Large Scale 3D Reconstruction
(50 mins)

Yasutaka Furukawa @ Washington University in St. Louis
Outline

1. Large scale MVS for organized photos
   (Aerial photos)

2. Large scale MVS for unorganized photos
   (Internet community photos)

3. Large scale indoor modeling
Common framework

- Divide
- Reconstruct
- Merge
Common framework

- Divide (the problem domain)
- Reconstruct (small problems in parallel)
- Merge (independent reconstructions)
Large Scale MVS for Organized Photos (Aerial photos)

[ Google Maps ]
Drones are getting hot
Drones are getting hot

CAPTURED BY ANY CAMERA
Input images
Input images
Divide and Reconstruct
Divide and Reconstruct
Simplest strategy for divide and reconstruct

- Given N images
- Divide into N depthmap-reconstruction problems
Merge

- The most difficult step
- Depends on the application (visualization, analysis, …)
Merge for visualization

- Hide gaps between reconstructions
• Merge for visualization
• Hide gaps between reconstructions

Ideal model

Model before merge

See through!
Merge for visualization

- Hide gaps between reconstructions

Ideal model

Model before merge

See through!

Model after merge

More consistent and fatter
Merge to a single consistent model

- Open problem for a polygonal mesh
  - “Multilevel Streaming for Out-of-Core Surface Reconstruction” [Bolitho et al., 2007]
  - Companies in industry might have something [Google, Microsoft, Apple, Pix4d, Accute3D, …]

- Point cloud representation is easy
  “Towards Internet-scale Multi-View Stereo” [Furukawa et al., 2010]
Merged 3D Point-cloud
Example (Google Maps)
Outline

1. Large scale MVS for organized photos (Aerial photos)

2. Large scale MVS for unorganized photos (Internet community photos)

3. Large scale indoor modeling
Large-Scale MVS for Unorganized Photos (Internet Photos)

“Building Rome in a Day” [ Agarwal, Snavely, et al., 2009 ]
Divide is a challenge
Divide is a challenge

Divide in the image space?
Divide is a challenge

Divide in the model space?
Image-clustering based on a model

“Towards Internet-scale Multi-View Stereo” [furukawa et al., 2010]

Image cluster
Image-clustering based on a model

“Towards Internet-scale Multi-View Stereo” [furukawa et al., 2010]
One way to look at the clustering problem

![Diagram with orange nodes and black lines showing connections]

- Orange circle: Image
- Black line: Common features
One way to look at the clustering problem

Normalized Cuts

- Image
- Common features
Formulation

P1 is reconstructed well in C1 based on (image resolutions/distribution)

“P1 is covered by C1.”

“P4 is covered by C2.”
We want to
1. remove redundant images
2. keep each cluster small (memory limit)
3. “cover” many points
Formulation

SFM points \( \{P_1, P_2, \ldots\} \)

Images \( \{I_1, I_2, \ldots\} \)

Image clusters \( \{C_1, C_2, \ldots\} \)

Minimize \( \sum_{k} |C_k| \) subject to

- \( \forall k \quad |C_k| \leq \alpha, \)
- \( \forall i \quad \frac{\text{# of covered points in } I_i}{\text{# of points in } I_i} \geq \delta. \)
Image Clustering Algorithm

SFM points \( \{P_1, P_2, \ldots\} \)

Images \( \{I_1, I_2, \ldots\} \)

Image clusters \( \{C_1, C_2, \ldots\} \)

Minimize \( \sum_k |C_k| \) subject to

- \( \forall k\ |C_k| \leq \alpha \), (compactness)
- \( \forall i\ \frac{\# \text{ of covered points in } I_i}{\# \text{ of points in } I_i} \geq \delta \). (coverage)
Image Clustering Algorithm

Minimize $\sum_k |C_k|$ subject to

- $\forall k \; |C_k| \leq \alpha$, (compactness)
- $\forall i \; \frac{\text{# of covered points in } I_i}{\text{# of points in } I_i} \geq \delta$. (coverage)
Image Clustering Algorithm

1. Start with all the images in a cluster

Minimize $\sum_k |C_k|$ subject to

- $\forall k \quad |C_k| \leq \alpha$, (compactness)
- $\forall i \quad \frac{\text{# of covered points in } I_i}{\text{# of points in } I_i} \geq \delta$. (coverage)
Image Clustering Algorithm

1. Start with all the images in a cluster
2. Optimize *compactness* while keeping *coverage*

\[
\text{Minimize } \sum_{k} |C_k| \quad \text{subject to} \quad \begin{align*}
\forall k & \quad |C_k| \leq \alpha, \\
\forall i & \quad \frac{\# \text{ of covered points in } I_i}{\# \text{ of points in } I_i} \geq \delta.
\end{align*}
\]

*compactness* (size) (coverage)
Image Clustering Algorithm

1. Start with all the images in a cluster
2. Optimize **compactness** while keeping **coverage**
3. If **size** constraint is broken, split a cluster

Minimize \( \sum_{k} |C_k| \) subject to
- \( \forall k \quad |C_k| \leq \alpha \),
- \( \forall i \quad \frac{\# \text{ of covered points in } I_i}{\# \text{ of points in } I_i} \geq \delta \).

- (compactness)
- (size)
- (coverage)
Image Clustering Algorithm

1. Start with all the images in a cluster
2. Remove redundant images while keeping coverage
3. If size constraint is broken, split a cluster
4. Add an image to each cluster to satisfy coverage

\[
\text{Minimize } \sum_{k} |C_k| \quad \text{subject to} \quad \begin{align*}
\forall k & |C_k| \leq \alpha, \quad \text{(size)} \\
\forall i & \left\{ \frac{\text{# of covered points in } I_i}{\text{# of points in } I_i} \right\} \geq \delta. \quad \text{(coverage)}
\end{align*}
\]
Image Clustering Algorithm

1. Start with all the images in a cluster
2. Remove redundant images while keeping coverage
3. If size constraint is broken, split a cluster
4. Add an image to each cluster to satisfy coverage
5. Repeat (3,4) until size and coverage are satisfied

Minimize \( \sum_k |C_k| \) subject to:

- \( \forall k \quad |C_k| \leq \alpha, \) (size)
- \( \forall i \quad \frac{\# \text{ of covered points in } I_i}{\# \text{ of points in } I_i} \geq \delta. \) (coverage)
Merge to a single consistent model

- Open problem for a polygonal mesh
  - “Multilevel Streaming for Out-of-Core Surface Reconstruction” [Bolitho et al., 2007]
  - “Fusion of Depth Maps with Multiple Scales” [Fuhrmann et al., 2011]

- Companies in industry might have something [Google, Microsoft, Apple, Pix4d, Accute3D, …]

- Point cloud representation is easy
  “Towards Internet-scale Multi-View Stereo” [Furukawa et al., 2010]
St. Peter’s Basilica

- 1275 images
- 4 clusters
- 6M 3D points
# Statistics

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Basilica</th>
<th>Trevi</th>
<th>Colosseum</th>
<th>Dubrovnik</th>
<th>San Marco</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input</strong></td>
<td>1275</td>
<td>1936</td>
<td>1167</td>
<td>6304</td>
<td>13709</td>
</tr>
<tr>
<td><strong>After image selection</strong></td>
<td>298</td>
<td>212</td>
<td>490</td>
<td>2097</td>
<td>3535</td>
</tr>
<tr>
<td><strong>Sum over output clusters</strong></td>
<td>333</td>
<td>218</td>
<td>528</td>
<td>2628</td>
<td>5917</td>
</tr>
<tr>
<td><strong># of clusters</strong></td>
<td>4</td>
<td>2</td>
<td>7</td>
<td>28</td>
<td>67</td>
</tr>
<tr>
<td><strong>View clustering</strong></td>
<td>1.3</td>
<td>3.3</td>
<td>1.5</td>
<td>21.1</td>
<td>39.3</td>
</tr>
<tr>
<td><strong>Running time [min]</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PMVS</td>
<td>305 (84)</td>
<td>165 (84)</td>
<td>232 (45)</td>
<td>1202 (180)</td>
<td>2103 (165)</td>
</tr>
<tr>
<td>Merge</td>
<td>39.9 (10.2)</td>
<td>12.6 (6.5)</td>
<td>73.0 (14.2)</td>
<td>474.6 (41.4)</td>
<td>666.5 (11.3)</td>
</tr>
</tbody>
</table>
Outline

1. Large scale MVS for organized photos (Aerial photos)

2. Large scale MVS for unorganized photos (Internet community photos)

3. Large scale indoor modeling
Large-Scale Indoor Modeling

“Reconstructing the World's Museums” [Xiao and Furukawa, 2012]
Best Student Paper Award at ECCV
Technical Contributions

- Architectural shape priors
- Single consistent 3D model (real merging)
Technical Contributions

- Architectural shape priors
- Single consistent 3D model (real merging)

**Inverse** Constructive Solid Geometry (CSG)
Technical Contributions

- Architectural shape priors
- Single consistent 3D model (real merging)

Inverse Constructive Solid Geometry (CSG)
Technical Contributions

• Architectural shape priors

• Single consistent 3D model (real merging)

**Inverse** Constructive Solid Geometry (CSG)
CGAL

- From CSG to a mesh
Algorithm

3D point cloud → 2D CSG (floorplan) → 3D CSG model → Wall model → Final textured model
Cut into Slices

point count

3D point cloud → 2D CSG (floorplan) → 3D CSG model → Wall model → Final textured model
Cut into Slices

3D point cloud → 2D CSG (floorplan) → 3D CSG model → Wall model → Final textured model
2D CSG Reconstruction

1. Generate primitives

2. Build 2D CSG
Primitive Generation and Fitting (bottom-up)

- 2D rectangle
- 2D line
- Point
2D CSG Reconstruction

1. Generate primitives

point $\rightarrow$ line
2D CSG Reconstruction

1. Generate primitives

- point $\rightarrow$ line
- line $\rightarrow$ rectangle

From 4 line segments
2D CSG Reconstruction

1. Generate primitives

2. Build 2D CSG
2D CSG Reconstruction

Explain the data

- **Free space**
- **Laser points**
2D CSG Reconstruction

1. Generate primitives
2. Build 2D CSG

Repeat in each slice
3D CSG Reconstruction

1. Generate primitives (cuboids)

2. Build 3D CSG (out of primitive candidates)
3D CSG Reconstruction

1. Generate primitives (cuboids)
3D CSG Reconstruction

1. Generate primitives (cuboids)
3D CSG Reconstruction

1. Generate primitives (cuboids)

2D CSG

3D point cloud → 2D CSG (floorplan) → 3D CSG model → Wall model → Final textured model
3D CSG Reconstruction

1. Generate primitives (cuboids)

2D CSG

Rectangle primitive
3D CSG Reconstruction

1. Generate primitives (cuboids)
2. Build a 3D CSG
Algorithm on Run

Step-by-step visualization of 3D CSG model reconstruction
Last Step

1. Remove Ceiling
2. Texture Mapping
Challenging Navigation

Frick Collection Gallery (New York City)

Goal

Start

Frick Collection Gallery (New York City)
# Statistics

<table>
<thead>
<tr>
<th></th>
<th>National Gallery</th>
<th>Frick</th>
<th>Tretyakov</th>
<th>Hermitage</th>
<th>Uffizi</th>
<th>Met</th>
</tr>
</thead>
<tbody>
<tr>
<td># of sub problems</td>
<td>1</td>
<td>8</td>
<td>19</td>
<td>20</td>
<td>48</td>
<td>32</td>
</tr>
<tr>
<td># of images</td>
<td>2,670</td>
<td>8,466</td>
<td>3,274</td>
<td>21,174</td>
<td>11,622</td>
<td>42,474</td>
</tr>
<tr>
<td># of laser points [million]</td>
<td>12.3</td>
<td>39.0</td>
<td>15.2</td>
<td>75.7</td>
<td>43.8</td>
<td>201.8</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>additive</th>
<th>subtractive</th>
<th></th>
<th>additive</th>
<th>subtractive</th>
<th></th>
<th>additive</th>
<th>subtractive</th>
<th></th>
<th>additive</th>
<th>subtractive</th>
</tr>
</thead>
<tbody>
<tr>
<td># of primitives</td>
<td></td>
<td>14</td>
<td>2</td>
<td></td>
<td>66</td>
<td>9</td>
<td></td>
<td>73</td>
<td>6</td>
<td></td>
<td>137</td>
<td>23</td>
</tr>
<tr>
<td># of triangles</td>
<td></td>
<td>1,050</td>
<td>4,962</td>
<td></td>
<td>3,830</td>
<td>6,902</td>
<td></td>
<td>18,882</td>
<td>22,610</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ground</td>
<td>Aerial</td>
<td>Ground+Aerial</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------</td>
<td>----------</td>
<td>----------</td>
<td>---------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Outdoor</td>
<td><img src="Image" alt="Google Streetview" /></td>
<td><img src="Image" alt="Google/Bing/NASA" /></td>
<td><img src="Image" alt="Google MapsGL" /></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Indoor</td>
<td><img src="Image" alt="Furukawa et al." /></td>
<td><img src="Image" alt="This paper" /></td>
<td><img src="Image" alt="This paper" /></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Summary

1. Large scale MVS for organized photos (Aerial photos)

2. Large scale MVS for unorganized photos (Internet community photos)

3. Large scale indoor modeling

• Divide -> Reconstruct -> Merge
Summary

1. Large scale MVS for organized photos (Aerial photos)

2. Large scale MVS for unorganized photos (Internet community photos)

3. Large scale indoor modeling

- Divide -> Reconstruct -> Merge

- Divide is a challenge for unorganized data
Summary

1. Large scale MVS for organized photos (Aerial photos)

2. Large scale MVS for unorganized photos (Internet community photos)

3. Large scale indoor modeling

- Divide -> Reconstruct -> Merge
- Divide is a challenge for unorganized data
- Merge is always a challenge (a real solution for indoors)
Thank You!

For slides and materials, check the tutorial website through our home pages.