High-Resolution Stereo Matching

Sudipta N. Sinha

Interactive Visual Media Group
Microsoft Research
Stereo matching

- Input: rectified image pair
- Output: disparity map

SGM [Hirschmuller 2005]

GT

6% errors $|\Delta d| \leq 1$ pixel in non-occluded regions
Stereo – applications

• Dense 3D reconstruction
• Robot navigation
• Automated driving
• Gaming, user interfaces
• Virtual viewpoint correction
• 3D movie editing
High-resolution Cameras

- High resolution cameras are everywhere
- 8+ MP on most commodity camera phones
- Impact on stereo matching techniques
  - Accuracy
  - Speed

Nokia Lumia 1020
(41 MP sensor)
Outline

• Stereo Matching
  - State of the art
  - High-resolution and Scalability

• High-Resolution Stereo Matching using Local Plane Sweeps

• Surface-based stereo matching

• Multi-view Stereo + Photometric Stereo
Middlebury Stereo benchmark

Images up to 450 x 375 (< 0.2 MP), D = 16 ... 60

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Avg.</th>
<th>Error Threshold = 1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Error Threshold...</td>
</tr>
<tr>
<td></td>
<td></td>
<td>▼</td>
</tr>
<tr>
<td>4C Census [52]</td>
<td>10.9</td>
<td>1.07 14 1.48 13 5.73 19</td>
</tr>
<tr>
<td>AdaLinBF [116]</td>
<td>14.2</td>
<td>1.11 18 1.37 8 5.79 21</td>
</tr>
<tr>
<td>CoopRegion [98]</td>
<td>14.8</td>
<td>0.87 4 1.16 19 4.51 46</td>
</tr>
<tr>
<td>PDP [67]</td>
<td>19.2</td>
<td>0.97 2 1.39 10 5.00 9</td>
</tr>
<tr>
<td>MultiSBE [129]</td>
<td>19.8</td>
<td>1.32 41 1.56 17 6.02 28</td>
</tr>
<tr>
<td>DoubleBF [34]</td>
<td>20.0</td>
<td>0.69 12 1.29 21 4.75 15</td>
</tr>
<tr>
<td>MDPM [140]</td>
<td>20.3</td>
<td>1.12 18 1.60 22 6.14 31</td>
</tr>
<tr>
<td>OutlierCon [160]</td>
<td>20.5</td>
<td>0.69 12 1.43 17 4.74 8</td>
</tr>
<tr>
<td>Adaboost [117]</td>
<td>24.3</td>
<td>0.73 20 1.53 14 5.26 24</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Average percent of bad pixels (explanation)</td>
</tr>
</tbody>
</table>
## KITTI benchmark

Image size 1241 x 376 (< 0.5 MP), D ≈ 70...150

### The KITTI Vision Benchmark Suite
A project of Karlsruhe Institute of Technology and Toyota Technological Institute at Chicago

Andreas Geiger (MPI Tübingen) | Phillip Lenz (KIT) | Christoph Stiller (KIT) | Raquel Urtasun (University of Toronto)

### Stereo Evaluation

<table>
<thead>
<tr>
<th>Rank</th>
<th>Method</th>
<th>Setting Code</th>
<th>Out-Noc</th>
<th>Out-All</th>
<th>Avg-Noc</th>
<th>Avg-All</th>
<th>Density</th>
<th>Runtime</th>
<th>Environment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SceneFlow</td>
<td>⊗/⊕/⊗</td>
<td>2.98 %</td>
<td>3.97 %</td>
<td>0.8 px</td>
<td>1.0 px</td>
<td>100.00 %</td>
<td>35 s</td>
<td>1 core @ 3.5 Ghz (C/C++)</td>
</tr>
</tbody>
</table>

Anonymous submission

<table>
<thead>
<tr>
<th>Rank</th>
<th>Method</th>
<th>Setting Code</th>
<th>Out-Noc</th>
<th>Out-All</th>
<th>Avg-Noc</th>
<th>Avg-All</th>
<th>Density</th>
<th>Runtime</th>
<th>Environment</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>PCP-SS</td>
<td>⊗/⊕/⊗</td>
<td>3.40 %</td>
<td>4.72 %</td>
<td>0.8 px</td>
<td>1.0 px</td>
<td>100.00 %</td>
<td>5 min</td>
<td>4 cores @ 2.5 Gz (Matlab + C/C++)</td>
</tr>
</tbody>
</table>


<table>
<thead>
<tr>
<th>Rank</th>
<th>Method</th>
<th>Setting Code</th>
<th>Out-Noc</th>
<th>Out-All</th>
<th>Avg-Noc</th>
<th>Avg-All</th>
<th>Density</th>
<th>Runtime</th>
<th>Environment</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>gTIF-SS</td>
<td>⊗/⊕/⊗</td>
<td>3.83 %</td>
<td>4.59 %</td>
<td>0.9 px</td>
<td>1.0 px</td>
<td>100.00 %</td>
<td>1 min</td>
<td>1 core @ 3.5 Ghz (Matlab + C/C++)</td>
</tr>
</tbody>
</table>

Anonymous submission

<table>
<thead>
<tr>
<th>Rank</th>
<th>Method</th>
<th>Setting Code</th>
<th>Out-Noc</th>
<th>Out-All</th>
<th>Avg-Noc</th>
<th>Avg-All</th>
<th>Density</th>
<th>Runtime</th>
<th>Environment</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>StereoSLIC</td>
<td>⊗/⊕/⊗</td>
<td>3.92 %</td>
<td>5.11 %</td>
<td>0.9 px</td>
<td>1.0 px</td>
<td>99.89 %</td>
<td>2.3 s</td>
<td>1 core @ 3.0 Ghz (C/C++)</td>
</tr>
</tbody>
</table>


<table>
<thead>
<tr>
<th>Rank</th>
<th>Method</th>
<th>Setting Code</th>
<th>Out-Noc</th>
<th>Out-All</th>
<th>Avg-Noc</th>
<th>Avg-All</th>
<th>Density</th>
<th>Runtime</th>
<th>Environment</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>PR-SLF</td>
<td>⊗/⊕/⊗</td>
<td>4.02 %</td>
<td>4.87 %</td>
<td>0.9 px</td>
<td>1.0 px</td>
<td>100.00 %</td>
<td>200 s</td>
<td>4 cores @ 3.0 Gz (Matlab + C/C++)</td>
</tr>
</tbody>
</table>


<table>
<thead>
<tr>
<th>Rank</th>
<th>Method</th>
<th>Setting Code</th>
<th>Out-Noc</th>
<th>Out-All</th>
<th>Avg-Noc</th>
<th>Avg-All</th>
<th>Density</th>
<th>Runtime</th>
<th>Environment</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>PGBP</td>
<td>⊗/⊕/⊗</td>
<td>4.04 %</td>
<td>5.37 %</td>
<td>0.9 px</td>
<td>1.1 px</td>
<td>100.00 %</td>
<td>5 min</td>
<td>4 cores @ 2.5 Gz (Matlab + C/C++)</td>
</tr>
</tbody>
</table>

Limitations

• 100+ new algorithms published (benchmarked) since 2002.
  - Middlebury: focus on accuracy
  - KITTI: focus on robust performance

Neither require high accuracy on hi-res images

• Most of the existing methods do not scale.
Limitations

- Searching full disparity space requires \( O(P \times D) \) time
  \[ = O(s^3) \text{ for image size } s \]

<table>
<thead>
<tr>
<th>Middlebury Teddy</th>
<th>KITTI</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.16MP x 60 = 10 Mdisp.</td>
<td>0.5MP x 80 = 40 Mdisp.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Middlebury New</th>
<th>Disney Mansion</th>
</tr>
</thead>
<tbody>
<tr>
<td>6MP x 256 = 1.5 Gdisp.</td>
<td>19MP x 1000 = 19 Gdisp.</td>
</tr>
</tbody>
</table>
New datasets and benchmarks

• Need more challenging datasets for algorithm design
• Middlebury Stereo Eval v.3 is underway *
• 30 new datasets
  - a subset is discussed here -- “Middlebury New 7”

* The benchmark is not public yet.
  If interested, contact Daniel Scharstein (schar@middlebury.edu)
Scalability

• (Most) existing methods are $O(P \times D)$, or $O(P \times D^2)$  
  \[ P: \text{pixels} \]

• These methods do not scale  
  \[ D: \text{disparities} \]

• Ideally, want $O(P)$

• Key Issues:
  - Does higher resolution even help?
  - Does it make sense to enumerate disparities?
  - Coarse to fine strategies?
  - Practical optimization techniques
Promising Approaches

• Efficient approximate energy minimization
  • Only run at low-res; Upsample and refine disparities
    [Ferst et. al. 2013, Ma et. al. 2013]
  • Semi-global Matching (SGM)
    [Hirschmüller 2005]

• Avoid exploring the whole DSI
  • Coarse-to-fine [long tradition]
  • Seed & Grow [Cech & Sara 2007, ...]
  • PatchMatch stereo [Bleyer et al. 2011]
  • ELAS [Geiger et al. 2010]
  • Local Plane Sweep Stereo [this talk]
Semi-global Matching

[Hirschmüller 2005]

- Aggregate along 1D minimum cost paths ending at pixel $p$
- Only cost of this path needed; not the path itself
- Efficiently computed via message-passing.
- Winner-take-all disparity selection
Efficient High-Resolution Stereo Matching using Local Plane Sweeps

Will be presented at CVPR 2014

Daniel Scharstein  
Rick Szeliski
Local Plane Sweep Stereo

- Sparse feature matching; refine vertical disparities
- Generate plane hypotheses (with unknown extents)
Plane hypothesis generation
Local Plane Sweep Stereo

- Sparse feature matching; refine vertical disparities
- Generate plane hypotheses (with unknown extents)
- Perform local plane sweeps (LPS) around hypothesized planes
  - local stereo problem with narrow disparity range; solved using SGM
Local Plane Sweeps

**Plane 2**

Local Plane Sweep solution

Cost Map

In-range disparities (ground truth)

Cost Map

-3
-2
-1
0
1
2
3

**Plane 1**

Local Plane Sweep solution
Local Plane Sweep Stereo

• Sparse feature matching; refine vertical disparities
• Generate plane hypotheses (with unknown extents)
• Perform local plane sweeps (LPS) around hypothesized planes
  - local stereo problem with narrow disparity range; solved using SGM
• Tile structure
  - Perform LPS on tiles and propagate planes to adjoining tiles
Proposal Propagation

• Initial proposals:
  - planes $\pi$ with feature points per tile

• Repeat nR=3 times:
  - Compute surfaces via local plane sweeps per tile
  - Update winner-take-all label map ($L_{WTA}$)
  - Use $L_{WTA}$ to predict well-supported plane proposals
  - Propagate these planes to the neighboring tiles
Local Plane Sweep Stereo

• Sparse feature matching; refine vertical disparities
• Generate plane hypotheses (with unknown extents)
• Perform local plane sweeps (LPS) around hypothesized planes
  - local stereo problem with narrow disparity range; solved using SGM
• Tile structure
  - Perform LPS on tiles and propagate planes to adjoining tiles
• Global optimization
  - Assign pixels to surface proposals
  - Fast approximate energy minimization (via SGM)
  - Extend SGM to exploit tile structure and sparse label sets
Global Optimization (via SGM)

- Message passing on 1D paths (8 directions)

**SGM**
- fixed label set at all pixels

**LPS**
- Label sets vary across tiles
Disparity Selection

• Original SGM:
  • Aggregated costs
  • WTA at every pixel

• LPS:
  • Aggregated costs
  • Top-m candidates at every pixel ($m = 2$)
  • Median filter on candidates within a small window.
Experiments

• Evaluation:
  - PatchMatch Stereo [Bleyer et al. 2011]
  - SGM (our impl.)
  - SGM-HH [Hirschmüller 2005]
  - ELAS [Geiger et al. 2010]
  - LPS

• Metric:
  - 1 and 2 pixel disparity error at non-occluded pixels.
Midd9 (1.4–2.7 MP)

Subset of full-resolution 2003-2006 Middlebury datasets
MiddNew7  (5.1–6.0 MP)

Subset of new 2011-2014 Middlebury training sets
Disney4 (4.5–19 MP)

C. Kim, H. Zimmer, Y. Pritch, A. Sorkine-Hornung, and M. Gross
Scene reconstruction from high spatio-angular resolution light fields
SIGGRAPH 2013

* We treat their results (computed from 100 images) as GT
Results – accuracy

Bad pixels (%), thresh = 1.0 pixel
Error maps (Motorcycle)

PatchMatch

3330 seconds

err1 = 33.8 %

err2 = 24.2 %
Error maps (Motorcycle)

SGM

51.4 seconds  err1 = 29.3%  err2 = 15.1%
Error maps (Motorcycle)

ELAS

5.0 seconds  err1 = 34.0%  err2 = 19.1%
Error maps (Motorcycle)

LPS

Ground Truth  Occlusion Mask

9.6 seconds  err1 = 12.2 %  err2 = 6.5 %
Results – Accuracy vs. Runtime

Avg. error vs. runtime, thresh = 1.0 pixel
Results – Scalability

• With increasing disparity range

![Runtime vs. disparity range](image1)

![Error vs. disparity range](image2)
Results – more proposals

Avg. error vs. number of rounds
LPS – Summary

• Plane proposals from matched features
• Refine into surfaces using LPS
• Propagate promising planes
• Final pixel assignment to surfaces using global optimization
LPS – Benefits

• Doesn’t explore full search space
• Runtime independent of disparity range
• No fronto-parallel bias
• Excellent recovery of slanted surfaces, even with weak texture
• Easy integration of vertical disparity correction
LPS – Limitations

- Can miss surfaces if not among initial proposals
- Cannot handle completely untextured surfaces
- Need “stopping criterion” for proposal generation
- No occlusion reasoning
Future work

• Other types of proposals
  • Hierarchical (coarse to fine)
  • Line, edge features
  • New proposals via “residual analysis”

• Add color models and occlusion reasoning into final pixel assignment

• Better modeling of calibration errors
Promising directions

• Moving away from monolithic optimization
• Global pixel-to-surface assignment (like segmentation; not matching)
• Local stereo matching for proposal generation
• Residual analysis to guide additional search
Surface-based stereo matching

- Piecewise planar stereo
  - Birchfield and Tomasi 2001
  - Furukawa et al. 2008
  - Sinha et al. 2009

- Surface stereo
  - Zebedin et al. 2008
  - Gallup et al. 2010
  - Bleyer et al. 2010, 2011
Multiple View Object Cosegmentation using Appearance and Stereo

Adarsh Kowdle, Sudipta N. Sinha and Rick Szeliski (ECCV 2012)

Using multiple views, infer what constitutes the foreground object.

Input images

Piecewise planar stereo

- Learn per-plane local color models
- Combine stereo and color cues
- Accurate occlusion boundaries

Stereo matching

Plane hypotheses

Plane labels

Depth map
Multiple View Object Cosegmentation using Appearance and Stereo

Adarsh Kowdle, Sudipta N. Sinha and Rick Szeliski (ECCV 2012)
Application:
Image-based Rendering
Photosynth 2 (www.photosynth.net)

• Capture the world in 3D;
• Novel view synthesis
• Interactive viewer
Multiview Photometric Stereo using Planar Mesh Parameterization

Jaesik Park, Sudipta N. Sinha, Yasuyuki Matsushita, Yu-Wing Tai and In So Kweon (ICCV 2013)

- Automatic 3D reconstruction from RGB images
- Multi-view stereo gives coarse shape
- Multi-view Photometric stereo refines shape
Multiview Photometric Stereo using Planar Mesh Parameterization

Jaesik Park, Sudipta N. Sinha, Yasuyuki Matsushita, Yu-Wing Tai and In So Kweon (ICCV 2013)
Multiview Photometric Stereo using Planar Mesh Parameterization

Jaesik Park, Sudipta N. Sinha, Yasuyuki Matsushita, Yu-Wing Tai and In So Kweon (ICCV 2013)
Multiview Photometric Stereo using Planar Mesh Parameterization

Jaesik Park, Sudipta N. Sinha, Yasuyuki Matsushita, Yu-Wing Tai and In So Kweon (ICCV 2013)

Acquisition Setup
Multiview Photometric Stereo using Planar Mesh Parameterization

Input image  Base Mesh (Multiview Stereo)  Final 3D model

Fine geometric details recovered from hi-res images (10+ MP)
Multiview Photometric Stereo using Planar Mesh Parameterization

Input image  Base Mesh (Multiview Stereo)  Final 3D model
Conclusion

• New directions for hi-res stereo matching
• New datasets and benchmarks are coming!
• Local Plane Sweep stereo
  - avoid exploring the whole DSI
  - moving away from monolithic optimization
• Surface-based stereo matching
  - robust; allows a range of priors to be incorporated
• Combining multi-view stereo and photometric stereo
Acknowledgements

Daniel Scharstein, Rick Szeliski, Adarsh Kowdle, Michael Bleyer, Carsten Rother, Pushmeet Kohli, Jaesik Park, Yasuyuki Matsushita, Yu-Wing Tai, In So Kweon
Thanks!