3D Visual SLAM in Unstructured Domains

Weizhen Zhou
Outline

- A little bit about myself
- The search and rescue scenario
- 3D SLAM
- What’s new
A little bit about myself

- BE & Ph.D. from University of Technology, Sydney
- Research assistant at the ARC Centre of Excellence for Autonomous Systems
- Research engineer at the Defence Science and Technology Organisation
An international competition for urban search and rescue robots, in which robots compete to find victims in a simulated earthquake environment

Main goal: Reliably assisting rescue operations through*:

- Submission of camera pictures and other sensor information
- Mapping of the area
- Identification and localisation of victims

RoboCup Rescue Robot League

Simultaneous localization and mapping (SLAM) problems arise when a robot does not have access to a map of the environment, nor does it know its own pose.

The goal for the robot is to acquire a map of its environment, and at the same time localize itself in relation to the map.

3D SLAM
RGB-D Sensing

- Consists of a time-of-flight range camera (SwissRanger 3000, low resolution, 176x144 pixels) and a conventional pin-hole camera (Point Grey Dragonfly2, high resolution, 1024x768 pixels)

- Provides a low-cost, lightweight and robust 3D sensor compared to laser scanners and stereo cameras
Feature Extraction and Registration

- Scale Invariant Feature Transformation (SIFT)\* were used as:
  - Scale- and rotation-invariant
  - Compact landmark descriptor allows quick data association
  - Descriptor is rich enough to be highly discriminatory

- Least-squares 3D point set registration\**

- Outlier removal through RANSAC\***, obtain initial camera and feature poses

---


Formulation

- State vector $X$ contains 6 DoF camera positions and 3D feature poses: $(x, y, z, \alpha, \beta, \gamma)^T$ and $(x_f, y_f, z_f)^T$

- Observation model:
  \[
  \begin{pmatrix}
  \hat{x}_f \\
  \hat{y}_f \\
  \hat{z}_f 
  \end{pmatrix} = \begin{pmatrix}
  \hat{x}_c \\
  \hat{y}_c \\
  \hat{z}_c 
  \end{pmatrix} + \text{RPY}(\hat{\alpha}_c, \hat{\beta}_c, \hat{\gamma}_c)^T \begin{pmatrix}
  x_l \\
  y_l \\
  z_l 
  \end{pmatrix}
  \]

- Information filter update:
  \[
  \hat{X} = I^{-1}i, P = I^{-1}
  \]
  \[
  I(k + 1) = I(k) + \nabla H_{k+1}^T Q_{k+1}^{-1} \nabla H_{k+1}
  \]
  \[
  i(k + 1) = i(k) + \nabla H_{k+1}^T Q_{k+1}^{-1} [z(k + 1) - H_{k+1}(\hat{X}(k)) + \nabla H_{k+1}\hat{X}(k)]
  \]

Where $Q_{k+1}$ is the covariance matrix of the observation noise and $z(k + 1)$ is the observation vector.

The state vector estimation $\hat{X}(k + 1)$ can be computed by solving

\[
I(k + 1)\hat{X}(k + 1) = i(k + 1)
\]

Recovering $\hat{X}(k + 1)$ is costly, log(det($I(k + 1)$)) is used as the measurable quantity of the information update. The larger the information gain the smaller the uncertainty.

Algorithm

- **Challenges:**
  - Large amount of visual data to be processed to counter motion blurry and undulating terrains
  - State recovering is computational expensive in EIF
  - Statistically most consistent input may not be the best

- **Solutions:**
  - “Look ahead and search backwards”, avoid kidnapping
  - Dynamic gain for multi-objective exploration
Results

Data collected with hand-held sensor package at prox. 5Hz

3D map obtained from filtering of 118 (out of 366) frames of the 6 × 3 meters search and rescue arena.

Distribution of selected frames over the entire data sequence

x(red circle), y(blue cross), z(green square) covariance of camera poses.
Results

Illustration of the incremental accumulation of convex hulls by each new frame, projected in 2D XZ plan (a, b) Outlines the spatial coverage when $\omega$ is set to dynamic value. (c) Outlines the spatial coverage when $\omega = 0.9$ (conservative case). (d) $\omega = 0.1$ (greedy case)

\[
Q = \omega E_I + (1 - \omega) E_S
\]

\[
\omega = \frac{n_{\text{old}}}{n_{\text{all}}}
\]

Computational performance comparison with different $\omega$ values

<table>
<thead>
<tr>
<th>Measurement</th>
<th>Full SLAM</th>
<th>Dynamic $\omega$</th>
<th>$\omega = 0.9$</th>
<th>$\omega = 0.1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time(s)</td>
<td>1230</td>
<td>488</td>
<td>647</td>
<td>431</td>
</tr>
<tr>
<td>No. features</td>
<td>1996</td>
<td>699</td>
<td>858</td>
<td>655</td>
</tr>
<tr>
<td>No. cameras</td>
<td>354</td>
<td>76</td>
<td>92</td>
<td>72</td>
</tr>
<tr>
<td>Volume ($m^3$)</td>
<td>39.5</td>
<td>42.4</td>
<td>34.1</td>
<td>41.0</td>
</tr>
</tbody>
</table>

Position comparison with full SLAM

<table>
<thead>
<tr>
<th>Measurement</th>
<th>Dynamic $\omega$</th>
<th>$\omega = 0.9$</th>
<th>$\omega = 0.1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mu_X \pm \sigma_X$</td>
<td>0.04 \pm 0.15</td>
<td>0.09 \pm 0.12</td>
<td>-0.2 \pm 0.20</td>
</tr>
<tr>
<td>$\mu_Y \pm \sigma_Y$</td>
<td>-0.03 \pm 0.14</td>
<td>-0.06 \pm 0.16</td>
<td>-0.56 \pm 0.37</td>
</tr>
<tr>
<td>$\mu_Z \pm \sigma_Z$</td>
<td>0.02 \pm 0.14</td>
<td>-0.07 \pm 0.13</td>
<td>-0.12 \pm 0.21</td>
</tr>
<tr>
<td>$\mu_\alpha \pm \sigma_\alpha$</td>
<td>-0.009 \pm 0.11</td>
<td>-0.06 \pm 0.07</td>
<td>0.01 \pm 0.17</td>
</tr>
<tr>
<td>$\mu_\beta \pm \sigma_\beta$</td>
<td>0.02 \pm 0.12</td>
<td>-0.05 \pm 0.09</td>
<td>0.04 \pm 0.11</td>
</tr>
<tr>
<td>$\mu_\gamma \pm \sigma_\gamma$</td>
<td>0.007 \pm 0.08</td>
<td>-0.01 \pm 0.07</td>
<td>0.007 \pm 0.12</td>
</tr>
</tbody>
</table>

What’s new

- RGB-D SLAM based solutions become very popular fuelled by increased computational capability such as GPU and low-cost sensor unit such as Microsoft Kinect
- RGB-D SLAM dataset and benchmark
  - http://vision.in.tum.de/data/datasets/rgbd-dataset
- Wide range of applications of accurate 3D reconstruction:
  - Object modelling
  - 3D printing
  - Robot navigation
  - Archaeology
  - ...

Thank You & Questions?