# ▶ The Full Problem of Matching and Fundamental Matrix Estimation

Problem: Given two sets of image points $X = \{x_i\}_{i=1}^m$ and $Y = \{y_j\}_{j=1}^n$ and their descriptors D, find the most probable

1. inliers $S_X \subseteq X$, $S_Y \subseteq Y$
2. one-to-one perfect matching $M : S_X \to S_Y$ perfect matching: 1-factor of the bipartite graph
3. fundamental matrix $\mathbf{F}$ such that $\operatorname{rank} \mathbf{F} = 2$
4. such that for each $x_i \in S_X$ and $y_j = M(x_i)$ it is probable that
   a. the image descriptor $D(x_i)$ is similar to $D(y_j)$, and
   b. the total geometric error $\sum_{ij} e_{ij}^2(\mathbf{F})$ is small   note a slight change in notation: e_{ij}
5. inlier-outlier and outlier-outlier matches are improbable
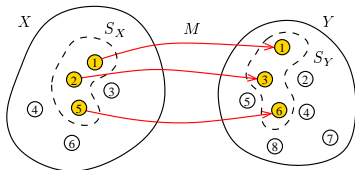
$$(M^*, \mathbf{F}^*) = \arg\max_{M, \mathbf{F}} p(M, \mathbf{F} \mid X, Y, D) \tag{17}$$

- probabilistic model: an efficient language for task formulation
- the (17) is a p.d.f. for all the involved variables   (there is a constant number of variables!)
- binary matching table $M_{ij} \in \{0, 1\}$ of fixed size $m \times n$
  - each row/column contains at most one unity
  - zero rows/columns correspond to unmatched point x_i/y_j

For algorithmic efficiency, instead of $(M^*, \mathbf{F}^*) = \arg\max\limits_{M, \mathbf{F}} p(M, \mathbf{F} \mid X, Y, D)$ we will solve

$$\mathbf{F}^* = \arg\max\limits_{\mathbf{F}} p(\mathbf{F} \mid X, Y, D) \qquad (18)$$

by marginalization of $p(M, \mathbf{F} \mid X, Y, D)$ over M this simplification changes the problem!

$$p(M, \mathbf{F} \mid X, Y, D) \simeq p(M, \mathbf{F}, X, Y, D) = p(X, Y, D, M \mid \mathbf{F}) \cdot p(\mathbf{F})$$

assuming correspondence-wise independence:

$$p(X, Y, D, M \mid \mathbf{F}) = \prod_{i=1}^{m} \prod_{j=1}^{n} p(x_i, y_j, D, m_{ij} \mid \mathbf{F}) \stackrel{\text{def}}{=} \prod_{i=1}^{m} \prod_{j=1}^{n} p_e(e_{ij}, d_{ij}, m_{ij} \mid \mathbf{F})$$

- e_{ij} represents geometric error for match $x_i \leftrightarrow y_i$: $e_{ij}(x_i, y_i \mid \mathbf{F})$
- d_{ij} represents descriptor similarity for match $x_i \leftrightarrow y_i$: $d_{ij} = \|\mathbf{d}(x_i) - \mathbf{d}(y_j)\|$

<u>Marginalization:</u>

$$\sum_{m_{11} \in \{0,1\}} \sum_{m_{12}} \cdots \sum_{m_{mn}} p(X, Y, D, M \mid \mathbf{F}) = \sum_{m_{11}} \sum_{m_{12}} \cdots \sum_{m_{mn}} \prod_{i=1}^{m} \prod_{j=1}^{n} p_e(e_{ij}, d_{ij}, m_{ij} \mid \mathbf{F}) =$$

$$= \cdots = \prod_{i=1}^{m} \prod_{j=1}^{n} \underbrace{\sum_{m_{ij} \in \{0,1\}} p_e(e_{ij}, d_{ij}, m_{ij} \mid \mathbf{F})}_{\text{we will continue with this term}} = p(X, Y, D \mid \mathbf{F})$$

Robust Matching Model (cont'd)

$$\sum_{m_{ij} \in \{0,1\}} p_e(e_{ij}, d_{ij}, m_{ij} \mid \mathbf{F}) = \sum_{m_{ij} \in \{0,1\}} p_e(e_{ij}, d_{ij} \mid m_{ij}, \mathbf{F}) \cdot p(m_{ij} \mid \mathbf{F}) =$$

$$= \underbrace{p_e(e_{ij}, d_{ij} \mid m_{ij} = 1, \mathbf{F})}_{p_1(e_{ij}, d_{ij} \mid \mathbf{F})} \cdot \underbrace{p(m_{ij} = 1 \mid \mathbf{F})}_{1 - \alpha_0} + \underbrace{p_e(e_{ij}, d_{ij} \mid m_{ij} = 0, \mathbf{F})}_{p_0(e_{ij}, d_{ij} \mid \mathbf{F})} \cdot \underbrace{p(m_{ij} = 0 \mid \mathbf{F})}_{\alpha_0} =$$

$$= (1 - \alpha_0)\, p_1(e_{ij}, d_{ij} \mid \mathbf{F}) + \alpha_0\, p_0(e_{ij}, d_{ij} \mid \mathbf{F}) \quad (19)$$

- the $p_0(e_{ij}, d_{ij} \mid \mathbf{F}) \approx \mathrm{const}$ is a penalty for 'missing a correspondence' but it should be a p.d.f. (cannot be a constant) (see Slide 108 for a simplification)

$$\alpha_0 \to 1, \quad p_0 \to 0 \quad \text{so that} \quad \frac{\alpha_0}{1 - \alpha_0}\, p_0 \approx \mathrm{const}$$

- the $p_1(e_{ij}, d_{ij} \mid \mathbf{F})$ is typically an easy-to-design component: assuming independence of geometric error and descriptor similarity:

$$p_1(e_{ij}, d_{ij} \mid \mathbf{F}) = p_1(e_{ij} \mid \mathbf{F}) \cdot p_1(d_{ij})$$

- we choose, eg.

$$p_1(e_{ij} \mid \mathbf{F}) = \frac{1}{T_e(\sigma_1, \mathbf{F})}\, e^{-\frac{e_{ij}^2(\mathbf{F})}{2\sigma_1^2}}, \quad p_1(d_{ij}) = \frac{1}{T_d(\sigma_d, \dim \mathbf{d})}\, e^{-\frac{\|\mathbf{d}(x_i) - \mathbf{d}(y_j)\|^2}{2\sigma_d^2}} \quad (20)$$

- $\sigma_1, \sigma_d, \alpha_0$ are 'hyper-parameters'
- the form of $T(\sigma_1, \mathbf{F})$ depends on error definition
- we will continue with the result from (19)

▶Simplified Robust Energy (Error) Function

- assuming the choice of p_1 as in (20), we are simplifying

$$p(X, Y, D \mid \mathbf{F}) = \prod_{i=1}^{m} \prod_{j=1}^{n} \Big[(1 - \alpha_0)\, p_1(e_{ij}, d_{ij} \mid \mathbf{F}) + \alpha_0\, p_0(e_{ij}, d_{ij} \mid \mathbf{F}) \Big] \qquad (21)$$

- we define 'energy' as: $V(x) = -\log p(x)$ this helps simplify the formulas
- for simplicity, we omit d_{ij}
- we choose $\sigma_0 \gg \sigma_1$ and the missed-correspondence penalty function as

$$p_0(e_{ij} \mid \mathbf{F}) = \frac{1}{T_e(\sigma_0, \mathbf{F})}\, e^{-\frac{e_{ij}^2(\mathbf{F})}{2\sigma_0^2}}$$

- then

$$V(X, Y, D \mid \mathbf{F}) = \sum_{i=1}^{m} \sum_{j=1}^{n} \left[\underbrace{-\log \frac{1-\alpha_0}{T_e(\sigma_1, \mathbf{F})}}_{\Delta(\mathbf{F})} - \log\Big(e^{-\frac{e_{ij}^2(\mathbf{F})}{2\sigma_1^2}} + \underbrace{\frac{\alpha_0}{1-\alpha_0}\, \frac{T_e(\sigma_1, \mathbf{F})}{T_e(\sigma_0, \mathbf{F})}\, e^{-\frac{e_{ij}^2(\mathbf{F})}{2\sigma_0^2}}}_{t\, \approx\, \text{const}} \Big) \right]$$
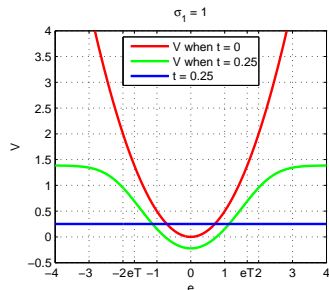
- by choosing representative of $\mathbf{F}$ such that $\Delta(\mathbf{F}) = \text{const}$, we get

$$V(X, Y, D \mid \mathbf{F}) = m\, n\, \Delta + \sum_{i=1}^{m} \sum_{j=1}^{n} \underbrace{-\log\Big(e^{-\frac{e_{ij}^2(\mathbf{F})}{2\sigma_1^2}} + t \Big)}_{\hat{V}(e_{ij})} \qquad (22)$$

note that m, n are fixed

Example for $\hat{V}(e)$ from (22):

red – the usual (non-robust) error $\qquad$ when $t = 0$

blue – the rejected correspondence penalty t

green – 'robust energy' (22)

- if the error of a correspondence exceeds a limit, it is ignored
- then $\hat{V}(e) = \mathrm{const}$ and we essentially count outliers in (22)
- t controls the 'turn-off' point
- the inlier/outlier threshold is e_T is the error for which
  $(1 - \alpha_0)\, p_1(e_T) = \alpha_0\, p_0(e_T)$: $\qquad$ note that $t \approx 0$

$$e_T = \sigma_1 \sqrt{-\log t^2} \qquad (23)$$

The full optimization problem is (18):

$$\mathbf{F}^* = \arg\max_{\mathbf{F}} p(\mathbf{F} \mid X, Y, D) = \arg\max_{\mathbf{F}} \frac{\overbrace{p(X, Y, D \mid \mathbf{F})}^{\text{likelihood}} \cdot \overbrace{p(\mathbf{F})}^{\text{prior}}}{\underbrace{p(X, Y, D)}_{\text{evidence}}} =$$

$$= \arg\min_{\mathbf{F}} \big\{ V(X, Y, D \mid \mathbf{F}) + V(\mathbf{F}) \big\}$$

- typically we take $V(\mathbf{F}) = 0$ unless we need to stabilize a computation, e.g. when video camera moves smoothly (on a high-mass vehicle) and we have a prediction for $\mathbf{F}$
- evidence is not needed unless we want to compare different models
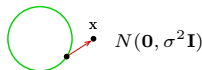
Discussion: On The Art of Probabilistic Model Design. . .

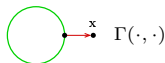- a few models for fitting zero-centered circle C of radius r to points in $\mathbb{R}^2$

The table structure shows three models across columns with row labels on the left.

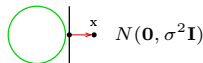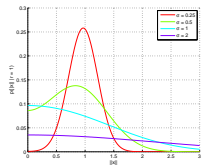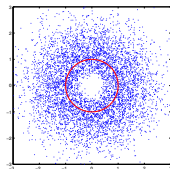|	marginalized over C	orthogonal deviation from C	Sampson approximation
error model	$N(\mathbf{0}, \sigma^2\mathbf{I})$	$\Gamma(\cdot, \cdot)$	$N(\mathbf{0}, \sigma^2\mathbf{I})$
radial p.d.f.			
random sample			
$p(\mathbf{x} \mid r)$	$\approx \dfrac{1}{\sigma\sqrt{(2\pi)^3 r \, \|\mathbf{x}\|}}\, e^{-\frac{(\|\mathbf{x}\|-r)^2}{2\sigma^2}}$	$\dfrac{1}{2\pi\Gamma(\frac{r^2}{\sigma})}\dfrac{1}{\|\mathbf{x}\|^2}\left(\dfrac{r\|\mathbf{x}\|}{\sigma}\right)^{\frac{r^2}{\sigma}} e^{-\frac{r\|\mathbf{x}\|}{\sigma}}$	$\dfrac{1}{r\sigma\sqrt{(2\pi)^3}}\, e^{-\frac{e^2(\mathbf{x};r)}{2\sigma^2}}$
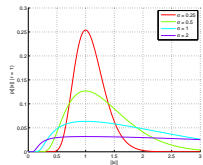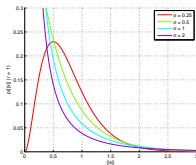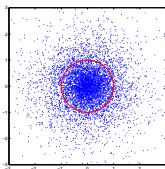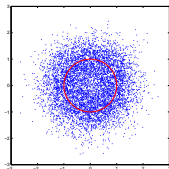
- mode inside the circle
- models the inside well
- tends to normal distrib.

- peak at the center
- unusable for small radii
- tends to Dirac distrib.

- mode at the circle
- hole at the center
- tends to normal distrib.

How To Find the Global Maxima (Modes) of a PDF?
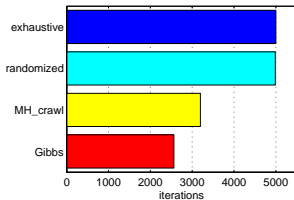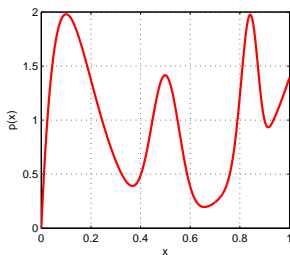
- consider the function $p(x)$ at left p.d.f. on $[0, 1]$, mode at 0.1
- consider several methods:
  1. exhaustive search
     ```
     step = 1/(iterations-1);
     for x = 0:step:1
      if p(x) > bestp
       bestx = x; bestp = p(x);
      end
     end
     ```
     - slow algorithm (definite quantization); faster variants exist • fast to implement
  2. randomized search with uniform sampling
     ```
     x = rand(1);
     if p(x) > bestp
      bestx = x; bestp = p(x);
      end
     ```
     - slow algorithm but better convergence • fast to implement • how to stop it?
  3. random sampling from $p(x)$ (Gibbs sampler)
   - faster algorithm • fast to implement but often infeasible (e.g. when $p(x)$ is data dependent (our case))
  4. Metropolis-Hastings sampling
     - almost as fast (with care) • not so fast to implement • rarely infeasible • RANSAC belongs here

- averaged over 10^4 trials
- number of proposals before $|x - x_{\text{true}}| \leq$ step
- uniform and Gibbs give the theoretical result

How To Generate Random Samples from a Complex Distribution?

target (red) and scaled proposal (blue) distributions
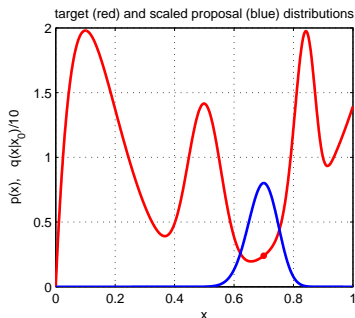
- red: probability density function $p(x)$ of a toy distribution on the unit interval   *target distribution*

$$p(x) = \sum_{i=1}^{4} \alpha_i \, \mathrm{Be}(x; \alpha_i, \beta_i), \ \sum_{i=1}^{4} \alpha_i = 1, \ \alpha_i \geq 0$$

$$\mathrm{Be}(x; \alpha, \beta) = \frac{1}{\mathrm{B}(\alpha, \beta)} \cdot x^{\alpha-1}(1-x)^{\beta-1}$$

- note we can generate samples from this $p(x)$ *how?*

- suppose we cannot sample from $p(x)$ but we can sample from some 'simple' distribution, given the last sample x_0 (blue)   *proposal distribution*

$$q(x \mid x_0) = \begin{cases} \mathrm{U}_{0,1}(x) & \text{(independent) uniform sampling} \\ \mathrm{Be}(x; \frac{x_0}{T}+1, \frac{1-x_0}{T}+1) & \text{'beta' diffusion (crawler)} \quad T-\text{temperature} \\ p(x) & \text{(independent) Gibbs sampler} \end{cases}$$

- note we have unified all the random sampling methods on the previous slide
- how to transform proposal samples $q(x \mid x_0)$ to target distribution $p(x)$ samples?

▶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 $C_{i+1} := C_i$

'Programing' an MH sampler

1. design a proposal distribution 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
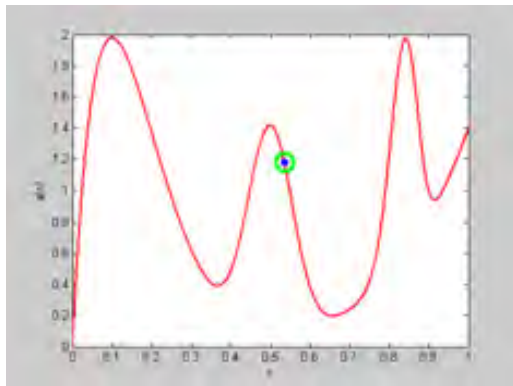
sampling process (video, 7:33)

initial sample

final distribution of visited states

- blue point: current sample
- green circle: best sample so far   quality $= \pi(x)$
- histogram: current distribution of visited states
- the vicinity of modes are the most often visited states
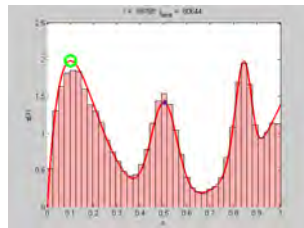
Demo Source Code (Matlab)

```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 = target_p(x)
% target distribution pi(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


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
```

```matlab
%% 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
n = histc(X, 0:0.025:1);
h = bar(0:0.025:1, n/sum(n)/0.025, '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 <u>inlier correspondences</u>

  the minimization will be over a <u>discrete</u> 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 $\quad a = \dfrac{p(S')}{p(S_i)} \cdot \dfrac{q(S_i)}{q(S')}$
  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]

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
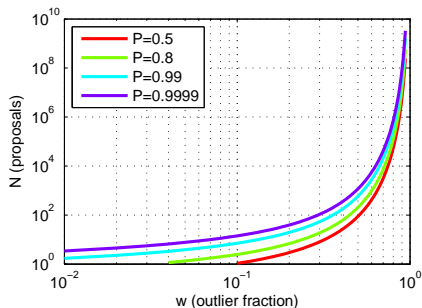- $1 - P =$ all proposals contained an outlier $= (1 - (1 - w)^s)^N$

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 stoppig criterion

RANSAC = five ideas: [Fischler & Bolles 1981]

1. proposal distribution is given by the <u>empirical distribution</u> of data sample:
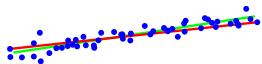
- • pairs of points define line distribution from $p(\mathbf{n} \mid X)$ (left)
- • random correspondence tuples drawn uniformly propose samples of $\mathbf{F}$ from a data-driven distribution $q(\mathbf{F} \mid X, Y)$

2. stopping based on the probability of mode-hitting   → Slide 117

3. standard RANSAC replaces probability maximization with <u>consensus maximization</u>
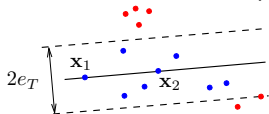
the e_T is the inlier/outlier threshold from (23)

4. when counting inliers, do not work with all m_{ij} but with a set of <u>tentative correspondences</u> 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–b until the table is empty
   d. return the remembered set

5. each time a new best sample occurs, start <u>local optimization</u> 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
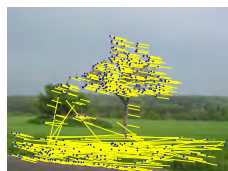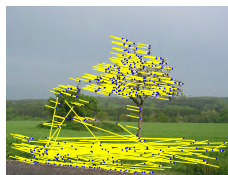
input images   interest points (ca. 3600)   tentative coresp. (416)   matching (340) notice wrong matches

- the minimization os 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.

Model

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

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

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

    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 explaning data, RANSAC will return one of them randomly

Thank You