

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 nanostructures of their 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 dependent on 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 directly. Furthermore, we describe a method for interactive rendering of diffraction effects due to interaction of light with biological nanostructures such as those on snake skins. Our method uses discrete height fields of natural gratings acquired by using atomic force microscopy (AFM) as an input and employs Fourier Optics to simulate far-field diffraction. Based on a 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 nanostructures of two snake species, namely the *Elaphe* and the *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

1	Derivations	1
1.1	Problem Statement and Challenges	1
1.2	Approximate a FT by a DFT	2
1.2.1	Reproduce FT by DTFT	2
1.2.2	Spatial Coherence and Windowing	4
1.2.3	Reproduce DTFT by DFT	6
1.3	Adaption of Stam's BRDF for Discrete Height Fields	8
1.3.1	Reflected Radiance of Stam's BRDF	8
1.3.2	Relative Reflectance	9
1.4	Optimization using Taylor Series	11
1.5	Spectral Rendering using DFTs	13
1.6	An Alternative Approach	14
1.6.1	PQ factors	14
1.6.2	Sinc Interpolation in Frequency Domain	16
2	Evaluation and Data Acquisition	18
2.1	Diffraction Gratings	18
2.2	Data Acquisition	22
2.3	Verifications	23
2.3.1	Numerical Comparisons	24
2.3.2	Virtual Testbench	25
3	Results	28
3.1	BRDF maps	28
3.2	Rendering Surface Geometries	41
A	Evaluation Data Generation	50
B	Rendering Surface Geometries using the NMM approach	52
	Bibliography	55

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 color production as shown in figure 1.2, which we can see whenever light is diffracted from a natural grating. For this purpose we need the following input data (see figure 1.1):

- A mesh representing a snake surface¹ as shown in figure 1.1(a).
- A natural diffraction grating represented as a height field, its maximum height and its pixel width².
- A vector field which describes how the given nanostructure patch is oriented on the surface (see figure 1.1(c)).

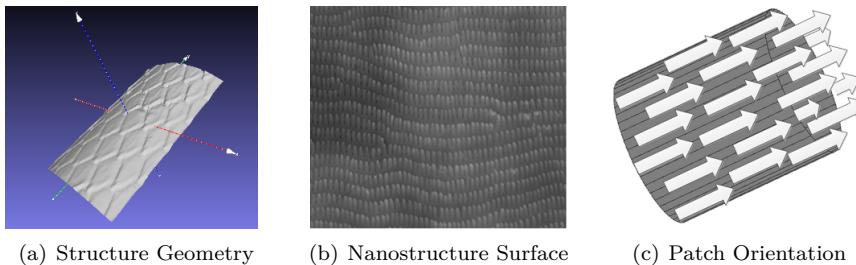


Figure 1.1: Input for our simulation

We want to rely on the integral equation ?? derived by J. Stam in his paper [Sta99] about diffraction shaders. This equation represents a BRDF which models the effect of diffraction caused when a beam of light hits a grating structure. His BRDF is formulated as a function of the Fourier transform of certain correlation functions relating to the given height field. It assumes that the structure of a given grating (i.e. the height field) exhibits a certain degree of regularity. This

¹In our simulation it is an actual reconstruction of a real snake skin. These measurements are provided by the Laboratory of Artificial and Natural Evolution at Geneva. See their website: www.lanevol.org.

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

homogeneity assumption of the structure enables him to accurately estimate the correlation function relying on statistical properties of the given height field. However, modelling the complexity of a biological nanostructure sufficiently and accurately by relying on statistical methods is a non-trivial task. 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 at interactive rates.

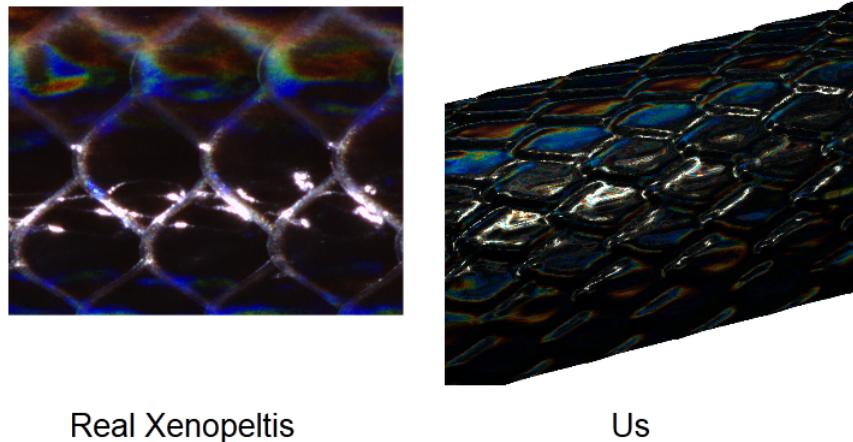


Figure 1.2: Output: Rendered Structural Colors

1.2 Approximate a FT by a DFT

Before we will start with our actual adaption of Stam's BRDF model, we first derive some useful identities which are later used during our main derivation in section 1.3. The goal of this section is to provide an approximation of a FT using the DFT when dealing with a bandlimited signal. For this purpose we will exploit the concept of windowing a function and spatial coherence as described in section 1.2.2.

1.2.1 Reproduce FT by DTFT

In the following sections, we derive an identity to approximate the FT by the DTFT. This identity will be used to derive the reflected spectral radiance $L_\lambda(\omega_r)$ when using Stam's BRDF for a given input height field. However, its expression, which is stated in equation 1.10 requires an evaluation of the Fourier transform applied on a given height field³ along every possible viewing and light direction. Figure 1.3 visualized the idea of how to obtain the DTFT from the FT for a one dimensional signal⁴

³actually it requires the computation of the inverse Fourier Transform of a transformed version of the given height field, the function $p(x,y)$ defined in equation ??.

⁴For 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.

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

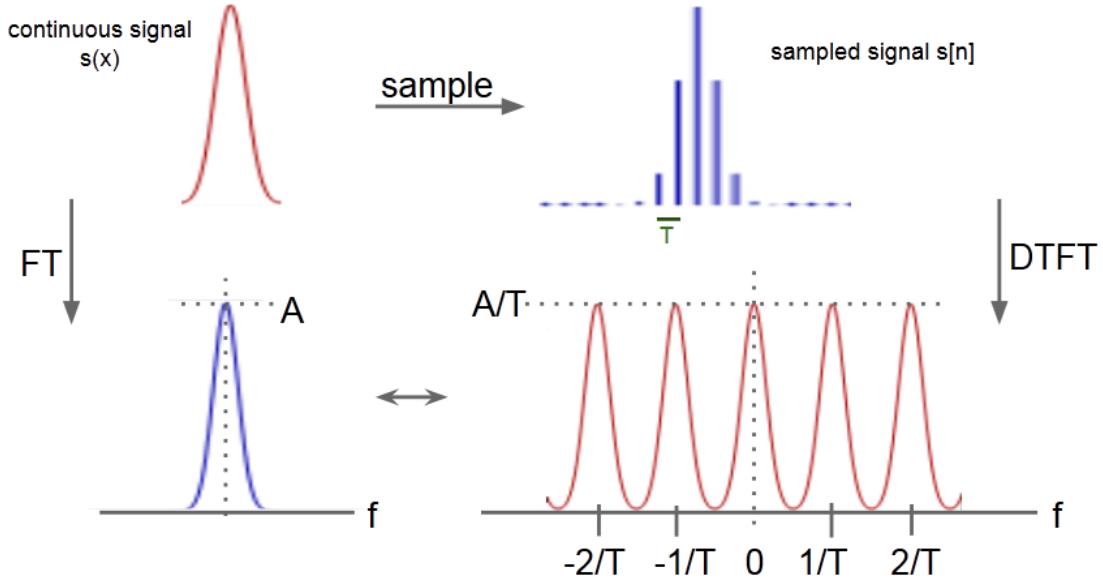


Figure 1.3: Illustration of how to approximate the analytical Fourier Transform (FT)⁵ of a given continuous signal by a Discrete Time Fourier Transform (DTFT). The DTFT applied on a band-limited, discretized signal yields a continuous, periodic response in frequency space.

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 original signal when it has a limited bandwidth. 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 transformation operator on the discretized signal \hat{s} which gives us the following expression:

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

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

⁶n denotes the number of samples.

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 remainder of this thesis we introduce the following definition:

$$P_{dtft} \equiv \mathcal{F}_{DTFT}\{p\} \quad (1.2)$$

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. Previously we have seen that wave-theory 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 interference is limited by the quantity coherence area. For filtered sunlight on earth this is equal to $65\mu m$ ⁸.

⁷By 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 ??.

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

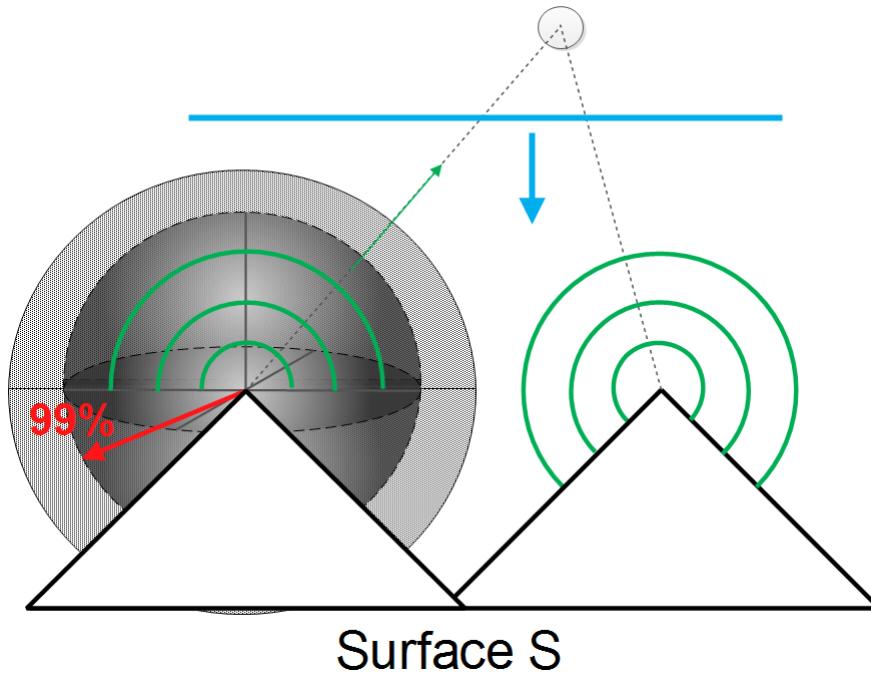


Figure 1.4: A plane wave encounters a surface. According to Huygens principle, secondary wavelets are emitted off from this surface. The resulting wave at a certain point in space (here indicated by a gray circle) depends on the interference among all waves encountering at this position. The amount of 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 Huygen'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 modelled by a Gaussian Random Process. For any such Gaussian Process we can use a spatial gaussian window $g(x)$ which is equal:

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

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 to saying that 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

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

in frequency space as:

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

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 approach.

1.2.3 Reproduce DTFT by DFT

In this section we explain how and under what assumptions the DTFT of a discretized signal¹⁰ can be approximated by a DFT. The whole idea how to reproduce the DTFT by DFT is schematically illustrated in figure 1.5.

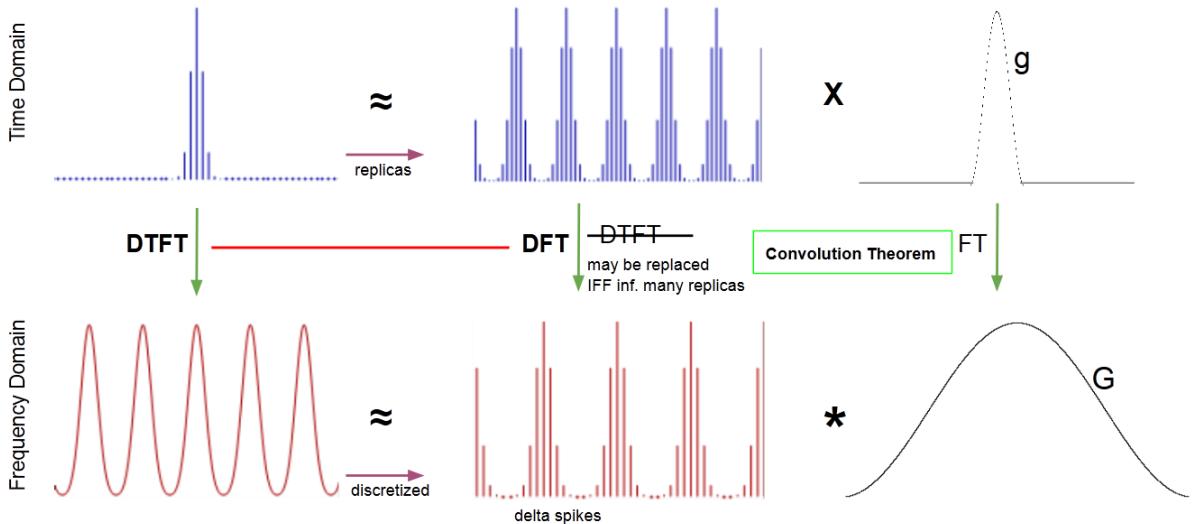


Figure 1.5: Illustration of how to approximate the DTFT¹¹ by the DFT relying on the Convolution Theorem, using a gaussian window function.

Lets say, we are given a spatial, band-limited and discretized one dimensional signal \hat{s} . Our goal is to approximate this spatial signal in a way such that when taking the DFT of this approximated signal, it will yield the same response as taking the DTFT of the original sampled \hat{s} . For this purpose we will use the previously introduced concept of gaussian windows and the so called Convolution Theorem which is a fundamental property of all Fourier transformations.

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:

$$\mathcal{F}\{f \cdot g\} = \mathcal{F}\{f\} * \mathcal{F}\{g\} \quad (1.5)$$

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

¹¹Images 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 principal issue is how to approximate our given signal \hat{s} . Therefore, let us consider another signal s_N^* which is the N times replicated version of \hat{s} (blue signal at center top in figure).

Remember that, in general, the wave magnitude at a certain point in space is the result of interference among all wavelength meeting at that position. In our scenario, the source of those signals are emitted secondary wavelets. The interference strength between wavelets is related to their spatial coherence. Windowing the signal by a gaussian window g is akin to modelling the effects of spatial coherence interference on the surface. 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 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 band-limited. 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 s_N^* and g is the same as convolving the DTFT of 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 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 s_N^* .

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

In general, for every non-windowed signal, we cannot compute its DTFT ?? numerically due to finite computer arithmetic and hence working with the DFT is our only option. Furthermore, there are numerically fast algorithms 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. Now, let a spatial gaussian window g having a standard deviation for which $4\sigma_s$ is equal $65\mu m$. Then, from before, it follows:

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

Therefore we can deduce the following expression:

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

where

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

1.3 Adaption of Stam's BRDF for Discrete Height Fields

Our goal is to render structural colors using Stam's BRDF¹² using natural gratings. According to our problem statement stated in section 1.1 we have to reformulate his BRDF equation. We will end up with an identity of the reflected spectral radiance. This is where the rendering equation from section ?? comes into play. Having looked at some basic derivations from section 1.2 we start our adaption of Stam's BRDF model in the following sections.

1.3.1 Reflected Radiance of Stam's BRDF

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

$$\begin{aligned}
BRDF(\omega_i, \omega_r) &= \frac{k^2 F^2 G}{4\pi^2 A w^2} \langle |P(ku, kv)|^2 \rangle \\
&= \frac{4\pi^2 F^2 G}{4\pi^2 A \lambda^2 w^2} \langle |P(ku, kv)|^2 \rangle \\
&= \frac{F^2 G}{A \lambda^2 w^2} \left\langle \left| P\left(\frac{2\pi u}{\lambda}, \frac{2\pi v}{\lambda}\right) \right|^2 \right\rangle
\end{aligned} \tag{1.9}$$

Going back to equation ?? and plugging equation 1.9 into it, using the definition of equation ?? and the equation ?? for ω we will get the following:

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

$$\begin{aligned}
L_\lambda(\omega_r) &= \frac{F^2(1 + \omega_i \cdot \omega_r)^2}{A\lambda^2 \cos(\theta_i) \cos(\theta_r) \omega^2} \left\langle \left| P\left(\frac{2\pi u}{\lambda}, \frac{2\pi v}{\lambda}\right) \right|^2 \right\rangle \cos(\theta_i) I(\lambda) \\
&= I(\lambda) \frac{F^2(1 + \omega_i \cdot \omega_r)^2}{\lambda^2 A \omega^2 \cos(\theta_r)} \left\langle \left| P\left(\frac{2\pi u}{\lambda}, \frac{2\pi v}{\lambda}\right) \right|^2 \right\rangle
\end{aligned} \tag{1.10}$$

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 arguments.

So far we just plugged Stam's BRDF identity into the rendering equation and hence have not deviated from his formulation. Keep in mind that P denotes the Fourier transformation of a given discrete height field. Its FT 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 lot¹³. According to section 1.2.1, when applying the Fourier transformation to a discretized function, we have to apply the DTFT operator. Thus when interested in computing the Fourier transformation of a given discrete height field, we have to apply the DTFT operator to it. One particular strategy to solve this issue is to approximate P by the Discrete Fourier Transform (DFT)¹⁴ and separate its computation such that terms for many directions can be precomputed and then later retrieved by look ups. For the approximation of P we rely on the derivations from section 1.2. Firstly, we approximated the FT by DTFT and then we approximated the DTFT by the DFT. For the basics of signal processing and Fourier transformations please refer to the appendix ??.

Using the insight gained by equation 1.1 allows us to further simplify equation 1.10:

$$\begin{aligned}
L_\lambda(\omega_r) &= I(\lambda) \frac{F^2(1 + \omega_i \cdot \omega_r)^2}{\lambda^2 A \omega^2 \cos(\theta_r)} \left\langle \left| P\left(\frac{2\pi u}{\lambda}, \frac{2\pi v}{\lambda}\right) \right|^2 \right\rangle \\
&= I(\lambda) \frac{F^2(1 + \omega_i \cdot \omega_r)^2}{\lambda^2 A \omega^2 \cos(\theta_r)} \left\langle \left| T^2 P_{dtft}\left(\frac{2\pi u}{\lambda}, \frac{2\pi v}{\lambda}\right) \right|^2 \right\rangle
\end{aligned} \tag{1.11}$$

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

1.3.2 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 corresponding radiance will be denoted

¹³Even a fast variant of computation for the two dimensional Fourier transform has a runtime complexity of $O(N^2 \log N)$ for a N by N discretized signal.

¹⁴See appendix ?? for further information about different kinds of Fourier transformations.

as $L_\lambda^{spec}(\omega_0)$. We define the relative reflected radiance for our problem 1.10 by simply taking the fraction between $L_\lambda(\omega_r)$ and $L_\lambda^{spec}(\omega_0)$ which is denoted by:

$$\rho_\lambda(\omega_i, \omega_r) = \frac{L_\lambda(\omega_r)}{L_\lambda^{spec}(\omega_0)} \quad (1.12)$$

Notice that the third component w from the vector in equation ?? is squared equal to $(\cos(\theta_i) + \cos(\theta_r))^2$ ¹⁵. But first, let us derive the following expression:

$$\begin{aligned} L_\lambda^{spec}(\omega_0) &= I(\lambda) \frac{F(\omega_0, \omega_0)^2 (1 + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix})^2}{\lambda^2 A (\cos(0) + \cos(0))^2 \cos(0)} \langle \left| T_0^2 P_{dtft}(0, 0) \right|^2 \rangle \\ &= I(\lambda) \frac{F(\omega_0, \omega_0)^2 (1+1)^2}{\lambda^2 A (1+1)^2 1} \left| T_0^2 N_{sample} \right|^2 \\ &= I(\lambda) \frac{F(\omega_0, \omega_0)^2}{\lambda^2 A} \left| T_0^2 N_{sample} \right|^2 \end{aligned} \quad (1.13)$$

Where $N_{samples}$ is the number of samples of the DTFT ?. Thus, we can plug our last derived expression 1.13 into the definition for the relative reflectance radiance 1.12 in the direction ω_r and we get:

$$\begin{aligned} \rho_\lambda(\omega_i, \omega_r) &= \frac{L_\lambda(\omega_r)}{L_\lambda^{spec}(\omega_0)} \\ &= \frac{I(\lambda) \frac{F(\omega_i, \omega_r)^2 (1 + \omega_i \cdot \omega_r)^2}{\lambda^2 A (\cos(\theta_i) + \cos(\theta_r))^2 \cos(\theta_r)} \langle \left| T_0^2 P_{dtft}\left(\frac{2\pi u}{\lambda}, \frac{2\pi v}{\lambda}\right) \right|^2 \rangle}{I(\lambda) \frac{F(\omega_0, \omega_0)^2}{\lambda^2 A} \left| T_0^2 N_{sample} \right|^2} \\ &= \frac{F^2(\omega_i, \omega_r) (1 + \omega_i \cdot \omega_r)^2}{F^2(\omega_0, \omega_0) (\cos(\theta_i) + \cos(\theta_r))^2 \cos(\theta_r)} \langle \left| \frac{P_{dtft}\left(\frac{2\pi u}{\lambda}, \frac{2\pi v}{\lambda}\right)}{N_{sample}} \right|^2 \rangle \end{aligned} \quad (1.14)$$

For simplification and better readability, let us define the following gain-factor:

$$C(\omega_i, \omega_r) = \frac{F^2(\omega_i, \omega_r) (1 + \omega_i \cdot \omega_r)^2}{F^2(\omega_0, \omega_0) (\cos(\theta_i) + \cos(\theta_r))^2 \cos(\theta_r) N_{sample}^2} \quad (1.15)$$

Using equation 1.15, we get the following expression for the relative reflectance radiance from equation 1.14:

$$\rho_\lambda(\omega_i, \omega_r) = C(\omega_i, \omega_r) \langle \left| P_{dtft}\left(\frac{2\pi u}{\lambda}, \frac{2\pi v}{\lambda}\right) \right|^2 \rangle \quad (1.16)$$

Using the previous definition for the relative reflectance radiance equation 1.12:

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

¹⁵Consult section ?? in the appendix

which we can rearrange to the expression:

$$L_\lambda(\omega_r) = \rho_\lambda(\omega_i, \omega_r) L_\lambda^{spec}(\omega_0) \quad (1.17)$$

Let us choose $L_\lambda^{spec}(\omega_0) = S(\lambda)$ such that it 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:

$$\begin{aligned} X &= \int_\lambda L_\lambda(\omega_r) \bar{x}(\lambda) d\lambda \\ Y &= \int_\lambda L_\lambda(\omega_r) \bar{y}(\lambda) d\lambda \\ Z &= \int_\lambda L_\lambda(\omega_r) \bar{z}(\lambda) d\lambda \end{aligned} \quad (1.18)$$

where $\bar{x}, \bar{y}, \bar{z}$ are the color matching functions. Combining our last finding from equation 1.17 for $L_\lambda(\omega_r)$ with the definition of the tristimulus values from equation 1.18, allows us to derive a formula for computing the colors values using Stam's BRDF formula relying on the rendering equation ???. Without any loss of generality it suffices to derive an explicit expression for just one tristimulus term, for example Y , the luminance:

$$\begin{aligned} Y &= \int_\lambda L_\lambda(\omega_r) \bar{y}(\lambda) d\lambda \\ &= \int_\lambda \rho_\lambda(\omega_i, \omega_r) L_\lambda^{spec}(\omega_0) \bar{y}(\lambda) d\lambda \\ &= \int_\lambda \rho_\lambda(\omega_i, \omega_r) S(\lambda) \bar{y}(\lambda) d\lambda \\ &= \int_\lambda C(\omega_i, \omega_r) \left\langle \left| P_{dtft} \left(\frac{2\pi u}{\lambda}, \frac{2\pi v}{\lambda} \right) \right|^2 \right\rangle S(\lambda) \bar{y}(\lambda) d\lambda \\ &= C(\omega_i, \omega_r) \int_\lambda \left\langle \left| P_{dtft} \left(\frac{2\pi u}{\lambda}, \frac{2\pi v}{\lambda} \right) \right|^2 \right\rangle S(\lambda) \bar{y}(\lambda) d\lambda \\ &= C(\omega_i, \omega_r) \int_\lambda \left\langle \left| P_{dtft} \left(\frac{2\pi u}{\lambda}, \frac{2\pi v}{\lambda} \right) \right|^2 \right\rangle S_y(\lambda) d\lambda \end{aligned} \quad (1.19)$$

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

1.4 Optimization using Taylor Series

Our final goal is to render structural colors resulting by the effect of wave diffraction. So far, we have derived an expression which can be used for rendering. Nevertheless, our current equation 1.19 used for computing structural colors, cannot directly be used for interactive rendering, since P_{dtft} had to be recomputed for every change in any direction¹⁶.

¹⁶According to changes in viewing- or incident light direction.

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 separate P_{dtft} in a certain way such that some computational expensive terms can be precomputed. The main idea is to formulate P_{dtft} as a series expansion relying on the definition of Taylor Series, as defined in equation ???. Further, we will provide an error bound for our approximation approach for a given number of terms. Lastly, we will substitute our approximation into our BRDF formula from equation 1.19.

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 wavelength λ . For any complex number t the power series expansion of the exponential function is equal to:

$$e^t = 1 + t + \frac{t^2}{2!} + \frac{t^3}{3!} + \dots = \sum_{n=0}^{\infty} \frac{t^n}{n!} \quad (1.20)$$

Now, when we use the exponent¹⁷ of $p(x, y)$ as an input argument for equation 1.20 we get:

$$\begin{aligned} e^t &= e^{ikwh} \\ &= 1 + (ikwh) + \frac{1}{2!}(ikwh)^2 + \frac{1}{3!}(ikwh)^3 + \dots \\ &= \sum_{n=0}^{\infty} \frac{(ikwh)^n}{n!}. \end{aligned} \quad (1.21)$$

where i is the imaginary unit for complex numbers. For simplification, in the reminder of this section we omitted the arguments of h . Equation 1.21 gives us an expression for an exponential series expansion for the exponent of $p(x, y)$. Please note that the above Taylor series is convergent for any complex valued number. Therefore the equation 1.21 is equal to

$$p(x, y) = \sum_{n=0}^{\infty} \frac{(ikwh(x, y))^n}{n!} \quad (1.22)$$

and thus gives us a series representation of $p(x, y)$. Next, calculating the Fourier transformation \mathcal{F} of equation 1.22 gives us the identity:

$$\begin{aligned} \mathcal{F}\{p\} &\equiv \mathcal{F}\left\{\sum_{n=0}^{\infty} \frac{(ikwh)^n}{n!}\right\} \\ &\equiv \sum_{n=0}^{\infty} \mathcal{F}\left\{\frac{(ikwh)^n}{n!}\right\} \\ &\equiv \sum_{n=0}^{\infty} \frac{(ikw)^n}{n!} \mathcal{F}\{h^n\} \end{aligned} \quad (1.23)$$

Where we have exploited the fact that the Fourier transformation is a linear operator. Therefore, in equation 1.23, 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 ¹⁸ in order to determine the Fourier transformation of p from equation ?? equation 1.23 then correspond to:

¹⁷This exponent is a complex valued function, equal to $ikwh(x,y)$.

¹⁸This identifier P may be subscripted by $dtft$ which will denote the DTFT variant of P .

$$P(\alpha, \beta) = \sum_{n=0}^{\infty} \frac{(ikw)^n}{n!} \mathcal{F}\{h^n\}(\alpha, \beta) \quad (1.24)$$

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

$$\tilde{P}_N(\alpha, \beta) := \sum_{n=0}^N \frac{(ikwh)^n}{n!} \mathcal{F}\{h^n\}(\alpha, \beta) \approx P(\alpha, \beta) \quad (1.25)$$

\tilde{P}_N is a good approximation of P , i.e. their absolute difference is small¹⁹. But first, the following two facts would have to be proven²⁰:

1. Show that there exist such an $N \in \mathbb{N}$ s.t. the approximation of equation 1.25 holds true.
2. Find a value for N s.t. this approximation is below a certain error bound, e.g. close to machine precision ϵ .

Our facts are valid and proven (see appendix ??) implies that there actually exists such an N . Thus, we can make use of the taylor series approximation from equation 1.25 and use it for approximating P_{dft} . This idea allows us to adapt equation 1.19, which is used for computing the structural colors of our BRDF model, in a numerically fast way. Finally the equation for the luminance is equal to:

$$\begin{aligned} Y &= C(w_i, w_r) \int_{\lambda} \left\langle \left| P_{dft} \left(\frac{2\pi u}{\lambda}, \frac{2\pi v}{\lambda} \right) \right|^2 \right\rangle S_y(\lambda) d\lambda \\ &= C(w_i, w_r) \int_{\lambda} \left\langle \sum_{n=0}^N \frac{(wk)^n}{n!} \mathcal{F}\{i^n h^n\} \left(\frac{2\pi u}{\lambda}, \frac{2\pi v}{\lambda} \right) \right\rangle^2 S_y(\lambda) d\lambda \end{aligned} \quad (1.26)$$

Notice that equation 1.26 is constrained by N and hence is an approximation of equation 1.19. Furthermore, it is possible to seperate out all the Fourier Terms in the summation and precompute them. This is why the approach in equation 1.26 is fast in order to compute structural color values according to our BRDF model. In other words this means that the Fourier terms in equation 1.26 are not anymore dependent on ω when using this Taylor series approximation.

1.5 Spectral Rendering using DFTs

In this section we describe how our final model for rendering structural colors due to diffraction will look like. For this purpose we use all our previous findings and plug them together to one big equation. For a given height field h representing the surface of a grating, we want to compute the resulting color due to light diffracted on that grating. For rendering we rely on the CIE_{XYZ} colorspace. For given direction vectors w_i and w_r as shown in figure ?? the DFT of the height field is equal to:

$$DFT_n\{h\}(u, v) = F_{dft}\{i^n h^n\} \left(\frac{2\pi u}{\lambda}, \frac{2\pi v}{\lambda} \right) \quad (1.27)$$

¹⁹Mathematically speaking, this statement correspond to $\|\tilde{P}_N - P\| \leq \epsilon$, where $\epsilon > 0$ is a small number.

²⁰Please have a look in section ?? in the appendix

From section 1.2.3 we know that we can reproduce a FT by applying a Gaussian window on the DFT from equation 1.27. This windowing approach gives us:

$$W_n(u, v) = \sum_{(r,s) \in \mathcal{N}_1(u,v)} |DFT_n\{h\}(u - w_r, v - w_s)|^2 \phi(u - w_r, v - w_s) \quad (1.28)$$

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 an the k-ring neighborhood around (u, v) with k as a parameter.

In section 1.4 we derived equation 1.26 which tells us how to compute the CIE_{XYZ} color value of a particular color channel using our relative reflectance BRDF model from section 1.3.2. Plugging all these findings together and using our windowing approach, listed in equation 1.28, 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:

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = C(\omega_i, \omega_r) \int_{\Lambda} \sum_{n=0}^N \frac{(2\pi w)^n}{\lambda^n n!} W_n(u, v) \begin{pmatrix} S_x(\lambda) \\ S_y(\lambda) \\ S_z(\lambda) \end{pmatrix} d\lambda \quad (1.29)$$

Where $C(\omega_i, \omega_r)$ is the defined in equation 1.15. Note that equation 1.29 integrates over a given wavelength spectrum, denoted by Λ . Usually, this Λ is equal to $[\lambda_{min}, \lambda_{max}]$ where $\lambda_{min} = 380nm$ and $\lambda_{max} = 780nm$.

1.6 An Alternative Approach

1.6.1 PQ factors

In this section we are presenting an alternative approach to the previous Gaussian window approach described in section 1.2.3 in order to solve the issue of working with the $DTFT$ instead of 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 repetitive function, i.e. $\forall x \in \mathbb{R} : S(x) = S(x + nT)$, where T is its fundamental period and $n \in \mathbb{N}_0$. Thus, the surfaces can be written formally as:

$$S(x) = \sum_{n=0}^N f(x + nT) \quad (1.30)$$

What we are looking for is an identity for the Fourier transform²¹ of our surface S , required in order to simplify the (X, Y, Z) colors from 1.26:

²¹Remember that we are using the definition of Fourier Transform used in electrical engineering where \mathcal{F} actually corresponds to the inverse Fourier Transform.

$$\begin{aligned}
\mathcal{F}\{S\}(w) &= \int f(x)e^{iwx}dx \\
&= \int_{-\infty}^{\infty} \sum_{n=0}^N f(x+nT)e^{iwx}dx \\
&= \sum_{n=0}^N \int_{-\infty}^{\infty} f(x+nT)e^{iwx}dx
\end{aligned} \tag{1.31}$$

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

$$\begin{aligned}
x &= y - nT \\
dx &= dy
\end{aligned} \tag{1.32}$$

Plugging this substitution back into equation 1.31 we will get:

$$\begin{aligned}
\mathcal{F}\{S\}(w) &= \sum_{n=0}^N \int_{-\infty}^{\infty} f(x+nT)e^{iwx}dx \\
&= \sum_{n=0}^N \int_{-\infty}^{\infty} f(y)e^{iw(y-nT)}dy \\
&= \sum_{n=0}^N e^{-iwnT} \int_{-\infty}^{\infty} f(y)e^{iwy}dy \\
&= \sum_{n=0}^N e^{-iwnT} \mathcal{F}\{f\}(w) \\
&= \mathcal{F}\{f\}(w) \sum_{n=0}^N e^{-iwnT}
\end{aligned} \tag{1.33}$$

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:

$$\begin{aligned}
\sum_{n=0}^N e^{-iwnT} &= \sum_{n=0}^N (e^{-iwT})^n \\
&= \frac{1 - e^{iwT(N+1)}}{1 - e^{-iwT}}
\end{aligned} \tag{1.34}$$

We recognize the geometric series identity for the left-hand-side of equation 1.34. Mainly relying on trigonometric identities, equation 1.33 can be further simplified to:

$$\mathcal{F}\{S\}(w) = (p + iq)\mathcal{F}\{f\}(w) \tag{1.35}$$

where p and q are defined as:

$$\begin{aligned} p &= \frac{1}{2} + \frac{1}{2} \left(\frac{\cos(wTN) - \cos(wT(N+1))}{1 - \cos(wT)} \right) \\ q &= \frac{\sin(wT(N+1)) - \sin(wTN) - \sin(wT)}{2(1 - \cos(wT))} \end{aligned} \quad (1.36)$$

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 nano-scaled surface snake shed represented as a two dimensional height field $h(x, y)$. 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 that 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 1.35.

$$\mathcal{F}\{S\}(w_1, w_2) = (p + iq)\mathcal{F}_{DTFT}\{h\}(w_1, w_2) \quad (1.37)$$

Note that a detailed derivation of equation 1.37 can be found in the appendix in section ?? and we have defined :

$$\begin{aligned} p &:= (p_1 p_2 - q_1 q_2) \\ q &:= (p_1 p_2 + q_1 q_2) \end{aligned} \quad (1.38)$$

For the identity of equation 1.37 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 1.32, but this time twice, once for x_1 and once for x_2 separately. The last step in equation 1.37, 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 Sinc Interpolation in Frequency Domain

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 the DFT applied on our height field. We assume that our height field is a superposition of periodically aligned substructures (i.e. finger structures). This so called PQ approach allows us to integrate over one period of a substructure in our height field, instead of iterating over the whole domain. Nevertheless, this main finding, described in equation 1.37, is using the DTFT. Thus, since our original height field is supposed to be a continuous-time band-limited function we can reconstruct it by applying a sinc-interpolation.

The aim of using a *sinc* interpolation is to reconstruct a spatial, bandlimited and continuous signal from a given sequence of samples. Therefore, for a given sequence of real numbers $x[n]$, representing a digital signal, its correspond continuous function is:

$$x(t) = \sum_{n=-\infty}^{\infty} x[n] \text{sinc} \left(\frac{t - nT}{T} \right) \quad (1.39)$$

which has the Fourier transformation $X(f)$ whose non-zero values are confined to the region $|f| \leq \frac{1}{2T}$. 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 less than the Nyquist frequency, then $x(t)$ is a perfect reconstruction of the original function. Figure 1.6 illustrates a reconstruction of a 1d signal relying on a sinc-interpolation.

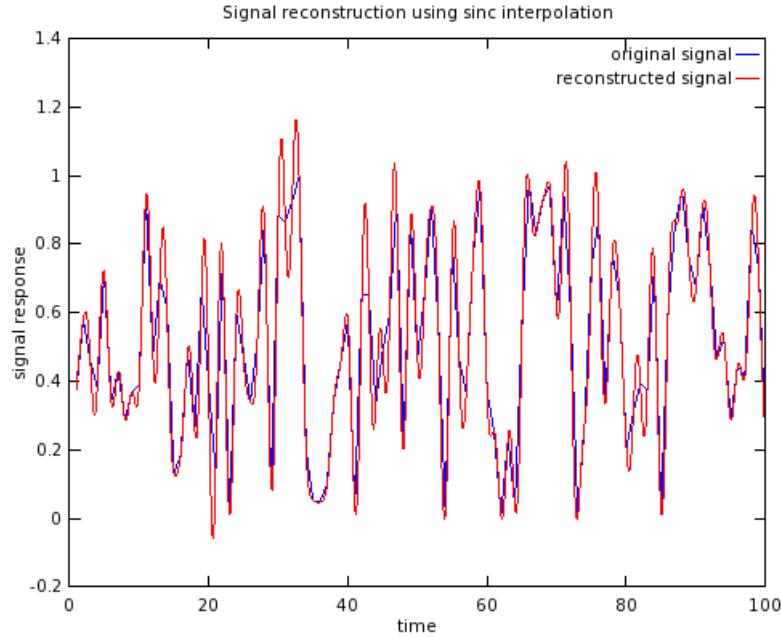


Figure 1.6: Comparison between a given random one dimensional input signal $s(t)$ and its sinc interpolation $\hat{s}(t)$. Notice that for the interpolation there were $N = 100$ samples from the original signal provided.

We use the sinc interpolation in order to reconstruct the DFT terms of our height field. Similarly, as discussed in section 1.2.3, we reconstruct the DFT by a windowing approach. Instead of using a Gaussian window we will use the sinc interpolation.

In the next chapter we talk about how to use this sinc-interpolation for rendering purposes (see section ??).

Chapter 2

Evaluation and Data Acquisition

In this chapter we will provide and discuss an evaluation of our rendering approaches. For this purpose we compare our method's (peak) viewing angles with maximum reflectance for a fixed incident beam with those resulting from the grating equation at different wavelengths. But first we revisit the term diffraction grating and provide a detailed definition.

2.1 Diffraction Gratings

In order to evaluate the quality of our simulations, it is important to understand all underlying elements involved in the rendering process. One particular element, which we have not investigated in detail is the diffraction grating represented by our height field. Thus, in this section we will examine in detail, what a diffraction grating actually is and how it works.

By the term *diffraction grating* we are referring to the surface of a flat piece of an opaque material that contains a large number of parallel, closely and evenly spaced slits¹ or bumps. Therefore, these slits are forming a periodically packed, groove-like structure along the surface of the material.

A diffraction grating alters the state of an incident light beam by diffracting its component waves. According to Huygen's principle (see section ??), when an incident light beam hits the grating points on the slits, the grating will act as point light sources that emits spherical wavelets. Basically, a diffraction grating can be either transmissive (see figure 2.2(*a*)) or reflective (see figure 2.7). As light transmits through or reflects off a grating, the grooves on the grating cause different wavelengths of the light to diffract differently and thus divide the light into its component wavelengths. This also implies that the emitted wave will have a different outgoing angle with peak intensity than the incident light. In general, the closer the spacing of the slits is to the wavelength of the incident wave, the more the emitted wave will be diffracted.

Figure 2.1 illustrates what happens when a monochromatic light passes through a transmissive grating. Using a spectrometer, we see that the outgoing angle of the emitted wave will be different from the incident angle. Hence, the diffracted light is composed of the sum of interfering wave components emanating from each slit in the grating.

¹Usually, these slits are either engraved or etched into the surface of the material.

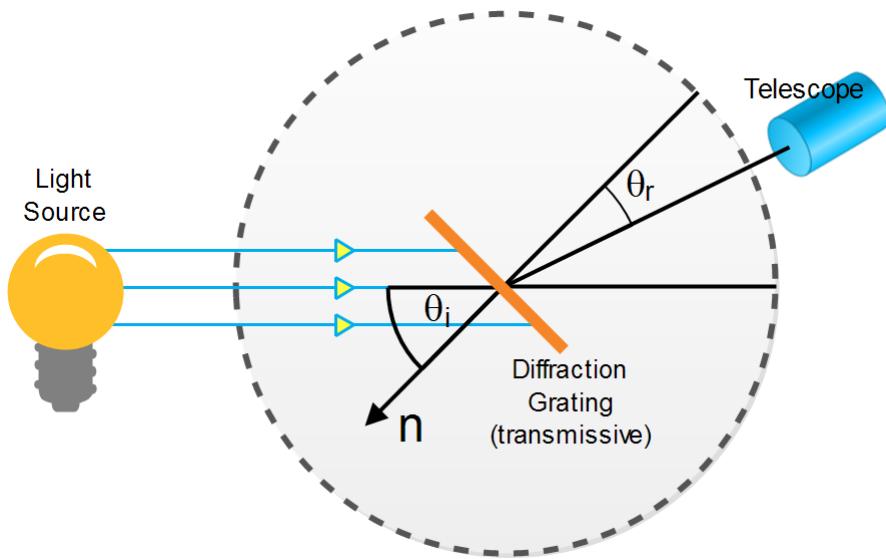


Figure 2.1: Spectrometer: When a beam of monochromatic light passes through a grating placed on a spectrometer, images of the sources can be seen through the telescope at different angles.

Suppose an incident plane wavefront, composed of a monochromatic light source, is directed at a transmissive diffraction grating, parallel to its axis(i.e. its surface normal) as shown in figure 2.2. Let the distance between successive slits on the grating be equal to d . Furthermore, at a distance L , there is a screen parallel to the grating. Then the emitted waves will form a diffraction pattern on the screen which is the result of interference effects (constructive or destructive interference) among outgoing wavelets as shown in figure 2.2(a).

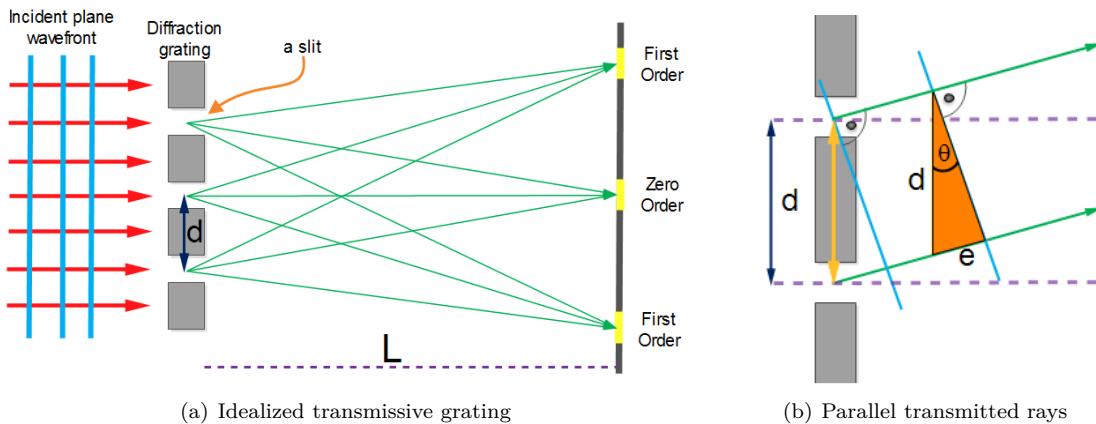


Figure 2.2: Light directed to parallel to grating

If the distance to the screen is much larger than the slits width, i.e. $L \gg d$, then all the rays emanating from the surface and ending up at the receiver are parallel. Thus, the path difference between waves from any two adjacent slits can be derived by drawing a perpendicular line between

the parallel rays. Applying simple trigonometry gives us this path difference as $e = ds\sin(\theta)$ as shown in figure 2.2(b). If the path difference equals one wavelength or a multiple of the wave's wavelength, the emerging, reflected waves from all slits will be in phase and a bright line will be observed at that point. Therefore, the condition for a local maxima in the interference pattern is:

$$ds\sin(\theta) = m\lambda \quad (2.1)$$

where $m \in \mathbb{N}_0$ is the order of diffraction and λ is the wavelength. Because d is very small for a diffraction grating, a beam of monochromatic light passing through it is split into very narrow bright fringes at large angles θ (see figure 2.4). Without the loss of generality, either for a transmissive or a reflective diffraction grating type, an analogous derivation for equation 2.1 can be derived.

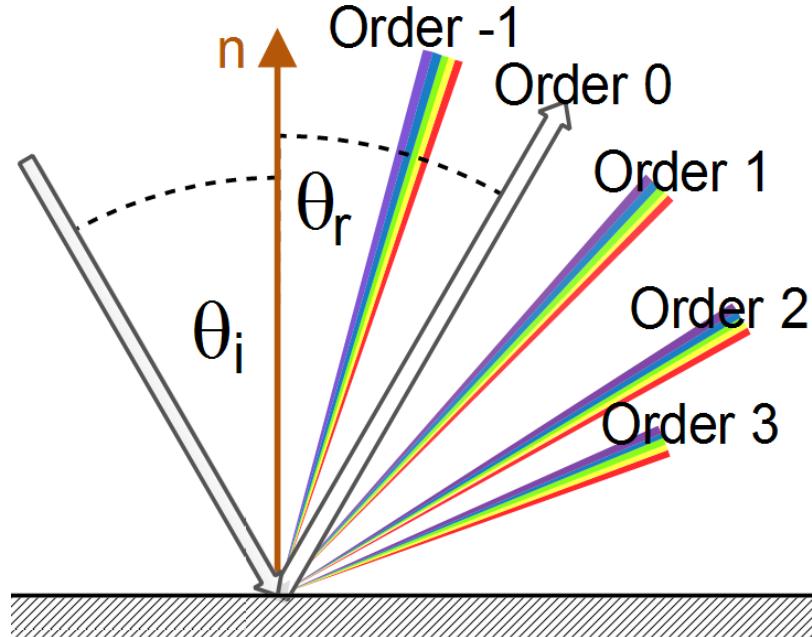


Figure 2.3: Illustration of different diffraction orders when light is diffracted on a reflective diffraction grating. An incident beam of light hits a surface at an angle θ_i w.r.t. the surface normal n . The angle θ_r denotes the angle of the reflected beam (zero order).

When a beam of white light is directed at a diffraction grating along its axis, instead of a monochromatic bright fringe, a set of colored spectra are observed on both sides of the central white band. Figure 2.4 illustrates this for different number of slits on a diffraction grating.

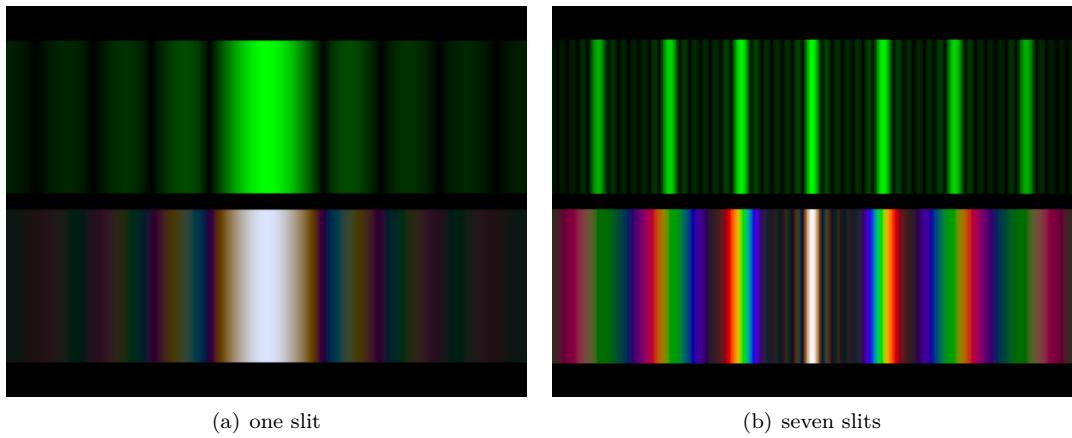


Figure 2.4: Difference of diffraction pattern² between a monochromatic (top) and a white (bottom) light spectra for different number of slits.

Since the reflection angle θ_r for a maxima increases with wavelength λ , red light, which has the longest wavelength, is diffracted through the largest angle. Similarly violet light has the shortest wavelength and is therefore diffracted the least. Thus, white light is split into its component colors from violet to red light. The spectrum is repeated in the different orders of diffraction, emphasizing certain colors differently, depending on their order of diffraction like shown in figure 2.3. Note that only the zero order spectrum is pure white. Figure 2.5 shows the relative intensity resulting when a beam of light hits a diffraction grating with different number of slits. From the graph we recognise that the more slits a grating has, the sharper more slopes the function of intensity gets. This is similar like saying that, the more periods a grating has, the sharper the diffracted color spectrum gets like shown in figure 2.4.

²These images have been taken from <http://www.itp.uni-hannover.de/~zawischa/ITP/multibeam.html>

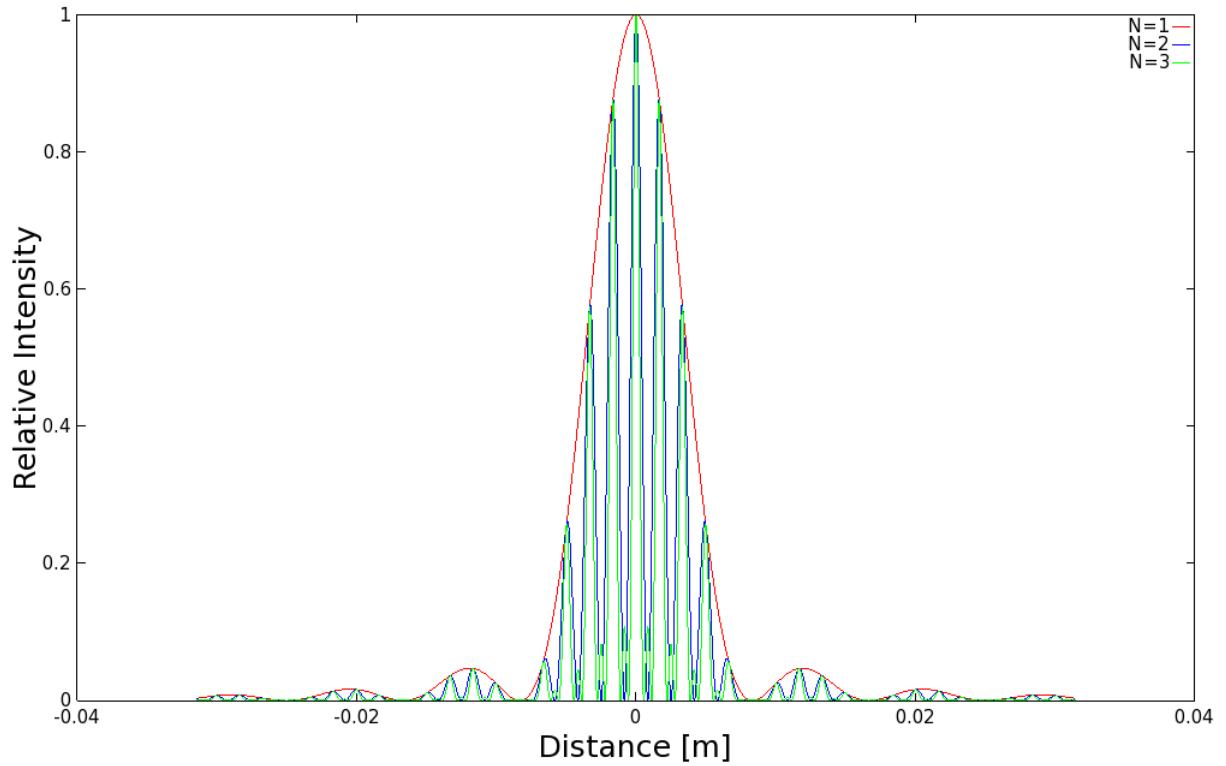


Figure 2.5: Relative intensities of a diffracted beam of light with wavelength $\lambda = 500\text{nm}$ on a grating for different number of slits N . A slit width of 30 microns and a slit separation of 0.15 mm was used. The viewer is 0.5m away from the grating.

2.2 Data Acquisition

Our goal is to perform physically accurate simulations of diffraction effects due to natural gratings. As for every simulation, its outcome highly depends on the input data and thus we also require measurements³ of real natural gratings. For that purpose, samples of skin sheds of Xenopeltis and Elaphe snake species were fixed on a glass plate. Then, by using an Atomic Force Microscope (AFM), their surface topography was measured and digitally stored⁴. In general, an AFM is a microscope that uses a tiny probe mounted on a cantilever to scan the surface of an object. The probe is extremely close to the surface, but does not touch it. As the probe traverses the surface, attractive and repulsive forces arising between it and the atoms on the surface inducing forces on the probe that bends the cantilever. The amount of bending is measured and recorded, providing a depth-map of the atoms on the surface. An atomic force microscope is a very high-resolution probe scanner with its demonstrated resolution on the order of a fraction of a nanometer, which is more than 1000 times better than the optical diffraction limit. The resolution of any optical system is limited by the optical diffraction limit due to the effect of diffraction. Thus, no matter how well a lens of an optical system is corrected, its resolution is fundamentally limited by this

³All measured data has been provided by the Laboratory of Artificial and Natural Evolution at Geneva - Website:www.lanevol.org

⁴Note these measured height fields can be visualized using grayscale images, indicating their relative depth.

optical barrier.

2.3 Verifications

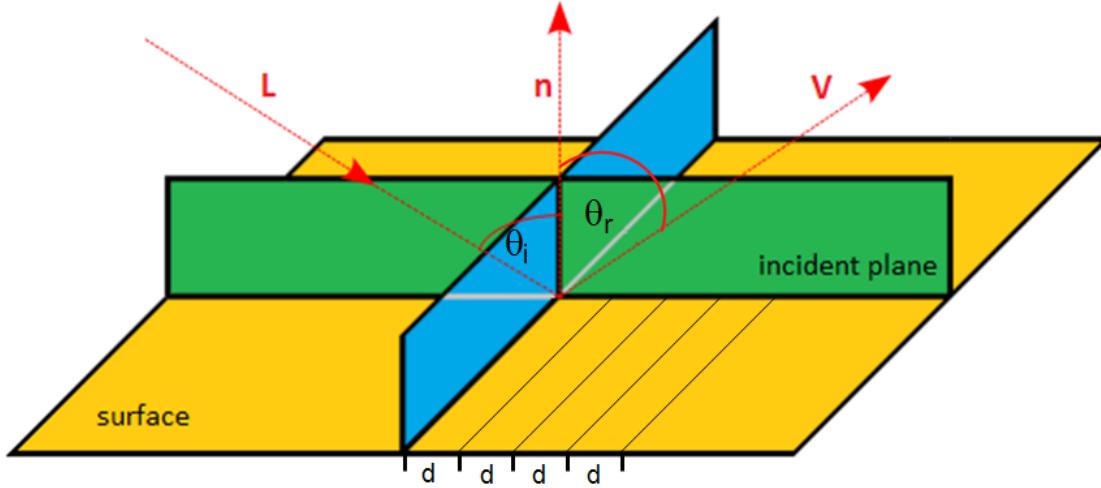


Figure 2.6: Experimental setup for evaluation: A light beam with direction L hits the surface, representing a grating pattern with periodicity d , at the incident plane⁵ relative to the surface normal n at angle θ_i and emerges at an angle θ_r with viewing direction V .

The physical reliability of our BRDF models has been verified by applying it to a height field for a synthetic blazed grating. Figure 2.6 illustrates the geometrical setup for our evaluation approach: A monochromatic beam of light with wavelength λ hits a surface with periodicity d at an angle θ_i relative to the normal n along its incident plane. The beam emerges from the surface at the angle θ_r with certain intensity as predicted by our model. Note that actually two angles are necessary in order to define a direction vector, using spherical coordinates. However, in our evaluation we fixed the azimuthal angle of these directional vectors (Further information can be found in section 2.3.1).

In our evaluation we compare the local peak angles predicted by our model with those resulting from the grating equation 2.2. The grating equation models the relationship between the grating spacing, the incident light angle and the angle for a local maxima for the diffracted light beam.

⁵Remember that in general a direction vector is determined by two angles. In our evaluation setup one angle is initially fixed. Thus, regardless what value we choose for our free angle parameter the incident light will always be parallel to a so called incident plane (green plane shown in figure 2.6).

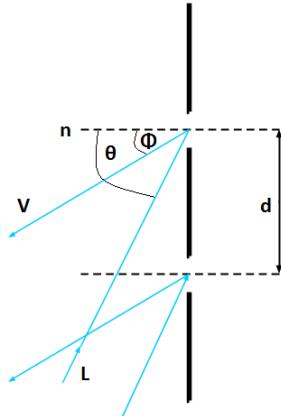


Figure 2.7: Reflecting grating: When the incident light direction is not parallel to its axis at the grating there is another $\sin(\phi)$ involved. See also the grating equation 2.2.

Figure 2.7 shows that if the incident light is not along the axis of the diffraction gratings then it effects the optical path differences. The angles with locally maximum intensity is given by the grating equation derived from the equation 2.1 following figure 2.7:

$$\sin(\theta_i) = \sin(\theta_r) + \frac{m\lambda}{d} \quad (2.2)$$

In our evaluation we are interested in the first order diffraction, i.e. $m = 1$. We further assume that the incident light direction L is given. In contrast the direction of the reflected wave V is a free parameter.

In Mathematics, a three dimensional direction vector is fully defined by two angles, i.e. it can be represented by spherical coordinates. Hence, θ_i , is a given constant whereas θ_r is a free parameter for our evaluation simulation. Therefore, we are going to compare the maxima or the peak viewing angles corresponding to each wavelength using data produced from our method against the maxima resulting by the grating equation 2.2.

2.3.1 Numerical Comparisons

In this section we explain how we evaluated the quality of our BRDF models. For a fixed incident light direction we want to compare the peak viewing angles with maximum reflectance for our method with those resulting from the grating equation for different wavelengths. For this purpose we realized the BRDF models corresponding to different gratings for each of our shading approaches, FLSS, NMM and PQ, in Java. By fixing the azimuth angle⁶ of the incident light L and viewing direction V , we reduced the degrees of freedom due to these directions, during our evaluation. Thus, any BRDF is then defined by a function that expects as input arguments a wavelength λ , the inclination angle of the incident light θ_i - and the viewing direction θ_r . The return value of such a function, denoted by $BRDF(\lambda, \theta_i, \theta_r)$, is the intensity value of the corresponding BRDF at these given input arguments.

⁶Each direction vector in space can be expressed by spherical coordinates. Then, such a unit vector is defined by a pair of two angles, the inclination and the azimuth angle. For further information, please have a look at appendix ??

For further details how we evaluated our BRDF in Java and how we generated the corresponding evaluation plots, please refer the appendix chapter A.

2.3.2 Virtual Testbench

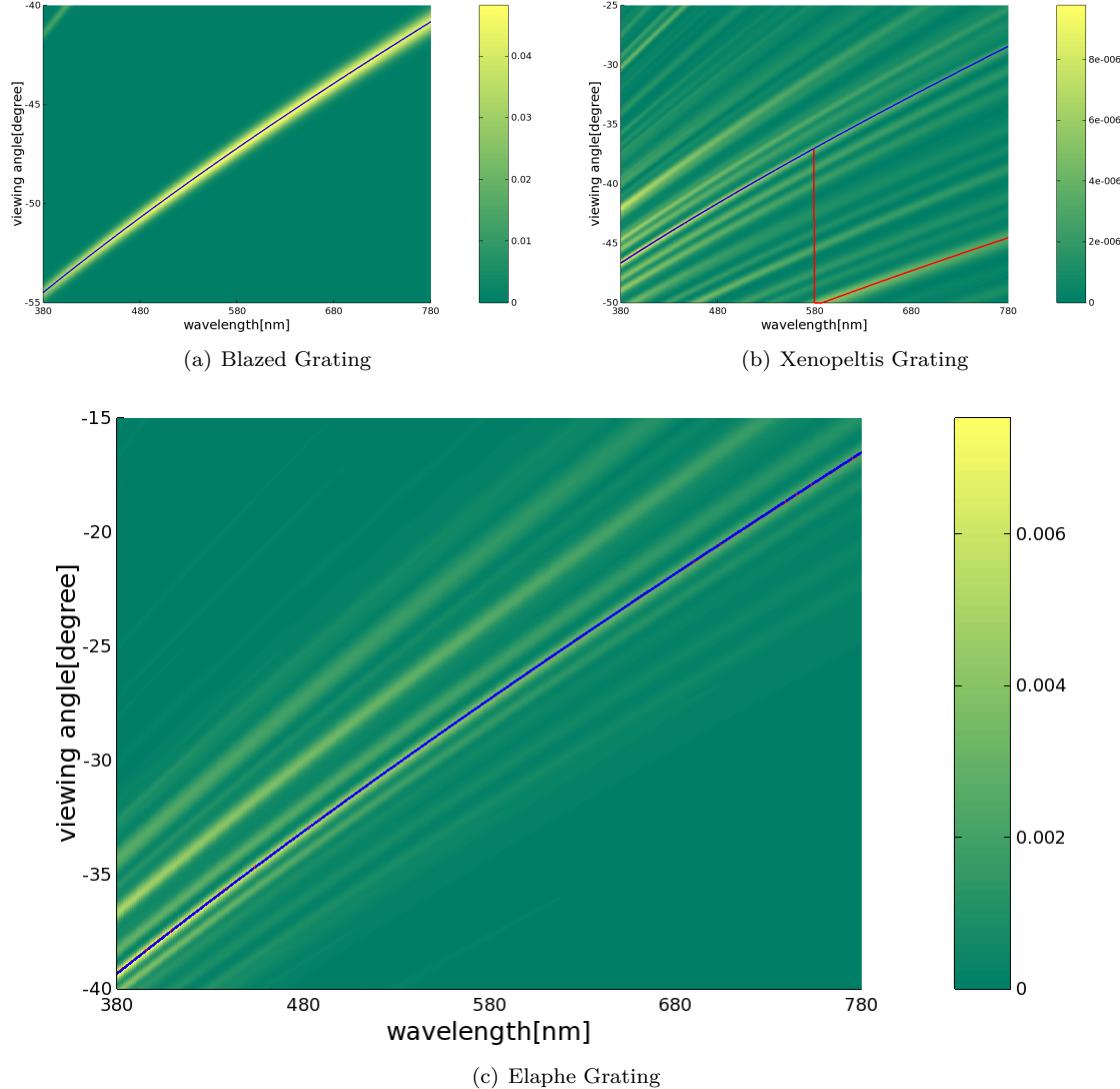


Figure 2.8: Reflectance obtained by using the FLSS approach described in algorithm ??.

In this section we discuss the quality of our BRDF models applied to different surface structures. For that purpose we compare the resulting relative reflectance computed as described in section 2.3.1 for each of our BRDF models to the idealized grating equation 2.2.

Measuring periodicity d in [nm]	Method					
	FLSS		NMM		PQ	
	mean	variance	mean	variance	mean	variance
Blazed Grating	2499.997	0.377	2499.997	0.377	2502.428	0.390
Elpae Grading	1144.262	0.401	1144.179	0.677	1052.308	49.678
Xenopeltis Grating	1552.27	0.45	-	-	-	-

Table 2.1: Statistics of periodicity d of our used gratings 3.2 estimated by using the grating equation 2.2.

Figure 2.8 shows the reflectance graphs (BRDF response) resulting from the FLSS approach over the *wavelength-spectrum-reflected-light-angle* grid (Λ, Θ) (according to the equations A.1 and A.2 as described in the appendix chapter A) as described in algorithm ???. This evaluation was applied to an idealized periodic structure, namely to the Blaze- 2.8(a) and to two natural gratings, the Elaphe- 2.8(c) and Xenopeltis grating 2.8(b). For all our evaluation plots, we used an illumination angle of θ_i equal to 75° .

Note that higher response values are plotted in yellow and lower values in green. For each of the graphs we determine the viewing angles with peak reflectance for each wavelength and then plot these peak viewing angles versus corresponding wavelengths as solid red curves. The blue curve represents diffraction angles for an idealized periodic structure with a certain periodicity d according to the grating equation 2.2. In the following we explain how the value d is computed.

By rearranging the terms of the grating equation defined in equation 2.2 and using the peak reflectance angle α_{r_k} derived like in algorithm 1 using the matrix R of equation A.3, we can compute a periodicity vale d_k for any wavelength λ_k .

$$d_k = \frac{\lambda}{\sin(\alpha_{r_k}) + \sin(\theta_i)} \quad (2.3)$$

By computing the mean value of these d_k values for all λ in Λ we can compute an estimated periodicity value d . We estimated these periodicity values for every grating structure and every method we are using. Thes corresponding periodicity values are tabulated in table 2.1.

The red and blue curve are closely overlapping in the figure 2.8(a) (for a blazed grating) and 2.8(c) (for an Elaphe grating). For Blaze and Elaphe there is only diffraction along one direction perceivable. Since the Blazed grating is synthetic we use its exact periodicity to plot the blue curve instead of estimating it. For Xenopeltis it is interesting to see that the red curve for the peak viewing angle toggles between two ridges corresponding to two different periodicities. This happens because there are multiple subregions of the nanostructure with slightly different orientations and periodicity. Each subregion carves out a different yellowish ridge. Depending on the viewing angle, the reflectance of such a subregion can be higher than others.

Figure 2.9 shows the evaluation plots for the NMM approach applied to the Blazed- (see figure 2.9(b)) and the Elaphe-grating (see figure 2.9(b)). The NMM approach is an optimization of the FLSS approach and is discussed in section ???. The response curve of each plotted NMM approach graph closely matches the corresponding grating equation curve. Furthermore, the evaluation graphs of the NMM approach look similar to the corresponding evaluation plots of the FLSS approach shown in figure 2.8. Thus, this confirms that the NMM optimization works well.

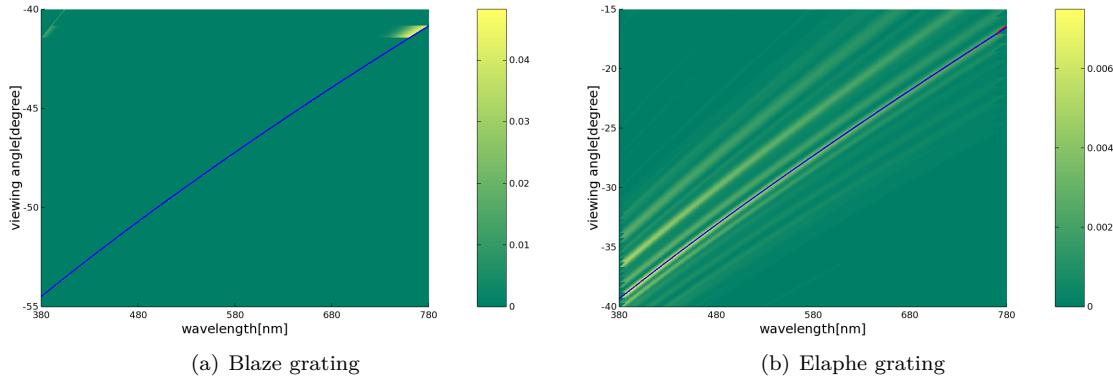


Figure 2.9: Reflectance obtained using NMM optimization approach.

Last, let us consider the evaluation graphs in figure 2.10 for the PQ approach described in algorithm ???. The PQ approach assumes the given grating being periodically distributed on the surface of a shape. For this approach we have plotted evaluation graphs of the Blaze- (See figure 2.10(a)) and Elaphe grating (See figure 2.10(b)). The response curves of both PQ evaluation graphs exhibit some similarities, but also some differences, compared to their corresponding grating equation curves. We could say that the response curve of the blaze grating is weakly oscillating around the grating equation curve (blue), but basically following it even with some outliers. The response curve of the Elpahe grating is not following its corresponding first order grating equation curve well. This could be due to the PQ's assumption that a given height field has to be periodically distributed along the surface. But in general, for natural gratings, this assumption usually does not hold true. Nevertheless, the red curve fits one of the response curves. We conclude that the PQ approach produces inaccurate results compared to the FLSS approach. Thus, the PQ approach is sub-optimal and unusable in practice.

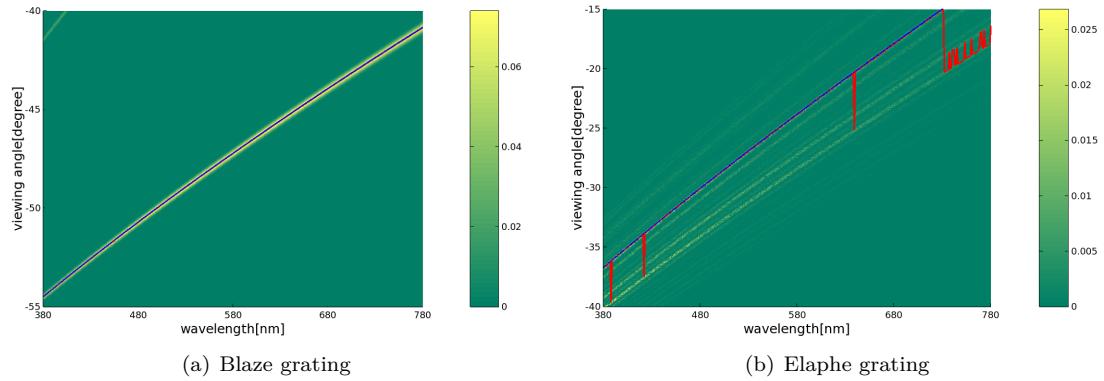


Figure 2.10: Reflectance obtained using PQ optimization approach.

Chapter 3

Results

In this chapter we examine the rendered output results of our implementation of our BRDF models applied to different input patches such as the Blaze grating and the Elaphe ?? and Xenopeltis ?? snake skins nanostructures. We are discussing and comparing both, their BRDF maps 3.1 and the corresponding renderings on a snake geometry as shown in section 3.2, for various input parameters. Last we also show a real experimental image showing the effect of diffraction for setting readers perception about expected results.

3.1 BRDF maps

A BRDF map shows a shader's output for all possible viewing directions for a given fixed, incident light direction. We assume that each viewing direction is expressed in spherical coordinates (See appendix ??) (θ_v, ϕ_v) and is represented in the map at a point

$$(x, y) = (\sin(\theta_v)\cos(\phi_v), \sin(\theta_v)\sin(\phi_v)) \quad (3.1)$$

with its origin at the center of the map. The light direction for normal incidence (θ_i, ϕ_i) has been fixed to $(0, 0)$ for our rendered results.

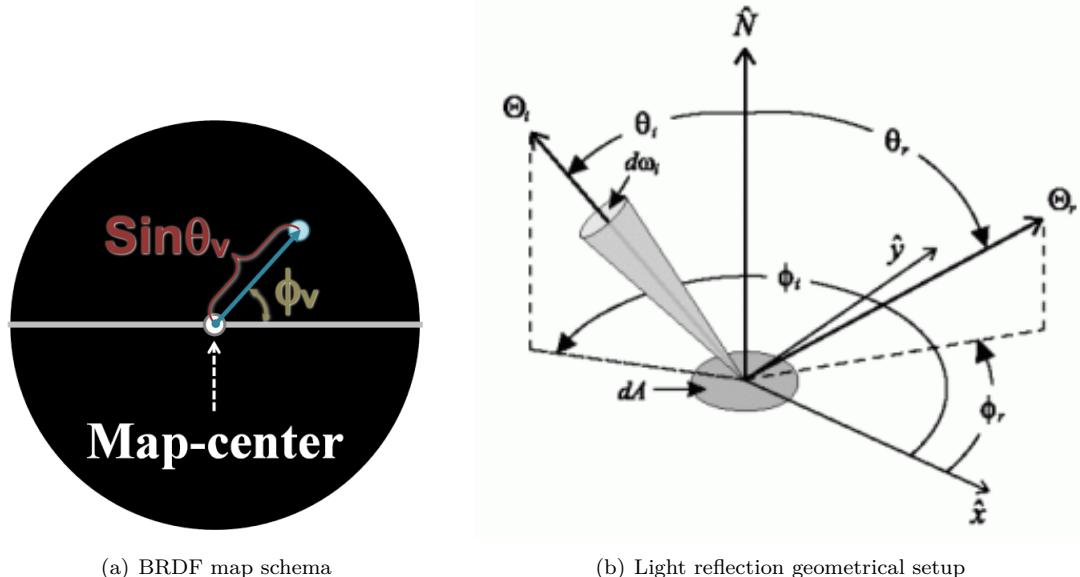


Figure 3.1: BRDF maps¹ for different patches: $\Theta = (\theta_i, \phi_i)$ is the direction of light propagation

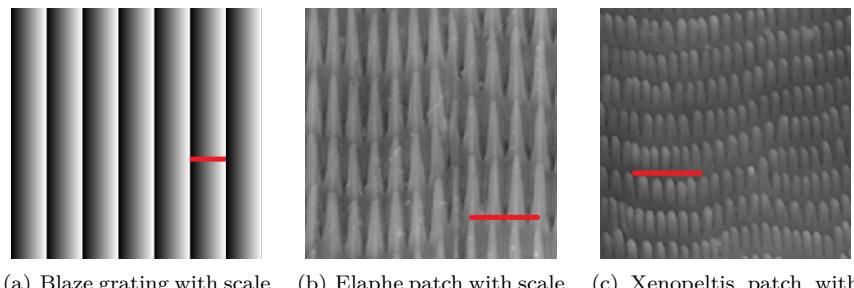


Figure 3.2: Cutouts of our nano-scaled surface gratings used for rendering with actual our shader with a scale indicator (red line) for each patch. Note that for actual rendering, we use larger patches.

Figure 3.3 shows the BRDF maps of the full lambda space sampling approach (**FLSS**) as described in section ?? applied on different nanoscale surface gratings shown in figure 3.2. In sub-figure 3.3(a) we see the BRDF map for the Blazed grating, showing high relative brightness for its first order diffraction, i.e. for the Blazed gratings most of the diffracted spectral energy lies in its first order. Notice that the surface of blazed grating is forming a step structure for which the angle between the step normal and the grating normal is denoted by *blaze angle*. Every blazed grating is

¹image source of figure:

- 3.1(a): Taken from D.S.Dhillon et al's Paper [D.S14]
- 3.1(b): Taken from <http://math.nist.gov/~FHunt/appearance/brdf.html>

manufactured in the Littrow² configuration. This means that the blaze angle is chosen such that for a chosen mode m and a chosen wavelength λ with maximum intensity and the incidence angle are identical. Thus a blazed grating has its maximal efficiency for the chosen wavelength of the light used. Higher diffraction modes are still perceivable (second and higher diffraction orders) but with a much lower relative brightness. The asymmetry in the brightness of the pattern is due to the asymmetric geometry of the grating 3.2(a).

The finger-like structures contained in the Elaphe surface grating 3.2(b) are considerably regularly aligned and hence diffraction occurs along the horizontal axis for the BRDF map as shown in figure 3.3(b). The reason for not seeing any strong diffraction color contribution along other directions in the BRDF map is due to the fact that these ‘nano-fingers’ overlap across layers and thus do not exhibit any well-formed periodicity along finger direction.

For the Xenopeltis surface grating shown in figure 3.2(c), we observe diffraction along many different, almost vertical directions in the BRDF map 3.3(c) since the layers of the finger-like structures do not overlap and are shifted significantly along their length but still exhibit some local consistency. A similar argument holds true for diffraction across locally periodic finger patches with slightly different orientations.

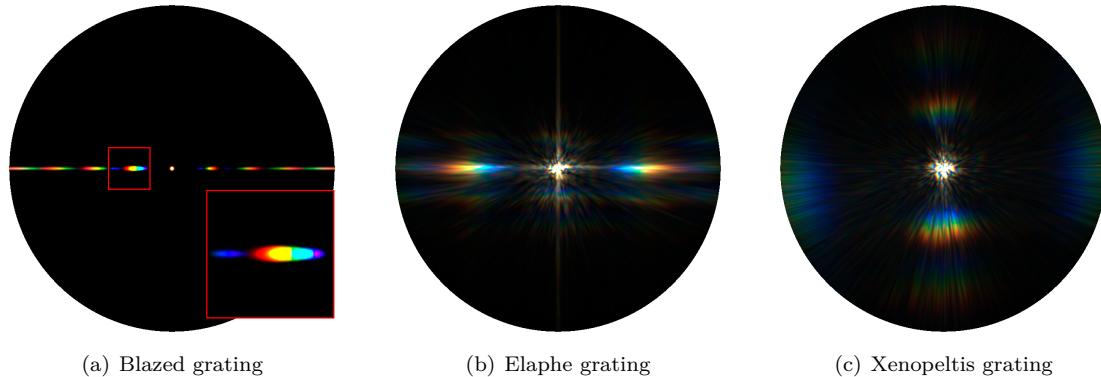


Figure 3.3: BRDF maps for different patches

Figure 3.4 shows BRDF maps of all our BRDF models applied on the Blaze grating. Figure 3.4(a) shows the FLSS shading approach result for our blazed grating and it is used in order to compare with our other rendering approaches.

Figure 3.4(b) shows the BRDF map for the NMM approach, introduced in section ??, which is close to the FLSS approach as verified in section 2.3 (see figure 2.9(a)). Nevertheless there is a small, noticeable difference: For the NMM approach we see a white, circular spot around the map center. Nevertheless, apart from this white spot, the NMM approach resembles the FLSS approach. The reason for this differences is due to the fact that the NMM approach treats the center of a BRDF map as a special case, as described in section ???. Technically, every location around a small ϵ -circumference from the map center gets white color assigned.

²For further information please see http://en.wikipedia.org/wiki/Blazed_grating.

Figure 3.4(c) shows the BRDF map for the PQ approach which relies on sinc-interpolation. The PQ BRDF map and the FLSS results are visual alike. In contrast to the evaluation plots in figure 2.10, the BRDF map for the PQ approach matches well with the one for the FLSS approach. Compared to FLSS, one difference we notice is that the first order of diffraction is a little spread for the PQ approach.

Last, let us consider figure 3.4(d) which shows the BRDF map produced by using Nvidia Gem’s implementation [JG04] of Stam’s BRDF model corresponding to periodic like structures with regularly repeated bumps (along y-axis of the BRDF map). This corresponds to a one dimensional diffraction grating, along the x-axis. This model only uses the spacing d of a given grating. It also always produces highly symmetric results and fails to confirm with the Littrow configuration.

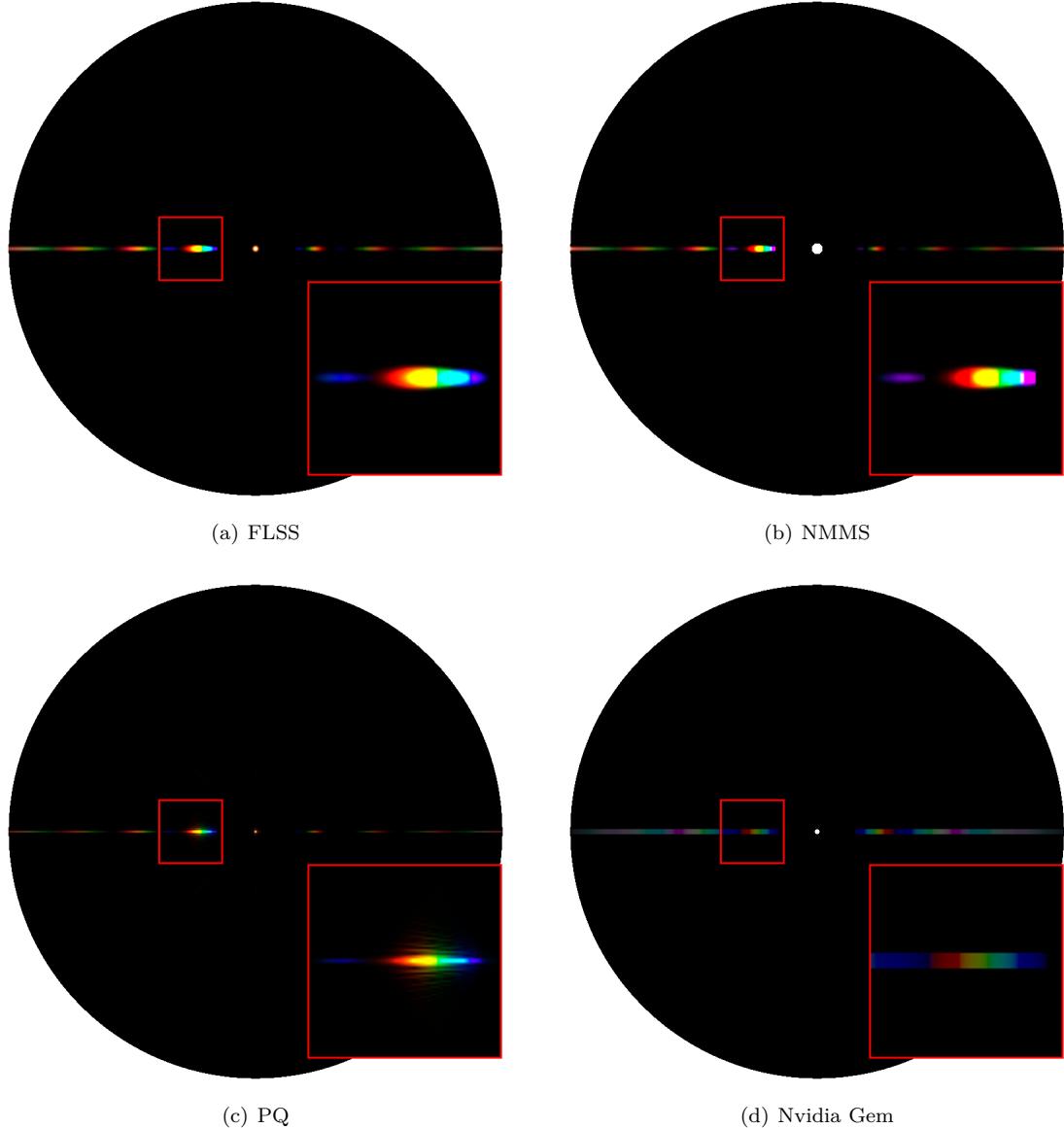


Figure 3.4: BRDF maps for Blazed grating comparing our different rendering approaches

Figure 3.5 and figure 3.6 show the BRDF maps for different wavelength step sizes used in the fragment shader for the FLSS approach applied to the blazed grating and the Elaphe snake shed, respectively. Within our fragment shaders the outermost loop iterates over the range [380nm, 780nm] for a given step size λ_{step} to integrate over the wavelength spectrum as illustrated in algorithm ???. Having bigger step sizes implies having fewer λ -samples which will reduce the overall runtime of a shader but it will also introduce artifacts and therefore lower the overall shading quality. For an Elaphe surface grating, artifacts are perceivable when $\lambda_{step} \geq 10\text{nm}$. Results produced by using 5nm step sizes do not differ from those produced by using $\lambda_{step} = 1\text{nm}$. This allows us to set λ_{step} at 5nm. For a Blazed grating we may chose even bigger step sized without losing any rendering

quality(see figure 3.5).

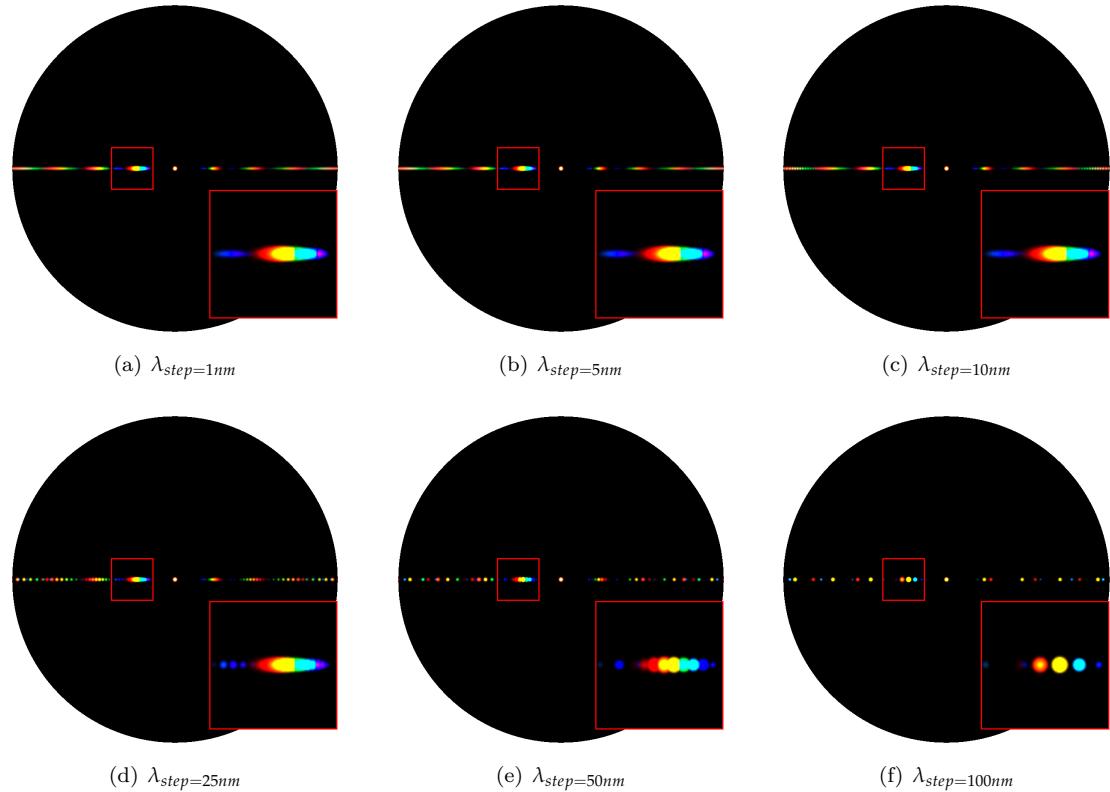
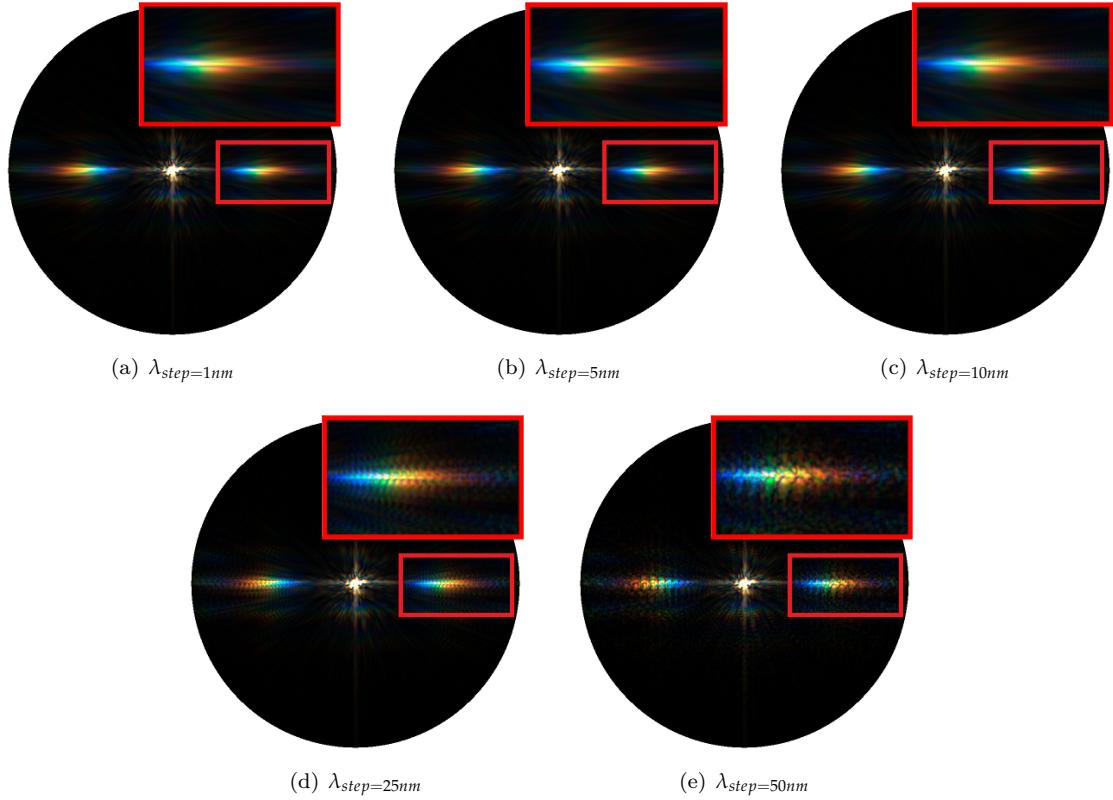


Figure 3.5: Blazed grating at $2.5\mu m$: Different λ step sizes

Figure 3.6: Elaphe grating at $65\mu m$: Different λ step sizes

The figures 3.7, 3.8, 3.9 show a comparison of the BRDF maps produced by the FLSS approach (on the left) and the PQ shading approach (on the right) applied on all our patches. For Blazed grating, as already mentioned, we notice that both approaches, FLSS and PQ, look similar but have notable differences. In the PQ map, the first order diffraction color contribution is spread. In general, a Blazed Grating is manufactured in a way that a large fraction of the incident light is diffracted preferentially into the first order. Therefore, most of the energy in its BRDF map lies in the first order of diffraction at its blaze angle. This implies that largest portion of the color contribution, visible on the corresponding BRDF map, lies at that angle. In figure 3.7, in contrast to the results produced by the FLSS approach, we see color fringes at the first order modes in the BRDF map produced by our PQ approach. This implies that the PQ approach does not produce reliable results which also affirms our evaluation plots shown in figure 2.10. The BRDF resulting BRDF map for the Elaphe and Xenopeltis gratings using the PQ approach are similar to those resulting by the FLSS approach. Nevertheless they also exhibit some artifacts, similar to those discussed for the Blazed grating.

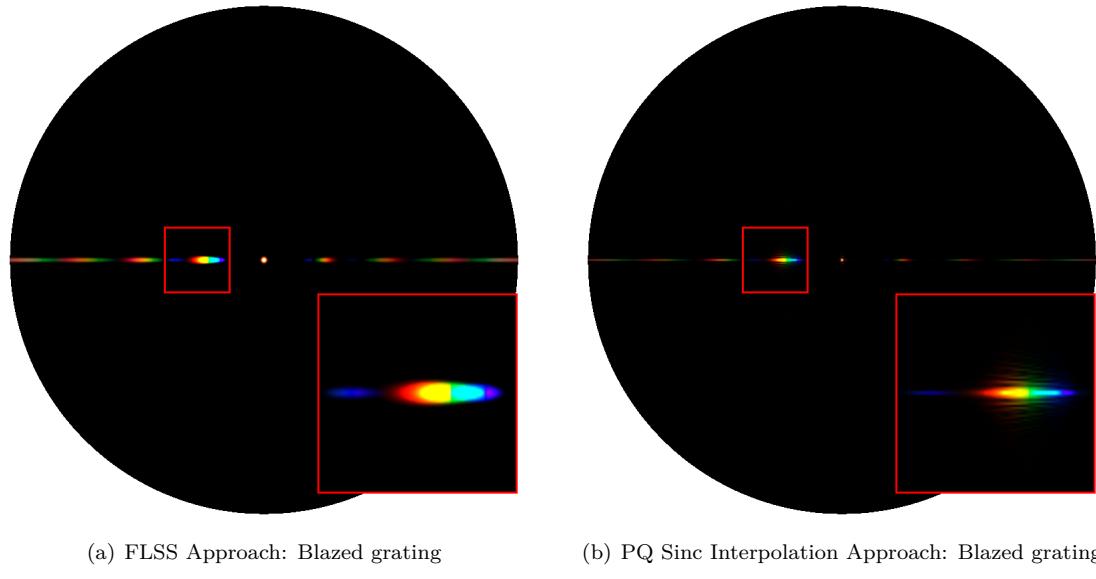


Figure 3.7: A comparison between the PQ- and the FLSS approach applied on an Blazed grating.

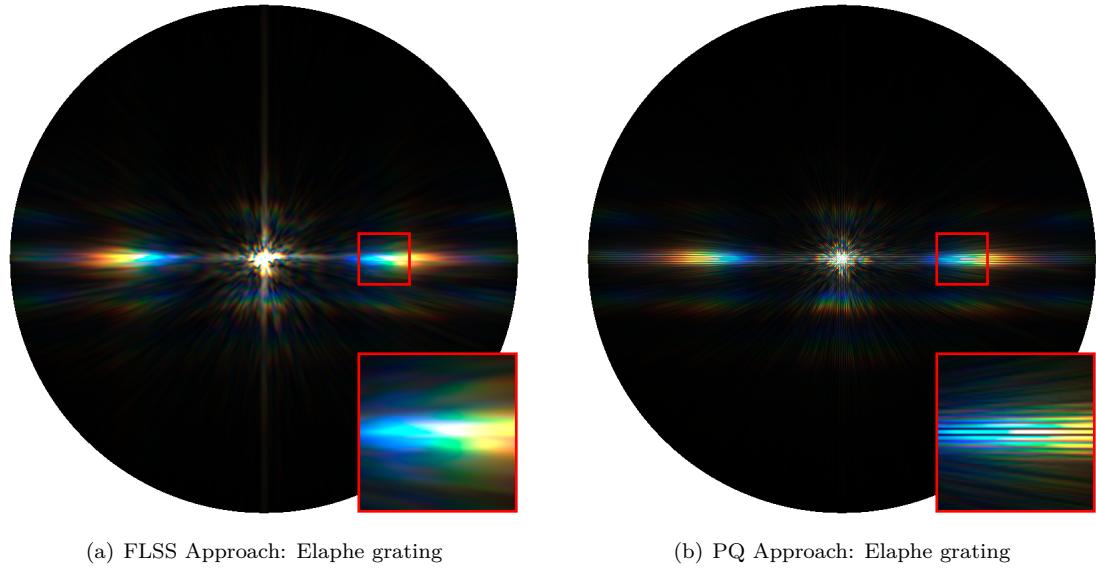


Figure 3.8: A comparison between the PQ- and the FLSS approach applied on an Elaphe grating.

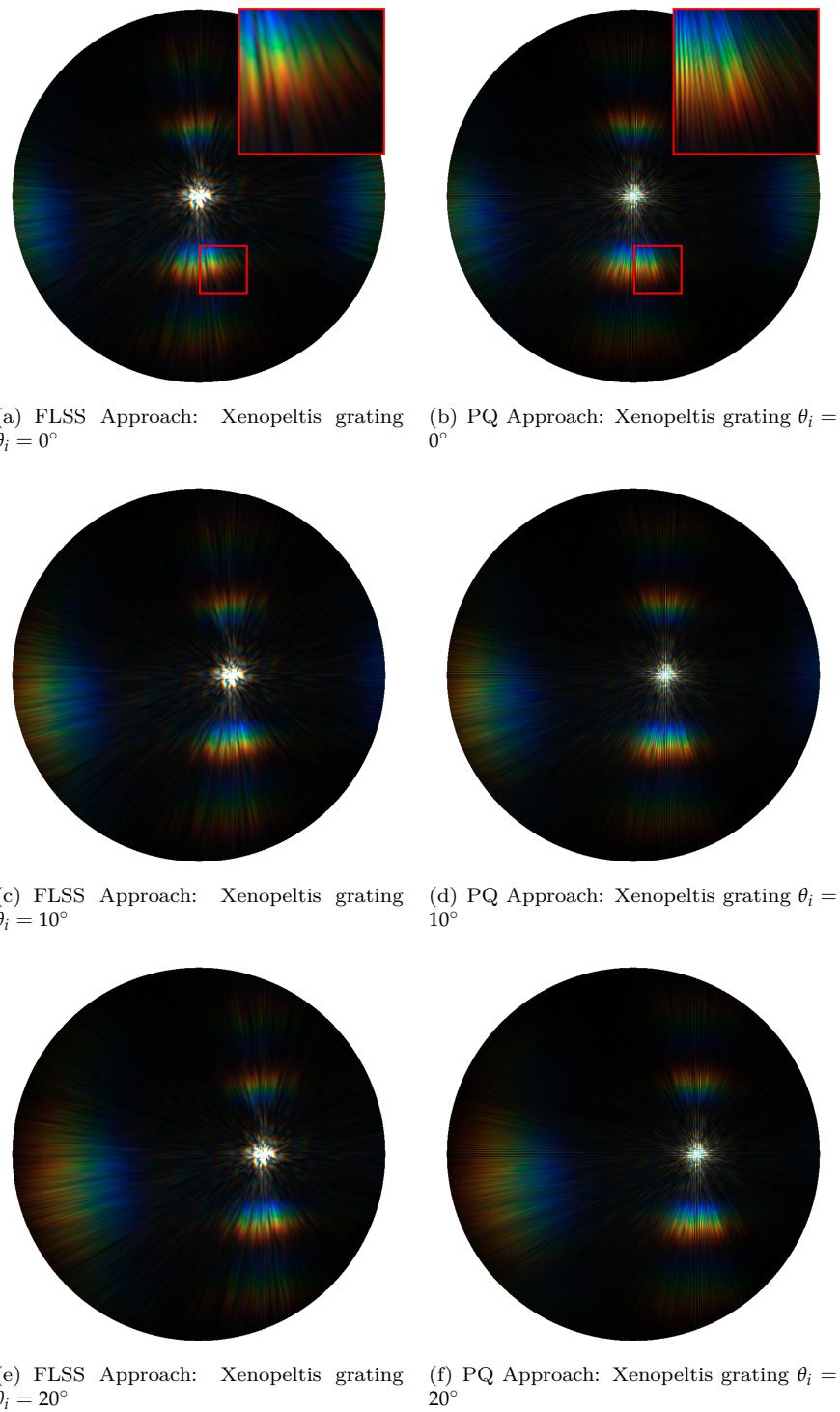


Figure 3.9: A comparison between the PQ- and the FLSS approach applied on an Xenopeltis grating.

Figure 3.10 shows BRDF maps for the FLSS approach when applied to the Blazed grating, while varying the value for the spatial variance σ_s for the coherence window. This is akin to changing the coherence length for the incident light. We see that the lower the coherence length gets, the fewer of the interacting grating periods are involved. Having fewer periods cause the production of overlapping blurry diffraction bands (i.e. blobs, see e.g. figure 3.10(b)) for different wavelengths λ . Thus, when reducing the number of periods this will produce blurry artifacts which ends in having poorly resolved colors.

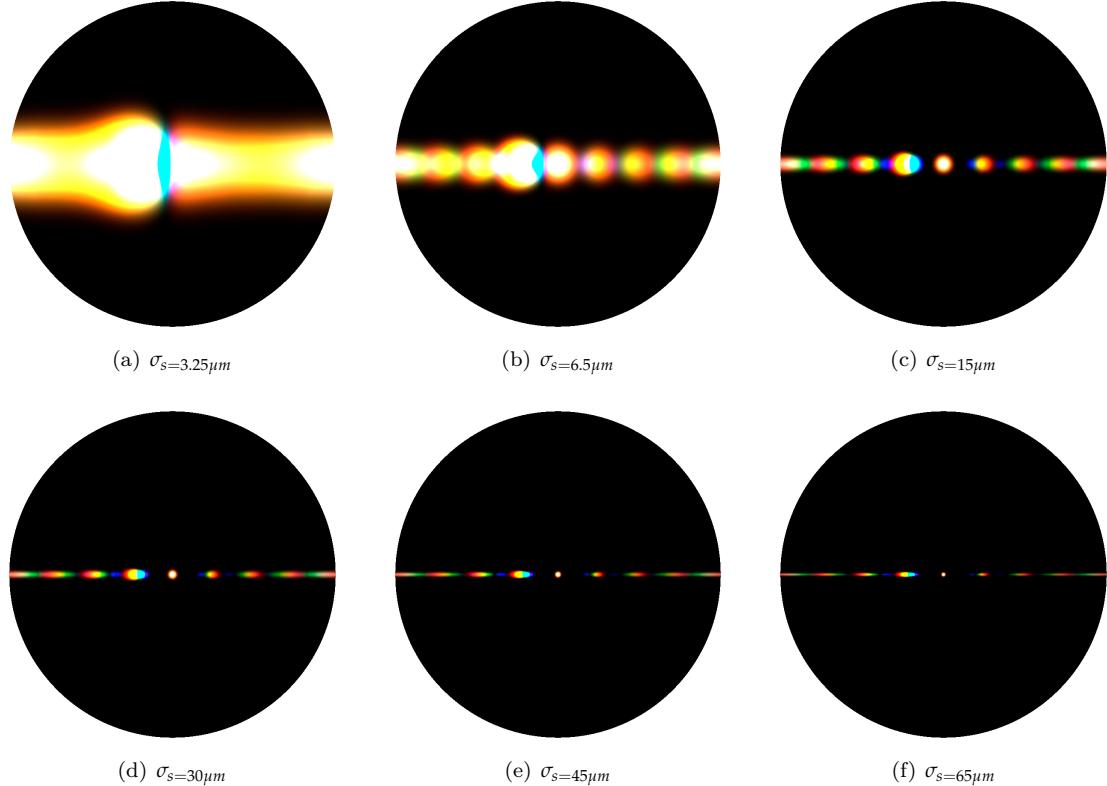


Figure 3.10: Blazed grating with periodicity of $2.5\mu m$: Different σ_s

Figures 3.11 and 3.12 show the BRDF maps for the FLSS approach using a different number of terms N used in the Taylor series approximation. For both input patches we clearly, visually observe the convergence of the taylor series for higher values of N . We visually observe convergence of the Taylor series for all our patches for a very large value of N^3 .

As discussed in section 1.4 there exists a certain value of N for which our approach converges. For all our shading approaches, applied on our gratings, we visually observed a convergence of their BRDF maps when using $N \geq 39$ DFT terms. Furthermore, for a Blazed grating it suffices to use only $N \geq 7$ - and for an Elaphe grating only $N \geq 9$ DFT terms. Notice, that these numbers of required DFT terms were empirically determined with a trial and error strategy.

³Using N equal to 40 lead to visual convergence for all our used gratings.

However, by making use of Taylor error term estimates, as introduced in the appendix section ??, we can derive an upper bound for N . This computation is dependent on many aspects, such as on the grating spacing, the sampling rate dH and the used wavelength spectrum. Thus, it is usually simpler to empirically determine an actual value for N to be used.

In algorithm ?? we compute different powers of DFT terms of our height fields h by evaluating the expression $DFT(h)^n \cdot i^n$, where i denotes the imaginary number i and n is a natural number. Note that when applying the DFT operator on a real or complex number (e.g. such as one particular value of our height field) we get a complex number. Using our Taylor series approximation has basically four possible convergence images, each having its own convergence radius. The reason for this is due to the fact we are multiplying the DFT terms by i^n .

In order to understand the reason behind this, let us consider the complex unit i . Raising i to the power of a natural number n leaves us four possible results as shown in equation 3.2:

$$i^n = \begin{cases} 1 & \text{if } n \equiv_4 0 \\ i & \text{if } n \equiv_4 1 \\ -1 & \text{if } n \equiv_4 2 \\ -i & \text{otherwise} \end{cases} \quad (3.2)$$

Thus multiplying a complex number c , such as particular value of a DFT term, by i raised to a certain power, will permute the the value of c . Since we compute $\sum_{k=0}^N DFT(h)^k \cdot i^k$ we can split this summation into four partial summations, according to their applied power factor i . Each of these summands converges with a different convergence radius.

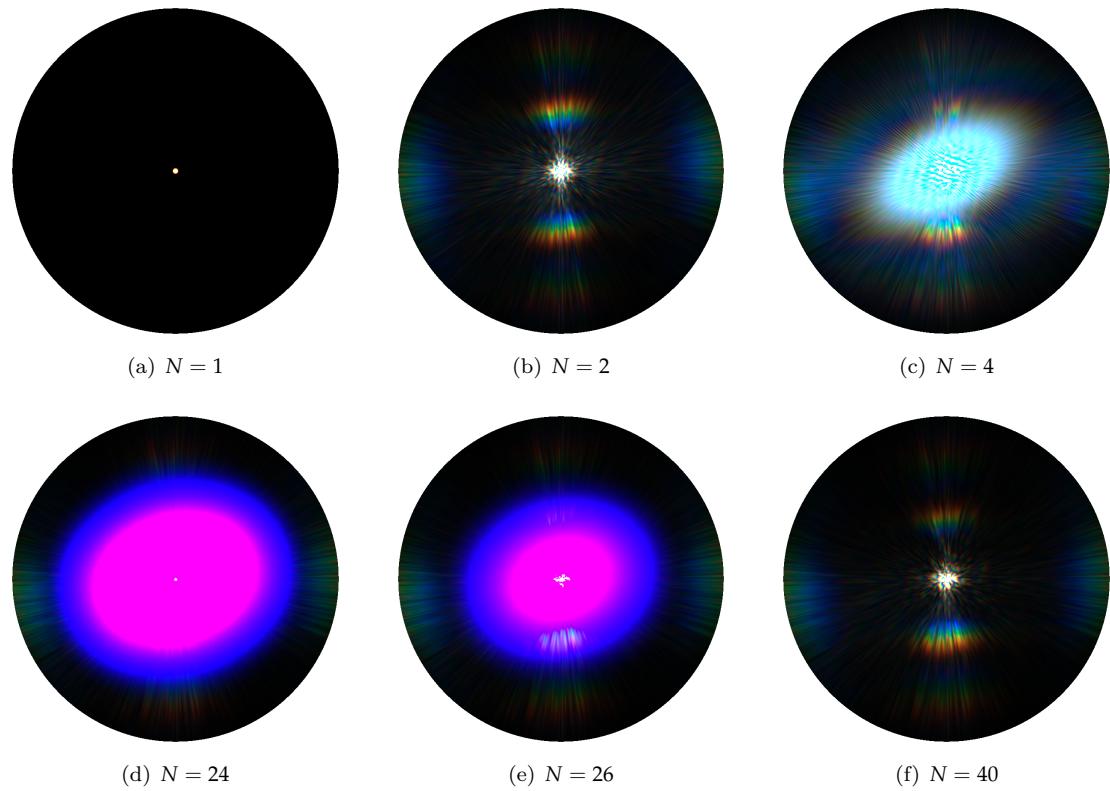


Figure 3.11: Blazed grating at $65\mu m$: N Taylor Iterations

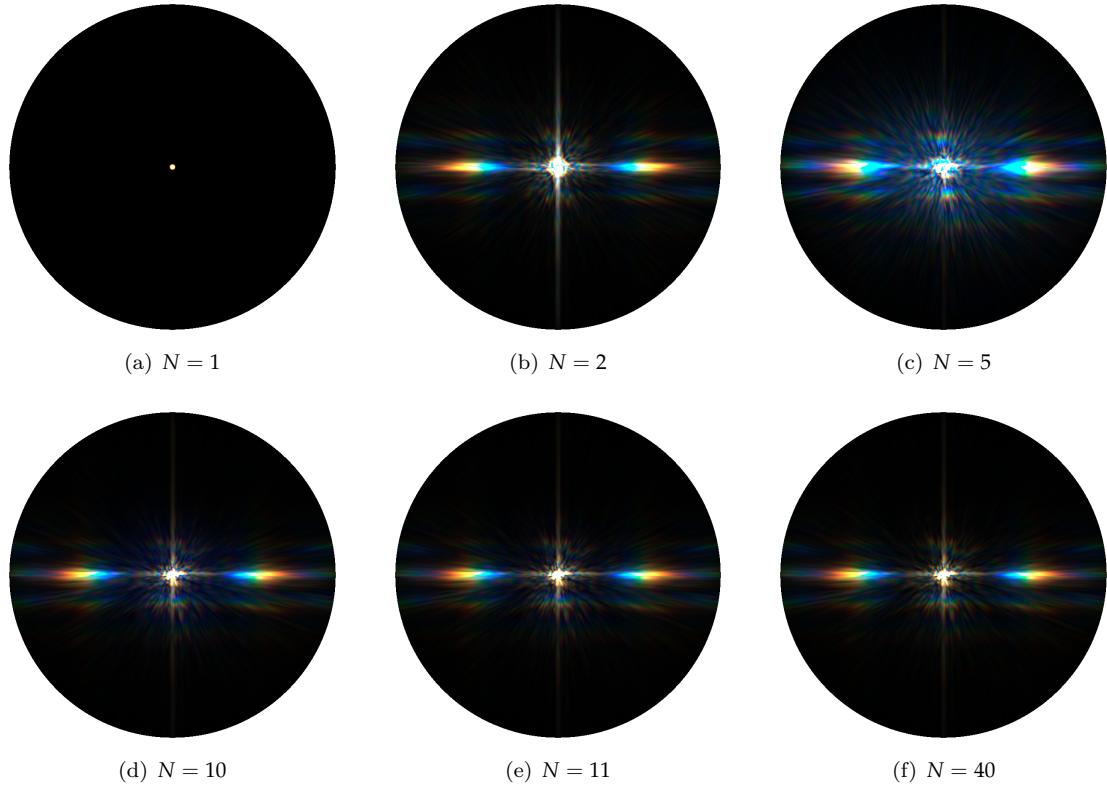
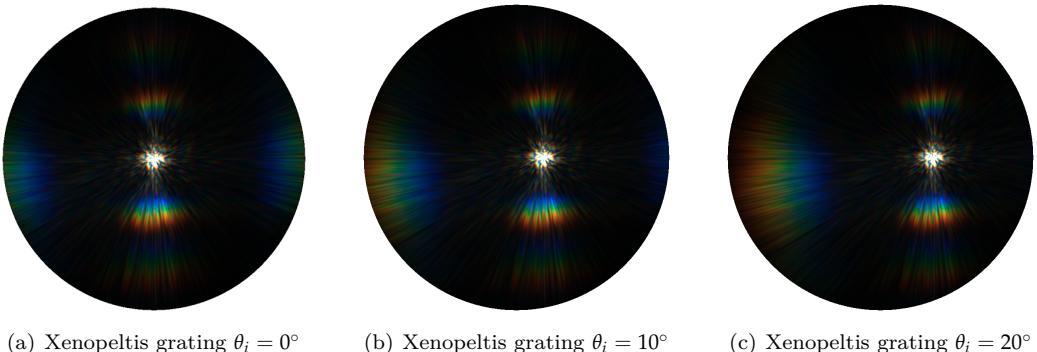
Figure 3.12: Elaphe grating at $65\mu m$: N Taylor Iterations

Figure 3.13 shows the BRDF maps of the FLSS approach applied on the Xenopeltis snake shed, using different θ_i incident angles. When slightly moving the incident angle θ_i , we can observe how the BRDF map changes. For higher values of θ_i we start seeing additional diffraction color contributions on the right side of the BRDF map.

Figure 3.13: BRDF maps for Xenopeltis grating: different θ_i angles

3.2 Rendering Surface Geometries

In this section we are going to present our actual renderings. These renderings are simulating the effects of diffraction produced when a directional light source encounters different nano-scaled surfaces on a given curved snake skin mesh. We will see that diffraction colors change dramatically with changes in the light direction, surface normals and the viewing direction. This is typical for diffraction colors observed in nature. Therefore, the aim of this section is to provide a visual comparison between diffraction patterns occurring in nature as shown in figure ?? and the renderings produced by our shader. Last we compare the results produced by our shader against those produced by Stam's approach. For this purpose we compare applied these methods on a synthetic grating (on a CD) and on a natural grating (on a Xenopeltis grating).

For rendering we are going to rely on our FLSS approach. Unfortunately, this approach is rather slow and can barely be considered as being interactively performing. The NMM approach on the other hand, has interactive runtime. In general, both approaches are valid according to their evaluation plots shown in figure 2.8 and in figure 2.9. Therefore, their rendered results may be considered as being accurate. However, the reason for choosing the FLSS approach instead of the NMM approach is that the renderings resulting from the NMM approach look purplish compared to those produced by the FLSS approach. This color-tone shift towards the purple color region for NMM renderings does not correspond to the reality and is a result of its non-uniform wavelength spectrum sampling. Further information about NMM's tone shift issue and how it can be addressed is discussed in the appendix section *B*.

The Laboratory of Artificial and Natural Evolution in Geneva provided us with a triangular mesh of a snake. This mesh was produced by a 3D scan for an Elaphe snake species and it consists of 11696 vertices and 22950 faces. Note that, for all our renderings, we used this snake mesh.

For the snake species under consideration, their macroscopic geometry is highly similar. Only the geometry of their nano-structures varies and is responsible for a snake's iridescence. Thus, we can use the same snake surface model to render diffraction for different species. Table 3.1 lists the system specifications of the machine I used in order to produce the rendered images.

Processor	Intel i7 CPU 970 @ 3.20 GHz (12 CPUs)
Memory	12288 MB RAM
Graphics Card	GeForce GTX 770
Graphics Clock	1150 MHz
Graphics Memory	4096 MB
Graphics Memory Bandwidth	230.4 GB/s

Table 3.1: Hardware specifications of the machine used to render snake surface. Statistics are provided using the tool *NVIDIA GeForce Experience*.

Figure 3.14 shows renderings produced by the FLSS approach applied on our snake mesh for different, given input patches. Due to the fact that a Blazed grating has its maximum intensity for a certain direction and the geometry of the snake mesh is curved i.e. it is non-flat, we can expect rather less diffraction color contribution as shown in figure 3.14(*b*).

In contrast, for both the renderings, we see colorful patterns on the skin of our snake species, Elaphe and Xenopeltis, due to the effect of diffraction. We see much less colorful patterns for

Elaphe as shown in figure 3.14(b) than for Xenopeltis as shown in figure 3.14(c). This is consistent with the observations in the real world as shown in figure ???. As observable in figure 3.2(b), the substructures (the finger like structures) in the height field of a Elaphe snake skin overlap (i.e are not very regularly aligned) along the y-axis. This is why the Elaphe species is less iridescent than the other specie. The Xenopeltis snake has a brownish body with no pigmentation, which makes the iridescence more spectacular than on Elaphe as seen in figure ??.

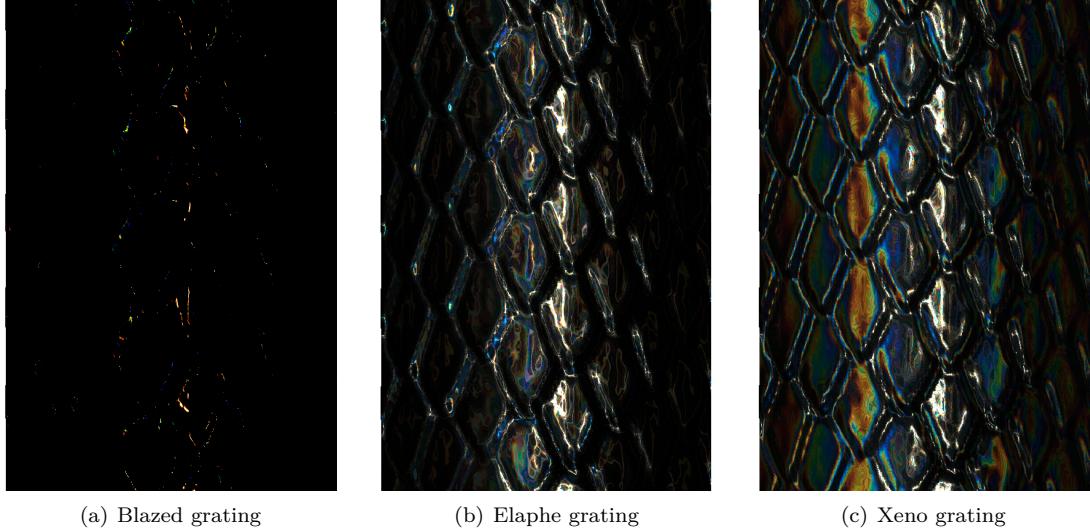


Figure 3.14: Diffraction of different snake skin gratings rendered on a snake geometry

Figure 3.15 shows a set of subfigures for rendering the effect of diffraction produced by the FLSS approach (used as our reference approach), applied on our snake mesh using the Elaphe nano structure. Figure 3.15(b) shows the final diffraction color contribution result with texture-blending. We note that the diffraction color contribution is not significant in this subfigure which resembles quite well to the reality as shown in figure ???. In subfigure 3.15(d) we see the light cone in order to show the direction of the light source besides the rendered results. Subfigure 3.15(e) is a sample visualization of a Fourier transformation for an Elaphe nano-scale surface structure as shown in figure 3.15(d).

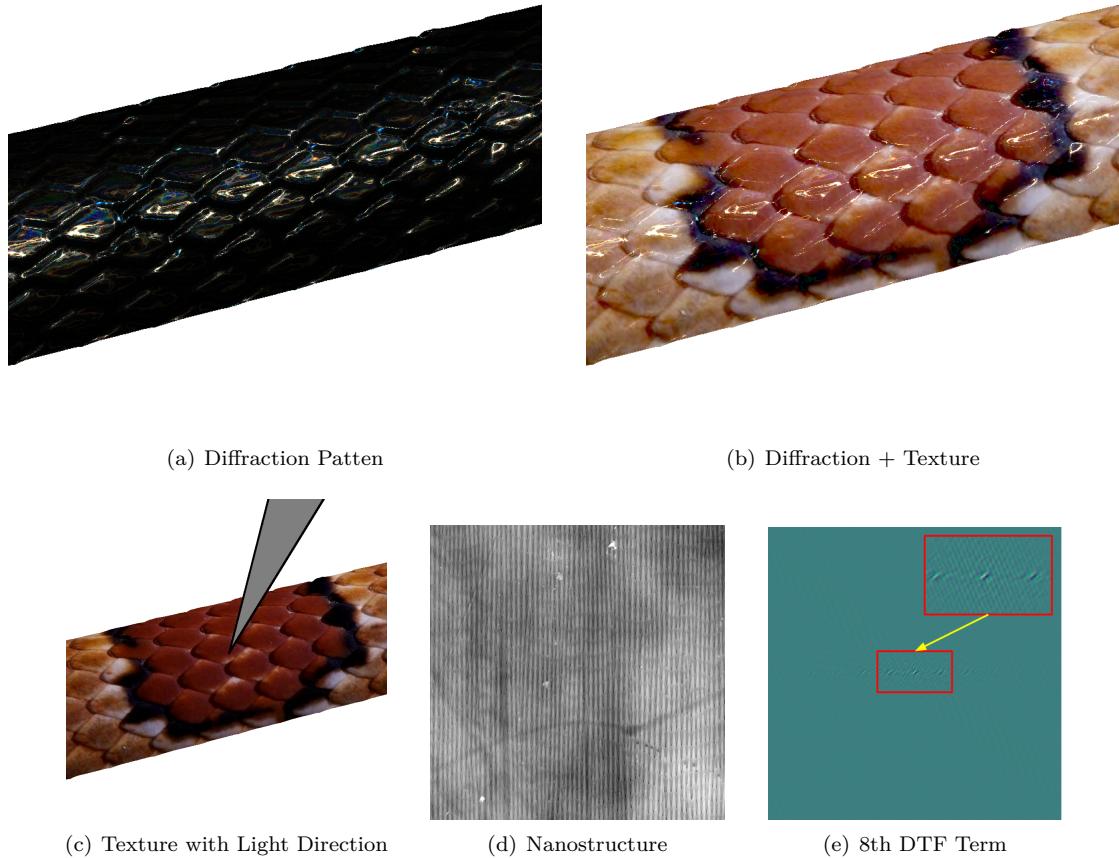
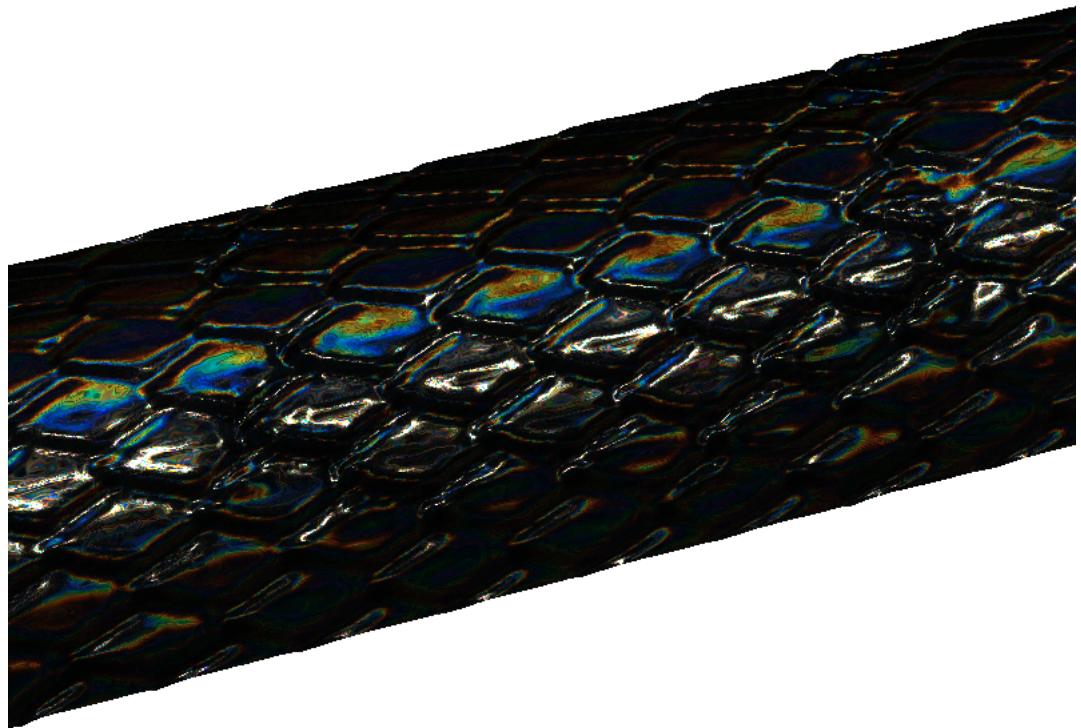
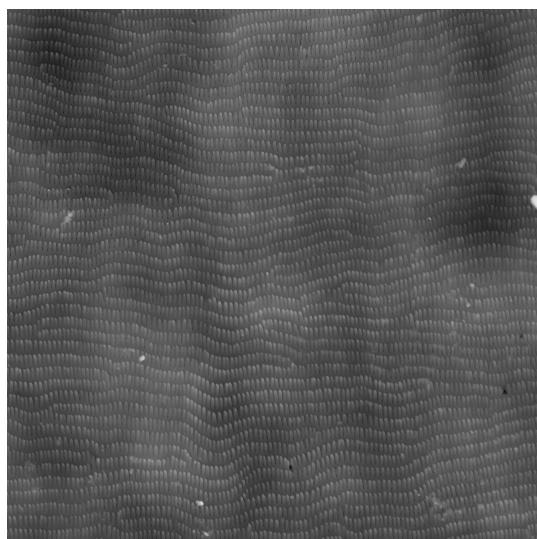


Figure 3.15: Diffraction for Elaphe snake skin produced by our reference approach.

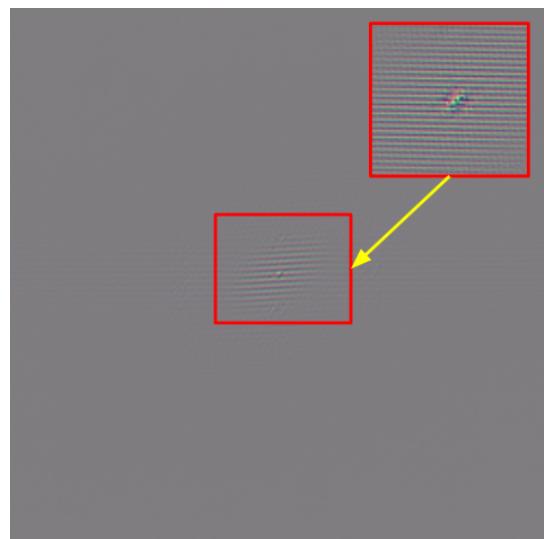
Figure 3.16 shows a set of subfigures for the effect of diffraction for the Xenopeltis snake surface. In our Xenopeltis rendering as shown in figure 3.16(c) when using a grating as shown in figure 3.16(b) we see a lot of iridescent color contribution. Comparing this to a real image 3.16(a) we notice much resemblance regarding the reflectance strength and colorful pattern.



(a) Diffraction Patten



(b) Nanostructure



(c) 3th DTF Term

Figure 3.16: Diffraction for Xenopeltis snake skin produced by our reference approach.

Figure 3.17 shows the diffraction pattern for an Elaphe snake shed at different zoom levels for a fixed incident light and a fixed viewing direction. We changed the zoom-levels by adjusting the field of view angle of our camera. For each image in this figure, the one to its right side is a five times zoomed-in version of the region within its red box. The close up perspectives exhibit complex and colorful diffraction patterns.

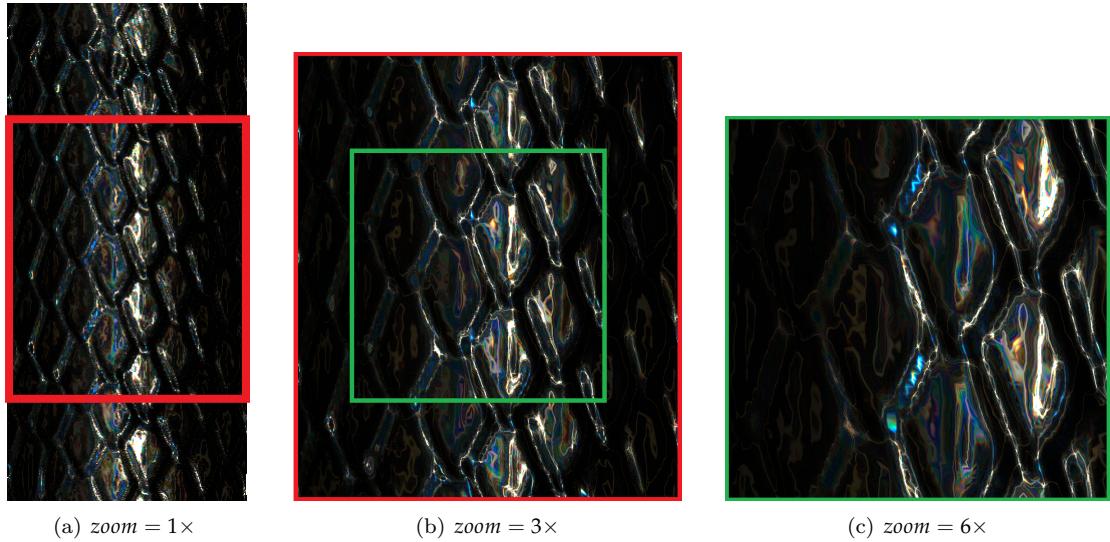


Figure 3.17: Showing the diffraction pattern on an Elaphe grating for different camera zoom levels by varying the field of view.

Figure 3.18 shows how the diffraction pattern changes when the incident light direction is moved slightly. This figure gives us an impression of what kind of complex, perspective-dependent pattern the diffraction phenomenon produces. The figure column from figure 3.18(a) to figure 3.18(e) models a scene in which a virtual sun (i.e. a directional light source) moves from south (S) to north (N). For each of this subfigure we can see in its inset (red box) how the diffraction color changes according to how the position of the sun changes. The figure column from figure 3.18(b) to figure 3.18(f) models a scenario when a sun moves from south-east (SE) to west (W). Similarly, the diffraction patterns changes according to the position of the sun.

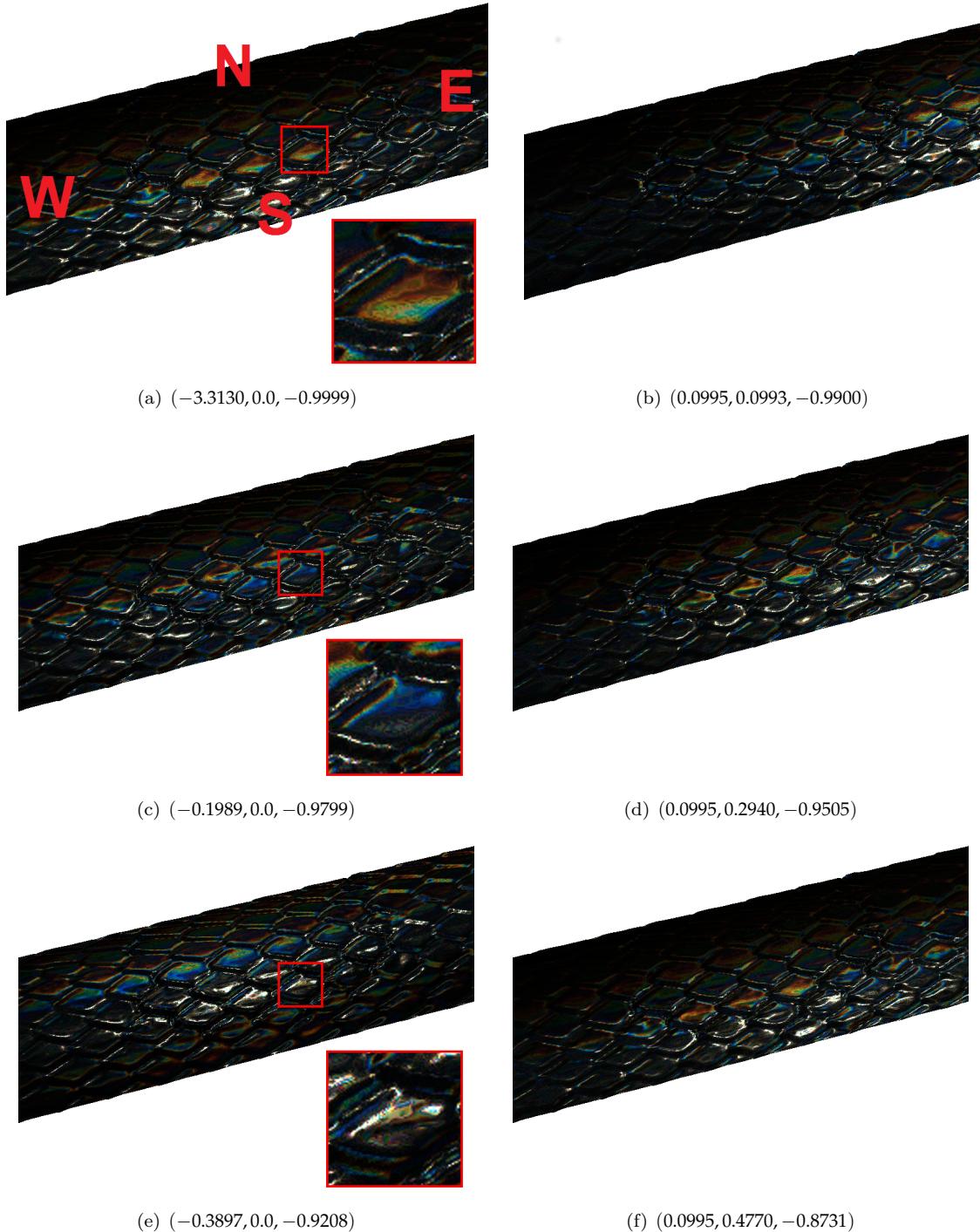


Figure 3.18: Showing the change of the diffraction pattern on an Elaphe grating for different light directions. Figure 3.18(a) shows the geographic direction in order to describe a local orientation. The first subfigure column the sun moves from south (S) to north (N). In the second column, the sun moves from south-east to west (W).

Figure 3.19 shows a photo of an experimental setup for demonstrating the effect of diffraction using a Elaphe snake grating. The exact parameters for the experimental setup are unknown. Nevertheless this image gives us an impression of how close our model is to the reality comparing it with our simulated results since we notice similar diffraction patterns for our simulated results using an Elaphe snake shed.



Figure 3.19: Diffraction Elaphe: Pattern for a sample setup

In the following a brief comparison between Stam's⁴ and our FLSS method using two different kinds of gratings, a synthetic, regularly aligned grating and a natural, complex structured grating. These gratings are shown in figure 3.20.

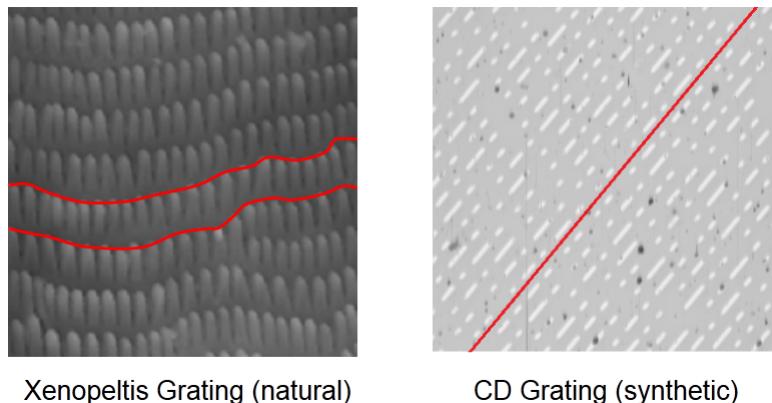


Figure 3.20: Alignment of nano-structures in diffraction gratings. On the left a complex, natural grating of the Elaphe snake species and on the right a synthetic, very regularly aligned grating of a CD.

Figure 3.21 shows an example of a case where Stam's approach performs well. Considering the red-line in the figure we notice that the nano-scaled structures of a compact disc are very regularly

⁴A reference implementation of Stam's Diffraction Shader[Sta99] is provided by Nvidia's GPU Gems at http://http.developer.nvidia.com/GPUGems/gpugems_08.html

aligned along the surface. Tracks of a CD are uniformly spaced and bumps along a track are distributed according to a poisson distribution⁵. We notice that the diffraction patterns in both approaches look similar.

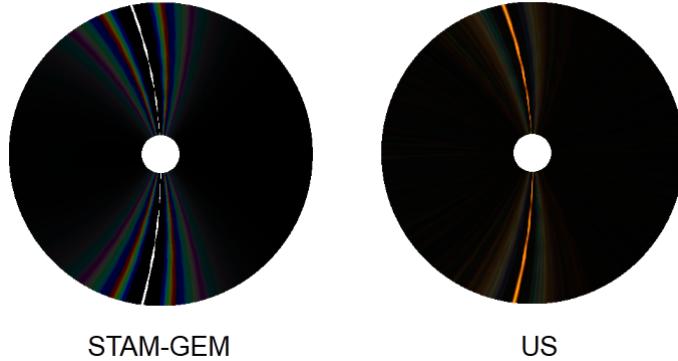
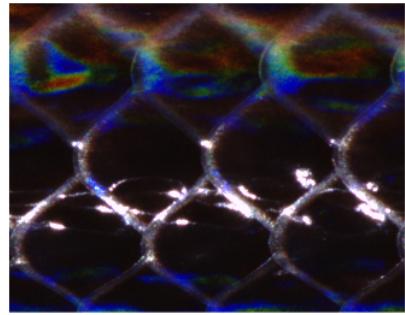


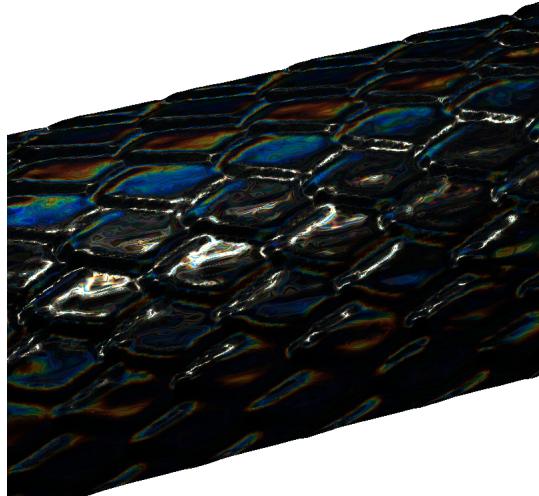
Figure 3.21: Comparison of our approach against a reference implementation of Stam’s method provided by Nvidia Gem. For synthetic diffraction gratings, which have a very regular structure, Stam’s approach is doing well. Both approaches produce similar diffraction patterns.

Finally, tried to reproduce a real Xenopeltis image (as shown in figure 3.22(a)) for a unknown light and viewing direction. For this purpose I used our FLSS rendering approach and compared its results against Nvidia Gem’s implementation. The results are shown in figure 3.22. Even the results of Stam’s method look convincing, they have some issues. In some regions in his renderings close to specular regions, there are missing colors. The color distribution in renderings produces by his approach is rather discrete in Stam’s. Also notice that Stam’s approach always produces a symmetric reflectance map (i.e. BRDF map) exhibiting results in regions where it must not.

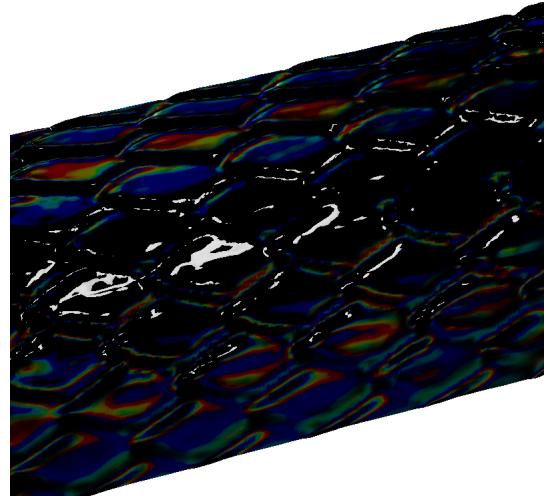
⁵See http://en.wikipedia.org/wiki/Poisson_distribution



(a) Real Xenopeltis Photograph



(b) FLSS



(c) Stam

Figure 3.22: Comparison of our FLSS approach(see figure 3.22(b)) against a reference implementation of Stam’s method(see figure 3.22(c)) provided by Nvidia Gem. We attempt to reproduce a real Xenopeltis skin coloration(see figure 3.22(a)). For natural diffraction gratings, which have a rather complex structure, Stam’s approach is qualitatively different from the real image.

Appendix A

Evaluation Data Generation

The aim of this appendix chapter is to provide some further information and insight about how we generated the data for our evaluation graphs and how we plotted them discussed in chapter 5. For this purpose we first explain how we discretized the input data in order to compute the BRDF values. This is followed by discussing the algorithm how we plotted our graphs as illustrated in algorithm 1.

In our evaluation Java program (as mentioned in section 2.3.1) we fixed the incident light angle θ_i^1 . This allows us to remove the argument θ_i from our BRDF function. Our java program computes each BRDF function at a given discrete wavelength-viewing-angle grid, denoted by $[\Lambda, \Theta]$. The wavelength space $\Lambda = [\lambda_{min}, \lambda_{max}]$ and the viewing angle range $\Theta = [\alpha_{min}, \alpha_{max}]$ of our free parameter θ_r are discretized using equidistant steps. The step sizes, denoted by $(\lambda_{step}, \alpha_{step})$, are provided as input arguments for our Java evaluation program.

Next, let us have a closer look at how our discrete $[\Lambda, \Theta]$ grid is constructed. The wavelength space Λ , which is ranging from λ_{min} to λ_{max} , is discretized like the following:

$$\Lambda = \{\lambda = \lambda_{min} + k \cdot \lambda_{step} | k \in \{0, \dots, steps_\lambda - 1\}\} \quad (\text{A.1})$$

where $steps_\lambda = \lceil \frac{\lambda_{max} - \lambda_{min}}{\lambda_{step}} \rceil$. We similarly discretise the viewing angle space Θ by setting a minimal and maximal viewing-angle boundary α_{min} and α_{max} . Then $\lceil \frac{\alpha_{max} - \alpha_{min}}{\alpha_{step}} \rceil$ is the number of angle $steps_\alpha$. And thus, our Θ space it defined like the following:

$$\Theta = \{\alpha = \alpha_{min} + k \cdot \alpha_{step} | k \in \{0, \dots, steps_\alpha - 1\}\} \quad (\text{A.2})$$

Then, every BRDF java function is applied to the grid $[\Lambda, \Theta]$ and the resulting spectral response is stored in a matrix

$$R = \{BRDF(\lambda_i, \theta_r^j) | i \in Index(\Lambda), j \in Index(\Theta)\} \quad (\text{A.3})$$

The generation process of the evaluation plots, which we discuss in section 2.3.2, is described in algorithm 1. This algorithm takes the matrix R from equation A.3 as input argument. For the maximal reflectance of our methods at any wavelength, it computes the corresponding peak viewing angles and compares it to the angle resulting from the grating equation.

¹In practice, we set the θ_i equal to 75°.

Algorithm 1 BRDF Evaluation Graph Plotter

Input: R Matrix with *BRDF* intensity values of (Λ, Θ) grid
 Λ discretized wavelength space used to compute R
 (α_{min}) minimum value of viewing angle space Θ
 (α_{step}) discretization level of viewing angle space Θ
 d estimated periodicity of height field
 θ_i fixed incident angle

Procedure: $getMaxIntensGridPointsOf(matrix, r)$: get the column-index of the largest
 intensity value in the row $matrix(r, *)$

$plotPoint(x, y)$: draw a point at (x, y)

Output: Evaluation Plot of given BRDF model applied on given height field

```

1: Foreach  $\lambda_k \in \Lambda$  do
2:    $\tilde{\alpha} = getMaxIntensGridPointsOf(R, \lambda_k)$                                  $\triangleright \tilde{\alpha} \equiv$  index viewing angle of max. R
3:    $\tilde{\alpha}_{r_k} = \alpha_{min} + \alpha_{step} \cdot \tilde{\alpha}$ 
4:    $\theta_{r_k} = \arcsin\left(\frac{\lambda}{d} - \sin(\theta_i)\right)$                           $\triangleright$  graph resulting from our BRDF model
5:    $plotPoint(\lambda_k, \tilde{\alpha}_{r_k})$                                                $\triangleright$  graph resulting from grating equation
6:    $plotPoint(\lambda_k, \theta_{r_k})$ 
7: end for

```

Algorithm 1 iterates over the wavelength space Λ and generates our evaluation plots. For any wavelength it computes the viewing angle with maximum reflectance and the angle resulting from the grating equation as defined in equation 2.2. Both angles are then plotted for the current wavelength in the current iteration. In the next section we will discuss the generated evaluation plots.

Appendix B

Rendering Surface Geometries using the NMM approach

In section 3.2 we mentioned that we use the FLSS approach instead of the NMM approach for producing our renderings applied on a snake mesh. The reason for choosing the FLSS approach was that it produces reliable results (according to its evaluation plots discussed in section 2.3.2). Furthermore, the renderings resulting from the NMM approach look purplish compared to those produced by the FLSS approach as shown in figure B.1. This figure shows renderings of our snake mesh when using an Elaphe grating produced by the FLSS approach (figure B.1(a)) and the NMM approach (figure B.1(b)). We observe that pixels, which have a bluish color tone in FLSS renderings exhibit a purplish color tone in the corresponding pixel in a NMM rendering. This color-tone shift towards the purple color region for NMM renderings does not correspond to the reality and is a result of its non-uniform wavelength spectrum sampling.

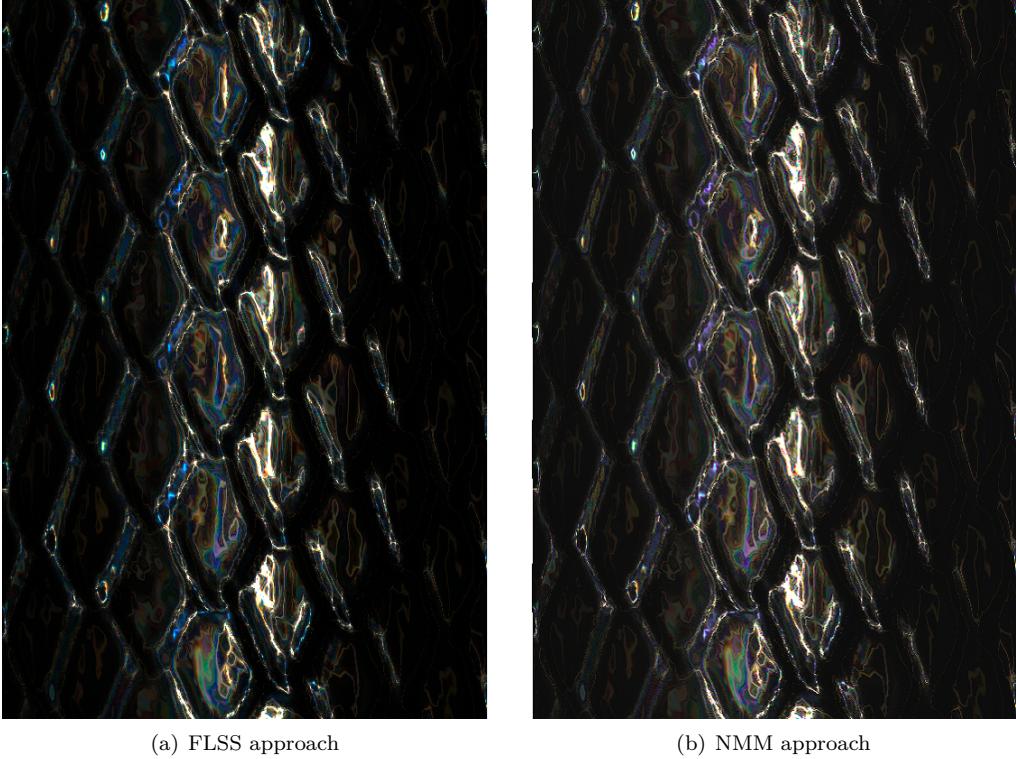


Figure B.1: Comparing the FLSS rendering approach by the NMM approach by rendering an Elaphe grating.

In order to address this color-tone issue let us revisit the idea of how we actually compute our color values. For this purpose let us consider the equation 1.18 which tells us how to compute CIE XYZ color values. One particular example is the computation of the luminance Y which is equal to:

$$Y = \int_{\Lambda} L_{\lambda}(\omega_r) \bar{y}(\lambda) d\lambda$$

In this formulation we integrate over the whole wavelength spectrum Λ in order to compute an actual color value for Y . However, in the NMM approach we perform a *non-uniform* integration over the wavelength spectrum. Instead of directly integrating over the wavelength spectrum we integrate uniformly over the minimum and maximum wavenumber as explained in section ???. Thus, we no longer integrate over the wavelength spectrum and we rather integrate over the corresponding wavenumber range. Hence, this depicts a change of integration variables. Unfortunately, I have not taken care of this factor in the NMM approach. In the following I describe what this factor is equal to.

The wavenumber for a given wavelength λ is equal to

$$k = \frac{2\pi}{\lambda} \quad (\text{B.1})$$

For the NMM approach integrate over dk instead over $d\lambda$. By rearranging the definition of the wavenumber k we get the following identitiy for the wavelength:

$$\lambda = \frac{2\pi}{k}$$

thus, the integrations variable change factor between $d\lambda$ and dk can be computed as the following:

$$\begin{aligned} \frac{d\lambda}{dk} &= \frac{d}{dk} \left(\frac{2\pi}{k} \right) = \frac{2\pi}{k^2} \\ \Rightarrow d\lambda &= \frac{2\pi}{k^2} dk \end{aligned}$$

This will lead us to the final representation for intgrating over the wavenumber range:

$$\begin{aligned} Y &= \int_{\Lambda} L_{\lambda}(\omega_r) \bar{y}(\lambda) d\lambda \\ &= \int_{N_{min}}^{N_{max}} L_k(\omega_r) \bar{y}(k) \frac{2\pi}{k^2} dk \end{aligned}$$

E.g. in order to compute the lumminance Y , we h
there is another issue - weighting

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. Dachsbaucher* (2014), January
- [DM12] DENNIS M, Sullivan: *Quantum Mechanics for Electrical Engineers*. John Wiley Sons, 2012, 2012. – ISBN 978-0470874097
- [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: Structurnal 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