Dense correspondence recovery involving images and 3D models

Sudipta N. Sinha
Microsoft Research

University of Utah,
March 27, 2018
Introduction

- Estimate correspondences and align multiple entities
  - Image to image (stereo, optical flow, scene flow … )
  - Image to 3D model (object recognition, pose estimation … )

- Applications:
  - **Vision**: image stitching, structure from motion, visual odometry, SLAM, camera localization, 3D mapping, 4D reconstruction, …
  - **Augmented Reality**: object recognition, 6D pose recovery, tracking
  - **Robotics**: localization, avoiding obstacles, object grasping & moving
Dense Image Correspondence

Binocular stereo

Multi-view stereo

Optical flow

Scene Flow

Dense Image Correspondence

Binocular stereo

Multi-view stereo

Optical flow

Scene Flow

Dense Image Correspondence

Binocular stereo

Multi-view stereo

Optical flow

Scene Flow

Dense Image Correspondence

Binocular stereo

Multi-view stereo

Optical flow

Scene Flow
Task: recognize object instances in an image, find pose of associated 3D models; project 3D model to get dense alignment.

Need training data (images, models, annotation); real images vs. CG

Challenges: scene clutter, low texture, difficult lighting, low resolution
Outline

- Global Stereo Matching with piecewise-planar priors
- Semi-global Matching (SGM)
  - Local plane-sweep stereo
  - SGM with surface orientation priors
- Stereoscopic Scene Flow
- Deep Single-shot 6D Object Instance Detection
Stereo Matching

- Dense pixel correspondence in rectified pairs
- Disparity Map: $D(x, y)$
  $$x' = x + D(x, y), \ y' = y$$
- Depth Map: $Z(x, y) = \frac{bf}{D(x,y)}$
Stereo Matching

Left

Right

x

y

z
Binocular Stereo Matching

\[ x \]

\[ y \]

\[ z \]

Left

Right

\[ x \]
Local Optimization

- Minimize matching cost at each pixel independently
- Winner-take-all (WTA)

\[
C_{SAD}(p, d) = \sum_{q \in N_p} |I_L(q) - I_R(q - d)|
\]

\[
C_{ZNCC}(p, d) = \frac{\sum_{q \in N_p} (I_L(q) - \bar{I}_L(p))(I_R(q - d) - \bar{I}_R(p - d))}{\sqrt{\sum_{q \in N_p} (I_L(q) - \bar{I}_L(p))^2 \sum_{q \in N_p} (I_R(q - d) - \bar{I}_R(p - d))^2}}
\]

- Convolutional Neural Nets

[Zbontar and Lecun 2015]
Stereo benchmarks

KITTI (2012—15)

Middlebury (2005)

Middlebury (2014)

Kim+ 2013

ETH3D (2017)

(5 — 6 MPixels)

(10—20 Mpixels)
Still challenging ...

- Corner cases:
  - Challenging geometry
  - Complex appearance
- High resolution imagery
- Real-time platforms, resource-constraints
Priors for Stereo Matching

- Stereo matching is an ill-posed problem
- Priors provides robustness to ambiguity and noise, e.g.
  - Smoothness prior ($1^{\text{st}}$–order, $2^{\text{nd}}$–order ...)
  - Discontinuities favored at image edges
  - Soft color segmentation cues (superpixels ...)
- Priors explicitly added to optimization objective
- Priors terms in objective can be learned from training data
Outline

- Global Stereo Matching with piecewise-planar priors
- Semi-global Matching (SGM)
  - Local plane-sweep stereo
  - SGM with surface orientation priors
- Stereoscopic Scene Flow
- Deep Single-shot 6D Object Instance Detection
Global Optimization

- Find a per-pixel label map \( D \) (find all disparities jointly)
- Labels are discrete (ordered), \( d \in L_D \)
  \[
  L_D = [d_{\text{min}}, d_{\text{max}}]
  \]
- Optimize:
  \[
  E(D) = E_{\text{data}}(D) + E_{\text{smooth}}(D)
  \]
- Data term encodes matching costs
- Smoothness term encodes prior/regularization
  - Example: neighboring pixels favored to take similar labels
Global Optimization

- Inference on Markov Random Fields (MRF)
- Minimize objective (energy):

\[
E(D) = E_{\text{data}}(D) + E_{\text{smooth}}(L)
\]

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

- \( C_p(d_p) \): matching cost term (\textit{tabular representation})
- \( V_{pq}(d, d') \): pairwise term (Potts, truncated linear or quadratic ...)

\textit{contrast sensitive Potts model} favors discontinuity at image edges
Global Optimization

▪ Binary MRFs:
  ▪ Efficient, exact methods known
  ▪ Submodular $V(\ast,\ast)$: s-t mincut problem

▪ Multi-label MRFs:
  ▪ NP-Hard, for useful choice of $V_{pq}(\ast,\ast)$
    ▪ *Discontinuity-preserving* Potts model.
  ▪ Approximation algorithms
    ▪ Move-making (via binary graph cuts)
Stereo matching with planar priors

[Sinha, Steedly, Szeliski 2009]
Stereo matching with planar priors
Image-based Rendering

Brownhouse (55 images)
Image-based Rendering

Castle (30 images)
Stereo matching with planar priors

- Tackle more general scenes
- Plane hypotheses generated via local fitting
- Now, alternate between
  - Learning surface color models (online)
  - Graph cut optimization

[Kowdle, Sinha, Szeliski 2012]

Semi-global stereo (SGM) → Plane label map
Find planes → Depth map

[Kowdle, Sinha, Szeliski 2012]
Stereo matching with planar priors

[Kowdle, Sinha, Szeliski 2012]
Image-based Rendering
Review: Stereo with planar priors

- MRF labels: planes (surfaces), NOT disparities.
- Estimated depth maps often approximate
  - ✓ accurate recovery of occlusion boundaries, surface normals
  - ✓ effective 2.5D proxies for novel view synthesis
- Limitations:
  - ✗ Planarity prior too strong for general scenes
  - ✗ Plane proposal generation is key; often imperfect
Outline

- Global Stereo Matching with piecewise-planar priors
- Semi-global Matching (SGM)
  - Local plane-sweep stereo
  - SGM with surface orientation priors
- Stereoscopic Scene Flow
- Deep Single-shot 6D Object Instance Detection
Semi Global Matching [Hirschmüller 2005]

- MRF inference (graph cuts, BP, ..) too slow
- SGM: Approximate even more; use heuristics
  - Parallelizable; practical on FPGA / GPUs
  - Widely used for assisted driving, robotics, aerial mapping …
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 [Hirschmϋller 2005]

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

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

- related to Belief Propagation

[Drory et al. 2014]
Semi Global Matching [Hirschmüller 2005]

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

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

- Evaluates the whole DSI
- Inefficient for high-resolution images

- Related to BP, TRW
  - [Drory et al. 2014]
Outline

- Global Stereo Matching with piecewise-planar priors
- Semi-global Matching (SGM)
  - Local plane-sweep stereo
  - SGM with surface orientation priors
- Stereoscopic Scene Flow
- Deep Single-shot 6D Object Instance Detection
Local Plane Sweep (LPS) Stereo

[Sinha, Scharstein, Szeliski 2014]

- Solve many local plane sweep stereo (LPS) problems
- Generates *surface* proposals; fuse them into a disparity map

Local Plane Sweep (LPS) Stereo

[Sinha, Scharstein, Szeliski 2014]
Local Plane Sweep (LPS) Stereo

[Sinha, Scharstein, Szeliski 2014]
Local Plane Sweep (LPS) Stereo

[Sinha, Scharstein, Szeliski 2014]
Local Plane Sweep (LPS) Stereo

- Lower bias towards piecewise planar reconstructions
- Faster (avoids evaluating the whole DSI)
- More accurate than SGM

BUT, tends to be inaccurate near weakly textured surfaces

[Sinha, Scharstein, Szeliski 2014]
Outline

- Global Stereo Matching with piecewise-planar priors
- Semi-global Matching (SGM)
  - Local plane-sweep stereo
  - SGM with surface orientation priors
- Stereoscopic Scene Flow
- Deep Single-shot 6D Object Instance Detection
Semi Global Matching [Hirschmüller 2005]

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

\[ E(D) = \sum_{p} C_{p}(d_{p}) + \sum_{p,q \in \mathcal{N}} V(d_{p}, d_{q}) \]

\[ V(d, d') = \begin{cases} 
0 & \text{if } d = d' \\
P_1 & \text{if } |d - d'| = 1 \\
P_2 & \text{if } |d - d'| \geq 2 
\end{cases} \]
Semi Global Matching [Hirschmüller 2005]

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

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

- Fronto parallel bias
- Inaccurate on slanted untextured surfaces
▪ This subtitle is 20 points
▪ Bullets are blue
▪ They have 110% line spacing, 2 points before & after
▪ Longer bullets in the form of a paragraph are harder to read if there is insufficient line spacing. This is the maximum recommended number of lines per slide (seven).

SGM @ quarter resolution

SGM @ full resolution (6 MP)
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

<table>
<thead>
<tr>
<th></th>
<th>Adirondack disparities</th>
<th>Adirondack error map</th>
<th>Motorcycle disparities</th>
<th>Motorcycle error map</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SGM</strong></td>
<td><img src="image1" alt="Image" /></td>
<td><img src="image2" alt="Image" /></td>
<td><img src="image3" alt="Image" /></td>
<td><img src="image4" alt="Image" /></td>
</tr>
<tr>
<td><strong>SGM-EPi</strong></td>
<td><img src="image5" alt="Image" /></td>
<td><img src="image6" alt="Image" /></td>
<td><img src="image7" alt="Image" /></td>
<td><img src="image8" alt="Image" /></td>
</tr>
<tr>
<td><strong>SGM-GS</strong></td>
<td><img src="image9" alt="Image" /></td>
<td><img src="image10" alt="Image" /></td>
<td><img src="image11" alt="Image" /></td>
<td><img src="image12" alt="Image" /></td>
</tr>
</tbody>
</table>
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, doesn’t hurt accuracy – Good!
Fast Multi-frame Stereo Scene Flow with Motion Segmentation
Taniai, Sinha, Sato 2017

Input:
Stereo Video

Output
Disparity Map
Optical Flow
Moving object segmentation
Fast Multi-frame Stereo Scene Flow with Motion Segmentation
Taniai, Sinha, Sato 2017

Input: $I_t^0, I_t^1$

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

Ego-motion: $P$
Disparity: $D$
Flow: $F_{rig}$
Final segmentation: $P + D$
## Fast Multi-frame Stereo Scene Flow with Motion Segmentation

**Taniai, Sinha, Sato 2017**

### 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>6.79</td>
<td>6.79</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>7.57</td>
<td>21.28</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

- Global Stereo Matching with piecewise-planar priors
- Semi-global Matching (SGM)
  - Local plane-sweep stereo
  - SGM with surface orientation priors
- Stereoscopic Scene Flow
- Deep Single-shot 6D Object Instance Detection
Object recognition + pose estimation

- **Task:** Given a RGB image (with known camera intrinsics), recognize the object instance and predict its 3D position and orientation.

- **CNN-based**
  - Rad + Lepetit 2017,
  - Kehl+ 2017,
  - Xiang+ 2017

- Global deep features
- Geometry not used
- small, texture-less objects
- Huge training set needed

**Local features (SIFT, Affine invariance)**
- Textured, distinctive objects
- Geometric verification
- A few training images are fine ..
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]

  ![Diagram](image)

  - **2d detector**
  - **pose estimator**
  - **pose refinement**

- **SSD-6D** [Kehl+ 2017]

  ![Diagram](image)

  - **2d detector, viewpoint classifier**
  - **heuristic**
  - **pose refinement**

- **Ours**

  ![Diagram](image)

  - **3D bounding box corner predictor**
  - **pose solver**

  - **PnP**
  - **6D pose**
Real-Time Seamless Single Shot 6D Object Pose Prediction
Tekin, Sinha, Fua 2018 (in CVPR, to appear)

- Single-shot 2D object detection (YOLO, SSD)
- Our CNN predicts 2D projections of 3D bounding box vertices (+ centroid). We run PnP solver on 9 2D-3D correspondences.
- Accurate, fast (50-90 fps); detects multiple objects in one pass.
Real-Time Seamless Single Shot 6D Object Pose Prediction
Tekin, Sinha, Fua 2018 (in CVPR, to appear)

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)
Real-Time Seamless Single Shot 6D Object Pose Prediction
Tekin, Sinha, Fua 2018 (in CVPR, to appear)
Real-Time Seamless Single Shot 6D Object Pose Prediction
Tekin, Sinha, Fua 2018 (in CVPR, to appear)
Real-Time Seamless Single Shot 6D Object Pose Prediction
Tekin, Sinha, Fua 2018 (in CVPR, to appear)

- Accuracy w.r.t. two metrics (2D projection, 3D overlap)
  - Percentage of test images where the error was within a threshold

<table>
<thead>
<tr>
<th>2D metric</th>
<th>3D metric</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Method</strong></td>
<td>w/o Refinement</td>
</tr>
<tr>
<td>Ape</td>
<td>-</td>
</tr>
<tr>
<td>Benchvise</td>
<td>-</td>
</tr>
<tr>
<td>Cam</td>
<td>-</td>
</tr>
<tr>
<td>Can</td>
<td>-</td>
</tr>
<tr>
<td>Cat</td>
<td>-</td>
</tr>
<tr>
<td>Driller</td>
<td>-</td>
</tr>
<tr>
<td>Duck</td>
<td>-</td>
</tr>
<tr>
<td>Eggbox</td>
<td>-</td>
</tr>
<tr>
<td>Glue</td>
<td>-</td>
</tr>
<tr>
<td>Holepuncher</td>
<td>-</td>
</tr>
<tr>
<td>Iron</td>
<td>-</td>
</tr>
<tr>
<td>Lamp</td>
<td>-</td>
</tr>
<tr>
<td>Phone</td>
<td>-</td>
</tr>
<tr>
<td><strong>Average</strong></td>
<td>69.5</td>
</tr>
</tbody>
</table>
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.
Summary

- **Image – Image Correspondence**
  - Stereo Matching
    - Algorithmic improvements with different trade-offs
  - Unified Stereoscopic Scene flow estimation
    - Main insight: Solving *more* improves accuracy but *also* *speed*

- **Image – 3D model alignment**
  - 6D Object detection and Pose Estimation
    - Predict 2D *control point locations* in image; solve pose algebraically
Collaborators

Tatsunori Taniai
RIKEN, Tokyo

Daniel Scharstein
Middlebury College

Rick Szeliski
Facebook

Drew Steedly
Microsoft

Yoichi Sato
Univ. of Tokyo

Adarsh Kowdle
Google

Bugra Tekin
EPFL

Pascal Fua
EPFL