StructSLAM: Visual SLAM with Building Structure Lines

Danping Zou
Assistant Professor
Key Laboratory of Navigation and Location-based Services
Shanghai Jiao Tong University
dpzou@sjtu.edu.cn
http://drone.sjtu.edu.cn/dpzou
What’s SLAM?

- **SLAM**: Simultaneous Localization And Mapping.

Constructing a map of an unknown environment while simultaneously keeping track of the robot’s location.

- Autonomous navigation
- motion planning
Active areas for SLAM research

- **Robotics**
  - UAVs, Service robots

- **Augmented Reality (AR)**
SLAM with different kinds of sensors

- **Active sensing**
- Heavy and bulky
- Active sensing
- Energy consuming

Since 2005, there has been intense research into VSLAM (Visual SLAM) using primarily visual (camera) sensors.

- **Passive sensing**
- Light & compact
- Energy saving
- Ubiquity

2D laser rangefinder
3D liDAR
RGB-D camera
List of Visual SLAM methods

- MonoSLAM
- FastSLAM
- GraphSLAM
- PTAM
- ORB-SLAM
- **CoSLAM***
- DTAM
- LSD-SLAM

Extended Kalman filter

Sparse feature points

Structure-from-motion

Dense / Semi dense

Knowing issues for Visual SLAM

- Scale ambiguity
- Lack of Illumination
- Accumulated error (Drift error)
- Lack of Texture

Vision + X

- GNSS (GPS, Beidou)
- Inertial Units (Gyro+Acc+Compass)
- Wireless (Wifi, Bluetooth)
- Map / Floor plan
What about if we consider one special case: indoor scenes

Natural scenes

Man-made scenes
StructSLAM

- A novel visual SLAM method aided with structure lines
  - Structure lines: The line feature aligned with dominant orientations


- Special session for indoor localization
Indoor scenes

Motivations:

- Lines are better landmarks in texture-less scenes (like many indoor scenes with only white walls) than points.
- Some lines (structure lines) encode the global orientation information.

Less accumulated error
StructSLAM:
Visual SLAM with Building Structure Lines

Hui Zhong Zhou, Danping Zou et al.

Shanghai Key Laboratory of Navigation and Location Based Services
Shanghai Jiao Tong University
April, 2014

http://drone.sjtu.edu.cn/dpzou/project/structslam.php
Extended Kalman filter:
- Simple to be implemented
- Easily fused with other sensors (IMU, odometers)
- More robust than structure-from-motion pipeline.
The man-made world are generally dominated by several directions.

- Vertical direction (always points to the sky)
- Horizontal directions (are usually perpendicular to each other, although not always)
Parameter planes

- Parameter plane is one of $xy, yz, xz$ planes of the world frame.
- A structure line is represented by a point on the parameter plane.
- The parameter plane is selected so as to make sure it is the most perpendicular to the dominant direction.
Each structure line is represented by a point on the parameter plane, denoted by a 4x1 vector.

It is in fact a 2D inverse depth representation*

\[
\mathbf{l} = \begin{pmatrix}
    c_a \\
    c_b \\
    \theta \\
    h
\end{pmatrix}
\]

- Projection of camera center
- Direction
- Inverse depth

StructSLAM – State representation

State vector and covariance matrix (MonoSLAM)

\[
\begin{bmatrix}
X_c \\
X_p \\
X_l
\end{bmatrix}, \quad \Sigma = \begin{bmatrix}
\Sigma_{cc} & \Sigma_{cp} & \Sigma_{cl} \\
\Sigma_{pc} & \Sigma_{pp} & \Sigma_{pl} \\
\Sigma_{lc} & \Sigma_{lp} & \Sigma_{ll}
\end{bmatrix}
\]

Camera pose + Points + Structure Lines

\[
x_c = \begin{bmatrix}
p^w_c \\
q^{wc} \\
v^w \\
\omega^c
\end{bmatrix}, \quad x_p = \begin{bmatrix}
m_1 \\
m_2 \\
\vdots
\end{bmatrix}, \quad x_p = \begin{bmatrix}
l_1 \\
l_2 \\
\vdots
\end{bmatrix}
\]
Initialization

**Step 1:**
- Use LSD line detector* to detect line segments on the image.

---

Initialization

Step 2:

• Apply J-linkage* to classify parallel line segments into groups and detect vanishing points

Step 3:
- Estimate the dominant direction from the vanishing points.

\[ \eta \propto R_{cw}^{wc} K^{-1} v \]

\[ v = KR_{cw} \eta \]

\( K \) : Camera intrinsic
\( R_{cw}^{wc} \) : Rotation from the world frame to the camera frame

\( R_{cw}^{wc} = (R_{cw})^T \)
Initialization

Step 4:

- Refine the dominant directions by non-linear least square optimization
- Initialize new lines (See the feature management section)
State prediction

Camera – constant velocity or odometer data (if available)

\[ f_c(x_c) = \begin{pmatrix} \bar{p}^w \\ \bar{q}^{wc} \\ \bar{v}^w \\ \bar{\omega}^c \end{pmatrix} = \begin{pmatrix} p^w + v^w \Delta t \\ q^{wc} \cdot q(\omega^c) \Delta t \end{pmatrix}. \]

Odometer velocity

Landmarks – static

\[ f_p(x_p) = x_p \quad f_l(x_l) = x_l \]

\[ F(x) = \begin{bmatrix} f_c(x_c) \\ x_p \\ x_l \end{bmatrix} + n \]
State prediction

Covariance propagation

\[
\bar{\Sigma} = F_x \Sigma F_x^T + F_n \Sigma_n F_n^T
\]

\[
F(x) = \begin{bmatrix}
  f_c(x_c) \\
  x_p \\
  x_l
\end{bmatrix} + n
\]

\[
F_x = \begin{bmatrix}
  \frac{\partial f_c}{\partial x_c} & 0 \\
  0 & I
\end{bmatrix} \\
F_n = \begin{bmatrix}
  \frac{\partial f_c}{\partial n} \\
  0
\end{bmatrix}
\]
Data association

Find the line segments corresponding to the structure line
Data association

- Find the line segments corresponding to the structure line

- One-to-multiple matching
- There could be false matchings (outliers).
Data association

Step 1: Get candidate matching by $\chi^2$-distance

\[ \chi^2 = \mathbf{r}_i^T (\mathbf{H}_i \mathbf{H}_i^T)^{-1} \mathbf{r}_i \]

- $\mathbf{r}_i$ : residual vector
- $\mathbf{H}_i$ : Jacobian of observation function

\[ \chi^2 < 5.99 \quad (Probability > 95\%) \]
Data association

无忧 Step 2: Comparing appearance by ZNCC (zero mean normalized cross-correlation)

Previous frame

Current frame

ZNCC > 0.8
Step 3: One-feature RANSAC to eliminate false matchings (outliers):

- Randomly sample a candidate matching
- Run a tentative EKF update using the sampled matching and check the number of inliers
- Keep the inlier set with the maximum number
- Repeat the above steps
- Use the inlier set to run EKF update.
State update

Observation model:

\[
\begin{align*}
\bar{I}_i &= \frac{s_j^0 \cdot \bar{I}_i}{\sqrt{(\bar{I}_i^1)^2 + (\bar{I}_i^2)^2}} \\
\bar{I}_j &= \frac{s_j^0 \cdot \bar{I}_j}{\sqrt{(\bar{I}_j^1)^2 + (\bar{I}_j^2)^2}} \\
\end{align*}
\]

Initialization

State prediction

Data Association

State update

Initialize new features

Remove old features

Structure line
State update

Project a structure line onto the image

\[ \mathbf{v} = \mathbf{K} \mathbf{R}^{cw} \mathbf{\eta} \]

\[ l^w h = \mathbf{P}^T \left( \begin{bmatrix} c_a, c_b \end{bmatrix}^T h + \begin{bmatrix} \cos(\theta), \sin(\theta) \end{bmatrix}^T \right) \]

\[ l^c = \mathbf{R}^{cw} l^w h - \mathbf{R}^{cw} \mathbf{p}^w h \]

\[ l^i = \mathbf{K} l^c \]

\[ \mathbf{l} = \mathbf{v} \times l^i \]
Observation model

- Observation function (measurement function):

\[
h(x) = \begin{bmatrix}
    \vdots \\
    m_{ij} \\
    \vdots 
\end{bmatrix}
\]

- Since the desired distance is zero, the residual is computed as:

\[
r(x) = -h(x)
\]
State update

Standard Extended Kalman Filter:

Predicted state and covariance: \( \bar{x}, \bar{\Sigma} \)

Innovation covariance: \( S = H\bar{\Sigma}H^T + N \)

Kalman gain: \( K = \bar{\Sigma}H^TS^{-1} \)

State update: \( x \leftarrow \bar{x} + Kr \)

Covariance update: \( \Sigma \leftarrow \bar{\Sigma} - KS\bar{K}^T \)

\( H = \frac{\partial m}{\partial x} \) : Jacobian of observation function

\( N \) : Observation noise - Uncertainty of detected line segments

\[
\begin{bmatrix}
\cdots \\
4 \\
4 \\
\cdots
\end{bmatrix}
\]
Multiple-pass EKF update
(Point + Structure lines)
Feature management

Initialize new structure lines

- Initialization
- State prediction
- Data Association
- State update
- Initialize new features
- Remove old features

η : Dominant direction

Point in world frame: \( \mathbf{m} = \mathbf{R}^{wc} \mathbf{K}^{-1} \dot{\mathbf{m}} + \mathbf{p}^w \)

Line though the point: \( \mathbf{L} = \mathbf{m} \eta^T - \eta \mathbf{m}^T \)

Intersection with parameter plane:

\( \tilde{\mathbf{l}}^w = \mathbf{L} \pi \)

Expressed in parameter plane (xy-plane):

\[
\mathbf{l}^p = \mathbf{P} \tilde{\mathbf{l}}^w
\]

\[
\mathbf{P} = \begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0
\end{bmatrix}
\]
Feature management

- Initialize new structure lines

 Initialization

 State prediction

 Data Association

 State update

 Initialize new features

 Remove old features
Feature management

- Initialize new structure lines

\[ \mathbf{l} = \begin{bmatrix} c_a, c_b, \theta, h \end{bmatrix}^T = \begin{bmatrix} \mathbf{o}^p(1), \mathbf{o}^p(2), \arctan \left( \frac{l^p(2) - \mathbf{o}^p(2)}{l^p(1) - \mathbf{o}^p(1)} \right), h_0 \end{bmatrix}^T \]
Feature management

- The number of features is limited in the state.
- For each dominant direction, we keep a maximum number of structure lines.
- Old features are removed according to the number of matching failure (NOF)
Simulated case

Points

Lines

Structure lines

Points and structure lines
Results

Simulated case

Lines V.S. structure lines
Results

Real-world case (Using one video camera)
• Indoor texture-less scenes
Results of real-word indoor scenes

Real world case (Using one video camera)

- Outdoor texture rich scenes
Benchmark datasets

- **Rawseeds datasets (all methods fused the odometer information)**
  - **Bicacco-02-25b**: A 774m trajectory in the indoor scene

![Graphs showing performance of different SLAM methods against ground truth.](image)
Benchmark results

(a) StructSLAM  (b) CI-Graph  (c) MonoSLAM  (d) LineSLAM
Benchmark datasets

Bicacco-02-27a:

(a) StructSLAM
(b) MonoSLAM
(c) LineSLAM
Benchmark datasets

Comparision

<table>
<thead>
<tr>
<th>Bicocca 25b [774 m]</th>
<th>Bicocca 27a [967m]</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Position[m]</strong></td>
<td><strong>Yaw[rad]</strong></td>
</tr>
<tr>
<td>Mean</td>
<td>Max</td>
</tr>
<tr>
<td>MonoSLAM</td>
<td>12.22</td>
</tr>
<tr>
<td>LineSLAM</td>
<td>27.63</td>
</tr>
<tr>
<td>CI-GraphSLAM</td>
<td>2.236</td>
</tr>
<tr>
<td>StructSLAM</td>
<td><strong>0.797</strong></td>
</tr>
</tbody>
</table>

**TABLE II. ERROR COMPARISON.**

StructSLAM : Without any loop-closing algorithms being applied!
Running time efficiency

A common PC with i7 4-core cpu (2.7GHz) (c++ implementation)

Average running time: 25.8 ms
Peak running time: 62.9 ms
Conclusion & Discussion

StructSLAM is more robust in texture-less indoor scenes than conventional SLAM methods.

With global orientation information encoded in structure lines, StructSLAM produces much less drift error.

It is well fit for robotic and augmented reality (AR) applications in indoor scenes.
Conclusion & Discussion

There are also some problems:

- Line are not distinguishable as points (ZNCC does not work well for large view angle changes)
- Dominant directions should be captured in the image at initialization stage.
- Dominant directions is fixed after initialization
The Third UAV competition in SJTU

AI technology (computer vision, learning, autonomous exploration) on micro drones

Coming competition (New URL): http://drone.sjtu.edu.cn
Past competition (Old URL): http://mediosoc.sjtu.edu.cn/wordpress
Thanks!