Sony Pictures Imageworks's Lighting Model Integration Report

2020/07/10

SQUARE ENIX CO., LTD.
Advanced Technology Division
Mizokuchi Tomohiro
Vongsay Kitti

Agenda

- Introduction
- Goals
- Energy Conservation
- Environment Lighting
- Optimization
- Results
- Future Work

Introduction

- We started to integrate the Sony Pictures Imageworks lighting model.
  - Revisiting Physically Based Shading at Imageworks [Kulla 2017].

☺ Pros

- Perfect conservation of specular energy.
- Perfect conservation of specular and diffuse energy for dielectrics.

☹ Cons

- Energy is not conserved for semiconductor materials.
- No information on how to implement environment lighting.

Goals

- Energy conservation for semiconductors.
- Real-time rendering of environment lighting.
  - Based on Split Sum Integral [Karis 2013].
- Approximate the LUTs (Look Up Table) with some basis functions to reduce their size.

Sony Pictures Imageworks

Ours

ENERGY CONSERVATION

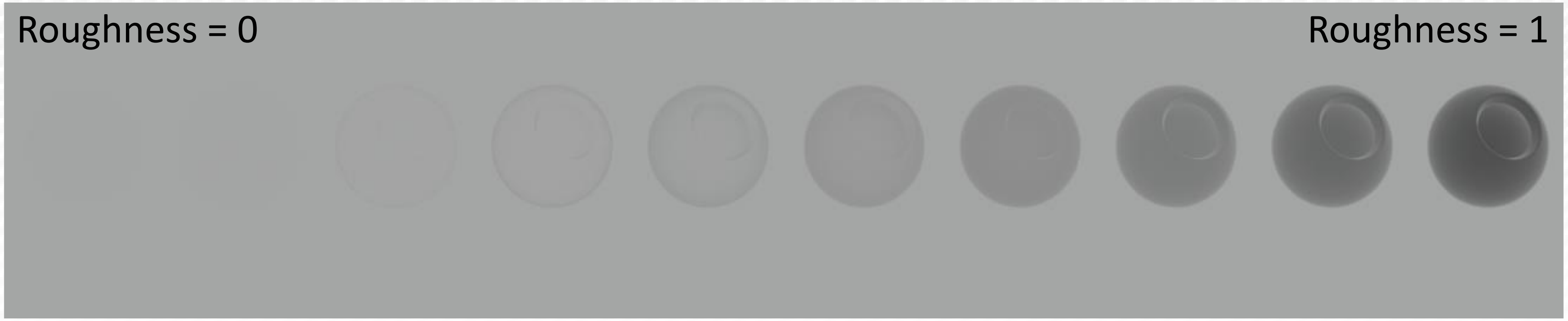
Energy Conservation - Imageworks

- Specular Term
  - Normal distribution is GGX and height-correlated masking & shadowing.
  - The specular assume single-scattering, becoming darker for higher roughness values.
- Matte Term
  - Kelemen et al model is used to conserve energy [Kelemen 2001].
  - Specular is rendered with energy loss.
  - That energy loss is compensated for with Lambertian diffuse models.
- Diffuse Term
  - Kelemen et al model is also used for the diffuse term.
  - Diffuse lighting is rendered with the absorbed energy of specular and matte.
  - Energy conservation can only be guaranteed for dielectric materials.

Energy - Specular Term

$$E(\mu_o) = \int_0^{2\pi} \int_0^1 f'\underbrace{(\mu_o, \mu_i, \phi)}_{\text{Microfacet Specular BRDF}} \mu_i d\mu_i d\phi \,.$$

- $E(\mu_o)$ is referred to as the directional albedo.
- Greek letter μ is $cos\theta$ term.
- Furnace test was used to visualize lost energy.



Furnace test

Energy - Matte Term

$$f'_{ms}(\mu_o,\mu_i) = \frac{\big(1 - E(\mu_o)\big)\big(1 - E(\mu_i)\big)}{\pi(1 - E_{avg})}, E_{avg} = 2\int_0^1 E(\mu)\mu d\mu\,.$$

- As introduced by Kelemen et al.
- By adding the new BRDF lobe f'_{ms}, energy is perfectly conserved.

Furnace test

Fresnel

- For dielectric materials, some energy is absorbed.
- Multi-scattering lights absorption needs to be considered.
  - Can be roughly modeled by averaging Fresnel with a cosine weight [Jakob 2014].
  - $F_{avg} = 2\int_0^1 F(\mu)\mu d\mu$.
  - $F_{ms} = \frac{(F_{avg}E_{avg})}{1 - F_{avg}(1 - E_{avg})}$.
- F_{ms} is multiplied by the matte term.
- Our integration used Artist Friendly Fresnel [Gulbrandsen 2014] as F.
  - Artists can tweak the edge color with **fresnel specular (f0)** and **edgetint**.

Energy - Diffuse Term

- Simply adding the diffuse term will not conserve energy.
- Kelemen et al's approach is also used for the diffuse term.
- The amount of energy depends on the specular and matte terms.
  - Some energy is absorbed by the specular and matte terms.
- Diffuse term is only considered for dielectric materials.
  - Edgetint is fixed at 0.
- So if the material is a semiconductor, the energy will not be conserved.

Roughness = 0 f0 = 0.5, edgetint = 1, albedo = 1 Roughness = 1

Energy Visualization

Dependency

- E: directional albedo of specular.
  - Dependency: $\mu_o, roughness$.
- E_{avg}: averaged directional albedo of specular.
  - Dependency: $roughness$.
- E': absorbed energy by specular and matte terms (edgetint=0).
  - Dependency: $\mu_o, roughness, fresnel\ specular$.
- E'_{avg}: specular and matte terms absorbed energy average.
  - Dependency: $roughness, fresnel\ specular$

Sony Pictures Imageworks

Ours

ENERGY CONSERVATION

Edgetint

- Diffuse term is only considered for dielectric materials.
  - Edgetint is fixed at 0.
- If the material is close to a semiconductor, the energy will not be conserved.
- Edgetint is close to 1 for real world conductors.
- Edgetint is close to 0 for real world dielectrics.
- ➢ We conserve energy by scaling the diffuse term for dielectrics.
  - ➢ The scale will determine edgetint value.

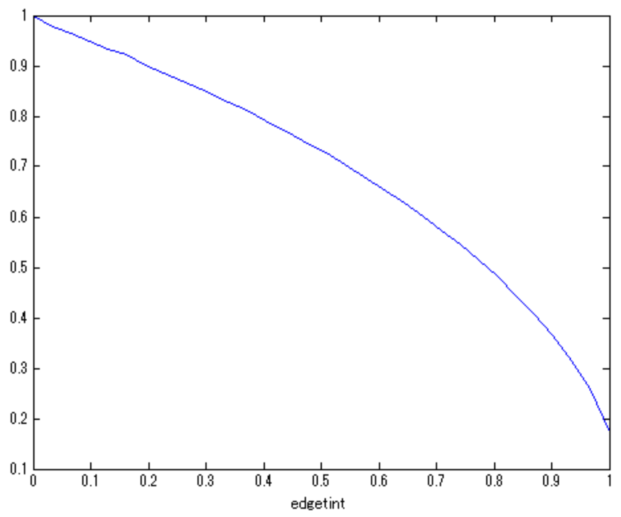
Scaling Parameter

- We determine the scale required to conserve energy for semiconductors.

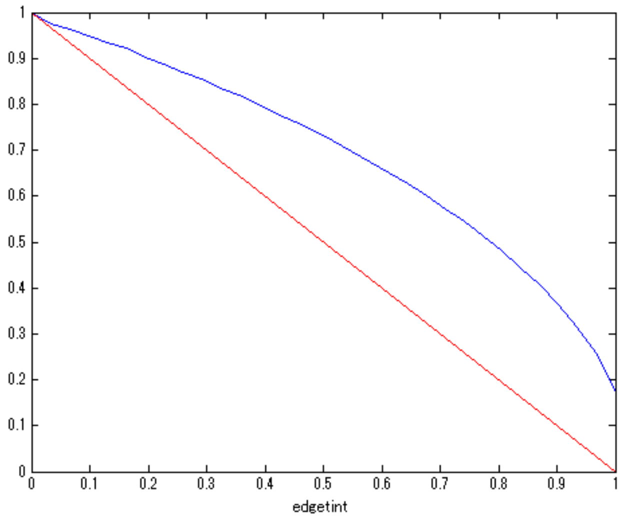
1. Calculate the absorbed energy E'' of specular and matte terms.
   - Edgetint isn't fixed.
   - Dependency: $\mu_o, roughness, fresnel\ specular, edgetint$.
2. Calculate the parameter k that is the ratio of E'' and E'.
   - $k = \frac{E''}{E'}$
   - The dimension of k is 4 ($\mu_o, roughness, fresnel\ specular,\ edgetint$).
3. Minimize k into a dimension of 3 ($\mu_o, roughness, fresnel\ specular$).
4. Finally k end up as a 1D LUT depending on edgetint.

Scaling Parameter

- Energy is conserved by multiplying the diffuse term by k.
- k is the only condition to preserve energy conservation.
- We tried to fit the 1D LUT at first but the calculation costs seemed excessive so we decided to approximate k as 1-edgetint.



The visualization of k

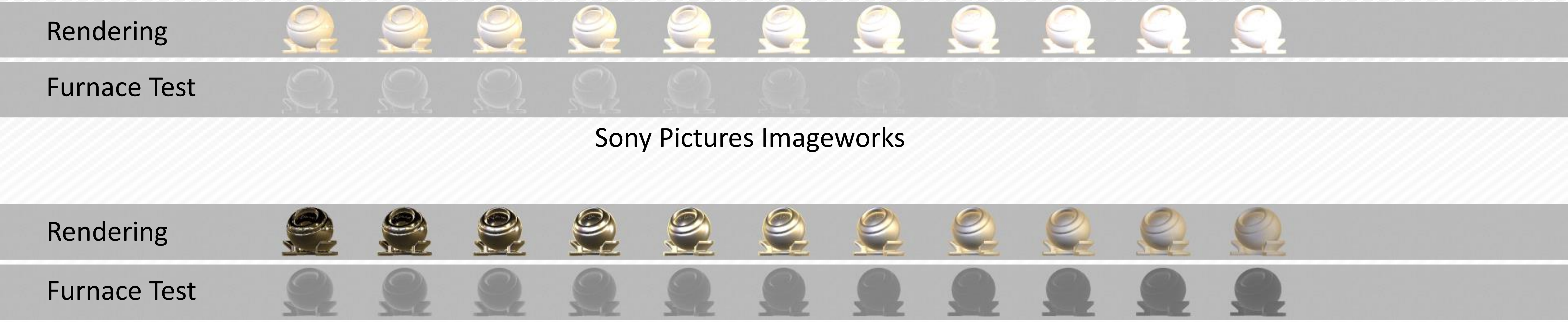


Approximated k (blue) as 1 – edgetint (red)

Energy Conservation

- Sony Pictures Imageworks: Energy is not conserved when roughness is close to 0.
- Ours: Energy is conserved for any roughness values.

Roughness = 0 | Energy visualization (f0 = 0.5, edgetint = 1, albedo = 1) | Roughness = 1



Ours

Importance Sampling

Split Integral

ENVIRONMENT LIGHTING

Importance Sampling

- Importance sampling is used for offline rendering.
  - Can sample the radiance from a cube map against a BRDF lobe.

☺ Pros

- Integration is very simple.
- Can do the fast rendering with filtered importance sampling [Krivanek 2008].

☹ Cons

- Calculation cost is very expensive.

Split Integral (Unreal)

- This is often used for environment lighting in games.
  - Real Shading in Unreal Engine 4 [Karis 2014].
- $\int_\Omega L_i(\mathbf{l}) f(\mathbf{l}, \mathbf{v}, \mathbf{n})\, |\mathbf{n} \cdot \mathbf{l}|\, d\mathbf{l} = \frac{\int_\Omega L_i(\mathbf{l}) f(\mathbf{l},\mathbf{v},\mathbf{n}) |\mathbf{n} \cdot \mathbf{l}| d\mathbf{l}}{\int_\Omega f(\mathbf{l},\mathbf{v},\mathbf{n}) |\mathbf{n} \cdot \mathbf{l}| d\mathbf{l}} \int_\Omega f(\mathbf{l}, \mathbf{v}, \mathbf{n}) |\mathbf{n} \cdot \mathbf{l}| d\mathbf{l}$.

$$\approx \frac{F(|\mathbf{n} \cdot \mathbf{v}|) \int_\Omega L_i(\mathbf{l}) \acute{f}(\mathbf{l}, \mathbf{v}, \mathbf{n}) |\mathbf{n} \cdot \mathbf{l}| d\mathbf{l}}{F(|\mathbf{n} \cdot \mathbf{v}|) \int_\Omega \acute{f}(\mathbf{l}, \mathbf{v}, \mathbf{n}) |\mathbf{n} \cdot \mathbf{l}| d\mathbf{l}} \int_\Omega f(\mathbf{l}, \mathbf{v}, \mathbf{n}) |\mathbf{n} \cdot \mathbf{l}| d\mathbf{l}.$$

Fresnel

BRDF w/o Fresnel

Pre-filtered cube map

Directional albedo

l: incident light direction
v: view direction
n: normal
Ω: hemisphere

- Schlick Fresnel [Schlick 1994] is used as the fresnel term F.
- This isn't directly applicable to our lighting model.

Our Directional Albedo

Our Pre-filtered Cube Map

SPLIT INTEGRAL

Directional Albedo

$$\int_{\Omega} f(\mathbf{l}, \mathbf{v}, \mathbf{n})|\mathbf{n} \cdot \mathbf{l}|d\mathbf{l} = E_{specular} + E_{matte} + albedo \ * \ E_{diffuse}.$$

- $E_{specular}$: Specular term's directional albedo.
- E_{matte} : Matte term's directional albedo.
- $E_{diffuse}$: Diffuse term's directional albedo.

$E_{specular}$

$$E_{specular} = \int_{\Omega} f(\mathbf{l}, \mathbf{v}, \mathbf{n})|\mathbf{n} \cdot \mathbf{l}| d\mathbf{l}$$
$$\approx F(\mathbf{v}, \boldsymbol{d}) \int_{\Omega} f'(\mathbf{l}, \mathbf{v}, \mathbf{n})|\mathbf{n} \cdot \mathbf{l}| d\mathbf{l}$$
$$\approx F(\mathbf{v}, \boldsymbol{d}) E(\mu_o).$$

- $\boldsymbol{d}$: Dominant direction of the reflection lobe.
- We extract from the integral F to reduce the dimension of the LUT.

E_{matte}

$$E_{matte} = F_{ms} \int_{\Omega} f'_{ms}(\mathbf{l}, \mathbf{v}, \mathbf{n}) |\mathbf{n} \cdot \mathbf{l}| d\mathbf{l}$$

$$= F_{ms} \left(1 - \int_{\Omega} f'(\mathbf{l}, \mathbf{v}, \mathbf{n}) |\mathbf{n} \cdot \mathbf{l}| d\mathbf{l} \right)$$

$$= F_{ms}\big(1 - E_s(\mu_o)\big).$$

- E baked as 2D LUT (*roughness*, μ_o).
- E_{avg} approximated using LMS (Least Mean Squares).

$E_{diffuse}$

$$E_{diffuse} = \int_{\Omega} f_{diffuse}(\mathbf{l}, \mathbf{v}, \mathbf{n}) |\mathbf{n} \cdot \mathbf{l}| d\mathbf{l}$$

$$= 1 - \int_{\Omega} \big(F(\mathbf{v}, \boldsymbol{d}) f'(\mathbf{l}, \mathbf{v}, \mathbf{n}) + F_{ms} f_{ms}(\mathbf{l}, \mathbf{v}, \mathbf{n})\big) |\mathbf{n} \cdot \mathbf{l}| d\mathbf{l}$$

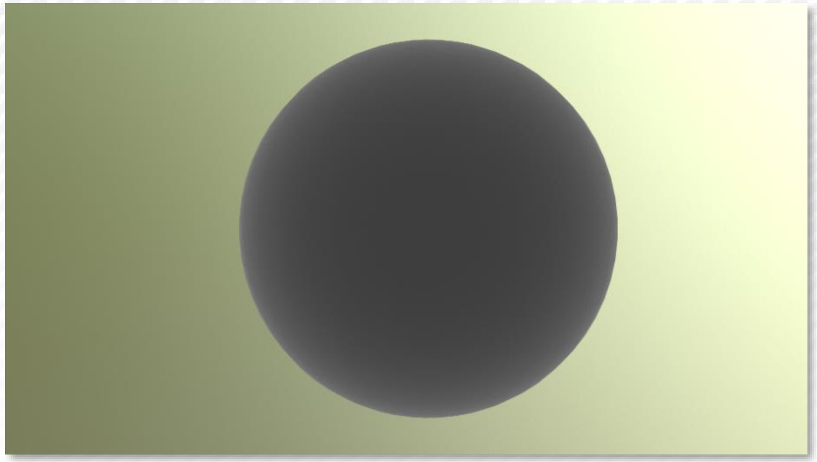
$$\approx (1 - edgetint) \left(1 - \int_{\Omega} (F\, f' + F_{ms}\, f_{ms}) |\mathbf{n} \cdot \mathbf{l}| d\mathbf{l}\right)$$

$$\approx (1 - edgetint)(1 - E'(\mu_o)).$$

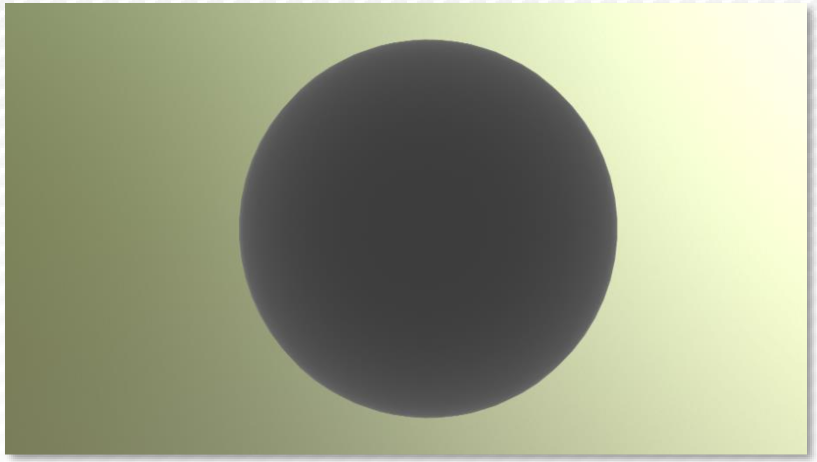
- $f_{diffuse}$ is a lobe based on Kelemen et al.
- E' baked as 3D LUT (roughness, $\mathrm{v} \cdot \mathrm{n}$, specular).
- E'_{avg} baked as 2D LUT (roughness , specular).

Fresnel Term ($F(\mathbf{v}, \boldsymbol{d})$)

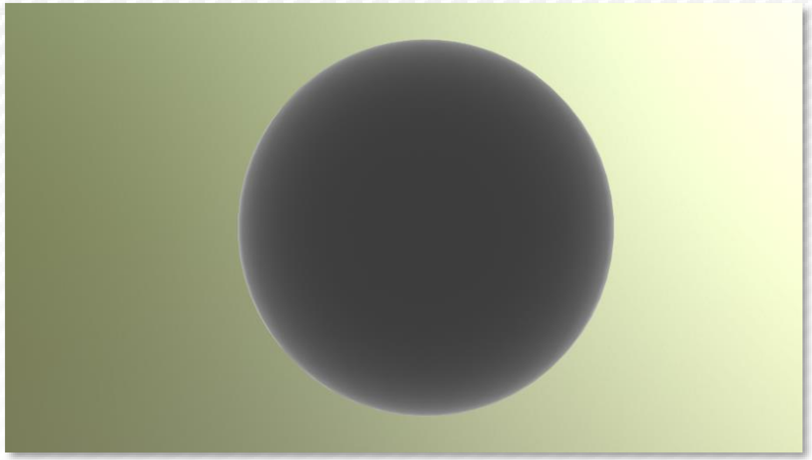
- We tested $F(\mathbf{v}, \boldsymbol{d})$ with the dominant directions for further improvements.
  - Moving Frostbite to Physically Based Rendering 3.0 [Lagarde 2014].
  - Unity Scriptable Render Pipeline [Lagarde 2018].
- We decided to use the approach of Unity Scriptable Render Pipeline.



Reference



Unity HDRP



Frostbite
(For a light probe w/o fitting curve)

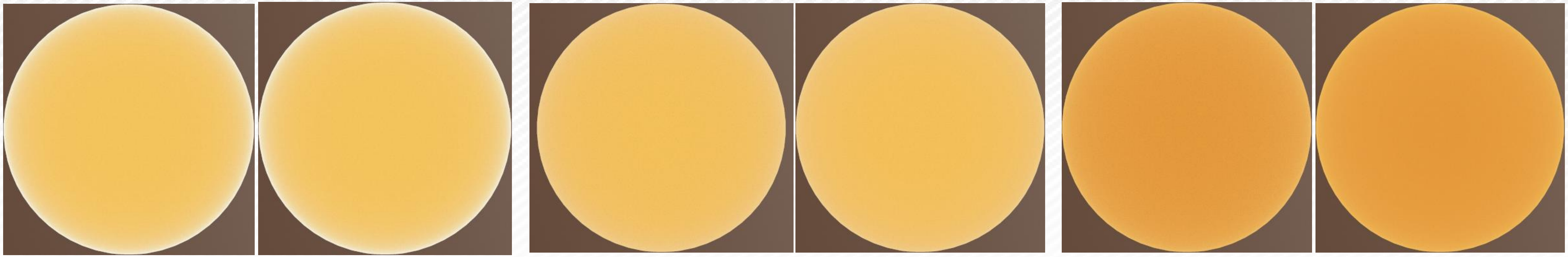
$E_{specular} + E_{matte} + E_{diffuse}$ (Roughness = 0.6, Specular = 0.27, Edgetint = 0.0)

LUT Data Size

- E : R8_UNORM, Resolution = 32*32, Data Size = 1 Kbytes.
- E' : R8_UNORM, Resolution = 32*32*32, Data Size = 32 Kbytes.
- E'_{avg}: R8_UNORM, Resolution = 32*32, Data Size = 1 Kbytes.

Result: Directional Albedo

- Material: Gold (Au).
  - Fresnel specular: 245, 197, 94.
  - Edgetint: 254, 250, 186.
  - Albedo: 0, 0, 0.

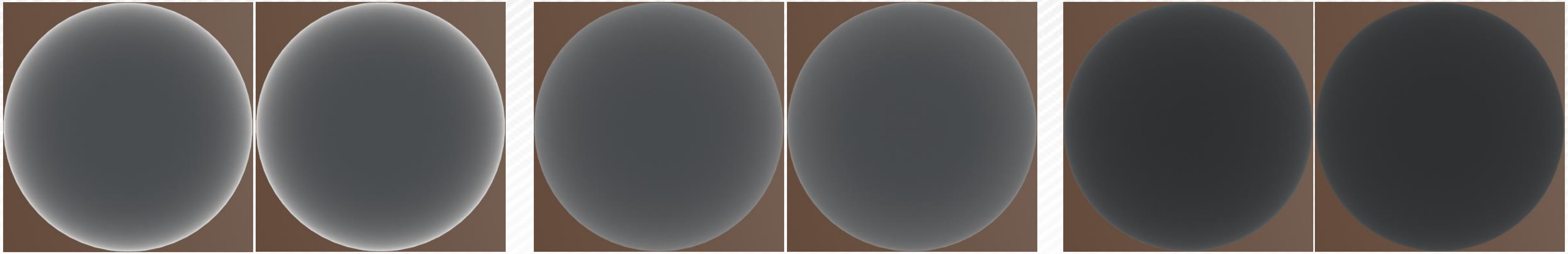


Importance Sampling | Split Integral | Importance Sampling | Split Integral | Importance Sampling | Split Integral

Roughness = 0 | Roughness = 0.5 | Roughness = 1

Result: Directional Albedo

- Material: Silicon Carbide.
  - Fresnel specular: 75, 78, 81.
  - Edgetint: 1, 4, 9.
  - Albedo: 0, 0, 0.

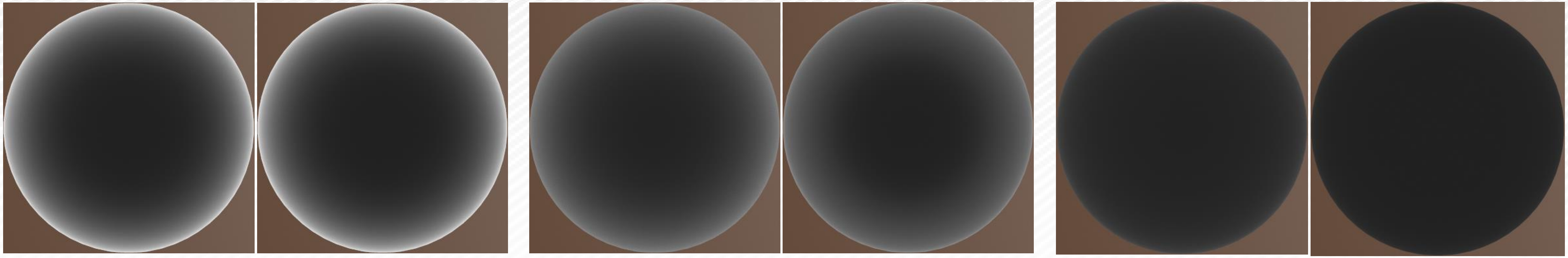


Importance Sampling | Split Integral | Importance Sampling | Split Integral | Importance Sampling | Split Integral

Roughness = 0 | Roughness = 0.5 | Roughness = 1

Result: Directional Albedo

- Material: Glass.
  - Fresnel specular: 13, 13, 13.
  - Edgetint: 0, 0, 0.
  - Albedo: 29, 29, 29.



Importance Sampling | Split Integral | Importance Sampling | Split Integral | Importance Sampling | Split Integral

Roughness = 0 | Roughness = 0.5 | Roughness = 1

Our Directional Albedo

Our Pre-filtered Cube Map

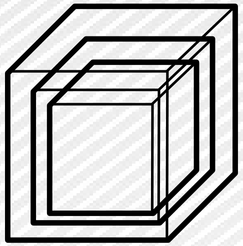
SPLIT INTEGRAL

Pre-filtered Cube Map

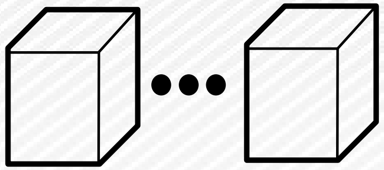
- Our pre-filtering is based on the Split Sum Integral [Karis 2013].
- Output is pre-filtered cube maps per lighting term.
  - Specular: created using $f'(\mathbf{l}, \mathbf{v}, \mathbf{n})$ weight.
  - Matte: created using $f'_{ms}(\mathbf{l}, \mathbf{v}, \mathbf{n})$ weight.
  - Diffuse: created using $f_{diffuse}$ weight.

Data Structure (Specular)

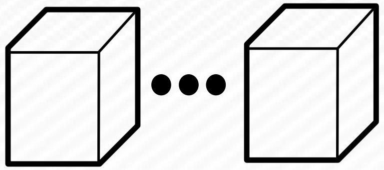
- Specular
  - Mip level represents roughness.
  - Mip level = $\sqrt{roughness}$ * mipCount.
  - The resolution is higher because of high frequency.
- Matte
  - 1D Array index is based on roughness.
  - The resolution is lower because of the low frequency.
- Diffuse
  - 2D Array index is based on roughness, specular.
  - The resolution is lower because of the low frequency.



Cube map with mip



Cube map array



Cube map array

OPTIMIZATION

First Implementation of The Pre-filtered Cube Map

- **Specular Term,** Cube map with mipmaps.
  - Format = BC6H_UF16.
  - Resolution = 512*512*6 (faces) with mipmaps.
- **Matte Term,** Cube map array.
  - Format = BC6H_UF16.
  - Resolution = 64*64*6 (faces).
  - Index = $\sqrt{roughness}$*8. ($\because 8 = The\ number\ of\ arrays$)
- **Diffuse Term,** Cube map array.
  - Format = BC6H_UF16.
  - Resolution = 64*64*6 (faces).
  - Index = ($\sqrt{roughness}$, specular)*(4, 4). ($\because (4,4) = The\ number\ of\ 2D\ arrays$)

Data Compression

- There are a lot of pre-filtered cube maps.
  - Even compressed as BC6H, it is too much data for real-time games.
- Matte term and diffuse term can be approximated with basis functions.
  - Because the cube maps are low frequency.
- We saved on data size by approximating using various basis functions.
  - Spherical Harmonics (SH): SH3, SH4, SH5.
  - Ambient Dice (AD): RGB, YCoCg, SRBF.
- Final data size.
  - Matte term: 864bytes ~ 2.4Kbytes.
  - Diffuse term: 432bytes ~ 1.2Kbytes.
- Limitation
  - Depending on cube maps, there might be negative values.

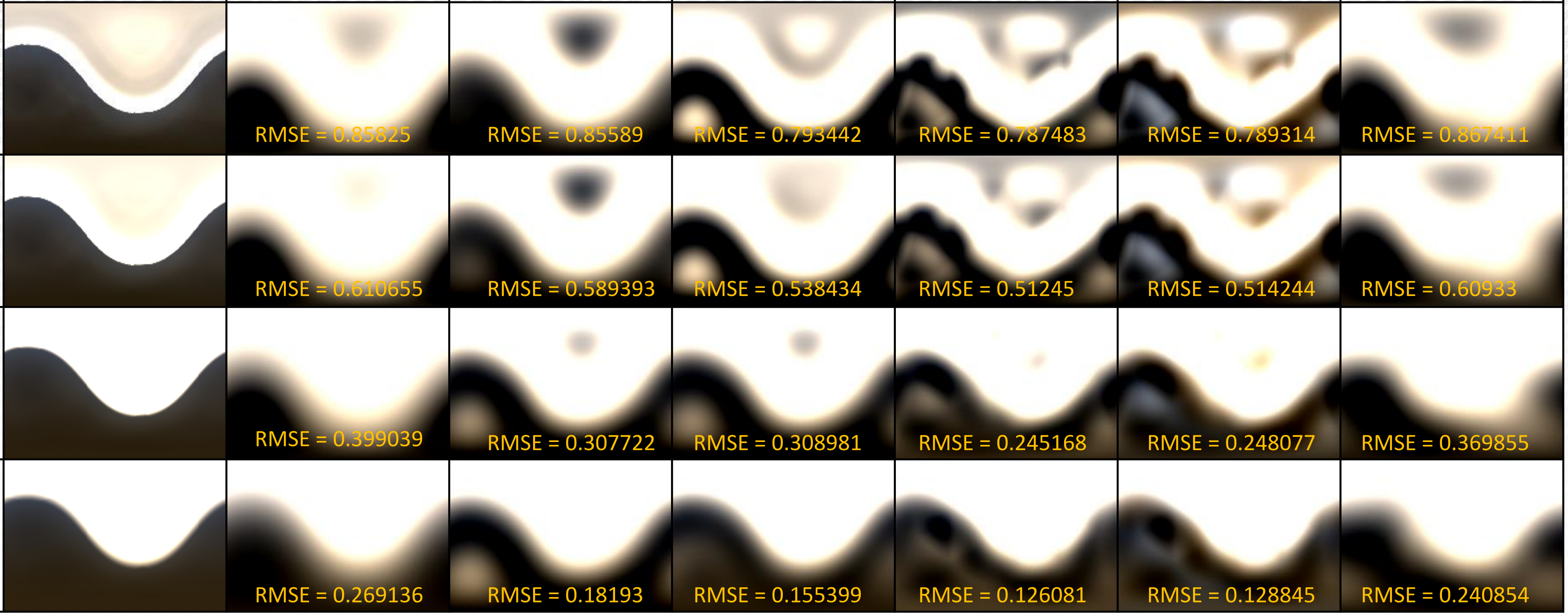
Environment map

Experiment1

Experiment2

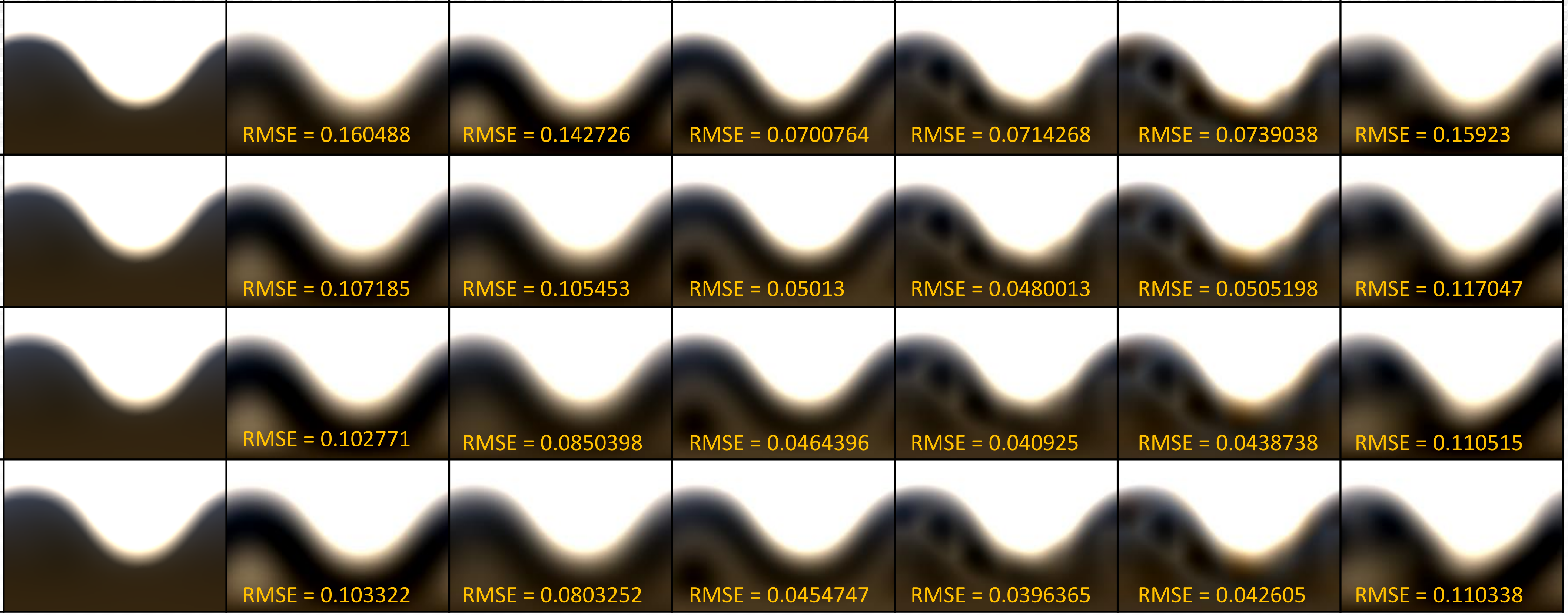
PRE-FILTERED CUBE MAP (MATTE)

Approximation (Reference Format = Cube Map Array)



$\sqrt{roughness}$	Reference	SH3	SH4	SH5	AD RGB	AD YCoCg	AD SRBF
0.002		RMSE = 0.85825	RMSE = 0.85589	RMSE = 0.793442	RMSE = 0.787483	RMSE = 0.789314	RMSE = 0.867411
0.14		RMSE = 0.610655	RMSE = 0.589393	RMSE = 0.538434	RMSE = 0.51245	RMSE = 0.514244	RMSE = 0.60933
0.28		RMSE = 0.399039	RMSE = 0.307722	RMSE = 0.308981	RMSE = 0.245168	RMSE = 0.248077	RMSE = 0.369855
0.42		RMSE = 0.269136	RMSE = 0.18193	RMSE = 0.155399	RMSE = 0.126081	RMSE = 0.128845	RMSE = 0.240854

Approximation (Reference Format = Cube Map Array)



$\sqrt{roughness}$	Reference	SH3	SH4	SH5	AD RGB	AD YCoCg	AD SRBF
0.57		RMSE = 0.160488	RMSE = 0.142726	RMSE = 0.0700764	RMSE = 0.0714268	RMSE = 0.0739038	RMSE = 0.15923
0.71		RMSE = 0.107185	RMSE = 0.105453	RMSE = 0.05013	RMSE = 0.0480013	RMSE = 0.0505198	RMSE = 0.117047
0.85		RMSE = 0.102771	RMSE = 0.0850398	RMSE = 0.0464396	RMSE = 0.040925	RMSE = 0.0438738	RMSE = 0.110515
1		RMSE = 0.103322	RMSE = 0.0803252	RMSE = 0.0454747	RMSE = 0.0396365	RMSE = 0.042605	RMSE = 0.110338

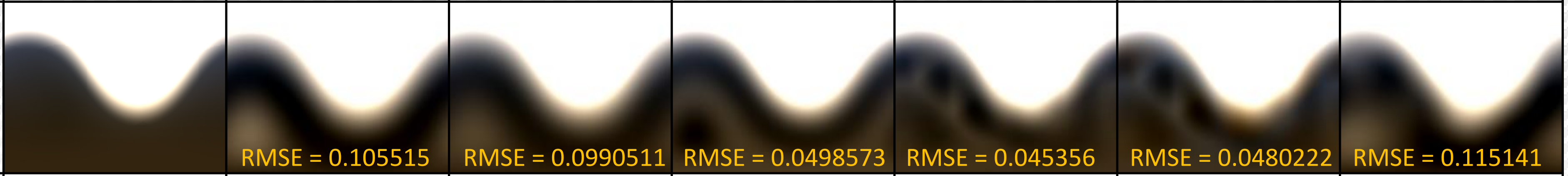
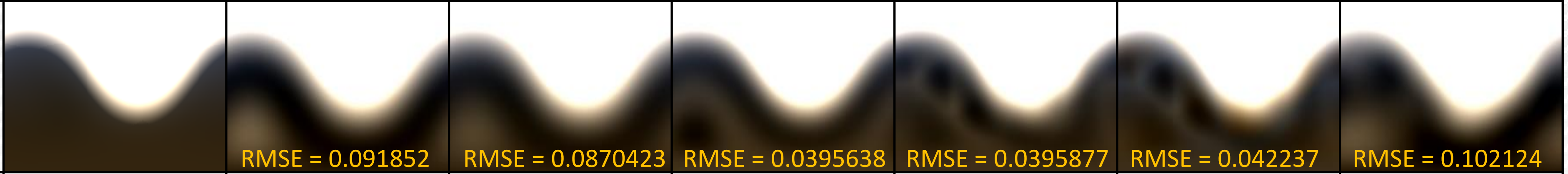
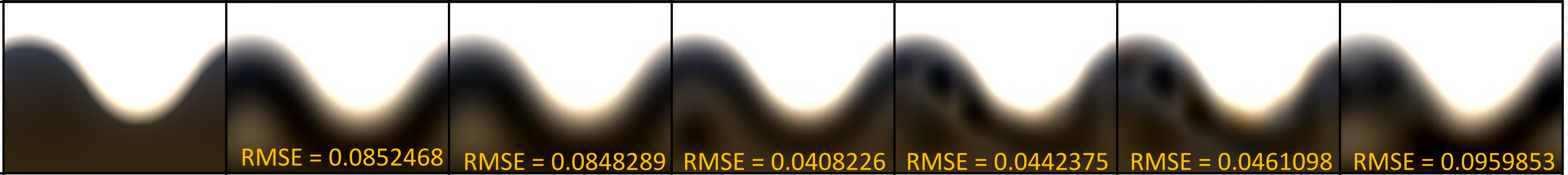
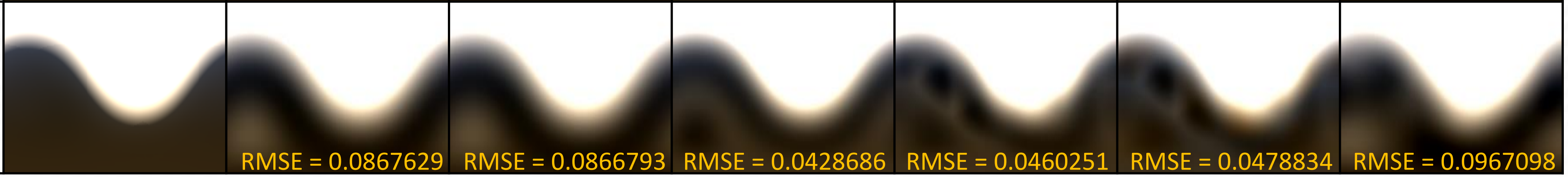
Environment map

Experiment 1

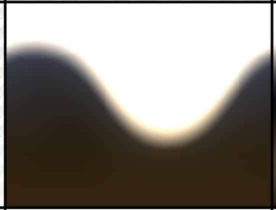
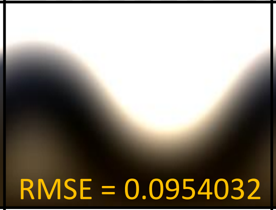
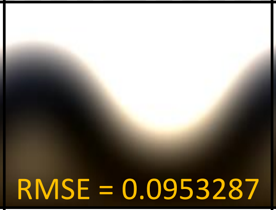
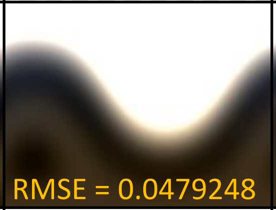
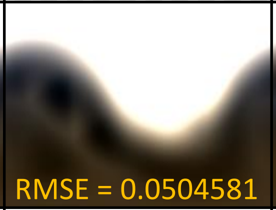
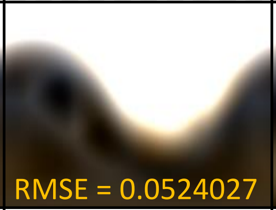
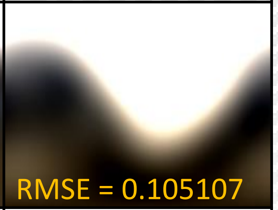
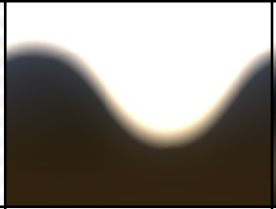
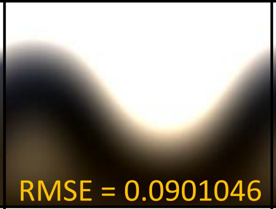
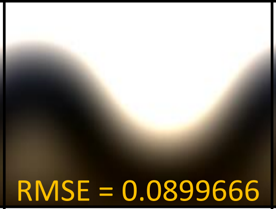
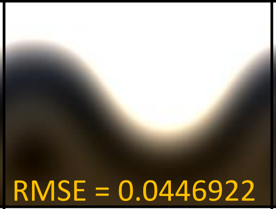
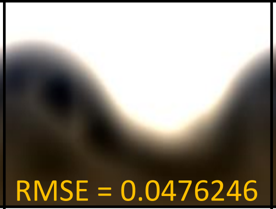
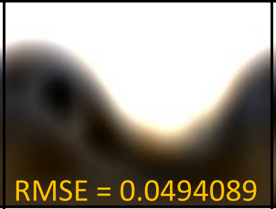
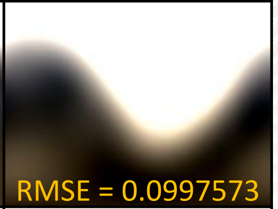
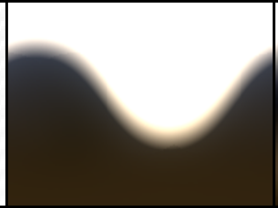
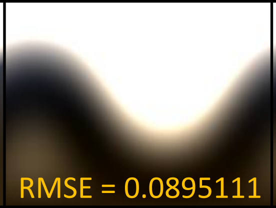
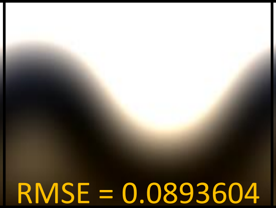
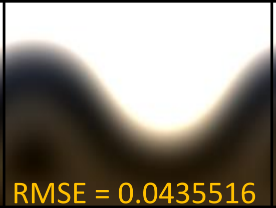
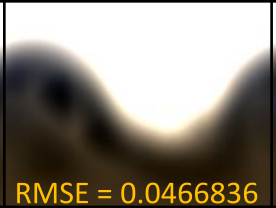
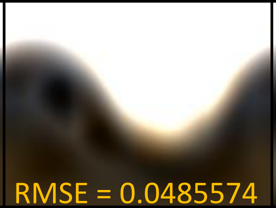
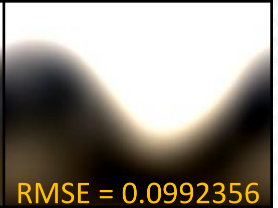
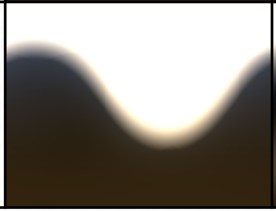
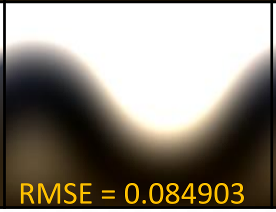
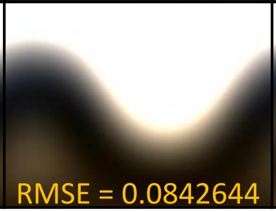
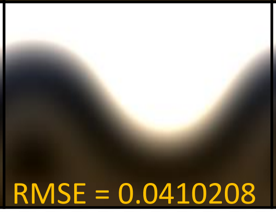
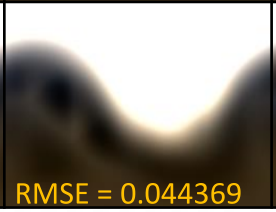
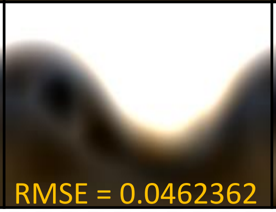
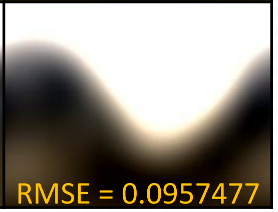
Experiment 2

PRE-FILTERED CUBE MAP (DIFFUSE)

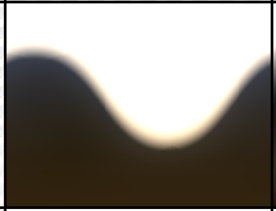
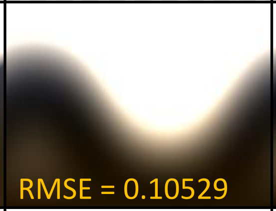
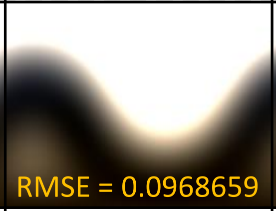
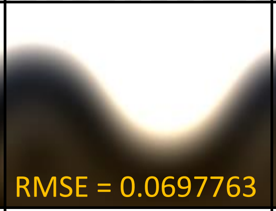
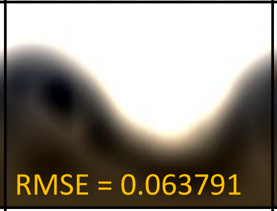
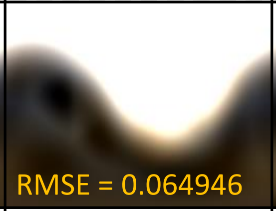
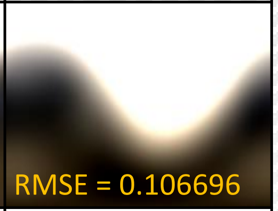
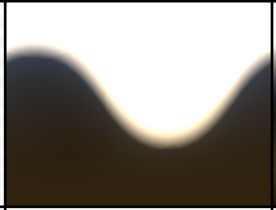
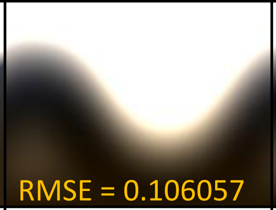
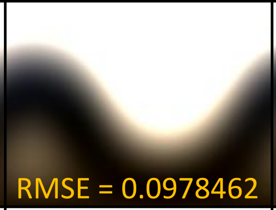
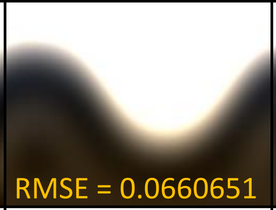
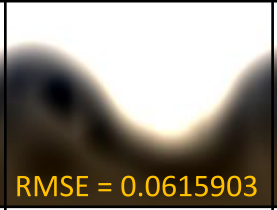
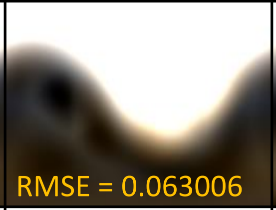
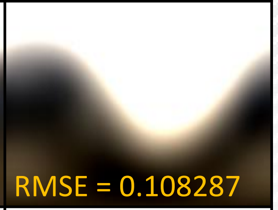
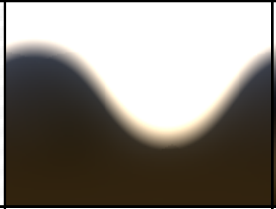
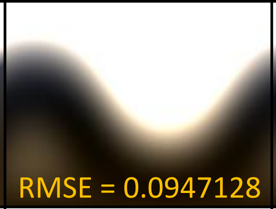
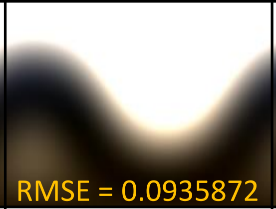
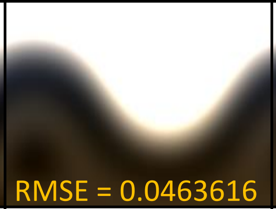
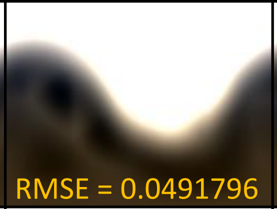
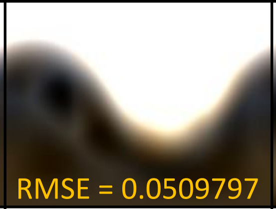
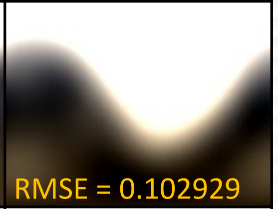
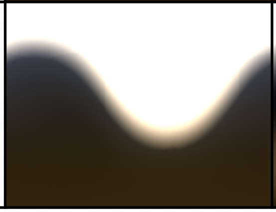
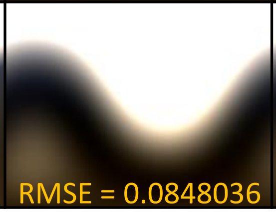
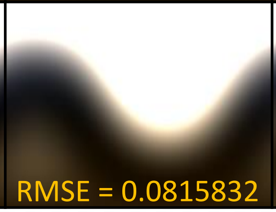
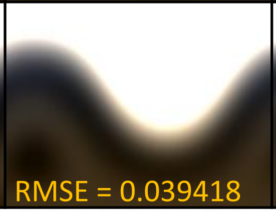
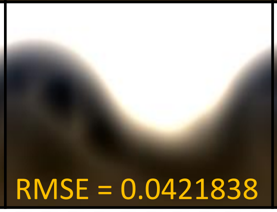
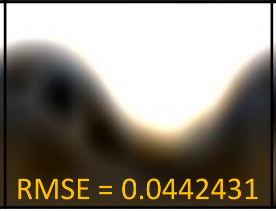
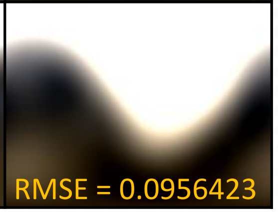
Approximation (Reference Format = Cube Map Array)

$specular$	$\sqrt{roughness}$	Reference	SH3	SH4	SH5	AD RGB	AD YCoCg	AD SRBF
0	0		RMSE = 0.105515	RMSE = 0.0990511	RMSE = 0.0498573	RMSE = 0.045356	RMSE = 0.0480222	RMSE = 0.115141
0	0.33		RMSE = 0.091852	RMSE = 0.0870423	RMSE = 0.0395638	RMSE = 0.0395877	RMSE = 0.042237	RMSE = 0.102124
0	0.66		RMSE = 0.0852468	RMSE = 0.0848289	RMSE = 0.0408226	RMSE = 0.0442375	RMSE = 0.0461098	RMSE = 0.0959853
0	1		RMSE = 0.0867629	RMSE = 0.0866793	RMSE = 0.0428686	RMSE = 0.0460251	RMSE = 0.0478834	RMSE = 0.0967098

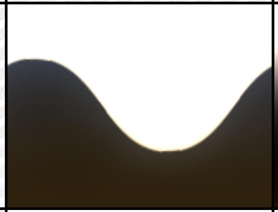
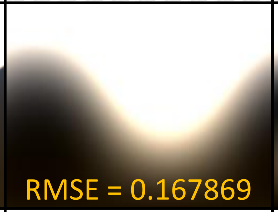
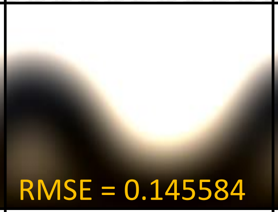
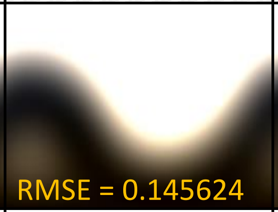
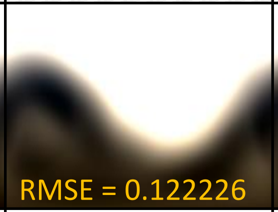
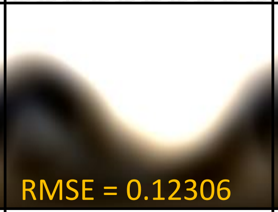
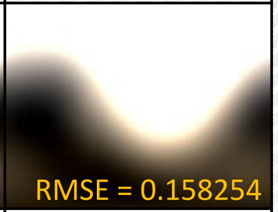
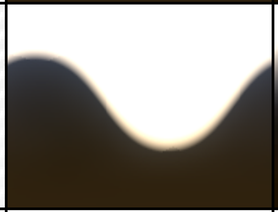
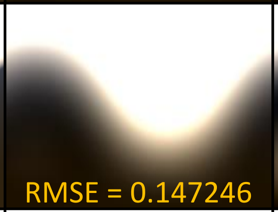
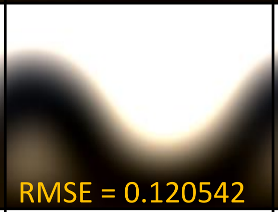
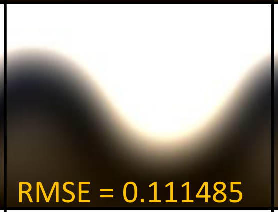
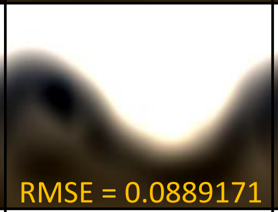
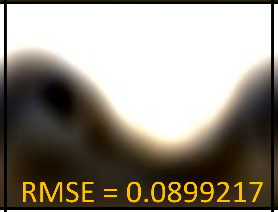
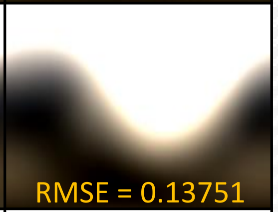
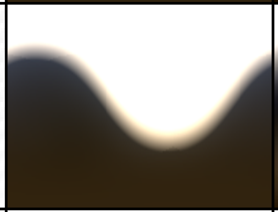
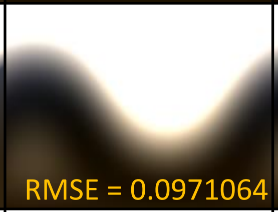
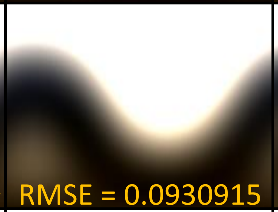
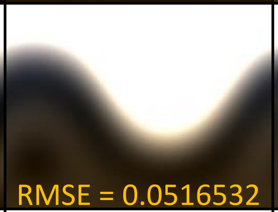
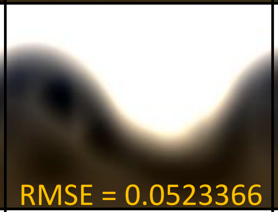
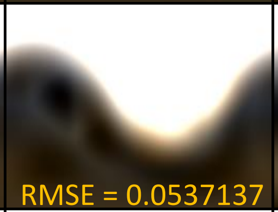
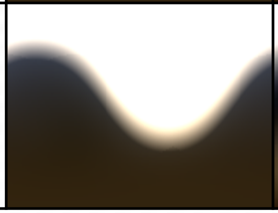
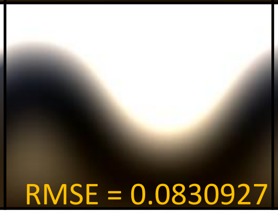
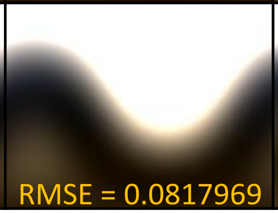
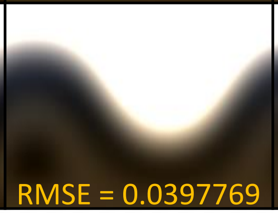
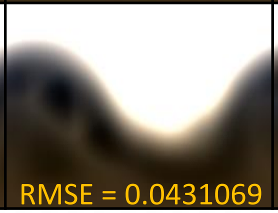
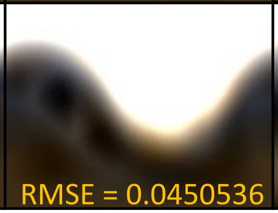
Approximation (Reference Format = Cube Map Array)

specular	$\sqrt{roughness}$	Reference	SH3	SH4	SH5	AD RGB	AD YCoCg	AD SRBF
0.33	0		 RMSE = 0.0954032	 RMSE = 0.0953287	 RMSE = 0.0479248	 RMSE = 0.0504581	 RMSE = 0.0524027	 RMSE = 0.105107
0.33	0.33		 RMSE = 0.0901046	 RMSE = 0.0899666	 RMSE = 0.0446922	 RMSE = 0.0476246	 RMSE = 0.0494089	 RMSE = 0.0997573
0.33	0.66		 RMSE = 0.0895111	 RMSE = 0.0893604	 RMSE = 0.0435516	 RMSE = 0.0466836	 RMSE = 0.0485574	 RMSE = 0.0992356
0.33	1		 RMSE = 0.084903	 RMSE = 0.0842644	 RMSE = 0.0410208	 RMSE = 0.044369	 RMSE = 0.0462362	 RMSE = 0.0957477

Approximation (Reference Format = Cube Map Array)

specular	$\sqrt{roughness}$	Reference	SH3	SH4	SH5	AD RGB	AD YCoCg	AD SRBF
0.66	0		 RMSE = 0.10529	 RMSE = 0.0968659	 RMSE = 0.0697763	 RMSE = 0.063791	 RMSE = 0.064946	 RMSE = 0.106696
0.66	0.33		 RMSE = 0.106057	 RMSE = 0.0978462	 RMSE = 0.0660651	 RMSE = 0.0615903	 RMSE = 0.063006	 RMSE = 0.108287
0.66	0.66		 RMSE = 0.0947128	 RMSE = 0.0935872	 RMSE = 0.0463616	 RMSE = 0.0491796	 RMSE = 0.0509797	 RMSE = 0.102929
0.66	1		 RMSE = 0.0848036	 RMSE = 0.0815832	 RMSE = 0.039418	 RMSE = 0.0421838	 RMSE = 0.0442431	 RMSE = 0.0956423

Approximation (Reference Format = Cube Map Array)

$specular$	$\sqrt{roughness}$	Reference	SH3	SH4	SH5	AD RGB	AD YCoCg	AD SRBF
1	0		 RMSE = 0.167869	 RMSE = 0.145584	 RMSE = 0.145624	 RMSE = 0.122226	 RMSE = 0.12306	 RMSE = 0.158254
1	0.33		 RMSE = 0.147246	 RMSE = 0.120542	 RMSE = 0.111485	 RMSE = 0.0889171	 RMSE = 0.0899217	 RMSE = 0.13751
1	0.66		 RMSE = 0.0971064	 RMSE = 0.0930915	 RMSE = 0.0516532	 RMSE = 0.0523366	 RMSE = 0.0537137	 RMSE = 0.102814
1	1		 RMSE = 0.0830927	 RMSE = 0.0817969	 RMSE = 0.0397769	 RMSE = 0.0431069	 RMSE = 0.0450536	 RMSE = 0.0940711

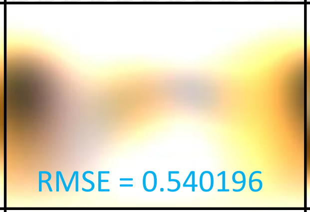
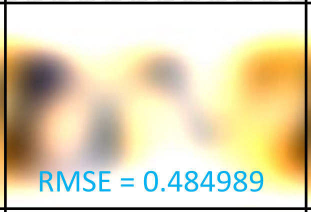
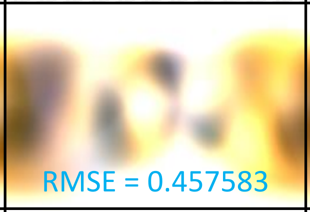
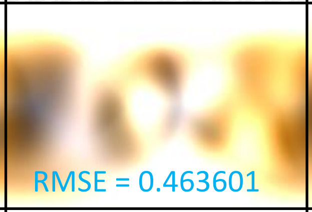
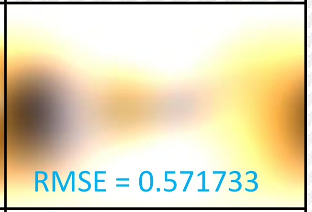
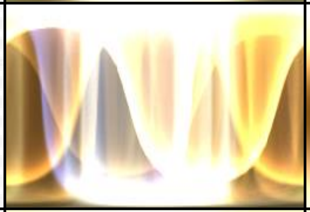
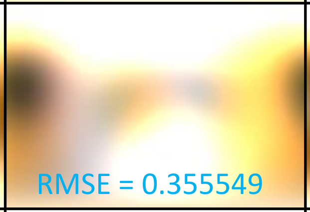
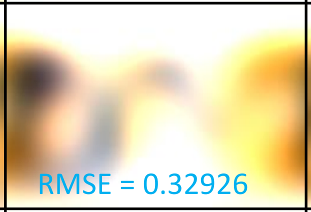
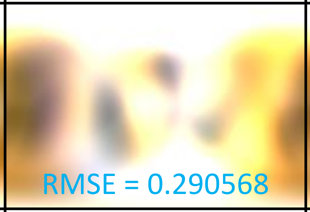
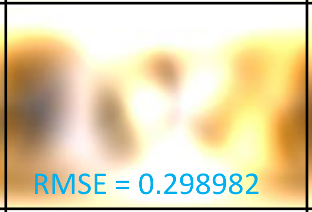
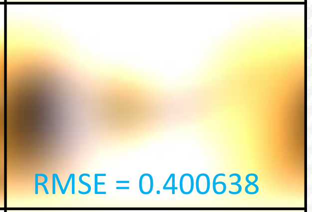
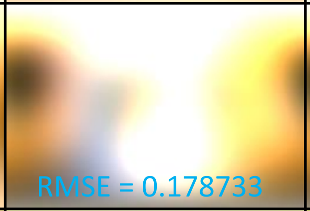
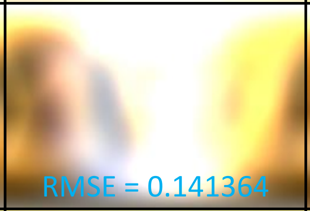
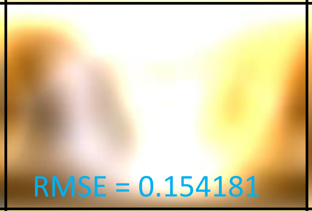
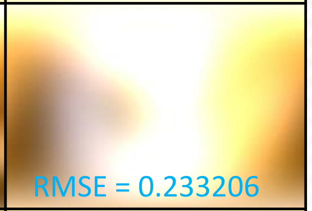
Environment map

Experiment 1

Experiment 2

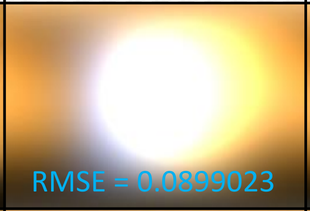
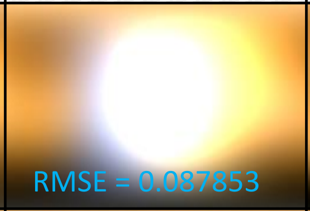
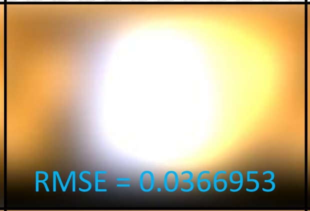
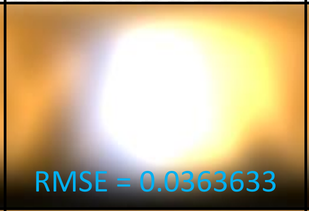
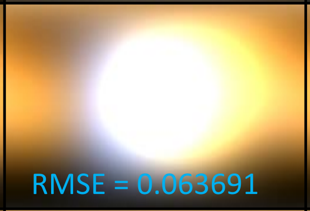
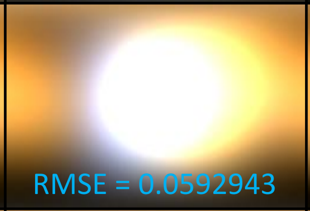
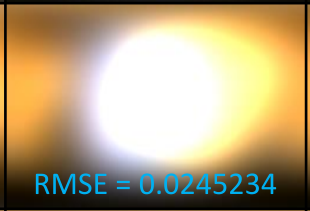
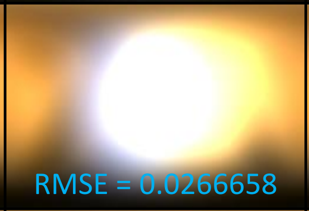
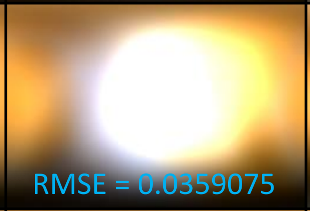
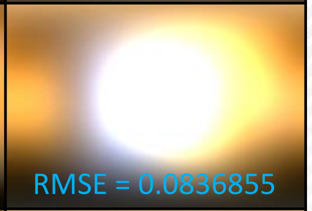
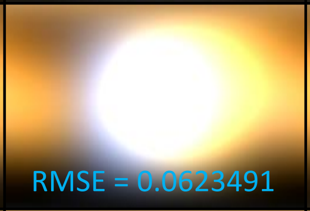
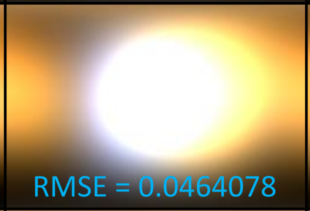
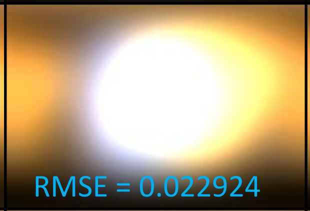
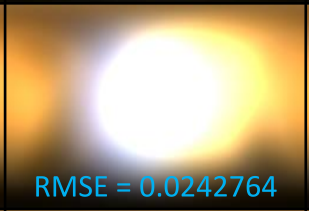
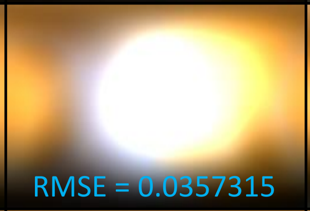
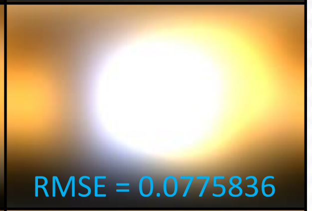
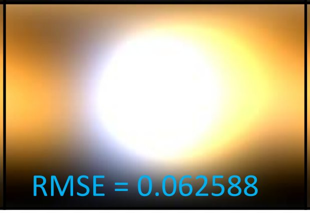
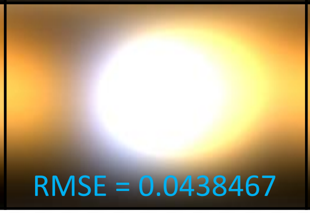
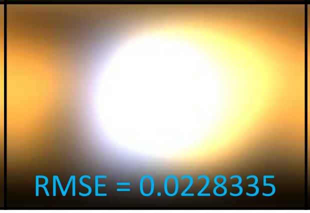
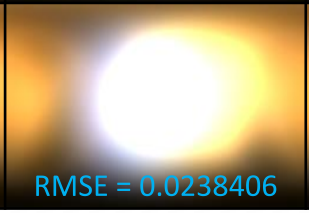
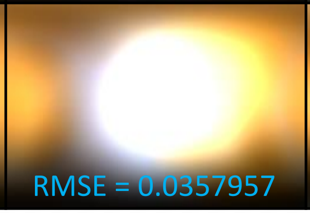
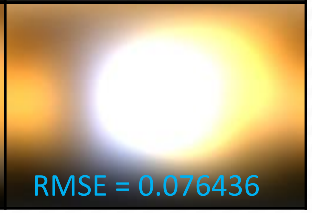
PRE-FILTERED CUBE MAP (MATTE)

Approximation (Reference Format = Cube Map Array)

$\sqrt{roughness}$	Reference	SH3	SH4	SH5	AD RGB	AD YCoCg	AD SRBF
0.002		 RMSE = 0.571352	 RMSE = 0.540196	 RMSE = 0.484989	 RMSE = 0.457583	 RMSE = 0.463601	 RMSE = 0.571733
0.14		 RMSE = 0.414258	 RMSE = 0.355549	 RMSE = 0.32926	 RMSE = 0.290568	 RMSE = 0.298982	 RMSE = 0.400638
0.28		 RMSE = 0.266122	 RMSE = 0.178733	 RMSE = 0.17727	 RMSE = 0.141364	 RMSE = 0.154181	 RMSE = 0.233206
0.42		 RMSE = 0.158992	 RMSE = 0.118812	 RMSE = 0.075932	 RMSE = 0.0645063	 RMSE = 0.075604	 RMSE = 0.147906

SQUARE ENIX ADVANCED TECHNOLOGY DIVISION

Approximation (Reference Format = Cube Map Array)

$\sqrt{roughness}$	Reference	SH3	SH4	SH5	AD RGB	AD YCoCg	AD SRBF
0.57		 RMSE = 0.0899023	 RMSE = 0.087853	 RMSE = 0.0366953	 RMSE = 0.0363633	 RMSE = 0.0439921	 RMSE = 0.105077
0.71		 RMSE = 0.063691	 RMSE = 0.0592943	 RMSE = 0.0245234	 RMSE = 0.0266658	 RMSE = 0.0359075	 RMSE = 0.0836855
0.85		 RMSE = 0.0623491	 RMSE = 0.0464078	 RMSE = 0.022924	 RMSE = 0.0242764	 RMSE = 0.0357315	 RMSE = 0.0775836
1		 RMSE = 0.062588	 RMSE = 0.0438467	 RMSE = 0.0228335	 RMSE = 0.0238406	 RMSE = 0.0357957	 RMSE = 0.076436

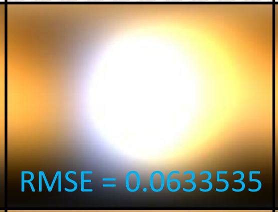
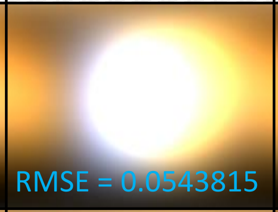
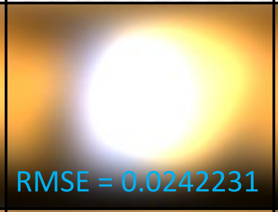
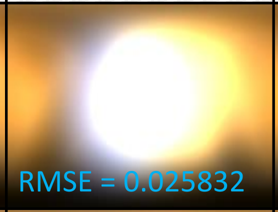
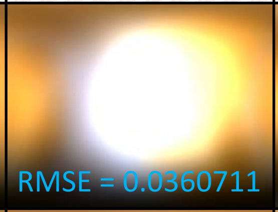
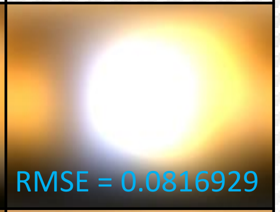
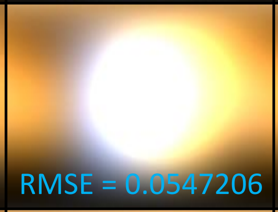
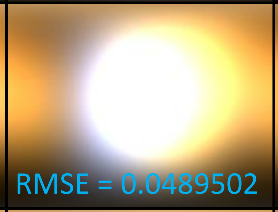
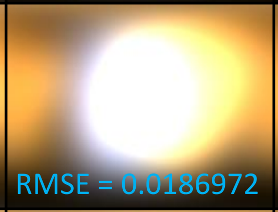
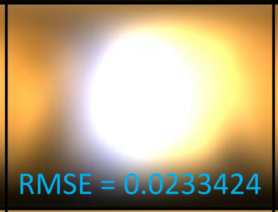
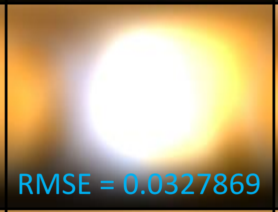
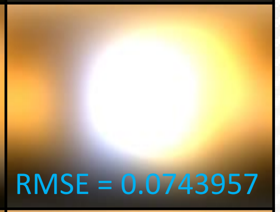
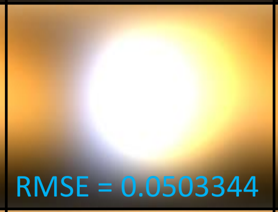
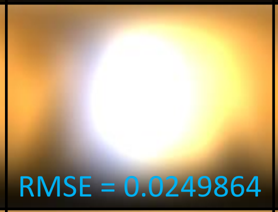
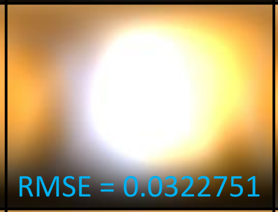
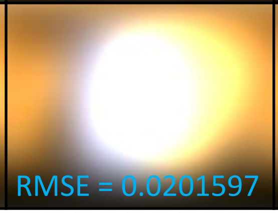
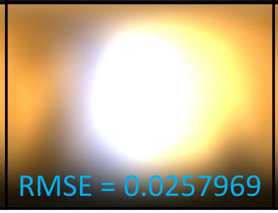
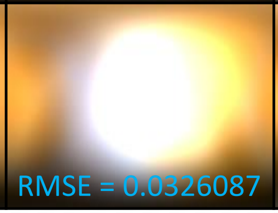
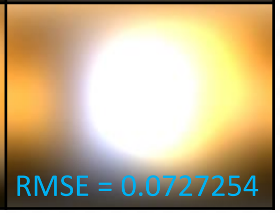
Environment map

Experiment 1

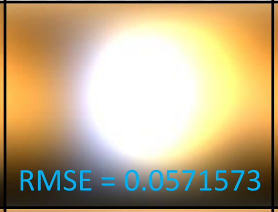
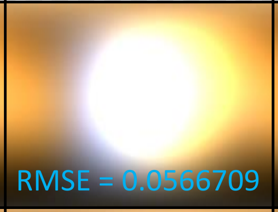
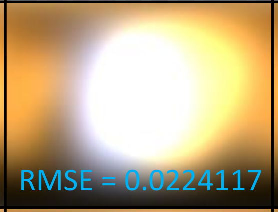
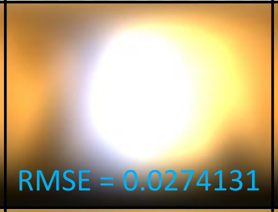
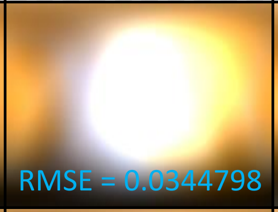
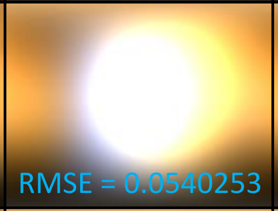
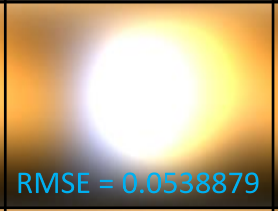
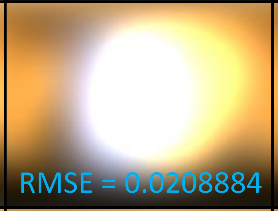
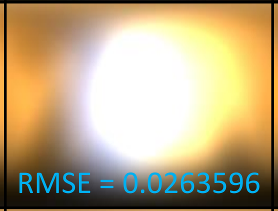
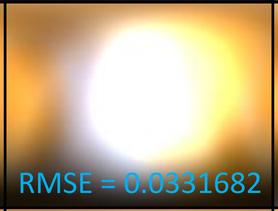
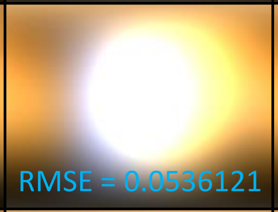
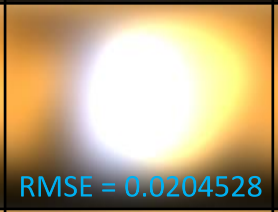
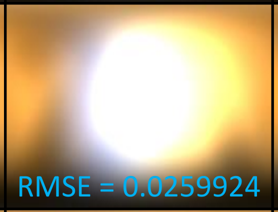
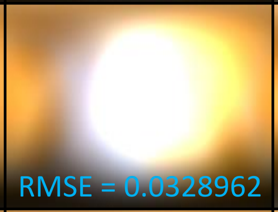
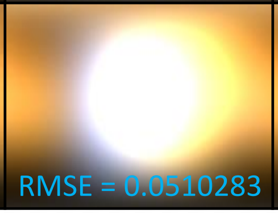
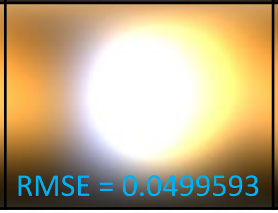
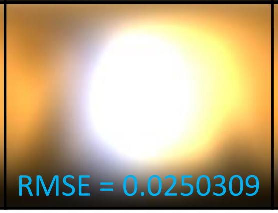
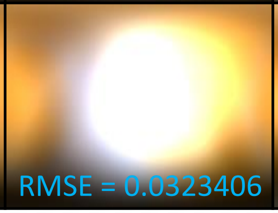
Experiment 2

PRE-FILTERED CUBE MAP (DIFFUSE)

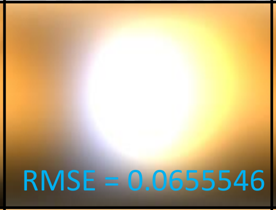
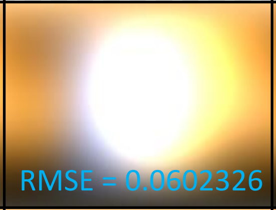
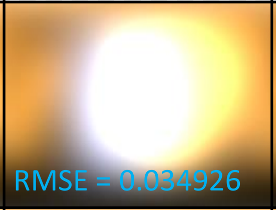
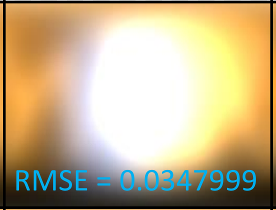
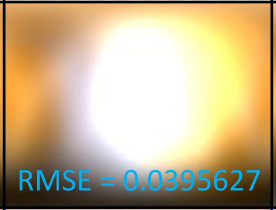
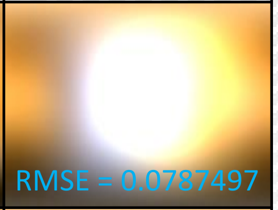
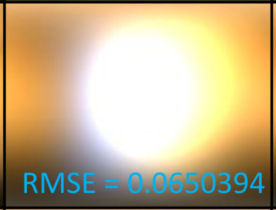
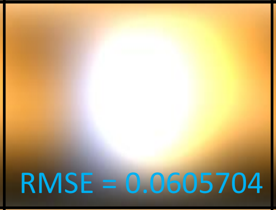
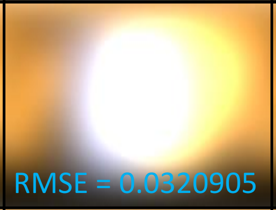
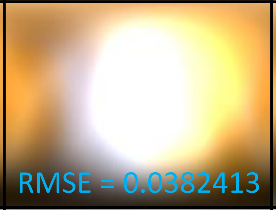
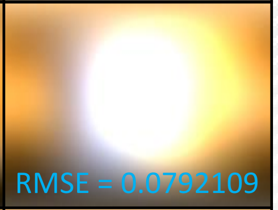
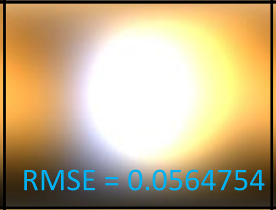
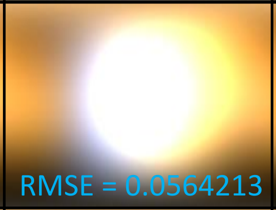
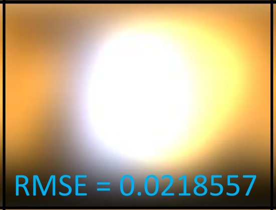
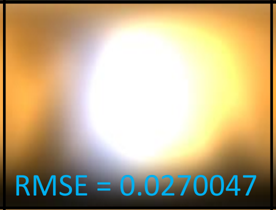
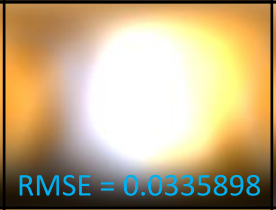
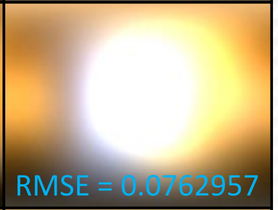
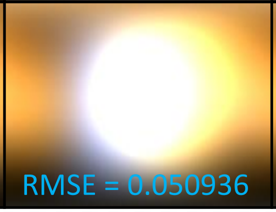
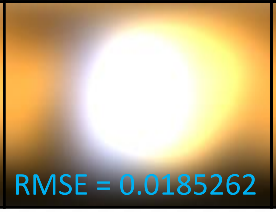
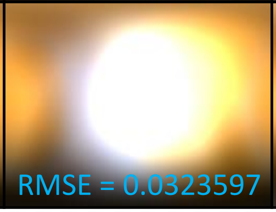
Approximation (Reference Format = Cube Map Array)

$specular$	$\sqrt{roughness}$	Input	SH3	SH4	SH5	AD RGB	AD YCoCg	AD SRBF
0	0		 RMSE = 0.0633535	 RMSE = 0.0543815	 RMSE = 0.0242231	 RMSE = 0.025832	 RMSE = 0.0360711	 RMSE = 0.0816929
0	0.33		 RMSE = 0.0547206	 RMSE = 0.0489502	 RMSE = 0.0186972	 RMSE = 0.0233424	 RMSE = 0.0327869	 RMSE = 0.0743957
0	0.66		 RMSE = 0.0511797	 RMSE = 0.0503344	 RMSE = 0.0191675	 RMSE = 0.0249864	 RMSE = 0.0322751	 RMSE = 0.0720122
0	1		 RMSE = 0.0521126	 RMSE = 0.0518774	 RMSE = 0.0201597	 RMSE = 0.0257969	 RMSE = 0.0326087	 RMSE = 0.0727254

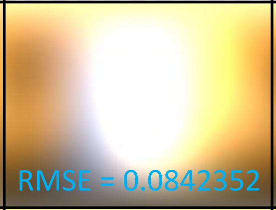
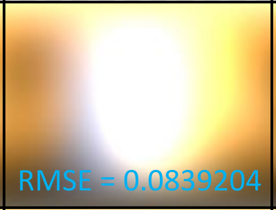
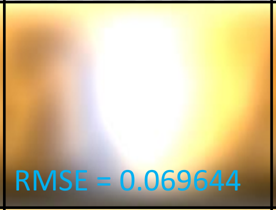
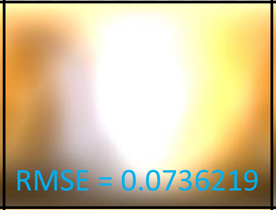
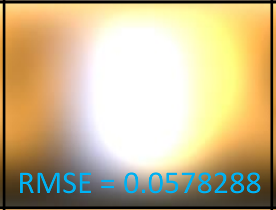
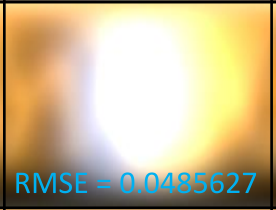
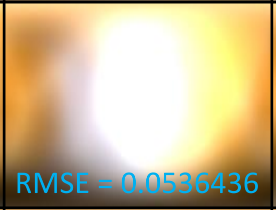
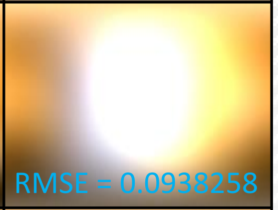
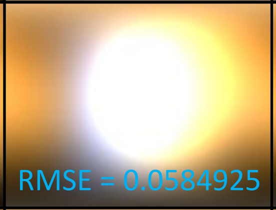
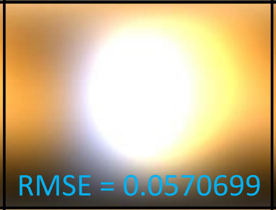
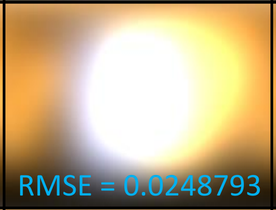
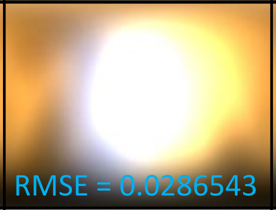
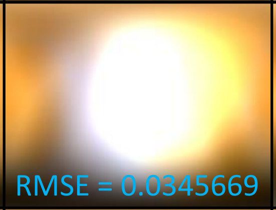
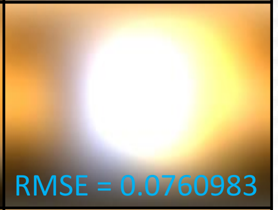
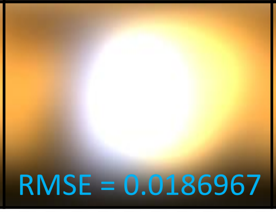
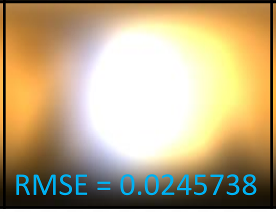
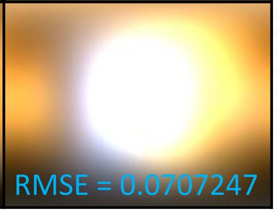
Approximation (Reference Format = Cube Map Array)

specular	$\sqrt{roughness}$	Input	SH3	SH4	SH5	AD RGB	AD YCoCg	AD SRBF
0.33	0		 RMSE = 0.0571573	 RMSE = 0.0566709	 RMSE = 0.0224117	 RMSE = 0.0274131	 RMSE = 0.0344798	 RMSE = 0.0779214
0.33	0.33		 RMSE = 0.0540253	 RMSE = 0.0538879	 RMSE = 0.0208884	 RMSE = 0.0263596	 RMSE = 0.0331682	 RMSE = 0.0745169
0.33	0.66		 RMSE = 0.0536121	 RMSE = 0.0534815	 RMSE = 0.0204528	 RMSE = 0.0259924	 RMSE = 0.0328962	 RMSE = 0.0741174
0.33	1		 RMSE = 0.0510283	 RMSE = 0.0499593	 RMSE = 0.0192056	 RMSE = 0.0250309	 RMSE = 0.0323406	 RMSE = 0.0718611

Approximation (Reference Format = Cube Map Array)

$specular$	$\sqrt{roughness}$	Input	SH3	SH4	SH5	AD RGB	AD YCoCg	AD SRBF
0.66	0		 RMSE = 0.0655546	 RMSE = 0.0602326	 RMSE = 0.034926	 RMSE = 0.0347999	 RMSE = 0.0395627	 RMSE = 0.0787497
0.66	0.33		 RMSE = 0.0650394	 RMSE = 0.0605704	 RMSE = 0.0320905	 RMSE = 0.0331192	 RMSE = 0.0382413	 RMSE = 0.0792109
0.66	0.66		 RMSE = 0.0564754	 RMSE = 0.0564213	 RMSE = 0.0218557	 RMSE = 0.0270047	 RMSE = 0.0335898	 RMSE = 0.0762957
0.66	1		 RMSE = 0.050936	 RMSE = 0.0476247	 RMSE = 0.0185262	 RMSE = 0.0241725	 RMSE = 0.0323597	 RMSE = 0.0714479

Approximation (Reference Format = Cube Map Array)

$specular$	$\sqrt{roughness}$	Input	SH3	SH4	SH5	AD RGB	AD YCoCg	AD SRBF
1	0		 RMSE = 0.11225	 RMSE = 0.0842352	 RMSE = 0.0839204	 RMSE = 0.069644	 RMSE = 0.0736219	 RMSE = 0.10493
1	0.33		 RMSE = 0.0938861	 RMSE = 0.0719985	 RMSE = 0.0578288	 RMSE = 0.0485627	 RMSE = 0.0536436	 RMSE = 0.0938258
1	0.66		 RMSE = 0.0584925	 RMSE = 0.0570699	 RMSE = 0.0248793	 RMSE = 0.0286543	 RMSE = 0.0345669	 RMSE = 0.0760983
1	1		 RMSE = 0.0499161	 RMSE = 0.0483398	 RMSE = 0.0186967	 RMSE = 0.0245738	 RMSE = 0.0320775	 RMSE = 0.0707247

Conclusion

- Ambient Dice – RGB, YCoCg.
  - ☺ The quality is the same as SH4 or SH5 according to RMSE (Root Mean Square Deviation).
  - ☺ High frequency cube map approximation is better than SH4 and SH5.
  - ☹ Artifacts appear around lower values areas.

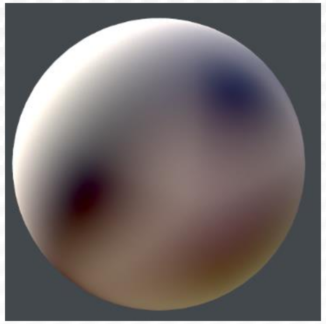
Reference

SH5

AD RGB

Pre-filtered Cube Map Approximation

Pre-filtered cube map (AD RGB)



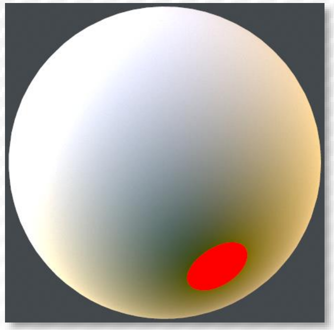
Rendered with sphere

Specific Artifacts

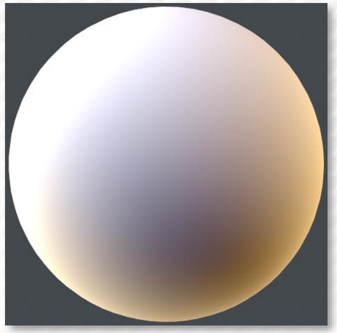
Conclusion

- Ambient Dice – SRBF.
  - ☺ The quality is the same as SH3 according to RMSE.
  - ☹ Depending on the input cube map, negative values might appear.
- We chose to use AD SRBF because of overall quality and stability.

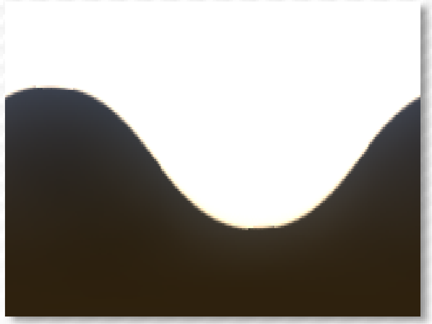
Pre-filtered cube map (reference)



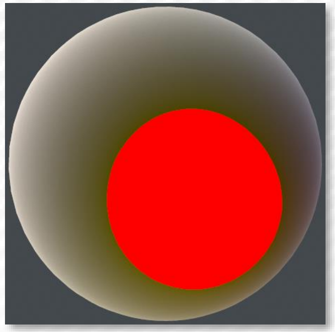
SH3



AD SRBF



Pre-filtered cube map (reference)



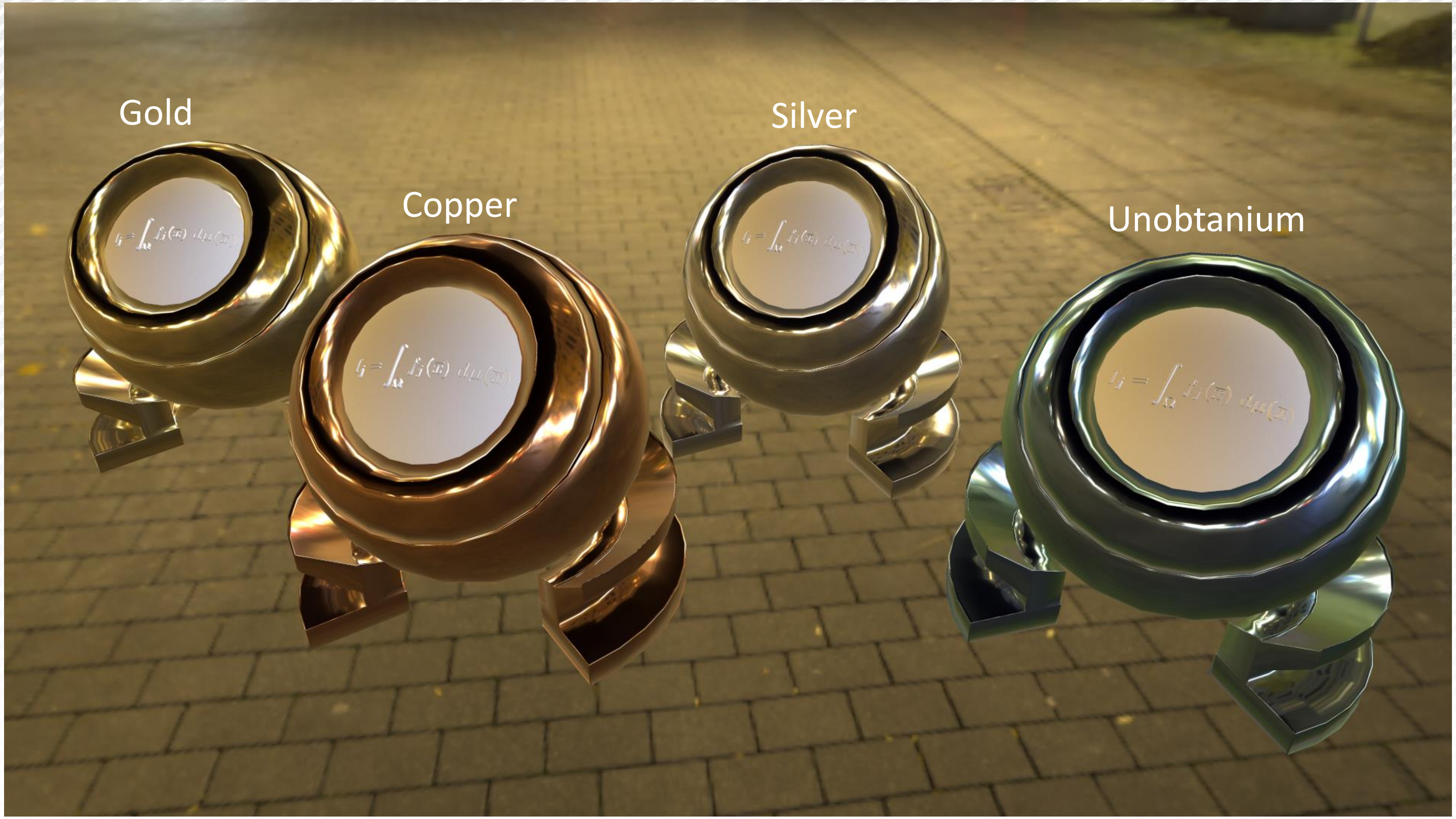
SH3

AD SRBF

Visualization of Negative Value (Red term is negative)

RESULTS

Conductor (Fresnel f0 and edgetint from [Gulbrandsen 2014])



Conductor – roughness

Roughness = 0 Roughness = 1

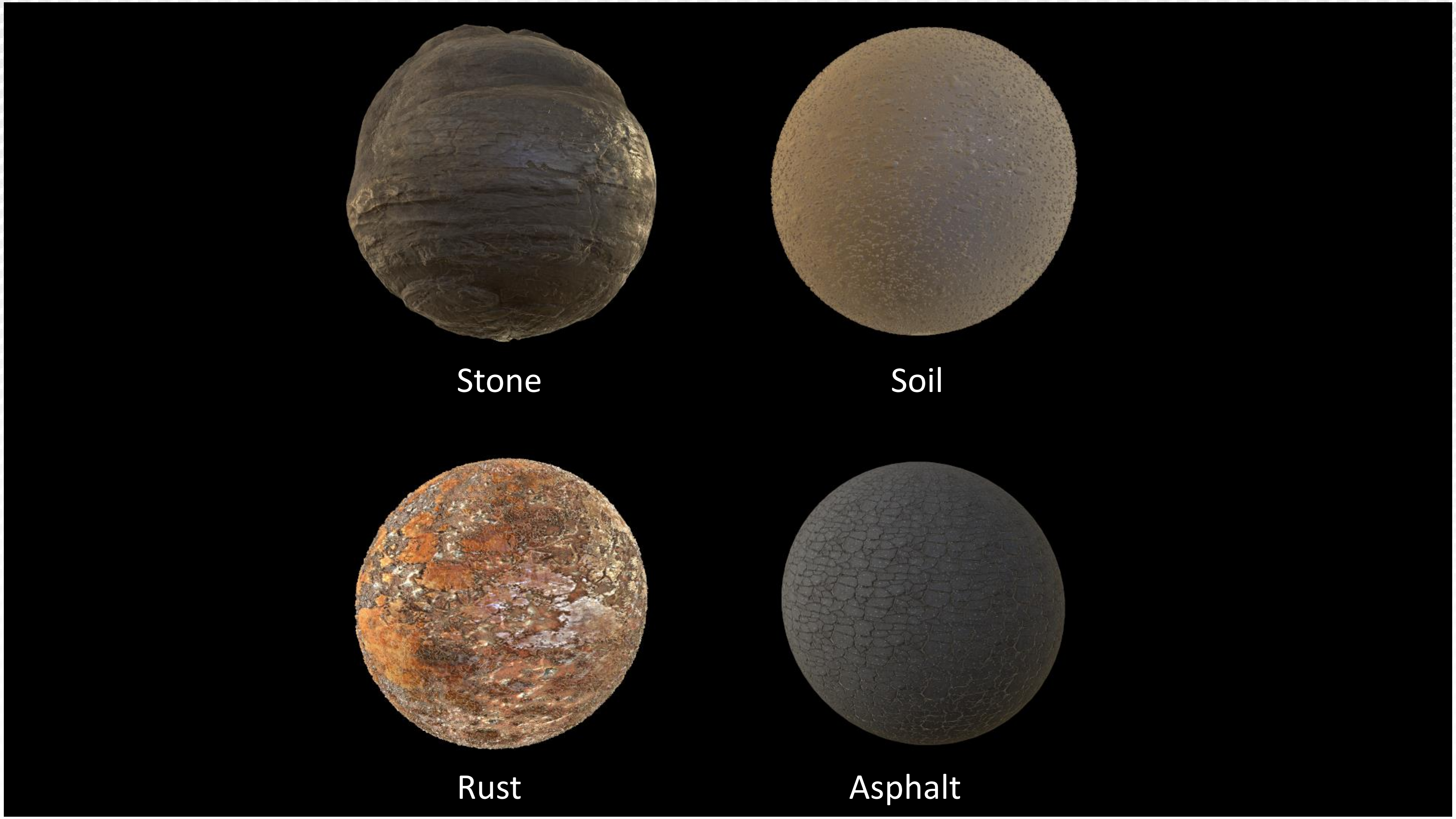
Gold

Copper

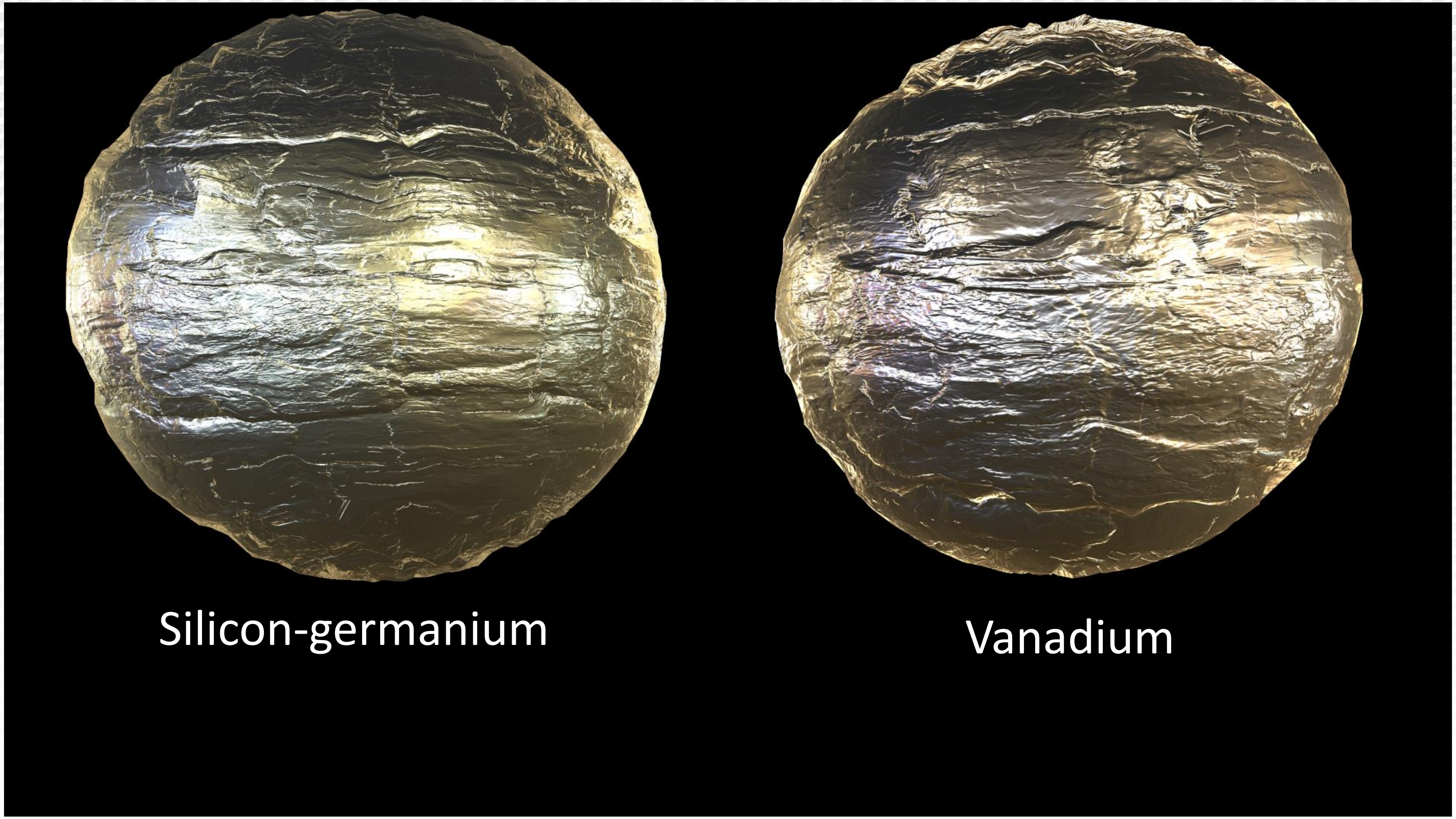
Silver

Unobtanium

Dielectric



Semiconductor (Fresnel f0 and edgetint from RefractiveIndex.Info)



Future Work

- Add support for anisotropy.
- Resolve the negative value artifacts of AD SRBF.
- Measure the performance on consoles.

Acknowledgements

- Adelle Bueno
- Eduardo Mosena
- Yusuke Tokuyoshi

References

1. [Gulbrandsen 2014] Ole Gulbrandsen. Artist Friendly Metallic Fresnel.
2. [Iwanicki 2017] Michal Iwanicki and Peter-Pike Sloan. Ambient Dice.
3. [Jakob 2014] Wenzel Jakob, Eugene d'Eon, Otto Jakob and Steve Marschner. A Comprehensive Framework for Rendering Layered Materials.
4. [Karis 2013] Brian Karis. Real Shading in Unreal Engine 4.
5. [Kelemen 2001] Csaba Kelemen and Laszlo Szirmay-Kalos. A Microfacet Based Coupled Specular-Matte BRDF Model with Importance Sampling.
6. [Krivanek 2008] Jaroslav Krivanek and Mark Colbert. Real-time Shading with Filtered Importance Sampling.
7. [Kulla 2017] Christopher Kulla and Alejandro Conty. Revisiting Physically Based Shading at Imageworks.
8. [Lagarde 2014] Sebastien Lagarde and Charles de Rousiers. Moving Frostbite to Physically Based Rendering 3.0.
9. [Lagarde 2018] Sebastien Lagarde and Evgenii Golubev. The Road Toward Unified Rendering with Unity's High Definition Render Pipeline.
10. [Schlick 1994] Schlick, C., An Inexpensive BRDF Model for Physically-based Rendering.

Resources

Image:

p.7: Revisiting Physically Based Shading at Imageworks, 2017, pp12, https://blog.selfshadow.com/publications/s2017-shading-course/imageworks/s2017_pbs_imageworks_slides_v2.pdf.

p.8: Revisiting Physically Based Shading at Imageworks, 2017, pp13, https://blog.selfshadow.com/publications/s2017-shading-course/imageworks/s2017_pbs_imageworks_slides_v2.pdf.

p.10: Mori Knob, Yasutoshi Mori, https://casual-effects.com/data/, CC by 3.0.

p.36: HDRIHAVEN, delta_2_2k.hdr, https://hdrihaven.com/hdri/?c=outdoor&h=delta_2, CC by 0.0.

p.44: HDRIHAVEN, hansaplatz_2k.hdr, https://hdrihaven.com/hdri/?c=outdoor&h=hansaplatz, CC by 0.0.

Trademark:

-SONY PICTURES is a trademark or registered trademark of Sony Corporation.