Grid Maps

1. What it is

① grid map is a way to represent the map. World is divided into grid of cells 

We can use this to draw a map (large map may need large memory).

② Assumption 1: each cell is totally occupied or totally free

③ Assumption 2: world is static. Cells don't change state. (no true in reality)

④ Assumption 3: cells are independent (not very true in reality).

2. Representation.

naturally we use a binary random variable to model a cell:

$p(m_i)=1$ occupied. $p(m_i)=0$ not occupied.

but we also allow $p(m_i)=0.5$ not known

Given sensor data.

e.g.

m_1	m_2
m_3	m_4

$m_1=0.9$
$m_2=0.5$
$m_3=0.8$
$m_4=0.1$

Probability of a state.

$$P(M=\boxed{\ }) = \prod_{i=1}^{4} p(M_i=m_i) = 0.9\times(1-0.5)\times0.8\times(1-0.1)$$

3. Mapping with known poses.

- The problem: Given sensor data and pose of sensor (robot), estimate a map.
- Mathematically:

$$p(m|z_{1:t},x_{1:t}) = \prod_i p(m_i|z_{1:t},x_{1:t})$$

Idea 1: To deal with binary variable, we can compute ratio $\frac{p(A)}{1-p(A)}$ for estimation

Idea 2: using Bayes Recursive Filter. That is, find relation between current state and previous state, thus form a recursive algorithm

- Go for $p(m_i|z_{1:t},x_{1:t})$.

knowing x_t, past pose of obs does not help

$$p(m_i|z_{1:t},x_{1:t}) \underset{\text{Bayes Rule}}{=} \frac{p(z_t|m_i,z_{1:t-1},x_{1:t})\,p(m_i|z_{1:t-1},x_{1:t})}{p(z_t|z_{1:t-1},x_{1:t})}$$

$$\left(p(m_i|z_t)=\frac{p(z_t|m_i)\,p(m_i)}{p(z_t)}\right)$$

at x_t, no z_t. so x_t is not usefull

$$\underset{\text{Markov}}{=} \frac{p(z_t|m_i,x_t)\,p(m_i|z_{1:t-1},x_{1:t-1})}{p(z_t|z_{1:t-1},x_{1:t})}$$

• Do the same for $\neg m_i$

$$p(\neg m_i | z_{1:t}, x_{1:t}) = \frac{p(z_t | \neg m_i, x_t)\, p(\neg m_i | z_{1:t-1}, x_{1:t-1})}{p(z_t | z_{1:t-1}, x_{1:t})}$$

• ratio:

$$\frac{p(m_i | z_{1:t}, x_{1:t})}{1 - p(m_i | z_{1:t}, x_{1:t})} = \frac{\dfrac{p(z_t | m_i, x_t)\, p(m_i | z_{1:t-1}, x_{1:t-1})}{\cancel{p(z_t | z_{1:t-1}, x_{1:t})}}}{\dfrac{p(z_t | \neg m_i, x_t)\, p(\neg m_i | z_{1:t-1}, x_{1:t-1})}{\cancel{p(z_t | z_{1:t-1}, x_{1:t})}}} = \underbrace{\frac{p(z_t | m_i, x_t)}{p(z_t | \neg m_i, x_t)}}_{\text{bad.}} \underbrace{\boxed{\frac{p(m_i | z_{1:t-1}, x_{1:t-1})}{p(\neg m_i | z_{1:t-1}, x_{1:t-1})}}}_{\text{good}}$$

• Bayes for swapping: $p(z_t | m_i, x_t) \rightarrow p(m_i | z_t, x_t)$. (You see why swapping later)

$$p(z_t | m_i, x_t) = \frac{p(m_i | z_t, x_t) \cdot p(z_t | x_t)}{p(m_i | x_t)}$$

markov: if no Observation. x_t does not help.

$$= \frac{p(m_i | z_t, x_t)\, p(z_t | x_t)}{p(m_i)}$$

$$\Rightarrow p(z_t | \neg m_i, x_t) = \frac{p(\neg m_i | z_t, x_t)\, p(z_t | x_t)}{p(\neg m_i)}.$$

$$\Rightarrow \frac{p(m_i | z_{1:t}, x_{1:t})}{p(\neg m_i | z_{1:t}, x_{1:t})} = \underbrace{\frac{p(m_i | z_t, x_t)}{p(\neg m_i | z_t, x_t)}}_{\text{current observation and state}} \underbrace{\boxed{\frac{p(\neg m_i)}{p(m_i)}}}_{\text{prior}} \underbrace{\boxed{\frac{p(m_i | z_{1:t-1}, x_{1:t-1})}{p(\neg m_i | z_{1:t-1}, x_{1:t-1})}}}_{\text{recursion}}$$

Recursive: for the task of $p(m_i | z_{1:t}, x_{1:t})$, we can keep its history. for each new state, I only need prior and ~~[illegible]~~ (obs, pos) for that timestamp to solve the mapping problem.

• Ratio to probability

$$\frac{p(x)}{1-p(x)} = Y \Rightarrow p(x) = \frac{1}{1 + \frac{1}{Y}}$$

$$\Rightarrow p(m_i | z_{1:t}, x_{1:t}) = \left[1 + \frac{1 - p(m_i | z_t, x_t)}{p(m_i | z_t, x_t)} \frac{p(m_i)}{1 - p(m_i)} \frac{1 - p(m_i | z_{1:t-1}, x_{1:t-1})}{p(m_i | z_{1:t-1}, x_{1:t-1})}\right]^{-1}$$

- For efficiency reason. We use log-odds notation.

$$l(m_i|z_{1:t}, x_{1:t}) \equiv \log\left(\frac{p(m_i|z_{1:t}, x_{1:t})}{1-p(m_i|z_{1:t}, x_{1:t})}\right)$$

thus the transform becomes:

$$l(x) = \log \frac{p(x)}{1-p(x)} \Rightarrow p(x) = 1 - \frac{1}{1+\exp l(x)}$$

And the formula becomes

$$l(m_i|z_{1:t}, x_{1:t}) \equiv l_{t,i}$$

$$= \underbrace{l(m_i|z_t, x_t)}_{\text{inverse sensor model}} + \underbrace{l(m_i|z_{1:t-1}, x_{1:t-1})}_{\text{recursive term}} - \underbrace{l(m_i)}_{\text{prior}}$$

$\Rightarrow$ Algorithm: given $\{l_{t-1,i}\}$, x_t, z_t.

for each cell m_i

  if m_i is in perception field of z_t.

    $l_{t,i} = l_{t-1,i} + \text{inv\_sensor\_model}(m_i, x_t, z_t) - l_0$.

  else

    $l_{t,i} = l_{t-1,i}$.

  end

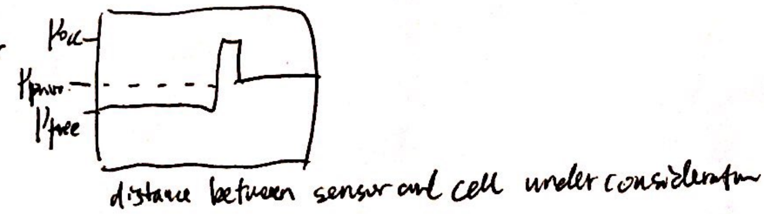
end.

Remark: There are many assumptions in this algorithm

1. Grid cells are binary
2. Grid cells are independent
3. Grid cells are static → we don't have a prediction step.
4. We know the poses perfectly → not true

4. Inverse sensor model example

laser range finder



5. Scan match.

In reality, motion is noisy. while sensor is rather precise

=) scan matching tries to incrementally align two scans or a ~~sca~~ map to a scan without revising past/map

It is in essence, pose correction.

$$x_t^* = \arg\max_{x_t} \{ \underbrace{p(z_t | x_t, m_{t-1})}_{\text{measurement}} \underbrace{p(x_t | u_{t-1}, x_{t-1}^*)}_{\text{motion}} \}$$

However, scan matching can only have locally consistent estimates and is not sufficient to build a large consistent map.

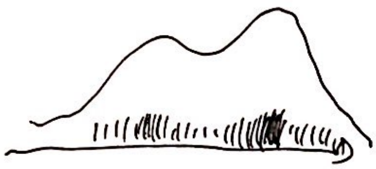
Particle Filters

1. Kalman Filter Assumes a Gaussian model. Particle Filter uses a non-parametric Approach for approximating a distribution.

- Natural idea would be to use samples $\mathcal{X} = \{ \langle x^{[j]}, w^{[j]} \rangle \}$ $j = 1 \cdots J$.

  particle sample ($x^{[j]}$), weight ($w^{[j]}$). (we will see later what this means)

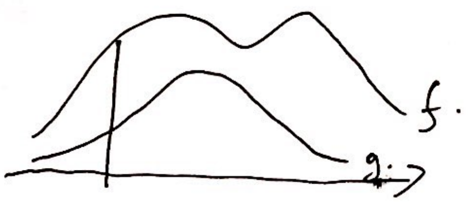
- The more particles fall into a region, the higher the probability of the region.



2. Question Comes: How to obtain the sample, in a way that makes sense?

- We know how to sample a gaussian uniformly: $x \leftarrow \frac{1}{2} \sum_{i=1}^{12} \text{rand}(-6, 6)$.

- Importance sampling: use a different distribution g to generate sample from f. And we know how to sample well in g.



① Introduce weight: $w = \frac{f}{g}$.
each sample will have a weight

② weights can be normalized by $\tilde{w}_i = \frac{w_i}{\sum w_i}$

③ weights and samples together characterizes a distribution. $p(x) = \sum_{j=1}^{J} w^{[j]} \delta_{x^{[j]}}(x)$. (weighted counts of particles in that region)

④ High weight means: one sample may "effectively" count for several samples.

~~[illegible]~~ This means we need to know our target f.

Obs: for {high dimension distribution, Large variance distribution}, we need many samples for a good approximation

3. ~~[illegible]~~ We can use a resampling strategy to implicitly weight the samples:

Resampling: Given $\langle x_t^{[j]}, w_t^{[j]} \rangle$ $j = 1 \cdots J$. $= \mathcal{X}_t$

Repeat J times: draw $i \in 1 \cdots J$ with probability $\propto w_t^{[i]}$ and add sample to $\mathcal{X}_t$.

Obs:

(1) effect of resampling is to replace unlikely samples by more likely ones "survival of fittest"

(2) Reduce number of samples to maintain.

⇒ particle Filter Algorithm. (purpose: use sampling method to approximate a distribution. Assumption: have proposal. have target.)

$\bar{\mathcal{X}}_t = \mathcal{X}_t = \phi$

(1) sampling $\mathcal{X}_t$: for $j = 1 : J$.

sample $x_t^{[j]} \sim \pi(x_t)$ ← proposal.

$$w_t^{[j]} = \frac{p(x_t^{[j]})}{\pi(x_t^{[j]})} \leftarrow \text{target.}$$

$\Rightarrow \bar{\mathcal{X}}_t = \bar{\mathcal{X}}_t + \langle x_t^{[j]}, w_t^{[j]} \rangle$ Add to $\bar{\mathcal{X}}_t$.

end.

(2) resample.

for $j = 1 : J$.

draw $i \in 1 \cdots J$ with prob $\propto w_t^{[i]}$

add sample to $\mathcal{X}_t$.

end.

A well implemented algorithm

4. How to do the resampling?

Stochastic universal sampling.

(low variance)

5. Apply Particle Filter For Localization

Given $\mathcal{X}_{t-1}, u_t, z_t$. find $\mathcal{X}_t$ (recursive filter)

(1) where do we sample from? $x_t^{[i]} \sim p(x_t \mid u_t, x_{t-1}^{[j]})$ odometry model

(2) How to compute weight? $w_t^{[j]} = p(z_t \mid x_t^{[j]})$. observation for correction

Interpretation: Given current state is $x_t^{[i]}$, How likely will I observe z_t?

obs: each particle is a trajectory proposal. Then they survival according to "survival of fittest"

move u_t → Sample. make observations. → remain filters whose observation are close to our sensor observation

we Are here!

It's important to sample here. not just translate the samples

Fast SLAM 1.0.

1. Particle Filter.

1.1. Three steps: Sample from proposal distribution.

Compute importance weights

Resampling.

1.2. Particle Filter can be used for localization: Given ~~x_{t-1}~~, u_t, z_t, find x_t.

1.3. works well in low dimension, but sample a high-dim distribution is hard.

1.4. Jump to SLAM.

localization $\longrightarrow$ SLAM

$x_t = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}$. $x = (x_{1:t}, m_{1,x}, m_{1,y} \dots m_{n,x}, m_{n,y})^T$

high-dim state vector

Solving this issue is important.

2. Rao-Blackwellization

2.1. Idea: mapping with known poses is simple (grid maps). so we can focus our state as $x = (x_{1:t})^T$.

This is to say, each sample represents a possible trajectory. And if this is true, we apply mapping with known poses, we get a map from each sample and compare with what we see

2.2. Mathematics.

$p(a,b) = p(b|a)\, p(a)$.

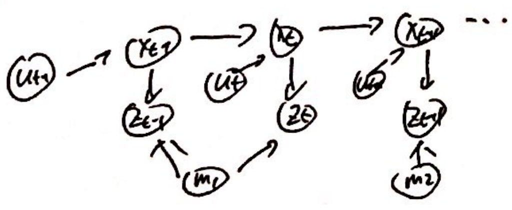
If $p(b|a)$ can be easily computed, then sample $p(a)$ only. And compute $p(b/a)$ for each sample. In essence, we sampled $p(a,b)$.

2.3 RB For SLAM.

when poses are known controls can be ignored

$p(x_{0:t}, m_{1:m} | z_{1:t}, u_{1:t}) = \underbrace{p(x_{0:t} | z_{1:t}, u_{1:t})}_{\text{particle}}\ \underbrace{p(m_{1:m} | x_{0:t}, z_{1:t})}_{\text{mapping}}$.

2.4. The ~~[illegible]~~ dependence model.



$x_{0:t}$ are known. $u_{1:t}$ are independent. Landmarks are also independent

2.5. Computation.

$$p(x_{0:t}, m_{1:M} | z_{1:t}, u_{1:t}) = \underbrace{p(x_{0:t} | z_{1:t}, u_{1:t})}_{\text{particle Filter MCL.}} \prod_{i=1}^{M} p(m_i | x_{0:t}, z_{1:t}).$$

2x2 covariance matrix

2-dim EKF

build map for.

note Also each sample is a path hypothesis but we don't sense past pose. So we only need to maintain a 3-dim pose vector. for the next state

3. Fast SLAM Algorithm

3.1. particles.

2x2 EKF.

state: $\boxed{x, y, \theta. \quad m_1, \quad m_2, \ \ldots m_M}$

3.2. Sampling:

$x_t^{[k]} \sim p(x_t | x_{t-1}^{[k]}, u_t)$ for each sample, apply odometry model, draw sample to get $x_t^{[k]}$

3.3. Importance weight:

current

predicted obs

$$w^{[k]} = |2\pi Q|^{-\frac{1}{2}} \exp\{-\frac{1}{2}(z_t - \hat{z}^{[k]})^T Q^{-1} (z_t - \hat{z}^{[k]})\}.$$

actual obs.

every particle has his own prediction.

3.4. The Algorithm.

Known: $z_t, c_t, u_t, \chi_{t-1}$.

① sample:

for $k=1$ to N do

Let $\langle x_{t-1}^{[k]}, \langle \mu_{1,t-1}^{[k]}, \Sigma_{1,t-1}^{[k]} \rangle \ldots \rangle$ be particle k in χ_{t-1}.

$x_t^{[k]} \sim p(x_t | x_{t-1}^{[k]}, u_t)$

② Correction.

$j = c_t$ observed feature.

if feature j never seen before.

Initialize.

$$\mu_{j,t}^{[k]} = h^{-1}(z_t, x_t^{[k]})$$

$$H = h'(\mu_{j,t}^{[k]}, x_t^{[k]})$$

$$\Sigma_{j,t}^{[k]} = H^{-1} Q_t (H^{-1})^T$$

$$w^{[k]} = p_0.$$

else. update landmark:

$$\langle \mu_{j,t}^{[k]}, \Sigma_{j,t}^{[k]} \rangle = \text{EKF-Update}(\,)$$

$$w^{[k]} = |2\pi Q|^{-\frac{1}{2}} \exp\{-\tfrac{1}{2}(z_t - \hat{z}^{[k]})^T Q^{-1} (z_t - \hat{z}^{[k]})\}$$

$$Q = H \Sigma_{j,t-1}^{[k]} H^T + Q_t. \quad \text{(Do before update)}$$

endif.

leave all unobserved features j' as they are.

endfor.

$$\chi_t = \text{Resample}\left(\langle x_t^{[k]}, \langle \mu_{1,t}^{[k]}, \Sigma_{1,t}^{[k]} \rangle, \overset{\text{landmarks}}{\ldots}, w^{[k]} \rangle_{k=1\ldots N}\right)$$

EKF-update

$$\hat{z}^{[k]} = h(\mu_{j,t-1}^{[k]}, x_t^{[k]}) \quad \text{— measure predict}$$

$$H = h'(\mu_{j,t-1}^{[k]}, x_t^{[k]}) \quad \text{— Jacobian}$$

$$Q = H \Sigma_{j,t-1}^{[k]} H^T + Q_t \quad \text{— measurement covariance.}$$

$$K = \Sigma_{j,t-1}^{[k]} H^T Q^{-1} \quad \text{— Kalman Gain}$$

$$\left.\begin{aligned} \mu_{j,t}^{[k]} &= \mu_{j,t-1}^{[k]} + K(z_t - \hat{z}^{[k]}) \\ \Sigma_{j,t}^{[k]} &= (I - KH)\Sigma_{j,t-1}^{[k]} \end{aligned}\right\} \text{update}$$

4. Importance weights Derivation

$$w^{[k]} = \frac{\text{target}(x^{[k]})}{\text{proposal}(x^{[k]})} \qquad \frac{p(x_{1:t} \mid z_{1:t}, u_{1:t})}{p(x_{1:t} \mid z_{1:t-1}, u_{1:t})}$$

Proposal is used step by step.

$$p(x_{1:t}|z_{1:t-1},u_{1:t}) = \underbrace{p(x_t|x_{t-1},u_t)}_{\text{from } \chi_{t-1} \text{ to } \bar{\chi}_t} \underbrace{p(x_{1:t-1}|z_{1:t-1},u_{1:t-1})}_{\chi_{t-1}}$$

$$\Rightarrow w^{[k]} = \frac{p(x_{1:t}^{[k]}|z_{1:t},u_{1:t})}{p(x_t^{[k]}|x_{t-1},u_t)\,p(x_{1:t-1}^{[k]}|z_{1:t-1},u_{1:t-1})}$$

$$\underline{\underline{\text{Bayes.}}} \quad \frac{\eta\, p(z_t|x_{1:t}^{[k]},z_{1:t-1})\,p(x_t|x_{t-1}^{[k]},u_t)}{p(x_t^{[k]}|x_{t-1}^{[k]},u_t)} \quad \frac{p(x_{1:t-1}^{[k]}|z_{1:t-1},u_{1:t-1})}{p(x_{1:t-1}^{[k]}|z_{1:t-1},u_{1:t-1})}$$

$$= \eta\, p(z_t|x_{1:t}^{[k]}, z_{1:t-1})$$

$$= \eta \int p(z_t|x_{1:t}^{[k]}, z_{1:t-1}, m_j)\, p(m_j|x_{1:t}^{[k]}, z_{1:t-1})\, dm_j$$

$$= \eta \int \underbrace{p(z_t|x_t^{[k]}, \overset{\text{d marker}}{m_j})}_{\mathcal{N}(z_t;\,\hat{z}^{[k]},\,Q_t)}\, \underbrace{p(m_j|x_{1:t-1}^{[k]}, z_{1:t-1})}_{\substack{\text{2x2 EKF.} \\ \mathcal{N}(m_j;\,\mu_{j,t-1}^{[k]},\,\Sigma_{j,t-1}^{[k]})}}\, dm_j$$

$$\Rightarrow Q = H\, \Sigma_{j,t-1}^{[k]} H^T + Q_t.$$

$$\Rightarrow w^{[k]} \simeq |2\pi Q|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(z_t - \hat{z}^{[k]})^T Q^{-1} (z_t - \hat{z}^{[k]})\right\}$$

5. Advantages

① per-particle data association. simple but effective