Local Tracking and Mapping for Direct Visual SLAM

Pablo Rodríguez Palafox
Technical University of Munich
Chair of Helicopter Technology & Computer Vision Group
Garching, October 11, 2019
Problem Statement

Direct Sparse Odometry, Engel et al.
Problem Statement

When doing marginalization of keyframes / points in VO,

reusing map points

(when revisiting already mapped areas)

is not possible.
Problem Statement

Past, Present, and Future of Simultaneous Localization And Mapping: Towards the Robust-Perception Age, Cadena et al.
Problem Statement

New KF
Problem Statement

New KF
Approach

Original idea from: *Direct Sparse Mapping*, Zubizarreta et al.
Approach
Approach

Temporal

Covisible

New KF
Approach

Temporal

Covisible

New KF

Approach

Inactive

Active Covisible

Inactive Covisible

Temporal
Approach

Overview

Input Video

→ Tracking
  SE(3) alignment to Local Covisibility Window

→ Distance Estimation
  Need KF?
  Refine temporal KFs
  Create new KF

→ Map Optimization

→ Recompute Local Covisibility Window
Approach
Overview

Input Video

Tracking
SE(3) alignment to Local Covisibility Window

Distance Estimation
Need KF?
Refine temporal KFs
Create new KF

Map Optimization

Recompute Local Covisibility Window
Approach

Tracking
Approach

Tracking
Approach

Tracking
Approach
Building the Image Pyramids
Approach

(Inverse Distance Formulation)

\[ d_p := \text{Inverse distance} \]

\[ d_p = \frac{1}{\|p\|} \]

\[ p = \|p\| \cdot b = \frac{b}{d_p} \]
Approach

Tracking

Direct Image Alignment

**Forward Additive**

\[ r_i(T \oplus \xi) = I_t(w(T \oplus \xi, u)) - I_h(u) \]

**Inverse Compositional**

\[ r_i(\xi) = I_h(w(I \oplus \xi, u)) - I_t(w(T, u)) \]
Approach

Tracking

Direct Image Alignment

Forward Additive

\[ r_i(T \oplus \xi) = I_t(w(T \oplus \xi, u)) - I_h(u) \]
Inverse Compositional Direct Image Alignment

\[ r_i(\xi) = I_h(w(I \oplus \xi, u)) - I_t(w(T, u)) \]
Approach

Overview

Input Video → Tracking
- SE(3) alignment to Local Covisibility Window

Distance Estimation
- Need KF?
  - Refine temporal KFs
  - Create new KF

Map Optimization

Recompute Local Covisibility Window
Approach
Candidate Point Tracking

Epipolar Curve Search

$C_{kf}$

Current frame

$C_{cur}$

$P_{max}^s$

$P_{min}^s$

$p_L = \alpha P_{max}^s + (1 - \alpha)P_{min}^s$

$d_{max}$

$d_{min}$

Figure based on that presented in OmniDSO
Approach
Candidate Point Tracking

Epipolar Curve Search

\[ p' = Rp + t \]
\[ = R \frac{b}{dp} + t \]

\[ b' = \frac{p'}{||p'||} = \frac{R \frac{b}{dp} + t}{||R \frac{b}{dp} + t||} \]
Approach

Candidate Point Tracking

Epipolar Curve Search

\[ p' = Rp + t = R \frac{b}{d_p} + t \]

\[ b'_x = \frac{p'_x}{||p'||} = \frac{r_0 \cdot b}{d_p} + t_x \]

\[ b'_y = \frac{p'_y}{||p'||} = \frac{r_1 \cdot b}{d_p} + t_y \]
Approach
Candidate Point Tracking

Epipolar Curve Search

\[ p' = Rp + t \]
\[ = R \frac{b}{d_p} + t \]

\[
\begin{align*}
    b'_x &= \frac{p'_x}{\|p'\|} = \frac{r_0 \cdot b}{d_p} + t_x \\
    b'_y &= \frac{p'_y}{\|p'\|} = \frac{r_1 \cdot b}{d_p} + t_y \\
\end{align*}
\]

\[
\begin{align*}
    b'_x &= \frac{r_0 \cdot b}{d_p} + t_x \\
    b'_y &= \frac{r_1 \cdot b}{d_p} + t_y \\
\end{align*}
\]
Approach
Candidate Point Tracking

Epipolar Curve Search

\[ p' = Rp + t \]
\[ = R \frac{b}{d_p} + t \]

\[ d_p = \frac{b_y r_0 \cdot b - b_x' r_1 \cdot b}{b_x' t_y - b_y' t_x} \]
Approach

Overview

Input Video

Tracking
SE(3) alignment to Local Covisibility Window

Distance Estimation
Need KF?
Refine temporal KFs
Create new KF

Map Optimization

Recompute Local Covisibility Window
Approach
Recomputing Local Covisibility Window
Approach
Recomputing Local Covisibility Window
Approach

Recomputing Local Covisibility Window
Approach
Overview

Input Video → Tracking
SE(3) alignment to Local Covisibility Window → Distance Estimation
Need KF?
→ Refine temporal KFs
→ Create new KF

→ Recompute Local Covisibility Window

Map Optimization
Approach

Photometric Bundle Adjustment

- Ceres
  (w/o coarse-to-fine)

- Manual Solver
  (w/ and w/o coarse-to-fine)
Approach

Robustification

Coarse-to-Fine

Residual Distribution

- normal distribution before removing gross outliers
- normal distribution after removing gross outliers
- t-distribution before removing gross outliers
- t-distribution after removing gross outliers

probability density function (PDF)
Results

Influence of Candidate Point Selection on Tracking
Results
Candidate Point Tracking
Results
Candidate Point Tracking
Results
Candidate Point Tracking

Upgraded to Landmarks
Results
Candidate Point Tracking
Results
Candidate Point Tracking

Upgraded to Landmarks
Results
Candidate Point Tracking

Remaining as Candidate Points
Results
Candidate Point Tracking

Remaining as Candidate Points

Upgraded to Landmarks
Results
Photometric Bundle Adjustment

Before PBA

PBA (Photometric Bundle Adjustment)

After PBA
## Results

### Photometric Bundle Adjustment

Max Number of Iterations per Level: **10**

<table>
<thead>
<tr>
<th>Solver Type</th>
<th>Metric</th>
<th>Iterations</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Total</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>Manual w/o pyrs</td>
<td>ATE (m)</td>
<td>0.00131</td>
</tr>
<tr>
<td>Ceres w/o pyrs</td>
<td>Runtime (s)</td>
<td>0.43</td>
</tr>
<tr>
<td>Manual w/ pyrs</td>
<td>ATE (m)</td>
<td>0.00054</td>
</tr>
<tr>
<td>Ceres w/ pyrs</td>
<td>Runtime (s)</td>
<td>2.42</td>
</tr>
</tbody>
</table>

Table 7.3: Results for different solver types and configurations.

Table 7.4: Detailed iteration counts for each solver type and configuration.

---

Figure 7.3: Sample map before Photometric Bundle Adjustment. Out of the total 12 cameras (green pyramids), 6 correspond to the first pass of the sensor across the region, while the other half belong to a second pass of the sensor. The purple pyramids represent groundtruth poses, while the purple line interpolates the groundtruth trajectory of the sensor. Note there are two purple lines, each corresponding to a different pass of the camera over the region.
Results

Full System
Results

Full System

RMSE ATE: 0.00589 m
Results

Full System

RMSE ATE: 1.05 m
Conclusion

Direct **SLAM**

+ Modular & flexible **framework** for future development

**Input Video**

**Tracking**

SE(3) alignment to Local Covisibility Window

**Distance Estimation**

Need KF?

- Refine temporal KFs
- Create new KF

**Map Optimization**

**Recompute Local Covisibility Window**
6. Back-End

Figure 6.3: Inverse distance computation from the best match.

\[ b_0^t = p_0^t k_{0^t} = r_0 \cdot b_{dp} + t_x^t k_{0^t} \] \hspace{1cm} (6.11)

\[ b_0^t = p_0^t k_{0^t} = r_1 \cdot b_{dp} + t_y^t k_{0^t} \] \hspace{1cm} (6.12)

Divide the former by the latter:

\[ \frac{b_0^t}{b_0^t} = \frac{r_0 \cdot b_{dp} + t_x^t k_{0^t}}{r_1 \cdot b_{dp} + t_y^t k_{0^t}} \] \hspace{1cm} (6.13)

Multiply (6.14) by \( b_{dp} \):

\[ b_0^t r_1 \cdot b_{dp} + b_0^t t_y^t = b_0^t r_0 \cdot b_{dp} + b_0^t t_x^t \] \hspace{1cm} (6.15)

We can finally solve for \( b_{dp} \) (i.e., the inverse distance of the candidate point with respect to the...
Future Work

- Refine system and find good balance for user-defined parameters
- Pose-graph optimization to close larger loops:
  - Double-window optimization (accurate pose-point & soft pose-pose)
- Test the system on real datasets (e.g., EuRoC)
Thank you very much for your attention.

Technical University of Munich
Pablo Rodríguez Palafox
pablo.rodriguez-palafox@tum.de