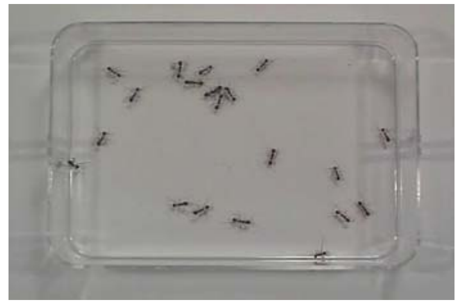
Sampling Methods for Bayesian Inference

A Tutorial

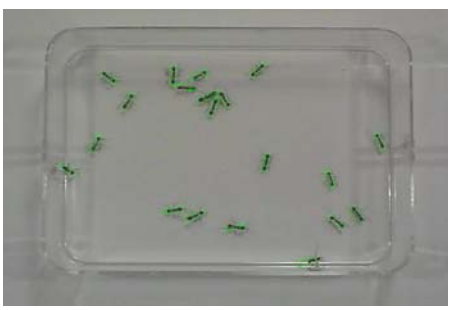
Frank Dellaert

Motivation

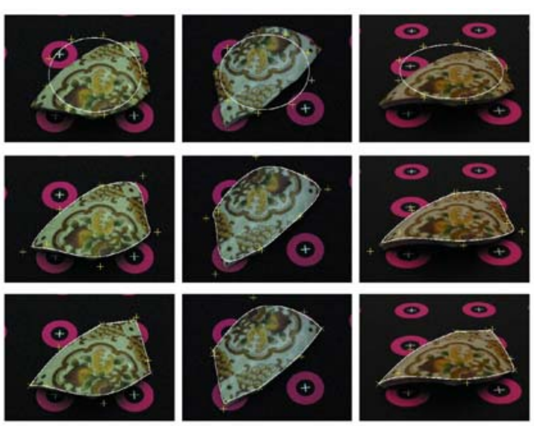
- How to track many INTERACTING targets ?

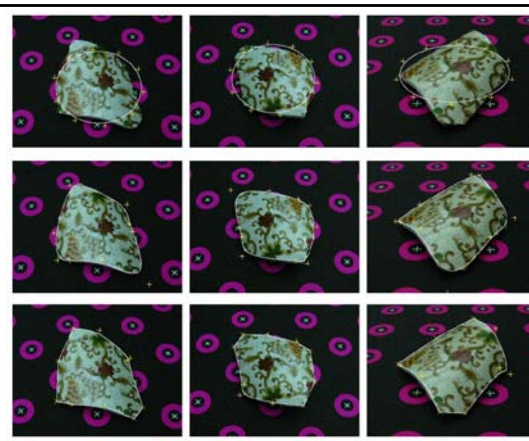


Results: MCMC

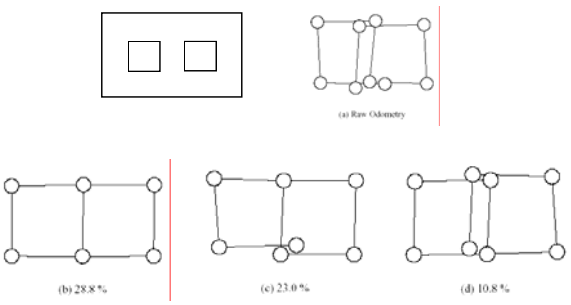


Dancers, q=10, n=500

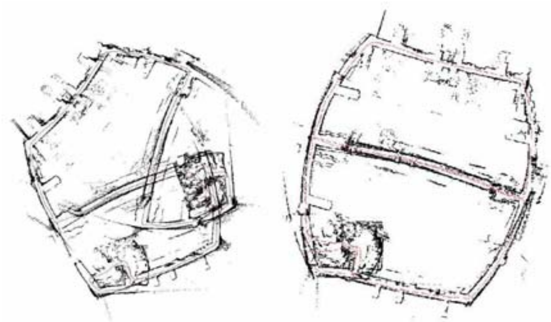




Probabilistic Topological Maps

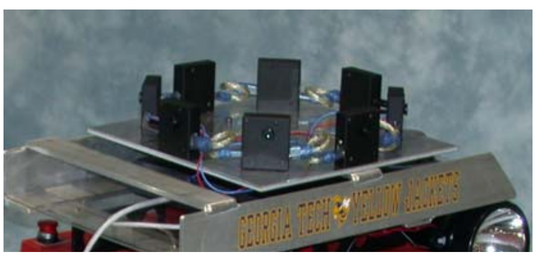


Results



Real-Time Urban Reconstruction

- 4D Atlanta, only real time, multiple cameras ☺
- Large scale SFM: closing the loop



Current Main Effort: 4D Atlanta

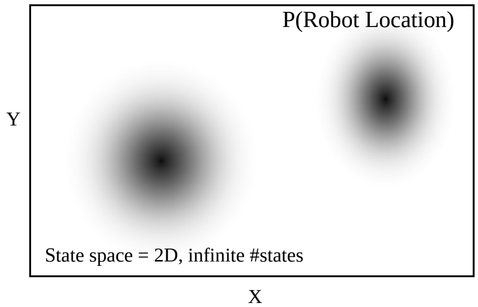
Goals

- **Bayesian paradigm** is a useful tool to
  - Represent knowledge
  - Perform inference
- **Sampling** is a nice way to implement the Bayesian paradigm, e.g. Condensation
- **Markov chain Monte Carlo** methods are a nice way to implement sampling

References

- **Neal**, Probabilistic Inference using MCMC Methods
- **Smith & Gelfand**, Bayesian Statistics Without Tears
- **MacKay**, Introduction to MC Methods
- **Gilks et al**, Introducing MCMC
- **Gilks et al**, MCMC in Practice

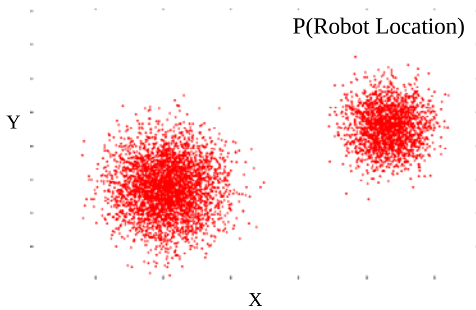
Probability of Robot Location



Density Representation

- Gaussian centered around mean x,y
- Mixture of Gaussians
- Finite element i.e. histogram
- Larger spaces -> We have a problem !

Sampling as Representation



Sampling Advantages

- Arbitrary densities
- Memory = O(#samples)
- Only in "Typical Set"
- Great visualization tool !

- minus: Approximate

How to Sample ?

- Target Density P(x)
- Assumption: we can evaluate P(x) up to an arbitrary multiplicative constant

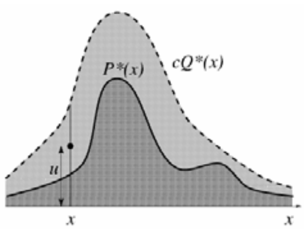
- Why can't we just sample from P(x) ??

How to Sample ?

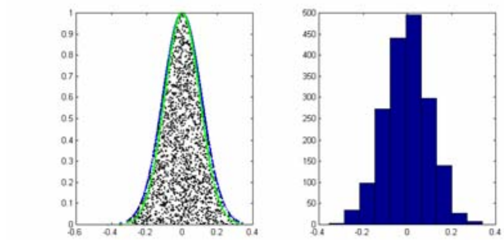
- Numerical Recipes in C, Chapter 7
- Transformation method: Gaussians etc…
- Rejection sampling
- Importance sampling
- Markov chain Monte Carlo

Rejection Sampling

- Target Density P
- Proposal Density Q
- P and Q need only be known up to a factor: P* and Q*
- must exist c such that cQ*>=P* for all x

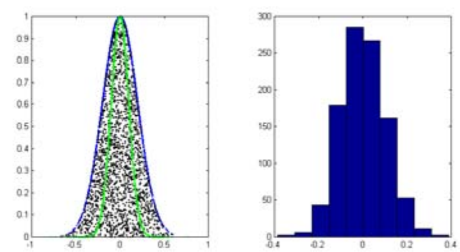


The Good…



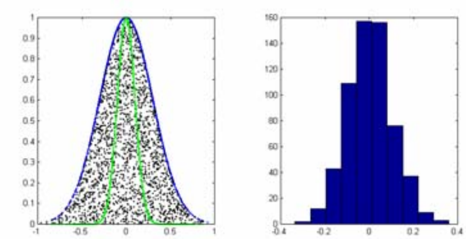
9% Rejection Rate

…the Bad…



50% Rejection Rate

…and the Ugly.



70% Rejection Rate

Mean and Variance of a Sample

Mean

$$\mu = \int_x x P(x) dx$$

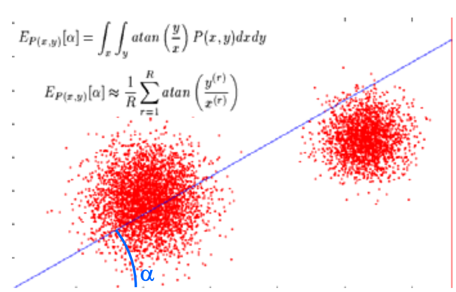
$$\mu \approx \frac{1}{R} \sum_{r=1}^{R} x^{(r)}$$

Variance (1D)

$$\sigma^2 = \int_x (x-\mu)^2 P(x) dx$$

$$\sigma^2 \approx \frac{1}{R} \sum_{r=1}^{R} (x^{(r)} - \hat{\mu})^2$$

Monte Carlo Expected Value



Expected angle = 30°

Monte Carlo Estimates (General)

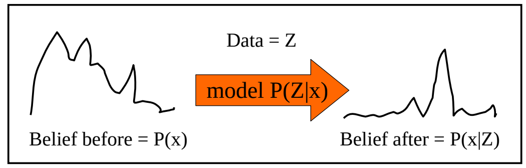
- Estimate expectation of any function f:

$$E_{P(x)}[f(x)] = \int_x f(x)P(x)d^N x$$

$$E_{P(x)}[f(x)] \approx \frac{1}{R}\sum_{r=1}^{R} f(x^{(r)})$$

Bayes Law

$$P(x|Z) \sim P(Z|x)P(x)$$



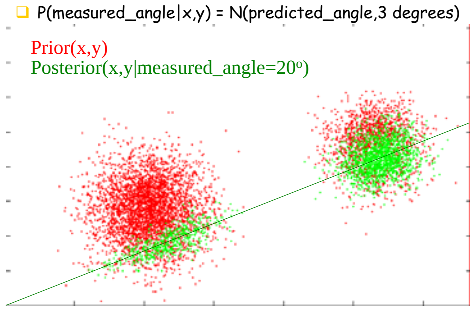
Prior Distribution of x

Likelihood of x given Z

Posterior Distribution of x given Z

Inference by Rejection Sampling

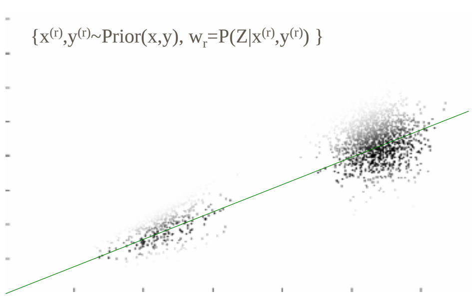
- P(measured_angle|x,y) = N(predicted_angle,3 degrees)



Importance Sampling

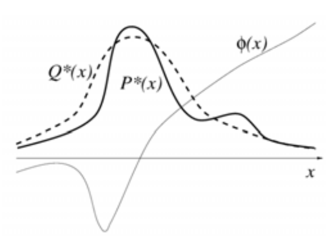
- Good Proposal Density would be: prior !
- Problem:
  - No guaranteed c s.t. c P(x)>=P(x|z) for all x
- Idea:
  - sample from P(x)
  - give each sample $x^{(r)}$ a importance weight equal to $P(Z|x^{(r)})$

Example Importance Sampling



Importance Sampling (general)

- Sample $x^{(r)}$ from Q^*
- $w_r = P^*(x^{(r)})/Q^*(x^{(r)})$



Important Expectations

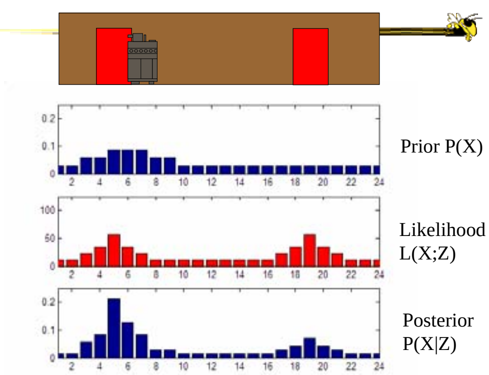
- Any expectation using weighted average:

$$w_r = \frac{P^*(x^{(r)})}{Q^*(x^{(r)})}$$

$$E_{P(x)}[f(x)] \approx \frac{\sum_{r=1}^{R} w_r f(x^{(r)})}{\sum_{r=1}^{R} w_r}$$

Particle Filtering

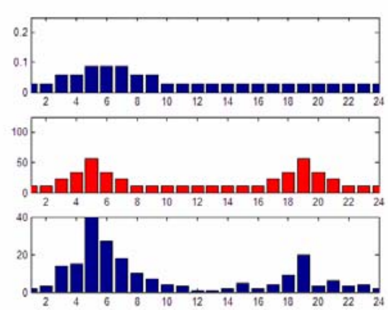
1D Robot Localization



Importance Sampling

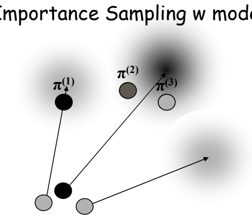
- Histogram approach does not scale
- Monte Carlo Approximation
- Sample from P(X|Z) by:
  - sample from prior P(x)
  - weight each sample $x^{(r)}$ using an importance weight equal to likelihood $L(x^{(r)};Z)$

1D Importance Sampling



Particle Filter

= Recursive Importance Sampling w modeled dynamics

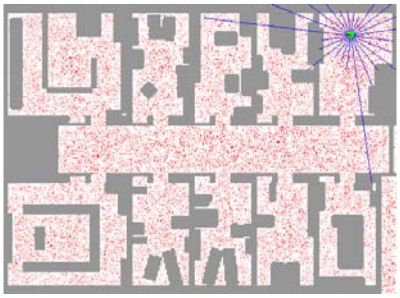


$$\pi_t^{(s)} = P(Z_t|X_t^{(s)})$$

First appeared in 70's, re-discovered by Kitagawa, Isard, …

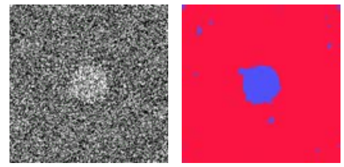
3D Particle filter for robot pose: Monte Carlo Localization

Dellaert, Fox & Thrun ICRA 99



Segmentation Example

- Binary Segmentation of image



Probability of a Segmentation

- Very high-dimensional
- 256*256 pixels = 65536 pixels
- Dimension of state space N = 65536 !!!!

- # binary segmentations = finite !
- 65536^2 = 4,294,967,296

Representation P(Segmentation)

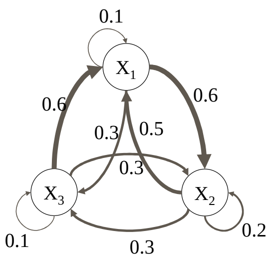
- Histogram ? I don't think so !
- Assume pixels independent
  $P(x_1x_2x_2...)=P(x_1)P(x_2)P(x_3)...$
- Markov Random Fields
  - Pixel is independent given its neighbors
- Clearly a problem !
- Giveaway: samples !!!

Sampling in High-dimensional Spaces

- Exact schemes ?
  - If only we were so lucky !
- Rejection Sampling
  - Rejection rate increase with N -> 100%
- Importance Sampling
  - Same problem: vast majority weights -> 0

Markov Chains

A simple Markov chain



K= [

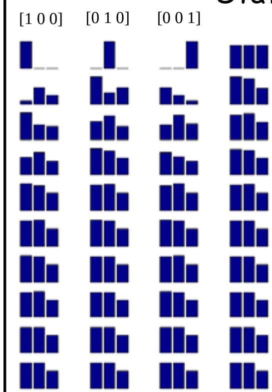
0.1 0.5 0.6

0.6 0.2 0.3

0.3 0.3 0.1

]

Stationary Distribution



q_0

$q_1 = K\, q_0$

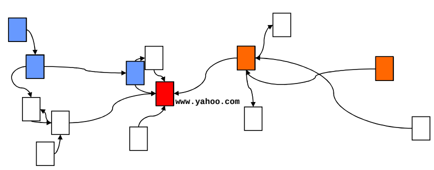
$q_2 = K\, q_1 = K^2\, q_0$

$q_3 = K\, q_2 = K^2\, q_1 = K^3\, q_0$

$q_{10} = K\, q_9 = \ldots\, K^{10}\, q_0$

The Web as a Markov Chain

Where do we end up if we click hyperlinks randomly ?



Answer: stationary distribution !

Eigen-analysis

K =

0.1000	0.5000	0.6000
0.6000	0.2000	0.3000
0.3000	0.3000	0.1000

E =

0.6396	0.7071	-0.2673
0.6396	-0.7071	0.8018
0.4264	0.0000	-0.5345

D =

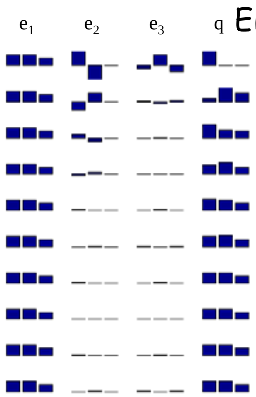
1.0000	0	0
0	-0.4000	0
0	0	-0.2000

KE = ED

Eigenvalue v_1 always 1

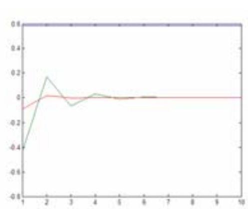
Stationary = e_1/sum(e_1)
i.e. Kp = p

Eigen-analysis



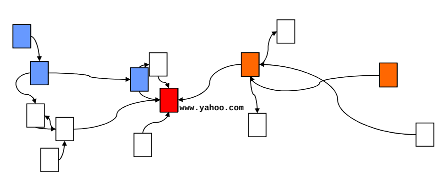
$q_n = K^n\, q_0 = E\, D^n\, c$

$= p + c_2\, v_2^{\,n}\, e_2 + c_3\, v_3^{\,n}\, e_{3+\ldots}$



Google Pagerank

Pagerank == First Eigenvector of the Web Graph !



Computation assumes a 15% “random restart” probability

Sergey Brin and Lawrence Page , The anatomy of a large-scale hypertextual {Web} search engine, Computer Networks and ISDN Systems, 1998

Markov chain Monte Carlo

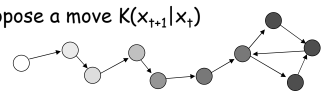
Brilliant Idea!

- Published June 1953
- Top 10 algorithm !

- Set up a Markov chain
- Run the chain until stationary
- All subsequent samples are from stationary distribution

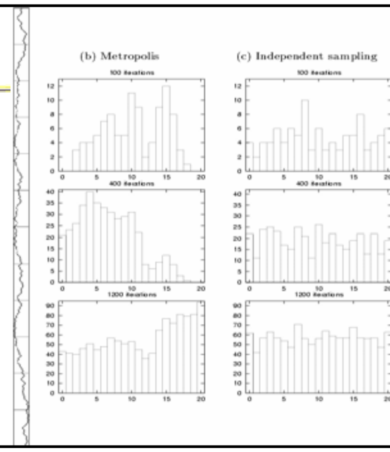
Markov chain Monte Carlo

- In high-dimensional spaces:
  - Start at $x_0 \sim q_0$
  - Propose a move $K(x_{t+1}|x_t)$



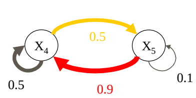
- K never stored as a big matrix ☺
- K as a function/search operator

Example



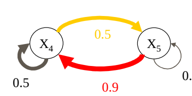
How do get the right chain ?

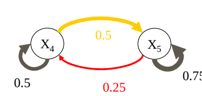
- Detailed balance:
  - $K(y|x)\ p(x) = K(x|y)\ p(y)$
- 0.5 * 9/14 = 0.9 * 5/14



Reject fraction of moves !

- Detailed balance:
  - $K(y|x)$ 1/3 = $K(x|y)$ 2/3
- 0.5 * 1/3 = a * 0.9 * 2/3
- a = 0.5 * 1/3 / (0.9 * 2/3) = 5/18





Metropolis-Hastings Algorithm

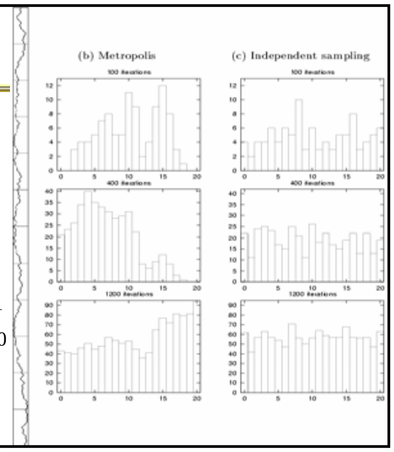
- pick $x^{(0)}$, then iterate over:
1. propose x' from $Q(x';x^{(t)})$
2. calculate ratio

$$a = \frac{P^*(x')}{P^*(x^{(t)})} \frac{Q(x^{(t)}; x')}{Q(x'; x^{(t)})}.$$

3. if a>1 accept $x^{(t+1)}=x'$
   else accept with probability a
   if rejected: $x^{(t+1)}=x^{(t)}$

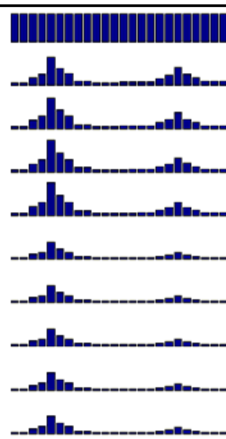
Again !

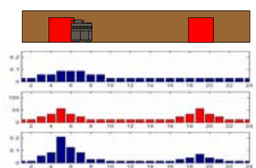
1. $x^{(0)}=10$
2. Proposal:
   $x'=x-1$ with Pr 0.5
   $x'=x+1$ with Pr 0.5
3. Calculate a:
   a=1 if x' in [0,20]
   a=0 if $x'=-1$ or $x'=21$
4. Accept if 1, reject if 0
5. Goto 2



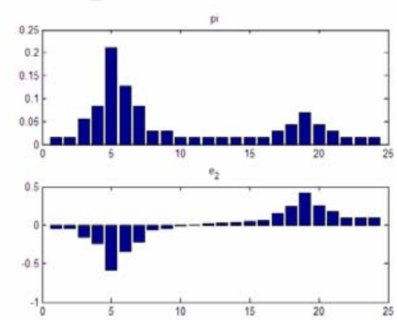
1D Robot Localization

Chain started at random
Converges to posterior



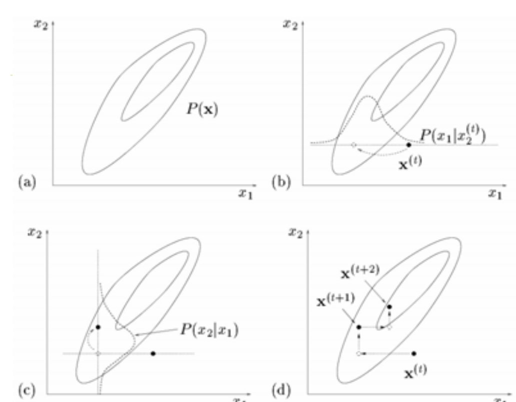


Localization Eigenvectors



Gibbs Sampling

- MCMC method that always accepts
- Algorithm:
  - alternate between x_1 and x_2
  - 1. sample from $x_1 \sim P(x_1|x_2)$
  - 2. sample from $x_2 \sim P(x_2|x_1)$
- Rationale: easy conditional distributions
- = Gauss-Seidel of samplers



Sampling Segmentations

- Prior model: Markov Random Field
- Likelihood: 1 or 0 plus Gaussian noise
- Gibbs Sampling method of choice
  - Conditional densities are easy in MRF

Samples from Prior

Forgiving Prior   Stricter Prior

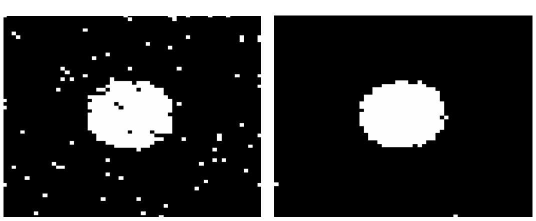
Sampling Prior

- P(being one|others)=
  - HIGH if many ones around you
  - LOW if many zeroes around you

Sampling Posterior

- P(being one|others)
  - pulled towards 0 if data close to 0
  - pushed towards 1 if data close to 1
  - and influence of prior…

Samples from Posterior



Forgiving Prior   Stricter Prior

Relation to Belief Propagation

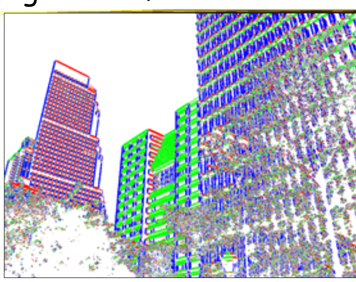
- In poly-trees: BP is exact
- In MRFs: BP is a variational approximation
- Computation is very similar to Gibbs
- Difference:
  - BP Can be faster in yielding a good estimate
  - BP exactly calculates the wrong thing
  - MC might take longer to converge
  - MC approximately calculates the right thing

Relation to Belief Propagation

Application: Edge Classification

Given vanishing points of a scene, classify each pixel according to vanishing direction

MAP Edge Classifications



Red: VP1 Green: VP2 Blue: VP3 Gray: Other White: Off

Bayesian Model

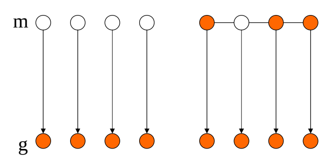
p(M | G,V) = p(G | M,V) p(M) / Z

M = classifications, G = gradient magnitude/direction, V = vanishing points

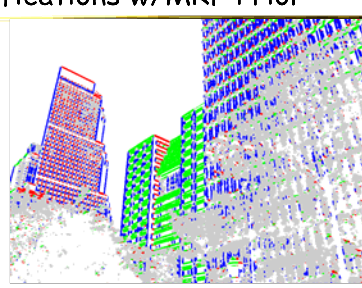
Independent Prior MRF Prior

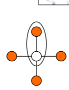
Prior: p(m)

Likelihood: p(g | m,V)



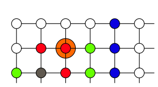
Classifications w/MRF Prior





Gibbs sampling over 4-neighbor lattice w/ clique potentials defined as: A if i=j, B if i <> j

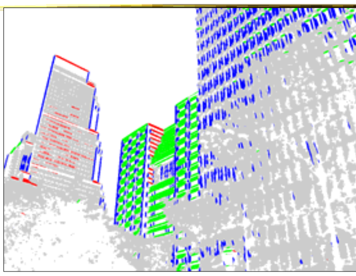
Gibbs Sampling & MRFs

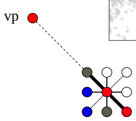


Sample from distribution over labels for one site conditioned on all other sites in its Markov blanket

Gibbs sampling approximates posterior distribution over classifications at each site (by iterating and accumulating statistics)

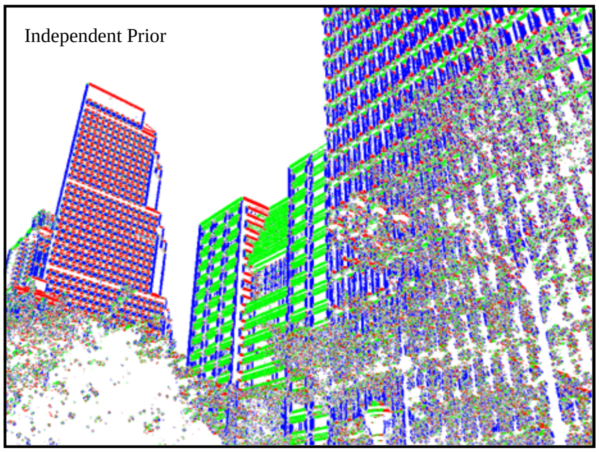
Directional MRF



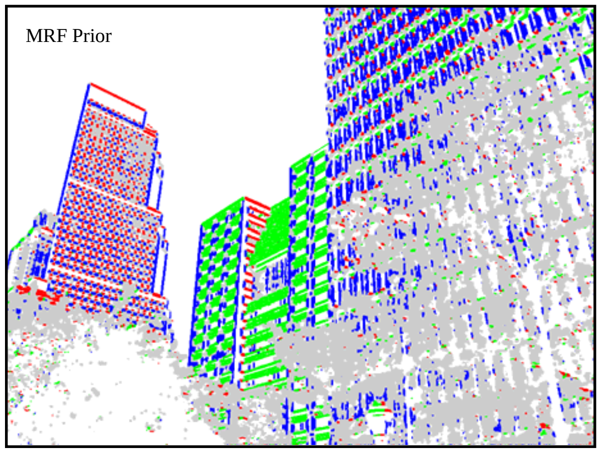


Give more weight to potentials of neighbors which lie along the vanishing direction of current model

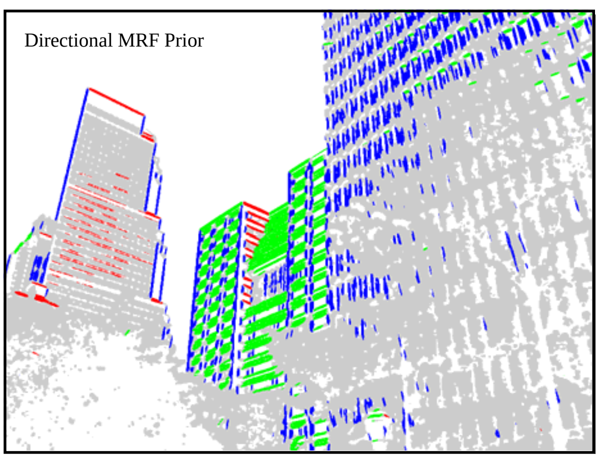
Original Image



Independent Prior



MRF Prior

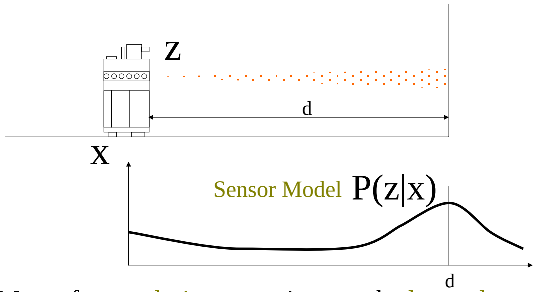


Directional MRF Prior

Take Home Points !

- **Bayesian paradigm** is a useful tool to
  - Represent knowledge
  - Perform inference
- **Sampling** is a nice way to implement the Bayesian paradigm, e.g. Condensation
- **Markov chain Monte Carlo** methods are a nice way to implement sampling

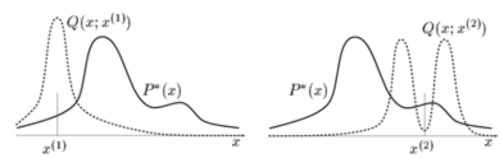
World Knowledge



Most often analytic expression, can be learned

Proposal Density Q

- Q(x';x) that depends on x



Step Size and #Samples

- Too large: all rejected
- Too small: random walk
- E[d]=e sqrt(T)
- Rule of thumb: $T>=(L/e)^2$

- Bummer: just a lower bound

Discussion Example

- e=1
- L=20
- T>=400
- Moral: avoid random walks

MCMC in high dimensions

- $e=s_{min}$
- $L=s_{max}$
- $T=(s_{max}/s_{min})^2$
- Good news: no curse in N
- bad news: quadratic dependence