Structure and Synthesis of Robot Motion

Information: Seeking it, Managing without it

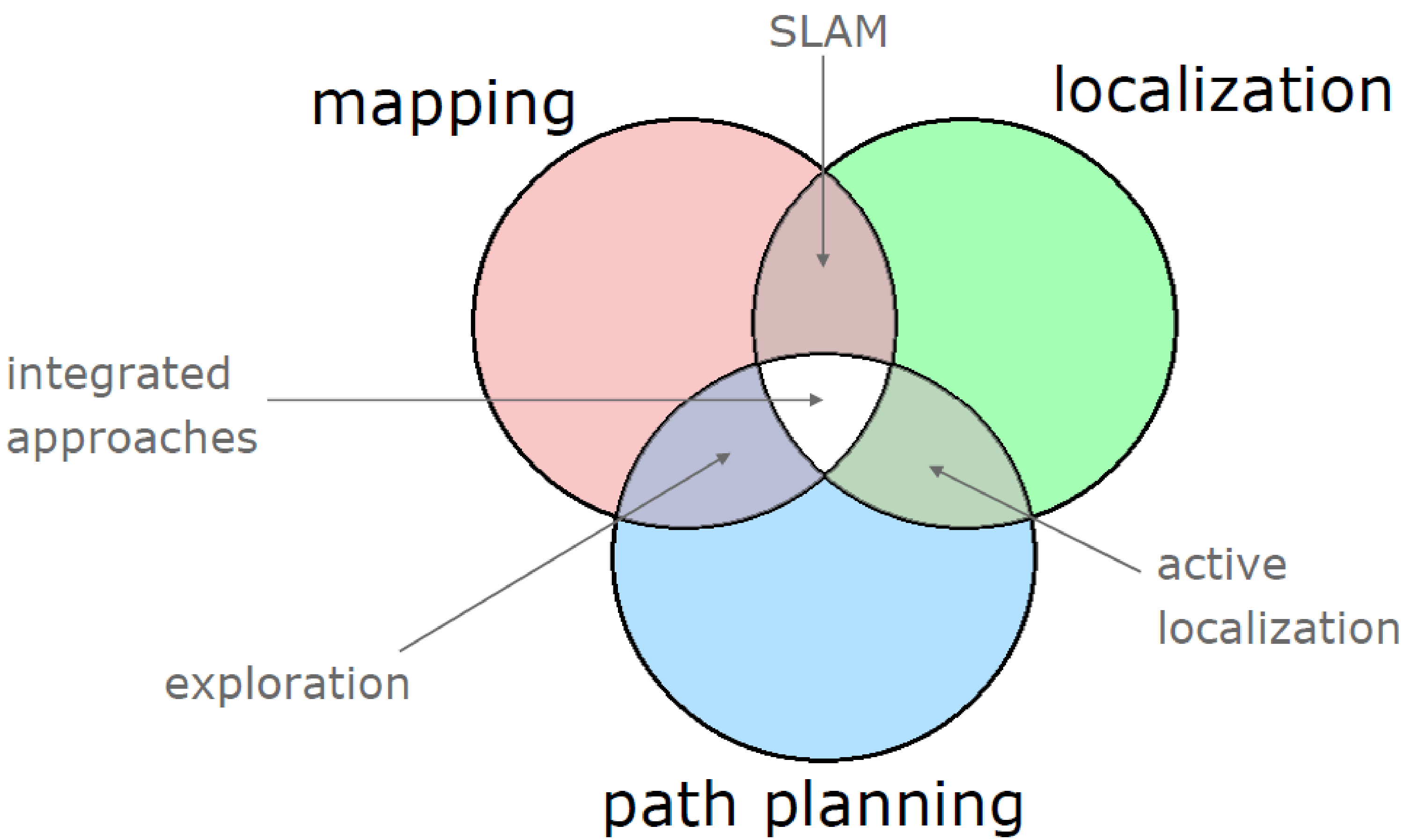
Subramanian Ramamoorthy
School of Informatics

26 March, 2012

Three Issues

- How do we pose the objective of actively seeking information as a part of the motion synthesis problem?
- How do we devise motion strategies that accommodate ‘minimal sensing’?
- The problem of information asymmetry and its relevance to robotics (just a few remarks on this one)

Q1: In Terms of Tasks of Mobile Robots



Exploration and SLAM

- SLAM is typically passive, because it consumes incoming sensor data
- Exploration actively guides the robot to cover the environment with its sensors
- Exploration in combination with SLAM: Acting under pose and map uncertainty
- Uncertainty should/needs to be taken into account when selecting an action

Mapping with Particle Filters

- Each particle represents a possible trajectory of the robot

- Each particle
  - maintains its own map and
  - updates it upon “mapping with known poses”

- Each particle survives with a probability proportional to the likelihood of the observations relative to its own map

Factorized Mapping Problem (Rao-Blackwellization)

poses map observations & odometry

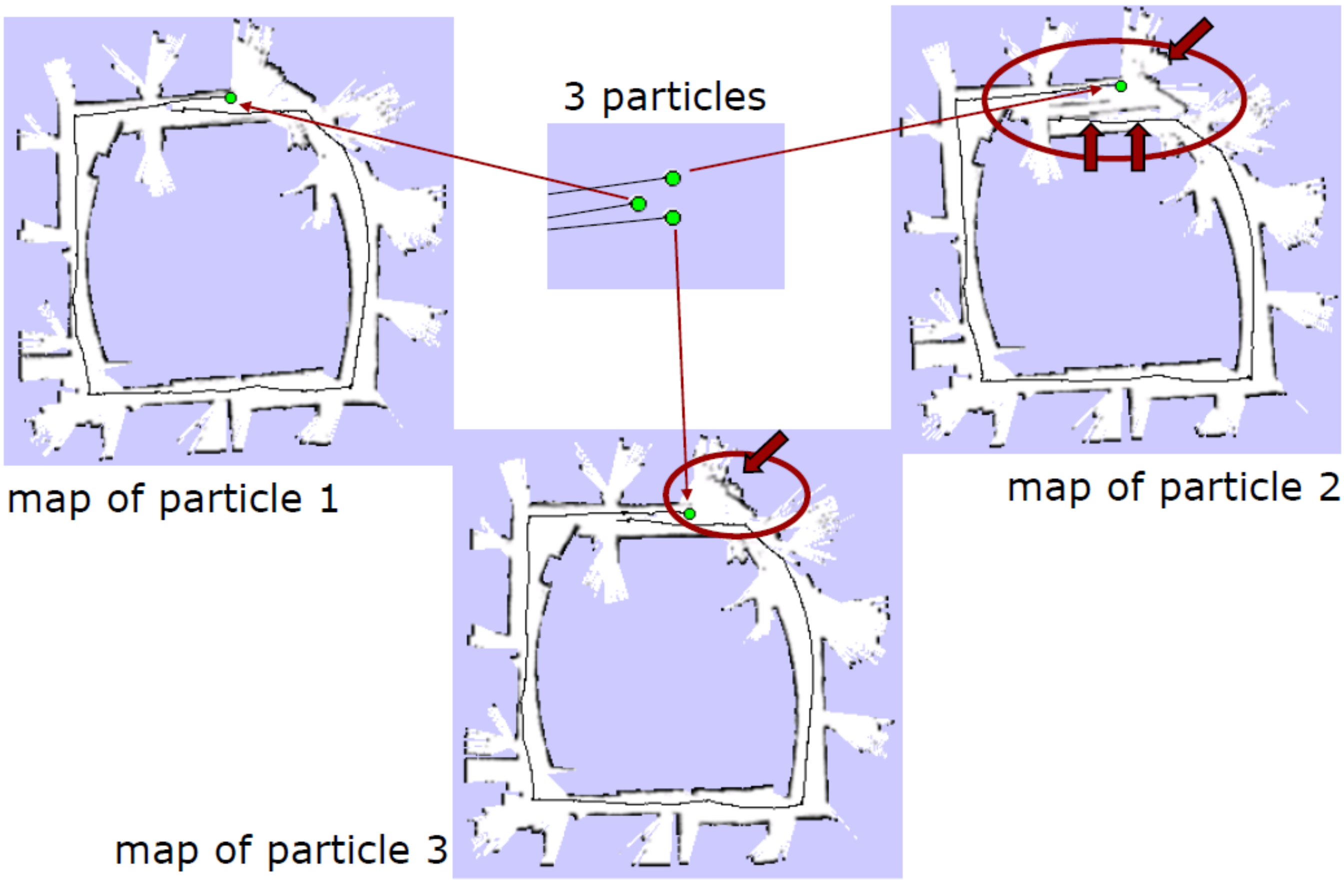
$$p(x, m \mid z, u)$$

$$= \; p(m \mid x, z, \cancel{u})\, p(x \mid z, u)$$

Mapping with known poses

Particle filter representing trajectory hypotheses

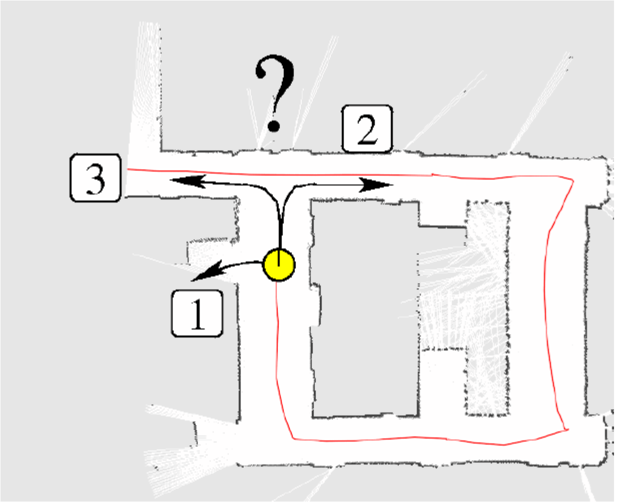
Particle Filter for Mapping



Combining Exploration and SLAM

- The previous approaches are purely passive
- By reasoning about control, the mapping process can be made much more effective

- Question: Where to move next?

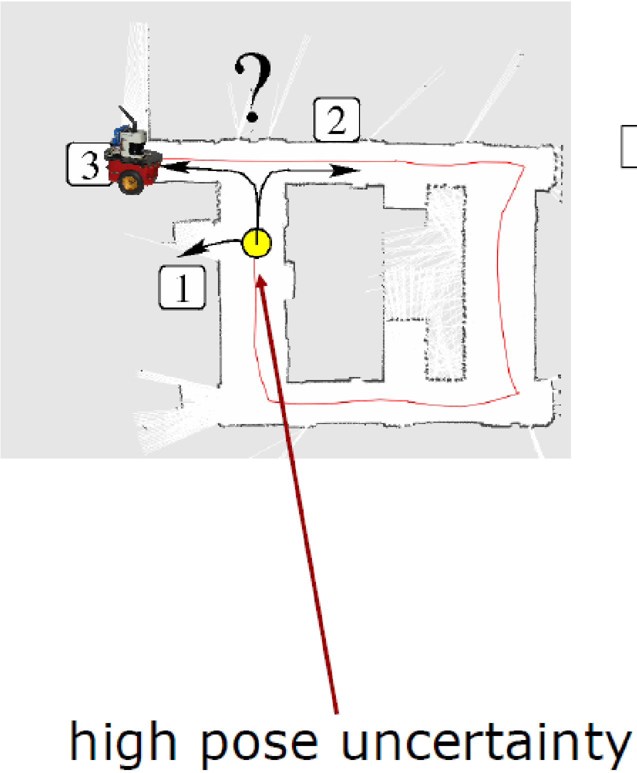


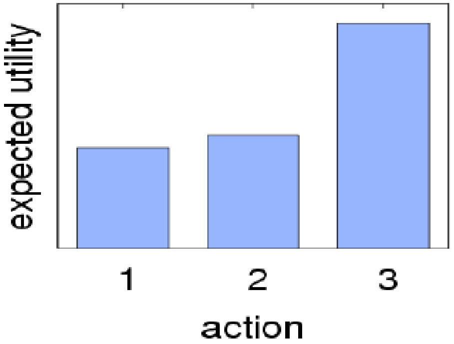
Decision Theoretic Approach

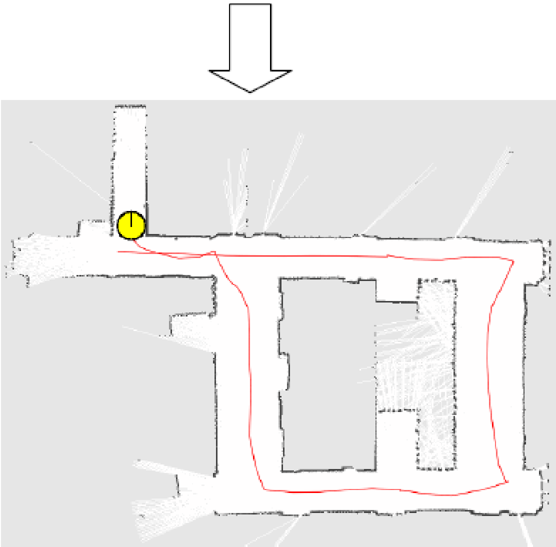
- Learn the map using a Rao-Blackwellized particle filter
- Consider a set of potential actions
- Apply an exploration approach that minimizes the overall uncertainty

Utility = uncertainty reduction - cost

Exploration Problem







Uncertainty of a Posterior

- Entropy is a general measure for the uncertainty of a posterior

$$\begin{aligned} H(p(x)) &= -\int_x p(x) \log p(x)\, dx \\ &= E_x[-\log(p(x))] \end{aligned}$$

- Information Gain = Uncertainty Reduction

$$I(t+1 \mid t) = H(p(x_t)) - H(p(x_{t+1}))$$

Entropy Computation

$$
\begin{aligned}
& H(p(x,y)) \\
= \ & E_{x,y}[-\log p(x,y)] \\
= \ & E_{x,y}[-\log(p(x)\, p(y \mid x))] \\
= \ & E_{x,y}[-\log p(x)] + E_{x,y}[-\log p(y \mid x)] \\
= \ & H(p(x)) + \int_{x,y} -p(x,y) \log p(y \mid x)\; dx\; dy \\
= \ & H(p(x)) + \int_{x,y} -p(y \mid x) p(x) \log p(y \mid x)\; dx\; dy \\
= \ & H(p(x)) + \int_x p(x) \int_y -p(y \mid x) \log p(y \mid x)\; dy\; dx \\
= \ & H(p(x)) + \int_x p(x) H(p(y \mid x))\; dx
\end{aligned}
$$

Computing Map and Pose Uncertainty

data (laser and odometry)

$$
\begin{aligned}
& H(p(x, m \mid d)) \\
= \quad & H(p(x \mid d)) + \int_x p(x \mid d) H(p(m \mid x, d))\, dx \\
\approx \quad & H(p(x \mid d)) + \sum_{i=1}^{\#particles} \omega^{[i]} H(p(m^{[i]} \mid x^{[i]}, d))
\end{aligned}
$$

trajectory uncertainty

particle weight

map uncertainty

Computing Entropy of the Map Posterior

Occupancy Grid map *m*:

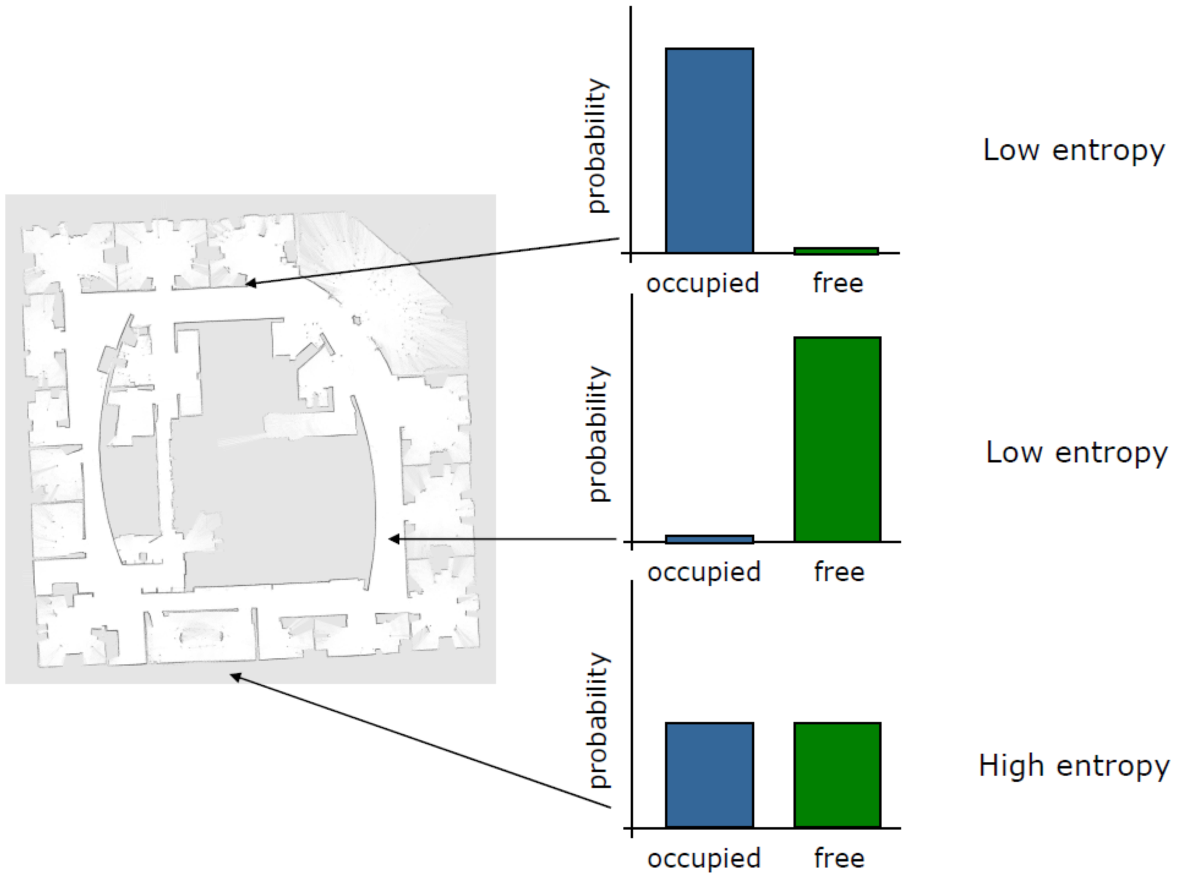
$$H(p(m)) \; = \; -\sum_{c \in m} p(c) \log p(c) + (1 - p(c)) \log(1 - p(c))$$

map uncertainty

grid cells

probability that the cell is occupied

Map Entropy

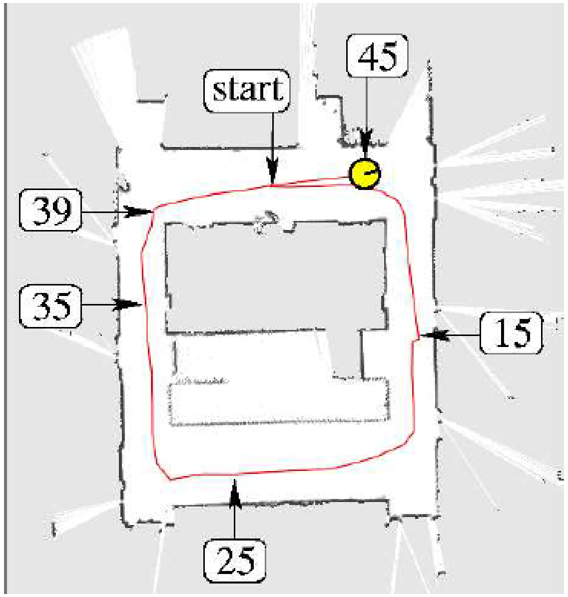


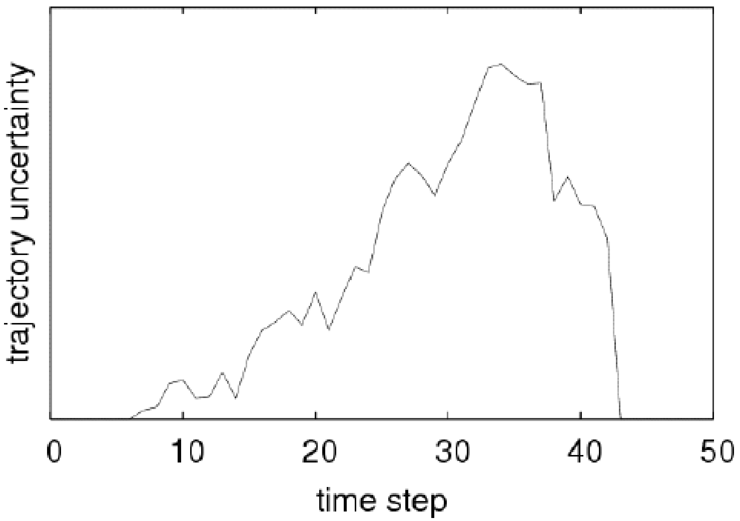
The overall entropy is the sum of the individual entropy values

Trajectory Posterior Entropy

Average pose entropy over time:

$$H(p(x_{1:t} \mid d)) \approx \frac{1}{t} \sum_{t'=1}^{t} H(p(x_{t'} \mid d))$$





Information Gain

- The reduction of entropy in the model

observations
to be obtained

action

H before action
is carried out

$$
\begin{aligned}
I(\hat{z}, a) = {} \\
& H(p(m, x \mid d)) - \\
& H(p(m, x, \hat{x} \mid d, a, \hat{z}))
\end{aligned}
$$

H after action is
carried out

new poses introduced
by action

Computing Expected Information Gain

- To compute the information gain one needs to know the observations obtained when carrying out an action
- This quantity is not known! Reason about potential measurements

$$E[I(a)] = \int_{\hat{z}} p(\hat{z} \mid a, d) \cdot I(\hat{z}, a) \, d\hat{z}$$

The Utility

- To take into account the cost of an action, we compute a utility

$$U(a) \;=\; I(a) - \alpha \cdot cost(a)$$

- Select the action with the highest expected utility

$$a^* \;=\; \underset{a}{\operatorname{argmax}} \{E[U(a)]\}$$

Q2: In Terms of I-Spaces

When there are sensors, planning naturally lives in an *information space*.

We need to develop:

- Formulations of sensor models, I-spaces
- Models of complexity, computation over I-spaces
- Sampling-based planning methods
- Combinatorial planning methods

For C-spaces, the early steps were already done (Lagrangian mechanics).

History of Information Spaces

Where have *information spaces* arisen?

Early appearance of concept: H. Kuhn, 1953

- **Extensive form games**
  Unknown state information regarding other players.
- **Stochastic control theory**
  Disturbances in prediction and measurements cause imperfect state information.
- **Robotics/AI**
  Uncertainty due to limited sensing.

Alternative names: belief states, knowledge states, hyperstates

What is a Sensor?

Light-dependent resistor

GPS unit

Wireless card

Toilet float mechanism

We know it when we see it, but will not try to formally classify.

What is a Sensor, again?

- *Transfer function* converts physical phenomenon to sensor reading:
$$g : \mathbb{R} \rightarrow \mathbb{R}.$$

- Domain of g may be *absolute* vs. *relative*.
- g itself may be *linear* or *nonlinear*.
- *Resolution* is given by set of possible $g(x)$.

- *Sensitivity* is set of stimuli that produce same reading.
- *Repeatability* is producing same readings under same phenomena.
- *Calibration* eliminates systematic errors.

You will find these notions in sensor handbooks.

Physical vs. Virtual Sensors

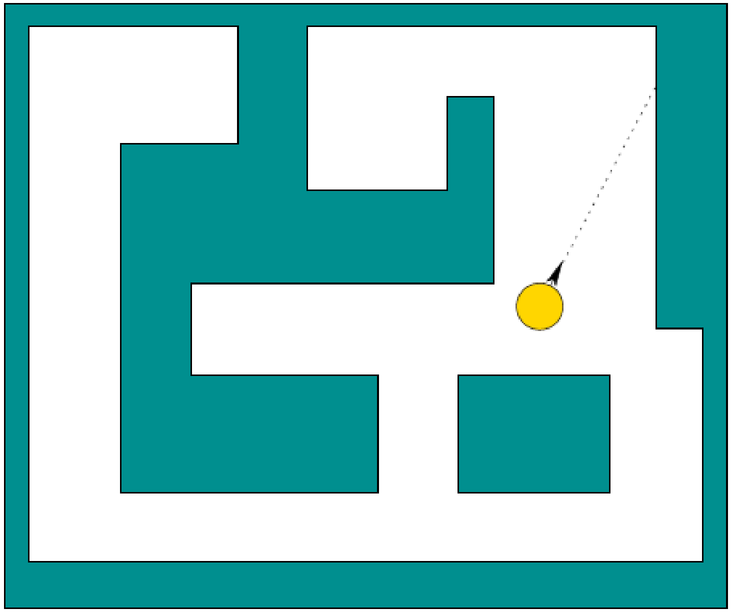
Physical sensor: The real thing.

Virtual sensor: Mathematical model of information obtained from a sensing system.

A virtual sensor could have many alternative physical-sensor implementations.

Identifying which *virtual* sensor is required will lead to better filter design and planning algorithms.

Consider this Mobile Robot



- Observation: The wall is 3 meters away.
- What possible external physical worlds are consistent with that?

Problem Structure

- Localization only: Set of possible configurations
- Mapping only: Set of possible environments
- Both: Set of configuration-environment pairs

Let $\mathcal{Z}$ be any set of sets.

Each $Z \in \mathcal{Z}$ is a “map” .
Each $z \in Z$ is the configuration or “place” in the map.

Unknown configuration and map yields a state space as:
All (z, Z) such that $z \in Z$ and $Z \in \mathcal{Z}$.

State Space for Planar Mobile Robot

Without any obstacles:

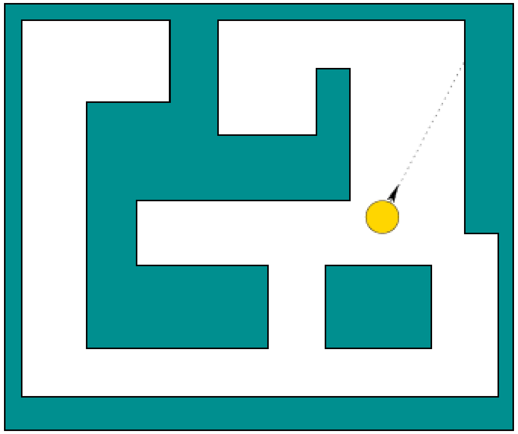
- Any position $(q_x, q_y) \in \mathbb{R}^2$
- Any orientation $q_\theta \in [0, 2\pi)$
- Let *state space* X be all positions and orientations

Can imagine $X \subset \mathbb{R}^3$; however, for orientation, we have additional topology since $q_\theta = 0 = 2\pi$.

Could write $X = \mathbb{R}^2 \times S^1$, in which S^1 is a circle and the set of all orientations.

Could write $X = SE(2)$, set of all 2D rigid-body transformations.

State Space given a Map



Suppose $E \subset \mathbb{R}^2$ is known to be the set of allowable positions.

Must have $(q_x, q_y) \in E$.

State space: $X = E \times S^1$

State Space for One of Many Maps

Given a set of k possible maps:

$$\mathcal{E} = \{E_1, E_2, \ldots, E_k\}$$

For example, could be given 5 maps:

$$\mathcal{E} = \{E_1, E_2, E_3, E_4, E_5\}$$

X is all (q, E_i) in which $(q_x, q_y) \in E_i$ and $E_i \in \mathcal{E}$.

Recall the common structure.

State Space for Unknown Map

Given an infinite *map family*, $\mathcal{E}$, of environments.

Examples:

- The set of all connected, bounded polygonal subsets that have no interior holes (formally, they are *simply connected*).
- The previous set expanded to include all cases in which the polygonal region has a finite number of polygonal holes.
- All subsets of $\mathbb{R}^2$ that have a finite number of points removed.
- All subsets of $\mathbb{R}^2$ that can be obtained by removing a finite collection of nonoverlapping discs.
- All subsets of $\mathbb{R}^2$ obtained by removing a finite collection of nonoverlapping convex sets.
- A collection of piecewise-analytic subsets of $\mathbb{R}^2$.

State Space for Unknown Map

In spite of larger $\mathcal{E}$, there is no difference:

X is all pairs (q, E) in which $(q_x, q_y) \in E$ and $E \in \mathcal{E}$.

We can write $X \subset \mathbb{R}^2 \times S^1 \times \mathcal{E}$.

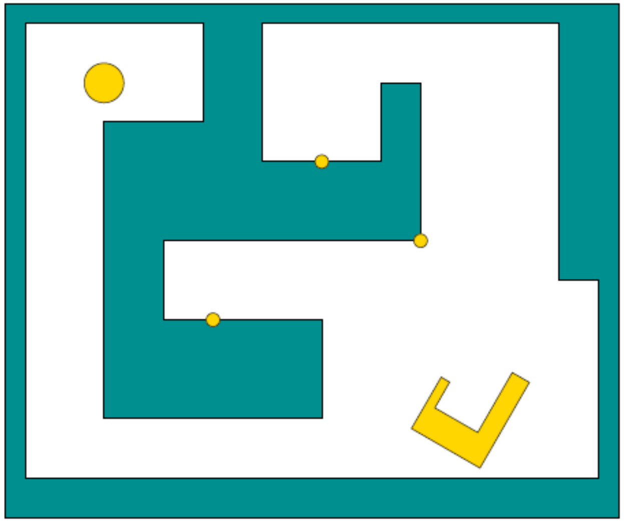
X is enormous! But that is fine here. We do not compute directly on it.

Note: Putting useful probability densities over X might be difficult or impossible.

X is usually **not a manifold** (doesn't look like C-space)

Placing Bodies into Environments

Place a *body* B into E.



Each could have a configuration space $SE(2)$, so that we transform it: $B(q_x, q_y, q_\theta) \subset E$.

Sensor Mapping

Let X be any physical state space.

Let Y denote the *observation space*, which is the set of all possible sensor observations.

A virtual sensor is defined by a *sensor mapping*:

$$h : X \to Y.$$

Note similarity to transfer function for physical sensors.

When $x \in X$, the sensor instantaneously observes $y = h(x) \in Y$.

Sensor Mapping: Extreme Examples

The weakest possible sensor

DUMMY SENSOR:
$Y = \{0\}$ and $h(x) = 0$ for all $x \in X$

The strongest possible sensor(s)

IDENTITY SENSOR:
$Y = X$ and $y = h(x) = x$
Just give me the state!

BIJECTIVE SENSOR:
h is bijective function from X to Y.
x can be reconstructed as $x = h^{-1}(y)$.

Projection Sensor

PROJECTION SENSOR:
Choose some components of X.

$X = \mathbb{R}^3$ and $x = (x_1, x_2, x_3) \in X$.

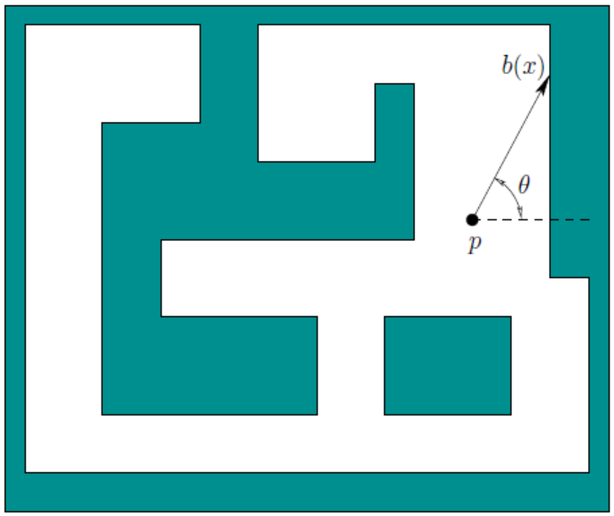
$Y = \mathbb{R}^2$

$y = h(x) = (x_1, x_2)$

$X = \mathbb{R}^2 \times S^1$
A state is $(q_x, q_y, q_\theta) \in X$.

Position sensor: Observes (q_x, q_y) and leaves q_θ unknown.
Ideal compass: Observes q_θ and leaves q_x and q_y unknown.

More Interesting: Directional Depth



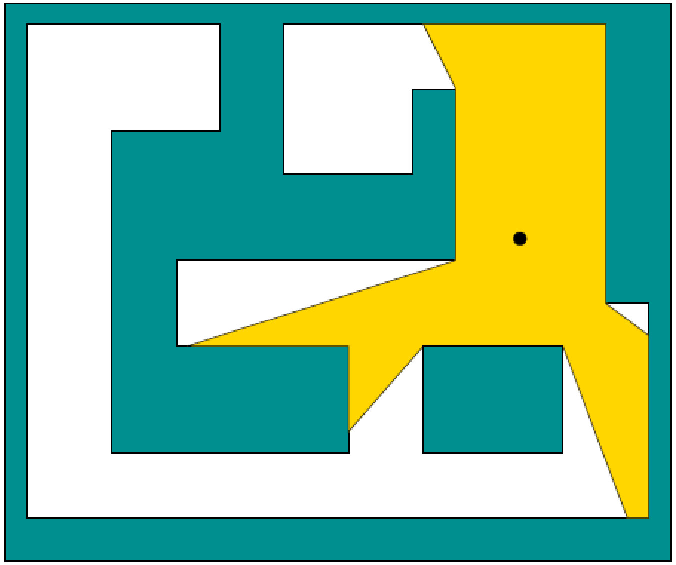
DIRECTIONAL DEPTH SENSOR:

$$h_d(p, \theta, E) = \|p - b(x)\|$$

Let $p = (q_x, q_y)$ and $\theta = q_\theta$ (shorthand notation)
$b(x)$ is point on boundary ∂E hit by ray.

Omnidirectional Version

Like an infinite-dimensional vector of observations

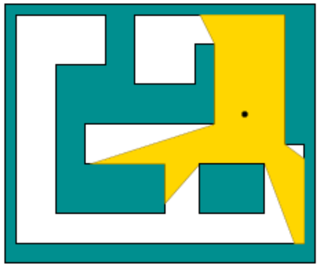


OMNIDIRECTIONAL DEPTH SENSOR:

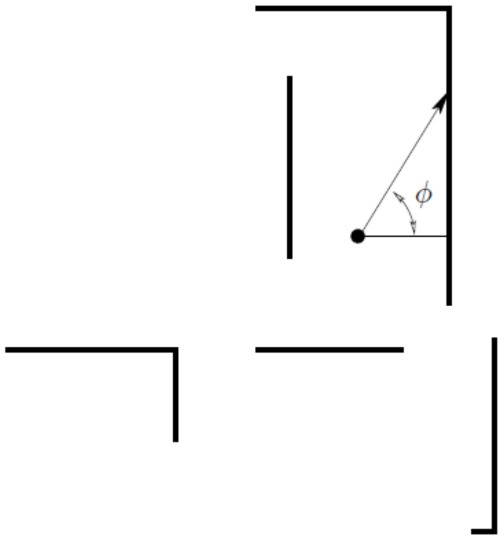
$h_{od}(x) = y$, in which $y : S^1 \to [0, \infty)$

$$y(\phi) = h_{od\phi}(p, \theta, E).$$

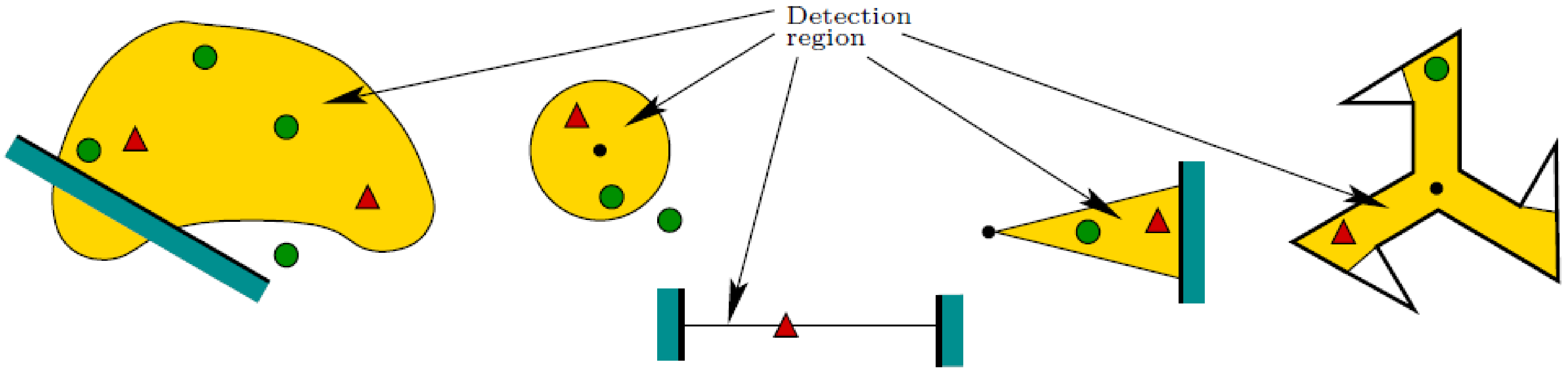
Understanding the Omnidirectional Sensor



How does the observation $y : S^1 \rightarrow [0, \infty)$ look?



New Category: Detection Sensor



Is a body in the field of view, or *detection region*?

Relational Sensors

Consider any relation R on the set of all bodies.

For a pair of bodies, B_1 and B_2, examples of $R(B_1, B_2)$ are:

- B_1 is in front of B_2
- B_1 is to the left of B_2
- B_1 is on top of B_2
- B_1 is closer than B_2
- B_1 is bigger than B_2.

More precisely, Let $R_x(i, j)$ mean B_i is related to B_j, when the system is at state x.

Idea is due to Guibas

Gap Sensor

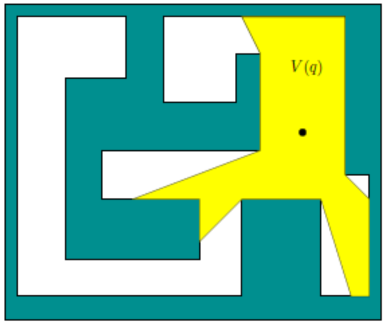
Report information obtained along the boundary of $V(q)$, which is denoted as $\partial V(q)$

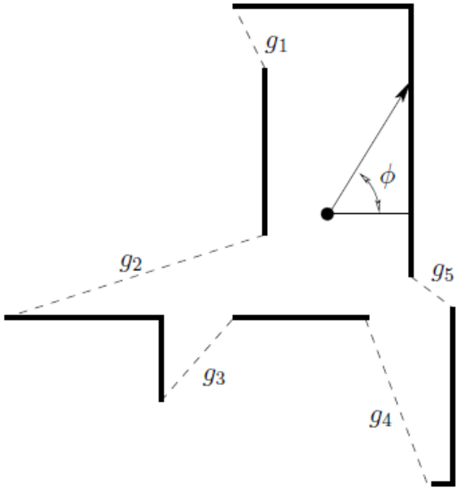
Two qualitatively different parts of $\partial V(q)$:

1. A piece of a body boundary
2. A gap (discontinuity in depth)

A gap sensor reports how these parts alternate.

Simple Gap Sensor





SIMPLE GAP SENSOR:

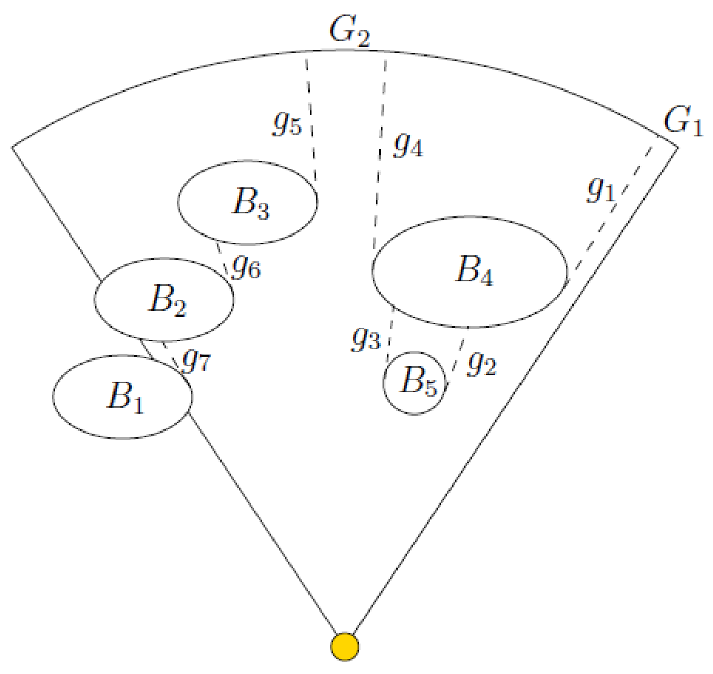
Alternating between boundary and gaps:

$$y = (B_0, g_1, B_0, g_2, B_0, g_3, B_0, g_4, B_0, g_5)$$

Equivalently:

$$y = (g_1, g_2, g_3, g_4, g_5)$$

Multibody Gap Sensing



MULIBODY GAP SENSOR:

$$y = (G_1, g_1, B_4, g_2, B_5, g_3, B_4, g_4, G_2, g_5, B_3, g_6, B_2, g_7, B_1)$$

So what?!

Can we Build “Filters”?

There are two general kinds of filters:

1. **Spatial:** Combining simultaneous observations from multiple sensors.

2. **Temporal:** Incrementally incorporating observations from a sensor at discrete stages.

Of course, we can make spatio-temporal filters.

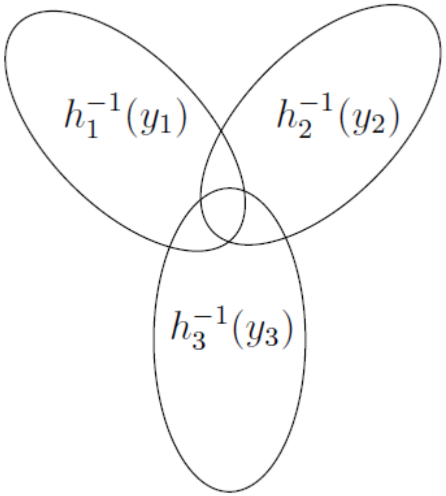
Triangulation: Preimage Intersection

Consider any n sensor mappings $h_i : X \to Y_i$ for i from 1 to n.

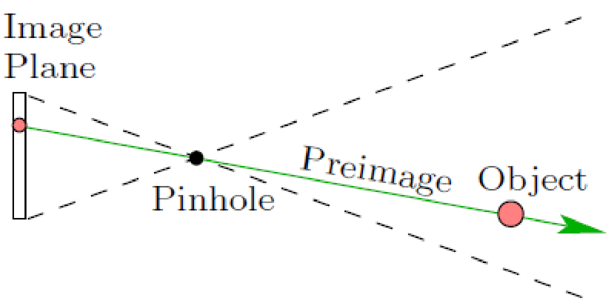
The *triangulation* of a set of the observations $y_1, \ldots, y_n$ is:

$$\Delta(y_1, \ldots, y_n) = h_1^{-1}(y_1) \cap h_2^{-1}(y_2) \cap \cdots \cap h_n^{-1}(y_n),$$

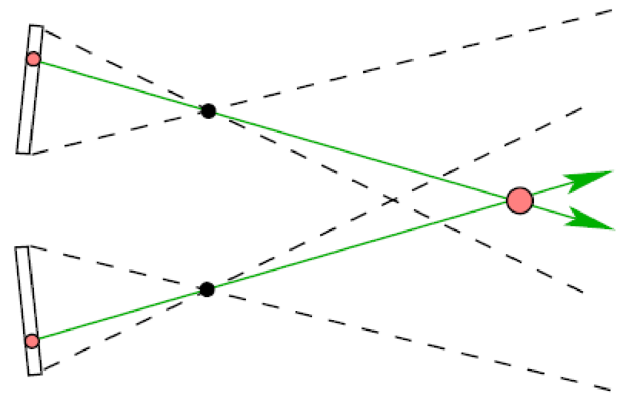
which is a subset of X.



Triangulation in Stereo Vision



One camera



Triangulation

Observation: Object location in image plane

Preimages: Infinite rays

Triangulation: $\Delta(y_1, y_2)$ is a point.

Relation to Linear Algebra

Precisely how does information improve from multiple observations?

Linear case: $y_i = C_i x$, with $Y = \mathbb{R}^{m_i}$ and $X = \mathbb{R}^n$.
Assume C_i has rank k.

Each $h_i^{-1}(y_i)$ is a $n-k$-dimensional hyperplane through the origin of X.

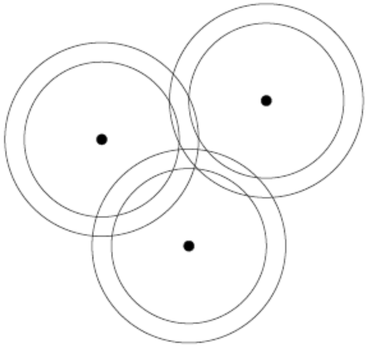
$\Delta(y_1, \ldots, y_n)$ is the intersection of hyperplanes.

Preimage dimension and linear independent are crucial.

Nonlinear case: Similar, but tricky due to geometry.

Handling Disturbances

Nondeterministic disturbances:



Probabilistic disturbances:

$$p(x|y_1, \ldots, y_n) = \frac{p(y_1|x)p(y_2|x)\cdots p(y_n|x)p(x)}{p(y_1, \ldots, y_n)}$$

The *least squares* optimization problem:

$$\min_{\hat{x} \in X} \sum_{i=1}^{n} d_i^2(\hat{x}, y_i)$$

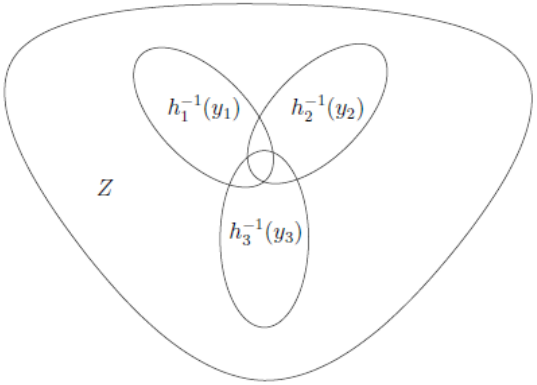
Over State-Time Space

Recall state-time space $Z = X \times T$.

A sensor is $h : Z \to Y$.

Triangulation intersections chunks of state-time space:

$$\Delta(y_1, \ldots, y_n) = h_1^{-1}(y_1) \cap h_2^{-1}(y_2) \cap \cdots \cap h_n^{-1}(y_n),$$



Important example: GPS simultaneously estimates position and time.

Filtering Over Time

Given state space X and sensor $h : X \rightarrow Y$.

Let $\tilde{x} : [0, t] \rightarrow X$ be a state trajectory.

Let $\tilde{y} : [0, t] \rightarrow Y$ be an *observation history*.

When presented with $\tilde{y}$, there are two fundamental questions:

1. What is the set of state trajectories $\tilde{x} : [0, t] \rightarrow X$ that might have occurred?
2. What is the set of possible current states, $\tilde{x}(t)$?

Time Parameterized Sensor Mapping

Apply $h : X \rightarrow Y$ for every $t' \in [0, t]$.

Every $t' \in [0, t]$ yields some observation $\tilde{y}(t') = h(\tilde{x}(t'))$.

Let $\tilde{X}$ be all state trajectories.

Let $\tilde{Y}$ be all possible observation histories.

Applying h over $[0, t]$, we obtain the induced map:

$$H : \tilde{X} \rightarrow \tilde{Y}$$

Answers to Our Questions

This preimage answers 1st question:

$$H^{-1}(\tilde{y}) = \{\tilde{x} \in \tilde{X} \mid \tilde{y} = H(\tilde{x})\}$$

"all state trajectories that could have produced $\tilde{y}$"

Answer to 2nd question:

$$\{x \in X \mid \exists \tilde{x} \in H^{-1}(\tilde{y}) \text{ such that } \tilde{x}(t) = x\}$$

"all possible current states, considering the history $\tilde{y}$"

Moving On: Nondeterministic Filters

Models: $h : X \rightarrow \mathrm{pow}(Y)$ and $F(x_k, u_k) \subseteq X$

The I-space: $\mathcal{I}_{ndet} = \mathrm{pow}(X)$

Initial I-state: $X_1 \subseteq X$

The filter:

$$X_{k+1}(\eta_{k+1}) = \phi(X_k(\eta_k), u_k, y_{k+1})$$

After first observation y_1:

$$X_1(\eta_1) = X_1(y_1) = X_1 \cap h^{-1}(y_1)$$

(Intersect initial constraint with observation preimage.)

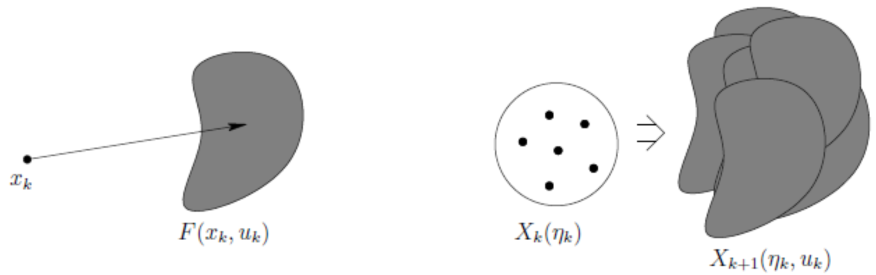
Operation of Nondeterministic Filters

Inductively, $X_k(\eta_k)$ is given.

Determine $X_{k+1}(\eta_{k+1})$ using $X_k(\eta_k)$, u_k, and y_{k+1}.

Using u_k,

$$X_{k+1}(\eta_k, u_k) = \bigcup_{x_k \in X_k(\eta_k)} F(x_k, u_k).$$



Using y_{k+1},

$$X_{k+1}(\eta_{k+1}) = X_{k+1}(\eta_k, u_k, y_{k+1}) = X_{k+1}(\eta_k, u_k) \cap h^{-1}(y_{k+1}).$$

Combinatorial Filters

Now we attempt to reduce filter complexity.

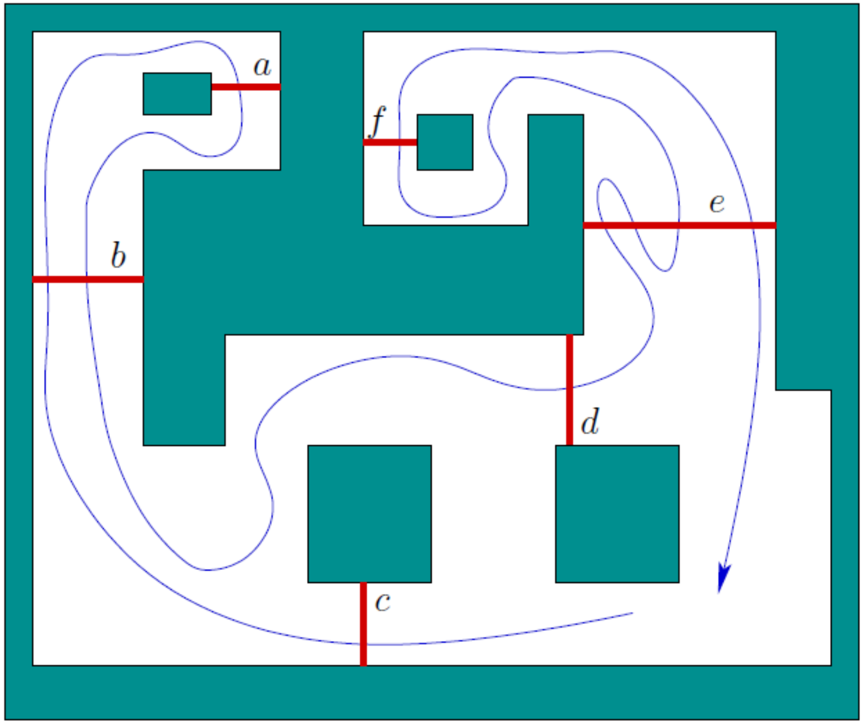
Introducing *combinatorial filters*

Three examples:

1. Obstacles and beams
2. Shadow information spaces
3. Gap navigation trees

Many, many more should be possible from the numerous virtual sensor models already given.

Obstacles and Beams



A point body moves in a known environment.

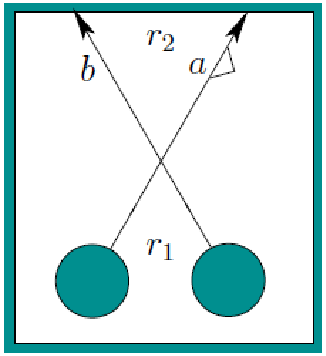
$X = E \subset \mathbb{R}^2$ and $\tilde{y} = cbabdeeefe$

What state trajectories are possible?

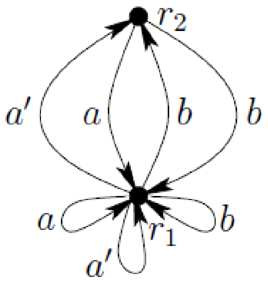
Multigraph Representation

Let G be a multigraph:

- There is one *vertex* for every $r \in R$.
- A *directed edge* is made from $r_1 \in R$ to $r_2 \in R$ if and only if the body can cross a single beam to go from r_1 to r_2.
- Each edge is labeled with the beam label and the direction, if needed.



Two beams



The multigraph G

Nondeterministic Region Filter

Let $\mathcal{I} = \mathrm{pow}(R)$ and $\iota_0 = R_0$, an initial region set.

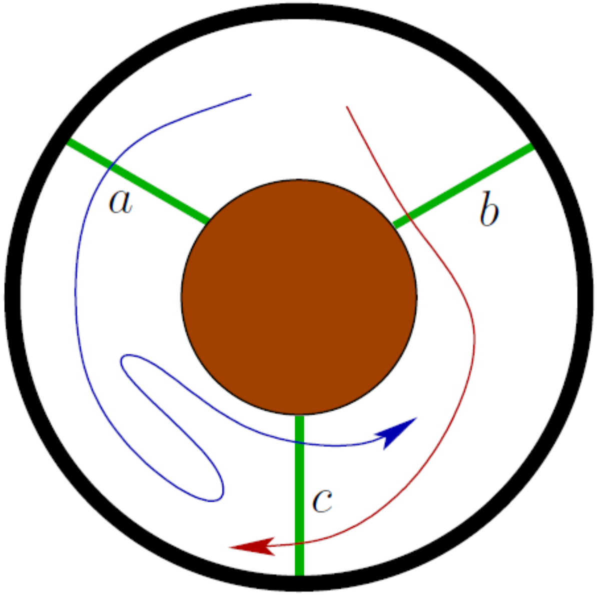
Filter:

$$R_{k+1} = \phi(R_k, y_{k+1})$$

In particular:

1. Let $k = 0$ and $R_k = R_0$.
2. Let $R_{k+1} = \emptyset$.
3. For vertex in R_k and outgoing edge that matches y_{k+1}, insert the destination vertex/region into R_{k+1}.
4. Increment k, and go to Step 2.

Two Bodies



In a given annulus E, we have two bodies, yielding $X = E^2 \subset \mathbb{R}^4$.

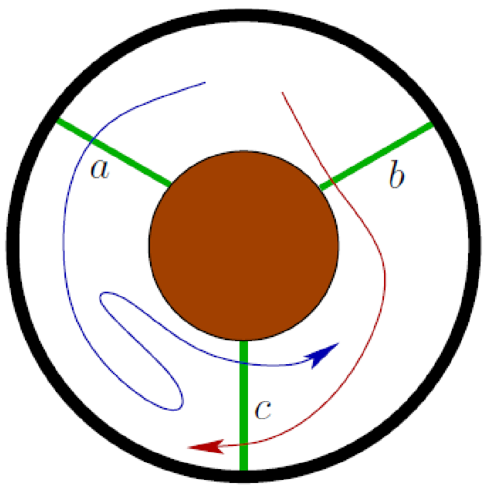
There are three disjoint, distinguishable, undirected beams a, b, c.

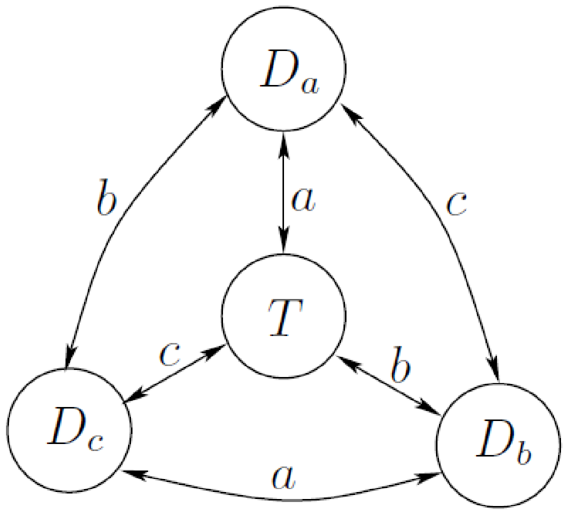
There are 3 regions, and nine combinations:
$(1,1)$, $(1,2)$, $(1,3)$, $(2,1)$, $(2,2)$, $(2,3)$, $(3,1)$, $(3,2)$, and $(3,3)$

Two-bit Filter

Use a task to reduce complexity MUCH further.

Task: Determine whether the bodies in a room *together*?





The previous I-space would have 511 I-states.

Here, the I-space is: $\mathcal{I} = \{T, D_a, D_b, D_c\}$

Filter: $\iota_k = \phi(\iota_{k-1}, y_k)$

Multi-Body Filter

What if more than one body move around?
For n bodies, $X \subseteq \mathbb{R}^{2n}$.

Let $R^n = R \times R \times \cdots \times R$

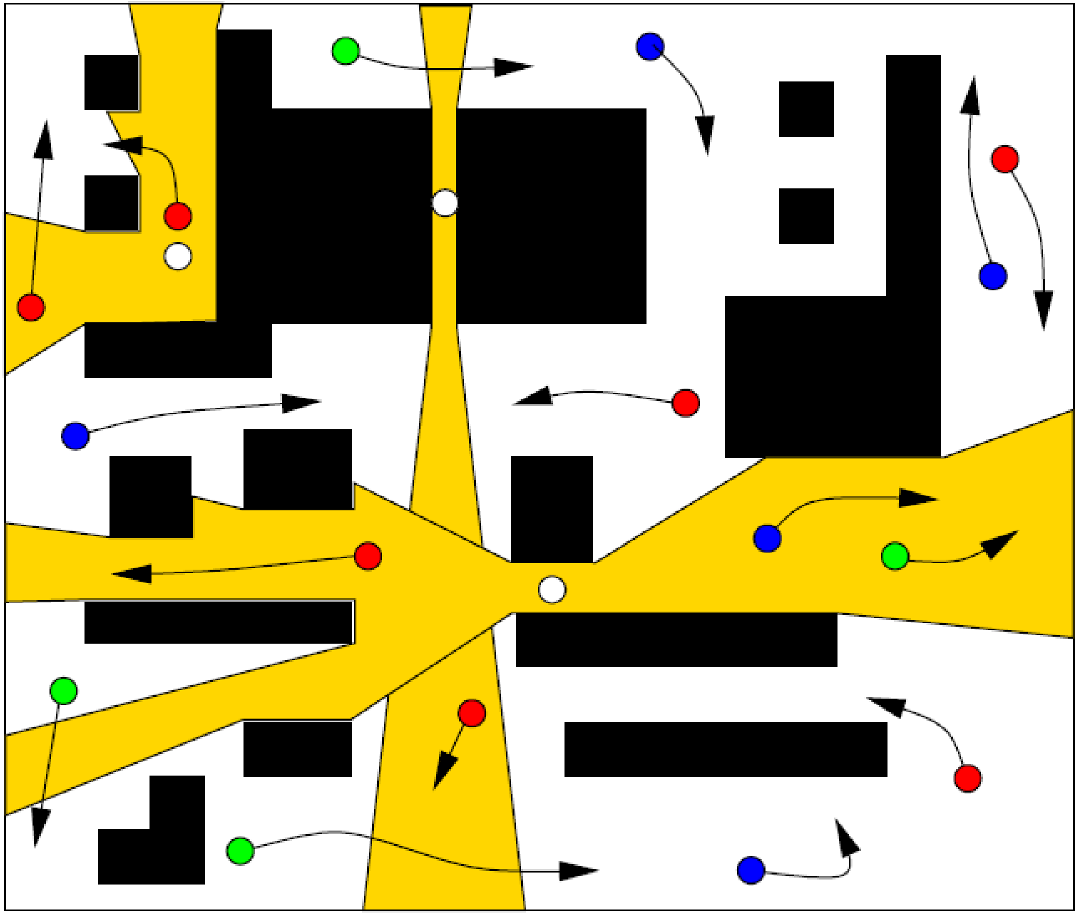
I-space: $\mathcal{I} = \mathrm{pow}(R^n)$

Compute the multigraph G, and form a product G^n.

Vertices of G^n are region assignments $(r_1, \ldots, r_n)$.
Edges of G^n correspond to possible transitions.

Extend the one-body filter directly to G^n.
Problem: Number of vertices is exponential in n.

Challenge



Q3: Asymmetry in Strategic Settings

- A big issue! Occurs in numerous robotics problems such as human-robot interactions
- Modelling this is an on-going challenge
- Some model from social sciences, e.g., market for lemons
  - Decisions with ‘quality uncertainty’
  - One person (seller) knows more than another (buyer)
  - What will be interaction look like? What should they do?

Acknowledgement

Parts of this lecture are based on:

- Lecture slides due to W. Burgard et al. at U. Freiburg
- Tutorial at IROS 2009 by S. LaValle (related reference: S. LaValle, “Filtering and Planning in Information Spaces”, UIUC Tech Report 2009)