Fast Plane Extraction in Organized Point Clouds Using Agglomerative Hierarchical Clustering

Chen Feng¹), Yuichi Taguchi²), and Vineet R. Kamat¹)

¹) University of Michigan, USA
²) Mitsubishi Electric Research Labs, USA

June 4, 2014
• Real-time plane extraction is crucial to various applications in robotics, computer vision, and 3D modeling:
  – Table-top object manipulation
  – Landmarks for SLAM
  – Compact and semantic scene modeling
• Real-time plane extraction is crucial to various applications in robotics, computer vision, and 3D modeling:
  – Table-top object manipulation
  – Landmarks for SLAM
  – Compact and semantic scene modeling
• We present an efficient and reliable fast plane extraction algorithm applicable to organized point clouds, such as depth maps obtained by Kinect sensors.
• Previous Work
• Previous Work
  – RANSAC-based
    • “Surfels” from Hough Transform (Oehler et al. 2011)
    • RANSAC on local region (Taguchi et al. 2013; Hulik et al. 2012; Lee et al. 2012)
• Previous Work
  – RANSAC-based
    • “Surfels” from Hough Transform (Oehler et al. 2011)
    • RANSAC on local region (Taguchi et al. 2013; Hulik et al. 2012; Lee et al. 2012)
  – Region-grow-based
    • Point-plane distance/MSE threshold (Hahnel et al. 2003; Poppinga et al. 2008)
    • Surface normal deviation threshold (Holz & Behnke 2012)
    • Line segments grow (Georgiev et al. 2011)
• Previous Work
  – RANSAC-based
    • “Surfels” from Hough Transform (Oehler et al. 2011)
    • RANSAC on local region (Taguchi et al. 2013; Hulik et al. 2012; Lee et al. 2012)
  – Region-grow-based
    • Point-plane distance/MSE threshold (Hahnel et al. 2003; Poppinga et al. 2008)
    • Surface normal deviation threshold (Holz & Behnke 2012)
    • Line segments grow (Georgiev et al. 2011)
  – Graph-based (Strom et al. 2010; Wang et al. 2013)
• Previous Work
  – RANSAC-based
    • “Surfels” from Hough Transform (Oehler et al. 2011)
    • RANSAC on local region (Taguchi et al. 2013; Hulik et al. 2012; Lee et al. 2012)
  – Region-grow-based
    • Point-plane distance/MSE threshold (Hahnel et al. 2003; Poppinga et al. 2008)
    • Surface normal deviation threshold (Holz & Behnke 2012)
    • Line segments grow (Georgiev et al. 2011)
  – Graph-based (Strom et al. 2010; Wang et al. 2013)
  – Other
    • Normal space clustering (Holz et al. 2011)
    • Gradient-of-depth feature (Enjarini et al. 2012)
• Previous Work

Average FPS for VGA (640x480) point clouds
• **Analogy to Line Regression** *(Nguyen et al. 2005; April Robotics Toolkit, 2010)*
  
  – Exploit the neighborhood information given by the order of points
• Analogy to Line Regression (Nguyen et al. 2005; April Robotics Toolkit, 2010)
  – Exploit the neighborhood information given by the order of points
• **Analogy to Line Regression** (Nguyen et al. 2005; April Robotics Toolkit, 2010)
  
  – Exploit the neighborhood information given by the order of points

![Diagram of 2D point sequences and double linked list]

2D point sequences

Build double linked list

a b c d e f g h i j
• Analogy to Line Regression (Nguyen et al. 2005; April Robotics Toolkit, 2010)

  – Exploit the neighborhood information given by the order of points
• Analogy to Line Regression (Nguyen et al. 2005; April Robotics Toolkit, 2010)
  – Exploit the neighborhood information given by the order of points
• **Analogy to Line Regression** (Nguyen et al. 2005; April Robotics Toolkit, 2010)
  
  – Exploit the neighborhood information given by the order of points

![Diagram of 2D point sequences and extracted line segments]
• Non-trivial Generalization to 3D
• Non-trivial Generalization to 3D
  – None-overlapping nodes
• Non-trivial Generalization to 3D
  – None-overlapping nodes
• Non-trivial Generalization to 3D
  – None-overlapping nodes

  – Number of merging attempts
    \( \leq 2 \)
• Non-trivial Generalization to 3D
  – None-overlapping nodes
  – Number of merging attempts $\leq 2$
• Algorithm Overview

**Agglomerative Hierarchical Clustering**

- **(a) Initialize graph**
- **(b) Find node A with min MSE**
- **(c) Merge with neighbor node B which gives min merging MSE**
- **(d) Extract Coarse Planes**
- **(e) Refine details**

**Repeat if merging MSE ≤ threshold**

Otherwise don’t merge but extract A
• Algorithm Overview
• Graph Initialization
• Graph Initialization
  – Non-overlapping node initialization
• **Graph Initialization**
  – Non-overlapping node initialization
  – Rejecting “bad” nodes
• Graph Initialization
  – Non-overlapping node initialization
  – Rejecting “bad” nodes

1) High MSE
• Graph Initialization
  – Non-overlapping node initialization
  – Rejecting “bad” nodes

2) Missing Data

1) High MSE
• Graph Initialization
  – Non-overlapping node initialization
  – Rejecting “bad” nodes

2) Missing Data
3) Depth Discontinuities
1) High MSE
• Graph Initialization
  – Non-overlapping node initialization
  – Rejecting “bad” nodes

2) Missing Data

3) Depth Discontinuities

4) At Boundary Between Planes

1) High MSE
• Graph Initialization
  – Non-overlapping node initialization
  – Rejecting “bad” nodes

  – Good! Avoid per-point normal estimation
• Agglomerative Hierarchical Clustering
• Agglomerative Hierarchical Clustering
• Agglomerative Hierarchical Clustering

1 Cluster Step(s)
• Agglomerative Hierarchical Clustering

Current min MSE node
Best node to merge

1 Cluster Step(s)
• Agglomerative Hierarchical Clustering

2 Cluster Step(s)
• Agglomerative Hierarchical Clustering

3 Cluster Step(s)
• Agglomerative Hierarchical Clustering

300 Cluster Step(s)
• Agglomerative Hierarchical Clustering

472 Cluster Step(s)
• Average Number of Merging Attempts
• **Average Number of Merging Attempts**
  – Small irrespective of initial number of nodes
• Average Number of Merging Attempts
  – Small irrespective of initial number of nodes
  – Planar graph! Average node degree < 6
• Average Number of Merging Attempts
  – Small irrespective of initial number of nodes
  – Planar graph! Average node degree < 6
  – Merging is empirically a constant-time operation
    • $O(n \log n)$, only arise from maintaining the min-heap
• Implementation Details
• Implementation Details
  – Disjoint set
• Implementation Details
  – Disjoint set
  – Min-heap
• Implementation Details
  – Disjoint set
  – Min-heap
  – Second-order statistics
• Implementation Details
  – Disjoint set
  – Min-heap
  – Second-order statistics
  – Depth discontinuity/MSE threshold
    (Holzer et al. IROS 2012; Khoshelham & Elberink, 2012)
• Implementation Details
  – Disjoint set
  – Min-heap
  – Second-order statistics
  – Depth discontinuity/MSE threshold
    (Holzer et al. IROS 2012; Khoshelham & Elberink, 2012)
  – Avoid strip-like initial node shape
• Segmentation Refinement
• Segmentation Refinement
  – Artifacts

Sawtooth
• Segmentation Refinement
  – Artifacts

Sawtooth

Unused Data
• Segmentation Refinement
  – Artifacts

Sawtooth

Over-segmentation
• Segmentation Refinement
  – Artifacts

  – Pixel-wise region-grow refinement
    • Only check boundary blocks and points
• Segmentation Refinement
  – Artifacts

– Pixel-wise region-grow refinement
  • Only check boundary blocks and points
**Simulated Data**

- Robustness to uniformly distributed depth noise (Georgiev et al., IROS 2011)
- Noise magnitude $E = 0, 10, \ldots, 200\text{mm}$
- Ground truth depth ranges from $1396\text{mm}$ to $3704\text{mm}$
• Real-World Kinect Data
  – 2102 frames of an indoor scene
  – 640 × 480 pixel/frame
• Real-World Kinect Data
  – 2102 frames of an indoor scene
  – 640 × 480 pixel/frame

initial node size vs. average processing time

27.3 ± 6.9ms/frame > 35Hz
• Real-World Kinect Data

Algorithm Breakdown

1) Graph Initialization
• SegComp Datasets (Hoover et al. PAMI 1996)

– ABW-TEST

(a) 21/27, 1 under, 4 misses, 3 noises
(b) 18/19, 1 miss
(c) 14/17, 3 misses
(d) 15/17, 2 misses
(e) 10/10

– PERCEPTRON-TEST

(a) 6/6
(b) 6/6
(c) 10/11, 1 over, 1 noise
(d) 12/13, 1 miss, 1 noise
(e) 20/30, 1 over, 1 under, 7 misses, 3 noises
- **SegComp Benchmark** (Gotardo et al. CVPR 2003; Oehler et al. ICIRA 2011; Holz & Behnke IAS 2012)

<table>
<thead>
<tr>
<th>Approach</th>
<th>SegComp ABW data set (30 test images) by Hoover et al. [26], assuming 80% pixel overlap as in [27]</th>
<th>SegComp PERCEPTRON data set (30 test images) by Hoover et al. [26], assuming 80% pixel overlap as in [27]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Regions in ground truth</td>
<td>Correctly detected</td>
</tr>
<tr>
<td>USF [27]</td>
<td>15.2</td>
<td>12.7 (83.5%)</td>
</tr>
<tr>
<td>WSU [27]</td>
<td>15.2</td>
<td>9.7 (63.8%)</td>
</tr>
<tr>
<td>UB [27]</td>
<td>15.2</td>
<td>12.8 (84.2%)</td>
</tr>
<tr>
<td>UE [27]</td>
<td>15.2</td>
<td>13.4 (88.1%)</td>
</tr>
<tr>
<td>OU [27]</td>
<td>15.2</td>
<td>9.8 (64.4%)</td>
</tr>
<tr>
<td>PPU [27]</td>
<td>15.2</td>
<td>6.8 (44.7%)</td>
</tr>
<tr>
<td>UA [27]</td>
<td>15.2</td>
<td>4.9 (32.2%)</td>
</tr>
<tr>
<td>UFPR [27]</td>
<td>15.2</td>
<td>13.0 (85.5%)</td>
</tr>
<tr>
<td>Oehler et al. [2]</td>
<td>15.2</td>
<td>11.1 (73.0%)</td>
</tr>
<tr>
<td>Holz et al. [8]</td>
<td>15.2</td>
<td>12.2 (80.1%)</td>
</tr>
<tr>
<td><strong>Ours</strong></td>
<td>15.2</td>
<td>12.8 (84.2%)</td>
</tr>
<tr>
<td></td>
<td>SegComp ABW data set (30 test images) by Hoover et al. [26], assuming 80% pixel overlap as in [27]</td>
<td>SegComp PERCEPTRON data set (30 test images) by Hoover et al. [26], assuming 80% pixel overlap as in [27]</td>
</tr>
<tr>
<td>USF [27]</td>
<td>14.6</td>
<td>8.9 (60.9%)</td>
</tr>
<tr>
<td>WSU [27]</td>
<td>14.6</td>
<td>5.9 (40.4%)</td>
</tr>
<tr>
<td>UB [27]</td>
<td>14.6</td>
<td>9.6 (65.7%)</td>
</tr>
<tr>
<td>UE [27]</td>
<td>14.6</td>
<td>10.0 (68.4%)</td>
</tr>
<tr>
<td>UFPR [27]</td>
<td>14.6</td>
<td>11.0 (75.3%)</td>
</tr>
<tr>
<td>Oehler et al. [2]</td>
<td>14.6</td>
<td>7.4 (50.1%)</td>
</tr>
<tr>
<td>Holz et al. [8]</td>
<td>14.6</td>
<td>11.0 (75.3%)</td>
</tr>
<tr>
<td><strong>Ours</strong></td>
<td>14.6</td>
<td>8.9 (60.9%)</td>
</tr>
</tbody>
</table>
• We presented an efficient plane extraction algorithm based on agglomerative clustering for organized point clouds.
• We presented an efficient plane extraction algorithm based on agglomerative clustering for organized point clouds.

• We analyzed the complexity of the clustering algorithm and shown that it is log-linear in the number of initial nodes.
• We presented an efficient plane extraction algorithm based on agglomerative clustering for organized point clouds.

• We analyzed the complexity of the clustering algorithm and shown that it is log-linear in the number of initial nodes.

• We demonstrated real-time performance with the accuracy comparable to state-of-the-art algorithms.
• Acknowledgement
  – MERL
    • Jay Thornton
    • Srikumar Ramalingam
  – Rackham Graduate School, University of Michigan
  – NSF Student Travel Grant
Chen Feng (in Chinese: 冯晨)
PhD Candidate

Department of Civil and Environmental Engineering
University of Michigan, Ann Arbor

E-mail: cforrest@umich.edu
Web: http://www.umich.edu/~cforrest/