Cluster-based 3D Reconstruction of Aerial Video

Scott Sawyer (scott.sawyer@ll.mit.edu)
MIT Lincoln Laboratory

HPEC ’12
12 September 2012

This work is sponsored by the Assistant Secretary of Defense for Research and Engineering under Air Force contract FA8721-05-C-0002. Opinions, interpretations, conclusions, and recommendations are those of the author and are not necessarily endorsed by the United States Government.
Given a collection of photos...
...determine their 3D structure
...determine their 3D structure
Outline

• Feature-based 3D Reconstruction
• Reconstructing Aerial Video
• Cluster-based Structure from Motion
• Results
Outline

- Feature-based 3D Reconstruction
  - Reconstructing Aerial Video
  - Cluster-based Structure from Motion
  - Results
“Structure from Motion” (SfM)

Photo Collection → Feature Extraction → Feature Matching → Sparse Reconstruction → Dense Reconstruction → 3D Point Cloud

Applications:

3D Maps

Robotic Navigation

Video Geo-registration
Feature Extraction

Photo Collection → Feature Extraction → Feature Matching → Sparse Reconstruction → Dense Reconstruction → 3D Point Cloud

Scale Invariant Feature Transform (SIFT)

- Identifies distinctive keypoints in an image and provides 128-dimensional descriptor vector
- Based on finding local extrema in scale space
- Invariant to scale, orientation and lighting conditions (to an extent) and aspect angles up to 30 degrees
- Published by David Lowe in 1999 and widely used in computer vision
Euclidean Distance Approximate Nearest Neighbors (ANN)

- Euclidean distance between SIFT keypoint descriptors used to determine matches
- Tree-based approximation reduces computation
- SIFT points matched between each pair of images – $O(n^2)$
Sparse Reconstruction

Projective Structure from Motion

- Given \( m \) images and \( n \) 3D points (with point correspondences)
- Estimate \( m \) projection matrices \((P_i, i = 1, \ldots, m)\) and \( n \) 3D points \((X_j, j = 1, \ldots, n)\) from the \( mn \) correspondences \((x_{ij})\)
- Iteratively minimize re-projection error ("bundle adjustment"): 
  \[
  E(P, X) = \sum_{i=1}^{m} \sum_{j=1}^{n} D(x_{ij}, P_iX_j)^2
  \]
Identify Patches of Rigid Structure

- Mature software ("PMVS2") forms patched-based reconstruction (open source, parallel)
- Removes spurious and non-rigid points
- This step not included in benchmark analysis
Outline

• Feature-based 3D Reconstruction
• Reconstructing Aerial Video
• Cluster-based Structure from Motion
• Results
Lincoln Laboratory performed a campaign of aerial data collections in 2011:

- Aerial platform circles ground subject of interest
- Camera gimbal remains fixed on ground point
- Video frames synchronized to GPS
- Challenge: Processing must keep up with rate of collection (e.g. one-hour mission timeline)
Sample Video
Bundler (single-core)

- Works on unordered image collections (no assumptions about input)
- Sequential implementation (single-core)
- Optimization solution considered baseline (truth)

Bundler (v. 0.4) by Noah Snavely

Image Set Size vs. Runtime (hours)

- Feat. Extr.
- Feat. Match
- Bundle Adj.
VisualSFM (v. 0.5.15) by Changchang Wu

- Restructures optimization to better enable parallelization of dense matrix kernels
- Feature extraction and bundle adjustment use GPU acceleration (CUDA)
- Multi-threaded feature matching (pThreads)

Run on quad-core Intel Xeon E5620 (2.4 GHz), 6 GB RAM, nVIDIA Quadro FX1800 (64 compute cores)
Outline

• Feature-based 3D Reconstruction
• Reconstructing Aerial Video
• Cluster-based Structure from Motion
• Results
LLGrid Computing Environment

Job submission

Scheduler assigns each job to a single core (no threading or shared memory)

Jobs may or may not be on the same node

Scheduler

Master Node

Network File System (Lustre)

Compute Nodes

Custom Python framework for submitting jobs, mapping data and synchronizing between steps

Inter-process communication supported via file system or network

Number of cores used varied experimentally up to 64
Cluster-based SfM Overview

1. Feature Extraction
   - Job 1
   - Job 2
   - ... Job N_p
   - Remap
   - SIFT keypoints
   - Runtime Growth: $O(N)$

2. Feature Matching
   - Job 1
   - Job 2
   - ... Job N_p
   - Remap
   - Matched Feature Graph
   - Runtime Growth: $O(N^2)$

3. Reconstruction
   - Job 1
   - ... Job M
   - Remap
   - Partial Point Clouds
   - Runtime Growth: $O(N^2/M)$

4. Point Cloud Fusion
   - Job 1
   - Job 2
   - ... Job N_p
   - Remap
   - Single Job
   - Final Point Cloud
   - Runtime Growth: $O(P^2/M)$

Combine Results

N: Input photo count; M: Partition count; P: Sparse point count
Feature Matching

Matching Pair Matrix

<table>
<thead>
<tr>
<th>Photo ID</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>...</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>...</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>...</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>...</td>
</tr>
<tr>
<td>N-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>N</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Matching Graph

(1,10) - (2,12) - (3,15) - (4,75) - (1,50) - (5,81)

Vertices described by tuple (Photo ID, Keypoint ID)

If \((i,j) = 1\), then match photo \(i\) to photo \(j\)

- Matching is most expensive step, growing \(O(N^2)\) with number of photos
- Results of matching represented as graph \(G\):
  - Vertices \(V\) represent SIFT points
  - Edges \(E\) represent SIFT matches
Flight path partitioned using block-cyclic mapping:

- $N$ photos mapped to $M$ partitions
- Assume photo set covers one revolution
- Angular gap between photos should be less than 30 degrees (to help with SIFT matching)
- Photos per partition ($N/M$) must be greater than 25
- Block size and partition count set automatically

Reconstructing partial photo sets independently creates challenges:

- Point clouds are in different arbitrary coordinate systems
- Many points are duplicated across the partial point clouds
- 3D points have shorter view lists (fewer iterative improvements)

Example: Block size: 2; $M=3$
Identifying Duplicate Points

Each 3D point has a view list comprised of SIFT keypoints (vertices in G). View lists from different partitions will always be disjoint because photos are mapped to exactly one partition.

3D points are considered duplicates if any vertices in B are in the neighborhood of A.
Transforming to a Common Space

Determine rotation, scale and translation based on duplicate points:

- Robustly eliminate outlier 3D points
- Solve for transformation matrix that minimizes error
- Combine duplicate points as a weighted average

Sparse point clouds after independent bundle adjustment on 4 partitions

Fused sparse point cloud after transformation
Outline

• Feature-based 3D Reconstruction
• Reconstructing Aerial Video
• Cluster-based Structure from Motion
• Results
Speed-Up & Scalability

Cluster-based SfM Speed-up

Cluster SfM Scalability (400 photos)

<table>
<thead>
<tr>
<th>Cluster Size</th>
<th>8</th>
<th>16</th>
<th>32</th>
<th>64</th>
</tr>
</thead>
<tbody>
<tr>
<td>Partitions</td>
<td>8</td>
<td>16</td>
<td>16</td>
<td>16</td>
</tr>
<tr>
<td>Frame rate (per min.)</td>
<td>1.2</td>
<td>2.4</td>
<td>4.3</td>
<td>7.6</td>
</tr>
<tr>
<td>Points per Frame (x10^3)</td>
<td>2.0</td>
<td>.87</td>
<td>.87</td>
<td>.87</td>
</tr>
</tbody>
</table>
Reconstruction Quality

RMS Error of Reconstructed Points (meters)

<table>
<thead>
<tr>
<th>Input size (N)</th>
<th>100</th>
<th>200</th>
<th>300</th>
<th>400</th>
</tr>
</thead>
<tbody>
<tr>
<td>VisualSFM</td>
<td>0.20</td>
<td>7.15</td>
<td>1.05</td>
<td>0.44</td>
</tr>
<tr>
<td>Cluster-based SfM</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4 Partitions</td>
<td>0.10</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8 Partitions</td>
<td>0.27</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12 Partitions</td>
<td>0.26</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16 Partitions</td>
<td>0.24</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Sparse Point Cloud Density

- Bundler
- VisualSFM
- Cluster (N/M=25)
Summary

- Creating 3D reconstructions of photo collections scales well to the cluster
  - 64-core cluster shows 14x speedup over multi-core workstation with GPU
- Our algorithm effectively parallelizes reconstruction for the case of aerial video around a ground point
- For algorithms that aren’t “pleasantly parallel”, deep knowledge or application and computing environment necessary to parallelize
- Future work:
  - Reduce feature matching complexity for different types of input sets
  - Point cloud geo-registration by matching against reference data (e.g. LIDAR or satellite imagery)
Scott Sawyer (scott.sawyer@ll.mit.edu)

SIGMA Team:
• Karl Ni
• Alex Vasile
• Nick Armstrong-Crews
• Ethan Phelps
• Nadya Bliss
• Prof. Noah Snavely (Cornell University)