Some slides in this lecture are courtesy to Prof. Manmohan and Prof. Silvio Savarese
Structure from Motion

Several images observe a scene from different viewpoints
Feature detection

Detect features using, for example, SIFT [Lowe, IJCV 2004]
Feature matching

Match features between each pair of images
Feature matching
Feature matching

Refine matching using RANSAC to estimate fundamental matrix between each pair
Fundamental Matrix

\[ x_1 \leftrightarrow x_2 \]

\[ x_1^T F x_2 = 0 \]
8-point algorithm

Given \( n \) point correspondences, set up a system of equations:

\[
\begin{bmatrix}
  u_1 u_1' & v_1 u_1' & u_1' & u_1 v_1' & v_1 v_1' & v_1' & u_1 & v_1 & 1 \\
  u_2 u_2' & v_2 u_2' & u_2' & u_2 v_2' & v_2 v_2' & v_2' & u_2 & v_2 & 1 \\
  \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
  u_n u_n' & v_n u_n' & u_n' & u_n v_n' & v_n v_n' & v_n' & u_n & v_n & 1 \\
\end{bmatrix}
\begin{bmatrix}
f_{11} \\
f_{12} \\
f_{13} \\
f_{21} \\
f_{22} \\
f_{23} \\
f_{31} \\
f_{32} \\
f_{33}
\end{bmatrix}
= 0
\]

- In reality, instead of solving \( Af = 0 \), we seek \( f \) to minimize \( \|Af\| \).
8-point algorithm – Problem?

- \( \mathbf{F} \) should have rank 2
- To enforce that \( \mathbf{F} \) is of rank 2, \( \mathbf{F} \) is replaced by \( \mathbf{F}' \) subject to the rank constraint.

- This is achieved by SVD. Let \( \mathbf{F} = \mathbf{U} \Sigma \mathbf{V}^T \), where
  \[
  \Sigma = \begin{bmatrix}
  \sigma_1 & 0 & 0 \\
  0 & \sigma_2 & 0 \\
  0 & 0 & \sigma_3
  \end{bmatrix}, \quad \Sigma' = \begin{bmatrix}
  \sigma_1 & 0 & 0 \\
  0 & \sigma_2 & 0 \\
  0 & 0 & 0
  \end{bmatrix}
  \]

then \( \mathbf{F}' = \mathbf{U} \Sigma' \mathbf{V}^T \) is the solution.
RANSAC

1. Randomly choose $s$ samples (correspondences)
   – For fundamental matrix, what is the size of $s$?

2. Fit the model (fundamental matrix) to those samples
   – How would you fit the model to $s$ correspondences?

3. Count the number of inliers among all other correspondences
   – How can you determine which points are inliers?

4. Repeat $N$ times
   – How do you choose $N$?

5. Choose the model with the largest set of inliers
We have:

\[ q^\top [t] \times R p = 0 \]

Define:

\[ E = [t] \times R \]

Then, we have:

\[ q^\top E p = 0 \]
• Essential matrix constraint in pixel space: \((K_2^{-1}q')^\top E(K_1^{-1}p') = 0\).
• Rearranging: \(q'^\top (K_2^{-\top}EK_1^{-1})p' = 0\)
• Define: \(F = K_2^{-\top}EK_1^{-1}\)
• Then, we have: \(q'^\top Fp' = 0\)
Motion from correspondences

• Use 8-point algorithm to estimate F
• Get E from F:

\[ F = K_2^{-\top} E K_1^{-1} \]
\[ E = K_2^{-\top} F K_1 \]

• Decompose E into skew-symmetric and rotation matrices:

\[ E = [t] \times R \]
Triangulation

$X \in \mathbb{R}^1, t_1$

$X \in \mathbb{R}^2, t_2$

Image 1
$R_1, t_1$

Image 2
$R_2, t_2$
Structure from motion

\[ \prod_1 X_1 \sim p_{11} \]

\[
\begin{align*}
\text{minimize} & \quad g(R, T, X) \\
\text{non-linear least squares}
\end{align*}
\]
Bundle adjustment

• Minimize sum of squared reprojection errors:

\[ g(X, R, T) = \sum_{i=1}^{m} \sum_{j=1}^{n} w_{ij} \cdot \left\| \mathbf{P}(x_i, R_j, t_j) - [u_{i,j}, v_{i,j}] \right\|^2 \]

  – Optimized with non-linear least squares
  – Levenberg-Marquardt is a popular choice

• Practical challenges?
  – Initialization
  – Outliers

indicator variable: whether point $i$ visible in image $j$

predicted image location

observed image location
Large systems built on these steps
Benchmark datasets

Visual Odometry / SLAM Evaluation 2012

The odometry benchmark consists of 22 stereo sequences, saved in lossless png format: We provide 11 sequences (00-10) with ground truth trajectories for training and 11 sequences (11-21) without ground truth for evaluation. For this benchmark you may provide results using monocular or stereo visual odometry, laser-based SLAM or algorithms that combine visual and LIDAR information. The only restriction we impose is that your method is fully automatic (e.g., no manual loop-closure tagging is allowed) and that the same parameter set is used for all sequences. A development kit provides details about the data format.

- Download odometry data set (grayscale, 22 GB)
- Download odometry data set (color, 65 GB)
- Download odometry data set (velodyne laser data, 80 GB)
- Download odometry data set (calibration files, 1 MB)
- Download odometry ground truth poses (4 MB)
- Download odometry development kit (1 MB)
### Benchmark datasets

#### Additional information used by the methods

- **Stereo**: Method uses left and right (stereo) images
- **Laser Points**: Method uses point clouds from Velodyne laser scanner
- **Loop Closure Detection**: This method is a SLAM method that detects loop closures
- **Additional training data**: Use of additional data sources for training (see details)

<table>
<thead>
<tr>
<th>Method</th>
<th>Setting</th>
<th>Code</th>
<th><strong>Translation</strong></th>
<th><strong>Rotation</strong></th>
<th>Runtime</th>
<th>Environment</th>
<th>Compare</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 V-LOAM</td>
<td>![Image]</td>
<td>☐</td>
<td>0.68 %</td>
<td>0.0015 [deg/m]</td>
<td>0.1 s</td>
<td>2 cores @ 2.5 Ghz (C/C++)</td>
<td>☐</td>
</tr>
<tr>
<td>2 LOAM</td>
<td>![Image]</td>
<td>☐</td>
<td>0.70 %</td>
<td>0.0017 [deg/m]</td>
<td>0.1 s</td>
<td>2 cores @ 2.5 Ghz (C/C++)</td>
<td>☐</td>
</tr>
<tr>
<td>3 SOFTZ</td>
<td>![Image]</td>
<td>☐</td>
<td>0.72 %</td>
<td>0.0016 [deg/m]</td>
<td>0.1 s</td>
<td>2 cores @ 2.5 Ghz (C/C++)</td>
<td>☐</td>
</tr>
<tr>
<td>4 LG-SLAM</td>
<td>![Image]</td>
<td>☐</td>
<td>0.82 %</td>
<td>0.0020 [deg/m]</td>
<td>0.2 s</td>
<td>2 cores @ 2.5 Ghz (C/C++)</td>
<td>☐</td>
</tr>
<tr>
<td>5 GDVO</td>
<td>![Image]</td>
<td>☐</td>
<td>0.86 %</td>
<td>0.0031 [deg/m]</td>
<td>0.09 s</td>
<td>1 core @ &gt;3.5 Ghz (C/C++)</td>
<td>☐</td>
</tr>
<tr>
<td>6 HypERROCC</td>
<td>![Image]</td>
<td>☐</td>
<td>0.88 %</td>
<td>0.0027 [deg/m]</td>
<td>0.25 s</td>
<td>2 cores @ 2.0 Ghz (C/C++)</td>
<td>☐</td>
</tr>
<tr>
<td>7 SOFT</td>
<td>![Image]</td>
<td>☐</td>
<td>0.88 %</td>
<td>0.0022 [deg/m]</td>
<td>0.1 s</td>
<td>2 cores @ 2.5 Ghz (C/C++)</td>
<td>☐</td>
</tr>
<tr>
<td>8 RotRocc</td>
<td>![Image]</td>
<td>☐</td>
<td>0.88 %</td>
<td>0.0025 [deg/m]</td>
<td>0.3 s</td>
<td>2 cores @ 2.0 Ghz (C/C++)</td>
<td>☐</td>
</tr>
<tr>
<td>9 EDVO</td>
<td>![Image]</td>
<td>☐</td>
<td>0.89 %</td>
<td>0.0030 [deg/m]</td>
<td>0.1 s</td>
<td>1 core @ 2.5 Ghz (C/C++)</td>
<td>☐</td>
</tr>
<tr>
<td>10 svo2</td>
<td>![Image]</td>
<td>☐</td>
<td>0.94 %</td>
<td>0.0021 [deg/m]</td>
<td>0.2 s</td>
<td>1 core @ 2.5 Ghz (C/C++)</td>
<td>☐</td>
</tr>
</tbody>
</table>