Computing Multiple Solutions

Computing multiple distinct solution is of great importance for preserving the ambiguities and uncertainty. We introduce two algorithms

1, C4: Clustering with Cooperative and Competitive Constraints.

2, The K-adventure's algorithm in DDMCMC.

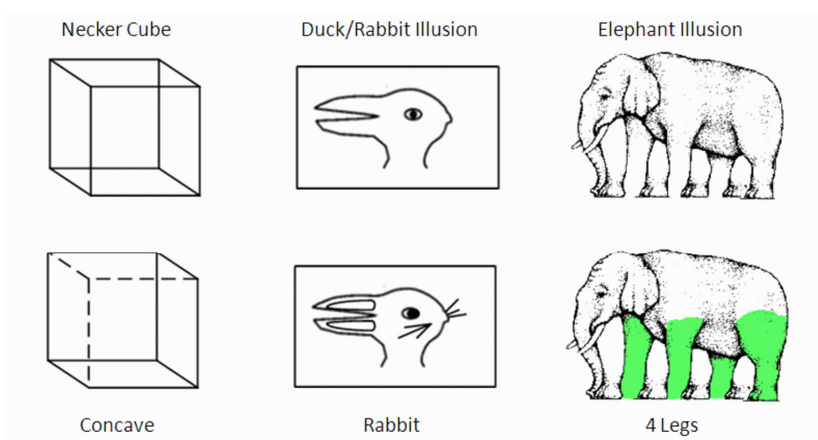

Jake Porway

Objective: Computing Multiple Solutions

Two criteria for designing "effective algorithms":

1, Fast convergence to global optimal solutions: "**not greedy**"

2, Exploring diverse solutions (liquidate, fast mixing) : "**not stubborn**"

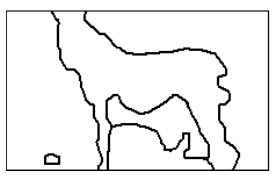


Computing multiple solutions to avoid premature commitments.

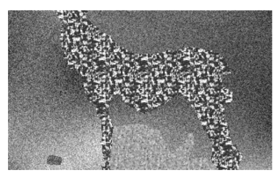
Sometime, to design ambiguities is to create art.

Computing Ambiguity in Visual Inference

a. Input image



b. Segmented texture regions



c. synthesis by texture models

d. curve processes + bkgd region

e. synthesis by curve models

Ambiguities are ubiquitous in images

Local interpretations are often strongly coupled !

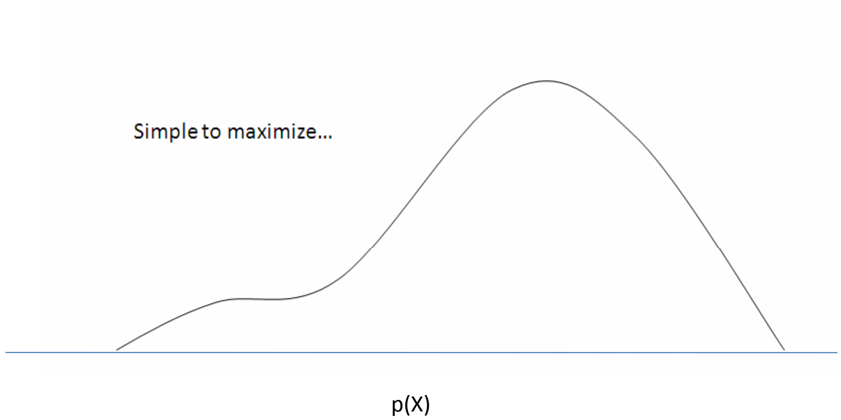
They form “clusters” and the search algorithms often get stuck.

Sampling Probabilities with Multiple Modes

The two criteria for MCMC design:

1, Short “burn-in” period --- The MC reaches the equilibrium fast

2, Fast “mixing rate” --- The MC states are less correlated in time



Revisit: Swendsen-Wang 1987

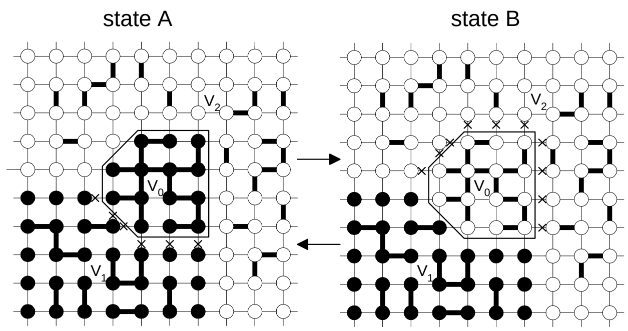
Ising model: $p(X)=\frac{1}{Z}\prod_{<i,j>}\varphi(x_i,x_j);\quad \varphi(x_i,x_j)=e^{\beta\cdot\delta(x_i=x_j)};\quad \delta()\in\{-1,+1\}$

(a) Augment with auxiliary bonding variable U along edges E:

$$U=\{u_{ij},\, i,\, j\in X\,;\, u_{ij}\in\{-1,+1\}\}$$

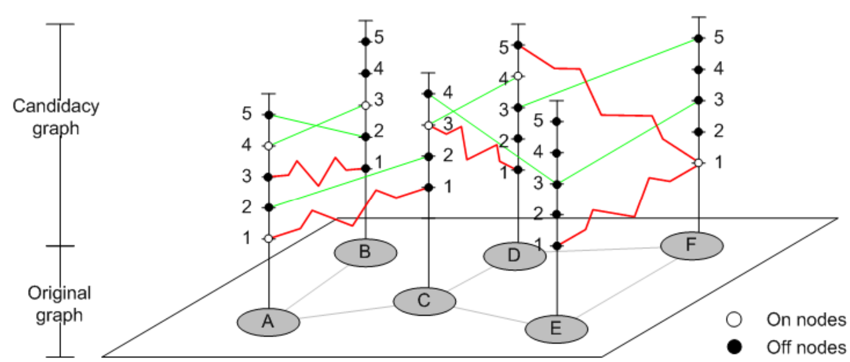
(b) Turn edges "on" (u_{ij} = +1) or "off" (u_{ij} = -1) probabilistically.

(c) Select a connected component V_0 and update its nodes' labels.

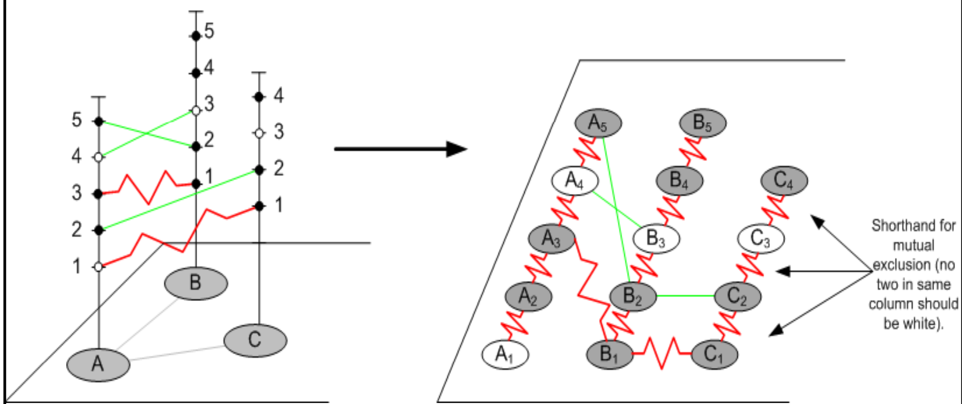


Representation: Candidacy Graphs

We formulate a candidacy graph representation, as it can represents
1, **MRF** and **CRF** structures
2, **Soft** and **hard** constraints.
3, **Positive** (collaborative) and **negative** (competitive) edges.

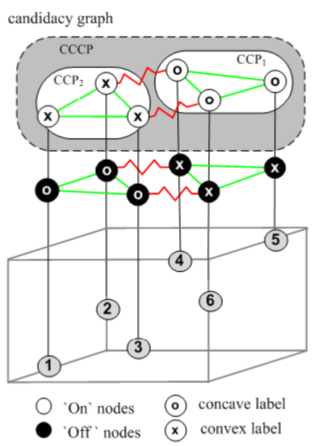


Flattening the Candidacy Graphs



This becomes a graph coloring problem: each color is a solution.

Ex: Candidacy Graph for Necker Cube



Some labels/edges omitted for clarity

Potts Model with Negative Edges

Split our edge set E into E^+ and E^- to represent positive and negative interactions:

$$p(X) = \frac{1}{Z} \prod_{<i,j> \in E^+} \varphi^+(x_i, x_j) \prod_{<i,j> \in E^-} \varphi^-(x_i, x_j)$$

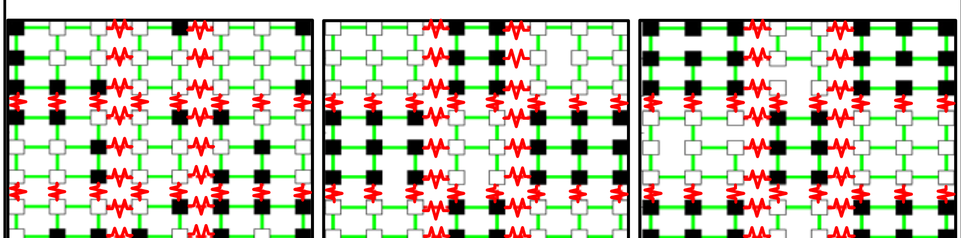
$$\varphi^+(x_i, x_j) = e^{\beta \cdot \delta(x_i = x_j)}, \ \varphi^-(x_i, x_j) = e^{\beta \cdot \delta(x_i \neq x_j)}$$

Form components similarly to Swendsen-Wang.

Components now consist of positively connected sub-components connected by negative interactions.

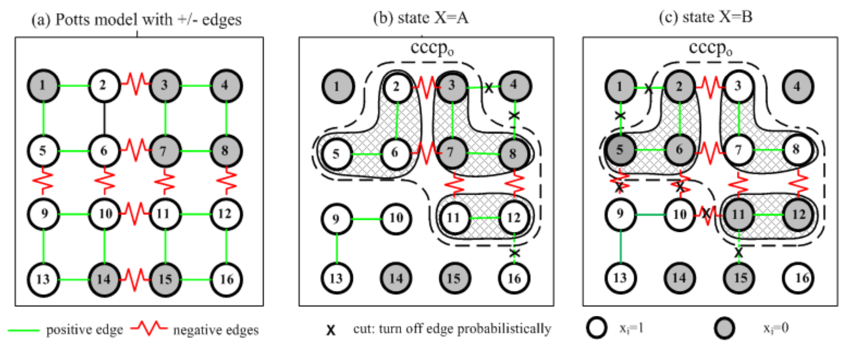
Experiments on Negative Edge Ising Model

Created "checkerboard" constraint problem.



Initial State | Solution 1 | Solution 2

CCCP: Composite Connected Components



Edge probabilities for clustering cccp's

For positive edge, the edge variable follows a Bernoulli probability

$u_e \sim \text{Bernoulli}(q_e 1(x_s = x_t))$

For negative edge, it follows

$u_e \sim \text{Bernoulli}(q_e 1(x_s \neq x_t))$

$$p^+(u_{ij} \mid x_i = x_j) = \begin{cases} 1-\rho & u_{ij} = 0 \\ \rho & u_{ij} = 1 \end{cases}$$

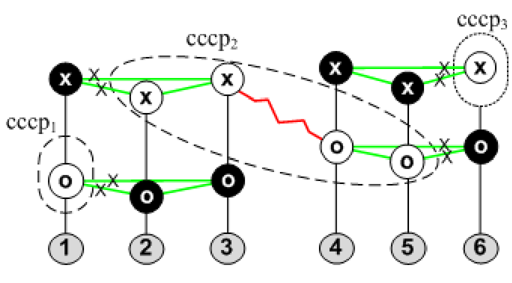
$$p^+(u_{ij} \mid x_i \neq x_j) = \begin{cases} 1 & u_{ij} = 0 \\ 0 & u_{ij} = 1 \end{cases}$$

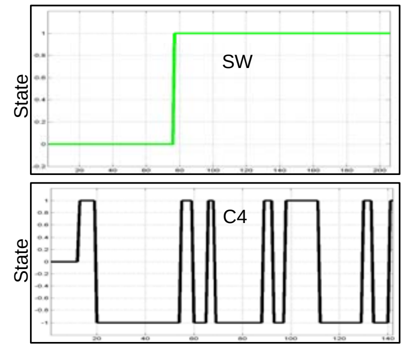
$$p^-(u_{ij} \mid x_i \neq x_j) = \begin{cases} 1-\rho & u_{ij} = 0 \\ \rho & u_{ij} = 1 \end{cases}$$

$$p^-(u_{ij} \mid x_i = x_j) = \begin{cases} 1 & u_{ij} = 0 \\ 0 & u_{ij} = 1 \end{cases}$$

Mixing rate

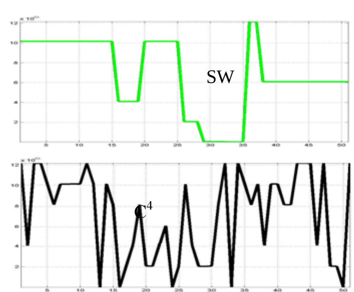
solution states over time for Ising model (two states)

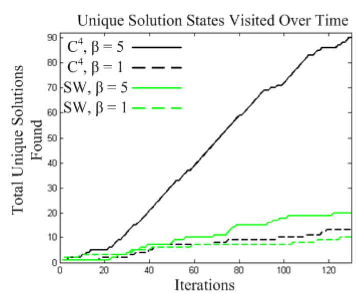




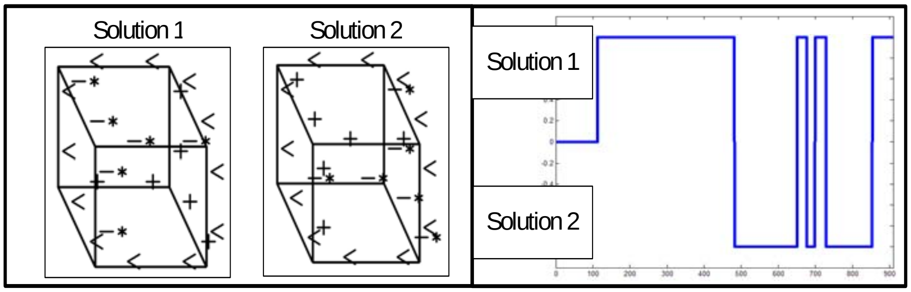
C^4 has
- Low burn-in time (converges quickly).
- High mixing rate (samples remaing unbiased over short run).

Simulating the Potts model



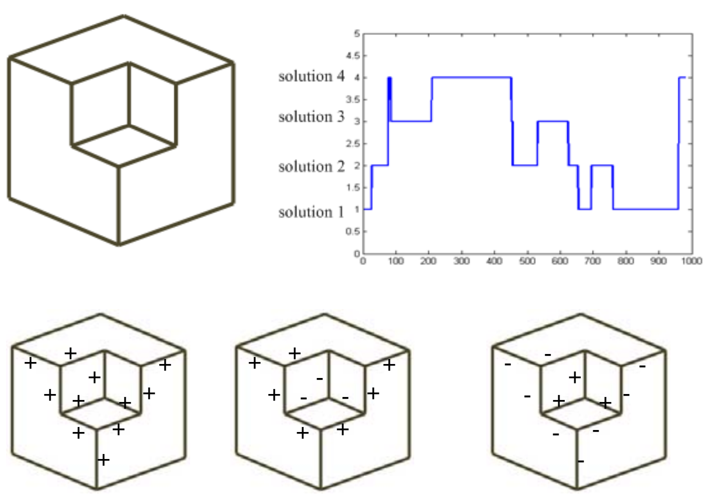


Application I: Line Drawing



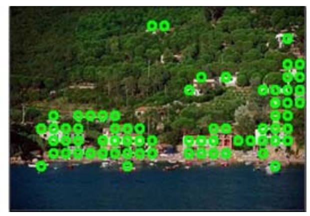
C4 finds both solutions, swaps corner labels continuously.

Line drawing interpretation



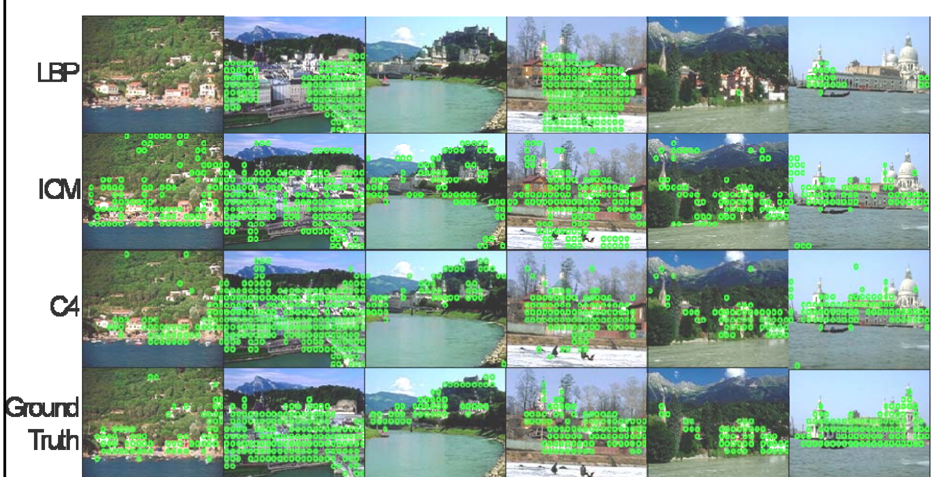
Application II: CRF for Building Detection

- Learn a CRF for building detection[1].
- CRF labels each window of image as {0,1} .



[1]S. Kumar, M. Hebert, "Man-Made Structure Detection in Natural Images using a Causal Multiscale Random Field", *CVPR, 2003.*

Comparison: CRF for Building Detection



LBP = Loopy Belief Propagation ICM = Iterated Conditional Modes

Comparison: CRF Building Detection

Algorithm	FP (per image)	Detection Rate (%)
Loopy BP	23.18	0.451
Swendsen-Wang	156.05	0.468
C4	**47.12**	0.696
ICM	61.78	**0.697**

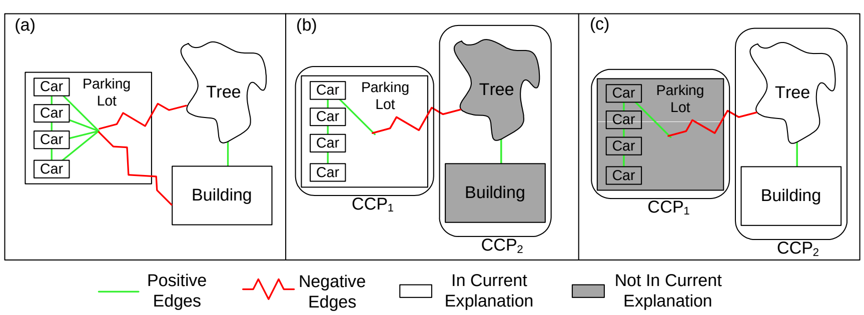
Note: Couldn't recreate Kumar's exact results, so compared our CRF with his inference method (ICM) and ours (C4).

Application III: Aerial Image Segmentation

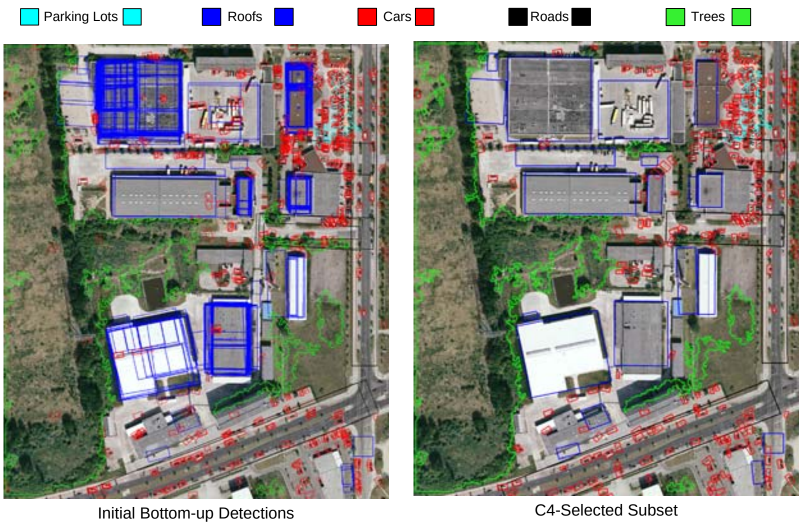
-Detect objects in an aerial image.

- MANY false positives.

- Have C^4 determine which ones best explain the image by turning detected objects "on" and "off".

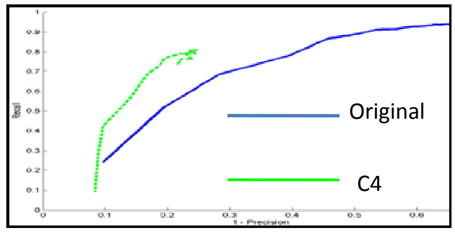


Results: Aerial Image Segmentation

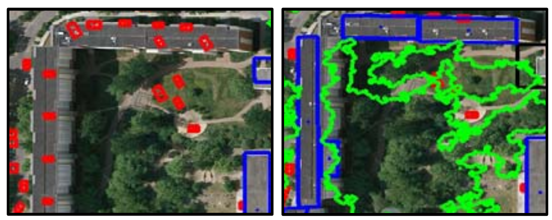


Results: Aerial Image Segmentation

P-R curve for original detections vs. C4 pruned detections.

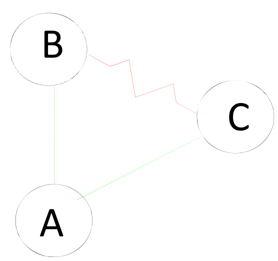


Example of C4's benefits: Misinterpreted section of image is later updated by large composite move.

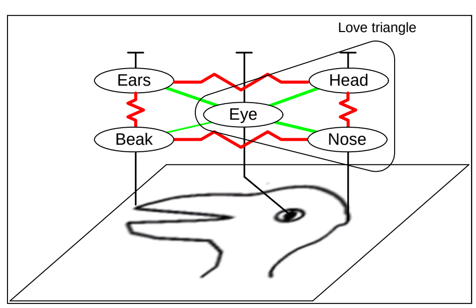


Problem with "Flat"-- C^4

The "love triangles":
create inconsistent clusters

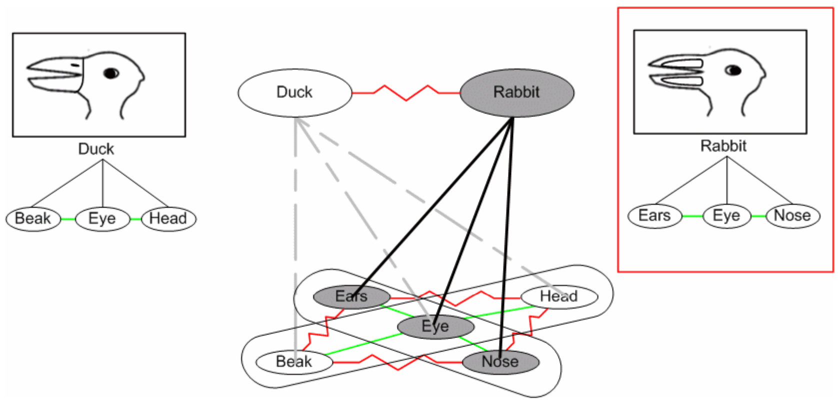


e.g. Love triangles in the duck/rabbit illusion.



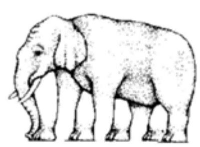
Hierarchical C4

The candidacy graph so far represent pairwise edges, high-order relations are represented by extended candidacy graphs

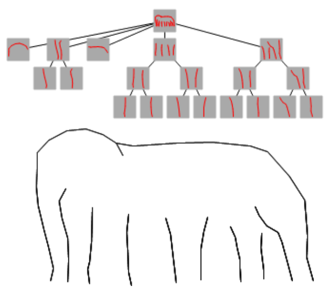


System will now flip between duck and rabbit without love triangle issue.

Elephant Illusion



Hierarchical part model



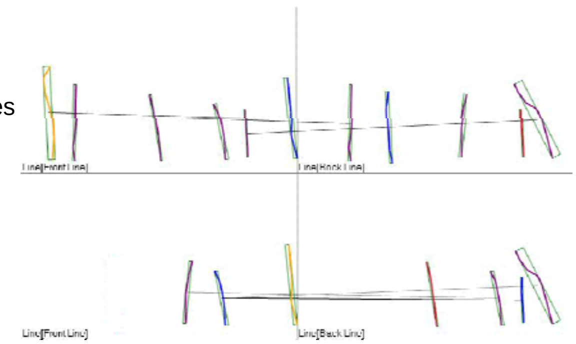
Layered representation of hierarchy

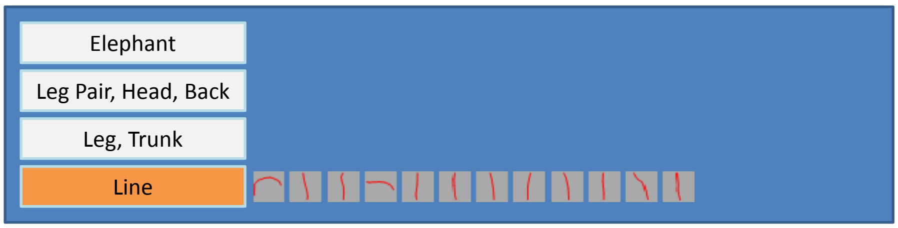
Elephant
Leg Pair, Head, Back
Leg, Trunk
Line

Composing the Bottom Layer

Trunk compatibilities

Leg compatibilities

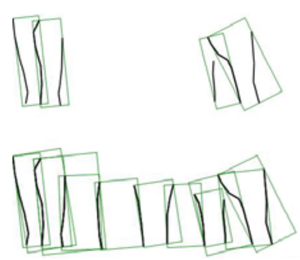


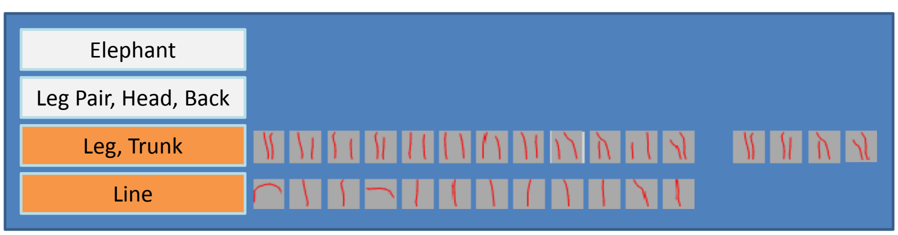


Part Binding for Next Layer

Trunk bindings

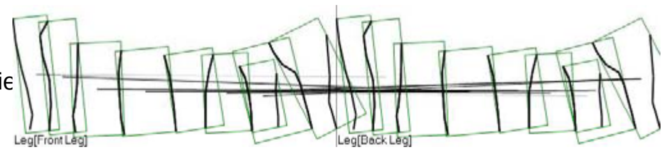
Leg bindings



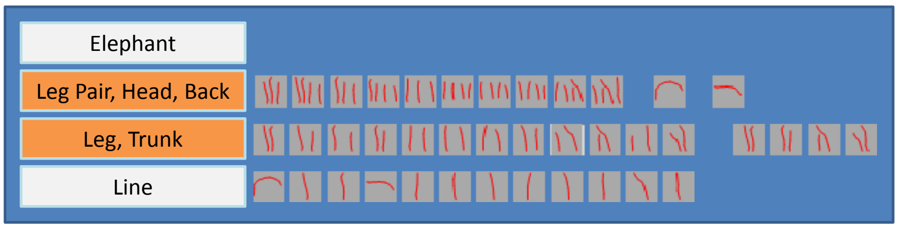


Continue Sampling/Binding for each layer

Leg Pair compatibilities

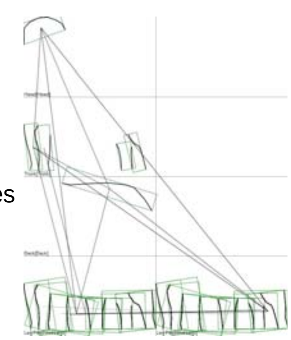


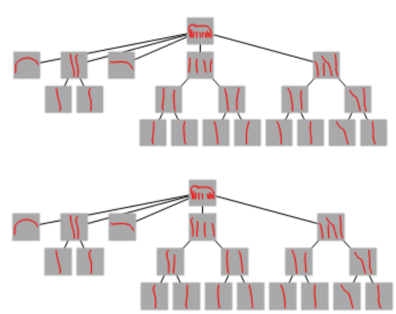
Leg Pair bindings

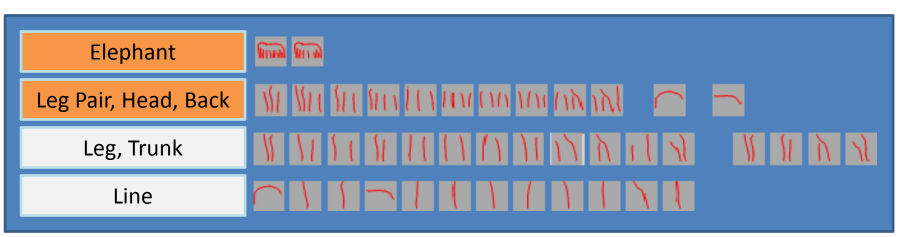


Top Level Bindings and sampling

Elephant compatibilities







Summary

1, A **candidacy graph representation** incorporates

MRF and **CRF** structures
Soft and **hard** constraints.
Positive (collaborative) and **negative** (competitive) edges.

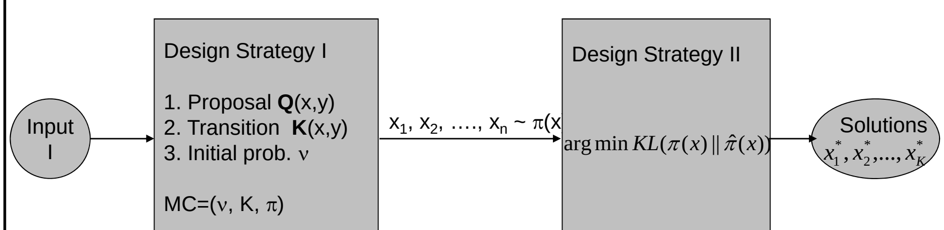
hierarchic compositional structures for high-order relations.

2, An **algorithm** with large composite moves

Fast convergence to global solutions
Effective mixing between solutions

Relaxation labeling → Gibbs sampling → SW Cut → C^4
(Rosenfeld/Zucker/Hammel 1978) (Geman/ Geman 1984) (Barbu/Zhu 2005) (Porway/ Zhu 2009)

General MCMC Design



Computing Multiple Solutions

To faithfully preserve the posterior probability $p(W|\mathbf{I})$,
We compute *a set of weighted scene particles* $\{W_1, W_2, \ldots, W_M\}$,

$$\hat{p}(W|\mathbf{I}) = \sum_{i=1}^{M} \alpha_i G(W - W_i), \quad \sum_{i=1}^{M} \alpha_i = 1$$

A mathematical principle:

$$S^* = \{W_1, W_2, \ldots, W_M\} = \arg\min_S D(p||\hat{p})$$

Computing Multiple Solutions

To faithfully preserve the posterior probability $p(W|\mathbf{I})$,
We compute *a set of weighted scene particles* $\{W_1, W_2, \ldots, W_M\}$,

$$\hat{p}(W|\mathbf{I}) = \sum_{i=1}^{M} \alpha_i G(W - W_i), \quad \sum_{i=1}^{M} \alpha_i = 1$$

A mathematical principle:

$$S^* = \{W_1, W_2, \ldots, W_M\} = \arg\min_S D(p||\hat{p})$$

Stat232B. Stat Computing and Inference, S.C. Zhu

Pursuit of Multiple Solutions

$$D(p||\hat{p}) \approx \log \frac{\varpi}{\boldsymbol{\omega}} + \sum_{n=1}^{N} \frac{\omega_n}{\boldsymbol{\omega}}[(E(\mathbf{x}_{\tau(c(n))}) - E(\mathbf{x}_n)) + \frac{(\mathbf{x}_n - \mathbf{x}_{\tau(c(n))})^2}{2\sigma^2}]$$
$$= \hat{D}(p||\hat{p})$$

The Kullback-Leibler divergence can be computed if we assume mixture of Gaussian distributions

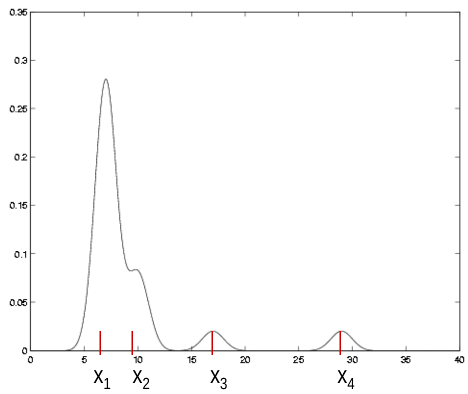
--- a simple fact: the KL-divergence of two Gaussians is the signal-to-noise rat

Intuition: S includes global maximum, local modes, apart from each other.

A *k*-adventurer algorithm

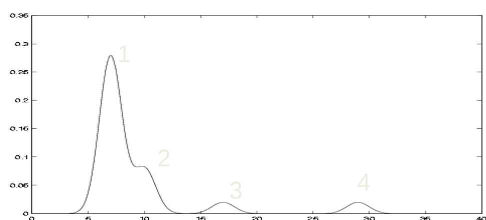
1. Initializing S_k by one $\mathbf{x}$ repeated k times and initializing $\hat{p}(\mathbf{x})$.
2. Repeat
3. Get k new weighted particle $(\omega'_i, \mathbf{x}'_i)$ by k MCMCs starting at $\mathbf{x}_i$ respectively.
4. $S_+ \leftarrow S_k \cup \{(\omega'_i, \mathbf{x}'_i), i = 1, ..., k\}$.
5. $p(\mathbf{x}) \leftarrow \hat{p}(\mathbf{x})$ by adding new particles.
6. $s^* = \arg\min_{|s|=k, s \in S_+} D(p||\hat{p}_+(s))$.
7. $S_k \leftarrow s^*$.

Preserving Distinct Particles



An Example of Keeping Multiple Solutions

An example of illustration:

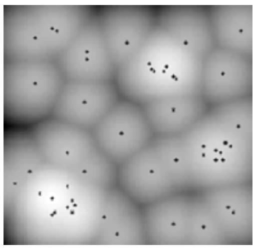


chosen S_3:	$\{x_1, x_2, x_3\}$	$\{x_1, x_2, x_4\}$	$\{x_1, x_3, x_4\}$	$\{x_2, x_3, x_4\}$
$D(p\|\|\hat{p})$:	3.5487	1.1029	0.5373	2.9430
$\hat{D}(p\|\|\hat{p})$:	3.5487	1.1044	0.4263	2.8230
$\|p - \hat{p}\|$:	0.1000	0.1000	0.3500	1.2482

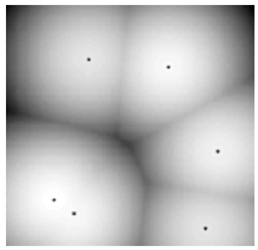
Preserving Distinct Particles

An example of illustration:

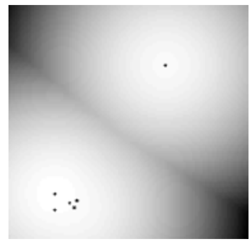
min $D(q \,||\, q_1)$ min $| q - q_2 |$



q



q_1



q_2

A model *p* with 50 particles two approximate models *q* with 6 particles