SIGGRAPH2012

The 39th International Conference and Exhibition
on Computer Graphics and Interactive Techniques

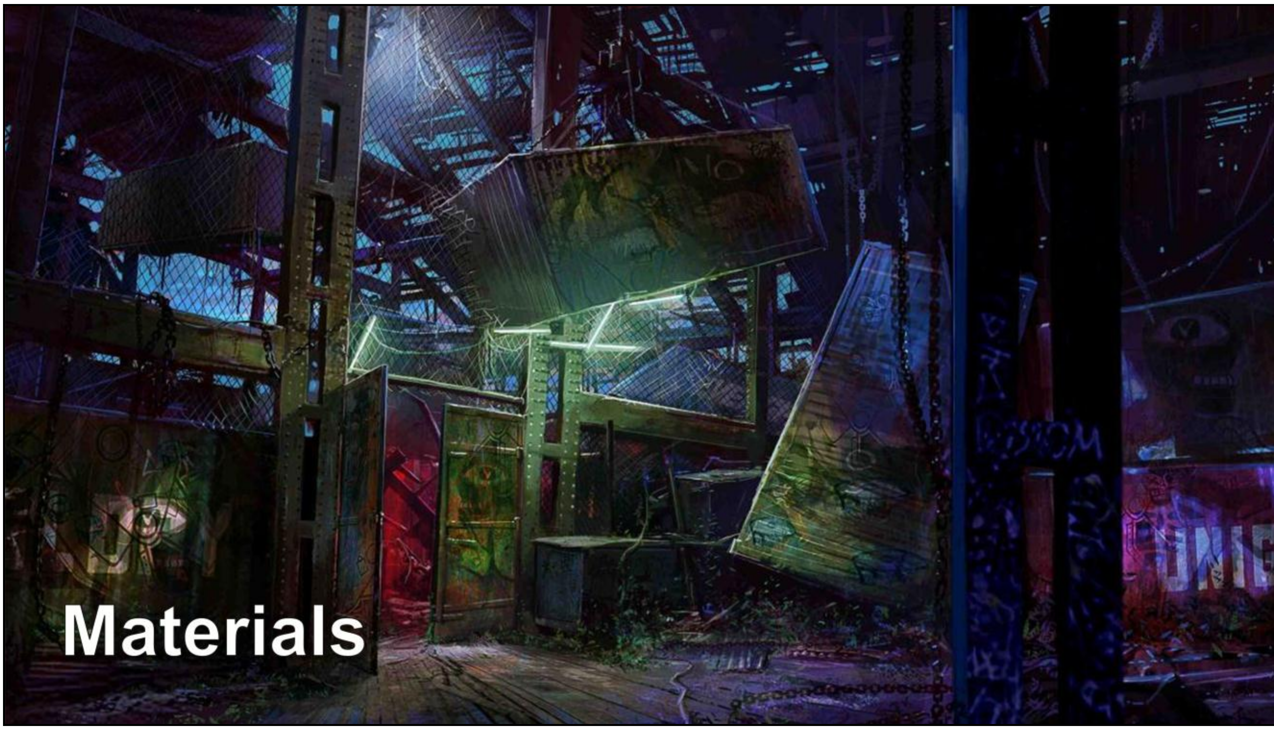
Calibrating Lighting and Materials in Far Cry 3

Stephen McAuley, Ubisoft Montréal

stephen.mcauley@ubisoft.com

SIGGRAPH2012

What motivated us to calibrate our lighting and materials on Far Cry 3? Far Cry 3 is a huge game, which puts a lot of pressure on content creation, a pressure which is only going to increase as hardware improves, consumers' continue to demand better quality and we wish to create bigger and bigger games. With a time of day cycle and indoor and outdoor environments, we need to make sure that our art looks good and holds up in all situations, and also to make sure it's easy to achieve that goal. We need to create better tools, improve our material models, expose better parameters, link things that should be linked…



On Far Cry 3, we began to make these changes to help our artists create high quality visuals, looking at our shading, our lighting but also our materials.

Diffuse Albedo

SIGGRAPH2012

Getting our diffuse albedo right was very important to us. If you just look at the concept art, you can see bright and saturated colours, which are really, really easy to get wrong. (Hence how many games look rather desaturated, and grey and brown.)

Moreover, balancing diffuse albedo textures causes a lot of problems during development. Maybe at one time or another, you've heard the phrase "My characters don't sit in the scene." Your character modellers and your environment artists are two separate teams, creating materials to different standards. Or an artist creates a material that looks fine outside, but for some reason looks pitch black as soon as it's moved to an indoor environment. Of course, he first goes to the lighters to complain, and there's a horrible cycle of iteration between lighters and material artists trying to work out who's to blame (and let's not get started on the postprocessing effects).

BRDF

SIGGRAPH2012

- Diffuse albedo:

$$f(\mathbf{l},\mathbf{v}) = \frac{\mathbf{c}_{\mathrm{diff}}}{\pi} + \frac{F(\mathbf{l},\mathbf{h})G(\mathbf{l},\mathbf{v},\mathbf{h})D(\mathbf{h})}{4(\underline{\mathbf{n}\cdot\mathbf{l}})(\underline{\mathbf{n}\cdot\mathbf{v}})}$$

So what do we mean by diffuse albedo? Well, it's part of our BRDF equation…

BRDF

- Torrance-Sparrow microfacet BRDF:

$$f(\mathbf{l},\mathbf{v}) = \frac{\mathbf{c}_{\mathrm{diff}}}{\pi} + \boxed{\frac{F(\mathbf{l},\mathbf{h})G(\mathbf{l},\mathbf{v},\mathbf{h})D(\mathbf{h})}{4(\underline{\mathbf{n}\cdot\mathbf{l}})(\underline{\mathbf{n}\cdot\mathbf{v}})}}$$

This BRDF is a combination of the Torrance-Sparrow microfacet BRDF for specular reflectance…

BRDF

SIGGRAPH2012

- Lambert diffuse BRDF:

$$f(\mathbf{l},\mathbf{v}) = \boxed{\frac{\mathbf{c}_{\mathrm{diff}}}{\pi}} + \frac{F(\mathbf{l},\mathbf{h})G(\mathbf{l},\mathbf{v},\mathbf{h})D(\mathbf{h})}{4(\underline{\mathbf{n}\cdot\mathbf{l}})(\underline{\mathbf{n}\cdot\mathbf{v}})}$$

…and Lambert diffuse BRDF. (Remember the N.L term is part of the reflectance integral not the BRDF.) This is where our diffuse albedo comes in, and we can see that it’s a physical property and can be measured. If we get it wrong, this whole equation will be imbalanced, so it doesn’t make much sense to have a physically-based specular reflectance model without sorting out your diffuse albedo at the same time.

Motivation

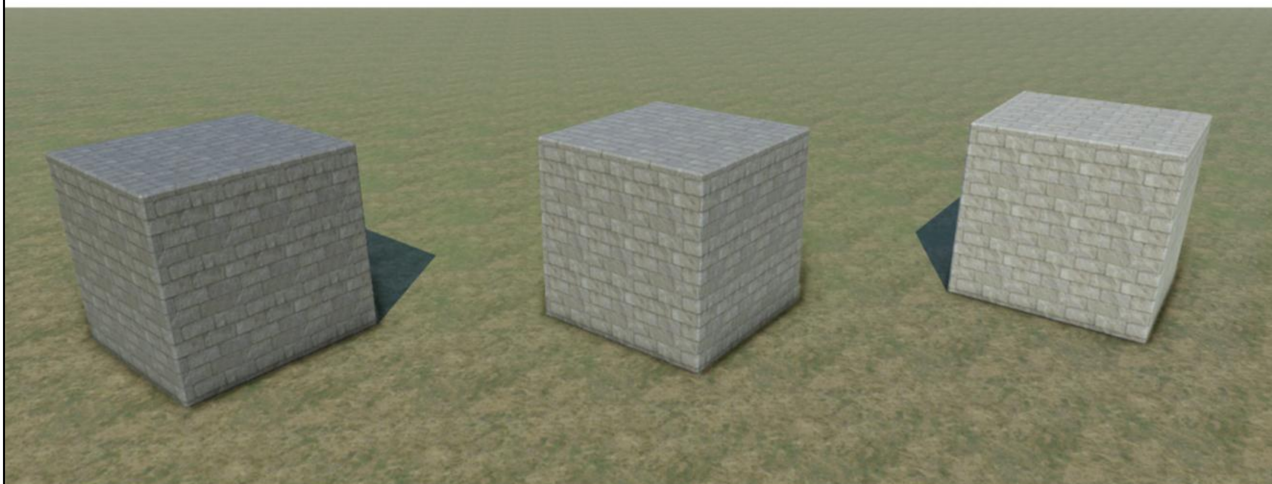
SIGGRAPH2012

- Natural, saturated colours
- Consistent across materials
- Consistent under all lighting conditions
- Diffuse lighting balanced with specular

Let’s sum up our motivation.

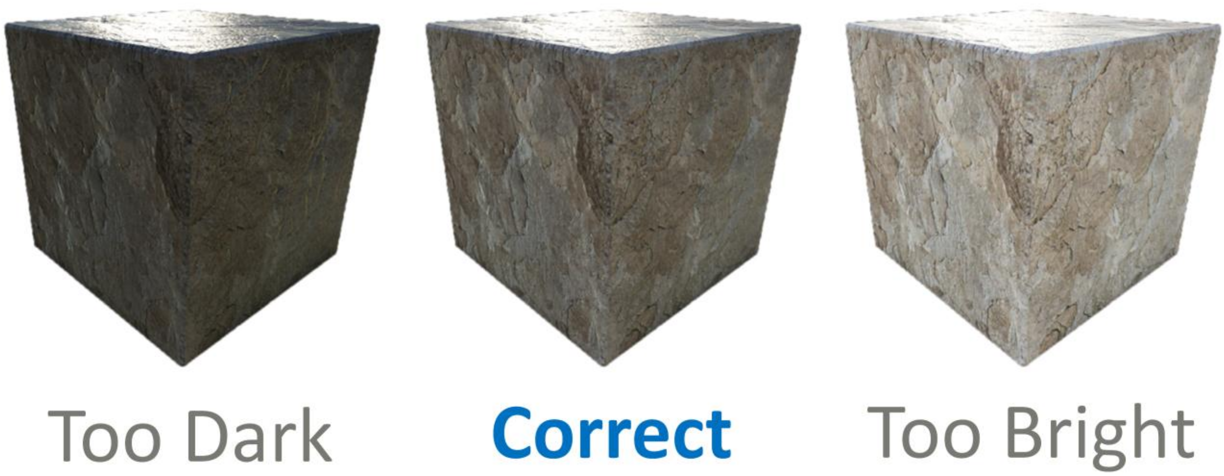
Diffuse Albedo vs Diffuse Albedo

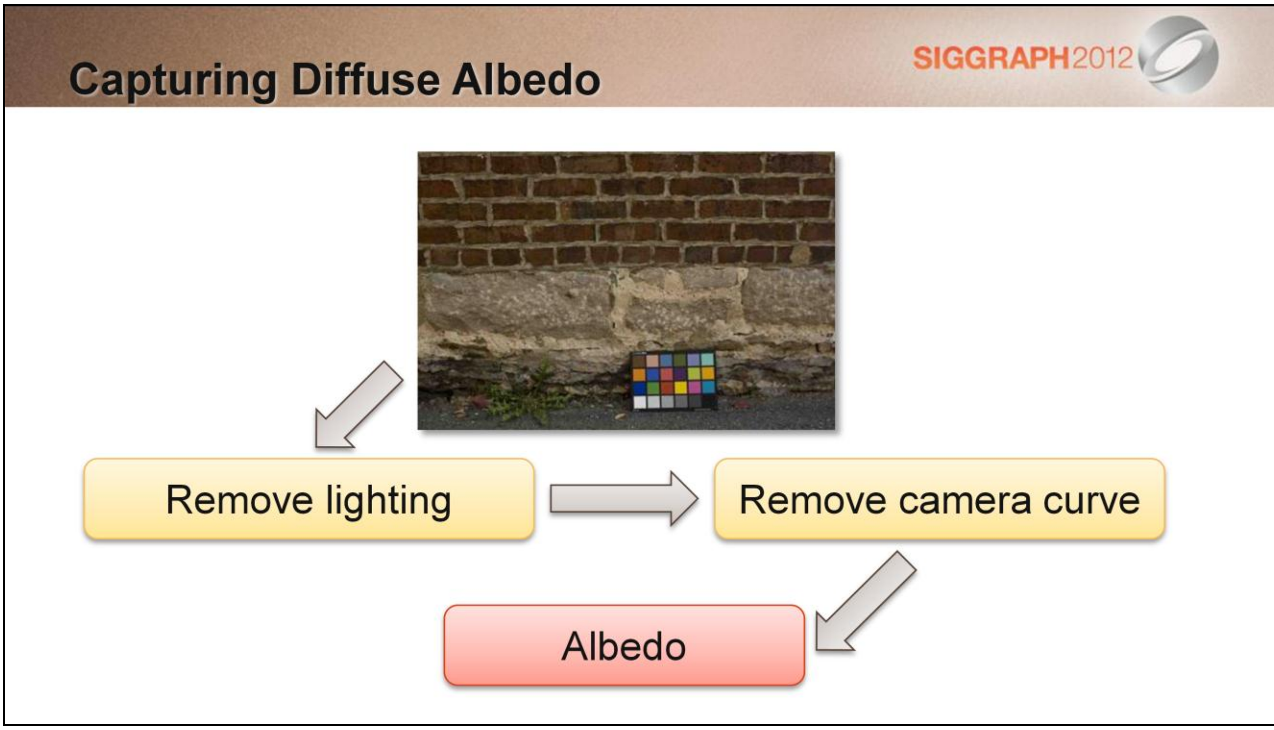
SIGGRAPH2012



Diffuse Reflectance vs Specular Reflectance

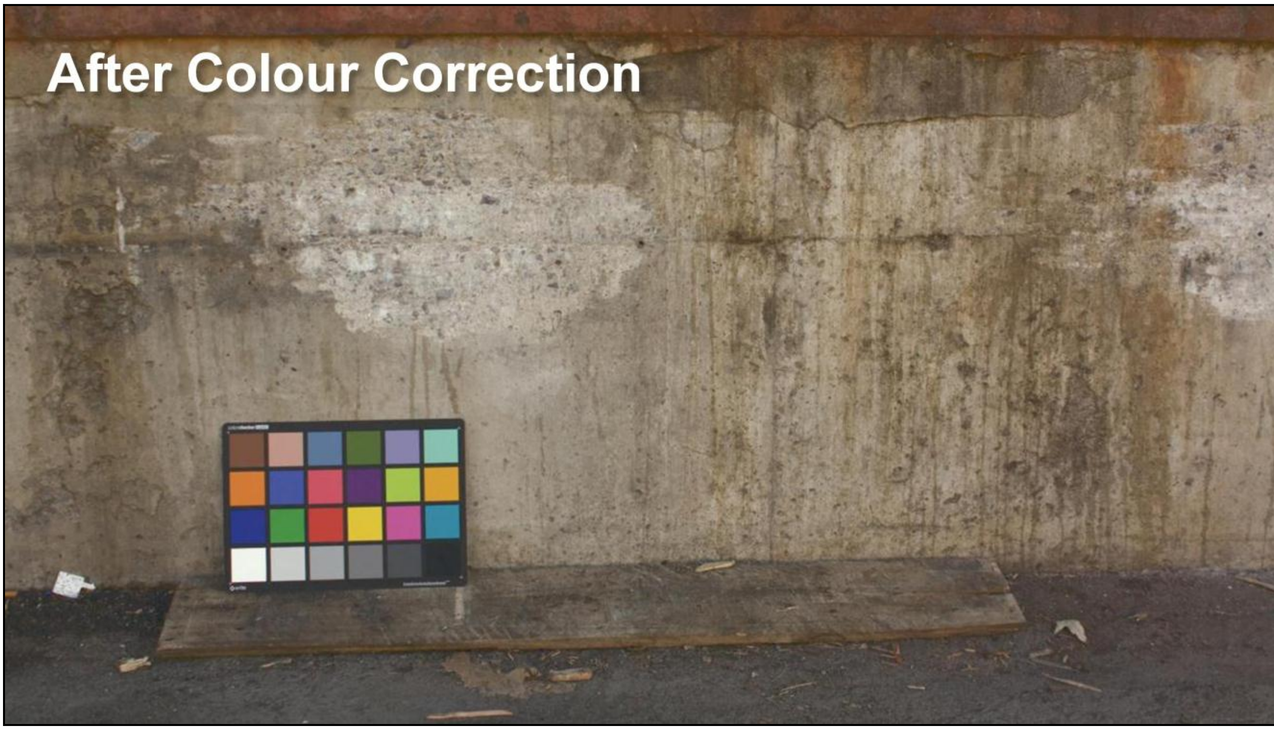
SIGGRAPH2012





If we assume only diffuse lighting here.

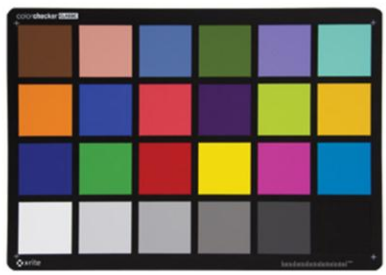

Original


After Colour Correction

Capturing Diffuse Albedo

SIGGRAPH2012

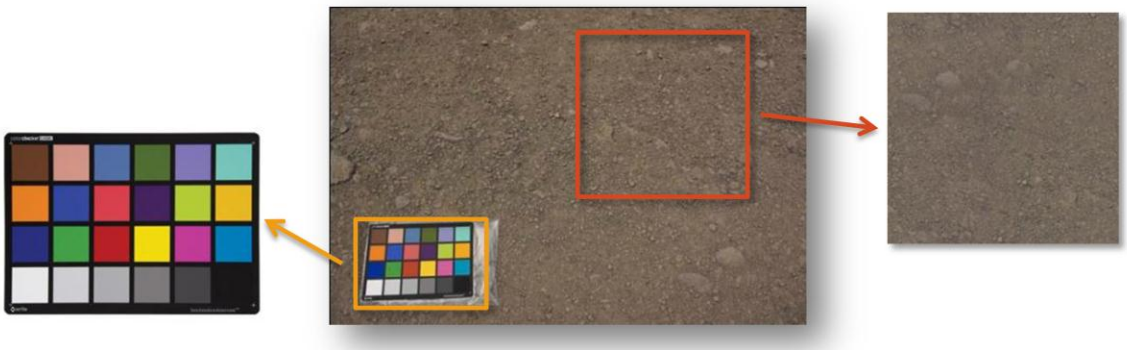
- Use the Macbeth ColorChecker™ as reference.
- Made by X-Rite, http://www.xrite.com/
- 24 colour patches with known sRGB values.



Capturing Diffuse Albedo

SIGGRAPH2012

- Photograph material alongside ColorChecker™.
- Lighting must be consistent.
- Use ColorChecker™ patches to find transform.



It’s obviously really important to have consistent lighting across the material that you’re capturing and the ColorChecker™ itself. You also want to minimise specular reflection. This means they both should be parallel to the camera plane, with direct lighting minimised as much as possible. Overcast days are ideal.

Affine Transform [Malin11]

- **Pro:** removes cross-talk between channels
- **Con:** linear transform

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \end{bmatrix} \begin{bmatrix} x_r \\ x_g \\ x_b \\ 1 \end{bmatrix} = \begin{bmatrix} y_r & y_g & y_b \end{bmatrix}$$

Many thanks to Paul Malin of Activision Central Tech, for sharing with me his colour correction algorithm and allowing me to present it.

Polynomial Transform [Malin11]

SIGGRAPH2012

- **Pro:** accurately adjusts levels
- **Con:** channel independent

$$\forall i \in \{r, g, b\}, y_i = \sum_{0}^{m} a_m x_i^m$$

Linear Least Squares

SIGGRAPH2012

- Given **X** and **Y**, find **A** such that:

$$\mathbf{X}\mathbf{A} = \mathbf{Y}$$

- Linear least squares: best approximation is:

$$\hat{\mathbf{A}} = \left(\mathbf{X}^{\mathsf{T}}\mathbf{X}\right)^{-1}\mathbf{X}^{\mathsf{T}}\mathbf{Y}$$

- Solve using Gauss-Jordan elimination.

Thus to find the best approximate affine and polynomial transforms, we need to find appropriate X and Y, and then find the matrix inverse. Thankfully, we can do this relatively simply by Gauss-Jordan elimination.

Affine Transform

SIGGRAPH2012

- Let $\mathbf{x_j}$ be our photographed ColorChecker™ patches.
- Let $\mathbf{y_j}$ be the target ColorChecker™ values.

$$\begin{bmatrix} x_{0,r} & x_{0,g} & x_{0,b} & 1 \\ x_{1,r} & x_{1,g} & x_{1,b} & 1 \\ \vdots & & & \vdots \\ x_{n,r} & x_{n,g} & x_{n,b} & 1 \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \\ a_{41} & a_{42} & a_{43} \end{bmatrix} = \begin{bmatrix} y_{0,r} & y_{0,g} & y_{0,b} \\ y_{1,r} & y_{1,g} & y_{1,b} \\ \vdots & & \vdots \\ y_{n,r} & y_{n,g} & y_{n,b} \end{bmatrix}$$

Polynomial Transform

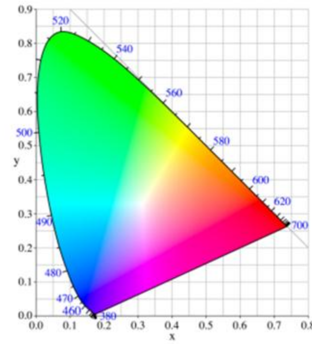
- Let $\mathbf{x}_j$ be our photographed ColorChecker™ patches.
- Let $\mathbf{y}_j$ be the target ColorChecker™ values.

$$\forall i \in \{r, g, b\}, \begin{bmatrix} 1 & x_{0,i} & x_{0,i}^2 & \dots & x_{0,i}^m \\ 1 & x_{1,i} & x_{1,i}^2 & \dots & x_{1,i}^m \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n,i} & x_{n,i}^2 & \dots & x_{n,i}^m \end{bmatrix} \begin{bmatrix} a_{0,i} \\ a_{1,i} \\ a_{2,i} \\ \vdots \\ a_{m,i} \end{bmatrix} = \begin{bmatrix} y_{0,i} \\ y_{1,i} \\ y_{2,i} \\ \vdots \\ y_{n,i} \end{bmatrix}$$

Colour Correction Tool

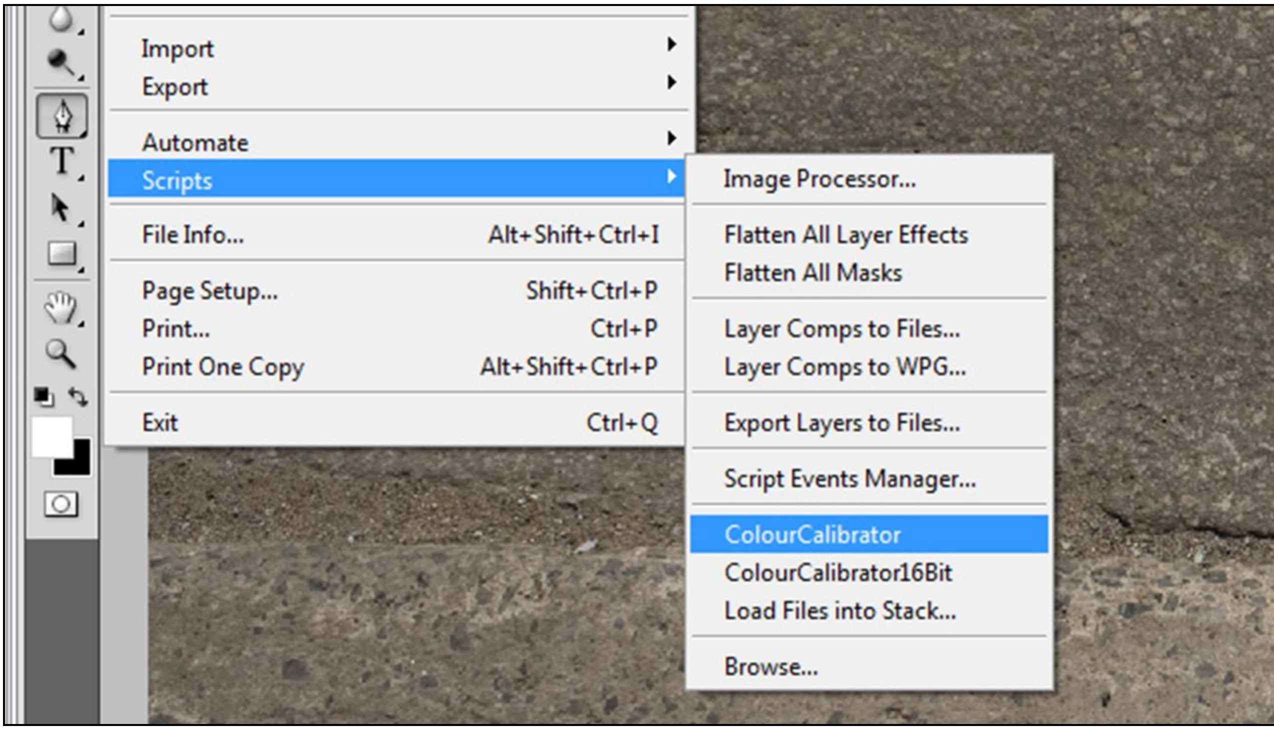
SIGGRAPH2012

- Command line tool.
- Launched by a Photoshop script.
- Operates in xyY colour space.
- Applies the following transforms [Malin11]:



Affine → Polynomial → Affine

In Photoshop we first have to click around the ColorChecker to tell the tool where to find it.



Then we can run the colour calibration script.

Future Work

SIGGRAPH2012

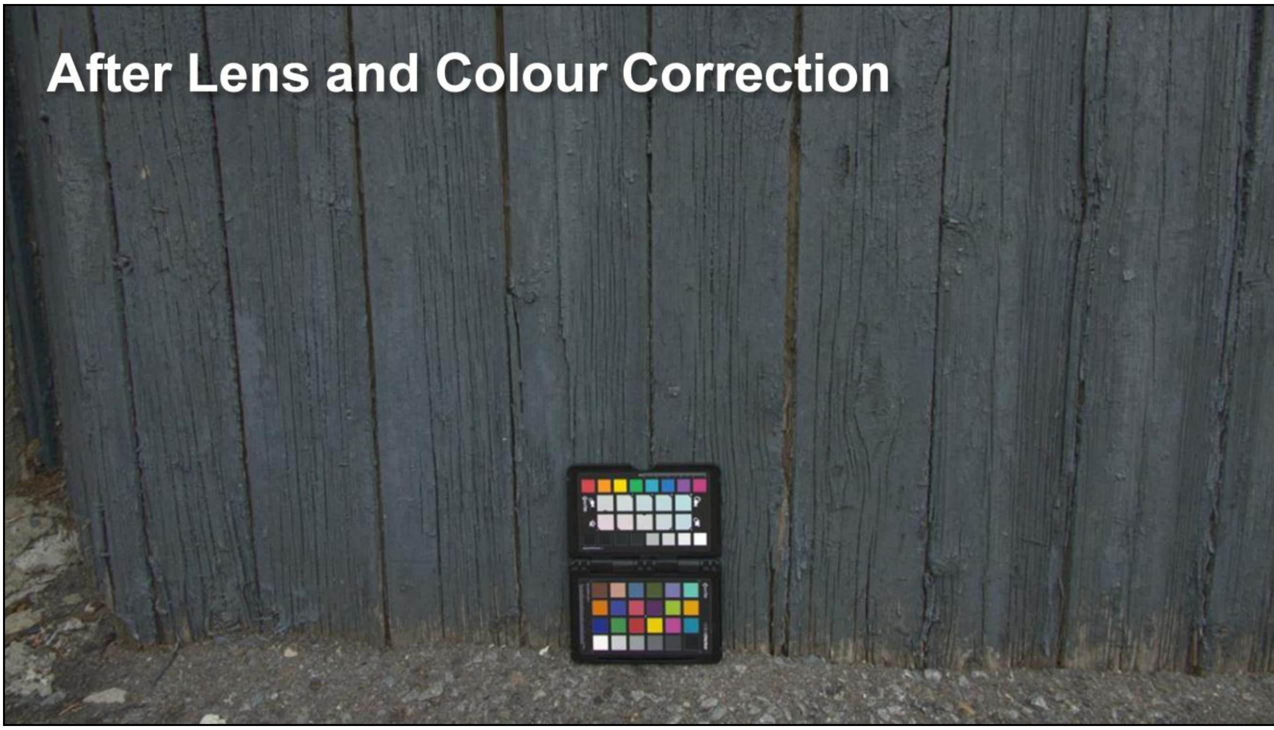
Improve tool:

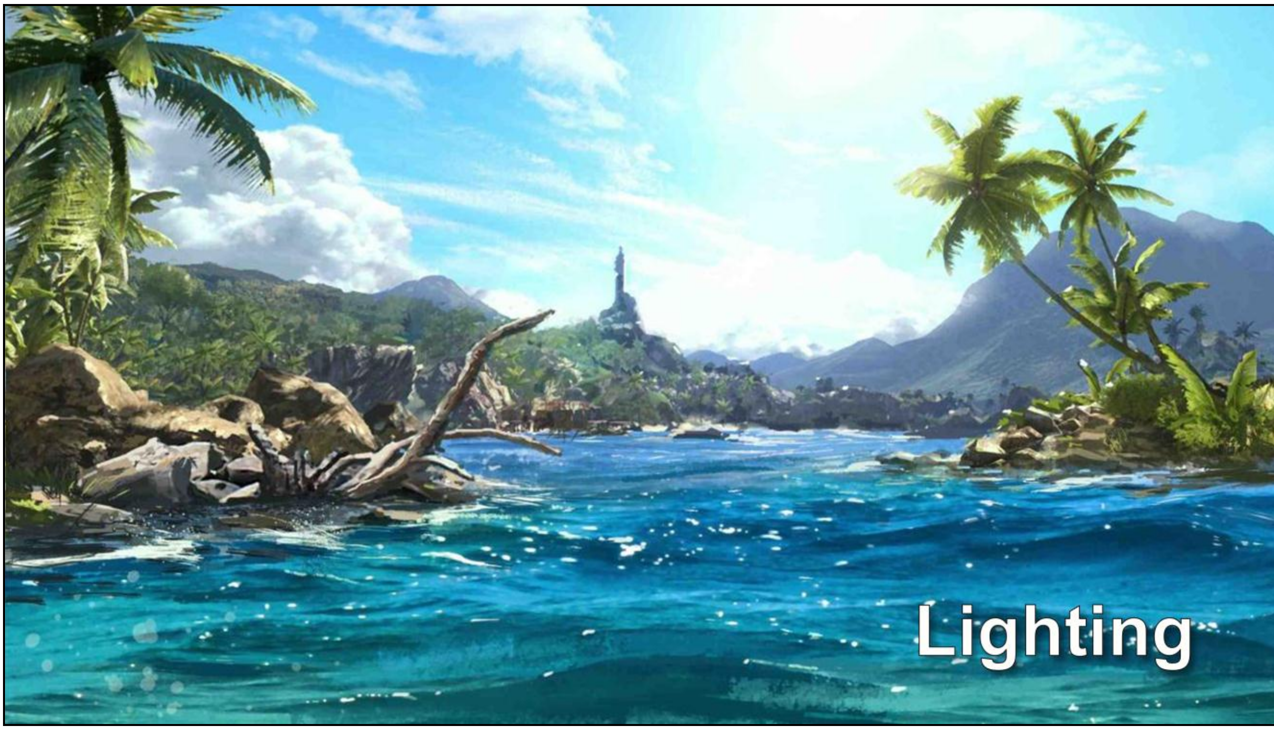
- ColorChecker™ detection
- `dcraw` to obtain raw sensor data
- Adobe Camera Model for lens correction

Improve capturing:

- Investigate polarising filters.

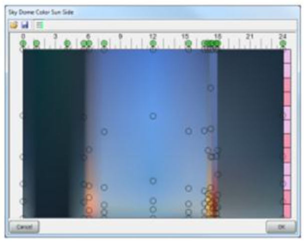

Original


After Lens and Colour Correction

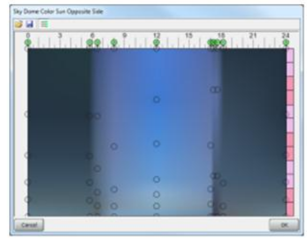

Lighting

Sky Colour

- Hue and saturation given by two gradient textures:



Sun side



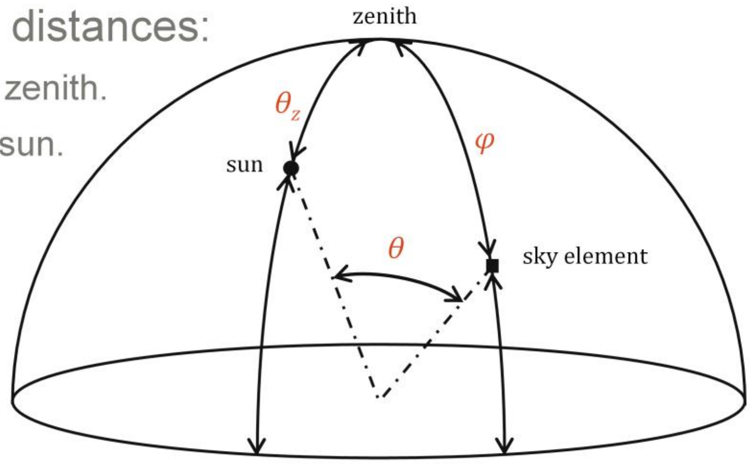
Sun opposite side

- Luminance given by CIE Sky Model.
- Cloud layers blended on top.

CIE Sky Model

SIGGRAPH2012

- Models the *luminance* distribution of the sky.
- Based on angular distances:
  - φ - sky element to zenith.
  - θ - sky element to sun.
  - θ_z - sun to zenith.



This only models the luminance of the sky, not its hue or saturation.

CIE Sky Model

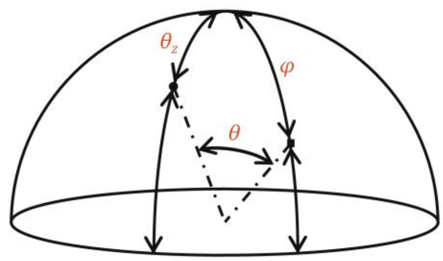
SIGGRAPH2012

- Luminance is given relative to that of the zenith:

$$\frac{L_{\theta,\phi}}{L_{\theta_z}} = \frac{(1 + a\exp(\frac{b}{\cos\phi}))(1 + c(\exp(d\theta) - \exp(d\frac{\pi}{2}) + e\cos^2\theta)}{(1 + a\exp(b))(1 + c(\exp(d\theta_z) - \exp(d\pi/2)) + e\cos^2\theta_z)}$$

- Five coefficients to give different sky models.

φ - sky element to zenith
θ - sky element to sun
θ_z - sun to zenith



The coefficients can change the model from a clear sky model to an overcast sky model, as well as differentiating between different atmospheric conditions.

CIE Sky Model

SIGGRAPH2012

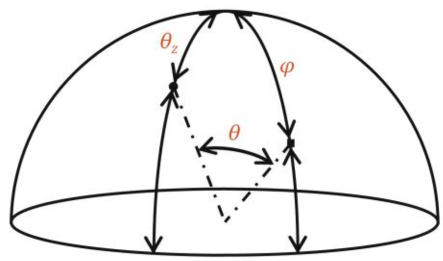
- Make relative to sun intensity not zenith intensity:
  - More consistent sky luminance over all times of day

$$\frac{L_{\theta,\phi}}{L_{\theta_z}} = \frac{(1 + a\exp(\frac{b}{\cos\phi}))(1 + c(\exp(d\theta) - \exp(d\frac{\pi}{2}) + e\cos^2\theta)}{(1 + a\exp(\frac{b}{\cos\theta_z}))(1 + c(1 - \exp(d\pi/2)) + e)}$$

φ - sky element to zenith

θ - sky element to sun

θ_z - sun to zenith



At dawn and dusk, the intensity at the sun can be 30 times that of the zenith. Thus if you make everything relative to the zenith, overall your sky will get much, much brighter at sunrise and sunset. By making everything relative to the sun intensity, you remove this problem, and in fact, we didn't have to adjust the sky intensity at all.

CIE Sky Model

SIGGRAPH2012

- Chose custom coefficients:

	a	b	c	d	e
CIE clear sky model	-1.0	-0.32	-10.0	-3.0	0.45
Far Cry 3 sky model	-1.0	-0.08	-24.0	-3.0	0.30

Sky Lighting

SIGGRAPH2012

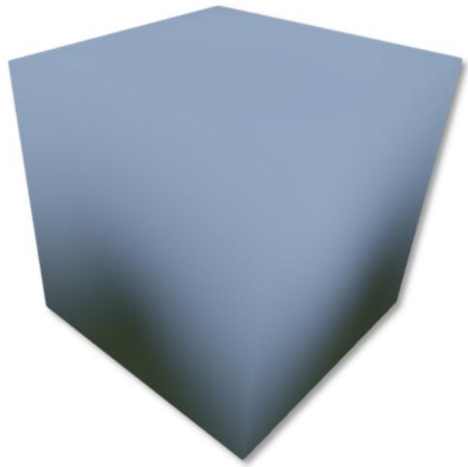
- Use sky dome as hemispherical light.

- Create SH coefficients from gradients and CIE model.
- Sky lighting and sky will always match.
- Variations in sky lighting.

This ensures that sky lighting and the sky itself will always match up.

Original Light Probe

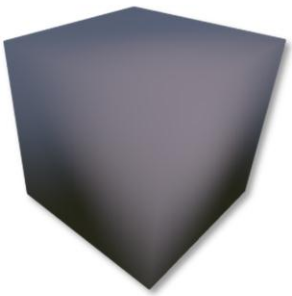
SIGGRAPH2012



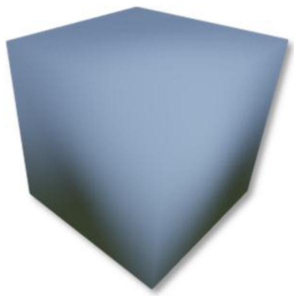
Midday

Final Light Probes

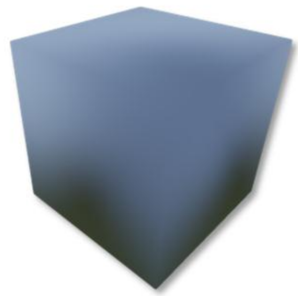
SIGGRAPH2012



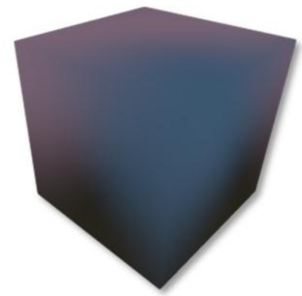
Dawn



Morning



Afternoon



Dusk


Before


After

Problems

SIGGRAPH2012

- Loss of control:
  - Concept art had different ambient and sky colours.

- Saturation:
  - Sky lighting too saturated.
  - Sky dome not saturated enough.

Artists obviously found the loss of control difficult, especially when trying to replicate concept art. They often wanted to achieve a mix of colours, so one colour for the ambient and another for the sky, to achieve the desired artistic goal.

Polarising Filter

SIGGRAPH2012

Without a polarising filter

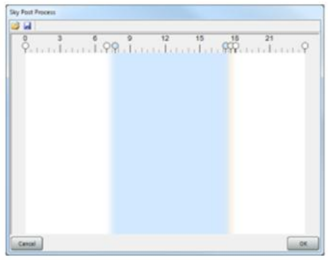
With a polarising filter

PiccoloNamek at the English language Wikipedia

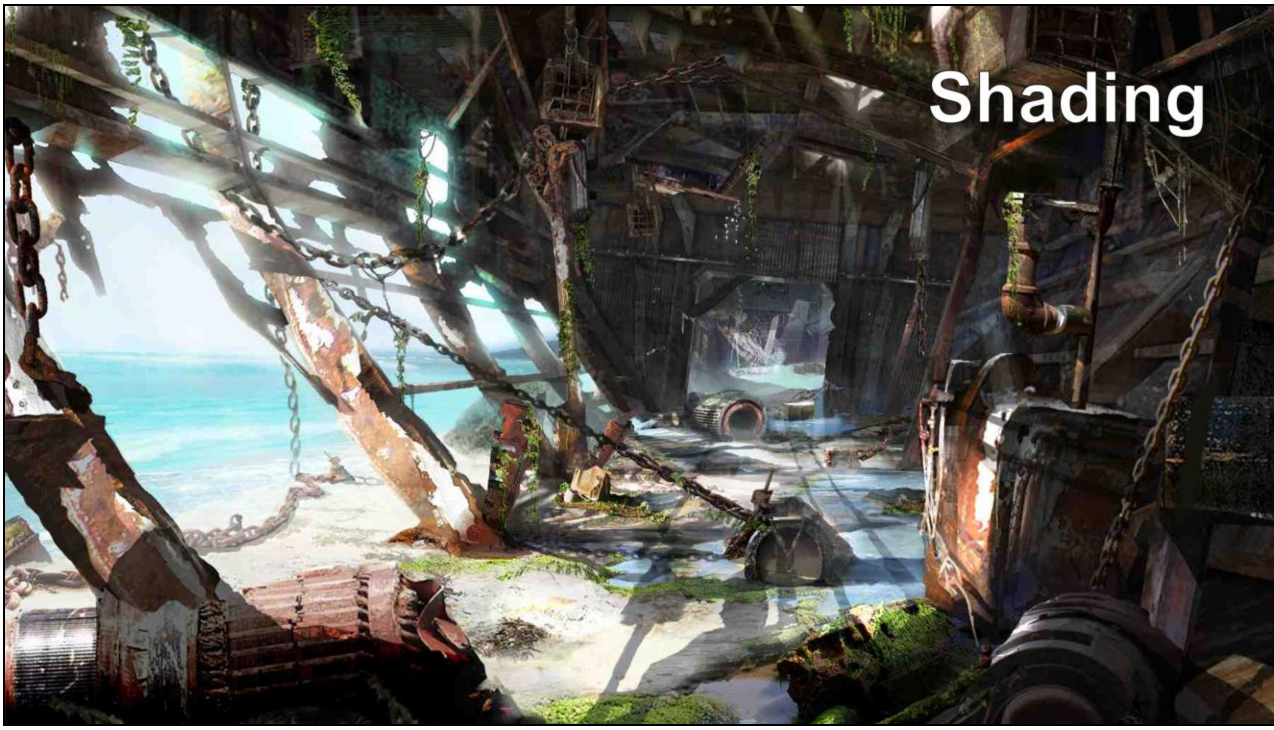
Polarising Filter

- Apply a fake polarising filter in the sky dome shader:

```
skyColor.rgb *= g_SkyPostProcess.rgb
```



- Also reduces direct sky reflection:
  - Applied to the sky in captured cube maps.


Shading

Microfacet BRDF

SIGGRAPH2012

- Torrance-Sparrow microfacet BRDF: [Lazarov11]

$$f(\mathbf{l},\mathbf{v}) = \frac{F(\mathbf{v},\mathbf{h})G(\mathbf{l},\mathbf{v},\mathbf{h})D(\mathbf{h})}{4(\underline{\mathbf{n}\cdot\mathbf{l}})(\underline{\mathbf{n}\cdot\mathbf{v}})}$$

Here, we very much follow from Dimitar Lazarov’s presentation at SIGGRAPH 2011 on Physically-Based Rendering in Call of Duty: Black Ops. This is just the specular part, we keep Lambertian diffuse on top of this. This model is great because you can pick and choose distribution and geometric terms to use.

Distribution Function

SIGGRAPH2012

- Use normalised Blinn-Phong:

$$\frac{1}{4}D(\mathbf{h}) = \frac{m+2}{8\pi}(\underline{\mathbf{n}\cdot\mathbf{h}})^m$$

- Specular power ***m*** in range [1, 8192]
- Encoded as glossiness ***g*** in [0, 1] where: $m = 2^{13g}$

Fresnel Term

SIGGRAPH2012

- Schlick's approximation for Fresnel:

$$F_{Schlick}(\mathbf{v},\mathbf{h}) = \mathbf{c}_{\text{spec}} + (1-\mathbf{c}_{\text{spec}})(1-\underline{\mathbf{h}\cdot\mathbf{v}})^5$$

- Spherical gaussian approximation: [Lagarde12]

$$F_{SG}(\mathbf{v},\mathbf{h}) = \mathbf{c}_{\text{spec}} + (1-\mathbf{c}_{\text{spec}})e^{-6\underline{\mathbf{h}\cdot\mathbf{v}}}$$

Visibility Term

- Call the remaining terms the *visibility term*:

$$V(\mathbf{l}, \mathbf{v}, \mathbf{h}) = \frac{G(\mathbf{l}, \mathbf{v}, \mathbf{h})}{(\underline{\mathbf{n} \cdot \mathbf{l}})(\underline{\mathbf{n} \cdot \mathbf{v}})}$$

- Many games have *V(**l**, **v**, **h**) = 1* making specular too dark. [Hoffman10] [Lazarov11]

Visibility Term

SIGGRAPH2012

- Schlick-Smith:
  - Roughness dependent.

$$V(\mathbf{l},\mathbf{v},\mathbf{h}) = \frac{1}{((\underline{\mathbf{n}\cdot\mathbf{l}})(1-a)+a)((\underline{\mathbf{n}\cdot\mathbf{v}})(1-a)+a)}$$

- Calculate ***a*** from Beckmann roughness ***k*** or Blinn-Phong specular power ***m***: [Lazarov11]

$$a = \sqrt{\frac{2}{\pi}}k = \frac{1}{\sqrt{\frac{\pi}{4}m + \frac{\pi}{2}}}$$

Approximating Schlick-Smith

SIGGRAPH2012

- Cost of Schlick-Smith is 8 instructions per light. [Lazarov2011]
- Too expensive for our budget:
  - Especially when applied to every light.
- Approximate by integrating Schlick-Smith over the hemisphere.

Approximating Schlick-Smith

SIGGRAPH2012

$$\frac{1}{2\pi}\int_{\Omega}\frac{1}{((\underline{\mathbf{n}\cdot\mathbf{l}})(1-a)+a)((\underline{\mathbf{n}\cdot\mathbf{v}})(1-a)+a)}\mathrm{d}\Omega$$

$$=\left[\int_{0}^{\frac{\pi}{2}}\frac{\sin(\theta)}{(\cos(\theta)(1-a)+a)}\mathrm{d}\theta\right]^2$$

$$=\left[\frac{-\log(a)}{1-a}\right]^2$$

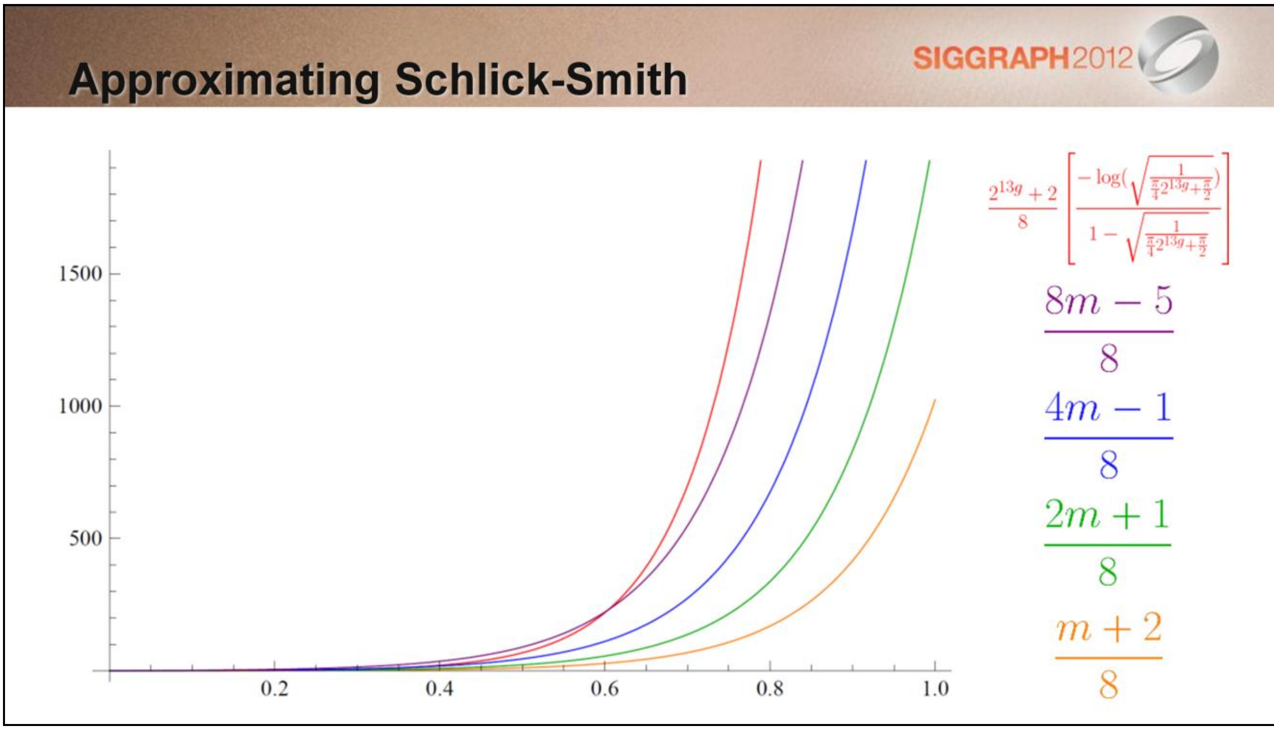
Approximating Schlick-Smith

SIGGRAPH2012

- Still 8 instructions! ☹
- Dependent only on glossiness…
- …just like our existing energy conservation term.
- Find a cheap curve to approximate both:

$$E(g) = \frac{2^{13g}+2}{8}\left[\frac{-\log(\sqrt{\frac{1}{\frac{\pi}{4}2^{13g}+\frac{\pi}{2}}})}{1-\sqrt{\frac{1}{\frac{\pi}{4}2^{13g}+\frac{\pi}{2}}}}\right]^2$$

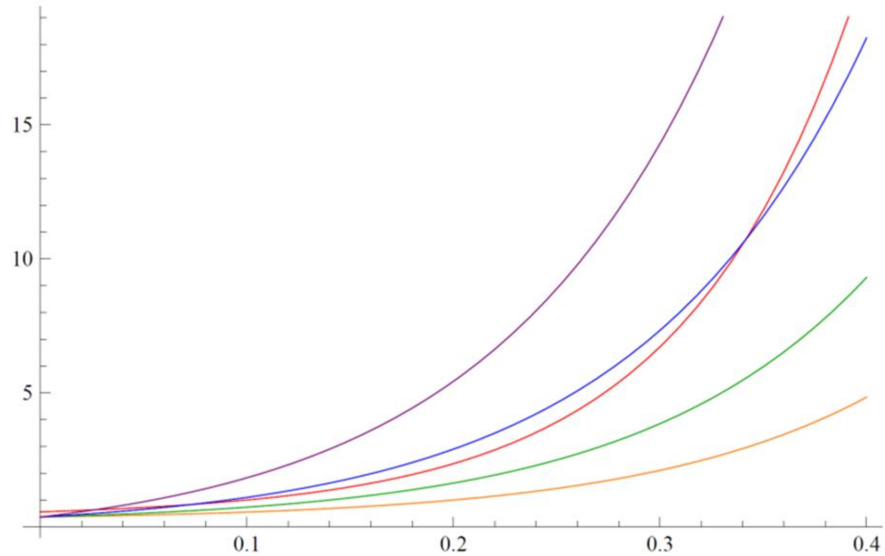
Approximate relative to glossiness g as it’s very close to a linear representation of highlight size.



We looked at functions of the form (am + b)/8, just a simple modification of our initial energy conservation term (in yellow at the bottom). Observe that we're careful to keep the value when m = 1 the same. This is because at low roughnesses, our artists desired to keep specular as dull as possible, and any brightening was incredibly noticeable.

Approximating Schlick-Smith

SIGGRAPH2012



$$\frac{2^{13g}+2}{8}\left[\frac{-\log(\sqrt{\frac{1}{\frac{\pi}{4}2^{13g}+\frac{\pi}{2}}})}{1-\sqrt{\frac{1}{\frac{\pi}{4}2^{13g}+\frac{\pi}{2}}}}\right]$$

$$\frac{8m-5}{8}$$

$$\frac{4m-1}{8}$$

$$\frac{2m+1}{8}$$

$$\frac{m+2}{8}$$

Only the yellow and green lines stay beneath the red curve at all points.

Approximating Schlick-Smith

SIGGRAPH2012

- Chose the following approximation:

$$E(g) \simeq \frac{2m+1}{8}$$

- Simple (and free) modification of initial energy conservation term.
- Preferred to underestimate:
  - Schlick-Smith known to over brighten.
  - Makes artefacts less visible.

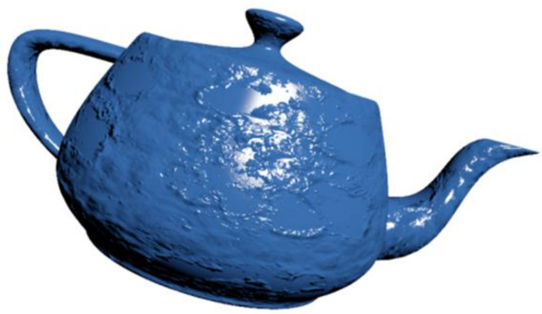
As this is an approximation, it both overestimates and underestimates the correct values, and any overestimation was really noticeable.

Also, despite our best efforts to calibrate our diffuse albedo, it erred on the dark side, so our specular was naturally brighter because of that.

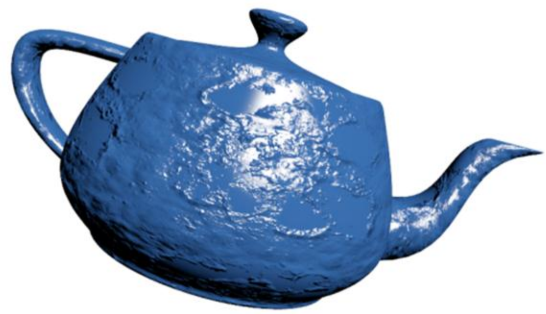
Specular Filtering

SIGGRAPH2012

- Needed specular filtering to:
  - Reduce shimmering.
  - Preserve appearance in the distance.



Without Filtering

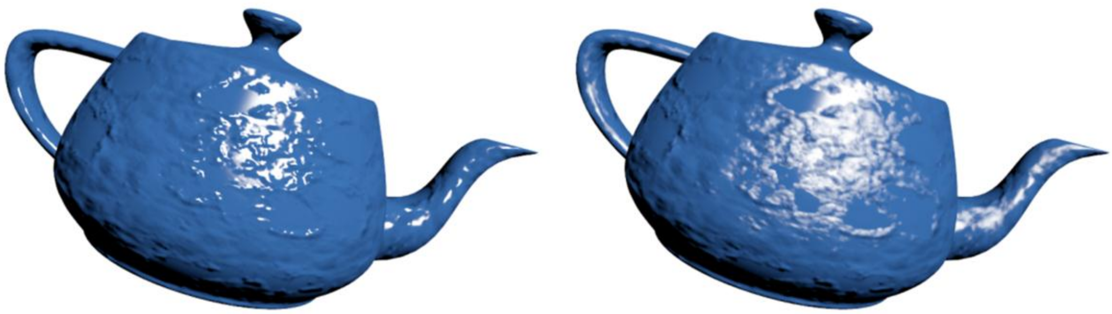


With Filtering

Specular Filtering

SIGGRAPH2012

- Needed specular filtering to:
  - Reduce shimmering.
  - Preserve appearance in the distance.



Without Filtering　　　　With Filtering

Specular Filtering

SIGGRAPH2012

- Use Toksvig's formula to scale the specular power, using the length of the filtered normal: [Hill11]

$$m' = \frac{||\mathbf{n}_a||m}{||\mathbf{n}_a|| + m(1 - ||\mathbf{n}_a||)}$$

- Store these scales in a texture.
- Ideally use to adjust gloss map.

Specular Filtering

SIGGRAPH2012

- Cost of adding an extra texture too high.
- Gloss maps not used in every shader.
  - Could not combine Toksvig map with existing gloss map.

- Free channels in our DXT5 compressed normal maps:

0xFF	Y	0x00	X

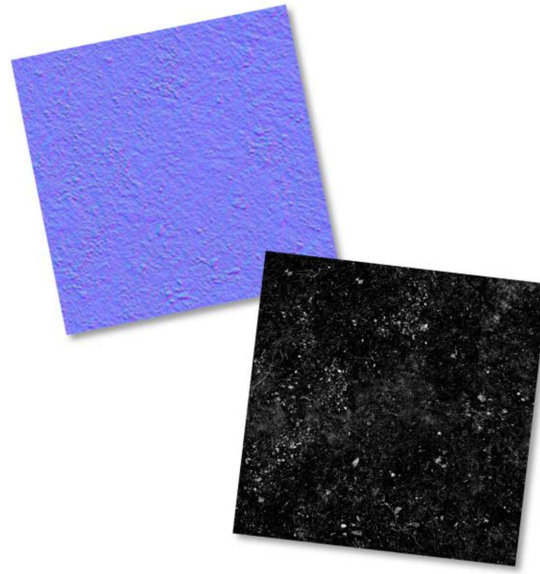
Cost of an extra map was too high for both performance and memory.

Specular Filtering

SIGGRAPH2012

- Artists paint gloss in **alpha** channel of normal map.
- Store gloss combined with Toksvig in red channel:

Gloss	Y	0x00	X



In Photoshop, the artists could paint a gloss map and place it into the alpha channel of their normal map. On export, Toksvig was applied and the result was placed into the red channel of the DXT5 compressed normal map.

Toksvig Maps

SIGGRAPH2012

- Compression of normal y component affected:

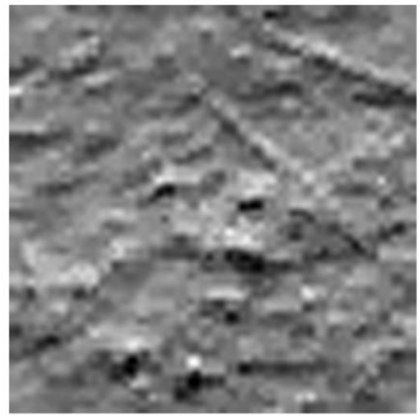
Without gloss

With gloss

Toksvig Maps

- Compression of normal y component affected:

Without gloss

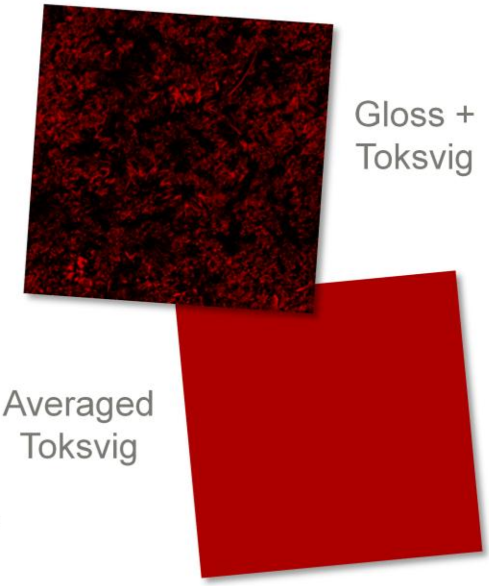


With gloss

Toksvig Maps

SIGGRAPH2012

- Gloss map is optional.
- With:
  - Combine with Toksvig.
- Without:
  - Average Toksvig to single value.



Averaging Toksvig to a single value allowed us to have some form of specular filtering with no extra textures and no compression artefacts.


Toksvig Off

This is just using the average Toksvig factor.

Conclusions

SIGGRAPH2012

- Calibrate your materials!
- Physically-based is a great starting point:
  - Tweaking is needed.
  - Not everything is simulated.
  - Good tools are essential.
- Can make the change late in a project.
- Possible on current-generation consoles.

References

SIGGRAPH2012

- [Darula02] Stanislav Darula and Richard Kittler, *CIE General Sky Standard Defining Luminance Distributions*, eSim 2002, http://mathinfo.univ-reims.fr/IMG/pdf/other2.pdf
- [Hill11] Stephen Hill, *Specular Showdown in the Wild West*, http://blog.selfshadow.com/2011/07/22/specular-showdown/
- [Hoffman10] Naty Hoffman, *Crafting Physically Motivated Shading Models for Game Development*, SIGGRAPH 2010, http://renderwonk.com/publications/s2010-shading-course/
- [Lagarde12] Sébastien Lagarde, *Spherical Gaussian Approximation for Blinn-Phong, Phong and Fresnel*, 2012, http://http://seblagarde.wordpress.com/2012/06/03/spherical-gaussien-approximation-for-blinn-phong-phong-and-fresnel/
- [Lazarov11] Dimitar Lazarov, *Physically Based Lighting in Call of Duty: Black Ops*, SIGGRAPH 2011, http://advances.realtimerendering.com/s2011/index.html
- [Malin11], Paul Malin, personal communication

Thanks

SIGGRAPH2012

- **Naty Hoffman**, *2K Games* (@renderwonk)
- **Paul Malin**, *Activision Central Tech* (@p_malin)
- **Stephen Hill**, *Ubisoft Montréal* (@self_shadow)
- **Philippe Gagnon**, *Ubisoft Montréal*
- **Mickael Gilabert**, *Ubisoft Montréal* (@mickaelgilabert)
- **Vincent Jean**, *Ubisoft Montréal*
- **Minjie Wu**, *Ubisoft Montréal*
- **Derek Nowrouzezahrai**, *Université de Montréal*

Any Questions?

stephen.mcauley@ubisoft.com

@stevemcauley

http://blog.stevemcauley.com/

SIGGRAPH2012