EKF SLAM

D.A. Forsyth, UIUC

Thread

- With two cameras and some calibration:
  - we can recover the position of 3D points
    - in the vehicle's coordinate system
- Together with an EKF, we can use this to recover
  - points in world coordinates (a map)
  - vehicle location
- In fact, we can do all this with one camera
  - with some minor care

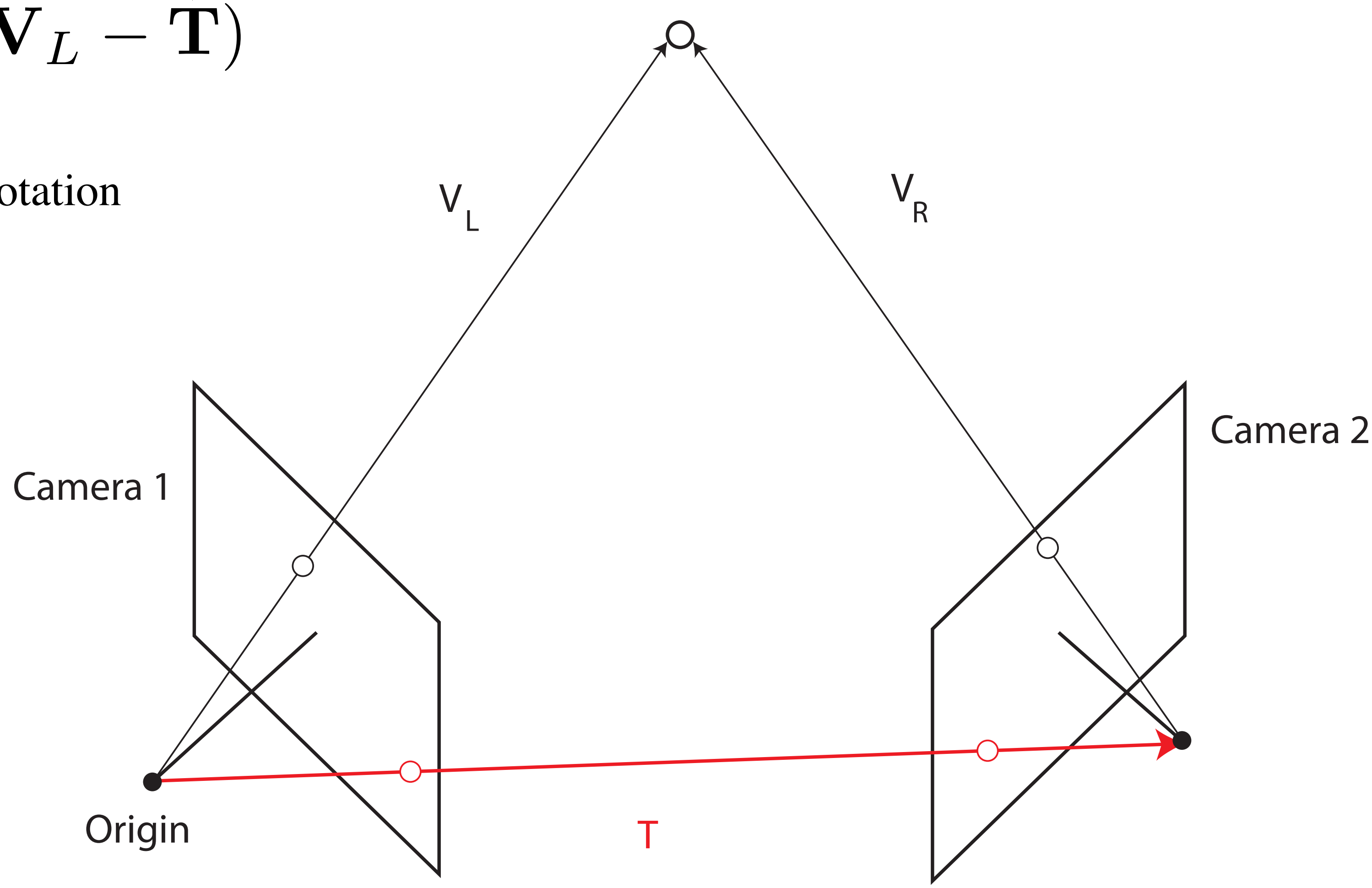
Two calibrated cameras yield 3D points, I

Two calibrated cameras yield 3D points, I

Camera translation

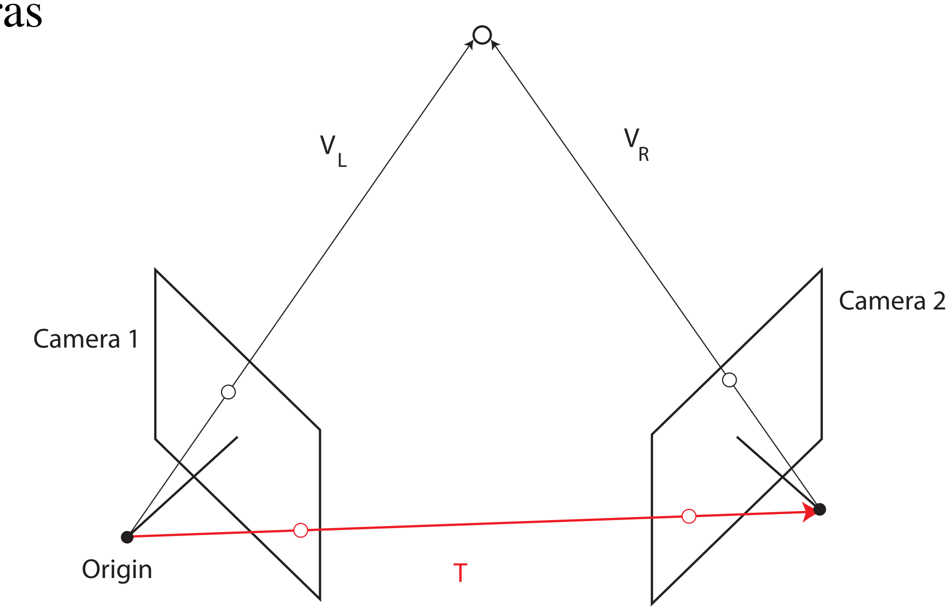
$$\mathbf{V}_R = \mathcal{R}(\mathbf{V}_L - \mathbf{T})$$

Camera rotation



Visual odometry established

- Correspondences between left and right yield
  - the fundamental matrix, and so the essential matrix
- The essential matrix yields
  - the rotation between two cameras
  - the translation *up to scale*



We can reconstruct in 3D

- 3D points:

And

$$\mathbf{X} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$$

$$\begin{pmatrix} x_1' \\ x_2' \\ x_3' \end{pmatrix} = \mathcal{R}\left[\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} - \mathbf{t}\right]$$

Point in camera one's coordinate system

Point in camera two's coordinate system

We can reconstruct in 3D, II

- Image points are:
  - (remember we know camera calibration!)

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} x_1/x_3 \\ x_2/x_3 \end{pmatrix} \qquad \begin{pmatrix} y'_1 \\ y'_2 \end{pmatrix} = \begin{pmatrix} x'_1/x'_3 \\ x'_2/x'_3 \end{pmatrix}$$

Recovering the depth

$$\mathbf{X} = \mathbf{y}x_3$$

$$x'_3\left(x'_1/x'_3\right) = x'_1$$

$$\left(\mathbf{r}_3^T\mathbf{X} - \mathbf{r}_3^T\mathbf{t}\right)y'_1 = \left(\mathbf{r}_1^T\mathbf{X} - \mathbf{r}_1^T\mathbf{t}\right)$$

$$\left(\mathbf{r}_3^T\mathbf{y}x_3 - \mathbf{r}_3^T\mathbf{t}\right)y'_1 = \left(\mathbf{r}_1^T\mathbf{y}x_3 - \mathbf{r}_1^T\mathbf{t}\right)$$

$$x_3 = \frac{(y'_1\mathbf{r}_3 - \mathbf{r}_1)^T\mathbf{t}}{(y'_1\mathbf{r}_3 - \mathbf{r}_1)^T\mathbf{y}}$$

The effect of scale

$$x_3 = \frac{(y_1' \mathbf{r}_3 - \mathbf{r}_1)^T \mathbf{t}}{(y_1' \mathbf{r}_3 - \mathbf{r}_1)^T \mathbf{y}}$$

- If we scale t
  - point coordinates scale
    - x_1=y_1 x_3, x_2=y_2 x_3
- Assume that scale is known
  - Easiest: fix two cameras in some position
  - Then we have points in 3D
  -

Thread

- With two cameras and some calibration:
  - we can recover the position of 3D points
    - in the vehicle's coordinate system
- Together with an EKF, we can use this to recover
  - points in world coordinates (a map)
  - vehicle location
- In fact, we can do all this with one camera
  - with some minor care

The SLAM Problem

- SLAM stands for simultaneous localization and mapping
- The task of building a map while estimating the pose of the robot relative to this map

- Why is SLAM hard?
  Chicken-or-egg problem:
  - a map is needed to localize the robot and
    a pose estimate is needed to build a map

From Burgard et al slides

Simplest case

- Vehicle moves in 2D
- Each measurement is
  - a 2D measurement
  - of position of a known beacon in vehicle coords
    - (i.e. we know which measurement corresponds to which 3D point)

State

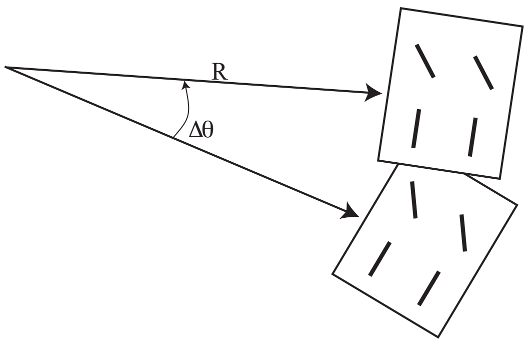
Position and orientation of the robot

$$\mathbf{x} = \begin{bmatrix} \mathcal{R} \\ \mathcal{M} \end{bmatrix} = \begin{bmatrix} \mathcal{R} \\ \mathcal{L}_1 \\ \ldots \\ \mathcal{L}_n \end{bmatrix}$$

Landmark 1 position in OCF

All landmark positions in original coordinate frame

A general movement model



$$\begin{bmatrix} x \\ y \\ \theta \end{bmatrix} \rightarrow \begin{bmatrix} x + R(\sin(\theta + \Delta\theta) - \sin\theta) \\ y - R(\cos(\theta + \Delta\theta) - \cos\theta) \\ \theta + \Delta\theta \end{bmatrix}$$

THIS ISN'T LINEAR!

v_t = velocity
omega_t = rotational velocity

$$\begin{bmatrix} x \\ y \\ \theta \end{bmatrix} \rightarrow \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} + \begin{bmatrix} \frac{v_t}{\omega_t}(\sin(\theta + \omega_t \Delta t) - \sin(\theta)) \\ -\frac{v_t}{\omega_t}(\cos(\theta + \omega_t \Delta t) - \cos(\theta)) \\ \omega_t \Delta t \end{bmatrix}$$

State update

$$\mathbf{x}_i = f(\mathbf{x}_{i-1}, \mathbf{n})$$

$$\mathbf{x} = \begin{bmatrix} \mathcal{R} \\ \mathcal{M} \end{bmatrix} = \begin{bmatrix} \mathcal{R} \\ \mathcal{L}_1 \\ \ldots \\ \mathcal{L}_n \end{bmatrix}$$

- The vehicle moves, as above;
  - but the landmarks don't move
  - and there isn't any noise

$$\begin{bmatrix} \mathcal{R} \\ \mathcal{L}_1 \\ \mathcal{L}_2 \\ \ldots \\ \mathcal{L}_n \end{bmatrix} \rightarrow \begin{bmatrix} h(\mathcal{R}) + \xi \\ \mathcal{L}_1 \\ \mathcal{L}_2 \\ \ldots \\ \mathcal{L}_n \end{bmatrix}$$

Recall: The extended Kalman filter

- Linearize:

$$\mathbf{x}_i = f(\mathbf{x}_{i-1}, \mathbf{n})$$

$$\mathcal{F}_x = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \cdots \\ \cdots & \frac{\partial f_i}{\partial x_j} & \cdots \end{bmatrix}$$

$$\mathcal{F}_n = \begin{bmatrix} \frac{\partial f_1}{\partial n_1} & \cdots & \cdots \\ \cdots & \frac{\partial f_i}{\partial n_j} & \cdots \end{bmatrix}$$

Posterior covariance of x_{i-1}

$$\mathbf{x}_i \sim N(f(\bar{\mathbf{x}}^+_{i-1}, \mathbf{0}), \mathcal{F}_x \Sigma^+_{i-1} \mathcal{F}_x^T + \mathcal{F}_n \Sigma_{n,i} \mathcal{F}_n^T)$$

Noise covariance

Measuring position

- Landmark is at: $\begin{bmatrix} u \\ v \end{bmatrix}$
  - in world coordinate system
- We record position in vehicle's frame:

vehicle orientation in world coords

point posn in world coords

Observation

$$\begin{bmatrix} x_v \\ y_v \end{bmatrix} = \mathcal{R}_{-\theta} \begin{bmatrix} (u - x) \\ (v - y) \end{bmatrix}$$

THIS ISN'T LINEAR!

point posn in vehicle coords

vehicle posn in world coords

The steps, EKF:

Have:

$$\overline{X}^{+}_{i-1} \qquad \Sigma^{+}_{i-1}$$

Construct:

$$\bar{X}^{-}_{i} = f_i(\bar{\mathbf{x}}^{+}_{i-1}, \mathbf{0}) \qquad \Sigma^{-}_{i} = \mathcal{F}_x \Sigma^{+}_{i-1} \mathcal{F}^{T}_{x} + \mathcal{F}_n \Sigma_{n,i} \mathcal{F}^{T}_{n}$$

Measurement arrives: $\mathbf{y}_i = g_i(\mathbf{x}_i, \mathbf{n})$

Now construct:

$$\bar{X}^{+}_{i} = \bar{X}^{-}_{i} + \mathcal{K}_i \left[\mathbf{y}_i - g_i(\bar{X}^{-}_{i}, \mathbf{0}) \right] \qquad \Sigma^{+}_{i} = \left[Id - \mathcal{K}_i \mathcal{G}_x \right] \Sigma^{-}_{i}$$

Where:

$$\mathcal{K}_i = \Sigma^{-}_{i} \mathcal{G}^{T}_{x} \left[\mathcal{G}_x \Sigma^{-}_{i} \mathcal{G}^{T}_{x} + \mathcal{G}_n \Sigma_{m,i} \mathcal{G}^{T}_{n} \right]^{-1}$$

In principle, now easy

- Rather horrid from the point of view of complexity
    - looks like we have to invert a 3+2N by 3+2N matrix!
- BUT
  - F_x is much simpler than it might look
    - the landmarks do not move!
  - F_n ditto
    - there is no noise in the landmark updates - the landmarks are fixed
  - Outcome:
    - We can deal with landmarks one by one
      - and so do many small matrix inversions rather than one large one

State update

$$\mathbf{x}_i = f(\mathbf{x}_{i-1}, \mathbf{n})$$

$$\mathbf{x} = \begin{bmatrix} \mathcal{R} \\ \mathcal{M} \end{bmatrix} = \begin{bmatrix} \mathcal{R} \\ \mathcal{L}_1 \\ \ldots \\ \mathcal{L}_n \end{bmatrix}$$

- The vehicle moves, as above;
  - but the landmarks don't move
  - and there isn't any noise

$$\begin{bmatrix} \mathcal{R} \\ \mathcal{L}_1 \\ \mathcal{L}_2 \\ \ldots \\ \mathcal{L}_n \end{bmatrix} \rightarrow \begin{bmatrix} h(\mathcal{R}) + \xi \\ \mathcal{L}_1 \\ \mathcal{L}_2 \\ \ldots \\ \mathcal{L}_n \end{bmatrix}$$

State update, II

- BUT
  - F_x is much simpler than it might look
    - the landmarks do not move!
  - F_n ditto
    - there is no noise in the landmark updates - the landmarks are fixed

N=Number of landmarks

3   2 N

$$\mathcal{F}_x = \begin{bmatrix} \frac{\partial f_{\mathcal{R}}}{\partial \mathcal{R}} & 0 \\ 0 & \mathcal{I} \end{bmatrix}$$

$$\mathcal{F}_n = \begin{bmatrix} \frac{\partial f_{\mathcal{R}}}{\partial \mathbf{n}} & 0 \\ 0 & 0 \end{bmatrix}$$

State update, III

- Imagine we have 2 landmarks

Recall EKF: $$\mathbf{x}_i \sim \mathcal{N}(f(\mathbf{x}_{i-1}, \mathbf{0}), \mathcal{F}_x \Sigma_{i-1}^{+} \mathcal{F}_x^T + \mathcal{F}_n \Sigma_{n,i} \mathcal{F}_n^T$$

$$\mathcal{F}_x = \begin{bmatrix} \mathcal{W} & 0 & 0 \\ 0 & \mathcal{I} & 0 \\ 0 & 0 & \mathcal{I} \end{bmatrix}$$

$$\Sigma_{i-1}^{+} = \begin{bmatrix} \mathcal{A} & \mathcal{B} & \mathcal{C} \\ \mathcal{B}^T & \mathcal{D} & \mathcal{E} \\ \mathcal{C}^T & \mathcal{E}^T & \mathcal{F} \end{bmatrix}$$

$$\mathcal{F}_x \Sigma_{i-1}^{+} \mathcal{F}_x^T = \begin{bmatrix} \mathcal{W}\mathcal{A}\mathcal{W}^T & \mathcal{W}\mathcal{A} & \mathcal{W}\mathcal{B} \\ \mathcal{B}^T\mathcal{W}^T & \mathcal{D} & \mathcal{E} \\ \mathcal{C}^T\mathcal{W} & \mathcal{E}^T & \mathcal{F} \end{bmatrix}$$

Notice fewer matrix multiplies!

State update, IV

- Imagine we have 2 landmarks

Recall EKF: $\mathbf{x}_i \sim \mathcal{N}(f(\mathbf{x}_{i-1}, \mathbf{0}), \mathcal{F}_x \Sigma^+_{i-1} \mathcal{F}_x^T + \mathcal{F}_n \Sigma_{n,i} \mathcal{F}_n^T$

$$\mathcal{F}_n = \begin{bmatrix} \mathcal{V} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad \Sigma_{n,i} = \begin{bmatrix} \mathcal{H} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\mathcal{F}_n \Sigma_{n,i} \mathcal{F}_n^T = \begin{bmatrix} \mathcal{V}\mathcal{H}\mathcal{V}^T & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Notice fewer matrix multiplies!

More simplifications

- BUT
  - G_x is much simpler than it might look
    - each set of measurements affected by only one landmark!

N N=Number of landmarks

3 2

$$\mathcal{G}_x = \begin{bmatrix} \frac{\partial \mathcal{O}_1}{\partial \mathcal{R}} & \frac{\partial \mathcal{O}_1}{\partial \mathcal{L}_1} & 0 & 0 & 0 & 0 \\ \frac{\partial \mathcal{O}_2}{\partial \mathcal{R}} & 0 & \frac{\partial \mathcal{O}_2}{\partial \mathcal{L}_2} & 0 & 0 & 0 \\ \ldots & & & & & \\ \frac{\partial \mathcal{O}_N}{\partial \mathcal{R}} & 0 & 0 & 0 & 0 & \frac{\partial \mathcal{O}_N}{\partial \mathcal{L}_N} \end{bmatrix}$$

2 N

More simplifications

- BUT
  - G_n is usually much simpler than it might look
    - noise is usually additive normal noise
    - This means that the term: $\mathcal{G}_n \Sigma_{n,i} \mathcal{G}_n^T$
      - is actually a block diagonal matrix

Big simplification

- The nasty bit…

$$\left[\mathcal{G}_x \Sigma_i^- \mathcal{G}_x^T + \mathcal{G}_n \Sigma_{m,i} \mathcal{G}_n^T\right]^{-1}$$

- But notice key point
  - measurements interact only through the position/orientation of the vehicle
  - each measurement depends on only one landmark and pose of v.
  - OR measurements are conditionally independent conditioned on pose of v.
  - OR you could subdivide time and update measurements one by one
  - OR matrix G_x has the sparsity structure above
- (the same point, manifesting in different ways)

Subdividing time…

- We receive measurements of landmarks in some order
  - a measurement of the position of landmark i affects the whole state
    - because it changes your estimate of the pose of the vehicle
      - and that affects your estimate of state of every landmark
    - BUT
      - the change in estimate of pose depends ONLY on
        - pose
        - landmark i

Subdividing time…

- Sequence
  - repeat
    - move (so make predictions)
    - landmark 1 measurement arrives (update pose and so all based on 1)
    - …
    - landmark N measurement arrives (update pose and so all based on N)

Steps in EKF

$$\mathcal{K}_i = \Sigma_i^- \mathcal{G}_x^T \left[\mathcal{G}_x \Sigma_i^- \mathcal{G}_x^T + \mathcal{G}_n \Sigma_{m,i} \mathcal{G}_n^T\right]^{-1}$$

$$\mathbf{x}_i^+ = \mathbf{x}_i^- + \mathcal{K}_i \left[\mathbf{y}_i - g(\mathbf{x}_i^-, \mathbf{0})\right]$$

$$\Sigma_i^+ = \left[\mathcal{I} - \mathcal{K}_i \mathcal{G}_x\right] \Sigma_i^-$$

One measurement from one landmark!

Steps in EKF

3+2Nx2 | 3+2N x 2 | 2 x 2

$$\mathcal{K}_i = \Sigma_i^- \mathcal{G}_x^T \left[\mathcal{G}_x \Sigma_i^- \mathcal{G}_x^T + \mathcal{G}_n \Sigma_{m,i} \mathcal{G}_n^T\right]^{-1}$$

Notice:
Inverting only a small matrix

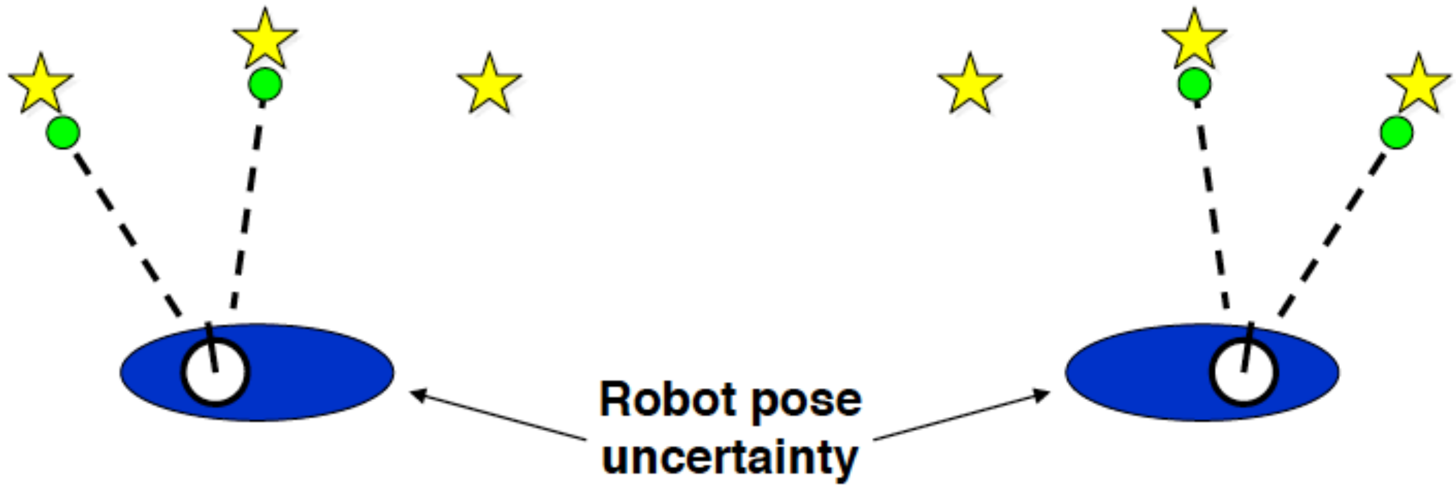
3+2Nx2 | 2 x 1

$$\mathbf{x}_i^+ = \mathbf{x}_i^- + \mathcal{K}_i \left[\mathbf{y}_i - g(\mathbf{x}_i^-, \mathbf{0})\right]$$

Notice:
But affecting the whole state!

$$\Sigma_i^+ = \left[\mathcal{I} - \mathcal{K}_i \mathcal{G}_x\right] \Sigma_i^-$$

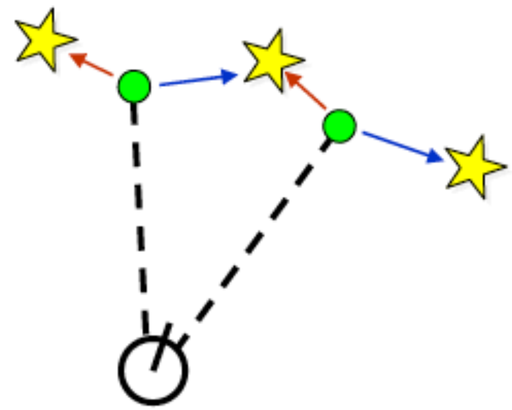
Why is SLAM a hard problem?



- In the real world, the mapping between observations and landmarks is unknown
- Picking wrong data associations can have catastrophic consequences
- Pose error correlates data associations

From Burgard et al slides

Data Association Problem



- A data association is an assignment of observations to landmarks
- In general there are more than $\binom{n}{m}$ (n observations, m landmarks) possible associations
- Also called "assignment problem"

From Burgard et al slides

Landmarks

- Which measurement comes from which landmark?
  - data association -
    - use some form of bipartite graph matching
      - Idea: $\overline{\mathbf{x}}_i^-$
        - predicts landmark positions, vehicle position before obs
          - compute distances between all pairs of
            - predicted obs, real obs
          - bipartite graph matcher
          - OR greedy

Landmarks

- No measurement from a landmark?
  - structure of EKF means you can process landmarks one by one
    - that’s what all the matrix surgery was about
    - so don’t update that landmark
- How do we know no measurement from a landmark?
  - refuse to match if distance in greedy/bipartite is too big
  - other kinds of matching problem (color, features, etc)

New landmarks

- Sequence
  - repeat
    - move (so make predictions)
    - landmark 1 measurement arrives (update pose and so all based on 1)
    - …
    - landmark N measurement arrives (update pose and so all based on N)
    - check if there is a new landmark
      - if so
        - initialize landmark position and covariance
          - conditioned on current state and measurement
        - process from now on (we have N+1 landmarks)

Initializing new landmarks, I

- Recall we have state estimate, state covar

vehicle orientation in world coords

point posn in world coords

Observation

$$\begin{bmatrix} x_v \\ y_v \end{bmatrix} = \mathcal{R}_{-\theta} \begin{bmatrix} (u - x) \\ (v - y) \end{bmatrix}$$

point posn in vehicle coords

vehicle posn in world coords

Initializing new landmarks, II

- Recall we have state estimate, state covar

vehicle orientation in world coords

Observation

point posn in world coords

$$\mathbf{R}_\theta \begin{bmatrix} x_u \\ x_v \end{bmatrix} + \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} u \\ v \end{bmatrix}$$

vehicle posn in world coords

Inverse observation model

$$\mathbf{R}_\theta \begin{bmatrix} x_u \\ x_v \end{bmatrix} + \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} u \\ v \end{bmatrix} \qquad \begin{bmatrix} u \\ v \end{bmatrix} = h(\text{state}, \text{meas})$$

- Where the landmark is
  - conditioned on measurement and state
- Advantage:
  - when a new landmark is encountered, we can introduce it

Initializing new landmarks, III

- Recall we have state estimate, state covar

$$\begin{bmatrix} u \\ v \end{bmatrix} = h(\text{state}, \text{meas})$$

$$\text{state} \sim N(\bar{X}_i^+, \Sigma_i^+) \qquad \text{(our model)}$$

- Previous results yield compute mean, covar of landmark!

Measuring distance and orientation

- Landmark is at:
  - in global coordinate system

$$\begin{bmatrix} u \\ v \end{bmatrix}$$

- We record distance and heading:
  - measurement

$$\begin{bmatrix} d \\ \phi \end{bmatrix} = \begin{bmatrix} \sqrt{(x-u)^2+(y-v)^2} \\ \mathrm{atan2}(y-u, x-v) - \theta \end{bmatrix}$$

THIS ISN'T LINEAR!

A further trick: inverting measurement

- Example: measure distance and orientation to point

$$\begin{bmatrix} u \\ v \end{bmatrix}$$

← point posn in world coords

vehicle posn in world coords ↓

Observation

$$\begin{bmatrix} d \\ \phi \end{bmatrix} = \begin{bmatrix} \sqrt{(x-u)^2 + (y-v)^2} \\ \mathrm{atan2}(y-u, x-v) - \theta \end{bmatrix}$$

↑ vehicle orientation in world coords

Range and bearing

Landmark position

Observation → $$\begin{bmatrix} d \\ \phi \end{bmatrix} = \begin{bmatrix} \sqrt{(x-u)^2 + (y-v)^2} \\ \text{atan2}(y-u, x-v) - \theta \end{bmatrix}$$

Vehicle state

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} x + (d+\xi)\sin(\phi+\zeta+\theta) \\ y + (d+\xi)\cos(\phi+\zeta+\theta) \end{bmatrix}$$

These are measurements of landmark ONLY

Noise affecting measurements

Here use the current estimate of vehicle state

Bearing only

- Important case
  - cameras
- EKF is fine
  - we're OK with one measurement of two degrees of freedom
- but how do we initialize?
  - inverse observation model idea needs work

Initializing bearing only

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} x + (d + \xi)\sin(\phi + \zeta + \theta) \\ y + (d + \xi)\cos(\phi + \zeta + \theta) \end{bmatrix}$$

Noise affecting measurements

Don't know d!

- Inverse observation model presents problems
  - because location could be anywhere on a line
- Apply a prior!
  - d ~ N(something, something big)
  - now previous results yield mean, covar of [u, v]'