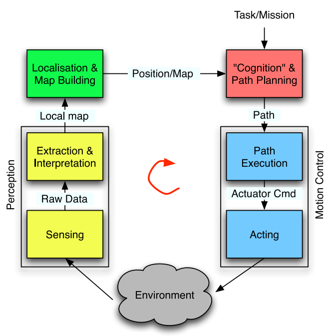
Robot Localization

Henrik I. Christensen

Robotics and Intelligent Machines @ GT
College of Computing
Georgia Institute of Technology
Atlanta, GA
hic@cc.gatech.edu

February 5, 2008

The Robot Structure



Outline

1. Introduction
2. Representations
3. Prediction
4. Updating
   - Topological Pose Estimation
   - Gaussian PDF
   - Sum of Gaussian PDF
   - Probability Grids
   - Monte-Carlo Methods
5. Examples
6. Wrap-up

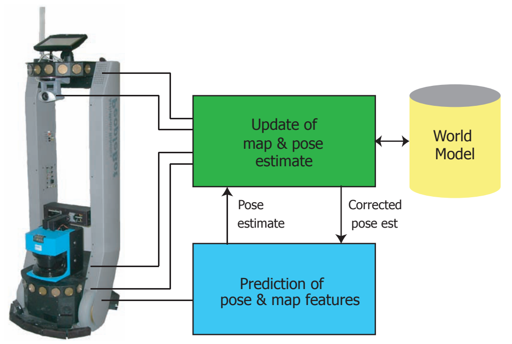
Introduction

- A fundamental part of mobile robotics
- Two problems:
  1. Pose initialisation / Kidnapped Robot
     - Initialising the system / recovery
  2. Pose Maintenance
     - Updating pose as the robot moves about

Pieces for the puzzle

- A model for the robot
- Model of the environment
  - Representation of environment
- A method for feature extraction
- A strategy to match features to the model
- A method to update the pose estimate

The basic pieces in localisation



The issues involves

- Managing the uncertainty for the pose estimate
  - 1 vs Many hypotheses for pose
- Selecting a model for the environment
- Modelling of the system
- Efficient Implementation(s)

Approach

- Representations
- Uncertainty in vehicle model
- Updating the pose estimate
  - Selection of different models for uncertainty

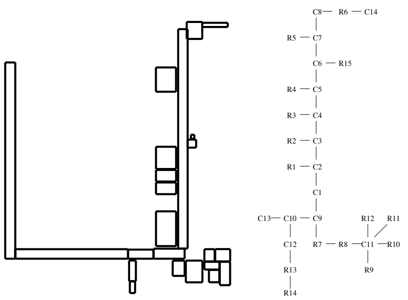
Outline

1. Introduction
2. Representations
3. Prediction
4. Updating
   - Topological Pose Estimation
   - Gaussian PDF
   - Sum of Gaussian PDF
   - Probability Grids
   - Monte-Carlo Methods
5. Examples
6. Wrap-up

Representation

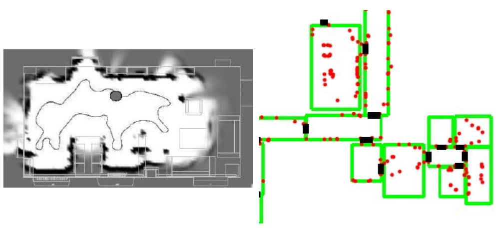
- Already covered in lecture on sensors and features
- A number of different options:
  1. Topological maps
  2. Feature maps
  3. Probability Grid
  4. Appearance based/raw models

Topological Map Example

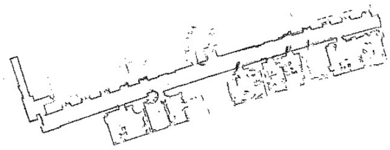


(a) Metric map **(b)** Topologic map

Metric Maps: Grids and Features



Appearance / Raw Sensor Map Example



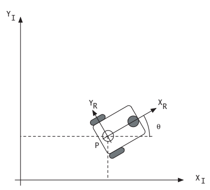
Outline

1. Introduction
2. Representations
3. Prediction
4. Updating
   - Topological Pose Estimation
   - Gaussian PDF
   - Sum of Gaussian PDF
   - Probability Grids
   - Monte-Carlo Methods
5. Examples
6. Wrap-up

Prediction of vehicle motion

- Consider the process as sense-move-sense-....
- As part of the movement step there is a need to estimate the new pose of the robot and the associated uncertainty in the position.
- Prediction is based entirely on odometric information and a model of the robot as discussed in kinematic modelling.
- Uncertainty is modelled by covariance propagation as discussed in under sensing

Model for differential drive robot



- Assume a pose estimate of

$$p = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}$$

- A motion of left and right wheel by Δs_l and Δs_r respectively. The distance between the wheel is assumed to be $2l$

Pose prediction – Differential drive robot

- Consequently:

$$
\begin{aligned}
\Delta s &= \frac{\Delta s_l + \Delta s_r}{2} \\
\Delta \theta &= \frac{\Delta s_r - \Delta s_l}{2l} \\
\Delta x &= \Delta s \cos\left(\theta + \frac{\Delta\theta}{2}\right) \\
\Delta y &= \Delta s \sin\left(\theta + \frac{\Delta\theta}{2}\right)
\end{aligned}
$$

Pose prediction – Differential drive robot

- Or in condensed form:

$$
p' = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} + \begin{bmatrix} \Delta s \cos\left(\theta + \frac{\Delta\theta}{2}\right) \\ \Delta s \sin\left(\theta + \frac{\Delta\theta}{2}\right) \\ \frac{\Delta s_r - \Delta s_l}{2l} \end{bmatrix}
$$

Pose Prediction – Uncertainty estimate

- Need to provide an estimate of uncertainty in position $\Sigma_{p'}$
- Assume the initial uncertainty is Σ_p
- Assume motion uncertainty is

$$
\Sigma_\Delta = \begin{bmatrix} k_r|\Delta s_r| & 0 \\ 0 & k_l|\Delta s_l| \end{bmatrix}
$$

- Assumption of uncertainty, and proportional to distance travelled.
- k_i is determined by calibration

Pose prediction – Uncertainty estimate

- Update can be generate by covariance propagation, i.e.:

$$
\Sigma_{p'} = \nabla_p f \Sigma_p \nabla_p f^T + \nabla_{\Delta_{rl}} f \Sigma_\Delta \nabla_{\Delta_{rl}} f^T
$$

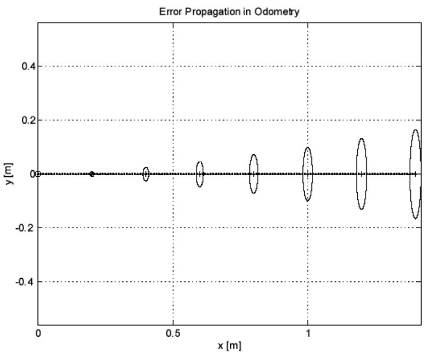
- Where $\nabla_p f$ and $\nabla_{\Delta_{rl}} f$ is the Jacobians for the pose and the motion, respectively.

Pose Updating – Pose/Motion Jacobian

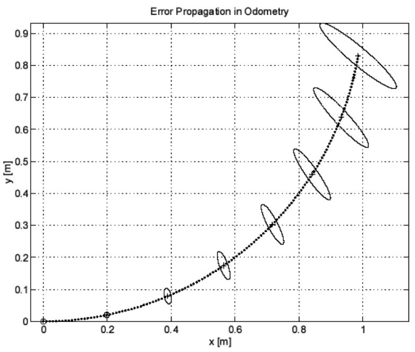
$$\nabla_p f = \begin{bmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial y} & \frac{\partial f}{\partial \theta} \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\Delta s \sin\left(\theta + \frac{\Delta\theta}{2}\right) \\ 0 & 1 & \Delta s \cos\left(\theta + \frac{\Delta\theta}{2}\right) \\ 0 & 0 & 1 \end{bmatrix}$$

$$\nabla_{\Delta_{rl}} = \begin{bmatrix} \frac{1}{2}\cos\left(\theta + \frac{\Delta\theta}{2}\right) - \frac{\Delta s}{4l}\sin\left(\theta + \frac{\Delta\theta}{2}\right) & \frac{1}{2}\cos\left(\theta + \frac{\Delta\theta}{2}\right) + \frac{\Delta s}{4l}\sin\left(\theta + \frac{\Delta\theta}{2}\right) \\ \frac{1}{2}\sin\left(\theta + \frac{\Delta\theta}{2}\right) - \frac{\Delta s}{4l}\cos\left(\theta + \frac{\Delta\theta}{2}\right) & \frac{1}{2}\sin\left(\theta + \frac{\Delta\theta}{2}\right) + \frac{\Delta s}{4l}\cos\left(\theta + \frac{\Delta\theta}{2}\right) \\ \frac{1}{4l} & \frac{1}{4l} \end{bmatrix}$$

Pose prediction – Linear Motion Example



Pose prediction – Circular Motion Example



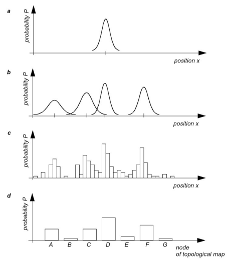
Outline

1. Introduction
2. Representations
3. Prediction
4. Updating
   - Topological Pose Estimation
   - Gaussian PDF
   - Sum of Gaussian PDF
   - Probability Grids
   - Monte-Carlo Methods
5. Examples
6. Wrap-up

Pose Updating

- Given an estimate of where the robot might be p'
- Acquire sensor data
- Extract Features
- Match features to model
- Update the pose estimate based on feature information

Pose updating – Uncertainty Model



- The selection of an uncertainty model
  - Single hypothesis
  - Sum of Gaussians
  - Probability grid
  - Topological Graph
  - Particle Based

Pose updating - Uncertainty Model

- The selection of an uncertainty model influences the updating methodology
- The uncertainty model is coupled to the environmental representation
- The model influences strongly the computational requirements

Uncertainty Modelling – Markov Approach

- Assume the world is divided into places/states $s \in P$
- Estimation of $p(s_t)$ given s_{t-1} and sensory data z_t
- Formally

$$p(s_t|z_t) = \int p(s_t|s'_{t-1}, z_t)p(s_{t-1})ds'_{t-1}$$

Integration needed as s_t could be reached from multiple locations

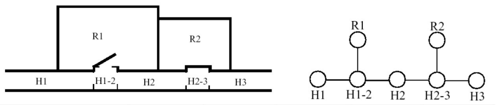
Uncertainty modelling – Markov Approach

- Markov assumption: all knowledge encoded in the pose/state estimate
- There is a probability model for motion updating
- There is a model for $p(z|s)$ i.e. a sensor model, as

$$p(s|z) = \frac{p(z|s)p(s)}{p(z)}$$

  where $p(s)$ is location model and $p(z)$ is the sensor noise model
- These assumptions are relative weak

Topological modelling – dervish example



	Wall	Closed door	Open door	Open hallway	Foyer
Nothing detected	0.70	0.40	0.05	0.001	0.30
Closed door detected	0.30	0.60	0	0	0.05
Open door detected	0	0	0.90	0.10	0.15
Open hallway detected	0	0	0.001	0.90	0.50

Topological modelling – dervish example

- Here the probability updating is used for direct lookup of $p(s|z)$, where s is any of the nodes in the topological map
- As robot moved through environment the graph is updated with new information
- The probability table is small and efficient to handle
- The localisation is coarse (location oriented)

Pose estimation with Gaussian Model

- The pose is approximated by a single Gaussian function

$$p(s) = \frac{1}{\sqrt{2\pi\Sigma_s}} \exp\left(\frac{1}{2}(s-\bar{s})\Sigma_s^{-1}(s-\bar{s})^T\right)^2$$

- s is here a continuous function and Σ_s is the associated uncertainty estimate
- Updating is normally performed using a Kalman filter model

Kalman filter – State space model

$$
\begin{aligned}
s_t &= F s_{t-1} + G u_t + w_t \\
z_t &= H s_t + v_t
\end{aligned}
$$

- where F is the system model, G is the deterministic input, H is a prediction of where features are in the world, w is the system noise, and v is the measurement noise

Detour – Probability Updating

- Assume two measurement x_1 and x_2 with associated uncertainties σ_1 and σ_2. How does one generate an optimal estimate $\hat{x}$?
- Doing a weighted least square

$$S = \sum_{i=1,2} w_i(\hat{x} - x_i)^2$$

what are the optimal weights w_i?

- From $\frac{\partial S}{\partial \hat{x}} = 0$ we get . . .

Detour – Probability Updating

$$\hat{x} = \frac{\sum w_i q_i}{\sum w_i}$$

with $w_i = \frac{1}{\sigma_i^2}$ we get

$$\hat{x} = \frac{\sigma_2^2}{\sigma_1^2 + \sigma_2^2} x_1 + \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2} x_2$$

and

$$\sigma_{\hat{x}} = \frac{\sigma_1^2 \sigma_2^2}{\sigma_1^2 + \sigma_2^2}$$

Detour – Probability Updating

- The update can be rewritten to

$$\hat{x} = x_1 + \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2}(x_2 - x_1)$$

- I.e. the updating = the value + a correction term

The Kalman model

- Composed of a prediction and an update of the estimate
- Prediction is the same as kinematic prediction of pose
- Update is based on the sensory feedback

Kalman Prediction

$$\begin{aligned} s_{t|t-1} &= F s_{t-1|t-1} + G u_t \\ \Sigma_{t|t-1} &= F \Sigma_{t-1|t-1} F^T + Q_t \end{aligned}$$

where Q is the uncertainty of the kinematic motion (odometric uncertainty)

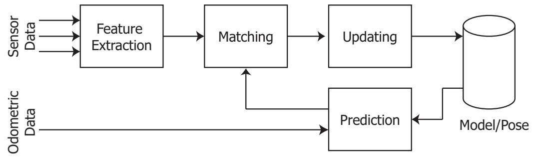
Kalman Updating

$$\begin{aligned} s_{t|t} &= s_{t|t-1} + K_t(z_t - H s_{t|t-1}) \\ K_t &= \Sigma_{t|t-1} H^T S_t^{-1} \\ S_t &= H \Sigma_{t|t-1} H^T + R_t \\ \Sigma_{t|t} &= (I - K_t H)\Sigma_{t|t-1} \end{aligned}$$

Kalman interpretation

- F is the kinematic update
- H is the measurement prediction $p(z|s)$ ie a prediction of where features in the world are in the sensory frame
- Σ is the uncertainty in the pose/state estimate $s_{t|t}$.
- S_t is the uncertainty in the sensory measurements
- R_t is the sensor noise

Use of the Kalman filter

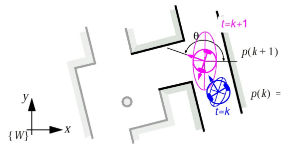


- Prediction / Update as described above
- Matching typically based on the Mahalanobis distance

$$M = z_t S_t^{-1} z_t^T$$

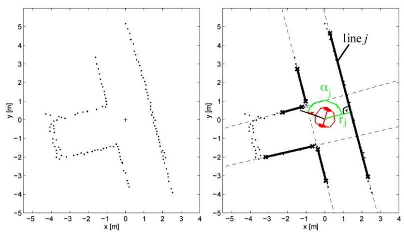
- A validation gate may be used: $M < \rho$

Kalman Example – Prediction

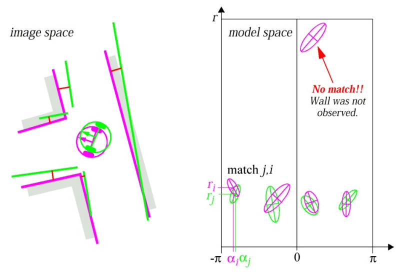


- Kinematic prediction of position and uncertainty
- Standard model

Kalman Example – Features

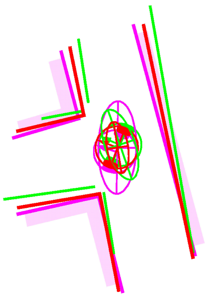


Kalman Example – Matching



using nearest neighbour and predicted features matching is easy

Kalman Example – Updating



- Matched features generates an error in estimates (purple - model), (green - measured), and (red - update)
- Updating is now trivial

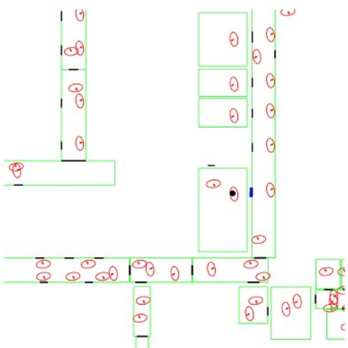
Kalman – Discussion

- Makes an assumption of a single pose estimate
- By far the most frequently used model
- Easy to compute.
- Estimation of F and H can be difficult. For non-linear systems that are Jacobians and must be computed for each step.

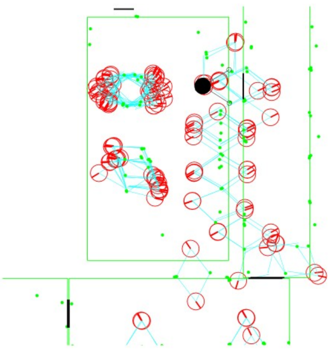
Sum of Gaussians

- Sometimes there might be multiple interpretations of the feature matches
- Each interpretation generates a hypothesis for the robot pose
- Multiple Distributions are generated and a number of them are used in "different" Kalman updates.
- When a model receives no matches or the uncertainty grows too large, i.e. $trace(\Sigma) > \delta$ then the model is terminated.
- Can be computationally challenging

Sum of Gauss/Hypothesis – Door Detection



Sum of Gauss/Hypothesis – More Evidence

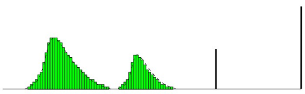


Probability Grids

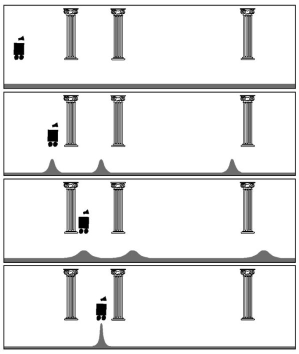
- Space can be tessellated and the update can be performed directly in pose space

$$p(s) = \alpha \sum_{s' \in P} p(s')p(s|s', o)$$

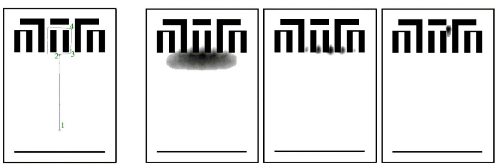
where α is a normalising factor



Grid updating during motion



Grid updating example



Path of the robot *Belief states at positions 2, 3 and 4*

Probability grids

- For accurate localisation in pose space the process can become challenging
- Say, a 100×100 m space with a resolution of 10 cm and an angular resolution of 0.1 deg the space is 1000×1000 or ≈ 3.6 GB of data that must be updated in each time step.
- Various approximation (windowing) methods can be used.

Monte-Carlo Based Methods

- Monte-Carlo based methods is using a sample model for approximation of the pose estimate
- Using a grid model as presented earlier
- Assume with have a number of particles in a collection

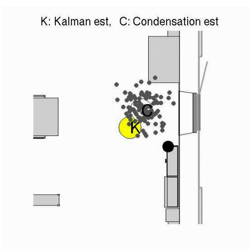
$$S_t = \left\{ (s_t^{(i)}, \pi_t^{(i)}) | \ i = 1..N \right\}$$

  each particle is a hypothesis for the position of the robot, and $\pi_t^{(i)}$ is an associated weight
- We can now approximate $p(s_t | z_0, z_1, ...z_t)$ for any distribution of the pose hypotheses

Monte-Carlo Strategy

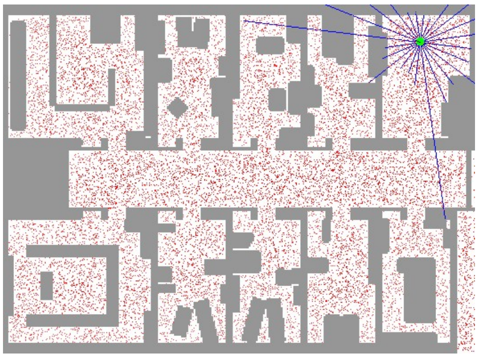
1. Draw N samples from an initial PDF. Typically a uniform distribution. Give each sample a weight of $\frac{1}{N}$
2. Propagate the motion information and draw a new sample from the distribution $p(s_{t+1}^{(i)} | s_t^{(i)}, o_t)$
3. Set the weight of the sample to $\pi_{t+1}^{(i)} = p(z_{t+1} | s_{t+1}^{(i)}) * \pi_t^{(i)}$ based on sensory input
4. Generate a new sample set by drawing samples from the current set and a basis distribution (typically uniform). Normalize the weights
5. Go back to step 2

Monte-Carlo Example



- Example of particle distribution about estimate of position
- Sonar readings for update of the position
- Video of system in operation

Monte-Carlo Example – Burgard, Fox & Thrun



Monte-Carlo Discussion

- Efficient to approximate any distribution of the pose
- The number of particles can be adopted to a particular platform
- Can be used both for simple and multi robot localisation

Outline

1. Introduction
2. Representations
3. Prediction
4. Updating
   - Topological Pose Estimation
   - Gaussian PDF
   - Sum of Gaussian PDF
   - Probability Grids
   - Monte-Carlo Methods
5. Examples
6. Wrap-up

Localisation / Mapping Example

- Now mapping and localisation is also integrated to allow for autonomous operation in general environments
- The mapping and localisation can be integrated to generate – Simultaneous Localisation and Mapping (SLAM)
- Indoor example VIDEO
- Outdoor example VIDEO

Outline

1. Introduction
2. Representations
3. Prediction
4. Updating
   - Topological Pose Estimation
   - Gaussian PDF
   - Sum of Gaussian PDF
   - Probability Grids
   - Monte-Carlo Methods
5. Examples
6. Wrap-up

Wrap-Up

- Localisation is a fundamental competence in mobile robotics
- Involves two major steps
  - Prediction of motion (kinematic modelling)
  - Updating of pose estimate(s)
- The method used depends upon the adopted model for handling of uncertainty and the associated world model
- Brief introduction to the main methods for estimation
- A few illustrative examples