SIGGRAPH2005

SIGGRAPH2005

# Precomputed Radiance Transfer: Theory and Practice


| Peter-Pike Sloan | Jaakko Lehtinen | Jan Kautz |
|---|---|---|
| Microsoft | Helsinki Univ. of Techn.<br>&<br>Remedy Entertainment | MIT |

# Light Representation

Jan Kautz

MIT

## Light Representation

SIGGRAPH2005

- Many ways of creating incident light
  - Synthetic environment map
  - HDR light probes (natural lighting)
  - Directly from analytical solutions
    - e.g. solution for a disk light source, angle $\tau$

There are many ways to create appropriate incident lighting.

## Overview

SIGGRAPH2005

- Light Representation
  - Analytical Solutions (SH)
    - Directional, disc, and spherical light sources
  - Environment Maps (SH/Haar)
    - Light probes
    - Rotation (SH/Haar)
    - Sample on-the-fly
  - Interpolate from precomputed incident lighting (SH)
    - Use gradient for better interpolation (SH)

In this part, we will talk about several ways to create such incident lighting.

## Analytical Solutions (SH)

SIGGRAPH2005

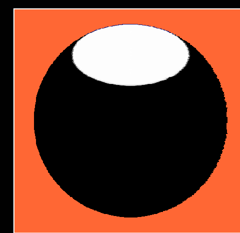
- Directional light source
  - Coming from a single direction $\Psi$
  - Incident illumination is basically a Dirac peak
- Integration against basis functions boils down to evaluating basis functions in that direction: $l_i = y_i(\Psi)$
- Good for e.g. sun light

The easiest way to create a light source is to integrate the basis functions against a directional light source (basically a Dirac).

## Analytical Solutions (SH)

- Disc sources (center around $z$):

$$d_\tau(\theta,\varphi)=\begin{cases}1 & (\tau-\theta)>0\\ 0 & otherwise\end{cases}$$



- Integrate symbolically in Maple:

$$l_i=\int_{\varphi=0}^{2\pi}\int_{\theta=0}^{\tau} y_i(\theta,\varphi)\sin\theta\, d\theta\, d\varphi$$

A disc-shaped area source can be analytically projected into SH.

It is preferred to center the disc around the z-axis. In that case, only very few coefficients are non-zero. Here are the non-zero coefficients for the first 5 bands:

*l*(l=0,m=0) = -1.772453851*cos(tau)+1.772453851

*l*(l=1,m=0) = 1.534990062*sin(tau)^2

*l*(l=2,m=0) = -1.981663648*cos(tau)^3+1.981663648*cos(tau)

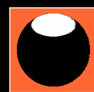
*l*(l=3,m=0) = -2.930920064*cos(tau)^4+3.517104076*cos(tau)^2-.5861840127

*l*(l=4,m=0) = -4.652691359*cos(tau)^5+6.646701941*cos(tau)^3-1.994010582*cos(tau)

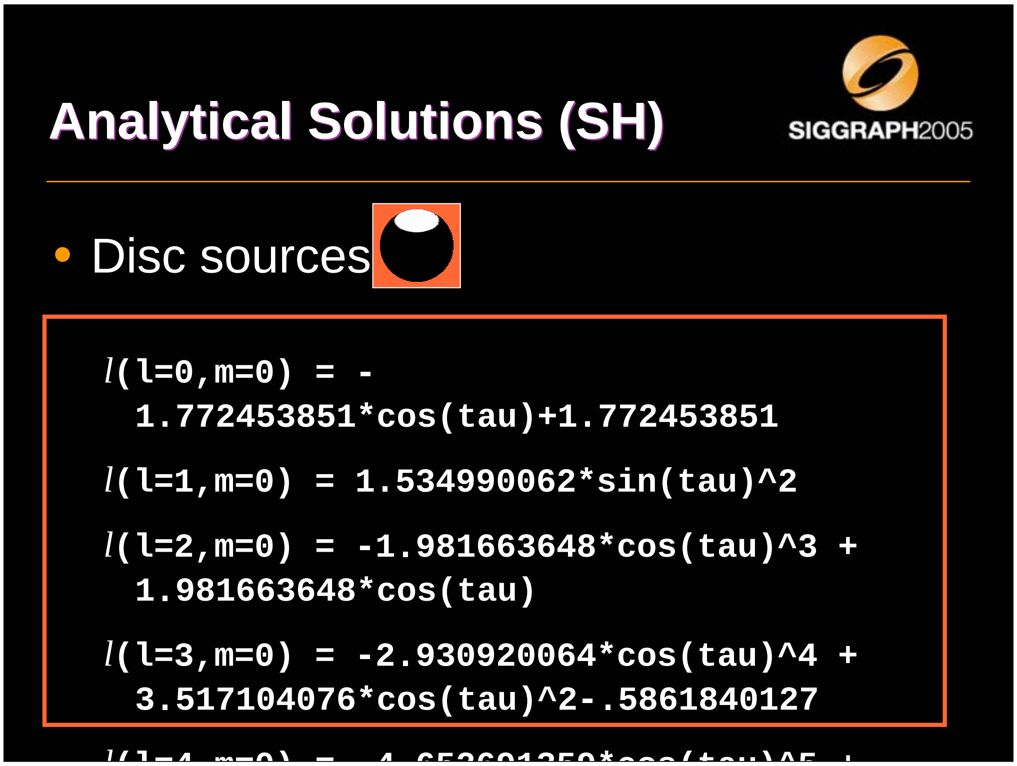
The light source can be rotated around using the SH rotation matrices (see later slides).

## Analytical Solutions (SH)

SIGGRAPH2005

- Disc sources: 
  - Due to SH construction any function that is rotationally symmetric around z only has non-zero coefficients for modes $m=0$ (Zonal Harmonics)
  - Efficient representation and also rotation
  - Use SH rotation matrix to move light

It's interesting to note, that the lighting vector will only have very few coefficients in that case (only for modes m=0), i.e. for the first 5 bands, only 5 out of the 25 coefficients will be non-zero. This also makes the rotation (see later slides) more efficient.

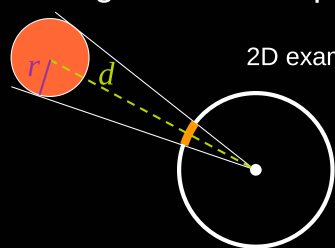



These are the formulas for a disc source.

## Analytical Solutions (SH)

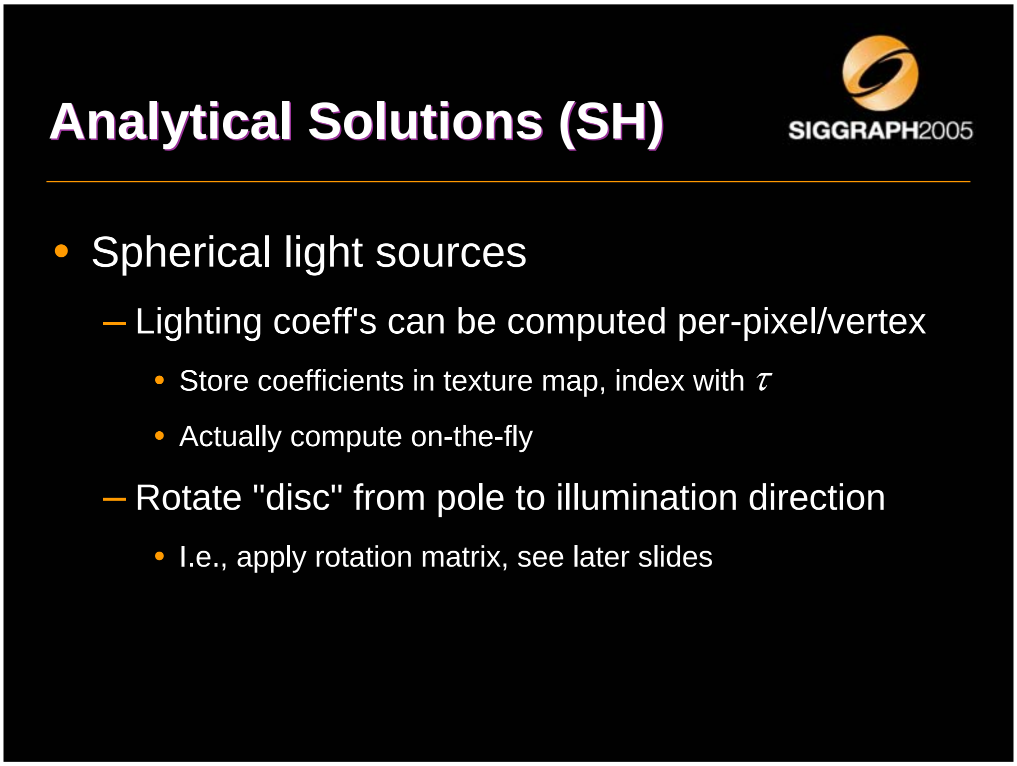
SIGGRAPH2005

- Spherical light sources
  - Spherical light sources appear as disc sources

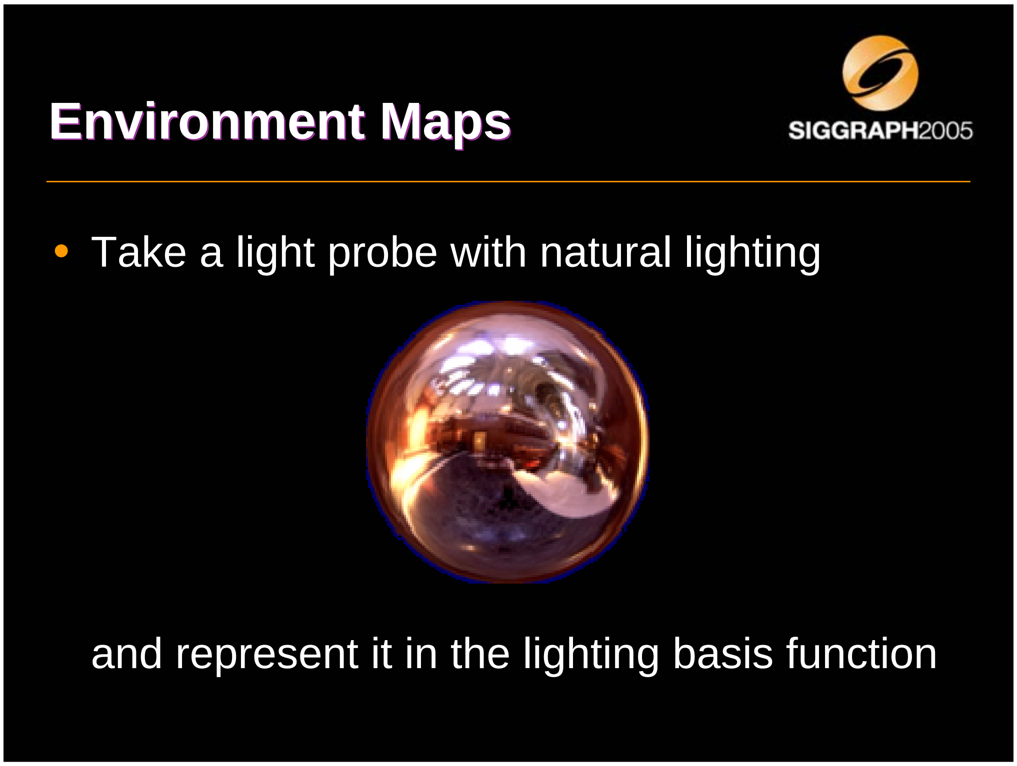



  - Distance of spherical light to center $d$ and sphere radius $r$ determines size of cap: $\sin\tau = r/d$

A special case are spherical light sources. They correspond to disc-like sources, but their size changes with the distance to the sample point, where the light is sampled. This sampling can be actually performed on-the-fly.




Since the formulas are very simple, the lighting coefficients can actually be computed on the fly.




(High-dynamic range) environment maps are a simple way of "creating" incident lighting.

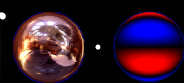
## Environment Maps

SIGGRAPH2005

- Spherical Harmonics
  - Integrate basis functions against fixed HDR map

$$l_i = \int L_{env}(\vec{s})\, y_i(\vec{s})\, d\vec{s}$$

$$l_i = \int$$



  - 1) Monte-Carlo integration
  - 2) Precompute maps in same space (e.g. cube map) that contain the basis functions $\Rightarrow$ big dot-product for each coeff.

This is a standard scenario. A HDR environment map is given and projected into basis functions.

If the environment map is given as a cube map, it is necessary to know the solid angle of a texel in the cube map. It is: 4/( (X^2 + Y^2 + Z^2)^(3/2) ), where [X Y Z] is the vector to the texel (not normalized).

## Environment Maps

SIGGRAPH2005

- Haar Wavelets
  - Represent environment maps as cube maps
  - Do standard wavelet transform on each face
  - Compress/quantize data (select most significant coefficients)

Reference 4096 coeffs. 100 coeffs.

## Rotating Lighting / Env.Maps

SIGGRAPH2005

- Rotation of lighting – Spherical Harmonics
  - There is a simple linear matrix that will rotate SH coefficients directly:

$$\mathbf{l}' = \mathbf{R}_{SH}\,\mathbf{l}$$

  - The matrix is block diagonal
    - No spill across bands
  - For the first few SH bands, it can be computed explicitly
  - General case: recurrence formula

$$R_{SH} = \begin{bmatrix} \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\ 0 & X & X & X & 0 & 0 & 0 & 0 & 0 & \cdots \\ 0 & X & X & X & 0 & 0 & 0 & 0 & 0 & \cdots \\ 0 & X & X & X & 0 & 0 & 0 & 0 & 0 & \cdots \\ 0 & 0 & 0 & 0 & X & X & X & X & X & \cdots \\ 0 & 0 & 0 & 0 & X & X & X & X & X & \cdots \\ 0 & 0 & 0 & 0 & X & X & X & X & X & \cdots \\ 0 & 0 & 0 & 0 & X & X & X & X & X & \cdots \\ 0 & 0 & 0 & 0 & X & X & X & X & X & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

Spherical Harmonics are a steerable basis. It means that no information is lost under rotation.

There are various ways of creating this rotation matrix. There are recurrence formulas, but in case only a few bands are needed, the matrix can be created explicitly.

## Rotating Lighting / Env.Maps

SIGGRAPH2005

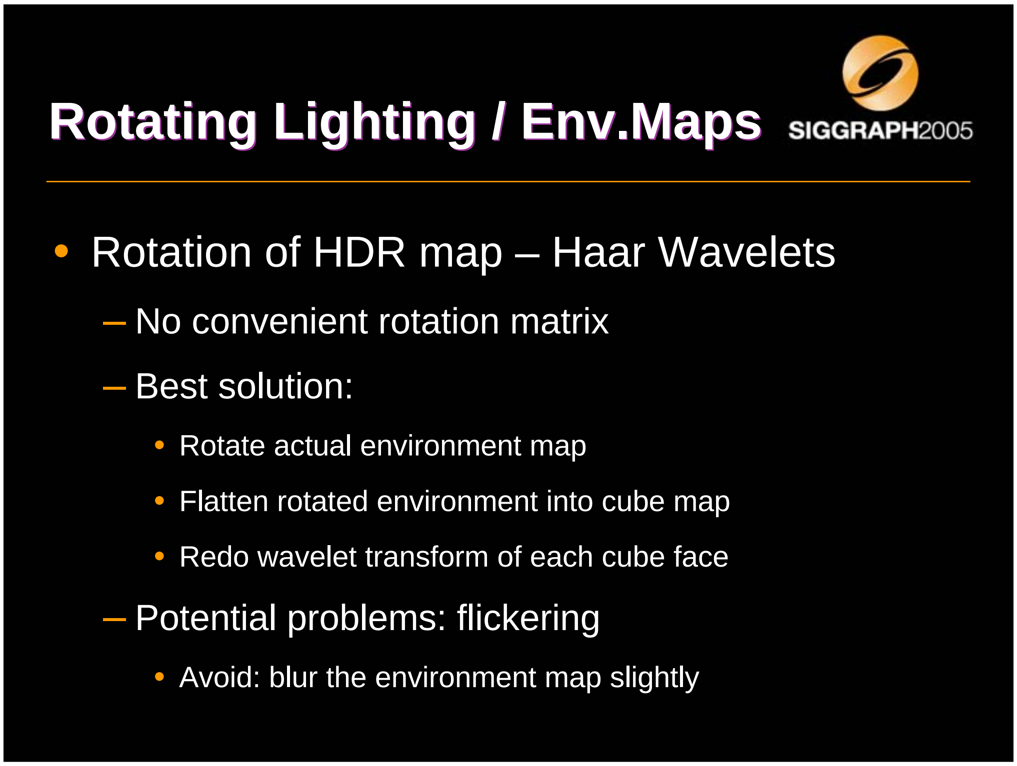
- Special case: disc sources
  - Remember: only $m$=0 coeff's are non-zero

$$l_i^R = R_{SH} \cdot l_i = \begin{bmatrix} \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\ 0 & \mathbf{X} & \mathbf{a} & \mathbf{X} & 0 & 0 & 0 & 0 & 0 & \cdots \\ 0 & \mathbf{X} & \mathbf{b} & \mathbf{X} & 0 & 0 & 0 & 0 & 0 & \cdots \\ 0 & \mathbf{X} & \mathbf{c} & \mathbf{X} & 0 & 0 & 0 & 0 & 0 & \cdots \\ 0 & 0 & 0 & 0 & \mathbf{X} & \mathbf{X} & \mathbf{d} & \mathbf{X} & \mathbf{X} & \cdots \\ 0 & 0 & 0 & 0 & \mathbf{X} & \mathbf{X} & \mathbf{e} & \mathbf{X} & \mathbf{X} & \cdots \\ 0 & 0 & 0 & 0 & \mathbf{X} & \mathbf{X} & \mathbf{f} & \mathbf{X} & \mathbf{X} & \cdots \\ 0 & 0 & 0 & 0 & \mathbf{X} & \mathbf{X} & \mathbf{g} & \mathbf{X} & \mathbf{X} & \cdots \\ 0 & 0 & 0 & 0 & \mathbf{X} & \mathbf{X} & \mathbf{h} & \mathbf{X} & \mathbf{X} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \cdot \begin{bmatrix} \mathbf{C} \\ 0 \\ \mathbf{C} \\ 0 \\ 0 \\ 0 \\ \mathbf{C} \\ 0 \\ 0 \\ \vdots \end{bmatrix} = \begin{bmatrix} \mathbf{C} \\ \mathbf{a}\cdot\mathbf{C} \\ \mathbf{b}\cdot\mathbf{C} \\ \mathbf{c}\cdot\mathbf{C} \\ \mathbf{d}\cdot\mathbf{C} \\ \mathbf{e}\cdot\mathbf{C} \\ \mathbf{f}\cdot\mathbf{C} \\ \mathbf{g}\cdot\mathbf{C} \\ \mathbf{h}\cdot\mathbf{C} \\ \vdots \end{bmatrix}$$

$\Rightarrow$ becomes **dot product**

see [Sloan05]

In case of z-rotationally symmetric sources (represented solely with Zonal Harmonics, see [Sloan05]), the rotation matrix becomes a dot-product.




In case of Haar Wavelets, there is no convenient rotation matrix. All one can do is to rotate in image space and then reproject.

## Varying Lighting

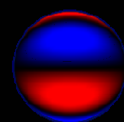
SIGGRAPH2005

- Up to now: lighting didn't really change
- What if character moves within scene?
  - Expect lighting on character to change
- Two solutions:
  - Sample lighting on-the-fly
  - Precompute for static scenes

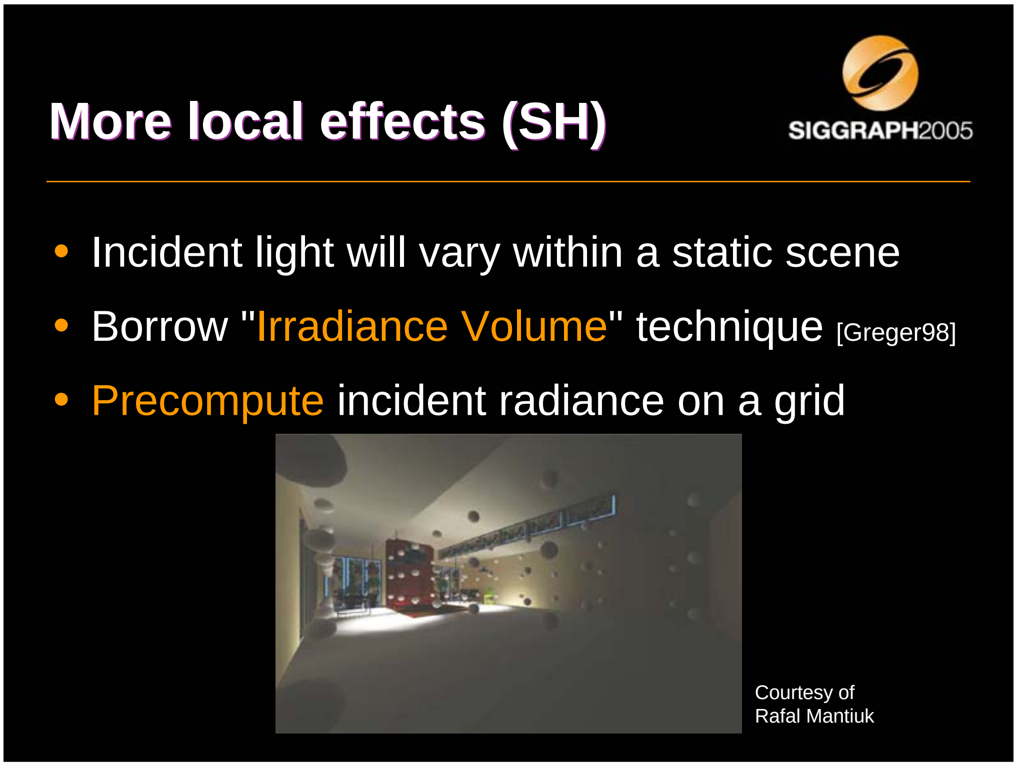
## Environment Maps – Sample Incident Lighting

SIGGRAPH2005

- Sample lighting:
  - Render scene from center of object ***p***
  - 6 times: for each cube face
  - Compute lighting coefficients (SH/Haar):

$$l_i^p = \int$$  $\cdot$ 

  - No need to rotate lighting then




Sofar, we have assumed that lighting is infinitely far away, but that doesn't go well with interior scenes.

In that case, we can sample the incident radiance at grid points throughout the scene (assuming the scene doesn't change) and store that in SH.

## More local effects (SH)

SIGGRAPH2005

- Incident light will vary within a static scene
- Borrow "Irradiance Volume" technique [Greger98]
- Precompute incident radiance on a grid
  - I.e., for each sample point $p_k$ precompute a light vector $\mathbf{l}_{p_k}$ (assumes static lighting)
  - For new points $q$, linearly interpolate light from nearby sample points $p_k$ [Mantiuk02]
- E.g., this technique is used in Max Payne 2

At run-time, one just queries the closest neighbors and interpolates the lighting coefficient vectors.

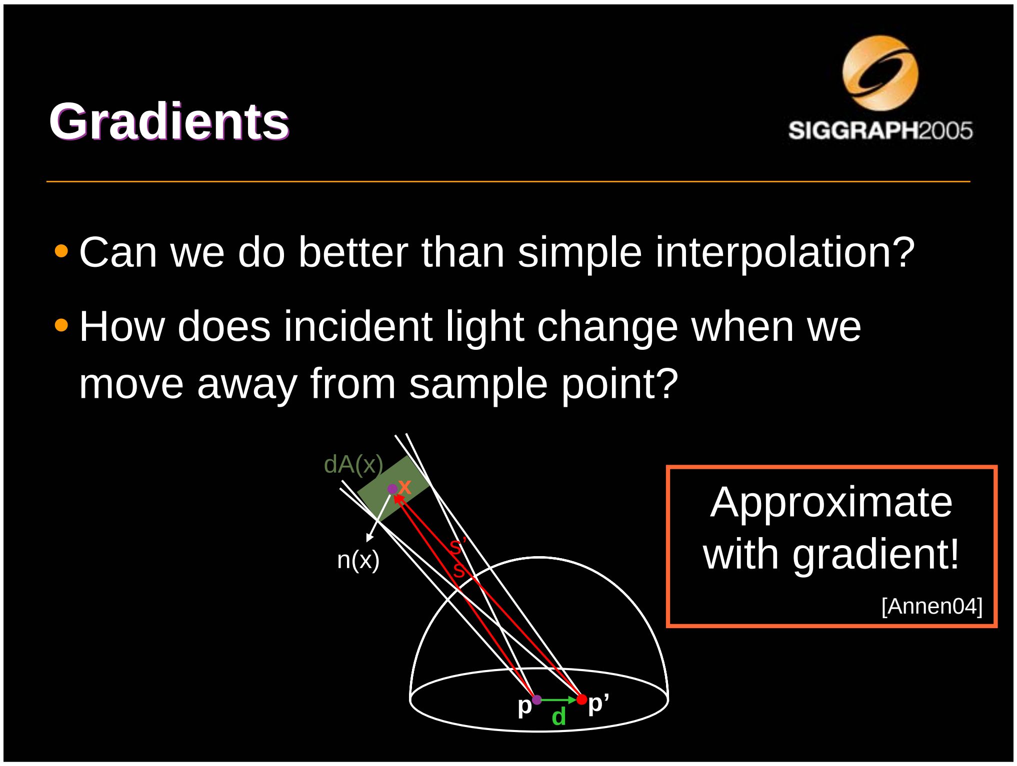
## Radiance Volumes

SIGGRAPH2005

- Note how bunny is illuminated differently

Courtesy of
Rafal Mantiuk

This gives a nice sense of an object being "in the scene". Even just the first 2 SH bands (4 coefficients) gives a great sense.




Instead of just linear interpolation, one can use a gradient in addition.

## Gradients

- Lighting is projected into SH as follows:

$$l_i^p = \int_\Omega L_S(p \leftarrow \vec{s}) y_i(\vec{s}) d\vec{s}$$

- We want to compute the gradient when we move $\boldsymbol{d}$ away from $p$
- And reuse at run-time:

$$l_i^{p+\boldsymbol{d}} = l_i^p + \left(\nabla l_i^p \cdot \boldsymbol{d}\right)$$

## Analytical Gradient

SIGGRAPH2005

- Rewrite integral as integral over surfaces (because of implicit dependency on $p$)
- Take gradient: (analytical integrand)

$$\nabla l_i^p = -\int_S \nabla \left( y_i\left(\frac{\vec{\sigma}}{\|\vec{\sigma}\|}\right) \frac{\vec{n}(\underline{x}) \cdot \vec{\sigma}}{\|\vec{\sigma}\|^3} \right) L_S(p \leftarrow \underline{x})\, dA$$

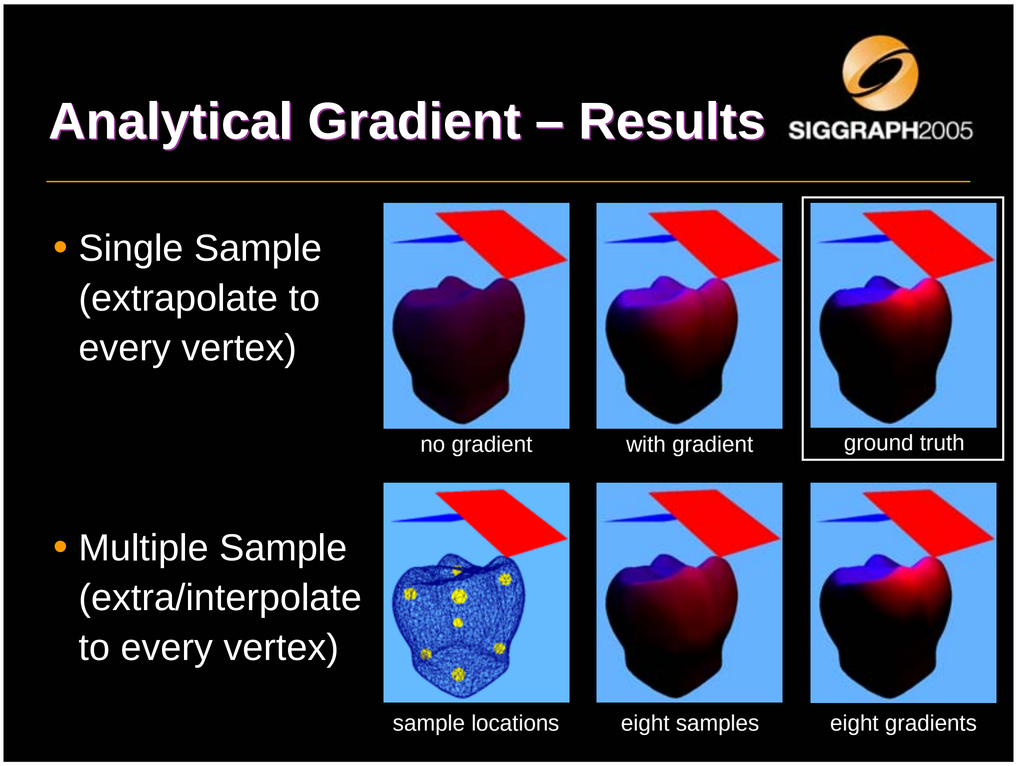
(Code is available from the authors)

$\vec{n} = \text{normal}$
$\vec{\sigma} = \underline{x} - p$

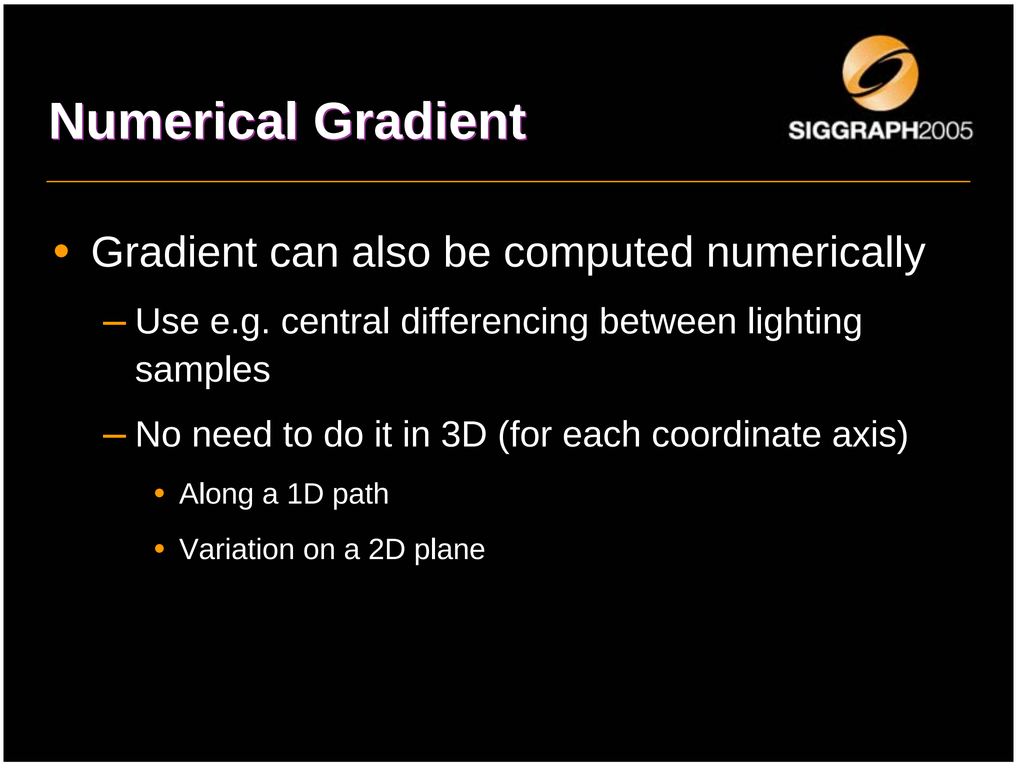
The actual derivation is slightly involved and omitted here. It is available in the paper [Annen04]. Code is available from the authors.

## Analytical Gradient – Putting it all together

- Compute incident lighting
  - Store distance and normal as well
- Compute coefficients $l_i^p = \int \;*\;$
  - Per-pixel integration with SH basis
- Compute gradient $\nabla l_i^p = \int \left( (\nabla\;) * \; + \; * (\nabla\;) \right)$
  - Per-pixel integration of simple analytical formula
  - Requires distance & normal
- For rendering:
  - Extrapolate to coefficients $l_i^{p+\boldsymbol{d}}$




The gradient helps to give a much more "local feel" of the light sources.




Numerical computation of the gradient is viable as well.

## Numerical Gradient

SIGGRAPH2005

- Gradient can also be computed numerically

From ATI – Ruby Demo

SIGGRAPH2005

# *Demos*

Given by Natalya Tatarchuk from ATI

## Conclusion

SIGGRAPH2005

- Lighting can be more than a static cube map
  - Analytical models
  - Sample on the fly
  - Precompute (interpolate)
  - Or even extrapolate using gradients

SIGGRAPH2005