Practical Course: Vision Based Navigation

Lecture 4: Structure from Motion (SfM)

Dr. Vladyslav Usenko, Nikolaus Demmel, David Schubert
Prof. Dr. Daniel Cremers

Version: 16.05.2020
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 \( \xi_i = (\log T_i)^\vee \quad i \in \{0,1,\ldots,t\} \)
  
  Environment map \( M \)

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**
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

\[ M = \{m_1, m_2, \ldots, m_S\} \]

• 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

\[
E(\xi_{0:t}, M) = \frac{1}{2} (\xi_0 \ominus \xi^0)^T \Sigma_{0, \xi}^{-1} (\xi_0 \ominus \xi^0) \\
+ \frac{1}{2} \sum_{\tau=0}^{t} \sum_{i=1}^{N_{\tau}} \left( y_{\tau,i} - h(\xi_\tau, m_{c,i}) \right)^T \Sigma_{y,\tau,i}^{-1} \left( y_{\tau,i} - h(\xi_\tau, m_{c,i}) \right)
\]

- 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
Energy Function as Non-linear Least Squares

- Residuals as function of state vector \( \mathbf{x} \)

\[
\begin{align*}
\mathbf{r}^0(\mathbf{x}) & := \xi_0 \Theta \xi^0 \\
\mathbf{r}^y_t(\mathbf{x}) & := \mathbf{y}_t, i - h(\xi_t, \mathbf{m}_{c_t, i})
\end{align*}
\]

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

\[
\begin{align*}
\mathbf{r}(\mathbf{x}) & := \begin{pmatrix} \mathbf{r}^0(\mathbf{x}) \\ \mathbf{r}^y_{0,1}(\mathbf{x}) \\ \vdots \\ \mathbf{r}^y_{t,N_t}(\mathbf{x}) \end{pmatrix} \\
\mathbf{W} & := \begin{pmatrix} 
\Sigma_{0,\xi}^{-1} & 0 & \cdots & 0 \\
0 & \Sigma_{y_{0,1}}^{-1} & \ddots & \vdots \\
\vdots & \ddots & \ddots & 0 \\
0 & \cdots & 0 & \Sigma_{y_{t,N_t}}^{-1} 
\end{pmatrix}
\end{align*}
\]

- Rewrite energy function as

\[
E(\mathbf{x}) = \frac{1}{2} \mathbf{r}(\mathbf{x})^T \mathbf{W} \mathbf{r}(\mathbf{x})
\]
Recap: Gauss-Newton Method

- Idea: Approximate Newton’s method to minimize $E(x)$
  - Approximate $E(x)$ through linearization of residuals

$$
\tilde{E}(x) = \frac{1}{2} \tilde{r}(x)^T W \tilde{r}(x)
$$

$$
= \frac{1}{2} \left( r \left( x_k \right) + J_k \left( x - x_k \right) \right)^T W \left( r \left( x_k \right) + J_k \left( x - x_k \right) \right)
$$

$$
= \frac{1}{2} r \left( x_k \right)^T W r \left( x_k \right) + r \left( x_k \right)^T W J_k \left( x - x_k \right) + \frac{1}{2} \left( x - x_k \right)^T J_k^T W J_k \left( x - x_k \right)
$$

- Finding root of gradient as in Newton’s method leads to update rule

$$
\nabla_x \tilde{E}(x) = b_k^T + (x - x_k)^T H_k
$$

$$
\nabla_x \tilde{E}(x) = 0 \quad \text{iff} \quad x = x_k - H_k^{-1} b_k
$$

- Pros:
  - Faster convergence than gradient descent (approx. quadratic convergence rate)

- Cons:
  - Divergence if too far from local optimum ($H$ not positive definite)
  - Solution quality depends on initial guess
Structure of the Bundle Adjustment Problem

- \( \mathbf{b}_k \) and \( \mathbf{H}_k \) sum terms from individual residuals:

\[
\mathbf{b}_k = \mathbf{b}_k^0 + \sum_{\tau=0}^{t} \sum_{i=1}^{N_{\tau}} \mathbf{b}^{\tau,i}_k = (\mathbf{J}_k^0)^\top \mathbf{\Sigma}_{0,\xi}^{-1} \mathbf{r}_\xi^0 (\mathbf{x}_k) + \sum_{\tau=0}^{t} \sum_{i=1}^{N_{\tau}} \left( (\mathbf{J}^{\tau,i}_k)^\top \mathbf{\Sigma}_{y,\tau,i}^{-1} \mathbf{r}^y_{\tau,i} (\mathbf{x}_k) \right)
\]

\[
\mathbf{H}_k = \mathbf{H}_k^0 + \sum_{\tau=0}^{t} \sum_{i=1}^{N_{\tau}} \mathbf{H}^{\tau,i}_k = (\mathbf{J}_k^0)^\top \mathbf{\Sigma}_{0,\xi}^{-1} (\mathbf{J}_k^0) + \sum_{\tau=0}^{t} \sum_{i=1}^{N_{\tau}} \left( (\mathbf{J}^{\tau,i}_k)^\top \mathbf{\Sigma}_{y,\tau,i}^{-1} (\mathbf{J}^{\tau,i}_k) \right)
\]

\( \mathbf{J}_k^0 \) \quad Jacobian of pose prior

\( \mathbf{J}^{\tau,i}_k \) \quad Jacobian of residuals for feature \( i \) in image \( \tau \)

- What is the structure of these terms?
Structure of the Bundle Adjustment Problem

\[
\begin{align*}
\mathbf{b}_k &= \mathbf{b}_k^0 + \sum_{\tau=0}^{t} \sum_{i=1}^{N_\tau} \mathbf{b}_{\tau,i}^k \\
&= (J_k^0)^\top \Sigma_{0,\xi}^{-1} \mathbf{r}^0(\mathbf{x}_k) + \sum_{\tau=0}^{t} \sum_{i=1}^{N_\tau} (J_{\tau,i}^k)^\top \Sigma_{y,\tau,i}^{-1} \mathbf{r}^{y,i}(\mathbf{x}_k)
\end{align*}
\]
Structure of the Bundle Adjustment Problem

\[ H_k = H^0_k + \sum_{\tau=0}^{t} \sum_{i=1}^{N_\tau} H^{\tau,i}_k = (J^0_k)^T \Sigma^{-1}_{0,\xi} (J^0_k) + \sum_{\tau=0}^{t} \sum_{i=1}^{N_\tau} (J^{\tau,i}_k)^T \Sigma^{-1}_{y_{\tau,i}} (J^{\tau,i}_k) \]
Example Hessian of a BA Problem

\[ H_k = \]

Lourakis et al., 2009

Landmark dimensions (982 landmarks)

Pose dimensions (10 poses)

Large, but sparse!

How to invert efficiently?
Exploiting the Sparse Structure

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

\[
\begin{align*}
H_k \Delta x &= -b_k \\
\begin{pmatrix} H_{\xi \xi} & H_{\xi m} \\ H_{m \xi} & H_{mm} \end{pmatrix} \begin{pmatrix} \Delta x_\xi \\ \Delta x_m \end{pmatrix} &= - \begin{pmatrix} b_\xi \\ b_m \end{pmatrix} \\
\Delta x_\xi &= - \left( H_{\xi \xi} - H_{\xi m} H_{mm}^{-1} H_{m \xi} \right)^{-1} \left( b_\xi - H_{\xi m} H_{mm}^{-1} b_m \right) \\
\Delta x_m &= - H_{mm}^{-1} \left( b_m + H_{m \xi} \Delta x_\xi \right)
\end{align*}
\]

- Is this any better?
Exploiting the Sparse Structure

- What is the structure of the two sub-problems?

- Poses:

\[
\begin{align*}
\Delta x_\xi &= - \left( H_{\xi\xi} - H_{\xi m} H_{m m}^{-1} H_{m \xi} \right)^{-1} \left( b_\xi - H_{\xi m} H_{m m}^{-1} b_m \right) \\
H_{\xi\xi} - H_{\xi m} H_{m m}^{-1} H_{m \xi} &= H_{\xi\xi} - \sum_{j=1}^{S} H_{\xi m_j} H_{m_j m_j}^{-1} H_{m_j \xi} \\
b_\xi - H_{\xi m} H_{m m}^{-1} b_m &= b_\xi - \sum_{j=1}^{S} H_{\xi m_j} H_{m_j m_j}^{-1} b_{m_j}
\end{align*}
\]

Reduced pose Hessian
Exploiting the Sparse Structure

• What is the structure of the two sub-problems?

• Poses:

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

\[
H_{\xi \xi} - \sum_{j=1}^{S} H_{\xi m_j} H_{m_j m_j}^{-1} H_{m_j \xi}
\]

Poses that observe landmark \( j \)

\[
H_{\xi \xi} \quad - \sum_{j=1}^{S} H_{\xi m_j} H_{m_j m_j}^{-1} H_{m_j \xi}
\]

=  

\[
b_\xi \quad - \sum_{j=1}^{S} H_{\xi m_j} H_{m_j m_j}^{-1} b_m
\]
Exploiting the Sparse Structure

• What is the structure of the two sub-problems?

• Landmarks:

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

\[
\Delta x_{m_j} = - H_{m_j, m_j}^{-1} \left( b_{m_j} + H_{m_j} \xi \Delta x_{\xi} \right)
\]

• Landmark-wise solution
• Comparably small matrix operations
• Only involves poses that observe the landmark
Exploiting the sparse structure

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

As a result, only a much smaller matrix has to be inverted
Exploiting the Sparse Structure

- Reduced pose Hessian can still have a sparse structure
- For many camera poses with many shared observations, the inversion of the reduced pose Hessian is still computationally expensive!
- Exploit further structure, e.g. using variable reordering or hierarchical decomposition
Effect of Loop Closures on the Hessian

Full Hessian

Reduced pose Hessian

Band matrix

Before loop closure
Effect of Loop Closures on the Hessian

Full Hessian

Reduced pose 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 $\tilde{x} = (x, y, z, w)^T \in \mathbb{P}^3$ from 2D image observations $\{y_1, \ldots, y_N\}$ for known camera poses $\{T_1, \ldots, T_N\}$

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

  - For each image $i$, let $T_i = \begin{pmatrix} p_1 & \vdots & p_3 & 0 & 0 & 0 & 1 \end{pmatrix}$ and $y_i = \begin{pmatrix} u \\ v \end{pmatrix}$

  - Projecting $\tilde{x}$ yields $y'_i = \pi(T_i\tilde{x}) = \begin{pmatrix} p_1\tilde{x}/p_3\tilde{x} \\ p_2\tilde{x}/p_3\tilde{x} \end{pmatrix}$

  - Requiring $y'_i = y_i$ gives two linear equations per image: $p_1\tilde{x} = up_3\tilde{x}$, $p_2\tilde{x} = vp_3\tilde{x}$

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

• Non-linear least squares on reprojection errors (more accurate):

  $$\min_x \left\{ \sum_{i=1}^N \|y_i - y'_i\|_2^2 \right\}$$

• 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;
```
<table>
<thead>
<tr>
<th></th>
<th>Original</th>
<th>Reduced</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter blocks</td>
<td>4896</td>
<td>4892</td>
</tr>
<tr>
<td>Parameters</td>
<td>15354</td>
<td>15324</td>
</tr>
<tr>
<td>Effective parameters</td>
<td>15190</td>
<td>15162</td>
</tr>
<tr>
<td>Residual blocks</td>
<td>24014</td>
<td>24014</td>
</tr>
<tr>
<td>Residuals</td>
<td>48028</td>
<td>48028</td>
</tr>
</tbody>
</table>

Minimizer                  | TRUST_REGION |
Sparse linear algebra library | SUITE_SPARSE |
Trust region strategy       | LEVENBERG_MARQUARDT |

<table>
<thead>
<tr>
<th></th>
<th>Given</th>
<th>Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear solver</td>
<td>SPARSE_SCHUR</td>
<td>SPARSE_SCHUR</td>
</tr>
<tr>
<td>Threads</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>Linear solver ordering</td>
<td>AUTOMATIC</td>
<td>4730,162</td>
</tr>
<tr>
<td>Schur structure</td>
<td>2,3,6</td>
<td>2,3,6</td>
</tr>
</tbody>
</table>

Cost:
<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial</td>
<td>3.979886e+03</td>
</tr>
<tr>
<td>Final</td>
<td>3.766801e+03</td>
</tr>
<tr>
<td>Change</td>
<td>2.130843e+02</td>
</tr>
</tbody>
</table>

Minimizer iterations | 21
Successful steps    | 21
Unsuccessful steps   | 0

Time (in seconds):
<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Preprocessor</td>
<td>0.048047</td>
</tr>
<tr>
<td>Residual only evaluation</td>
<td>0.069569  (20)</td>
</tr>
<tr>
<td>Jacobian &amp; residual evaluation</td>
<td>0.388923  (21)</td>
</tr>
<tr>
<td>Linear solver</td>
<td>0.586967   (20)</td>
</tr>
<tr>
<td>Minimizer</td>
<td>1.134797</td>
</tr>
<tr>
<td>Postprocessor</td>
<td>0.001068</td>
</tr>
<tr>
<td>Total</td>
<td>1.183913</td>
</tr>
</tbody>
</table>

Termination: NO_CONVERGENCE (Maximum number of iterations reached. Number of iterations: 20.)
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