Efficient and Accurate 3D Scene Reconstruction and Object Pose Prediction

Sudipta Sinha
Microsoft Research

University of Illinois at Urbana-Champaign
May 9, 2018
Overview

- Structure from Motion
- Dense Stereo
- Photometric Stereo
- Scene Flow
- Multiview Tracking

Image-based Rendering

Aerial Mapping with UAVs

Image-based Camera Localization

Object 6D Pose

Augmented Reality
Outline

- Stereo Matching: New Trends
- Semi-Global Stereo Matching (SGM)
  - SGM with Surface Orientation Priors
- Stereo Scene Flow with Motion Segmentation
- Trajectory Planning for Aerial Multi View Stereo
- Deep 6D Object Pose Prediction
Stereo Matching

$x$

$y$

$z$

Left

Right

$x$
Stereo Matching

- $x$
- $y$
- $z$
- Left
- Right

Diagram showing stereo matching with images from left and right viewpoints.
Still Challenging

- Fore-shortening
- Specular
- Transparency, reflections
- Different lighting
- Dynamic range
- Untextured slanted surfaces

State of the art methods are still …

- Inaccurate in many corner cases
- Too slow for real-time, resource constrained systems
Stereo benchmarks

- KITTI (2012–15)
- Middlebury (2005)
- Middlebury (2014)
- Kim+ 2013
- ETH3D (2017)

Stereo benchmarks:

- **Middlebury** (2005)
- **Middlebury** (2014)
- **KITTI** (2012–15)
- **ETH3D** (2017)

Benchmarks for stereo vision with different resolutions:

- **Middlebury** (2005):
  - (5 — 6 MPixels)
- **KITTI** (2012–15):
  - (5 — 6 MPixels)
- **ETH3D** (2017):
  - (10 — 20 Mpixels)
Classical Methods (MRF inference)

- Find a per-pixel label (disparity map) $D$, by minimizing energy:

$$E(D) = E_{\text{data}}(D) + E_{\text{smooth}}(D)$$

$$= \sum_{p \in I} C_p(d_p) + \sum_{(p,q) \in N} V_{pq}(d_p, d_q)$$

- Data (cost) term encodes matching costs
- Smoothness (cost) term encodes prior
- Discrete vs. Continuous labels
- Inference: Graph Cuts, Belief Propagation, PatchMatch-style optim.
Piecewise Planar Stereo

[Sinha, Steedly, Szeliski 2009]

Multiple Plane Detection

3D Line Reconstruction

MRF energy minimization via graph cuts

Structure from motion

Novel View
Piecewise Planar Stereo *Revisited* [Kowdle, Sinha, Szeliski 2012]

- Local plane fitting (more flexible)
- CRF models photo-consistency (stereo cue) and color segmentation (monocular cue)
- Learn color models per-surface
- Alternate between graph cuts and learning

Semi-global stereo (SGM)  ➔  Find planes  ➔  Label map

Depth map
Piecewise Planar Stereo *Revisited*  
[Kowdle, Sinha, Szeliski 2012]
New Trends

- Learning the matching cost:
  - MC-CNN [Zbontar + Lecun 2015], Chen+ 2015, Luo+ 2016
- Continuous MRFs: [Taniai+ 2017] (Rank 1 on Middlebury 2014!)
- Deep stereo regression (end to end training)
  - FlowNet [Dosovitskiy+ 2015], DispNet [Mayer+ 2016]
- Return of “Correlation”
  - DispNetCorr [Mayer+ 2016]
  - GC-Net [Kendall+ 2017]
- Return of “CRFs” (Hybrid CNN-CRF models)
  - Seki and Pollefeys 2017, Knobelreiter+ 2017
Stereo Benchmark Rankings

Middlebury 2014

KITTI 2015
Group A and B have no methods in common!
Group A entries all use MC-CNN acrt but no other “deep learning” technique!
Group B methods do NOT use MC-CNN acrt; they use ResNet, 3D convolutions, 3D deconvolutions, U-shaped Nets, RNNs; End to end learning is very popular!
Conclusions

Must train **one** model on combined training set and submit to all benchmarks!
Outline

- Stereo Matching: New Trends
- Semi-Global Stereo Matching (SGM)
  - SGM with Surface Orientation Priors
- Stereo Scene Flow with Motion Segmentation
- Trajectory Planning for Aerial Multi View Stereo
- Deep 6D Object Pose Prediction
Semi Global Matching [Hirschmüller 2005]

- MRF inference (Graph Cuts, BP, ..) too slow
- SGM: Approximate even more; use heuristics
  - Widely used: assisted driving, robotics, aerial mapping …
  - Runs in real-time on FPGAs, GPUs …
Scanline Optimization (1D)

Minimize:

\[ E(D) = \sum_{p \in I} C_p(d_p) + \sum_{(p,q) \in N} V_{pq}(d_p, d_q) \]

- Consider the above problem on a 1D scanline.
- Compute an aggregated matching cost

\[ L_r(p, d) = C_p(d) + \min_{d' \in D} (L_r(p - r, d') + V(d, d')) \]

- \( r = (1, 0) \): start at leftmost pixel, scan left
Semi Global Matching (SGM)

- For 8 directions
  - calculate aggregated costs

\[
L_r(p, d) = C_p(d) + \min_{d' \in D} (L_r(p - r, d') + V(d, d')).
\]

- Finally, sum the costs and select per-pixel minima.

\[
S(p, d) = \sum_r L_r(p, d)
\]

\[
D_p = \arg \min_d S(p, d).
\]
Semi Global Matching (SGM)
Semi Global Matching [Hirschmüller 2005]

Approximates 2D MRF using 1D optimization along 8 cardinal directions

\[
E(D) = \sum_p C_p(d_p) + \sum_{p,q \in \mathcal{N}} V(d_p, d_q)
\]

- related to Belief Propagation, TRW-S

[Drory et al. 2014]
Semi Global Stereo Matching with Surface Orientation Priors

3DV 2017

Daniel Scharstein
Middlebury College

Tatsunori Taniai
RIKEN, Tokyo

Sudipta Sinha
Microsoft Research
Semi Global Matching [Hirschmüller 2005]

Approximates 2D MRF using 1D optimization along 8 cardinal directions

\[ E(D) = \sum_{p} C_p(d_p) + \sum_{p,q \in N} V(d_p,d_q) \]

\[ \int 0 \quad \text{if } d = d' \]

- Fronto parallel bias
- Inaccurate on slanted untextured surfaces
SGM\(^*\) (quarter resolution)

SGM\(^*\) (full resolution (6 MP))

* Confidence measure used to prune uncertain pixels (black holes)
SGM-P: SGM with orientation priors

- [Scharstein, Taniai, Sinha, 3DV 2017]
- What if we knew the surface slant?
- Replace fronto-parallel bias with bias parallel to surface

Idea:
- *Rasterize* disparity surface prior (at arbitrary depth)
- Adjust $V(d, d')$ to follow discrete disparity “steps”
SGM-P: 2D orientation priors

\[ V_S(d_p, d'_p) = V(d_p + j_p, d'_p) \]
SGM-P: 3D orientation priors

\[ V_S(d_p, d'_p) = V(d_p + j_p(d_p), d'_p) \]

Jump locations vary with disparity
SGM-P: Where do we get priors?

- Matched features + triangulation
- Matched features + plane fitting
- Low-res matching + plane fitting
- Ground truth oracle
- Semantic analysis
- Manhattan-world assumptions
SGM-P: Results

Adirondack disparities error map

Motorcycle disparities error map

SGM

SGM-EPi

SGM-GS
SGM-P: Results

% Disparity Error > 2.0 (F)

% Error reduction over SGM (F)
SGM-P: Results

- Huge performance gains for slanted untextured scenes
- Soft constraint; inaccurate normals don’t hurt accuracy

![Graph showing % Disparity Error > 2.0 and % Error reduction over SGM](image)
Outline

- Stereo Matching: New Trends
- Semi-Global Stereo Matching (SGM)
  - SGM with Surface Orientation Priors
- Stereo Scene Flow with Motion Segmentation
- Trajectory Planning for Aerial Multi View Stereo
- Deep 6D Object Pose Prediction
Fast Multi-frame Stereo Scene Flow with Motion Segmentation

CVPR 2017

Tatsunori Taniai
RIKEN, Tokyo

Sudipta Sinha
Microsoft Research

Yoichio Sato
Univ. of Tokyo
Multi-frame Stereo Scene Flow

- Stereo video from moving stereo camera rig (calibrated)
- Scene Flow equivalent to stereo matching and optical flow estimation
Application

Object scene flow for autonomous vehicles
Menze and Geiger 2015

Action recognition by dense trajectories
Wang+ 2011

- Depth and flow sequences are useful in many applications
Motivation

Efficient, unified method for

- Stereo
- Optical Flow
- Moving object segmentation
- Visual Odometry (Camera ego-motion)
Main Idea: Dominant Rigid Scene Assumption

- Most of the scene is rigid; hence, camera motion determines the *rigid optical flow*.
- Given *rigid flow map*, only find regions with moving objects and recompute their flow.
Proposed Approach

Input

- Binocular stereo
- Visual odometry
- Epipolar stereo
- Initial motion segmentation
- Optical flow
- Flow fusion

Flow fusion

- Flow
- Motion seg.

Flow fusion

- $I_t^0, I_t^1$
- $I_t^0, I_t^0 + \tilde{D}$
- $I_t^0, I_t^0 + P$
- $I_t^0, I_t^0 + P, D$
- $I_t^0, I_t^0 + \tilde{S}$
- $I_t^0, I_t^0 + F_{rig}, F_{non}$

Flow fusion

- $\tilde{D}$
- $P$
- $D$
- $\tilde{S}$
- $F_{rig}$
- $F_{non}$
Results – KITTI 2015 Scene Flow Benchmark (Nov 2016)

<table>
<thead>
<tr>
<th>Rank</th>
<th>Method</th>
<th>D1-bg</th>
<th>D1-fg</th>
<th>D1-all</th>
<th>D2-bg</th>
<th>D2-fg</th>
<th>D2-all</th>
<th>Fl-bg</th>
<th>Fl-fg</th>
<th>Fl-all</th>
<th>SF-bg</th>
<th>SF-fg</th>
<th>SF-all</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>PRSM [43]</td>
<td>3.02</td>
<td>10.52</td>
<td>4.27</td>
<td>5.13</td>
<td>15.11</td>
<td>6.79</td>
<td>5.33</td>
<td>17.02</td>
<td>7.28</td>
<td>6.61</td>
<td>23.60</td>
<td>9.44</td>
<td>300 s</td>
</tr>
<tr>
<td>2</td>
<td>OSF [30]</td>
<td>4.54</td>
<td>12.03</td>
<td>5.79</td>
<td>5.45</td>
<td>19.41</td>
<td>7.77</td>
<td>5.62</td>
<td>22.17</td>
<td>8.37</td>
<td>7.01</td>
<td>28.76</td>
<td>10.63</td>
<td>50 min</td>
</tr>
<tr>
<td>3</td>
<td>FSF+MS (ours)</td>
<td>5.72</td>
<td>11.84</td>
<td>6.74</td>
<td>5.77</td>
<td>12.28</td>
<td>9.85</td>
<td>8.48</td>
<td>29.62</td>
<td>12.00</td>
<td>11.17</td>
<td>37.40</td>
<td>15.54</td>
<td>2.7 s</td>
</tr>
<tr>
<td>8</td>
<td>PCOF + ACTF [10]</td>
<td>6.31</td>
<td>19.24</td>
<td>8.46</td>
<td>19.15</td>
<td>36.27</td>
<td>22.00</td>
<td>14.89</td>
<td>62.42</td>
<td>22.80</td>
<td>25.77</td>
<td>69.35</td>
<td>33.02</td>
<td>0.08 s (GPU)</td>
</tr>
<tr>
<td>12</td>
<td>GCSF [8]</td>
<td>11.64</td>
<td>27.11</td>
<td>14.21</td>
<td>32.94</td>
<td>35.77</td>
<td>33.41</td>
<td>47.38</td>
<td>45.08</td>
<td>47.00</td>
<td>52.92</td>
<td>59.11</td>
<td>53.95</td>
<td>2.4 s</td>
</tr>
</tbody>
</table>

200 road scenes with multiple moving objects
Outline

- Stereo Matching: New Trends
- Semi-Global Stereo Matching (SGM)
  - SGM with Surface Orientation Priors
- Stereo Scene Flow with Motion Segmentation
- Trajectory Planning for Aerial Multi View Stereo
- Deep 6D Object Pose Prediction
Submodular Trajectory Optimization for Aerial 3D Scanning

ICCV 2017

Mike Roberts\textsuperscript{1,2} Debadeepta Dey\textsuperscript{2} Anh Truong\textsuperscript{3} Sudipta Sinha\textsuperscript{2}
Shital Shah\textsuperscript{2} Ashish Kapoor\textsuperscript{2} Pat Hanrahan\textsuperscript{1} Neel Joshi\textsuperscript{2}

\textsuperscript{1}Stanford University \textsuperscript{2}Microsoft Research \textsuperscript{3}Adobe Research
Acquiring imagery using drones
Multi-view Stereo Reconstruction
Manual Planning Prior to Capture

- Waypoints planned by human experts …
  - Several redundant flight trajectories were flown
- 3,500 images from 6 days with 19 ten-minute flights
- Projeto Redentor (Pix4D whitepaper, 2015)
Our Goal

- Automatically generate optimized trajectories for 3D scanning using drones, such that
  1. the acquired images will produce an accurate 3D model when processed using a Multi View Stereo (MVS) algorithm.
  2. the UAV makes best use of its limited flight time budget.

- Processing happens post flight.
- Battery typically lasts 15—20 minutes.
Related Work

- **View selection** [Hornung+ 2011]
  - First, acquire dense imagery
  - Later, select subset & process

- **Next-best-view planning**
  - Information-gain maximization [Isler+ 2016]
  - Robotic RGB-D 3D scanning [Wu+ 2014]
  - No travel budget constraints

- **Real-time Drone view planning** [Mostegel+ 2016]
  - Greedy technique; heuristic-based
Diverse Viewpoints help Multi View Stereo

Preference for
- diverging viewing angles
- close-up views
- fronto parallel views of surfaces
Coverage Measure

For a surface point $S$ observed from multiple cameras, we define coverage as the area of the union of all the blue disks on a hemisphere.
Similarly, we define coverage for multiple surface points observed from multiple camera viewpoints.
Method

- Evaluating coverage function requires knowledge of scene geometry
- Thus, we follow a two-staged procedure.

1. Fly an easy-to-generate trajectory;
2. Compute coarse reconstruction (SFM $\rightarrow$ MVS $\rightarrow$ meshing)
3. Plan *optimized* trajectory based on mesh from step 2.
4. Fly trajectory computed in step 3.
5. Run SFM + MVS on images from step 1 and 4.
Planning Optimized Trajectories

Graph of all possible camera location (and orientation); edge weights are Euclidean distances between locations.

Propose to solve the problem in two steps.

1. Solve optimal set of orientations; ignoring path constraints

2. Then, find the set of locations by solving a graph orienteering problem
Solving for Camera Orientations

- Coverage set function is submodular
  - Adding new elements to an existing set gives diminishing returns
- Cardinality and Mutual Exclusion Constraint
  - Select exactly one look-at vector at each position
- Constrained submodular maximization
  - Always, pick the next best element with the most marginal reward
  - Greedy algorithm; good theoretical approximation guarantee
Solving for Camera Positions

- Graph Orienteering Problem
  - NP-Hard; related to TSP and Knapsack
  - Find short paths that let you collect most rewards (at nodes).
- In standard orienteering, rewards are additive.
- But, our reward function is submodular, not additive!
- Hence, we must solve a submodular orienteering problem.
Solving for Camera Positions

- Choose a good sub-gradient
  (additive approximation)
  for our submodular function

- Approximation yields an instance of the orienteering problem

\[
\text{maximize } \sum c(s) \\
\text{subject to } T(S) < B
\]

- Solve as an integer linear program (ILP)
Results

- Pix4D for 3D reconstruction
  - Outputs texture-mapped 3D model
- Baselines:
  - **Overview**: Lawn-mower pattern
  - **Random**:
    - recover coarse 3d model; estimate free space.
    - select random points in free space.
    - compute TSP tour.
Results

Our computed trajectories visualized in Google Earth

Barn

MSR Redmond
Results
Insert video (ICCV supplementary video here)
Outline

- Stereo Matching: New Trends
- Semi-Global Stereo Matching (SGM)
  - SGM with Surface Orientation Priors
- Stereo Scene Flow with Motion Segmentation
- Trajectory Planning for Aerial Multi View Stereo
- Deep 6D Object Pose Prediction
3D Recognition, 2D-3D Model Alignment

Given a RGB image (with known intrinsics), recognize the objects and predict their 3D position and orientation within the scene.

Classical methods:

- Recognizing Image Patches
  - Scale, Affine invariant features
  - Geometric verification (rigid scenes)

  Lowe 2001

- Rothganger+ 2005

  - Worked for textured, distinctive objects
  - Required a small # of training images

  Lepetit+ 2005
Object 6D Pose Estimation

Given a RGB image (with known intrinsics), recognize the objects and predict their 3D position and orientation within the scene.

RGB-D methods:
- Lai+ 2010
- Hinterstoisser+ 2012
- Brachmann+ 2014, 2016

- Classical Object Recognition
- Fast Image Retrieval

CNN methods:
- Rad + Lepetit 2017
- Kehl+ 2017
- Xiang+ 2017
- Tekin+ 2018
- Oberweger+ 2018

- Global deep feature representations
- Not much use of geometry
- Promising for small, texture-less objects
- Huge training set needed
Texture-less Object 6D Pose Datasets

LINEMOD [2012]
15 objects

T-LESS [2017]
30 objects

YCB-VIDEO [2018]
21 objects
Deep 6D object pose estimation

- BB8 [Rad and Lepetit 2017]

- CNN$_1$ (2d detector)
- cropped image
- CNN$_2$ (pose estimator)
- CNN$_3$ (pose refinement)
- 6D pose
Our Method

- Single-shot 2D object detection (YOLO, SSD)
- Our CNN predicts 2D projections of 3D bounding box vertices (and the centroid). We run PnP solver on 9 2D-3D correspondences.
- Accurate, fast (50-90 fps); detects multiple objects in one pass.
Our Method

Training:
- ground truth 2D coordinates of the 9 control points are the targets
- modify YOLO loss function (for confidence estimation)
- data augmentation

Testing:
- Subpixel refinement
- PnP (RANSAC, least squares)
CNN Architecture

Convolutional 3x3x32
Max-pool 2x2/2
Conv.
3x3x64
Max-pool 2x2/2
Convolutional 3x3x128
1x1x64
3x3x128
Max-pool 2x2/2
Convolutional 3x3x256
1x1x128
3x3x256
Max-pool 2x2/2
Convolutional 3x3x512
1x1x256
3x3x512
1x1x256
3x3x512
Max-pool 2x2/2
Convolutional 3x3x1024
1x1x512
3x3x1024
1x1x512
3x3x1024
3x3x1024
Merge
Convolutional 3x3x1024
Convolutional 1x1x(9x2+1+C)

<table>
<thead>
<tr>
<th>Layer</th>
<th>Output Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>416 x 416</td>
</tr>
<tr>
<td>1</td>
<td>208 x 208</td>
</tr>
<tr>
<td>2</td>
<td>104 x 104</td>
</tr>
<tr>
<td>3</td>
<td>52 x 52</td>
</tr>
<tr>
<td>4</td>
<td>26 x 26</td>
</tr>
<tr>
<td>5</td>
<td>512 x 512</td>
</tr>
<tr>
<td>6</td>
<td>128 x 128</td>
</tr>
<tr>
<td>7</td>
<td>64 x 64</td>
</tr>
<tr>
<td>8</td>
<td>256 x 256</td>
</tr>
<tr>
<td>9</td>
<td>1024 x 1024</td>
</tr>
</tbody>
</table>
CNN Architecture

- Convolutional 3x3x32
- Max-pool 2x2/2 Conv. 3x3x64
- Max-pool 2x2/2 Convolutional 3x3x128 1x1x64 3x3x128
- Max-pool 2x2/2 Convolutional 3x3x256 1x1x128 3x3x256
- Max-pool 2x2/2 Convolutional 3x3x512 1x1x256 3x3x512 1x1x256 9x3x512
- Max-pool 2x2/2 Convolutional 3x3x1024 1x1x512 3x3x1024 1x1x512 3x3x1024 3x3x1024
- Merge
- Convolutional 3x3x1024
- Convolutional 1x1x(9x2+1+C)
Results on LineMOD dataset

- Two accuracy metrics (2D image projection, 3D model overlap).
- Percentage of test images where the error was lower than specified thresholds.

### 2D metric

<table>
<thead>
<tr>
<th>Method</th>
<th>w/o Refinement</th>
<th>w/ Refinement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ape</td>
<td>-</td>
<td>95.3</td>
</tr>
<tr>
<td>Benchwise</td>
<td>-</td>
<td>80.0</td>
</tr>
<tr>
<td>Cam</td>
<td>-</td>
<td>80.9</td>
</tr>
<tr>
<td>Can</td>
<td>-</td>
<td>84.1</td>
</tr>
<tr>
<td>Cat</td>
<td>-</td>
<td>97.0</td>
</tr>
<tr>
<td>Driller</td>
<td>-</td>
<td>74.1</td>
</tr>
<tr>
<td>Duck</td>
<td>-</td>
<td>81.2</td>
</tr>
<tr>
<td>Eggbox</td>
<td>-</td>
<td>87.9</td>
</tr>
<tr>
<td>Glue</td>
<td>-</td>
<td>89.0</td>
</tr>
<tr>
<td>Holepuncher</td>
<td>-</td>
<td>90.5</td>
</tr>
<tr>
<td>Iron</td>
<td>-</td>
<td>78.9</td>
</tr>
<tr>
<td>Lamp</td>
<td>-</td>
<td>74.4</td>
</tr>
<tr>
<td>Phone</td>
<td>-</td>
<td>77.6</td>
</tr>
<tr>
<td>Average</td>
<td>69.5</td>
<td>83.9</td>
</tr>
</tbody>
</table>

### 3D metric

<table>
<thead>
<tr>
<th>Method</th>
<th>w/o Refinement</th>
<th>w/ Refinement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ape</td>
<td>-</td>
<td>27.9</td>
</tr>
<tr>
<td>Benchwise</td>
<td>-</td>
<td>62.0</td>
</tr>
<tr>
<td>Cam</td>
<td>-</td>
<td>40.1</td>
</tr>
<tr>
<td>Can</td>
<td>-</td>
<td>48.1</td>
</tr>
<tr>
<td>Cat</td>
<td>-</td>
<td>45.2</td>
</tr>
<tr>
<td>Driller</td>
<td>-</td>
<td>58.6</td>
</tr>
<tr>
<td>Duck</td>
<td>-</td>
<td>32.8</td>
</tr>
<tr>
<td>Eggbox</td>
<td>-</td>
<td>40.0</td>
</tr>
<tr>
<td>Glue</td>
<td>-</td>
<td>27.0</td>
</tr>
<tr>
<td>Holepuncher</td>
<td>-</td>
<td>42.4</td>
</tr>
<tr>
<td>Iron</td>
<td>-</td>
<td>67.0</td>
</tr>
<tr>
<td>Lamp</td>
<td>-</td>
<td>39.9</td>
</tr>
<tr>
<td>Phone</td>
<td>-</td>
<td>35.2</td>
</tr>
<tr>
<td>Average</td>
<td>32.3</td>
<td>43.6</td>
</tr>
</tbody>
</table>
Results on LineMOD dataset

- Running Times:
  - On TitanX or similar GPU.
  - using cuDNN

<table>
<thead>
<tr>
<th>Method</th>
<th>Overall speed for 1 object</th>
<th>Refinement runtime</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brachmann et al. [2]</td>
<td>2 fps</td>
<td>100 ms/object</td>
</tr>
<tr>
<td>Rad &amp; Lepetit [25]</td>
<td>3 fps</td>
<td>21 ms/object</td>
</tr>
<tr>
<td>Kehl et al. [10]</td>
<td>10 fps</td>
<td>24 ms/object</td>
</tr>
<tr>
<td>OURS</td>
<td>50 fps</td>
<td>-</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>2D projection metric</th>
<th>Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>416 × 416</td>
<td>89.71</td>
<td>94 fps</td>
</tr>
<tr>
<td>480 × 480</td>
<td>90.00</td>
<td>67 fps</td>
</tr>
<tr>
<td>544 × 544</td>
<td>90.37</td>
<td>50 fps</td>
</tr>
<tr>
<td>688 × 688</td>
<td>90.65</td>
<td>43 fps</td>
</tr>
</tbody>
</table>

When input image is resized, our method remains accurate and runs much faster.
Conclusions

- State of the art in stereo matching; new challenges
- Improvements to Semi Global Matching
  - Incorporating soft surface orientation priors
- Fast scene flow with motion segmentation
- Camera path planning for improved multi-view stereo
- Deep single shot 6D object pose estimation
  - CNN architecture conceptually simpler (~YOLO architecture) and faster