Multiview Stereo

3D Vision
University of Illinois

Derek Hoiem

Many slides adapted from Lana Lazebnik, Steve Seitz, Yasu Furukawa, Noah Snavely
Creating a 3D model from multiple images

1. **Match** features across images

2. **SfM**: Solve for camera poses and 3D positions of matched features

3. **MVS**: Propose and verify 3D points by matching patches

4. **Meshing**: Fit a surface to the points
Today: Multi-view stereo

Given several images with known pose/parameters of the same object or scene, compute a representation of its 3D shape

Source: C. Hernandez, N. Snavely
Outline

• Applications and use cases

• Overview and key concepts

• COLMAP MVS algorithm

• State-of-art and open research
Applications
Applications
Applications
(Construction)

Point Cloud / Mesh

360 Photo Tour

Ortho or “Floor Map”

Surface Map
2-view stereo vs. multi-view stereo

2-view
• Provided two paired images
  – Nearly 100% overlap
  – Lighting usually consistent
• Solve for **discrete disparity** at each pixel
• Left/right consistency check

Multi-view
• Provided set of possible overlapping images
  – Must choose “source views”
  – Limited overlap with any view
  – Lighting often inconsistent
  – May be significant occlusions
• Solve for **continuous depth/normal** at each pixel
• Multiview consistency check
Mo views, mo problems

- How to choose source views (overall / per pixel)
- How to be robust to occlusion
- How to efficiently solve large continuous optimization problem
Multi-view stereo: Basic idea

Source: Y. Furukawa
Multi-view stereo: Basic idea

Source: Y. Furukawa
Multi-view stereo: Basic idea

Source: Y. Furukawa
Multi-view stereo: Basic idea

Source: Y. Furukawa
Multiview Stereo Algorithm

Input: Set of calibrated images, sparse correspondences

For each reference view:
1. Select “global” source views – candidate images for matching
2. For each iteration, for each pixel:
   i. Select pixel source views
   ii. Choose candidate depth/normal values
   iii. Compute/agregate photometric scores (e.g. NCC) between reference and source views
   iv. Select candidate with best score

Filter inconsistent depth values and fuse depth maps

Output: 3D point cloud
Source view selection challenge

Cameras 4 and 5 can more clearly see point p.

Source: N. Snavely
Cameras 3 and 4 can more clearly see point q.

Source: N. Snavely
Source view selection challenge

Camera 5 can’t see point r.

Source: N. Snavely
Source view selection challenge

Camera 1 can’t see point \( s \).

Source: N. Snavely
Principles for choosing source views

Choose 3 best views overall
Principles for choosing source views

• Sparse points in common
Principles for choosing source views

- Sparse points in common
- Moderate angle between cameras and point
Principles for choosing source views

- Sparse points in common
- Moderate angle between cameras and point
- Similar distance as reference
Principles for choosing source views

- Sparse points in common
- Moderate angle between cameras and point
- Similar distance as reference
- Not too oblique to the surface
Principles for choosing source views

• Sparse points in common
• Moderate angle between cameras and point
• Similar distance as reference
• Not too oblique to the surface
• Not redundant with other views
Principles for choosing source views

- Sparse points in common
- Moderate angle between cameras and point
- Similar distance as reference
- Not too oblique to the surface
- Not redundant with other views

Pixelwise
- Angle, orientation, distance priors
- Good photometric score with reference (tricky because we aren’t sure of depth yet)
- Views for nearby pixels should usually be the same
Approaches for depth/normal candidate generation
Plane Sweep

- Sweep family of planes at different depths w.r.t. a reference camera
- For each depth, project each input image onto that plane (homography) and compare the resulting stack of images

R. Collins, A space-sweep approach to true multi-image matching, CVPR 1996
Plane sweep stereo

Scene surface

Sweeping plane

Image 1

Image 2
Region growing

1. Propagate along plane (defined by depth/normal) from pixel with good photometric score to its neighbors
Region growing

1. Propagate along plane (defined by depth/normal) from pixel with good photometric score to its neighbors

2. Accept, reject, or refine by gradient descent
Region growing

1. Propagate along plane (defined by depth/normal) from pixel with good photometric score to its neighbors

2. Accept, reject, or refine by gradient descent

3. Add to queue for expansion if accepted
Neighboring pixels are often co-planar, even when depth varies.

Plane equation:

\[ ax + by + cz + d = 0 \]

Normal:
\[ [a \ b \ c] \]

Plane distance:
\[ d \]
Patch match

Initialize by randomly assigning depth/normal values to each pixel

For each iteration, for each pixel:
1. Sample depth/normal candidates from pixel neighborhood, randomly perturb current hypothesis
2. Select candidate with best photometric score

Why it works:
1. “labels” (plane candidates) are piecewise constant
2. High chance of at least one pixel being assigned a good label; then it propagates quickly
## Pros and cons of candidate generation approaches

<table>
<thead>
<tr>
<th>Plane sweep</th>
<th>Region growing</th>
<th>Patch match</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓ Simple, easy to optimize</td>
<td>✓ Few candidates to evaluate</td>
<td>✓ Does not rely on initial depth estimates</td>
</tr>
<tr>
<td>✓ Can refine post-process on cost volume</td>
<td>✓ Allows precise depth+normal estimates</td>
<td>✓ Allows precise depth+normal estimates</td>
</tr>
<tr>
<td>✓ Can refine post-process on cost volume</td>
<td>✓ Acts as implicit prior that can complete smooth surfaces</td>
<td>✓ Can be solved efficiently using GPU</td>
</tr>
<tr>
<td>- True surface may be different orientation, resulting in poor estimates</td>
<td>- Relies on initial depth estimates for some pixels</td>
<td>- Efficient + effective propagation requires more complex schemes</td>
</tr>
<tr>
<td>- Fixed number of depth estimates may provide coarse estimation or high memory/compute cost</td>
<td>- Incorrect estimates can propagate to create incorrect regions</td>
<td>- Thin structures and small surfaces may be missed</td>
</tr>
</tbody>
</table>
How to compute photometric scores given depth/normal candidate

1. Compute NCC with each source image (illustrate patch projection on the board)

2. Sum NCC values, possibly weighted by confidence that each source image views the same scene point
Outline

• Applications and use cases

• Overview and key concepts

• COLMAP MVS algorithm

• State-of-art and open research
Start with SfM result
- Known camera poses and intrinsic parameters
- “Sparse” 3D points with links to corresponding views
Pixelwise view priors
Select global view candidates

- In paper’s experiments, all images used view candidates
- In practice, typically ~20 images for each reference image that have significant triangulation angle, similar resolution, similar angle, computed as sum over sparse points

View Priors
Photometric scores

• Bilaterally weighted NCC on patches (e.g. 25x25)
  – NCC normalizes for mean/std intensity
  – Bilateral weighting gives more weight to pixels close to and with similar color to the center pixel

\[
\rho_l^m = \frac{\text{cov}_w(w_l, w_l^m)}{\sqrt{\text{cov}_w(w_l, w_l) \cdot \text{cov}_w(w_l^m, w_l^m)}}
\]

• Mapped into probability

\[
P(X_l^m | Z_l^m, \theta_l) = \begin{cases} 
\frac{1}{NA} \exp \left( - \frac{(1-\rho_l^m(\theta_l))^2}{2\sigma^2_\rho} \right) & \text{if } Z_l^m = 1 \\
\frac{1}{N} U & \text{if } Z_l^m = 0,
\end{cases}
\]
PatchMatch candidates

- Each iteration, sweep in 4 directions (left, up, right, down)
- Generate candidates from previous pixel

\[
\{(\theta_i, n_i), (\theta_{i-1}^{\text{prp}}, n_{i-1}), (\theta_i^{\text{rnd}}, n_i), (\theta_i, n_i^{\text{rnd}}), (\theta_i^{\text{rnd}}, n_i^{\text{rnd}}), (\theta_i^{\text{prp}}, n_i), (\theta_i, n_i^{\text{prp}})\}
\]
Algorithm

1. Select global candidates (for efficiency)
2. Solve probability that each pixel is observed by each source image, given current depth/normal estimate
   - Combining view prior, photometric score, and spatial and “temporal” (keep same as last iteration) smoothness in MRF formulation
3. Solve depth/normal for each pixel via PatchMatch candidate generation/scoring, given view probability
   - Score is patchwise bilateral NCC, summed over source images, weighted by view probability
   - In later iterations, score also accounts for geometric consistency with depth estimated from source views (illustrate on board)
4. Fusion / filtering
Filtering

- Keep depth values if
  - Photometric score high enough
  - Geometrically consistent depth/normal with at least N source images
- Compute centroid of similar points and combine points from all images
## Results

<table>
<thead>
<tr>
<th></th>
<th>[59]</th>
<th>[24]</th>
<th>[13]</th>
<th>[57]</th>
<th>[51]</th>
<th>[26]</th>
<th>[15]</th>
<th>\N</th>
<th>\P</th>
<th>\S</th>
<th>\B</th>
<th>\PSB</th>
<th>\G</th>
<th>Ours</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Fountain</strong></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></td>
<td><strong>0.827</strong></td>
</tr>
<tr>
<td>2cm</td>
<td>0.769</td>
<td>0.754</td>
<td>0.731</td>
<td>0.712</td>
<td>0.732</td>
<td>0.824</td>
<td>0.693</td>
<td>0.799</td>
<td>0.824</td>
<td>0.825</td>
<td>0.826</td>
<td>0.817</td>
<td>0.804</td>
<td><strong>0.827</strong></td>
</tr>
<tr>
<td>10cm</td>
<td>0.929</td>
<td>0.930</td>
<td>0.838</td>
<td>0.832</td>
<td>0.822</td>
<td>0.973</td>
<td>0.838</td>
<td>0.937</td>
<td>0.972</td>
<td>0.973</td>
<td>0.973</td>
<td>0.965</td>
<td>0.949</td>
<td><strong>0.975</strong></td>
</tr>
<tr>
<td><strong>Herzjesu</strong></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></td>
<td><strong>0.931</strong></td>
</tr>
<tr>
<td>2cm</td>
<td>0.650</td>
<td>0.649</td>
<td>0.646</td>
<td>0.220</td>
<td>0.658</td>
<td><strong>0.739</strong></td>
<td>0.283</td>
<td>0.673</td>
<td>0.686</td>
<td>0.688</td>
<td>0.690</td>
<td>0.688</td>
<td>0.679</td>
<td>0.691</td>
</tr>
<tr>
<td>10cm</td>
<td>0.844</td>
<td>0.848</td>
<td>0.836</td>
<td>0.501</td>
<td>0.852</td>
<td>0.923</td>
<td>0.455</td>
<td>0.901</td>
<td>0.928</td>
<td>0.927</td>
<td>0.929</td>
<td>0.921</td>
<td>0.907</td>
<td><strong>0.931</strong></td>
</tr>
</tbody>
</table>
Benchmarks for MVS

- Middlebury: https://vision.middlebury.edu/mview
- DTU: https://roboimagedata.compute.dtu.dk
- Tanks and Temples: https://www.tanksandtemples.org/
- ETH3D: https://www.eth3d.net/
Recent advances

- COLMAP: as described
- ACMH: like COLMAP, with adaptive propagation for PatchMatch
- ACMM: ACMH + multiscale
- ACMP: ACMM + plane-based completion of smooth regions
- DeepC-MVS: ACMM + deep confidence filtering
- TV Prior: Volumetric surface reconstruction with total variation prior (not PatchMatch based)
Important GPU speedup: red-black scheme

• Instead of sweeping, update all “red” pixels or all “black” pixels at once

• Much faster for GPU because massively parallel
  – Sweeps take $O(\text{width} + \text{height} + \text{candidates})$ when parallelized
  – Red-black takes $O(\text{candidates})$ when parallelized

Galliani et al. 2016 (Gipuma)
<table>
<thead>
<tr>
<th>Method</th>
<th>Info</th>
<th>all</th>
<th>high-res-</th>
<th>multi-view</th>
<th>indoor</th>
<th>outdoor</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ETH3D High Res</strong> – F1 Score</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>Info</th>
<th>all</th>
<th>high-res-</th>
<th>multi-view</th>
<th>indoor</th>
<th>outdoor</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeepC-MVS fast</td>
<td></td>
<td>79.65</td>
<td>86.91</td>
<td>86.62</td>
<td>87.76</td>
<td>93.52</td>
</tr>
<tr>
<td>ACMP</td>
<td></td>
<td>74.13</td>
<td>81.51</td>
<td>80.57</td>
<td>84.36</td>
<td>89.42</td>
</tr>
<tr>
<td>Qingshan Xu and Wenbing Tao</td>
<td>Planar Priors Assisted PotMat: Multi-View Stereo. AAAI 2020</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ACMM</td>
<td></td>
<td>73.20</td>
<td>80.78</td>
<td>79.84</td>
<td>83.58</td>
<td>89.31</td>
</tr>
<tr>
<td>Qingshan Xu and Wenbing Tao</td>
<td>Multi-Scale Geometric Consistency Guided Multi-View Stereo. CVPR 2019</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PCF-MVS</td>
<td></td>
<td>73.52</td>
<td>80.38</td>
<td>78.84</td>
<td>85.01</td>
<td>87.71</td>
</tr>
<tr>
<td>Andreas Kuhn, Shan Lin, Oliver Erleler, Plane Completion and Filling for Multi-View Stereo Reconstruction. GCPR 2016</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PLC</td>
<td>C</td>
<td>70.83</td>
<td>78.05</td>
<td>76.37</td>
<td>83.08</td>
<td>91.88</td>
</tr>
<tr>
<td>Jia Liao, Yingying Fu, Chingan Yan, Chunhua Shen: Pyramid Multi-View Stereo with Local Consistency. Pacific Graphics 2019</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ACMH</td>
<td></td>
<td>67.68</td>
<td>75.89</td>
<td>73.93</td>
<td>81.77</td>
<td>88.71</td>
</tr>
<tr>
<td>Qingshan Xu and Wenbing Tao</td>
<td>Multi-Scale Geometric Consistency Guided Multi-View Stereo. CVPR 2019</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>COLMAP ROB</td>
<td>C</td>
<td>66.92</td>
<td>73.01</td>
<td>70.41</td>
<td>80.81</td>
<td>87.13</td>
</tr>
</tbody>
</table>
Open Research Problems

• Improve speed and completeness

• Reduce number of views required for good results
  – Typically 3-5 views per scene point is currently required

• Produce more structured models, e.g. directly solve for surfaces rather than points
Summary

• Multiview stereo brings several new problems: selecting source views per pixel, solving for continuous normal/depth, filtering

• There are many algorithms and optimizations strategies – I focused on the currently dominant family of algorithms

• As with two-view stereo, main challenges are textureless, reflective, and transparent surfaces, and boundaries
Next class

• “Classic” MVS papers
• 8 groups
• Read paper and submit individual review. Can discuss with group, but write your review on your own.
• Discuss with your group before class Tues and make 1-2 summary slides – put in same Google Slides as before
• I will choose order based on content of papers and give brief introduction
• Presentations are 5-6 minutes + 2 min for question/comment