

## ►Metropolis-Hastings (MH) Sampling

$C$  – configuration (of all variable values)

Here  $C = \mathbf{F}$  and  $p(C) = p(\mathbf{F} \mid X, Y, D)$

**Goal:** Generate a sequence of random samples  $\{C_i\}$  from  $p(C)$

- setup a Markov chain with a suitable transition probability function so that it generates the sequence

### Sampling procedure

1. given  $C_i$ , generate random sample  $S$  from  $q(S \mid C_i)$

$q$  may use some information from  $C_i$  (Hastings)

2. compute acceptance ratio

the evidence term drops out

$$a = \frac{p(S)}{p(C_i)} \cdot \frac{q(C_i \mid S)}{q(S \mid C_i)}$$

3. generate random number  $u$  from unit-interval uniform distribution  $U_{0,1}$
4. if  $u < a$  then  $C_{i+1} := S$  else  $C_{i+1} := C_i$

### ‘Programming’ an MH sampler

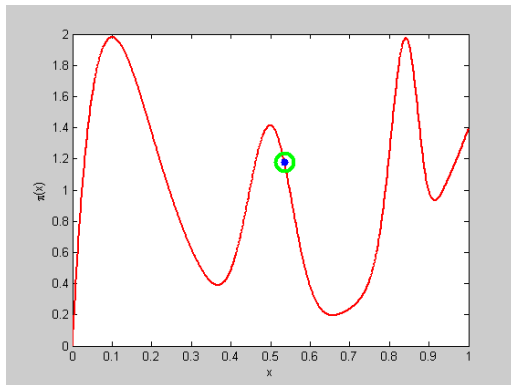
1. design a proposal distribution (mixture)  $q$  and a sampler from  $q$
2. write functions  $q(C_i \mid S)$  and  $q(S \mid C_i)$  that are proper distributions

not always simple

### Finding the mode

- remember the best sample fast implementation but must wait long to hit the mode
- use simulated annealing very slow
- start local optimization from the best sample good trade-off between speed and accuracy

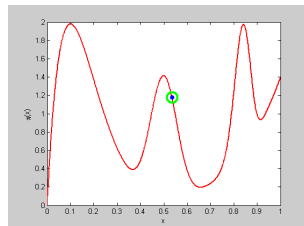
# MH Sampling Demo



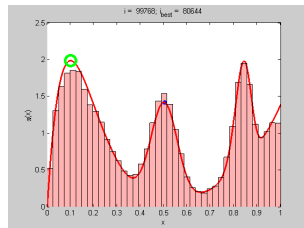
sampling process (video, 7:33, 100k samples)

- blue point: current sample
- green circle: best sample so far
- histogram: current distribution of visited states
- the vicinity of modes are the most often visited states

$$\text{quality} = \pi(x)$$



initial sample



final distribution of visited states

# Demo Source Code (Matlab)

```
function x = proposal_gen(x0)
% proposal generator q(x | x0)

    T = 0.01; % temperature
    x = betarnd((x0)/T+1, (1-x0)/T+1);
end

function p = proposal_q(x, x0)
% proposal distribution q(x | x0)

    T = 0.01;
    p = betapdf(x, x0/T+1, (1-x0)/T+1);
end

function p = target_p(x)
% target distribution p(x)

% shape parameters:
a = [2 40 100 6];
b = [10 40 20 1];

% mixing coefficients:
w = [1 0.4 0.253 0.50]; w = w/sum(w);
p = 0;
for i = 1:length(a)
    p = p + w(i)*betapdf(x,a(i),b(i));
end
end
```

```
% DEMO script

k = 10000; % number of samples
X = NaN(1,k); % list of samples

x0 = proposal_gen(0.5);
for i = 1:k
    x1 = proposal_gen(x0);
    a = target_p(x1)/target_p(x0) * ...
        proposal_q(x0,x1)/proposal_q(x1,x0);
    if rand < a
        X(i) = x1; x0 = x1;
    else
        X(i) = x0;
    end
end

figure(1)
x = 0:0.001:1;
plot(x, target_p(x), 'r', 'linewidth',2);
hold on
binw = 0.025; % histogram bin width
n = histc(X, 0:binw:1);
h = bar(0:binw:1, n/sum(n)/binw, 'histc');
set(h, 'facecolor', 'r', 'facealpha', 0.3)
xlim([0 1]); ylim([0 2.5])
xlabel 'x'
ylabel 'p(x)'
title 'MH demo'
hold off
```

## ► From MH Sampling to RANSAC

- configuration =  $k$ -tuple of inlier correspondences  
the minimization will be over a discrete set of epipolar geometries proposable from 7-tuples
- data-driven proposals  $q$ :
  1. select  $k$ -tuple from data independently and uniformly  $q(S) = \binom{mn}{k}^{-1}$
  2. solve the minimal geometric problem  $\mapsto$  geometry proposal (e.g.  $\mathbf{F}$  from  $k = 7$ )
- independent sampling  $a = \frac{p(S')}{p(S_i)} \cdot \frac{q(S_i)}{q(S')}$ 
  1.  $q$  uniform, then  $a = \frac{p(S')}{p(S_i)}$  MAPSAC ( $p(S)$  includes the prior)
  2.  $q$  dependent on descriptor similarity PROSAC (similar pairs are proposed more often)

---

### LO-MAPSAC

1. generate random sample  $S_b$  from  $q(S)$
2. set initial  $N := \binom{mn}{k}$
3. repeat  $N$ -times:
  - a. generate random sample  $S'$  from  $q(S)$
  - b. if  $p(S') > p(S_b)$  then
    - i.  $S_b := S'$
    - ii. threshold-out inliers using  $e_T$  from (23)
    - iii. start local optimization from  $S_b$  and update  $S_b$  with the result
    - iv. re-estimate  $N$  from inlier counts using the standard formula for RANSAC termination, see Slide 117
4. output  $S_b$

- see the  MPV course for RANSAC details

see also [Fischler & Bolles 1981], [25 years of RANSAC]

## ► Stopping RANSAC

**Principle:** what is the number of proposals  $N$  that are needed to hit an all-inlier sample?

$$N \geq \frac{\log(1 - P)}{\log(1 - (1 - w)^s)}$$

- $(1 - w)^s$  – proposal does not contain an outlier
- $1 - (1 - w)^s$  – proposal contains at least one outlier
- $1 - P =$  all proposals contained an outlier  $= (1 - (1 - w)^s)^N$

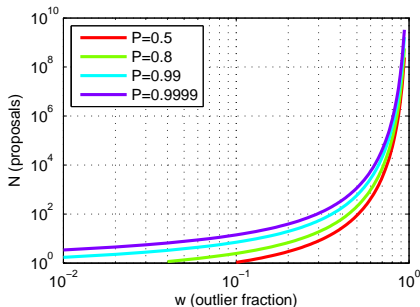
$P$  – probability that at least one sample is all-inlier

$w$  – the fraction of outliers among tentative correspondences

$s$  – sample size (7 in 7-point algorithm)

$N$  for  $s = 7$

$w$	$P$	
	0.8	0.99
0.5	205	590
0.8	$1.3 \cdot 10^5$	$3.5 \cdot 10^5$
0.9	$1.6 \cdot 10^7$	$4.6 \cdot 10^7$

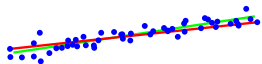


- $N$  can be re-estimated using the current estimate for  $w$  (if there is LO, then after LO)  
the quasi-posterior estimate for  $w$  is the average over all samples generated so far
- for  $w \rightarrow 1$  we gain nothing over the standard MH-sampler stopping criterion

# ►The Difference between RANSAC and a General MH Sampler

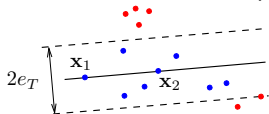
**RANSAC = five ideas:** [Fischler & Bolles 1981]

1. proposal distribution is given by the empirical distribution of data sample:



- pairs of points define line distribution from  $p(\mathbf{n} \mid X)$  (left)
- random correspondence tuples drawn uniformly propose samples of  $\mathbf{F}$  from a data-driven distribution  $q(\mathbf{F} \mid X, Y)$

2. stopping based on the probability of mode-hitting → Slide 117
3. standard RANSAC replaces probability maximization with consensus maximization

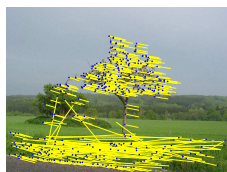


the  $e_T$  is the inlier/outlier threshold from (23)

4. when counting inliers, do not work with all  $m_{ij}$  but with a set of tentative correspondences that form a matching, e.g. selected by stable matching:
  - a. find a pair  $m_{ij}$  of greatest  $p_1(d_{ij})$  and remember it
  - b. remove row  $i$  and column  $j$  from the matching table (needs some bookkeeping and reindexing)
  - c. repeat Steps a–c until the table is empty
  - d. return the remembered set
5. each time a new best sample occurs, start local optimization from inliers

or LO weighted by posterior  $p(m_{ij})$  [Chum et al. 2003]  
LM optimization with Sampson error (and re-weighting)

# Example Matching Results for the 7-point Algorithm with RANSAC



input images

interest points (ca. 3600) tentative corresp. (416)

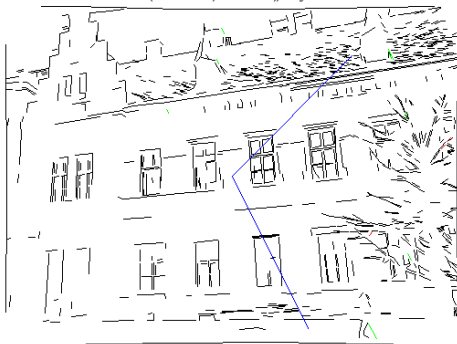
matching (340)  
notice wrong matches

- the minimization is over a discrete set of epipolar geometries proposable from 7-tuples

# Example: MH Sampling for a More Complex Problem

**Task:** Find two vanishing points from line segments detected in input image.

iter: 10 (acc TOT=0.0%, HMC=NaN%); Eavg = 14.597

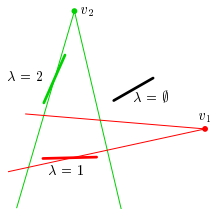


video

## Model

- principal point known, square pixel
- explicit variables
  1. two unknown vanishing points  $v_1, v_2$
- latent variables
  1. each line has a vanishing point label  $\lambda_i \in \{\emptyset, 1, 2\}$ ,  $\emptyset$  represents an outlier
  2. 'mother lines' passing through vanishing points

$$\arg \min_{v_1, v_2, \Lambda, L} V(v_1, v_2, \Lambda, L \mid S)$$



## simplifications

- vanishing points restricted to the set of all pairwise segment intersections
- mother lines fixed by segment centroid



# Beyond RANSAC

Note that by simplification in (18) on Slide 106 we have lost constraints on  $M$  (eg. uniqueness). One can choose a better model when not marginalizing:

$$p(M, \mathbf{F}, X, Y, D) = \underbrace{p(X, Y \mid M, \mathbf{F})}_{\text{geometric error}} \cdot \underbrace{p(D \mid M)}_{\text{similarity}} \cdot \underbrace{p(M)}_{\text{constraints}} \cdot \underbrace{p(\mathbf{F})}_{\text{prior}}$$

this is a global model: decisions on  $m_{ij}$  are no longer independent!

## In the MH scheme

- one can work with full  $p(M, \mathbf{F} \mid X, Y, D)$ , then  $S = (M, \mathbf{F})$ 
  - explicit labeling  $m_{ij}$  can be done by, e.g. sampling from

$$q(m_{ij} \mid \mathbf{F}) \sim ((1 - \alpha_0) p_1(e_{ij} \mid \mathbf{F}), \alpha_0 p_0(e_{ij} \mid \mathbf{F}))$$

when  $p(M)$  uniform then always accepted,  $a = 1$

⊗ derive

- additional proposals from  $q(\mathbf{F} \mid M)$  are possible, with explicit inliers Hybrid Monte Carlo
- we can compute the posterior probability of each match  $p(m_{ij})$  by histogramming  $m_{ij}$  over  $\{S_i\}$
- local optimization can then use explicit inliers and  $p(m_{ij})$
- error can be estimated for elements of  $\mathbf{F}$  from  $\{S_i\}$  does not work in RANSAC!
- large error indicates problem degeneracy this is not directly available in RANSAC
- good conditioning is not a requirement we work with the entire distribution  $p(\mathbf{F})$
- one can find the most probable number of epipolar geometries by reversible jump MCMC (homographies or other models)

if there are multiple models explaining data, RANSAC will return one of them randomly

Thank You





