Lighting And Material Of Halo3

Hao Chen (haochen@bungie.com)

Bungie

Microsoft Research Asia

NON FACETE NOBIS CALCITRARE VESTRVM

Talk Overview

- Introduction
- Halo3 Lighting
- Halo3 Material Model
- HDR Rendering
- Results
- Acknowledgement
- Q&A

Halo2

- per pixel light map
- per vertex dominant lightdir+"ambientess"
- Bump maps modulates light map color.
- Objects sample LM and build fake lights.

- Bumps only see single directional light
- Bumps disappear In shadowy area
- Specular from point lights only
- LDR pipeline

Halo3

- What do we want?
  - global illumination
  - handle variety of lighting environments
  - consistent lighting everywhere
  - render bump maps “correctly”
  - complex material under complex lighting
  - HDR

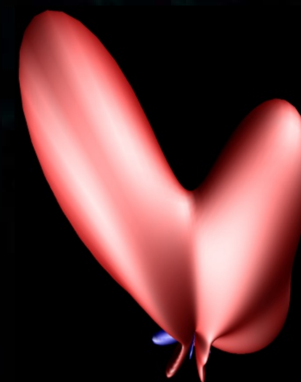
Definitions

- Radiance
  - *L(ω)*: density of light energy through a given point in a given direction.
- Irradiance
  - *I:* incident radiance integrated over the hemisphere of the surface normal with the cosine lobe.
- BRDF
  - *f(V, L):* Bidirectional reflectance distribution function.
- Fresnel
  - *F:* Predicts ratio of reflected and transmitted light when light travels between different mediums.
  - *F0*: Reflectance at near normal incident angle.

Related Work

- Irradiance Volume [Greger98][ATI05]
- SH Irradiance Environment Map [Ramamoorthi01]
- Pre-computed Radiance Transfer [Sloan02]
- SH Light Maps [Good, Taylor05]
- Sky and Atmosphere [Preetham99][Hoffman02]
- Reflectance Models [CookTorrance82][Schlick94]
- Low Frequency Glossy Material [Kautz02][Sloan03]
- Frequency Space Environment Map [Ramamoorthi02]

SH Irradiance Env Map



[Ramamoorthi00]

$$L_{lm} = \iint_{\theta,\phi} L(\theta,\phi) Y_{lm}(\theta,\phi) \sin(\theta) d\theta d\phi$$

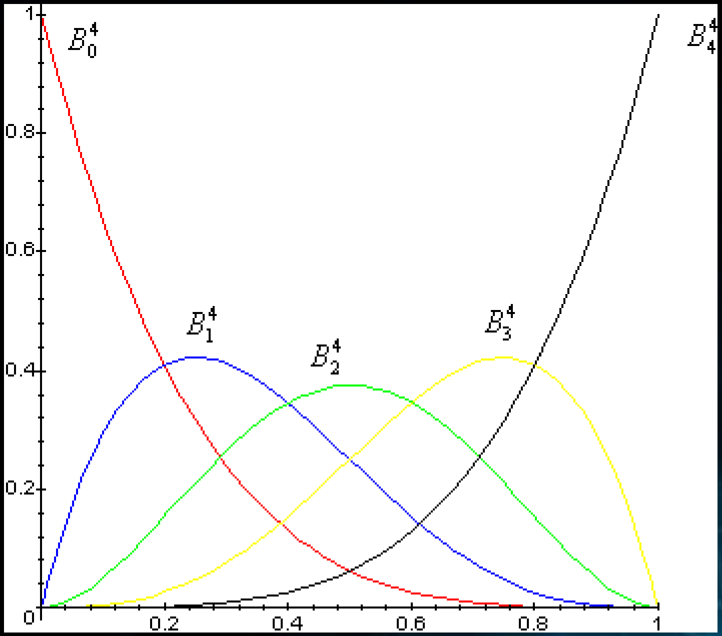
distant radiance at given direction

SH basis evaluated at given direction

diff solid angle

Harmoniques sphériques

- Base de fonctions
  - Espace vectoriel de … fonctions
  - Exemple : base de fonction de polynômes



Addon
Cours

Harmoniques sphériques

- Décomposition d'une fonction

$$f(x) = \sum_{i:indices} f_i \, \Gamma_i(x)$$

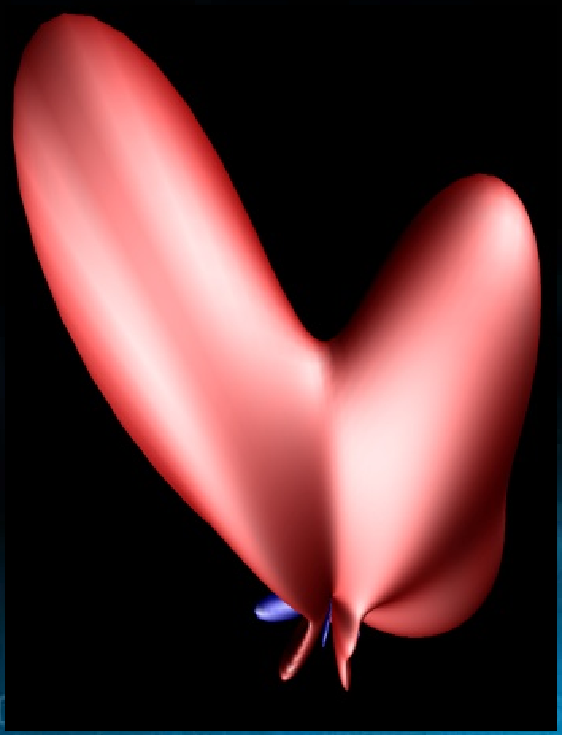
- Intérêt :
  - Précalculs sur les Γ
  - Stocke une fonction avec *n* coefficients
    - Décomposition en espace
    - Décomposition en fréquence

Harmoniques sphériques

- Problématique :
  - Capturer une fonction variant dans des coordonnées sphériques

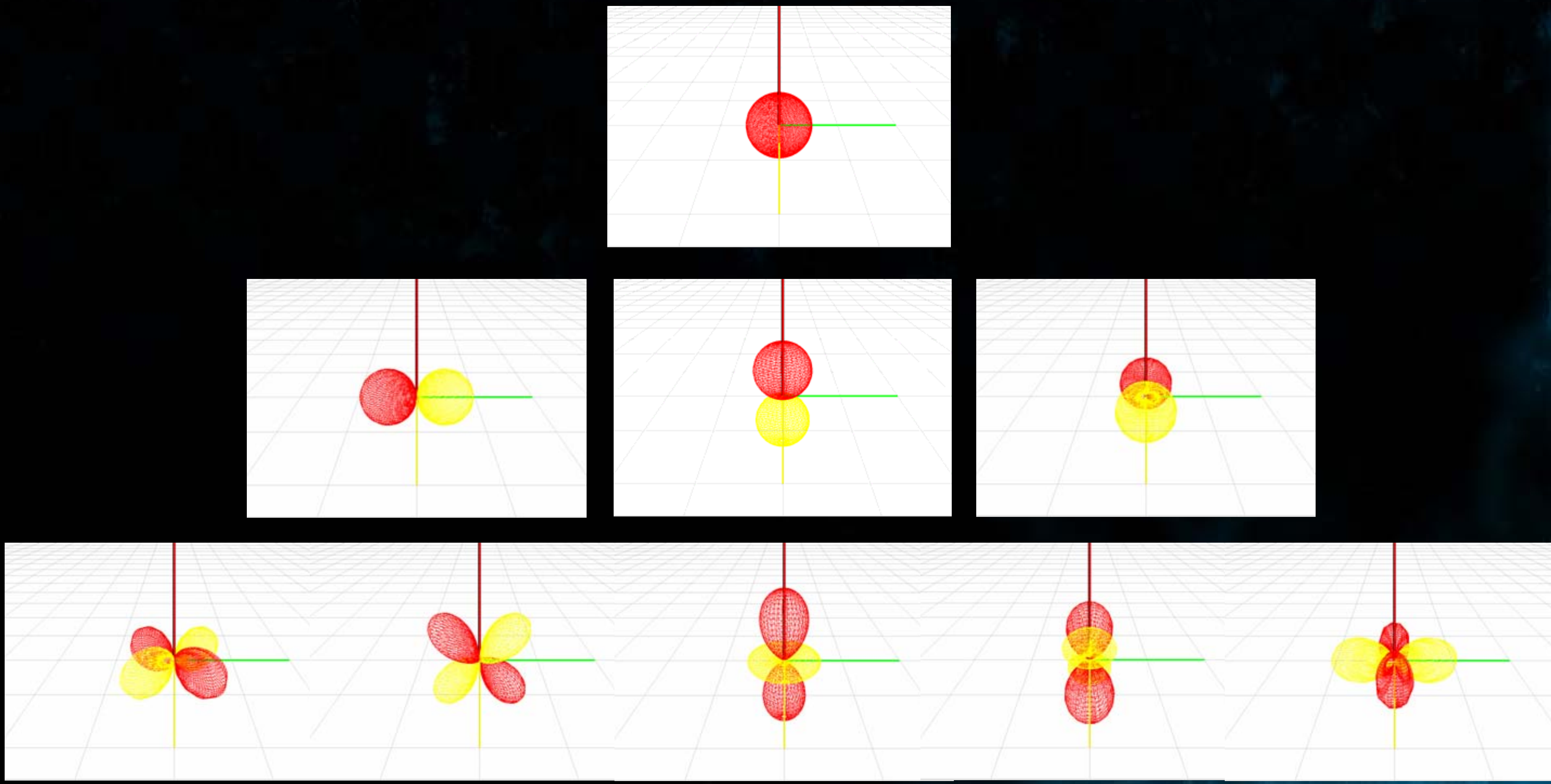
Décomposition en fréquence

En 1 point donné



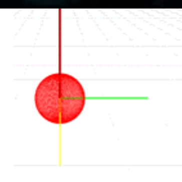
BUNGIE

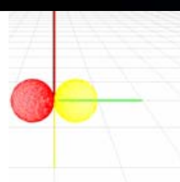
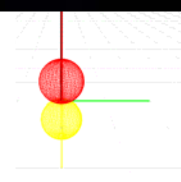
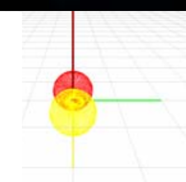
Harmoniques sphériques

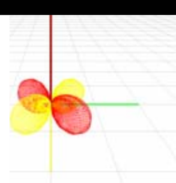
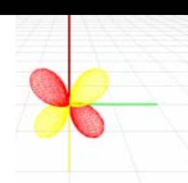
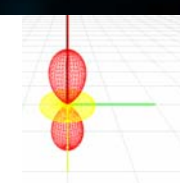


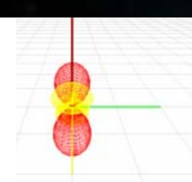
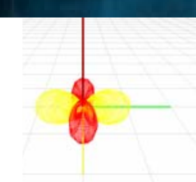
Harmoniques sphériques

- Décomposition d’une fonction

$L(\omega) = c_0^0 \times$  $+$

$c_1^{-1} \times$  $+$ $c_1^{0} \times$  $+$ $c_1^{1} \times$  $+$

$c_2^{-2} \times$  $+$ $c_2^{-1} \times$  $+$ $c_2^{0} \times$  $+$

$c_2^{1} \times$  $+$ $c_2^{2} \times$ 

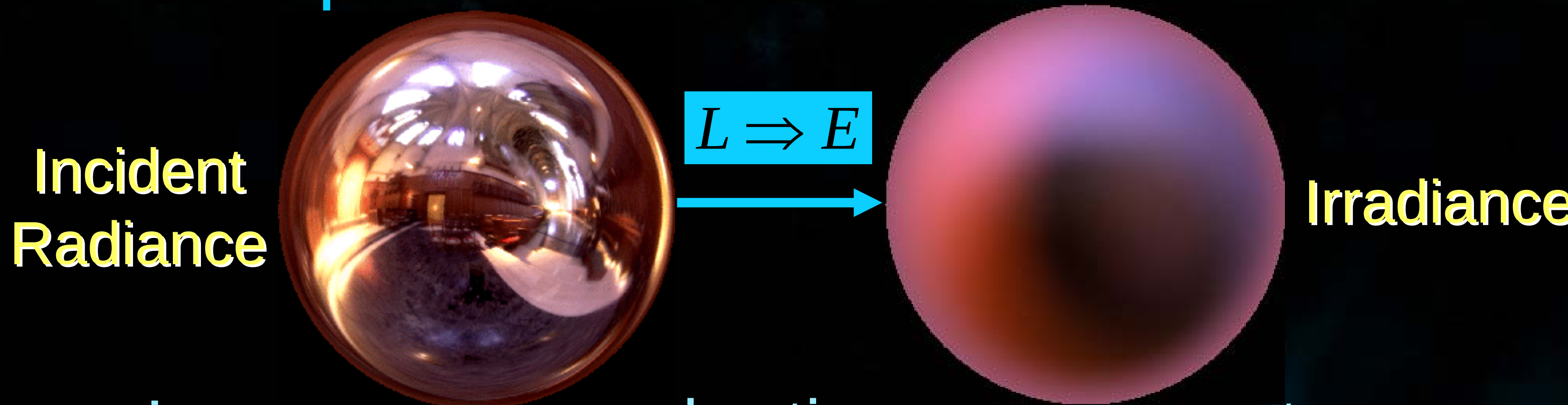
Harmoniques sphériques

- Fonction représentée par un ensemble de coefficient :

$$\Lambda = [c_l^m] = \begin{bmatrix} c_0^0 & c_1^{-1} & \cdots & c_{n-1}^{n-1} \end{bmatrix}^T$$

Exemple : calcul de l'irradiance

- A priori, pour chaque pixel, intégrale sur l'hémisphère



  - Les surfaces lambertiennes agissent comme des filtres passe bas
  - Idée : Faire une décomposition (analyse) en fréquence

An Efficient Representation for Irradiance Environment Maps
Ravi Ramamoorthi, Pat Hanrahan

Harmoniques sphériques

- Sources wolfram :

$$Y_l^m(\theta,\phi) = \sqrt{\frac{2l+1}{4\pi}\frac{(l-m)!}{(l+m)!}}\,P_l^m(\cos\theta)\,e^{im\phi}$$

Coefficient de pondération

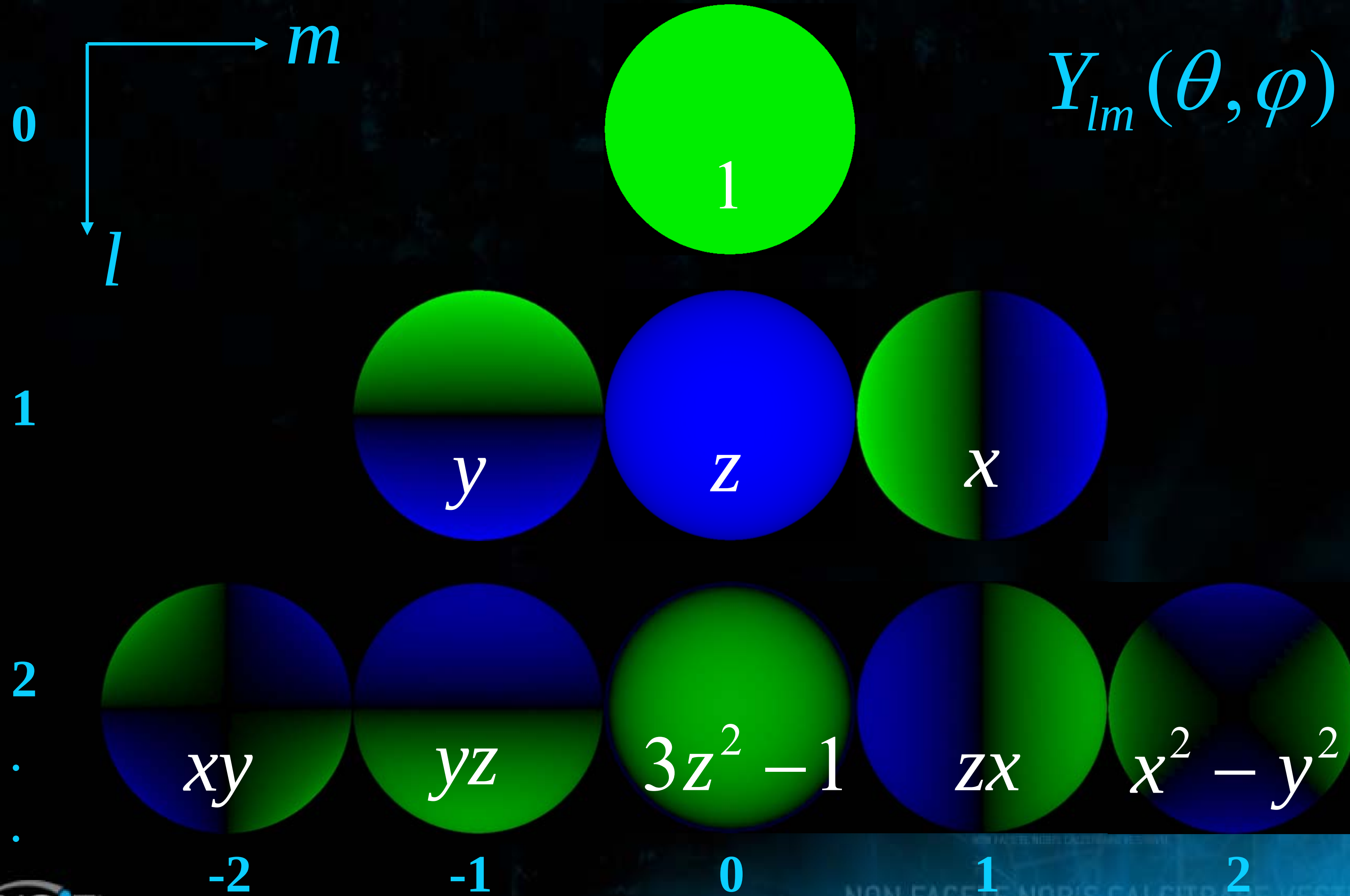
Polynôme de Legendre (agit sur le cosinus)

Exponentielle complexe !

Solution de l'équa. dif.

$$\left[\frac{1}{\sin\theta}\frac{\partial}{\partial\theta}\left(\sin\theta\frac{\partial}{\partial\theta}\right)+\frac{1}{\sin^2\theta}\frac{\partial^2}{\partial\phi^2}+l(l+1)\right]u=0,$$

Harmoniques sphériques

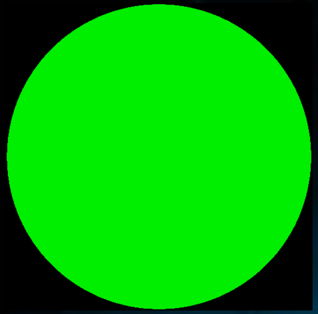
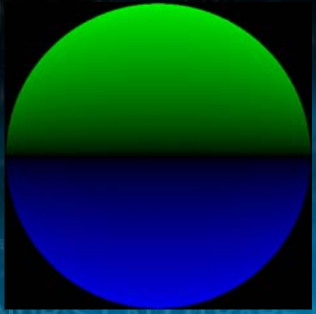


Utilisation des harmoniques sphériques

Sur l'illumination (L) et l'irradiance (E)

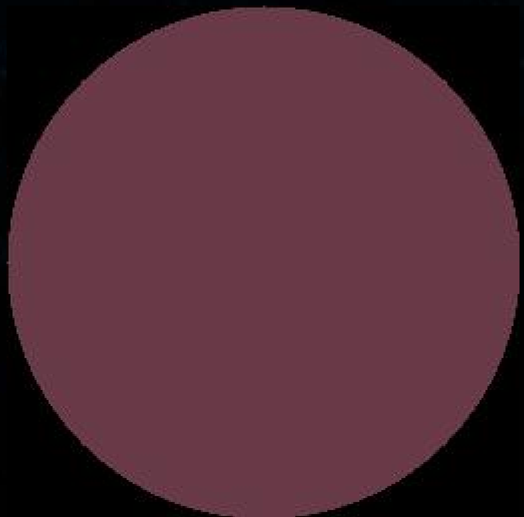
$$L(\theta,\phi) = \sum_{l=0}^{\infty} \sum_{m=-l}^{+l} L_{lm} Y_{lm}(\theta,\phi)$$

$$E(\theta,\phi) = \sum_{l=0}^{\infty} \sum_{m=-l}^{+l} E_{lm} Y_{lm}(\theta,\phi)$$

 $= .67$  $+ .36$  $+ \ldots$

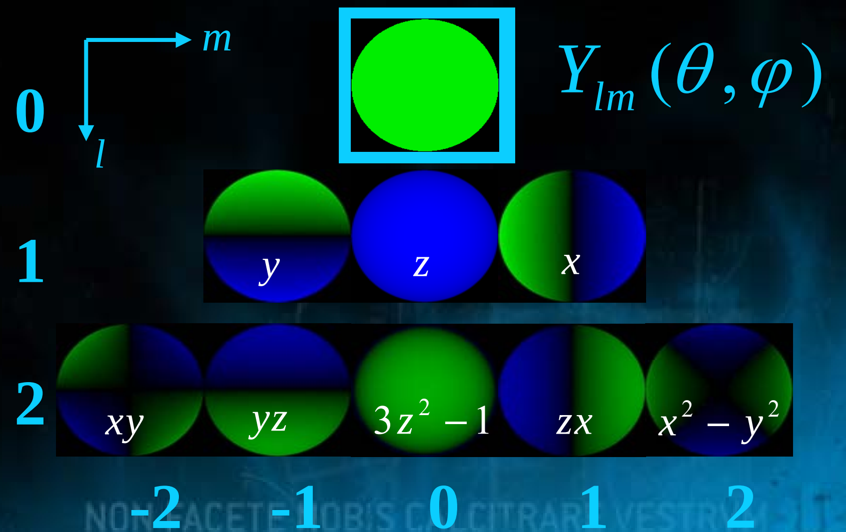
Approximation 9 paramètres

Image exacte



Ordre 0
1 terme

Erreur RMS = 25 %

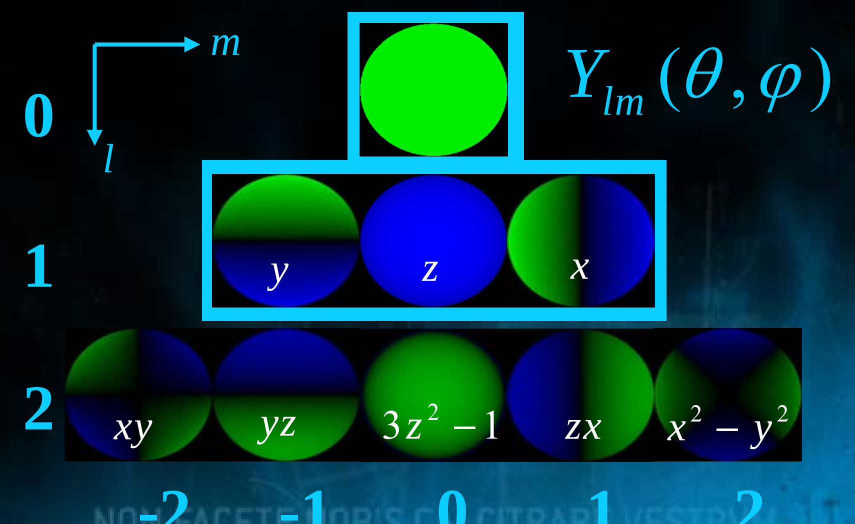


Approximation 9 paramètres

Image exacte

Ordre 1
4 termes

Erreur RMS = 8%



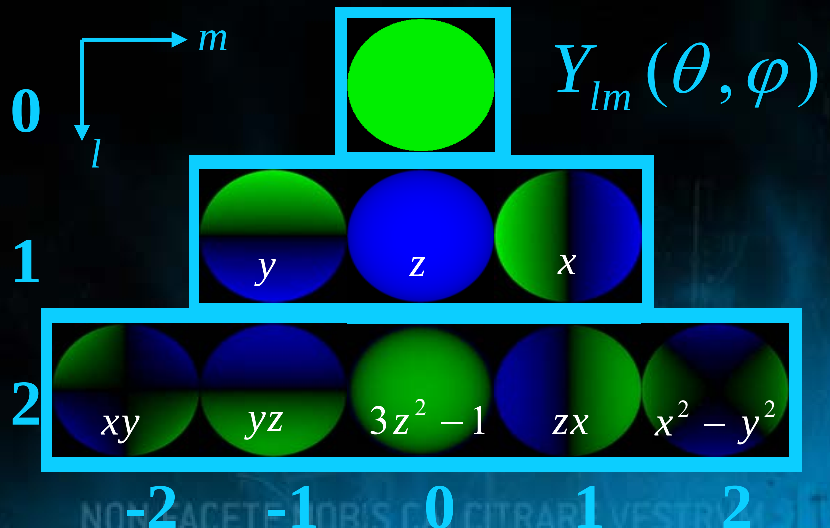
Approximation 9 paramètres

Image exacte

Ordre 2
9 termes

Erreur RMS = 1%

Pour toute illumination,
erreur moyenne < 3%
[Basri Jacobs 01]



BUNGIE

Calcul des coefficients

- Calcul des 9 coefficient L_{lm}
  - 9 coefficients au lieu d'une intégrale pour chaque pixel !
  - Peuvent être considérés comme les moments de l'illumination

$$L_{lm} = \int_{\theta=0}^{\pi} \int_{\phi=0}^{2\pi} L(\theta,\phi)\, Y_{lm}(\theta,\phi)\, \sin\theta \, d\theta d\phi$$

  - Somme pondéré pour la texture d'env.

$$L_{lm} = \sum_{pixels(\theta,\phi)} envmap[pixel] \times basisfunc_{lm}[pixel]$$

Comparaison

Illumination
incidente
300x300

Irradiance map
Texture: 256x256
Intégrale
hémisphérique 2h

Irradiance map
Texture: 256x256
Coef harmoniques
sphériques 1sec

$\text{Time} \propto 300 \times 300 \times 256 \times 256$

$\text{Time} \propto 9 \times 256 \times 256$

Rendu

Irradiance approchée par interpolation quadratique

$$E(n) = c_4 L_{00} 1 + 2c_2 L_{11} x + 2c_2 L_{1-1} y + 2c_2 L_{10} z + c_5 L_{20} (3z^2 - 1) + 2c_1 L_{2-2} xy + 2c_1 L_{21} xz + 2c_1 L_{2-1} yz + c_1 L_{22} (x^2 - y^2)$$

$$E(n) = n^t M n$$

Matrice 4x4 (linéaire sur les coefficients L_{lm})

Vecteur normal $\begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$

Implementation GPU

$$E(n) = n^t M n$$

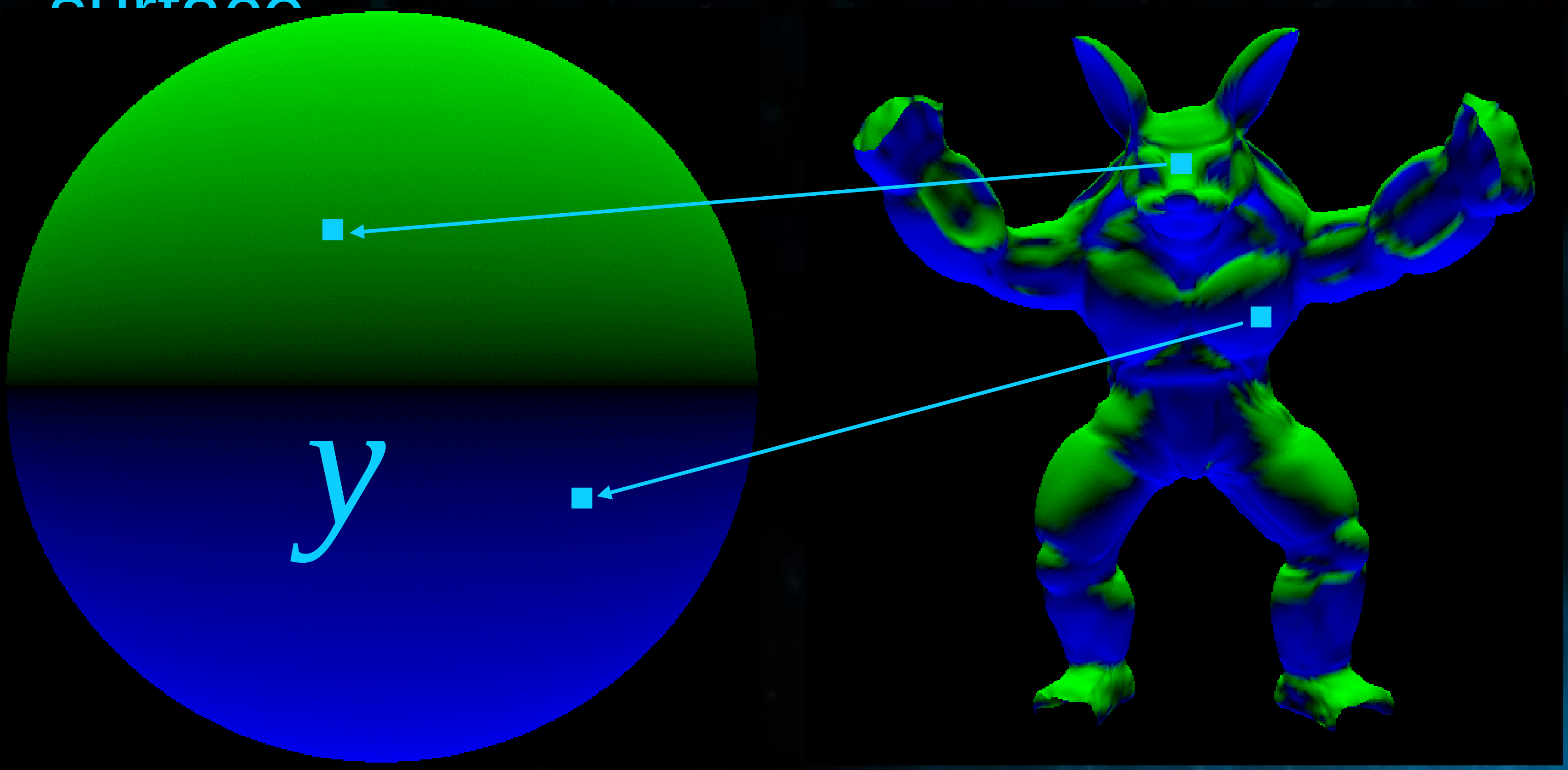
Rendu simple procedural

- Pas de texture
- Multiplication matrice vecteur et prod. scalaire
- Pixel shader

```
surface float1 irradmat (matrix4 M, float3 v) {
        float4 n = {v,1};
        return dot(n,M*n);
}
```

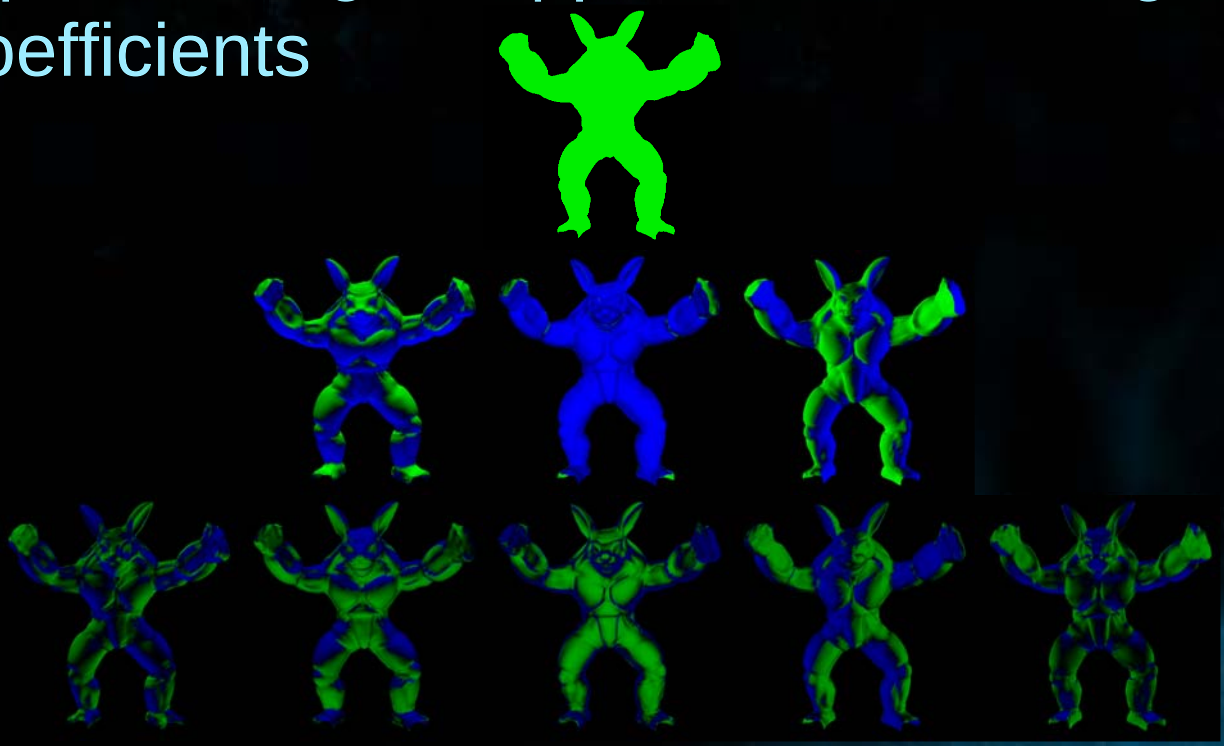
Géométrie complexe

Utilisation seulement des normales de surface



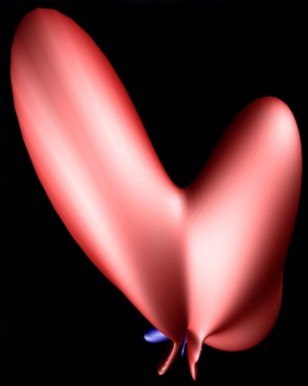
Lighting Design

- Somme pondéré par L_{lm} des fonctions
  - On peut changer l'apparence en changeant les coefficients



Demo (2001)

SH Irradiance Env Map



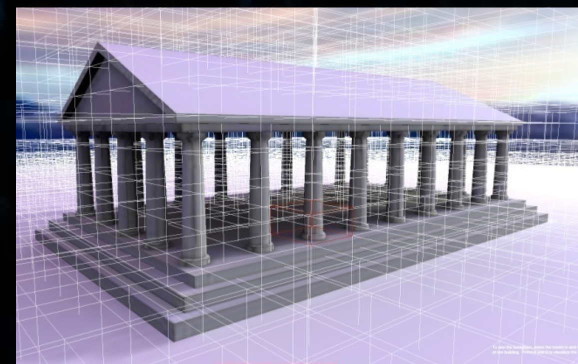
$$E(n) = n^t M n$$

- Irradiance distribution as SH vector.
- <3% error with just 9 terms [Basri Jacobs 01]
- Evaluate normal to get irradiance.
- Only represent a single point in space.
- Only for infinite lighting environment.
- What about local lighting?

Retour talk

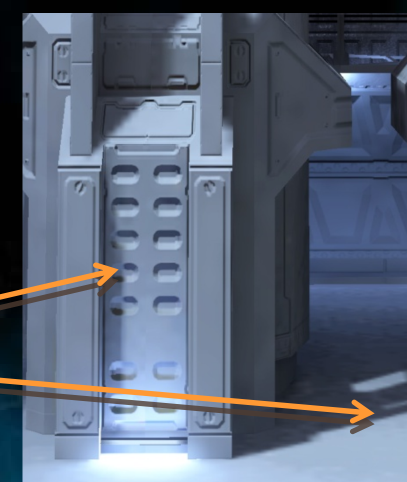
Irradiance Volumes

[Greger 98]

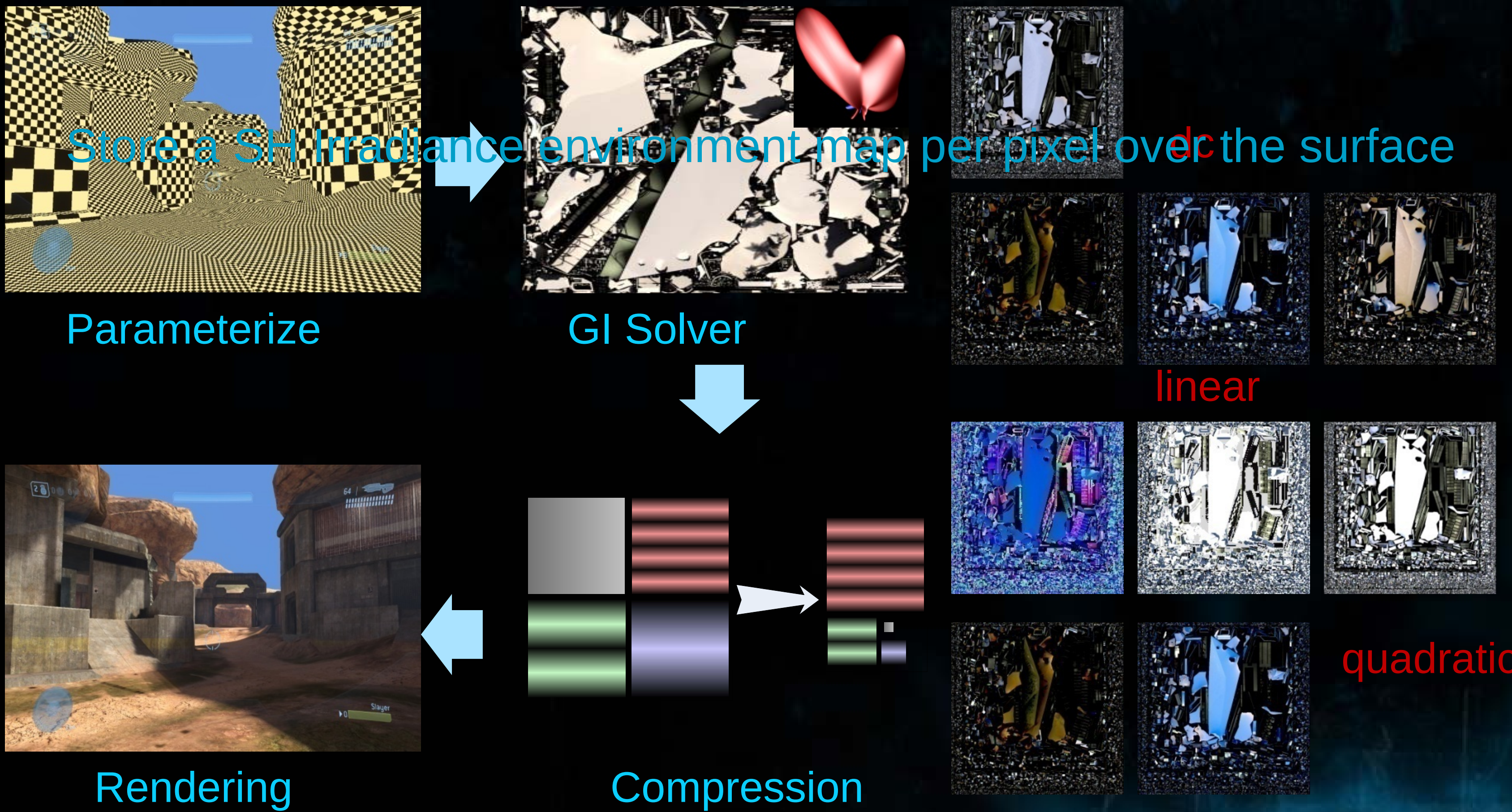


[ATI 05]

- Spatially divide volume into cells.
- irradiance volume per cell.
- Interpolate between samples.
- Sharp shadow boundaries?
- Bump maps?



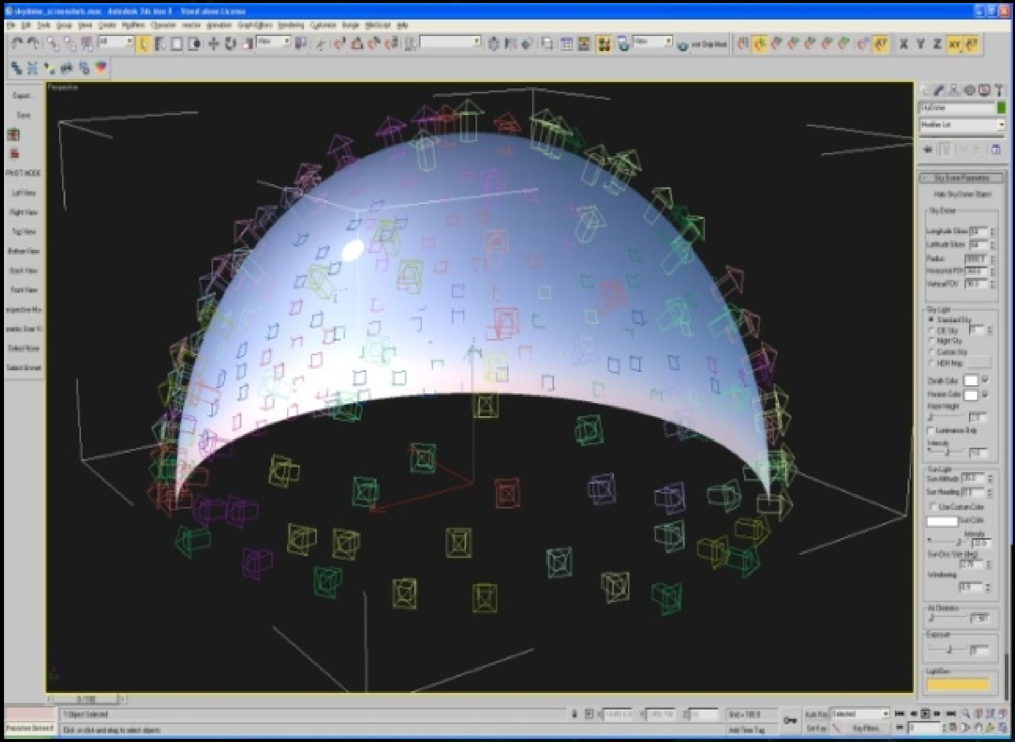
SH Light Map



Other basis would work too, e.g. Half Life Basis, ZH, etc.

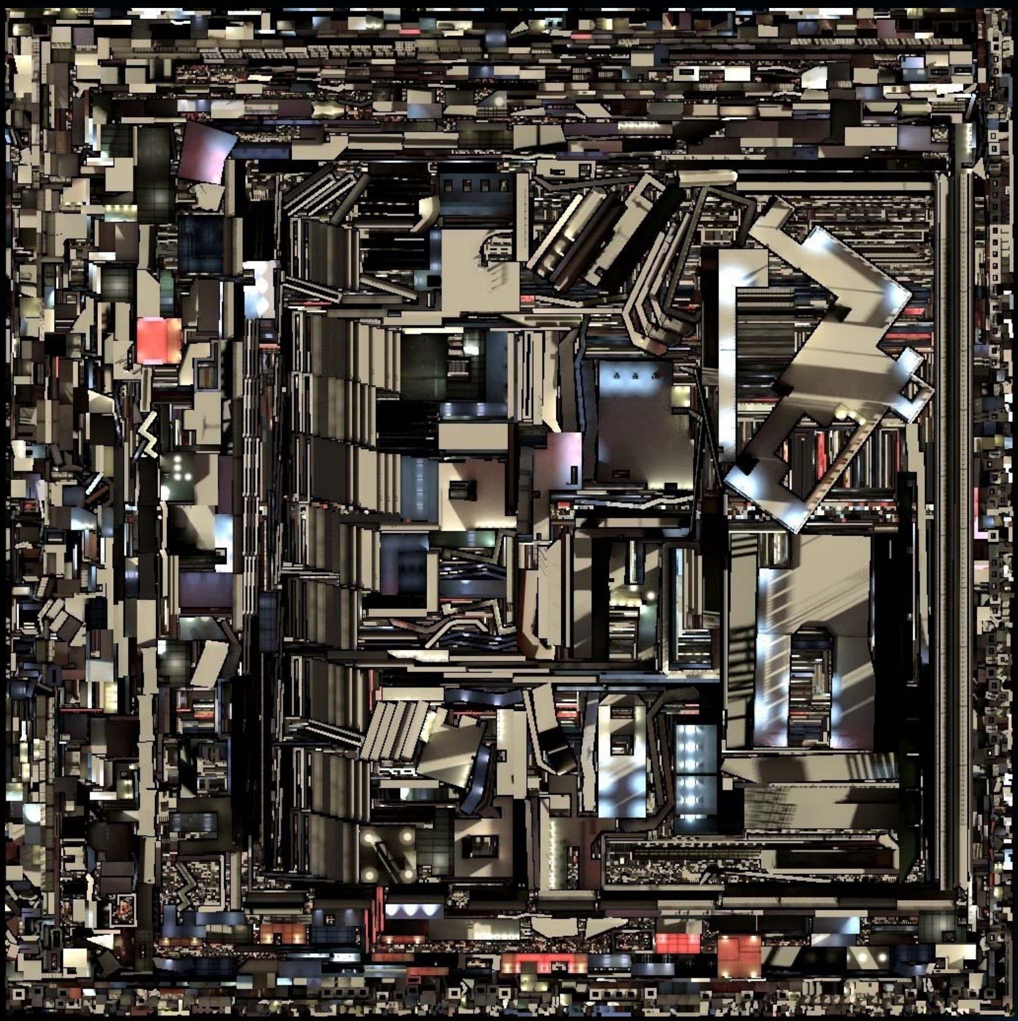
Sky and Sun

- HDR pipeline means starting with HDR light sources.
- Custom sky plugin for 3DStudio Max.
- Procedural Sky/Sun Model[Preetham99]
- CIE Sky Model
- Can also use HDR Cubemap



HDR Sky Example

Parameterization



- UVAtlas (DirectX SDK)
  - Minimize distortion
  - Efficient Packing
  - Input “importance”
- Halo3 improvements:
  - small charts placement.
  - long and thin charts.
- > 80% texel utilization
  - Halo2: < 50%

Photon Mapping Farm

Initialization → Direct Illumination → Photon Cast → Radiance Estimate → Exit Illumination → Final Gather → Signal Compression → DXT Compression

256 servers; 456 Processors; 1GB memory per proc; see Luis's talk.

Compression

Signal Optimization:

- Compute signal gradient
- Resample to half dim.
- Preserve high freq charts
- Squeeze low freq charts

DXT Compression

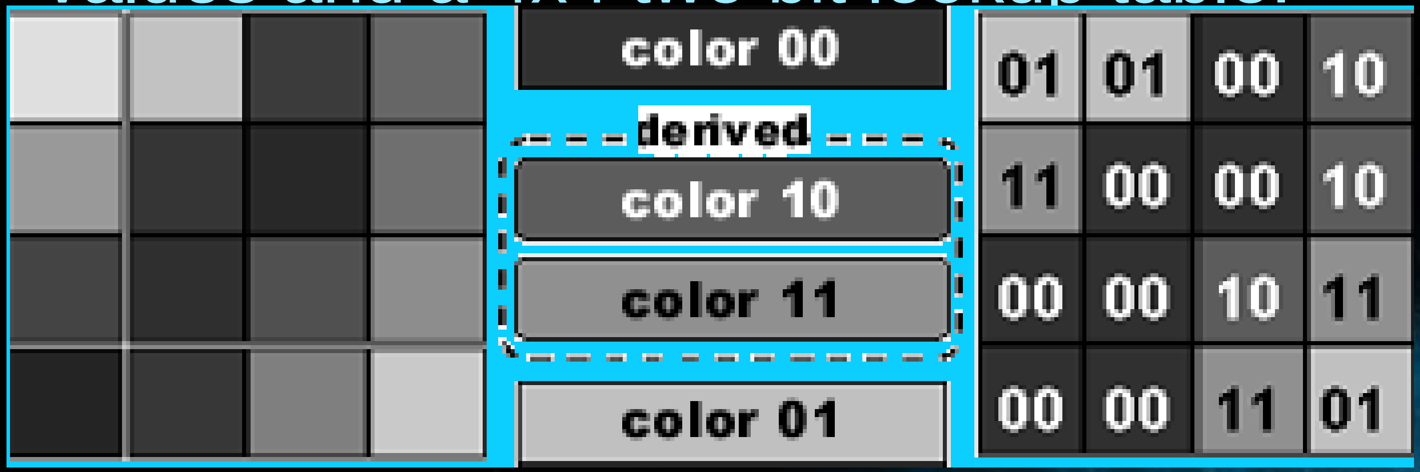
- Use 2 DXT5 for each FP16
- Color space conversion
- 12 bits (2 DXT5 alpha) luminance
- 3:1 compression ratio

Details in a separate talk by Yaohua Hu.

DXT Compression

Addon Cours

- DXT 1 :
  - DXT1 is the smallest variation of S3TC, storing 16 input pixels in 64 bits of output, consisting of two 16-bit RGB 5:6:5 color values and a 4x4 two bit lookup table.



DXT Compression

- DXT 1
  - If the first color value ($c0$) is numerically greater than the second color value ($c1$), then two other colors are calculated :

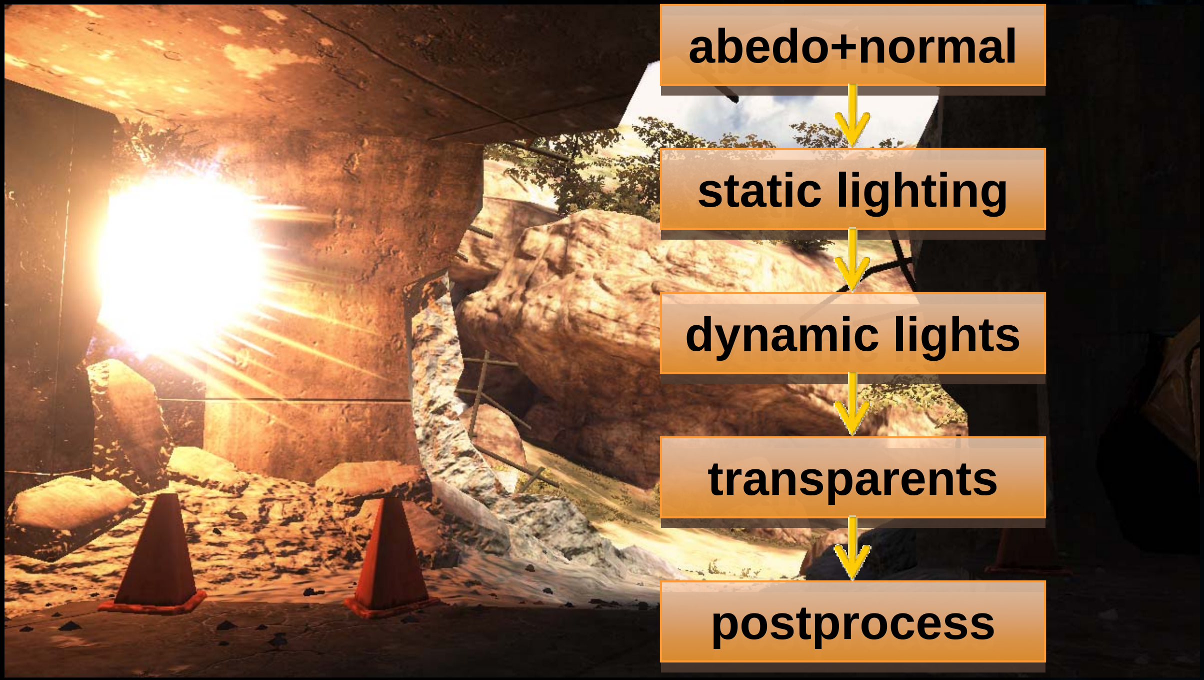
$$c_2 = \frac{2}{3}c_0 + \frac{1}{3}c_1 \qquad c_3 = \frac{1}{3}c_0 + \frac{2}{3}c_1$$

  - Otherwise, if $c_0 \leq c_1$

    then $$c_2 = \frac{1}{2}c_0 + \frac{1}{2}c_1$$

    and $c3$ is transparent.

Rendering Passes



Object Lighting

sample surface

↓

PRT

↓

VS: **PRT/Unshadowed**

→

PS: **Unshadowed * ratio**

Optimization

- given a quadratic SH vecotor, i=0,…8
  - Pull out dominant light.
  - Store SH linear + dominant light.
  - In Shader, do N dot L + shirm(sh[] – c * Y(d), N)

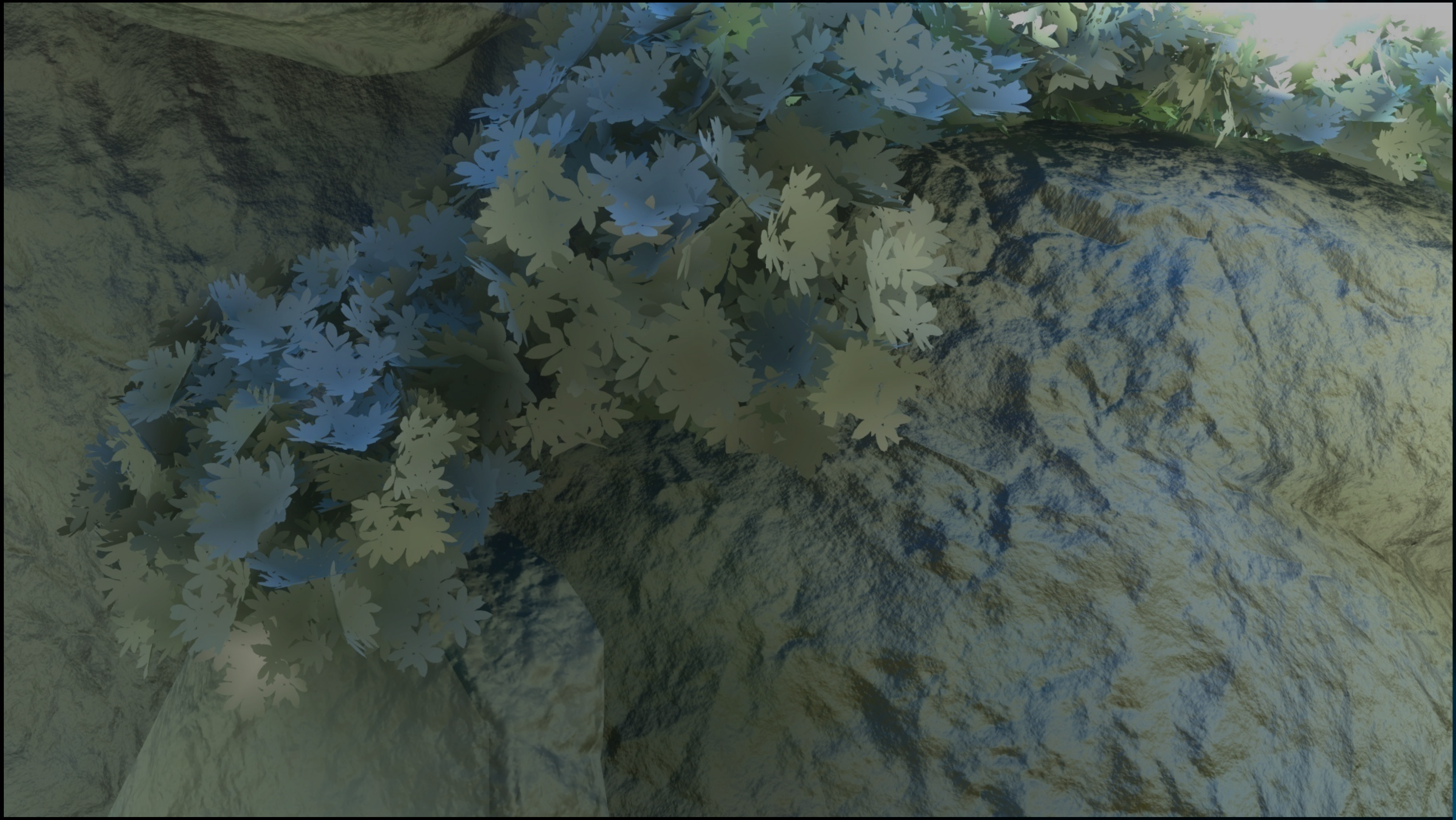
$$E = \sum_{i=0,\ldots 8} (\lambda_i - cY_i(d))^2, E^{'} = 0$$

$$c = \sum_{i=0,\ldots 8} (\lambda_i Y_i(d)) / \sum_{i=0,\ldots 8} Y_i(d)^2$$

See Peter Pike Sloan’s talk “Stupid Spherical Harmonics Tricks".

Lighting Examples

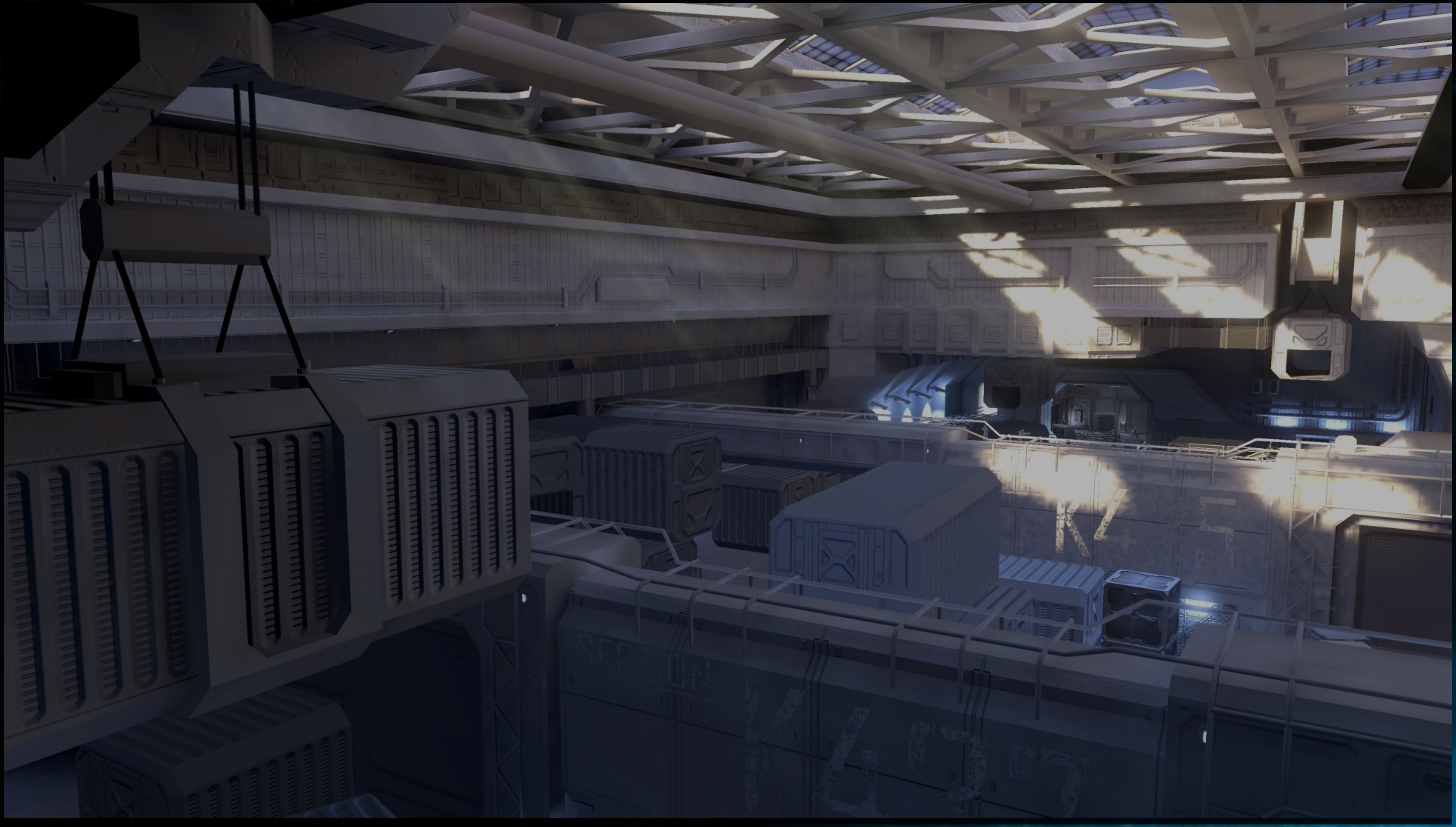
Lighting Examples



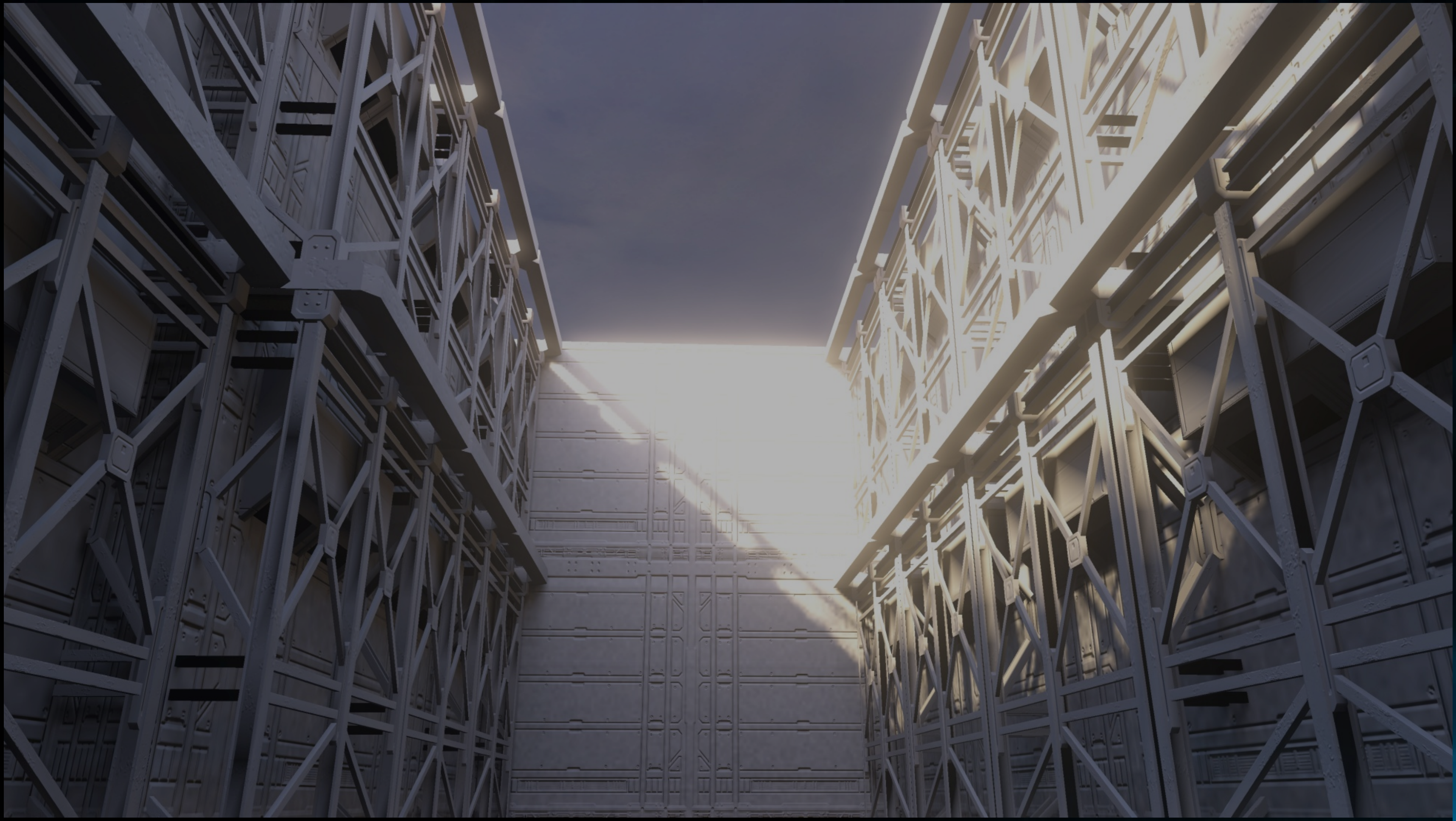
Lighting Examples

Lighting Examples

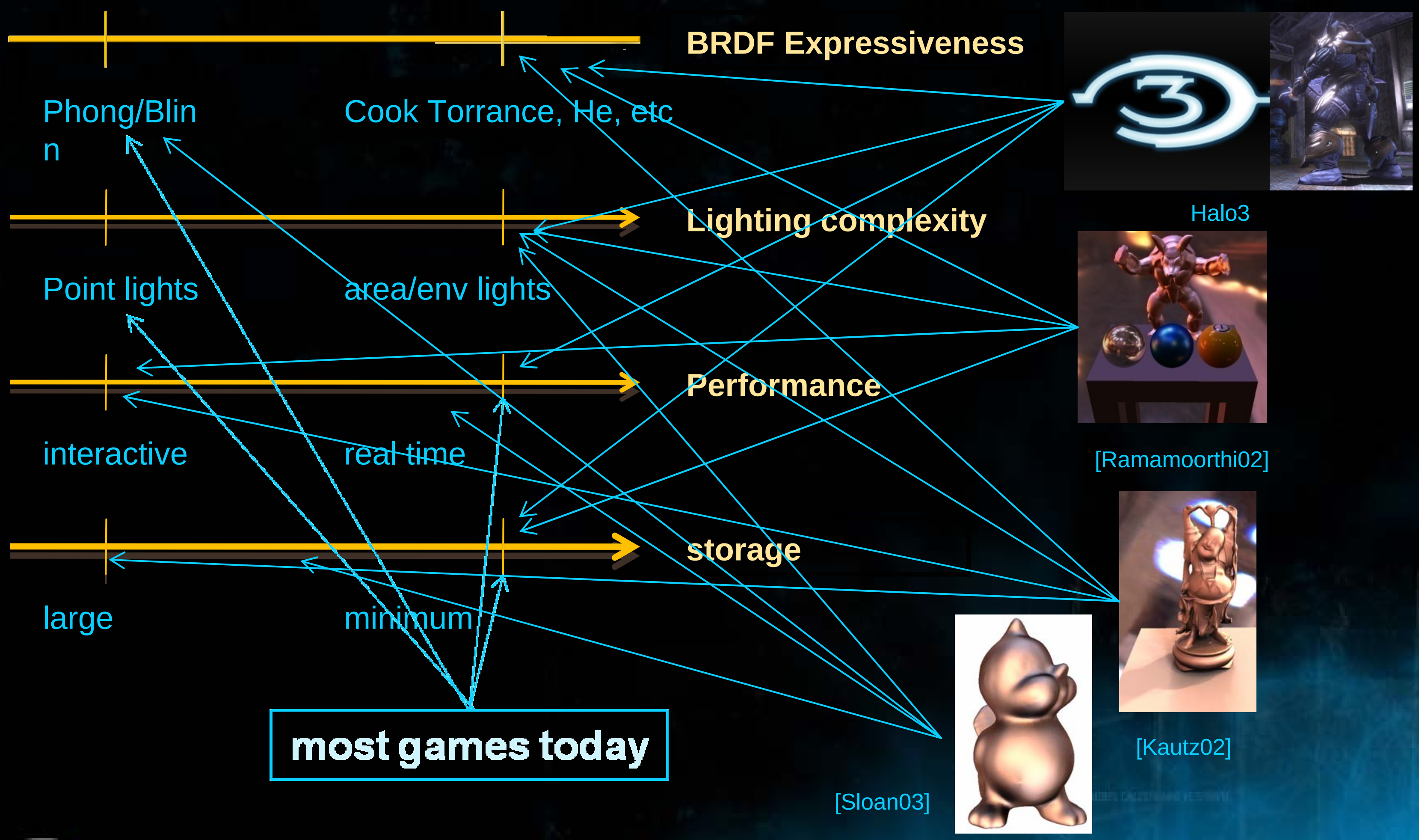
Lighting Examples



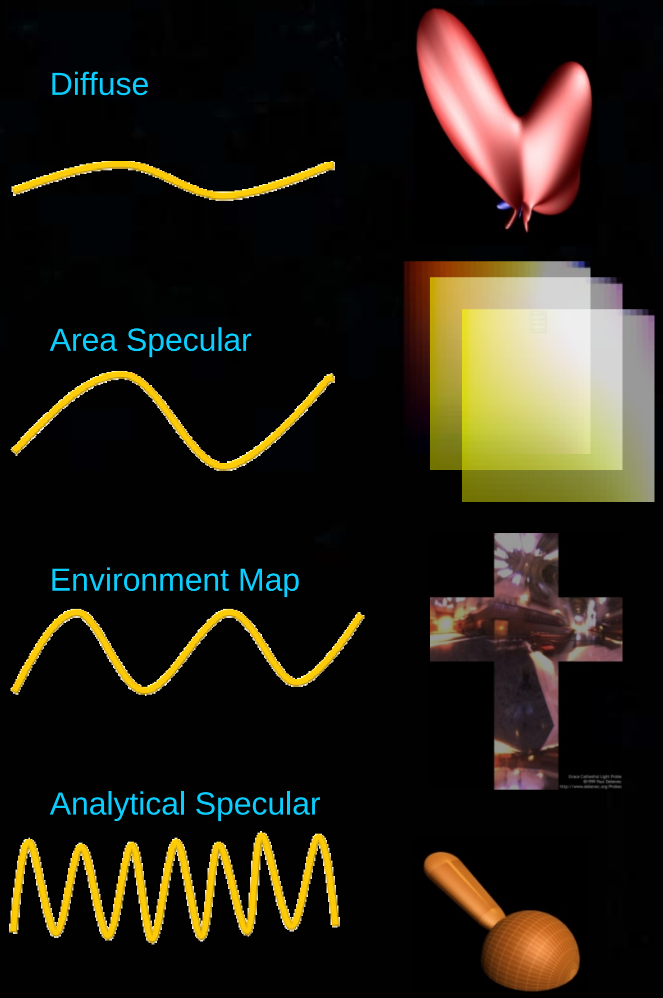
Lighting Examples



Material Motivation



Halo3 Material Model



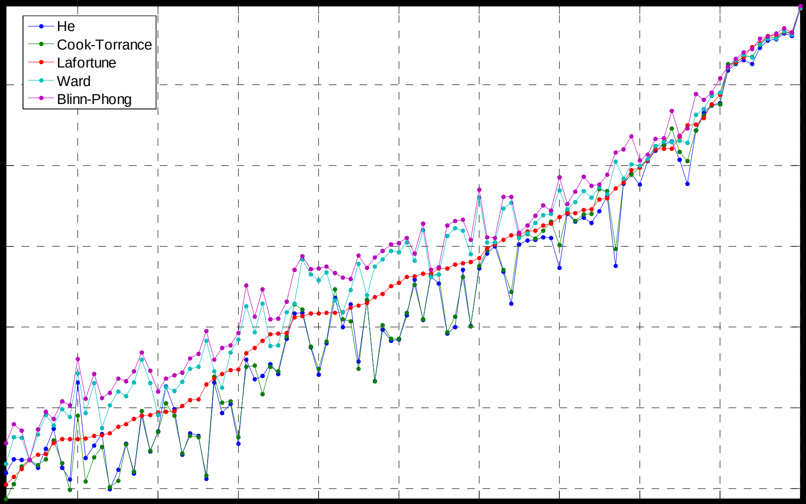
- The basic Idea
  - Separate material into diffuse parts and
  - Low, med, high freq glossy parts.
  - SH irradiance envmap for diffuse
  - New area specular model for low frequency glossy.
  - Prefiltered env map for mid frequency glossy.
  - BRDF evaluated directly with point lights for high freq.

Cook Torrance BRDF

[CookTorrance81]

$$f(V, L) = k_d R_d + k_s \cdot F \cdot R_m$$

view direction light direction diffuse & specular scale Fresnel specular lobe



[Ngan05]

$$R_m(V, L) = \frac{DG}{\pi(N \cdot L)(N \cdot V)}$$

D: microfacet distribution function

G: geomery term

Cook Torrance & Point Lights

- Option1: Evaluate directly in shader
  - E.g. [Dempski Viale 2005]
  - Somewhat expensive, not too bad.
- Option2: Store D, G, F, terms in textures.
- What about lights that are not point lights?
  - Need to integrate light from all directions.
  - Not trivial to do.

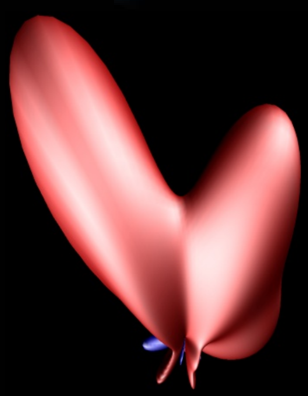
Light Integration

$$I(V) = \oint f(V,L)\cos(\theta)L(\omega)d\omega$$

$$k_d R_d \oint \cos(\theta)L(\omega)d\omega + k_s \oint F R_m(V,L)\cos(\theta)L(\omega)d\omega$$

↓

??



SH irradiance env. map

Light Integration in SH

$$I_s(V) = k_s \oint FR_m(V,L)\cos(\theta) L(\omega) d\omega$$

Project light into SH basis.

$$L(\omega) = \sum_{i=0}^{8} \lambda_i Y_i(\omega)$$

Project BRDF and cosine term in SH basis

$$B_{m,i}(V) = \oint \frac{F}{F_0} R_m(V,L)\cos(\theta) Y_i(\omega) d\omega$$

$$I_s(V) = K_s F_0 \sum_{i=0}^{8} \lambda_i B_{m,i}(V)$$

Dot product to convolve

Light Integration in SH Cont.

$$F \approx F_0 + (1 - F_0)(1 - (L \cdot H))^5$$ [Shilick85]

$$B_{m,i}(V) = \oint \frac{F_0 + (1 - F_0)(1 - (L \cdot H))^5}{F_0} R_m(V, L)\cos(\theta) Y_i(\omega) d\omega$$

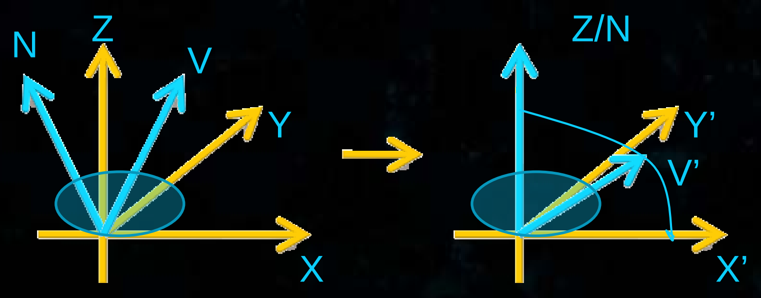
$$C_{m,i}(V) = \oint R_m(V, L)\cos(\theta) Y_i(\omega) d\omega$$

$$D_{m,i}(V) = \oint (1 - (L \cdot H)^5) R_m(V, L)\cos(\theta) Y_i(\omega) d\omega$$

$$B_{m,i}(V) = C_{m,i}(V) + \frac{1 - F_0}{F_0} D_{m,i}(V)$$

Preintegration

Pre-integration

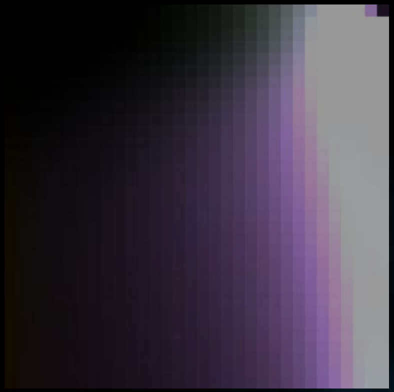


Reflective symmetry means:

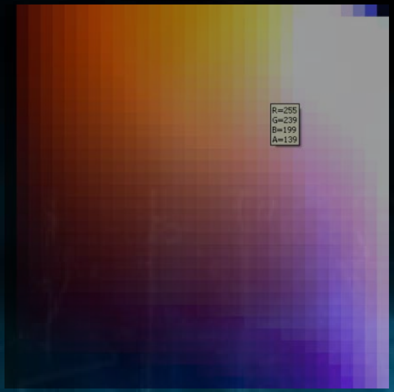
$$C_{m,i}(V) = D_{m,i}(V) = 0, i = 1,4,5.$$

Isotropic BRDF = any coordinate frame16 m values, and 8 V directions is enough.

C (i=0,2,3,6)

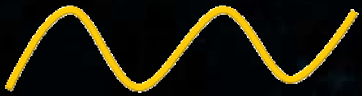


D (i=0,2,3,6)



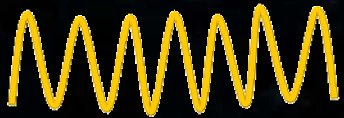
C,D (i=7,8)

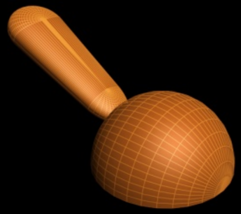
BUNGIE



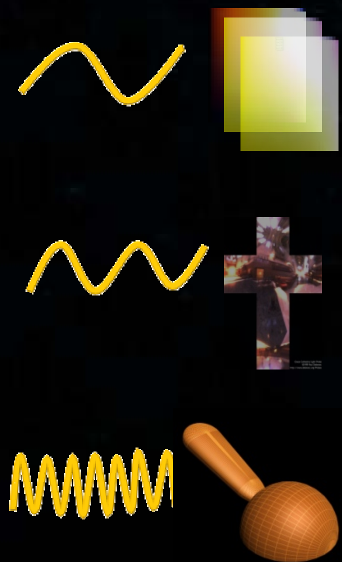
Grace Cathedral Light Probe
©1999 Paul Debevec
http://www.debevec.org/Probes

BUNGIE






TRAXUS

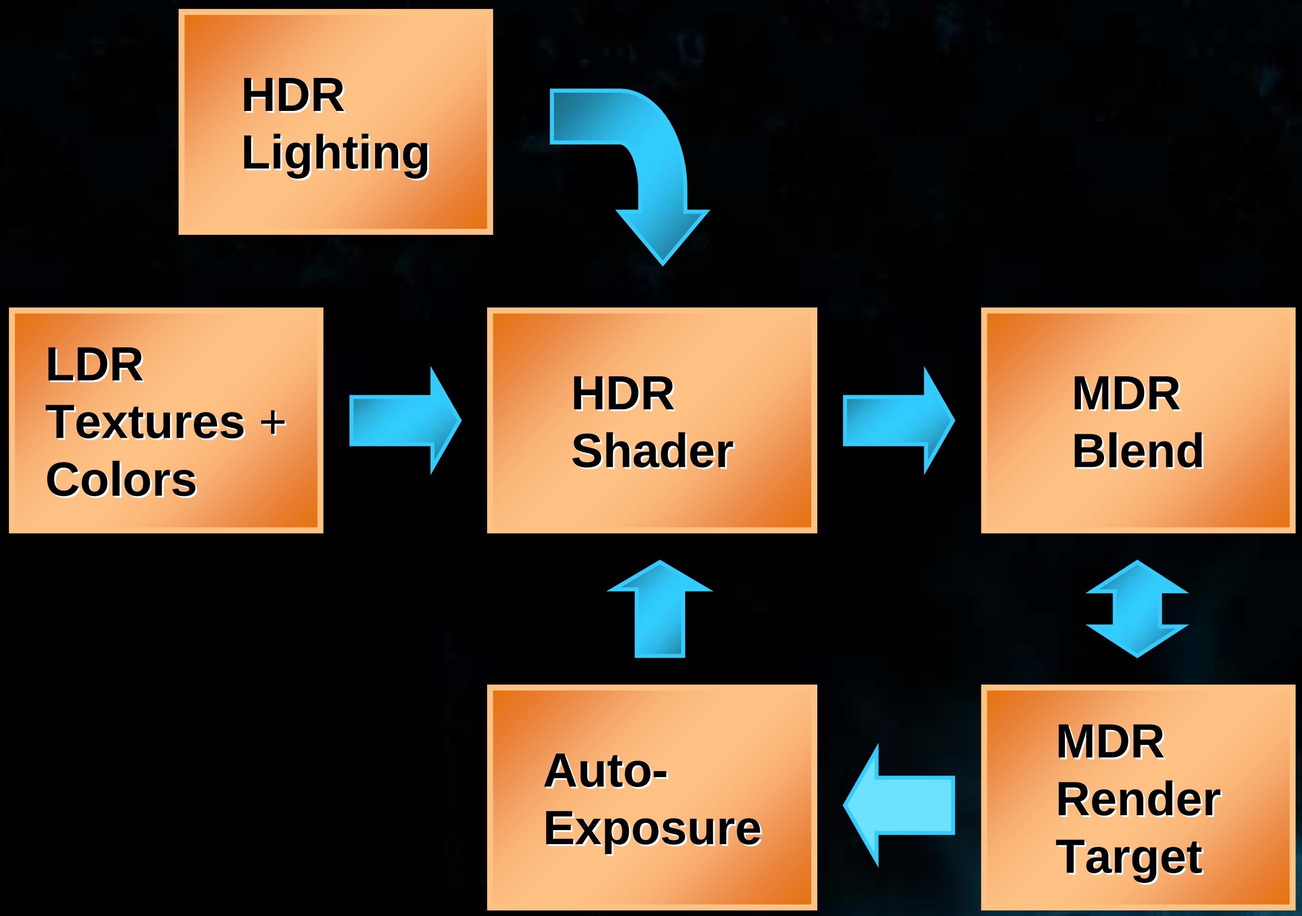


BUNGiE

HDR Rendering

HDR Rendering

Rendering Pipeline



Render Target Considerations

- Memory Size
- Render Speed
- Hardware Blend Support
- Dynamic Range
- Banding

Useable Exposure Range (Dynamic Range, Banding)

Xbox 360 Render Targets

	16f	10-bit 7e3	10-bit lin.	8-bit xRGB	16-bit lin.
Exposure Range	30 stops	3 stops	0 stops	0 stops	5 stops
Blend Support	NO ☹	YES	YES	YES	YES
Memory Size	2x	1x	1x	1x	2x
Blend/Fill Rate	50%	50-100%	50-100%	50[illegible]%	50%

	8-bit xRGB + 8-bit xRGB
Exposure Range	7 stops
Blend Support	YES
Memory Size	2x
Blend/Fill Rate	25-50%

x 2

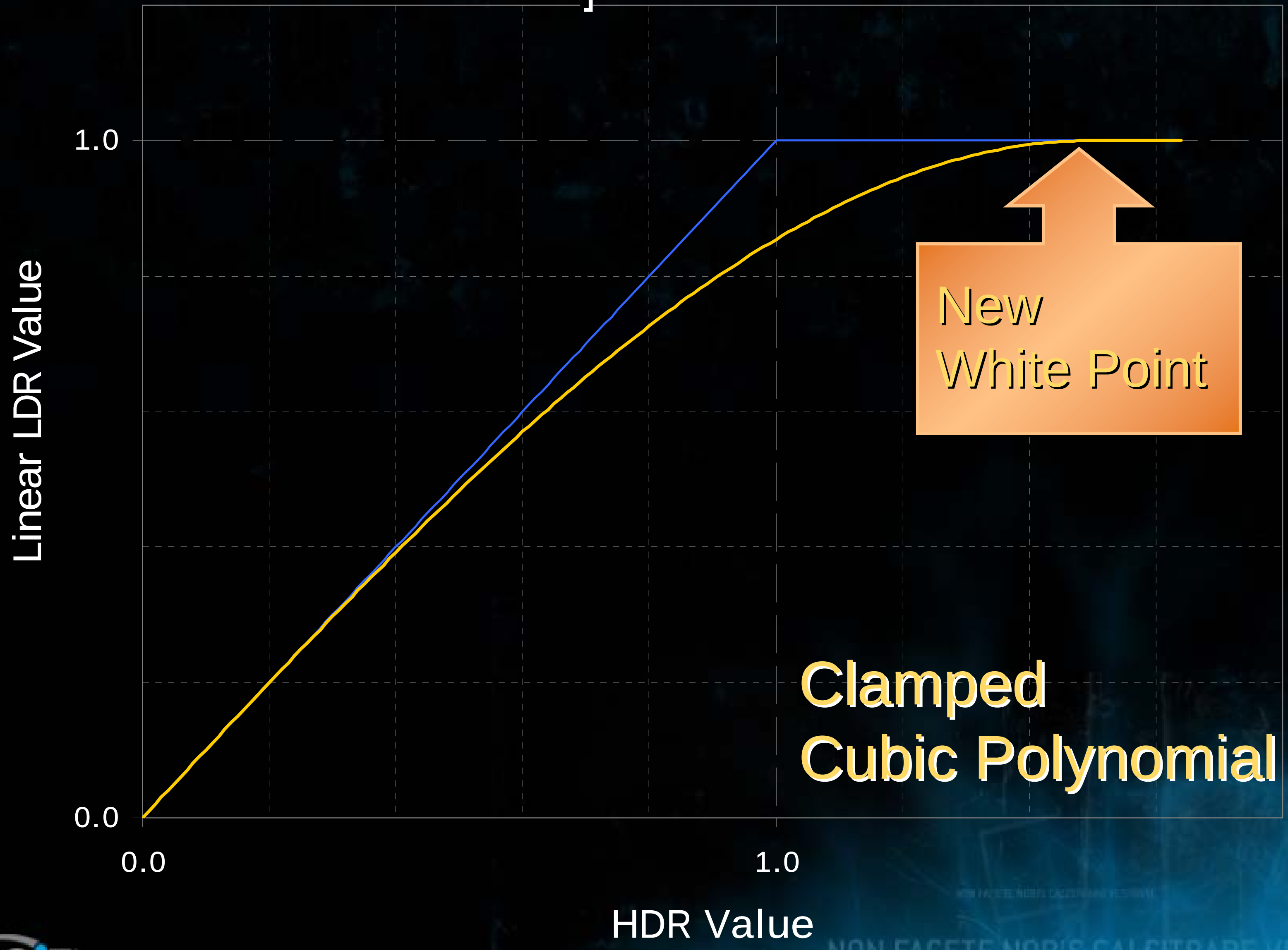

Detail
Bloom
(-7 stops)
BUNGIE


Lack of sufficient dynamic range
makes the dark areas look hazy!
Artificially increase dynamic range…
BUNGIE

Bloom Curve

Bloom Postprocessing

Final Step: Tone Curve



No Tone Curve

Tone Curve

BUNGIE

NON FACETE NOBIS CALCITRARE VESTRVM

NON FACETE NOBIS CALCITRARE VESTRVM

NON FACETE NOBIS CALCITRARE VESTRVM


NON FACETE NOBIS CALCITRARE VESTRVM
BUNGIE

Conclusion

- SH light map is a natural extension to the traditional light-mapping pipeline.
- Separating material into layers is a good approximation for all frequency reflectance.
- Area specular is critical for achieving seamless lighting and material integration.
- ALU is cheap, and will get cheaper, take maximum advantage of it.

Future Work

- Global Illumination with local, moving lights.
- GI for dynamic and semi-dynamic scenes.
- Better lighting basis (less ringing, higher frequency).
- Area specular model with complex transport.
- Measured BRDF.
- Non photo-realistic rendering.

Acknowledgement

- Authors
  - Bungie: Hao Chen, Ben Wallace, Chris Tchou, David Cook, Xi Wang, Yaohua Hu.
  - MSRA: Xinguo Liu, Zhipeng Hu, Xin Huang, Minmin Gong, Kun Zhou.
- Special Thanks
  - Bungie artists.
  - Peter Pike Sloan, Baining Guo, Harry Shum
  - Kutta Srinivasan, Matt Lee, Mikey Wetzel.

These slides will be available online
Feb. 25th, 2008
http://www.bungie.net/publications/

We're Hiring!
- Come see us at the Career Pavillion in Booth 913 (West Hall)

http://www.bungie.net/jobs/