Introduction: the SLAM problem

Data Association

- Given an environment map
- And a set of sensor observations
- Associate observations with map elements
Importance of Data Association

- Measurement $y$ is used to improve estimation of $x$:
  \[
  \hat{x}_k^B = \hat{x}^B_{k|k-1} + K_k(z_k - h_k(\hat{x}^B_{k|k-1}))
  \]
  \[
  P_k^B = (I - K_k H_k) P^B_{k|k-1}
  \]
  \[
  K_k = P^B_{k|k-1} H_k^T (H_k P^B_{k|k-1} H_k^T + R_k)^{-1}
  \]

- If the association of $E_i$ with feature $F_j$ is:
  - correct: $\hat{x}$
  - spurious: $\hat{x} - \hat{x}$

Error: $x - \hat{x}$
Covariance: $P$

Consistency Divergence!

Difficulties: clutter

- Influence of the type, density, precision and robustness of features considered:

Laser scanner:
- Small amount of features ($n$)
- Small amount of measurements ($m$)
- Low spuriousness

High clutter

Difficulties: imprecision

- Both the sensor and the vehicle introduce imprecision

High clutter

Vertical Edge Trinocular vision:
- variable depth precision
- good angular precision

Robot imprecision:
- introduces CORRELATED error
Approaches to Data Association

- **Search in configuration space**: find robot location with maximal data to map overlapping
  - Can be done with raw data
  - Or with features
    - Speed of convergence?

- **Search in correspondence space**: find a consistent correspondence hypothesis and compute robot location
  1. Extract features from data
    - If sparse data, move and build a local map
  2. Feature-based map (points, lines, trees, …)
  3. Search for data feature to map feature correspondences
    - Exponential number of solutions?

Example

- **Vehicle with SICK laser**
- **Detect trees**

```
; (ui, vi): observation i relative to robot
; (xj, yj): absolute location of map feature j
mv = 0;
for x = xmin to xmax step xstep,
  for y = ymin to ymax step ystep,
    for t = tmin to tmax step tstep,
      v = 0;
      for i = 1 to m,
        (xi, yi) = compose(x, y, t, ui, vi);
        for j = 1 to n,
          d = dist(xi, yi, xj, vj);
          if d < dmax
            v = v + 1;
          fi
        fi
      fi
    fi
  fi
fi
```

Search in configuration space

- Evaluate all vehicle locations looking for tree matchings (or deploy an army of particles looking for trees)

¿where is the vehicle?
Data-driven search in configuration space (let the trees vote)

- Each pairing constrains the possible location of the vehicle (to a circle in this case).

\[
\begin{align*}
&; (u_i, v_i): \text{observation } i \text{ relative to robot } \\
&; (x_j, y_j): \text{absolute location of map feature } j \\
&; R = \text{rotation}(t); \\
&; \text{for } t = t_{\text{min}} \text{ to } t_{\text{max}} \text{ step } t_{\text{step}}, \\
&; \text{for } i = 1 \text{ to } m, \\
&; \quad (x_i, y_i) = R \ast (u_i, v_i); \\
&; \text{for } j = 1 \text{ to } n, \\
&; \quad (x, y) = (x_j, y_j) - (x_i, v_i); \\
&; \quad \text{addvote}(x, y, t); \\
&; \text{rof} \\
&; \text{rof} \\
\end{align*}
\]

Results

- The most voted solution(s) stands out

Feature extraction: Laser

- Classical algorithms (split & merge):

Not robust to complex and/or spurious data
Alternative: RANSAC

- **p**: no. of points
- **n**: points to build model
- **w**: probability that a point is good

\[ z = \text{acceptable probability of failure} \]
\[ t = \left\lceil \frac{\log z}{\log (1 - w^n)} \right\rceil \]

- **w**: 0.5
- **n**: 2
- **z**: 0.05
- **t**: 11

\[ O(p^n). \text{possible models} \]

Robust statistics deal with spuriousness

Our solution: RANSAC

RANSAC for 3D planes

Feature extractions: Sonar

- Very sparse and noisy data

“It is unclear how the performance of our approach degrades with inaccuracy of the sensors. For example, it is unclear if sonar sensors are sufficiently accurate to yield good results” (Thrun, 2001)

Move and build a local map

Exploit redundancy
Use a good sensor model

Hough Transform

- Sonar returns vote for lines
- Look for local maxima

The Hough gives robust local data associations

A local map is built using standard techniques
Data Association (in correspondence space)

- **n** map features:
  \[ \mathcal{F} = \{ F_1 \ldots F_n \} \]
- **m** sensor measurements:
  \[ \mathcal{E} = \{ E_1 \ldots E_m \} \]
- **Goal**: obtain a hypothesis that associates each observation \( E_i \) with a feature \( F_j \)
  \[ \mathcal{H}_m = \{ j_1 \ldots j_i \ldots j_m \} \]
- Non matched observations:
  \[ j_i = 0 \]

(n + 1)^m possible hypotheses

Interpretation tree (Grimson et al. ’87):

Use constraints to prune the tree

- **Map:**
  - \( F_1 \)
  - \( F_2 \)
  - \( F_3 \)
  - \( F_4 \)
  - \( F_5 \)
  - \( F_6 \)

- **Observations:**
  - \( E_1 \)
  - \( E_2 \)
  - \( E_3 \)
  - \( E_4 \)
  - \( E_5 \)

- **Constraints:**
  - Feature location (needs an estimation of robot location)
  - Geometric relations: angles, distances,… (location independent)

Individual Compatibility

- Measurement equation for observation \( E_i \) and feature \( F_j \)
  \[ z_i = h_{ij}(x_j^F) + w_i \]
  \[ E[w_i w_i]^T = R_i \]
  \[ z_i \approx h_{ij}(\hat{x}_j^F) + H_{ij}(x_i^E - \hat{x}_j^F) \]
  \[ H_{ij} = \frac{\partial h_{ij}}{\partial x_j^F}(\hat{x}_j^F) \]

- \( E_i \) and \( F_j \) are compatible if:
  \[ D_{ij}^2 = (z_i - h_{ij}(\hat{x}_j^F))^T P_{ij}^{-1}(z_i - h_{ij}(\hat{x}_j^F)) < \chi^2_{d_i, \alpha} \]
  \[ P_{ij} = H_{ij}^T \Sigma_j H_{ij} + R_i \]

- Nearest Neighbour (NN) rule:
  - associate \( E_i \) with the feature \( F_j \) having smaller Mahalanobis distance \( D_{ij} \)

Nearest Neighbor

Greedy algorithm: \( O(mn) \)

\[ \text{NN:} \]
\[ \text{for } i = 1 \text{ to } m \text{ -- measurement } E_i \]
\[ D_{2\text{min}} = \text{Mahalanobis}^2(E_i, F_j) \]
\[ \text{nearest } = 1 \]
\[ \text{for } j = 2 \text{ to } n \text{ -- feature } F_j \]
\[ D_{ij} = \text{Mahalanobis}^2(E_i, F_j) \]
\[ \text{if } D_{ij} < D_{2\text{min}} \text{ then} \]
\[ \text{nearest } = j \]
\[ D_{2\text{min}} = D_{ij} \]
\[ \text{fi} \]
\[ \text{fi} \]
\[ \text{if } D_{2\text{min}} < \text{Chi}_2^2(d_i, \alpha) \text{ then} \]
\[ H(i) = \text{nearest} \]
\[ \text{else} \]
\[ H(i) = 0 \]
\[ \text{fi} \]
\[ \text{fi} \]
**Example: MonoRob**

Nearest Neighbor: $\mathcal{H} = \{1, 2, 0\}$
True Association: $\mathcal{H} = \{1, 0, 2\}$

---

**The fallacy of the Nearest Neighbour**

- Parings may be individually compatible, but not jointly compatible:

  $(y_1, x_1)$ is IC compatible
  $(y_2, x_2)$ is IC compatible
  $(y_3, x_2)$ is IC compatible

  \[
  \{(y_1, x_1), (y_2, x_2)\}
  \]

---

**Joint Compatitibily**

- Given a hypothesis $\mathcal{H} = \{j_1, j_2, \ldots, j_s\}$
- Joint measurement equation
  \[
  z_\mathcal{H} = h_\mathcal{H}(x_{\mathcal{H}}^R) + w_\mathcal{H}
  \]
  \[
  h_\mathcal{H} = \begin{bmatrix}
  h_{1,j_1} \\
  h_{2,j_2} \\
  \vdots \\
  h_{s,j_s}
  \end{bmatrix}
  \]
- The joint hypothesis is compatible if:
  \[
  D^2_\mathcal{H} = (z_\mathcal{H} - h_\mathcal{H}(\hat{x}_\mathcal{H}^R))^T C^{-1}_\mathcal{H} (z_\mathcal{H} - h_\mathcal{H}(\hat{x}_\mathcal{H}^R)) < \chi^2_{d,\alpha}
  \]
  \[
  C_\mathcal{H} = H_\mathcal{H} P_{\mathcal{H}}^R H_\mathcal{H}^T + R_\mathcal{H}
  \]

  \[d = \text{length}(z)\]

---

**Individual vs. Joint Compatibility**

- Joint Compatibility assures consistency:

  $(y_1, x_1)$ and $(y_3, x_2)$ are jointly compatible

- $(y_1, x_1)$ and $(y_2, x_2)$ are NOT jointly compatible

---

Joint Compatibility Branch and Bound

- Find the largest hypothesis with jointly consistent pairings

```python
procedure JCBB (H, i): -- find pairings for observation E_i
    if i > m -- leaf node?
        if pairings(H) > pairings(Best)
            Best = H
        fi
    else
        for j in {1...n}
            if individual.compatibility(i, j) and then
                joint.compatibility(H, i, j)
                JCBB([H j], i + 1) -- pairing (E_i, E_j) accepted
            fi
        rof
    if pairings(H) + m - i > pairings(Best) -- can do better?
        JCBB([H 0], i + 1) -- star node, E_i not paired
    fi
```

SLAM without odometry

- No estimation of the vehicle motion
- Vehicle motion estimation using Joint Compatibility
- Assuming small motions
- Segments in the environment

The loop closing problem

Nearest Neighbor

Individually compatible pairings

Wrong answer!

Joint Compatibility at work

Jointly compatible pairings
No robot estimation: use geometric constraints

- Location independent constraints
  - Unary: depend on a single matching (size, color, …)
    » Tree diameter
    » Wall length: the observed length should be less than the length in the map (if map is complete)
  - Binary: depend on two matchings (angles, distances)
    » Distances between points
    » Angles between lines
    » "Distances" between segments
  - N-ary: more complex, little pruning

3. Vehicle Relocation

Previously built map (A)  Local map (B)

• Is the robot in the map?
• If so, where?

Geometric Constraints Branch and Bound (Grimson, 1990)

procedure GCBB(H, i):
if i > m -- leaf node?
  if pairings(H) > pairings(Best) -- did better?
    estimate.location(H)
    if joint.compatibility(H)
      Best = H
    fi
  fi
else
  for j in {1..n}
    if unary(i, j) \& binary(i, j, H)
      GCBB([H j], i + 1) -- (E_i, F_j) accepted
    fi
  rof
if pairings(H) + m - i > pairings(Best)
  GCBB([H 0], i + 1) -- try star node
fi
Robot relocation example

100% success rate for n=30 local maps (but exponential on m)

RANSAC

```java
procedure RANSAC(H) {
    if i > m
        if pairings(H) > pairings(Best)
            Best = H
        fi
    elseif pairings(H) == 3
        estimate location(H)
        if joint.compatibility(H)
            JCB::H, i = hypothesis verification
        fi
    else -- branch and bound without star node
        for j in {1...n}
            if unary(i, j) && binary(i, j, H)
                RS([H, j], i + 1)
            fi
        fi
    fi
}
```

Exploiting locality

- Limit search in the global map to subsets of covisible features
- Locality makes relocation linear with the global map size
Information Matrix vs. Covisibility

- Sydney park dataset (Nebot et al.)

Experiments

1. No significant difference in effectiveness of the considered algorithms
2. All algorithms are made linear with the size of the map
3. Efficiency when the vehicle is in the map

4. When the vehicle is NOT in the map:
   random measurements, 100 for each m=3..30

Multirobot map building

Match, join and fuse to one full stochastic map

Multirobot map building

- Randomly select a **feature** in one map
- Try to associate its **covisible features** in the other map

![Unsuccessful tries](image)

Multivehicle SLAM

![Final map](image)

4. Relocation using laser and vision

- 2D laser scanner:
  (+) structure.
  (-) poor discrimination.

- Vision camera:
  (+) visual appearance.
  (-) just viewing directions.
  (-) viewpoint dependent.

- Joint usage of laser+vision.
  – structure + visual appearance.
Recognition of Corresponding Observations.

- Which observations are correspondent?
  (horizontal, different height)

- Picture vs structure:
  – Variant appearance vs invariance.

Recognition of Corresponding Observations.

- Detect laser segments
  (regard as vertical planes).

- Obtain orthoimages
  – Perspective deformation compensation:
    » foreshortening
    » scale

- Corresponding planes look similar

Recognition of Corresponding Observations.

- Detect interest points in the orthoimages

- Find corresponding points using normalized cross-correlation
  – Non-planar elements produce spurious matchings

- Fit the relative translation using robust statistics

- Reject the hypotheses with little support

- Compute the relative location

Relative location computed
Relocation: experiments

• Given a map of a corridor build from 8 laser+vision snapshots

• Relocate 192 new observations within the map.

Results for 192 x 8 relocations

<table>
<thead>
<tr>
<th>Ref. View</th>
<th>True positives</th>
<th>True negatives</th>
<th>False positives</th>
<th>False negatives</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a)</td>
<td>31</td>
<td>128</td>
<td>0</td>
<td>23</td>
</tr>
<tr>
<td>(b)</td>
<td>24</td>
<td>128</td>
<td>0</td>
<td>40</td>
</tr>
<tr>
<td>(c)</td>
<td>32</td>
<td>128</td>
<td>0</td>
<td>32</td>
</tr>
<tr>
<td>(d)</td>
<td>19</td>
<td>138</td>
<td>2</td>
<td>33</td>
</tr>
<tr>
<td>(e)</td>
<td>47</td>
<td>132</td>
<td>0</td>
<td>13</td>
</tr>
<tr>
<td>(f)</td>
<td>30</td>
<td>128</td>
<td>0</td>
<td>34</td>
</tr>
<tr>
<td>(g)</td>
<td>54</td>
<td>108</td>
<td>0</td>
<td>30</td>
</tr>
<tr>
<td>(h)</td>
<td>30</td>
<td>141</td>
<td>0</td>
<td>16</td>
</tr>
<tr>
<td>TOTAL</td>
<td>272</td>
<td>1041</td>
<td>2</td>
<td>221</td>
</tr>
</tbody>
</table>

- Low false positive rate
- Most false negatives due to poor image overlap
- Much better than laser alone

Example of false positive

• Due to strong visual symmetries
  - Minor differences are regarded as outliers

Mapping using Laser and Vision

• Map = observations + relative locations

5. Conclusions

• Data association in continuous SLAM:
  – Algorithms based on some form of consensus have provided the best results
    » RANSAC
    » Hough
    » Joint Compatibility (forget NN for loop closing)

• Vehicle relocation and map matching
  – Robust, linear time algorithm based on RANSAC
  – Visual relocation works where laser fails (corridors!)

Recommended Readings


