First scan matching algorithms

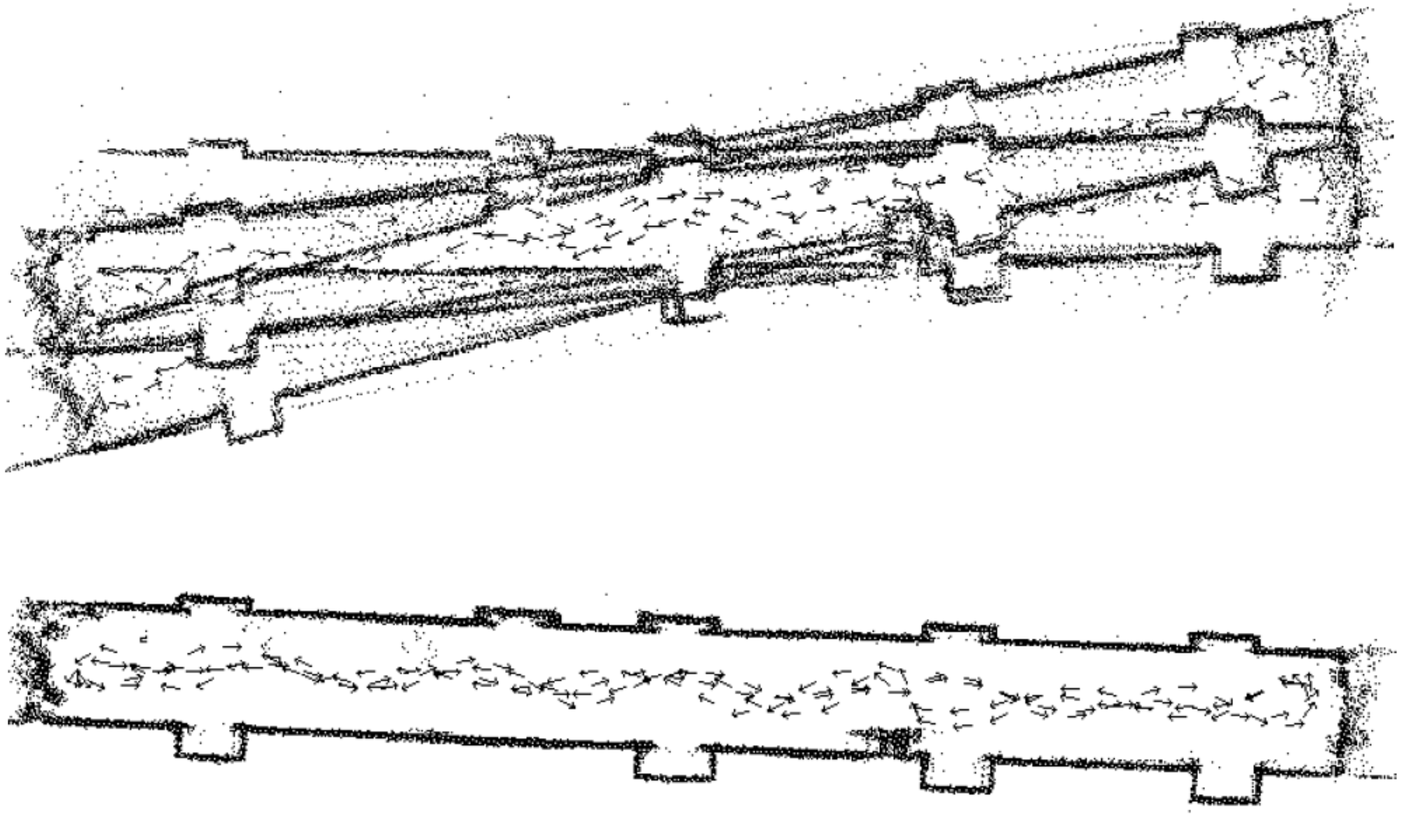
Alberto Quattrini Li

2015-10-22

University of South Carolina

Robot mapping through scan-matching

- Framework for consistent registration of multiple frames of measurements (e.g., scans) [Lu and Milios, 1997, Auton Robot]



Source: [Lu and Milios, 1997, Auton Robot]

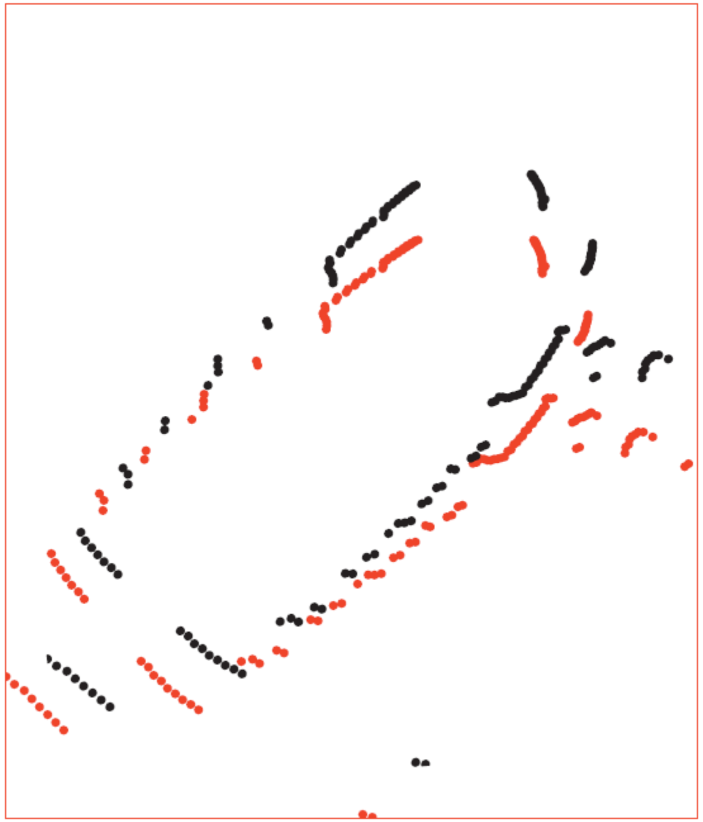
Scan

- *Scan*: sequence of *scan points* representing the contour curve of the local environment
  - *Scan point*: represented with polar coordinates
  - A scan is relative to a pose typically stored as Cartesian coordinates

Problem statement

- Input:
  - Starting pose P_{ref} and the associated scan S_{ref}
  - New scan S_{new} from P'_{new}
- Output:
  - Rotation ω
  - translation T

  for S_{new} such that S_{new} is aligned with S_{ref}

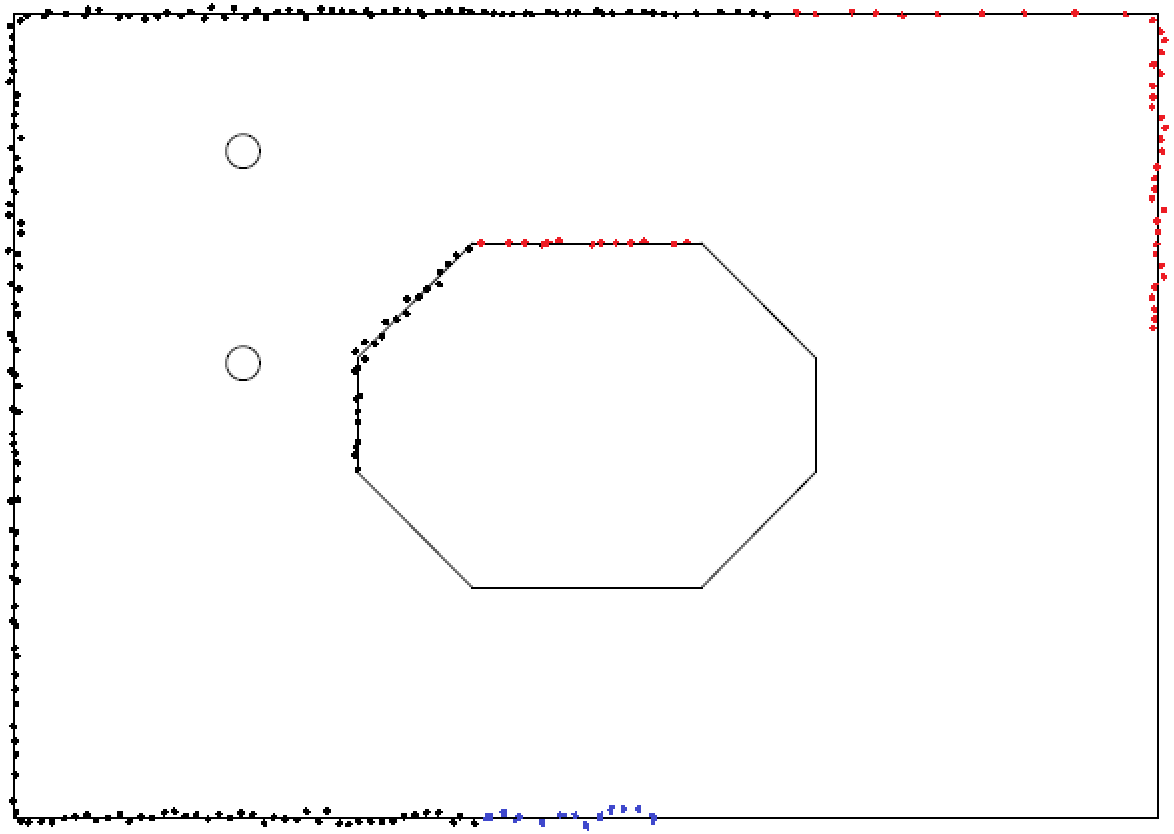


Source: Courtesy of Noel Welsh

Scan

- Characterized by two types of discrepancies
  - Random sensing noise
  - Occlusion

Source: Adaptation of [Lu and Milios, 1997, Auton Robot]



Scan alignment search overview

- Restrict search to translation and rotation (rigid transform)
- Start with an initial guess from odometry
- Find matching points/outliers
- Minimize a distance function between the matching points of the scans

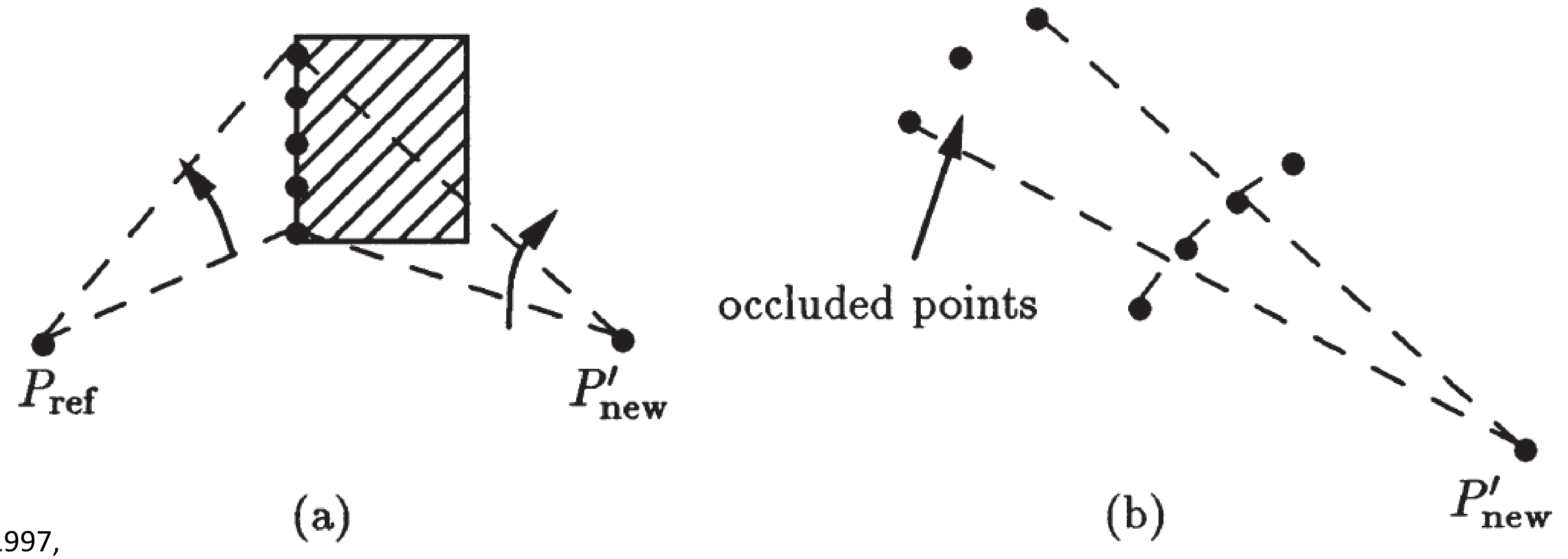
- Two methods by Lu and Milios [1997, J Intell Robot Syst]

Method 1. - Search/Least-Squares Matching

```
Input: P_ref, S_ref, P'_new, S_new
Output: ω, T
Project S_ref to P'_new;
Compute the tangent directions on each scan point;
for i ← 1 to N_iterations do
    Select ω from a global search procedure that minimizes the matching
    distance function;
    For each point on S_new find an approximate corresponding point S_ref;
    Find least-squares solution of T;
    Update the rigid transformation;
end
```

Step 1 (Projection of reference scan)

- Change of coordinate system for points in S_{ref}
- Determine each point in S_{ref} is visible from P'_{new}



Source: [Lu and Milios, 1997,
J Intell Robot Syst]

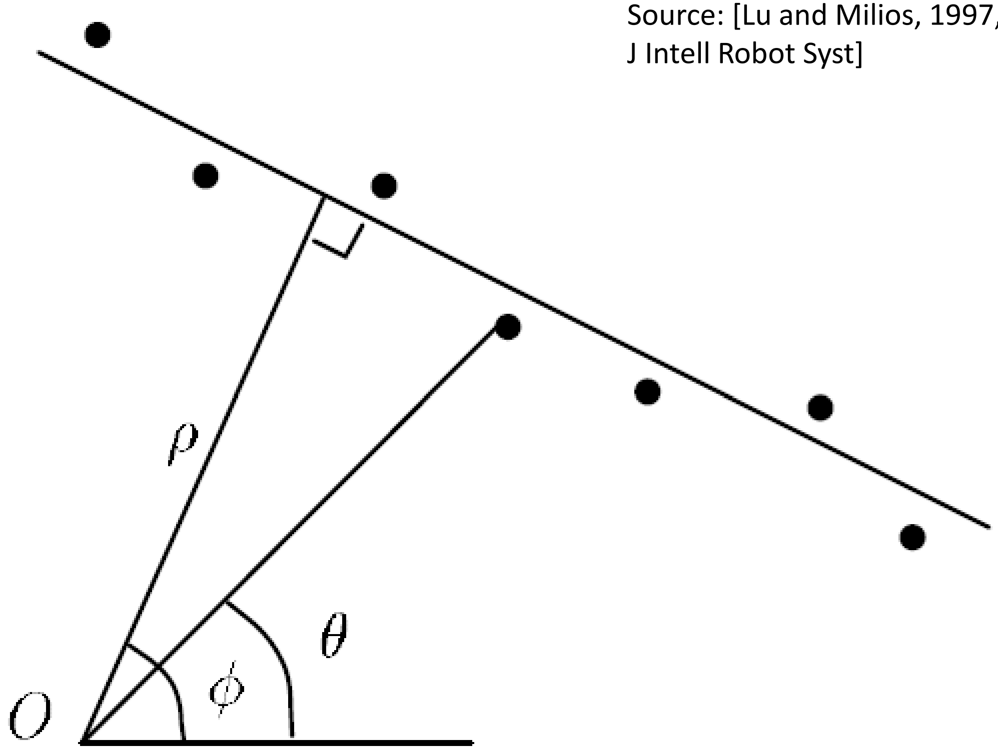
Step 2 (tangent lines)

- Minimize the error

$$E_{\text{fit}} = \sum_{i=1}^{n} (x_i \cos\phi + y_i \sin\phi - \rho)^2$$

where

- n neighborhood size
- x_i, y_i scan point
- ρ, ϕ distance from the origin to the line and direction of a normal to the line

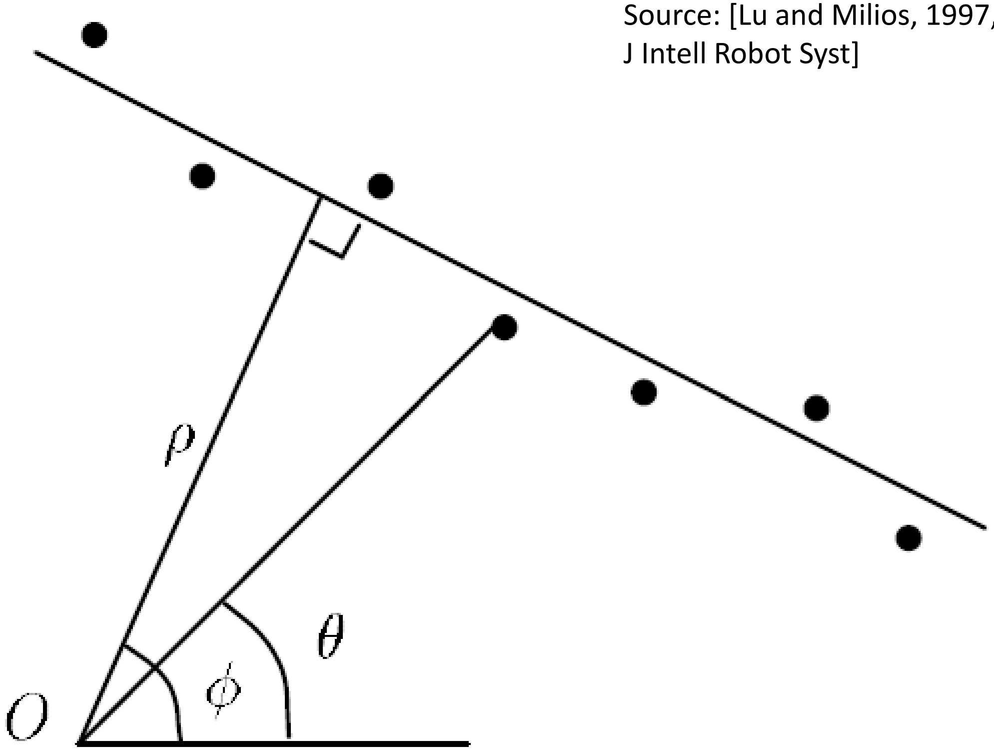


Source: [Lu and Milios, 1997, J Intell Robot Syst]

Step 2 (tangent lines)

- Discard lines
  - Near corners
    - Upper threshold on E_{fit}
  - Occlusion boundaries
    - Upper threshold on incidence angle $\theta_i - \phi$

Source: [Lu and Milios, 1997, J Intell Robot Syst]



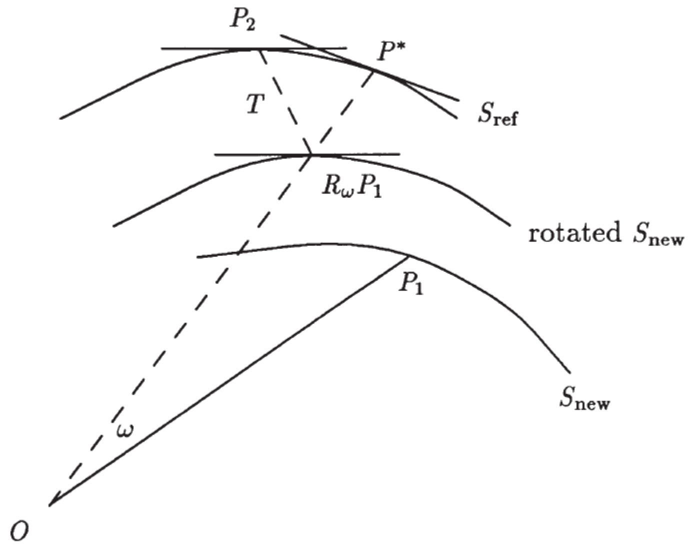
Step 4 (find ω) – matching points

- A correspondence pair (matching points P_i, P^*) is accepted only if

$$(R_\omega \vec{n}_i) \cdot \vec{n}^* \geq \cos\alpha \wedge |D_i| \leq H_d$$

where

- R_ω is the rotation matrix
- $\vec{n}_i, \vec{n}^*$ normal directions of the tangent lines
- α, H_d thresholds
- D_i expresses $P*$ translation



Step 4 (find ω)

- Search using the golden section method

$$E_{\text{match}}(\omega) = \frac{1}{n_p + n_o}(\min_T E(\omega, T) + n_o H_d^2)$$

where

- n_p and n_o number of matching pairs of points and outliers, respectively
- H_d^2 is the constant cost of an outlier

Step 5-6 (find T)

- Distance between two scans in which we find T given ω

$$E(\omega, T) = \sum_{i=1}^{n_p} (C_i^x T_x + C_i^y T_y - D_i)^2$$

by using the least-squares solution

- Note that the coefficients can be derived from the following relationship

$$(R_\omega \vec{n}_1 + \vec{n}^*)T \approx (R_\omega \vec{n}_1 + \vec{n}^*)(P^* - R_\omega P_1)$$

Summing up

- The rotation search/least-squares algorithm is able to robustly solve for the transformation even in the presence of large initial pose error
- The solution may not be highly accurate

Method 2. - IDC algorithm

Input: $P_{\text{ref}}, S_{\text{ref}}, P'_{\text{new}}, S_{\text{new}}$

Output: ω, T

while *Error* $>$ *Threshold* **do**

- $P_{\text{cp}} \leftarrow \text{closestPoint}(S_{\text{ref}}, S_{\text{new}})$;
- $P_{\text{mrp}} \leftarrow \text{matchingRangePoint}(S_{\text{ref}}, S_{\text{new}})$;
- Find $(\omega_{\text{cp}}, T_{\text{cp}})$ from P_{cp} and $(\omega_{\text{mrp}}, T_{\text{mrp}})$ from P_{mrp} by minimizing the error (least-squares solution);
- Choose transformation of $(\omega_{\text{mrp}}, T_{\text{cp}})$;

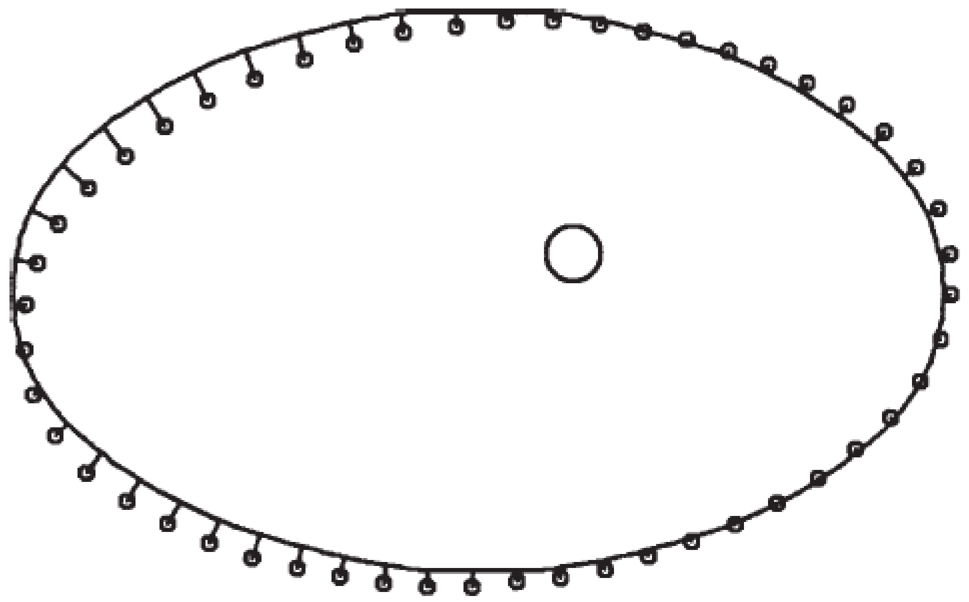
end

Iterative Dual Correspondence

- IDC uses two methods to define matching points:
  - Closest point
  - Matching range point

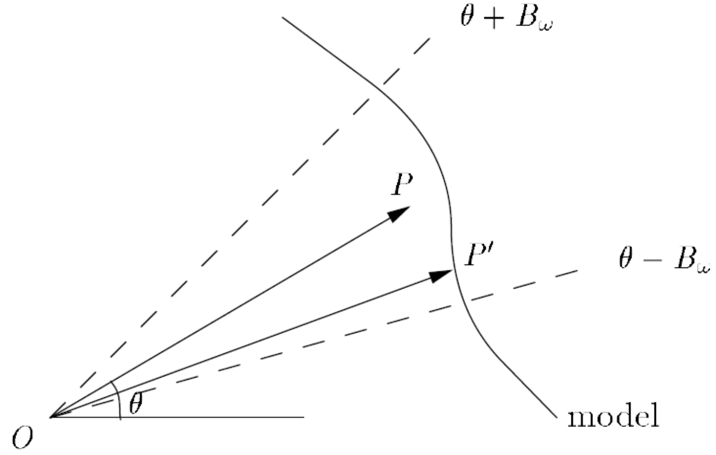
Closest point rule

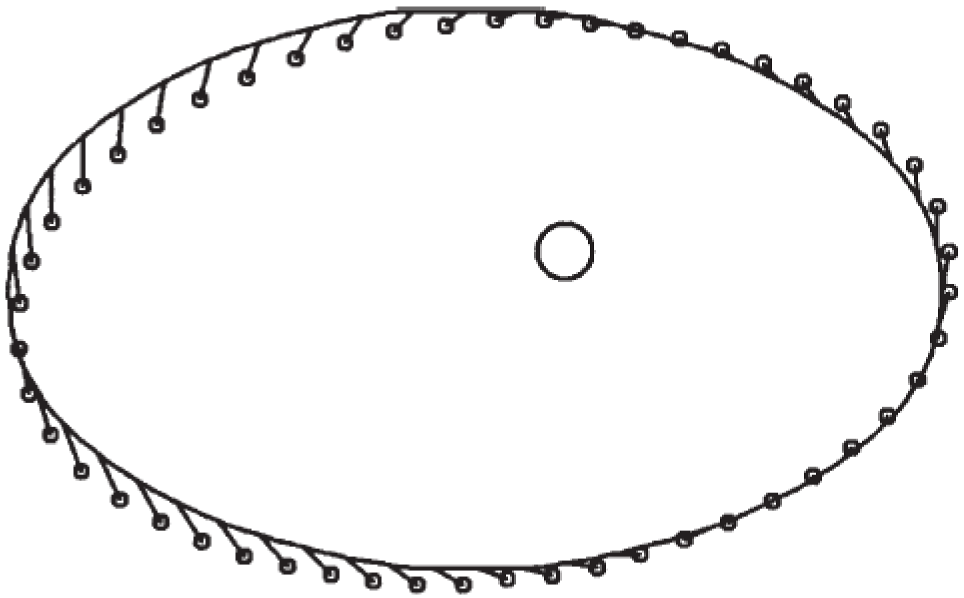
- Match two points that are closest together
- Typically informative of translation but not rotation



Matching range point rule

- Match points that have the same distance (range) from their respective poses and are within a predetermined rotation of one another
- Typically informative of rotation but not translation
- Assumes initial poses are close together (T is negligible)





Find corresponding points

- Corresponding points P (of S_{new}) and P' (of S_{ref}) are found using the two rules
- Interpolation is necessary as S_{ref} is discrete
  - Interpolation from simply connecting two adjacent points with a line segment (closest point rule)
  - Linear interpolation between two points (matching range point rule)

$$\hat{r} = \frac{r_1 r_2(\theta_2 - \theta_1)}{r_1(\hat{\theta} - \theta_1) + r_2(\theta_2 - \hat{\theta})}$$

- Outliers are detected and discarded according to

$$||P'| - |P|| > B_r$$

Minimizing the error

- To find the translation and rotation, minimize the squared distance between the matching points:

$$E_{dist}(\omega, T) = \sum_{i=1} n|R_\omega P_i + TP_i'|^2$$

where:

- ω is the angle
- T is the translation
- P_i and P_i' are matching points
- R_ω is the rotation matrix for a rotation of angle ω

Summing up

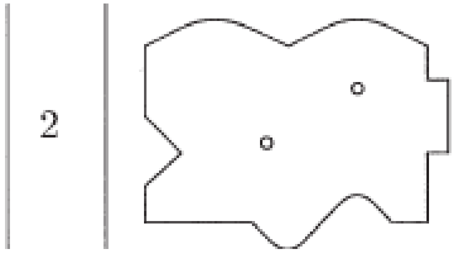
- More accurate solution as long as it converges (experimentally found 15-20 iterations are sufficient)

Experiments

- Robot with a laser range finder mounted on a pan/tilt unit that rotates to have a uniformly distributed scan points
- Simulation
  - Matching process run 1000 times with randomly generated initial pose error and sensing noise in different environments
- Real robot
  - Some test in some environments with ARK robot with Optech G150 laser rangefinder
  - Ground truth not available

Results in simulation

- Results for the two algorithms ran independently
- The IDC algorithm performs better for rotation
- Justifies the use of the combination of two

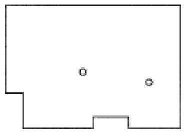
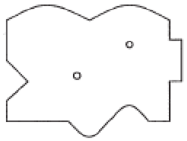
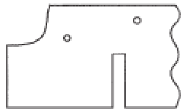
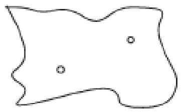
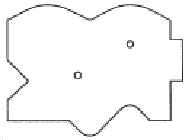
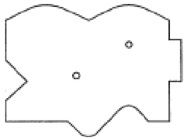


Residual		First algorithm	Second algorithm
Rotation	σ_ω	$0.5375°$	$0.1599°$
Translation x	σ_x	0.7652 cm	0.7827 cm
Translation y	σ_y	0.7998 cm	0.6514 cm

Results in simulation

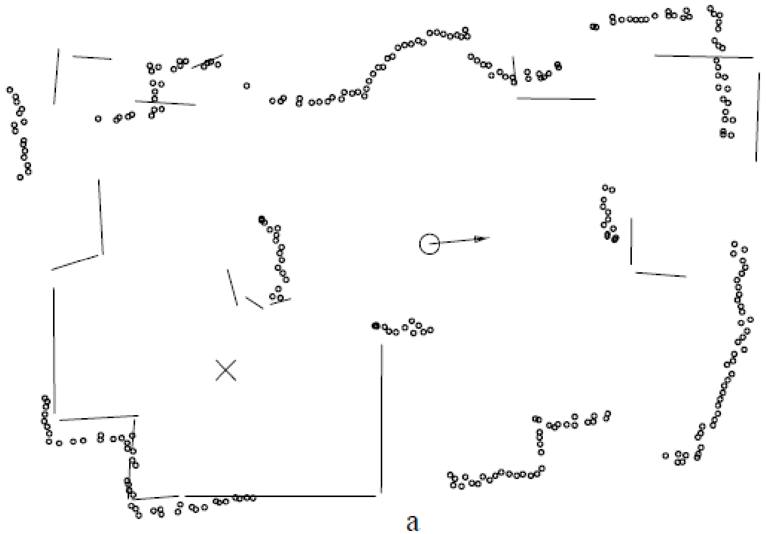
- The algorithms are able to deal with non-polygonal environments
- As the sensing noise increases, also the residuals standard deviations increase

Table I. Statistics of experiments in simulated environments. Maximum initial rotation and translation are set at $\pm 14.3^\circ$ and 50 cm, respectively

No.	Simulated Environments and Maximum Sensing Noise		Residual Standard Deviations		
				After Stage 1	After Stage 2
1		noise: $\pm$5cm	σ_ω	0.2786°	0.0547°
			σ_x	0.3909 cm	0.3418 cm
			σ_y	0.4531 cm	0.2702 cm
2		noise: $\pm$5cm	σ_ω	0.3668°	0.0754°
			σ_x	0.4150 cm	0.3592 cm
			σ_y	0.3886 cm	0.3146 cm
3		noise: $\pm$10cm	σ_ω	0.4414°	0.1876°
			σ_x	1.3449 cm	1.0436 cm
			σ_y	1.5336 cm	0.8532 cm
4		noise: $\pm$10cm	σ_ω	0.3970°	0.1824°
			σ_x	0.8723 cm	0.9535 cm
			σ_y	0.7836 cm	0.8446 cm
5		noise: $\pm$15cm	σ_ω	0.6090°	0.3027°
			σ_x	1.2268 cm	1.2604 cm
			σ_y	1.1269 cm	1.1438 cm
6		noise: $\pm$20cm	σ_ω	1.1517°	0.6230°
			σ_x	2.2832 cm	2.5478 cm
			σ_y	2.1961 cm	2.1811 cm

Comparison with Cox algorithm

- Cox algorithm [1991, IEEE T Robotic Autom]
  - Iterative closest point
  - Adapted to have a line-segment based map of the environment
- The proposed approach and the Cox algorithm perform relatively good

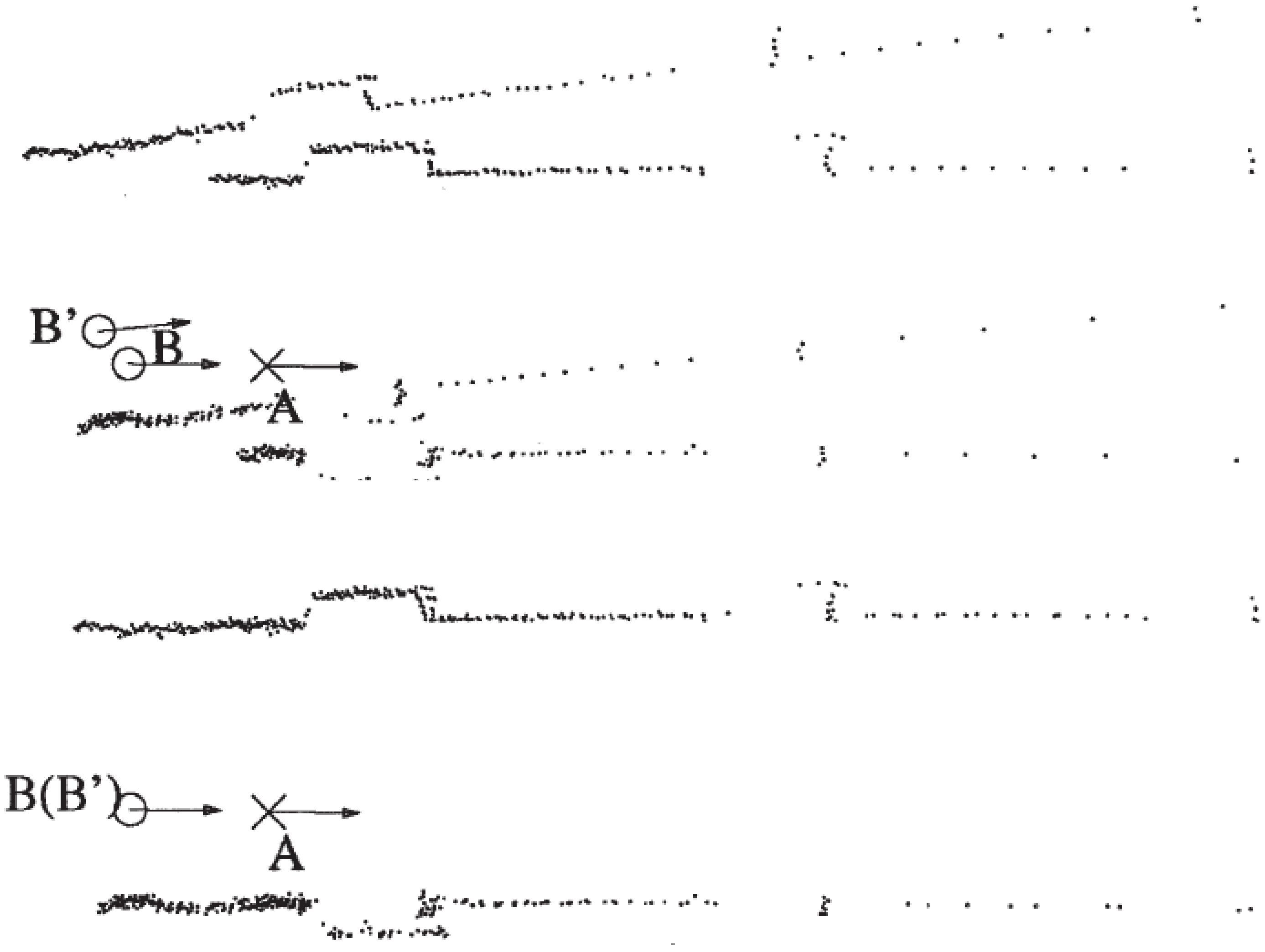


a

Residual		Point-Line Algo.	Point-Point Algo.
Rotation	σ_ω	0.2277 °	0.1992 °
Translation x	σ_x	1.8281 cm	1.6375 cm
Translation y	σ_y	1.2338 cm	1.3128 cm

Results with real data

- Errors in sensing and odometry are not very high, compared to simulation
- The algorithms work in some real indoor environments



Discussion

- Straight corridors?
- Limited field of view?
- What if last scan is misaligned?
- What kind of representation for the map?
- Multiple robots?
- …