Robust VI-SLAM and HD-Map Reconstruction for Location-based Augmented Reality

Guofeng Zhang
State Key Lab of CAD&CG, Zhejiang University
SLAM

• **A Basic Problem in Robotics & Computer Vision**
  • Simultaneously estimate the device pose and 3D scene structure in an unknown environment.

• **Wide Applications**
  • Augmented Reality, Virtual Reality
  • Robotics, Automated Driving
Visual-Inertial SLAM

• **Main Sensors**
  - Single or Multiple Cameras
  - Inertial Measurement Unit

• **Advantages**
  - Low cost
  - High localization accuracy at least in a small workspace
  - Inside-out solution: no scene setup
Traditional SLAM Framework

**Input**
- Sensor data

**Foreground Thread**
- Compute the device pose in real-time

**Background Thread**
- Optimizing local or global maps
- Loop closure detection

**Output**
- Device poses
- 3D points

**Optimize & reduce accumulation error**

**Loop closure detection**
Challenges of Visual SLAM and VI-SLAM

Challenge 1

Accuracy and Robustness
- Dynamic environment
- Lots of outliers
- Unstable optimization

Challenge 2

Real-Time Performance
- Large scene
- Huge computation
- Limited computation resources on a mobile device
Key Idea for Robust Estimation

Robust Estimation

Accuracy of Constraints

Sufficiency of Constraints

outliers detection and removement

distribution prior

change detection

motion priors

structure priors

... $I_{t-3}$ $I_{t-2}$ $I_{t-1}$ $I_t$

$H_1$ $H_2$ $H_3$
Outliers Detection and Removal

- Detect the changed 3D points and update the keyframes adaptively

ISMAR 2013

Our SLAM Result
Visual-Inertial Odometry with Multi-Plane Priors

Motivation
- A fast and lightweight VIO method for low-end mobile devices.
- Planes commonly exist in human-made scenes, and can be utilized.

Plane extraction and expansion from VIO point cloud
- Reprojection Consensus

\[ \varepsilon_k = \sum_i \| u_{ik}(\lambda_k) - \tilde{u}_{ik} \|^2, \quad \varepsilon_k^+ = \sum_i \| u_{ik}(\lambda_k^+) - \tilde{u}_{ik} \|^2 \]

- Ordinary RPE
- Point-on-Plane RPE

Plane Constraints
- VIP-PnP : solving the BA as if some points lying on planes

Structureless Plane-Distance Error
- Augmented triangulation with Plane Constraint
- Minimize Point-to-Plane Distance Error

For 1 landmark, \( m \) reprojection error \(\to\) 1 structureless error

Experimental Results

### Trajectories on TUM-VI Outdoors1

![Graph showing trajectories on TUM-VI Outdoors1 with lines for Ours, VINS-Mono, and DSO]

<table>
<thead>
<tr>
<th>Dataset</th>
<th>ORB-SLAM2</th>
<th>SVO2</th>
<th>DSO</th>
<th>VINS-Mono</th>
<th>PVIO</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Loop</td>
<td>Loop</td>
<td></td>
<td>Loop</td>
<td>Plane</td>
</tr>
<tr>
<td>MH.01</td>
<td>0.92</td>
<td>0.03</td>
<td>0.10</td>
<td>0.06</td>
<td>0.05</td>
</tr>
<tr>
<td>MH.02</td>
<td>0.03</td>
<td>0.03</td>
<td>0.12</td>
<td>0.07</td>
<td>0.05</td>
</tr>
<tr>
<td>MH.03</td>
<td>0.17</td>
<td>0.35</td>
<td>0.40</td>
<td>0.20</td>
<td>0.31</td>
</tr>
<tr>
<td>MH.04</td>
<td>0.15</td>
<td>0.37</td>
<td>0.43</td>
<td>0.37</td>
<td>0.29</td>
</tr>
<tr>
<td>MH.05</td>
<td>0.06</td>
<td>0.40</td>
<td>0.30</td>
<td>0.30</td>
<td>0.28</td>
</tr>
<tr>
<td>V1.01</td>
<td>0.03</td>
<td>0.03</td>
<td>0.07</td>
<td>0.05</td>
<td>0.12</td>
</tr>
<tr>
<td>V1.02</td>
<td>0.15</td>
<td>0.03</td>
<td>0.21</td>
<td>0.09</td>
<td>0.11</td>
</tr>
<tr>
<td>V1.03</td>
<td>0.49</td>
<td>0.10</td>
<td>0.09</td>
<td>0.11</td>
<td>0.09</td>
</tr>
<tr>
<td>V2.01</td>
<td>0.03</td>
<td>0.03</td>
<td>0.11</td>
<td>0.03</td>
<td>0.13</td>
</tr>
<tr>
<td>V2.02</td>
<td>0.15</td>
<td>0.03</td>
<td>0.11</td>
<td>0.13</td>
<td>0.16</td>
</tr>
<tr>
<td>V2.03</td>
<td>0.70</td>
<td>0.40</td>
<td>0.40</td>
<td>0.16</td>
<td>0.29</td>
</tr>
<tr>
<td>Room1</td>
<td>x</td>
<td>0.10</td>
<td>x</td>
<td>x</td>
<td>0.06</td>
</tr>
<tr>
<td>Room2</td>
<td>x</td>
<td>0.12</td>
<td>x</td>
<td>x</td>
<td>0.11</td>
</tr>
<tr>
<td>Room3</td>
<td>(0.04)</td>
<td>x</td>
<td>x</td>
<td>(0.04)</td>
<td>0.12</td>
</tr>
<tr>
<td>Corridor1</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>5.43</td>
</tr>
<tr>
<td>Outdoor1</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>74.55</td>
</tr>
</tbody>
</table>

**Notes:**
- Single thread
- No loop-closure & relocalization
Key Idea for Efficient Estimation

- **Bundle Adjustment**
  - Jointly optimize all cameras and points
  - Time-consuming and require large memory space

\[
\text{arg min}_{C_1, \ldots, C_{N_c}, X_1, \ldots, X_{N_p}} \sum \left\| \pi(X_i, C_j) - x_{ij} \right\|^2
\]
Sparse Bundle Adjustment

\[
\arg \min_{C_1, \ldots, C_{N_c}, X_1, \ldots, X_{N_p}} \sum \left\| \pi(X_i, C_j) - x_{ij} \right\|^2
\]

Sparse Bundle Adjustment

- Runtime significantly increases with the number of cameras

Four Steps in one iteration
1. Normal equation
2. Schur complement
3. Solve cameras
4. Solve points by back substitution
Batch VS Incremental BA

- Batch BA

- Incremental BA
Batch VS Incremental BA

• Batch BA

• Incremental BA
Batch VS Incremental BA

- Batch BA

- Incremental BA
Batch VS Incremental BA

- Batch BA

- Incremental BA
Incremental Bundle Adjustment

- **Most cameras and points are nearly unchanged**
  - Contribution of most projection functions nearly remains the same
  - No need to re-compute at each iteration

- **Incremental approaches**
  - iSAM, iSAM2, SLAM++
  - Our EIBA & ICE-BA

- **Key ideas**
  - Incremental update: makes maximum use of intermediate computation for efficiency
  - Detect the actually changed variables and adaptively update them
ICE-BA: Incremental, Consistent and Efficient BA for VI-SLAM

- Factor graph representation

ICE-BA: Incremental, Consistent and Efficient BA for VI-SLAM

- Factor graph representation

- New cameras or points come
ICE-BA: Incremental, Consistent and Efficient BA for VI-SLAM

- Factor graph representation

- Points have changed after iteration
ICE-BA: Incremental, Consistent and Efficient BA for VI-SLAM

- Factor graph representation
- Cameras have changed after iteration
Step 1: Normal Equation

- **Batch BA**

  \[
  \begin{align*}
  U = 0; \ V = 0; \ W = 0; \ u = 0; \ v = 0 \\
  \text{for each point } j \text{ and each camera } i \in \mathcal{V}_j \text{ do} \\
  \quad \text{Construct linearized equation (11)} \\
  \quad U_{ii}^+ = J_{C_{ij}}^T J_{C_{ij}} \\
  \quad V_{jj}^+ = J_{X_{ij}}^T J_{X_{ij}} \\
  \quad u_i^+ = J_{C_{ij}}^T e_{ij} \\
  \quad v_j^+ = J_{X_{ij}}^T e_{ij} \\
  \quad W_{ij}^+ = J_{C_{ij}}^T J_{X_{ij}} \\
  \text{end for}
  \end{align*}
  \]

- **ICE-BA**

  \[
  \begin{align*}
  \text{for each point } j \text{ and each camera } i \in \mathcal{V}_j \text{ that } C_i \text{ or } X_j \text{ is changed do} \\
  \quad \text{Construct linearized equation (11)} \\
  \quad S_{ii}^- = A_{ij}^U; \quad A_{ij}^U = J_{C_{ij}}^T J_{C_{ij}}; \quad S_{ii}^+ = A_{ij}^U \\
  \quad V_{jj}^- = A_{ij}^V; \quad A_{ij}^V = J_{X_{ij}}^T J_{X_{ij}}; \quad V_{jj}^+ = A_{ij}^V \\
  \quad g_i^- = b_{ij}^u; \quad b_{ij}^u = J_{C_{ij}}^T e_{ij}; \quad g_i^+ = b_{ij}^u \\
  \quad v_j^- = b_{ij}^v; \quad b_{ij}^v = J_{X_{ij}}^T e_{ij}; \quad v_j^+ = b_{ij}^v \\
  \quad W_{ij}^+ = J_{C_{ij}}^T J_{X_{ij}} \\
  \quad \text{Mark } V_{jj} \text{ updated}
  \end{align*}
  \]
Step 2: Schur Complement

- **Batch BA**

\[ S = U \]
\[
\text{for each point } j \text{ and each camera pair } (i_1, i_2) \in V_j \times V_j \\
\quad \text{do} \\
\quad \quad S_{i_1 i_2} = W_{i_1 j} V_{j j}^{-1} W_{i_2 j}^T \\
\text{end for} \\
\]
\[
g = u \\
\text{for each point } j \text{ and each camera } i \in V_j \text{ do} \\
\quad \quad g_i = W_{i j} V_{j j}^{-1} v_j \\
\text{end for} \\
\]

- **ICE-BA**

\[
\text{for each point } j \text{ that } V_{j j} \text{ is updated and each camera pair } (i_1, i_2) \in V_j \times V_j \text{ do} \\
\quad S_{i_1 i_2}^+ = A_{i_1 i_2}^S \\
\quad A_{i_1 i_2}^S = W_{i_1 j} V_{j j}^{-1} W_{i_2 j}^T \\
\quad S_{i_1 i_2}^- = A_{i_1 i_2}^S \\
\text{end for} \\
\]
\[
\text{for each point } j \text{ that } V_{j j} \text{ is updated and each camera } i \in V_j \text{ do} \\
\quad g_i^+ = b_{i j}^g; \quad b_{i j}^g = W_{i j} V_{j j}^{-1} v_j; \quad g_i^- = b_{i j}^g \\
\text{end for} \]
Sub-track Improvement for Local BA

- In Local BA, most points may be observed by most frames in the sliding window
  - Dense Schur complement
  - A large portion need to be re-computed
- Split the original long feature track $X_i$ into several short overlapping sub-tracks $X_{i_1}, X_{i_2}, \ldots$
Efficiency Comparison

- Local BA (LBA)
  - ICE-BA (50 frames)
  - Ceres (10 key frames)

- Global BA (GBA)
  - ICE-BA: almost $O(1)$
  - Ceres: $O(n^2)$
Efficiency Comparison

- **Local BA (LBA)**
  - ICE-BA (50 frames)
  - OKVIS (8 key frames)
  - 10x speedup

- **Global BA (GBA)**
  - ICE-BA: steady and smooth
  - iSAM2: steep and peaks
  - 20x speedup
## Accuracy Comparison on EuRoc dataset

<table>
<thead>
<tr>
<th>Seq.</th>
<th>Ours w/ loop</th>
<th>Ours w/o loop</th>
<th>OKVIS</th>
<th>SVO</th>
<th>iSAM2</th>
</tr>
</thead>
<tbody>
<tr>
<td>MH_01</td>
<td>0.11</td>
<td>0.09</td>
<td>0.22</td>
<td>0.06</td>
<td>0.07</td>
</tr>
<tr>
<td>MH_02</td>
<td>0.08</td>
<td>0.07</td>
<td>0.16</td>
<td>0.08</td>
<td>0.11</td>
</tr>
<tr>
<td>MH_03</td>
<td>0.05</td>
<td>0.11</td>
<td>0.12</td>
<td>0.16</td>
<td>0.12</td>
</tr>
<tr>
<td>MH_04</td>
<td>0.13</td>
<td>0.16</td>
<td>0.18</td>
<td>-</td>
<td>0.16</td>
</tr>
<tr>
<td>MH_05</td>
<td>0.11</td>
<td>0.27</td>
<td>0.29</td>
<td>0.63</td>
<td>0.25</td>
</tr>
<tr>
<td>V1_01</td>
<td>0.07</td>
<td>0.05</td>
<td>0.03</td>
<td>0.06</td>
<td>0.07</td>
</tr>
<tr>
<td>V1_02</td>
<td>0.08</td>
<td>0.05</td>
<td>0.06</td>
<td>0.12</td>
<td>0.08</td>
</tr>
<tr>
<td>V1_03</td>
<td>0.06</td>
<td>0.11</td>
<td>0.12</td>
<td>0.21</td>
<td>0.12</td>
</tr>
<tr>
<td>V2_01</td>
<td>0.06</td>
<td>0.12</td>
<td>0.05</td>
<td>0.22</td>
<td>0.10</td>
</tr>
<tr>
<td>V2_02</td>
<td>0.04</td>
<td>0.09</td>
<td>0.07</td>
<td>0.16</td>
<td>0.13</td>
</tr>
<tr>
<td>V2_03</td>
<td>0.11</td>
<td>0.17</td>
<td>0.14</td>
<td>-</td>
<td>0.20</td>
</tr>
<tr>
<td>Avg</td>
<td><strong>0.08</strong></td>
<td><strong>0.12</strong></td>
<td><strong>0.14</strong></td>
<td><strong>0.20</strong></td>
<td><strong>0.13</strong></td>
</tr>
</tbody>
</table>
## Comparison and Analysis

<table>
<thead>
<tr>
<th></th>
<th>iSAM2</th>
<th>EIBA / ICE-BA</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Motion style</strong></td>
<td>✓ keep forward</td>
<td>✗ to and fro</td>
</tr>
<tr>
<td><strong>Variable ordering</strong></td>
<td>By algebraic method</td>
<td>By tricks of standard BA</td>
</tr>
<tr>
<td></td>
<td>- The best ordering are changed time to time</td>
<td>- Don’t care about which camera/point comes first</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Always marginalize points first</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- PCG explicitly leverage sparsity of camera Hessian</td>
</tr>
<tr>
<td><strong>Incremental calculation vs sparseness</strong></td>
<td>Trade off</td>
<td>Re-linearize wherever necessary without affecting sparseness</td>
</tr>
<tr>
<td></td>
<td>- Fix linearization: matrix becomes denser and denser during to and fro motion.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Reordering: re-calculation</td>
<td></td>
</tr>
</tbody>
</table>

Source Code: [https://github.com/baidu/ICE-BA](https://github.com/baidu/ICE-BA)
Key Idea: Cloud-Edge-Terminal Combination

Cloud + Edge Computing

3D Registration
- Multi-sensor fusion
- SLAM/semantic SLAM
- Object Recognition & Tracking
- 3D Reconstruction of Scenes

Large-Scale AR
- Light Estimation
- Realistic Rendering
- Occlusions Handling
- Physical Simulation

HD Map

Sensor Data

5G
High Bandwidth, Low Latency

Optimization results
Pose information
3D map
Real-time rendering & feedback

Localization & Navigation

Multiple-Persons Sharing

Apply

Apply
Visual Localization & AR Navigation
### Localization & AR Navigation

#### Traditional Solutions

- Error up to 10 meters
- Not available in indoor environments

#### Visual Solutions

- GPS
- WIFI, Blue Tooth

#### Advantages
- Low cost
- Non-intrusive
- High precision
- Intuitive with AR effect

#### Challenges
- Lack of visual features
- Environment change
- Heavy computation
Main Techniques in Visual Localization & AR Navigation

**Sparse Map Reconstruction**
- Extract visual features
- Recover the 3D structure

**Dense Map Reconstruction**
- Handling occlusions and collisions
- Free-viewpoint 3D navigation

**Visual Localization & Tracking**
- Real-time 6DOF camera pose recovery for AR
Sparse Map Reconstruction

Challenges

• Many textureless regions
• Visual Ambiguity
• Large Scale

Key Ideas

• Capture Panorama Videos
• Integrating SLAM with SfM
• Divide and conquer
Dense Map Reconstruction

Challenges
- Many textureless regions
- Large-scale reconstruction

Key Ideas
- Accurate dense depth maps estimation and fusion
  - By multi-level feature matching
- Extendable accurate dense mesh reconstruction
  - Out-of-core reconstruction of large-scale meshes
Visual Localization & Tracking

Challenges

- Real-time
- Long distance
- View point, illumination, appearance variations

Key Ideas

- Cloud and terminal cooperation
- Tightly couple SLAM with global relocalization
- Learning based visual features
AR Multiple-Persons Sharing
Reconstructed 3D Map
Softwares

- ENFT-SFM or LS-ACTS
  - [http://www.zjucvg.net/ls-acts/ls-acts.html](http://www.zjucvg.net/ls-acts/ls-acts.html)
- RKSLAM: [http://www.zjucvg.net/rkslam/rkslam.html](http://www.zjucvg.net/rkslam/rkslam.html)
- RDSLAM: [http://www.zjucvg.net/rdslam/rdslam.html](http://www.zjucvg.net/rdslam/rdslam.html)
- ACTS: [http://www.zjucvg.net/acts/acts.html](http://www.zjucvg.net/acts/acts.html)
- SenseSLAM
  - [http://www.zjucvg.net/senseslam/](http://www.zjucvg.net/senseslam/)
Source Code

- ENFT-SfM
  - https://github.com/zju3dv/ENFT-SfM
- Segment-based Bundle Adjustment
  - https://github.com/zju3dv/SegmentBA
- Efficient Incremental Bundle Adjustment
  - EIBA: https://github.com/zju3dv/EIBA
  - ICE-BA: https://github.com/baidu/ICE-BA
VSLAM/VISLAM Technology Trends (1)

- **Reduce Textureless Problem**
  - Edge Tracking
  - Direct Tracking
  - Learning based methods or incorporating scene prior/semantic information
    (predict scene layout/semantic information, depth map and normal map)

Edge Feature (Klein et al., 2008)  
Plane Feature (Concha et al., 2014)  
Semi-dene Tracking (Engel et al., 2014)  
Scene Layout (Salas et al., 2015)  
Semantic SLAM (Nicholson et al., 2018)
VSLAM/VISLAM Technology Trends (2)

- **Multiple Sensors Fusion**
  - Combining GPS, depth camera, odometer, WiFi, 5G

Survey of cellular mobile radio localization methods (Peral-Rosado et al., 2017)

Even-camera based VIO (Rebecq et al., 2017)

Multiple Sensors Fusion

https://www.intellias.com/sensor-fusion-autonomous-cars-helps-avoid-deaths-road/
VSLAM/VISLAM Technology Trends (3)

**Dense 3D Reconstruction**
- Real-time single / multiple camera based methods
- Real-time depth camera based methods
- Real-time reconstruction of non-rigid objects

- **Dense 3D Reconstruction based AR application**
  (Schöps et al., 2014)

- **MobileFusion**
  (Ondrúška et al., 2015)

- **RKD-SLAM**
  (Liu et al., 2017)

- **DynamicFusion**
  (Newcombe et al., 2015)

- **Keyframe-based Dense Planar SLAM**
  (Hsiao et al., 2017)

- **CNN-SLAM**
  (Tateno et al., 2017)
Thanks!