IMAGE-BASED MODELING & RENDERING (1)

Some next lecture slides are adopted from slides of Prof. Ravi Ramamoorthi (USCD), Marc Levoy (Google), Michael Cohen (Facebook), Leonard McMillan (UNC) and Richard Szeliski (Facebook).
Modeling light

- How do we generate new scenes and animations from existing ones?
- Classic “3D Vision + Graphics”:  
  - take (lots of) pictures  
  - recover camera pose  
  - build 3D model  
  - extract texture maps / BRDFs  
  - synthesize new views
Computer Graphics

slides from Michael Cohen

Computer Vision

slides from Michael Cohen
Visual Computing (Combined)

But, vision technology falls short

slides from Michael Cohen
... and so does graphics.

![Diagram](image1)

Image-Based Rendering

![Diagram](image2)

slides from Michael Cohen
Traditional Modeling and Rendering

Model

User Input
- texture maps
- survey data

Modeling

Geometry
- Reflectance
- Light sources

Rendering

Images

For Photorealism:

Modeling is Hard   Rendering is Slow

Next few slides courtesy Paul Debevec; SIGGRAPH 99 course notes

Traditional Modeling and Rendering

Can we model and render this?
What do we want to do with the model?
Image-based modeling & rendering


Image-Based Modeling and Rendering

Images, user input, range scans → Model → Images
IBM / IBR

Image-based modeling (IBM) & Image-Based Rendering (IBR)

“The study of image-based modeling and rendering is the study of sampled representations of geometry.”

Geometry-based vs. image-based rendering

conceptual world

model construction

geometry

geometry-based rendering

flythrough of scene

real world

image acquisition

images

computer vision

image-based rendering

flythrough of scene
Shortcutting the vision/Graphics pipeline

real world

<table>
<thead>
<tr>
<th>vision pipeline</th>
</tr>
</thead>
</table>

geometry

<table>
<thead>
<tr>
<th>graphics pipeline</th>
</tr>
</thead>
</table>

views

image-based rendering

(from M. Cohen)

Game 1: Increase the dimensionality

<table>
<thead>
<tr>
<th>2D rgb</th>
<th>texture</th>
</tr>
</thead>
<tbody>
<tr>
<td>2D rgbz</td>
<td>range image</td>
</tr>
<tr>
<td>2.5D rgbαz</td>
<td>layered depth images</td>
</tr>
<tr>
<td>4D rgb</td>
<td>light field / Lumigraph</td>
</tr>
<tr>
<td>4D rgbz</td>
<td>array of range images</td>
</tr>
<tr>
<td>4.5D rgbαz</td>
<td>layered light fields</td>
</tr>
</tbody>
</table>
Game2: Replace the Quality Represented

4D rgb 
\{u, v, s, t\}
light field / Lumigraph

5D rgb 
\{x, y, z, \theta, \phi\}
plenoptic function

6D \rho 
\{u, v, s, t\} \times \{\theta_i, \phi_i\}
free-space BRDF field

7D \rho 
\{x, y, z\} \times \{\theta_i, \phi_i, \theta_f, \phi_f\}
BRDF volume

Image-based representations: the classics

- **3D**
  - model + texture/reflectance map [Blinn78]
  - model + displacement map [Cook84]
  - volume rendering [Levoy87, Drebin88]

- **2D + Z**
  - range images [Binford73]
  - disparity maps [vision literature]

- **2.5D**
  - sprites [vis-sim, games]

- **n 2D**
  - epipolar plane images [Bolles87]
  - movie maps [Lippman78]

- **2D**
  - environment maps, a.k.a. panoramas [19th century]
Recent additions

- **full model**
  - view-dependent textures [Debevec96]
  - surface light fields [Wood00]
  - Lumigraphs [Gortler96]

- **sets of range images**
  - view interpolation [Chen93, McMillan95, Mark97]
  - layered depth images [Shade98]
  - relief textures [Oliveira00]

- **feature correspondences**
  - plenoptic editing [Seitz98, Dorsey01]

- **camera pose**
  - image caching [Schaufler96, Shade96]
  - sprites + warps [Lengyel97]
  - light fields [Levoy96]

- **no model**
  - outward-looking QTVR [Chen95]

---

**Spectrum of IBMR**

- **Model**
  - Kinematics, etc.
  - Geometry + Materials
  - Geometry + Images
  - Images + Depth
  - Light Field
  - Movie Map
  - Panorama
  - Image

- **Image-Based Modeling**
  - Images, renderings, user input, range scans

- **Image-Based Rendering**
  - Images
IBR: Pros and Cons

• Advantages
  – Easy to capture images: photorealistic by definition
  – Simple, universal representation
  – Often bypass geometry estimation?
  – Independent of scene complexity?

• Disadvantages
  – WYSIWYG but also WYSIAYG
  – Explosion of data as flexibility increased
  – Often discards intrinsic structure of model?

• Today, IBR-type methods also often used in synthetic rendering (e.g. real-time rendering PRT)
  – General concept of data-driven graphics, appearance
  – Also, data-driven geometry, animation, simulation
  – Spawned light field cameras for image capture

Image-Based Models: What do they allow?

<table>
<thead>
<tr>
<th>Model</th>
<th>Movement</th>
<th>Geometry</th>
<th>Lighting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Geometry + Materials</td>
<td>Continuous</td>
<td>Global</td>
<td>Dynamic</td>
</tr>
<tr>
<td>Geometry + Images</td>
<td>Continuous</td>
<td>Global</td>
<td>Fixed</td>
</tr>
<tr>
<td>Images + Depth</td>
<td>Continuous</td>
<td>Local</td>
<td>Fixed</td>
</tr>
<tr>
<td>Light Field</td>
<td>Continuous</td>
<td>None</td>
<td>Fixed</td>
</tr>
<tr>
<td>Movie Map</td>
<td>Discrete</td>
<td>None</td>
<td>Fixed</td>
</tr>
<tr>
<td>Panorama</td>
<td>Rotation</td>
<td>None</td>
<td>Fixed</td>
</tr>
<tr>
<td>Image</td>
<td>None</td>
<td>None</td>
<td>Fixed</td>
</tr>
</tbody>
</table>
IBR: A brief history

- Texture maps, bump maps, environment maps [70s]
- Poggio MIT 90s: Faces, image-based analysis/synthesis
- Mid-Late 90s
  - Chen and Williams 93, View Interpolation [Images+depth]
  - Chen 95 Quicktime VR [Images from many viewpoints]
  - McMillan and Bishop 95 Plenoptic Modeling [Images w disparity]
  - Gortler et al, Levoy and Hanrahan 96 Light Fields [4D]
  - Shade et al. 98 Layered Depth Images [2.5D]
  - Debevec et al. 00 Reflectance Field [4D]
  - Inverse rendering (Marschner, Sato, Yu, Boivin,...)
- Today: IBR hasn’t replaced conventional rendering, but has brought sampled and data-driven representations to graphics

FOUNDATIONS FOR IBMR

Some next lecture slides are adopted from slides of Prof. Ravi Ramamoorthi (USCD), Marc Levoy (Google), Michael Cohen (Facebook), Leonard McMillan (UNC) and Richard Szeliski (Facebook).
Images as a Collection of Rays

An image is a subset of the rays seen from a given point - this "space" of rays occupies two dimensions.

The Plenoptic Function

The set of rays seen from all points ... 

\[ p = P(\theta, \phi, x, y, z, \lambda, t) \]
Image-based Rendering is about

...reconstructing a plenoptic function from a set of samples taken from it.

✓ Ignoring time, and selecting a discrete set of wavelengths gives a 5-D plenoptic function

Where to Begin?

✓ Pinhole camera model

- Defines a mapping from image points to rays in space
Mapping from Rays to Points

Simple Derivation

\[ \mathbf{P} = \begin{bmatrix} u_x & v_x & o_x \\ u_y & v_y & o_y \\ u_z & v_z & o_z \end{bmatrix} \]

\[ \mathbf{X} = \mathbf{C} + t \mathbf{P} \mathbf{x} \]

Correspondence

\[ \hat{\mathbf{C}}_2 + t_2 \mathbf{P}_2 \bar{\mathbf{x}}_2 = \hat{\mathbf{C}}_1 + t_1 \mathbf{P}_1 \bar{\mathbf{x}}_1 \]

\[ t_1 \mathbf{P}_1 \bar{\mathbf{x}}_2 = \mathbf{C}_1 - \hat{\mathbf{C}}_2 + t_1 \mathbf{P}_1 \bar{\mathbf{x}}_1 \]

\[ t_2 \bar{\mathbf{x}}_2 = \mathbf{P}_2^{-1}(\hat{\mathbf{C}}_1 - \hat{\mathbf{C}}_2) + t_1 \mathbf{P}_1^{-1} \mathbf{P}_1 \bar{\mathbf{x}}_1 \]

\[ \frac{1}{t_2} \bar{\mathbf{x}}_2 = \frac{1}{t_2} \mathbf{P}_2^{-1}(\hat{\mathbf{C}}_1 - \hat{\mathbf{C}}_2) + \mathbf{P}_1^{-1} \mathbf{P}_1 \bar{\mathbf{x}}_1 \]

\[ \bar{\mathbf{x}}_2 = \frac{1}{t_2} \mathbf{P}_2^{-1}(\hat{\mathbf{C}}_1 - \hat{\mathbf{C}}_2) + \frac{1}{t_2} \mathbf{P}_1^{-1} \mathbf{P}_1 \bar{\mathbf{x}}_1 \]
Warping in Action

✓ A 3D Warp

Visibility

✓ The warping equation determines where points go...

... but that is not sufficient
Partition Reference Image

- Project the desired center-of-projection onto the reference image

Enumeration

- Drawing toward the projected point guarantees an occlusion compatible ordering
- Ordering is consistent with a painter’s algorithm
- Independent of the scene’s contents
- Easily generalized to other viewing surfaces
- No auxiliary information required
Reconstruction

- Typical images are discrete, not continuous
- An image can be formed by different geometries

IMAGE-BASED MODELING

Some next lecture slides are adopted from slides of Prof. Ravi Ramamoorthi (USCD), Marc Levoy (Google), Michael Cohen (Facebook), Leonard McMillan (UNC), Richard Szeliski (Facebook), and Giljoo Nam (KAIST).
3D Imaging

- passive
  - shape from stereo
  - shape from focus
  - shape from motion, etc.

- active
  - texture-assisted shape-from-X
  - triangulation using structured-light
  - time-of-flight

A General Pipeline for 3D Imaging

Pre-processing
- Camera/Projector Model
- Camera/Projector Calibration

Partial Geometry
- Triangulation
- Structured Light Imaging

Geometry Enhancement
- Photometric Stereo
- Optimizing Geometry

Complete 3D Geometry
- RBT + ICP
- Surface Reconstruction
**Line Closest Point**

- This method can reconstruct the 3D point, but it is not robust to noise.
- More practical approach usually exploits line-plane intersection point.

**Line-Plane Intersection**

- Except for parallel situation, they always intersect at a point.
- Line : a ray from camera
- Plane : a plane from projector
Line-Plane Intersection

\[ P = \{ p : n' ( p - q_p ) = 0 \} \]

\[ L = \{ p = q_C + \lambda v \} \]

\[ n' ( p - q_p ) = n' ( \lambda v + q_C - q_p ) = 0 \]

\[ \lambda = \frac{n'(q_p - q_C)}{n'v} \]

\[ p = q_C + \lambda v \]

---

Line-Plane Intersection

- Plane sweep
  - We sweep the plane and capture each plane with the camera
  - Laser 3D scanners
Camera-Projector Calibration Results


Structured Lighting: Encoding & Decoding Scheme

- Encoding on a projector
  - Encoding the corresponding points of a projector
- Decoding on a camera
  - Decoding the corresponding points of a camera
Structured Lighting:
Encoding Patterns

Structured Lighting:
Decoding Patterns
Structured Lighting per View

Estimating Normals: Photometric Stereo

- First proposed by [Woodham 1980]
  - Use at least 3 light sources and one camera
  - Assume diffuse reflection

Enhance Geometry using Normals

- Where does the low frequency bias come from?
  - Directional light assumption

Enhance Geometry using Normals

- Where does the low frequency bias come from?
  - Directional light assumption
  - Perspective camera projection
  - Etc.
• The apparent brightness of a diffuse surface is the same regardless of the observer’s viewpoint.

\[ I = \rho (\overrightarrow{N} \cdot \overrightarrow{L}) \]

- \( I \): observed intensity
- \( \rho \): diffuse reflectance
- \( \overrightarrow{N} \): surface normal (unit vector)
- \( \overrightarrow{L} \): light direction (unit vector)

\[ I_1 = \rho \left[ L_{1,x} \quad L_{1,y} \quad L_{1,z} \right] \left[ \begin{array}{c} N_x \\ N_y \\ N_z \end{array} \right] \]

- \( I_1 \): observed intensity
- \( \rho \): diffuse reflectance
- \( \overrightarrow{N} \): surface normal (unit vector)
- \( \overrightarrow{L} \): light direction (unit vector)
Diffuse Reflection

\[ I = \rho (\vec{N} \cdot \vec{L}) \]

\[ \begin{bmatrix} I_1 \\ I_2 \\ I_3 \end{bmatrix} = \rho \begin{bmatrix} L_{1,x} & L_{1,y} & L_{1,z} \\ L_{2,x} & L_{2,y} & L_{2,z} \\ L_{3,x} & L_{3,y} & L_{3,z} \end{bmatrix} \begin{bmatrix} N_x \\ N_y \\ N_z \end{bmatrix} \]

- \( I_1 \): observed intensity
- \( \rho \): diffuse reflectance
- \( \vec{N} \): surface normal (unit vector)
- \( \vec{L} \): light direction (unit vector)
Diffuse Reflection

\[ I = \rho \left( \vec{N} \cdot \vec{L} \right) \]

- \( \rho \vec{N} = (\vec{L} \vec{L})^{-1} \vec{L} ' I \)
- \( \vec{N} = \text{normalize}(N) \)
- \( \rho = \| \rho \vec{N} \| \)

- \( I \): observed intensity
- \( \rho \): diffuse reflectance
- \( \vec{N} \): surface normal (unit vector)
- \( \vec{L} \): light direction (unit vector)

Photometric Stereo

- normal map color coding:
  \[ \{ R, G, B \} = \left\{ \frac{N_x + 1}{2}, \frac{N_y + 1}{2}, \frac{N_z + 1}{2} \right\} \]

- diffuse reflectance per channel:
  \[ \rho = \{ \rho_R, \rho_G, \rho_B \} \in \mathbb{R}^3 \]
  \[ \rho_R, \rho_G, \rho_B \in [0,1] \]
Photometric Stereo Summary

<table>
<thead>
<tr>
<th>Acquisition Setup</th>
<th>Observation</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Diagram" /></td>
<td><img src="image2" alt="Image" /></td>
<td><img src="image3" alt="Image" /></td>
</tr>
</tbody>
</table>


Light Calibration

Calculating light direction

\[
\tilde{L} = 2(\tilde{N} \cdot \tilde{R})\tilde{N} - \tilde{R}
\]

\[
\tilde{R} = \tilde{V} = [0,0,1]^T
\]

Photometric Stereo

Enhance Geometry using Normals

- Enhance Geometry
  - [Nehab et al. 2005] presented in SIGGRAPH 2005
  - Geometry: Good Low Frequency + High Frequency Errors
  - Normal: Low Frequency Bias + Good High Frequency

Enhance Geometry using Normals

Original geometry + Normals = Optimized geometry

A General Pipeline for 3D Imaging

Complete 3D Geometry
- RBT + ICP
- Surface Reconstruction

Register all partial geometries into a common global coordinates (RBT + ICP)

Get implicit function for 3D surface (Poisson reconstruction)

Mesh indexing (Marching cubes)

Done
Merging Multi-view Geometries

Merging Two Geometries

- Suppose the object is rigid, then this problem is considered as rigid-body transformation
- Select at least 3 pairs of same points of the object automatically or interactively
Merging Multi-view Point Clouds

Surface from Implicit function

- Empty vertices: the implicit function value is positive
- Yellow vertices: the implicit function value is negative
- Then, depending on combination of each cell, draw the surface as figures above
3D Imaging Result

3D object

3D geometry

3D printed

Image Based Models

Modeling and Rendering Architecture from Photographs

Original photograph with marked edges

Recovered model

Model edges projected onto photograph

Synthetic rendering
Image-Based Modeling

- Create 3D model (and texture maps) from images
- automated
  - (structure from motion, stereo)
- interactive
  - Façade system

Façade

1. Select building blocks
2. Align them in each image
3. Solve for camera pose and block parameters (using constraints)
View-dependent texture mapping

1. Determine visible cameras for each surface element
2. Blend textures (images) depending on distance between original camera and novel viewpoint

Model-based stereo

• Compute offset from block model

• Some more results:
Image-Based Faces

- Estimate shape from images
- Match metrics to shape
- Project video onto shape
  - Texture map
- Animate

Graphics/Imaging Continuum

Geometry centric

- Fixed geometry
- View-dependent texture
- View-dependent geometry
- Sprites with depth

Image centric

- Concentric mosaics
- Lumigraph
- Light field

Polygon rendering + texture mapping

Warping

Interpolation
KAIST-VCLAB IBMR Example

• See video