

# Adaptive Measurement of Anisotropic Material Appearance

## SUPPLEMENTAL MATERIAL

Radomír Vávra, Jiří Filip  
Institute of Information Theory and Automation of the CAS, Czech Republic

This document provides supplemental material to the paper *Adaptive Measurement of Anisotropic Material Appearance* accepted to Pacific Graphics 2017.

### CONTENTS:

#### Tested Uniform Sampling Schemes (pages 2–5)

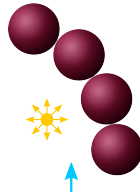
- Fig. 1 (page 2) illustrates 30 different uniform sampling schemes used for comparison of the proposed method.
- Tab. 1 (pages 3-5) shows the detailed angular steps for the schemes.

#### Test BRDF Data (pages 6–7)

- Fig. 2 (page 6) shows BRDFs of the tested materials and results of their analytical fitting by Kurt et al. (2010) model used as our dense reference BRDF data.
- Tab. 2 (page 7) lists corresponding model's parameters for individual materials.

#### Proposed Method's Performance Analysis (pages 8–22)

- Fig. 3 (page 8) shows relative errors (MRE [%]) for all ten tested materials and all methods as a function of number of samples.
- Fig. 4 (page 9) explains projection of 4D BRDF into a 2D image. The projection is used in the next figures.
- Fig. 5 – Fig. 14 (pages 10–19) illustrate the proposed method's performance in comparison with uniform sampling interpolated by barycentric and RBF interpolations for schemes 14 and 19 (8 911 and 18 712 samples respectively). The figures compare BRDFs and their rendering in illumination environment.
- Fig. 15 (page 20) shows result of measurement experiment comparing two uniform and one proposed adaptive sampling approaches in the sampling schemes 14 and 19. The 3D scene used is shown below.



- Fig. 16 (page 21) shows performance of Kurt and Ward model (fitted to 8,911 samples) when used for the 3D scene reconstruction.
- Finally, (page 22) outlines stability test of slices placement.

#### Other Supplementary Materials (pages 23–26)

- Section 5 details an algorithm of adaptive samples placement along the one-dimensional slices.
- Section 7 rewrites the equations for interpolation in the 2D BRDF subspace introduced by Filip et al. (2013) to make transition into the 4D space possible. Then, equations for interpolation of any value in the BRDF space from the sparse 4D structure (in the form of four types of the BRDF slices) are provided.
- Section 8 describes implementation of the interpolation method on graphics hardware.

# 1 Tested Uniform Sampling Schemes

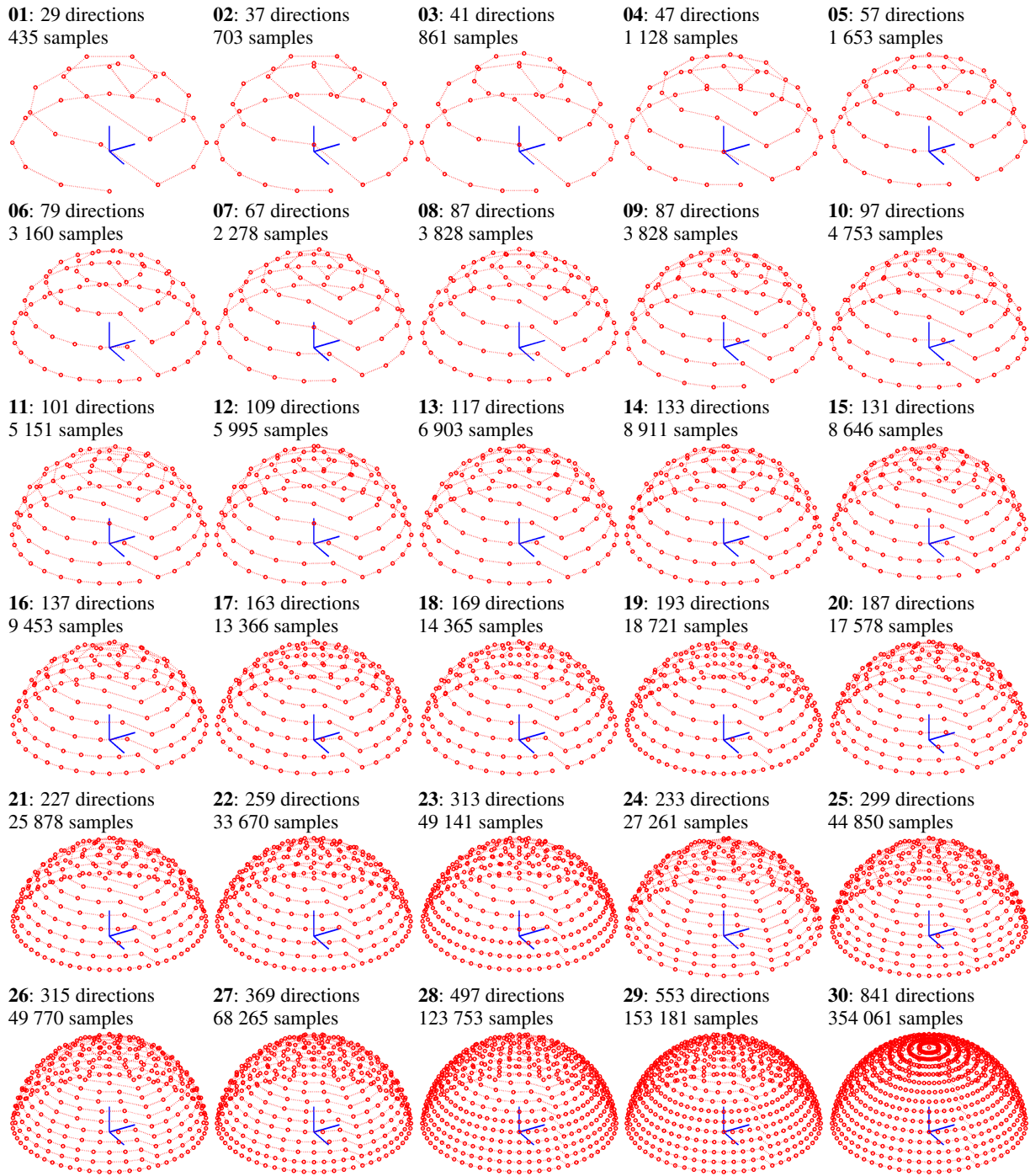


Figure 1: Overview of thirty sampling schemes used for uniform sampling of hemisphere using barycentric and RBF based methods. Each scheme is accompanied with number of directions per hemisphere and corresponding number of reciprocal samples used for BRDF measurement.



Table 1: Uniform sampling schemes and their azimuthal steps and sample counts on individual elevations.

scheme 1			scheme 2			scheme 3			scheme 4		
$\theta$ [°]	$\Delta\varphi$ [°]	#n	$\theta$ [°]	$\Delta\varphi$ [°]	#n	$\theta$ [°]	$\Delta\varphi$ [°]	#n	$\theta$ [°]	$\Delta\varphi$ [°]	#n
0	360	1	0	360	1	0	360	1	0	360	1
28	60	6	28	60	6	28	36	10	20	60	6
56	36	10	56	30	12	56	30	12	40	36	10
80	30	12	80	20	18	80	20	18	60	30	12
									80	20	18
sum		29	sum		37	sum		41	sum		47
illum. $\times$ view		841	illum. $\times$ view		1369	illum. $\times$ view		1681	illum. $\times$ view		2209
reciprocal		435	reciprocal		703	reciprocal		861	reciprocal		1128

scheme 5			scheme 6			scheme 7			scheme 8		
$\theta$ [°]	$\Delta\varphi$ [°]	#n	$\theta$ [°]	$\Delta\varphi$ [°]	#n	$\theta$ [°]	$\Delta\varphi$ [°]	#n	$\theta$ [°]	$\Delta\varphi$ [°]	#n
0	360	1	0	360	1	0	360	1	0	360	1
20	60	6	20	36	10	16	60	6	16	60	6
40	30	12	40	20	18	32	36	10	32	30	12
60	20	18	60	18	20	48	30	12	48	20	18
80	18	20	80	12	30	64	20	18	64	18	20
						80	18	20	80	12	30
sum		57	sum		79	sum		67	sum		87
illum. $\times$ view		3249	illum. $\times$ view		6241	illum. $\times$ view		4489	illum. $\times$ view		7569
reciprocal		1653	reciprocal		3160	reciprocal		2278	reciprocal		3828

scheme 9			scheme 10			scheme 11			scheme 12		
$\theta$ [°]	$\Delta\varphi$ [°]	#n	$\theta$ [°]	$\Delta\varphi$ [°]	#n	$\theta$ [°]	$\Delta\varphi$ [°]	#n	$\theta$ [°]	$\Delta\varphi$ [°]	#n
0	360	1	0	360	1	0	360	1	0	360	1
14	60	6	14	60	6	12	90	4	12	60	6
28	36	10	28	36	10	24	60	6	24	36	10
42	30	12	42	30	12	36	36	10	36	30	12
56	20	18	56	20	18	48	30	12	48	30	12
70	18	20	70	18	20	60	20	18	60	20	18
80	18	20	80	12	30	72	18	20	72	18	20
						80	12	30	80	12	30
sum		87	sum		97	sum		101	sum		109
illum. $\times$ view		7569	illum. $\times$ view		9409	illum. $\times$ view		10201	illum. $\times$ view		11881
reciprocal		3828	reciprocal		4753	reciprocal		5151	reciprocal		5995

scheme 13			scheme 14			scheme 15			scheme 16		
$\theta$ [°]	$\Delta\varphi$ [°]	#n	$\theta$ [°]	$\Delta\varphi$ [°]	#n	$\theta$ [°]	$\Delta\varphi$ [°]	#n	$\theta$ [°]	$\Delta\varphi$ [°]	#n
0	360	1	0	360	1	0	360	1	0	360	1
12	60	6	12	60	6	10	90	4	10	90	4
24	36	10	24	36	10	20	60	6	20	60	6
36	30	12	36	30	12	30	36	10	30	36	10
48	20	18	48	20	18	40	30	12	40	30	12
60	18	20	60	18	20	50	20	18	50	20	18
72	18	20	72	12	30	60	18	20	60	18	20
80	12	30	80	10	36	70	12	30	70	12	30
						80	12	30	80	10	36
sum		117	sum		133	sum		131	sum		137
illum. $\times$ view		13689	illum. $\times$ view		17689	illum. $\times$ view		17161	illum. $\times$ view		18769
reciprocal		6903	reciprocal		8911	reciprocal		8646	reciprocal		9453

scheme 17			scheme 18			scheme 19			scheme 20		
$\theta$ [°]	$\Delta\varphi$ [°]	#n	$\theta$ [°]	$\Delta\varphi$ [°]	#n	$\theta$ [°]	$\Delta\varphi$ [°]	#n	$\theta$ [°]	$\Delta\varphi$ [°]	#n
0	360	1	0	360	1	0	360	1	0	360	1
10	60	6	10	60	6	10	60	6	8	90	4
20	36	10	20	36	10	20	36	10	16	60	6
30	30	12	30	30	12	30	30	12	24	36	10
40	20	18	40	20	18	40	20	18	32	30	12
50	18	20	50	18	20	50	18	20	40	20	18
60	12	30	60	12	30	60	12	30	48	18	20
70	12	30	70	10	36	70	10	36	56	18	20
80	10	36	80	10	36	80	6	60	64	12	30
									72	12	30
									80	10	36
sum		163	sum		169	sum		193	sum		187
illum. × view		26569	illum. × view		28561	illum. × view		37249	illum. × view		34969
reciprocal		13366	reciprocal		14365	reciprocal		18721	reciprocal		17578

scheme 21			scheme 22			scheme 23			scheme 24		
$\theta$ [°]	$\Delta\varphi$ [°]	#n	$\theta$ [°]	$\Delta\varphi$ [°]	#n	$\theta$ [°]	$\Delta\varphi$ [°]	#n	$\theta$ [°]	$\Delta\varphi$ [°]	#n
0	360	1	0	360	1	0	360	1	0	360	1
8	90	4	8	60	6	8	60	6	6	90	4
16	60	6	16	36	10	16	36	10	12	60	6
24	36	10	24	30	12	24	30	12	18	60	6
32	30	12	32	20	18	32	20	18	24	36	10
40	20	18	40	18	20	40	18	20	30	36	10
48	18	20	48	12	30	48	12	30	36	30	12
56	12	30	56	12	30	56	10	36	42	30	12
64	12	30	64	10	36	64	6	60	48	20	18
72	10	36	72	10	36	72	6	60	54	20	18
80	6	60	80	6	60	80	6	60	60	18	20
									66	18	20
									72	12	30
									78	12	30
									80	10	36
sum		227	sum		259	sum		313	sum		233
illum. × view		51529	illum. × view		67081	illum. × view		97969	illum. × view		54289
reciprocal		25878	reciprocal		33670	reciprocal		49141	reciprocal		27261

scheme 25			scheme 26			scheme 27		
$\theta$ [°]	$\Delta\varphi$ [°]	#n	$\theta$ [°]	$\Delta\varphi$ [°]	#n	$\theta$ [°]	$\Delta\varphi$ [°]	#n
0	360	1	0	360	1	0	360	1
6	90	4	6	60	6	6	60	6
12	60	6	12	60	6	12	36	10
18	60	6	18	36	10	18	30	12
24	36	10	24	30	12	24	30	12
30	30	12	30	30	12	30	20	18
36	30	12	36	20	18	36	20	18
42	20	18	42	20	18	42	18	20
48	20	18	48	18	20	48	18	20
54	18	20	54	18	20	54	12	30
60	12	30	60	12	30	60	12	30
66	12	30	66	12	30	66	10	36
72	10	36	72	10	36	72	10	36
78	10	36	78	10	36	78	6	60
80	6	60	80	6	60	80	6	60
sum		299	sum		315	sum		369
illum. × view		89401	illum. × view		99225	illum. × view		136161
reciprocal		44850	reciprocal		49770	reciprocal		68265

scheme 28			scheme 29			scheme 30		
$\theta$ [°]	$\Delta\varphi$ [°]	#n	$\theta$ [°]	$\Delta\varphi$ [°]	#n	$\theta$ [°]	$\Delta\varphi$ [°]	#n
0	360	1	0	360	1	0	360	1
6	90	4	6	60	6	6	6	60
12	60	6	12	36	10	12	6	60
18	36	10	18	30	12	18	6	60
24	30	12	24	20	18	24	6	60
30	20	18	30	18	20	30	6	60
36	18	20	36	12	30	36	6	60
42	12	30	42	10	36	42	6	60
48	10	36	48	6	60	48	6	60
54	6	60	54	6	60	54	6	60
60	6	60	60	6	60	60	6	60
66	6	60	66	6	60	66	6	60
72	6	60	72	6	60	72	6	60
78	6	60	78	6	60	78	6	60
80	6	60	80	6	60	80	6	60
sum		497	sum		553	sum		841
illum. × view		247009	illum. × view		305809	illum. × view		707281
reciprocal		123753	reciprocal		153181	reciprocal		354061

## 2 Test BRDF Data

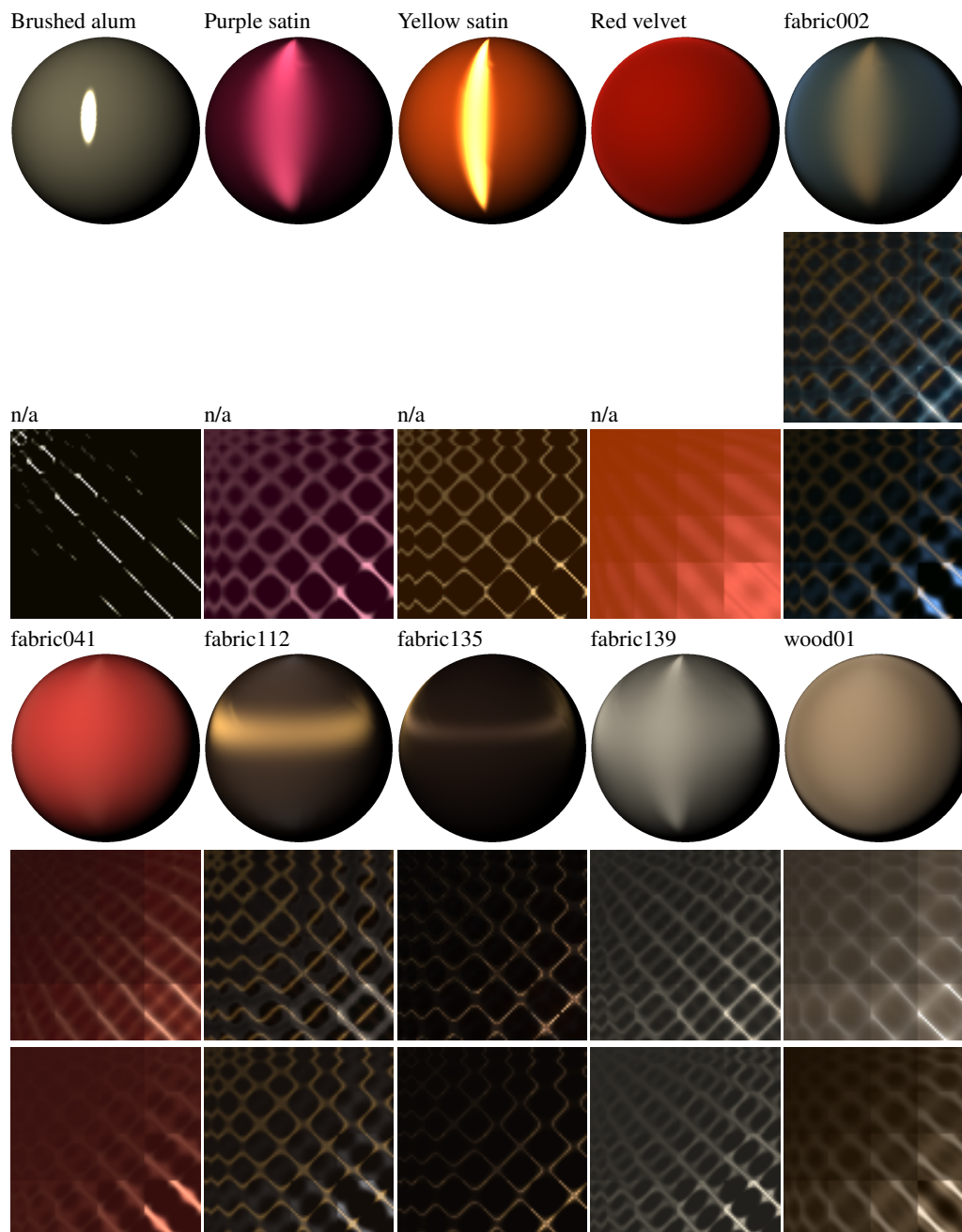


Figure 2: Comparison of anisotropic BRDF measurement of the tested materials with their approximation using 1 or 2 lobes of Kurt BRDF model [KSKK10], which has been used as our dense reference dataset. Note that for the first four materials fits from [KSKK10] were used and the reference data are not available.

Table 2: Parameters of Kurt BRDF model for ten materials.

		Brushed alum.	Purple satin	Red velvet	Yellow satin	fabric002	fabric041	fabric112	fabric135	fabric139	wood01
	$k_{dr}$	0.0036	0.0026	0.0048	0.0066	0.0805	0.2565	0.0838	0.0395	0.1607	0.1548
	$k_{dg}$	0.0034	0.0004	0.0005	0.0022	0.0945	0.0703	0.0596	0.0258	0.153	0.1219
	$k_{db}$	0.0026	0.0011	0	0.0004	0.081	0.0613	0.0434	0.0205	0.135	0.0836
lobe 1	$k_{sr}$	0.0115	0.1404	0.1938	0.0542	0.0798	0.0211	0.1419	0.014	0.0483	0.0768
	$k_{sg}$	0.0105	0.0522	0.0333	0.0345	0.1205	0.019	0.1078	0.011	0.0461	0.0741
	$k_{sb}$	0.0075	0.0711	0.0267	0.0131	0.1796	0.0146	0.0521	0.0087	0.0414	0.0706
	$f_0$	0.999	0.055	0.041	0.207	0.3431	1	1	1	1	1
	$m_x$	0.035	0.339	2.337	0.129	1.3508	0.3665	1.2101	1.2025	0.2948	0.5672
	$m_y$	0.129	1.256	2.644	1.084	0.8121	1.1075	0.2139	0.1029	1.3215	0.7619
	$\alpha$	0.005	0	0	0.197	0.8359	0.4984	0	0.5043	0.5753	0.5051
lobe 2	$k_{sr}$					0.0798	0.0211	0.1419	0.014	0.0483	0.0768
	$k_{sg}$					0.1205	0.019	0.1078	0.011	0.0461	0.0741
	$k_{sb}$					0.1796	0.0146	0.0521	0.0087	0.0414	0.0706
	$f_0$					1	0.1222	1	0.3078	1	1
	$m_x$					0.3168	1.1864	0.7046	2.038	1.3281	1.7007
	$m_y$					1.0061	0.7247	1.2292	0.3867	0.3677	1.923
	$\alpha$					0.0905	0.8356	0.976	0.5723	0.667	0.2285

### 3 Proposed Method's Performance Analysis

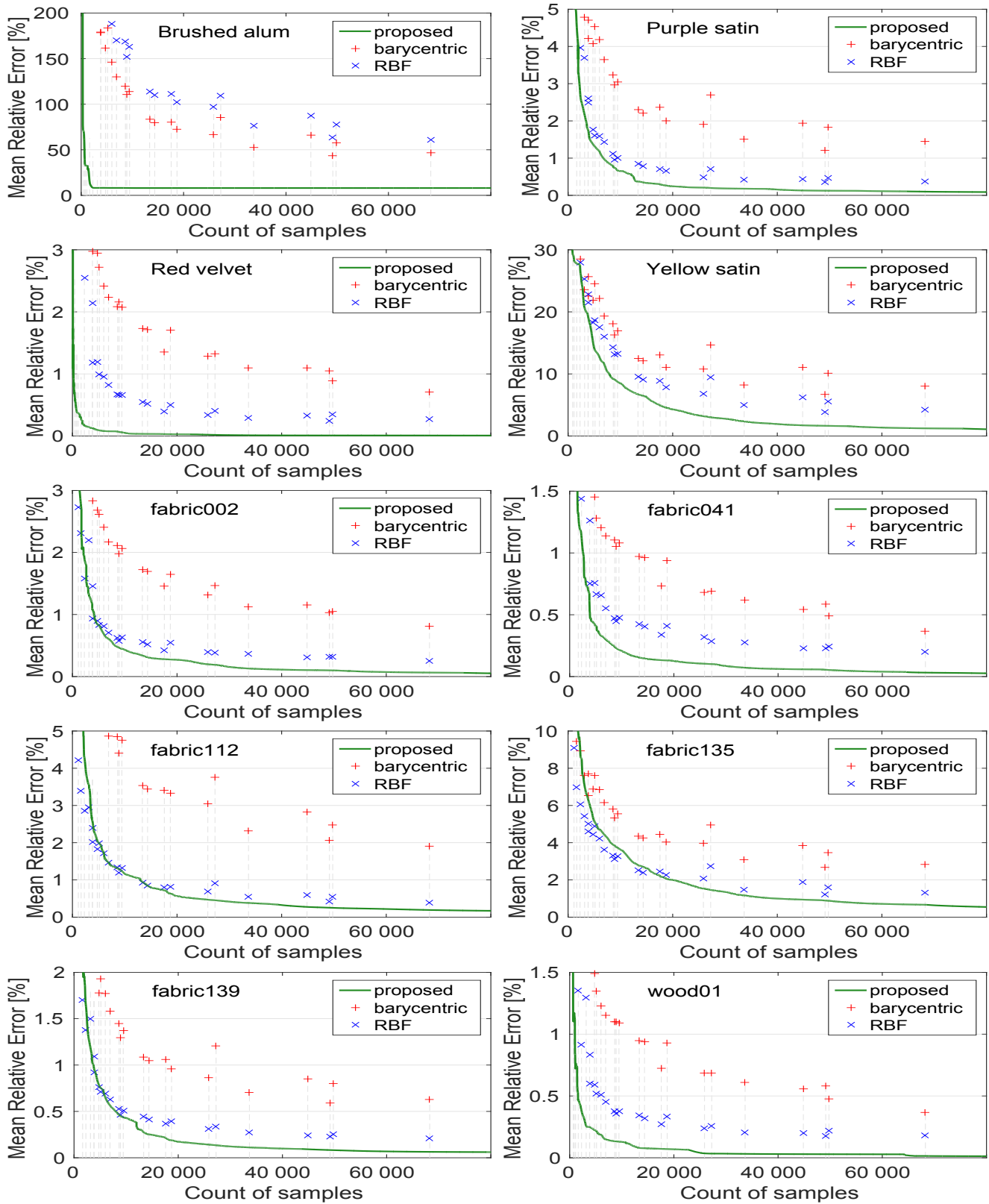


Figure 3: Relative error [%] of BRDF reconstruction of the proposed method, barycentric and RBF-based uniform interpolation methods as a function of number of samples for all tested materials.

As the visualization of an entire 4D BRDF at an angular density of  $2^\circ$  is impossible, we therefore analyze performance of the tested methods in a 2D preview comprised of four types of sparse projections distributed uniformly in the BRDF space and illustrating its major azimuthal- and elevation-dependent behavior as explained in Fig. 4.

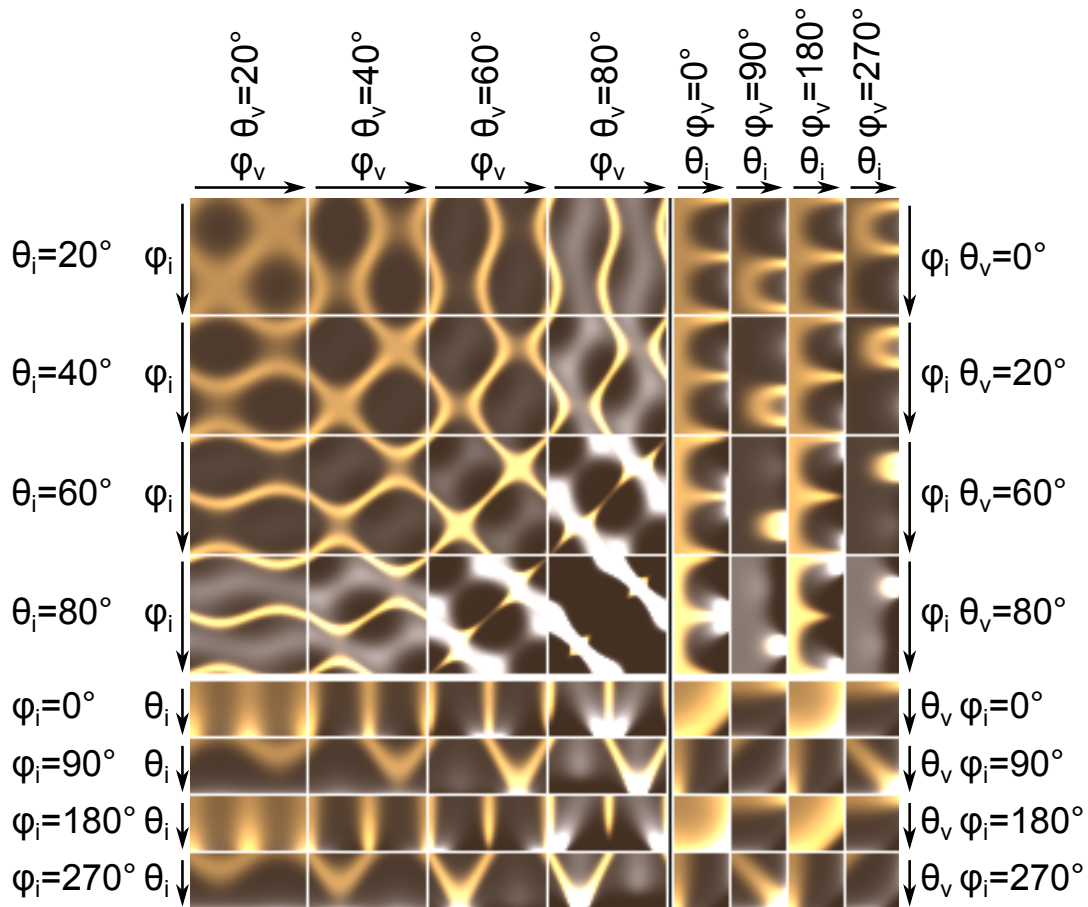


Figure 4: Example of 4D BRDF projection into 2D image in four different ways. Top-left area: continuous subspaces of azimuthal angles ( $\varphi_i, \varphi_v$ ) for several combinations of fixed elevation angles ( $\theta_i, \theta_v$ ). Top-right area: continuous subspaces of angles ( $\varphi_i, \theta_i$ ) for several combinations of angles ( $\theta_v, \varphi_v$ ). Bottom-left area: continuous subspaces of angles ( $\theta_i, \varphi_v$ ) for several combinations of angles ( $\varphi_i, \theta_v$ ). Bottom-right area: continuous subspaces of elevation angles ( $\theta_v, \theta_i$ ) for several combinations of azimuthal angles ( $\varphi_i, \varphi_v$ ).



Brushed alum		rendering ( <i>grace</i> environment)		BRDF	
	(a) reference	difference 10×	errors	difference 10×	MRE [%]
scheme 14 ( 8 911 samples)	(b) barycentric		9.7 27.5 19.39 0.932 68.4		110.7
	(c) RBF		11.9 28.6 19.03 0.907 69.6		152.1
	(d) proposed		7.3 24.0 20.56 0.950 75.3		24.5
scheme 19 ( 18 721 samples)	(e) barycentric		8.0 23.1 20.88 0.945 81.6		72.4
	(f) RBF		9.0 23.7 20.67 0.932 80.5		102.1
	(g) proposed		4.5 15.1 24.60 0.966 81.2		18.9

Figure 5: Columns 1,2: rendering of the reference BRDF of material *Brushed alum* (a) compared with its reconstruction using all tested methods for the same number of 8 911 reciprocal samples (scheme 14) and 18 721 samples (scheme 19): (b),(e) barycentric, (c),(f) RBF, (d),(g) proposed. Columns 3,4: corresponding previews of reference and reconstructed BRDFs. Errors from the reference data for renderings are in CIE  $\Delta E$ / RMSE/ PSNR[dB]/ SSIM/ VDP2 and for BRDFs in MRE [%]. All difference images are scaled 10×.

Purple satin

	rendering ( <i>grace</i> environment)			BRDF		
(a) reference						
		difference 10×	errors	difference 10×	MRE [%]	
scheme 14 ( 8 911 samples)	(b) barycentric		0.5 0.8 50.21 0.999 95.9		3.0	
	(c) RBF		0.3 0.5 54.84 0.999 95.9		1.0	
	(d) proposed		0.2 0.3 57.99 0.999 95.9		0.8	
scheme 19 ( 18 721 samples)	(e) barycentric		0.4 0.6 52.70 0.999 95.9		2.0	
	(f) RBF		0.2 0.4 56.94 0.999 95.9		0.7	
	(g) proposed		0.1 0.3 59.83 0.999 95.9		0.5	

Figure 6: Columns 1,2: rendering of the reference BRDF of material *Purple satin* (a) compared with its reconstruction using all tested methods for the same number of 8 911 reciprocal samples (scheme 14) and 18 721 samples (scheme 19): (b),(e) barycentric, (c),(f) RBF, (d),(g) proposed. Columns 3,4: corresponding previews of reference and reconstructed BRDFs. Errors from the reference data for renderings are in CIE  $\Delta E$ / RMSE/ PSNR[dB]/ SSIM/ VDP2 and for BRDFs in MRE [%]. All difference images are scaled 10×.



**Yellow satin**

	rendering ( <i>grace</i> environment)		BRDF	
(a) reference				
		difference 10×	errors	difference 10× MRE [%]
scheme 14 ( 8 911 samples)	(b) barycentric		1.7 3.3 37.72 0.996 94.5	2.2
	(c) RBF		1.5 2.9 39.03 0.996 94.5	0.7
	(d) proposed		1.3 2.7 39.46 0.997 94.7	0.1
scheme 19 ( 18 721 samples)	(e) barycentric		1.2 2.3 41.03 0.997 95.4	1.7
	(f) RBF		1.0 1.9 42.67 0.997 95.5	0.5
	(g) proposed		1.0 2.2 41.32 0.997 94.7	0.1

Figure 7: Columns 1,2: rendering of the reference BRDF of material *Yellow satin* (a) compared with its reconstruction using all tested methods for the same number of 8 911 reciprocal samples (scheme 14) and 18 721 samples (scheme 19): (b),(e) barycentric, (c),(f) RBF, (d),(g) proposed. Columns 3,4: corresponding previews of reference and reconstructed BRDFs. Errors from the reference data for renderings are in CIE  $\Delta E$ / RMSE/ PSNR[dB]/ SSIM/ VDP2 and for BRDFs in MRE [%]. All difference images are scaled 10×.


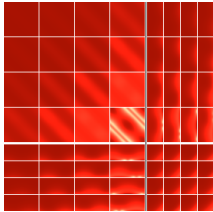

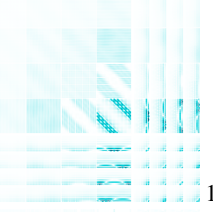

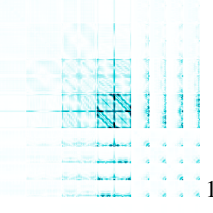





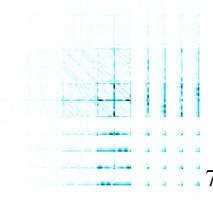


Red velvet		rendering ( <i>grace</i> environment)		BRDF	
		difference 10×	errors	difference 10×	MRE [%]
(a) reference					
scheme 14 ( 8 911 samples)	(b) barycentric		0.4 0.6 53.16 0.999 95.9		16.3
	(c) RBF		0.2 0.3 58.76 0.999 95.9		13.2
	(d) proposed		0.0 0.1 70.09 1.000 95.9		10.4
scheme 19 ( 18 721 samples)	(e) barycentric		0.3 0.4 55.19 0.999 95.9		11.0
	(f) RBF		0.2 0.3 59.42 0.999 95.9		7.8
	(g) proposed		0.0 0.1 73.46 1.000 95.9		5.7

Figure 8: Columns 1,2: rendering of the reference BRDF of material *Red velvet* (a) compared with its reconstruction using all tested methods for the same number of 8 911 reciprocal samples (scheme 14) and 18 721 samples (scheme 19): (b),(e) barycentric, (c),(f) RBF, (d),(g) proposed. Columns 3,4: corresponding previews of reference and reconstructed BRDFs. Errors from the reference data for renderings are in CIE  $\Delta E$ / RMSE/ PSNR[dB]/ SSIM/ VDP2 and for BRDFs in MRE [%]. All difference images are scaled 10×.



**fabric002**

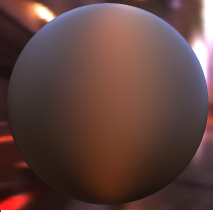
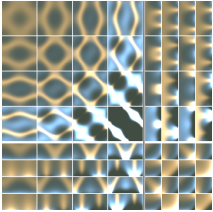
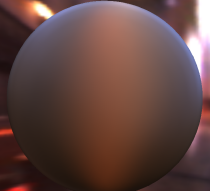
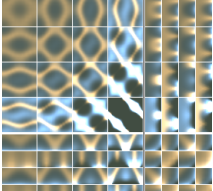
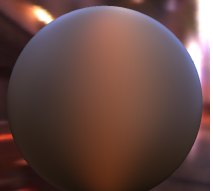
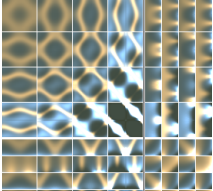
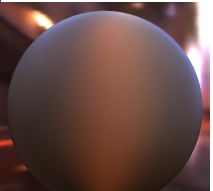
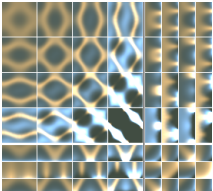
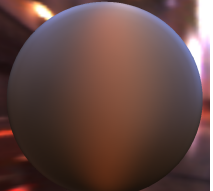
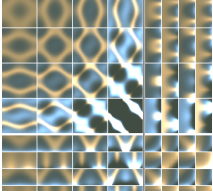
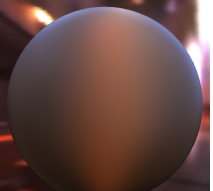
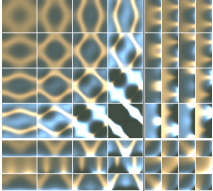
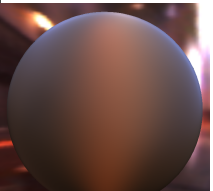
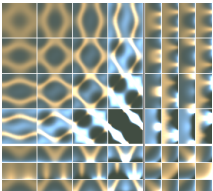
	rendering ( <i>grace</i> environment)			BRDF	
(a) reference					
		difference 10×	errors	difference 10×	MRE [%]
scheme 14 ( 8 911 samples)	(b) barycentric		0.6 1.0 48.54 0.999 95.9		2.0
	(c) RBF		0.3 0.5 54.44 0.999 95.9		0.6
	(d) proposed		0.1 0.3 60.09 0.999 95.9		0.5
scheme 19 ( 18 721 samples)	(e) barycentric		0.5 0.9 49.14 0.999 95.9		1.6
	(f) RBF		0.3 0.5 54.03 0.999 95.9		0.5
	(g) proposed		0.1 0.2 62.35 1.000 95.9		0.3

Figure 9: Columns 1,2: rendering of the reference BRDF of material *fabric002* (a) compared with its reconstruction using all tested methods for the same number of 8 911 reciprocal samples (scheme 14) and 18 721 samples (scheme 19): (b),(e) barycentric, (c),(f) RBF, (d),(g) proposed. Columns 3,4: corresponding previews of reference and reconstructed BRDFs. Errors from the reference data for renderings are in CIE  $\Delta E$ / RMSE/ PSNR[dB]/ SSIM/ VDP2 and for BRDFs in MRE [%]. All difference images are scaled 10×.

**fabric041**



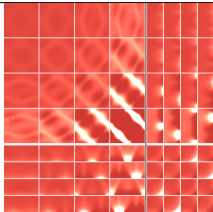
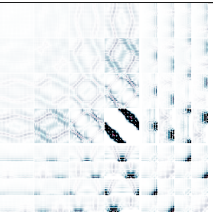
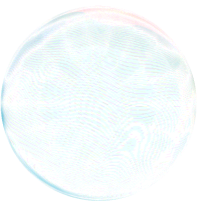
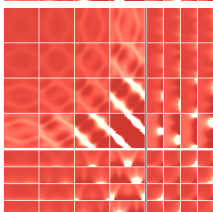
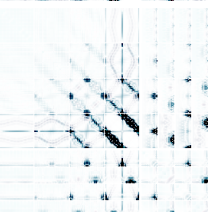

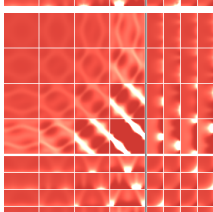
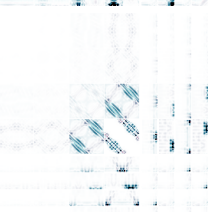

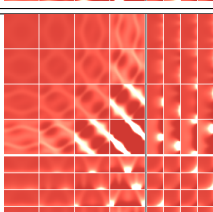
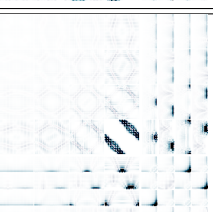

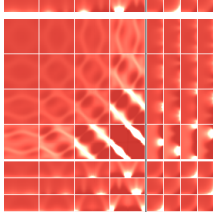
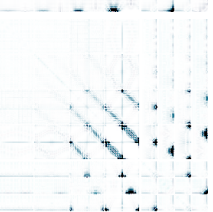

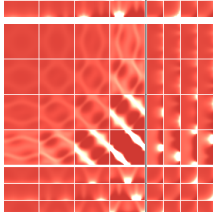
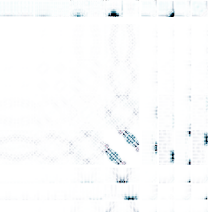
	rendering ( <i>grace</i> environment)		BRDF			
(a) reference						
		difference 10×	errors			
				difference 10× MRE [%]		
scheme 14 ( 8 911 samples)	(b) barycentric		0.3 0.6 53.08 0.999 95.9			1.1
	(c) RBF		0.3 0.4 56.21 0.999 95.9			0.5
	(d) proposed		0.0 0.1 64.96 1.000 95.9			0.2
scheme 19 ( 18 721 samples)	(e) barycentric		0.3 0.6 53.20 0.999 95.9			0.9
	(f) RBF		0.3 0.4 56.04 0.999 95.9			0.4
	(g) proposed		0.0 0.1 66.64 1.000 95.9			0.1

Figure 10: Columns 1,2: rendering of the reference BRDF of material *fabric041* (a) compared with its reconstruction using all tested methods for the same number of 8 911 reciprocal samples (scheme 14) and 18 721 samples (scheme 19): (b),(e) barycentric, (c),(f) RBF, (d),(g) proposed. Columns 3,4: corresponding previews of reference and reconstructed BRDFs. Errors from the reference data for renderings are in CIE  $\Delta E$ / RMSE/ PSNR[dB]/ SSIM/ VDP2 and for BRDFs in MRE [%]. All difference images are scaled 10×.



fabric112

	rendering ( <i>grace</i> environment)			BRDF		
(a) reference						
		difference 10×	errors		difference 10×	MRE [%]
scheme 14 ( 8 911 samples)	(b) barycentric		1.0 1.7 43.75 0.998 94.9			4.4
	(c) RBF		0.4 0.7 51.28 0.999 95.8			1.2
	(d) proposed		0.2 0.4 55.19 0.999 95.9			1.3
scheme 19 ( 18 721 samples)	(e) barycentric		0.8 1.2 46.46 0.999 95.4			3.3
	(f) RBF		0.3 0.5 54.62 0.999 95.9			0.8
	(g) proposed		0.2 0.4 56.55 0.999 95.9			0.9

Figure 11: Columns 1,2: rendering of the reference BRDF of material *fabric112* (a) compared with its reconstruction using all tested methods for the same number of 8 911 reciprocal samples (scheme 14) and 18 721 samples (scheme 19): (b),(e) barycentric, (c),(f) RBF, (d),(g) proposed. Columns 3,4: corresponding previews of reference and reconstructed BRDFs. Errors from the reference data for renderings are in CIE  $\Delta E$ / RMSE/ PSNR[dB]/ SSIM/ VDP2 and for BRDFs in MRE [%]. All difference images are scaled 10×.



**fabric135**

	rendering ( <i>grace</i> environment)		BRDF		
(a) reference					
		difference 10×	errors	difference 10× MRE [%]	
scheme 14 ( 8 911 samples)	(b) barycentric		0.5 0.8 49.61 0.999 95.7		5.4
	(c) RBF		0.4 0.7 51.33 0.999 95.8		3.1
	(d) proposed		0.3 0.5 53.68 0.999 95.9		3.9
scheme 19 ( 18 721 samples)	(e) barycentric		0.4 0.7 51.54 0.999 95.8		4.0
	(f) RBF		0.3 0.5 54.57 0.999 95.9		2.3
	(g) proposed		0.2 0.4 56.10 0.999 95.9		2.5

Figure 12: Columns 1,2: rendering of the reference BRDF of material *fabric135* (a) compared with its reconstruction using all tested methods for the same number of 8 911 reciprocal samples (scheme 14) and 18 721 samples (scheme 19): (b),(e) barycentric, (c),(f) RBF, (d),(g) proposed. Columns 3,4: corresponding previews of reference and reconstructed BRDFs. Errors from the reference data for renderings are in CIE  $\Delta E$ / RMSE/ PSNR[dB]/ SSIM/ VDP2 and for BRDFs in MRE [%]. All difference images are scaled 10×.

**fabric139**

	rendering ( <i>grace</i> environment)		BRDF	
(a) reference				
		difference 10×	errors	difference 10× MRE [%]
scheme 14 ( 8 911 samples)	(b) barycentric		0.4 0.7 51.87 0.999 95.9	1.3
	(c) RBF		0.3 0.4 55.83 0.999 95.9	0.5
	(d) proposed		0.2 0.3 58.46 0.999 95.9	0.5
scheme 19 ( 18 721 samples)	(e) barycentric		0.3 0.5 53.48 0.999 95.9	1.0
	(f) RBF		0.3 0.4 56.11 0.999 95.9	0.4
	(g) proposed		0.1 0.2 60.36 1.000 95.9	0.3

Figure 13: Columns 1,2: rendering of the reference BRDF of material *fabric139* (a) compared with its reconstruction using all tested methods for the same number of 8 911 reciprocal samples (scheme 14) and 18 721 samples (scheme 19): (b),(e) barycentric, (c),(f) RBF, (d),(g) proposed. Columns 3,4: corresponding previews of reference and reconstructed BRDFs. Errors from the reference data for renderings are in CIE  $\Delta E$ / RMSE/ PSNR[dB]/ SSIM/ VDP2 and for BRDFs in MRE [%]. All difference images are scaled 10×.



<b>wood01</b>		rendering ( <i>grace</i> environment)		BRDF	
		difference 10×	errors	difference 10×	MRE [%]
(a) reference					
scheme 14 ( 8 911 samples)	(b) barycentric		0.5 0.8 50.00 0.999 95.9		1.1
	(c) RBF		0.3 0.5 54.77 0.999 95.9		0.4
	(d) proposed		0.1 0.2 63.76 1.000 95.9		0.2
scheme 19 ( 18 721 samples)	(e) barycentric		0.4 0.7 51.86 0.999 95.9		0.9
	(f) RBF		0.3 0.4 55.63 0.999 95.9		0.3
	(g) proposed		0.0 0.1 65.90 1.000 95.9		0.1

Figure 14: Columns 1,2: rendering of the reference BRDF of material *wood01* (a) compared with its reconstruction using all tested methods for the same number of 8 911 reciprocal samples (scheme 14) and 18 721 samples (scheme 19): (b),(e) barycentric, (c),(f) RBF, (d),(g) proposed. Columns 3,4: corresponding previews of reference and reconstructed BRDFs. Errors from the reference data for renderings are in CIE  $\Delta E$ / RMSE/ PSNR[dB]/ SSIM/ VDP2 and for BRDFs in MRE [%]. All difference images are scaled 10×.

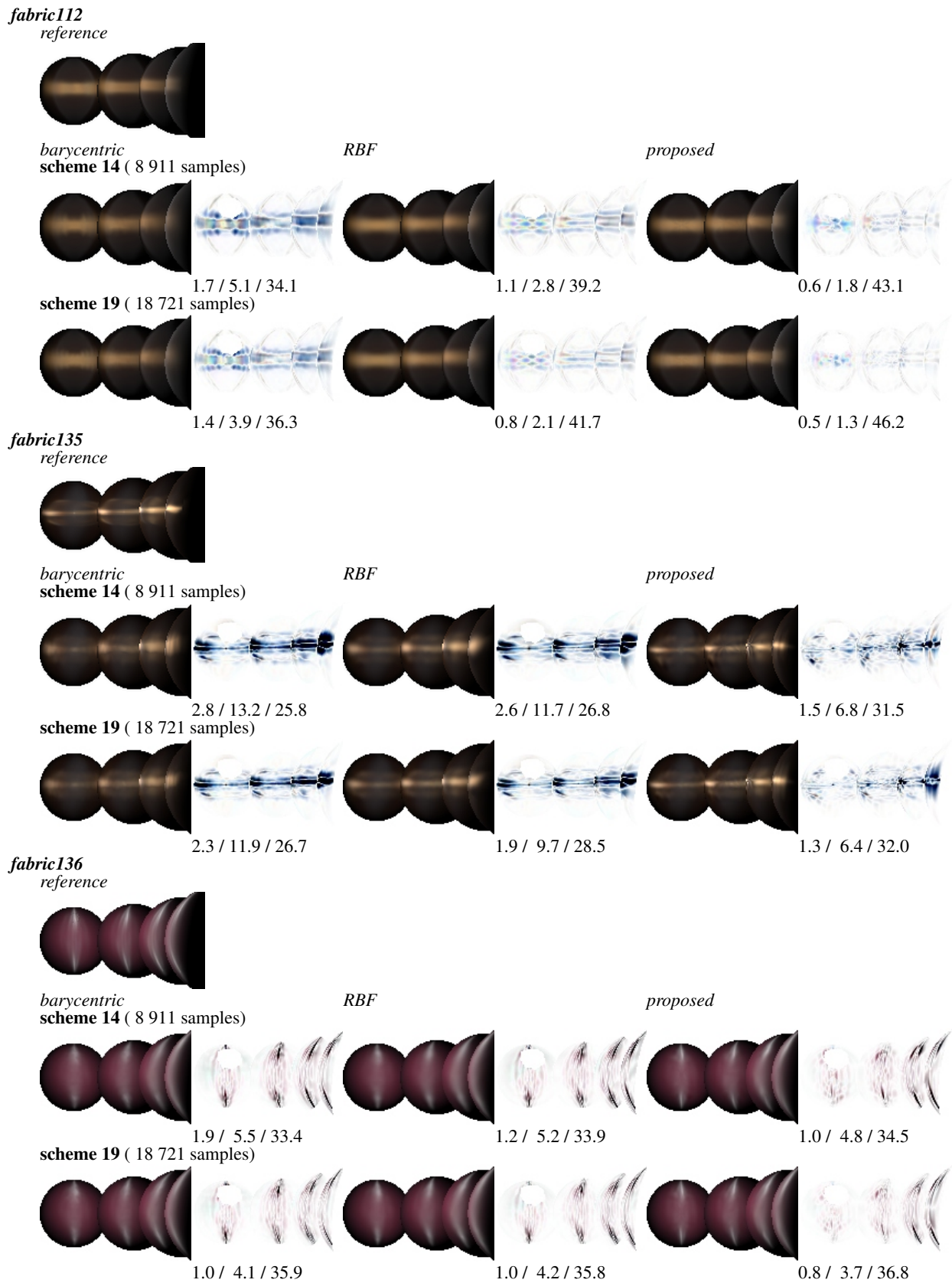


Figure 15: Measurement experiment: All directions used in this image were measured as reference and compared with uniform measurement combined with barycentric and RBF interpolations, as well as with the proposed adaptive measurement using 8911 and 18721 samples. Errors from the reference data are in CIE  $\Delta E$ / RMSE/ PSNR[dB] and the difference images are scaled  $10\times$ .

## 4 A Comparison with Analytical Models

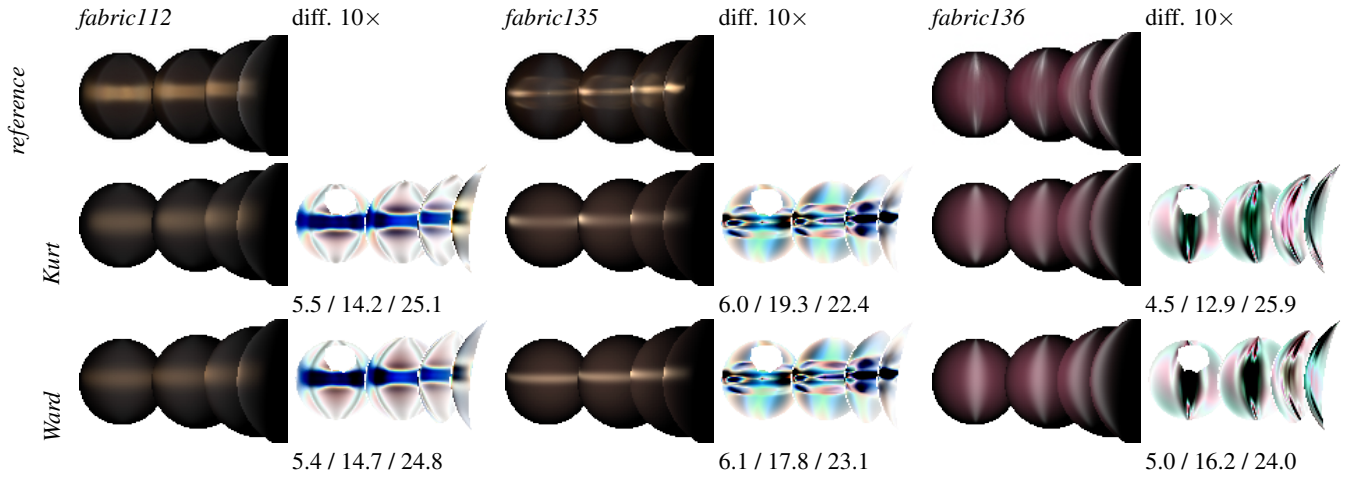


Figure 16: All the reachable directions in the virtual scene (see the first page) measured and compared with the barycentric and RBF interpolations of uniformly measured data, fits of two analytical models ([KSKK10], [Ward92]), and the proposed adaptive measurement. 8,911 samples are used, the difference values are in CIE  $\Delta E$  / RMSE / PSNR [dB].

## 5 Adaptive Sampling Along the Slices

Each slice can be interpreted as an unknown one-dimensional signal that we need to measure and reconstruct. In case of the axial or diagonal slice the signal is periodic with the period of  $360^\circ$ . To our best knowledge, adaptive measurement of an unknown one-dimensional signal with limited apriory assumptions is still an open problem. Therefore, we propose the enhanced version of the heuristic algorithm which was first introduced in [FVH\*13] to deal with the problem.

Any slice can be sampled uniformly with a defined step (e.g.,  $1^\circ$ ) or adaptively decreasing the number of samples on the one hand and increasing reconstruction accuracy in areas with high variance of the signal on the other. As the behavior of the signal is unknown, the adaptive algorithm can work with already measured samples only; adding new samples in areas where it can improve accuracy of the reconstructed signal. When the samples are taken in proper directions and their count is sufficient, every value of the slice can be interpolated precisely enough using, e.g., a piece-wise cubic splines.

We propose heuristic Algorithm 1 that enables very efficient adaptive sampling of the slices using a given count of samples  $n$ . The algorithm is based on a simple assumption. If the value of a sample can be predicted well by the neighboring samples, the neighborhood of the sample can probably be predicted well too. Therefore, there is no need to place new samples there. On the other hand, if the value of any sample cannot be predicted well by the neighboring samples, there is a possibility that neighborhood of the sample cannot be predicted well even by the sample itself together with its neighbors. Therefore, we should place new samples there.

The algorithm consists of three steps as shown in Figure 17. In the first step, the algorithm samples the signal only in the directions of intersections of the slices to collect at least some information about the signals. Then, several iterations of adaptive measurement are performed relying on the ASI (Adaptive Sampling Iteration) function, which adaptively selects and measures  $n_i$  sampling candidates.

**function**  $T = \text{ASI}(T, n_i)$

- 1: Evaluate the leave-one-out cross-validation error for each direction in  $T$  (for each slice independently), i.e., interpolate the value of a sample in  $T$  based on neighboring samples and evaluate the error by comparison with the true value of the sample.
- 2: Make an ordered list of the potential directions (see crosses in Fig. 17) in descending value of their weight. The potential direction is the one in the middle of each two neighboring already measured directions in  $T$ . Its weight is the maximal error value of the neighboring directions.
- 3: Measure values of the first  $n_i$  directions in the list and append them to  $T$ .

The count of samples  $n_i$  in each iteration depends on parameters  $k, p_1, p_2$ . The parameter  $k$  defines the ratio of the count of samples measured in the second step of the Algorithm 1 versus the count of samples taken in the third step. We have found experimentally that reasonable choice is  $k = 0.9$  expressing the ratio 90% : 10%. The parameters  $p_1$  and  $p_2$  define how many iterations are performed in each step and our default choice is  $p_1 = p_2 = 5$ . Using these parameters, the algorithm forces to divide even those intervals on the slices where the error is lower in the second step as there could be some hidden variation of the signal. In the third step, only the intervals where the high cross-validation error prevails are divided. Although the algorithm is not very sensitive to the parameters, one should not decrease the total number of iterations  $p_1 + p_2$  too much as intervals on the slices could not be divided sufficiently then. On the contrary, using too many iterations (up to addition of only one new sample during an iteration) is not recommended either as the algorithm would be too focused on the specular highlights and might overlook variations of the signal in unexplored intervals.

The ASI function demands the table  $T$  of already measured directions and their values and the count of samples  $n_i$  that will be identified and measured by the function. The output of the function is the appended table  $T$ . The function itself consists of three steps. First, the leave-one-out cross-validation error is evaluated in each already measured sample. The error is evaluated independently for each slice, i.e., the sample at the intersection of the slices has several error values. The evaluation for each sample is done by exclusion of the evaluated sample from the dataset, linear interpolation of a value in the location of the evaluated sample from the neighboring samples, and computation of a distance of the interpolated value and the actual value of the evaluated sample (see Fig. 17-right).

As a distance measure, we use the maximum difference over all color channels. Then, in the second step of the ASI function, a list of all directions where new measurement could be performed is prepared. Each direction has assigned a weight equal to the maximum of cross-validation errors of its already measured neighbors (see Fig. 17-right). The list is then sorted in descending manner, values of  $n_i$  first directions from the list are measured, and the table  $T$  is appended by the newly measured directions and their values.

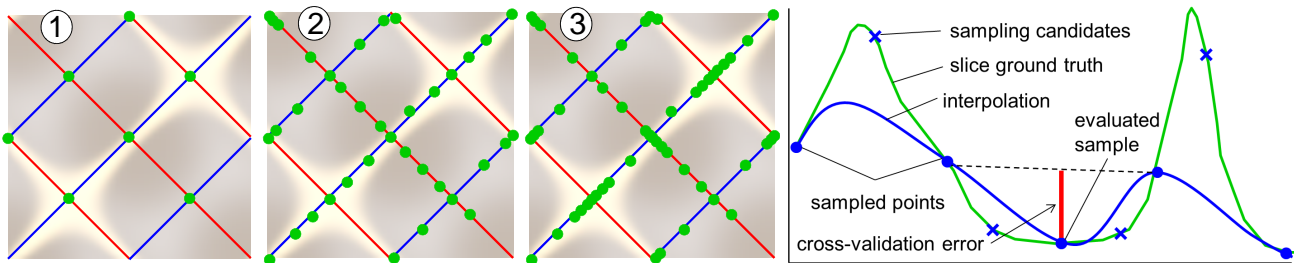


Figure 17: A description of individual steps of the adaptive sampling algorithm (Algorithm 1), and a description of the cross-validation procedure (right).

---

**Algorithm 1** Adaptive sampling along the slices

---

**Input:**  $n, k, p_1, p_2$ .

**Output:**  $T$  - a table of measured directions and their values.

- 1: Measure values of all  $n_0$  samples at all intersections of the BRDF slices creating  $T$ .
  - 2:  $p_1$ -times perform:  $T = \text{ASI}(T, [n - n_0] \cdot k/p_1)$ .
  - 3:  $p_2$ -times perform:  $T = \text{ASI}(T, [n - n_0] \cdot [1-k]/p_2)$ .
- 

Once the measurement is done, we can reconstruct values of the four-dimensional BRDF by the method introduced in [FVH\*13]. In the supplementary material, we show how to rewrite their equations to make transition to four dimensions possible. Also, we provide there equations for reconstruction of the desired value in the four dimensional space, and we describe implementation of the equations on graphics hardware.



## 6 Stability of the Slices Placement

To test stability of the proposed selection of placement of the slices we perform the following experiment. We compute optimal values of the parameters  $|m|$ -times using Equation 1; each time with one material left out of the computation.

$$(\hat{a}_s, \hat{e}_s)(n) = \arg \min_{(a'_s \in a_s, e'_s \in e_s)} \sum_{m=1}^{|m|} \frac{err_{m, a'_s, e'_s}(n)}{\min_{(a''_s \in a_s, e''_s \in e_s)} err_{m, a''_s, e''_s}(n)}. \quad (1)$$

As a consequence there are  $|m|$  different results, and we compare them with the results in Table 3. This is graphically illustrated in Figure 18, where the first line corresponds to the computation for all  $|m|$  materials, and each other line corresponds to the computation with one material left out.

Table 3: Optimal values of parameters  $a_s, e_s$  depending on demanded count of samples.

$n \leq$	$e_s$	$a_s$	$n \leq$	$e_s$	$a_s$
667	28°	180°	17 096	12°	36°
932	20°	180°	20 969	10°	36°
1 034	16°	180°	22 291	8°	36°
1 060	28°	60°	33 879	12°	20°
2 272	20°	60°	38 735	10°	20°
3 230	16°	60°	79 469	8°	20°
4 928	14°	60°	184 655	6°	20°
5 645	16°	36°	$\infty$	6°	12°
9 660	14°	36°			

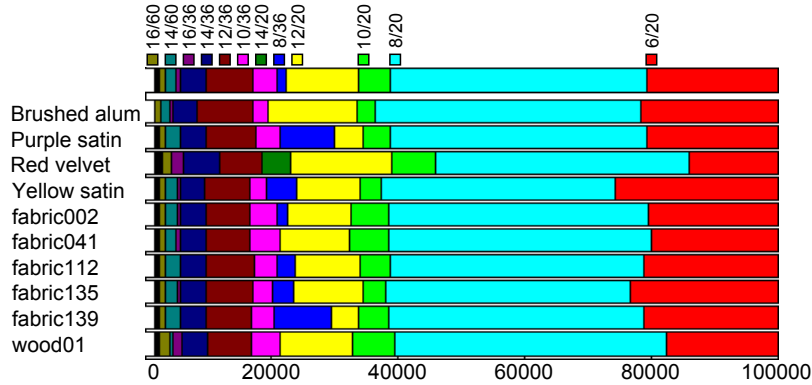


Figure 18: First line: the color coded optimal placement of the slices in terms of  $e_s/a_s$  for various count of samples. Other lines: the stability test of the optimal placement of the slices. At each line one material is left out from computation of the optimal placement.

As one can see, boundaries of individual regions (representing combinations of parameters  $e_s/a_s$ ) vary slightly but not dramatically. When we leave material *Red velvet* out, there is a green region (14/20) which does not appear on other lines substituting purple (10/36) and blue (8/36) regions. When we leave material *Brushed alum*, *fabric041* or *wood01* out, the blue region (8/36) is missing. Note that usage of the parameters of neighboring regions brings only slightly worse reconstruction errors.

We evaluate increase of the MRE due to usage of the optimal placement of the BRDF slices according to Table 3 instead of usage of the best placement for the tested material (which is unknown for a newly measured material). Evaluation is performed across a broad count ranging from 55 samples (resulting from count of intersections of the slices in the sparsest configuration  $e_s = 28^\circ/a_s = 180^\circ$ ) to 354 061 samples (resulting from the densest tested uniform distribution of samples across a hemisphere, see Sec. 1). Results are summarized in Table 4. While maximal increase of the error is almost 1.5-times, the average MRE across all the materials is only 1.2-times worse than if we use the best possible placement of the slices. We conclude that usage of the optimal placement according to Equation 1 is a good practice in the measurement of an unknown material.

Table 4: Increase of the MRE using the optimal placement of the slices instead of the best possible placement.

Brush. alum	Purple satin	Red velvet	Yellow satin	fabric 002	fabric 041	fabric 112	fabric 135	fabric 139	wood 01	mean
1.31×	1.19×	1.49×	1.32×	1.02×	1.03×	1.05×	1.27×	1.11×	1.17×	1.20×

## 7 Reconstruction from the Sparse Structure

This section rewrites the equations for interpolation in the 2D BRDF subspace introduced by Filip et al. (2013) to make transition into the 4D space possible. Then, equations for interpolation of any value in the BRDF space from the sparse 4D structure (in the form of four types of the BRDF slices) are provided.

It can be proven that changing interpolation order of  $x, y$  axes yields the same result. Also, the equations can be rewritten as follows to make transition into four dimensions possible. First, we apply bilinear interpolation of corner values introducing

$$c_{xy} = (1 - x) \cdot [(1 - y) \cdot c_{00} + y \cdot c_{01}] + x \cdot [(1 - y) \cdot c_{10} + y \cdot c_{11}] .$$

Values of the slices are interpolated linearly and differences are computed as:

$$\begin{aligned} p_{xy} &= (1 - y) \cdot p_{x0} + y \cdot p_{x1} , \\ q_{xy} &= (1 - x) \cdot q_{0y} + x \cdot q_{1y} , \\ \Delta p_{xy} &= p_{xy} - c_{xy} , \\ \Delta q_{xy} &= q_{xy} - c_{xy} . \end{aligned}$$

The final value of the reconstructed function is

$$\begin{aligned} r_{xy} &= \max(c_{xy} + \Delta p_{xy} + \Delta q_{xy}, \min_{xy}) = \max(p_{xy} + q_{xy} - c_{xy}, \min_{xy}) , \\ \min_{xy} &= \min(p_{x0}, p_{x1}, q_{0y}, q_{1y}) . \end{aligned}$$

Extension into four dimensions is straightforward. We apply multi-linear interpolation to all sixteen corners of the four-dimensional hypercube  $c_{\bar{x}\bar{y}\bar{z}\bar{w}}$ , where  $\bar{x}, \bar{y}, \bar{z}, \bar{w} \in \{0, 1\}$ :

$$\begin{aligned} c_{xyzw} &= (1 - x) \cdot \left\{ (1 - y) \cdot \left\{ (1 - z) \cdot [(1 - w) \cdot c_{0000} + w \cdot c_{0001}] \right. \right. \\ &\quad + z \cdot [(1 - w) \cdot c_{0010} + w \cdot c_{0011}] \\ &\quad \left. \left. + y \cdot \left\{ (1 - z) \cdot [(1 - w) \cdot c_{0100} + w \cdot c_{0101}] \right. \right. \right. \\ &\quad \left. \left. \left. + z \cdot [(1 - w) \cdot c_{0110} + w \cdot c_{0111}] \right\} \right\} \\ &\quad + x \cdot \left\{ (1 - y) \cdot \left\{ (1 - z) \cdot [(1 - w) \cdot c_{1000} + w \cdot c_{1001}] \right. \right. \\ &\quad + z \cdot [(1 - w) \cdot c_{1010} + w \cdot c_{1011}] \\ &\quad \left. \left. + y \cdot \left\{ (1 - z) \cdot [(1 - w) \cdot c_{1100} + w \cdot c_{1101}] \right. \right. \right. \\ &\quad \left. \left. \left. + z \cdot [(1 - w) \cdot c_{1110} + w \cdot c_{1111}] \right\} \right\} . \end{aligned}$$

Then, values of the slices are interpolated in three remaining dimensions (see Fig. 3). Values of the eight axial slices  $p_{x\bar{y}\bar{z}\bar{w}}$ , where  $\bar{y}, \bar{z}, \bar{w} \in \{0, 1\}$  are interpolated as

$$\begin{aligned} p_{xyzw} &= (1 - y) \cdot \left\{ (1 - z) \cdot [(1 - w) \cdot p_{x000} + w \cdot p_{x001}] + z \cdot [(1 - w) \cdot p_{x010} + w \cdot p_{x011}] \right\} \\ &\quad + y \cdot \left\{ (1 - z) \cdot [(1 - w) \cdot p_{x100} + w \cdot p_{x101}] + z \cdot [(1 - w) \cdot p_{x110} + w \cdot p_{x111}] \right\} , \end{aligned}$$

values of the eight diagonal slices  $q_{\bar{x}y\bar{z}\bar{w}}$ , where  $\bar{x}, \bar{z}, \bar{w} \in \{0, 1\}$  are interpolated as

$$\begin{aligned} q_{xyzw} &= (1 - x) \cdot \left\{ (1 - z) \cdot [(1 - w) \cdot q_{0y00} + w \cdot q_{0y01}] + z \cdot [(1 - w) \cdot q_{0y10} + w \cdot q_{0y11}] \right\} \\ &\quad + x \cdot \left\{ (1 - z) \cdot [(1 - w) \cdot q_{1y00} + w \cdot q_{1y01}] + z \cdot [(1 - w) \cdot q_{1y10} + w \cdot q_{1y11}] \right\} , \end{aligned}$$

values of the eight horizontal slices  $s_{\bar{x}\bar{y}z\bar{w}}$ , where  $\bar{x}, \bar{y}, \bar{w} \in \{0, 1\}$  are interpolated as

$$\begin{aligned} s_{xyzw} &= (1 - x) \cdot \left\{ (1 - y) \cdot [(1 - w) \cdot s_{00z0} + w \cdot s_{00z1}] + y \cdot [(1 - w) \cdot s_{01z0} + w \cdot s_{01z1}] \right\} \\ &\quad + x \cdot \left\{ (1 - y) \cdot [(1 - w) \cdot s_{10z0} + w \cdot s_{10z1}] + y \cdot [(1 - w) \cdot s_{11z0} + w \cdot s_{11z1}] \right\} , \end{aligned}$$

and values of the eight vertical slices  $t_{\bar{x}\bar{y}\bar{z}w}$ , where  $\bar{x}, \bar{y}, \bar{z} \in \{0, 1\}$  are interpolated as

$$\begin{aligned} t_{xyzw} &= (1 - x) \cdot \left\{ (1 - y) \cdot [(1 - z) \cdot t_{000w} + z \cdot t_{001w}] + y \cdot [(1 - z) \cdot t_{010w} + z \cdot t_{011w}] \right\} \\ &\quad + x \cdot \left\{ (1 - y) \cdot [(1 - z) \cdot t_{100w} + z \cdot t_{101w}] + y \cdot [(1 - z) \cdot t_{110w} + z \cdot t_{111w}] \right\} . \end{aligned}$$

Final value of reconstructed function is

$$\begin{aligned} r_{xyzw} &= \max(p_{xyzw} + q_{xyzw} + s_{xyzw} + t_{xyzw} - 3 \cdot c_{xyzw}, \min_{xyzw}) , \\ \min_{xyzw} &= \min(\min_p, \min_q, \min_s, \min_t) , \\ \min_p &= \min(p_{x000}, p_{x001}, p_{x010}, p_{x011}, p_{x100}, p_{x101}, p_{x110}, p_{x111}) , \\ \min_q &= \min(q_{0y00}, q_{0y01}, q_{0y10}, q_{0y11}, q_{1y00}, q_{1y01}, q_{1y10}, q_{1y11}) , \\ \min_s &= \min(s_{00z0}, s_{00z1}, s_{01z0}, s_{01z1}, s_{10z0}, s_{10z1}, s_{11z0}, s_{11z1}) , \\ \min_t &= \min(t_{000w}, t_{001w}, t_{010w}, t_{011w}, t_{100w}, t_{101w}, t_{110w}, t_{111w}) . \end{aligned}$$

## 8 Graphics Hardware Implementation

Another key aspect that sets our method apart from the RBF interpolation is its feasibility for very fast interpolation on a graphics hardware (GPU). Although the barycentric interpolation allows for this as well, it does not support an adaptive non-uniform placement of the samples and thus cannot achieve a high reconstruction performance.

To reconstruct a BRDF on a GPU using our method one needs to store data in four textures. The first texture contains values at intersections of the slices, the second contains pre-interpolated values of the axial slices, the third contains pre-interpolated values of the diagonal slices and the fourth texture contains pre-interpolated values of the horizontal slices (identical to the vertical slices). The first three textures can be easily stacked together. Note that the interpolation along the slices has to be precomputed to avoid intricate lookups for measured values. Despite that, the data are still very compact. Table 5 compares the size of the data structures required by the tested methods using 8911 and 18721 reciprocal samples.

Table 5: The GPU storage space required by the barycentric and proposed interpolation methods for 8 911 samples ( $e_s = 14^\circ/a_s = 36^\circ$ ) and 18 721 samples ( $e_s = 10^\circ/a_s = 36^\circ$ ).

method	8 911 samples	18 721 samples
barycentric	3.5 MB (scheme 14)	3.6 MB (scheme 19)
proposed	3.3 MB	5.1 MB

Rendering speed on the GPU is 270 FPS for the barycentric interpolation and 130 FPS for the proposed method regardless of the complexity of a 3D scene as the majority of the computation is performed by the fragment shader rendering window of size  $800 \times 800$  pixels. The timings are obtained using nVidia GeForce GTX 570 graphics card. The slower performance of the proposed approach results from the higher complexity of the BRDF reconstruction from the slices, which is in contrast to implementation of the barycentric interpolation using fast indexing in precomputed cube-maps stored as textures. The visual performance of the GPU rendering is identical to that of the CPU rendering, i.e., the proposed method clearly provides superior visual quality to the barycentric interpolation.

Let us describe content of the four textures in detail. The first texture containing values at intersections of the slices consists of  $\frac{e_n(e_n+1)}{2}$  areas, where  $e_n$  is count of elevations (see Fig. 3). Note that due to Helmholtz reciprocity we do not need to store values for subspaces where  $\theta_i > \theta_v$ . Each area covers  $(a_n + 1) \times (2a_n + 1)$  pixels, where  $a_n$  is count of slices of one type (axial or diagonal) in a subspace and for simplicity we assume that the number of axial slices is equal to the number of diagonal slices. Then,  $2a_n$  is the number of intersections in each subspace. An additional row and column in each area is there due to easier interpolation of the values, which can be performed implicitly by a GPU during texture value lookup and explicit interpolation has to be performed in two remaining dimensions ( $\theta_v, \theta_i$ ) only.

The second texture containing values of the axial slices also consists of  $\frac{e_n(e_n+1)}{2}$  areas. Width of each area is  $a_n + 1$  as one axial slice is repeated for easier interpolation and height of the texture depends on a selected resolution (e.g., for resolution  $1^\circ$  the height is 360 pixels). As interpolation in two dimensions is provided by a GPU during texture value lookup, only interpolation in two other dimensions has to be implemented. The third texture covers the same area as the second texture, contains values of the diagonal slices and is interpolated in the same manner.

The last texture contains values of all horizontal slices. There is  $2a_n^2$  horizontal slices for each measured illumination elevation. Therefore, with padding for easier interpolation, width of the texture is  $a_n(2a_n + 1)$  and height is, e.g.,  $90 \cdot e_n$  for resolution  $1^\circ$ . Interpolation in two dimensions is, again, performed during texture value lookup by a GPU and only two remaining dimensions has to be interpolated explicitly.