

# INTERACTIVE POINT SPREAD FUNCTION SIMULATION WITH DIFFRACTION AND INTERFERENCE EFFECTS

Tom Cuypers, Tom Mertens, Philippe Bekaert

*Expertise Centre for Digital Media, Hasselt University - tUL - IBBT, Hasselt, Belgium*  
{firstname.lastname}@uhasselt.be

Se Baek Oh<sup>1</sup>, Ramesh Raskar<sup>2</sup>

<sup>1</sup>MIT Mechanical Engineering, MIT, Cambridge, U.S.A.

<sup>2</sup>MIT MediaLab, MIT, Cambridge, U.S.A.  
sboh@mit.edu, raskar@media.mit.edu

Keywords: PSF, Wigner distribution function, Diffraction, Interference, Grating, Monte Carlo.

Abstract: Interactive simulation of point spread functions is an invaluable tool for evaluating optical designs. We present an interactive method for simulating the point spread function for designs that require diffraction and interference effects. These effects occur when the design contains apertures whose size approaches the wavelength of light, typically in the form of gratings or masks. Traditional ray-based techniques are not suitable here, whereas wave-based methods are not immediately amenable to an efficient implementation due to their complexity. We propose a method based on the Wigner Distribution function. This function models wave optics at gratings, but does so in a ray-based framework. This enables us to simulate diffraction and interference effects efficiently, even for multiple gratings. The resulting computation is in the order of a fraction of a second, thereby enabling the user to interactively manipulate the optical configuration or the projection plane. The proposed method can be scaled down in precision in order to achieve real-time performance.

## 1 INTRODUCTION

Designing optical instruments such as lenses is an important task in various fields such as scientific imaging, photography, microscopy and holography. Before actually building, it is desirable to evaluate the design performance through a simulation. These designs often consist of a set of lenses and masks or apertures placed at different depths. The goal of the simulation is to obtain a point spread function (PSF), which is the recorded light intensity on the image sensor created by a single point light source. The PSF is indicative for the performance of the lens which in practice may translate into the image quality of a photographic lens, for instance. This paper deals with an efficient method to carry out such simulations.

Rendering a point spread function using geometric optics is straightforward and computationally efficient. Yet, it is inherently limited to large scale apertures as diffraction and interference effects are ignored. As the size of the masks shrinks to a scale closer to the size the wavelength of the light, these ef-

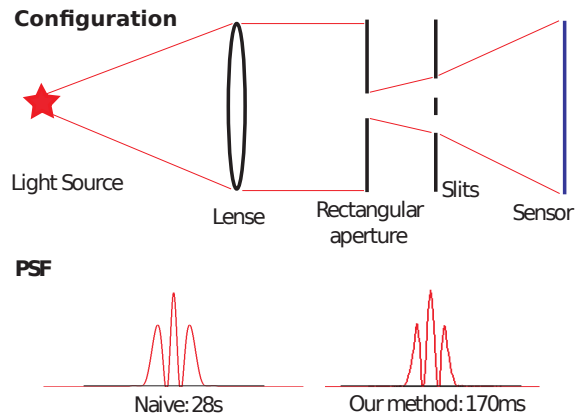


Figure 1: We present a point spread function simulation of light passing through a set of lenses and gratings. (Top) An example configuration of a lens system with several masks creating a diffraction pattern on the sensor. (Bottom) The point spread function calculated using simple phase tracking and our speed optimized method.

fects become more prominent. For instance, designs based on wavefront coding (Dowski and Cathey, ;

Horstmeyer et al., 2010; Schechner et al., 1996) will demand diffraction and interference. A typical application example is a lens with a depth-independent PSF.

Diffraction and interference effects are governed by laws from wave optics rather than geometric optics. This increases the implementation complexity but also the computation time of the simulation. The latter is not desirable when multiple iterations are required to find the optimal positions for each mask. The Wigner distribution function (Wigner, 1932) is a popular light representation and is applicable for diffraction and interference simulations in the optics community. It basically models light transport through a grating as a mathematical operation, and can be applied in a successive fashion for multiple gratings. The Wigner distribution function is defined in the space–spatial frequency domain which has recently been shown to have similar properties as the space–angle domain of the light field representation (Zhang and Levoy, 2009; Oh et al., 2010). Oh et al (Oh et al., 2010) demonstrated this idea to render interference patterns. This technique is valid in both close or far range (near-field and far-field), however it is a slow process as it relies on brute-force ray tracing.

We propose a Monte Carlo-based simulation technique for the Wigner distribution function. As these calculations are easy to perform in parallel, a GPU implementation is presented. We show an example configuration and corresponding PSF on Figure 1 and show the calculation speed–up compared to the naive calculation. The resulting computation is in the order of a fraction of a second, thereby enabling the user to interactively manipulate the optical configuration or the projection plane. The proposed method can be scaled down in precision in order to achieve real-time performance.

## 2 RELATED WORK

Light is often described as an electromagnetic field with amplitude and phase. The Huygens–Fresnel principle is often used to represent wave propagation, which is a convolution of point scatterers (Goodman, 2005). In contrast, geometrical optics treats light as a collection of rays. Among the extensive efforts to connect wave and ray optics (Wolf, 1978), the notable ones are the generalized radiance proposed by Walther (Walther, 1973) and the Wigner Distribution Function (Bastiaans, 1977; Bastiaans, 1981; Bastiaans, 1979), where light is described in terms of local spatial frequency, which has a simple relationship with the angular domain. Although the generalized

radiance or the WDF can be negative, they exhibit convenient properties and explain diffraction rigorously and conveniently (Bastiaans, 1997). We prefer this light representation as it allows us to create a probability function in space and spatial frequency for a more efficient Monte Carlo sampling.

In computer graphics, light simulation often involves solving the rendering equation (Kajiya, 1986) that describes the light propagation. Multiple techniques have been proposed to render wave phenomena in computer graphics. Moravec proposed a wave model to render complex light transport efficiently (Moravec, 1981), which is based on phase tracking. This technique keeps track of the travel distance of a ray and calculates its phase. Ziegler et al. developed a wave-based framework (Ziegler et al., 2008), where complex values can be assigned for occluders to account for phase effects. They also implemented hologram rendering based on wave propagation (with the spatial frequency) (Ziegler et al., 2007). Stam implemented a diffraction shader based on the Kirchhoff integral (Stam, 1999) for random or periodic patterns. Unfortunately, this technique assumes the light source and observer to be at infinity, and therefore not suitable for our system.

## 3 WIGNER DISTRIBUTION FUNCTION

The Wigner distribution function is a representation of light commonly used in the optics community. It is used to simulate of light in both near–field and far–field provided that the paraxial approximation is valid. This approximation assumes that the incoming light direction is close to the normal direction. For the purpose of plane to plane propagation of light, this assumption is valid.

The microstructure geometry of a diffracting surface can be represented as a complex function  $t(x)$  in space. The amplitude  $a(x)$  of  $t(x)$  is the amount of light passing through at position  $x$ . The phase part  $\phi(x)$  of  $t(x)$  represents the phase delay introduced to the light due to the thickness(height profile) and/or the refractive index of the surface. We can calculate the Wigner distribution function (Wigner, 1932) of the microstructure as

$$W(x, u) = \int t\left(x + \frac{x'}{2}\right) t^*\left(x - \frac{x'}{2}\right) e^{-i2\pi x' u} dx' \quad (1)$$

where  $x$  is the position,  $u$  the spatial frequency and  $*$  is the complex conjugate operator. As an incoming wavefront parallel with the diffracting surface is distorted due to the phase delay, the outgoing wave front

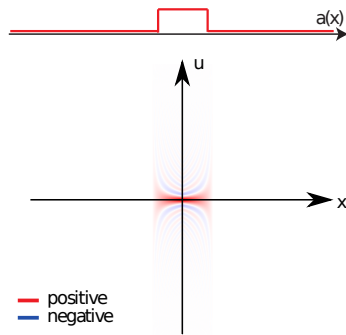


Figure 2: The Wigner distribution function of an aperture. (Top) The aperture has an amplitude function which is 1 where the light is passing and 0 elsewhere. (Bottom) The Wigner distribution of this aperture calculated using Eq 1.

is represented using the same equation in position and spatial frequency. An example is shown on Figure 2.

The spatial frequency contains the directional information of a wavefront. There is a simple relationship with the propagation direction  $\theta$ :

$$u = \sin(\theta)/\lambda \quad (2)$$

where  $\lambda$  is the wavelength of the light. The basic concept of the Wigner distribution function is to decompose the correlation function of a complex wavefront into a set of local plane wavefronts with different starting positions and spatial frequencies. The intensity of each local plane wavefront is a real value, which could be negative as well.

### 3.1 Properties

In order to simulate light propagating through a system of lenses and gratings we need a few additional operators of the Wigner distribution function:

**Propagation through Mid-air.** The Wigner distribution function  $W_z(x, u)$  of a complex wavefront will shear due to the traveling distance  $z$ , similar to the light field.

$$W_z(x, u) = W(x - \lambda uz, u) \quad (3)$$

**Propagation through a Grating.** The Wigner distribution function  $W_o$  of an outgoing wavefront is a convolution in spatial frequency of the Wigner distribution function of an incoming wavefront  $W_i$  and the Wigner distribution function of the grating  $W$ :

$$W_o(x, u) = \int W_i(x, u' - u)W(x, u')du' \quad (4)$$

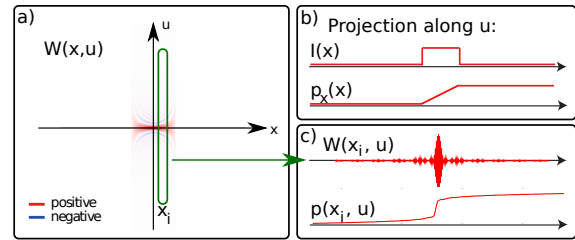


Figure 3: Schematic overview of the calculation of  $p_x$  and  $p$ . (a) The Wigner distribution function of the wavefront in position–spatial frequency (b) The projection of the wavefront along  $u$  is used to calculate  $p_x$  (c) At a position  $x_i$ , we construct  $p(x_i, u)$  using the absolute values of  $W(x_i, u)$ .

**Projection on a Surface.** The measured intensity when the light hits a surface such as a camera sensor is the projection along all spatial frequencies  $u$

$$I(x) = \int W(x, u)du \quad (5)$$

Even though the Wigner distribution function contains possible negative values, the observed intensity  $I(x)$  on a surface is always positive (Bastiaans, 1997).

### 3.2 Monte Carlo Simulation

In order to speed up these calculations, we want to numerically estimate the PSF using Monte Carlo simulations. As a simple example, we show the projection of our wavefront on a plane:

$$I(x) = \int W(x, u)du \quad (6)$$

$$\approx (u_{min} - u_{max}) \sum_{i=1}^N s(x, a) \quad (7)$$

where  $s$  is a sample contributing to the PSF at position  $x$ . This sample is calculated as

$$s(x, a) = W(x, a)p(x, a) \quad (8)$$

Which states that we can randomly sample a spatial frequency  $a$  at a position  $x$  in the wavefront and add this value to the total intensity  $I$  at position  $x$ . The function  $p$  defines the probability of selecting this spatial frequency and is added for normalization. In a case where we uniformly sample  $a$  is

$$p(x, a) = \frac{1}{N} \quad (9)$$

with  $N$  the amount of samples. This however is similar to phase tracking as we select  $N$  different paths throughout the optical elements.

Alternatively, we can efficiently sample  $I(x)$  by selecting our spatial frequency samples ( $a$ ) according to its probability function ( $p$ ). If we sample  $a$  within

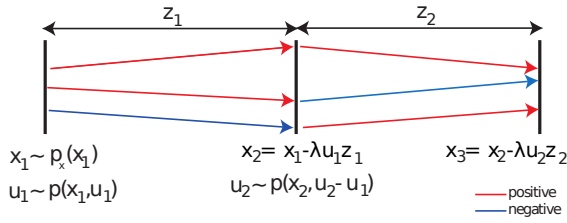


Figure 4: An overview of the Monte Carlo simulation of the point spread function. An initial random position and spatial frequency is sampled according to  $p_x$  and  $p$ . With each additional grating, a new outgoing spatial frequency is sampled until it reaches the final plane. The intensity of the sample is calculated based upon the Wigner distribution function of each grating.

a range of  $[u_{min}, u_{max}]$ , the probability function is calculated as

$$p(x, u) = \frac{\int_{u_{min}}^u |W(x, a)| da}{\int_{u_{min}}^{u_{max}} |W(x, a)| da}. \quad (10)$$

This is not possible with phase tracking as the importance of a direction is unknown. An example is demonstrated on Figure 3(a)(c).

When the wavefront has to propagate through space before it gets projected we have to include the traveling distance of the light  $z$  and a sample is can therefore be calculated as

$$s(x + \lambda zu) = W(x, a)p(x, a) \quad (11)$$

We can solve this by forward propagation, this involves choosing a random start position  $x_i$  and sample the spatial frequency  $u_i$  according to  $p$ . This sample will define a position  $x$  after a propagation distance  $z$  and is added to  $I(x)$ . Similar to the selective sampling of the spatial frequency we can sample the start position  $x_i$  using the probability function

$$p_x(x) = \frac{\int_{x_{min}}^x \int |W(a, u)| du dx}{\int_{x_{min}}^{x_{max}} \int |W(a, u)| du dx}. \quad (12)$$

This function is also shown on Figure 3(b).

Finally, when a wavefront passes through a grating, we can also estimate this convolution as

$$W_o(x, u_o) = \int W(x, u_o - u_i) W_i(x, u_i) du_i \quad (13)$$

$$\approx \sum_{i=1}^N \frac{W(x, u_o - a)}{p(x, u_o - a)} W_i(u, a) \quad (14)$$

Which we simulate by sampling an outgoing spatial frequency  $u_i$  according to  $p(x, u)$ . A schematic overview of this simulation is illustrated in Figure 4.

## 4 IMPLEMENTATION

Because of the nature of Monte Carlo simulations, this technique is very suitable for parallel execution. Therefore we implement part of the algorithm on the GPU. The implementation can be divided into a pre-processing step calculating the Wigner distribution functions and other lookup tables, and rendering step calculating the diffraction pattern. The implementation on the CPU is written in C++ for speed and efficiency. The GPU part is implemented using OpenGL and Cg shaders.

### 4.1 Pre-processing

We start by calculating a discrete Wigner distribution function for each diffraction grating micro-structure  $t$ , provided by the user. For example, a rectangular aperture as shown in Figure 2 has an amplitude function of  $a(x) = \text{rect}(x/A)$ , where  $A$  is the size of the aperture. We used the fast Fourier transform for an efficient calculation of the correlation function and the Fourier decomposition in Eq. (1). This information is stored into a lookup table in the form of a texture. As textures are build to hold positive valued intensities, the positive values of the Wigner distribution function is stored in the red channel and the negative values in the green channel.

Having this lookup table also allows precomputing the probability functions  $p(x, u)$  and  $p_x(x)$  as provided by Eq. (10) and Eq. (12). As we work with a discrete function, we can easily invert both these functions which we store in the blue and alpha channel of the texture. This makes sampling of a position and orientation according to the probability very cheap as it only requires a single texture lookup.

### 4.2 Rendering

The rendering consists of  $N$  samples which travel from the first grating through the system until they reach a projection surface. We start by creating these samples as a set of vertex points in OpenGL. The coordinates of these vertices are randomly chosen between 0 and 1 as we do not have a function to generate random values on the GPU. We used the NVidia Cg library to create a vertex shader that performs the Monte Carlo simulation based on the random values provided by the coordinates of the vertex and the probability function provided by the precomputed textures.

For each sample, we calculate the position on the projection screen and translate the sample to this position in the vertex shader. The intensity is calculated

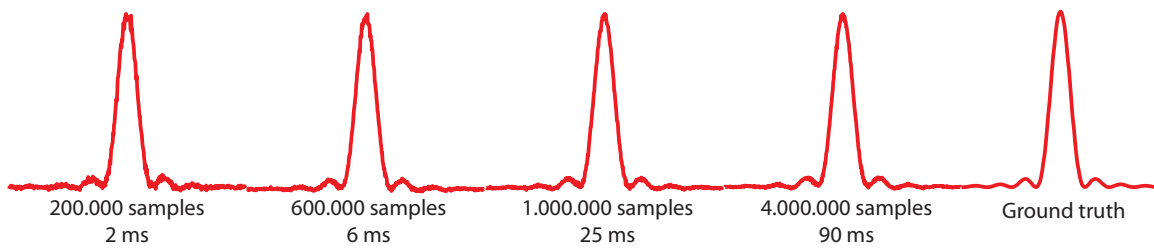


Figure 5: The amount of samples creates a trade-off between accuracy and speed. Here we show four different renderings of light passing through a single aperture.

through this path and saved in the red and green color channels of the vertex similar to the creation of the textures. These samples are then summed up using the blend function of the GPU. Due to the separation of the positive and negative values of each sample (we can enable the blending using `glBlendFunc(GL_ONE, GL_ONE)`). For precision we render this process to a 32bit floating point frame buffer.

## 5 RESULTS

The accuracy of the Monte Carlo simulation greatly depends on the amount of samples we use. This, however, also increases the calculation speed. Figure 5 shows a comparison between the PSF calculated using our method and a ground truth of a rectangular aperture. We notice that by increasing the amount of samples our results converges to the ground truth.

In the preprocessing step, the lookup-tables are created and stored into a texture on the GPU. This step depends on the resolution of the lookup-tables. To calculate a Wigner distribution lookup table of a resolution of  $1024 \times 1024$  and the probability lookup tables with a resolution of 1024 takes around 800 ms to calculate on a Intel Core2 6300 – 1.8GHz CPU. The rest of the result in this section are performed on an NVidia GeForce 8800 GT.

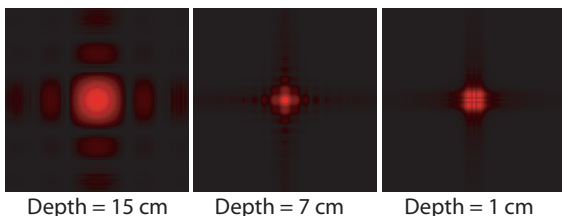


Figure 6: The Monte Carlo sampling works both in near-field and far-field. We show the PSF of a rectangular aperture measured at three different depths. We used 3 million samples to generate these renderings, which is performed at 80 ms for each rendering.

A speed-up compared to the naive implementation of the operators of the Wigner distribution function is presented in Figure 1. The propagation through a single grating using the naive method requires a computation of 150 ms, which can be reduced to a 15 ms using our method. Propagating through multiple gratings can reduce the time from 28s to 0.2s.

Finally, we show its applicability in both near-field and far-field, which is necessary to construct PSFs of systems with relative small distance between the masks. Figure 6 illustrates a PSF of a rectangular aperture in both near- and far-field. The PSF is calculated in both  $x$  and  $y$  directions and multiplied to achieve the presented results.

## 6 CONCLUSIONS

We presented an efficient implementation for calculating PSFs using the Wigner distribution function. Using this representation, we are able to calculate the diffraction and interference created by small scale structures of gratings or lenses. Utilizing Monte Carlo sampling on this function, we are able to simulate the PSF at interactive speeds. This allows users to interactively adjust parameters.

### 6.1 Future Work

We demonstrate the calculation of a PSF created by a 1D or 2D gratings which are separable in the 2 dimensions. Extending the theory to a full 2D gratings is straight forward, but requires a large amount of memory as the Wigner distribution becomes a function of 4D variables. In order to cope with the limited memory, we require an intelligent compression method for data.

## ACKNOWLEDGEMENTS

The authors (Tom Cuyper, Tom Mertens and Philippe Bekaert) acknowledge financial support by

the ERDF (European Regional Development Fund), the European Commission (FP7 IP 2020 3D media) and the Flemish government. Furthermore, we would like to thank our colleagues and reviewers for their useful comments and suggestions.

Ziegler, R., Bucheli, S., Ahrenberg, L., Magnor, M., and Gross, M. (2007). A bidirectional light field - hologram transform. *Comp. Gr. Forum*.

Ziegler, R., Croci, S., and Gross, M. H. (2008). Lighting and occlusion in a wave-based framework. *Computer Graphics Forum*.

## REFERENCES

- Bastiaans, M. (1997). Application of the wigner distribution function in optics. In *The Wigner Distribution - Theory and Applications in Signal Processing*, pages 1227–1238. OSA.
- Bastiaans, M. J. (1977). Frequency-Domain Treatment of Partial Coherence. *Optica Acta*, 24(3):261–274.
- Bastiaans, M. J. (1979). Wigner Distribution Function and Its Application to 1st-order Optics. *J. Opt. Soc. Am.*, pages 1710–1716.
- Bastiaans, M. J. (1981). The Wigner Distribution Function of Partially Coherent-Light. *Optica Acta*, pages 1227–1238.
- Dowski, E. R. and Cathey, W. T. Extended Depth of Field through Wave-Front Coding. *Applied Optics*.
- Goodman, J. W. (2005). *Introduction to Fourier optics*. Roberts & Co., Englewood, Colo., 3rd edition.
- Horstmeyer, R., Oh, S. B., and Raskar, R. (2010). Iterative aperture mask design in phase space using a rank constraint. *Opt. Express*.
- Kajiya, J. T. (1986). The rendering equation. In *Proceedings of the 13th annual conference on Computer graphics and interactive techniques, SIGGRAPH '86*, pages 143–150, New York, NY, USA. ACM.
- Moravec, H. P. (1981). 3d graphics and the wave theory. In *Proceedings of the 8th annual conference on Computer graphics and interactive techniques, SIGGRAPH '81*, pages 289–296, NY, USA. ACM.
- Oh, S. B., Kashyap, S., Garg, R., Chandran, S., and Raskar, R. (2010). Rendering wave effects with augmented light fields. *EuroGraphics*.
- Schechner, Y. Y., Piestun, R., and Shamir, J. (1996). Wave propagation with rotating intensity distributions. *Phys. Rev. E*.
- Stam, J. (1999). Diffraction shaders. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques, SIGGRAPH '99*, pages 101–110, New York, NY, USA. ACM Press/Addison-Wesley Publishing Co.
- Walther, A. (1973). Radiometry and Coherence. *J. Opt Soc. Am.*
- Wigner, E. (1932). On the quantum correction for thermodynamic equilibrium. *Physical Review*.
- Wolf, E. (1978). Coherence and Radiometry. *Journal of Optical Society of America*.
- Zhang, Z. and Levoy, M. (2009). Wigner distributions and how they relate to the light field. In *IEEE ICCP*.