Probabilistic Robotics: Localization & SLAM

METR 4202: **Robotics** & Automation

Dr Surya Singh -- Lecture # 11

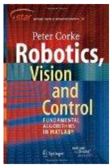
October 12, 2016

metr4202@itee.uq.edu.au
http://robotics.itee.uq.edu.au/~metr4202/ [http://**metr4202.com**]

Schedule of Events

Week	Date	Lecture (W: 12:05-1:50, 50-N202)
1	27-Jul	Introduction
2	3-Aug	Representing Position & Orientation & State (Frames, Transformation Matrices & Affine Transformations)
3	10-Aug	Robot Kinematics Review (& *Ekka Day*)
4	17-Aug	Robot Inverse Kinematics & Kinetics
5	24-Aug	Robot Dynamics (Jacobeans)
6	31-Aug	Robot Sensing: Perception & Linear Observers
7	7-Sep	Robot Sensing: Single View Geometry & Lines
8	14-Sep	Robot Sensing: Feature Detection
9	21-Sep	Robot Sensing: Multiple View Geometry
	28-Sep	*Study break*
10	5-Oct	Motion Planning
11	**12-Oct**	**Probabilistic Robotics: Localization & SLAM**
12	19-Oct	Probabilistic Robotics: Planning & Control
13	26-Oct	State-Space Automation (Shaping the Dynamic Response/LQR) + Course Review

Follow Along Reading:



Robotics, Vision & Control
by Peter Corke

Also online:SpringerLink

UQ Library eBook:
364220144X

Today

➔ SLAM ←

- SLAM
  - pp. 123-4 (§6.4-6.5)

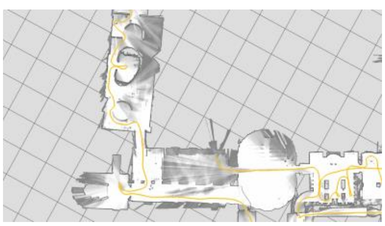
- Planning & Control
  - pp. ??

Next Time

Cool Robotics Share (It's Back!)

Cartographer

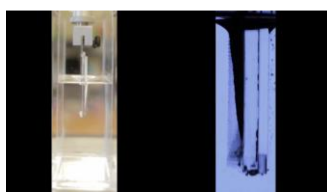
- Google Open Source SLAM



https://opensource.googleblog.com/2016/10/introducing-cartographer.html

Compliant Materials/Robotics

- Vision in ME Research

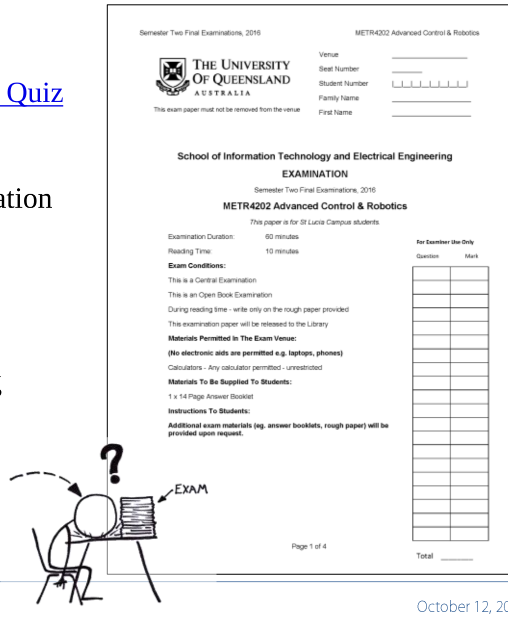


http://news.mit.edu/2016/beaver-inspired-wetsuits-surfers-1005

Final Exam!

- 4 Questions | 60 Minutes
- **Open Book**
- Similar in nature to the [2015 Quiz]

Topics:
- Position, orientation and location in space
- Robot analysis (forward/Inverse kinematics, recursive Newton-Euler formulations, etc.)
- Sensing geometry (including camera calibration)
- Multiple-view geometry
- Motion planning and control


Semester Two Final Examinations, 2016 METR4202 Advanced Control & Robotics

THE UNIVERSITY OF QUEENSLAND AUSTRALIA

This exam paper must not be removed from the venue

Venue
Seat Number
Student Number
Family Name
First Name

School of Information Technology and Electrical Engineering

EXAMINATION

Semester Two Final Examinations, 2016

METR4202 Advanced Control & Robotics

This paper is for St Lucia Campus students.

Examination Duration: 60 minutes

Reading Time: 10 minutes

Exam Conditions:

This is a Central Examination

This is an Open Book Examination

During reading time - write only on the rough paper provided

This examination paper will be released to the Library

Materials Permitted In The Exam Venue:

(No electronic aids are permitted e.g. laptops, phones)

Calculators - Any calculator permitted - unrestricted

Materials To Be Supplied To Students:

1 x 14 Page Answer Booklet

Instructions To Students:

Additional exam materials (eg. answer booklets, rough paper) will be provided upon request.

For Examiner Use Only

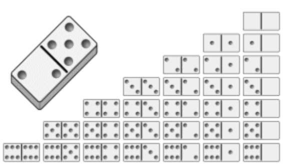
Question	Mark

Page 1 of 4

Total

Lab 3!: Sort | Play Domino

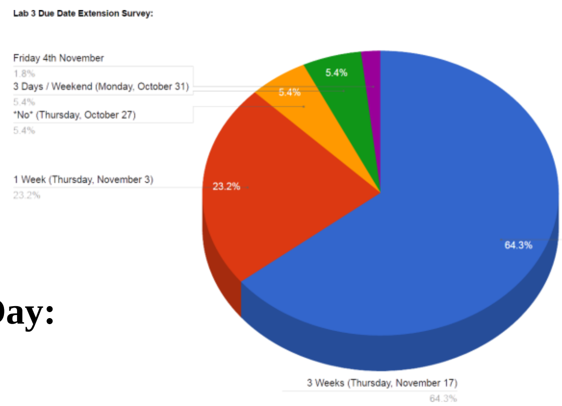
- Option 1: Sort Dominos
- Option 2: Play Dominos



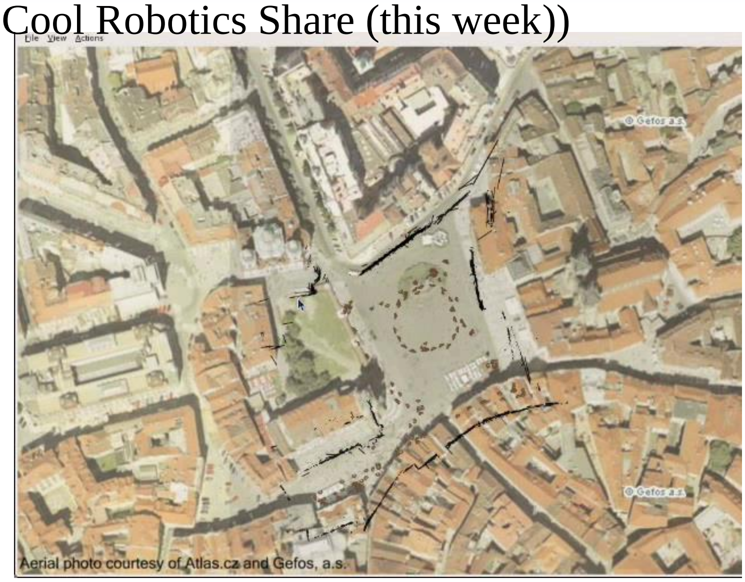
Source: https://upload.wikimedia.org/wikipedia/commons/0/04/Dominoes.jpg

Lab 3: Extension! [“Lab-ey McLabFace”]

- **Robot Grading**: November 3rd – 7th
- **Report:** November 17
- **Open-House/Demo Day:** November 21



SFM: Structure from Motion (& Cool Robotics Share (this week))



Structure [from] Motion

- Given a set of feature tracks,
  estimate the 3D structure and 3D (camera) motion.

- Assumption: orthographic projection

- Tracks: (u_{fp}, v_{fp}), f: frame, p: point
- Subtract out **mean** 2D position…

  $\mathbf{i}_f$: rotation, $\mathbf{s}_p$: position

$$u_{fp} = i_f^T s_p, v_{fp} = j_f^T s_p$$

From Szeliski, *Computer Vision: Algorithms and Applications*

Structure from motion

- How many points do we need to match?
- 2 frames:
  - (R,t): 5 dof + 3n point locations ≤
  - 4n point measurements ⇒
  - n ≥ 5
- k frames:
  - 6(k–1)-1 + 3n ≤ 2kn
- always want to use many more

$$\hat{u}_{ij} = f(\mathbf{K}, \mathbf{R}_j, \mathbf{t}_j, \mathbf{x}_i)$$
$$\hat{v}_{ij} = g(\mathbf{K}, \mathbf{R}_j, \mathbf{t}_j, \mathbf{x}_i)$$

From Szeliski, *Computer Vision: Algorithms and Applications*

Measurement equations

- Measurement equations

$u_{fp} = \boldsymbol{i}_f^T \boldsymbol{s}_p$ $\boldsymbol{i}_f$: rotation, $\boldsymbol{s}_p$: position

$v_{fp} = \boldsymbol{j}_f^T \boldsymbol{s}_p$

- Stack them up…

$\boldsymbol{W} = \boldsymbol{R}\,\boldsymbol{S}$

$\boldsymbol{R} = (\boldsymbol{i}_1,\ldots,\boldsymbol{i}_F, \boldsymbol{j}_1,\ldots,\boldsymbol{j}_F)^T$

$\boldsymbol{S} = (\boldsymbol{s}_1,\ldots,\boldsymbol{s}_P)$

From Szeliski, *Computer Vision: Algorithms and Applications*

Factorization

$\boldsymbol{W} = \boldsymbol{R}_{2F\times 3}\,\boldsymbol{S}_{3\times P}$

SVD

$\boldsymbol{W} = \boldsymbol{U}\,\boldsymbol{\Lambda}\,\boldsymbol{V}$ **Λ must be rank 3**

$\boldsymbol{W'} = (\boldsymbol{U}\,\boldsymbol{\Lambda}^{1/2})(\boldsymbol{\Lambda}^{1/2}\,\boldsymbol{V})$ $= \boldsymbol{U'}\,\boldsymbol{V'}$

Make $\boldsymbol{R}$ orthogonal

$\boldsymbol{R} = \boldsymbol{QU'}$, $\boldsymbol{S} = \boldsymbol{Q}^{-1}\boldsymbol{V'}$

$\mathbf{i}_f^T\mathbf{Q}^T\mathbf{Q}\mathbf{i}_f = 1 \ \ldots$

From Szeliski, *Computer Vision: Algorithms and Applications*

Results

- Look at paper figures…

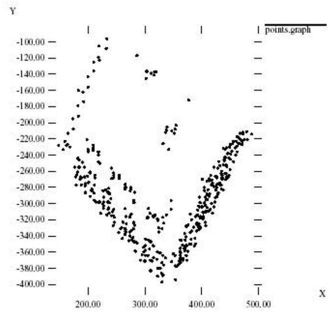


Figure 4.5: A view of the computed shape from approximately above the building (compare with figure 4.6).

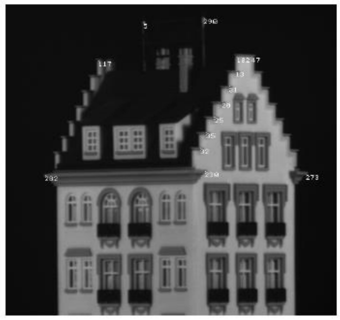


Figure 4.7: For a quantitative evaluation, distances between the features shown in the picture were measured on the actual model, and compared with the computed results. The comparison is shown in figure 4.8.

From Szeliski, *Computer Vision: Algorithms and Applications*

Bundle Adjustment

- What makes this non-linear minimization hard?
  - many more parameters: potentially slow
  - poorer conditioning (high correlation)
  - potentially lots of outliers
  - gauge (coordinate) freedom

$$\hat{u}_{ij} = f(\mathrm{K}, \mathrm{R}_j, \mathrm{t}_j, \mathrm{x}_i)$$
$$\hat{v}_{ij} = g(\mathrm{K}, \mathrm{R}_j, \mathrm{t}_j, \mathrm{x}_i)$$

From Szeliski, *Computer Vision: Algorithms and Applications*

More Cool Robotics Share!

SLAM!

(Better than SMAL! ☺)

What is SLAM?

- SLAM asks the following question:

  Is it possible for an autonomous vehicle to start at an unknown location in an unknown environment and then to incrementally build a map of this environment while simultaneously using this map to compute vehicle location?

- SLAM has many indoor, outdoor, in-air and underwater applications for both manned and autonomous vehicles.

- Examples
  - Explore and return to starting point (Newman)
  - Learn trained paths to different goal locations
  - Traverse a region with complete coverage (eg, mine fields, lawns, reef monitoring)
  - …

Components of SLAM

- Localisation
  - Determine pose given a priori map
- Mapping
  - Generate map when pose is accurately known from auxiliary source.
- SLAM
  - Define some arbitrary coordinate origin
  - Generate a map from on-board sensors
  - Compute pose from this map
  - Errors in map and in pose estimate are dependent.

SLAM: 30+ Year History!

Source: Leonard (MIT) Hartley and Zisserman, Cambridge University Press, p. 437

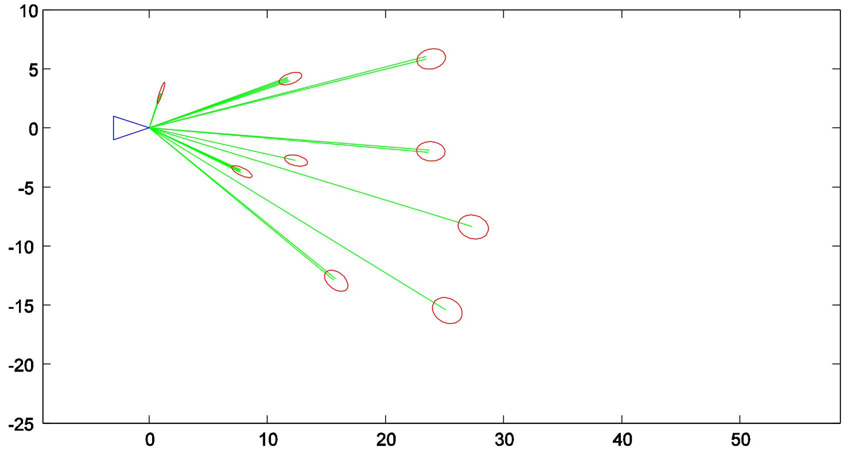
Jenkin Building Basement, Circa 1989

Source: Leonard (MIT)

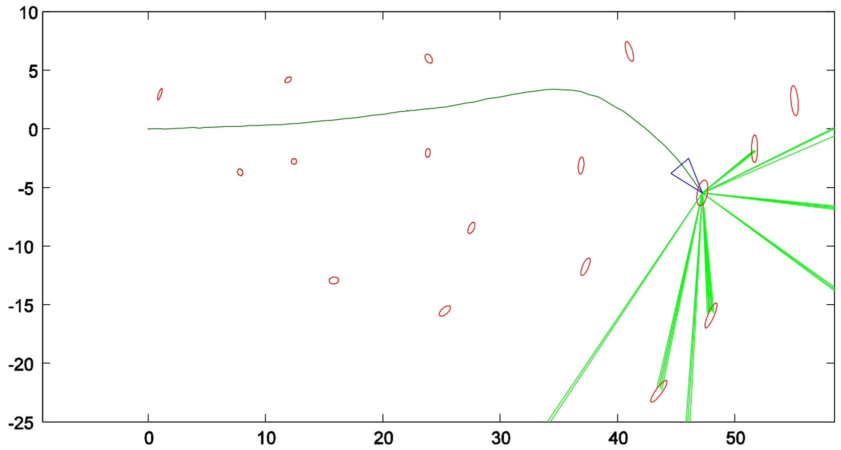
Basic SLAM Operation

Example: SLAM in Victoria Park

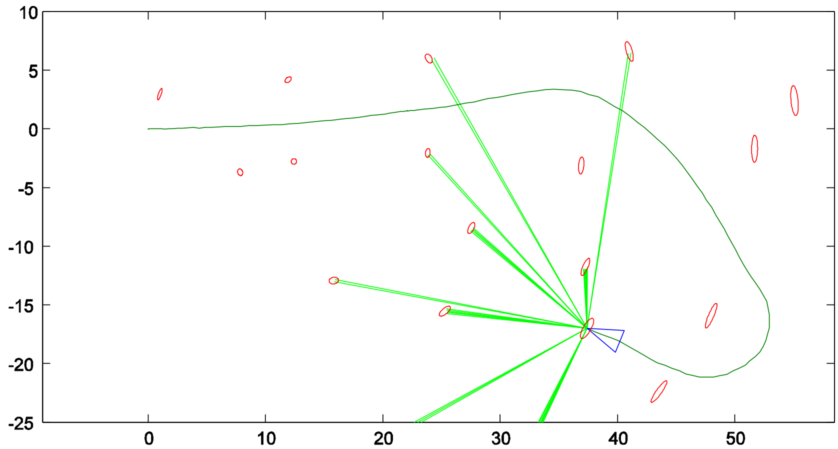
Basic SLAM Operation



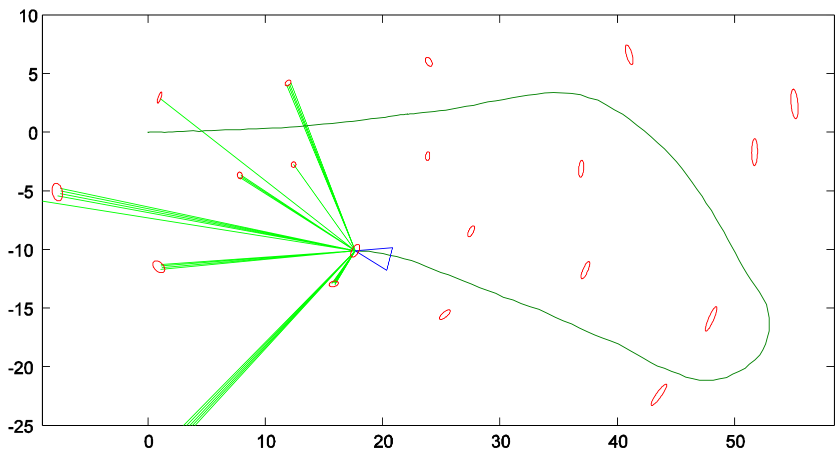
Basic SLAM Operation



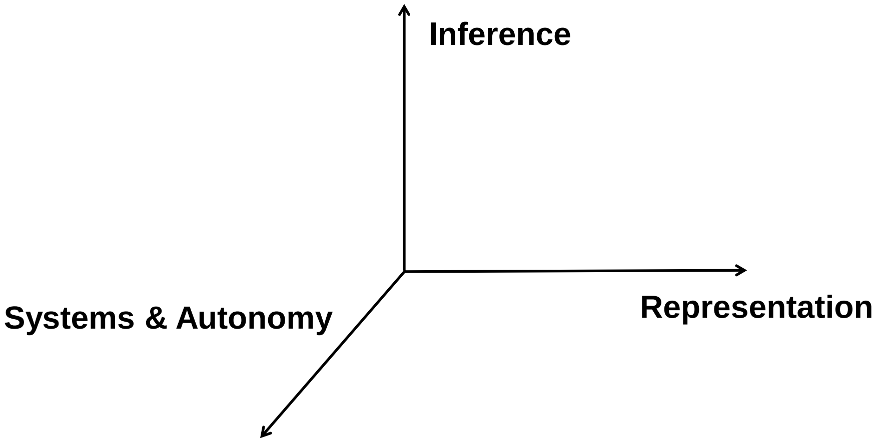
Basic SLAM Operation



Basic SLAM Operation

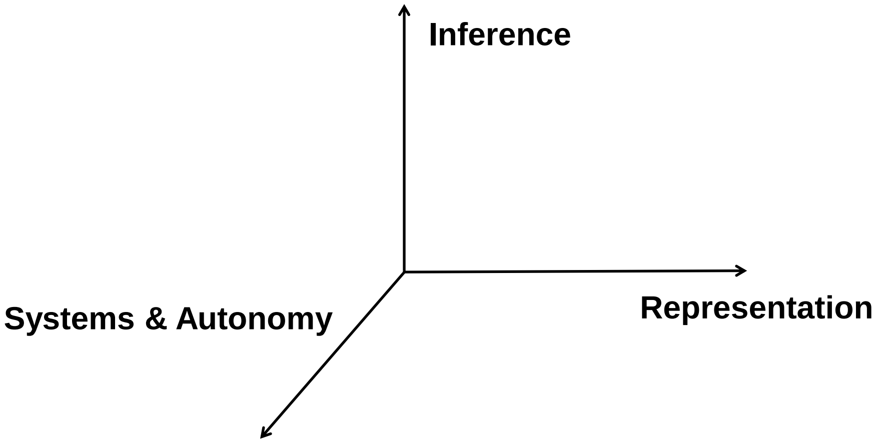


Why is SLAM Difficult?



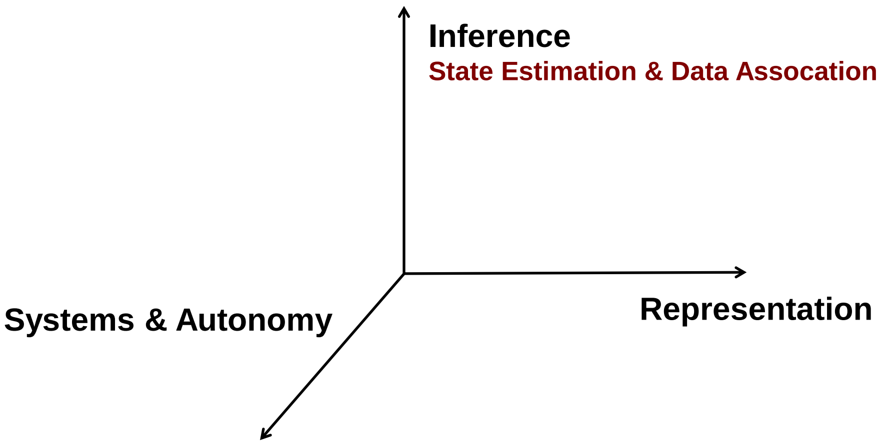
Source: Leonard (MIT)

Why is SLAM Difficult?



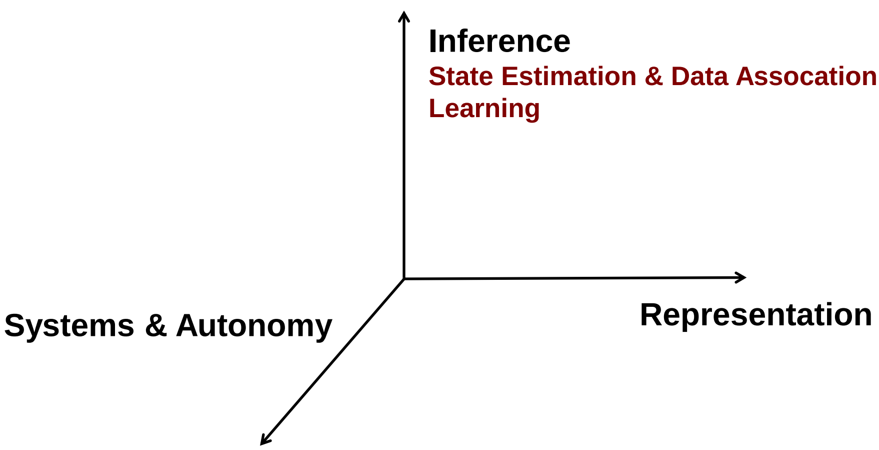
Source: Leonard (MIT)

Why is SLAM Difficult?



Source: Leonard (MIT)

Why is SLAM Difficult?



Source: Leonard (MIT)

Why is SLAM Difficult?

Inference
State Estimation & Data Assocation
Learning

Systems & Autonomy

Representation
Metric vs. Topological

Source: Leonard (MIT)

Why is SLAM Difficult?

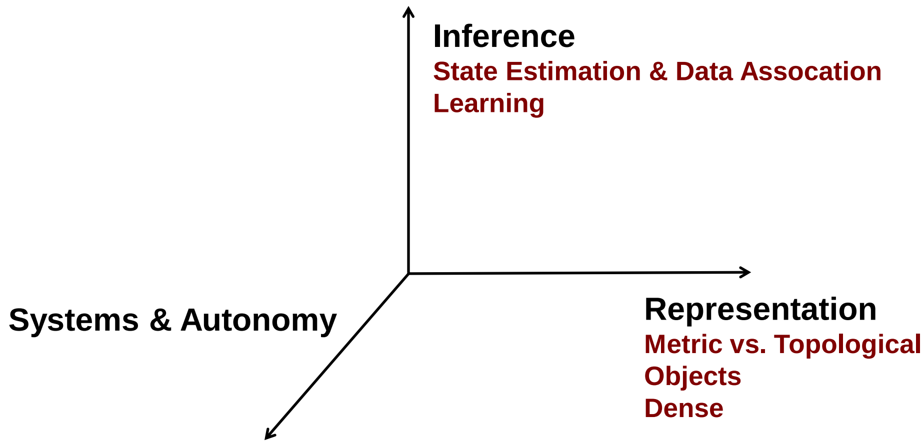
Inference
State Estimation & Data Assocation
Learning

Systems & Autonomy

Representation
Metric vs. Topological
Objects

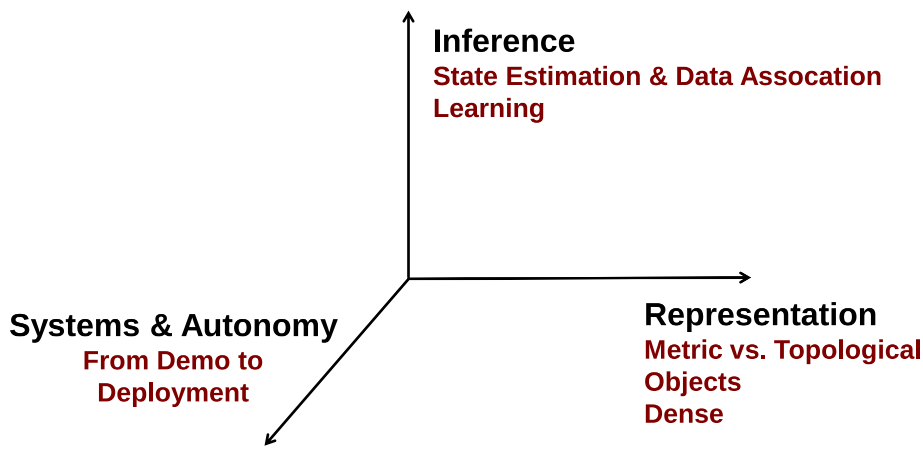
Source: Leonard (MIT)

Why is SLAM Difficult?



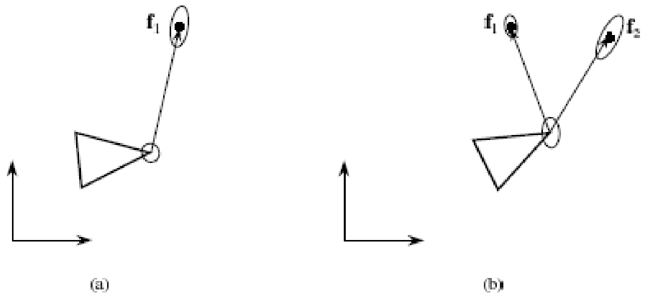
Source: Leonard (MIT)

Why is SLAM Difficult?

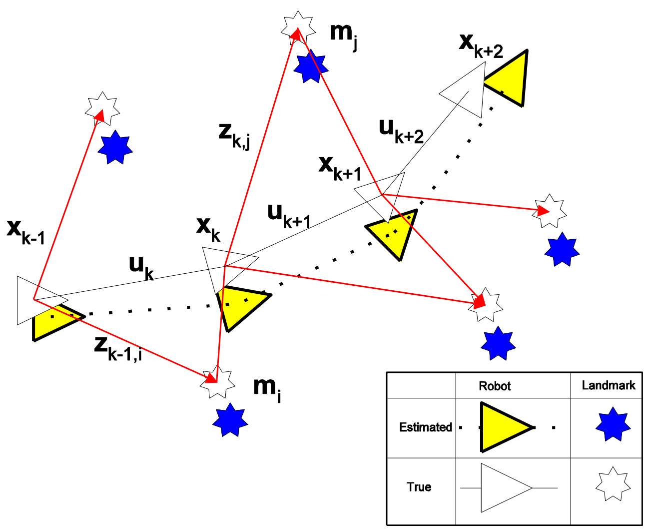


Source: Leonard (MIT)

Dependent Errors



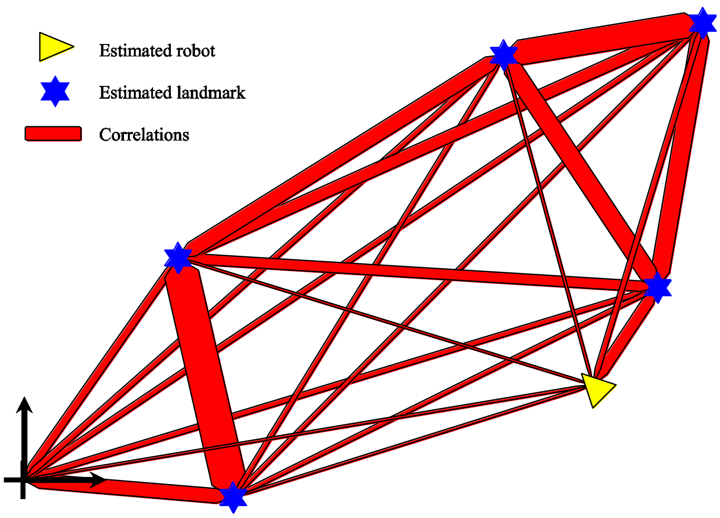
Correlated Estimates



SLAM Convergence

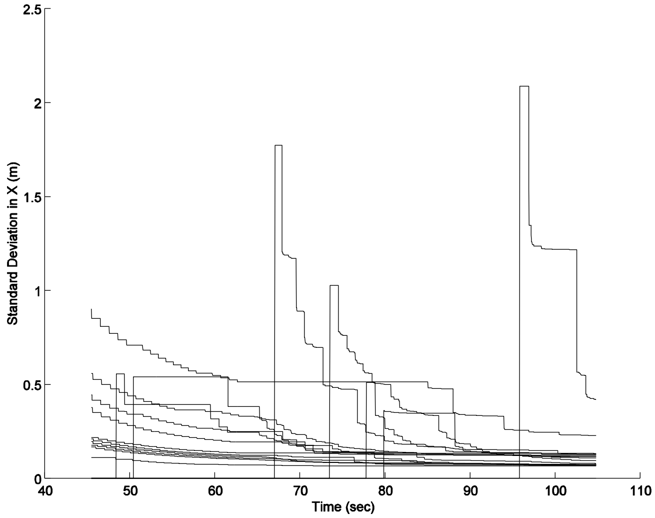
- An observation acts like a displacement to a spring system
  - Effect is greatest in a close neighbourhood
  - Effect on other landmarks diminishes with distance
  - Propagation depends on local stiffness (correlation) properties
- With each new observation the springs become increasingly (and monotonically) stiffer.
- In the limit, a rigid map of landmarks is obtained.
  - A perfect *relative* map of the environment
- The location accuracy of the robot is bounded by
  - The current quality of the map
  - The relative sensor measurement

Spring Analogy



Monotonic Convergence

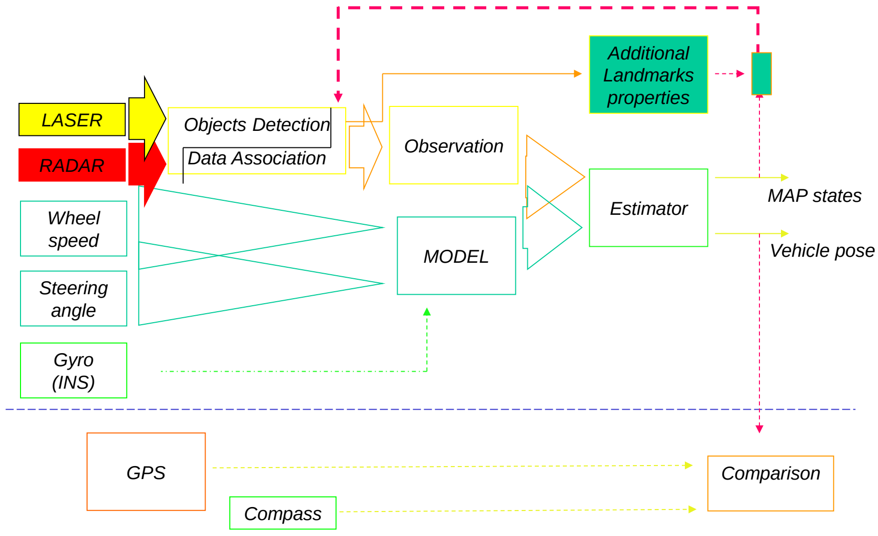
- With each new observation, the determinant decreases over the map and for any submatrix in the map.



Models

- Models are central to creating a representation of the world.
- Must have a mapping between sensed data (eg, laser, cameras, odometry) and the states of interest (eg, vehicle pose, stationary landmarks)
- Two essential model types:
  - Vehicle motion
  - Sensing of external objects

An Example System



States, Controls, Observations

Joint state with momentary pose

$$\mathbf{x}_k = \begin{bmatrix} \mathbf{x}_{v_k} \\ \mathbf{m}_1 \\ \vdots \\ \mathbf{m}_N \end{bmatrix}$$

Joint state with pose history

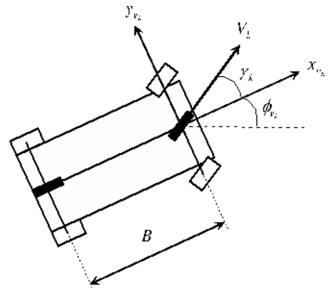
$$\mathbf{x}_k = \begin{bmatrix} \mathbf{x}_{v_k} \\ \mathbf{x}_{v_{k-1}} \\ \vdots \\ \mathbf{x}_{v_0} \\ \mathbf{m}_1 \\ \vdots \\ \mathbf{m}_N \end{bmatrix}$$

Control inputs: $\mathbf{U}_{0:k} = \{\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_k\} = \{\mathbf{U}_{0:k-1}, \mathbf{u}_k\}$

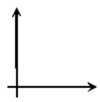
Observations: $\mathbf{Z}_{0:k} = \{\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_k\} = \{\mathbf{Z}_{0:k-1}, \mathbf{z}_k\}$

Vehicle Motion Model

- Ackerman steered vehicles: Bicycle model



- Discrete time model:



$$\mathbf{x}_{v_k} = \mathbf{f}_v\left(\mathbf{x}_{v_{k-1}}, \mathbf{u}_k\right) = \begin{bmatrix} x_{v_{k-1}} + V_k \Delta T \cos(\phi_{v_{k-1}} + \gamma_k) \\ y_{v_{k-1}} + V_k \Delta T \sin(\phi_{v_{k-1}} + \gamma_k) \\ \phi_{v_{k-1}} + \frac{V_k \Delta T}{B} \sin(\gamma_k) \end{bmatrix}$$

SLAM Motion Model

$$\mathbf{x}_{v_k} = \mathbf{f}_v\left(\mathbf{x}_{v_{k-1}}, \mathbf{u}_k\right) = \begin{bmatrix} x_{v_{k-1}} + V_k \Delta T \cos(\phi_{v_{k-1}} + \gamma_k) \\ y_{v_{k-1}} + V_k \Delta T \sin(\phi_{v_{k-1}} + \gamma_k) \\ \phi_{v_{k-1}} + \frac{V_k \Delta T}{B} \sin(\gamma_k) \end{bmatrix}$$

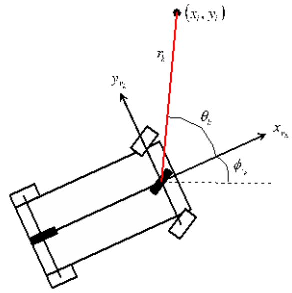
- Joint state: Landmarks are assumed stationary

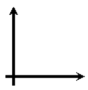
$$\mathbf{x}_k = \begin{bmatrix} \mathbf{f}_v(\mathbf{x}_{v_{k-1}}, \mathbf{u}_k) \\ \mathbf{m}_1 \\ \vdots \\ \mathbf{m}_N \end{bmatrix} \qquad \mathbf{x}_k = \begin{bmatrix} \mathbf{f}_v(\mathbf{x}_{v_{k-1}}, \mathbf{u}_k) \\ \mathbf{x}_{v_{k-1}} \\ \vdots \\ \mathbf{x}_{v_0} \\ \mathbf{m}_1 \\ \vdots \\ \mathbf{m}_N \end{bmatrix}$$

Observation Model

- Range-bearing measurement

$$\mathbf{z}_{i_k} = \mathbf{h}_i(\mathbf{x}_k) = \begin{bmatrix} \sqrt{(x_i - x_{v_k})^2 + (y_i - y_{v_k})^2} \\ \arctan \frac{y_i - y_{v_k}}{x_i - x_{v_k}} - \phi_{v_k} \end{bmatrix}$$





Applying Bayes to SLAM: Available Information

- States $\mathbf{X}_k$ (Hidden or inferred values)
  - Vehicle poses
  - Map; typically composed of discrete parts called landmarks or features
- Controls $\mathbf{U}_{0:k}$
  - Velocity
  - Steering angle
- Observations $\mathbf{Z}_{0:k}$
  - Range-bearing measurements

Augmentation: Adding new poses and landmarks

- Add new pose

$$\mathbf{x}_k = \begin{bmatrix} \mathbf{f}_v(\mathbf{x}_{v_{k-1}}, \mathbf{u}_k) \\ \mathbf{x}_{v_{k-1}} \\ \vdots \\ \mathbf{x}_{v_0} \\ \mathbf{m}_1 \\ \vdots \\ \mathbf{m}_N \end{bmatrix}$$

- Conditional probability is a Markov Model

$$\begin{aligned} p\left(\mathbf{x}_{v_k}|\mathbf{x}_{k-1}\right) &= \int p\left(\mathbf{x}_{v_k}|\mathbf{x}_{k-1}, \mathbf{u}_k\right) p\left(\mathbf{u}_k\right) d\mathbf{u}_k \\ &= \int \delta\left(\mathbf{x}_{v_k} - \mathbf{f}_v\left(\mathbf{x}_{v_{k-1}}, \mathbf{u}_k\right)\right) p\left(\mathbf{u}_k\right) d\mathbf{u}_k \\ &= p\left(\mathbf{x}_{v_k}|\mathbf{x}_{v_{k-1}}\right) \end{aligned}$$

Augmentation

$$\begin{aligned} p\left(\mathbf{x}_{v_k}|\mathbf{x}_{k-1}\right) &= \int p\left(\mathbf{x}_{v_k}|\mathbf{x}_{k-1}, \mathbf{u}_k\right) p\left(\mathbf{u}_k\right) d\mathbf{u}_k \\ &= \int \delta\left(\mathbf{x}_{v_k} - \mathbf{f}_v\left(\mathbf{x}_{v_{k-1}}, \mathbf{u}_k\right)\right) p\left(\mathbf{u}_k\right) d\mathbf{u}_k \\ &= p\left(\mathbf{x}_{v_k}|\mathbf{x}_{v_{k-1}}\right) \end{aligned}$$

- Product rule to create joint PDF $p(x_k)$

$$p\left(\mathbf{x}_{v_k}, \mathbf{x}_{k-1}\right) = p\left(\mathbf{x}_{v_k}|\mathbf{x}_{v_{k-1}}\right) p\left(\mathbf{x}_{v_{k-1}}, \ldots, \mathbf{x}_{v_0}, \mathbf{m}_1, \ldots, \mathbf{m}_N\right)$$

- Same method applies to adding new landmark states

Marginalisation:
Removing past poses and obsolete landmarks

- Augmenting with new pose and marginalising the old pose gives the classical SLAM prediction step

$$p\left(\mathbf{x}_{v_k}, \mathbf{m}_1, \ldots, \mathbf{m}_N\right) = \int p\left(\mathbf{x}_{v_k}, \mathbf{x}_{v_{k-1}}, \mathbf{m}_1, \ldots, \mathbf{m}_N\right) d\mathbf{x}_{v_{k-1}}$$

Fusion: Incorporating observation information

- Conditional PDF according to observation model

$$\begin{aligned} p\left(\mathbf{z}_{i_k}|\mathbf{x}_k\right) &= \int p\left(\mathbf{z}_{i_k}|\mathbf{x}_{v_k}, \mathbf{m}_i, \mathbf{r}_k\right) p\left(\mathbf{r}_k\right) d\mathbf{r}_k \\ &= \int \delta\left(\mathbf{z}_{i_k} - \mathbf{h}\left(\mathbf{x}_{v_k}, \mathbf{m}_i, \mathbf{r}_k\right)\right) p\left(\mathbf{r}_k\right) d\mathbf{r}_k \end{aligned}$$

- Bayes update:
  proportional to product of likelihood and prior

$$p\left(\mathbf{x}_k|\mathbf{Z}_{0:k}\right) = \frac{p\left(\mathbf{z}_{i_k} = \mathbf{z}_0|\mathbf{x}_k\right) p\left(\mathbf{x}_k|\mathbf{Z}_{0:k-1}\right)}{p\left(\mathbf{z}_{i_k} = \mathbf{z}_0\right)}$$

Implementing Probabilistic SLAM

- The problem is that Bayesian operations are intractable in general.
  - General equations are good for analytical derivations, not good for implementation
- We need approximations
  - Linearised Gaussian systems (EKF, UKF, EIF, SAM)
  - Monte Carlo sampling methods (Rao-Blackwellised particle filters)

EKF SLAM

- The complicated Bayesian equations for augmentation, marginalisation, and fusion have simple and efficient closed form solutions for linear Gaussian systems
- For non-linear systems, just linearise
  - EKF, EIF: Jacobians
  - UKF: use deterministic samples

Kalman Implementation

- So can we just plug the process and observation models into the standard EKF equations and turn the crank?

- Several additional issues:
  - Structure of the SLAM problem permits more efficient implementation than naïve EKF.
  - Data association.
  - Feature initialisation.

Structure of SLAM

- Key property of stochastic SLAM
  - Largely a *parameter* estimation problem

- Since the map is stationary
  - No process model, no process noise

- For Gaussian SLAM
  - Uncertainty in each landmark reduces monotonically after landmark initialisation
  - Map converges

- Examine computational consequences of this structure in next session.

Data Association

- Before the Update Stage we need to determine if the feature we are observing is:
  - An old feature
  - A new feature

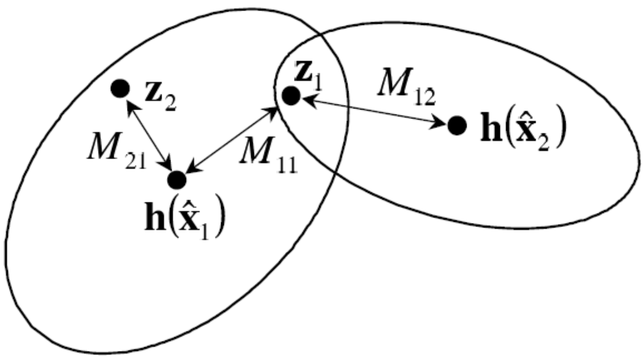
- If there is a match with only one known feature, the Update stage is run with this feature information.

$$\mu(k) = z(k) - h(\hat{x}(k/k-1))$$

$$S(k) = \nabla h_x(k) P(k/k-1) \nabla h_x^T(k) + R$$

$$\alpha = \mu^T(k) S^{-1}(k)\ \mu(k) < \chi^2_{0.95}$$

Validation Gating



New Features

- If there is no match then a potential new feature has been detected
- We do not want to incorporate a spurious observation as a new feature
  - It will not be observed again and will consume computational time and memory
  - It will add clutter, increasing risk of future mis-associations
  - The features are assumed to be static. We don't not want to accept dynamic objects as features: cars, people etc.

Acceptance of New Features

- **APPROACH 1**

  - Get the feature in a list of potential features
  - Incorporate the feature once it has been observed for a number of times
  - Advantages:
    - Simple to implement
    - Appropriate for High Frequency external sensor
  - Disadvantages:
    - Loss of information
    - Potentially a problem with sensor with small field of view: a feature may only be seen very few times

Acceptance of New Features

- **APPROACH 2**
  - The state vector is extended with past vehicle positions and the estimation of the cross-correlation between current and previous vehicle states is maintained. With this approach improved data association is possible by combining data form various points
    - J. J. Leonard and R. J. Rikoski. Incorporation of delayed decision making into stochastic mapping
    - Stephan Williams, PhD Thesis, 2001, University of Sydney
  - Advantages:
    - No Loss of Information
    - Well suited to low frequency external sensors (ratio between vehicle velocity and feature rate information)
    - Absolutely necessary for some sensor modalities (eg, range-only, bearing-only)
  - Disadvantages:
    - Cost of augmenting state with past poses
    - The implementation is more complicated

Incorporation of New Features

- **We have the vehicle states and previous map**

$$P_0 = \begin{bmatrix} P_{v,v}^0 & P_{v,m}^0 \\ P_{m,v}^0 & P_{m,m}^0 \end{bmatrix}$$

We observed a new feature and the covariance and cross-covariance terms need to be evaluated

$$P_1 = \begin{bmatrix} P_{v,v}^0 & P_{v,m}^0 & ? \\ P_{m,v}^0 & P_{m,m}^0 & ? \\ ? & ? & ? \end{bmatrix}$$

Incorporation of New Features

- Approach 1

$$P_0 = \begin{bmatrix} P^0_{vv} & P^0_{vm} & 0 \\ P^0_{mv} & P^0_{mm} & 0 \\ 0 & 0 & A \end{bmatrix}$$

With A very large

$$W(k) = P(k/k-1)\nabla h_x^T(k) S^{-1}(k)$$
$$S(k) = \nabla h_x(k) P(k/k-1) \nabla h_x^T(k) + R$$
$$P(k/k) = P(k/k-1) - W(k)S(k)W^T(k)$$

$$P_1 = \begin{bmatrix} P^1_{vv} & P^1_{vm} & P^1_{vn} \\ P^1_{mv} & P^1_{mm} & P^1_{mn} \\ P^1_{nv} & P^1_{nm} & P^1_{nn} \end{bmatrix}$$

- **Easy to understand and implement**
- **Very large values of A may introduce numerical problems**

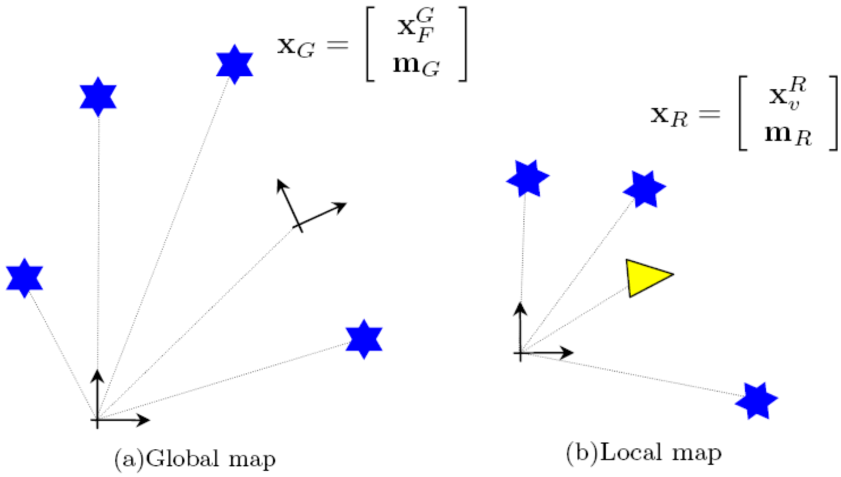
Analytical Approach

$$P_0 = \begin{bmatrix} P^0_{v,v} & P^0_{v,m} \\ P^0_{m,v} & P^0_{m,m} \end{bmatrix}$$

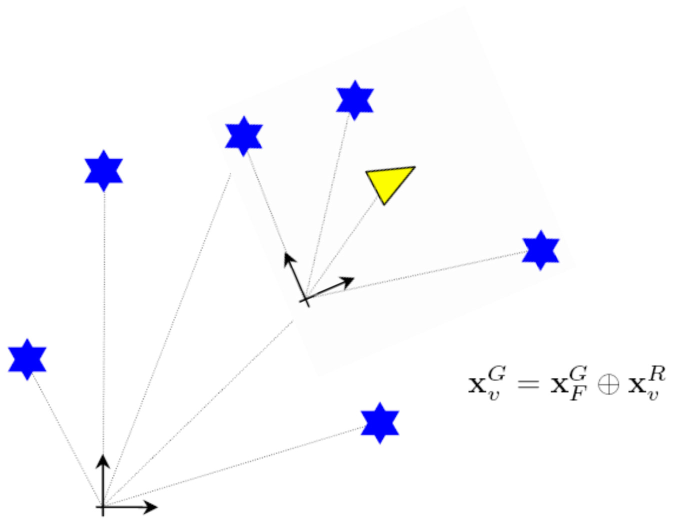
- **We can also evaluate the analytical expressions of the new terms**

$$P_1 = \begin{bmatrix} P^0_{v,v} & P^0_{v,m} & ? \\ P^0_{m,v} & P^0_{m,m} & ? \\ ? & ? & ? \end{bmatrix}$$

Constrained Local Submap Filter



CLSF Registration



CLSF Global Estimate