3D Vision

Marc Pollefeys, Viktor Larsson

Spring 2019
<table>
<thead>
<tr>
<th>Date</th>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feb 18</td>
<td>Introduction</td>
</tr>
<tr>
<td>Feb 25</td>
<td>Geometry, Camera Model, Calibration</td>
</tr>
<tr>
<td>Mar 4</td>
<td>Features, Tracking / Matching</td>
</tr>
<tr>
<td>Mar 11</td>
<td><strong>Project Proposals by Students</strong></td>
</tr>
<tr>
<td>Mar 18</td>
<td>Structure from Motion (SfM) + papers</td>
</tr>
<tr>
<td>Mar 25</td>
<td>Dense Correspondence (stereo / optical flow) + papers</td>
</tr>
<tr>
<td><strong>Apr 1</strong></td>
<td><strong>Bundle Adjustment &amp; SLAM + papers</strong></td>
</tr>
<tr>
<td>Apr 8</td>
<td><strong>Student Midterm Presentations</strong></td>
</tr>
<tr>
<td>Apr 15</td>
<td>Multi-View Stereo &amp; Volumetric Modeling + papers</td>
</tr>
<tr>
<td>Apr 22</td>
<td>Easter break</td>
</tr>
<tr>
<td>Apr 29</td>
<td>3D Modeling with Depth Sensors + papers</td>
</tr>
<tr>
<td>May 6</td>
<td>3D Scene Understanding + papers</td>
</tr>
<tr>
<td>May 13</td>
<td>4D Video &amp; Dynamic Scenes + papers</td>
</tr>
<tr>
<td>May 20</td>
<td>papers</td>
</tr>
<tr>
<td>May 27</td>
<td><strong>Student Project Demo Day = Final Presentations</strong></td>
</tr>
</tbody>
</table>
3D Vision – Class 6
Bundle Adjustment and SLAM

- [Montemerio, Thrun, Koller, Wegbreit, FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem, AAAI 2002]
- Section 2.5 from [Lee, Visual Mapping and Pose Estimation for Self-Driving Cars, PhD Thesis, ETH Zurich, 2014]

Slides : Gim Hee Lee
Lecture Overview

• Bundle Adjustment in Structure-from-Motion

• Simultaneous Localization & Mapping (SLAM)
Recap: Structure-From-Motion

• Two views initialization:
  – 5-Point algorithm (Minimal Solver)
  – 8-Point linear algorithm
  – 7-Point algorithm
Recap: Structure-From-Motion

- Triangulation: 3D Points

\[ E \to (R,t) \]
Recap: Structure-From-Motion

• Subsequent views: Perspective pose estimation
Recap: Structure-From-Motion
Bundle Adjustment

• Refinement step in Structure-from-Motion.
• Refine a visual reconstruction to produce jointly optimal 3D structures $P$ and camera poses $C$.
• Minimize total re-projection errors $\Delta z$.

Cost Function:

$$\underset{X}{\text{argmin}} \sum_i \sum_j \left\| x_{ij} - \pi(P_j, C_i) \right\|^2_{W_{ij}}$$

$$X = [P, C]$$
Bundle Adjustment

- Refinement step in Structure-from-Motion.
- Refine a visual reconstruction to produce jointly optimal 3D structures $P$ and camera poses $C$.
- Minimize total re-projection errors $\Delta z$.

Cost Function:

$$\arg\min_X \sum_i \sum_j \Delta z_{ij}^T W_{ij} \Delta z_{ij}$$

$W_{ij}^{-1}$: Measurement error covariance

$$X = [P, C]$$
Bundle Adjustment

• Minimize the cost function: $\arg\min_X f(X)$
  1. Gradient Descent
  2. Newton Method
  3. Gauss-Newton
  4. Levenberg-Marquardt
Bundle Adjustment

1. Gradient Descent

Initialization: $X_k = X_0$

Compute gradient: $g = \frac{\partial f(X)}{\partial X} \bigg|_{X=X_k} = \Delta Z^T W J$

Update: $X_k \leftarrow X_k - \eta g$

$J = \frac{\partial \pi}{\partial X}$ : Jacobian

$\eta$ : Step size

Slow convergence near minimum point!
Bundle Adjustment

2. Newton Method

$2^{nd}$ order approximation (Quadratic Taylor Expansion):

$$ f(X + \delta)|_{X=X_K} \approx f(X) + g\delta + \frac{1}{2} \delta^T H \delta |_{X=X_K} $$

Hessian matrix: $H = \frac{\partial^2 f(X + \delta)}{\partial \delta^2} |_{X=X_k}$

Find $\delta$ that minimizes $f(X + \delta)|_{X=X_K}$!
Bundle Adjustment

2. Newton Method

Differentiate and set to 0 gives:

$$\delta = -H^{-1}g$$

Update:  $$X_k \leftarrow X_k + \delta$$

Computation of $H$ is not trivial and might get stuck at saddle point!
Bundle Adjustment

3. Gauss-Newton

\[ H = J^T W J + \sum_i \sum_j \Delta Z_{ij} W_{ij} \frac{\partial^2 \pi_{ij}}{\partial \mathbf{X}^2} \]

\[ H \approx J^T W J \]

Normal equation:

\[ J^T W J \delta = -J^T W \Delta \mathbf{Z} \]

Update: \( X_k \leftarrow X_k + \delta \)

Might get stuck and slow convergence at saddle point!
Bundle Adjustment

4. Levenberg-Marquardt

Regularized Gauss-Newton with damping factor $\lambda$.

$$\left(J^T W J + \lambda I\right) \delta = -J^T W \Delta Z$$

$H_{LM}$

$\lambda \rightarrow 0$: Gauss-Newton (when convergence is rapid)

$\lambda \rightarrow \infty$: Gradient descent (when convergence is slow)

Adapt $\lambda$ during optimization:

- Decrease $\lambda$ when function value decreases
- Increase $\lambda$ otherwise
Structure of the Jacobian and Hessian Matrices

- Sparse matrices since 3D structures are locally observed.

\[
J = \begin{pmatrix}
\begin{array}{cccc}
A1 & A2 & B1 & B2 & B3 & B4 \\
A1 & A2 & C1 & C2 & C3 & C4 \\
A1 & A2 & D1 & D2 & D3 & D4 \\
A1 & A2 & E1 & E2 & E3 & E4 
\end{array}
\end{pmatrix}
\]
Efficiently Solving the Normal Equation

- Schur Complement: Exploit structure of $H$

\[
H_{LM} \delta = -J^T W \Delta Z
\]
Efficiently Solving the Normal Equation

- Schur Complement: Exploit structure of $H$

$$H_{LM} \delta = -J^T W \Delta Z$$
Efficiently Solving the Normal Equation

• Schur Complement: Obtain reduced system

\[ H_{LM} \delta = -J^T W \Delta Z \]

\[
\begin{bmatrix}
H_S & H_{SC} \\
H_{SC}^T & H_C
\end{bmatrix}
\begin{bmatrix}
\delta_S \\
\delta_C
\end{bmatrix} =
\begin{bmatrix}
\epsilon_S \\
\epsilon_C
\end{bmatrix}
\]

Multiply both sides by:

\[
\begin{bmatrix}
I & 0 \\
-H_{SC}^T H_S^{-1} & I
\end{bmatrix}
\]

\[
\begin{bmatrix}
H_S & H_{SC} \\
0 & H_C - H_{SC}^T H_S^{-1} H_{SC}
\end{bmatrix}
\begin{bmatrix}
\delta_S \\
\delta_C
\end{bmatrix} =
\begin{bmatrix}
\epsilon_S \\
\epsilon_C - \epsilon_S H_{SC}^T H_S^{-1}
\end{bmatrix}
\]

3D Structures
Camera Parameters
Efficiently Solving the Normal Equation

• Schur Complement: Obtain reduced system

\[
\begin{bmatrix}
H_S & H_{SC} \\
0 & H_C - H_{SC}^T H_S^{-1} H_{SC}
\end{bmatrix}
\begin{bmatrix}
\delta_S \\
\delta_C
\end{bmatrix} = \begin{bmatrix}
\varepsilon_S \\
\varepsilon_C - \varepsilon_S H_{SC}^T H_S^{-1}
\end{bmatrix}
\]

First solve for \( \delta_C \) from:

\[
(H_C - H_{SC}^T H_S^{-1} H_{SC})\delta_C = \varepsilon_C - \varepsilon_S H_{SC}^T H_S^{-1}
\]

Schur Complement
(Sparse and Symmetric Positive Definite Matrix)

Solve for \( \delta_{SC} \) by backward substitution.
Efficiently Solving the Normal Equation

\[(H_C - H_{sc}^T H_s^{-1} H_{sc}) \delta_C = \varepsilon_C - \varepsilon_s H_{sc}^T H_s^{-1} \quad \equiv \quad Ax = b\]

Don’t solve as \(x=\mathbf{A}^{-1}\mathbf{b}\): \(\mathbf{A}\) is sparse, but \(\mathbf{A}^{-1}\) is not!

• Use sparse matrix factorization to solve system

  1. LU Factorization \(A = LU\)  
  2. QR factorization \(A = QR\)  
  3. Cholesky Factorization \(A = LL^T\)

• Iterative methods

  1. Conjugate gradient
  2. Gauss-Seidel
Problem of Fill-In

Hessian

Natural Cholesky
Problem of Fill-In

• Reorder sparse matrix to minimize fill-in.

\[(P^T A P)(P^T x) = P^T b\]

Permutation matrix to reorder \(A\)

• NP-Complete problem.

• Approximate solutions:
  1. Minimum degree
  2. Column approximate minimum degree permutation
  3. Reverse Cuthill-McKee.
  4. ...
Problem of Fill-In
Robust Cost Function

• Non-linear least squares: \( \arg\min_X \sum_{ij} \Delta z_{ij}^T W_{ij} \Delta z_{ij} \)

• Maximum log-likelihood solution:

\[
\arg\min_X -\ln p(Z \mid X) = \arg\min_X -\ln \left( \prod_{ij} c_{ij} \exp(-\Delta z_{ij}^T W_{ij} \Delta z_{ij}) \right)
\]

\[
= \arg\min_X \sum_{ij} \Delta z_{ij}^T W_{ij} \Delta z_{ij}
\]

• Assume that:

1. \( X \) is a random variable that follows Gaussian distribution.
2. All observations are independent.
Robust Cost Function

• Gaussian distribution assumption is not true in the presence of outliers!
• Causes wrong convergences.
Robust Cost Function

\[ \arg\min_x \sum_{ij} \rho_{ij}(\Delta z_{ij}) = \arg\min_x \sum_{ij} \Delta z_{ij}^T S_{ij} \Delta z_{ij} \]

- Similar to iteratively re-weighted least-squares.
- Weight is iteratively rescaled with the attenuating factor \( \rho_{ij}'' \).
- Attenuating factor is computed based on current error.
Robust Cost Function

\[ \rho(.) \]

<table>
<thead>
<tr>
<th></th>
<th>( \rho(.) )</th>
<th>( \rho''(.) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cost function</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PDF</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Attenuation factor</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Squared-error**
  - Gaussian Distribution
  - Influence from high errors

- **Cauchy**
  - Cauchy Distribution
  - Reduced influence from high errors

- **Huber**
Robust Cost Function

Outliers are taken into account in Cauchy!
State-of-the-Art Solvers

• Google Ceres:
  – https://code.google.com/p/ceres-solver/

• g2o:
  – https://openslam.org/g2o.html

• GTSAM:
  – https://collab.cc.gatech.edu/borg/gtsam/

• Multicore Bundle Adjustment
Lecture Overview

• Bundle Adjustment in Structure-from-Motion

• Simultaneous Localization & Mapping (SLAM)
Simultaneous Localization & Mapping (SLAM)

• Robot navigates in unknown environment:
  – Estimate its own pose
  – Acquire a map model of its environment.

• Chicken-and-Egg problem:
  – Map is needed for localization (pose estimation).
  – Pose is needed for mapping.

• Highly related to Structure-From-Motion.
Full SLAM: Problem Definition

- Robot poses
- Observations
- Map landmarks
- Control actions
Full SLAM: Problem Definition

• Maximum a Posteriori (MAP) solution:

\[
\begin{align*}
\arg\max_{X,L} p(X,L | Z,U) &= \arg\max_{X_L} p(X_0) \prod_{i=1}^{M} p(x_i | x_{i-1}, u_i) \prod_{k=1}^{K} p(z_k | x_{ik}, l_{jk}) \\
\end{align*}
\]
Full SLAM

\[ \arg \max_{X,L} p(X,L \mid Z,U) = \arg \max_{X,L} p(X_0) \prod_{i=1}^{M} p(x_i \mid x_{i-1}, u_i) \prod_{k=1}^{K} p(z_k \mid x_{ik}, l_{jk}) \]

Negative log-likelihood

\[ = \arg \min_{X,L} \left\{ - \sum_{i=1}^{M} \ln p(x_i \mid x_{i-1}, u_i) - \sum_{k=1}^{K} \ln p(z_k \mid x_{ik}, l_{jk}) \right\} \]

Likelihoods:

\[ p(x_i \mid x_{i-1}, u_i) \propto \exp\left\{ -\|f(x_{i-1}, u) - x_i\|_\Lambda_i^2 \right\} \]

Process model

\[ p(z_k \mid x_{ik}, l_{jk}) \propto \exp\left\{ -\|h(x_{ik}, l_{jk}) - z_k\|_{\Sigma_k}^2 \right\} \]

Measurement model
Full SLAM

\[
\argmax_{X,L} p(X,L \mid Z,U) = \argmin_{X,L} \left\{ -\sum_{i=1}^{M} \ln p(x_i \mid x_{i-1}, u_i) - \sum_{k=1}^{K} \ln p(z_k \mid x_{ik}, l_{ik}) \right\}
\]

Putting the likelihoods into the equation:

\[
\argmax_{X,L} p(X,L \mid Z,U) = \argmin_{X,L} \left\{ \sum_{i=1}^{M} \| f(x_{i-1}, u_i) - x_i \|_{\Lambda_i}^2 + \sum_{k=1}^{K} \| h(x_{ik}, l_{ik}) - z_k \|_{\Sigma_k}^2 \right\}
\]

Minimization can be done with Levenberg-Marquardt (similar to bundle adjustment problem)!
Full SLAM

Normal Equations:

\[(J^T W J + \lambda I)\delta = -J^T W \Delta Z\]

Jacobian made up of \(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial u}, \frac{\partial h}{\partial x}, \frac{\partial h}{\partial l}\)

Weight made up of \(\Lambda_i, \Sigma_k\)

Can be solved with sparse matrix factorization or iterative methods

Solving the full SLAM problem rather expensive for larger scenes
Online SLAM: Problem Definition

• Estimate current pose $x_t$ and full map $L$:

$$p(x_t, L | Z, U) = \int \int \ldots \int p(X, L | Z, U) \, dx_1 \, dx_2 \ldots dx_{t-1}$$

Previous poses are marginalized out

• Inference with:
  1. (Extended) Kalman Filter (EKF SLAM)
  2. Particle Filter (FastSLAM)
EKF SLAM

• Assumes: pose $x_t$ and map $L$ are random variables that follow Gaussian distributions.
• Hence,

$$p(x_t, L | Z, U) \sim N(\mu, \Sigma)$$

• (Extended) Kalman Filter iteratively
  – Predicts pose & map based on process model
  – Corrects prediction based on observations
EKF SLAM

Prediction:
\[ \bar{\mu}_t = f(u_t, \mu_{t-1}) \]  
\[ \bar{\Sigma}_t = F_t \Sigma_{t-1} F_t^T + R_t \]

Correction:
\[ y_t = z_t - h(\bar{\mu}_t) \]
\[ K_t = \bar{\Sigma}_t H_t^T (H_t \bar{\Sigma}_t H_t^T + Q_t)^{-1} \]
\[ \mu_t = \bar{\mu}_t + K_t y_t \]
\[ \Sigma_t = (I - K_t H_t) \bar{\Sigma}_t \]

Measurement Jacobian  
\[ H_t = \frac{\partial h(\bar{\mu}_t)}{\partial x_t} \]

Process Jacobian  
\[ F_t = \frac{\partial f(u_t, \mu_{t-1})}{\partial x_{t-1}} \]
Covariance is a dense matrix that grows with increasing map features!

True robot and map states might not follow unimodal Gaussian distribution!
Particle Filtering: FastSLAM

• Particles represents samples from the posterior distribution $p(x_t, L | Z, U)$.

• $p(x_t, L | Z, U)$ can be any distribution (need not be Gaussian).
FastSLAM

Each particle represents:

\[
p^m_t = \{x^m_t, < \mu^m_{1,t}, \Sigma^m_{1,t} >, < \mu^m_{2,t}, \Sigma^m_{2,t} > ... < \mu^m_{N,t}, \Sigma^m_{N,t} > \}
\]

Robot state

Landmark state
(mean and covariance)

\[
x^m_t \sim p(x_t | x_{t-1}, u_t)
\]

Sample the robot state from the process model

\[
p(L^m_n,t | x^m_t, z_t)
\]

N Kalman filter
Landmark updates

\[
w^m_t \propto p(z_t | L^m_t, x^m_t)
\]

Weight update

Resampling based on current state
FastSLAM

• Many particles needed for accurate results.
• Computationally expensive for high state dimensions.
• Constraints: Relative pose estimates from 3D structure.
• Don’t update 3D structure (fixed wrt. to some pose).
• Optimizes poses as $\text{argmin}_X \sum_{ij} \| z_{ij} - h(v_i, v_j) \|^2_{\Sigma_{ij}}$
• Can be used to minimize loop-closure errors.
Summary

• **Bundle Adjustment**
  – Refine 3D points and poses in Structure-From-Motion.
  – Efficient computation by exploiting structure & sparsity.
  – Core step in every Structure-From-Motion (SFM) pipeline.

• **Simultaneous Localization and Mapping**
  – Very similar to Incremental SFM.
  – Typically includes some motion model.
  – Two general approaches to SLAM:
    • (Local) Bundle Adjustment (not discussed in lecture)
    • Filter-based techniques (EKF SLAM, FastSLAM)
  – Pose-Graph SLAM (loop-closure handling)
<table>
<thead>
<tr>
<th>Date</th>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feb 18</td>
<td>Introduction</td>
</tr>
<tr>
<td>Feb 25</td>
<td>Geometry, Camera Model, Calibration</td>
</tr>
<tr>
<td>Mar 4</td>
<td>Features, Tracking / Matching</td>
</tr>
<tr>
<td>Mar 11</td>
<td><strong>Project Proposals by Students</strong></td>
</tr>
<tr>
<td>Mar 18</td>
<td>Structure from Motion (SfM) + papers</td>
</tr>
<tr>
<td>Mar 25</td>
<td>Dense Correspondence (stereo / optical flow) + papers</td>
</tr>
<tr>
<td><strong>Apr 1</strong></td>
<td><strong>Bundle Adjustment &amp; SLAM + papers</strong></td>
</tr>
<tr>
<td>Apr 8</td>
<td><strong>Student Midterm Presentations</strong></td>
</tr>
<tr>
<td>Apr 15</td>
<td>Multi-View Stereo &amp; Volumetric Modeling + papers</td>
</tr>
<tr>
<td>Apr 22</td>
<td>Easter break</td>
</tr>
<tr>
<td>Apr 29</td>
<td>3D Modeling with Depth Sensors + papers</td>
</tr>
<tr>
<td>May 6</td>
<td>3D Scene Understanding + papers</td>
</tr>
<tr>
<td>May 13</td>
<td>4D Video &amp; Dynamic Scenes + papers</td>
</tr>
<tr>
<td>May 20</td>
<td>papers</td>
</tr>
<tr>
<td>May 27</td>
<td><strong>Student Project Demo Day = Final Presentations</strong></td>
</tr>
</tbody>
</table>
Next week: Midterm Presentations

Reminder: Prepare short presentation (3-5min) for Monday!