Recall SLAM factor graph

E.g., in “input_INTEL_g2o.g2o”

Odometric constraints:

<table>
<thead>
<tr>
<th>EDGE_SE2</th>
<th>i</th>
<th>1224</th>
<th>1225</th>
<th>0.625700</th>
<th>0.003903</th>
<th>0.013764</th>
<th>11.124947</th>
<th>1.838454</th>
<th>0.000000</th>
<th>255.403355</th>
<th>0.000000</th>
<th>2432.575223</th>
</tr>
</thead>
<tbody>
<tr>
<td>EDGE_SE2</td>
<td>i</td>
<td>1225</td>
<td>1226</td>
<td>0.106504</td>
<td>-0.000109</td>
<td>-0.016731</td>
<td>13.283390</td>
<td>-138.281257</td>
<td>0.000000</td>
<td>8813.714781</td>
<td>0.000000</td>
<td>2418.398573</td>
</tr>
<tr>
<td>EDGE_SE2</td>
<td>i</td>
<td>1226</td>
<td>1227</td>
<td>0.000000</td>
<td>0.000000</td>
<td>-0.000662</td>
<td>11.111282</td>
<td>-0.257444</td>
<td>0.000000</td>
<td>399.999830</td>
<td>0.000000</td>
<td>2496.693284</td>
</tr>
</tbody>
</table>

Loop closure constraints:

<table>
<thead>
<tr>
<th>EDGE_SE2</th>
<th>i</th>
<th>19</th>
<th>166</th>
<th>-2.459689</th>
<th>0.241111</th>
<th>0.252800</th>
<th>11.731352</th>
<th>1.696454</th>
<th>0.000000</th>
<th>15.751178</th>
<th>0.000000</th>
<th>1592.856013</th>
</tr>
</thead>
<tbody>
<tr>
<td>EDGE_SE2</td>
<td>i</td>
<td>19</td>
<td>172</td>
<td>1.067903</td>
<td>0.915786</td>
<td>0.149470</td>
<td>22.211594</td>
<td>-17.729393</td>
<td>0.000000</td>
<td>39.428022</td>
<td>0.000000</td>
<td>1892.102792</td>
</tr>
<tr>
<td>EDGE_SE2</td>
<td>i</td>
<td>25</td>
<td>172</td>
<td>-2.539634</td>
<td>0.852630</td>
<td>-0.029860</td>
<td>11.348228</td>
<td>0.783018</td>
<td>0.000000</td>
<td>13.696833</td>
<td>0.000000</td>
<td>2357.130503</td>
</tr>
</tbody>
</table>

Visual loop closure detection
(a.k.a. visual place recognition)

General LCD/VPR framework

Typical LCD processing pipeline:
1. Selection of top ranked map candidates
2. Verification of the candidates (loop closure verification)

Major challenges:
1. Scalability: LCD in a map of numerous poses (images)
2. Illumination Invariance: LCD in spite of significant illumination changes between visits
3. Viewpoint invariance: LCD in spite of significant viewpoint changes between visits

Do these images come from the same place?
Scalability: BoW-based solution


Image Matching

Current View = A Map Location

Yes

No

Brute-Force Feature Matching

\[ O(n^2dN) \]

- \( n \): number features in an image
- \( d \): dimension of a feature descriptor
- \( N \): number of map locations

\( E.g., \ 300^2 \times 128 \times 10^5 = 10^{11} \)

Image Matching?

53,000 Map Locations

Current View

Bag-of-Words (BoW)

\[ O(ndN/k) = O(ndN') \]

- \( n \): number features in an image
- \( d \): dimension of a feature descriptor
- \( N \): number of map locations
- \( k \): a constant scaler

\( E.g., \ 300 \times 128 \times 10^5 / 10^2 = 10^7 \)

Bag of Words?

- How does Google search the internet so quickly?
- Answer: indexing
- However, we are dealing with visual data and must express images as “text files” before indexing can be applied.
BoW Image Descriptor

1. Extract visual features from training images

2. Cluster the extracted features

Each image is described in terms of a histogram of visual word frequencies

BoW Inverted Index

Past image indexes containing the words seen in current image

\[ O(ndN') \]

Cost of ANN search for vector quantization

- \( n \): number features in an image
- \( d \): dimension of a feature descriptor
- \( N' \): \( N/k \)
- \( N \): number of map locations
- \( k \): a constant scaler
Bag-of-Words w/ Vocabulary Tree

Bag-of-Words w/ K-D Tree

Bag-of-Words w/ LSH

Bag-of-Words w/ GNNS

Location/Place Recognition

Whole Image Descriptor

Feature Match $O(n^2dN)$

BoW

Linear Search $O(ndN')$

Tree Search $O(nd\log(N'))$

$s\text{GNNS Search} < O(nd\log(N'))$

$N' = N/k$

$O(1*dN) = O(dN)$

$d$: dimension of the descriptor

$N$: number of map locations
**Location/Place Recognition**

- **Feature Match**
  - BoW: $O(n^2dN)$
  - Linear Search: $O(ndN/k)$
  - Tree Search: $O(nd\log(N/k))$
- **Image Match**
  - BoW: $O(dN)$
  - Linear Search: $O((n/k)^2dN)$
- **sGNNS Search**
  - $O(nd\log(N/k))$

**Whole Image Descriptor**

- **O(dN)**

Examples: Color Histogram, HOG, Gist, SIFT, SURF, Down Sampled (DS) Image, **Binarized DS Image**

**Whole Image Descriptor**

- **Histogram**
- **Local histogram**
- **Gradient based**

**Whole Image Descriptor**

- **Gabor-Gist**

$d = 16 \times 20 = 320$


**Scalable Location/Place Recognition**

- **Feature Match**
  - BoW: $O(n^2dN)$
  - Linear Search: $O(ndN/k)$
  - Tree Search: $O(nd\log(N/k))$
- **Image Match**
  - BoW: $O(dN)$
  - Linear Search: $O((n/k)^2dN)$
- **BDS MI Match**
  - $O(N)$

**Summary**

- **Solutions to scalability**
  - BoW based: sGNNS
  - Whole image based: BDS+MI
- **Open issue**: robustness to dynamic changes:
  - illumination
  - seasonable
  - moving objects
- **Ongoing work**
  - Semantics SLAM