Principles of Robot Autonomy I

SLAM backend: Bundle adjustment and Pose graph optimization techniques

Bayes rule, probabilistic occupancy grid mapping

Stanford University

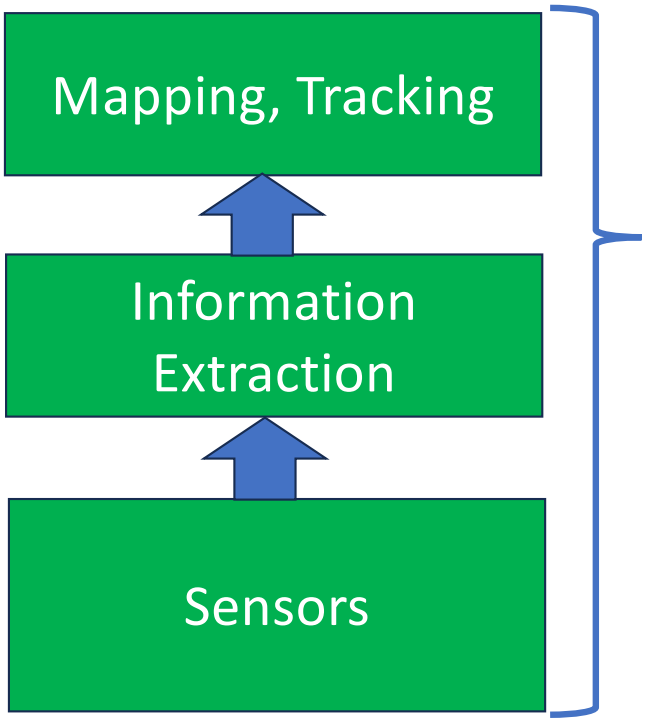
IPRL

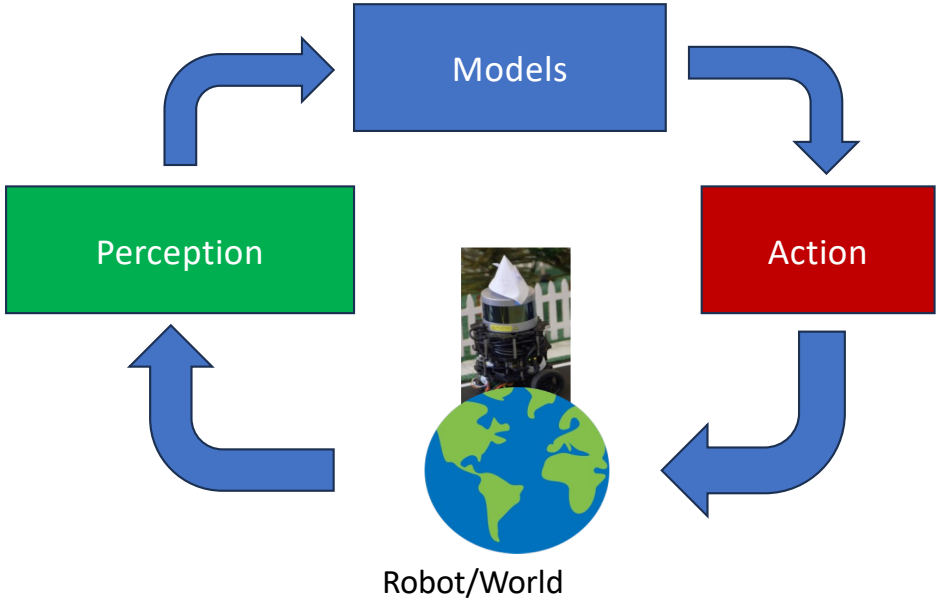
Logistics

- Homework 4: out Tues Nov 4 (After the midterm), due Thurs 11/13
- No labs: Wed–Fri this week (Midterm), and Mon–Tues next week (democracy day)
- No lecture: Tues, Nov 4 (Democracy day)
- Video lecture Thurs, Nov 6 (Mac traveling)
- Midterm window: Wed, Oct 29, 5pm – Sat, Nov 1
  - Take home, 72 hour window
  - Check out exam on gradescope
  - You will have personal 5 hour time slot
  - Open notes, book, HW solutions
  - No internet, no GenAI, no working with others
- Lecture 12:
  - SLAM Backend
  - Bundle adjustment, Pose graph optimization
  - Intro to Bayesian filtering, Prob. occupancy mapping

Robot Perception

Perception Stack



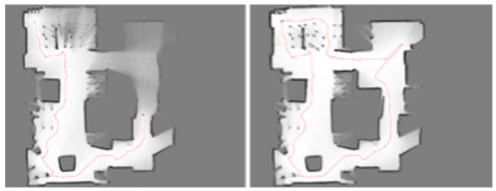


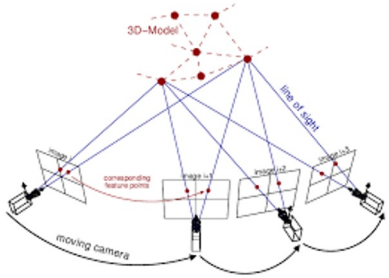
Perception Stack: Computer vision, filtering, SLAM

Use sensor data to update the models

Localization, Mapping, Tracking:

- EKF/Monte Carlo localization
- Occupancy grid mapping
- Factor graphs/SLAM
- Tracking (EKF and Particle Filter)
- **AA273: Filtering (Schwager)**
- **AA275: Navigation (Gao)**



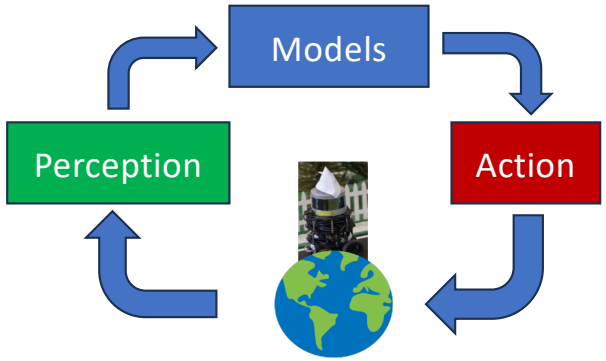


Information extraction

- Computer vision: features, correspondences, Structure from Motion (SfM), depth
- Lidar scan matching, ICP
- **CS231A: Comp Vision**

Sensors:

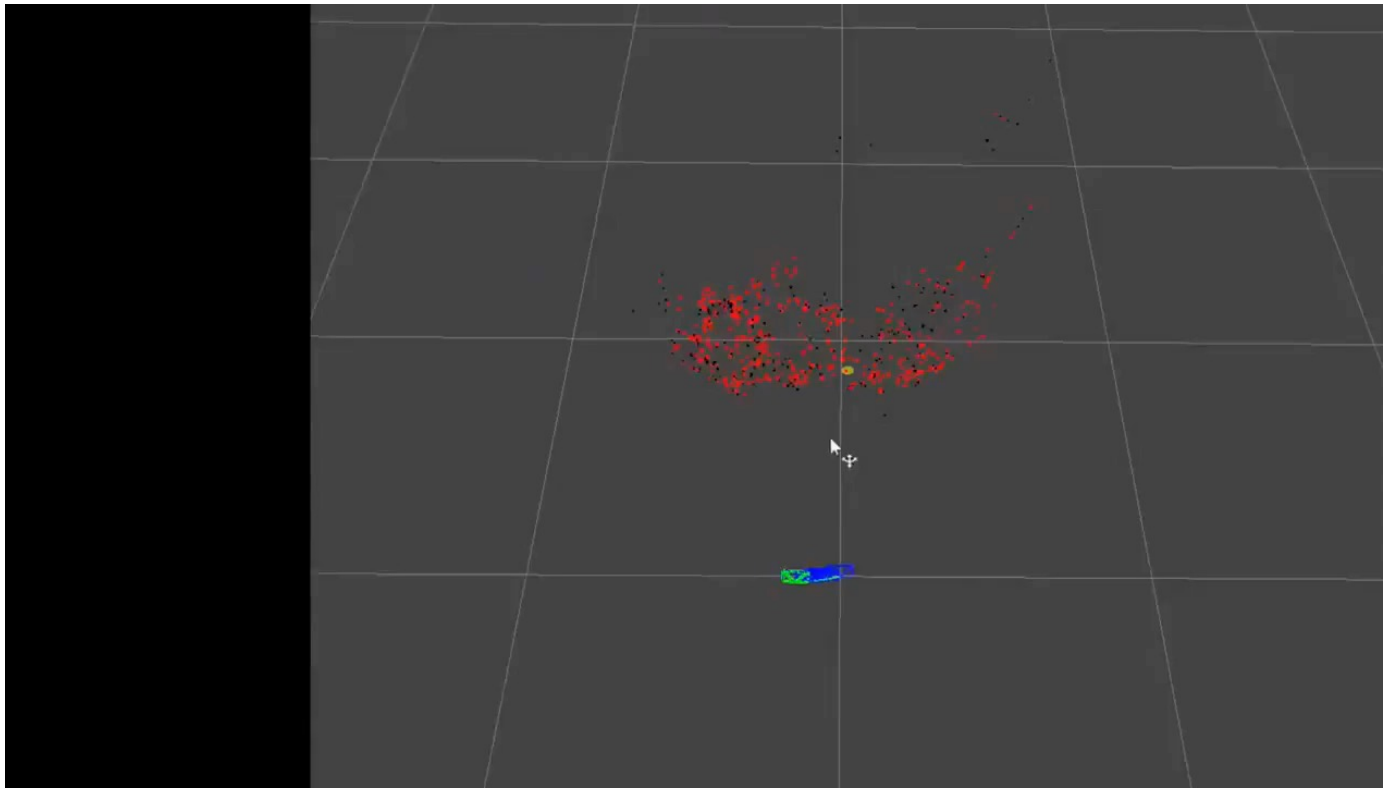
- RGB Cameras, RGB-D/stereo cameras, Lidar
- IMU, GPS, wheel encoders



NeRFBridge for SLAM with NeRFs

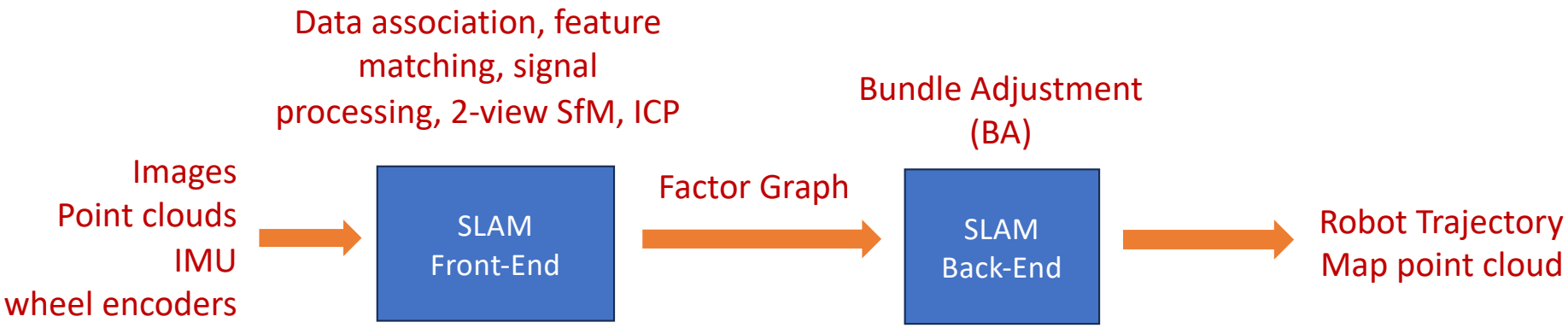
SLAM in ROS NAV2 toolbox

SLAM in Space for Asteroid Reconstruction



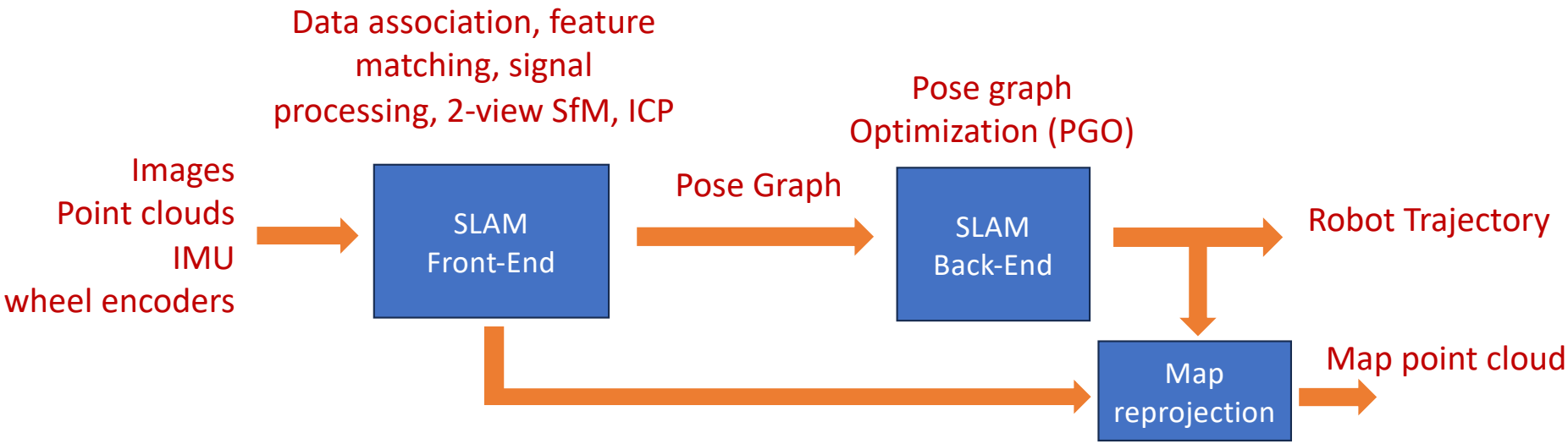
Factor Graphs: A Unifying Perspective for SLAM

(Frank Dellaert, 2012, 2017)



Bundle Adjustment SLAM - Produces both robot traj and map point cloud

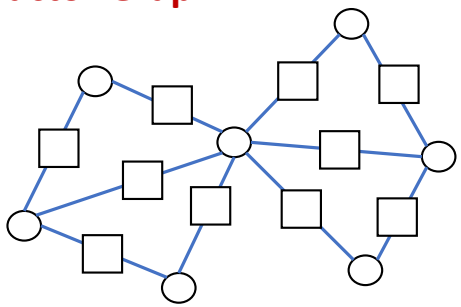
Factor Graphs: A Unifying Perspective for SLAM



Pose Graph SLAM - Produces robot traj first, map obtained in post processing.

Bundle Adjustment Optimization

Factor Graph:



Sum of all factors gives total cost:

Odometry factors (LIDAR, IMU/encoders)

$$J((R_i,\tau_i),q_k) = \sum_{(\hat{R}_{ij},\hat{\tau}_{ij})} (\lambda_{ij}^R \|R_i\hat{R}_{ij} - R_j\|^2 + \lambda_{ij}^\tau \|R_i\hat{\tau}_{ij} - (\tau_j - \tau_i)\|^2)$$

$$+ \sum_{\hat{q}_{ik}} \underbrace{\lambda_{ik}^D \|R_i\hat{q}_{ik} - (q_k - \tau_i)\|^2}_{\text{RGB-D factors}} + \sum_{(u_{ik},v_{ik})} \underbrace{\lambda_{ik}^V \|R_i K^{-1} p_{ik}^h - (q_k - \tau_i)\|^2}_{\text{RGB factors}}$$

Optimization Problem (Bundle Adjustment) :

$$\min_{(R_i,\tau_i),q_k} J((R_i,\tau_i),q_k)$$

$$\text{subject to} \quad \left.\begin{aligned} R_i^T R_i = I \\ \det(R_i) = 1 \end{aligned}\right\} \; SO(3)$$

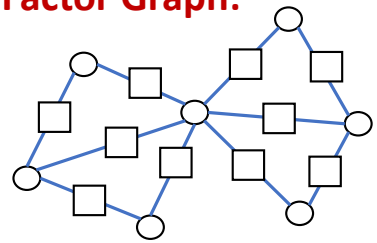
Interpretation: Find poses and feature locations that are *as consistent as possible* with all the measurements.

SLAM pose and map estimates:

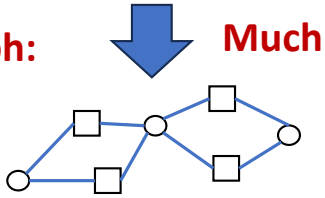
$$\{(R_i^*,\tau_i^*),q_k^*\} = \arg\min_{(R_i,\tau_i),q_k} J((R_i,\tau_i),q_k)$$

Pose Graph Optimization

Factor Graph:



Pose Graph:



Much simpler!

Interpretation: Still looking for poses *most consistent* with all relative pose measurements. Typically **much faster** than full bundle adjustment, but can be **less accurate**.

Pose Graph Cost Function:

Odometry factors (from all sensors)

$$J((R_i,\tau_i)_{i=1}^N) = \sum_{(\hat{R}_{ij},\hat{\tau}_{ij})} (\lambda_{ij}^R \|R_i\hat{R}_{ij} - R_j\|^2 + \lambda_{ij}^\tau \|R_i\hat{\tau}_{ij} - (\tau_j - \tau_i)\|^2)$$

Pose Graph Optimization (PGO):

$$\min_{(R_i,\tau_i)_{i=1}^N} J((R_i,\tau_i)_{i=1}^N)$$

$$\text{subject to} \quad R_i^T R_i = I$$

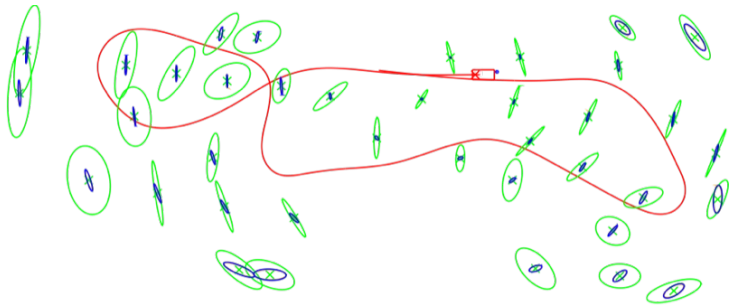
$$\det(R_i) = 1$$

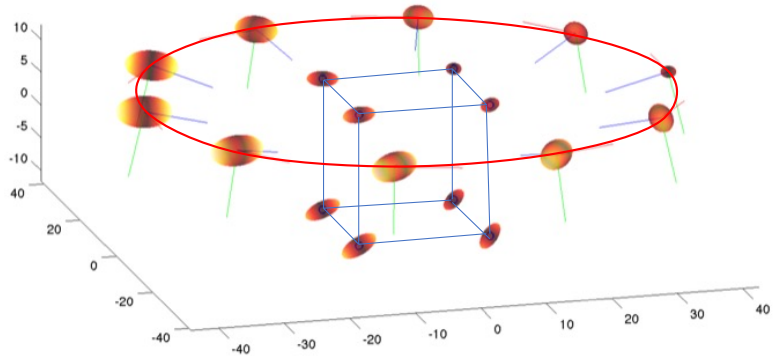
SLAM pose estimates only (no map!):

$$\{(R_i^*,\tau_i^*)_{i=1}^N\} = \arg\min_{(R_i,\tau_i)_{i=1}^N} J((R_i,\tau_i)_{i=1}^N)$$

GTSAM: The Standard Open-Source Factor Graph Optimizer

Frank Dellaert, and collaborators, GA Tech

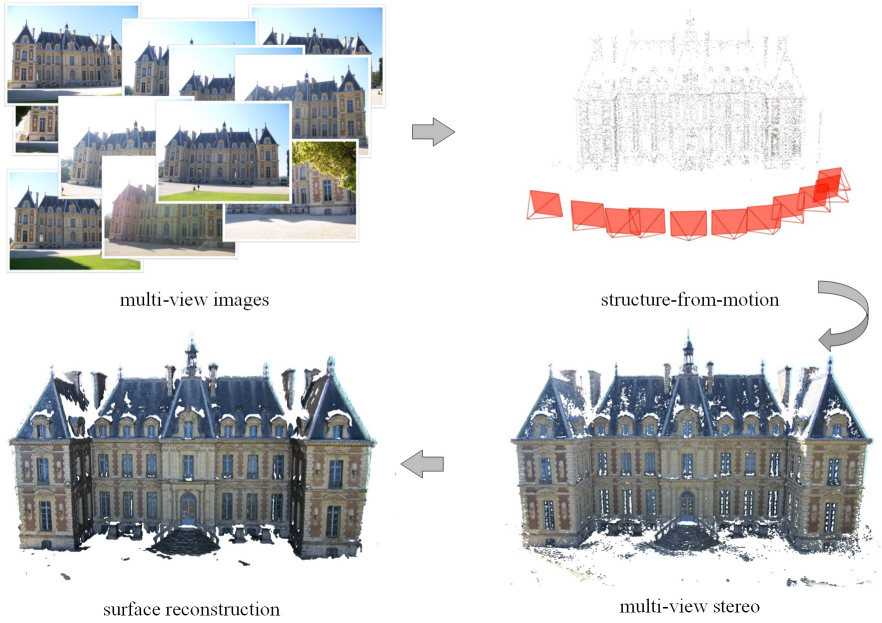




https://gtsam.org/

Colmap: The Standard Open-Source CV 3D Photogrammetry Package

Johannes Schönberger, and collaborators, ETH Zurich



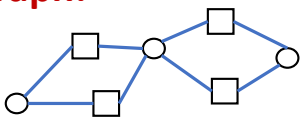
https://github.com/colmap/colmap

Obtaining the Map from PGO solution

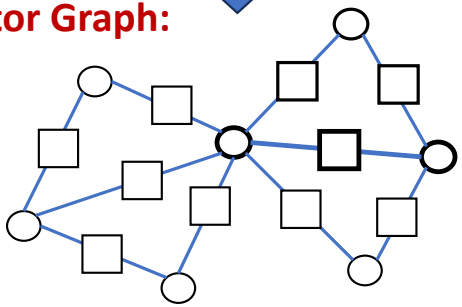
SLAM pose estimates:

$$\{(R_i^*, \tau_i^*)_{i=1}^N\} = \arg \min_{(R_i,\tau_i)_{i=1}^N} J((R_i, \tau_i)_{i=1}^N)$$

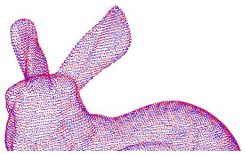
Pose Graph:

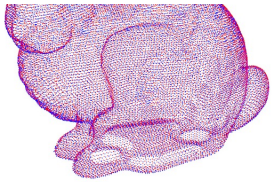


Factor Graph:



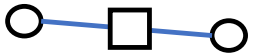
Lidar point clouds in local frames





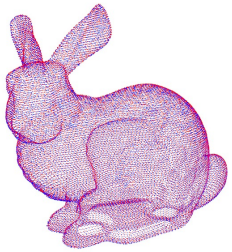
$P_i^* = R_i^* P_i + \tau_i^*$

$P_j^* = R_j^* P_j + \tau_j^*$



(R_i^*, τ_i^*) (R_j^*, τ_j^*)

Lidar point clouds in world frame



Fused point cloud in world frame

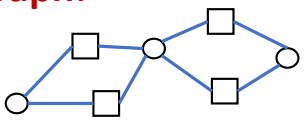
$\{P_i^* \cup P_j^*\}$

Obtaining the Map from PGO Solution

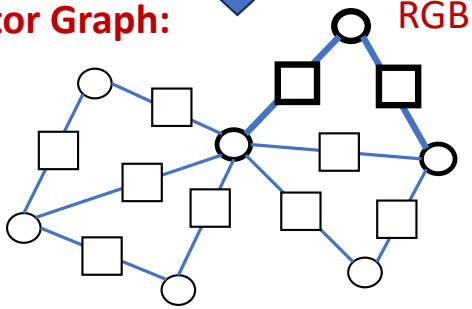
SLAM pose estimates:

$$\{(R_i^*, \tau_i^*)_{i=1}^N\} = \arg \min_{(R_i,\tau_i)_{i=1}^N} J((R_i, \tau_i)_{i=1}^N)$$

Pose Graph:

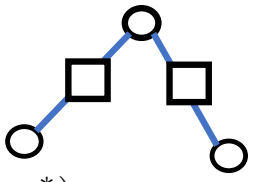


Factor Graph: RGB factors



Point in point cloud

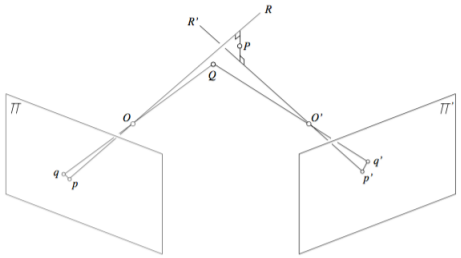
q_k^* unknown



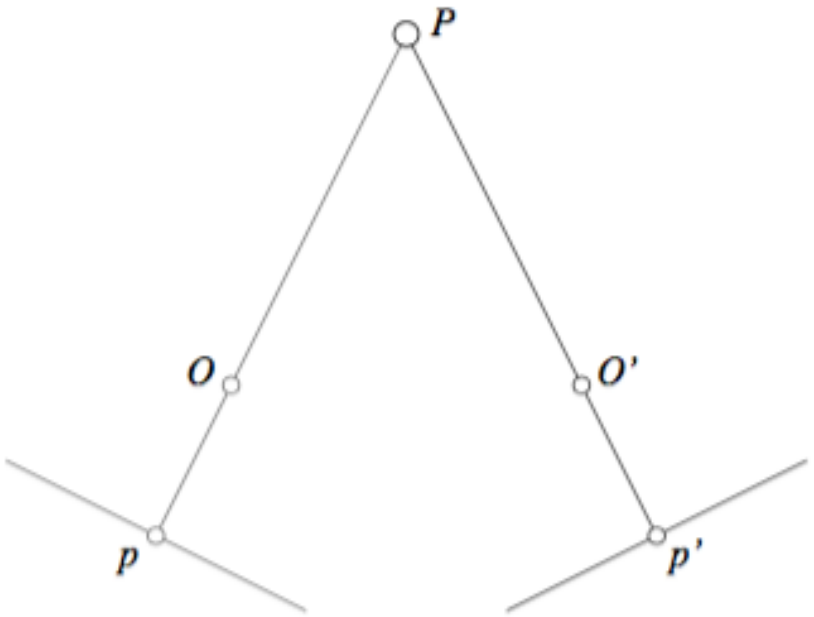
(R_i^*, τ_i^*) (R_j^*, τ_j^*)

known

Stereo Depth: find location of point in 3D from two RGB images with known poses.

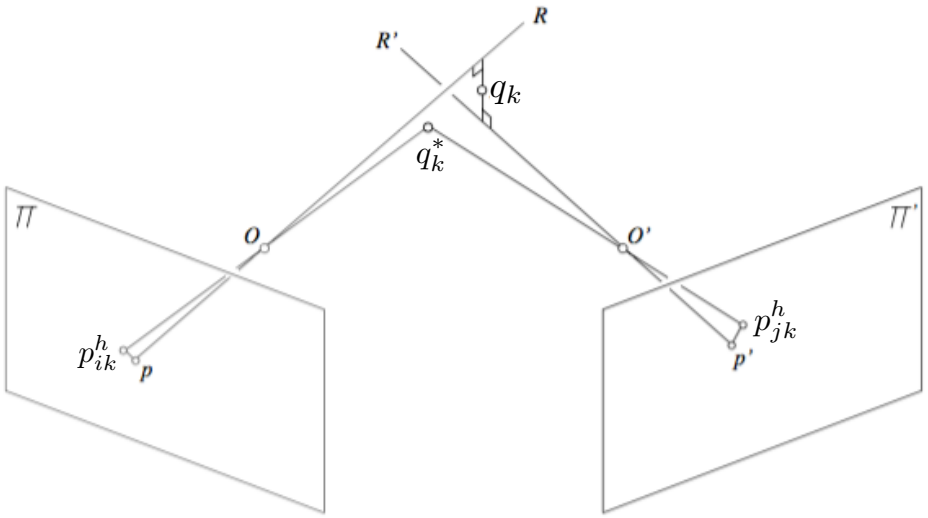


Binocular reconstruction



- Given: *calibrated* camera with known poses and two image matching points p and p'
- Find corresponding scene point by intersecting the two rays $\overline{Op}$ and $\overline{O'p'}$ (process known as triangulation)

Approximate triangulation



Camera calibration matrix

Known camera pose

RGB measurement:

$$p_{ik}^h = K R_i^T (q_k - \tau_i) + \nu_{ik}$$

$$p_{ik}^h = (u_{ik}, v_{ik}, 1) z_{ik}^h$$

Homogeneous pixel coords

Measured pixel coords in image

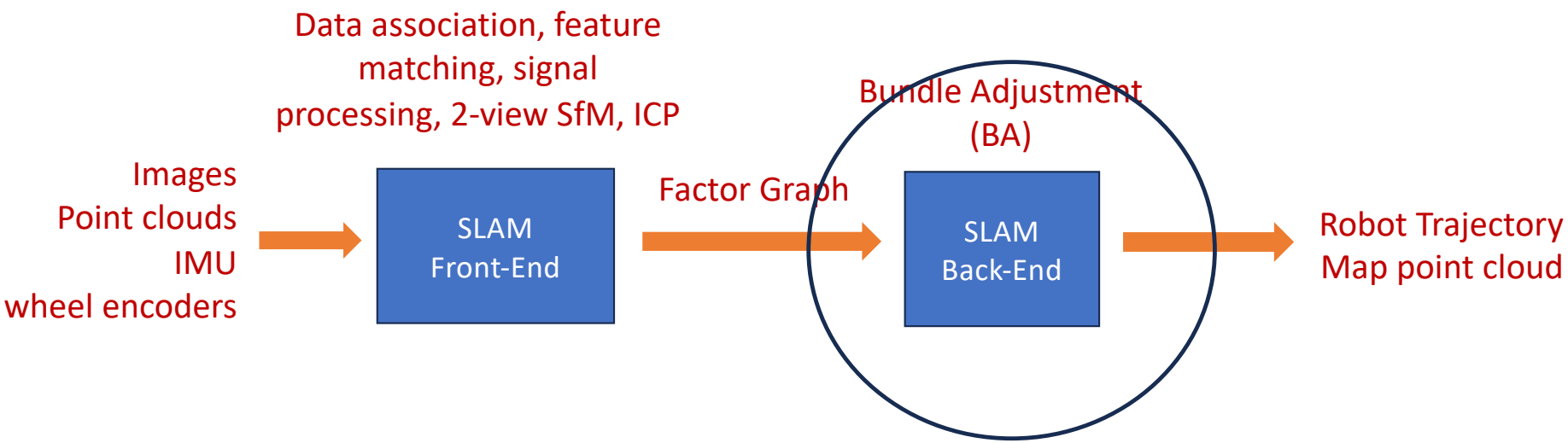
Unknown pixel depth

$$q_k^* = \arg\min_{q_k} \|R_i K^{-1} p_{ik}^h - (q_k - \tau_i)\|^2 + \|R_j K^{-1} p_{jk}^h - (q_k - \tau_j)\|^2$$

$$\text{subject to} \quad p_{ik}^h = (u_{ik}, v_{ik}, 1) z_{ik}^h$$

Factor Graphs: A Unifying Perspective for SLAM

(Frank Dellaert, 2012, 2017)



Bundle Adjustment SLAM - Produces both robot traj and map point cloud

Optimization techniques (e.g., iSAM, colmap)

Optimization Problem:

$$\min_{(R_i,\tau_i)_{i=1}^N,(q_k)_{k=1}^M} J((R_i,\tau_i)_{i=1}^N,(q_k)_{k=1}^M)$$

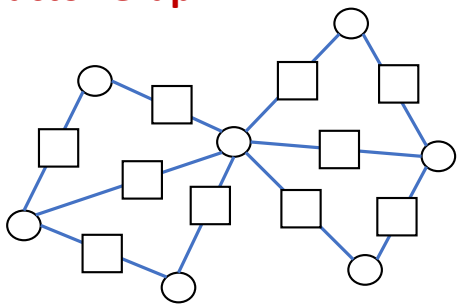
$$\text{subject to} \quad R_i^T R_i = I$$

$$\det(R_i) = 1$$

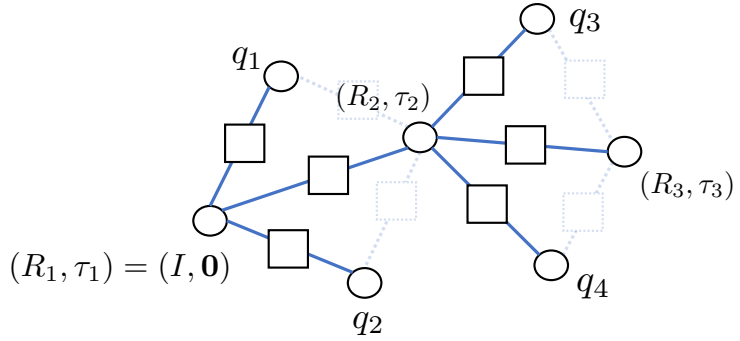
Large, non-convex, constrained problem.
In practice, many local minima.

Initialization: most optimization solvers require a rough initialization to converge to reasonable solution.

Factor Graph:



Initialization with min spanning tree:

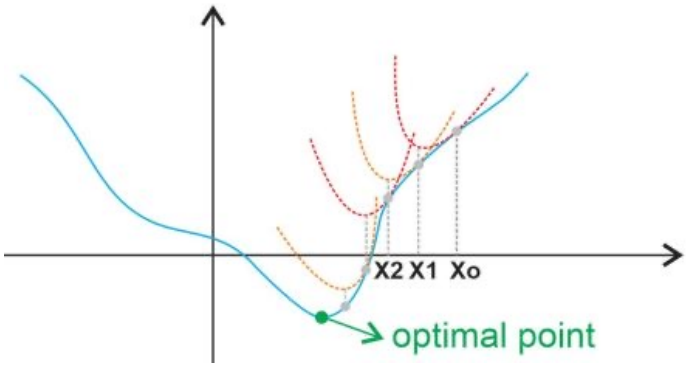


Optimization techniques (e.g., iSAM, colmap)

Optimization Problem:

$$\min_{(R_i,\tau_i)_{i=1}^N,(q_k)_{k=1}^M} J((R_i,\tau_i)_{i=1}^N,(q_k)_{k=1}^M)$$

$$\text{subject to} \quad R_i^T R_i = I$$
$$\det(R_i) = 1$$



Gauss-Newton Optimization

$$J = \sum_{(i,j)} f(T_i,T_j) = \sum_{(i,j)} \|l_{ij}(T_i,T_j)\|^2 \approx \sum_{(i,j)} \|l(T_i^0,T_j^0) + \nabla l(T_i^0,T_j^0)((T_i - T_j^0),(T_j - T_j^0))\|^2$$

Nonlinear least squares!

Linear least squares!

Taylor linearization about (T_i^0, T_j^0)

$$(T_i^1)_{i=1}^N = \arg\min \sum_{(i,j)} \|l(T_i,T_j) + \nabla l(T_i^0,T_j^0)((T_i - T_i^0),(T_j - T_j^0))\|^2$$

Iterate until convergence

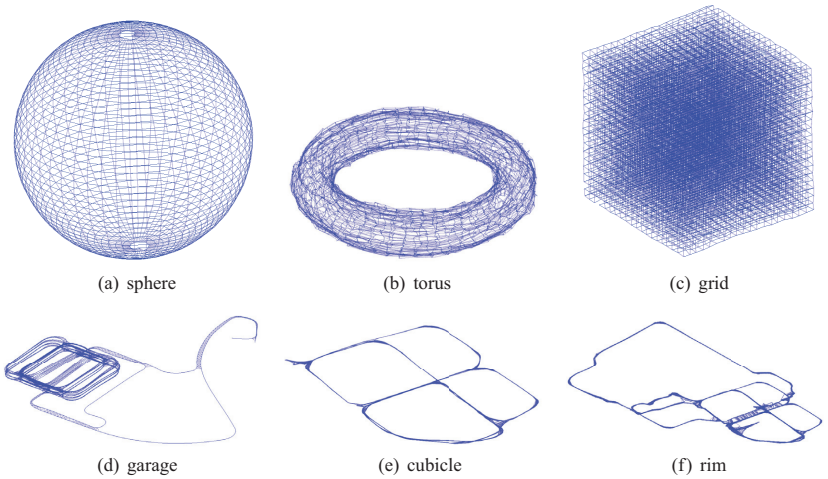
Levenberg-Marquart Optimization:

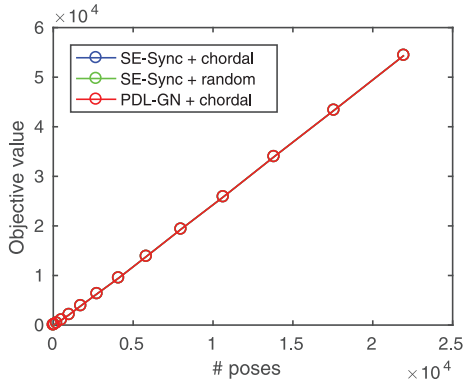
Combination of Gauss-Newton and Grad descent

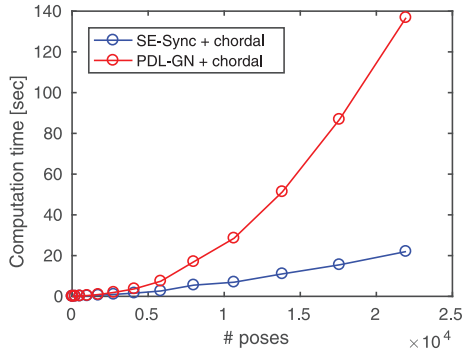
SE-Sync: Efficient Certifiable Solver for PGO

Rosen et al, IJRR 2018

Summary: Uses Riemannian optimization, duality, and semi-definite programming to solve PGO with suboptimality bound







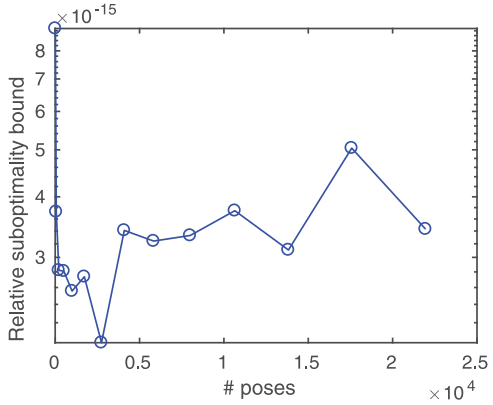
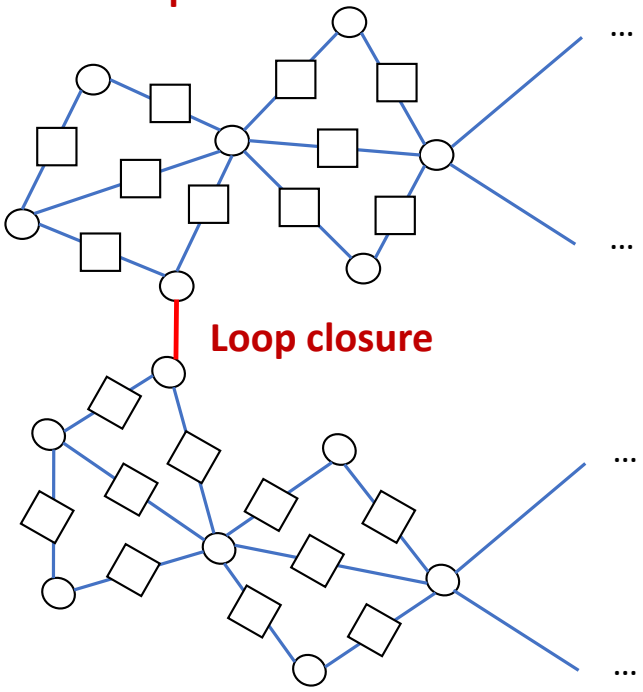
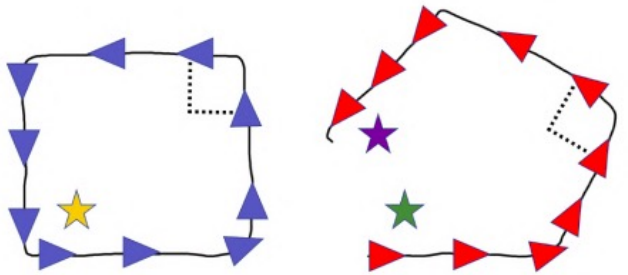


Fig. 5. Globally optimal solutions for the 3D SLAM benchmark datasets listed in Table 2.

Loop Closure

Factor Graph:

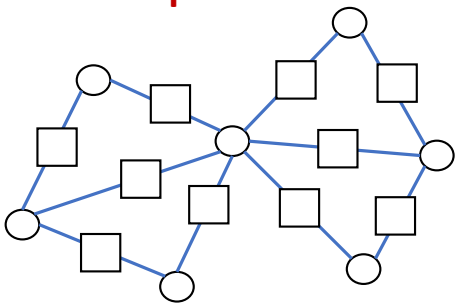


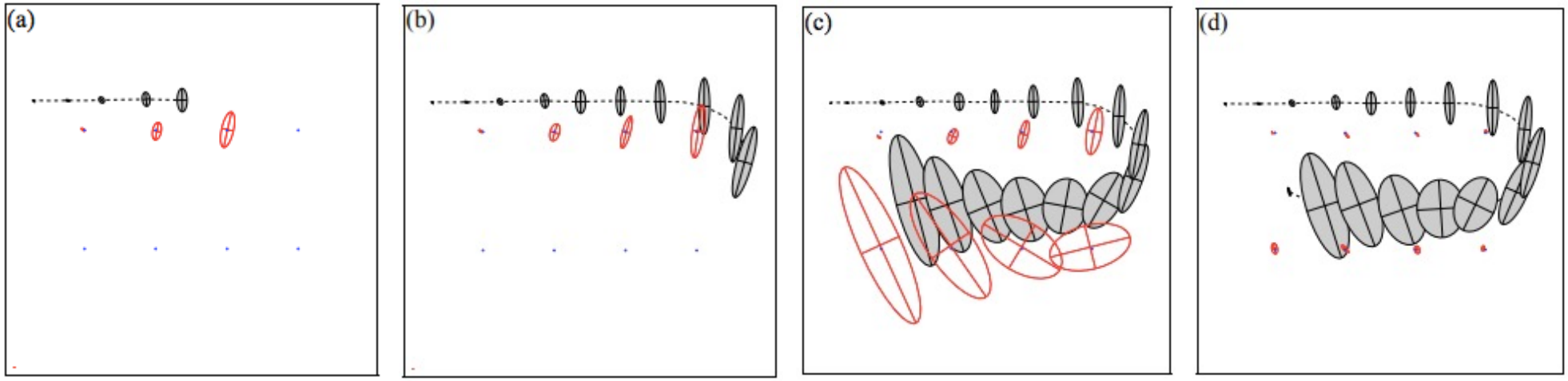


- **Long time separation**
- **Close spatial separation**
- **Requires place recognition or special loop closure module**

Loop Closure

Factor Graph:





Online vs Offline

- Online SLAM problem: estimate the posterior over the momentary pose along with the map

$$p(x_t, m \mid z_{1:t}, u_{1:t})$$

- Full SLAM problem: estimate posterior over the entire path along with the map

$$p(x_{1:t}, m \mid z_{1:t}, u_{1:t})$$

- Usually re-solve full SLAM with “warm start” iteratively online

Bayesian Filtering, Probabilistic Occupancy Grid Mapping

- Random variables
- Bayes' rule
- Stochastic models

Recursive State Estimation – Bayesian Estimation and Filtering

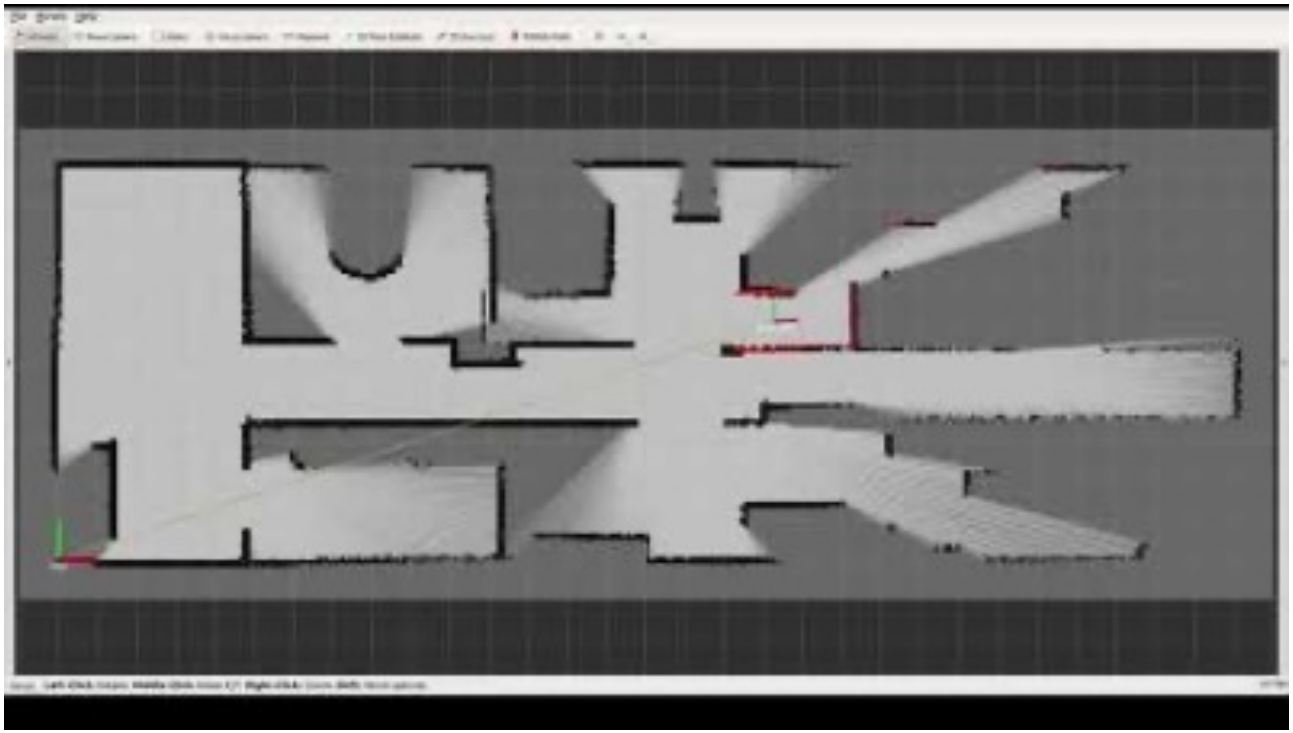
Mathematical Formalism to:

- continuously integrate measurements
- from different sensor sources
- to infer the state of a latent variable

To obtain high-level world or robot state information

- Mapping - where are the obstacles
- Localization - where is my robot
- Tracking - where are the other dynamic agents in the scene
- SLAM - joint mapping and localization

Example: Occupancy grid mapping



Probabilistic occupancy grid mapping

Occupancy map: $m \in \{0,1\}^{M \times N}$

Single grid cell: $m_{i,j} \in \{0,1\}$

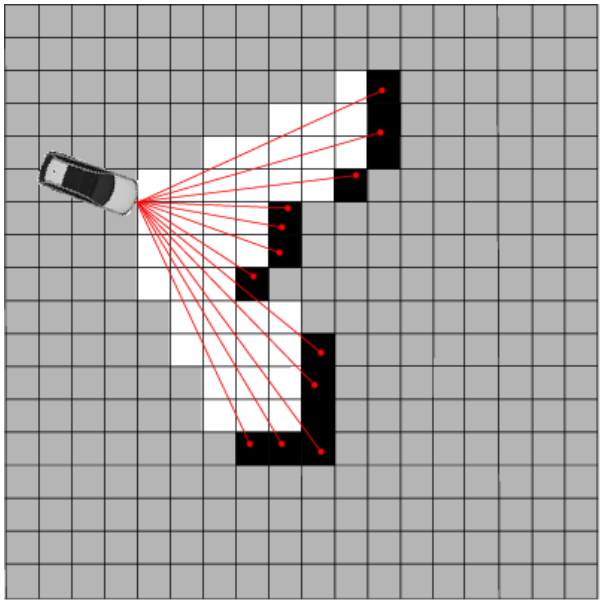
Measurement model (Lidar, stereo, vision):

See point in cell — Cell occupied

$$p(z_t(i,j) \mid m(i,j)) = \begin{bmatrix} p(0 \mid 0) & p(0 \mid 1) \\ p(1 \mid 0) & p(1 \mid 1) \end{bmatrix}$$

True negative — False negative

False positive — True positive



Cell (i,j) only gets a measurement if it is inside the footprint of sensor

Footprint depends on robot pose (assumed known)

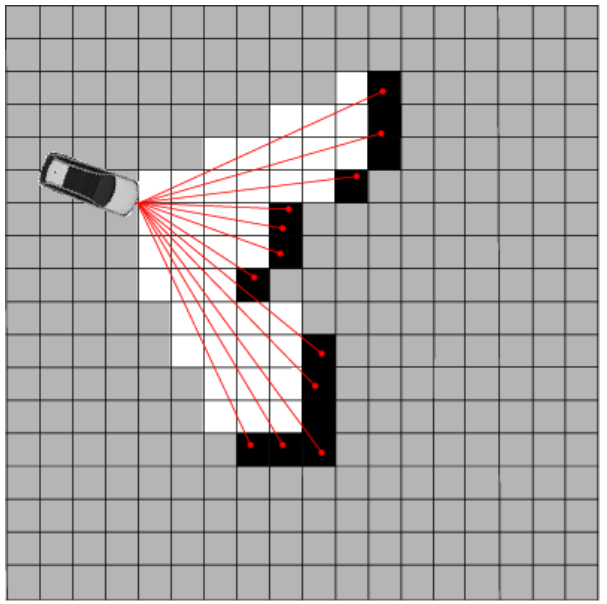
Probabilistic occupancy grid mapping

E.g.

$$p(z(i,j) \mid m(i,j)) = \begin{bmatrix} 0.9 & 0.05 \\ 0.1 & 0.95 \end{bmatrix}$$

True negative — False negative

False positive — True positive



Goal:

Find probability distribution over each occupancy cell given sequence of measurements

$$p(m(i,j) \mid z_{1:t}(i,j))$$

Sequence of measurements from 1 to t

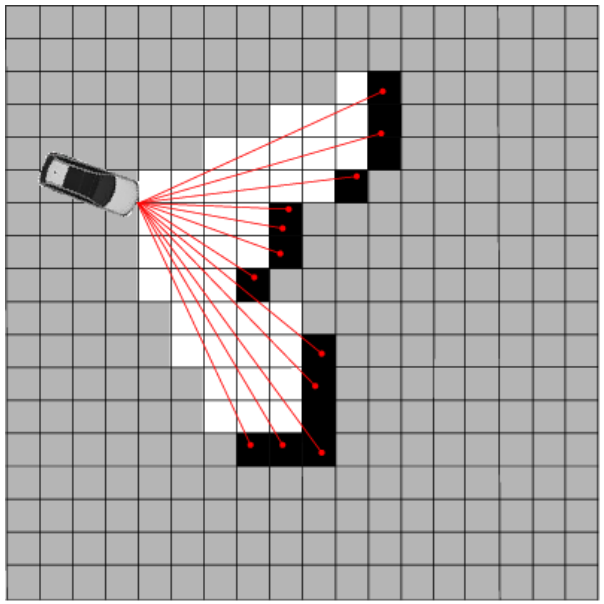
Probabilistic occupancy grid mapping

Recall:

$$p(m(i,j) = 1 \mid z_{1:t}(i,j)) \in [0,1]$$

Called a binary random variable,
or a Bernoulli random variable

Each cell in the map is a *probability*
Ground truth map is binary, but unknown.



Basic concepts in probability

- Key idea: quantities such as sensor measurements, states of a robot, and its environment are modeled as random variables (RVs)
- Discrete RV: the space of all the values that a random variable *X* can take on is *discrete*; characterized by probability mass function (pmf)

$$p(X = x) \quad (\text{or } p(x)), \qquad \sum_x p(X = x) = 1$$

Random variable

Specific value

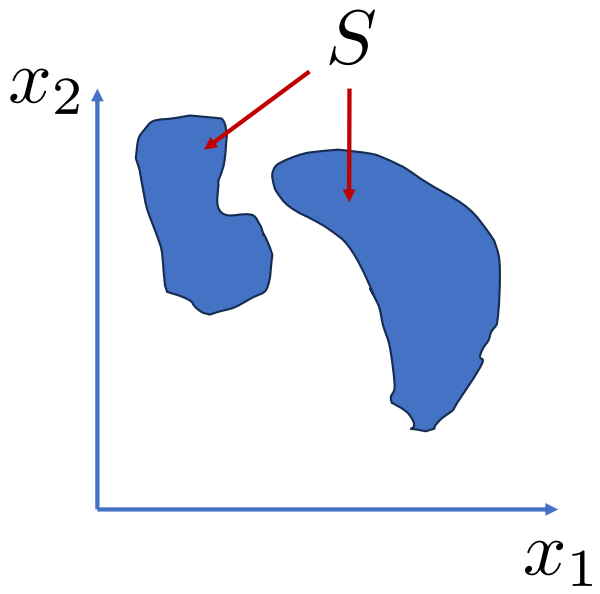
- Continuous RV: the space of all the values that a random variable *X* can take on is *continuous;* characterized by probability density function (pdf)

$$P(a \leq X \leq b) = \int_a^b p(x)\,dx, \quad P(X \in S) = \int_{x \in S} p(x)\,dx \qquad \int_{-\infty}^{\infty} p(x)\,dx = 1$$

Same for Random Vectors

$$X = [X_1 \quad X_2 \quad \cdots \quad X_n]^T \in \mathbb{R}^n$$

Pdf: $p(x) \geq 0 \quad \forall x \in \mathbb{R}^n$



$$P(X \in S) = \int_{x \in S} p(x)\, dx \qquad \int_{\mathbb{R}^n} p(x)\, dx = 1$$

Joint distribution, independence, and conditioning

- Joint distribution of two random variables *X* and *Y* is denoted as

$$p(x, y) := p(X = x \text{ and } Y = y)$$

- *X* and *Y* are independent iff

$$p(x, y) = p(x)p(y)$$

- Suppose we know that $Y = y$ (with $p(y) > 0$); conditioned on this fact, the probability that the X's value is x is given by

$$p(x \mid y) := \frac{p(x, y)}{p(y)}$$

Conditional probability

Note: if *X* and *Y* are independent

$$p(x \mid y) := p(x)!$$

Law of total probability

- For discrete RVs:

$$p(x) = \sum_y p(x, y) = \sum_y p(x \mid y)p(y)$$

- For continuous RVs:

$$p(x) = \int p(x, y)dy = \int p(x \mid y)p(y)dy$$

- Note: if $p(y) = 0$, define the product $p(x \mid y)p(y) = 0$

Bayes' rule

- Key relation between $p(x \mid y)$ and its "inverse," $p(y \mid x)$
- For discrete RVs:

$$p(x \mid y) = \frac{p(y \mid x)p(x)}{p(y)} = \frac{p(y \mid x)p(x)}{\sum_{x'} p(y \mid x')p(x')}$$

- For continuous RVs:

$$p(x \mid y) = \frac{p(y \mid x)p(x)}{p(y)} = \frac{p(y \mid x)p(x)}{\int p(y \mid x')p(x')\,dx'}$$

Bayes' rule and probabilistic inference

- Assume *x* is a quantity we would like to infer from *y*
- Bayes rule allows us to do so through the inverse probability, which specifies the probability of data *y* assuming that *x* was the cause

Posterior probability distribution

Prior probability distribution

$$p(x \mid y) = \frac{p(y \mid x)p(x)}{\int p(y \mid x')p(x')\,dx'}$$

Data

Normalizer, does not depend on $x := \eta^{-1}$

- Notational simplification

$$p(x \mid y) = \eta\, p(y \mid x)p(x)$$

Next time: pre-recorded lecture on Canvas

- More on prob. occ grid mapping