024 \times 1,024$ samples and (b,d) 768×768 samples for a reconstruction.

of the bunny is present in Fig. 4.54(d). Unfortunately, the bunny seems to contain intensity artifacts. We suspected that the overlapping might be the cause and verified it by calculating an optical field of the scene "Bunny" and reconstructing it. A reconstruction depicted in Fig. 4.55(b) shows intensity artifact located close to the base where the most of auxiliary patches are cumulated.

Verified by reconstructions, the chosen solution works. It does not require any precomputed optical fields and it generates additional structures adaptively according to the depth. Since the method handles auxiliary patches similar to regular patches, it is compatible with almost all acceleration approaches described in Sec. 4.2. The only exception is the grouping the might cause unwanted shift of auxiliary patches.

The only glitch of the approach is overlapping that causes slightly disturbing intensity artifacts. It is a product of visibility approximation and a random phase variation. Nevertheless, since the phase defines directions in which a patch emits light, we can solve the overlapping by modulating the phase variation. This is a problem close to a problem of digital diffusers that is out of scope of this work.[32]

[32]Actually, the only available solution [WB89, Luc94] requires an iterative approach that is rather slow. Since the patch is a plane parallel to the plane κ, we can easily incorporate it but we did not because the solution is a plain application of the brute force.



(a) (b)

Figure 4.55: Numerical reconstruction of optical fields calculated from scenes the scene "Bunny" either (a) without auxiliary patches or (b) with auxiliary patches. The dashed rectangle shows overlapping and images represents a detail of $1,024 \times 1,024$.

We showed that we can enhance our method such that it is able to handle large planes perpendicular to the plane κ. Nevertheless, we also showed, that most of the scenes that we use does not need the handle such planes. Our solution works but it has a minor glitch that can be removed by more sophistical definition of surface emittance properties.

4.4 Discussion and Summary

In this chapter we proposed a new method that calculates an optical filed from a virtual scene. We accelerated the method using features of the optical field. We also addressed the only weakness of the method and we showed that we can reduce its negative influence significantly. In this section, we summarise features, advantages and disadvantages. Also, we discuss other possible trends towards further acceleration.

Before we continue, let us discuss the difference between our method and the method proposed by Martin Janda [JHO08]. Following features of the method, we denote this method as the AngRay method. We decided to include a few paragraphs on this topic due to objections that we received for a reviewer. Let us first summarise similarities. Similar to majority of other methods, we employ the Rayleigh-Sommerfeld diffraction formulation [Goo05] and we use ray-casting both to solve the visibility and to examine a surface in a scene. Also, we can process triangular meshes and we calculate an optical field of surfaces in a virtual scene. Yet, these are the only similarities between the two methods.

The difference between our method and the AngRay method is that the AngRay method handles the surface differently. There is no pre-processing step. The AngRay method samples the surface from the optical field on the fly. Since rays are uniformly distributed in a given range of angles, the AngRay method converts the surface to tiny patches of irregular and uneven shape. It runs strictly in the spatial domain and it is able to handle detail comparable with the size of the sampling step. For that reason, it is quite slow.

Our method can easily process objects of any type that allows calculation of intersection between the object and a ray. It decomposes the scene to patches that acts similar to fragments created by a graphic card. A patch is a part of a plane that is parallel with the plane $\kappa : z = 0$ at which we evaluate the optical field. Since every patch has both the same

shape and the same size, we can accelerate the process of optical field calculation easily and we do not need any resampling due to rotation of a patch. Even without proposed acceleration, the fact that we decompose the scene to patches larger than PLS, leads to a speedup.

Since size of a patch defines the smallest detail we capture and it controls calculation time at the same time, we can generate easily fast previews. However, if we use a patch of a size that is equal to a sampling step, we reduce the method to a geometry-based one. In fact, the resulting calculation is even slower than a regular geometry-based method because we use propagation of the angular spectrum. Therefore, for final production we have to select an appropriate patch size. In this work we consider a patch size that is similar to a pixel size of contemporary LCD because LCD has a pixel that is large but it does not disturb the viewer. The other option, which we did not explore, is to follow abilities of human visual system that were applied by the MIT Holovideo solution [Luc94].

Our method decomposed the scene to pillar and it puts a patch in a front cap of every pillar. As a consequence, we block light by a sidewall of a pillar but we do not emit any light from it. This might cause a viewer to see dark places. Therefore, we proposed an approach that adds patches to emulate light emitters at sidewalls. While this solve the missing emitters, it introduces overlapping that leads to small intensity artifacts. The overlapping is caused by approximation of a half-shadow through a threshold. We assume that it can be solved by modulating phase variation of the patch. Since this is rather a problem of diffusers, it is out of scope of this work and we do not address it here.

Despite the proposed additional patches, out method is not suitable for processing of large and standalone unclosed meshes. Every such a mesh prevents our method from creating appropriately long pillars at sides of an object. Hence, large planes that are almost perpendicular will be decomposed to a cloud a pillar rather then a volume of pillars. However, this is not a serious limitation because a user case prepare or fix a proper mesh using already existing approaches of the computer graphics.

Our method does not address partial transparency because we assume the surface in scenes is opaque. If the transparent object does not deform the objects visible trough it, we may solve it by modifying evaluation of visibility without any additional computation time.[33] If the object had deformed passing waves, we would have had to properly modulate light emitter by every patch of the transparent object. Actually, we would have to calculate an optical field generate by occluded objects at that patch[ZCG08]. This would increase significantly the computation time.

The basic version of our method requires memory because it uses the 2D FFT. The 2D FFT is highly inefficient if the input array is stored in a external memory because it accesses the whole array. Nevertheless, if we apply the frequency masking approach, which was introduced in Sec. 4.2.2, we may decompose the optical field to tiles and solve each tile separately. The frequency masking approach requires one FFT at the end of computation and one FFT every time it generates a patch with a new phase variation. We can combine it with the spectrum library, which was introduced in Sec. 4.2.1. As we have shown, the number of required phase variations is much lower than number of patches and hence we can afford to execute the 2D FFT using the external memory.

Furthermore, our method suffers from unpleasant effects of periodicity enforced by FFT. Due to the periodicity we cannot capture patches that are outside of a subspace defined by a positive Z-axis and a rectangle that encloses samples on the plane $\kappa : z = 0$. If we had tried

[33] In such a case, a result of the visibility test would be a number in a range $[0, 1]$ rather than a boolean value.

it, we would have created overlapping patches because every patch repeats periodically and the length of a period is exactly a size of the optical field. Nevertheless, this limitation is not significant since a hologram acts as a window to a reality behind it.[34]

The more serious issue is the fact that a patch repeats periodically. As a consequence, we obtain a periodically repeating copies of the virtual scene. This disturbs a viewer because these copies will interfere with the original when perspective is applied. Since it is a side-effect of a discrete and finite Fourier transform, we cannot avoid it completely. We can pad a patch with thick frame of zeros such that the distance of copies increases then a viewer does not see them.[35] Consequently, this increases memory requirements. Thus, this is the only true limitation which we cannot avoid or reduce efficiently.

Even though we already discuss various acceleration approaches, we did not deal with brute-force acceleration through hardware means because we focused mostly on optimisations of the algorithm. Thus, let us discuss them briefly now. From a viewpoint of our method, a usage of multiple threads on a single machine is not efficient enough due to memory requirements. We may still use it but the resulting approach has a lot of sequential parts.

Using a distributed environment, the situation is different. The algorithm Alg. 2 processes every patch independently of each other. As a consequence, we can distribute patches among computational nodes, calculate them at each node, retrieve results and sum these results together. As shown in Sec. 4.2, the most time consuming part of computation is propagation of the angular spectrum. Thus, the time spend on calculation will greatly depend on number of patches. Hence, if we had had a homogenous network, we would have been able to distribute the patches statically among nodes and thus to eliminate synchronization almost completely as we did in [JHO08]. Nevertheless, we did not experiment with it due to time restrictions.

Also, we considered application of means that are very low level, i.e., streaming SIMD extensions (SSE) and a graphical processing unit (GPU). Thanks to the adjustment introduced in Sec. 4.2.4, we can use SSE to accelerate the propagation of the angular spectrum. As shown in this chapter, our method uses ray-casting to evaluate visibility and the 2D FFT to calculate the propagation. The rest of operations are piece-wise multiplications and summations. Since the 2D FFT can be implemented on GPU with a speedup, our method can be implemented on GPU as well. The only issue is a limited memory of GPU that is much smaller than a memory accessible by the CPU. This limits the maximum size of the maximum size of the optical field.

In this section we presented features of the proposed method. We showed that our method has a few real disadvantages that we can significantly reduce and only one of them is disturbing and cannot be avoided completely. Through discussion, we suggested that we might reduce the calculation time even further by applying brute-force acceleration such as distributed computing. We, however, did not experimented with it. Overall, we succeeded on designing a method that can calculate optical fields of a virtual scenes at significantly reduced time.

[34] Actually, in order to see na object that is reconstructed in front of a hologram, one have to look at a hologram. The reconstructed object cannot cross the boundary of a hologram.

[35] If the sampling step is larger then $\frac{1}{2}\lambda$, which allows to capture almost all frequencies according to Shannon sampling theorem, we may set the size of the frame according to a range of samples that might be influenced by the patch. Due to the maximum deflection angle, which is specified by the diffraction condition Eq. (2.24). As a consequence, the additional copies should not be able to receive any contribution.

Chapter 5

Hardware Acceleration of a Ray-based Method

In the previous chapter we proposed a new method and presented its acceleration. In this chapter we shall present acceleration of a method designed by Martin Janda [JHO08]. We begin with briefly describing a principle of the method. Then, we describe our contributions that are minor contribution of this thesis. We contributed to his work by proposing acceleration through graphical processing unit. Also, we adjusted a reduced occlusion version of the method to fit the programmable hardware. For the purpose of the description, we denote the method designed by Martin Janda as the AngRay method.

Before we proceed to the description of our contribution, let us briefly introduce the AngRay method that was developed by Martin Janda. We introduce only necessary facts, for more details refer to [JHO08]. We begin the description of the major algorithm and we continue with the proposed accelerations.

The AngRay method calculates an optical field samples from an virtual scene that consists of triangular meshes. In its core, the AngRay method is a PLS-based method that allows more efficient acceleration and that uses ray-casting [Wat00]. The major difference from a common geometry-based method that processes a cloud of PLS is that the AngRay method shoots rays from an optical field sample towards the scene. When a ray hits the surface, it generates PLS and calculates a contribution of PLS to the sample. A direction of a ray is defined by two discrete angle ψ_s and ξ_t as illustrated with Fig. 5.1. The AngRay method shoots rays with a uniform step in both angles and sums contributions.[1] This forms an algorithm that is summarised in Alg. 5.

Let us now introduce briefly calculation of a contribution done by the AngRay method.[2] The ray, which is shot towards the scene, is defined by an origin at $\hat{\mathbf{u}}_{mn}$ and two discrete angles ψ_s and ξ_t, i.e.,

$$R_{mnst} = \{\mathbf{x} : \mathbf{x} = \mathbf{u}_{mn} + r_{mnst}\hat{\mathbf{r}}_{st}\}, \tag{5.1}$$

where $\hat{\mathbf{r}}_{st}$ is a unit directional vector depicted in Fig. 5.1. Hence, at an intersection r_{mnst} is the distance between the sample at $\mathbf{u}_{mn}$ and PLS used in the spherical wave expression Eq. (2.9). The amplitude of PLS is calculated directly from the intersection using standard means of the computer graphics. The phase of PLS can be arbitrary. In this work we used

[1] The range of angles is not vital for description of our contribution. For more details, refer to [JHO08].

[2] Since we keep the notation equal to the Detail driven method, we differ slightly from the paper [JHO08].

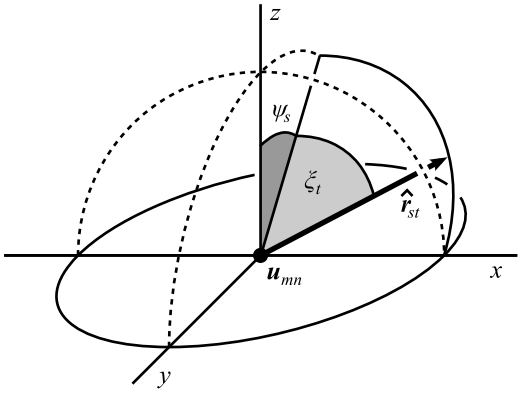


Figure 5.1: The directional vector $\hat{\mathbf{r}}_{st}$ of the ray R_{mnst}. [JHO08]

Algorithm 5 The core algorithm of the AngRay method. [JHO08]

```
1: Zero all samples u_mn.
2: for all samples u_mn of the optical field do
3:     for all angles ψ_s do
4:         for all angles ξ_t do
5:             Create the ray using angles ψ_s and ξ_t.
6:             Find the nearest intersection between the ray and the mesh.
7:             Create PLS at the intersection and shift it properly.
8:             Calculate the contribution of the create PLS to the sample u_mn.
9:             Add the contribution to the sample u_mn.
10:        end for
11:    end for
12: end for
```

two setups of phase: a constant and a random function based on a location of PLS.[3] If the ray does not hit the surface, its contribution will be zero.

The method samples the surface with rays. In a general case, a texture on the surface contains arbitrary frequencies that causes light to be emitted in arbitrary direction. However, it is not possible to capture all frequencies due to a finite sampling step. The texture has to be modified to contain only supported frequencies. Such a texture is easily manageable if the surface consist of planes parallel to the plane κ. In order to create such planes from the original surface, the method shifts each generated along the ray such that the new Z-axis coordinate of PLS becomes an integer multiply of the wavelength.

5.1 Acceleration through GPU

In this section we present our contribution to the AngRay method. Calculating an optical field of $M \times N$ samples, the method generates MN rays. This is a high number but there is a significant coherence between rays. We facilitate this coherence to use efficiently the graphical processing unit (GPU). First, we present design decisions, then we discuss accuracy issue that appeared and we close the discussion with results and measurements.

[3]Since the constant phase is not suitable for viewing, it is used only for testing purposes.

5.1.1 The Design

The GPU is designed to transform triangular meshes and sample them through a unit called a rasterizer. The output of the rasterizer is a uniform, rectangular grid of a samples. In terms of ray-casting, the rasterizer samples the scene by casting parallel rays. Following the expression Eq. (5.1), these are ray R_{mn00} with a directional vector $\hat{\mathbf{r}}_{00} = (0, 0, 1)$. By applying the depth buffer technique, which is hard-wired [Wat00], the rasterizer selects the first intersection for every ray.

Our goal is to exploit the intersection calculation done by GPU. In our first attempt, we tried to follow gathering of contributions done by steps 3–11 of the algorithm Alg. 5. This, however, revealed two issues that threatens the performance: an additional summation step and visibility errors. The latter is caused by the fact that GPU assumes parallel rays. As a consequence, the setup used by the algorithm Alg. 5, which is depicted in Fig. 5.2(a), has to be deformed to a setup depicted in Fig. 5.2(b). If the scene contains large triangles, this deformation will cause visibility errors because GPU transform only vertices of a triangle.[4] In order to avoid it, we have to split large triangles and this might be an expensive operation even for the latest GPU.[5] Besides that, after we evaluate all contributions, we have to sum them. Since we access all samples of a large memory block, we shall experience reduced performance [NVI08]. Therefore we searched for another, more simple solution.

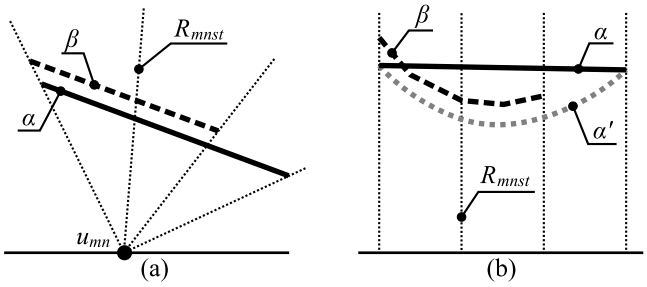


Figure 5.2: (a) A cross-section of two planes sampled by rays originating from the sample u_{mn} and (b) a cross-section of the same setup deformed to fit the rasterizer. While the plane α consists of a single triangle, the plane β consists of multiple triangles. As a consequence, a visibility error occurs in (b). The surface α' represents the plane α deformed accurately.

We notice a fact that the loops in the algorithm Alg. 5 have no sequential dependance and thus we can freely reorganise them. Hence, if we gather a single contribution for all samples u_{mn}, we shall match the scheme that is used by the GPU without any additional summation. Since the AngRay method shoots rays according to the angles ψ_s and ξ_t, all rays are parallel for a given pair of angles. Hence, we do not need any deformation similar to the one depicted in Fig. 5.2. For that reason, we choose to explore this solution. Still, in order to employ the rasterizer, we have to find a transformation $\mathcal{P}_{st}$ that maps the directional vector $\hat{\mathbf{r}}_{st}$ to a vector $\hat{\mathbf{r}}_{00}$.

The first step towards the transformation $\mathcal{P}_{st}$ is to find a transformation that maps a directional vector $\hat{\mathbf{r}}_{st}$ to a vector that is parallel with the vector $\hat{\mathbf{r}}_{00}$ regardless of its length.

[4]The rasterizer considers the triangle as a part of a plane and thus it will not deform the surface. Such a result might suffer from an improperly solved visibility as illustrated with Fig. 5.2(b).

[5]In fact, we can split triangles in that geometry shader that is on the chip. However, the geometry shader can generated only a limited number of additional triangles and it is one of the slowest units of GPU. Thus, we shall end up with an increased calculation time. Among others, at the time of development the geometry shader was not available and thus we searched for another solution that proved to be better.

We identified two such transformations: rotation and skewing. If the rotation is applied to all rays a centre of rotation $\mathbf{x}_{m_r n_r}$ has to be chosen. As a consequence, the distances r_{mnst} are not preserved except the distance $r_{m_r n_r st}$ as depicted in Fig. Fig. 5.3(b). Furthermore, the rotation displaces origins of the rays. Due to these features, the rotation is unacceptable.

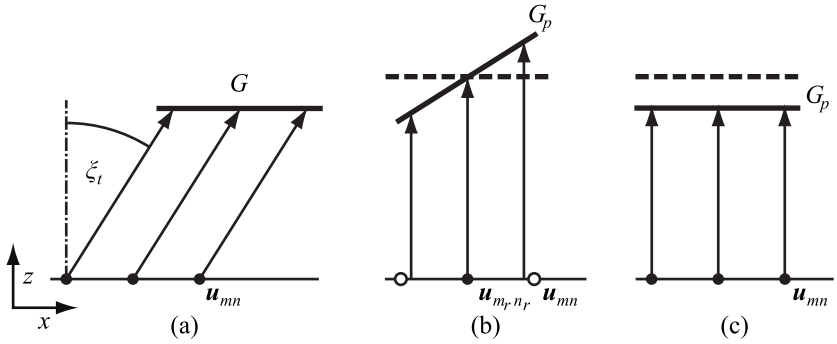


Figure 5.3: Transformations that are able to transform a direction $\hat{\mathbf{r}}_{st}$ to a vector parallel with a vector $\hat{\mathbf{r}}_{00}$. (a) The Mesh G consist of a single line that can be either (b) rotated or (c) skewed. While rotation leads scaling in all axes, skewing scales only along the Z-axis. The dashed line shows the desired output.

The remaining option is skewing that shifts the X-axis and the Y-axis coordinate by an offset. The offset is a function of the Z-axis coordinate. Applied to all rays, the skewing scales distances r_{mnst} by a constant while it keeps origins of the rays intact. As a consequence, the directional vector $\hat{\mathbf{r}}_{st}$ becomes parallel to the vector $\hat{\mathbf{r}}_{00}$ and Z-axis coordinates of intersections equal to corresponding distances r_{mnst} from Eq. (5.1) scaled by a constant as depicted in Fig. 5.3(c). Such behaviour suits our needs.

In the second step towards a proper transformation $\mathcal{P}_{lm}$ we compensate the scaling constant by introducing a multiplicative correction constant ζ_{st}. The correction constant is inversely proportional to a projection of the direction vector $\hat{\mathbf{r}}_{st}$ into the Z-axis, i.e., $\mathbf{z} \cdot \hat{\mathbf{r}}_{st} = 1/\zeta_{st}$. As illustrated with Fig. 5.4, the correction constant is

$$\zeta_{st} = \left(1 + \tan^2 \psi_s + \tan^2 \xi_t\right)^{\frac{1}{2}} . \tag{5.2}$$

Let us assume a left-haded coordinate system [Wat00].[6] Applying Eq. (5.2), the transformation $\mathcal{P}_{st}$ is a transformation matrix

$$\mathcal{P}_{st} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -\tan \xi_t & -\tan \psi_s & \zeta_{st} \end{bmatrix} . \tag{5.3}$$

The whole GPU generation is summarised in Alg. 6. Nevertheless, implementing the algorithm we run into an accuracy issue that prevented us from directly implementing the algorithm Alg. 6. We proposed a solutions that we discuss in the next section.

5.1.2 The Accuracy Issue

During implementation of the algorithm Alg. 6, we ran into accuracy issues. Due to performance reasons, we solved the issue by adjusting the computation rather than using larger

[6] Direct3D, which we used for actual implementation, uses the left-handed coordinate system. Nevertheless, the transformation $\mathcal{P}_{st}$ can be easily modified to fit the right-handed one.

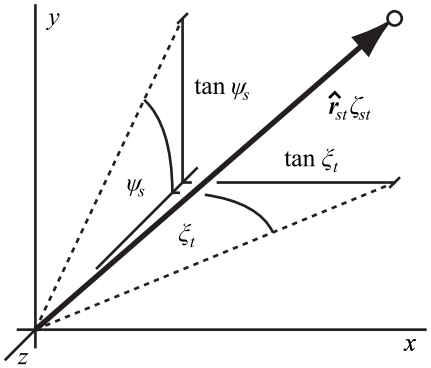


Figure 5.4: Calculation of the correction constant ζ_{st}. An orthogonal projection of the directional vector $\hat{\mathbf{r}}_{st}$ into the Z-axis multiplied by ζst equals to the vector $(0, 0, 1)$.

Algorithm 6 The algorithm of the AngRay method modified to fit GPU. Notice that lines 6–10 are executed entirely on GPU employing the parallel environment of GPU.

```
1: Zero all samples u_mn.
2: for all angles ψ_s do
3:     for all angles ξ_t do
4:         Calculate the constant ζ_st.                          ▷ Eq. (5.2)
5:         Calculate and set the transformation P_st              ▷ Eq. (5.3)
6:         for all samples u_mn do                               ▷ Begin GPU processing
7:             Find the nearest intersection for every ray R_mnst.
8:             Calculate the contribution.                       ▷ Eq. (2.9)
9:             Add the contribution to corresponding sample.
10:        end for                                               ▷ End GPU processing
11:    end for
12: end for
```

data type. GPU supports both single-precision floating point numbers (floats) and double-precision floating point numbers (doubles). However, the number of mathematical units of GPU in the case of floats is much higher than in the case of doubles [NVI08], i.e., execution using doubles will be slower. In this section we propose a modification that allow us to use floats.

Following the expression Eq. (2.9), the contribution of PLS consist of a real-valued amplitude and a phase. Since the resulting contribution are summed together, the phase is a complex number $\chi_{mnst} = \cos(2\pi\,\phi_{mnst}) + \mathrm{j}\sin(2\pi\,\phi_{mnst})$, where $\phi_{mnst} = \frac{1}{\lambda} r_{mnst} + \phi_{\mathrm{PLS}}$, ϕ_{PLS} is proportional to the initial phase of PLS, and r_{mnst} is a distance between the sample u_{mn} and PLS defined in Eq. (5.1).[7] Both the sine and the cosine are periodical functions and thus only a fractional part of the phase ϕ_{mnst} is relevant. As $\lambda \approx 10^{-7}$ m and the maximum distance $r_{mnst} \approx 10^{-1}$ m for usual scenes, $\phi_{st} \approx 10^{6} \sim 2^{20}$, i.e., the fractional part occupies only 4 bits of the 24-bit mantissa if floats are used[IEE85]. This proved to be not enough the phase shift properly because it introduced disturbing artifacts into the reconstruction.

To address the problem of accuracy let us define a plane $\kappa_i : z = iD_z$, $D_z = \text{const.}$ and a distance r_κ that equals the longest straight section of a ray R_{mnst} between two successive planes κ_i and κ_{i+1} as depicted in Fig. 5.5(a). In a general case, $r_\kappa \to \infty$ but thanks to a limited sampling step, a diffraction condition Eq. (2.24) and a limited bounding box of the

[7] Since we sum contributions later, we use the Euler formula rather than the phasor form.

scene, r_κ is finite. We chose D_z such that $\frac{1}{\lambda} r_\kappa$ can be represented in 24-bit mantissa with adequate accuracy of the fractional part. Following similar consideration as in Sec. 4.2.4, we experimented with a 10-bit fractional part.

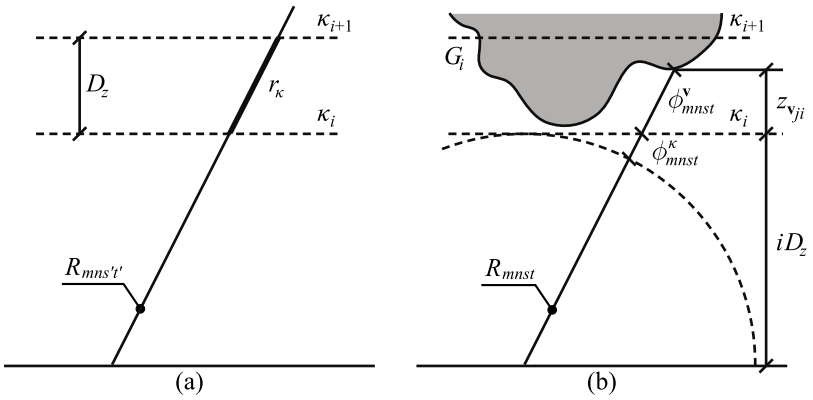


Figure 5.5: (a) Evaluation of the longest distance r_κ and (b) decomposition of the distance including corresponding phase shifts $\phi^{\mathbf{v}}_{mnst}$ and ϕ^{κ}_{mnst}. The longest distance is computed from D_z and the extreme ray $R_{mns't'}$.

Consequently, we define the triangular mesh G_i that is the triangular mesh G in a subspace between planes κ_i and κ_{i+1}.[8] Furthermore, we split the Z-axis coordinate of vertices of the mesh G_i. Let $\mathbf{v}_j$ be a location of a vertex in the mesh G and let $\mathbf{v}_{ji}$ be a location of the corresponding vertex in the mesh G_i. The new location $\mathbf{v}_{ji} = \mathbf{v}_j - (0, 0, iD_z)$ assures that its Z-axis coordinate $z_{\mathbf{v}_{ji}} \leq D_z$.

Following the definition of the Z-axis coordinate, we split the phase ϕ_{mnst} of a contribution to $\phi_{mnst} = \phi^{\mathbf{v}}_{mnst} + \phi^{\kappa}_{mnst}$, where

$$\begin{aligned} \phi^{\mathbf{v}}_{mnst} &= \zeta_{st} \frac{z_{\mathbf{v}_{ji}}}{\lambda}, \\ \phi^{\kappa}_{mnst} &= \zeta_{st} \frac{iD_z}{\lambda} \end{aligned} \tag{5.4}$$

as illustrated with Fig. 5.5(b). The fractional part of $\phi^{\mathbf{v}}_{mnst}$ can be computed on GPU with adequate accuracy because of an appropriately selected distance D_z. Since iD_z is constant for all vertices in the mesh G_i, the fractional part of ϕ^{κ}_{mnst} is computed on CPU using doubles. However, unlike $\phi^{\mathbf{v}}_{mnst}$ it is computed only once per each direction $\hat{\mathbf{r}}_{st}$. The resulting phase is a sum of fractional parts and this is accurate enough.

Another accuracy issue appear when $\phi_s \approx 0$ and $\xi_t \approx 0$. Applied to Eq. (5.2), the result is only slightly larger than one and when stored in floats before it is sent to GPU, it becomes one. We resolved this issue by sending $\zeta_{lm} - 1$ instead of ζ_{lm} and processing it in that decomposed form on GPU.

In this subsection we proposed solution of the accuracy issues. In the next section we present measurements and we compare the results with the results of the original algorithm Alg. 5.

[8] The mesh G_i can be computed using standard means of the computer graphics [JHO08].

5.1.3 The Results

In the previous sections we presented acceleration of the AngRay method using GPU. In this section we show that the proposed solution works and provides significant acceleration. For that purpose we use a large sampling step of 7.0 μm. As a consequence, both angular steps D_ψ and D_ξ will be small and $\psi_s \approx 0$ and $\xi_t \approx 0$ most of the time. Also, we select such scenes that contain objects at least at 0.4 m. This will test the proposed accuracy solution.

We calculate optical fields of scenes "Chess", "Plane", and "Primitives". In the case of the scene "Plane" every ray hits the surface and thus we can test real efficiency of GPU. The scene "Primitives" contains a small number of objects at different depths and a small number of triangles while the scene "Chess" contains a larger number of objects with a complicated visibility and a larger number of triangles. Through these scene we shall test influence of triangle number and visibility solution.

First, we tested the accuracy of the calculation described in Sec. 5.1.2. Since we know the accurate result, which is calculated by CPU, we can compare the optical fields numerically. For that purpose we calculated an optical field of $1,024 \times 1,024$ samples. We used the scene "Plane" because it consists of a single plane at 0.42 mm, i.e., there will be no blur due to focus in the reconstruction. Hence, we can measure how accurately we captured the surface.

Since intensity is the only measurable feature of the optical field, we compare intensities of optical fields propagated without lens. After the propagation, we normalised both fields such that the maximum intensity was 1.0 and we calculated the mean square error (MSE) of $M \times N$ samples as $\mathrm{MSE} = \frac{1}{MN} \sum_m \sum_n (|u_{mn}|^2 - |u'_{mn}|^2)^2$. The resulting MSE was 0.97×10^{-4}. Such MSE is neglectable and thus the proposed version that uses GPU is able to calculate properly an optical field.

Next, we measured a speedup. For that purpose we calculated optical fields of all scenes using both CPU (Intel Xeon 3.2 GHz) and GPU (NVIDIA GeForce 8800 GTX). Due time reasons, we used smaller optical fields of $1,024 \times 1,024$ samples.[9] As shown in the table Tab. 5.1, the speedup is significant. The more rays hit the void, the weaker is the speedup. The number of triangles has almost no influence on calculation using GPU because the scene "Chess" was calculated 1.2× slower than the scene "Primitives" even though the scene "Chess" contains 44× more triangles. On the other hand, the GPU employs a brute force approach while solving the visibility, i.e., it drops previously calculated contribution when it finds a closer one. This reduces the speedup as illustrated with the scene "Chess", which has a complicated visibility, compared to the scene "Plane".

Table 5.1: Calculation times and speedups using the GPU in comparison with the CPU. The table also shows estimated percentage of created rays that hits the scene.

Scene	*CPU*	*GPU*	*Speedup*	*Hit rays*
Chess	78.6 hr	0.3 hr	245.7	32 %
Primitives	65.4 hr	0.2 hr	327.1	30 %
Plane	218.9 hr	0.3 hr	718.9	100 %

Illustrated with measurements, we achieve the goal of accelerating the AngRay method. Using just a single GPU, we outperformed the CPU significantly. Despite our attempts on

[9] Even through the field contained rather low number of samples, it took about 218 hours to calculate using CPU and the original algorithm.

accuracy, the calculated optical field differs from the result of CPU but we showed that the difference is neglectable.

5.2 The Partial Quadratic Approximation

Besides the original AngRay method, we also collaborated on the reduced occlusion method that is a modification of the AngRay method. We designed an approximation that speeds up a the reduced occlusion method developed by Martin Janda and that is suitable for implementation on a programmable hardware. We begin with a brief overview of the modification and we continue with a description of our contribution.

First, let us briefly described the reduced occlusion method proposed by Martin Janda [JHS07, JHO08]. The basic principle is the same as the AngRay method. The major difference in handling of PLS and in range of angles. The reduced occlusion method ignores the Y-axis, i.e., the discrete angle ψ_s depicted in Fig. 5.1 is zero all the time. This removes one loop from the algorithm Alg. 5 and saves computation time. However, at the same time this creates a horizontal parallax only (HPO) hologram. If reconstructed without a setup used in this work, it will blurred vertically. In order to prevent the blur, each generated PLS contributes to the whole column of samples rather than to a single sample. For the purpose of this text, we denote this method as the ReOc method.

We collaborated with Martin Janda on improving the efficiency of the ReOc method. Thanks to the HPO-like structure, the ReOc method reduces number of lighting calculations. However, it does not reduce the number of calculated contribution because every PLS contributes to the whole column. We focus on that fact. The contribution of PLS to the column follows the spherical wave expression Eq. (2.9). The calculation can be simplified using various approximation but they have either restrictive spatial limitations [YIO00, IMY$^+$05] or require double-precision floating point [MT00]. Thus, we propose a new approximation that imposes quite loose restrictions and that is able to use 32-bit integers. The latter makes the approximation compatible with the programmable hardware (FPGA) and graphical processing unit (GPU).

5.2.1 The Approximation

In order to decrease the calculation time of the reduced occlusion method, we designed an approximation that calculates an column of samples generated by PLS. The approximation suits the reduce occlusion method and allows a fixed point arithmetic. This is crucial for use with a programmable hardware (FPGA). In this section we describe the principle of the approximation.

Let us now discuss a single PLS s generated by the ReOc method. PLS is located at $\mathbf{s} = (x_\mathbf{s}, y_\mathbf{s}, z_\mathbf{s})$. Without imposing a significant restriction, we can assume that all PLS are located in a subspace defined by the positive Z-axis and the XY-plane, i.e., $z_\mathbf{s} > 0$. It is self-luminous and it emits light with a complex amplitude $I_s^{1/2} \exp(\mathrm{j}2\pi\phi_s)$, where I_s is intensity of PLS and $2\pi\phi_s$ is the phase of PLS. Since PLS generates a spherical wave defined by Eq. (2.9), the contribution c_{mn} of PLS to the sample u_{mn} is

$$c_{mn} = \frac{I_s^{1/2}}{r}\frac{z}{r}\exp(\mathrm{j}2\pi\phi_r + \mathrm{j}2\pi\phi_s), \tag{5.5}$$

where $\phi_r = \frac{1}{\lambda}[(mD_x - x_\mathbf{s})^2 + (nD_y - y_\mathbf{s}) + z_\mathbf{s}^2]^{1/2}$ is proportional to the phase shift due to a distance r between PLS and the sample u_{mn}.

Since $\psi_s = 0$, the ReOc method generates PLS from a horizontal slice of a scene. Each slice is aligned to a row of samples and thus $y_\mathbf{s} = n_s D_y$. A single PLS contributes to a single column $\nu_m : u_{mn'}, n' \in [-M/2, M/2 - 1]$. Then, the phase shift ϕ_r is

$$\phi_r = \frac{1}{\lambda}[(n' - n_s)^2 \Delta_y^2 + \zeta^2]^{1/2}, \tag{5.6}$$

where $\zeta^2 = (x_\mathbf{s} - x_{mn_s})^2 + z_\mathbf{s}^2$ is constant for a given column ν_m and PLS located at $\mathbf{s}$.

A frequent approach to efficient evaluation of Eq. (5.6) employs the binomial series that approximate the square root function [IMY+05, YIO00]. This, however, enforces a minimum distance along the Z-axis in which the approximated ϕ_r is considered valid [Goo05]. The larger the scene is, the further away it has to be. Nevertheless, the function ϕ_r is smooth and in a small range $n' \in [n_0, n_0 + P - 1]$, where $P \in \mathbb{Z}$ is small, it resembles a quadratic function. Thus, we can split the column into subparts and approximate each subpart by a quadratic function. This removes the necessity of the square root function within the range similarly to the binomial series but it does not enforce the minimum distance.

Thus, we split the column ν_m into subparts $\mu_{m_i} = (u_{m_i p})$, where $u_{m_i p}$ is a sample of the subpart μ_{m_i} and $p \in [0, P - 1]$, i.e., a subpart contains P samples. We denote the first sample $u_{m_i 0}$ of a subpart ν_{m_i} as the node η_{m_i} and we evaluate the phase ϕ_r at the node accurately using the expression Eq. (5.6). Since the starting phase ϕ_s is constant for PLS, we approximate the sum $\phi_r + \phi_s$ with a quadratic function

$$\phi_r + \phi_s \approx \phi_{m_i p} = a t_x^2 + b t_x + c,\ t_x = \frac{p}{P}, \tag{5.7}$$

where a, b, and c are the quadratic coefficients, $t_x \in [0, 1)$ and P is a length of a subpart. The expression Eq. (5.7) approximates the phase between two nodes η_{m_i} and $\eta_{m_{i+1}}$. Since the nodes are uniformly distributed along the column, the coefficients a, b, and c are

$$\begin{aligned} a &= \frac{1}{2\lambda}(r_{m_{i+2}} + r_{m_i} - 2r_{m_{i+1}}), \\ b &= \frac{1}{2\lambda}(4r_{m_{i+1}} - 3r_{m_i} - r_{m_{i+2}}), \\ c &= \frac{1}{\lambda} r_{m_i} + \phi_s, \end{aligned} \tag{5.8}$$

where r_{m_i} is an accurate distance between PLS and the node η_{m_i}. Due to the applied principle, we denote this approximation as the partial quadratic approximation.

Following the expression Eq. (5.5), the amplitude of a contribution at the node η_{m_i} is

$$A_{m_i} = \frac{I_s^{1/2}}{r_{m_i}} \frac{z_s}{r_{m_i}}, \tag{5.9}$$

where r_{m_i} is the distance between PLS and the sample at the node η_{m_i} and I_s is PLS intensity. The amplitude function Eq. (5.9) exhibits similar properties as the function ϕ_r and thus it can be approximated similarly. However, we use much coarser approximation because the amplitude does not need to be represented as precisely as the phase [Goo05].[10] Using linear interpolation, we approximate the amplitude at the sample $u_{m_i p}$ as

$$A_{m_i p} = A_{m_i} + (A_{m_{i+1}} - A_{m_i}) t_x,\ t_x = \frac{p}{P}. \tag{5.10}$$

[10] Even if we ignore completely the amplitude, the hologram will provide a recognisable reconstruction [MNF+02, IMY+05].

Based on a contribution expression Eq. (5.5), the phase approximation Eq. (5.7), and the amplitude approximation Eq. (Eq. (5.10)), the contribution c_{m_ip} of PLS to the sample u_{m_ip} is approximated as

$$c_{m_ip} = A_{m_ip} \exp(\mathrm{j}2\pi\phi_{m_ip}) = A_{m_ip} \cos(2\pi\phi_{m_ip}) + \mathrm{j}A_{m_i}p \sin(2\pi\phi_{m_ip}). \tag{5.11}$$

5.2.2 The Length of a subpart

In the previous section, we described the approximation. In this section we estimate a proper length of a subpart. For that purpose we estimate error caused by the approximation and its influence on the reconstruction. Since the optical field is sensitive to the phase [MNF+02, IMY+05, Goo05] rather than the amplitude, we examine only the error of the phase.

First, we verified the shape of the error. For that purpose we evaluated the phase both using the approximation Eq. (5.7) and using the binomial series [YIO00, IMY+05] discussed in Sec. 3.2. For this purpose, we chose arbitrarily a length of a subpart as $P = 256$ samples, we put PLS at a distance of 0.3 m over the beginning of a column and the sampling step was 2.0 μm. Using various lengths of columns, we evaluated the error as a difference from an exact value at the last subpart of a column because we presumed that the error will be most noticeable at that location. The results presented in Fig. 5.6. Since the phase $\phi = \frac{1}{\lambda}r$, where r is a distance, the error equal to 1.0 means an error in phase of one period.

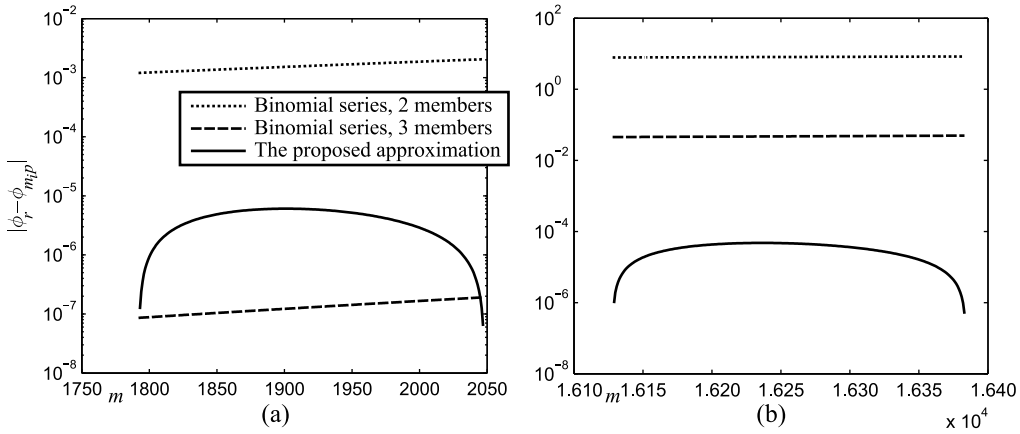


Figure 5.6: The difference of phases calculated using the binomial series and using the proposed approximation for a column of either (a) 2,048 samples and (b) 16,384 samples.

The results in Fig. 5.6 show that the difference is a smooth and concave-like function. Thus, we can use a maximum of the difference between the approximated phase ϕ_{m_ip} and the exact phase ϕ_r as the error of approximation. Also, the results shows that unlike the binomial series the error is almost independent on the size of a column. Hence, we can approximate accurately larger optical fields.

Now, let us discuss the influence of the approximation error on the reconstruction. We can expect that by increasing a length of the subpart we increase the error. Hence, if we had known the acceptable error, we would have been able identify by a computation a length that is appropriate. For that purpose we calculated optical fields generated by a single PLS and reconstructed them without a lens because we did not examine influence on the viewer. In order to create various error, we chose to use various lengths of subparts and various distance

of PLS. In all cases we used a sampling step of 0.5 μm that is the shortest sampling step we considered.[11] We present the reconstructions in Fig. 5.7.

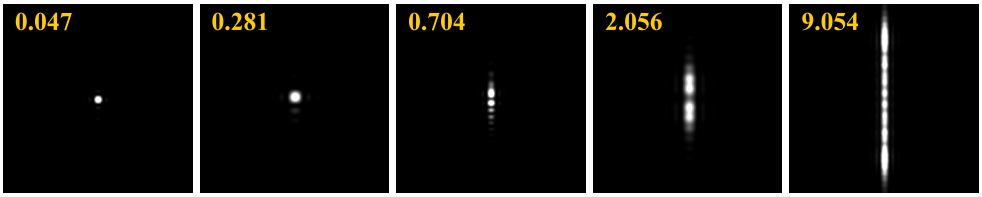


Figure 5.7: Numerical reconstructions of optical fields calculate using the proposed approximation. A number in a corner of each image is the maximum difference of phases in all possible subparts. The number of samples per a subpart was (a) 512, (b,c) 1,024, (d,e) 2,048. The distance of PLS was (a,c) 5.0 mm and (b,d,e) 3.0 mm.

As it is shown in Fig. 5.7, a difference in phases larger than one period has a devastating effect on a reconstruction. Even if difference is below one period, it still has a negative impact. The asymmetry present in Fig. 5.7(c) is caused by asymmetry of evaluation of interpolation coefficient in Eq. (5.8). Following experiments, we say that a difference greater than $\frac{1}{4}$ period is not acceptable. Due to implementation reasons, we restrict all possible subparts length to $P = 2^a$, $a \in \mathbb{Z}$. Since $P \leq M$ we can estimate the maximum appropriate length of a subpart by calculating the error fast enough.

In this section we showed that the proposed approximation has better accuracy than the binomial series. Also, we estimated a boundary error that assures a valid reconstruction and we applied this error to estimate the maximum appropriate length of a subpart. In the next section we discuss ability of the approximation to use fixed point arithmetics.

5.2.3 The Fixed-Point Calculation

In the previous sections we presented an approximation that uses a quadratic function. In this section we show that it compatible with fixed point arithmetic. For that purpose we define parameters of the scene and we estimate necessary bit lengths. This section is crucial for successful implementation using programmable hardware. For clarity of the text, let as assume that we are processing the subpart ν_{m_i}, i.e., we shall inherently assume the index m_i in this section if not noted otherwise.

The programmable hardware is not suitable for floating point operations. Also, fixed point multipliers are usually a limited resource and therefore we have to reformulate Eq. (5.7). Since both the sampling step D_y and the length P are constant, we can use a differential scheme that lacks multipliers. Based on the phase approximation Eq. (5.7) and the amplitude approximation Eq. (5.10), the differential scheme is

$$\begin{aligned} A_{p+1} &= A_p + \Delta A, \\ \phi_{p+1} &= \phi_p + \Delta\phi_{p+1}, \\ \Delta\phi_{p+1} &= \Delta\phi_p + \Delta\Delta\phi, \end{aligned} \tag{5.12}$$

where the phase ϕ_0 and the amplitude A_0 are evaluated accurately at the node η_{m_i}. Using the expression Eq. (5.8) that defines quadratic coefficients, the parameters of the differential

[11] We calculated na optical field of $2,048 \times 2,048$ samples and PLS was located over the centre of the field.

scheme are

$$\begin{aligned} \phi_0 &= c\frac{1}{P},\ \Delta\phi_0 = b\frac{1}{P} + a\frac{1}{P^2},\ \Delta\Delta\phi = 2a\frac{1}{P^2}, \\ \Delta A &= (A_{m_{i+1}0} - A_{m_i 0})\frac{1}{P}. \end{aligned} \tag{5.13}$$

Next, we limit the spatial extent of the virtual scene and we define parameters of a supported optical field. This step will allow us to select appropriate bit lengths of number representations. In all our computations we assume a wavelength of $\lambda = 635$ nm by default. We designed the solution for a sampling step of 0.5 μm and an optical field of $65,536 \times 65,536$ samples. We assume that intensity of all PLS is a range [0.0, 1.0] and Z-axis coordinate of all PLS is in a range of [0.092 m, 0.200 m]. We select the minimal distance according to the diffraction condition Eq. (2.24) applied to an amplitude-modulating hologram.[12] Using these parameters, we estimated a maximum size of a subpart to $P = 16,384$ samples. Let us denote this configuration as the configuration A. Since such a large optical field is not practical for testing purposes, we also used an optical field of $2,048 \times 2,048$ samples. In such a case, we estimated a subpart length of $P = 512$ samples and we shifted the depth range to [0.003 m, 0.200 m]. Let us denote this configuration as the configuration B.

Let us now define necessary bit ranges for parameters of the differential scheme Eq. (5.12). According to the expression Eq. (5.11), the argument of a contribution to a sample u_p is $2\pi\phi_p$. Both the sine and the cosine, which we use to express the complex number, are periodic and therefore we need just a few first bits of the fractional part of the phase ϕ_p. Since $\phi_p >> 1$, we can use a fractional part directly and we do not need to care about overflows in the integer part.

The scheme Eq. (5.12) implements a quadratic function. Hence, the parameter $\Delta\Delta\phi$ defines convexity of the function ϕ_p. Thus, we can estimate the minimum value of $\Delta\Delta\phi$ using the closest PLS and the maximum value using the furthest PLS as illustrated with Fig. 5.8(a). The reference sample is at edge of the optical field in order obtain extreme values. The parameter $\Delta\phi_p \propto \frac{\partial\phi_p}{\partial t_x}$ defines how the function ϕ_p changes when we move to a next sample. Thus, it reaches the minimum using PLS that is further away along the Z-axis and it reaches maximum using PLS whose function ϕ_p is the most convex one as illustrated with Fig. 5.8(b).

The amplitude function A_p, which is defined by Eq. (5.9), contains two components: a cosine-like function $\frac{z_s}{r}$ and a function $\frac{1}{r}$. As illustrated with Fig. 5.8(c,d), both functions reach their maximum and minimum using different PLS. Therefore, we estimated the range by multiplying maximums and minimums. Also, we assumed that intensity I_s of PLS is 1.0 at the maximum and $2^{-6} \approx 0.02$ at the minimum. Any PLS with intensity lower than the minimum is excluded. The results range contains all possible worst-cases because the function A_p never reaches the point when both $\frac{z_s}{r}$ and $\frac{1}{r}$ are at maximum or at minimum.

By applying the approach described above, we can calculate necessary bit ranges for both the configuration A and the configuration B. Uniting them and assuming that an error of the amplitude is not crucial, we obtained the minimal bit budget depicted in Fig. 5.9. The budget shows that we can use 32-bit fixed point numbers to calculate the optical field. This budget supports any other configuration that is between the configuration A and the configuration B. That this assumes that the minimal distance of PLS should be set according to the resolution

[12] An amplitude-modulating hologram is an array of real numbers. Hence, its Fourier transform is symmetric [Smi97]. Thus, exposing the hologram to a plane wave means that we can control only a half of the whole range without a risk of overlapping with the symmetrical copy.

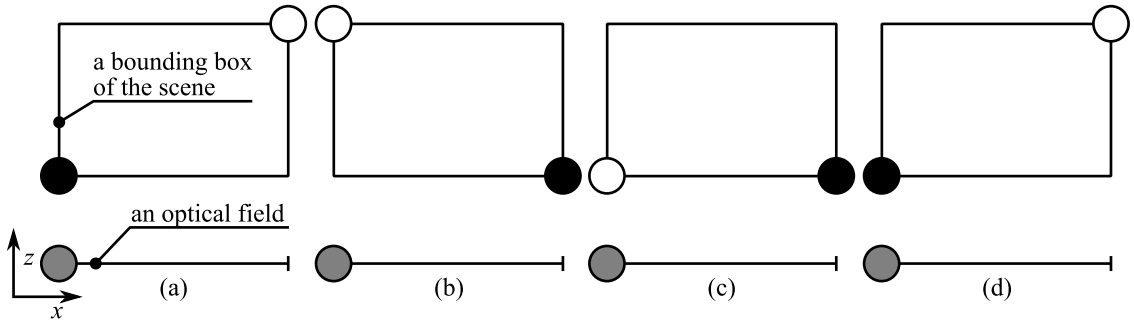


Figure 5.8: Positions of PLS that may lead to either the maximum or the minimum of (a) $\Delta\Delta\phi$, (b) $\Delta\phi_p$, (c) $\frac{z_s}{r}$, and (d) $\frac{1}{r}$, where r is a distance between the sample and PLS. PLS at the white circle leads the minimum and PLS at the black circle leads to the maximum. Both the maximum and the minimum are inspected from a sample marked by a gray circle.

of the optical field. If PLS is close than it should be, we suggest to limit the contribution only to a range defined by the diffraction condition Eq. (2.24).

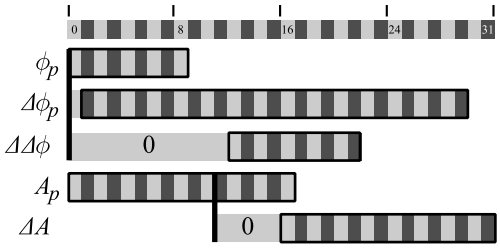


Figure 5.9: Estimated bit ranges necessary to evaluate the differential scheme Eq. (5.12). The thick vertical lines show locations of decimal points.

We use fixed point numbers to evaluate the fractional part of the phase ϕ_p. Hence, we replace the sine and the cosine function with a table. Using the results of experiments done in [RBD+99], we set the bit length of the table index to 8 bits. Since both the since and cosine function give values from a range $[0,\ 1]$, we use the same number of bits to represent table contents. After we retrieve the sine and the cosine, we multiply them with the function A_p and we round the result to 8 bits.

As long as we kept $\frac{1}{r} \gg 1$, we obtain working optical fields. However, when we experimented with a different settings that fits the bit budget but that uses too small distances for a given sampling step, we run into accuracy issues. We calculated an optical field of $4,096 \times 4,096$ samples using a sampling step of 7.0 μm, a subpart of $P = 4,096$ samples and a depth range of $[0.4\text{ m},\ 0.7\text{ m}]$. Using the original rounding to 8 bit we obtained an optical field with significantly disturbed phase as depicted in Fig. 5.10(a). The phase was almost constant and the optical field behaved like an in-line hologram as illustrated with Fig. 5.11(a). When we increased the bits left after the rounding, the situation improved. From results of Sec. 5.1 we knew that single-precision floating point numbers (floats) works. Since floats have 24-bit mantissa and we need at maximum 10 bit for the integer part, we can round to 14 bit fractional part safely. We experimented successfully with it as illustrated with Fig. 5.10(c) and Fig. 5.11(b). Thus, instead of rounding to 8 bits, we round to 14 bits.

As we increate the number of rounding bits, we reduce a number of PLS that we can accumulate less before an overflow. Since the integer part of the function A_p is 10 bits

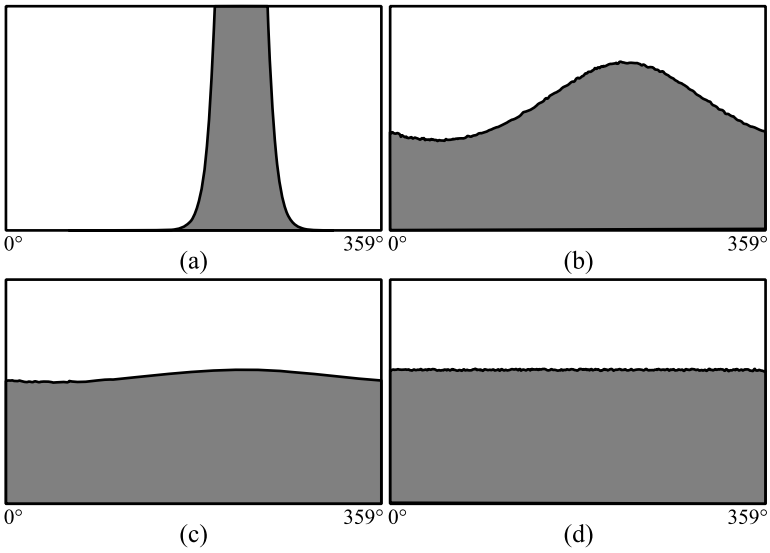


Figure 5.10: Histograms of phases of optical fields calculated using (a) an 8-bit, (b) 12-bit, (c) 14-bit, and (d) a 16-bit fractional part of a result. All histograms are scaled similarly, the top of a histogram is 0.5 % a total number of optical field samples.

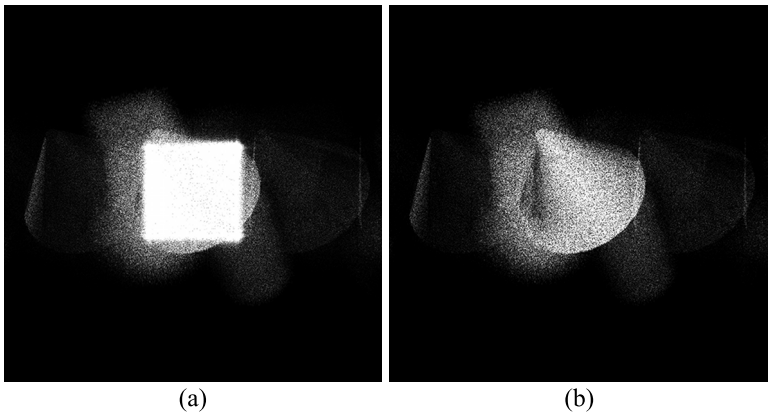


Figure 5.11: Numerical reconstructions of the same field. The optical fields were calculated using (a) an 8-bit and (b) a 14-bit fractional part of a result.

and the fractional part of the result is 14 bit, we can accumulate only 256 light sources. Afterwards, we have convert these sources to a floating point numbers and add them to the resulting optical field. This increases calculation time but as we show in the next section, the increase is not overly dramatic.

5.2.4 The Results

In the previous section we presented a new approximation that is based on a quadratic function. Using the approximation, we can calculate an optical field facilitating a fixed floating arithmetics. In this section we focus on time measurements. We show that increase of the rounding bits does not significantly increase the computation time and we discuss that our approximation is better than other ones.

First, we show that by increasing the rounding bits, we do not increase significantly the computation time. For that purpose we used the scene "Primitives", an optical field of $4,096 \times 4,096$ samples and a sampling step of 7.0 μm. Using PC Intel Xeon 3.2 GHz, we measured the computation time. The measured values that are presented in the table Tab. 5.2 shows that the calculation time is increased but the increase is not proportional.

Table 5.2: A reaction of calculation time to increase of rounding bits. The column denoted at "Relative increase" shows additional time when compared to 8-bit case. All times were measured using PC Intel Xeon 3.2 GHz.

Fractional bits left	Calculation time	Relative increase
8 bit	171.3 hr	0 %
12 bit	175.3 hr	2 %
14 bit	188.0 hr	10 %
16 bit	239.9 hr	40 %

As the next, we measure the actual speedup achieved by the proposed approximation. For that purpose, we used a smaller optical field of $2,048 \times 2,048$ samples, a sampling step of 0.5 μm and the scenes "Primitives" and "Bunny". Using these parameters, we used the subpart length of $P = 512$ samples. Besides our approximation, we implemented a full evaluation of a contribution Eq. (5.5) using double-precision floating point numbers (doubles) and we used measurements of this implementation to show a speedup achieved by our approximation.

Besides that we implemented a Fresnel approximation in a form of scaling of a precalculated optical field as described in Sec. 3.2.1. We used a single floating point numbers (floats) and a bilinear interpolation to retrieve the sample from a precalculated optical field. Also, we implemented a version of our method that uses just a linear interpolation instead of the quadratic one. In that case, we shrank a subpart to $P = 64$ samples. In both the linear interpolation and the quadratic one, we used 8-bit rounding. We measured times using PC Intel Xeon 3.2 GHzand we compared them to the computation time of the exact evaluation defined by Eq. (5.5). The results are presented in Fig. 5.12.

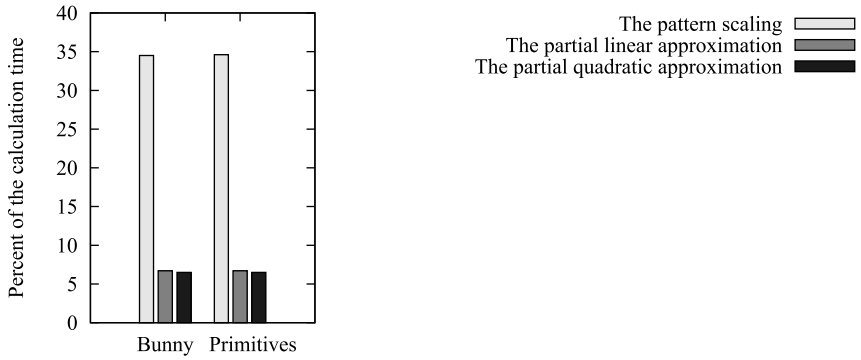


Figure 5.12: A speedup of calculation on CPU using various approximations. The speedup is expressed in percentage of calculation time using exact evaluation of the distance.

The results from Fig. 5.12 show that we achieved speedup. Using the table Tab. 5.2, we can estimate the speedup in a case of 14-bit rounding to 13× (7.6 %) lower computation time. The linear version of our interpolation provides similar speedup. This is caused by a simpler differential scheme. On the other hand, the quadratic version allows longer subparts. This

might lead to more efficient implementation on a hardware because we do not often disturb the computation by providing a new set of parameters.

Also, we did not compared directly our approximation with the recurrence formula [MT00]. According to the paper, the maximum error caused by the recurrence formula is comparable. If we apply additional error control to the recurrence formula, the error will be significantly lower than the partial quadratic approximation. Even though in such a case it is rather questionable whether it is necessary to decrease error beyond the boundary we discussed in Sec. 5.2.2. Unfortunately, the recurrence formula requires doubles for a sampling step shorter than $D_x \sim 10\lambda \approx 6$ μm. As a consequence, the original paper states that the recurrence formula achieve a speedup maximum of only $4\times$ (25.1 %) lower computation time than a full evaluation of a contribution Eq. (5.5). Hence, we are faster.

On the other hand, a solution proposed in [YIO00] uses fixed point numbers.[13] According to the paper, they are able to reach a speedup of $20\times$ (5.0 %). However, they use the binomial series and therefore they end up with a high error for larger optical fields in combination with closer objects as we have shown in Sec. 5.2.2. Thus, they are $1.5\times$ faster but we can handle objects that are much closer.

In this section we have shown that the proposed approximation is not the best in both the speedup and the accuracy. We showed that our approximation is either faster or more accurate than considered and existing approximations. Also, we have shown that the increase the number of rounding bits does not lead to a significant increase of computational time. Hence, our approximation is useful. Since it uses fixed point numbers and a subpart is quite long, we can expect that by implementing on a programmable hardware we gain another speedup. Since this is out of scope of this work, we did not implement it.

[13] Actually, other mentioned solution [IMY+05] is just reimplementation that omits amplitude in order to obtain better performance. For that reason, we did not compare with it.

Chapter 6

Summary

In the previous chapters, we gave a brief and necessary overview of holography. Then, we summarised existing method for hologram generation and we identified their weaknesses. Based on that, we proposed a new method and showed that under appropriate conditions it is faster. Next, we presented our contribution to the method designed by Martin Janda. Finally, in this chapter we give an overview of results achieved in this work. This chapter summarises the whole work.

In this work, we examined whether it is possible to combine different trends on digital hologram generation and gain a speedup at the same time. We verified that such a combination is possible and we showed that under appropriate conditions we can end up with low computation times. We estimated these conditions using theoretical evaluation and we verified them numerically. Based on that, we can state that these conditions are not tight so that they are not serious limitations of the proposed method.

Even though we achieved the goal of lower computation time, we experimented with the method even further. Thanks to the design of the method, we introduced modifications that allowed us to reduce the computation time of the basic version $10\times$. This means that we ended up in the order of minutes where we started in the order of hours. Notice that we did not applied any brute-force acceleration approaches through the hardware so far.

We designed our method following an analogy with the computer graphics. Similar to the computer graphics, we decompose the scene to small and uniform elements. In our case, we use patches that are parts of a plane. Since the plane is parallel to the plane at which we evaluate optical field samples, we are able to calculate efficiently an optical field generated by the patch. We handle patches as the whole, i.e., the patch is the smallest detail that we process. This has two important side effects. Firstly, we can use a ray-casting to approximate visibility of the whole patch. Secondly, we can trade lower computational time for tiny details that cannot by recognised by the viewer anyway. Thus, we can generate low-detail previews quickly.

The major disadvantage of our method is caused by the original design. Our method decomposes the scene to patches. If a surface of an object consists of large parts of planes that are perpendicular to the plane at which we evaluate the optical field samples, we cannot capture the scene properly. We just create a structure that blocks the light at these planes but we do not add any emitter. In this work we address this issue by proposing a solution that adds the emitter. The solution neither modifies the algorithm nor requires additional precalculated data. The only glitch of the solution is overlapping that may manifest itself as

an intensity artifact. This glitch can be removed by properly defining a light emitted by the patch. We, however, did not address it because it is an issue of a digital diffuser which is out of the scope of this work.

Our method requires that the whole optical field fits into the memory because it employs the 2D fast Fourier transform (FFT) to calculate an optical field of a patch. If it is not fulfilled, we may experience serious efficiency loss because the 2D FFT accesses the whole field. We addressed this issue by introducing the frequency masking that limits a number of executions of FFT and hence we can afford to execute the 2D FFT using the external memory.

Another issue of using FFT is the fact that FFT assumes periodicity of the input. As a consequence, we calculate an infinite number of copies of the scene. These copies might disturb the viewer due to perspective deformation. Since the periodicity assumption is essential component of FFT, we cannot avoid the copies. We, however, can reduce them by calculating more optical field samples. This, however, increases memory consumption.

Even though we proposed various modification of the method in order to accelerate it, there is still space for the future work. We employ FFT to calculate efficiently the optical field of a patch but this causes some issues of the method. Since the patch is a special case of a plane, with a high probability there could be another, slightly less efficient approach to optical field calculation for that special case. Also, we may continue our work on acceleration using a distributed environment or a hardware such as the programmable hardware or the graphical processing unit (GPU). Furthermore, our method will definitely benefit from any research about digital diffusers. With ability to define quickly a patch that emits light in a custom direction, our method will be able to handle surface other than the diffuse ones without almost any modification.

Besides the proposed method, which is the major contribution, we contributed to acceleration of the method proposed by Martin Janda. Our goal was to reduce the calculation time. We proposed an acceleration through graphical processing unit (GPU). For that purpose, we reorganised the original algorithm such that we were able to facilitate the mesh processing ability of GPU. As a result, we achieved a significant speedup.

Besides that, we designed an approximation that can be used by the reduced occlusion method, which was designed by Martin Janda as well. We designed the approximation such that it can be implemented efficiently on a programmable hardware. The major feature is that the approximation uses a fixed point arithmetics and it does not impose any significant limitations to the scene, i.e., it can be used for larger optical fields and closer objects then other approximations. Even if implemented on CPU, it yields a speedup. Hence, in both minor contributions we achieve the goal of reducing the calculation time.

Unlike the proposed method, we cannot suggest to continue the research in a direction of pure geometry based renderers because it is rather a matter of small adjustment and fixes. This is caused by a fact that the geometry-based methods are extremely slow because they have to process too many elements. Nevertheless, the possible way out of it could be through a detail control. Using this, it is be possible to significantly decrease the number of elements as we have shown by proposing our new method.

In this thesis we presented a new method. We combined two trends of digital hologram generation. We showed that such an approach is possible. As a side-effect, we created a method that intentionally decreases the detail of the scene through which we can control the

calculation time. Hence, similar to the computer graphics, we have shown that we can gain speed by counting on limited ability of the human visual system.

Appendix A

List of Reviewed Published Works

List of published, reviewed works and their relevance to individual chapters of this thesis. Publications are sorted in order of relevance.

Chapter 4

- I. Hanák, M. Janda, and V. Skala. Detail Driven Digital Hologram Generation. Accepted in *The Visual Computer Journal*, currently available as an On-line first preview.

Chapter 5

- I. Hanák, M. Janda, and V. Skala. Full-parallax hologram synthesis of triangular meshes using a graphical processing unit. In *3DTV Conference proc.*. Piscataway : IEEE, 2007. pp. 1–4, 2007.
- I. Hanak, P. Zemcik, M. Zadnik, and A. Herout. Hologram Synthesis Accelerated in FPGA by Partial Quadratic Interpolation. In *The Optical Engineering*, 48(08), 085802, 2009.
- M. Janda, I. Hanák, and L. Onural. Hologram synthesis for photorealistic reconstruction. *J. Opt. Soc. Am. A*, 25(12):3038–3096, 2008.
- M. Janda, I. Hanák, and V. Skala. HPO hologram synthesis for full-parallax reconstruction setup. In *3DTV Conference proc.*. Piscataway : IEEE, 2007. pp. 1–4, 2007.
- M. Janda, I. Hanák, and V. Skala. Digital HPO hologram rendering pipeline. In *EG2006 short papers conf. proc.*, pp. 81–84, 2006.

Appendix B

Used Symbols and Notation

Table B.1: A notation of symbols used text-wide.

notation	description
$\cdot$	A dot product
$*$	A piece-wise multiplication
$\star$	A convolution
j	An imaginary number, $\mathrm{j}^2 = -1$
x	A scalar or a complex number
$\mathbf{x}$	A vector, $\mathbf{x} = (x_\mathbf{x}, y_\mathbf{x}, z_\mathbf{x})$
$\hat{\mathbf{x}}$	A normalised vector, $\hat{\mathbf{x}} = \frac{\mathbf{x}}{\lvert\mathbf{x}\rvert}$
$\mathbf{X}$	A matrix
X$[i]$	An i-th element of a table X
$\mathcal{X}$	X in a frequency domain, i.e, an angular spectrum
$\Re\{x\}$	A real part of a number x
$\Im\{x\}$	An imaginary part of a number x

Table B.2: Symbols used by state of the art.

symbol	description
κ	The recording plane, $\kappa : z = 0$
κ_ξ	A plane parallel with the recording plane, $\kappa : z = \xi$
λ	A wavelength
ν	A result of the visibility check, $\nu \in \{0, 1\}$
ϕ	A value proportional to the phase, the phase is $2\pi\phi$
φ	A phase
ρ_η	A plane parallel with the XZ-plane, $\rho_\eta : y = \eta$
σ	Fresnel approximation scale factor

Table B.3: Symbols used by the reduced detail method.

symbol	description
D_x	Sampling step of optical field in the X-axis
D_y	Sampling step of optical field in the Y-axis, usually equals to D_x
e^d_{lo}	A d-th patch that corresponds to the cell g_{lo}
g_{lo}	A cell of the visibility grid
h_i	An i-th intersection of a ray and the mesh
k	Wavenumber, $k = \frac{2\pi}{\lambda}$
$\mathbf{n}_{h_i}$	A normal a the intersection h_i
p^d_{lo}	A d-th pillar that corresponds to the cell g_{lo}
S_n	A horizontal slice, intersection of the scene and the plane ρ_{nD_y}
t_{lo}	A member of the visibility map T
U	Optical field values, usually a matrix of complex numbers
$\mathcal{U}$	An angular spectrum of an optical field U or a sample of the angular spectrum
$u(x, y)$	A sample of a continuous optical field U at $(x, y, 0)$
$u(\mathbf{p})$	A sample of a continuous optical field U at $\mathbf{p}$ and $\mathbf{p}$ respectively
u_{mn}	A sample of a discrete optical field
$\mathbf{u}_{mn}$	A location of the sample of a discrete optical field
z_{h_i}	A distance of the intersection h_i along the Z-axis
η	Index of a frequency in the frequency domain that corresponds to the X-axis.
κ	The recording plane, $\kappa : z = 0$
κ_ξ	A plane parallel with the recording plane, $\kappa : z = \xi$
λ	A wavelength
ϕ	A value proportional to the phase, the phase is $2\pi\phi$
ψ	Index of a frequency in the frequency domain that corresponds to the Y-axis.
ρ_η	A plane parallel with the XZ-plane, $\rho_\eta : y = \eta$

Appendix C

Parameters of Testing Scenes

Table C.1: Parameters of scenes used to verify functionality of proposed methods. The scenes are scaled to fit a rectangle 2.0×2.0 mm, i.e., $4,096 \times 4,096$ samples, sampling step 0.5 μm. The scene scales proportionally with the rectangle.

Scene	# triangles	Depth [mm]	Figure
Bunny	61,747	[6.0, 7.0]	Fig. C.1(a)
Chess	42,566	[6.0, 7.0]	Fig. C.1(b)
Plane	2	6.0	Fig. C.1(c)
Primitives	972	[6.0, 30.5]	Fig. C.1(d)
Primitives2	1,964	[6.0, 9.1]	Fig. C.1(e)
StillLifeBunny	84,580	[9.0, 11.0]	Fig. C.1(f)

Table C.2: A meaning of a symbol in superscript used for some scenes. The symbol defines the sampling step, the resolution of the calculated optical field and the maximum size of the orthogonal projection of the scene.

Symbol	Resolution of the optical field	Projection of the scene	Sampling step
none	$4,096 \times 4,096$	2.0×2.0 mm	0.5 μm
†	$1,024 \times 1,024$	0.5×0.5 mm	0.5 μm
‡	$6,144 \times 6,144$	43.0×43.0 mm	7.0 μm
*	$2,048 \times 2,048$	1.0×1.0 mm	0.5 μm

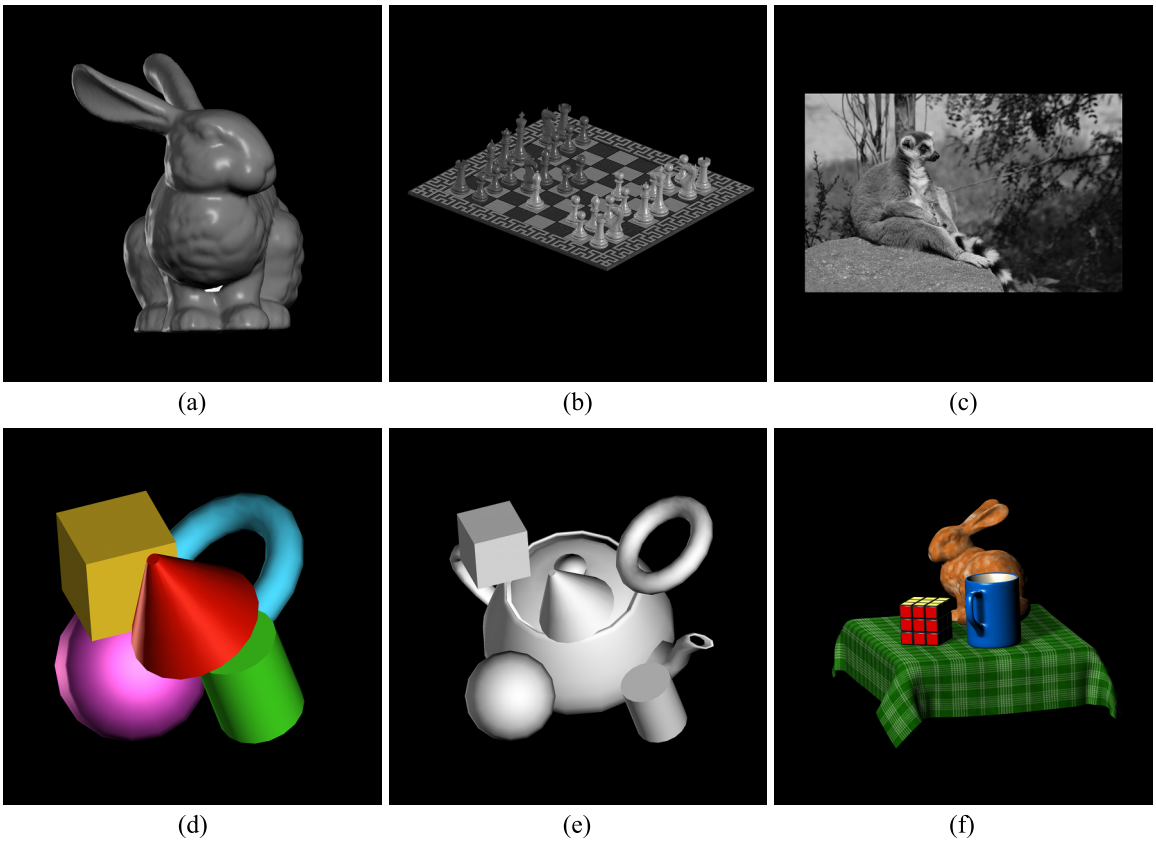


Figure C.1: Orthogonal projections of used scenes.

Bibliography

[ABMW06] L. Ahrenberg, P. Benzie, M. Magnor, and J. Watson. Computer generated holography using parallel commodity graphics hardware. *Opt. Express*, 14(17):7636–7641, 2006.

[ABMW08] L. Ahrenberg, P. Benzie, M. Magnor, and J. Watson. Computer generated holograms from three dimensional meshes using an analytic light transport model. *Appl. Opt.*, 47(10):1567–1574, 2008.

[AR03] D. Abookasis and J. Rosen. Computer-generated holograms of three-dimensional objects synthesized from their multiple angular viewpoints. *J. Opt. Soc. Am. A*, 20(8):1537–1545, 2003.

[BFJ+90] S. Bara, C. Frere, Z. Jaroszewicz, A. Kolodzijczyk, and D. Leseberg. Modulated on-axis circular zone plates for a generation of three-dimensional focal curves. *J. Mod. Opt.*, 37(8):1287–1295, 1990.

[BL66] B.R. Brown and A.W. Lohnmann. Complex spatial filtering with binary masks. *Appl. Opt.*, 5(6):967–969, 1966.

[BLCLO00] C. Buraga-Lefebvre, S. Coëtmellec, D. Lebrun, and C. Özkul. Application of wavelet transform to hologram analysis: three-dimensional location of particles. *Opt. Lasers Eng.*, 33:409–421, 2000.

[BW05] M. Born and E. Wolf. *Principles of Optics*. Cambridge University Press, 7th edition, 2005.

[EO06] G.B. Esmer and L. Onural. Computation of holographic patterns between tilted planes. In *Holography 2005*, volume 6252, page 62521K. SPIE, 2006.

[FJ] M. Frigo and S. G. Johnson. Fftw. WWW http://www.fftw.org/.

[FLB86] Ch. Frere, D. Leseberg, and O. Bryngdahl. Computer-generated holograms of three-dimensional objects composed of line segments. *J. Opt. Soc. Am. A*, 3(5):726–730, May 1986.

[FTI86] A. Fujimoto, T. Tanaka, and K. Iwata. Arts: Accelerated ray-tracing system. *IEEE Comput. Graphics Appl.*, 6(4):16–26, 1986.

[FVDFH96] J. D. Foley, A. Van-Dam, S. K. Feiner, and J. F. Hughes. *Computer Graphics: Principles and Practice*. Addison-Wesley, 1st edition, 1996.

[Gab49] D. Gabor. Microscopy by reconstructed wavefronts. *Proc. R. Soc. London, Ser. A*, 197:454–487, 1949.

[Goo05] J.W Goodman. *Introduction to Fourier Optics*. Roberts & Company Publishers, 3rd edition, 2005.

[Gra03] J.R. Graham. Wave optics. WWW http://grus.berkeley.edu/ jrg/ScalarWave/, 2003.

[Har96] P. Hariharan. *Optical Holography: Principles, techniques and applications*. Cambridge University Press, 2nd edition, 1996.

[Har05] A.H. Harker. 1b24 waves, optics and acoustics. WWW http://www.cmmp.ucl.ac.uk/ãhh/teaching/1B24n/, 2005.

[Hua71] T.S. Huang. Digital holography. *Proc. IEEE*, 59(9):1335–1346, 1971.

[IEE85] IEEE. Ieee standard for binary floating-point arithmetic (ansi/ieee std 754-1985), 1985.

[IMY+05] T. Ito, N. Masuda, K. Yoshimura, A. Shiraki, T. Shimobaba, and T. Sugie. Special-purpose computer horn-5 for a real-time electroholography. *Opt. Express*, 13(6):1923–1932, 2005.

[Int07] Intel. Intel sse4 programming reference. WWW `http://software.intel.com/en-us/articles/`, 2007.

[JHO08] M. Janda, I. Hanák, and L. Onural. Hologram synthesis for photorealistic reconstruction. *J. Opt. Soc. Am. A*, 25(12):3038–3096, 2008.

[JHS06] M. Janda, I. Hanák, and V. Skala. Digital HPO hologram rendering pipeline. In *EG2006 short papers conf. proc.*, pages 81–84, 2006.

[JHS07] M. Janda, I. Hanák, and V. Skala. Hpo hologram synthesis for full-parallax reconstruction setup. In *3DTV Conference proc.*, pages 1–4, 2007.

[JOPBV97] J. L. Juárez-Pérez, A. Olivares-Pérez, and L. R. Berriel-Valdos. Nonredundant calculation for creating digital fresnel holograms. *Appl. Opt.*, 36(29):7437–7443, 1997.

[KDS99] M. Koenig, O. Deussen, and T. Strothotte. Texture-based composition of holograms using triangular elements. In *IASTED*, pages 162–167, 1999.

[KDS01] M. Koenig, O. Deussen, and T. Strothotte. Texture-based hologram generation using triangles. In *Practical Holography XV and Holographic Materials VII*, volume 4296, pages 1–8. SPIE, 2001.

[KHL08] H. Kim, J. Hahn, and B. Lee. Mathematical modeling of triangle-mesh-modeled three-dimensional surface objects for digital holography. *Appl. Opt.*, 47(19):D117–D127, 2008.

[KIO+06] M. Kovachev, R. Ileva, L. Onural, G.B. Esmer, P. Benzie, J. Watson, and E. Mitev. Reconstruction of computer generated holograms by spatial light modulators. *Lecture Notes in Computer Science*, 2006.

[Kra04] F. Krausz. Photonics: Lecture notes, 2004.

[KSR07] B. Katz, N. T. Shaked, and J. Rosen. Synthesizing computer generated holograms with reduced number of perspective projections. *Opt. Express*, 15(20):13250–13255, 2007.

[KYY08] H. Kang, T. Yamaguchi, and H. Yoshikawa. Accurate phase-added stereogram to improve the coherent stereogram. *Appl. Opt.*, 47(19):D44–D54, 2008.

[LAe01] Y. Li, D. Abookasis, and J. editoren. Computer-generated holograms of three-dimensional realistic objects recorded without wave interference. *Appl. Opt.*, 40(17):2864–2870, 2001.

[Lal68] É. Lalor. Conditions for the validity of the angular spectrum of plane waves. *J. Opt. Soc. Am. A*, 58(9):1235–1237, 1968.

[LBL02] D.R. Luke, J.V. Burke, and R.G. Lyon. Optical Wavefront Reconstruction: Theory and Numerical Methods'. *SIAM Review*, 44:169–224, 2002.

[LBU04] M. Liebling, T. Blu, and M. Unser. Complex-wave retrieval from a single off-axis hologram. *J. Opt. Soc. Am. A*, 21(3):367–377, 2004.

[LF88] D. Leseberg and C. Frère. Computer-generated holograms of 3-D objects composed of tilted planar segments. *Appl. Opt.*, 27:3020–3024, 1988.

[LG95] M. Lucente and T. A. Galyean. Rendering interactive holographic images. In *SIGGRAPH '95*, pages 387–394, 1995.

[LHJ68] L.B. Lesem, P.M. Hirsch, and J.A. Jordan. Computer synthesis of holograms for 3-d display. *Commun. ACM*, 11(10):661–673, 1968.

[LL94] P. Lacroute and M. Levoy. Fast volume rendering using a shear-warp factorization of the viewing transformation. In *SIGGRAPH '94: Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 451–458. ACM, 1994.

[Lob08] P. Lobaz. Personal correspondence, 2008.

[Loh78] A. W. Lohmann. Three-dimensional properties of wave-fields. *Optik*, 51:105–117, 1978.

[Luc92] M. Lucente. Optimization of hologram computation for real-time display. In *Practical Holography VI*, volume 1667, pages 32–43. SPIE, 1992.

[Luc93] M. Lucente. Interactive computation of holograms using a look-up table. *J. El. Imag.*, 2:28–34, 1993.

[Luc94] M. Lucente. *Diffraction-Specific Fringe Computation for Electro-Holography.* PhD thesis, MIT, 1994.

[Luc96] M. Lucente. Computation holographic bandwidth compression. *IBM Systems Journal*, 35:349–365, 1996.

[Luc97] M. Lucente. Interactive three-dimensional holographic displays: Seeing the future in depth. *ACM SIGGRAPH Computer Graphics*, 31(2):63–67, 1997.

[Mac97] A.E Macgregor. Computer generated holograms from dot matrix and laser printers. *Am. J. Phys.*, 60(9):839–846, 1997.

[Mat05] K. Matsushima. Exact hidden-surface removal in digitally synthetic full-parallax hologram. In *Practical Holography XIX: Materials and Applications*, volume 5742, pages 25–32. SPIE, 2005.

[MIT+06] N. Masuda, T. Ito, T. Tanaka, A. Shiraki, and T. Sugie. Computer generated holography using parallel commodity graphics hardware. *Opt. Express*, 14(2):603–608, 2006.

[MK04] K. Matsushima and A. Kondoh. A wave optical algorithm for hidden-surface removal in digitally synthetic full-parallax holograms for three-dimensional objects. In *Practical Holography XVIII: Materials and Applications*, volume 5290, pages 90–97. SPIE, 2004.

[MKM06] K. Matsushima, S. Kobayashi, and H. Miyauchi. A high-resolution fringe printer for studying synthetic holograms. In *Practical Holography XX*, volume 6136, pages 347–354. SPIE, 2006.

[MNF+02] O. Matoba, T. J. Naughton, Y. Frauel, N. Bertaux, and B. Javidi. Three-dimensional object reconstruction using phase-only information from a digital hologram. In *Three-Dimensional TV, Video, and Display.*, volume 4864, pages 122–128. SPIE, 2002.

[MT00] K. Matsushima and M. Takai. Recurrence formulas for fast creation of synthetic three-dimensional holograms. *Appl. Opt.*, 39:6587–6594, 2000.

[NFJT02] T. J. Naughton, Y. Frauel, B. Javidi, and E. Tajahuerce. Compression of digital holograms for three-dimensional object reconstruction and recognition. *Appl. Opt.*, 41(20):4124–4132, 2002.

[NM08] T. Nakatsuji and K. Matsushima. Free-viewpoint images captured using phase-shifting synthetic aperture digital holography. *Appl. Opt.*, 47(19):D138–D143, 2008.

[NSM+05] S. Nishi, K. Shiba, K. Mori, S. Nakayama, and S Murashima. Fast calculation of computer-generated fresnel holograms utilizing distributed parallel processing and array operation. *Opt. Rev.*, 12(4):287–292, 2005.

[NVI08] NVIDIA. Nvidia cuda programming guide, version 2.1, 2008.

[Onu07] L. Onural. Exact analysis of the effects of sampling of the scalar diffraction field. *J. Opt. Soc. Am. A*, 24(2):359–367, 2007.

[Pho75] B.T. Phong. Illumination for computer generated pictures. *Commun. ACM*, 18(6):311–317, 1975.

[Ple03] W.J. Plesniak. Incremental update of computer-generated holograms. *Opt. Eng.*, 42:1560–1571, 2003.

[PM03] C. Petz and M. Magnor. Fast hologram synthesis for 3d geometry models using graphics hardware. In *Practical Holography XVII and Holographic Materials IX*, volume 5005, pages 266–275. SPIE, 2003.

[RBD+99] A. Ritter, J. Böttger, O. Deussen, M. König, and T. Strothotte. Hardware-based Rendering of Full-parallax Synthetic Holograms. *Appl. Opt.*, 38(11):1364–1369, 1999.

[SCS05] C. Slinger, C. Cameron, and M. Stanley. Computer-generated holography as a generic display technology. *Computer*, 38(8):46–53, 2005.

[SIY04] Y. Sando, M. Itoh, and T. Yatagai. Full-color computer-generated holograms using 3-D Fourier spectra. *Opt. Express*, 12:6246–6251, 2004.

[SJ05] U. Schnars and W. Juepner. *Optical Holography: Principles, techniques and applications*. Springer, 2005.

[Smi97] S.W. Smith. *The Scientist and Engineer's Guide to Digital Signal Processing*. California Technical Publishing, 1997.

[SMU04] Y. Sakamoto, M. Morishima, and A. Usui. Computer generated holograms on a cr-r disk. *Practical Holography XVIII*, 5290:42–49, 2004.

[SU94] Stanford Computer Graphics Laboratory Standford University. The stanford 3d scanning repository. WWW `http://www-graphics.stanford.edu/data/3Dscanrep/`, 1994.

[TB93] T. Tommasi and B. Bianco. Computer-generated holograms of tilted planes by a spatial frequency approach. *J. Opt. Soc. Am. A*, 10:299–305, February 1993.

[Und97] J.S. Underkoffler. Occlusion processing and smooth surface shading for fully computed synthetic holography. *Practical Holography XI and Holographic Materials III*, 3011:19–30, 1997.

[Wal95] A. Walther. *The Ray and Wave Theory of Lenses*. Cambridge University Press, 1995.

[Wat00] A. Watt. *3D Computer Graphics*. Addison-Wesley, 3rd edition, 2000.

[WB89] F. Wyrowski and O. Bryngdahl. Speckle-free reconstruction in digital holography. *J. Opt. Soc. Am. A*, 6(8):1171–1174, 1989.

[Wei] E. Weisstein. World of science. WWW http://scienceworld.wolfram.com/.

[YAC02] L. Yu, Y. An, and L. Cai. Numerical reconstruction of digital holograms with variable viewing angles. *Opt. Express*, 10:1250–+, 2002.

[YIO00] H. Yoshikawa, S. Iwase, and T. Oneda. Fast computation of fresnel holograms employing difference. In *Practical Holography XIV and Holographic Materials VI*, volume 3956, pages 48–55. SPIE, 2000.

[YZ97] I. Yamaguchi and T. Zhang. Numerical reconstruction of digital holograms with variable viewing angles. *Opt. Lett.*, 22(16):1268–1270, 1997.

[ZCG08] R. Ziegler, S. Croci, and M. Gross. Lighting and occlusion in a wave-based framework. *Computer Graphics Forum*, 27(2):211–220, 2008.

Index