Towards Linear-time Incremental Structure from Motion

Changchang Wu
University of Washington

---

**How Fast is Bundle Adjustment?**
- PCG spent most of time on matrix-vector multiplication, where
  - The time complexity of a single CG iteration is $O(n)$
  - The number of CG iterations depend on the condition number of the problems.

- BA can be done in linear time with truncation (max CG/LM iterations)

**Re-Triangulation (RT)**
- Revisits feature matches in a geometric sequence, which takes $O(n)$ time.
- Reduces accumulated drifts and allows for implicit loop-closing.

**Time Complexity**
- Still $O(n^2)$ due to linear scan for the partial BA and filtering.
- Large constant factor for the $O(n)$ portion for 15K cameras.

---

**Bundle Adjustment Strategy**
- There is no need to bundle adjust the entire model for ever image.
  - With a linear sequence: full BA when $n$ increases by $\alpha$
  - With a geometric sequence:
    - full BA when $n$ increases relatively by $r$
  - Perform a partial BA when not performing full BA, which adds to $O(n)$.
  - Point filtering also requires only $O(n)$ thanks to the geometric sequence.

**Reconstruction Summary**

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Image</th>
<th>Camera</th>
<th>Portion</th>
<th>Observations</th>
<th>Time/Ball (min)</th>
<th>Time/Partial</th>
<th>Time/Adding</th>
<th>Time/Filtering</th>
</tr>
</thead>
<tbody>
<tr>
<td>Central Rome</td>
<td>65191</td>
<td>80425</td>
<td>100</td>
<td>72408338</td>
<td>20:40</td>
<td>100</td>
<td>200</td>
<td>500</td>
</tr>
<tr>
<td>Art Quad</td>
<td>65191</td>
<td>80425</td>
<td>45</td>
<td>72408338</td>
<td>4:26</td>
<td>100</td>
<td>200</td>
<td>500</td>
</tr>
<tr>
<td>Loop</td>
<td>65191</td>
<td>80425</td>
<td>20</td>
<td>72408338</td>
<td>1:17</td>
<td>100</td>
<td>200</td>
<td>500</td>
</tr>
<tr>
<td>Colosseum</td>
<td>1164</td>
<td>1175</td>
<td>20</td>
<td>72408338</td>
<td>591</td>
<td>452</td>
<td>100</td>
<td>19</td>
</tr>
</tbody>
</table>

**Evaluation and Comparisons**

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Full BA</th>
<th>Partial BA</th>
<th>RT</th>
<th>n</th>
<th>t</th>
<th>l</th>
<th>f</th>
<th>p</th>
</tr>
</thead>
<tbody>
<tr>
<td>Central Rome</td>
<td>65191</td>
<td>80425</td>
<td>100</td>
<td>72408338</td>
<td>20:40</td>
<td>100</td>
<td>200</td>
<td>500</td>
</tr>
<tr>
<td>Art Quad</td>
<td>65191</td>
<td>80425</td>
<td>45</td>
<td>72408338</td>
<td>4:26</td>
<td>100</td>
<td>200</td>
<td>500</td>
</tr>
<tr>
<td>Loop</td>
<td>65191</td>
<td>80425</td>
<td>20</td>
<td>72408338</td>
<td>1:17</td>
<td>100</td>
<td>200</td>
<td>500</td>
</tr>
<tr>
<td>Colosseum</td>
<td>1164</td>
<td>1175</td>
<td>20</td>
<td>72408338</td>
<td>591</td>
<td>452</td>
<td>100</td>
<td>19</td>
</tr>
</tbody>
</table>

**Significant reduction of the matching cost**

- Precise feature matching
  - Sort the features in decreasing-scale order for each image.
  - Find the image pairs that need to be matched (VT, GPS, etc...).
  - Match the first $h$ features for each pair. Let $m_i(h)$ be the number of matches.
  - Standard feature matching for pairs that satisfy $m_i(h) \geq t_i$

**Related “Truncations”**
- Image matching using Vocabulary Tree and ANN, and filtering by GPS,
- Scene graph simplification (Skeletal graph or Iconic images),
- Bundle adjustment using Pre-conditioned Conjugated Gradient (PCG)

**Incremental Structure from Motion**

- Add Camera
- Partial BA
- Filter
- Re-triangulation & Full BA

---

**VisualSFM**

- Available as part of VisualSFM:
  - http://homes.cs.washington.edu/~ccwu/vsfm/
  - or http://ccwu.me/vsfm

---

**Contributions**
- We show that many sub-steps of incremental SIM, including BA and point filtering, require only $O(n)$ time in practice when using a geometric BA strategy.
- Without sacrificing the time-complexity, we introduce a re-triangulation step to deal with the problem of accumulated drifts without explicit loop closing.
- A simple preemptive feature matching for reducing image matching cost.

---

**Preemptive Feature Matching**
1. Sort the features in decreasing-scale order for each image.
2. Find the image pairs that need to be matched (VT, GPS, etc...).
3. Match the first $h$ features for each pair. Let $m_i(h)$ be the number of matches.
4. Standard feature matching for pairs that satisfy $m_i(h) \geq t_i$

---

**Evaluation**

- Significant reduction of the matching cost
- Precision feature matching
- Sort the features in decreasing-scale order for each image.
- Find the image pairs that need to be matched (VT, GPS, etc...).
- Match the first $h$ features for each pair. Let $m_i(h)$ be the number of matches.
- Standard feature matching for pairs that satisfy $m_i(h) \geq t_i$