6.801/866

Image-Based Rendering

T. Darrell
Vision for Graphics

So far: stereo, motion, tracking, model-based recognition, most focusing on recovering 3-D models with accurate shape…

One of the main applications of vision is making new pictures!

• Do we need detailed models?
• Do we need Euclidean 3-D shape?
• Are dense range images useful?
Image-based rendering

Synthesize new views from a set of pictures.

Pre-Recorded Images $I_1$, $I_2$, ..., $I_n$ → Model → Control → New Image
1. Build a 3-D model; re-render from new viewpoint
   - Multi-view stereo; “Virtualized Reality”
   - Visual Hulls
   - Model-based stereo

2. Establish correspondences; use view transfer
   - Affine view synthesis

3. Model sets of light rays
   - Lightfields, Lumigraphs, …
Taxonomy

1. Build a 3-D model; re-render from new viewpoint
   - Multi-view stereo; “Virtualized Reality”
   - Visual Hulls
   - Model-based stereo

2. Establish correspondences; use view transfer
   - Affine view synthesis

3. Model sets of light rays
   - Lightfields, Lumigraphs, …
Taxonomy

1. Build a 3-D model; re-render from new viewpoint
   - Multi-view stereo; “Virtualized Reality”
   - Visual Hulls
   - Model-based stereo

2. Establish correspondences; use view transfer
   - Affine view synthesis

3. Model sets of light rays
   - Lightfields, Lumigraphs, …
Taxonomy

1. Build a 3-D model; re-render from new viewpoint
   - Multi-view stereo; “Virtualized Reality”
   - Visual Hulls
   - Model-based stereo

2. Establish correspondences; use view transfer
   - Affine view synthesis

3. Model sets of light rays
   - Lightfields, Lumigraphs, …
Models from stereo

FIGURE 12.13: Correlation-based stereo matching: (a) a pair of stereo pictures; (b) a texture-mapped view of the reconstructed surface; (c) comparison of the regular (left) and refined (right) correlation methods in the nose region. Reprinted from [Devernay and Faugeras, 1994], Figures 5, 8 and 9.
CMU’s 3-D Room

49 camera 3-D room:

[Kanade et al. 1998]
Multi-view stereo for VR

Compute dense range image from 3-6 nearby cameras:

Merge into global mesh.
Texture and render new views....

[Kanade et al. 1998]
"Virtualized Reality"

[Real View 1]

Movement of virtual view

[Real View 2]

[Kanade et al. 1998]
Models

• Virtualized reality
  – very accurate
  – many correspondences
  – many cameras

• What can you do with a few cameras, and just silhouettes?
Visual Hulls

Visual Hull [Laurentini, 91]: the minimal object that produces the given silhouettes
- 3-D model contains the true object
- visual cone intersection
- texture mapped for a desired viewpoint

[Matusik]
Smoothed Visual Hull

Fit surface spline to mesh; relax model according to smoothness assumption. [Sullivan and Ponce]
Smoothed Visual Hull
Smoothed Visual Hull Result
Smoothed Visual Hull Result
Smoothed Visual Hull Result
Image-based Visual Hulls

Visual Hull can be computed in $O\left(Kn^2\right)$ from $K$ images with $n \times n$ pixels, without computing any explicit 3-D geometry (Matusik et al, 2001)

Exploit view-dependent texture mapping (more later…)

Image-Based Visual Hulls
Model-based SFM

• Assume parametric shape model
  – boxes
  – prisms
  – solids of revolution
  – unknown height, width, etc…
  – constraints between unknowns

• Given marked features, fit model to image using (relatively simple) non-linear search.
Façade

Visually compelling model from just a few photographs!

Three steps:

• Photogrammetry (Model-based SFM)
• View dependent Texture Mapping
• Model-based Stereopsis
Photogrammetry (Model-based SFM)

Line features  recovered model  model overlay  recovered texture
View-dependent texture
Model-based stereo for surface detail
Façade
Façade
Façade Movie
Calibration/model free IBR?

- Cameras are hard to calibrate...desirable to have IBR methods that work without external/scene knowledge
- Recover affine structure from motion
- Use to insert virtual objects that follow camera motion...
Taxonomy

1. Build a 3-D model; re-render from new viewpoint
   - Multi-view stereo; “Virtualized Reality”
   - Visual Hulls
   - Model-based stereo

2. Establish correspondences; use view transfer
   - Affine view synthesis

3. Model sets of light rays
   - Lightfields, Lumigraphs, …
Affine image transfer

Use affine model ... Given $P_0$-$P_3$, and

$$p = AP + b, \quad \text{where} \quad A = \begin{pmatrix} a_1^T \\ a_2^T \end{pmatrix}$$

With appropriate choice of 4 bases we can express projected location of points as:

$$p = (1 - x - y - z)p_0 + xp_1 + yp_2 + zp_3.$$ 

1. Given $m \geq 2$ images of $p_0$-$p_3$ and $p$ solve using least-squares for $x, y, z$

2. Use $x, y, z$ and positions of $p_0$-$p_3$ in new view to find $p$ in new view.
Augmented reality

Find cameras with black squares; add virtual object to scene with correct camera motion.
Model recovery

• View transfer good for many special effects and augmented reality applications.
• For model recovery, dense correspondence is needed!
• But correspondence is hard! … (and/or models are approximate)
• What can we do without correspondence?
• Model visible rays, not shape…. 
Taxonomy

1. Build a 3-D model; re-render from new viewpoint
   - Multi-view stereo; “Virtualized Reality”
   - Visual Hulls
   - Model-based stereo

2. Establish correspondences; use view transfer
   - Affine view synthesis

3. Model sets of light rays
   - Lightfields, Lumigraphs, …
The Plenoptic function

- IBR $\rightarrow$ recover geometric and photometric models from photographs, bypass the modeling process.
- *Plenoptic function*: images that can be seen!

- What parameterizes visible rays?
  - Camera position
  - Viewing angle
  - Wavelength
  - Time

(In a non-dispersive medium...)
The Plenoptic function

- Adelson and Bergen’s Plenoptic function
- $7D \rightarrow 5D \rightarrow 4D \rightarrow 2D$

$7D$: $(c_x, c_y, c_z, \theta, \phi, \lambda, t)$
$5D$: $(c_x, c_y, c_z, \theta, \phi)$
$4D$: $(x_1, y_1, x_2, y_2)$
$2D$: $(\theta, \phi)$

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Viewing space</th>
<th>Name</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>free</td>
<td>plenoptic function</td>
<td>1991</td>
</tr>
<tr>
<td>5</td>
<td>free</td>
<td>plenoptic modeling</td>
<td>1995</td>
</tr>
<tr>
<td>4</td>
<td>inside a 3D box</td>
<td>Lightfield/Lumigraph</td>
<td>1996</td>
</tr>
<tr>
<td>3</td>
<td>inside a 2D circle</td>
<td>concentric mosaics</td>
<td>1999</td>
</tr>
<tr>
<td>2</td>
<td>at a fixed point</td>
<td>panorama</td>
<td>1994</td>
</tr>
</tbody>
</table>

[Shum and He]
Lightfields

- Approximate Plenoptic function for fixed camera location, time, ...
- Reparametrize rays based on planar intersection
- A “light slab”:

\[ L(u, v, s, t) \]
Lightfields

- Generally, 2D slices of 4D data set
- For a new views compute other 2D slices
- Challenges:
  - Capture
  - Parameterization
  - Compression
  - Rendering
Alternate representations

- Point / angle
- Two points on a sphere
- Points on two planes
- Original images and camera positions…
Light-field rendering

• Compute intersection with \((u,v)\) and 
\((s,t)\) planes, take closest ray

• Interpolation possibilities
  – Bilinear in \((u,v)\) only
  – Bilinear in \((s,t)\) only
  – Quadrilinear in \((u,v,s,t)\)
Example lightfields
Example lightfields
Example lightfields
Unstructured Lumigraph

Generalize model-based view-dependent texture mapping (e.g., Façade) and Lightfield

Both are methods for interpolating color values for a desired ray as some combination of input rays.

  VDTM: use geometric model as proxy
  LFR: planar light “slab”

The Unstructured Lumigraph [Buehler 2001] is an IBR algorithm that includes VDTM and LFR as special cases, and has nice properties of each.
Unstructured Lumigraph

Desirable properties
Geometric proxies
Unstructured input
Epipole consistency
Minimal angular deviation
Continuity
Resolution Sensitivity
Equivalent ray consistency
Real-time
Unstructured Lumigraph Rendering

- Example: hallway with “tunnel” geometric proxy (inside of cube).
- Images gathered from translating robot.
- 3-D effect with no (local) 3-D structure….
1. Build a 3-D model; re-render from new viewpoint
   - Multi-view stereo; “Virtualized Reality”
   - Visual Hulls
   - Model-based stereo

2. Establish correspondences; use view transfer
   - Affine view synthesis

3. Model sets of light rays
   - Lightfields, Lumigraphs, …

[Figures from Forsythe and Ponce unless Attributed]
Endgame

• Exams due today
• Project show and tell on Tuesday—3 slides for L.M.—stand up and give a 2-3 minute overview (videotaped)
• Projects by 12/10 (or electronically by 12/15 with extension)

<table>
<thead>
<tr>
<th>Date</th>
<th>Topic</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>11/26</td>
<td>Model-Based Vision</td>
<td>Req: FP 18</td>
</tr>
<tr>
<td>11/28</td>
<td>Thanksgiving (NO LECTURE)</td>
<td></td>
</tr>
<tr>
<td>12/3</td>
<td>Image Databases</td>
<td>Req: FP 25</td>
</tr>
<tr>
<td>12/5</td>
<td>Image-Based Rendering</td>
<td>Req: FP 26</td>
</tr>
<tr>
<td>12/10</td>
<td>Project Show and Tell</td>
<td>Projects Due</td>
</tr>
</tbody>
</table>
# Recap

<table>
<thead>
<tr>
<th>#</th>
<th>Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>9/5</td>
<td>Course Introduction</td>
</tr>
<tr>
<td>2</td>
<td>9/10</td>
<td>Cameras, Lenses, and Sensors</td>
</tr>
<tr>
<td>3</td>
<td>9/12</td>
<td>Radiometry and Shading Models I</td>
</tr>
<tr>
<td>4</td>
<td>9/17</td>
<td>Radiometry and Shading Models II</td>
</tr>
<tr>
<td>5</td>
<td>9/19</td>
<td>Multiview Geometry</td>
</tr>
<tr>
<td>6</td>
<td>9/24</td>
<td>Stereo</td>
</tr>
<tr>
<td>7</td>
<td>9/26</td>
<td>Color</td>
</tr>
<tr>
<td>8</td>
<td>10/1</td>
<td>Shape from Shading</td>
</tr>
<tr>
<td>9</td>
<td>10/3</td>
<td>Image Filtering</td>
</tr>
<tr>
<td>10</td>
<td>10/8</td>
<td>Image Representations</td>
</tr>
<tr>
<td>11</td>
<td>10/10</td>
<td>Texture and Edges</td>
</tr>
<tr>
<td>12</td>
<td>10/17</td>
<td>Bayesian Analysis</td>
</tr>
<tr>
<td>13</td>
<td>10/22</td>
<td>Optic Flow and Direct SFM</td>
</tr>
<tr>
<td>14</td>
<td>10/24</td>
<td>Affine Reconstruction</td>
</tr>
<tr>
<td>15</td>
<td>10/29</td>
<td>Interactive Systems (Low-Level)</td>
</tr>
<tr>
<td>16</td>
<td>10/31</td>
<td>Face Detection and Recognition I</td>
</tr>
<tr>
<td>17</td>
<td>11/5</td>
<td>Face Detection and Recognition II</td>
</tr>
<tr>
<td>18</td>
<td>11/7</td>
<td>Projective Reconstruction</td>
</tr>
<tr>
<td>19</td>
<td>11/12</td>
<td>Segmentation I</td>
</tr>
<tr>
<td>20</td>
<td>11/14</td>
<td>Segmentation II</td>
</tr>
<tr>
<td>21</td>
<td>11/19</td>
<td>Tracking I</td>
</tr>
<tr>
<td>22</td>
<td>11/21</td>
<td>Tracking II</td>
</tr>
<tr>
<td>23</td>
<td>11/26</td>
<td>Model-Based Vision</td>
</tr>
<tr>
<td>24</td>
<td>12/3</td>
<td>Image Databases</td>
</tr>
<tr>
<td>25</td>
<td>12/5</td>
<td>Image-Based Rendering</td>
</tr>
<tr>
<td>26</td>
<td>12/10</td>
<td>Project Show and Tell</td>
</tr>
</tbody>
</table>
Thanks!

*We* learned a lot! (and we hope you did too!)
Comments are very welcome to refine this class in the future.

Thanks to:

- Erik
- Louis
- You