Bundle Adjustment Without Iterative Structure Estimation and its Application to Navigation

Vadim Indelman

Robotics and Intelligent Machines (RIM) Center
College of Computing
Georgia Institute Of Technology

April 2012
Introduction

- Navigation-aiding techniques are essential for reducing dead-reckoning or inertial navigation errors
- Vision-based navigation-aiding methods are commonly used
  - In particular, when GPS is unavailable
  - Typical scenarios include: indoor, urban, underwater environments
- Existing approaches for navigation aiding – typically use filtering techniques
- Another approach for information fusion: incremental optimization
- Bundle adjustment is commonly used for solving the full Simultaneous Localization and Mapping (SLAM) problem
  - Can be applied for navigation-aiding, but is computationally expensive!

- This work – computationally efficient bundle adjustment for fusing all vision observations
  - In other words – focus on the “Localization part” in SLAM
Introduction

- **Problem Formulation:**
  - **Input:** Sequence of incoming images
    Initial solution for camera poses
  - **Goal/Output:**
    Optimize camera poses in a sequence of images
    Recover coordinates of observed 3D points (Optional)
  - **Assumptions:**
    Solved correspondence – matching features between images are known
    Known camera calibration
  - **Applications:**
    - Autonomous navigation applications (initial solution – dead reckoning)
    - Multi-agent systems
    - Simultaneous Localization and Mapping (SLAM)
    - Structure from motion, augmented reality, …
Introduction (cont.)

- Bundle Adjustment (BA): Minimize overall re-projection errors

- **Light Bundle Adjustment** - Main idea:
  - Reduce computational cost by avoiding optimizing the 3D points
    - Use *multi-view constraints* instead of projection equations
    - Less variables to optimize
  - Recover landmarks based on optimized camera poses

- Previous work (in the context of using multi-view constraints for motion estimation)
  - **Sliding window of triplets of images**: “Incremental motion estimation through local bundle adjustment”, Z. Zhang and Y. Shan, 2001
  - **Avoid structure estimation using trifocal tensors**: “Threading fundamental matrices”, S. Avidan and A. Shashua, 2001
  - **Trifocal constraints for BA**: “Relative Bundle Adjustment based on Trifocal Constraints”, R. Steffen et al., 2010
  - **Three-view constraints**: “Distributed Vision-Aided Cooperative Localization and Navigation Based on Three-View Geometry”, V. Indelman et al., 2012
Contents

- Introduction
- Bundle Adjustment
- Light Bundle Adjustment
- Structure Reconstruction
- Results
- Conclusions
Bundle Adjustment (BA)

- Scenario:
  - $N$ cameras\views, observing $M$ 3D points
  - Not all cameras necessarily observe all 3D points

- Projection equation:
  - Between the $i$-th 3D point and the $j$-th camera pose
  \[
p_{ij} = K_j \begin{bmatrix} R_j & t_j \end{bmatrix} P_i = M_j P_i
  \]

- Optimized cost function – sum of re-projection errors (Mahalanobis distance):
  \[
  J^{BA} = \sum_{j=1}^{N} \sum_{i=1}^{M} \| p_{ij} - \text{Proj}(x_j, P_i) \|^2_{\Sigma_{ij}}
  \]
  - $p_{ij}$: Measured pixel
  - $x_j$: Pose of image $j$: $R_j$, $t_j$

- Optimized variables:
  \[
  x^{BA} = \begin{bmatrix} x_1^T & \ldots & x_N^T & P_1^T & \ldots & P_M^T \end{bmatrix}^T \in \mathbb{R}^{(6N+3M) \times 1}
  \]
Light Bundle Adjustment (LBA)

- Use multi-view constraints instead of projection equations
- Overall multi-view constraints (for all image sequence): \( h(\hat{x}, \hat{p}) = 0 \)
- Optimized (constrained) cost function:
  \[
  J^{LBA} \triangleq \sum_{j=1}^{N} \sum_{i=1}^{M} \left\| p_{ij} - \hat{p}_{ij} \right\|_{\Sigma_{ij}}^2 - 2\lambda^T h(\hat{x}, \hat{p})
  \]

- Cost function does not involve structure parameters!
- Approximation: Optimize only camera poses
- Optimized variables:
  \[
  x^{LBA} = \begin{bmatrix} x_1^T & \ldots & x_N^T \end{bmatrix}^T \in \mathbb{R}^{6N \times 1}
  \]

- As opposed to \( x^{BA} \in \mathbb{R}^{(6N+3M) \times 1} \)
Three-View Constraints

- Multi-view constraints – in this work: **Three-view constraints**
- Consider three views $k$, $l$ and $m$ observing the same unknown landmark

- $q_k$, $q_l$, $q_m$: Line of sight for pixel $p$: $q = K^{-1}p$
- $t_{i \rightarrow j}$: translation from camera $i$ to camera $j$
Three-View Constraints

- Three-view constraints for cameras $k$, $l$ and $m$:
  \[
  \bar{q}_k^T (\bar{t}_{k\rightarrow l} \times \bar{q}_l) = 0 \\
  \bar{q}_l^T (\bar{t}_{l\rightarrow m} \times \bar{q}_m) = 0 \\
  (\bar{q}_l \times \bar{q}_k) \cdot (\bar{q}_m \times \bar{t}_{l\rightarrow m}) = (\bar{q}_k \times \bar{t}_{k\rightarrow l}) \cdot (\bar{q}_m \times \bar{q}_l)
  \]
  - $\bar{a}$: ideal value of some vector $a$
  - $q$: Line of sight for pixel $p$: $q = K^{-1} p$
  - $t_{i \rightarrow j}$: translation from camera $i$ to camera $j$
  - All vectors should be expressed in the same coordinate frame

- **First two equations**: epipolar constraints between views $k,l$ and $l,m$
- **Third equation**: Scale consistency
- **Three-view constraints**:
  - Allow (also) to reduce position errors along motion heading in straight trajectories
  - Have been applied to: navigation aiding (incl. loop closures), cooperative navigation
Light Bundle Adjustment (cont.)

- In practice, due to image noise and errors in camera poses, the constraints will not be satisfied.

- Define residual error:
  \[ z_1 \triangleq q_k^T (t_{k \to l} \times q_l) \]
  \[ z_2 \triangleq q_l^T (t_{l \to m} \times q_m) \]
  \[ z_3 \triangleq (q_l \times q_k)^T (q_m \times t_{l \to m}) - (q_k \times t_{k \to l})^T (q_m \times q_l) \]

- Constraints error of views \( k, l, m \) observing the \( i \)-th 3D point:
  \[ z_{i, (k,l,m)} \triangleq \begin{bmatrix} z_1 & z_2 & z_3 \end{bmatrix}^T \]

- Non-linear (known) function \( h_{i, (k,l,m)} \):
  \[ z_{i, (k,l,m)} = h_{i, (k,l,m)}(\hat{x}_k, \hat{x}_l, \hat{x}_m, p_{ik}, p_{il}, p_{im}) \]

- \( h_{i, (k,l,m)} \) is part of the overall multi-view constraints function \( h(x, p) \)
Light Bundle Adjustment (cont.)

- What happens if a 3D point is observed by more than 3 views?
- Assume the $i$-th 3D point is observed by $n_i$ cameras: $\{k_1, \ldots, k_{n_i}\}$
  - Should use only independent constraints
  - After the first three views – apply a reduced version of three-view constraints
  - Views $1,2,3$:
    - $z_1$: epipolar constraint between views $k=1$ and $l=2$
    - $z_2$: epipolar constraint: between views $l=2$ and $m=3$
    - $z_3$: three-view constraint: between views $k=1$, $l=2$ and $m=3$

$z_{i}^{(k,l,m)} = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix}$
Light Bundle Adjustment (cont.)

- What happens if a 3D point is observed by more than 3 views?
- Assume the $i$-th 3D point is observed by $n_i$ cameras: $\{k_1, \ldots, k_{n_i}\}$
  - Should use only independent constraints
  - After the first three views – apply a reduced version of three-view constraints
  - Views 2,3,4:
    - $z_1$: epipolar constraint between views $k=2$ and $l=3$ is **not applied**
    - $z_2$: epipolar constraint: between views $l=3$ and $m=4$
    - $z_3$: three-view constraint: between views $k=2,l=3$ and $m=4$

\[ z_i^{(k,l,m)} \triangleq \begin{bmatrix} z_2 \\ z_3 \end{bmatrix} \]

Reduced version
Light Bundle Adjustment (cont.)

- Overall constraints for observing the \(i\)-th 3D point in images \(\{k_1, \ldots, k_{n_i}\}\)

\[
\mathbf{z}_i \triangleq \begin{bmatrix}
\mathbf{z}^{(k_1k_2k_3)}_i \\
\mathbf{z}^{(k_2k_3k_4)*}_i \\
\vdots \\
\mathbf{z}^{(k_{n_i-2}k_{n_i-1}k_{n_i})*}_i
\end{bmatrix}
\]

- Take into account all \(M\) observed 3D points:

\[
\mathbf{h} (\mathbf{x}, \mathbf{p}) \triangleq \begin{bmatrix}
\mathbf{z}_1^T \\
\vdots \\
\mathbf{z}_M^T
\end{bmatrix}
\]

- Optimized cost function

\[
J^{LBA} = \|\mathbf{p} - \hat{\mathbf{p}}\|_\Sigma^2 - 2\lambda^T \mathbf{h}(\hat{\mathbf{x}}, \hat{\mathbf{p}})
\]
Basic Example

- Scenario
  - 3 landmarks
  - 5 cameras

- Constraints:

\[ z_1 = z_{(1,3,4)}^{(1)} \in \mathbb{R}^{3 \times 1} \]
\[ z_2 = \begin{bmatrix} z_{(1,2,4)}^{(1)} \\ z_{(2,4,5)}^{(2)} \end{bmatrix} \in \mathbb{R}^{5 \times 1} \]
\[ z_3 = z_{(3,4,5)}^{(3)} \in \mathbb{R}^{3 \times 1} \]

\[ z = \begin{bmatrix} z_1^T \\ z_2^T \\ z_3^T \end{bmatrix}^T \in \mathbb{R}^{11 \times 1} \]
Light Bundle Adjustment (cont.)

- As in BA, optimization is up to a 7-DOF transformation
  - A proper regularization should be used

- A relative formulation is used:
  - Camera poses are expressed relative to the first frame
    - Fixes 6 of the 7 DOFs

- Scale constraint – fix the range between the first two views (from initial solution)

\[
J^{LBA} = \| p - \hat{p} \|_{\Sigma}^2 - 2\lambda_1^T h \left( \hat{x}^{rel}, \hat{p} \right) - 2\lambda_2^T g \left( \hat{x}^{rel} \right)
\]
Structure Reconstruction in LBA

- Structure reconstruction
  - Performed after convergence of LBA optimization
  - All or some of the observed 3D points can be recovered
  - Standard structure reconstruction procedure
  - Based on the optimized camera poses

- Observation of the $i$-th 3D point by the $j$-th camera:
  \[ p_{ij} = K_j \begin{bmatrix} R_j & t_j \end{bmatrix} P_i = M_j P_i \]

- Taking into account all cameras observing the $i$-th 3D point:
  \[ A \tilde{P}_i = b \]

- Standard estimation\optimization
Results

- Pozzoveggiani dataset (http://profs.sci.univr.it/~fusiello/demo/samantha/):
  - ~45 images
  - BA solution for camera poses and landmarks (considered as ground truth)
Results (cont.)

- Initial Conditions for LBA: BA camera pose solution corrupted with errors:
  - Position: 50 m error (1σ)
  - Rotation: 0.1 deg error (1σ)
  - Pixels: 0.5 pixel error (1σ)

- Illustration of camera poses and observed 3D points (only part of the data is shown)
Results (cont.)

Position estimation

Position estimation errors
Results (cont.)

Structure estimation errors

Structure estimation errors - zoom
Conclusions

- **Light Bundle Adjustment**
  - Optimization of camera poses based on multi-view constraints
  - Structure estimation is not part of the optimization
  - Reduced computational cost
  - Structure reconstruction based on optimized camera poses

- **Applications**
  - Structure from motion
  - Mobile robotics and autonomous systems
  - Multi-agent systems