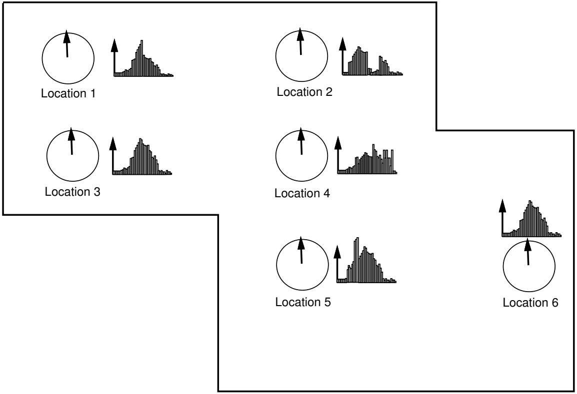
Robotics

Lecture 8: Simultaneous Localisation and Mapping (SLAM)

See course website
`http://www.doc.ic.ac.uk/~ajd/Robotics/` for up to date information.

Andrew Davison
Department of Computing
Imperial College London

Review: Practical 7



- Need repeatable spin and measurement.
- Recognising orientation too will be computationally costly without invariant descriptors.

Simultaneous Localisation and Mapping

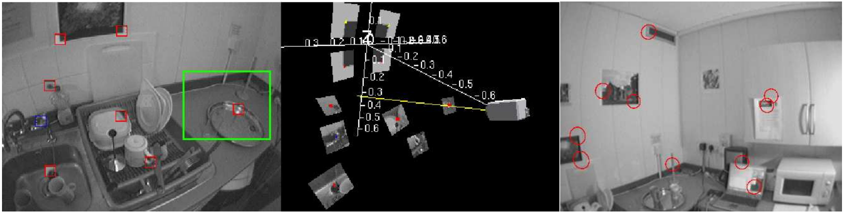
- One of the big successes of probabilistic robotics.

  A body with quantitative sensors
  moves through a previously unknown,
  static environment, mapping it
  and calculating its egomotion.

- When do we need SLAM?
  - When a robot must be truly autonomous (no human input).
  - When nothing is known in advance about the environment.
  - When we can't place beacons (or even use GPS like indoors or underwater).
  - And when the robot actually needs to know where it is.

Features for SLAM

- Most SLAM algorithms make maps of natural scene *features*.
- Laser/sonar: line segments, 3D planes, corners, etc.
- Vision: salient point features, lines, textured surfaces.

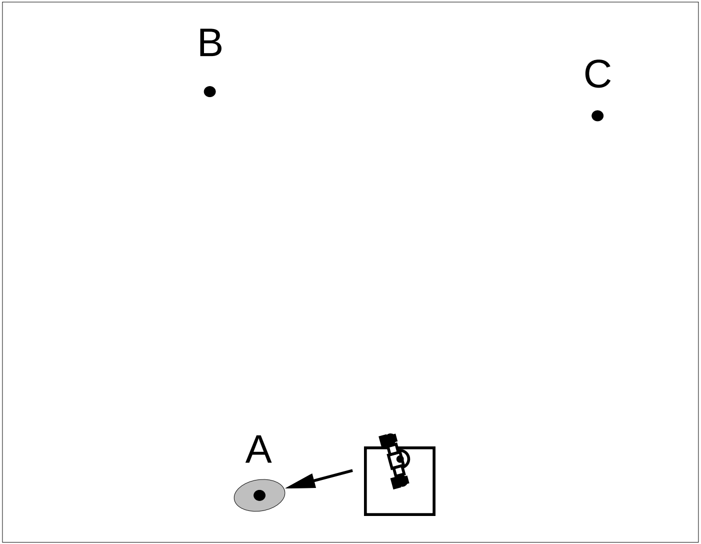


- Features should be distinctive; recognisable from different viewpoints (data association).

Propagating Uncertainty

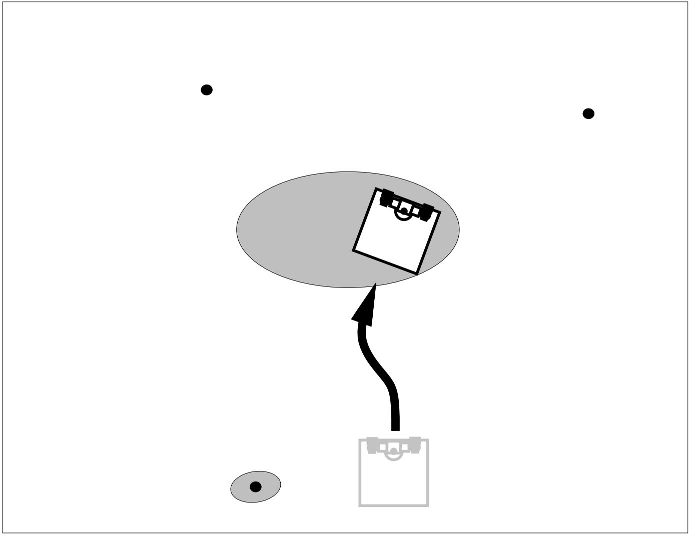
- SLAM seems like a chicken and egg problem — but we can make progress if we assume the robot is the only thing that moves.
- Main assumption: the world is static.

Simultaneous Localisation and Mapping



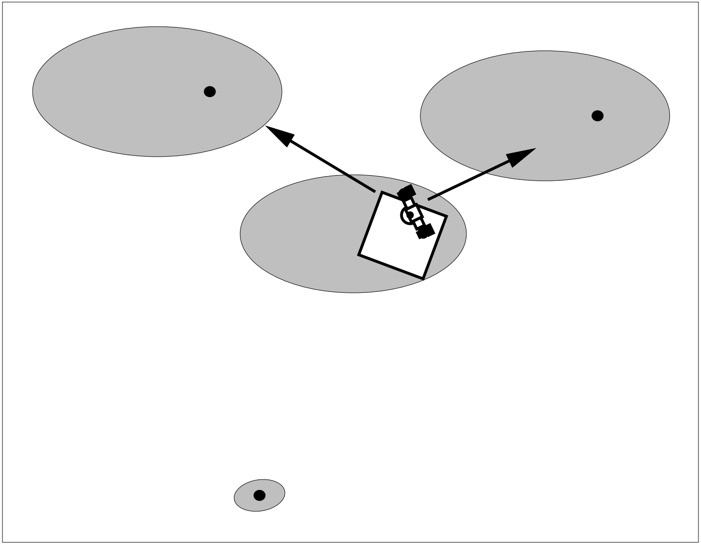
(a) Robot start (zero uncertainty); first measurement of feature A.

Simultaneous Localisation and Mapping



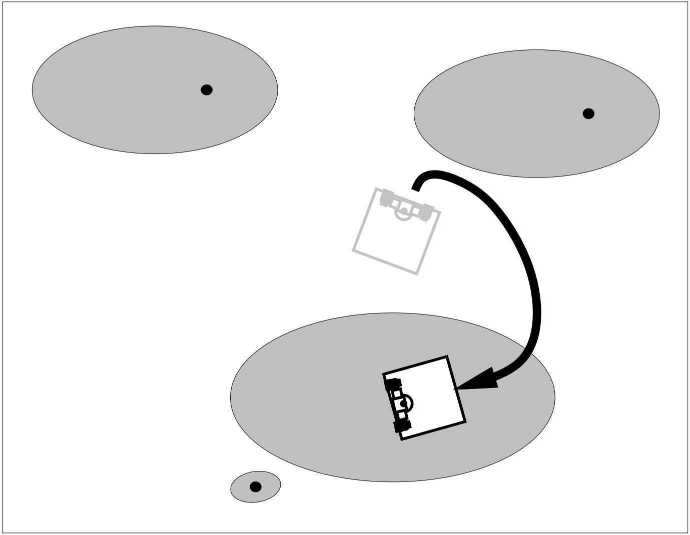
(b) Robot drives forwards (uncertainty grows).

Simultaneous Localisation and Mapping



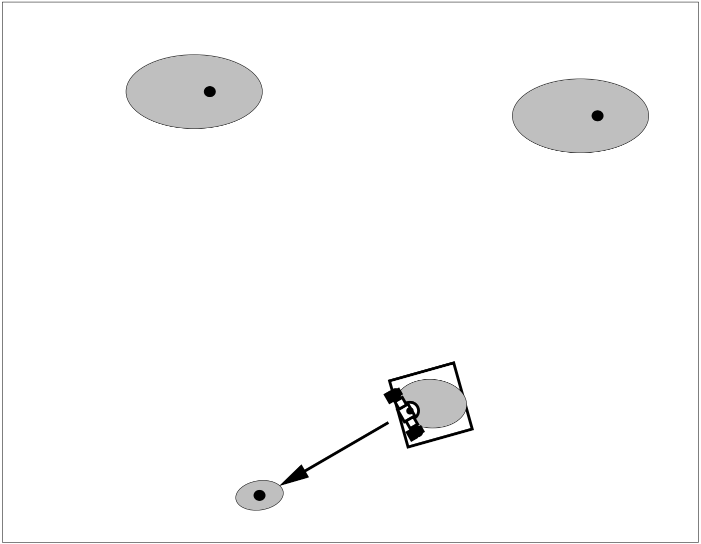
(c) Robot makes first measurements of B and C.

Simultaneous Localisation and Mapping



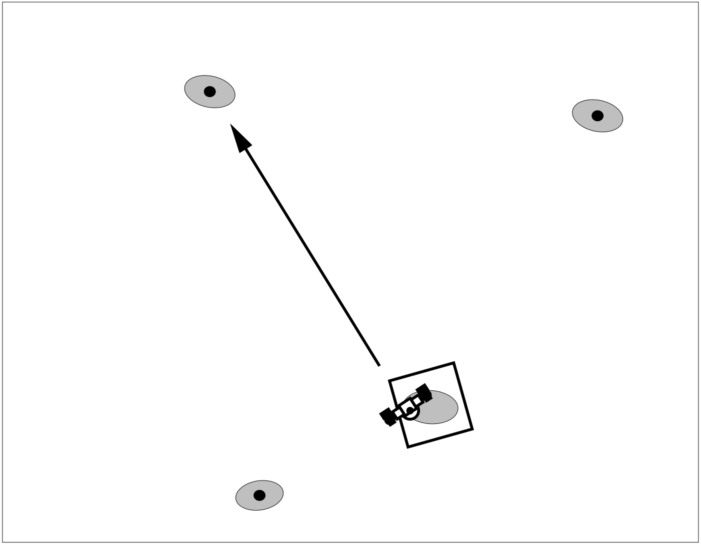
(d) Robot drives back towards start (uncertainty grows more)

Simultaneous Localisation and Mapping



(e) Robot re-measures A; *loop closure*! Uncertainty shrinks.

Simultaneous Localisation and Mapping



(f) Robot re-measures B; note that uncertainty of C also shrinks.

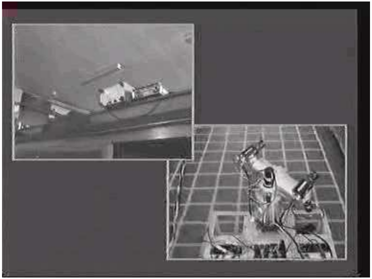
Simultaneous Localisation and Mapping

- First Order Uncertainty Propagation

$$\hat{\mathbf{x}} = \begin{pmatrix} \hat{\mathbf{x}}_v \\ \hat{\mathbf{y}}_1 \\ \hat{\mathbf{y}}_2 \\ \vdots \end{pmatrix} \quad , \quad \mathrm{P} = \begin{bmatrix} \mathrm{P}_{xx} & \mathrm{P}_{xy_1} & \mathrm{P}_{xy_2} & \dots \\ \mathrm{P}_{y_1x} & \mathrm{P}_{y_1y_1} & \mathrm{P}_{y_1y_2} & \dots \\ \mathrm{P}_{y_2x} & \mathrm{P}_{y_2y_1} & \mathrm{P}_{y_2y_2} & \dots \\ \vdots & \vdots & \vdots & \end{bmatrix}$$

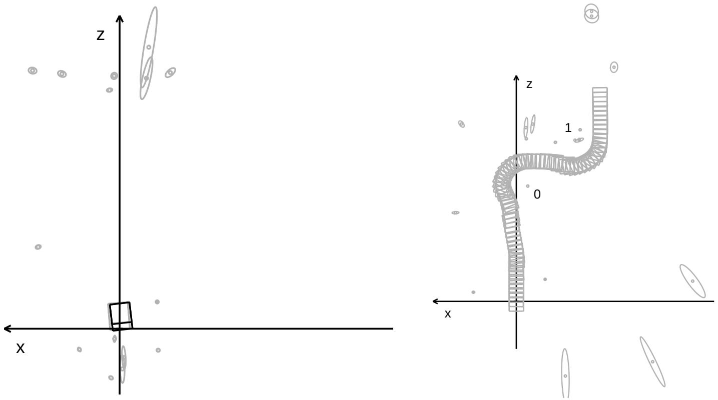
- $\mathbf{x}_v$ is robot state, e.g. (x, y, θ) in 2D; y_i is feature state, e.g. (X, Y) in 2D.
- PDF over robot and map parameters is modelled as a single multi-variate Gaussian and we can use the Extended Kalman Filter.
- PDF represented with state vector and covariance matrix.

SLAM Using Active Vision



- Stereo active vision; 3-wheel robot base.
- Automatic fixated active mapping and measurement of arbitrary scene features.
- Sparse mapping.

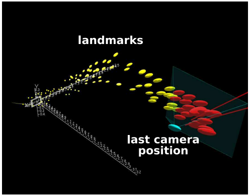
Limits of Metric SLAM



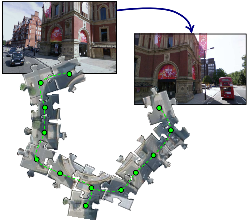
Purely metric probabilistic SLAM is limited to small domains due to:

- Poor computational scaling of probabilistic filters.
- Growth in uncertainty at large distances from map origin makes representation of uncertainty inaccurate.
- *Data Association* (matching features) gets hard at high uncertainty.

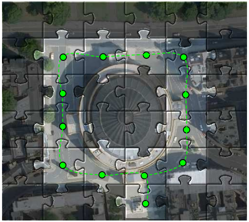
Large Scale Localisation and Mapping



Local Metric



Place Recognition



Global Optimisation

Practical modern solutions to large scale mapping follow a *metric/topological* approach. They need the following elements:

- Local metric mapping to estimate trajectory and possibly make local maps.
- Place recognition, to perform 'loop closure' or relocalise the robot when lost.
- Map optimisation/relaxation to optimise a map when loops are closed.

Global Topological: ‘Loop Closure Detection’

- Angeli *et al.*, IEEE Transactions on Robotics 2008.

Pure Topological SLAM

- Graph-based representation.
- Segmentation of the environment into linked distinct places.
- Adapted to symbolic planning and navigation.

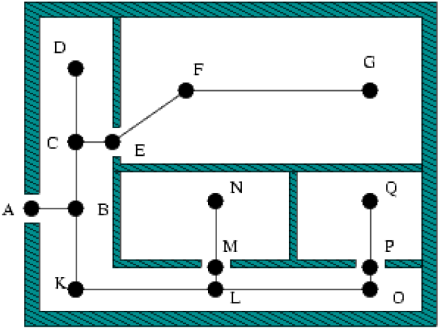
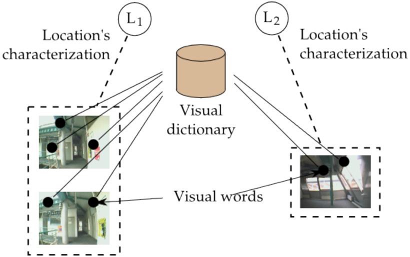
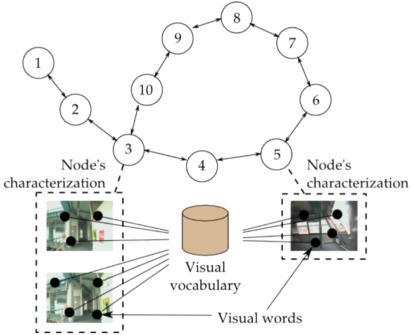


Figure: Topological representation

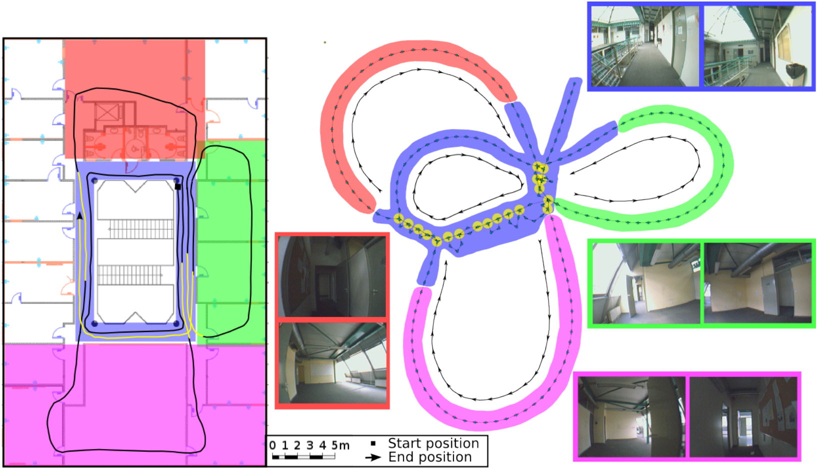
Environment Model

- Map defined as a graph of connected locations.
- Edges model relationships between locations (e.g. traversability, similarity).

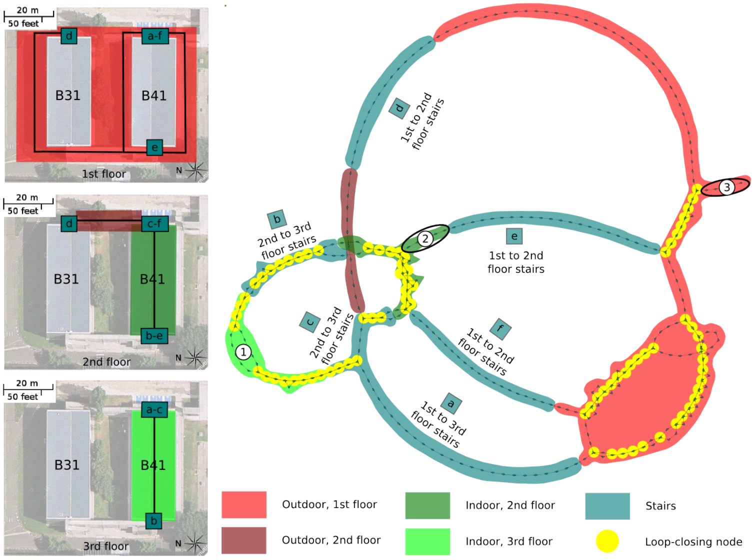




Indoor Topological Map

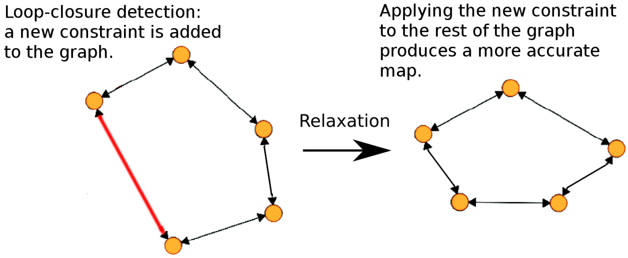


Mixed Indoor / Outdoor Topological Map, Several Levels



Adding Metric Information on the Edges

- Take advantage of odometry measurements from a wheeled robot to add relative displacement information between nodes.
- Apply simple graph-relaxation algorithm. to compute accurate 2D absolute positions for the nodes.



Relaxation Algorithm

1. Estimate position and variance of node i from each neighboring node j:

$$(x_i')_j = x_j + d_{ji}\cos(\theta_{ji} + \theta_j) \quad (y_i')_j = y_j + d_{ji}\sin(\theta_{ji} + \theta_j) \quad (\theta_i')_j = \theta_j + \varphi_{ji} \tag{1}$$

$$(v_i')_j = v_j + v_{ji} \tag{2}$$

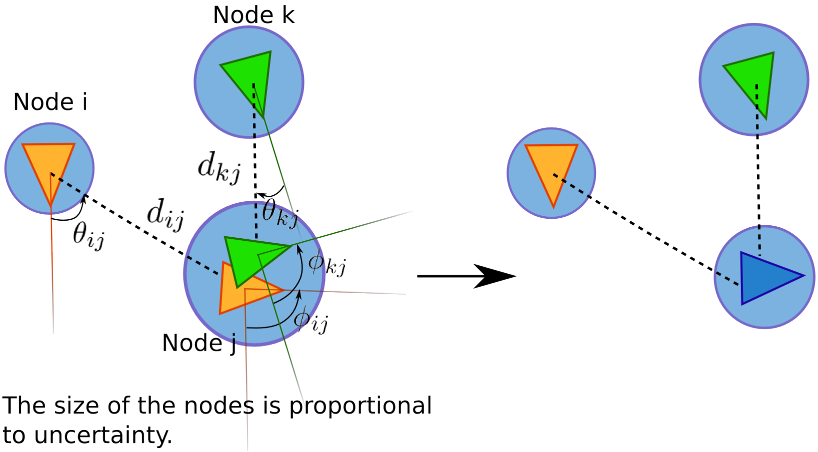
2. Estimate variance of node i using harmonic mean of estimates from neighbors (n_i = number of neighbors of node i):

$$v_i = \frac{n_i}{\sum_j \frac{1}{(v_i')_j}} \tag{3}$$

3. Estimate position of node i as the mean of the estimates from its neighbors:

$$x_i = \frac{1}{n_i}\sum_j \frac{(x_i')_j v_i}{(v_i')_j} \qquad y_i = \frac{1}{n_i}\sum_j \frac{(y_i')_j v_i}{(v_i')_j} \qquad \theta_i = \arctan\left(\frac{\sum_j \frac{\sin((\theta_i')_j)}{(v_i')_j}}{\sum_j \frac{\cos((\theta_i')_j)}{(v_i')_j}}\right) \tag{4}$$

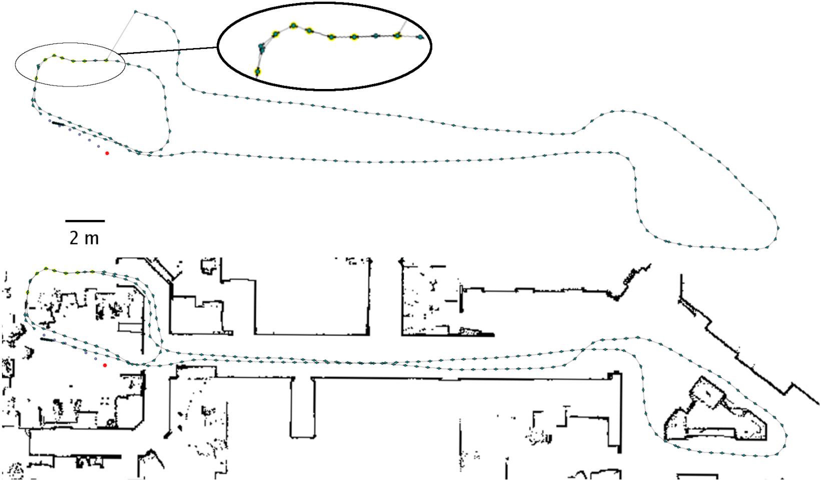
Relaxation Algorithm (Duckett, 2000): Illustration



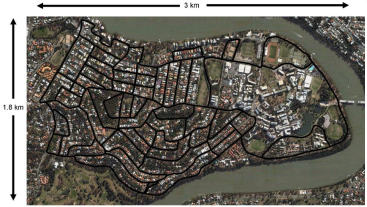
The size of the nodes is proportional to uncertainty.

The position and orientation of node *j* is obtained as the mean of the positions obtained from nodes *i* and *k* (i.e., by composing their positions with the corresponding relative displacements to node *j*).

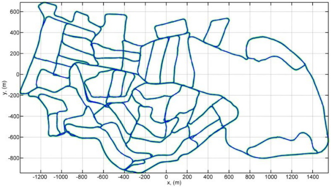
Map Relaxation: Good Odometry, One Loop Closure



Simple Large-Scale SLAM: RATSLAM



(a)



(b)

Milford and Wyeth, 2007.
`http://www.youtube.com/watch?v=-0XSUi69Yvs`

- Very simple 'visual odometry' gives rough trajectory.
- Simple visual place recognition provides *many* loop closures.
- Map relaxation/optimisation to build global map.