Topological Mapping

Discrete Bayes Filter

Vision Based Localization

Given a image(s) acquired by moving camera
determine the robot’s location and pose ?

- Towards localization without odometry
- What can be achieved using solely visual sensing ?
- Applications toward agumenting human navigational capabilities (indoors, outdoors)

Related Work

- Vision-based SLAM – pose maitenance [Stephens' 02, Se' 02]
- Landmark Based Methods [Sims,Dudek 2001, Taylor 1998]
- Appearance Based SLAM [Rybski et. al ' 03]
- Appearance based Topological localization [Ulrich' 00, Gaspar' 00]

- Approaches motivated by object recognition – given the image determine which location that image came from
- Approaches motivated by structure and motion estimation

- Integrate information over several channels [Torralba et al' 03]
  Rotation invariant image descriptors [Wolf-Burgard' 03]
  PCA based approaches [Leonardis' 01]
- Omni-directional cameras [Artac2002, Gaspar2000]

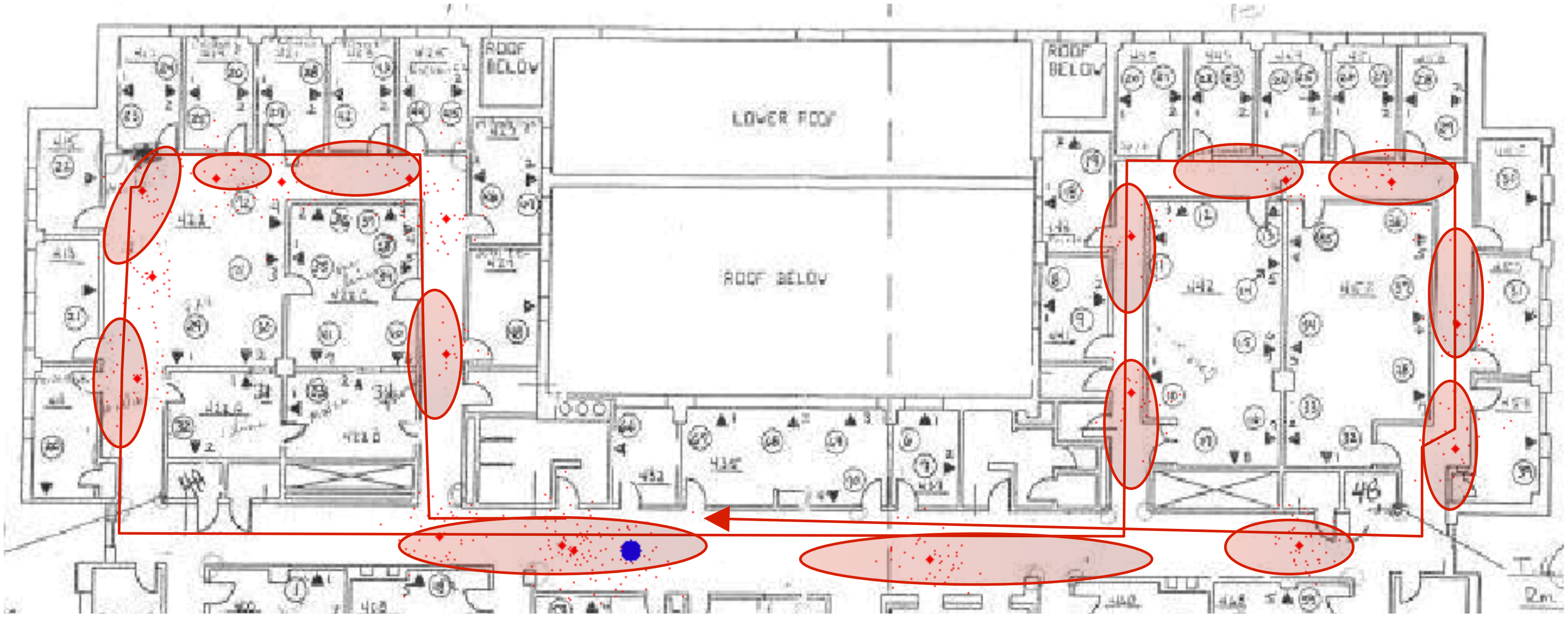
Challenges

- Metric and topological localization using only vision
- Applicable to large scale self-similar environments
- Robust to dynamic changes in the environment

Our Approach

- Acquire video sequence during the exploration
- Build the environment model in terms of locations and spatial relationships between them
- Topological localization by means of location recognition
- Metric localization by means of relative positioning

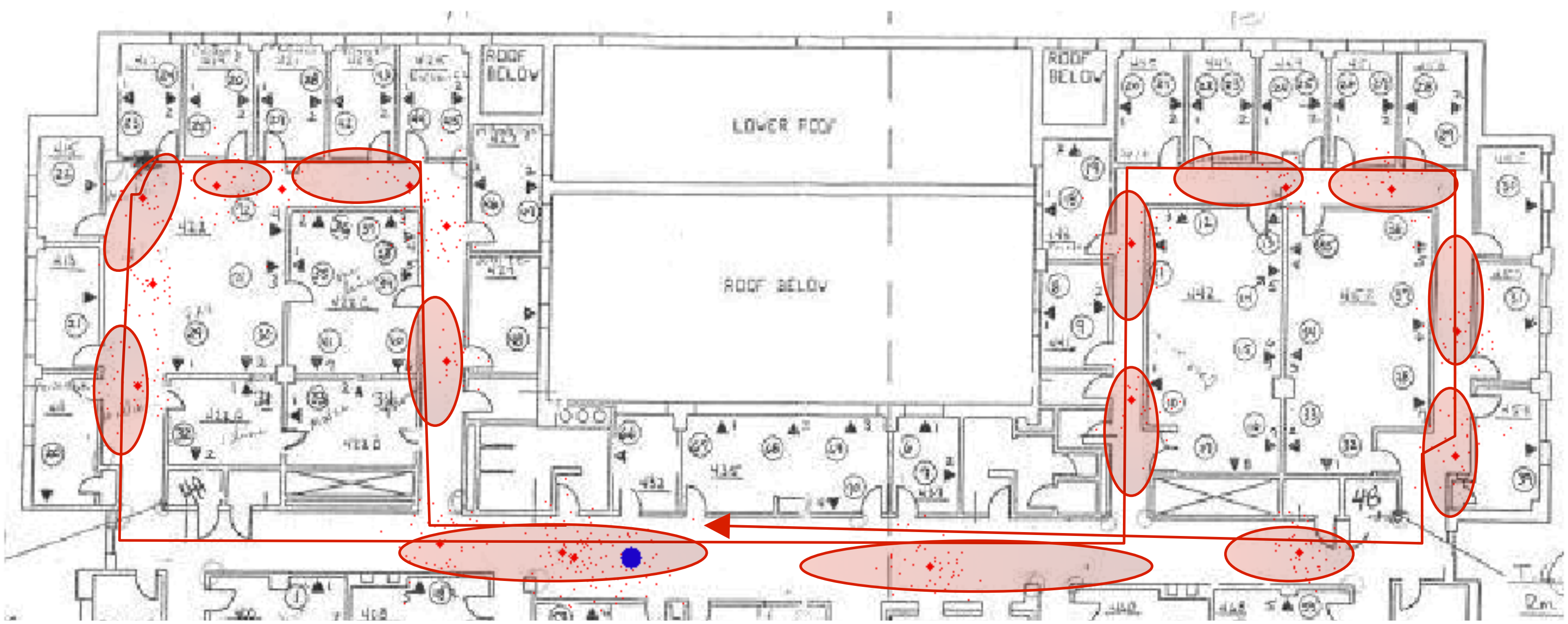
Vision Based Localization



Vision Based Localization

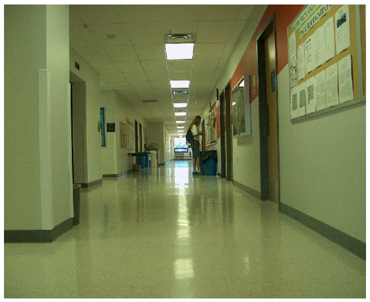
- Impose some discrete structure on the space of continuous visual observations (associate semantic labels with individual locations - corridor, hallway, office)

- Localization given the topological model

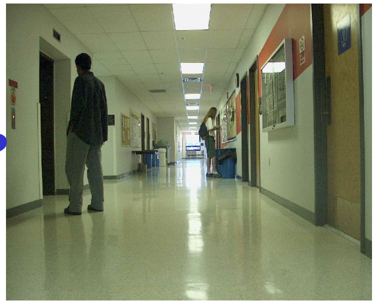


Issues

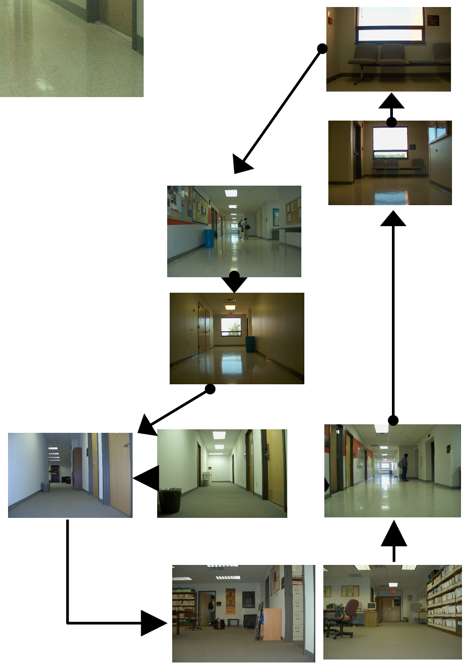
- Representation of individual locations
- Learning the representative location features
- Learning neighborhood relationships between locations



Same location ?



- Each view is represented by a set of scale invariant features or image histograms
- Locations correspond to sub-sequences across which features can be matched successfully
- Spatial relationships between locations are captured by Hidden Markov Model

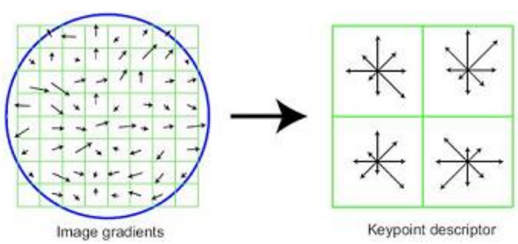


Scale Invariant Features

- Each image is characterized by a set of scale-invariant keypoints and their associated descriptors [D. Lowe,2000]
- Keypoints - extrema in DOG pyramid

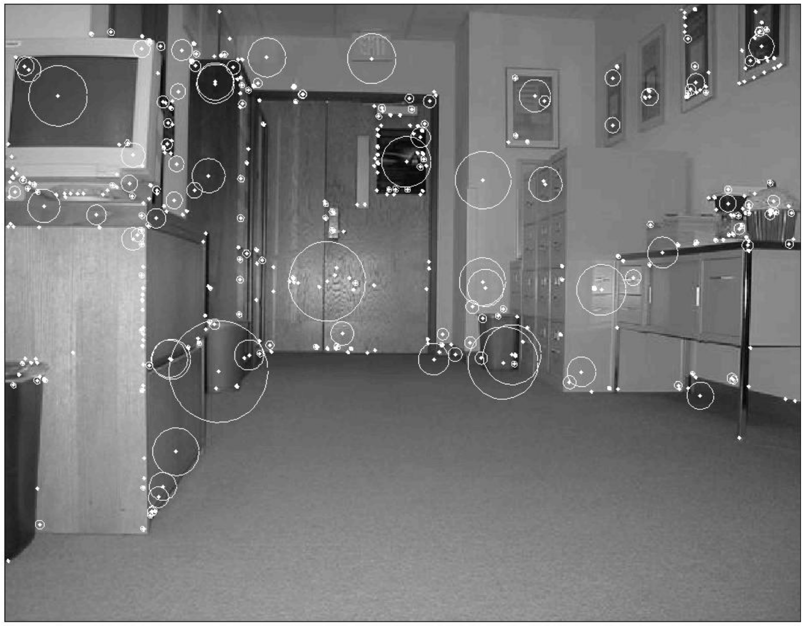
$$
\begin{aligned}
D(x, y, \sigma) &= (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \\
&= L(x, y, k\sigma) - L(x, y, \sigma).
\end{aligned}
$$

- Descriptor – 8 bin orientation histograms computed over 4 x 4 grid overlayed over pixel neighbourhood and stacked together to form a 128 dim feature vector



- Good repeatability across variations of scale and pose

Image Matching



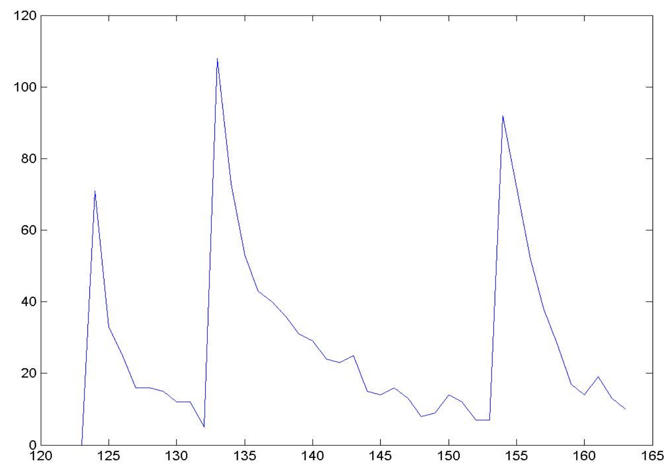
10 – 500 features for each view of the sequence

- For each keypoint find the discriminative nearest neighbor keypoint, based on Euclidean distance between two descriptors

- Image Distance (Score) - # of successfully matched features

Partitioning the video sequence

- Transitions between individual locations determined during exploration
- Location sub-sequence across which features can be matched successfully (# of successfully matched features is lower then 2*minimal number of features needed for pose estimation)
- Location Representation - set of representative views and their associated keypoints



# of matched features 1^{st} – i-th view

Representative views of locations

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. The location with maximum number of matches is the most likely location

- Recognition Rates

# of views	Training sequence	Test 1 sequence	Test 2 sequence
one	84 %	46%	44%
two	97%	68%	66%
four	100%	82%	83%

Location Recognition

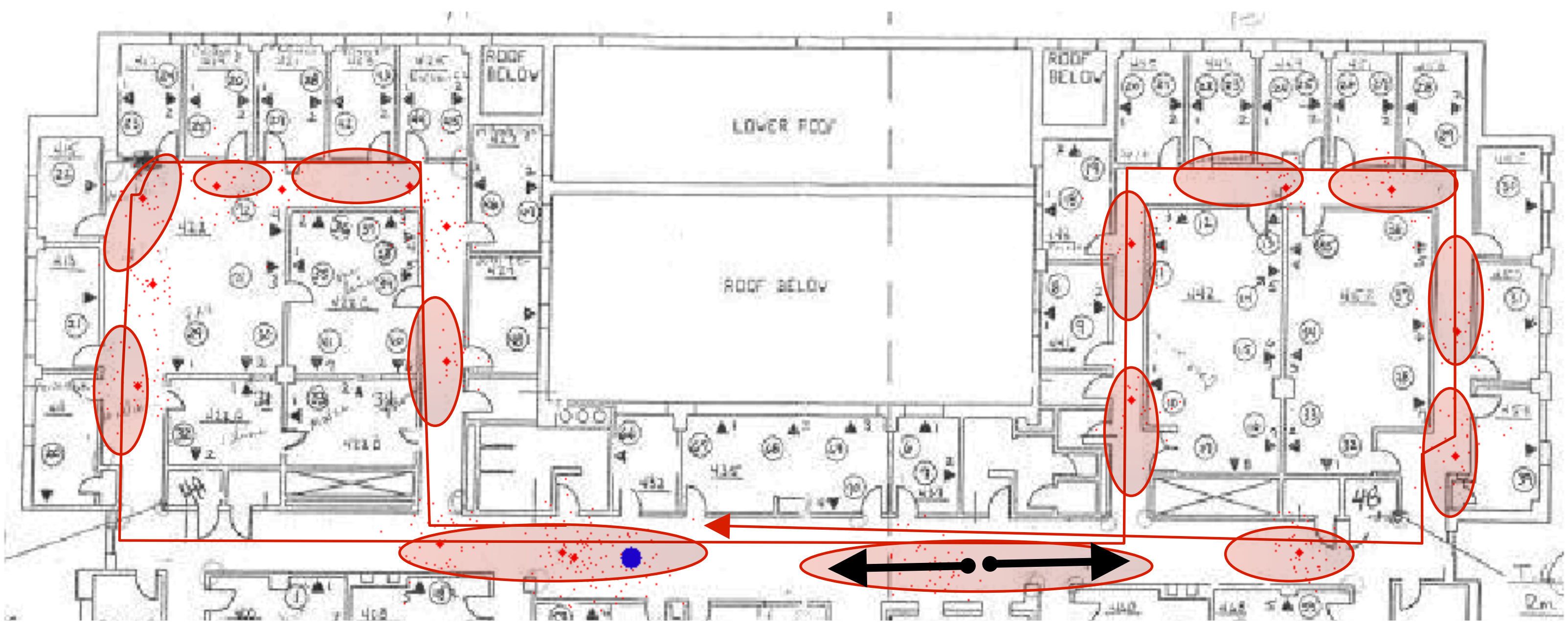
- Large changes in the view point -> misclassification
- Misclassification due to dynamic changes in the environment

- Exploit spatial relationships between individual locations to improve recognition

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, S') – transition function , Discrete Markov Model*



Markov Localization in the topological model

Given the sequences of views what is the most likely
Location the current view came from ?

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

Location posterior
P(location |observations)

Observation likelihood
P(image|location)

Markov Localization in the topological model

Given the sequences of views what is the most likely
Location the current view came from ?

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

Location posterior
P(location |observations)

Observation likelihood
P(image|location)

# of successfully matched features

Observation likelihood
P(image|location)

$$P(o_t | L_t = l_i) = \frac{C(i)}{\sum_j C(j)}$$

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

Location transition probability matrix

- Slight digression

Time and uncertainty

The world changes; we need to track and predict it

Diabetes management vs vehicle diagnosis

Basic idea: copy state and evidence variables for each time step

$\mathbf{X}_t$ = set of unobservable state variables at time t
e.g., $BloodSugar_t$, $StomachContents_t$, etc.

$\mathbf{E}_t$ = set of observable evidence variables at time t
e.g., $MeasuredBloodSugar_t$, $PulseRate_t$, $FoodEaten_t$

This assumes **discrete time**; step size depends on problem

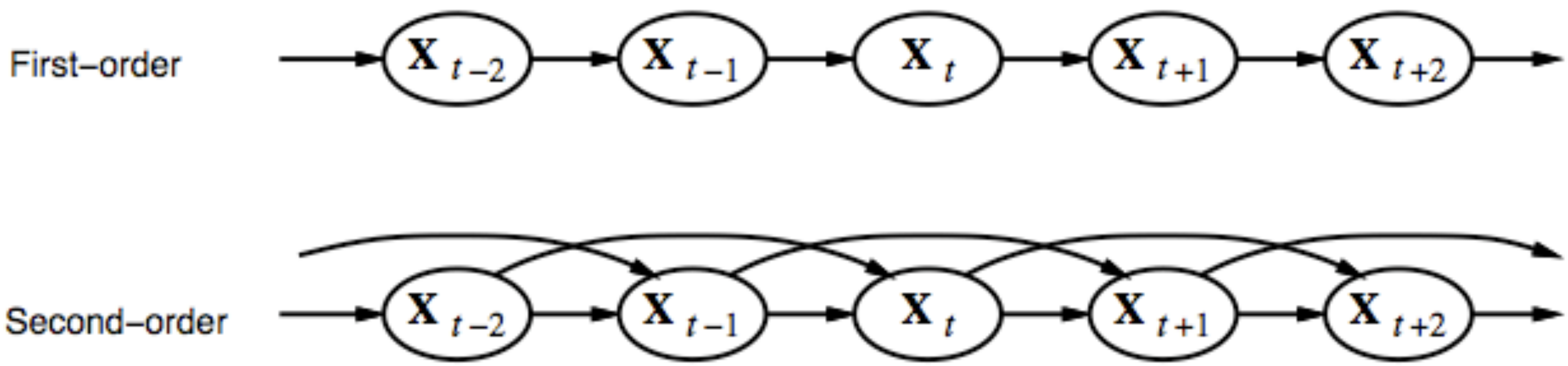
Notation: $\mathbf{X}_{a:b} = \mathbf{X}_a, \mathbf{X}_{a+1}, \ldots, \mathbf{X}_{b-1}, \mathbf{X}_b$

Markov Property

Construct a Bayes net from these variables: parents?

Markov assumption: $\mathbf{X}_t$ depends on **bounded** subset of $\mathbf{X}_{0:t-1}$

First-order Markov process: $\mathbf{P}(\mathbf{X}_t|\mathbf{X}_{0:t-1}) = \mathbf{P}(\mathbf{X}_t|\mathbf{X}_{t-1})$
Second-order Markov process: $\mathbf{P}(\mathbf{X}_t|\mathbf{X}_{0:t-1}) = \mathbf{P}(\mathbf{X}_t|\mathbf{X}_{t-2}, \mathbf{X}_{t-1})$



Sensor Markov assumption: $\mathbf{P}(\mathbf{E}_t|\mathbf{X}_{0:t}, \mathbf{E}_{0:t-1}) = \mathbf{P}(\mathbf{E}_t|\mathbf{X}_t)$

Stationary process: transition model $\mathbf{P}(\mathbf{X}_t|\mathbf{X}_{t-1})$ and sensor model $\mathbf{P}(\mathbf{E}_t|\mathbf{X}_t)$ fixed for all t

Inference Tasks

Filtering: $\mathbf{P}(\mathbf{X}_t|\mathbf{e}_{1:t})$
belief state—input to the decision process of a rational agent

Prediction: $\mathbf{P}(\mathbf{X}_{t+k}|\mathbf{e}_{1:t})$ for $k > 0$
evaluation of possible action sequences;
like filtering without the evidence

Smoothing: $\mathbf{P}(\mathbf{X}_k|\mathbf{e}_{1:t})$ for $0 \leq k < t$
better estimate of past states, essential for learning

Most likely explanation: $\arg\max_{\mathbf{x}_{1:t}} P(\mathbf{x}_{1:t}|\mathbf{e}_{1:t})$
speech recognition, decoding with a noisy channel

Filtering

Aim: devise a **recursive** state estimation algorithm:

$$\mathbf{P}(\mathbf{X}_{t+1}|\mathbf{e}_{1:t+1}) = f(\mathbf{e}_{t+1}, \mathbf{P}(\mathbf{X}_t|\mathbf{e}_{1:t}))$$

$$\begin{aligned}\mathbf{P}(\mathbf{X}_{t+1}|\mathbf{e}_{1:t+1}) &= \mathbf{P}(\mathbf{X}_{t+1}|\mathbf{e}_{1:t}, \mathbf{e}_{t+1}) \\ &= \alpha\mathbf{P}(\mathbf{e}_{t+1}|\mathbf{X}_{t+1}, \mathbf{e}_{1:t})\mathbf{P}(\mathbf{X}_{t+1}|\mathbf{e}_{1:t}) \\ &= \alpha\mathbf{P}(\mathbf{e}_{t+1}|\mathbf{X}_{t+1})\mathbf{P}(\mathbf{X}_{t+1}|\mathbf{e}_{1:t})\end{aligned}$$

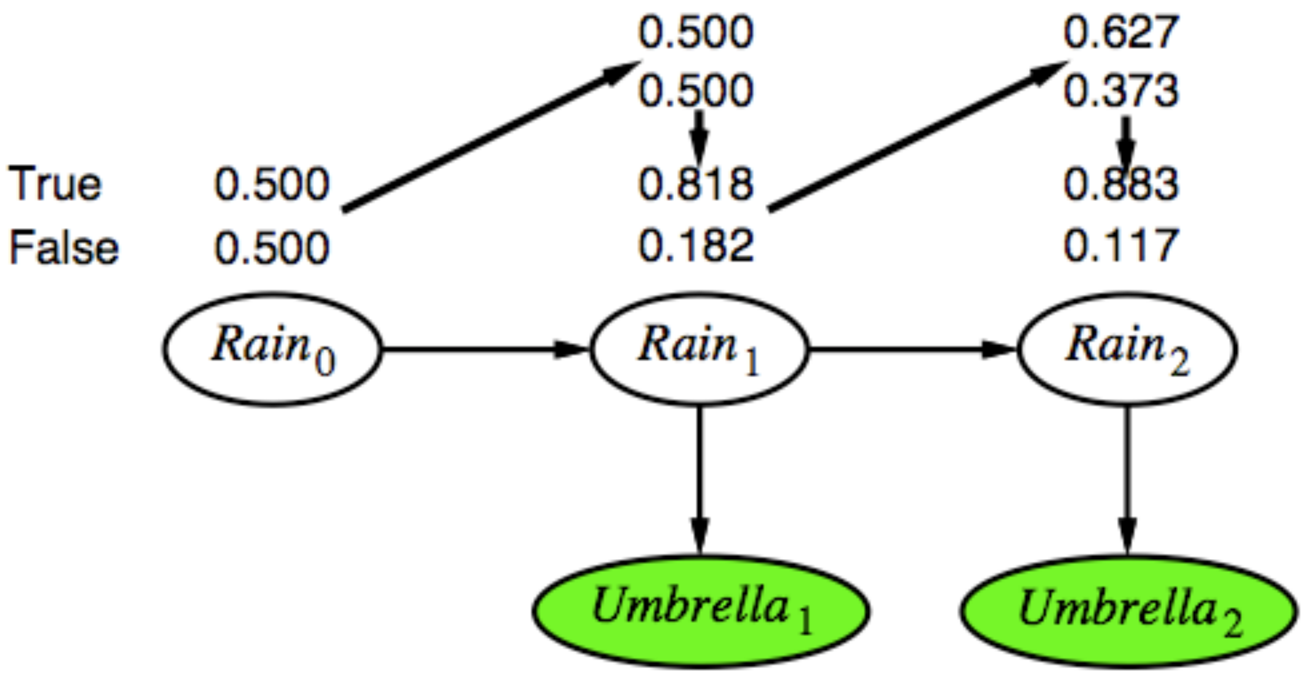
I.e., prediction + estimation. Prediction by summing out $\mathbf{X}_t$:

$$\begin{aligned}\mathbf{P}(\mathbf{X}_{t+1}|\mathbf{e}_{1:t+1}) &= \alpha\mathbf{P}(\mathbf{e}_{t+1}|\mathbf{X}_{t+1})\Sigma_{\mathbf{x}_t}\mathbf{P}(\mathbf{X}_{t+1}|\mathbf{x}_t, \mathbf{e}_{1:t})P(\mathbf{x}_t|\mathbf{e}_{1:t}) \\ &= \alpha\mathbf{P}(\mathbf{e}_{t+1}|\mathbf{X}_{t+1})\Sigma_{\mathbf{x}_t}\mathbf{P}(\mathbf{X}_{t+1}|\mathbf{x}_t)P(\mathbf{x}_t|\mathbf{e}_{1:t})\end{aligned}$$

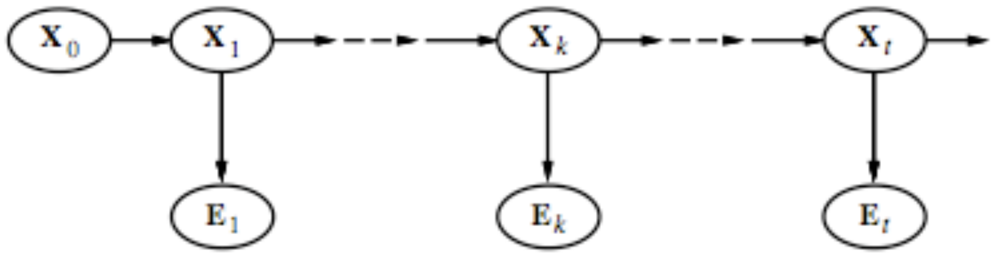
$\mathbf{f}_{1:t+1} = \textsc{Forward}(\mathbf{f}_{1:t}, \mathbf{e}_{t+1})$ where $\mathbf{f}_{1:t} = \mathbf{P}(\mathbf{X}_t|\mathbf{e}_{1:t})$

Time and space **constant** (independent of t)

Filtering Example



Smoothing



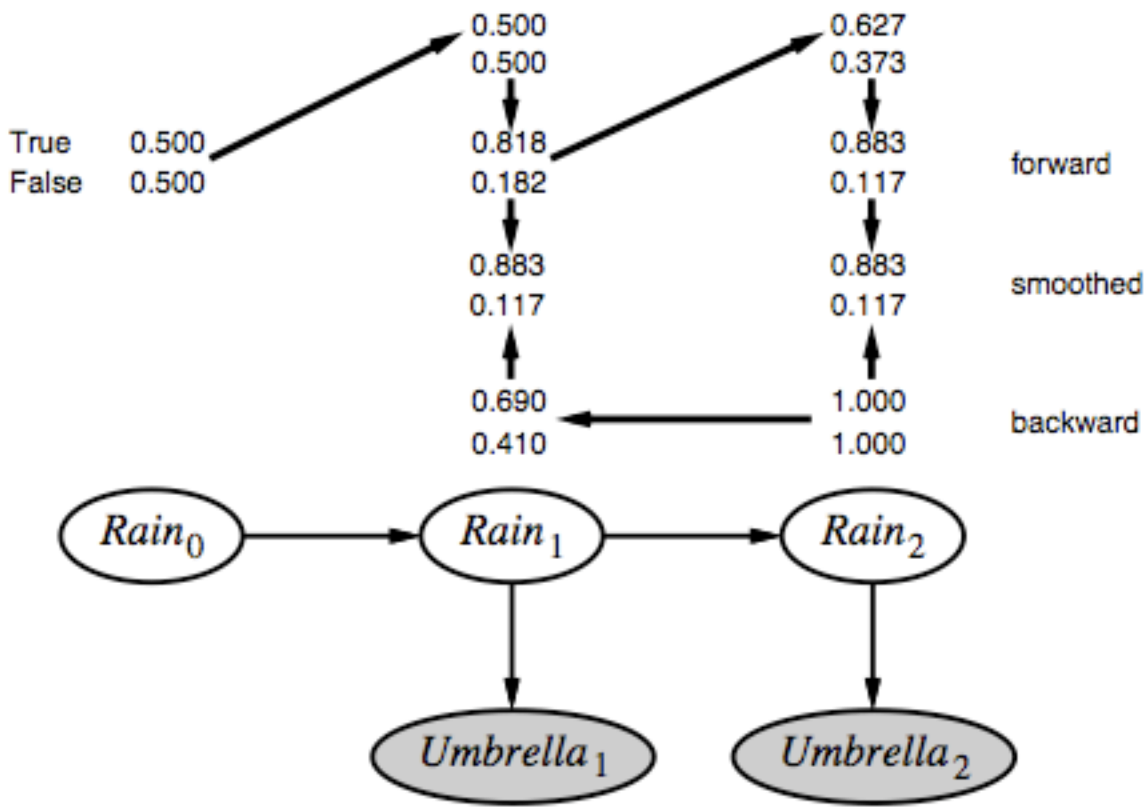
Divide evidence $\mathbf{e}_{1:t}$ into $\mathbf{e}_{1:k}$, $\mathbf{e}_{k+1:t}$:

$$\begin{aligned}
\mathbf{P}(\mathbf{X}_k|\mathbf{e}_{1:t}) &= \mathbf{P}(\mathbf{X}_k|\mathbf{e}_{1:k}, \mathbf{e}_{k+1:t}) \\
&= \alpha\mathbf{P}(\mathbf{X}_k|\mathbf{e}_{1:k})\mathbf{P}(\mathbf{e}_{k+1:t}|\mathbf{X}_k, \mathbf{e}_{1:k}) \\
&= \alpha\mathbf{P}(\mathbf{X}_k|\mathbf{e}_{1:k})\mathbf{P}(\mathbf{e}_{k+1:t}|\mathbf{X}_k) \\
&= \alpha\mathbf{f}_{1:k}\mathbf{b}_{k+1:t}
\end{aligned}$$

Backward message computed by a backwards recursion:

$$\begin{aligned}
\mathbf{P}(\mathbf{e}_{k+1:t}|\mathbf{X}_k) &= \Sigma_{\mathbf{x}_{k+1}}\mathbf{P}(\mathbf{e}_{k+1:t}|\mathbf{X}_k, \mathbf{x}_{k+1})\mathbf{P}(\mathbf{x}_{k+1}|\mathbf{X}_k) \\
&= \Sigma_{\mathbf{x}_{k+1}}P(\mathbf{e}_{k+1:t}|\mathbf{x}_{k+1})\mathbf{P}(\mathbf{x}_{k+1}|\mathbf{X}_k) \\
&= \Sigma_{\mathbf{x}_{k+1}}P(\mathbf{e}_{k+1}|\mathbf{x}_{k+1})P(\mathbf{e}_{k+2:t}|\mathbf{x}_{k+1})\mathbf{P}(\mathbf{x}_{k+1}|\mathbf{X}_k)
\end{aligned}$$

Smoothing



Forward–backward algorithm: cache forward messages along the way
Time linear in t (polytree inference), space $O(t|\mathbf{f}|)$

Most likely explanation

Most likely sequence $\neq$ sequence of most likely states!!!!

Most likely path to each $\mathbf{x}_{t+1}$
= most likely path to **some** $\mathbf{x}_t$ plus one more step

$$\max_{\mathbf{x}_1 \ldots \mathbf{x}_t} \mathbf{P}(\mathbf{x}_1, \ldots, \mathbf{x}_t, \mathbf{X}_{t+1} | \mathbf{e}_{1:t+1})$$
$$= \mathbf{P}(\mathbf{e}_{t+1} | \mathbf{X}_{t+1}) \max_{\mathbf{x}_t} \left(\mathbf{P}(\mathbf{X}_{t+1} | \mathbf{x}_t) \max_{\mathbf{x}_1 \ldots \mathbf{x}_{t-1}} P(\mathbf{x}_1, \ldots, \mathbf{x}_{t-1}, \mathbf{x}_t | \mathbf{e}_{1:t}) \right)$$

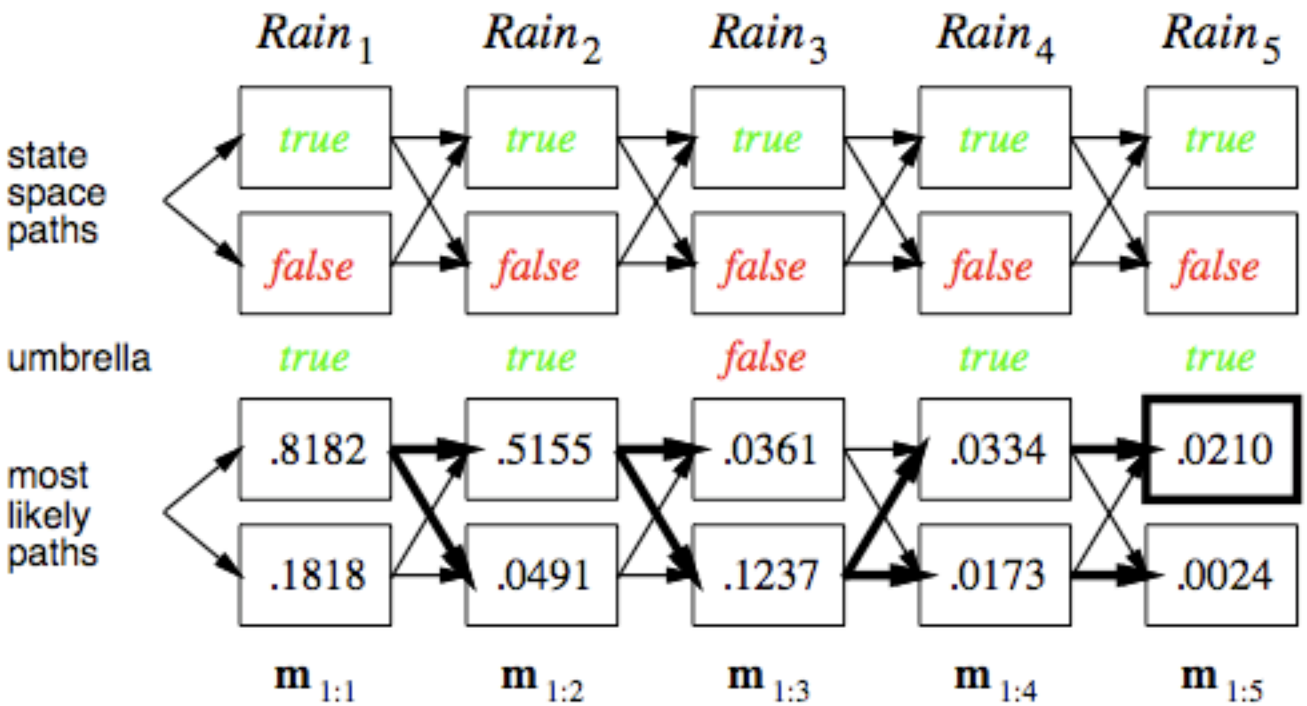
Identical to filtering, except $\mathbf{f}_{1:t}$ replaced by

$$\mathbf{m}_{1:t} = \max_{\mathbf{x}_1 \ldots \mathbf{x}_{t-1}} \mathbf{P}(\mathbf{x}_1, \ldots, \mathbf{x}_{t-1}, \mathbf{X}_t | \mathbf{e}_{1:t}),$$

I.e., $\mathbf{m}_{1:t}(i)$ gives the probability of the most likely path to state i.
Update has sum replaced by max, giving the Viterbi algorithm:

$$\mathbf{m}_{1:t+1} = \mathbf{P}(\mathbf{e}_{t+1} | \mathbf{X}_{t+1}) \max_{\mathbf{x}_t} (\mathbf{P}(\mathbf{X}_{t+1} | \mathbf{x}_t) \mathbf{m}_{1:t})$$

Viterbi example



Hidden Markov Models

$\mathbf{X}_t$ is a single, discrete variable (usually $\mathbf{E}_t$ is too)
Domain of X_t is $\{1, \ldots, S\}$

Transition matrix $\mathbf{T}_{ij} = P(X_t = j | X_{t-1} = i)$, e.g., $\begin{pmatrix} 0.7 & 0.3 \\ 0.3 & 0.7 \end{pmatrix}$

Sensor matrix $\mathbf{O}_t$ for each time step, diagonal elements $P(e_t | X_t = i)$
e.g., with $U_1 = true$, $\mathbf{O}_1 = \begin{pmatrix} 0.9 & 0 \\ 0 & 0.2 \end{pmatrix}$

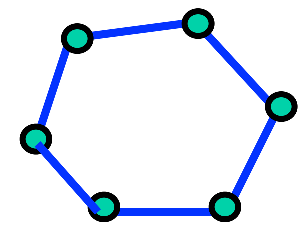
Forward and backward messages as column vectors:

$$\begin{aligned} \mathbf{f}_{1:t+1} &= \alpha \mathbf{O}_{t+1} \mathbf{T}^\top \mathbf{f}_{1:t} \\ \mathbf{b}_{k+1:t} &= \mathbf{T} \mathbf{O}_{k+1} \mathbf{b}_{k+2:t} \end{aligned}$$

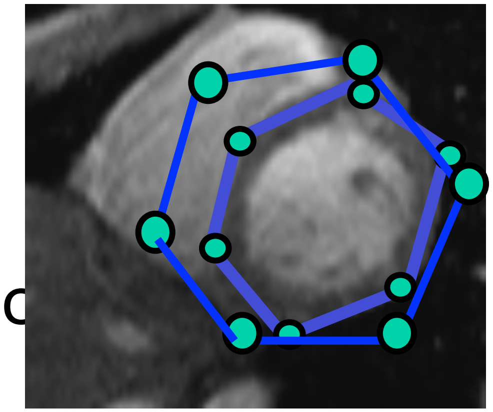
Forward-backward algorithm needs time $O(S^2 t)$ and space $O(St)$

Recap: deformable contour

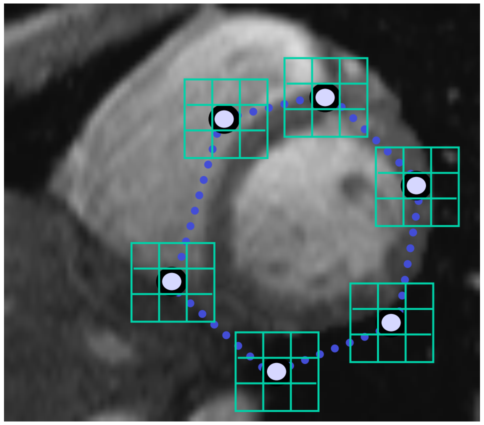
- A simple elastic snake is defined by:
  - A set of *n* points,
  - An internal energy term (tension, bending, plus optional shape prior)
  - An external energy term (gradient-based)



- To use to segment an object:
  - Initialize in the vicinity of the objec
  - Modify the points to minimize the total energy

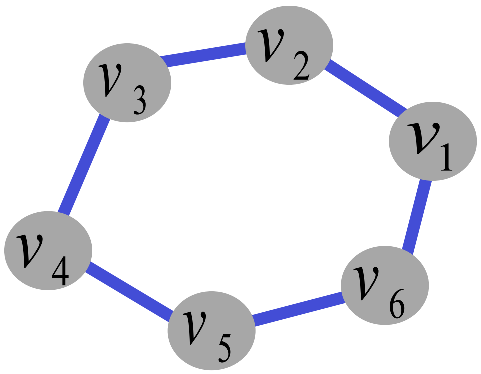


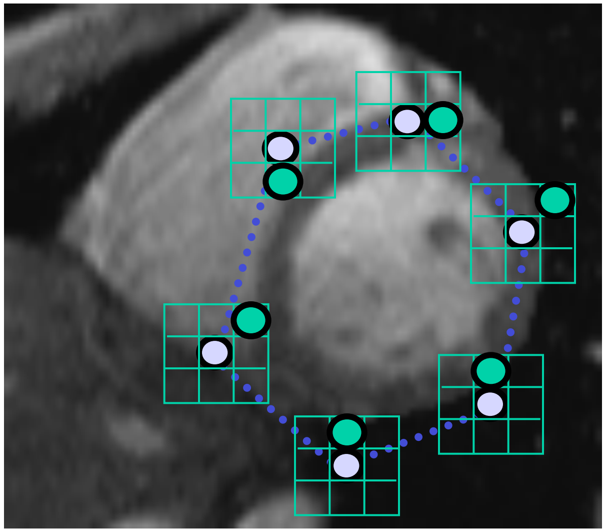
Energy minimization: greedy



- For each point, search window around it and move to where energy function is minimal
  - Typical window size, e.g., 5 x 5 pixels
- Stop when predefined number of points have not changed in last iteration, or after max number of iterations
- Note:
  - Convergence not guaranteed
  - Need decent initialization

Energy minimization: dynamic programming





With this form of the energy function, we can minimize using dynamic programming, with the *Viterbi* algorithm.

Energy minimization: dynamic programming

- Possible because snake energy can be rewritten as a sum of pair-wise interaction potentials:

$$E_{total}(\nu_1, \ldots, \nu_n) = \sum_{i=1}^{n-1} E_i(\nu_i, \nu_{i+1})$$

- Or sum of triple-interaction potentials.

$$E_{total}(\nu_1, \ldots, \nu_n) = \sum_{i=1}^{n-1} E_i(\nu_{i-1}, \nu_i, \nu_{i+1})$$

Snake energy: pair-wise interactions

$$E_{total}(x_1,\ldots,x_n,y_1,\ldots,y_n) = -\sum_{i=1}^{n-1} |G_x(x_i,y_i)|^2 + |G_y(x_i,y_i)|^2$$

$$+\,\alpha \sum_{i=1}^{n-1} (x_{i+1}-x_i)^2 + (y_{i+1}-y_i)^2$$

Re-writing the above with $v_i = (x_i, y_i)$:

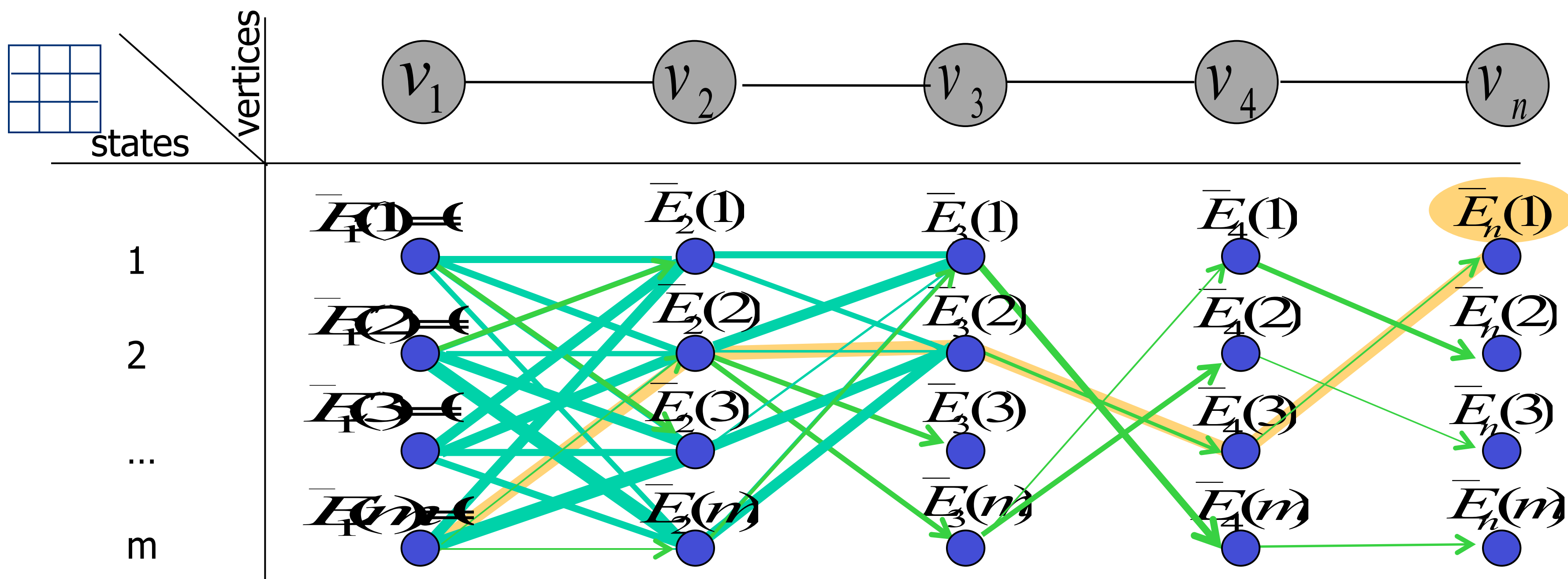
$$E_{total}(v_1,\ldots,v_n) = -\sum_{i=1}^{n-1} \|G(v_i)\|^2 + \alpha \sum_{i=1}^{n-1} \|v_{i+1}-v_i\|^2$$

$$E_{total} = E_1(v_1,v_2) + E_2(v_2,v_3) + \ldots + E_{n-1}(v_{n-1},v_n)$$

where $E_i(v_i,v_{i+1}) = -\|G(v_i)\|^2 + \alpha\|v_{i+1}-v_i\|^2$

Viterbi algorithm

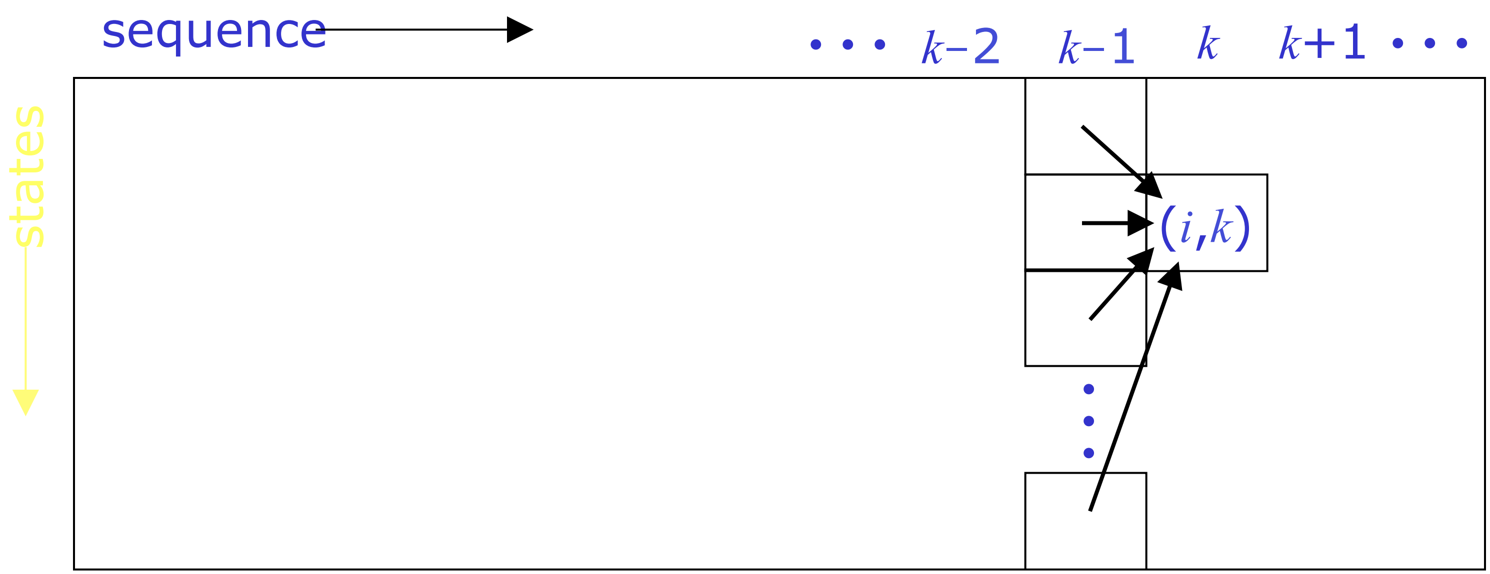
Main idea: determine optimal position (state) of predecessor, for each possible position of self. Then backtrack from best state for last vertex.



Complexity: $O(nm^2)$ vs. brute force search _____?

The Viterbi Algorithm

$$V(i,k) = \begin{cases} \max\limits_{j} V(j,k-1)P_t(q_i \mid q_j)P_e(x_k, q_i) & \text{if } k > 0, \\ P_t(q_i \mid q_0)P_e(x_0 \mid q_i) & \text{if } k = 0. \end{cases}$$



$$\phi_{\max} = \underset{\phi_{i,L-1}}{\arg\max}\, V(i, L-1)P_t(q_0 \mid q_i)$$

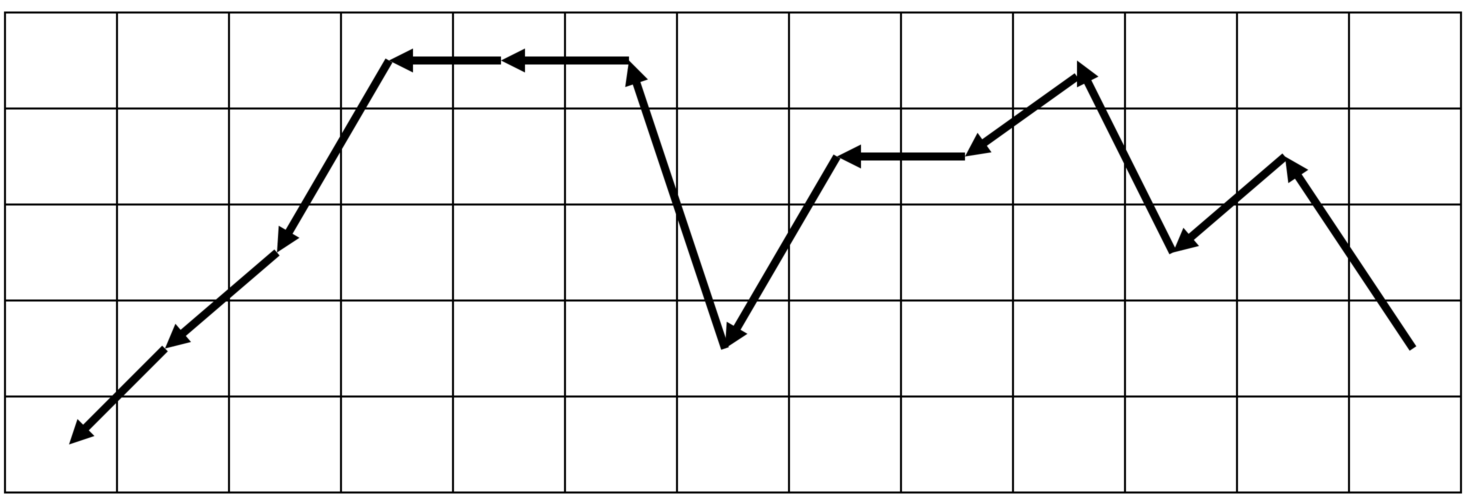
Duke
UNIVERSITY

Viterbi: Traceback

$$V(i,k) = \begin{cases} \max\limits_{j} V(j,k-1) P_t(q_i \mid q_j) P_e(x_k \mid q_i) & \text{if } k > 0, \\ P_t(q_i \mid q^0) P_e(x_0 \mid q_i) & \text{if } k = 0. \end{cases}$$

$$T(i,k) = \begin{cases} \underset{j}{\operatorname{argmax}}\, V(j,k-1) P_t(q_i \mid q_j) P_e(x_k \mid q_i) & \text{if } k > 0, \\ 0 & \text{if } k = 0. \end{cases}$$

$T(\ T(\ T(\ \ldots\ T(\ T(i, L-1), L-2)\ \ldots, 2), 1), 0) = 0$



Duke UNIVERSITY

Viterbi Algorithm in Pseudocode

$\lambda_{trans}[q_i] = \{q_j \mid P_t(q_i|q_j) > 0\}$

$\lambda_{emit}[s] = \{q_i \mid P_e(s|q_i) > 0\}$

```
procedure viterbi(Q,α,P_t,P_e,S,λ_trans,λ_emit)
    for k←0 up to |S|-1 do
      for i←0 up to |Q|-1 do
        V[i][k]←-∞;
        T[i][k]←NIL;
    for i←1 up to |Q|-1 do
      V[i][0]←log(P_t(q_i|q_0))+log(P_e(S[0]|q_i));
      if V[i][0]>-∞ then T[i][0]←0;
    for k←1 up to |S|-1 do
      foreach q_i∈λ_emit[S[k]] do
        foreach q_j∈λ_trans[ q_i]  do
          v←V[j][k-1]+log(P_t(q_i|q_j))+
                              log(P_e(S[k]|q_i));
          if v>V[i][k] then
            V[i][k]←v;
            T[i][k]←j;
    y←1;
    push φ,0;
    for i←2 up to |Q|-1 do
      if V[i][|S|-1]+log(P_t(q_0|q_i)) >
         V[y][|S|-1]+log(P_t(q_0|q_y)) then y←i;
    for k←|S|-1 down to 0 do
      push φ,y;
      y←T[y][k];
    push φ,0;
    return φ;
```

initialization

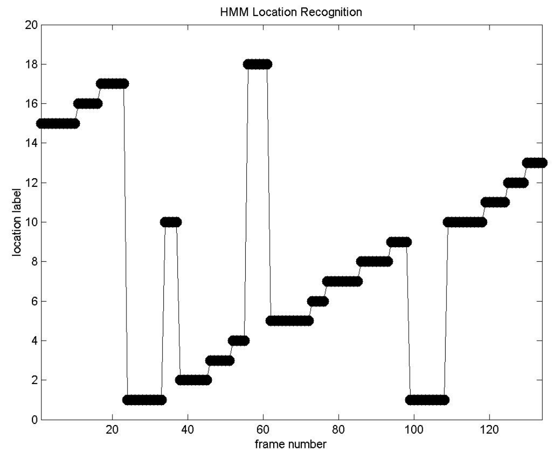
fill out main part of DP matrix

choose best state from last column in DP matrix

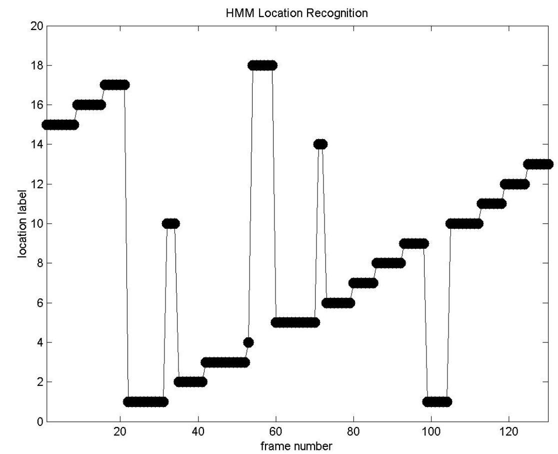
traceback

HMM Recognition

With HMM

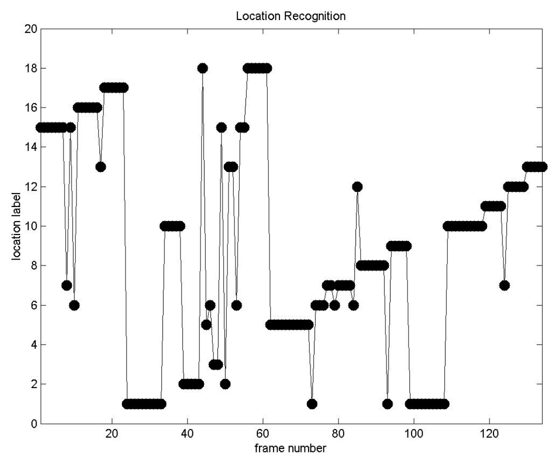


96.3%

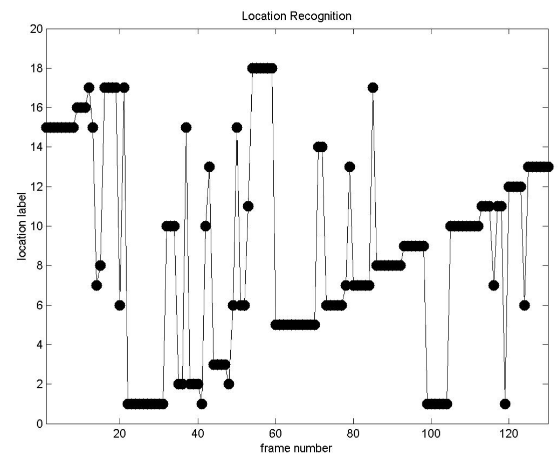


95.4%

Without HMM



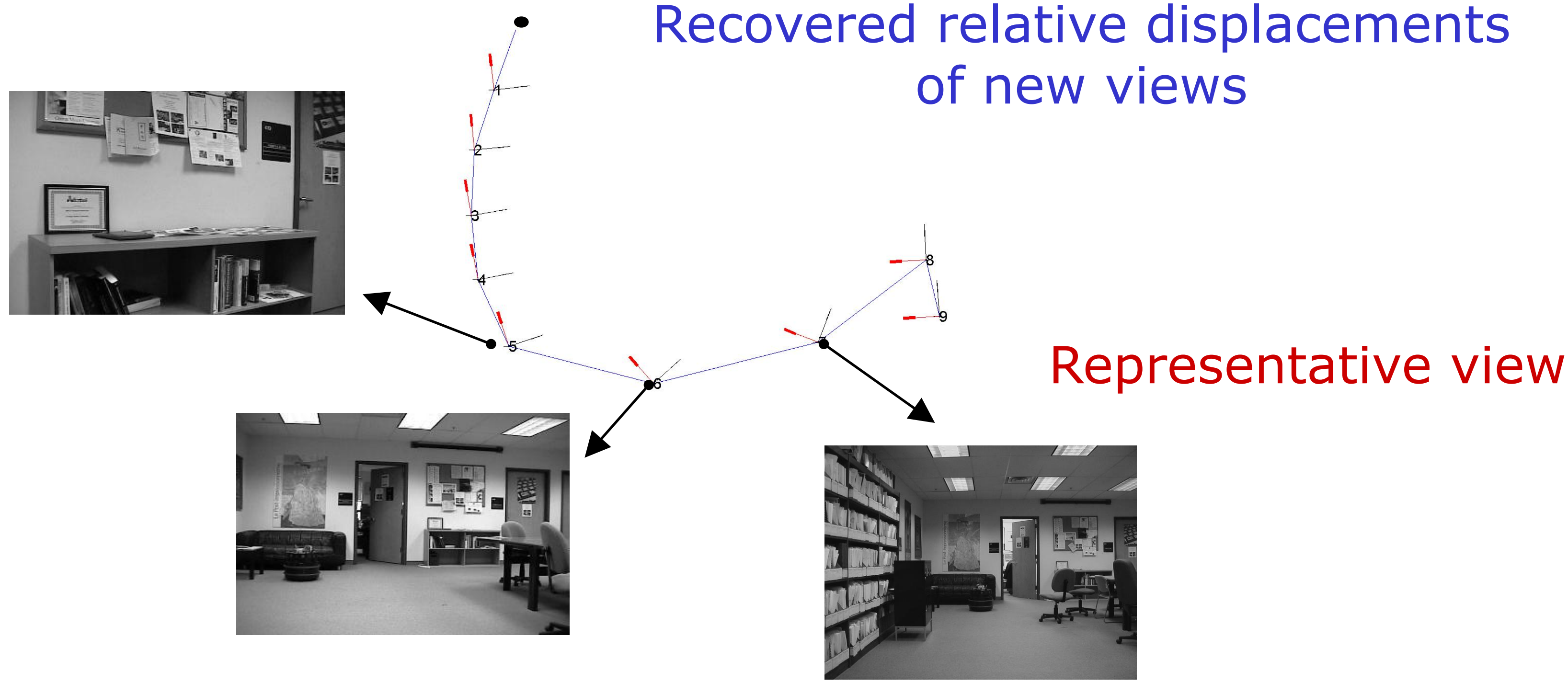
82%



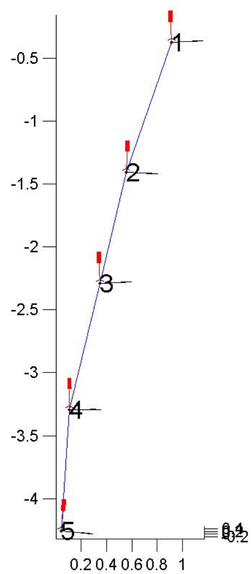
83%

Metric Localization within Location

1. Given closest representative view of the location
2. Establish exact correspondences between keypoints
3. Matching combining (epipolar) geometry, keypoint descriptors and intrinsic scale
4. Compute relative pose with respect to the reference view (despite the unknown focal length)

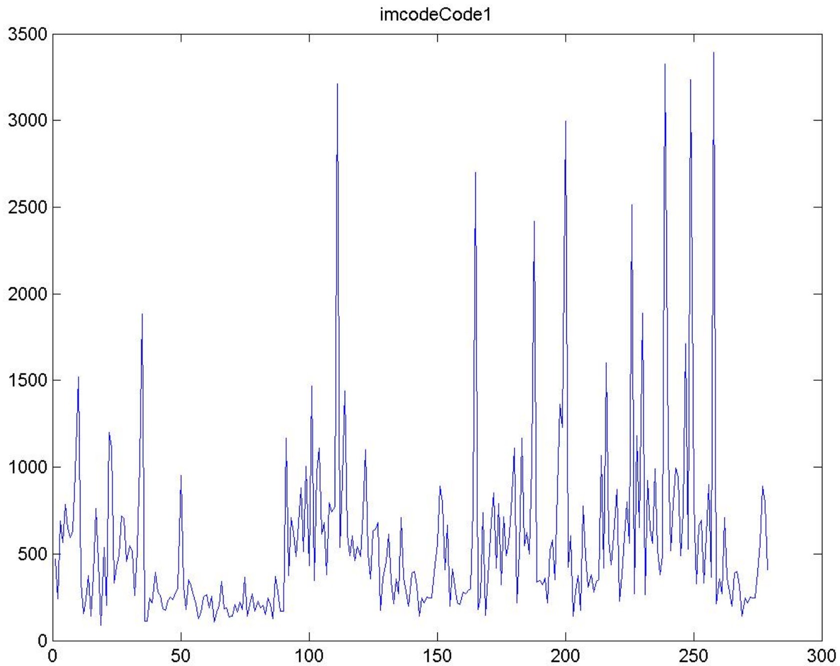


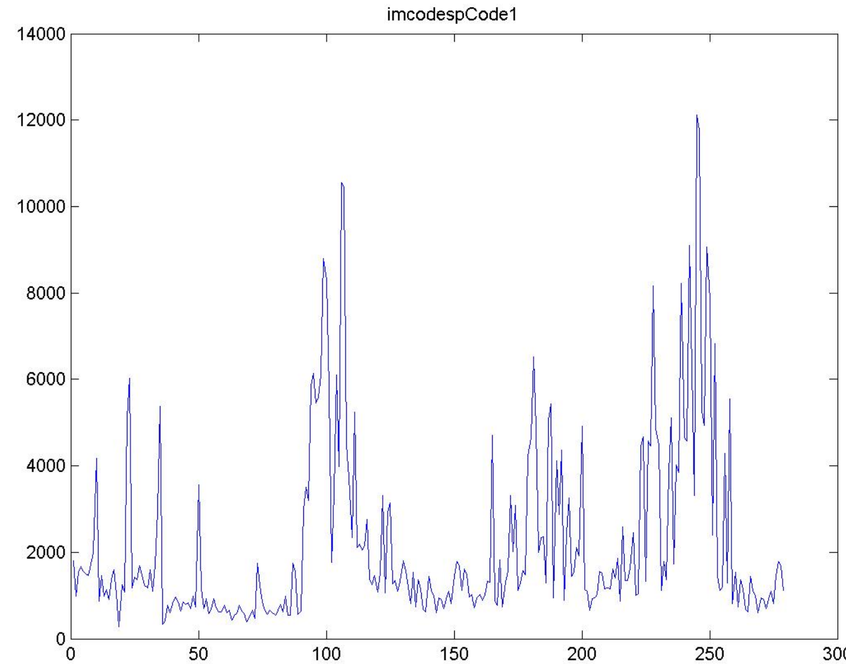
Metric Localization within Location

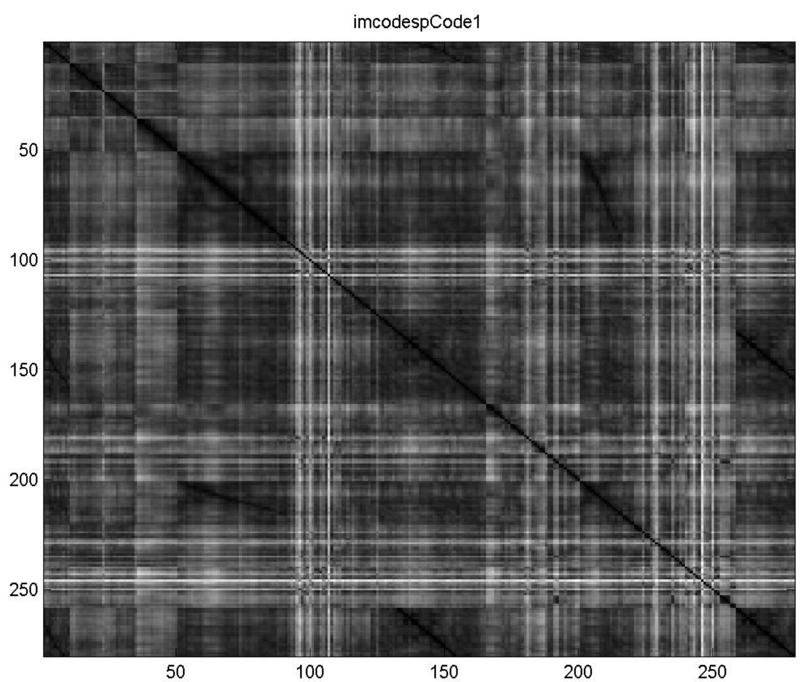


Conclusions and Future Work

- Robust and effective categorization and automatic segmentation of video into distinct locations and distinct categories (indoors, outdoors, office, hallway, crossing)
- Topological and metric localization using scale invariant features
- Extensions to outdoors environments (where the orientation cannot be coarsely quantized)
- Develop complete exploration strategies
- Enhancing matching and pose recovery methods for generic unstructured environments


imcodeCode1
3500
3000
2500
2000
1500
1000
500
0
0 50 100 150 200 250 300


imcodespCode1
14000
12000
10000
8000
6000
4000
2000
0
0 50 100 150 200 250 300


imcodespCode1
50
100
150
200
250
50 100 150 200 250

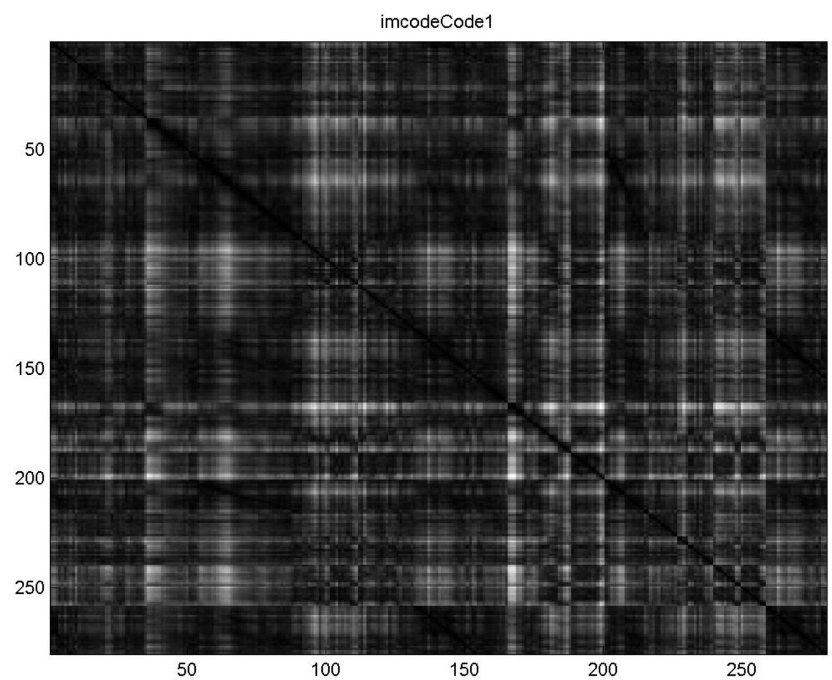
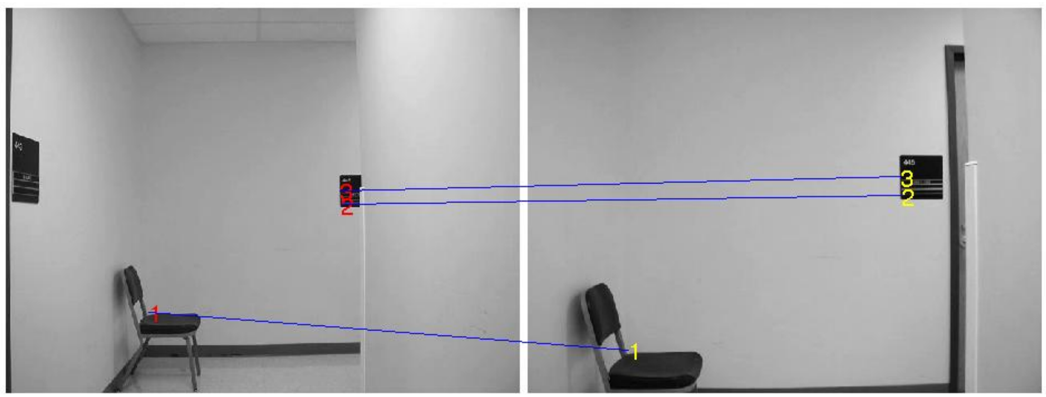

imcodeCode1
50
100
150
200
250
50 100 150 200 250

image 1100 (left), image 1101 (right), number of matches = 3



Pose Estimation

- Two view epipolar geometry
- Related Work [Sturm' 01, Agapito' 00, Ma et. al' 03]
- Calibrated case

$$\mathrm{x}_2^T \widehat{T} R \mathrm{x}_1 = \mathrm{x}_2^T E \mathrm{x}_1 = 0$$

- Essential matrix – planar case $R_z \in SO(3), T = [t_x, 0, t_z]^T$

$$E = \begin{bmatrix} 0 & -t_z & 0 \\ t_z c\theta + t_x s\theta & 0 & t_z s\theta - t_x c\theta \\ 0 & t_x & 0 \end{bmatrix}$$

- Partially calibrated case - unknown focal length

$$F = K^{-T} E K^{-1} \text{ with } K = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Pose Estimation

- Partially calibrated case - unknown focal length
- Fundamental matrix

$$F = K^{-T} \begin{bmatrix} 0 & f_1 & 0 \\ f_2 & 0 & f_3 \\ 0 & f_4 & 0 \end{bmatrix} K^{-1}$$

- Calibration constraints (Kruppa's equations)

$$FKK^T F^T = \lambda^2 \widehat{e} KK^T \widehat{e}^T$$

- With the epipole $e = [f_4, 0, -f_1]^T$
- In the planar motion case Kruppa's equations can be renormalized with

Focal Length Estimation

- Planar Kruppa's equations wi$e = [f_4, 0, -f_1]^T, \lambda = 1$

$$FKK^TF^T = \widehat{e}KK^T\widehat{e}^T$$

- Directly yields constraints on focal length

$$f_2^2 f^2 + f_3^2 = f_4^2 f^2 + f_1^2$$

- can be estimated in the closed form

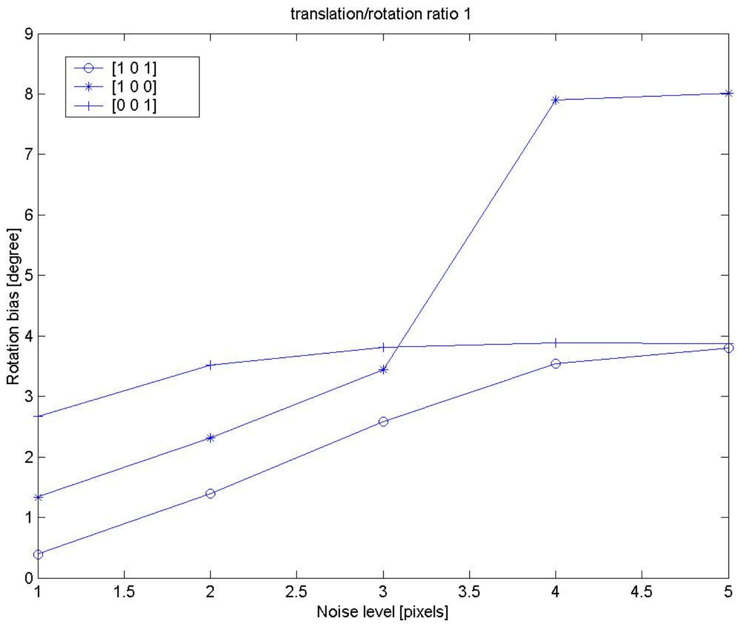
$$f = \sqrt{\frac{f_1^2 - f_3^2}{f_2^2 - f_4^2}}$$

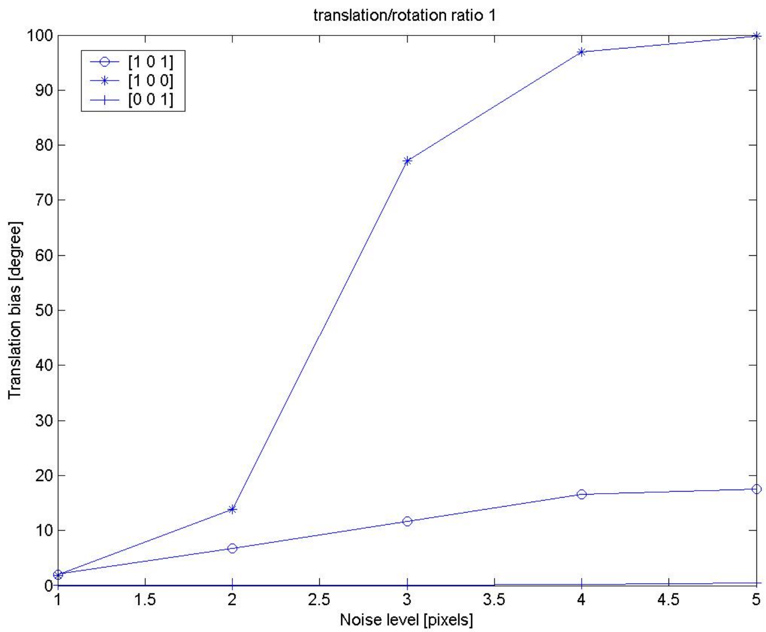
Robust Pose and Focal Length Estimation

- Modified random sampling strategy
- Incorporates the focal length constraint
  (enables faster convergence)

1. Generate number of hypothesis by sampling 4 points from the set of matches
2. Verify the which hypotheses satisfy the focal length constraint
3. Select the hypothesis which minimizes the total distance to the epipolar lines
4. Reject the matches with residual error above some threshold

Sensitivity of the motion estimates



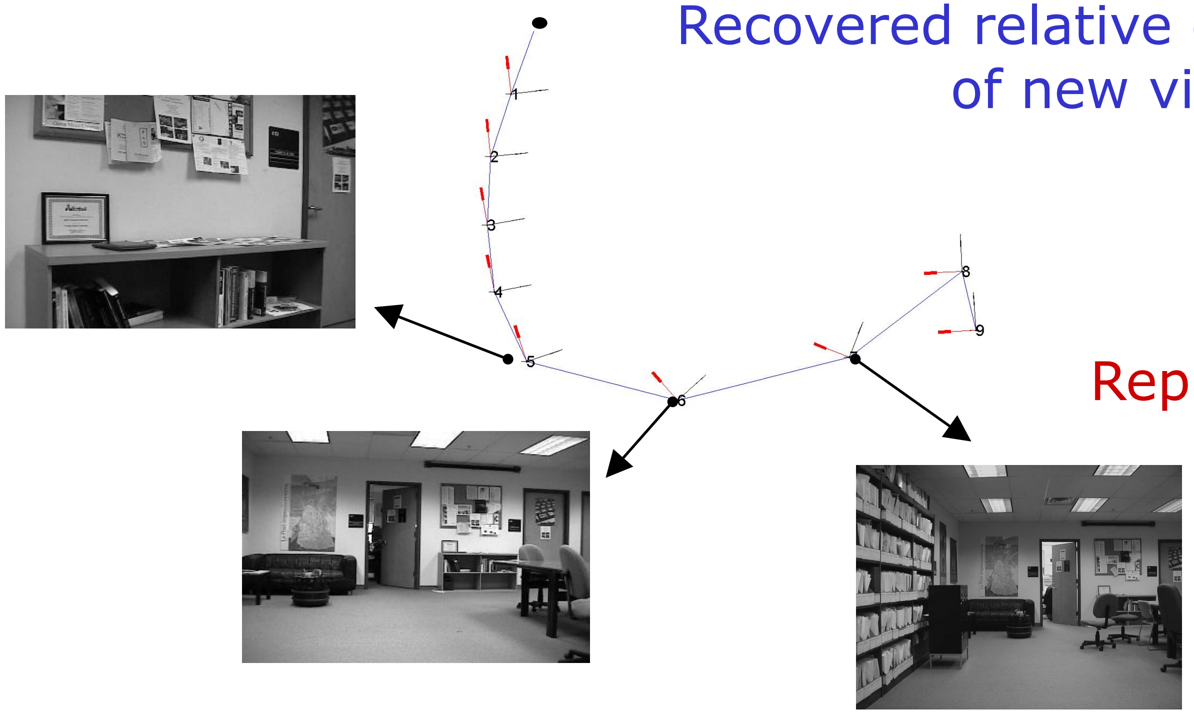


Simulation – 100 trials, different motion, error in correspondences measurements

Metric Localization within Location

1. Given closest representative view of the location
2. Establish exact correspondences between keypoints
3. Matching combining (epipolar) geometry,
   keypoint descriptors and intrinsic scale
4. Compute relative pose with respect to the reference view
   (despite the unknown focal length)

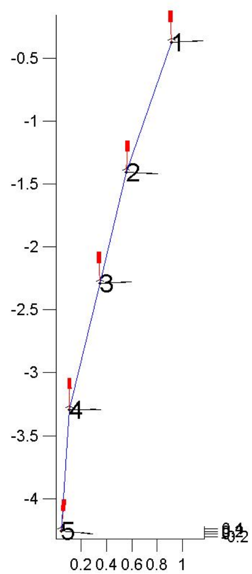
Recovered relative displacements of new views



Representative view

$$f = 624.25$$

Metric Localization within Location



$$f = 545.30$$

Conclusions and Future Work

- Robust and effective categorization and automatic segmentation of video into distinct locations and distinct categories (indoors, outdoors, office, hallway, crossing)
- Topological and metric localization using scale invariant features

- Exploit geometric relationships between features
- Alternative features/feature descriptors
- Extensions to outdoors environments

- Develop complete exploration strategies
- Improving the matching and pose recovery methods for generic unstructured environments