Multiple Robot Simultaneous Localization and Mapping

Sajad Saeedi[†], Liam Paull[†], Michael Trentini[⋆] and Howard Li[†]

[†]COBRA Group at the University of New Brunswick, Fredericton, Canada, http://www.ece.unb.ca/COBRA/, {sajad.saeedi.g, liam.paull, howard}@unb.ca

[⋆]Defence Research and Development Canada, Suffield, Alberta, Canada, Mike.Trentini@drdc-rddc.gc.ca

Problem statement

In this research, a decentralized platform for SLAM with multiple robots has been developed

- Given: Two occupancy grid maps, each developed by a robot.
- Find: The transformation which fuses two maps.
  - This is different than image registration, since there is no a priori knowledge about the shared areas in maps.

Significance

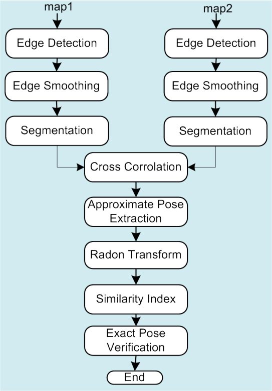
- Exploration and mapping can be done faster and more accurately by multiple robots.
- A distributed system is more robust.
- Applications in collaboration based operations: fire fighting in forest and urban areas, rescue operation in natural disasters, cleaning marine oil spills, underwater and space exploration, security, surveillance and maintenance investigations.

Contributions of research

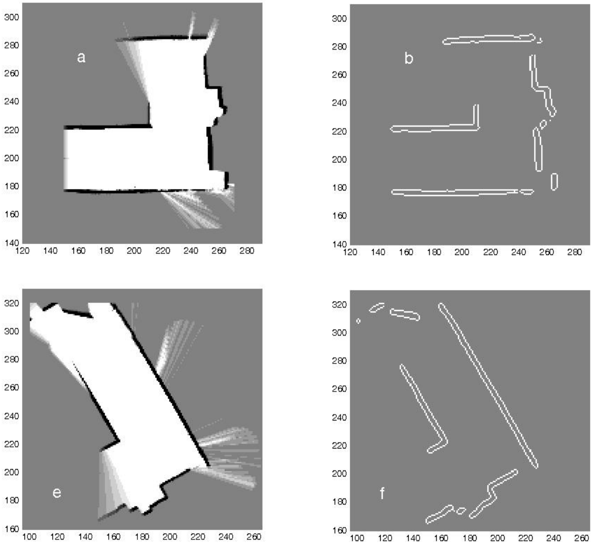
- Applying Canny edge detection and introducing a smoothing method to extract unique characteristics of an occupancy grid map,
- Using a segmentation method and applying a correlation technique to find the common parts within two maps,
- Estimating the relative transformation matrix of two maps by approximate and exact pose extractions,
- Applying the Radon transform to tune the orientation angle,
- The use of image entropy to tune the translation vector, as well as verify the final result.

Overview of proposed method

Main idea: find small segments present in both maps. Use them to find the transformation. Segments should not raise ambiguity.



- Problem: the occupancy grid mapping shows boundaries of obstacles as thicker than they are in reality due to the error in sensor measurements.
- Solution: Using Canny edge detection, it is possible to extract the exact obstacle boundaries to be used in the other blocks so that the map can be processed more accurately and efficiently.


a
300
280
260
240
220
200
180
160
140
120 140 160 180 200 220 240 260 280
b
300
280
260
240
220
200
180
160
140
120 140 160 180 200 220 240 260 280
e
320
300
280
260
240
220
200
180
160
100 120 140 160 180 200 220 240 260
f
320
300
280
260
240
220
200
180
160
100 120 140 160 180 200 220 240 260

- Problem: Edges provide redundancy. It is required to find the center of the objects.
- Solution: Smooth Canny edges by scanning in the vertical and horizontal directions.

0	1	1	1	1	1	1	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	1	1	1	0	0	0	0	0
0	1	0	1	0	0	0	0	0	0	0	0
0	1	0	1	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	1	0	0	1	0
0	0	0	0	0	0	0	1	0	0	1	0
0	0	0	0	0	0	0	0	1	0	1	0
0	0	0	0	0	0	0	1	0	1	0	0
0	0	0	0	0	0	1	0	1	0	0	0
0	0	0	0	0	1	0	1	0	0	0	0

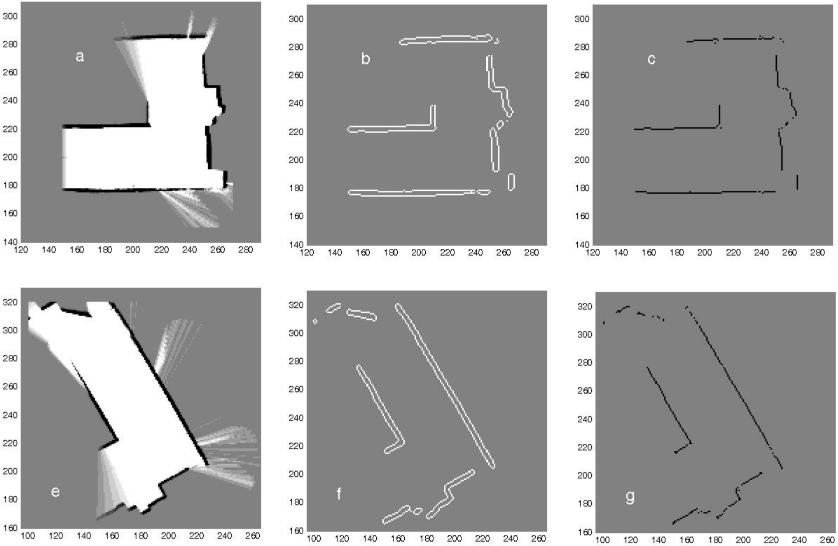
a) original map

0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	1	0	0	0	0	0

b) smoothed after vertical scan

0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	1	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	1	0	0	0	0	0

c) smoothed after horizontal scan


a
b
c
e
f
g

- Problem: look for segments of the map which have unique geometric properties.
- Solution: Choose a random start point($P(0)$) from the smoothed image. Then from the start point choose n points along the edge in one direction. Each point is:

$$P(i) = \begin{bmatrix} x_i \\ y_i \end{bmatrix}, i = 1..n, \tag{1}$$

Then Euclidian distance vector, D and the differential angle vector, Δ are calculated ($\boldsymbol{\Lambda} = (\boldsymbol{D}, \boldsymbol{\Delta})$):

$$\boldsymbol{D} = \left[d_1, d_2, \ldots, d_n\right]^T, \boldsymbol{\Delta} = \left[\delta_1, \delta_2, \ldots, \delta_{n-1}\right]^T, \tag{2}$$

where

$$d_i = ||P(i) - P(0)||,$$
$$\delta_i = \angle(P(i) - P(0)) - \angle(P(i-1) - P(0)). \tag{3}$$

- Problem: We need to make sure that the segments are unique and has enough geometric properties.
- Solution: Use distance and arc histograms:
  - distance histogram: based on $\boldsymbol{D}$.
  - arc histogram: arcs defined as: $\boldsymbol{\Omega} = \left[\omega_1, \omega_2, \ldots, \omega_{n-1}\right]^T$, each arc is defined as $\omega_i = d_{i+1} \sum_{j=1}^{i} \delta_j$.
- distance histogram: $hist_d$, arc histogram: $hist_a$:

$$hist_d = \left[hd_1, hd_2, \ldots, hd_{nd}\right]^T,$$

$$hist_a = \left[ha_1, ha_2, \ldots, ha_{na}\right]^T,$$

nd: number of the bins, hd_i, $i = 1, 2, \ldots, nd$ is the value of each bin for $hist_d$.
na: number of the bins, ha_i, $i = 1, 2, \ldots, na$ is the value of each bin for $hist_a$.

To verify the acceptability of a segment, the following conditions should be met:

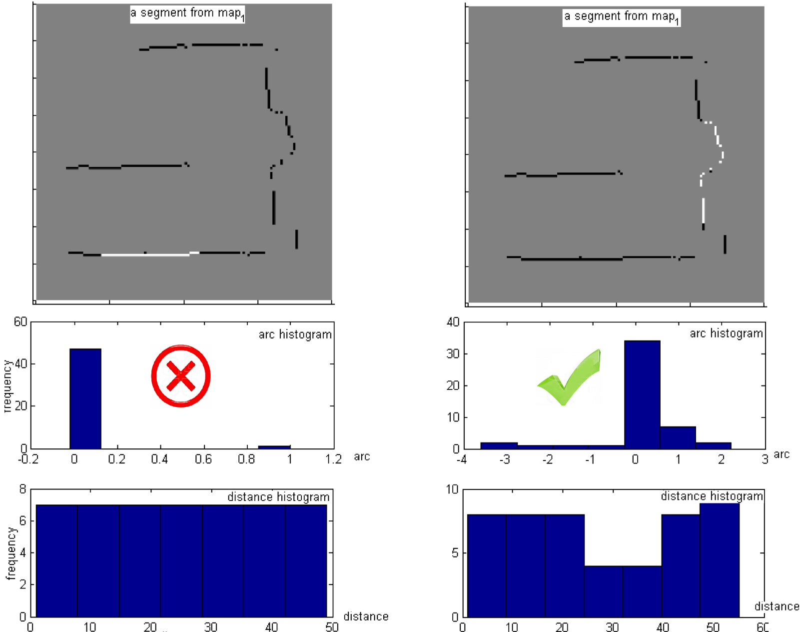
1. $hist_d(i) \neq 0, \forall i = 1 \dots nd$ (all bins should be non zero).
2. The distance histogram should not be smooth:

$$\max(hist_d) - \min(hist_d) \geq h_d, \quad (4)$$

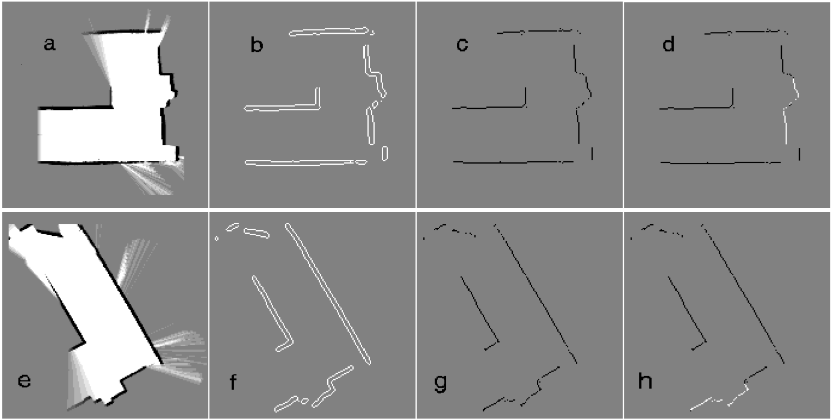
   where h_d is a predefined threshold value.
3. The segment should not be discontinuous. A discontinuity will cause a large value in the arc vector, so the condition:

$$\max(|\omega_i|) \leq \omega_c, \quad (5)$$

   where ω_c is a predefined threshold value will reject segments with discontinuities.


a segment from map1
a segment from map1
arc histogram
frequency
arc
arc histogram
arc
distance histogram
frequency
distance
distance histogram
distance

- To find the similarity of the selected segments, cross correlation is applied to the normalized:
  - distance histogram and
  - arc histogram.
- If the distance and arc histograms are similar, then the result of each cross correlation will be close to one and the segments are deemed to be a match.
- Otherwise, another segment from map_2 is selected and the process continues.
- If, for the selected segment from map_1, no correspondence from map_2 can be found, then another segment from map_1 should be selected.


a
b
c
d
e
f
g
h

After finding matching segments from both maps approximate relative pose is extracted. First rotation and then the translation.

- rotation:
  - the difference in the slopes of the two fitted lines on segments:
  - $\alpha_{app} = \arctan a_1 - \arctan a_2$
  - a_1: slope of the fitted line for the segment of map_1
  - a_2: slope of the fitted line for the segment of map_2
- translation:
  - : the difference in the centroids of segments:
  - $T_{app} = \begin{bmatrix} x_{app} \\ y_{app} \end{bmatrix} = \sum_{k=1}^{n_1} \frac{R_{\alpha_{app}} P_{seg1}(k)}{n_1} - \sum_{k=1}^{n_2} \frac{P_{seg2(k)}}{n_2}$
  - $R_{\alpha_{app}}$: the rotation matrix based on α_{app},
  - P_{seg1}, P_{seg2}: the sets of points in segments 1 and 2,
  - n_1, n_2: the cardinalities of P_{seg1} and P_{seg2}.

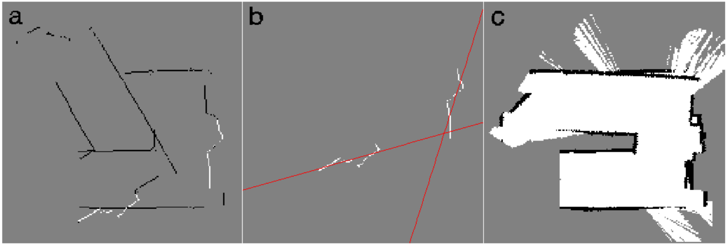
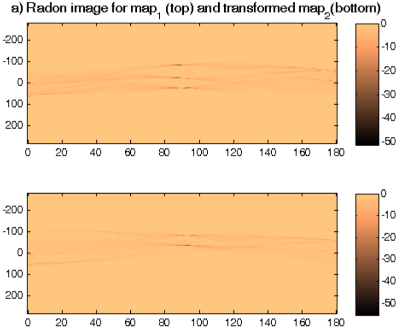
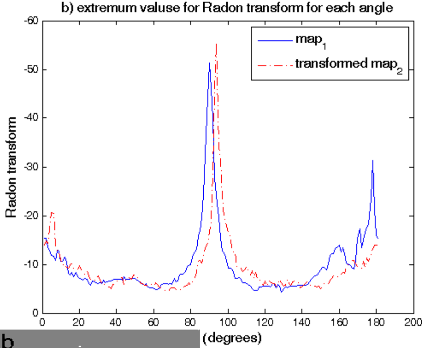


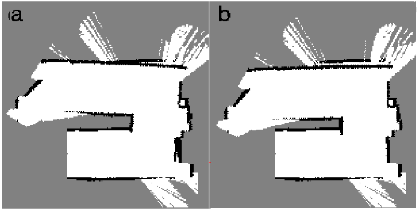
Figure: **a)** Both maps in the same coordinates with marked segments. **b)** Fitted lines on segments to extract relative orientation. **c)** Both maps in the coordinates of map_1 after approximate transformation.

The approximate relative pose is tuned.

- rotation:
  - Radon transform
  - $\alpha_{ext} = \alpha_{app} + \delta_{tuning}$,
  - $\delta_{tuning} = \arg\min_\theta(\mathcal{R}_{\theta=0:180}(m_1)) - \arg\min_\theta(\mathcal{R}_{\theta=0:180}(T_{x,y,\theta}(m_2)))$.
  - $\mathcal{R}_{\theta=0:180}(map)$: the Radon image of the input map.
  - $T_{x,y,\theta}(map)$: translation vector.
  - m_1 and m_2 are map_1 and map_2.
- translation:
  - Similarity Index
  - $agr(map_1, map_2) = \sharp\{p = (x, y)|map_1(p) = map_2(p)\}$,
  - $dis(map_1, map_2) = \sharp\{p = (x, y)|map_1(p) \neq map_2(p)\}$,
  - $J_{similarity} = dis(map_1, map_2) - agr(map_1, map_2)$.
  - $T_{ext} = \left[x_{ext}, y_{ext}\right]^T = \arg\max_{\mathcal{S}(x,y)}(|J_{similarity}(x, y)|)$ $\{\mathcal{S}(x, y) \subset R^2|x_{app} - \delta_x < x < x_{app} + \delta_x,$ $y_{app} - \delta_y < y < y_{app} + \delta_y\}$.


a) Radon image for map$_1$ (top) and transformed map$_2$(bottom)


b) extremum valuse for Radon transform for each angle
map$_1$
transformed map$_2$
Radon transform
(degrees)


a
b

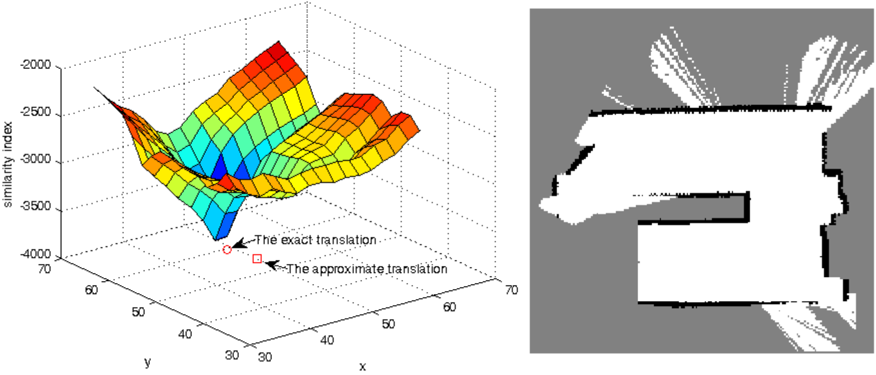


Figure: Tuning translation using similarity index

The validity of the transformation can be verified using either similarity index or image entropy.

- Similarity index:

$$V(m_1, m_2)\Big|_{R_{ext}, T_{ext}} = \frac{agr(m_1, m_2) \times 100\%}{agr(m_1, m_2) + dis(m_1, m_2)},$$

- image entropy:
  - $\mathcal{H} = -\sum p \log_2(p)$,
  - p: the normalized histogram of an image.

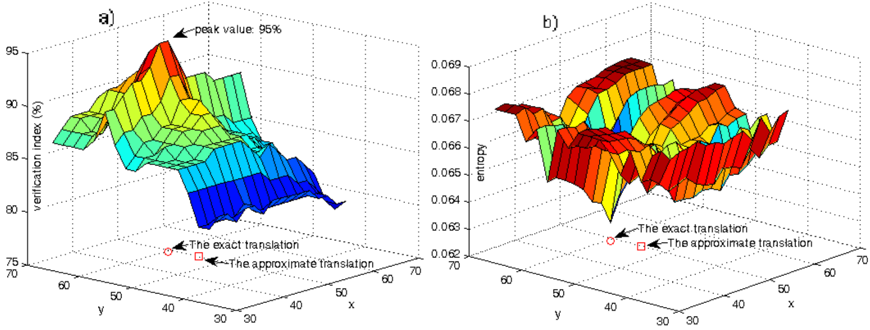
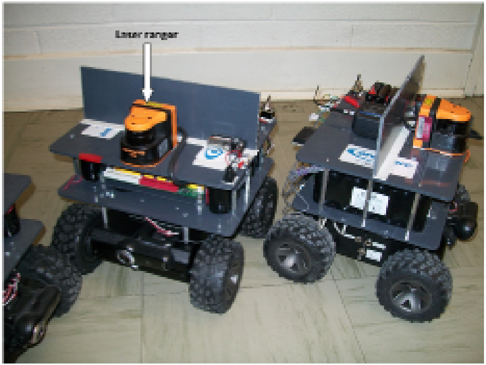
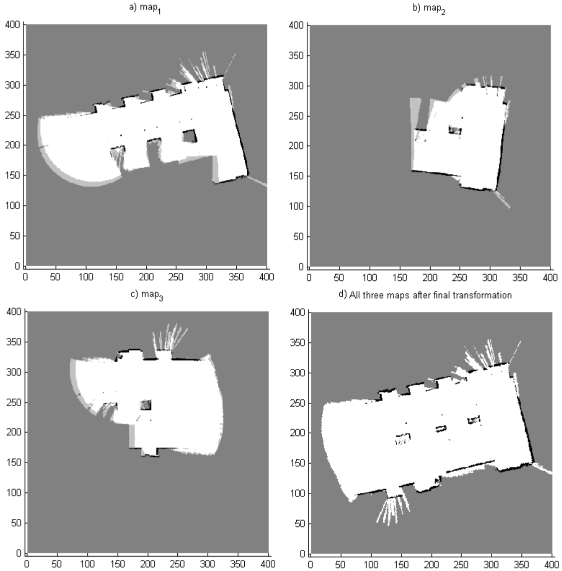


Figure: Final verification using **a)** similarity index and **b)** image entropy.

- CoroBots: Hokuyo UBG-05LN and Phidget Encoders



- Laser Odometry: Iterative Closest Point (ICP)
- Data Fusion: Extended Kalman Filter (EKF)



Processign time: this ($\approx$ 80), Random walk ($\approx$ 154)

- Map fusion achieved by applying a layerd algorithm. including two main steps:
  - Approximate pose finding and
  - Tunning the pose.
- The results were verified at the end.
- future work: improving the approximate pose extraction by using better search algorithms.

Thank You.