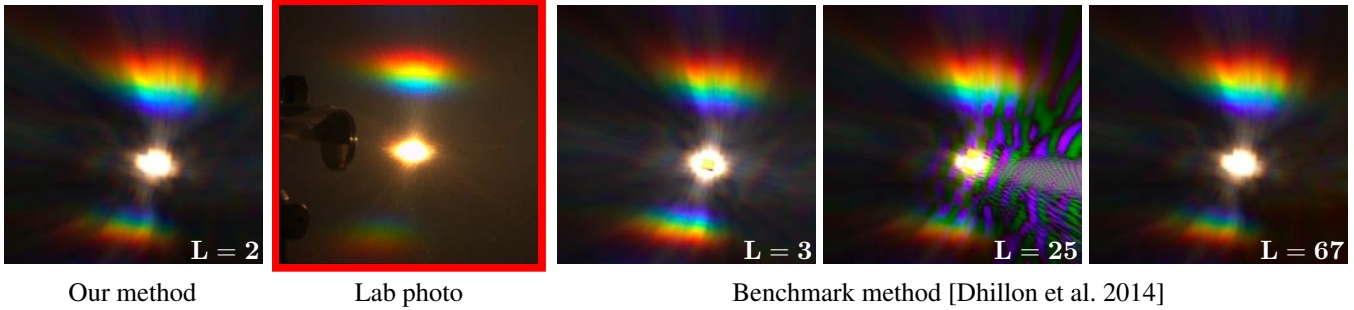


# Efficient Surface Diffraction Renderings with Chebyshev Approximations

Daljit Singh J. Dhillon\* and Abhijeet Ghosh†  
Imperial College London



**Figure 1:** Diffraction pattern seen on a screen due to a beam incident on a gold plated Sunbeam snakeskin patch (setup and lab photo from [Dhillon et al. 2014]). Rendering with our proposed method (left) qualitatively well matches the photograph (center left) with just two lookup tables including reproducing the characteristic asymmetry in the diffraction lobes. In comparison, Dhillon et al.’s method (right) requires a large number of tables for accurate rendering.

## Abstract

We propose an efficient method for reproducing diffraction colours on natural surfaces with complex nanostructures that can be represented as height-fields. Our method employs Chebyshev approximations to accurately model view-dependent iridescences for such a surface into its spectral bidirectional reflectance distribution function (BRDF). As main contribution, our method significantly reduces the runtime memory footprint from precomputed lookup tables without compromising photorealism. Our accuracy is comparable with current state-of-the-art methods and better at equal memory usage. Furthermore, a Chebyshev polynomial basis set with its near-best approximation properties allow for scalable memory-vs-performance trade-offs. We show realistic diffraction effects with just two lookup textures for natural, quasi-periodic surface nanostructures. Performance intensive applications like games and VR can benefit from our method, especially for low-end GPU or mobile platforms.

**Keywords:** Interactive rendering, diffraction, iridescence, structural colours, spectral BRDFs, shader, photorealism

## 1 Introduction

In computer graphics, photorealism is indispensable for virtual reality simulations and natural objects pose several challenges to it. We focus on one such challenge that deals with the rendering of natural surfaces exhibiting complex, colourful and iridescent appearances due to light diffraction. Common examples of it include snake skins, insect wings, certain gems and shells (shown below). Diffraction colours in nature are often hard to simulate since they are produced by complex surface nanostructures and



modelling them involves scalar wave optics. Most state-of-the-art diffraction shaders focus only on the speed and accuracy of their algorithm while ignoring their GPU memory footprints. In this paper, we propose a practicable method that is interactive, accurate and memory efficient, all at the same time, even for low-end GPU platforms.

Starting with Stam [1999], several authors proposed interactive methods for rendering far-field diffraction effects using spectral BRDFs. These methods often represent surface nanostructures as height-fields. Most of these methods either work only with simple or statistically known height-fields [Stam 1999; Wu and Zheng 2013] or compromise accuracy for interactivity by overly simplifying their models [Imura et al. 2009; Lindsay and Agu 2006]. On the other hand, highly accurate models have been proposed based on Wigner distribution functions (WDFs) [Cuypers et al. 2012] and finite difference time domain (FDTD) techniques [Musbach et al. 2013]. But these methods are not interactive for complex height-fields representing natural nanostructures. Recently, Dhillon et al. [2014] proposed a pre-computed lookup tables based method which is interactive as well as highly accurate for generic, discrete height-fields. However, with too few lookup tables it does not model some interesting aspects such as asymmetries in the diffraction pattern and adding just a few more tables results in color streak artifacts (see Figure 1 for an example). For convergence to accurate results, it requires a large number of lookup tables.

Practically, we need a method which provides a good trade-off between memory and performance (accuracy and speed). Also, performance improvements must scale with increased memory usage. To deliver these goals, we devise a method that employs Chebyshev approximations for modelling spectral BRDFs in a diffraction shader. For a given height-field, based on Fourier optics, we precompute coefficients to linearly combine Chebyshev basis polynomials to compute diffraction colours in CIE X,Y,Z components. Our basis polynomials depend only on the half-vector from view geometry. They can be easily generated and linearly combined at runtime. In summary, our main contributions is: A Chebyshev approximation based algorithm to produce realistic renderings for measured height-fields with a scalable trade-off between memory and performance.

We show with a natural, quasi-periodic, measured height-field from

\*†email: djdhillon@gmail.com, ghosh@imperial.ac.uk

**Table 1: Notations and formulae in  $\llcorner\llcorner$ term: description $\lrcorner\lrcorner$  format**

$\mathbf{p}$ : surface point under observation	$\hat{n}$ : unit local surface normal and Z axis
$m, n$ : discrete $X, Y$ spatial coordinates	$h(m, n)$ : discretised height-field
$\hat{\omega}_i$ : unit vector pointing at light source	$\hat{\omega}_v$ : unit vector pointing at viewpoint
$\hat{\omega}_h = \hat{\omega}_i + \hat{\omega}_v$ : half-vector	$(u, v, w) = -\hat{\omega}_h$ : lookup coordinates
$\lambda$ : wavelength for a given light-wave	$f_\lambda(\hat{\omega}_i, \hat{\omega}_v)$ : BRDF for spectral wave $\lambda$
$F$ : Fresnel term	DFT: discrete Fourier transform operator
$s, t$ : discrete $U, V$ frequency coordinates	$\xi_s, \xi_t$ : ordinary frequencies at $(s, t)$
$\sigma_w$ : spatial coherence length for light	$\xi_f = \frac{1}{2\pi\sigma_w}$ : spectral coherence length
$p_\lambda(m, n) = e^{i2\pi w h(m, n)/\lambda}$ : a phasor-like auxiliary function	
$G(\hat{\omega}_i, \hat{\omega}_v) = (1 + \hat{\omega}_i \cdot \hat{\omega}_v)^2 / (\hat{n} \cdot \hat{\omega}_i)(\hat{n} \cdot \hat{\omega}_v)$ : geometric attenuation	
$W(u, v, s, t, \lambda) = e^{-[(u/\lambda - \xi_s)^2 + (v/\lambda - \xi_t)^2]/2\xi_f^2}$ : coherence window	

a snakeskin that using just two lookup tables can produce realistic appearances. Furthermore, we also show that our approximations work equally well even with seemingly random height-fields.

## 2 Data-Driven Spectral BRDF Model

Consider a measured, discrete height-field  $h(m, n)$  representing given surface nanostructures. Based on Fourier optics and coherence properties of light, Dhillon et al. [2014] formulate its spectral BRDF. Their BRDF formulation is normalised against mirror reflections for a reference beam of normal incident white-light. Table 1 introduces all notations that are required to understand the following. We reformulate Equations (4), (6), (7) and (8) from Dhillon et al. [2014] to express their normalised spectral BRDF as

$$f_\lambda(\hat{\omega}_i, \hat{\omega}_v) = C(\hat{\omega}_i, \hat{\omega}_v) |S_\lambda(\hat{\omega}_i, \hat{\omega}_v)|^2 \quad (1)$$

with  $S_\lambda(\hat{\omega}_i, \hat{\omega}_v) = \sum_{s,t} \mathcal{F}_\lambda(\xi_s, \xi_t) W(u, v, s, t, \lambda)$ . (2)

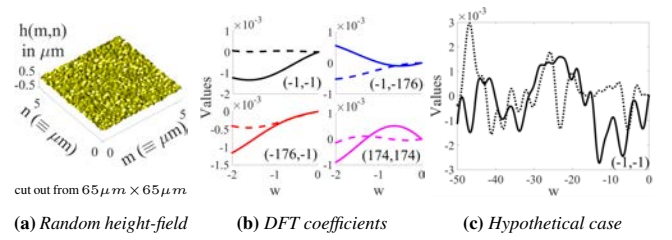
Here,  $C(\hat{\omega}_i, \hat{\omega}_v) = F^2 G / F_0^2 w^2$  with  $F_0$  as the Fresnel term for mirror reflection of the normal incident beam.  $\mathcal{F}_\lambda \equiv \text{DFT}\{p_\lambda(m, n)\}$  and  $W$  is a 2D Gaussian for light coherence, in the frequency domain. Please refer to Table 1 again for other notational details and our supplemental material for further technical details.

Dhillon et al. [2014] implement Equation (2) with a Taylor series expansion for the auxiliary function  $p_\lambda(m, n)$ . This is analogous to a Fourier series approximation of the BRDFs defined by Stam [1999]. Fourier series approximations suffer from Gibbs phenomena and until sufficiently large number of Taylor series terms are not used, their method exhibits disturbing visual artifacts. We propose a new approach to better approximate spectral BRDFs with fewer lookup tables that completely avoid such artifacts.

## 3 Chebyshev Approximation

The key challenge in evaluating Equation (2) involves repeating DFT computations at each viewed point  $\mathbf{p}$ . This view-dependency arises from the  $w$ -term in the definition for operand  $p_\lambda(m, n)$  to DFT.  $w$  is the projection of inverted half-vector  $-\hat{\omega}_h$  onto local surface normal  $\hat{n}$ . Our main idea is to directly model these  $w$ -dependent variations in each DFT coefficient with a separate polynomial approximation.

Consider a discretised, nano-scale height-field  $h$  as shown in Figure 2a. To keep things as general as possible, we choose  $h$  to be a random field. With a Taylor series expansion for the corresponding  $p_\lambda(m, n)$ , each DFT coefficient  $\mathcal{F}_\lambda(\xi_s, \xi_t)$  can be expressed as a complex polynomial in  $w$  [Dhillon et al. 2014, Equation (7)]. We studied  $w$ -dependent variations in several  $\mathcal{F}_\lambda(\xi_s, \xi_t)$  coefficients for our  $h(m, n)$  to find that most of them vary as smooth, slowly varying functions of  $w$ . Figure 2b illustrates this fact for a few of them. We cite three points that apparently explain these observations.



**Figure 2:  $w$ -dependent variations in DFT coefficients for our random height-field. With patch-size of  $700 \times 700$  pixels,  $(s, t)$  for each coefficient is indicated at bottom-right of its graph. The real and imaginary parts are shown in solid and dashed lines, respectively.**

For our experiments: (a)  $|w|$  has a small range  $[0, 2]$ , (b)  $\lambda$  for visible spectrum is in the range  $[380, 780]$ nm with a scale factor of  $\approx 2$  between the bounds, and (c)  $p_\lambda$  is a complex exponentiation of a band-limited, periodic function  $h$  and this bounds the magnitude of its discrete Fourier transform. Without above three points, a DFT coefficient can have high fluctuations as Figure 2c shows with an unreal, large range for  $w$ . Next, we performed elaborate studies with several different height-fields including natural measured nanostructures to empirically establish that smooth, slow varying nature is a general trend for DFT terms in Equation (2). In short, for our problem,  $\mathcal{F}_\lambda(\xi_s, \xi_t)$  can be approximated to a high-level of accuracy with only few higher order terms in the polynomial function of  $w$ .

We now propose to model DFT coefficients in Equation (2) with Chebyshev polynomial basis functions of the first kind. We explain the reasons for this choice in Section 3.2. Without loss of generality, with  $w \in [-2, 0]$ , we define DFT coefficients as functions of  $w$  with

$$\mathcal{F}_\lambda(\xi_s, \xi_t) = A_\downarrow^j + iB_\downarrow^j + \sum_{m=0}^M (A_\uparrow^j a_m^j + iB_\uparrow^j b_m^j) T_m(w). \quad (3)$$

Here  $\mathbf{j} = (s, t, \lambda)$  refers to a specific frequency point.  $[A_\downarrow^j, A_\uparrow^j]$  and  $[B_\downarrow^j, B_\uparrow^j]$  express ranges for real and imaginary parts of all DFT coefficients at  $(s, t)$  for given height-field  $h$  with incident wavelength  $\lambda$ . More importantly,  $T_m$  is the  $m$ -th order Chebyshev polynomial of the first kind and  $a_m^j, b_m^j$  are the corresponding Chebyshev coefficients for the polynomial fit.  $M$  represents the highest polynomial order used in a specific approximation. For a given height-field  $h$ , at each  $\mathbf{j}$ , we compute  $\mathcal{F}_\lambda(\xi_s, \xi_t)$  with different values of  $w$ . We then fit a linear combination of Chebyshev basis polynomial to generated  $\mathcal{F}_\lambda$  values to estimate all above terms. With our proposed model, the spectral power for  $S_\lambda$  in Equation (1) can be formulated with substitutions from Equations (2) and (3) as,

$$|S_\lambda(\hat{\omega}_i, \hat{\omega}_v)|^2 = S_\lambda S_\lambda^* = \sum_{m=0}^M t_m(u, v, \lambda) T_m(w). \quad (4)$$

The consolidated Chebyshev coefficients  $t_m$  are defined in terms of previously computed  $A_\downarrow^j, A_\uparrow^j, B_\downarrow^j, B_\uparrow^j, a_m^j$  and  $b_m^j$  from Equation (3) and a known coherence window  $W$  from Equation (2). Mathematical derivation and form for  $t_m$  terms is complex and left out for the sake of brevity. For a given pair  $(u, v)$ , we can precompute  $t_m(u, v, \lambda), \forall \lambda$ . Next, we integrate  $t_m$  terms over  $\lambda$  to tabulate final Chebyshev coefficients for lookup.

### 3.1 Interactive Rendering

For interactive rendering, we substitute BRDF formulations from Equations (1) and (4) into a general spectral rendering equation. Let  $I$  be the spectral power distribution in relative units (RU) [Dhillon

et al. 2014]. Let  $\Gamma_X, \Gamma_Y, \Gamma_Z$  be the respective CIE X, Y, Z colour component filters. The rendering equation then gives:

$$Y(u, v, w) = \int_{\lambda} f_{\lambda}(\hat{\omega}_i, \hat{\omega}_v)(\hat{n} \cdot \hat{\omega}_i) I(\lambda) \Gamma_Y(\lambda) d\lambda, \quad (5)$$

$$= C(\hat{\omega}_i, \hat{\omega}_v)(\hat{n} \cdot \hat{\omega}_i) \sum_{m=0}^M c_m^Y(u, v) T_m(w), \quad (6)$$

$$\text{with } c_m^Y(u, v) = \int_{\lambda} t_m(u, v, \lambda) I(\lambda) \Gamma_Y(\lambda) d\lambda. \quad (7)$$

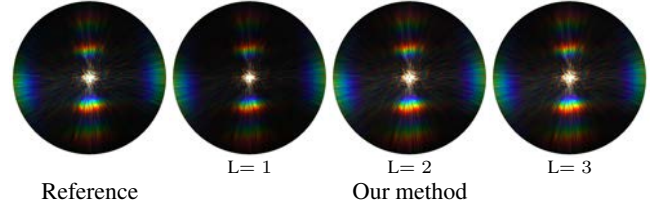
Similarly we obtain  $X$  and  $Z$  in terms of  $c_m^X$  and  $c_m^Z$  respectively to define diffraction colour at  $(u, v, w)$ . With previously estimated  $t_m$  terms, we discretise  $u-v$  space and tabulate  $c_m^X, c_m^Y$  and  $c_m^Z, \forall m$  (see Section 4 for details). We precompute  $L = M + 1$  such lookup tables to load them as textures in the GPU memory. At runtime, for each pixel, we compute the half-vector  $(u, v, w)$  from the view geometry, evaluate Chebyshev polynomials  $T_m(w)$  on-the-fly and simultaneously accumulate their linear sum as scaled by lookup terms from  $c_m^X, c_m^Y, c_m^Z$ . This summation gives us the CIE XYZ components for rendering diffraction colour at that pixel.

### 3.2 Advantages

The core advantage of our approach is that it drastically reduces the number of precomputed lookup tables required for accurate renderings. This leads to significant performance gains in terms of speed and accuracy against the amount of storage used. Furthermore, the accuracy of our approach monotonically increases with each additional table. Next, we explain these gains with technical details.

As discussed in Section 3, most of the DFT coefficients  $\mathcal{F}_{\lambda}(\xi_s, \xi_t)$  for a given discrete height-field  $h$  are smooth, slow varying functions of  $w$  and we can approximate them as polynomials of a low degree. For any continuous function  $f(w)$  defined over an interval for  $w$ , approximation theory states that there exists a unique polynomial of a given degree  $M$  that is its ‘best’ approximation [Burkill 1959]. A polynomial approximation is considered best if it minimizes the maximum approximation error over the whole interval for  $w$  and called as minimax polynomial for  $f(w)$ . While there is no well-defined strategy to estimate the minimax polynomial of function  $f(w)$ , it can be closely approximated with a linear combination of Chebyshev polynomials  $T_m(w)$ , in general [Burkill 1959]. Thus, Chebyshev approximation is popularly used in several problems to achieve near best approximations in a practicable manner.

Above properties of Chebyshev approximations imply that for a given number  $L$  of lookup tables, our approach produces the best practicable accuracy in rendering diffraction colours with polynomial fits. With fewer lookup tables, our approach reduces runtime computations from lookups as well as the memory required to store the tables. In comparison, Dhillon et al. [2014] also use polynomial approximations in terms of  $w$ . However, their lookup coefficients are based on Fourier-series approximations which result in strong ringing artifacts due to Gibbs phenomena unless a large number of lookup tables are used. With Chebyshev approximations, these artifacts translate into Runge phenomena. However, using nonlinear sampling of the  $w$ -range while estimating coefficients  $a_m^j$  and  $b_m^j$ , these Runge-phenomena related errors are minimised. With only a few polynomial terms in our method, these errors are very low for most of the approximated DFT coefficients and it rarely manifests into noticeable visual artifacts. Furthermore, for difficult cases with high overall brightness, accuracy is quickly improved by increasing the number of lookup tables. Thus, our method provides scalable gains for memory-vs-performance trade-offs.



**Figure 3:** BRDF slices for a Sunbeam snakeskin patch with a normal incident white lightbeam. Each map centre represent  $\omega_v \equiv (0, 0)$  and each point  $(x = \sin \theta_v \cos \phi_v, y = \sin \theta_v \sin \phi_v)$  represents a view direction  $\omega_v \equiv (\theta_v, \phi_v)$ . Leftmost image is generated with the reference Method\_BR at 4000RU. With just two lookup tables ( $L=2$ ), our method matches these expected results.

## 4 Experiments and Results

**Implementation:** We implement three methods for our evaluations. **Method\_M** is our main approach, implemented as a GLSL shader. We discretise  $u-v$  space and tabulate  $c_m^X$  for each discrete point  $(u, v)$  by evaluating the integral in Equation (7) with a trapezoidal rule. We take uniform steps of  $\Delta\lambda = 5nm$  for these integrations. Similarly, we tabulate  $c_m^Y$  and  $c_m^Z$ . For a given height-field  $h$  and specified number of lookup tables  $L$ , we precompute one 3-element texture array  $\mathbf{T}_m \equiv (c_m^X, c_m^Y, c_m^Z)$  with  $501 \times 501$  discrete points, for each  $m \in [0, L-1]$ . With  $u, v \in [-2, 2]$ , we discretise  $u-v$  space nonlinearly for better results [Dhillon et al. 2014]. For each pixel  $d_{ij}$  in the lookup table  $\mathbf{T}_m$ , where  $i, j \in [-N, -N+1, \dots, N]$ ,  $u(i, j) = 2(i/N)^5$  and  $v(i, j) = 2(j/N)^5$ . Except for the contents of the lookup tables, our Method\_M is implemented in same manner as that by Dhillon et al. [2014]. We implement their lookup table based approach, say **Method\_B**, for performance benchmarking and their non-interactive, reference shader as the ultimate reference for visual comparisons, say **Method\_BR**.

**Data:** We use five test height-fields: (a) two natural, quasi-periodic nanostructures from snakeskins (one each from Sunbeam and Corn snakes), (b) one synthetic strictly regular blazed grating, (c) one synthetic quasi-random and (d) one random height-field in our evaluations. Each sample patch is  $65\mu m \times 65\mu m$  in size to accommodate a span of  $4\sigma_s = 65\mu m$  for the Gaussian coherence window.

**Results:** Figure 3 shows BRDF slices for the Sunbeam snake with a normal incident light beam, for different number of lookups for our Method\_M. Even with a single lookup table, the BRDF slice looks subjectively very similar to the reference image from Method\_BR. However, overall brightness is apparently low. With two lookup tables, results look almost indistinguishable from the reference. Using three lookup tables bring about only marginal changes near the central specular region. Thereafter, increasing lookup tables did not bring any noticeable change. We made similar observations for all other height-fields in our data set. Figure 4 shows their BRDF slices at 4000RU, except for the random field which is at 20000RU.

For comparison with the benchmark Method\_B, we produce surface renderings. Figure 1 shows a reference photo with diffraction colours emerging from a Sunbeam skin sample in a laboratory setup. We simulate this lab setup to produce a subjectively similar rendering with just two lookup tables for our Method\_M. In particular, the lower brightness of the lower blob of colours in the reference photo is well-noticeable in our renderings. In comparison, Method\_B needs atleast 3 lookup tables to produce diffraction colours. With three tables, Method\_B only simulates a first-order Born approximation where the upper and lower blobs of colour have similar intensities. Increasing the number of lookups from three introduces disturbing visual artifacts as seen for  $L=25$ . These artifacts only disappear



when Method\_B converges with a large number of lookups  $L=67$ . Note that our Method\_M produces subjectively similar results as for the convergence of Method\_B. We emphasise that both Method\_B and Method\_M upon convergence match the accuracy of the ultimate reference, Method\_BR (not shown). However, in general, our method achieves this high accuracy with lot fewer tables.

Finally, we demonstrate the accuracy of our Method\_M on an actual Corn snake skin. Figure 5 shows that our method produces realistic secondary highlights with just 2 lookup tables at  $1000RU$ . The benchmark Method\_B cannot produce these secondary highlights with fewer lookup tables. In summary, our Method\_M reduces the required number of lookup tables for highly accurate renderings by an order of magnitude when compared to the benchmark Method\_B.

We used FRAPS to evaluate the speed of our Method\_M and the benchmark Method\_B on an NVIDIA GeForce GT 630 GPU with 192 CUDA Cores and shared system memory of 2048MB. With equal lookups both methods perform similarly, frequently surpassing 60fps for a rendering window of  $1024 \times 1024$  pixels. As our method requires fewer tables, in comparison, it is faster than the benchmark.

## 5 Conclusion

Memory efficiency is critical for practical realisations of interactive diffraction shaders. We improve upon current state-of-the-art in realistic, interactive rendering of natural surfaces with complex nanostructures. More specifically, we devise a new method based on Chebyshev approximations to compute lookup tables, which reduces runtime memory footprint of a recent state-of-the-art diffraction shader by an order of magnitude. We achieve these improvements without compromising on realistic appearances or the rendering speed. On the contrary, reducing required lookup tables further reduces computational load. With just two or three lookup tables, we demonstrate highly accurate renderings for real, natural, quasi-periodic nanostructures on snakeskins. Also, our method works equally well for quasi-random or completely random height-fields. In future, we would like to devise a quantitative analysis based approach to pre-determine required number of lookup tables for a given accuracy level. Also, our method can be adapted to efficiently examine structural colours due to seemingly random nanostructures on biological or other natural samples.

## Acknowledgements

This work is supported by SNSF Early Postdoc.Mobility Fellowship P2BEP2\_165343 to D. J. Dhillon, EPSRC Fellowship EP/N006259/1 and a Royal Society Wolfson Research Merit Award to A. Ghosh. We thank LANE lab and Dept. of QMP (previously DPMC) at University of Geneva for snakeskin height-fields and surface geometry.

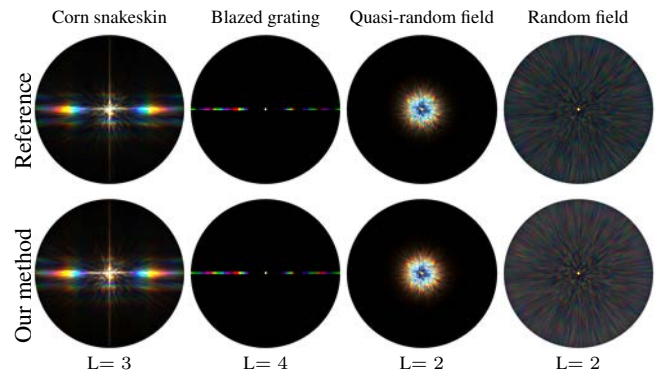
## References

BURKILL, J. C. 1959. *Lectures on approximation by polynomials*. No. 16. Tata Institute of Fundamental Research.

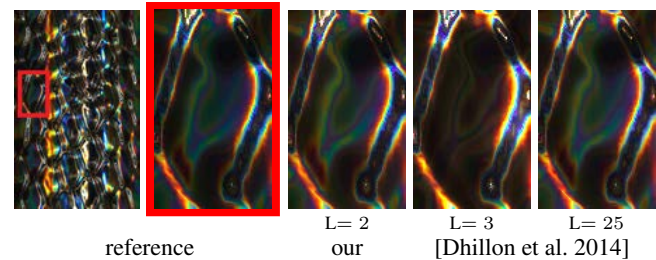
CUYPERS, T., HABER, T., BEKAERT, P., OH, S. B., AND RASKAR, R. 2012. Reflectance model for diffraction. *ACM Trans. Graph.* 31, 5 (Sept.), 122:1–122:11.

DHILLON, D. S., TEYSSIER, J., SINGLE, M., GAPONENKO, I., MILINKOVITCH, M. C., AND ZWICKER, M. 2014. Interactive diffraction from biological nanostructures. *Computer Graphics Forum* 33, 8, 177–188.

IMURA, M., OSHIRO, O., SAEKI, M., MANABE, Y., CHIHARA, K., AND YASUMURO, Y. 2009. A generic



**Figure 4:** BRDF slices for height-fields with different nanostructures. While the Corn snakeskin sample is measured and natural, other three are synthetic. Top row shows expected results from the reference Method\_BR. Our method (below) qualitatively matches these results with just two to four lookup tables.



**Figure 5:** Surface rendering of a real, 3D scanned Corn snakeskin patch. First two left images are generated by Method\_BR. Second image zooms into a single scale from the leftmost image. Our method (center) realistically generates the greenish and purple coloured highlights with just two lookup tables. In contrast, Method\_B (right) lacks those colours with fewer lookups ( $L=3$ ). It eventually produces accurate colours with  $L=25$  lookup tables.

real-time rendering approach for structural colors. In *Proceedings of the 16th ACM Symposium on Virtual Reality Software and Technology*, ACM, 95–102.

LINDSAY, C., AND AGU, E. 2006. Physically-based real-time diffraction using spherical harmonics. In *International Symposium on Visual Computing*, Springer, 505–517.

LIU, Y., SHIGLEY, J., AND HURWIT, K. 1999. Iridescence color of a shell of the mollusk *pinctada margaritifera* caused by diffraction. *Optics express* 4, 5, 177–182.

MATIN, T. R., LEONG, M. K., MAJLIS, B. Y., AND GEBESHUBER, I. C. 2010. Correlating nanostructures with function: Structural colors on the wings of a malaysian bee. *AIP Conference Proceedings* 1284, 1, 5–14.

MUSBACH, A., MEYER, G. W., REITICH, F., AND OH, S. H. 2013. Full wave modelling of light propagation and reflection. *Computer Graphics Forum* 32, 6, 24–37.

STAM, J. 1999. Diffraction shaders. In *Proceedings of SIGGRAPH 99*, ACM, Annual Conference Series, 101–110.

WU, F., AND ZHENG, C. 2013. A comprehensive geometrical optics application for wave rendering. *Graph. Models* 75, 6 (Nov.), 318–327.