An Interactive Shader for Natural Diffraction Gratings

Bachelorarbeit

der Philosophisch-naturwissenschaftlichen Fakultät
der Universität Bern

vorgelegt von

Michael Single

2014

Leiter der Arbeit:
Prof. Dr. Matthias Zwicker
Institut für Informatik und angewandte Mathematik

Abstract

In nature, animals exhibit structural colors because of the physical interaction of light with the surface nanostructure of their exterior skin. In his pioneering work, J.Stam developed a reflectance model based on wave optics capturing the effect of diffraction from surface nanostructures. His model is limited by an accurate estimate of the correlation function using statistical properties of the surface's height field. We propose an adaption of his BRDF model that can handle complex natural gratings. Furthermore, we describe a method for interactively rendering of diffraction effects due to interaction of light with biological nano-structures such as snake skin. As input data, our method uses discrete height fields of natural gratings acquired by using atomic force microscopy (AFM) and employ Fourier Optics. Based on Taylor Series approximation for the phase shifts at the nanoscale surface, we leverage the precomputation of the discrete Fourier Transformations, involved in our model, to achieve interactive rendering speed (about 5-15 fps). We demonstrate results of our approach using surface nano-structures of two snake species, namely the Elaphe and Xenopeltis species, when applied to a measured snake geometry. Lastly, we evaluate the quality of our method by comparing its (peak) viewing angles with maximum reflectance for a fixed incident beam with those resulting from the grating equation at different wavelengths. We conclude that our method produces accurate results for complex, natural gratings at interactive speed.

Contents

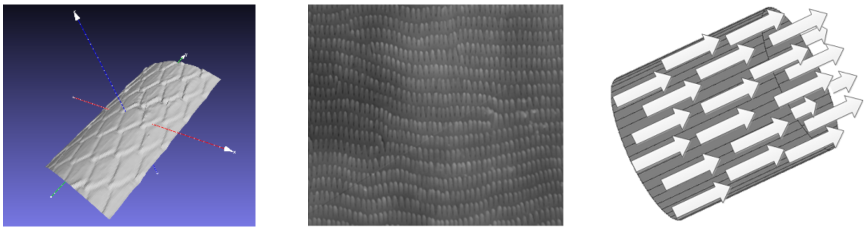
Chapter 1

Derivations

1.1 Problem Statement and Challenges

The goal of this thesis is to perform a physically accurate and interactive simulation of structural colors production like shown in figure 1.2, which we can see whenever a light source is diffracted on a natural grating. For this purpose we need to be provided by the following input data as shown in figure 1.1:

- A mesh representing a snake surface[1] with associated texture coordinates as shown in figure 1.1(a).
- A natural diffraction grating represented as a height field, its maximum height and its pixel-width-correspondence[2].
- A vectorfield which describes how fingers on a provided surface of the nanostructrue are aligned as shown in figure 1.1(c).



(a) Structure Geometry (b) Nanostructure Surface (c) Patch Orientation

Figure 1.1: Input for our simulation

We want to rely on the integral equation **??** derived by J. Stam in his paper [Sta99] about diffraction shaders. This equation formualtes a BRDF modeling the effect of diffraction under the assumption that a given grating can either be formulated as an analytical function or its structure is

[1] Which is in our simulation an actual reconstruction of a real snake skin. These measurements are provided by the Laboratory of Artificial and Natural Evolition at Geneva.See their website:`www.lanevol.org`.

[2] Since the nanostructure is stored as a grayscale image, we need a scale telling us what length and height one pixel cooresponds to in this provided image.

simple enough beeing modeled relying on statistical methods. These assumptions guarantee that **??** has an explicit solution. However, the complexisty of a biological nanostructures cannot sufficiently and accurately modeled simply using statistical methods. This is why interactive computation at high resolution becomes a hard task, since we cannot evaluate the given integral equation on the fly. Therefore, we have to adapt Stam's equation such that we are able to perform interactive rendering using explicitly provided height fields.

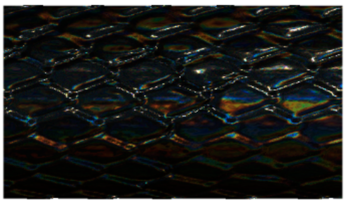


Figure 1.2: Output: Rendered Structural Colors

1.2 Approximate a FT by a DFT

1.2.1 Reproduce FT by DTFT

In the previous section, we have found an identity for the reflected spectral radiance $L_\lambda(\omega_r)$ when using Stam's BRDF for a given input height field. However, the derived expression in equation 1.13 requires to evaluate the Fourier Transform of our height field[3] for every direction. In this section we explain how to approximate the FT by the DTFT and apply it to our previous derivations. Figure 1.3 graphically shows how to obtain the DTFT from the FT for a one dimensional signal[4]

The first step is to uniformly discretize the given signal since computers are working finite, discrete arithmetic. We rely on the Nyquist–Shannon sampling theorem tells us how dense we have to sample a given signal $s(x)$ such that can be reconstructed its sampled version $\hat{s}[n]$[6]. In particular, a sampled version according to the Nyquist–Shannon sampling theorem will have the same Fourier Transform as its origianl singal. The sampling theorem states that if f_{max} denotes the highest frequency of $s(x)$, then, it has to be sampled by a rate of f_s with $2f_{max} \leq f_s$ in order to be reconstructable. By convention $T = \frac{1}{f_s}$ represent the interval length between two samples.

Next, we apply the Fourier Tranformation operator on the discretized singal $\hat{s}$ which gives us the following expression:

[3] actually it requires the computation of the inverse Fourier Transform of a transformed version of the given heightfield, the function p(x,y) defined in equation **??**.

[4] For our case we are dealing with a two dimensional, spatial signal, the given height field. Nevertheless, without any constraints of generality, the explained approach applies to multi dimensional problems.

[5] Images of function plots taken from `http://en.wikipedia.org/wiki/Discrete_Fourier_transform` and are modified.

[6] n denotes the number of samples.

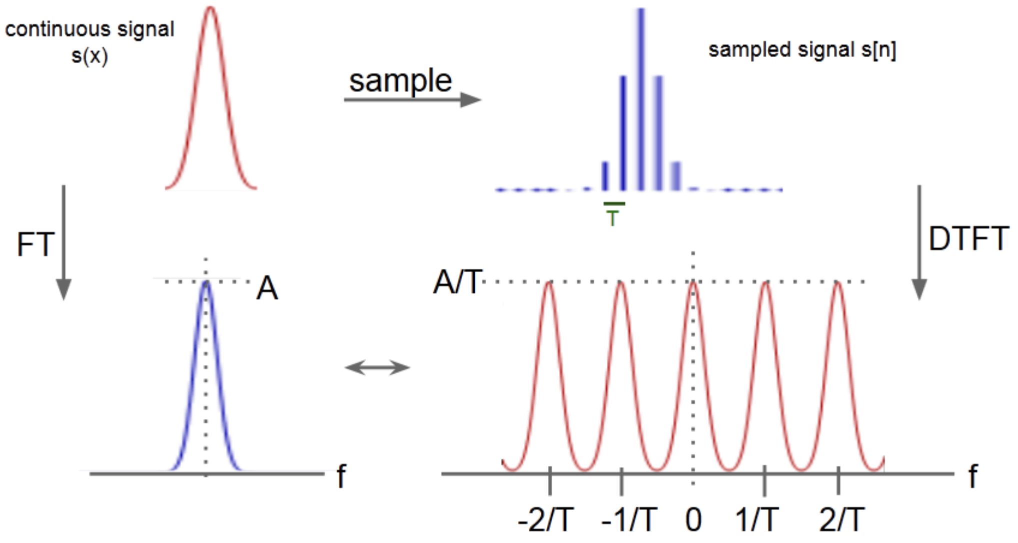


Figure 1.3: Illustration of how to approximate the analytical Fourier Transform (FT) [5] of a given continuous signal by a Discrete Time Fourier Transform (DTFT). The DTFT applied on a bandlimited, discretized signal yields a continuous, periodic response in frequency space.

$$
\begin{aligned}
\mathcal{F}_{FT}\{\hat{s}\}(w) &= \int_{\mathbb{R}} \hat{s}[n]e^{-iwx}dx \\
&= \int_{\mathbb{R}} mask(x)s(x)e^{-iwx}dx \\
&= T\sum_{x=-\infty}^{\infty} \hat{s}[x]e^{-iwx} \\
&= T\mathcal{F}_{DTFT}\{s\}(w)
\end{aligned}
\tag{1.1}
$$

Equation 1.1 tells us that if $\hat{s}$ is sufficiently sampled, then its DTFT corresponds to the FT of $s(x)$. Notice that the resulting DTFT from the sampled signal has a height of $\frac{A}{T}$ where A is the height of the FT of s and thus is a scaled version of the FT.

For a given height field h, let us compute Stam's auxiliary function p defined as in equation ??. For the reminder of this thesis we introduce the following definition:

$$P_{dtft} \equiv \mathcal{F}_{DTFT}\{p\} \tag{1.2}$$

Therefore P_{dtft} denotes the DTFT of a transformed version of our height field h [7].

1.2.2 Spatial Coherence and Windowing

Before we can derive a final expression in order to approximate a FT by a DFT, we first have to revisit the concept of coherence introduced in section ?? of chapter 2.Previousely we have seen

[7] By transformed height field we mean $p(x,y) = e^{i\frac{2\pi}{\lambda}wh(x,y)}$ which we get, when we plug h into equation ?? and this expression again plug into equation ??.

that Stam's BRDf tells us what is the total contribution of all secondary sources which allows us to say what is the reflected spectral radiance at a certain point in space. This is related to stationary interference which itself depends on the coherence property of the emitted secondary wave sources. The ability for two points in space, t_1 and t_2, to interfere in the extend of a wave when being averages over time is the so called spatial coherence. The spatial distance between such two points over which there is significant intererence is limited by the quantity coherence area. For filtered sunlight on earth this is equal to $65\mu m$ [8].

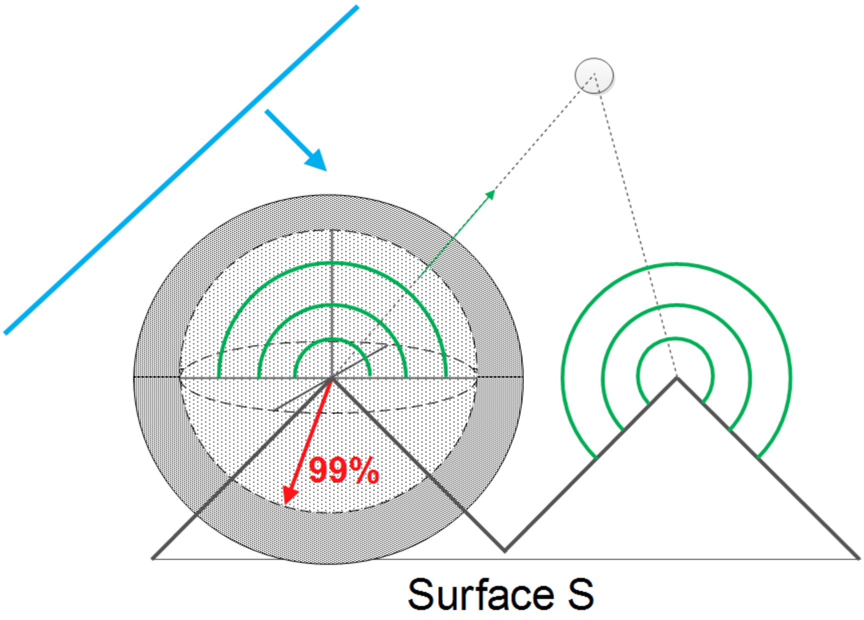


Figure 1.4: A plane wave encoungers a surface. According to Huygens principle, secondary wavelets are emitted of from this surface. The resulting wave at a certain point in space (here indicated by a gray circle) depends on the inteference among all waves encountering at this position. The amount of significant interference is directly affected by the spatial coherence property of all the wavelets.

Figure 1.4 illustrates the concept of spatial coherence. A wavefront (blue line) encounters a surface. Due to Hugen's Principle, secondary wavelets are emitted off from the surface. The reflected radiance at a certain point in space, e.g. at a viewer's eye position (denoted by the gray circle), is a result of interference among all wavelets at that point. This interference is directly affected by the spatial coherence property of all the emitted wavelets.

In physics spatial coherence is predicted by the cross correlation between t_1 and t_2 and usually modeled by by a Gaussian Random Process. For any such Gaussian Processes we can use a spatial gaussian window $g(x)$ which is equal:

$$g(x) = \frac{1}{\sqrt{2\pi} \cdot \sigma} \cdot e^{-\frac{x^2}{2\sigma^2}} \tag{1.3}$$

[8] A proof for this number can be looked up in the book Optical Coherence and Quantum Optics[LM95] on page 153 and 154.

We have chosen standard deviation σ_s of the window such that it fulfills the equation $4\sigma_s = 65\mu m$. This is equivalent like saying we want to predict about 99.99%[9] of the resulting spatial coherence interference effects in our model by a cross correlation function.

By applying the Fourier Transformation to the spatial window we get the corresponding window in frequency space will look like:

$$G(f) = e^{-\frac{f^2}{2\sigma_f^2}} \tag{1.4}$$

Notice that this frequency space window has a standard deviation σ_f equal to $\frac{1}{2\pi\sigma_s}$. Those two windows, the spatial- and the frequency space window, will be used in the next section in order to approximate the DTFT by the DFT by a windowing apporach.

1.2.3 Reproduce DTFT by DFT

In this section we explain how and under what assumptions the DTFT of a discretized signal[10] can be approximated by a DFT. The whole idea how to reproduce the DTFT by DFT is schematically illustrated in figure 1.5.

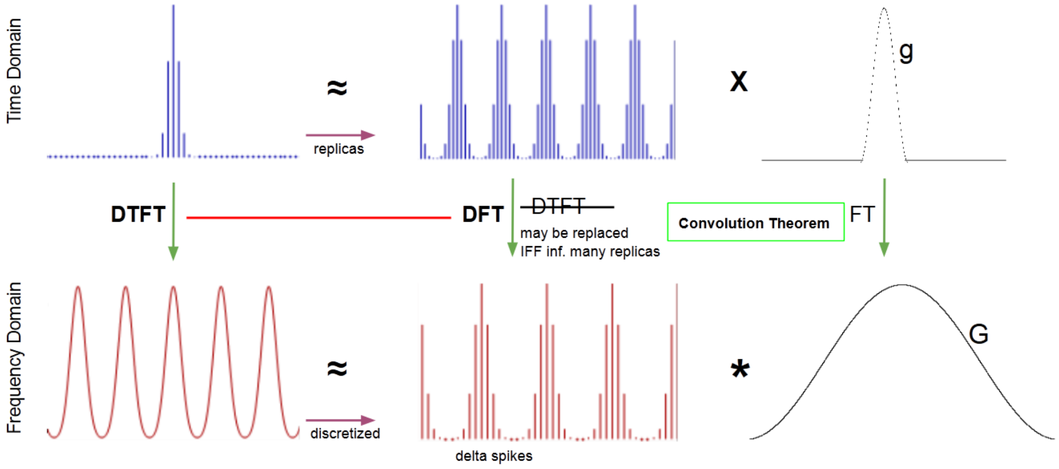


Figure 1.5: Illustration of how to approximate the DTFT [11] by the DFT relying on the Convolution Theorem, using a gaussian window function.

Given a spatial, bandlimited and discretized one dimensional signal $\hat{s}$. Our goal is to approximate this spatial signal in a way such that when taking the DTFT of this approximated signal, it will yield almost the same like taking the DTFT of the original sampled $\hat{s}$. For this purpose we will use the previouse introduced concept of gaussian windows and the so called Convolution Theorem which is a fundamental property of all Fourier Transformations.

[9] Standard deviation values from confidence intervals table of normal distribution provided by Wolfram MathWorld `http://mathworld.wolfram.com/StandardDeviation.html`.

[10] E.g. a sampled signal like already presented in figure 1.3

[11] Images of function plots taken from `http://en.wikipedia.org/wiki/Discrete_Fourier_transform` and are modified. Note that the scales in the graphic are not appropriate.

The Convolution Theorem states that the Fourier Transformation of a product of two functions, f and g, is equal to convolving the Fourier Transformations of each individual function. Mathematically, this statement corresponds to equation 1.5:

$$\mathcal{F}\{f \cdot g\} = \mathcal{F}\{f\} * \mathcal{F}\{g\} \tag{1.5}$$

The principal issue is how to approximate our given signal $\hat{s}$. Therefore, let us consider another signal $\hat{s_N}$ which is the N times replicated version of $\hat{s}$ (blue signal at center top in figure).

Remeber that in general, the signal repsonse at a certain point in space is the result of interference among all signals meeting at that position. In our scenario, the source of those signals are emitted secondary wavelets. The interference strength between these points is related to their spatial coherence. Windowing the signals by a gaussian window g will capture a certain percentage of all interference effects. From the previous section 1.2.2 we know that we can use gaussian window like in equation 1.3 in order to approximate such spatial signals interference effects.

Using this insight, we can approximate $\hat{s}$ by taking the product of $\hat{s_N}$ with a gaussian window g. This fact is illustrated in the first row of figure 1.3. So what will the DTFT of this approximation yield? We already know that the DTFT of $\hat{s}$ is a continuous, periodic signal, since $\hat{s}$ is bandlimited. Thus, taking the DTFT of this found approximation should give us approximatively the same continuous, periodic signal.

This is where the convolution theorem comes into play: Applying the DTFT to the product of $\hat{s_N}$ and g is the same as convolving the DTFT of $\hat{s_N}$ by DTFT of g. From equation 1.4 we already know that the DTFT of g is just another gaussian, denoted by G. On the other hand the DTFT of $\hat{s_N}$ yields a continuous, periodic signal. The higher the value of N, the sharper the signal gets (denoted by delta spiked) and the closer it converges toward to the DFT. This is why the DFT is the limit of a DTFT applied on periodic and discrete signals. Therefore, for a large number of N we can replace the DTFT by the DFT operator when applied on $\hat{s_N}$.

Lastly, we see that the DTFT of $\hat{s}$ is approximitely the same like convolving a gaussian window by the DFT of $\hat{s_N}$. This also makes sense, since convolving a discrete, periodic signal (DFT of $\hat{s_N}$) by a continuous window function G yiels a continuous, periodic function.

In practise, we cannot compute the DTFT **??** numerically due to finite computer arithmetic and hence working with the DFT is our only option. Furthermore, there are numerically fast algorithmes in order to compute the DFT values of a function, the Fast Fourier Transformation (FFT). The DFT **??** of a discrete height field is equal to the DTFT of an infinitely periodic function consisting of replicas of the same height field. Not, let a spatial gaussian window g having a standard deviation for which $4\sigma_s$ is equal μm. Then, from before, it followis:

$$\mathcal{F}_{dtft}\{\mathbf{s}\} \equiv \mathcal{F}_{dft}\{\mathbf{s}\} * G(\sigma_f) \tag{1.6}$$

Therefore we can deduce the following expression from this:

$$\begin{aligned}
\mathcal{F}_{dtft}\{\mathbf{t}\}(u,v) &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F_{dft}\{\mathbf{t}\}(w_u,w_v)\phi(u-w_u,v-w_v)dw_u dw_v \\
&= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\sum_i\sum_j F_{dft}\{\mathbf{t}\}(w_u,w_v) \\
&\quad \delta(w_u-w_i,w_v-w_j)\phi(u-w_u,v-w_v)dw_u dw_v \\
&= \sum_i\sum_j\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F_{dft}\{\mathbf{t}\}(w_u,w_v) \\
&\quad \delta(w_u-w_i,w_v-w_j)\phi(u-w_u,v-w_v)dw_u dw_v \\
&= \sum_i\sum_j F_{dft}\{\mathbf{t}\}(w_u,w_v)\phi(u-w_u,v-w_v) \qquad (1.7)
\end{aligned}$$

where

$$\phi(x,y) = \pi e^{-\frac{x^2+y^2}{2\sigma_f^2}} \qquad (1.8)$$

1.3 Adaption of Stam's BRDF Discrete Height Fields

1.3.1 Rendering Equation

As already discussed in the theoretical background chapter, colors are associated to radiance. Since we are starting with Stam's BRDF[12] formulation but want to perform a simulation rendering structural colors, we have to reformulate this BRDF equation such that we will end up with an identity of the reflected spectral radiance. This is where the rendering equation comes into play. Lets assume we have given an incoming light source with solid angle ω_i and θ_i is its angle of incidence, ω_r is the solid angle for the reflected light. Further let λ denote the wavelength[13] and Ω is the hemisphere of integration for the incoming light. Then, we are able to formulate a $BRDF_\lambda$ by using its definition **??**:

$$\begin{aligned}
f_r(\omega_i,\omega_r) &= \frac{dL_r(\omega_r)}{L_i(\omega_i)cos(\theta_i)d\omega_i} \\
\Rightarrow f_r(\omega_i,\omega_r)L_i(\omega_i)cos(\theta_i)d\omega_i &= dL_r(\omega_r) \\
\Rightarrow \int_\Omega f_r(\omega_i,\omega_r)L_i(\omega_i)cos(\theta_i)d\omega_i &= \int_\Omega dL_r(\omega_r) \\
\Rightarrow \int_\Omega f_r(\omega_i,\omega_r)L_i(\omega_i)cos(\theta_i)d\omega_i &= L_r(\omega_r) \qquad (1.9)
\end{aligned}$$

The last equation is the so called rendering equation . We assume that our incident light is a directional, unpolarized light source like sunlight and therefore its radiance is given as

$$L_\lambda(\omega) = I(\lambda)\delta(\omega-\omega_i) \qquad (1.10)$$

[12] Remember that a BRDF is the portion of a incident light source reflected off a given surface towards a specified viewing direction.

[13] Notice that, to keep our terms simple, we have droped all λ subscripts for spectral radiance quantites.

where $I(\lambda)$ is the intensity of the relative spectral power for the wavelength λ. By plugging the identity in equation 1.10 into our current rendering equation 1.9, we will get:

$$\begin{aligned} L_\lambda(w_r) &= \int_\Omega BRDF_\lambda(\omega_i, \omega_r) L_\lambda(\omega_i) cos(\theta_i) d\omega_i \\ &= BRDF_\lambda(\omega_i, \omega_r) I(\lambda) cos(\theta_i) \end{aligned} \tag{1.11}$$

where $L_\lambda(\omega_i)$ is the incident radiance and $L_\lambda(\omega_r)$ is the radiance reflected by the given surface. Note that the integral in equation 1.11 vanishes since $\delta(\omega - \omega_i)$ is only equal one if and only if $\omega = \omega_i$.

1.3.2 Reflected Radiance of Stam's BRDF

We are going to use Stam's main derivation (**??**) for the $BRDF(\omega_i, \omega_r)$ in 1.11 by applying the fact that the wavenumber is equal $k = \frac{2\pi}{\lambda}$:

$$\begin{aligned} BRDF(\omega_i, \omega_r) &= \frac{k^2 F^2 G}{4\pi^2 A w^2} \langle |P(ku, kv)|^2 \rangle \\ &= \frac{4\pi^2 F^2 G}{4\pi^2 A \lambda^2 w^2} \langle |P(ku, kv)|^2 \rangle \\ &= \frac{F^2 G}{A\lambda^2 w^2} \langle \left| P(\frac{2\pi u}{\lambda}, \frac{2\pi v}{\lambda}) \right|^2 \rangle \end{aligned} \tag{1.12}$$

Going back to equation 1.11 and plugging equation 1.12 into it, using the definition of equation **??** and the equation **??** for ω we will get the following:

$$\begin{aligned} L_\lambda(\omega_r) &= \frac{F^2(1 + \omega_i \cdot \omega_r)^2}{A\lambda^2 cos(\theta_i) cos(\theta_r) \omega^2} \left\langle \left| P\left(\frac{2\pi u}{\lambda}, \frac{2\pi v}{\lambda}\right) \right|^2 \right\rangle cos(\theta_i) I(\lambda) \\ &= I(\lambda) \frac{F^2(1 + \omega_i \cdot \omega_r)^2}{\lambda^2 A \omega^2 cos(\theta_r)} \left\langle \left| P\left(\frac{2\pi u}{\lambda}, \frac{2\pi v}{\lambda}\right) \right|^2 \right\rangle \end{aligned} \tag{1.13}$$

Note that the Fresnel term F is actually a function of (w_i, w_r), but in order to keep the equations simple, we omitted its arguemts. So far we just plugged Stam's BRDF identity into the rendering equation and hence have not significantly deviated from his formulation. Keep in mind that P deontes the Fourier transform of the provided height field which depends on the viewing and incidence light direction. Thus this Fourier Transform has to be recomputed for every direction which will slow down the whole computation quite a lot[14]. One particular strategy to solve this issue is to approximate P by the Discrete Fourier Transform (DFT)[15] and seperate its computation such that terms for many directions can be precomputed and then later retrieved by look ups. The approximation of P happens in two steps: First we approximate the Fourier Transform by the Discrete Time Fourier Transform (DTFT) and then, afterwards, we approximate the DTFT by the DFT. For further about basics of signal processing and Fourier Transformations please consult the appendix **??**.

[14] Even a fast variant of computation the Fourier Transform has a runtime complexitiy of O(N log N) where N is the number of sample.

[15] See appendix **??** for further information about different kinds of fourier transformations.

Using the insight gained by equation 1.1 allows us to further simplify equation 1.13:

$$\begin{aligned} L_\lambda(\omega_r) &= I(\lambda)\frac{F^2(1+\omega_i\cdot\omega_r)^2}{\lambda^2 Aw^2 cos(\theta_r)}\left\langle\left|P\left(\frac{2\pi u}{\lambda},\frac{2\pi v}{\lambda}\right)\right|^2\right\rangle \\ &= I(\lambda)\frac{F^2(1+\omega_i\cdot\omega_r)^2}{\lambda^2 Aw^2 cos(\theta_r)}\left\langle\left|T^2 P_{dtft}\left(\frac{2\pi u}{\lambda},\frac{2\pi v}{\lambda}\right)\right|^2\right\rangle \end{aligned} \tag{1.14}$$

Where P_{dtft} is a substitude for $\mathcal{F}_{DTFT}\{s\}(w)$. Furthermore T the sampling distance for the discretization of $p(x,y)$ assuming equal and uniform sampling in both dimensions x and y.

1.3.3 Relative Reflectance

In this section we are going to explain how to scale our BRDF formulation such that all of its possible output values are mapped into the range $[0,1]$. Such a relative reflectance formulation will ease our life for later rendering purposes since usually color values are within the range $[0,1]$, too. Furthermore, this will allow us to properly blend the resulting illumination caused by diffraction with a texture map.

Let us examine what $L_\lambda(\omega_r)$ will be for a purely specular surface, for which $\omega_r = \omega_0 = \omega_i$ such that $\omega_0 = (0,0,1)$. For this specular reflection case, the correspinding radiance will be denoted as $L_\lambda^{spec}(\omega_0)$. When we know the expression for $L_\lambda^{spec}(\omega_0)$ we would be able to compute the relative reflected radiance for our problem 1.13 by simply taking the fraction between $L_\lambda(\omega_r)$ and $L_\lambda^{spec}(\omega_0)$ which is denoted by:

$$\rho_\lambda(\omega_i,\omega_r) = \frac{L_\lambda(\omega_r)}{L_\lambda^{spec}(\omega_0)} \tag{1.15}$$

Notice that the third component w from the vector in equation **??** is squared eqaul $(cos(\theta_i)+cos(\theta_r))^2$[16]. But first, let us derive the following expression:

$$\begin{aligned} L_\lambda^{spec}(\omega_0) &= I(\lambda)\frac{F(\omega_0,\omega_0)^2(1+\begin{pmatrix}0\\0\\1\end{pmatrix}\cdot\begin{pmatrix}0\\0\\1\end{pmatrix})^2}{\lambda^2 A(cos(0)+cos(0))^2 cos(0)}\langle\left|T_0^2 P_{dtft}(0,0)\right|^2\rangle \\ &= I(\lambda)\frac{F(\omega_0,\omega_0)^2(1+1)^2}{\lambda^2 A(1+1)^2 1}\left|T_0^2 N_{sample}\right|^2 \\ &= I(\lambda)\frac{F(\omega_0,\omega_0)^2}{\lambda^2 A}\left|T_0^2 N_{sample}\right|^2 \end{aligned} \tag{1.16}$$

Where $N_{samples}$ is the number of samples of the DTFT **??**. Thus, we can plug our last derived expression 1.16 into the definition for the relative reflectance radiance 1.15 in the direction w_r and will get:

[16] Consult section **??** in the appendix

$$
\begin{aligned}
\rho_\lambda(\omega_i,\omega_r) &= \frac{L_\lambda(\omega_r)}{L_\lambda^{spec}(\omega_0)} \\
&= \frac{I(\lambda)\frac{F(\omega_i,\omega_r)^2(1+\omega_i\cdot\omega_r)^2}{\lambda^2 A(cos(\theta_i)+cos(\theta_r))^2 cos(\theta_r)}\langle\left|T_0^2 P_{dtft}(\frac{2\pi u}{\lambda},\frac{2\pi v}{\lambda})\right|^2\rangle}{I(\lambda)\frac{F(\omega_0,\omega_0)^2}{\lambda^2 A}\left|T_0^2 N_{sample}\right|^2} \\
&= \frac{F^2(\omega_i,\omega_r)(1+\omega_i\cdot\omega_r)^2}{F^2(\omega_0,\omega_0)(cos(\theta_i)+cos(\theta_r))^2 cos(\theta_r)}\langle\left|\frac{P_{dtft}(\frac{2\pi u}{\lambda},\frac{2\pi v}{\lambda})}{N_{samples}}\right|^2\rangle
\end{aligned} \tag{1.17}
$$

For simplification and better readability, let us introduce the following expression, the so called gain-factor:

$$
C(\omega_i,\omega_r) = \frac{F^2(\omega_i,\omega_r)(1+\omega_i\cdot\omega_r)^2}{F^2(\omega_0,\omega_0)(cos(\theta_i)+cos(\theta_r))^2 cos(\theta_r) N_{samples}^2} \tag{1.18}
$$

Using equation 1.18, we will get the following expression for the relative reflectance radiance from equation 1.17:

$$
\rho_\lambda(\omega_i,\omega_r) = C(\omega_i,\omega_r)\langle\left|P_{dtft}(\frac{2\pi u}{\lambda},\frac{2\pi v}{\lambda})\right|^2\rangle \tag{1.19}
$$

Using the previous definition for the relative reflectance radiance equation 1.15:

$$
\rho_\lambda(\omega_i,\omega_r) = \frac{L_\lambda(\omega_r)}{L_\lambda^{spec}(\omega_0)} \tag{1.20}
$$

Which we can rearrange to the expression:

$$
L_\lambda(\omega_r) = \rho_\lambda(\omega_i,\omega_r) L_\lambda^{spec}(\omega_0) \tag{1.21}
$$

Let us choose $L_\lambda^{spec}(w_0) = S(\lambda)$ such that is has the same profile as the relative spectral power distribution of CIE Standard Illuminant $D65$ discussed in **??**. Furthermore, when integrating over λ for a specular surface, we should get CIE_{XYZ} values corresponding to the white point for $D65$. The corresponding tristimulus values using CIE colormatching functions **??** for the CIE_{XYZ} values look like:

$$
\begin{aligned}
X &= \int_\lambda L_\lambda(\omega_r)\overline{x}(\lambda)d\lambda \\
Y &= \int_\lambda L_\lambda(\omega_r)\overline{y}(\lambda)d\lambda \\
Z &= \int_\lambda L_\lambda(\omega_r)\overline{z}(\lambda)d\lambda
\end{aligned} \tag{1.22}
$$

where $\overline{x}$, $\overline{y}$, $\overline{z}$ are the color matching functions. Combining our last finding from equation 1.21 for $L_\lambda(\omega_r)$ with the definition of the tristimulus values from equation 1.22, allows us to derive a formula for computing the colors values using Stam's BRDF formula relying on the rendering equation 1.9. Without any loss of generality it suffices to derive an explicit expression for just one tristimulus term, for example Y, the luminance:

$$\begin{aligned}
Y &= \int_\lambda L_\lambda(\omega_r)\overline{y}(\lambda)d\lambda \\
&= \int_\lambda \rho_\lambda(\omega_i,\omega_r)L_\lambda^{spec}(\omega_0)\overline{y}(\lambda)d\lambda \\
&= \int_\lambda \rho_\lambda(\omega_i,\omega_r)S(\lambda)\overline{y}(\lambda)d\lambda \\
&= \int_\lambda C(\omega_i,\omega_r)\langle\left|P_{dtft}(\frac{2\pi u}{\lambda},\frac{2\pi v}{\lambda})\right|^2\rangle S(\lambda)\overline{y}(\lambda)d\lambda \\
&= C(\omega_i,\omega_r)\int_\lambda \langle\left|P_{dtft}(\frac{2\pi u}{\lambda},\frac{2\pi v}{\lambda})\right|^2\rangle S(\lambda)\overline{y}(\lambda)d\lambda \\
&= C(\omega_i,\omega_r)\int_\lambda \langle\left|P_{dtft}(\frac{2\pi u}{\lambda},\frac{2\pi v}{\lambda})\right|^2\rangle S_y(\lambda)d\lambda \qquad (1.23)
\end{aligned}$$

Where we used the definition $S_y(\lambda)\overline{y}(\lambda)$ in the last step.

1.4 Optimization using Taylor Series

Our final goal is to render stractural colors resulting by the effect of wave diffraction. So far, we have derived an expression which can be used for rendering. Nevertthesless, our current equation 1.23 used for computing structural colors, cannot directly be used for interactive rendering, since P_{dtft} had do be recomputed for every change in any direction[17].

In this section, we will address this issue and deliver an approximation for P_{dtft} defined in equation 1.2. This approximation will allow us to seperate P_{dtft} in a certain way such that some computional expsensive terms can be precomputed. The main idea is to formulate P_{dtft} as a series expansion relying on the definition of Taylor Series, like defined in equation **??**. Further, we will provide an error bound for our approximation approach for a given number of terms. Last, we will plug our finding extend our current BRDF formula from equation 1.23 by the findings derived within this section.

Let us consider $p(x,y) = e^{ikwh(x,y)}$ form Stam's Paper **??** where $h(x,y)$ is a given height field and $k = \frac{2\pi}{\lambda}$ denotes the wavenumber of wavelenght λ. For any complex number t the power series expansion of the exponential function is equal to:

$$e^t = 1 + t + \frac{t^2}{2!} + \frac{t^3}{3!} + ... = \sum_{n=0}^{\infty}\frac{t^n}{n!} \qquad (1.24)$$

Now, when we use the exponent[18] of $p(x,y)$ as an input argument for equation 1.24 we get:

[17] According to changes in viewing- or incident light direction.

[18] This exponent is a complex valued function, equal to ikwh(x,y).

$$\begin{aligned} e^t &= e^{ikwh} \\ &= 1 + (ikwh) + \frac{1}{2!}(ikwh)^2 + \frac{1}{3!}(ikwh)^3 + ... \\ &= \sum_{n=0}^{\infty} \frac{(ikwh)^n}{n!}. \end{aligned} \tag{1.25}$$

where i is the imaginary unit for complex numbers. For simplification, in the reminder of this section we omitted the arguments of h. Equation 1.25 gives us an expression for an exponential series expansion for an for the exponent of $p(x,y)$. Please note that above's taylor series is convergent for any complex valued number. Therefore the equation 1.25 is equal to

$$p(x,y) = \sum_{n=0}^{\infty} \frac{(ikwh(x,y))^n}{n!} \tag{1.26}$$

and thus gives us a series representation of $p(x,y)$. Next, calculating the Fourier Transformation $\mathcal{F}$ of equation 1.26 gives us the identity:

$$\begin{aligned} \mathcal{F}\{p\} &\equiv \mathcal{F}\left\{\sum_{n=0}^{\infty} \frac{(ikwh)^n}{n!}.\right\} \\ &\equiv \sum_{n=0}^{\infty} \mathcal{F}\left\{\frac{(ikwh)^n}{n!}\right\} \\ &\equiv \sum_{n=0}^{\infty} \frac{(ikw)^n}{n!} \mathcal{F}\{h^n\} \end{aligned} \tag{1.27}$$

Where we have exploited the fact that the Fourier Transformation is a linear operator. Therefore, in equation 1.27, we have shown that the Fourier Transformation of a series is equal to the sum of the Fourier Transformation, applied on each individual series term. Reusing the identifier P[19] in order to determin the Fourier Transformation of Stams definition p from equation **??**, equation 1.27 then correspond to:

$$P(\alpha,\beta) = \sum_{n=0}^{\infty} \frac{(ikw)^n}{n!} \mathcal{F}\{h^n\}(\alpha,\beta) \tag{1.28}$$

Up to now we have found a infinity series representation for P_{dtft}. Next we are going to look for an upper bound $N \in \mathbb{N}$ such that

$$\tilde{P}_N(\alpha,\beta) := \sum_{n=0}^{N} \frac{(ikwh)^n}{n!} \mathcal{F}\{h^n\}(\alpha,\beta) \approx P(\alpha,\beta) \tag{1.29}$$

$\tilde{P}_N$ is a good approximation of P, i.e. their absolute difference is small[20]. But first, the following two facts would have to be proven[21]:

1. Show that there exist such an $N \in \mathbb{N}$ s.t. the approximation of equation 1.29 holds true.

[19] This identifier P may be subscripted by *dtft* which will denote the DTFT variant of P.

[20] Mathematically speaking, this statement correspond to $||\tilde{P}_N - P|| \leq \epsilon$, where $\epsilon > 0$ is a small number.

[21] Please have a look in section **??** in the appendix

2. Find a value for N s.t. this approximation is below a certain error bound, e.g. close to machine precision ϵ.

Assuming these facts are proven and there actually exists such an N, we can make use of the taylor series approximation from equation 1.29 and use it for approximating P_{dtft}. This idea allows us to adapt equation 1.23, which is used for computing the structural colors of our BRDF model, in nummerically fast way. E.g. the equation for the luminance is then be equal:

$$\begin{aligned} Y &= C(w_i, w_r) \int_\lambda \langle \left| P_{dtft}(\frac{2\pi u}{\lambda}, \frac{2\pi v}{\lambda}) \right|^2 \rangle S_y(\lambda) d\lambda \\ &= C(w_i, w_r) \int_\lambda \left| \sum_{n=0}^{N} \frac{(wk)^n}{n!} \mathcal{F}\{i^n h^n\}(\frac{2\pi u}{\lambda}, \frac{2\pi v}{\lambda}) \right|^2 S_y(\lambda) d\lambda \qquad (1.30) \end{aligned}$$

Notice that equation 1.30 is contrainted by N and hence is an approximation of equation 1.23. Furthermore, it is possible to seperate out all the Fourier Terms in the summation and precompute them. This is why the approach in equation 1.30 is fast in order to compute structural color values according to our BRDF model.

1.5 Spectral Rendering

As the last step of our series of derivations, we plug all our findings together to one big equation. in order to compute the color for each pixel on our mesh in the CIE_{XYZ} colorspace. For any given heigh-field $h(x, y)$ representing a grating of a nano structure and for given direction vectors w_i and w_r as shown in figure **??** the resulting color caused by the effect of diffraction can be computed the following way:

$$DFT_n\{h\}(u, v) = F_{dft}\{i^n h^n\}(\frac{2\pi u}{\lambda}, \frac{2\pi v}{\lambda}) \qquad (1.31)$$

$$W_n(u, v) = \sum_{(r,s) \in \mathcal{N}_1(u,v)} |DFT_n\{h\}(u - w_r, v - w_s)|^2 \phi(u - w_r, v - w_s) \qquad (1.32)$$

where $\phi(x, y) = \pi e^{-\frac{x^2+y^2}{2\sigma_f^2}}$ is the Gaussian window from equation 1.2.3 and $\mathcal{N}_1(u, v)$ denotes the one-neighborhood around (u, v). Then $\forall(u, v, w)$ like defined in equation **??**, our final expression for computing structural colors due to diffraction, using all our previous derivations, will be equal:

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = C(\omega_i, \omega_r) \int_\Lambda \sum_{n=0}^{N} \frac{(2\pi w)^n}{\lambda^n n!} W_n(u, v) \begin{pmatrix} S_x(\lambda) \\ S_y(\lambda) \\ S_z(\lambda) \end{pmatrix} d\lambda \qquad (1.33)$$

1.6 Alternative Approach

1.6.1 PQ factors

In this section we are presenting an alternative approach to the previous Gaussian window approach 1.2.3 in order to solve the issue working with $DTFT$ instead the DFT. We assume, that a given surface S is covered by a number of replicas of a provided representative surface patch f. In a simplified, one dimensional scenario, mathematically speaking, f is assumed to be a repetative function, i.e. $\forall x \in \mathbb{R} : S(x) = S(x + nT)$, where T is its period and $n \in \mathbb{N}_0$. Thus, the surfaces can be written formally as:

$$S(x) = \sum_{n=0}^{N} f(x + nT) \tag{1.34}$$

What we are looking for is an identity for the Fourier Transform[22] of our surface S, required in order to simplify the (X, Y, Z) colors from 1.30:

$$\begin{aligned}\mathcal{F}\{S\}(w) &= \int f(x)e^{iwx}dx \\ &= \int_{-\infty}^{\infty} \sum_{n=0}^{N} f(x + nT)e^{iwx}dx \\ &= \sum_{n=0}^{N} \int_{-\infty}^{\infty} f(x + nT)e^{iwx}dx\end{aligned} \tag{1.35}$$

Next, apply the following substitution $x + nT = y$ which will lead us to:

$$\begin{aligned}x &= y - nT \\ dx &= dy\end{aligned} \tag{1.36}$$

Plugging this substitution back into equation 1.35 we will get:

$$\begin{aligned}\mathcal{F}\{S\}(w) &= \sum_{n=0}^{N} \int_{-\infty}^{\infty} f(x + nT)e^{iwx}dx \\ &= \sum_{n=0}^{N} \int_{-\infty}^{\infty} f(y)e^{iw(y-nT)}dy \\ &= \sum_{n=0}^{N} e^{-iwnT} \int_{-\infty}^{\infty} f(y)e^{iwy}dy \\ &= \sum_{n=0}^{N} e^{-iwnT} \mathcal{F}\{f\}(w) \\ &= \mathcal{F}\{f\}(w) \sum_{n=0}^{N} e^{-iwnT}\end{aligned} \tag{1.37}$$

We used the fact that the exponential term e^{-iwnT} is a constant factor when integrating along dy and the identity for the Fourier Transform of the function f. Next, let us examine the series $\sum_{n=0}^{N} e^{-iwnT}$ closer:

[22] Remember that we are using the definiton of Fourier Transform used in electrical engineering where $\mathcal{F}$ actually corresponds to the inverse Fourier Transform.

$$\begin{aligned} \sum_{n=0}^{N} e^{-uwnT} &= \sum_{n=0}^{N} (e^{-iwT})^{n} \\ &= \frac{1 - e^{iwT(N+1)}}{1 - e^{-iwT}} \end{aligned} \tag{1.38}$$

We recognize the geometric series identity for the left-hand-side of equation 1.38. Mainly relying on trigonometric identities, equation 1.37 can further simplified to:

$$\mathcal{F}\{S\}(w) = (p + iq)\mathcal{F}\{f\}(w) \tag{1.39}$$

where p and q are defined like:

$$\begin{aligned} p &= \frac{1}{2} + \frac{1}{2}\left(\frac{cos(wTN) - cos(wT(N+1))}{1 - cos(wT)}\right) \\ q &= \frac{sin(wT(N+1)) - sin(wTN) - sin(wT)}{2(1 - cos(wT))} \end{aligned} \tag{1.40}$$

Please notice, all derivation steps can be found in the appendix in section **??**.

Now lets consider our actual problem description. Given a patch of a nanoscaled sureface snake shed represented as a two dimensional heightfield $h(x, y)$. We once again assume that this provided patch is representing the whole surface S of our geometry by some number of replicas of itself. Therefore, $S(x, y) = \sum_{n=0}^{N} h(x + nT_1, y + mT_2)$, assuming the given height field has the dimensions T_1 by T_2. In order to derive an identity for the two dimensional Fourier transformation of S we can similarly proceed like we did to derive equation 1.39.

$$\mathcal{F}\{S\}(w_1, w_2) = (p + iq)\mathcal{F}_{DTFT}\{h\}(w_1, w_2) \tag{1.41}$$

Note that a detailed derivation of equation 1.41 can be found in the appendix in section **??** and we have defined :

$$\begin{aligned} p &:= (p_1 p_2 - q_1 q_2) \\ q &:= (p_1 p_2 + q_1 q_2) \end{aligned} \tag{1.42}$$

For the identity of equation 1.41 we made use of Green's integration rule which allowed us to split the double integral to the product of two single integrations. Also, we used the definition of the 2-dimensional inverse Fourier transform of the height field function. We applied a similar substitution like we did in 1.36, but this time twice, once for x_1 and once for x_2 separately. The last step in equation 1.41, substituting with p and q in equation **??** will be useful later in the implementation. The insight should be, that the product of two complex numbers is again a complex number. We will have to compute the absolute value of $\mathcal{F}\{S\}(w_1, w_2)$ which will then be equal $(p^2 + q^2)^{\frac{1}{2}} \, |\mathcal{F}\{h\}(w_1, w_2)|$

1.6.2 Interpolation

In section 1.6.1 we derived an alternative approach to the gaussian window approach described in section 1.2.3 in order to approximate our height field. We assume that our height field is a superpostion of periodically aligned substructered (i.e. finger structures). This so called PQ approach allows us to integrate over one period of a substructre in our height field, instead iterating over the whole domain. Nevertheless, this main finding, discribed in equation 1.41, is using the DTFT. Thus, since our original height field is supposed to be a continuous-time bandlimited function we can reconstruct it by applying a sinc-interpolation.

In general, for a sinc-interpolation, we are interested in recovering an original analog signal $x(t)$ from its samples. Therfore, for a given sequence of real numbers $x[n]$, representing a digital signal, its correspond continuous function is:

$$x(t) = \sum_{n=-\infty}^{\infty} x[n] sinc\left(\frac{t - nT}{T}\right) \tag{1.43}$$

which has the Fourier transformation $X(f)$ whose non-zero values are confined to the region $|f| \leq \frac{1}{2T} = B$. When $x[n]$ represents time samples at interval T of a continuous function, then the quantity $f_s = \frac{1}{T}$ is known as its sample rate and $\frac{f_s}{2}$ denotes the Nyquist frequency. The sampling Theorem states that when a function has a Bandlimit B less than the Nyquist frequency, then $x(t)$ is a perfect reconstruction of the original function. Figure illustates a reconstruction of a 1d signal relying on a sinc-interpolation.

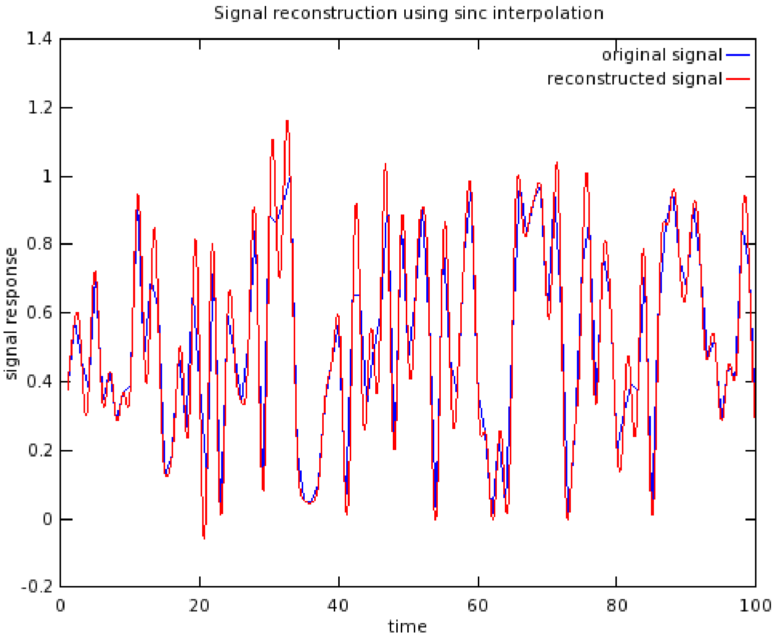


Figure 1.6: Comparission between a given random one dimensional input signal $s(t)$ and its sinc interpolation $\hat{s}(t)$. Notice that for the interpolation there were $N = 100$ samples from the original signal provided.

List of Tables

List of Figures

List of Algorithms

Bibliography

[Bar07] BARTSCH, Hans-Jochen: *Taschenbuch Mathematischer Formeln.* 21th edition. HASNER, 2007. – ISBN 978–3–8348–1232–2

[CT12] CUYPERS T., et a.: Reflectance Model for Diffraction. In: *ACM Trans. Graph. 31, 5* (2012), September

[DD14] D.S. DHILLON, et a.: Interactive Diffraction from Biological Nanostructures. In: *EUROGRAPHICS 2014/ M. Paulin and C. Dachsbacher* (2014), January

[D.S14] D.S.DHILLON, M.Single I.Gaponenko M.C. Milinkovitch M. J.Teyssier: Interactive Diffraction from Biological Nanostructures. In: *Submitted at Computer Graphics Forum* (2014)

[For11] FORSTER, Otto: *Analysis 3.* 6th edition. VIEWEG+TEUBNER, 2011. – ISBN 978–3–8348–1232–2

[I.N14] I.NEWTON: *Opticks, reprinted.* CreateSpace Independent Publishing Platform, 2014. – ISBN 978–1499151312

[JG04] JUAN GUARDADO, NVIDIA: Simulating Diffraction. In: *GPU Gems* (2004). `https://developer.nvidia.com/content/gpu-gems-chapter-8-simulating-diffraction`

[LM95] LEONARD MANDEL, Emil W.: *Optical Coherence and Quantum Optics.* Cambridge University Press, 1995. – ISBN 978–0521417112

[MT10] MATIN T.R., et a.: Correlating Nanostructures with Function: Structurral Colors on the Wings of a Malaysian Bee. (2010), August

[PAT09] PAUL A. TIPLER, Gene M.: *Physik für Wissenschaftler und Ingenieure.* 6th edition. Spektrum Verlag, 2009. – ISBN 978–3–8274–1945–3

[PS09] P. SHIRLEY, S. M.: *Fundamentals of Computer Graphics.* 3rd edition. A K Peters, Ltd, 2009. – ISBN 978–1–56881–469–8

[R.H12] R.HOOKE: *Micrographia, reprinted.* CreateSpace Independent Publishing Platform, 2012. – ISBN 978–1470079031

[RW11] R. WRIGHT, et a.: *OpenGL SuperBible.* 5th edition. Addison-Wesley, 2011. – ISBN 978–0–32–171261–5

[Sta99] STAM, J.: Diffraction Shaders. In: *SIGGRPAH 99 Conference Proceedings* (1999), August

[T.Y07] T.YOUNG: *A course of lectures on natural philosophy and the mechanical arts Volume 1 and 2.* Johnson, 1807, 1807