Lecture 4: Structure from Motion (SfM)
Topics Covered

• Introduction
  - Structure from Motion (SfM)
  - Simultaneous Localization and Mapping (SLAM)

• Bundle Adjustment
  - Energy Function
  - Non-linear Least Squares
  - Exploiting the Sparse Structure

• Triangulation
Structure from Motion


- 3D reconstruction using a set of unordered images
- Requires estimation of 6DoF poses
Simultaneous Localization and Mapping (SLAM)

- Estimate 6DoF poses and map from sequential image data
- Update once new frames arrive

Engel et al., “LSD-SLAM: Large-Scale Direct Monocular SLAM”, ECCV 2014
Problem Definition SfM / Visual SLAM

Estimate camera poses and map from a set of images

• Input

Set of images $I_{0:t} = \{I_0, I_1, \ldots, I_t\}$

Additional input possible
• Stereo
• Depth
• Inertial measurements
• Control input

• Output

Camera pose estimates $T_i \in SE(3)$,
also written as $\xi_i = \left(\log T_i\right)^\vee \quad i \in \{0,1,\ldots,t\}$

Environment map $M$

fr3/long_office_household sequence, TUM RGB-D benchmark

Mur-Artal et al., 2015
Typical SfM Pipeline

1) **Map initialization**
   - Using 2D-to-2D correspondences
   - Recover pose (stereo pair if available)
   - Triangulate landmarks using pose

2) **Localization** with known map
   - Using 2D-to-3D correspondences

3) **Mapping** with known poses
   - Using 2D-to-2D correspondences
     → **Triangulation**

4) **Joint refinement** of map and poses
   - Using 2D-to-2D correspondences
     → **Bundle adjustment**

Iterate to add new frames
Visual SLAM

SLAM ⊂ SfM, with special focus:
• Sequential image data
• Data arrives sequentially
• Preferably realtime
• More focus on trajectory

Technical solutions:
• Windowed optimization
• Selection of keyframes
• Removal of keyframes (e.g. marginalization)

Accumulation of drift
• Detect loop closures
• Global mapping in separate thread
  (e.g. pose graph optimization)

Odometry
• No global mapping
• Incremental tracking only
• Local map possible

Clemente et al., RSS 2007
Landmarks and Features

- The map consists of 3D locations of landmarks
  \[ M = \{ \mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_S \} \]

- For image \( \tau \), the set of 2D image coordinates of detected features is denoted
  \[ Y_\tau = \{ y_{\tau,1}, y_{\tau,2}, \ldots, y_{\tau,N} \} \]

- Known data association:
  Feature \( i \) in image \( \tau \) corresponds to landmark \( j = c_{\tau,i} \) \( (1 \leq i \leq N, 1 \leq j \leq S) \)
Bundle Adjustment Energy

\[
E(\xi_{0:t}, M) = \frac{1}{2} (\xi_0 \Theta \xi^0)^T \Sigma_{0,\xi}^{-1} (\xi_0 \Theta \xi^0)
\]

\[
+ \frac{1}{2} \sum_{\tau=0}^{t} \sum_{i=1}^{N_r} \left( y_{\tau,i} - h\left(\xi_\tau, m_{c,\tau,i}\right) \right)^T \Sigma_{y,\tau,i}^{-1} \left( y_{\tau,i} - h\left(\xi_\tau, m_{c,\tau,i}\right) \right)
\]

• Pose prior: Fix absolute pose ambiguity
  - In this case equivalent to keeping \(\xi_0 = \xi^0\)
  - Keep absolute pose information e.g. when first frame is marginalized
• Additional prior to fix scale ambiguity might be necessary

Absolute pose fixed by \(\xi^0\)

3D coordinates of map points

Reprojection errors

Camera poses
Energy Function as Non-linear Least Squares

- Residuals as function of state vector $x$

$$r^0(x) := \xi_0 \otimes \xi^0$$

$$r^y_{t,i}(x) := y_{t,i} - h(\xi_t, m_{c_{t,i}})$$

- Stack the residuals in a vector-valued function and collect the residual covariances on the diagonal blocks of a square matrix

$$r(x) := \begin{pmatrix} r^0(x) \\ r^y_{0,1}(x) \\ \vdots \\ r^y_{t,N_t}(x) \end{pmatrix}$$

$$W := \begin{pmatrix} \Sigma_{0,\xi}^{-1} & 0 & \cdots & 0 \\ 0 & \Sigma_{y_{0,1}}^{-1} & \ddots & \vdots \\ \vdots & \vdots & \ddots & 0 \\ 0 & \cdots & 0 & \Sigma_{y_{t,N_t}}^{-1} \end{pmatrix}$$

- Rewrite energy function as

$$E(x) = \frac{1}{2} r(x)^T W r(x)$$
Structure of the Bundle Adjustment Problem

\[
H_k = H_0^k + \sum_{\tau=0}^{t} \sum_{i=1}^{N_\tau} H_{\tau,i}^k = (J_0^k)^T \Sigma_{0,\xi}^{-1} (J_0^k) + \sum_{\tau=0}^{t} \sum_{i=1}^{N_\tau} (J_{\tau,i}^k)^T \Sigma_{y,\tau,i}^{-1} (J_{\tau,i}^k)
\]

Diagonal, typically \( S \gg t \)

Sparse!
Example Hessian of a BA Problem

\[ H_k = \]

Landmark dimensions (982 landmarks)

Pose dimensions (10 poses)

Lourakis et al., 2009

Large, but sparse!

How to invert efficiently?
Exploiting the Sparse Structure

• Idea:
  Apply the Schur complement to solve the system in a partitioned way

\[
H_k \Delta x = - b_k
\]

\[
\begin{pmatrix}
H_{\xi\xi} & H_{\xi m} \\
H_{m\xi} & H_{mm}
\end{pmatrix}
\begin{pmatrix}
\Delta x_{\xi} \\
\Delta x_m
\end{pmatrix} = -
\begin{pmatrix}
b_{\xi} \\
b_m
\end{pmatrix}
\]

\[
\Delta x_{\xi} = - \left( H_{\xi\xi} - H_{\xi m} H_{mm}^{-1} H_{m\xi} \right)^{-1} \left( b_{\xi} - H_{\xi m} H_{mm}^{-1} b_m \right)
\]

\[
\Delta x_m = - H_{mm}^{-1} \left( b_m + H_{m\xi} \Delta x_{\xi} \right)
\]

• Is this any better?
Exploiting the Sparse Structure

\[ \Delta x_\xi = - \left( H_{\xi \xi} - H_{\xi m} H_{m m}^{-1} H_{m \xi} \right)^{-1} \left( b_\xi - H_{\xi m} H_{m m}^{-1} b_m \right) \]
Effect of Loop Closures on the Hessian

Full Hessian

Reduced pose Hessian

Band matrix

Before loop closure
Effect of Loop Closures on the Hessian

No band matrix: costlier to solve

After loop closure
Further Considerations

Many methods to improve convergence / robustness / run-time efficiency, e.g.

- Use matrix decompositions (e.g. Cholesky) to perform inversions
- Levenberg-Marquardt optimization improves basin of convergence
- Heavier-tail distributions / robust norms on the residuals can be implemented using iteratively reweighted least squares
- Preconditioning
- Hierarchical optimization
- Variable reordering
- Delayed relinearization
Triangulation

- Find landmark position given the camera poses
- Ideally, the rays should intersect
- In practice, many sources of error: pose estimates, feature detections and camera model / intrinsic parameters
Triangulation

- Goal: Reconstruct 3D point $\tilde{x} = (x, y, z, w)^T \in \mathbb{R}^3$ from 2D image observations $\{\mathbf{y}_1, \ldots, \mathbf{y}_N\}$ for known camera poses $\{\mathbf{T}_1, \ldots, \mathbf{T}_N\}$

- Linear solution: Find 3D point such that reprojections equal its projection

  - For each image $i$, let $\mathbf{T}_i = \begin{pmatrix} p_1 & p_2 & p_3 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$ and $\mathbf{y}_i = \begin{pmatrix} u \\ v \end{pmatrix}$

  - Projecting $\tilde{x}$ yields $\mathbf{y}_i' = \pi(\mathbf{T}_i \tilde{x}) = \begin{pmatrix} p_1 \tilde{x}/p_3 \tilde{x} \\ p_2 \tilde{x}/p_3 \tilde{x} \end{pmatrix}$

  - Requiring $\mathbf{y}_i' = \mathbf{y}_i$ gives two linear equations per image:
    \begin{align*}
    p_1 \tilde{x} &= u p_3 \tilde{x} \\
    p_2 \tilde{x} &= v p_3 \tilde{x}
    \end{align*}

  - Leads to system of linear equations $\mathbf{A} \tilde{x} = \mathbf{0}$, two approaches to solve:
    - Set $w = 1$ and solve non-homogeneous least squares problem
    - Find nullspace of $\mathbf{A}$ using SVD, then scale such that $w = 1$

- Non-linear least squares on reprojection errors (more accurate): $\min_x \left\{ \sum_{i=1}^N \|\mathbf{y}_i - \mathbf{y}_i'\|_2^2 \right\}$

- Different solutions for different methods in the presence of noise
Exercises

Exercise sheet 4
• Implement components of SfM pipeline
• BA: Ceres can do the Schur complement
• Triangulation: use OpenGV’s triangulate function

Exercise sheet 5
• Implement components of odometry
• Similar to sheet 4, but:
  − More efficient 2D-3D matching using approximate pose of previous frame
  − New keyframe if number of matches too small
  − New landmarks by triangulation from stereo pair
  − Keep runtime bounded by removing old keyframes

```cpp
ceres::Solver::Options ceres_options;
ceres_options.max_num_iterations = 20;
ceres_options.linear_solver_type = ceres::SPARSE_SCHUR;
ceres_options.num_threads = 8;
ceres::Solver::Summary summary;
Solve(ceres_options, &problem, &summary);
std::cout << summary.FullReport() << std::endl;
```
Summary

SfM
- Estimate map and camera poses from set of images
- SLAM: Sequential data, real-time
- Odometry: No global mapping

Bundle Adjustment
- Non-linear least squares problem
- Sparse structure of Hessian can be exploited for efficient inversion

Triangulation
- Linear and non-linear algorithms
- Differences in the presence of noise