Bundled Depth-Map Merging for Multi-View Stereo

Jianguo Li, Eric Li, Yurong Chen, Lin Xu, Yimin Zhang
Intel Labs China
CVPR 2010
Outline

- Motivations
- Technique details
- Results
- Conclusions
Motivation

• MVS is crowded, but
  – Fast and accurate algorithm is still meaningful.
  – Application gamut not well studied

• Algorithms into several tech-categories
  – depth-map merging is typical and extensible; but
    • Requires sub-pixel stereo matching precision for accuracy
    • Challenges on merging inaccurate depth maps

• This paper proposes a bundle optimization based depth-map merging framework
Framework

Stereo matching

Bundle optimization
Stereo pair selection

- Not all image/stereo pairs are useful
  - Due to shear effect etc
- Based on three rules
  - Angle between view directions
    - In the range of 5~45
  - Distance between optical center
    - Remove too large or too small
  - Each view at most two pairs
Stereo matching

- Rectify stereo pairs
- For each pixel in a scan line of the left image
  - Find correspondence in the right image
  - Based on DAISY features
  - Using kd-tree to accelerate the search
  - Each pixel returns top-K (=10) results
  - Scan-line optimization to keep intra-line order consistency
- Keep reliable matches
Track building and optimization

Track building
- Tracks: connected matching pixels from $\geq 3$ views
- Depth-first search to find the tracks among stereo-pairs

stereo pair stereo pair stereo pair
Bundled track optimization (1)

• Track optimization
  - A track => the same 3D point
  - Find: 3D points $X$ such that
    $$\min \sum_{k \in t_k} w(x^k_i) \| x^k_i - P^k_i \hat{X}^k \|,$$
    where
    $$w(x^k_i) = \begin{cases} 
      1 & \text{if } \| x^k_i - P^k_i \hat{X}^k \| < \gamma, \\
      10 & \text{otherwise.}
    \end{cases}$$
  - Keep eligible tracks
    • reprojection error less than $\gamma$
    • $\beta \geq 3$ views
Bundled optimization (2)

- Refine the point position and estimate point normals with bounded optimization by photo-consistency (using L-BFGS-B)

\[ E_k = \sum_{i,j \in t_k} \| DF_i(x) - DF_j(H_{ij}(x; n, d)) \|, \]

\[
\begin{align*}
\min & \quad E_k, \\
\text{s.t.} & \quad (1) |\hat{x}_i - x_i| < \gamma, \\
& \quad (2) n_{ij} \ast \frac{X_{cam}^i O_i}{\|X_{cam}^i O_i\|} > 0.5,
\end{align*}
\]

- Normal initialization
  - Initial normal with plane fitting or
  - Grid search from point view direction

- Benefits from bounded optimization
  - Less function evaluation
  - Fast convergence
Implementation

- CPU: multi-core, core i7
- Code: all in C/C++
- Fine-grained code optimization
  - SIMD for hotspots
    - Kdtree, DAISY, etc
  - Data-level parallelization
    - Scanline in stereo match
    - Tracks in track-opt
    - 3D Points in refinement
- A typical 32, 2Mpix dataset, took less than 3min
Results on Middlebury

- Performances:
  - Very high accuracy:
    - dinoRing result ranked within top 5
    - Comparable results on templeRing
  - One of the most efficient methods among non-GPU methods

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Accuracy</th>
<th>Completeness</th>
<th>Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>dinoRing</td>
<td>0.43mm</td>
<td>99.7%</td>
<td>354</td>
</tr>
<tr>
<td>templeRing</td>
<td>0.64mm</td>
<td>98.2%</td>
<td>213</td>
</tr>
</tbody>
</table>

Note:
- During writing, use a 8-core core2duo.
- Better performance with Core i7.
Results on Real Datasets

Monster
34, 2Mpix
~25cm high
Complex surface

SculptFace
14, 6Mpix
~1m high
Highlight/metal surface
More Results

NikeShoe: 26, 4M pixel photos
Gum, Leather & Cloth, 23cm

Lion: 17, 6M pixel
Iron, ~1.2m high
Large highlight area

Reconstructed surface
Textured model
Results from video input

Input: ~1min HD video around a marble lion (Height/Width: ~2m/1m)

Reconstructed surface

Textured model
Conclusions

• Contributions
  – A bundle optimization framework for fast and accurate depth-map merging
  – Study MVS on complex/non-Lambertian/highlight surface
  – Introduce DAISY for MVS (both stereo matching and photo consistency optimization)
  – Parallel processing friendly and scalable well

Dataset available at http://sites.google.com/site/leeplus/bmvs/
Thank you!