Simultaneous Localization and Mapping (SLAM)

RSS Lecture 16

April 8, 2013

Prof. Teller

Text: Siegwart and Nourbakhsh S. 5.8

SLAM Problem Statement

- Inputs:
  - No external coordinate reference
  - Time series of proprioceptive and exteroceptive measurements* made as robot moves through an *initially unknown* environment
- Outputs:
  - A *map** of the environment
  - A robot *pose estimate* associated with each measurement, in the coordinate system in which the map is defined

*Not yet fully defined

SLAM Problem -- Incremental

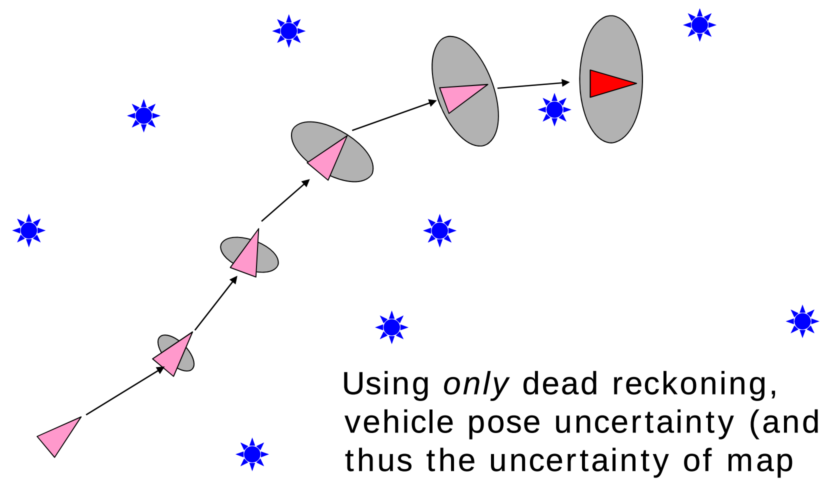
- State/Output:
  - –Map of env’t observed “so far”
  - –Robot pose estimate w.r.t. map
- Action/Input:
  - –Move to a new position/orientation
  - –Acquire additional observation(s)
- Update State:
  - –Re-estimate the robot’s pose
  - –Revise the map appropriately

SLAM Aspects

- What is a measurement?
- What is a map?
- How are map, pose coupled?
- How should robot move?
- What is hard about SLAM?

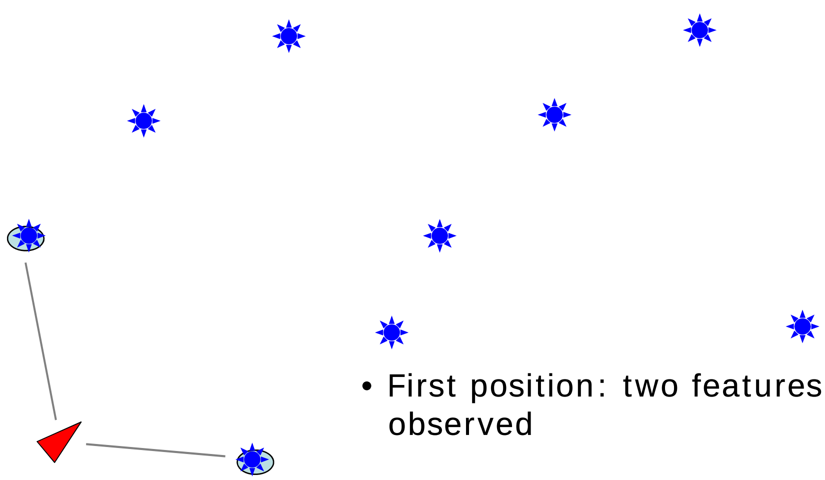
- But first: some intuition

Intuition: SLAM without Landmarks



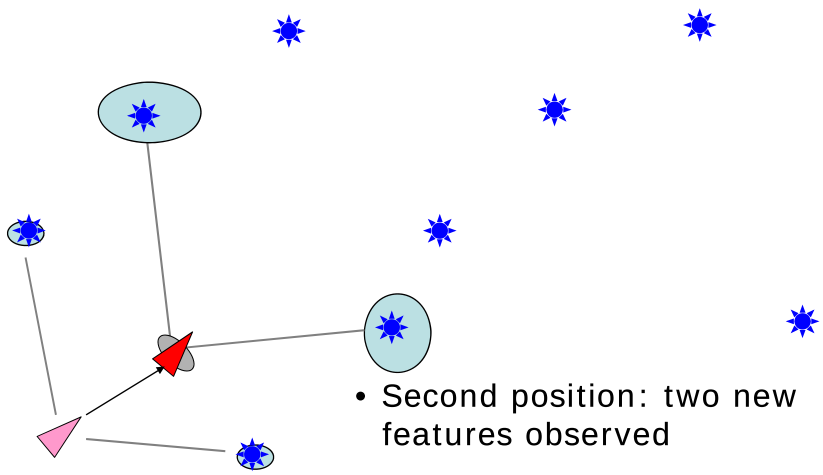
Using *only* dead reckoning, vehicle pose uncertainty (and thus the uncertainty of map features) grows without bound

With Landmark Measurements



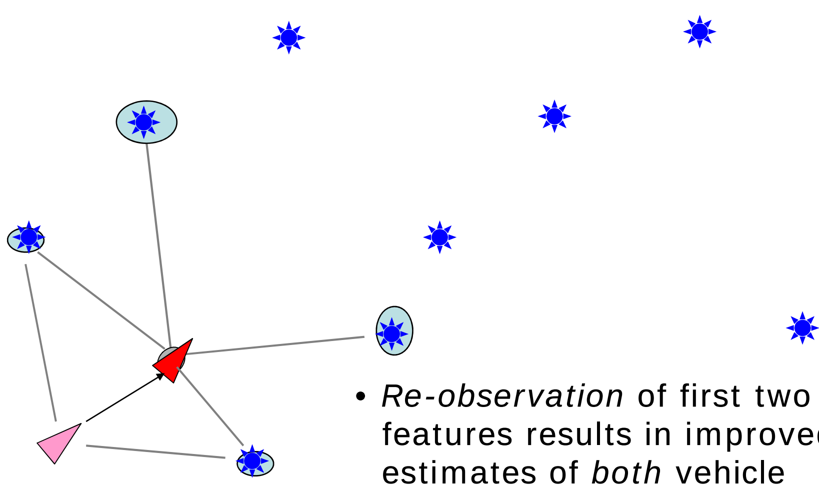
- First position: two features observed

Illustration of SLAM with Landmarks



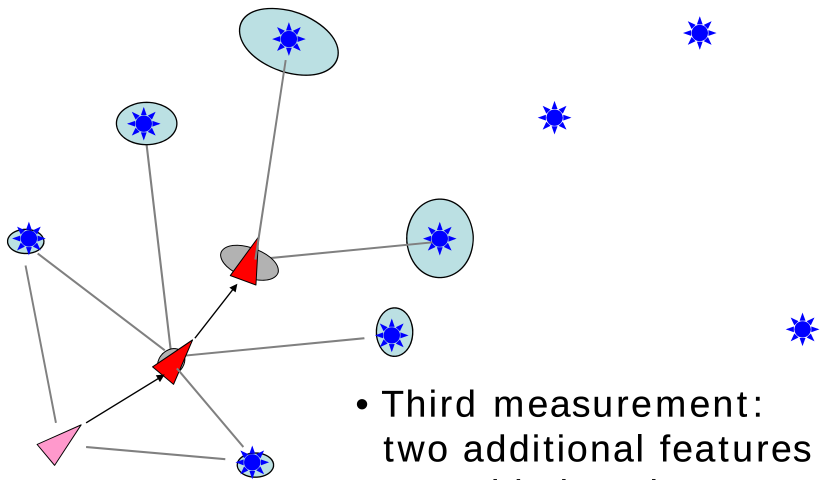
- Second position: two new features observed

Illustration of SLAM with Landmarks



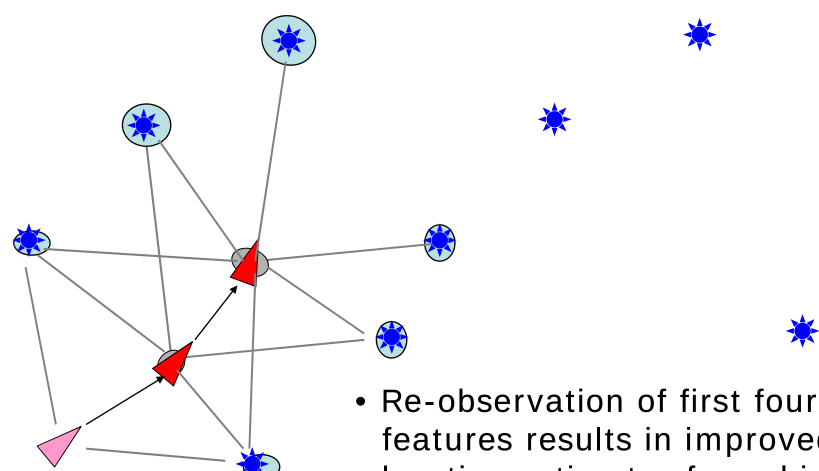
- *Re-observation* of first two features results in improved estimates of *both* vehicle pose and features

Illustration of SLAM with Landmarks



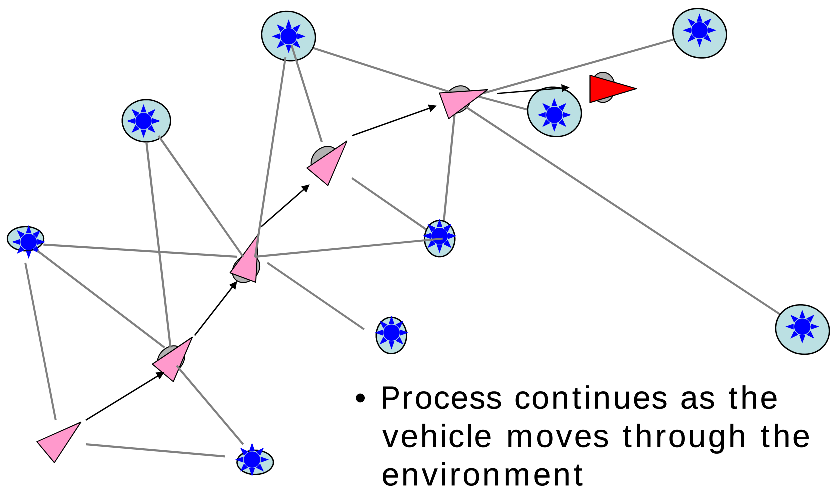
- Third measurement: two additional features are added to the map

Illustration of SLAM with Landmarks



- Re-observation of first four features results in improved location estimates for vehicle poses and all map features

Illustration of SLAM with Landmarks



- Process continues as the vehicle moves through the environment

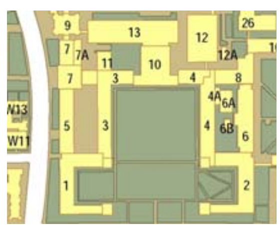
Why is SLAM Hard?

- “Grand challenge”-level robotics problem
  - Autonomous, persistent, collaborative robots mapping multi-scale, generic environments
- Map-making = learning
  - Difficult even for humans
  - Even skilled humans make mapping mistakes
- Scaling issues
  - Space: Large extent (combinatorial growth)
  - Time: Persistent autonomous operation
- “Chicken and Egg” nature of problem
  - If robot had a map, localization would be easier
  - If robot could localize, mapping would be easier
  - … But robot has neither; starts from blank slate
  - Must also execute an *exploration strategy*
- Uncertainty at every level of problem

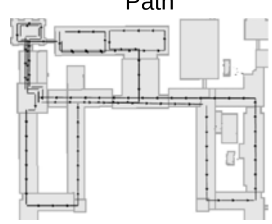
Uncertainty in Robotic Mapping

Uncertainty: / Scale:	Continuous	Discrete
Local	Sensor noise	Data association
Global	Navigation drift	Loop closing

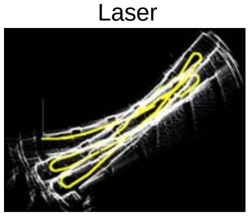
MIT Killian Court



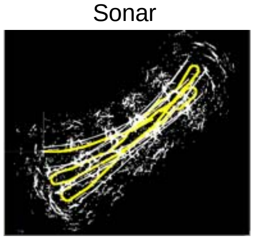
Path

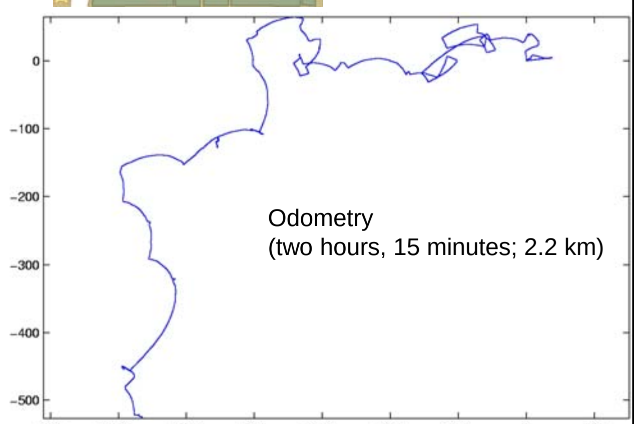


Laser



Sonar

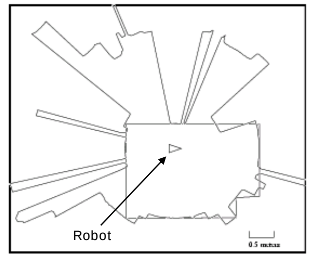




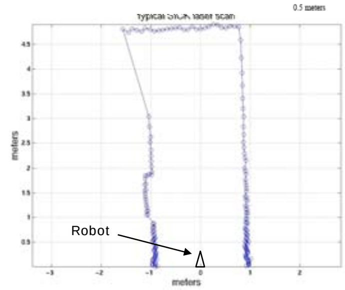
Common range-and-bearing sensors

Polaroid sonar ring
12 range returns, one per 30 degrees, at ~4 Hz

(+ servoed rotation)

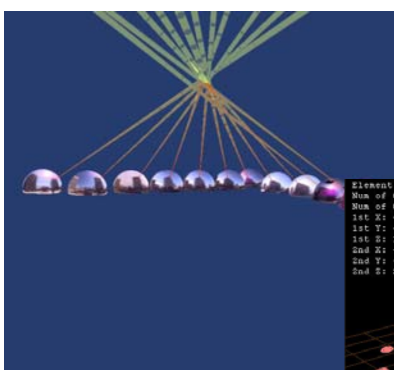


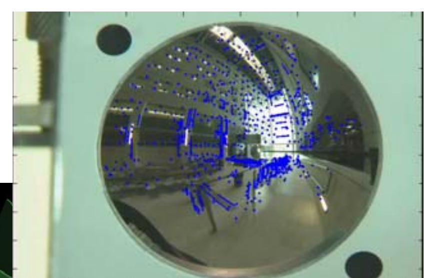
SICK laser scanner
180 range returns, one per degree, at 5-75 Hz



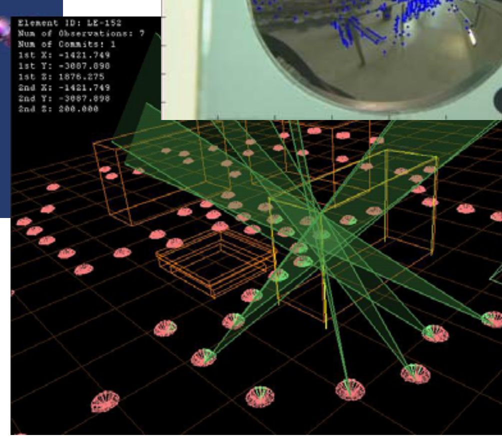
Other possibilities: Stereo/monocular vision; Robot itself (stall, bump sensing)

Tracking & long-baseline monocular vision



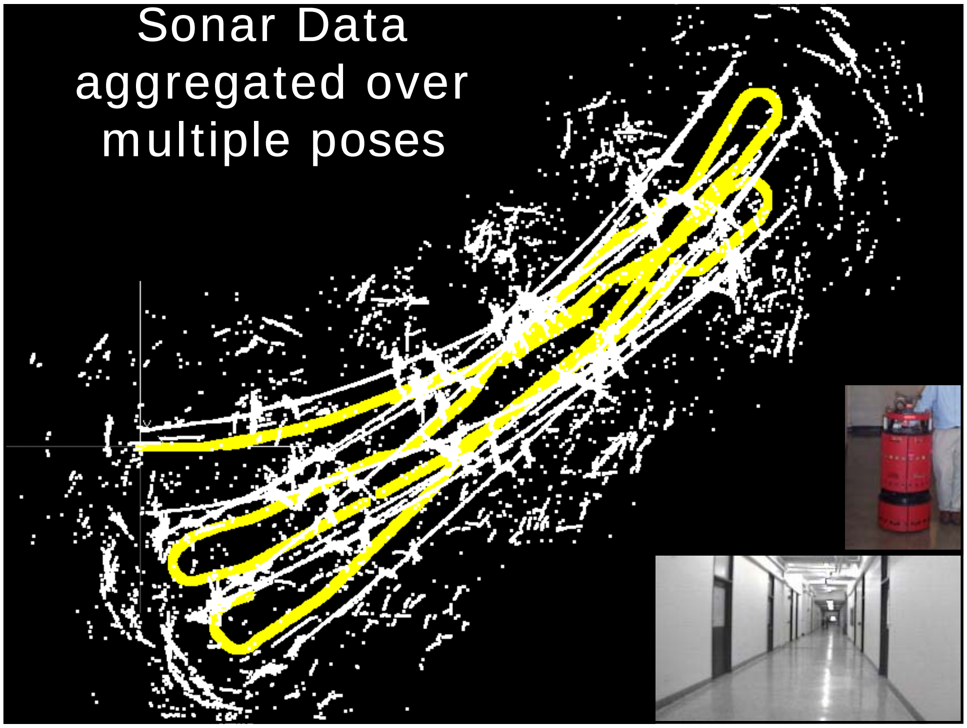


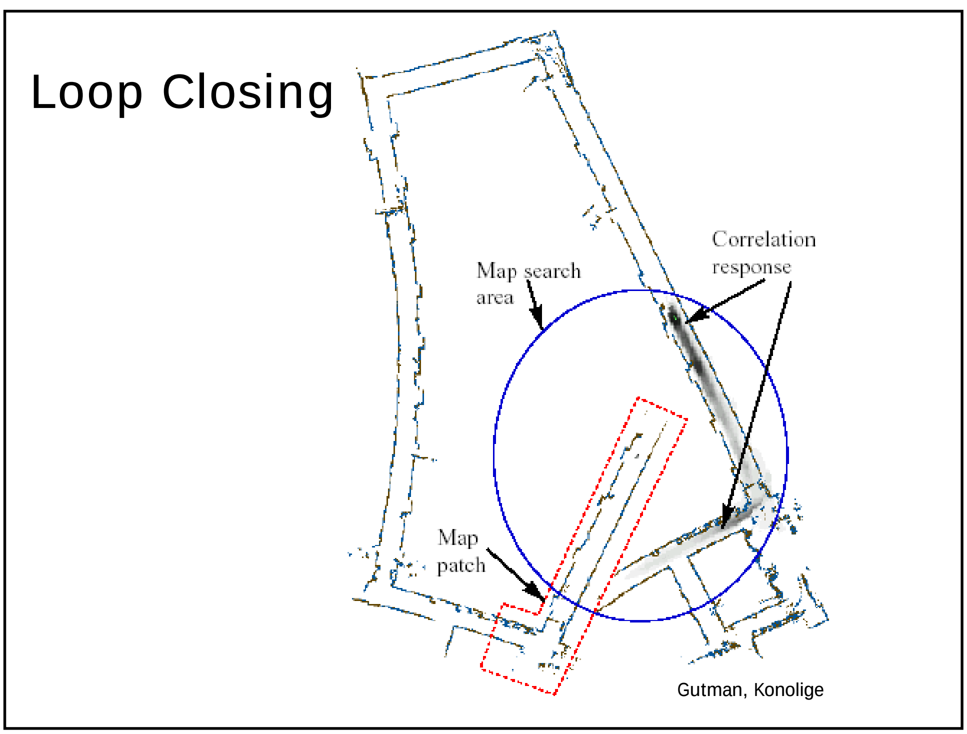
Bosse

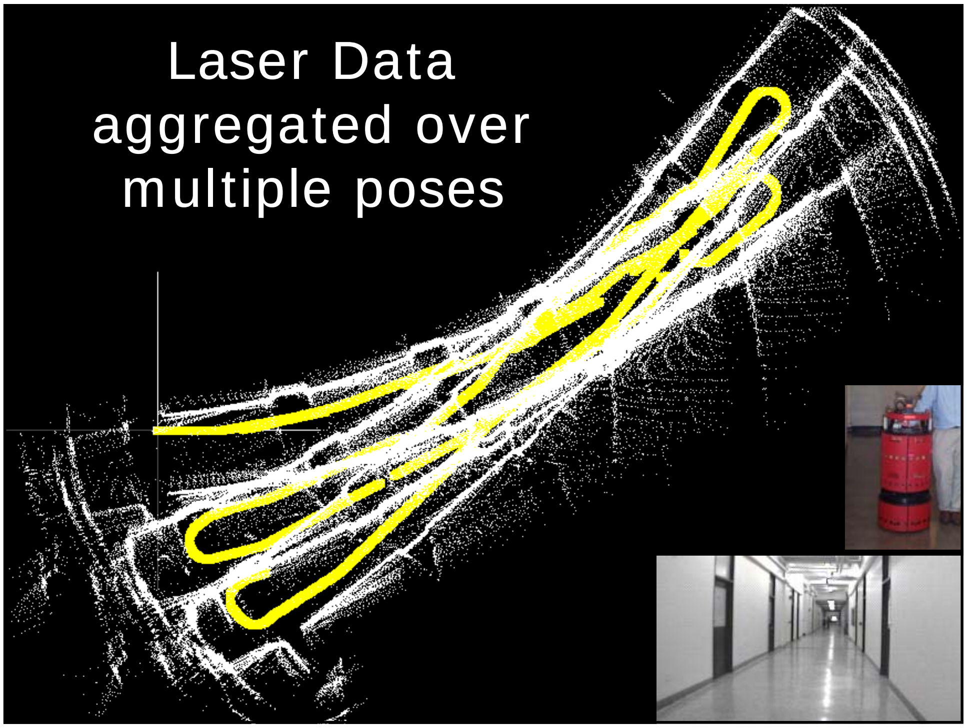


Track points, edges, texture patches from frame to frame; triangulate to recover local 3D structure. Also called "SFM," **S**tructure **F**rom camera **M**otion, or object motion in the image

Chou


Sonar Data
aggregated over
multiple poses

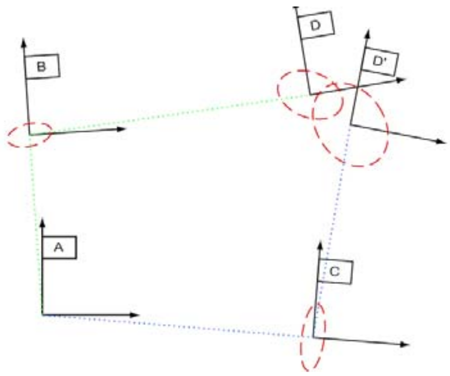

Loop Closing
Correlation
response
Map search
area
Map
patch
Gutman, Konolige

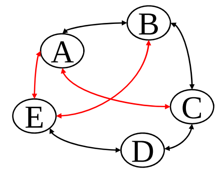


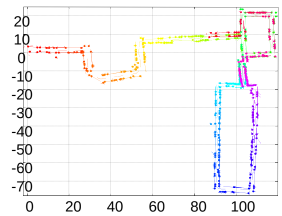
What is a map?

- Collection of *features* with some *relationship* to one another
- What is a *feature*? ← Uncertainty
  - Occupancy grid cell
  - Line segment
  - Surface patch
- What is a feature *relationship*?
  - Rigid-body transform (metrical mapping)
  - Topological path (chain of co-visibility)
  - Semantics (label, function, contents)

Atlas hybrid maps (Bosse et al.)



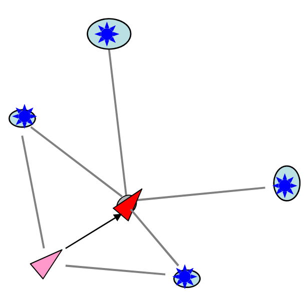




- Features: point, line, patch clouds
- Geometry: rigid frames, submaps
- Topology: map adjacencies
- Hybrid: uncertain map-to-map transformations

What is *pose* w.r.t. a map?

- Pose estimate that is (maximally) *consistent* with the estimated features observed from vicinity
- Consistency can be evaluated locally, semi-locally, or globally

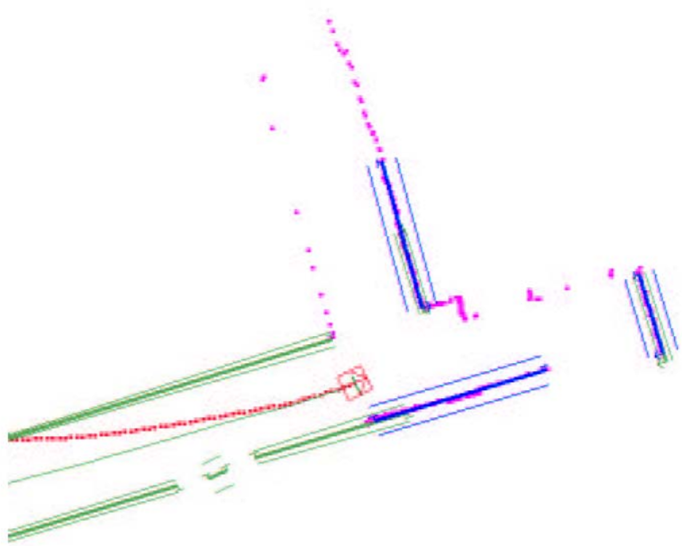


- Note tension between estimation *precision* and solution *consistency*

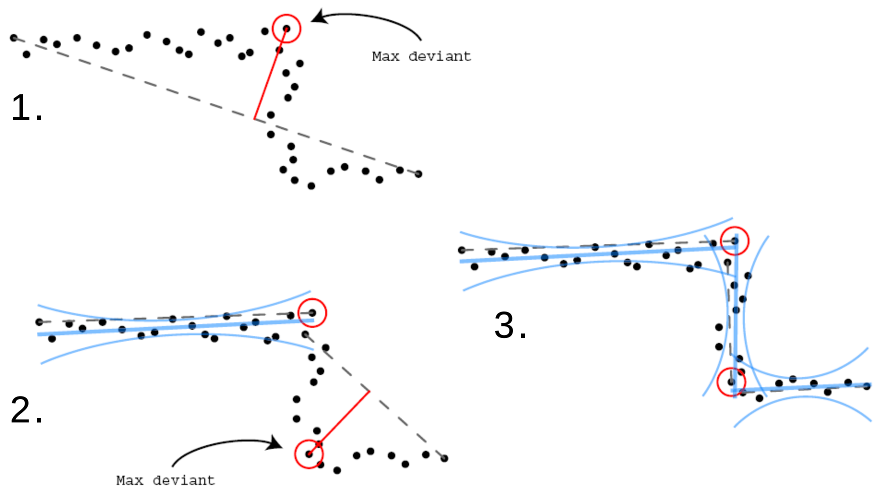
Example

- SLAM with laser scanning
- Observations
- Local mapping
  - Iterated closest point
- Loop closing
  - Scan matching
  - Deferred validation
  - Search strategies

Observations

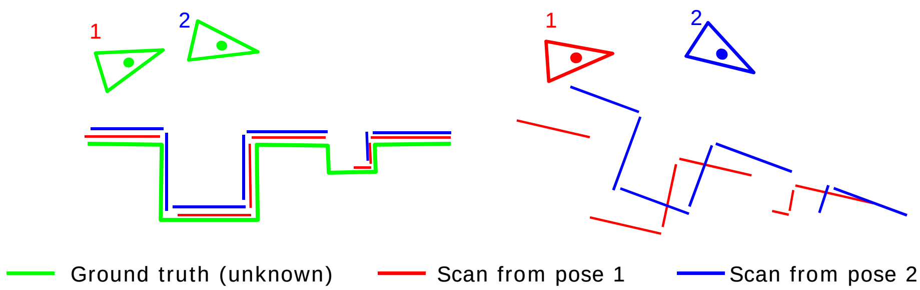


Observations



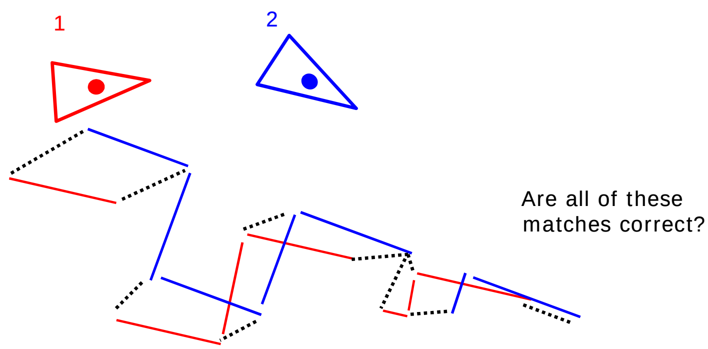
Scan Matching

- Robot scans, moves, scans again
- Short-term odometry/IMU error causes misregistration of scans
- *Scan matching* is the process of bringing scan data into alignment



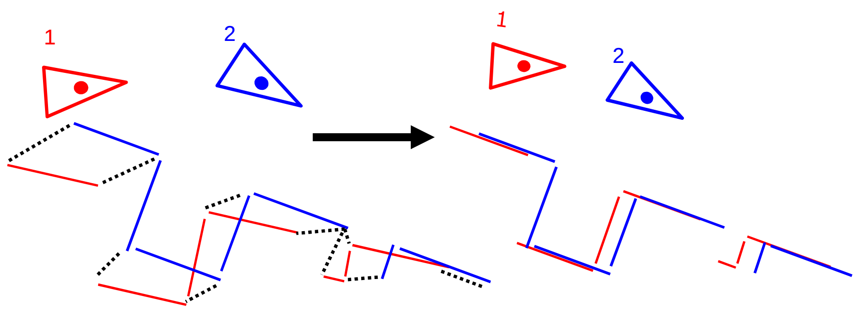
Iterated Closest Point

- For each point in scan 1
  - Find the *closest point* in scan 2 (how?)



Iterated Closest Point

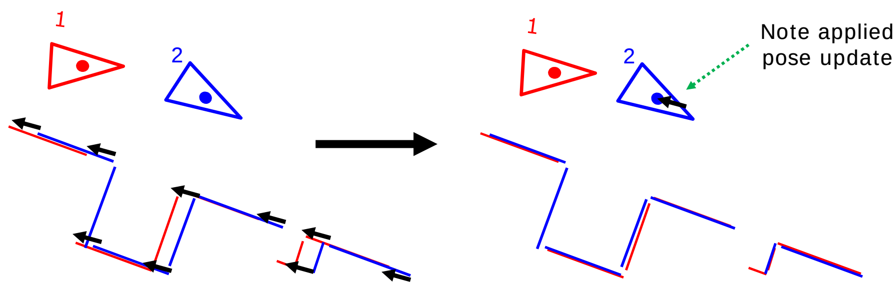
- Find the transformation that best aligns the matching sets of points



What happens to the estimate of the relative vehicle pose between sensor frames 1 & 2 ?

Iterated Closest Point

- ... Repeat until convergence



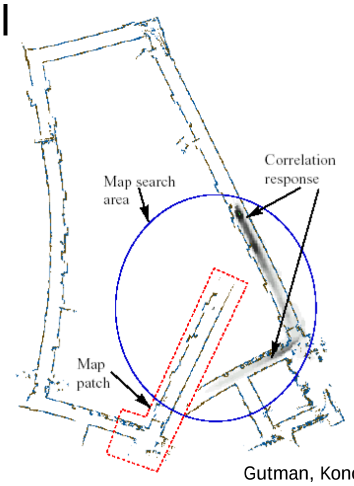
- Can do ICP across scans, across a scan and a (sub)map, or even across submaps!

Limitations / failure modes

- Computational cost (two scans of size *n*)
  - Naively, $O(n^2)$ plus cost of alignment step
- False minima
  - If ICP starts far from true alignment
  - If scans exhibit repeated local structure
- Bias
  - Anisotropic point sampling
  - Differing sensor fields of view (occlusion)
- Lots of research on improved ICP methods (see, e.g., Rusinkiewicz)

Loop Closing

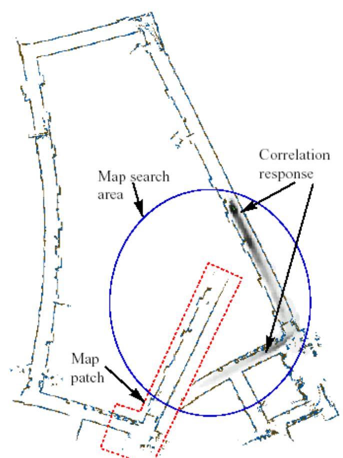
- ICP solves small-scale, short-duration alignment fairly well
- But now, consider:
  - Large scale
  - High uncertainty



Gutman, Konolige

Loop Closing

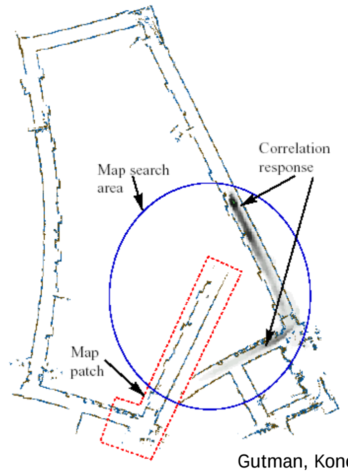
- Naive ICP ruled out:
  - Too CPU-intensive
- Assume we have a pose *uncertainty bound*
- This limits the portion of the existing map that must be searched
- Still have to face the problem of matching two partial scans that are far from aligned



Gutman, Konolige

Scan Matching Strategies

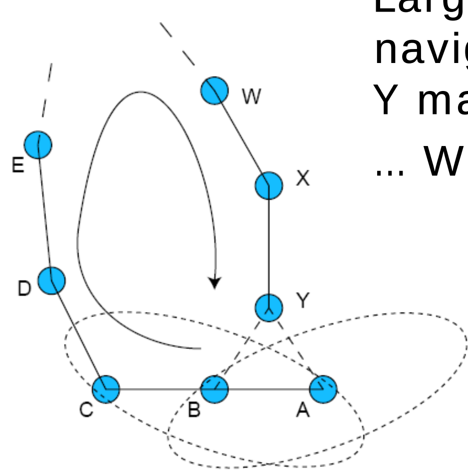
- Exhaustive search
  - Discretize robot poses
  - Find implied alignments
  - Assign score to each
  - Choose highest score
  - Pros, Cons?
- Randomized search
  - Choose minimal suff-icient match, at random
  - Align and score
  - Choose highest score
  - RANSAC (1981)
  - Pros, Cons?



Gutman, Konolige

Loop Closing Ambiguity

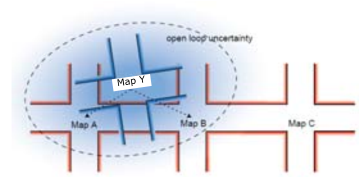
- Consider SLAM state after ABC … XY



Large open-loop navigation uncertainty

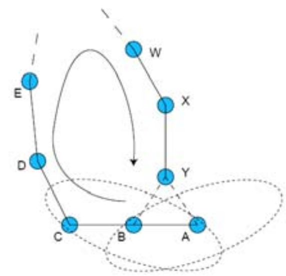
Y matches *both* A & B

… What to do?



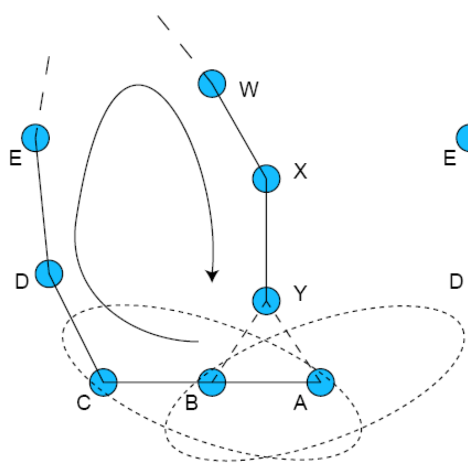
Loop Closing Choices

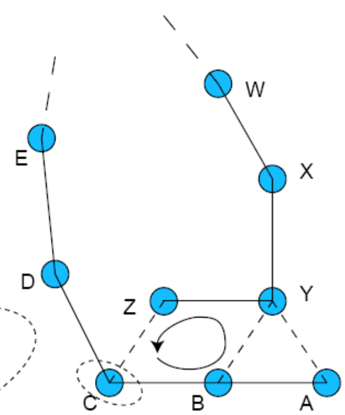
- Choose neither match
  - Pros, cons?
- Choose one match
  - Pros, cons?
- Choose both matches
  - Pros, cons?



Deferred Loop Validation

- Continue SLAM until Z matches C
- Examine graph for ~*identity cycle*





Some SLAM results

- See rvsn.csail.mit.edu group page

... But what's missing?

- Is topology enough?
- Are topology and geometry enough?
- ... What else is there?

Localization from a Prior Map

(Just the "L" part of SLAM)
The method shown here uses only a single Kinect

Method (Fallon et al.)

Expository Video

Summary

- SLAM is a hard robotics problem:
  - Requires sensor fusion over large areas
  - Scaling issues arise quickly with real data
- Key issue is managing *uncertainty*
  - At both low level and high level
  - Both continuous and discrete
- Saw several SLAM strategies
  - Local and global alignment
  - Randomization
  - Deferred validation
- SLAM is only part of the solution for most applications (need names, semantics)