The Middlebury Stereo Vision Benchmark

Daniel Scharstein

Department of Computer Science
Middlebury College

This work is supported by NSF grants IIS-0413169 and IIS-0917109
Joint work with ...

Heiko Hirschmüller, DLR (German aerospace agency)

York Kitajima ’15
Greg Krathwohl ’14
Duncan Levear ’15

Nera Nesic ’13
Xi Wang ’14
Porter Westling ’12
...building on joint work with

Rick Szeliski, Microsoft Research

Anna Blasiak ’07

Jeff Wehrwein ’08

Gonzalo Alonso ’06

Jiaxin (Lily) Fu ’03

Sarri Al-Nashashibi ’08

Brad Hiebert-Treuer ’07

Alexander Vandenberg-Rodes

Alan Lim ’09

Padma Ugbabe ’03
Outline

• Intro to stereo vision
• Middlebury benchmarks
• Ground truth via structured light
• Stereo benchmark v.3
Computer vision

• Goal: Extract information from images (both 2D and 3D)

• Hard problem:
  – Noisy data
  – Lots of it
  – Need additional assumptions
Stereo matching

- Input: rectified image pair
Stereo matching

• Input: rectified image pair
Stereo matching

- Input: rectified image pair
- Output: disparity map

SGM [HH, 2005]

GT

6% errors $|\Delta d| \leq 1$ in nonocc regions
Stereo – applications

• Dense 3D reconstruction
• Robot navigation
• Automated driving
• Gaming, user interfaces
• Virtual viewpoint correction
• 3D movie editing
• …
Stereo vision

• Infer 3D structure from 2 (or more) images of a scene

• Seems easy for humans...
Stereo algorithm

- Step 1: Image matching
- Step 2: 3D reconstruction
Stereo algorithm

• Step 1: Image matching – hard!

  Correspondence problem

• Step 2: 3D reconstruction – fairly easy
Stereo matching

Input: image pair
Output: disparity map (encoding of depth)

• Help: 1D motion only

  (epipolar constraint)
Stereo geometry
Stereo geometry
Stereo geometry

Disparity $d$

= difference in image position
Stereo geometry

Disparity $d$

= difference in image position
Stereo geometry

\[
\frac{d}{f} = \frac{b}{Z}
\]

Disparity \( d = bf \frac{1}{Z} \)
Disparity map (SSD, 11x11 window)

d=0

far

d=17

close
1D Dynamic programming
Why is matching hard?

– Untextured areas
– Noisy data / aliasing
– Depth discontinuties
– Occlusions
– Reflections / specularities
– Different camera responses
– Imperfect calibration
– ...

Which method is best?

Input (left)    Examples of published disparity maps, ca. 1995
Which method is best?

Input (right)    Examples of published disparity maps, ca. 1995
Datasets with ground truth

• Ground truth = true answer
  (e.g., true disparities)

• GT needed for *quantitative* analysis of algorithms (benchmarks)

• Middlebury hosts several computer vision benchmarks:

  http://vision.middlebury.edu/
Outline

• Intro to stereo vision

• Middlebury benchmarks

• Ground truth via structured light

• Stereo benchmark v.3
The Middlebury Computer Vision Pages

Welcome to vision.middlebury.edu. This site is a repository for computer vision evaluations and datasets. It contains:

- The Middlebury Stereo Vision Page, an evaluation of dense two-frame stereo algorithms (described in IJCV 2002)
- The Multi-view Stereo Page, an evaluation of multi-view stereo algorithms (presented at CVPR 2006)
- The MRF Page, an evaluation of energy minimization methods for Markov Random Fields (presented at ECCV 2006)
- The Optical Flow Page, an evaluation of optical flow algorithms (presented at ICCV 2007)
- The Color Page, providing datasets for evaluating the color processing of digital cameras (presented at BMVC 2009)

The material on this site has been developed by Daniel Scharstein and Richard Szeliski, as well as several other researchers, who are listed on the individual project pages. Support by Middlebury College, Microsoft Research, and the National Science Foundation is gratefully acknowledged. Any questions about content, server status, etc., should be directed to Daniel Scharstein. Other pages hosted on this server are listed here.
Middlebury Stereo Page

(Scharstein & Szeliski – CVPR 2001, IJCV 2002)

• vision.middlebury.edu/stereo

• Evaluator with web interface
Middlebury Stereo Page

(Scharstein & Szeliski – CVPR 2001, IJCV 2002)

- vision.middlebury.edu/stereo
- Evaluator with web interface

v.1 (2002) by Lily Fu ’03
Middlebury Stereo Page

(Scharstein & Szeliski – CVPR 2001, IJCV 2002)

- vision.middlebury.edu/stereo
- Evaluator with web interface

v.1 (2002) by Lily Fu ’03
v.2 (2006) by Anna Blasiak ’07
Middlebury Stereo Page

(Scharstein & Szeliski – CVPR 2001, IJCV 2002)

- vision.middlebury.edu/stereo
- Evaluator with web interface

Middlebury stereo benchmark

Stereo Table

Middlebury Stereo Benchmark

New features and improvements
Submit and evaluate

Open a new window

Error Threshold

Algorithm

Average percent of bad pixels
(explanation)

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Error</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>LCU [156]</td>
<td>0.34</td>
<td>3.83</td>
</tr>
<tr>
<td>TS Gol [143]</td>
<td>0.35</td>
<td>4.05</td>
</tr>
<tr>
<td>J OSCF + GCP [136]</td>
<td>0.34</td>
<td>4.18</td>
</tr>
<tr>
<td>ADCensus [3]</td>
<td>0.35</td>
<td>3.97</td>
</tr>
<tr>
<td>AdaptBP [13]</td>
<td>0.34</td>
<td>4.23</td>
</tr>
<tr>
<td>CoReF [3]</td>
<td>0.35</td>
<td>4.47</td>
</tr>
<tr>
<td>CC Radar [13]</td>
<td>0.34</td>
<td>4.73</td>
</tr>
<tr>
<td>RDP [87]</td>
<td>0.35</td>
<td>4.57</td>
</tr>
<tr>
<td>MultiRBF [12]</td>
<td>0.34</td>
<td>4.33</td>
</tr>
<tr>
<td>DoubleBP [3]</td>
<td>0.35</td>
<td>4.19</td>
</tr>
</tbody>
</table>
Middlebury stereo benchmark

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

Middlebury Stereo Evaluation - Version 2

New features and main differences to version 1.
Submit and evaluate your own results.

Open a new window for each link

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Avg. Rank</th>
<th>Tsukuba ground truth</th>
<th>Venus ground truth</th>
<th>Teddy ground truth</th>
<th>Cones ground truth</th>
<th>Average percent of bad pixels (explanation)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>nonocc all disc</td>
<td>nonocc all disc</td>
<td>nonocc all disc</td>
<td>nonocc all disc</td>
<td></td>
</tr>
<tr>
<td>LCU [156]</td>
<td>11.6</td>
<td>1.06 1.34 5.50 15.0</td>
<td>0.07 0.26 1.03</td>
<td>3.68 9.95 10.4 14</td>
<td>1.63 6.87 12 4.82</td>
<td>3.89</td>
</tr>
<tr>
<td>TSGO [143]</td>
<td>11.7</td>
<td>0.87 1.13 4.66 6</td>
<td>0.11 0.24 1.47</td>
<td>5.61 8.09 13.8 34</td>
<td>3.18 1.61 0.45 3</td>
<td>4.06</td>
</tr>
<tr>
<td>JSOS + GCP [151]</td>
<td>13.5</td>
<td>0.74 1.34 3.98 1</td>
<td>0.08 0.16 1.15</td>
<td>3.96 10.1 11.8 20</td>
<td>2.28 7.91 32 6.74 21</td>
<td>4.18</td>
</tr>
<tr>
<td>ADCensus [82]</td>
<td>15.8</td>
<td>1.07 1.48 5.73 23</td>
<td>0.09 0.25 1.15</td>
<td>4.10 6.22 10.9 16</td>
<td>2.42 7.25 18 6.95 25</td>
<td>3.97</td>
</tr>
<tr>
<td>AdaptingBP [16]</td>
<td>19.7</td>
<td>1.11 1.37 5.79 25</td>
<td>0.10 0.21 1.44</td>
<td>4.22 7.06 11.8 21</td>
<td>2.48 7.92 34 7.32 33</td>
<td>4.23</td>
</tr>
<tr>
<td>CoopRegion [39]</td>
<td>20.0</td>
<td>0.87 1.16 4.61 5</td>
<td>0.11 0.21 1.54</td>
<td>5.16 8.31 13.0 28</td>
<td>2.79 7.18 17 8.01 53</td>
<td>4.41</td>
</tr>
<tr>
<td>GCRADAR [452]</td>
<td>24.3</td>
<td>1.15 1.42 6.32 22</td>
<td>0.15 0.27 1.89 4</td>
<td>5.33 10.6 14.7 14</td>
<td>2.01 7.37 32 5.88 4</td>
<td>4.76</td>
</tr>
</tbody>
</table>
**KITTI stereo 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

---

### Stereo Evaluation

<table>
<thead>
<tr>
<th>Rank</th>
<th>Method</th>
<th>Setting</th>
<th>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>
<td></td>
</tr>
<tr>
<td>2</td>
<td>PCBP-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 Ghz (Matlab + C/C++)</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>gtRF-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 @ 2.5 Ghz (Matlab + C/C++)</td>
<td></td>
</tr>
<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>
<td></td>
</tr>
<tr>
<td>5</td>
<td>PR-Sf+D</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 Ghz (Matlab + C/C++)</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>PCBP</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 Ghz (Matlab + C/C++)</td>
<td></td>
</tr>
</tbody>
</table>

---
Limitations

• Searching full disparity space requires \( O(P^*D) \) time
  \( = O(s^3) \) for image size \( s \)

• E.g.
  – Midd Teddy: 450 x 375 x 60 = 1 Mdisp
  – New Motorcycle: 3000 x 2000 x 256 = 1.5 Gdisp
  – Disney Mansion: 19MP x 1000 = 19 Gdisp
Scalability

• Many existing methods don’t scale
  – want $O(P)$, not $O(P \times D)$, or $O(P \times D^2)$, ...

• Questions:
  – does higher resolution even help?
  – does it make sense to enumerate disps?

• Need more challenging datasets for algorithm design and testing
  – hi-res
  – complex geometry
  – realistic scenes
New Datasets

• 2011-2013 collected 33 new datasets
  – multi exposure, multi lighting
  – floating-point disparities (PFM)

• 2013-2014 improved processing at DLR

• Build on structured lighting method by Scharstein & Szeliski [CVPR 2003]
Outline

• Intro to stereo vision
• Middlebury benchmarks
• Ground truth via structured light
• Stereo benchmark v.3
Ground truth via structured light

- Use sequence of illumination patterns to uniquely “color” each scene point
- Compute stereo correspondences (now almost trivial)
- Bonus: Illumination pattern is another source of disparities
Experimental setup (2002)

- Two cameras, one or more projectors

- Alternately, move single camera on rail
Structured light

- Project series of binary images onto scene
- Use Gray code for position encoding
Structured light

- Use code images to uniquely identify each pixel in a scene
Structured light

- Threshold images to obtain Gray code bits
Decoded \((u, v)\) coordinates
Decoded \((u, v)\) coordinates

low-order bits
Disparity computation

• Match \((u,v)\) coordinates between left and right view

⇒ view disparities
View disparities
But...

- Also have correspondences between views and \((u,v)\) coordinates

- Camera-projector stereo matching!
  \[\text{illumination disparities}\]
Recovering the projection matrix $M$

For scene point $S = (x, y, d)$ and projection $P = (u, v)$:

$$
\begin{bmatrix}
    u \\
    v \\
    1
\end{bmatrix} \cong \begin{bmatrix}
\end{bmatrix} \begin{bmatrix}
    x \\
    y \\
    d \\
    1
\end{bmatrix}
$$

Can recover $M$ using robust least-squares fit

Then, $(u, v)$ code gives new disparity estimate!
View disparities
Illumination disparities
2004-2006

• Automated acquisition process
• 2005: 9 datasets    2006: 21 datasets

• But: limited resolution, realism
2011-2014

• Portable rig
  – 2 DSLRs, 2 consumer cameras
• Improved Gray codes
• Natural scenes
• Specular surfaces

2011: 5 datasets  
2012: 7 datasets  
2013: 21 datasets  

• 2014: Improved processing

[Gray codes with maximum min-SW]  
[Gupta et al., CVPR 2011]
Improved processing

• How to get subpixel accuracy at 6 MP?

• Contributions:
  – Robust interpolation of codes
  – Fast 2D subpixel correspondence search
  – Improved calibration via bundle adjustment
  – Improved self-calibration of projectors

• Results:
  – Rectification and disparity accuracy of < 0.2 pixels
  – “perfect” and “imperfect” versions of datasets

• Best paper award at GCPR 2014
Processing pipeline

old system

Calibration images → initial calibration → imperfect calibration

Code images
  → decoding, interpolation → 2D matching → 2D view disparities → merging → merged 2D disparities → bundle adjustment → perfect calibration
  → rectification → decoding, interpolation → 1D matching → 1D view disparities → merging → merged 1D disparities

ambient images → rectification

Demo

imperfect disparities

unknown dy

imperfect ambients

Perfect disparities

Perfect ambients
Rectification errors

Better rectified

Worse rectified

max $dy = -0.25$

min $dy = -1.5$

max $dy = +3.2$

min $dy = -5.2$
Bundle adjustment

- Minimize residual y-disps using nonlinear opt.
- Refine subset of camera params:

<table>
<thead>
<tr>
<th></th>
<th>left cam</th>
<th>right cam</th>
</tr>
</thead>
<tbody>
<tr>
<td>lens distortion</td>
<td>$\kappa_1, \kappa_2$</td>
<td>$\kappa_1, \kappa_2$</td>
</tr>
<tr>
<td>intrinsics</td>
<td>$f_x, f_y$</td>
<td>$f_x = f_y$</td>
</tr>
<tr>
<td></td>
<td>$c_x, c_y$</td>
<td>$c_x, c_y$</td>
</tr>
<tr>
<td>extrinsics</td>
<td>$R_0$</td>
<td>$R_1$</td>
</tr>
<tr>
<td></td>
<td>$T_0$</td>
<td>$T_1$ fixing $|T_1 - T_0|$</td>
</tr>
</tbody>
</table>
Bundle adjustment

- Rectification error reduction:

  averaged over all 33 datasets:

  max: 2.76 -> 0.67  
  avg: 0.77 -> 0.096
Projector self-calibration

• Extend linear model $P = MS$ to include lens distortion of projectors

• Model selection: use best of 0, 3, 4, 6 param lens distortion model

• Error reduction:
  – avg residual: $0.47 \rightarrow 0.26$
  – bad resids $> 1.0$: $7.3\% \rightarrow 0.75\%$
Qualitative Results

• Can render disparity maps in 3D using \texttt{sv, plyv} (part of Heiko’s cvkit)

• Compare 3D reconstructions using
  – new system
  – integer disps
  – old system
    (w/o novel self-calibration and subpixel matching)
Demo mesh viewer
Accuracy

• No absolute measurements available, but
• Can check consistency of disparity estimates from P different projectors:
  – avg sample stddev  $s = 0.20$
  – avg # of samples  $n = 7.7$
  – we provide these as PFM images
• Can check residuals in planar regions
  $r_0$: int disps  $r_1$: old  $r_2$: new
Challenge

• Test new datasets with stereo methods
  – Census [HH et al. 2002]
  – SGM [HH 2006, 2008]
  – ELAS [Geiger et al. 2010]

• Compare % of bad pixels (t=1.0)
  – also include full-size Teddy, Cones (2.7 MP)

• Compare perfect vs. imperfect rectification
Outline

• Intro to stereo vision
• Middlebury benchmarks
• Ground truth via structured light
• Stereo benchmark v.3
Middlebury Benchmark v.3

• Datasets:
  – 15 training pairs, 15 test pairs (hidden GT)
  – Full, Half, and Quarter resolution
  – Some pairs with varying exposure, illumination
  – Most pairs with imperfect rectification

• Evaluation:
  – Multiple performance metrics
  – Weighted average for overall ranking
  – Allows evaluation of “sparse” results
Middlebury Benchmark v.3

• Web interface by Duncan Levear ’15
  – Automatic upload and evaluation
  – Interactive sorting and plotting
  – Supports periodic changes of weights

http://vision.middlebury.edu/stereo/eval3/
Conclusion

• Benchmarks are important, stimulate research

• Creating ground-truth data is challenging, fun

• Hi-resolution images require new level of accuracy

• Stereo is not a solved problem!
Lessons learned on...

• ... value of benchmarks?
• ... value of dense GT
• ... pitfalls / drawbacks (and solutions)?
• ... maintaining benchmarks?
• ... future benchmarks?
Value of benchmarks

- Enables quantitative comparison
- Summarizes state of the art
- Stimulates new research
- Challenging data “pushes envelope”
Value of dense GT

Algorithm design

• Evaluate algorithm components
  – Robust data term
  – Smoothness priors

• (Learning – if enough data)
Pitfalls (1)

- Overfitting to test data
- Focus on ranking
- Deemphasizes aspects not evaluated
- “Rest” after initial “push”
Solutions (1)

• Provide separate training data
• Keep testing GT hidden
• Provide diverse datasets
• Provide multiple metrics, avoid single ranking

• Can analyze correlation among metrics and datasets
Pitfalls (2)

- Overfitting to test data
- Focus on ranking
- Deemphasizes aspects not evaluated
- “Rest” after initial “push”
- Table becomes saturated
- Easy / bad datasets can hurt progress
Solutions (2)

• Update benchmarks periodically
  – But difficult to “seed” new table
  – Difficult to create good datasets

• Have “rolling” benchmarks
  – Predictable (yearly?) change of weighting
  – Change of masks?
  – Remove easy datasets / regions?
Solutions (3)

• Other stereo/flow benchmarks:
  – [Strecha et al. 2008 mview benchmark]
  – KITTI benchmarks (automated driving) [Geiger, Lenz, Stiller, Urtasun, 2012]
  – MPI Sintel Flow (synthetic data) [Butler, Wulff, Stanley, Black, 2012]
  – ...You?
Some personal insights

• “It’s the UI, stupid” 😊
• Huge value in compact representation and visualization of results (& links to papers)
• Participation must be easy
• Strive for automatic scripts, but referee / moderator always needed
• Unless “one-shot contest,” can never completely avoid overfitting
• Significant time commitment
Questions:

• What kind of stereo datasets and scenes do we need, and how many?
  – Indoor/outdoor, resolution, content?
• Cover multiple challenges with single dataset or systematically cover “space”?
• What are values / limitations of other evaluation metrics (e.g. interpolation error)?
• Value of synthetic data?
• Value of realism?
• Can we use human annotation?
Wishlist

• Accurate GT for outdoor images
• “Internet vision” dataset w/ dense GT
• Datasets for scene flow (w/ indep. motion)
• More efforts by others (Sintel & KITTI good start)
• More work on synthetic images