Monocular Visual-Inertial Odometry with Planar Regularities

Chuchu Chen*, Patrick Geneva*, Yuxiang Peng, Woosik Lee, and Guoquan Huang

Robot Perception and Navigation Group (RPNG)
University of Delaware, USA
Introduction

- Man-made environments provide rich structural information (e.g. planes...)
- Estimation of planes enables scene understanding (e.g. AR / VR)
- Constraining 3D features to planes through point-on-plane regularization can improve efficiency (reduce state)
Introduction

- Man-made environments provide **rich structural** information (e.g. planes...)
- Estimation of **planes** enables scene understanding (e.g. AR / VR)
- Constraining 3D features to planes through point-on-plane **regularization** can improve efficiency (reduce state)

**Key Contributions:**

- Novel plane detection and tracking with only a **monocular** camera
- **Efficient** filter-based VIO with planar regularities
- **Open sourced** code and dataset
Monocular Plane Feature Detection and Tracking

- Sparse Point Features:
  - FAST detection
  - KLT provides frame-to-frame tracking
Monocular Plane Feature Detection and Tracking

- **Sparse Point Features:**
  - FAST detection
  - KLT provides frame-to-frame tracking

- **Point Feature Meshing**
  - 3D point feature recovery
  - 2D mesh (Delaunay triangulation)
  - Compute normals of triangles
Monocular Plane Feature Detection and Tracking

- **Sparse Point Features:**
  - FAST detection
  - KLT provides frame-to-frame tracking

- **Point Feature Meshing**
  - 3D point feature recovery
  - 2D mesh (Delaunay triangulation)
  - Compute normals of triangles
Monocular Plane Feature Detection and Tracking

- **Sparse Point Features:**
  - FAST detection
  - KLT provides frame-to-frame tracking

- **Point Feature Meshing**
  - 3D point feature recovery
  - 2D mesh (Delaunay triangulation)
  - Compute normals of triangles

- **Vertex Normals:**
  - Use neighboring triangles normals to compute avg. vertex normals
  - Remove invalidate vertices with high variance (points on the edge)
Monocular Plane Feature Detection and Tracking

- **Sparse Point Features:**
  - FAST detection
  - KLT provides frame-to-frame tracking

- **Point Feature Meshing**
  - 3D point feature recovery
  - 2D mesh (Delaunay triangulation)
  - Compute normals of triangles

- **Vertex Normals:**
  - Use neighboring triangles normals to compute avg. vertex normals
  - Remove invalidate vertices with high variance (points on the edge)
Monocular Plane Feature Detection and Tracking

- **Sparse Point Features:**
  - FAST detection
  - KLT provides frame-to-frame tracking

- **Point Feature Meshing**
  - 3D point feature recovery
  - 2D mesh (Delaunay triangulation)
  - Compute normals of triangles

- **Vertex Normals:**
  - Use neighboring triangles normals to compute avg. vertex normals
  - Remove invalidate vertices with high variance (points on the edge)

- **Vertex Matching Heuristics**
  - Pairwise comparison with neighbors
  - Normal difference, point-to-plane distance, statistical filter to remove outlier
Monocular Plane Feature Detection and Tracking

- **Sparse Point Features:**
  - FAST detection
  - KLT provides frame-to-frame tracking

- **Point Feature Meshing**
  - 3D point feature recovery
  - 2D mesh (Delaunay triangulation)
  - Compute normals of triangles

- **Vertex Normals:**
  - Use neighboring triangles normals to compute avg. vertex normals
  - Remove invalidate vertices with high variance (points on the edge)

- **Vertex Matching Heuristics**
  - Pairwise comparison with neighbors
  - Normal difference, point-to-plane distance, statistical filter to remove outlier

Shown efficient and robust from real-world experiments (e.g. ~3-4 ms on EuRoC Mav)
Planar Regularities

Camera bearing meas.
\[ z_c = h(x_C, {}^G p_f) + n_c \]

Point: \(^G p_f\)
Planar Regularities

\[ z_c = h(x_C, g_p_f) + n_c \]
Planar Regularities

$z_c = h(x, p_f) + n_c$

Plane: $G_{\Pi} = \mathbf{G}_n^G G_d^G$

Point: $G_{p_f}$

Camera bearing meas.
Planar Regularities

Plane: $G\Pi = G^r_n G^d_d$

Point: $^Gp_f$

Camera bearing meas.

$z_c = h(x_C, ^Gp_f) + n_c$

Point-on-plane regularity

$z_d = (^Gp_f^T G^r_n - G^d_d) + n_d$
Planar Regularities

Camera bearing meas.
\[ z_c = h(x_C, G_p f) + n_c \]

Plane:
\[ G_{\Pi} = G_n G_d \]

Point:
\[ G_{p f} \]

Point-on-plane regularity
\[ z_d = (G_p f \cdot G_n - G_d) + n_d \]
Planar Regularities

Plane: $G_{\Pi} = G_n G_d$

Point: $G_{p_f}$

Point-on-plane regularity

$z_d = (G_{p_f}^{-1} G_n - G_d) + n_d$
Planar Regularities

- Stack linearized bearing $\mathbf{z}_C$ and regularity measurements $\hat{\mathbf{z}}_d$
  \[
  \begin{bmatrix}
  \hat{\mathbf{z}}_C \\
  \hat{\mathbf{z}}_d 
  \end{bmatrix} = \begin{bmatrix}
  \mathbf{H}^C_T \\
  0
  \end{bmatrix} \mathbf{x}_C + \begin{bmatrix}
  \mathbf{H}^C_f \\
  \mathbf{H}^d_f
  \end{bmatrix} \mathbf{G}_f \mathbf{p}_f + \begin{bmatrix}
  0 \\
  \mathbf{H}^d_{\pi}
  \end{bmatrix} \mathbf{G}_\Pi + \begin{bmatrix}
  \mathbf{n}_C \\
  \mathbf{n}_d
  \end{bmatrix}
  \]
  \[
  \Rightarrow \hat{\mathbf{z}} = \mathbf{H}^T_T \mathbf{x}_C + \mathbf{H}^G_f \mathbf{p}_f + \mathbf{H}^G_{\pi} \mathbf{G}_\Pi + \mathbf{n}
  \]

Point-on-plane regularity
  \[
  z_d = (\mathbf{G}_f^T \mathbf{n} - \mathbf{G}_d) + n_d
  \]
Planar Regularities

- Stack linearized bearing $z_c$ and regularity measurements $\tilde{z}_d$

$$
\begin{bmatrix}
\tilde{z}_c \\
\tilde{z}_d
\end{bmatrix} =
\begin{bmatrix}
H_c^T \\
0
\end{bmatrix}
\dot{x}_C +
\begin{bmatrix}
H_c^T \\
H_f^T
\end{bmatrix}
G \tilde{p}_f +
\begin{bmatrix}
0 \\
H_d^T
\end{bmatrix}
G \tilde{\Pi} +
\begin{bmatrix}
n_c \\
n_d
\end{bmatrix}
$$

$$
\Rightarrow \tilde{z} = H_T \dot{x}_C + H_f^T \tilde{p}_f + H_\pi G \tilde{\Pi} + n
$$

- MSCKF and SLAM feature updates to balance accuracy and efficiency

- Planar regularities can constrain both in-state and out-of-state features

Point-on-plane regularity

$$
\tilde{z}_d = (G^T_{pf} n - G_d) + n_d
$$
Planar Regularities

- Stack linearized bearing $z_c$ and regularity measurements $\tilde{z}_d$:
  \[
  \begin{bmatrix}
  \tilde{z}_c \\
  \tilde{z}_d
  \end{bmatrix} = \begin{bmatrix} H^c_T \\ 0 \end{bmatrix} \tilde{x}_C + \begin{bmatrix} H^c_f \\
  H^d_f \\
  H^\pi \\
  0
  \end{bmatrix} G \tilde{p}_f + \begin{bmatrix} 0 \\
  H^d_f \\
  H^\pi \\
  0
  \end{bmatrix} G \tilde{\Pi} + \begin{bmatrix} n_c \\
  n_d
  \end{bmatrix}
  \]
  \[
  \Rightarrow \tilde{z} = H_T \tilde{x}_C + H_f G \tilde{p}_f + H^\pi G \tilde{\Pi} + n
  \]

- MSCKF and SLAM feature updates to balance accuracy and efficiency

- Planar regularities can constrain both in-state and out-of-state features

Point-on-plane regularity:
\[
\tilde{z}_d = (G \tilde{p}_f^{T} G n - G d) + n_d
\]

SLAM planes shown impressive performance by providing long-term tracking and regularization
Experimental Results: Detection and tracking

- Planes can be tracked much longer than points to better constrain the motion
- Efficient to include planes without an additional sensor
Experimental Results: Detection and tracking

- Planes can be tracked much longer than points to better constrain the motion
- Efficient to include planes without an additional sensor

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Feat. / PL</th>
<th>PL / Frame</th>
<th>Track Len.</th>
<th>PL Active</th>
<th>Time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>V1_01</td>
<td>19.6 ± 13.3</td>
<td>2.9 ± 1.3</td>
<td>53.4 ± 74.0</td>
<td>0.9 ± 0.7</td>
<td>3.3 ± 0.7</td>
</tr>
<tr>
<td>V1_03</td>
<td>10.1 ± 9.4</td>
<td>0.7 ± 1.0</td>
<td>24.9 ± 26.0</td>
<td>0.0 ± 0.2</td>
<td>2.0 ± 0.7</td>
</tr>
<tr>
<td>table_01</td>
<td>27.3 ± 13.1</td>
<td>2.7 ± 1.1</td>
<td>61.1 ± 227.6</td>
<td>1.1 ± 0.5</td>
<td>3.5 ± 0.7</td>
</tr>
<tr>
<td>table_02</td>
<td>82.0 ± 58.7</td>
<td>2.2 ± 1.3</td>
<td>49.1 ± 249.2</td>
<td>1.2 ± 0.6</td>
<td>4.1 ± 0.9</td>
</tr>
<tr>
<td>table_03</td>
<td>33.9 ± 21.3</td>
<td>3.0 ± 1.2</td>
<td>88.5 ± 337.4</td>
<td>1.5 ± 0.6</td>
<td>4.0 ± 0.7</td>
</tr>
</tbody>
</table>

*All computational results were performed in a single thread on an Intel(R) Xeon(R) E3-1505Mv6 @ 3.00GHz.
Experimental Results

Table 1: Simulation RPE results (degree / cm). M: MSCKF feature, S: SLAM feature, PT: Point, PL: Plane.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>60m</th>
<th>80m</th>
<th>100m</th>
<th>120m</th>
<th>NEES(3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>M-PT</td>
<td>0.37  / 4.3</td>
<td>0.44  / 5.0</td>
<td>0.50  / 5.6</td>
<td>0.55  / 6.2</td>
<td>3.39 / 1.75</td>
</tr>
<tr>
<td>M-PT &amp; M-PL</td>
<td>0.37  / 4.3</td>
<td>0.43  / 4.9</td>
<td>0.48  / 5.5</td>
<td>0.53  / 6.1</td>
<td>3.34 / 1.72</td>
</tr>
<tr>
<td>M-PT &amp; MS-PL</td>
<td><strong>0.36</strong> / 3.6</td>
<td><strong>0.42</strong> / 4.1</td>
<td><strong>0.48</strong> / 4.6</td>
<td><strong>0.53</strong> / 5.1</td>
<td><strong>3.99</strong> / 1.44</td>
</tr>
<tr>
<td>MS-PT</td>
<td>0.30  / 3.6</td>
<td>0.35  / 4.1</td>
<td>0.40  / 4.6</td>
<td>0.43  / 5.1</td>
<td>3.45 / 1.63</td>
</tr>
<tr>
<td>MS-PT &amp; M-PL</td>
<td>0.29  / 3.5</td>
<td><strong>0.33</strong> / 4.0</td>
<td><strong>0.37</strong> / 4.5</td>
<td><strong>0.41</strong> / 4.9</td>
<td><strong>3.09</strong> / 1.44</td>
</tr>
<tr>
<td>MS-PT &amp; MS-PL</td>
<td><strong>0.29</strong> / 2.9</td>
<td>0.35  / 3.3</td>
<td>0.39  / 3.7</td>
<td>0.42  / 4.1</td>
<td>3.38 / 1.20</td>
</tr>
</tbody>
</table>

- Impressive performance gain by introducing SLAM planes
- **Consistent** estimation with planar regularities!
Experimental Results

Table 2: Real-world AR Table Dataset (degree / cm)

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>table_01</th>
<th>table_02</th>
<th>table_03</th>
<th>table_04</th>
<th>Time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>M-PT</td>
<td>0.45 / 6.8</td>
<td>0.85 / 2.4</td>
<td>1.37 / 5.6</td>
<td>0.83 / 7.5</td>
<td>8.7 ± 1.7</td>
</tr>
<tr>
<td>M-PT &amp; M-PL</td>
<td>0.52 / 6.5</td>
<td>0.91 / 2.5</td>
<td>1.44 / 5.9</td>
<td>0.87 / 7.1</td>
<td>13.3 ± 3.2</td>
</tr>
<tr>
<td>M-PT &amp; MS-PL</td>
<td>0.67 / 4.6</td>
<td>0.72 / 2.0</td>
<td>0.96 / 3.0</td>
<td>0.75 / 3.2</td>
<td>13.9 ± 2.9</td>
</tr>
<tr>
<td>MS-PT</td>
<td>1.15 / 5.7</td>
<td>1.79 / 4.1</td>
<td>2.41 / 6.9</td>
<td>1.28 / 5.7</td>
<td>9.4 ± 2.0</td>
</tr>
<tr>
<td>MS-PT &amp; M-PL</td>
<td>1.32 / 5.5</td>
<td>0.89 / 2.5</td>
<td>1.03 / 4.5</td>
<td>1.10 / 4.7</td>
<td>15.0 ± 3.9</td>
</tr>
<tr>
<td>MS-PT &amp; MS-PL</td>
<td>1.25 / 5.1</td>
<td>0.65 / 2.3</td>
<td>1.05 / 4.6</td>
<td>0.79 / 5.0</td>
<td>14.7 ± 3.2</td>
</tr>
<tr>
<td>VINS-Fusion [1]</td>
<td>1.62 / 5.8</td>
<td>1.32 / 3.0</td>
<td>1.47 / 7.6</td>
<td>1.75 / 5.6</td>
<td>35.6 ± 17.0*</td>
</tr>
<tr>
<td>OKVIS [2]</td>
<td>2.48 / 9.0</td>
<td>2.01 / 7.7</td>
<td>3.94 / 15.3</td>
<td>2.05 / 16.2</td>
<td>85.5 ± 32.6*</td>
</tr>
</tbody>
</table>

- Including **planes** improves VIO accuracy!
- **Efficient** performance
- Outperform state-of-the-art point-based systems
Monocular Visual-Inertial Odometry with Planar Regularities

Chuchu Chen*, Patrick Geneva*, Yuxiang Peng, Woosik Lee, and Guoquan Huang

● Novel plane detection and tracking on monocular camera
● Efficient filter-based VIO with planar regularities to improve accuracy
● Open sourced code and dataset

ccchu@udel.edu
chuchuchen.net

Source Code
AR Table Dataset