Introduction to SLAM Part II

Paul Robertson

Review

- Localization
  - **Tracking**, Global Localization, Kidnapping Problem.
- Kalman Filter
  - Quadratic
  - Linear (unless EKF)
- SLAM
  - Loop closing
  - Scaling:
    - Partition space into overlapping regions, use rerouting algorithm.
- Not Talked About
  - Features
  - Exploration

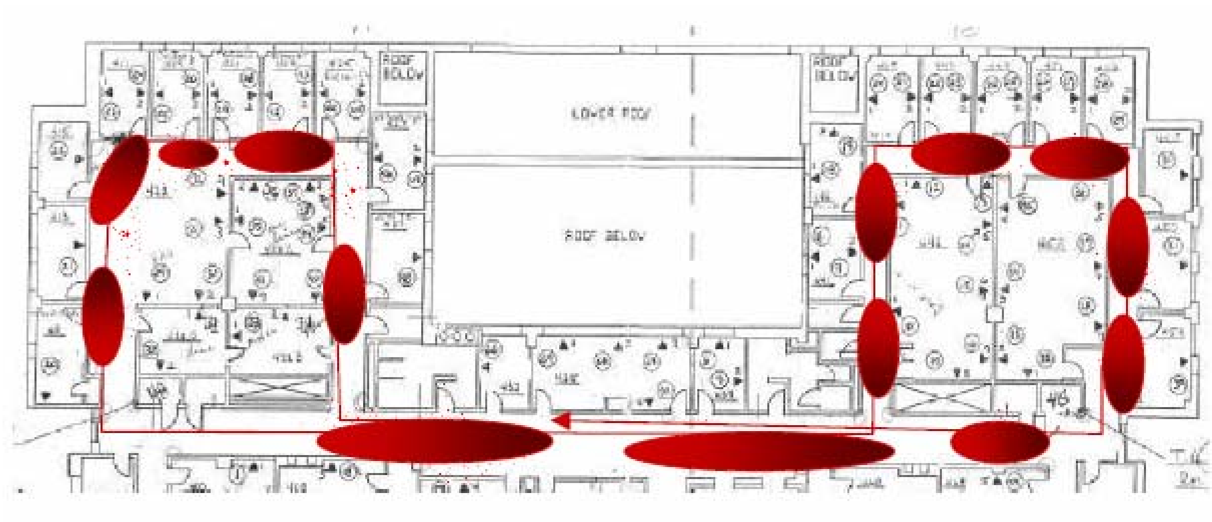
Outline

- Topological Maps
- HMM
- SIFT
- Vision Based Localization

Topological Maps

Idea:

Build a qualitative map where the nodes are similar sensor signatures and transitions between nodes are control actions.



Advantages of Topological maps

- Can solve the Global Location Problem.
- Can solve the Kidnapping Problem.
- Human-like maps
- Supports Metric Localization
- Can represent as a Hidden Markov Model (HMM)

Hidden Markov Models (HMM)

Scenario

- You have your domain represented as set of state variables.
- The states define what following state are reachable from any given state.
- State transitions involve action.
- Actions are observable, states are not.
- You want to be able to make sense of a sequence of actions

Examples

Part-of-speech tagging, natural language parsing, speech recognition, scene analysis, Location/Path estimation.

Overview of HMM

What a Hidden Markov Model is

- Algorithm for finding the most likely state sequence.
- Algorithm for finding the probability of an action sequence (sum over all allowable state paths).
- Algorithm for training a HMM.
- Only works for problems whose state structure can be characterized as FSM in which a single action at a time is used to transition between states.
- Very popular because algorithms are linear on the length of the action sequence.

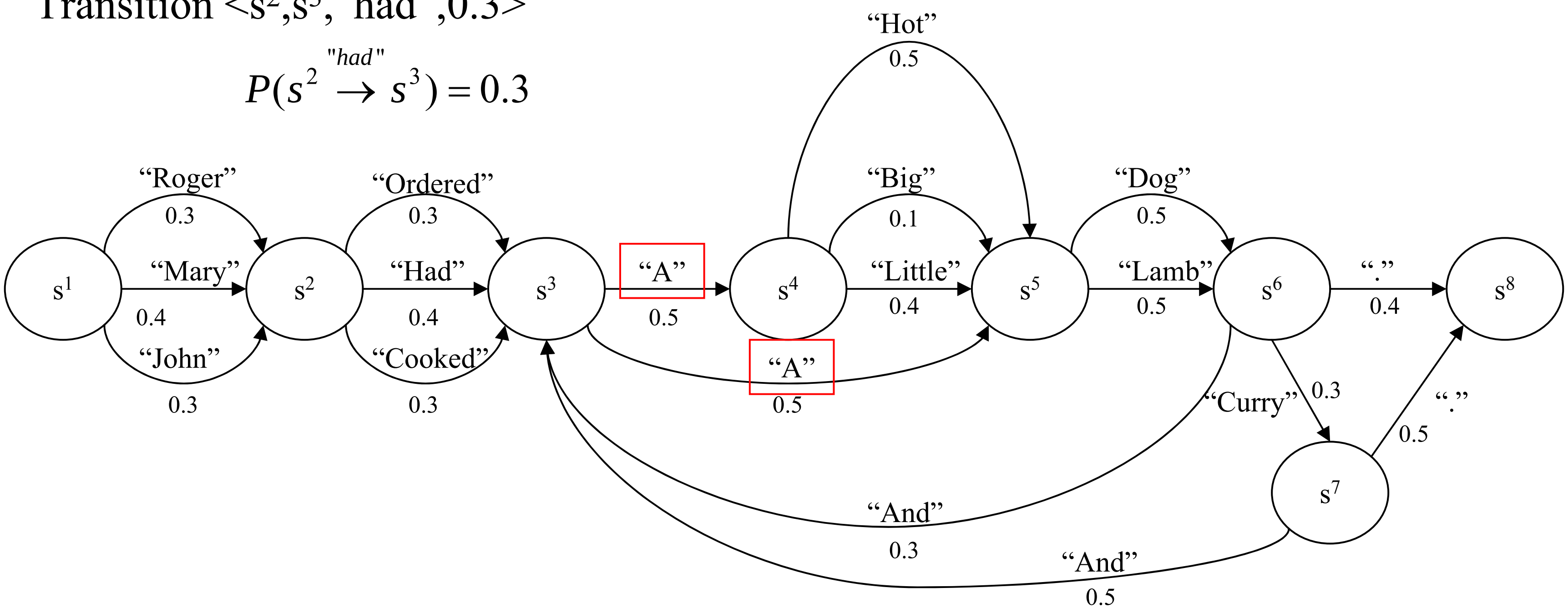
Hidden Markov Models

A finite state machine with probabilities on the arcs.

<s^1,S,W,E> where S={s^1,s^2,s^3,s^4,s^5,s^6,s^7,s^8}; W={"Roger", …}; E={<transition> …}

Transition <s^2,s^3,"had",0.3>

$$P(s^2 \xrightarrow{\text{"had"}} s^3) = 0.3$$



S_1: Mary had a little Lamb and a big dog.

S_2: Roger ordered a lamb curry and a hot dog.

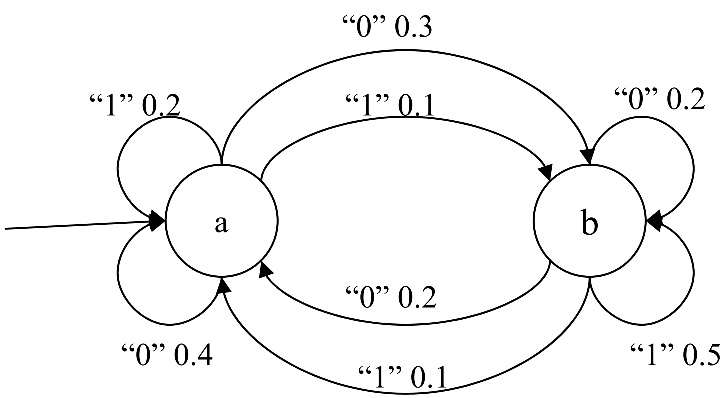
S_3: John cooked a hot dog curry.

$P(S_3)=0.3*0.3*0.5*0.5*0.3*0.5=0.003375$

Finding the most likely path

Viterbi Algorithm: For an action sequence of length t-1 finds:

$$\sigma(t) = \arg\max_{s_{1,t}} P(s_{1,t} \mid w_{1,t-1})$$ in linear time.



"1110"

Viterbi Algorithm:

For each state extend the most probable state sequence that ends in that state.

States		ε	1	11	111	1110
a	Sequence	a	aa	aaa	aaaa	abbba
	Probability	1.0	0.2	0.04	0.008	0.005
b	Sequence	b	ab	abb	abbb	abbbb
	Probability	0.0	0.1	0.05	0.025	0.005

Action Sequence Probabilities

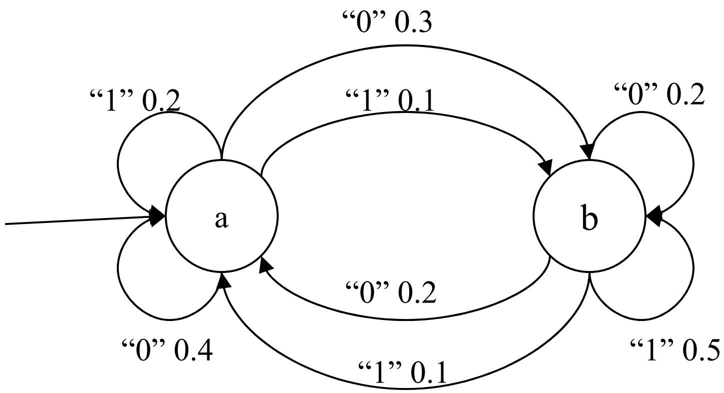
$$P(w_{1,n}) = \sum_{i=1}^{\sigma} P(w_{1,n}, S_{n+1} = s^i)$$

Let $\alpha_i(t)$ be the probability $P(w_{1,t-1}, S_t = s^i)$ so $P(w_{1,n}) = \sum_{i=1}^{\sigma} \alpha_i(n+1)$

$$\alpha_i(1) = \begin{cases} i = 1 \rightarrow 1.0 \\ otherwise \rightarrow 0 \end{cases}$$ (Must start in the start state).

$$\alpha_j(t+1) = \sum_{i=1}^{\sigma} \alpha_i(t) P(s^i \xrightarrow{w_t} s^j)$$

HMM forward probabilities



"1110"

t	1	2	3	4	5
	ε	1	1	1	0
$\alpha a(t)$	1.0	**0.2**	0.05	0.017	0.0148
$\alpha b(t)$	0.0	**0.1**	**0.07**	0.04	0.0131
$P(w_{1,t})$	1.0	0.3	0.12	0.057	**0.0279**

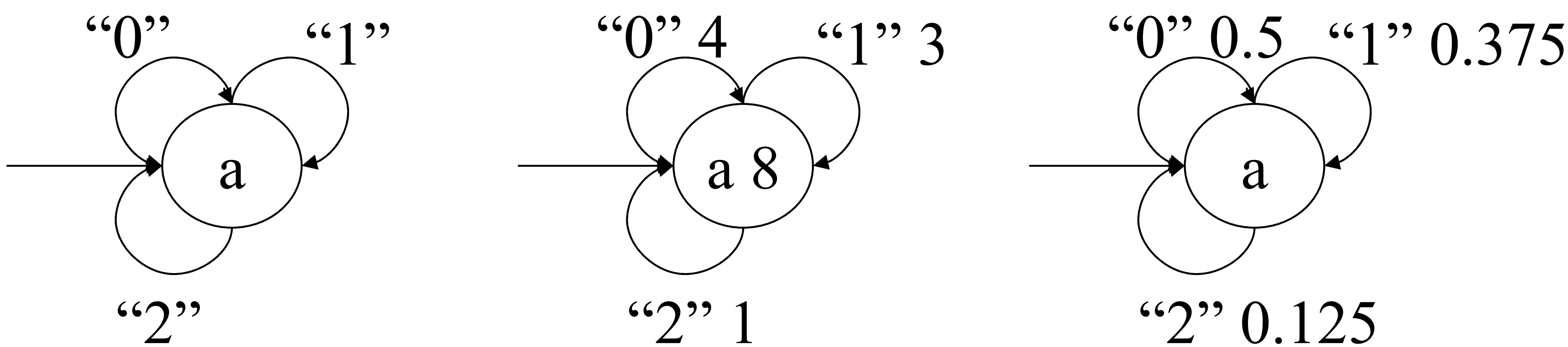
0.2*0.1=0.02

+

0.1*0.5=0.05

HMM Training
(Baum-Welch Algorithm)

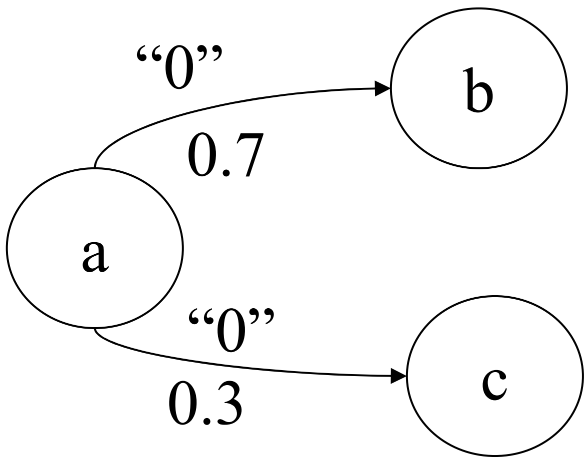
Given a training sequence, adjusts the HMM state transition probabilities to make the action sequence as likely as possible.



Training Sequence: 01010210

With Hidden States

Intuitively…



When counting transitions Prorate transitions by their Probability.

? But you don't know the transition probabilities!

1. Guess a set of transition probabilities.
2. (while (improving)

   (propagate-training-sequences))

"improving" is calculated by comparing the cross-entropy after each iteration. When the cross-entropy decreases by less than ε in an iteration we are done.

Cross entropy is:

$$-\frac{1}{n}\sum_{w_{1,n}} P_{M-1}(w_{1,n}) \log_2 P_M(w_{1,n})$$

Scale Invariant Feature Transform

David Lowe ‘Distinctive Image Features from Scale-Invariant Keypoints’ IJCV 2004.

Stages:

- Scale Space (Witkin ‘83) Extrema Extraction
- Keypoint Pruning and Localization
- Orientation Assignment
- Keypoint Descriptor

Scale space in SIFT

Motivation:

– Objects can be recognized at many levels of detail
– Large distances correspond to low l.o.d.
– Different kinds of information are available at each level

Idea: Extract information content from an image at each l.o.d. Detail reduction done by Gaussian blurring:

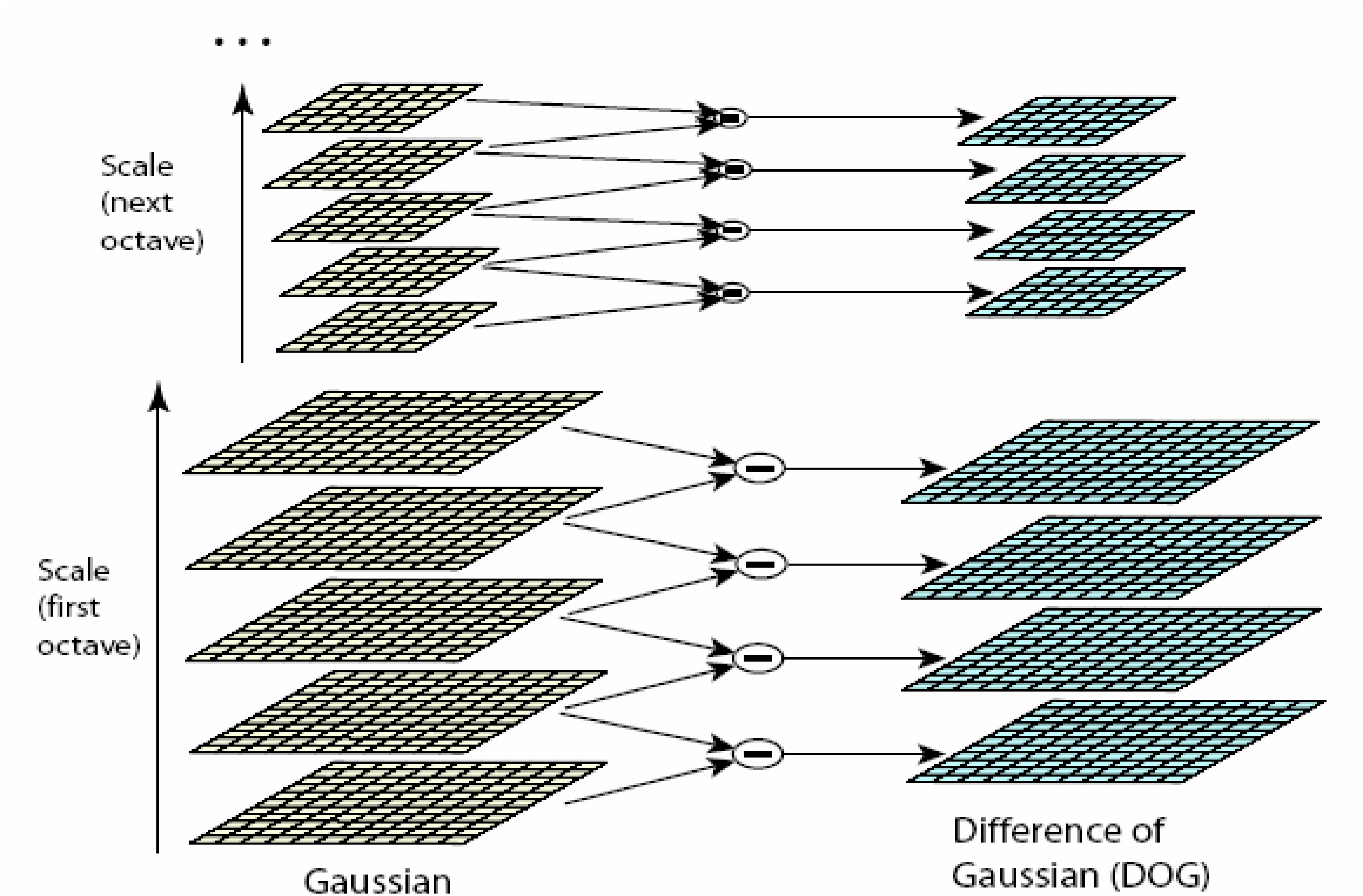
– $I(x, y)$ is input image. $L(x, y, \sigma)$ is rep. at scale σ.
– $G(x, y, \sigma)$ is 2D Gaussian with variance σ^2
– $L(x, y, \sigma) = G(x, y, \sigma) * I(x, y)$
– $D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma)$

Features of SIFT

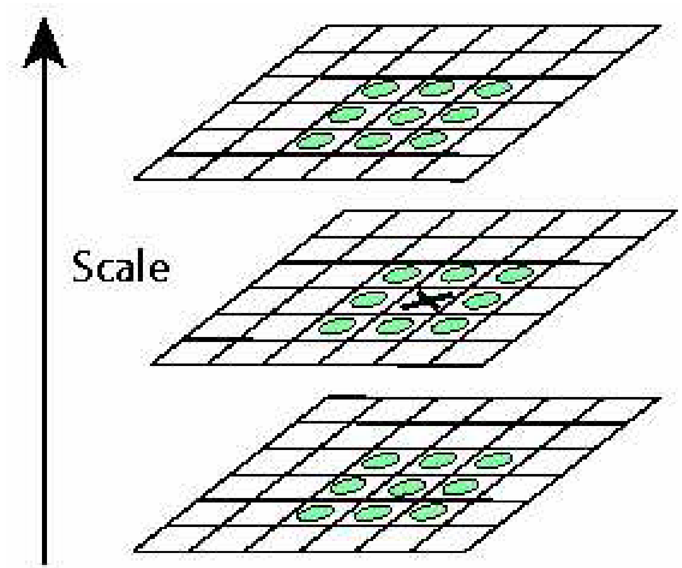
Invariant to:

- Scale
- Planar Rotation
- Contrast
- Illumination
- Large numbers of features

Difference of Gaussians



Scale Space

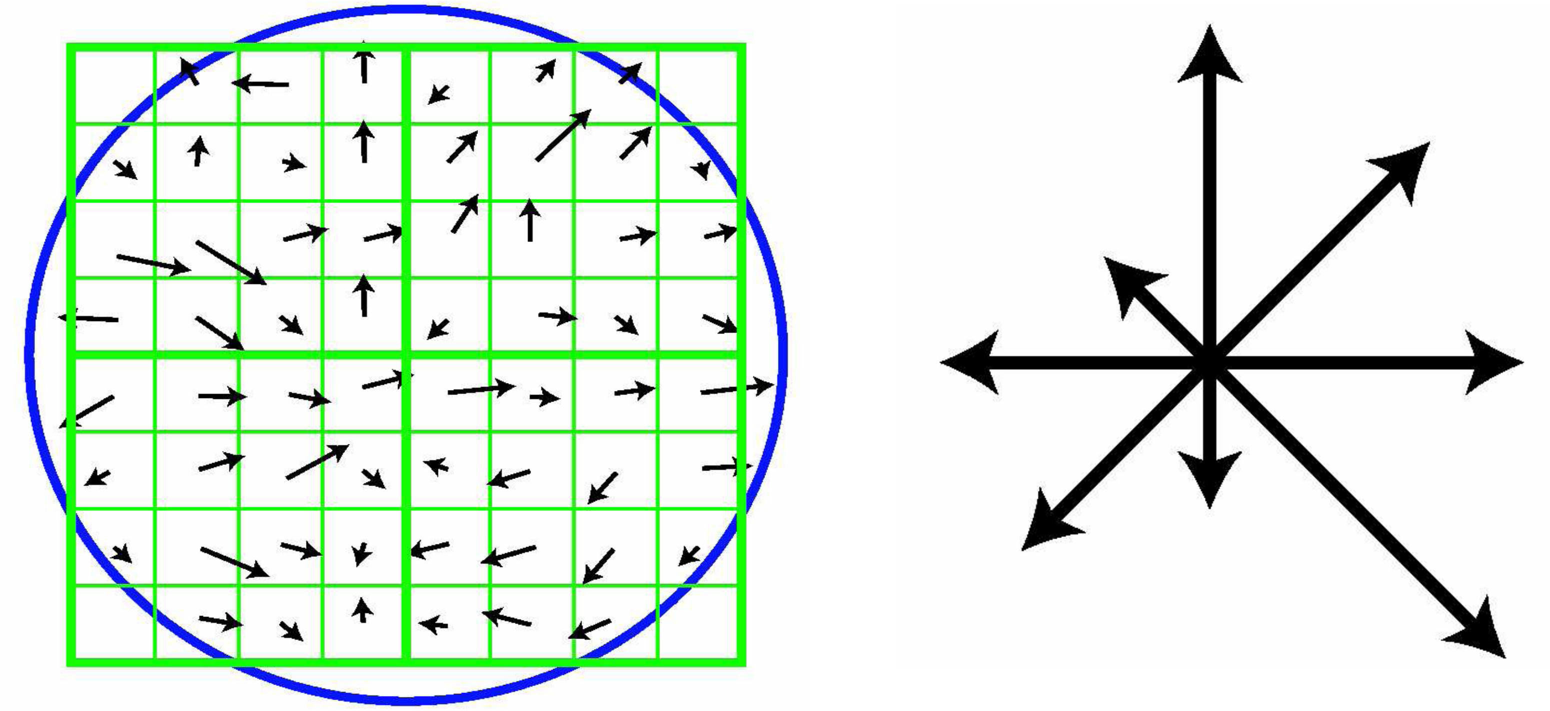


- Compute local extrema of D
- Each (x, y, σ) is a feature.
- (x, y) scale and planar rotation invariant.

Pruning for Stability

- Remove feature candidates that
  - Low Contrast
  - Unstable Edge Responses

Orientation Assignment



For each feature (x, y, σ):

– Find fixed-pixel-area patch in $L(x, y, \sigma)$ around (x, y)

– Compute gradient histogram; call this b_i

– For b_i within 80% of max, make feature (x, y, σ, b_i)

Vision Based SLAM

Readings:

Se, S., D. Lowe and J. Little, ‘Mobile Robot Localization and Mapping with Uncertainty using Scale-Invariant Visual Landmarks’, The International Journal of Robotics Research, Volume 21 Issue 08.

Kosecka, J. Zhou, L. Barber, P. Duric, Z. ‘Qualitative Image Based Localization in Indoor Environments’ CVPR 2003.

Predictive Vision-Based SLAM

1. Compute SIFT features from current location.
2. Use Stereo to locate features in 3D.
3. Move
4. Predict new location based on odometry and Kalman Filter.
5. Predict location of SIFT features based upon motion of robot.
6. Find SIFT features and find 3D position of each.
7. Compute position estimate from each matched feature.

Vision Based Localization

- Acquire video sequence during the exploration of new environment.
- Build environment model in terms of locations and spatial relationships between them.
- Topological localization by means of location recognition.
- Metric localization by computing relative pose of current view and representation of most likely location.

Same Location?

Global Topology, Local Geometry

Issues:

1. Representation of individual locations
2. Learning the representative location features
3. Learning neighborhood relationships between locations.
4. Each view represented by a set of SIFT features.
5. Locations correspond to sub-sequences across which features can be matched successfully.
6. Spatial relationships between locations are captured by a location graph.

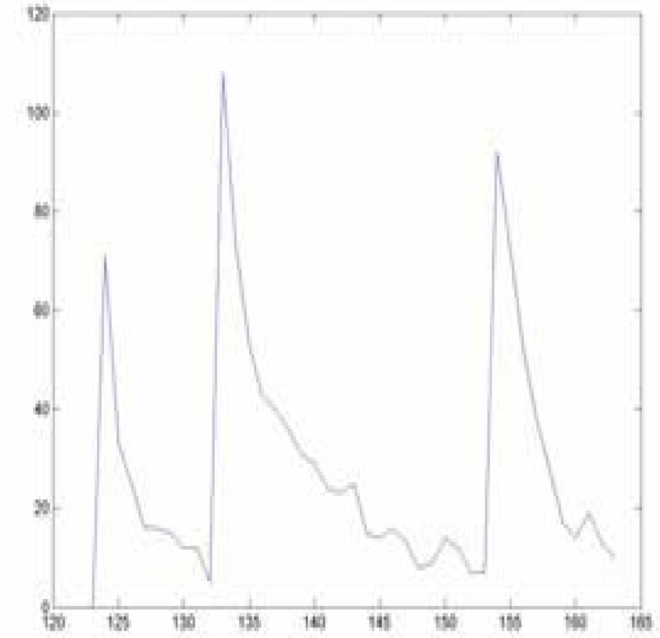
Image Matching

10 – 500 features per view

- For each feature find the discriminative nearest neighbor feature.
- Image Distance (Score) - # of successfully matched features.

Partitioning the Video Sequence

- Transitions determined during exploration.
- Location sub-sequence across which features can be matched successfully.
- Location Representation: set of representative views and their associated features.



Location Recognition

- Given a single view what is the location this view came from ?

Recognition – voting scheme

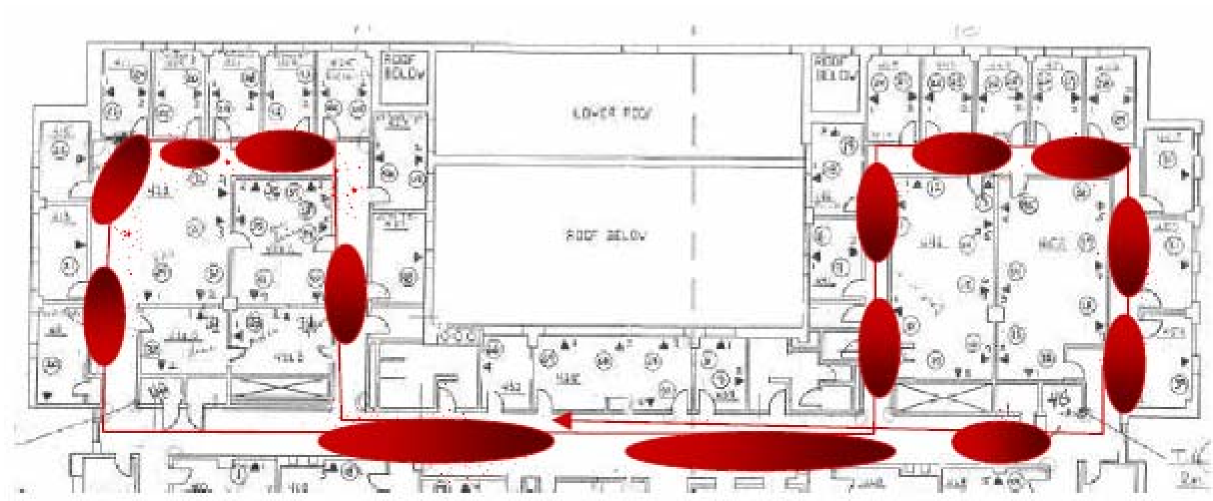
For each representative view selected in the exploration stage

1. Compute the number of matched features.
2. Location with maximum number of matches is the most likely location.

Markov Localization in the topological Model

Exploiting the spatial relationships between the locations

- *S – discrete set of states L x {N, W, S, E} locations and orientations*
- *A – discrete set of actions (N, W, S, E)*
- *T(S, A, S') – transition function , Discrete Markov Model*



Markov Localization

$$P(L_t=l_i|o_{1:t}) \quad \propto P(o_t|L_t=l_i)\ P(L_t=l_i|o_{1:t-1})$$

Location posterior — Observation likelihood

P(location|observations) — P(image|location)

Observation Likelihood $P(o_t|L_t=l_i) = \dfrac{C(i)}{\Sigma_j C(j)}$

P(image|location)

$$P(L_t=l_i|o_{1:t-1}) = \Sigma\ A(i,j)P(L_{t-1}=l_j|o_{1:t-1})$$

Location transition probability matrix

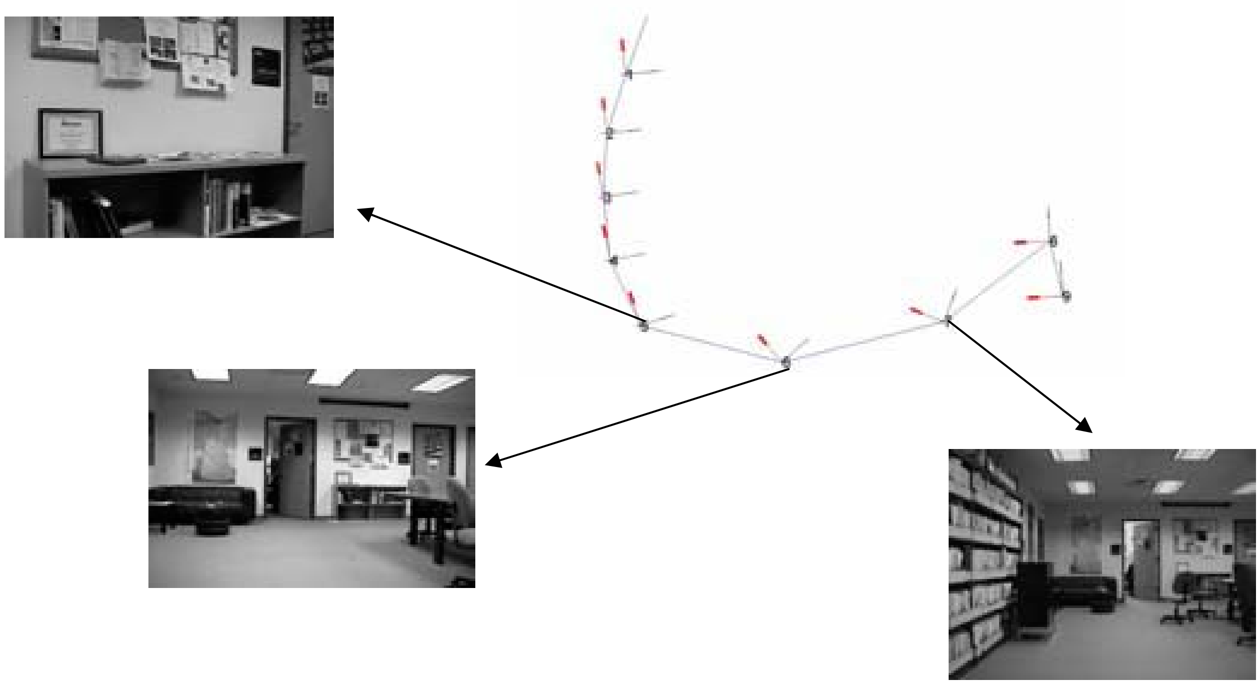
HMM Recognition

- Improved recognition rate from 82% to 96% in experimental tests

Metric Localization within Location

1. Given closest representative view of the location
2. Establish exact correspondences between keypoints
3. Probabilistic matching combining (epipolar) geometry, keypoint descriptors and intrinsic scale

Compute relative pose with respect to the reference view



Wrap up

- What we have covered:
  - Supporting Methods
    - Kalman Filter
    - HMM
    - SIFT
  - Localization and Mapping
    - Basic SLAM
    - Large Scale SLAM (Leonard)
    - Topological Maps
    - Vision Based Localization/SLAM