### To Do
- Assignment 3 milestone due Jun 3
  - 1-2 page PDF or website
  - What you have done so far (at least one image)
  - 1-2 para proposal of what you hope to accomplish
  - We may say ok or schedule time to meet, discuss
  - Talk to us if any difficulty finding project
    - Assignment gives some well specified, loose, other options; you can do something else too

### Motivation
- Next week: Image-Based Rendering. Use measured data (real photographs) and interpolate for realistic real-time
- Why not apply to real-time rendering?
  - Precompute (offline) some information (images) of interest
  - Must assume something about scene is constant to do so
  - Thereafter real-time rendering. Often accelerate hardware
- Easier and harder than conventional IBR
  - Easier because synthetic scenes give info re geometry, reflectance (but CG rendering often longer than nature)
  - Harder because of more complex effects (lighting from all directions for instance, not just changing view)
- Representations and Signal-Processing crucial

### My General Philosophy
- This general line of work is a large data management and signal-processing problem
- Precompute high-dimensional complex data
- Store efficiently (find right mathematical represent.)
- Render in real-time
  - Worry about systems issues like caching
  - Good signal-processing: use only small amount of data but guarantee high fidelity
- Many insights into structure of lighting, BRDFs, …
  - Not just blind interpolation; signal processing

### Precomputation-Based Relighting
- Analyze precomputed images of scene

### Precomputation-Based Relighting
- Analyze precomputed images of scene
Assumptions

- Static geometry
- Precomputation
- Real-Time Rendering (relight all-frequency effects)
- Exploit linearity of light transport for this
- Later, change viewpoint as well

Why is This Hard?

- Plain graphics hardware supports only simple (point) lights, BRDFs (Phong) without any shadows
- Shadow maps can handle point lights (hard shadows)
- Environment maps complex lighting, BRDFs but no shadows
- IBR can often do changing view, fixed lighting

- How to do complex shadows in complex lighting?
- With dynamically changing illumination and view?

Relighting as a Matrix-Vector Multiply

\[
P = \begin{bmatrix}
    \mathbf{T}_{11} & \mathbf{T}_{12} & \ldots & \mathbf{T}_{1M} \\
    \mathbf{T}_{2} & \mathbf{T}_{22} & \ldots & \mathbf{T}_{2M} \\
    \vdots & \vdots & \ddots & \vdots \\
    \mathbf{T}_{N1} & \mathbf{T}_{N2} & \ldots & \mathbf{T}_{NM}
\end{bmatrix}
\begin{bmatrix}
    L_{1} \\
    L_{2} \\
    \vdots \\
    L_{N}
\end{bmatrix}
\]

Matrix Columns (Images)

\[
\mathbf{T}_{ij} = \cos(\theta_{ij})
\]

Why is This Hard?

Plain graphics hardware supports only simple (point) lights, BRDFs (Phong) without any shadows

- Shadow maps can handle point lights (hard shadows)
- Environment maps complex lighting, BRDFs but no shadows
- IBR can often do changing view, fixed lighting

- How to do complex shadows in complex lighting?
- With dynamically changing illumination and view?
Precompute 2: Rasterize Matrix Rows

\[
\begin{bmatrix}
T_{11} & T_{12} & \cdots & T_{1M} \\
T_{21} & T_{22} & \cdots & T_{2M} \\
\vdots & \vdots & \ddots & \vdots \\
T_{N1} & T_{N2} & \cdots & T_{NM}
\end{bmatrix}
\]

Problem Definition

Matrix is Enormous
- 512 x 512 pixel images
- 6 x 64 x 64 cubemap environments

Full matrix-vector multiplication is intractable
- On the order of $10^{10}$ operations per frame

How to relight quickly?

Outline

- Motivation and Background
- Compression methods
  - Low frequency linear spherical harmonic approximation
  - Factorization and PCA
  - Local factorization and clustered PCA
  - Non-linear wavelet approximation
- Changing view as well as lighting
  - Clustered PCA
  - Factored BRDFs
  - Triple Product Integrals
- Precomputed Radiance Transfer
  - Better light integration and transport
    - dynamic, area lights
    - self-shadowing
    - interreflections
  - For diffuse and glossy surfaces
  - At real-time rates
  - Sloan et al. 02 (most cited rendering paper in last 15 years—1000, widely used in games, movie production; Spherical Harmonic Lighting)

Precomputation: Spherical Harmonics

Diffuse Transfer Results
### Idea of Compression

- The vector is projected onto low-frequency components (say 25). Size greatly reduced.
- Hence, only 25 matrix columns
- But each pixel still treated separately (still have 300000 matrix rows for 512 x 512 image)
- Actually, for each pixel, dot product of matrix row (25 elems) and lighting vector (25 elems) in hardware
- Good technique (common in games, movies) but useful only for broad low-frequency lighting

### Outline

- Motivation and Background
- **Compression methods**
  - Low frequency linear spherical harmonic approximation
  - Factorization and PCA
  - Local factorization and clustered PCA
  - Non-linear wavelet approximation
- Changing view as well as lighting
  - Clustered PCA
  - Factored BRDFs
  - Triple Product Integrals

### PCA or SVD factorization

- SVD:
  - $\mathbf{P}$
  - $\mathbf{S}$
  - $\mathbf{C}$
  - diagonal matrix (singular values)

- Applying Rank $b$:
  - $\mathbf{P}$
  - $\mathbf{S}_b$
  - $\mathbf{C}_b$

- Absorbing $\mathbf{S}_b$ values into $\mathbf{C}_b$:
  - $\mathbf{P}$
  - $\mathbf{S}_b$
  - $\mathbf{C}_b$
  - $\mathbf{C}_b$ smaller

### Idea of Compression

- Represent matrix (rather than light vector) compactly
- Can be (and is) combined with low frequency vector
- Useful in broad contexts.
  - BRDF factorization for real-time rendering (reduce 4D BRDF to 2D texture maps) McCool et al. 01 etc
  - Surface Light field factorization for real-time rendering (4D to 2D maps) Chen et al. 02, Nishino et al. 01
  - Factorization of Orientation Light field for complex lighting and BRDFs (4D to 2D) Latta et al. 02

- Not too useful for general precomput. relighting
  - Transport matrix not low-dimensional!
Local or Clustered PCA

- Exploit local coherence (in say 16x16 pixel blocks)
  - Idea: light transport is locally low-dimensional. Why?
  - Even though globally complex
  - See Mahajan et al. 07 for theoretical analysis
- Original idea: Each triangle separately
  - Example: Surface Light Fields 3D subspace works well
  - Vague analysis of size of triangles
  - Instead of triangle, 16x16 image blocks [Nayar et al. 04]
- Clustered PCA [Sloan et al., 2003]
  - Combines two widely used compression techniques: Vector Quantization or VQ and Principal Component Analysis
  - For complex geometry, no need for parameterization / topology

Outline

- Motivation and Background
  - Compression methods
    - Low frequency linear spherical harmonic approximation
    - Factorization and PCA
    - Local factorization and clustered PCA
    - Non-linear wavelet approximation
  - Changing view as well as lighting
    - Clustered PCA
    - Factored BRDFs
    - Triple Product Integrals

Sparse Matrix-Vector Multiplication

Choose data representations with mostly zeroes

Vector: Use non-linear wavelet approximation on lighting

Matrix: Wavelet-encode transport rows

Non-linear Wavelet Approximation

Wavelets provide dual space / frequency locality
- Large wavelets capture low frequency area lighting
- Small wavelets capture high frequency compact features

Non-linear Approximation
- Use a dynamic set of approximating functions (depends on each frame’s lighting)
- By contrast, linear approx. uses fixed set of basis functions (like 25 lowest frequency spherical harmonics)
- We choose 10’s - 100’s from a basis of 24,576 wavelets
Non-linear Wavelet Light Approximation

Wavelet Transform

Error in Lighting: St Peter's Basilica

Output Image Comparison

Non-linear Wavelet Light Approximation

Retain 0.1% – 1% terms

Outline

- Motivation and Background
- Compression methods
  - Low frequency linear spherical harmonic approximation
  - Factorization and PCA
  - Local factorization and clustered PCA
  - Non-linear wavelet approximation
- Changing view as well as lighting
  - Clustered PCA
  - Factored BRDFs
  - Triple Product Integrals
Changing Only The View

Problem Characterization

6D Precomputation Space
- Distant Lighting (2D)
- View (2D)
- Rigid Geometry (2D)

With ~ 100 samples per dimension
≈ 10^12 samples total!!: Intractable computation, rendering

Clustered PCA

- Use low-frequency light and view variation (Order 4 spherical harmonic = 25 for both; total = 25^2 = 625)
- 625 element vector for each vertex
- Apply CPCA directly (Sloan et al. 2003)
- Does not easily scale to high frequencies
  - Really cubic complexity (number of vertices, illumination directions or harmonics, and view directions or harmonics)
  - Practical real-time method on GPU

Factored BRDFs

- Sloan et al. 04, Wang et al. 04: All-frequency effects
- Combines lots of things: BRDF factorization, CPCA, nonlinear approx. with wavelets
- Idea: Factor BRDF to depend on incident, outgoing
  - Incident part handled with view-independent relighting
  - Then linearly combine based on outgoing factor
- Effectively, break problem into a few subproblems that can be solved view-independently and added up
  - Can apply nonlinear wavelet approx. to each subproblem
  - And CPCA to the matrices for further compression

Factored BRDFs: Critique

- Simple, reasonably practical method
- Problem: Non-optimal factorization, few terms
  - Can only handle less glossy materials
  - Accuracy not properly investigated [Mahajan et al 08]
- Very nice synthesis of many existing ideas
- Comparison to triple product integrals
  - Not as deep or cool, but simpler and real-time
  - Limits BRDF fidelity, glossiness much more
  - In a sense, they are different types of factorizations

Outline

- Motivation and Background
- Compression methods
  - Low frequency linear spherical harmonic approximation
  - Factorization and PCA
  - Local factorization and clustered PCA
  - Non-linear wavelet approximation
- Changing view as well as lighting
  - Clustered PCA
  - Factored BRDFs
  - Triple Product Integrals
**Factorization Approach**

\[
\begin{aligned}
&6D \text{ Transport} \\
\approx & \quad -10^6 \text{ samples} \\
\times & \quad 4D \text{ Visibility} \\
\approx & \quad 10^8 \text{ samples} \\
\times & \quad 10^8 \text{ samples} \\
\approx & \quad 10^8 \text{ samples}
\end{aligned}
\]

**Triple Product Integral Relighting**

\[
\text{Relit Images (3-5 sec/frame)}
\]

**Triple Product Integrals**

\[
B = \int_{\Omega^2} L(\omega) V(\omega) \bar{n}(\omega) \, d\omega
\]

\[
= \int_{\Omega^2} \left( \sum_i L_i \psi_i(\omega) \right) \left( \sum_j V_j \psi_j(\omega) \right) \left( \sum_k \bar{n}_k \psi_k(\omega) \right) \, d\omega
\]

\[
= \sum_i \sum_j \sum_k L_i V_j \bar{n}_k \int_{\Omega^2} \psi_i(\omega) \psi_j(\omega) \psi_k(\omega) \, d\omega
\]

\[
= \sum_i \sum_j \sum_k L_i V_j \bar{n}_k C_{ijk}
\]

**Basis Requirements**

1. Need few non-zero “tripling” coefficients

\[
C_{ijk} = \int_{\Omega^2} \psi_i(\omega) \psi_j(\omega) \psi_k(\omega) \, d\omega
\]

2. Need sparse basis coefficients

\[
L_i, V_j, \bar{n}_k
\]

**1. Number of Non-Zero Tripling Coefficients**

<table>
<thead>
<tr>
<th>Basis Choice</th>
<th>Number Non-Zero $C_{ijk}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>General (e.g. PCA)</td>
<td>$O(N^3)$</td>
</tr>
<tr>
<td>Pixels</td>
<td>$O(N)$</td>
</tr>
<tr>
<td>Fourier Series</td>
<td>$O(N^2)$</td>
</tr>
<tr>
<td>Sph. Harmonics</td>
<td>$O(N^{5/2})$</td>
</tr>
<tr>
<td>Haar Wavelets</td>
<td>$O(N \log N)$</td>
</tr>
</tbody>
</table>
2. Sparsity in Light Approx.

**Summary of Wavelet Results**
- Derive direct $O(N \log N)$ triple product algorithm
- Dynamic programming can eliminate $\log N$ term
- Final complexity linear in number of retained basis coefficients

**Broader Computational Relevance**
- Clebsch-Gordan triple product series for spherical harmonics in quantum mechanics (but not focused on computation)
- Essentially no previous work graphics, applied math
- Same machinery applies to basic operation: multiplication
  - Signal multiplication for audio, image compositing
  - Compressed signals/videos (e.g. wavelets JPEG 2000)

**Summary**
- Really a big data compression and signal-processing problem
- Apply many standard methods
  - PCA, wavelet, spherical harmonic, factor compression
- And invent new ones
  - VQPCA, wavelet triple products
- Guided by and gives insights into properties of illumination, reflectance, visibility
  - How many terms enough? How much sparsity?

**Subsequent Work**
- My survey linked from website (lecture only covers 2002-2004)
- Varied lighting/view. What about dynamic scenes, BRDFs
  - Much subsequent work [Zhou et al. 05, Ben-Artzi et al. 06]. But still limited for dynamic scenes
- Must work on GPU to be practical
- Sampling on object geometry remains a challenge
- Near-Field Lighting has had some work, remains a challenge
- Applications to lighting design, direct to indirect transfer
- New basis functions and theory
- Newer methods do not require precompute, various GPU tricks
- So far, low-frequency spherical harmonics used in games, all-frequency techniques have had limited applicability