Lecture 2

Inference in And-Or Graph

--- Scheduling Top-Down/Bottom-Up Processes
--- Computing Multiple Solutions

Song-Chun Zhu

University of California, Los Angeles, USA
Lotus Hill Research Institute, China

Ref: [1] X. Yang, T. F. Wu, and S.C. Zhu. "Evaluating Information Contributions of Bottom-up and Top-down Processes", ICCV09.
[2] J. Porway S.C. Zhu, "C4: Cluster Sampling with Collaborative and Competitive Constraints".

Beijing Summer School, July 2009.

Seeing as Statistical Inference

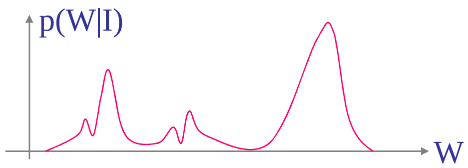
A basic assumption, since Helmholtz (1860), is that biologic and machine vision compute the most probable interpretation(s) from input images.

Let I be an image and W be a semantic representation of the world.

$$W^* = \arg\max_{W \in \Omega} p(W \mid I) = \arg\max_{W \in \Omega} p(I \mid W)\, p(W)$$

In statistics, we need to preserve the full posterior.

$$(W_1, W_2, \ldots, W_k) \sim p(W \mid I)$$



Bayes

Top-down / Bottom-up Inference at all levels

Objective: Constructing parse graphs on-line !

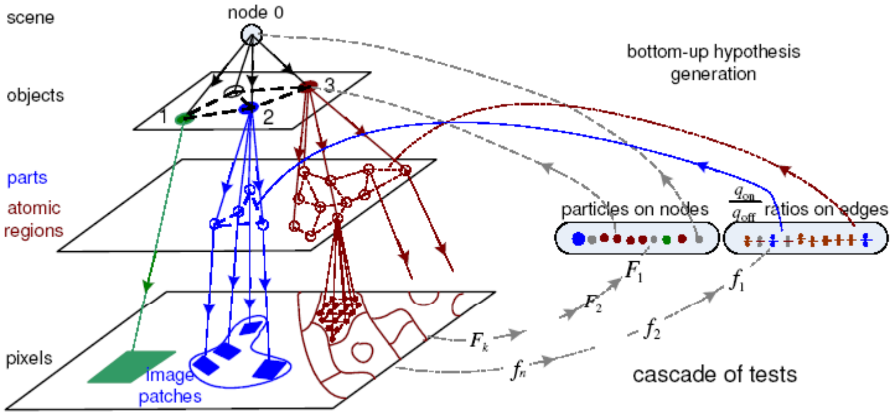
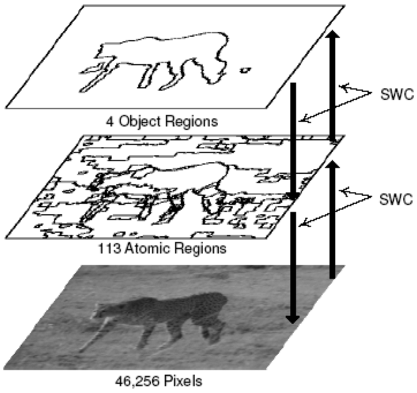


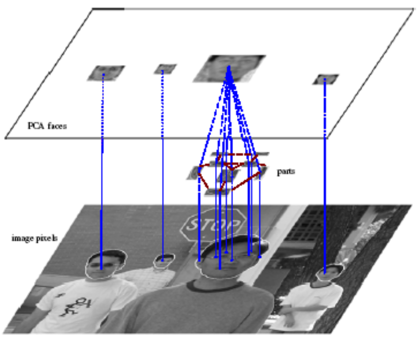
Image parsing by DDMCMC, Tu et al, 2002-05

Two Basic Computing Mechanisms: Bottom-up vs. Top-down

Some objects can be computed more effectively by bottom-up while others by top-down



(a) bottom-up graph construction



(b) Top-down graph construction

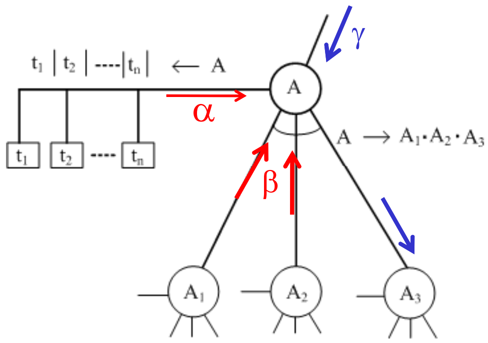
How to formulate this problem ?

Part 2.A: α, β and γ computing processes in AoG

The And-Or graph is a recursive structure. So, consider a node A.

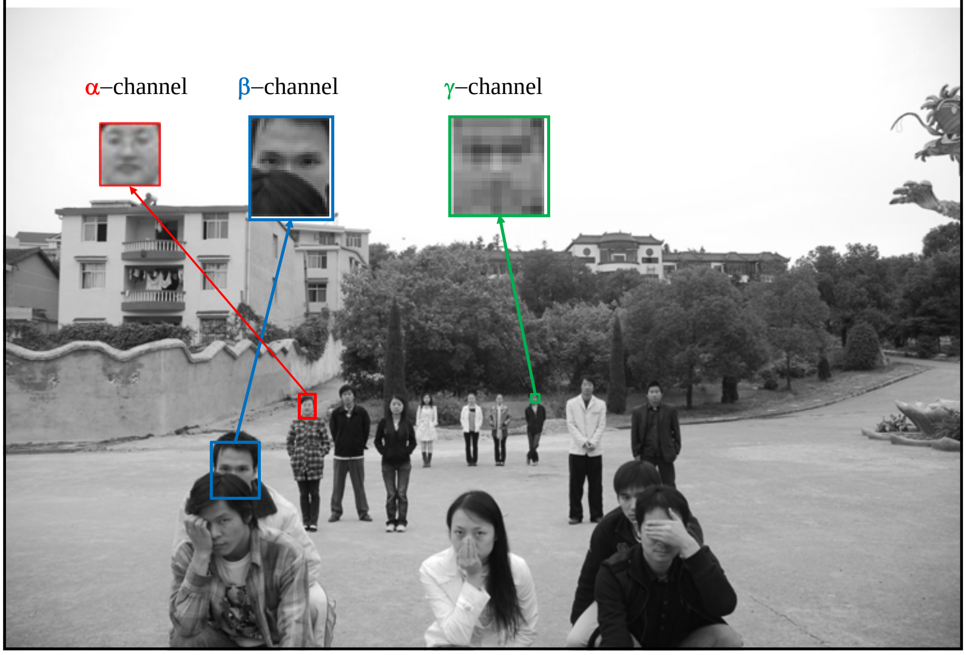
1, any node A terminate to leaf nodes at a coarse scale (ground).
2, any node A is connected to the root.

Starting the $\alpha/\beta/\gamma$ channels when they are applicable ---an optimal scheduling problem



Compositional boosting, T.F. Wu et al, CVPR 07

An example: human faces are computed in 3 channels



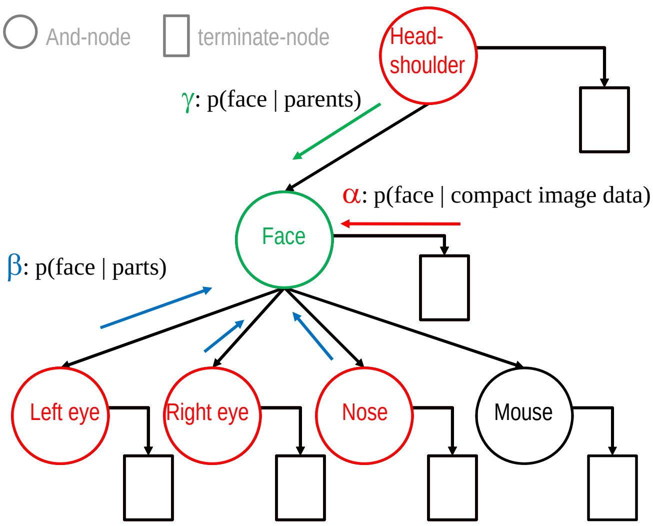
Human faces in real scenarios

α-channel

β-channel

γ-channel

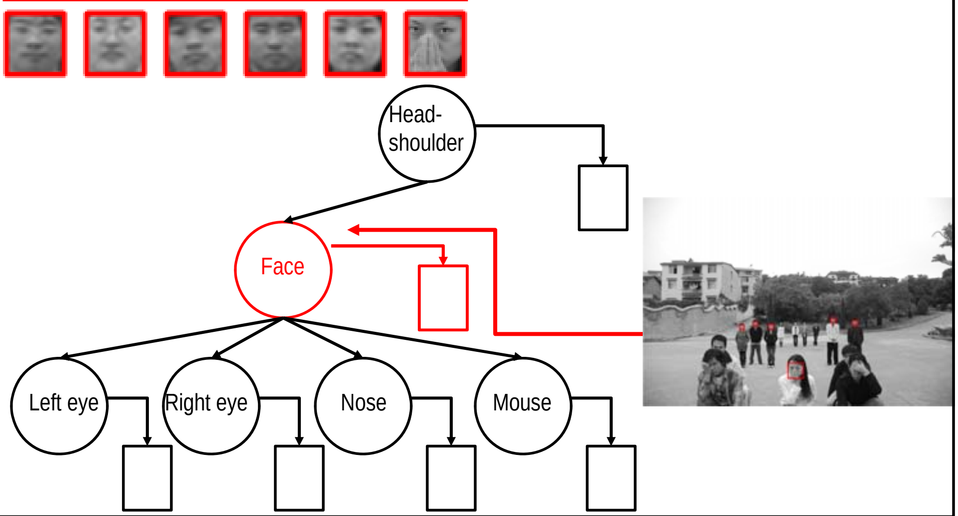
Hierarchical modeling and α, β and γ computing



1. Each node has its own α, β and γ computing processes.
2. How much does each channel contribute?

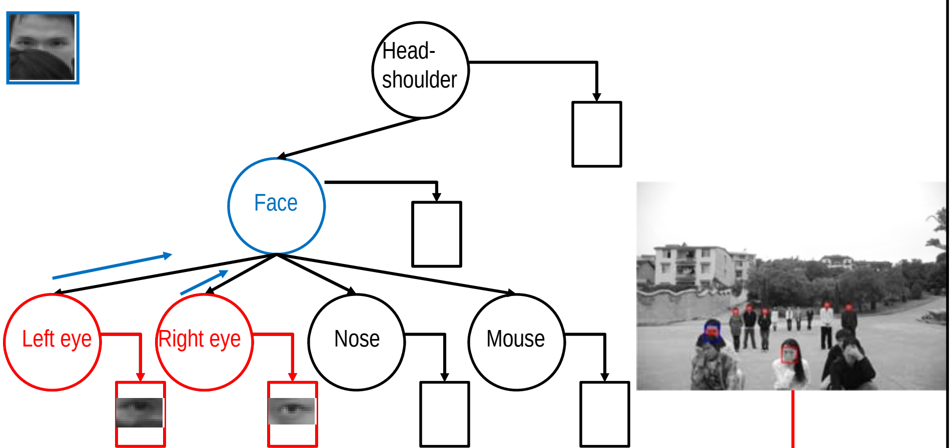
α processes for the face node

α-channels: p(face | compact image data)



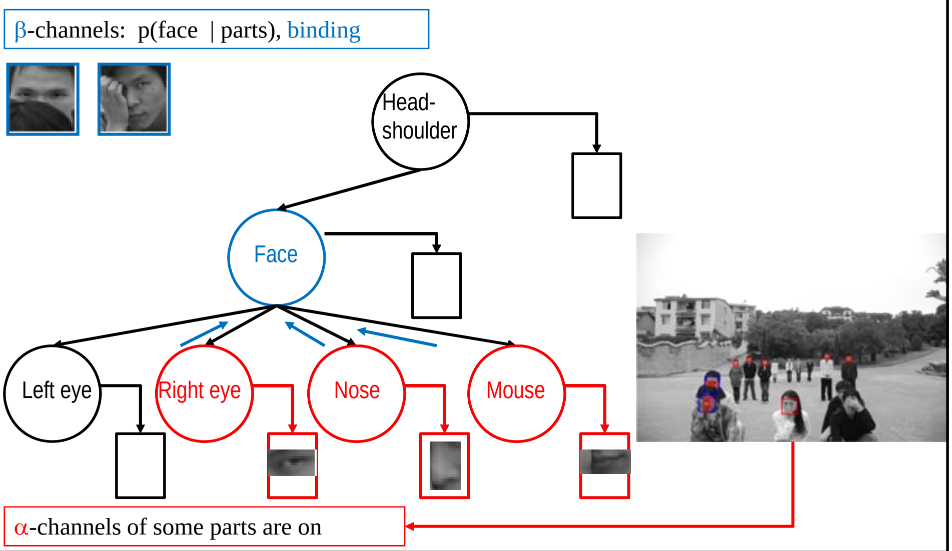
β processes for the face node(when its α is off)

β-channels: p(face | parts), binding

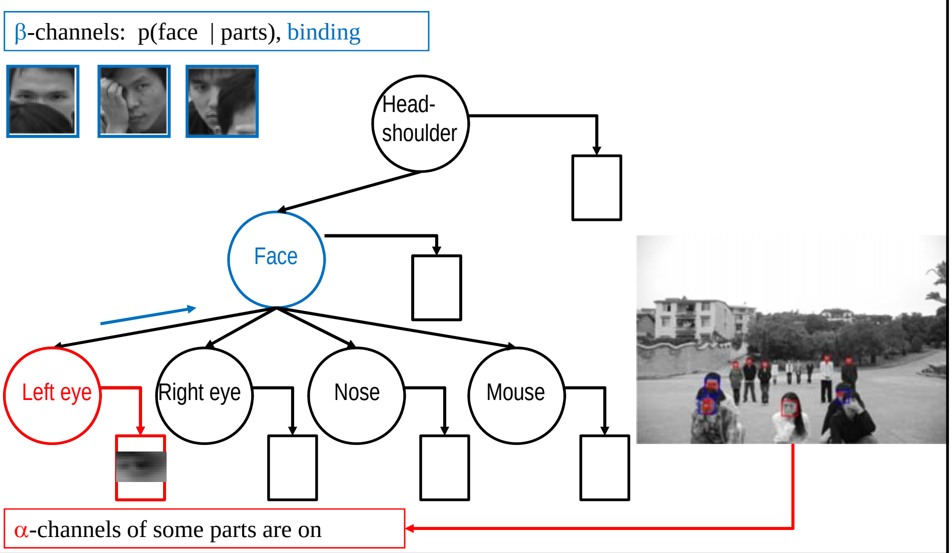


α-channels of some parts are on

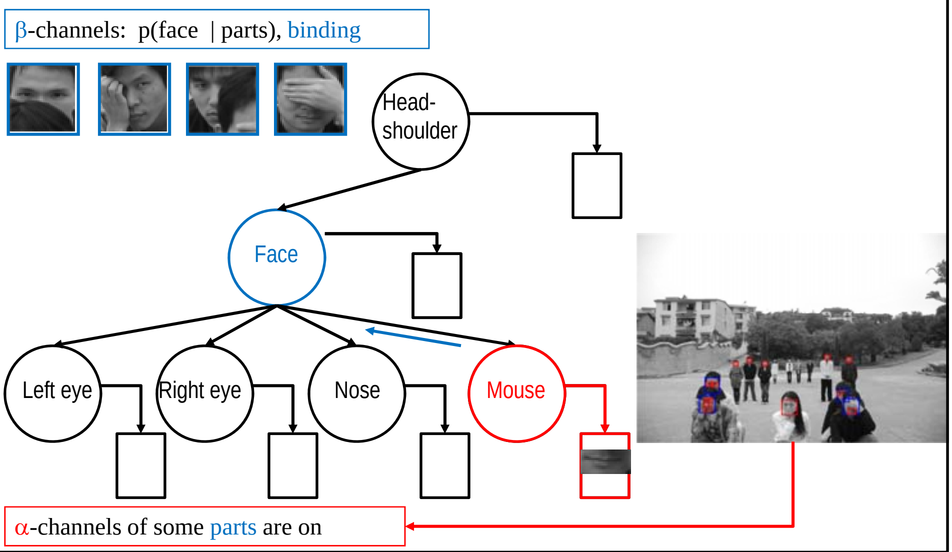
β processes for the face node(when its α is off)



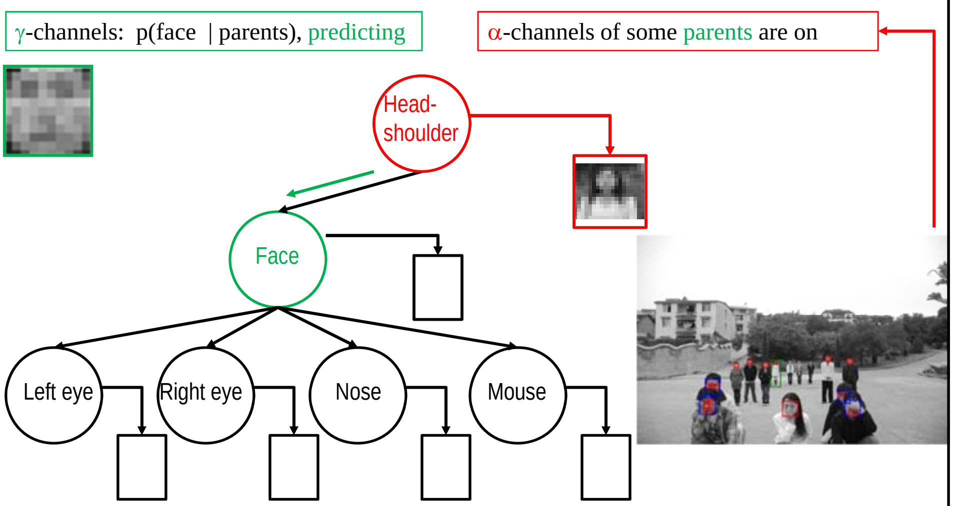
β processes for the face node(when its α is off)



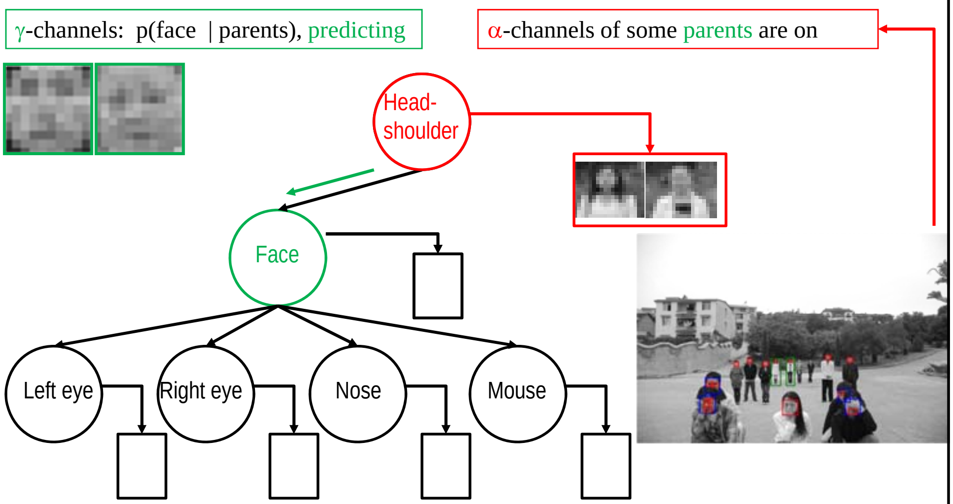
β processes for the face node(when its α is off)



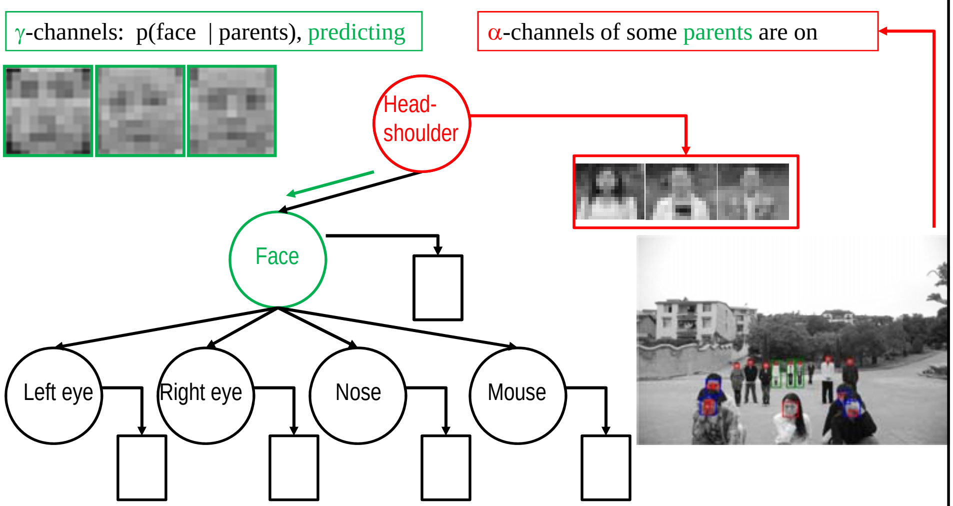
γ processes for the face node(when it's α and β is off)



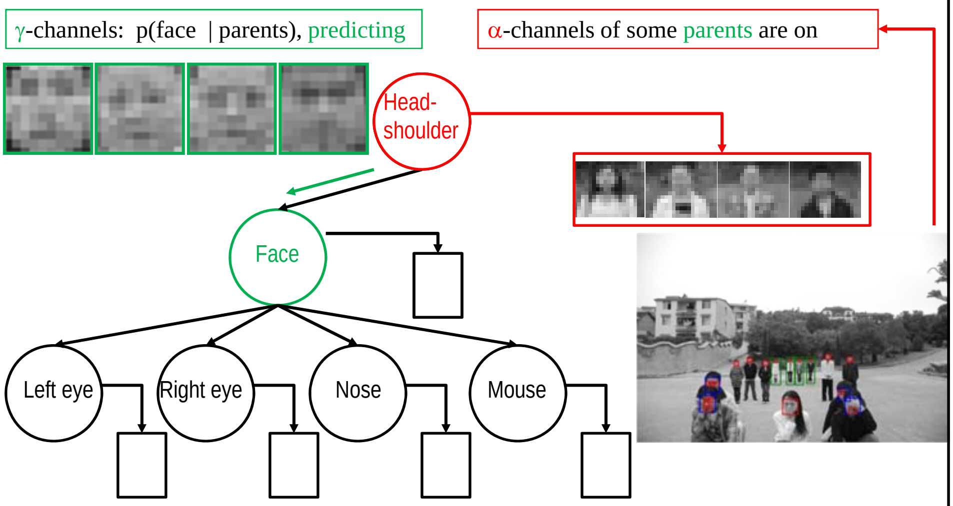
γ processes for the face node(when it's α and β is off)



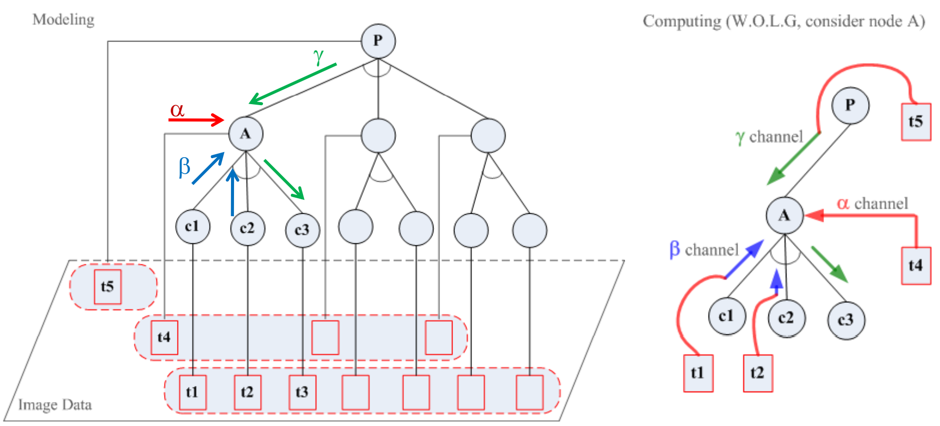
γ processes for the face node(when it's α and β is off)



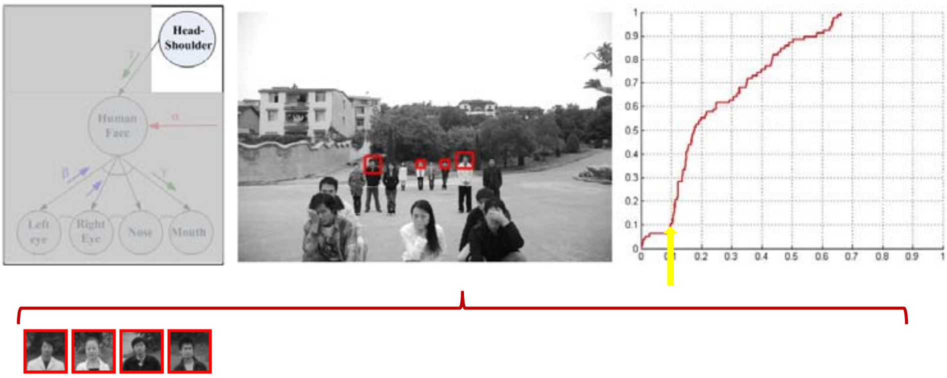
γ processes for the face node(when it's α and β is off)



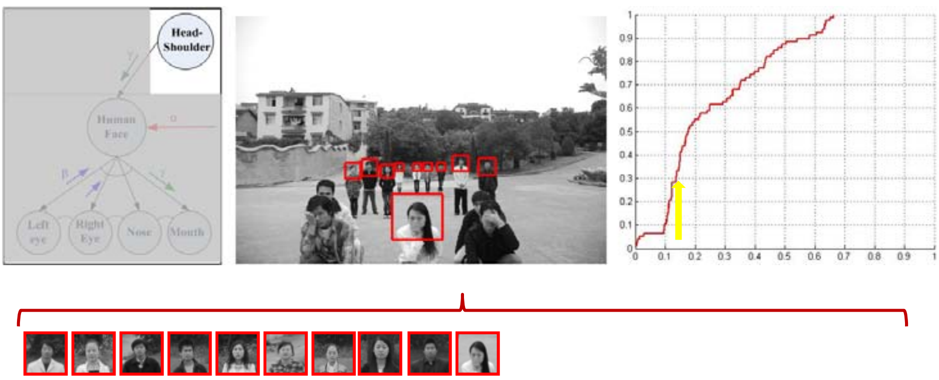
In general: recursive α, β and γ channels



α–channel: head-shoulder

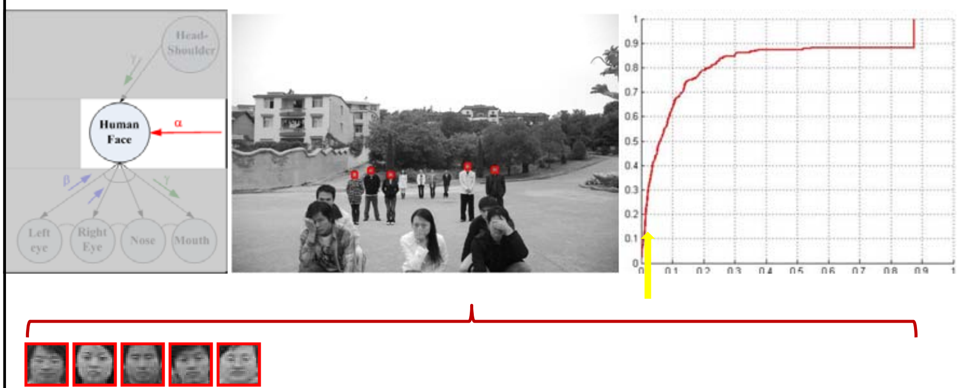


α–channel: head-shoulder

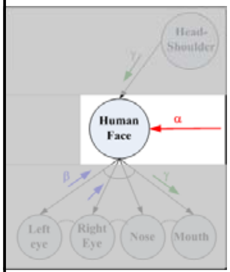


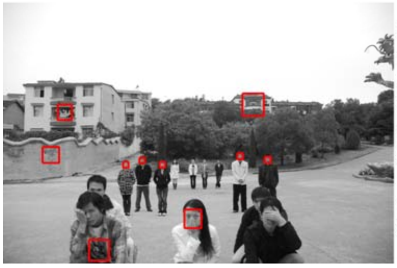
α–channel: head-shoulder

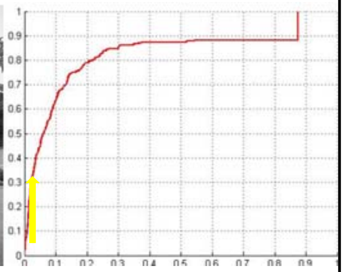
α–channel: face

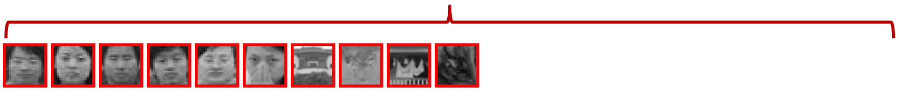


α–channel: face

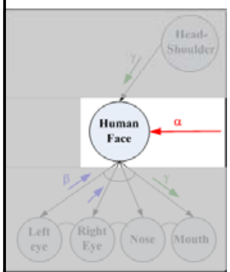


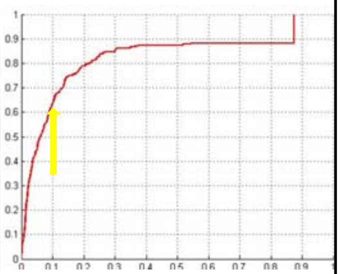


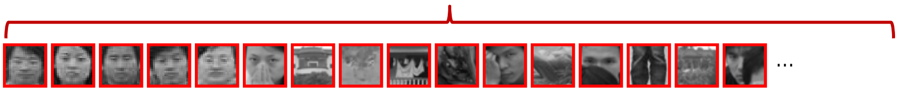




α–channel: face

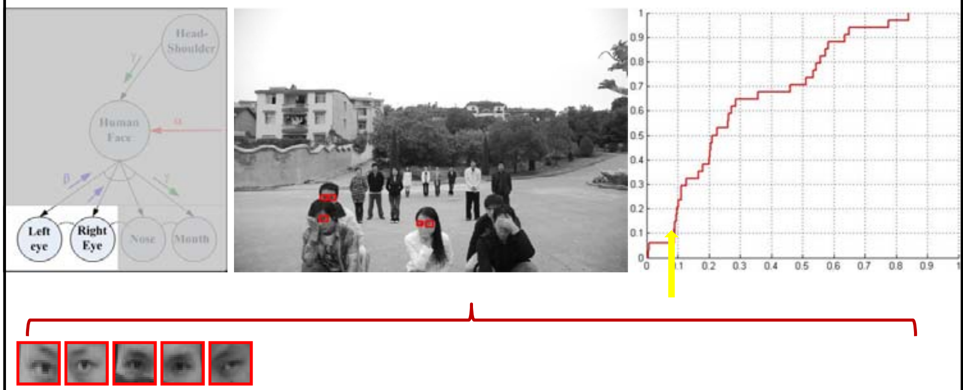






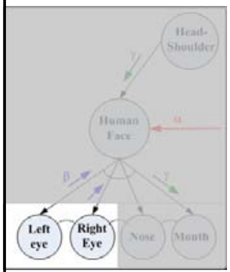
…

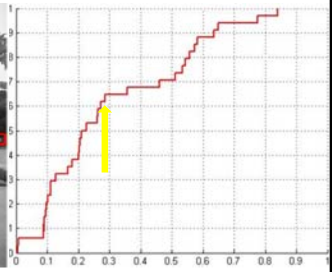
α–channel: eye

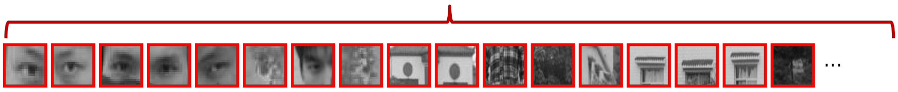


α–channel: eye

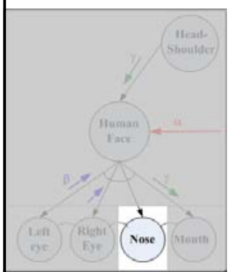
α–channel: eye

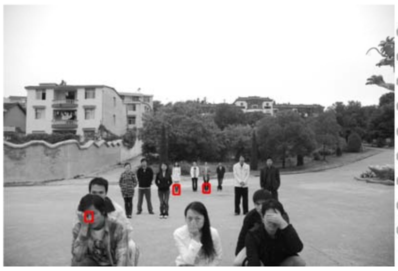


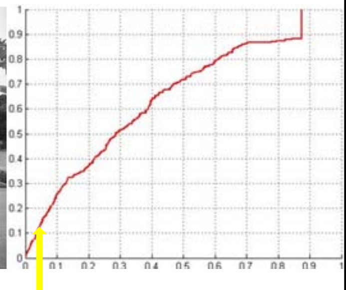


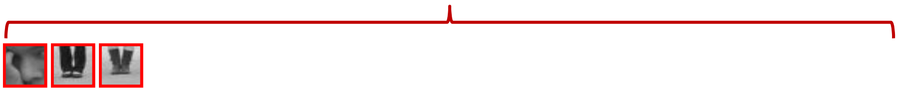

…

α–channel: nose

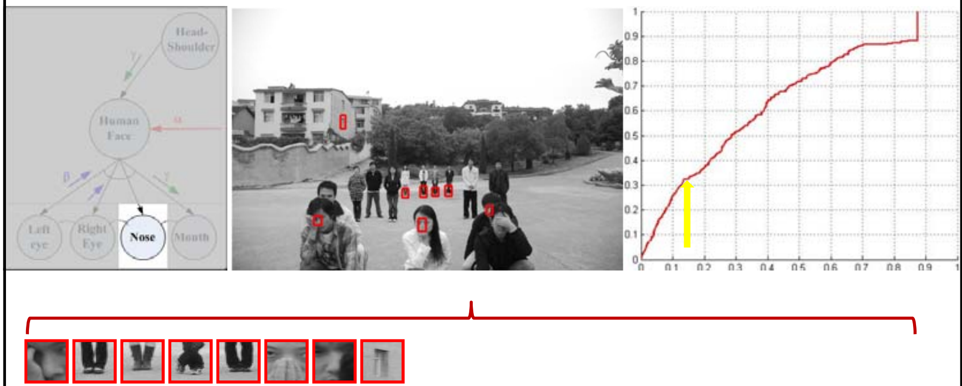




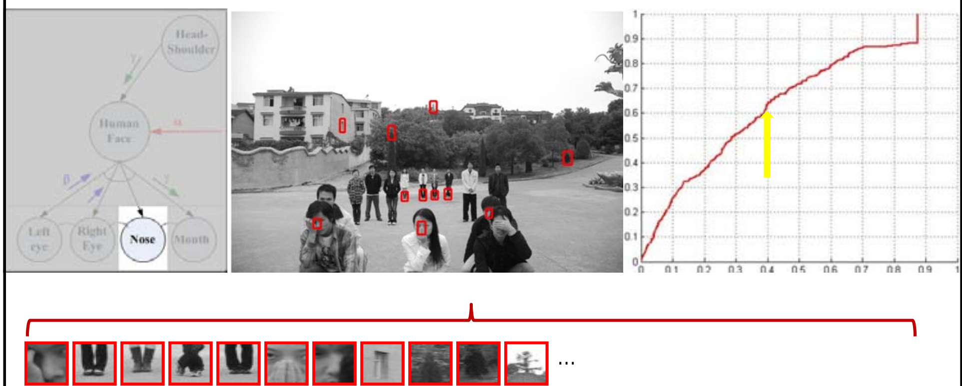




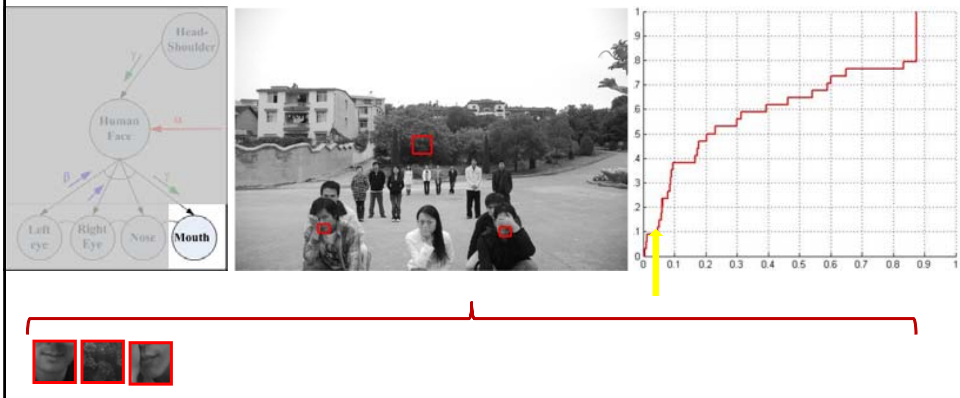
α–channel: nose



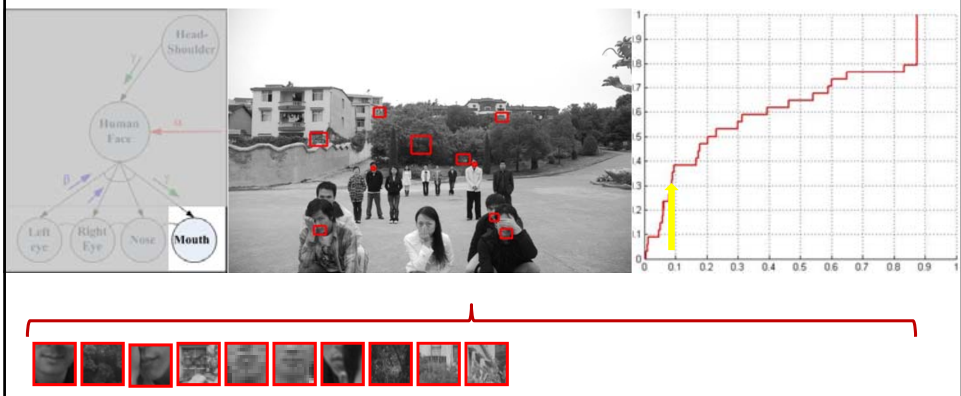
α–channel: nose



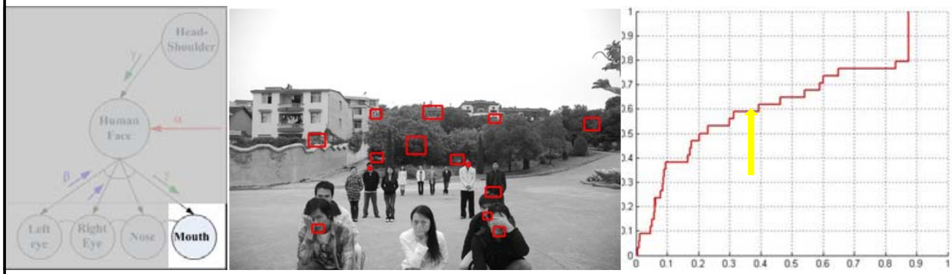
α–channel: mouth



α–channel: mouth

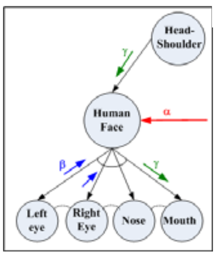


α–channel: mouth

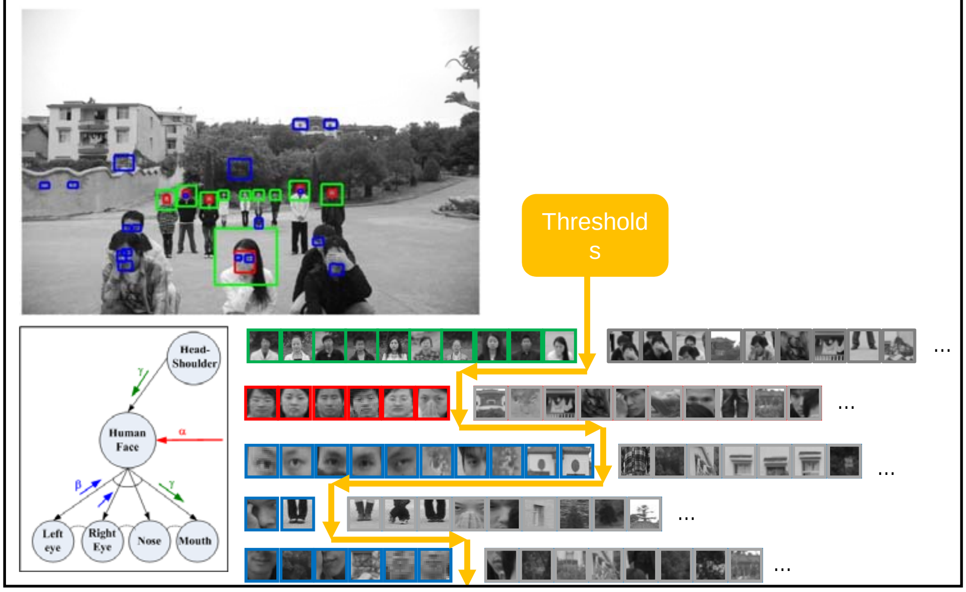


All α channels

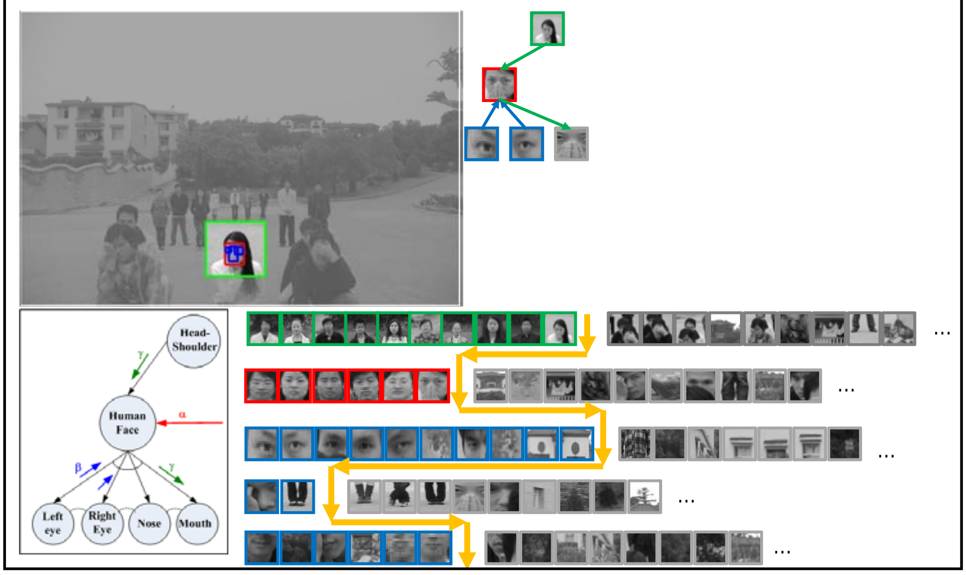
keep things and throw away stuffs by integrating α, β and γ channels



Integrating α, β and γ channels



Integrating α, β and γ channels

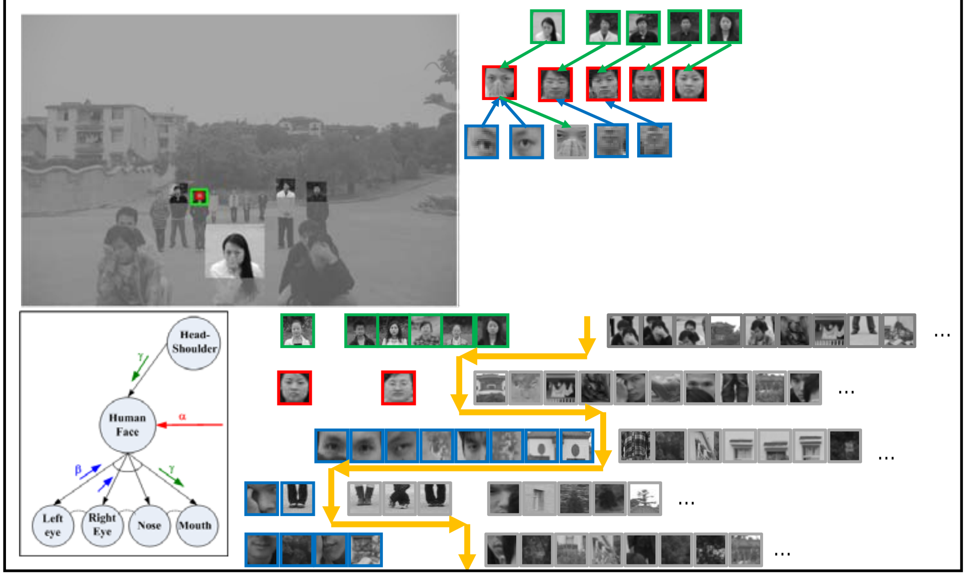


Integrating α, β and γ channels

Integrating α, β and γ channels

Integrating α, β and γ channels

Integrating α, β and γ channels



Integrating α, β and γ channels

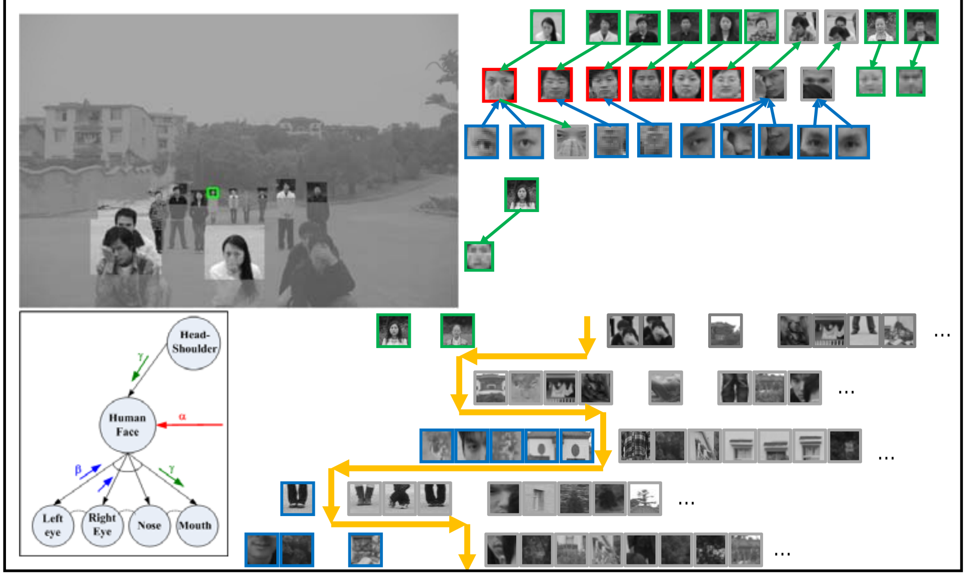
Integrating α, β and γ channels

Integrating α, β and γ channels

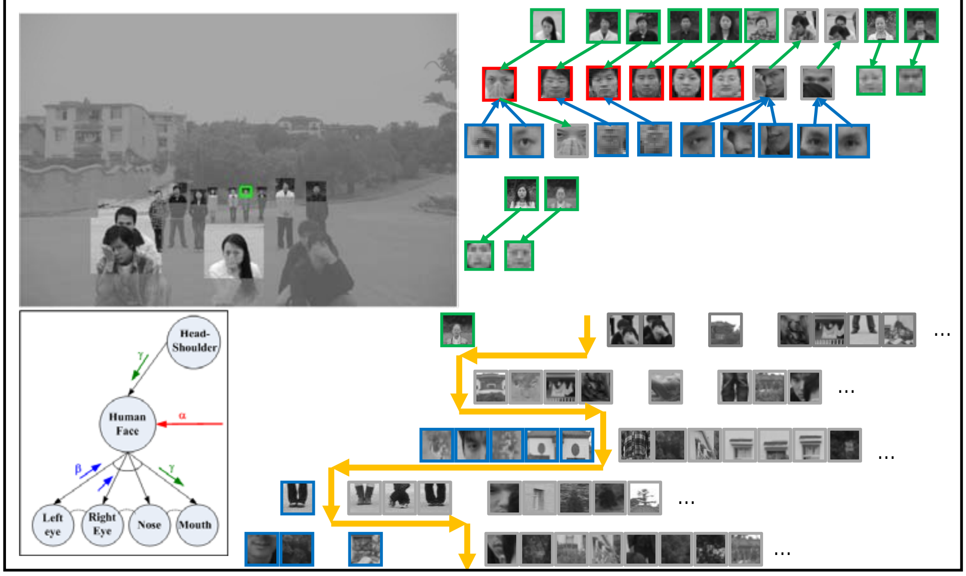
Integrating α, β and γ channels

Integrating α, β and γ channels

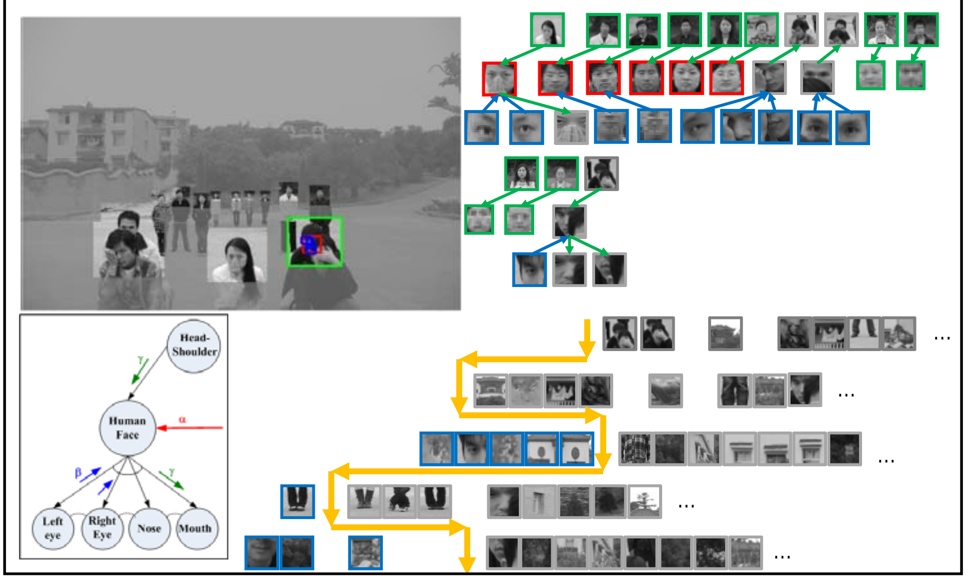
Integrating α, β and γ channels



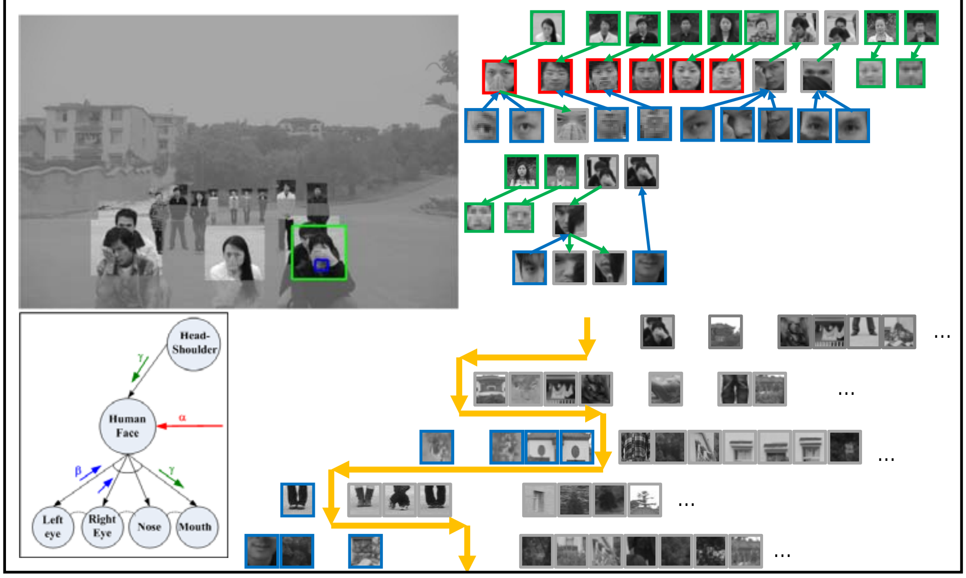
Integrating α, β and γ channels



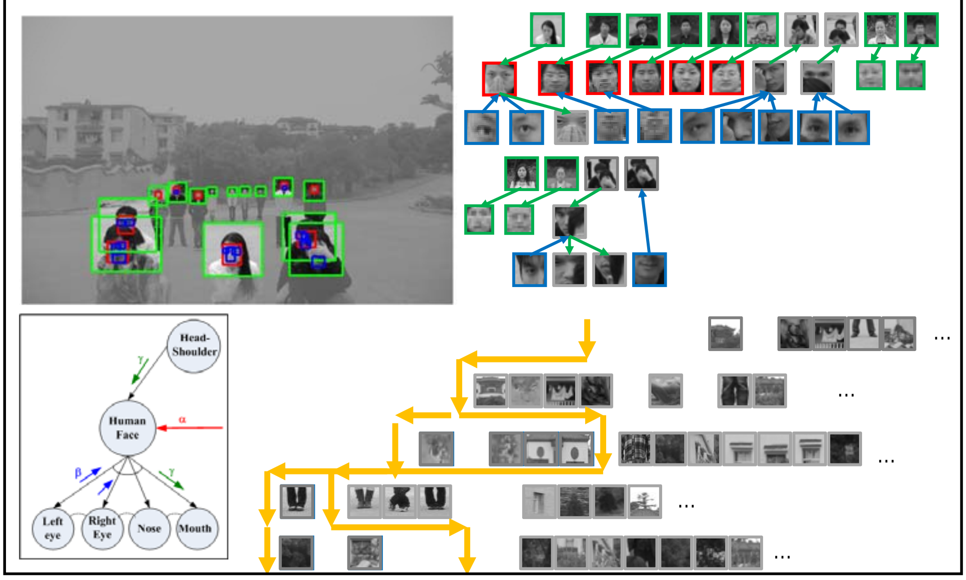
Integrating α, β and γ channels



Integrating α, β and γ channels

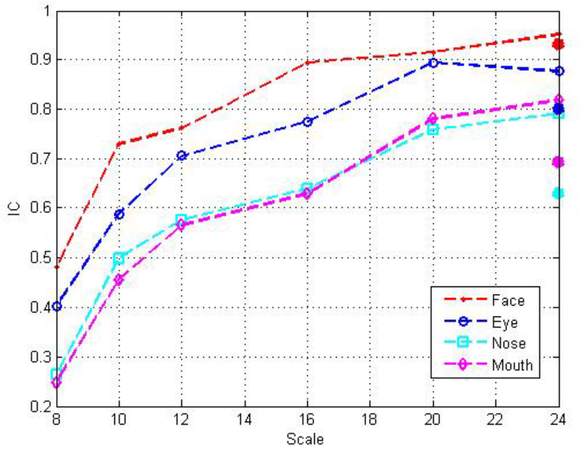


Integrating α, β and γ channels



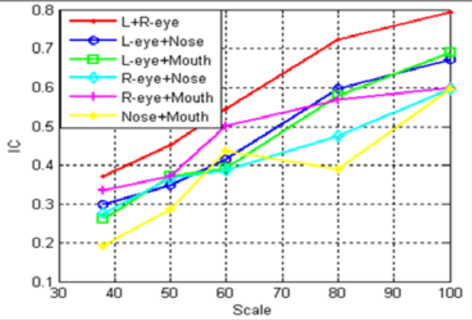
Information contribution

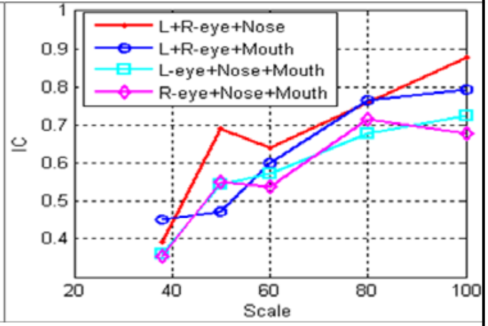
α channels



Information contribution

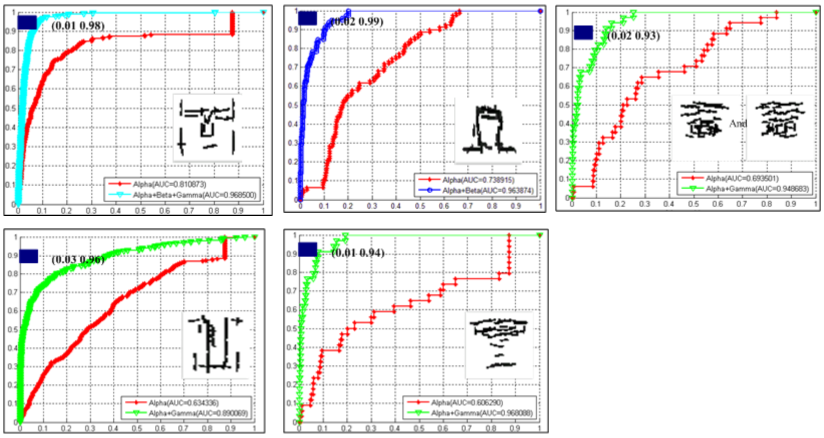
- β channels





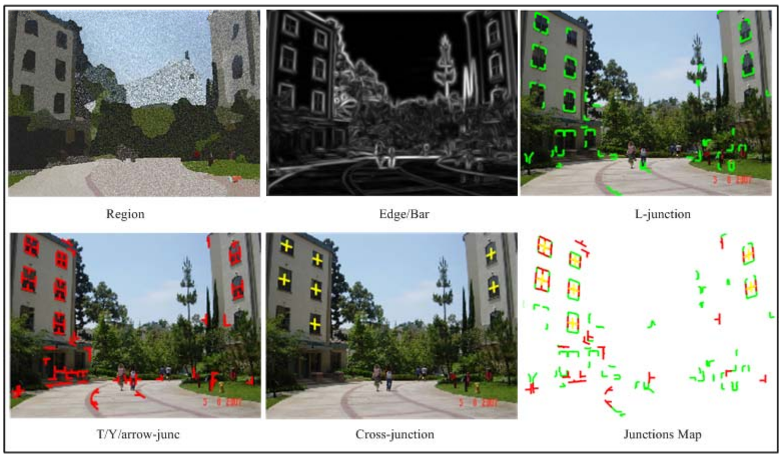
Performance improvement

red for α, blue for α+β, green for α+γ, cyan for α+β+γ channels



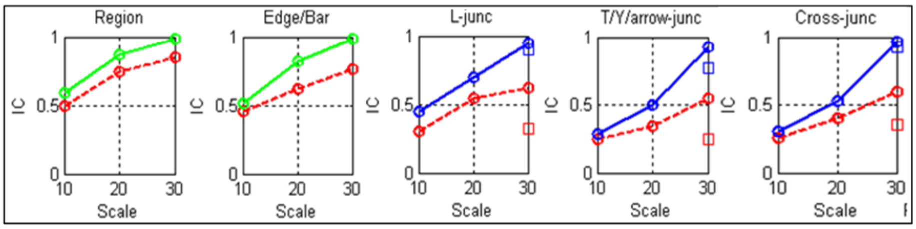
At low-middle level: α, β and γ channels

junctions: β channels dominate, say, binding.

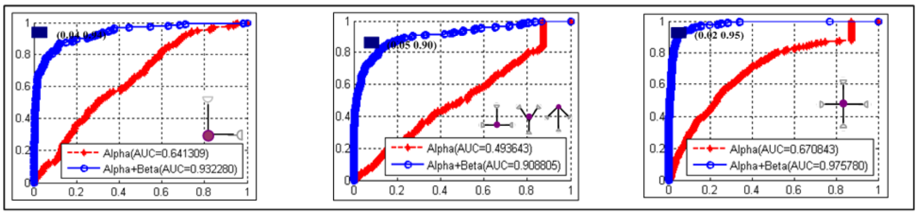


At low-middle level: α, β and γ channels

junctions: β channels dominate, say, binding.



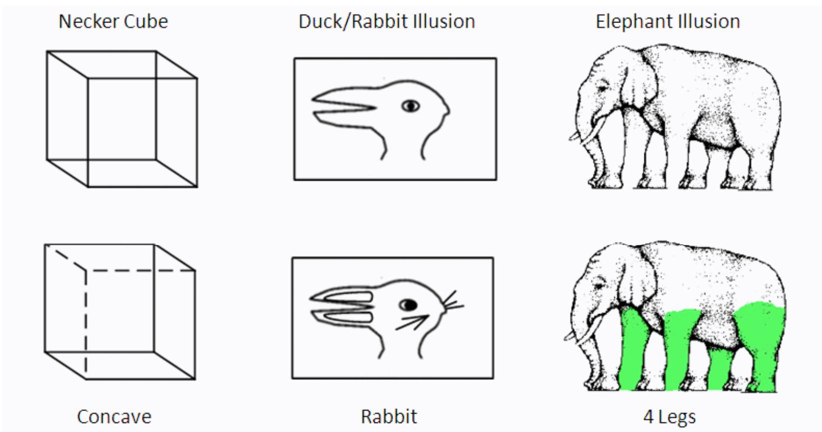
(a) Information contributions (IC) evaluated for the five low-level primitives: α(in red), $\alpha+\beta$(in blue) $\alpha+\beta+\gamma$(in green).



(b) The testing ROCs of α(in red) and $\alpha+\beta$(in blue) for L, T/Y/arrow and cross junctions, respectively.

2.B Computing Multiple Solutions

(1), preserve uncertainty, and (2), avoid premature commitments.



Here are two more examples

We showed that computers can dream. Can computers find these solutions ?

Ambiguities are ubiquitous in images

Local interpretations are often strongly coupled !

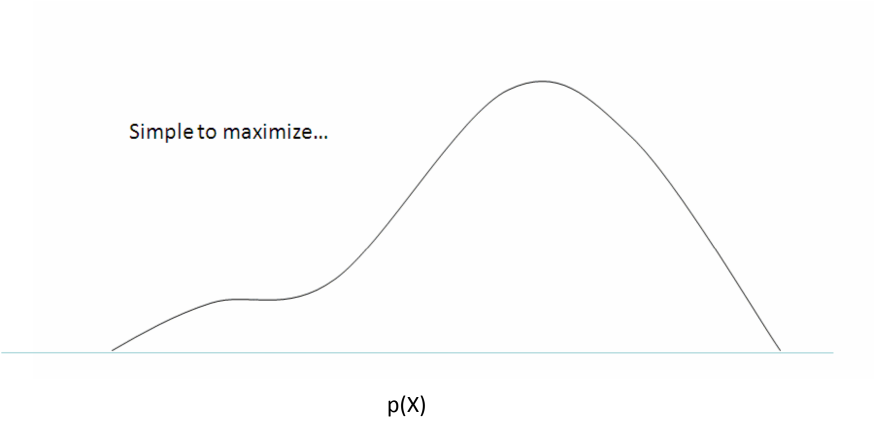
They form “clusters” and the search algorithms often get stuck.

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



Background: Swendsen-Wang 1987

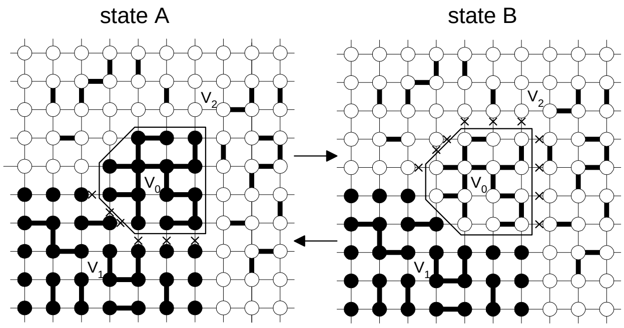
Ising model: $p(X) = \frac{1}{Z} \prod_{<i,j>} \varphi(x_i, x_j); \quad \varphi(x_i, x_j) = e^{\beta \cdot \delta(x_i = x_j)}; \quad \delta() \in \{-1, +1\}$

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

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

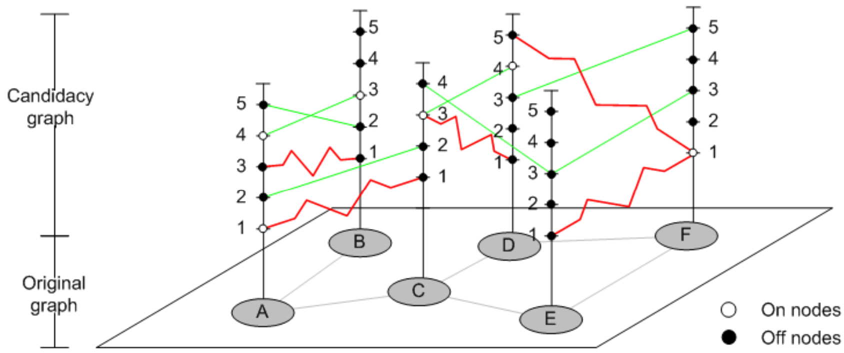
(b) Turn edges "on" (u_{ij} = +1) or "off" (u_{ij} = -1) probabilistically.

(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.



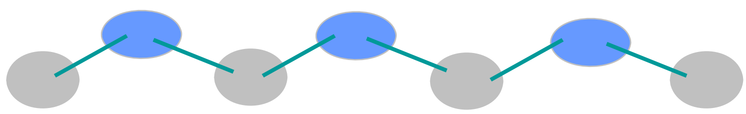
MCMC 101: Sampling with Auxiliary Variables

Augment the Ising model with bond variable U:

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

Define the joint probability so that

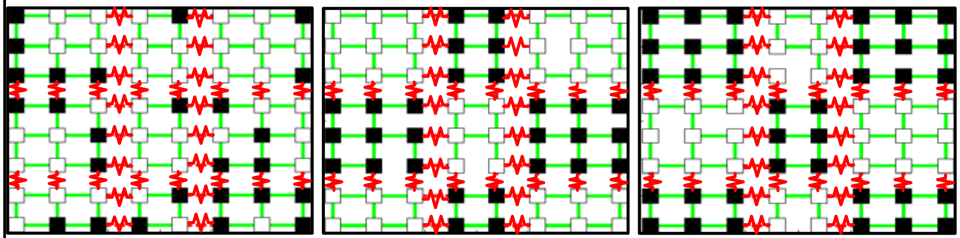
1. $\sum_U p(X,U)=p(X)$
2. p(U|X) and p(X|U) are easy to sample from.



(Edwards and Sokal, 1988)

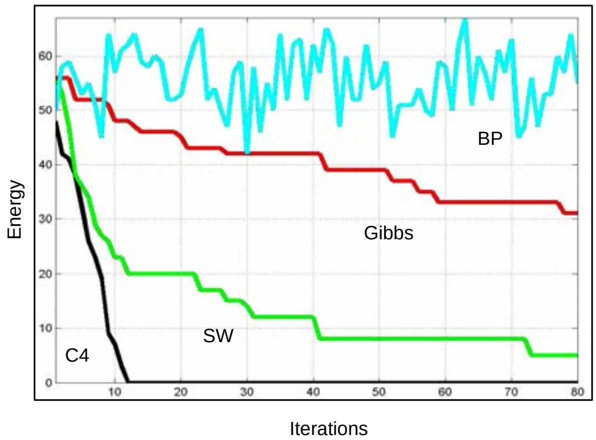
Experiments on Negative Edge Ising Model

Created “checkerboard” constraint problem.



Initial State Solution 1 Solution 2

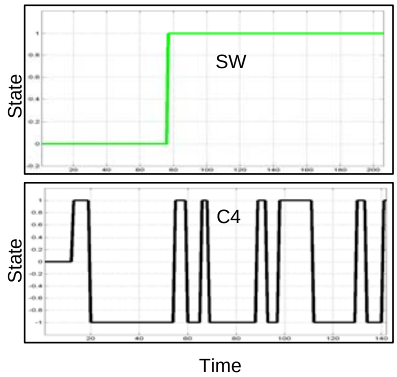
Convergence plot: Energy over iterations



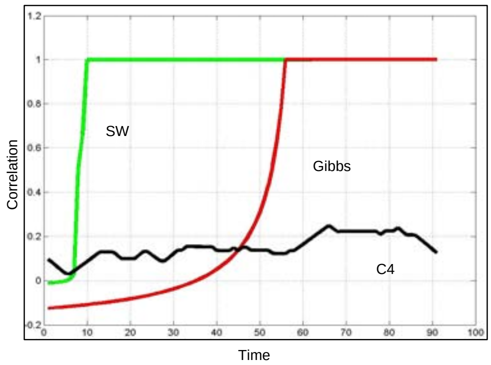
C4 converges fastest of all.
Gibbs takes huge amount of time to converge.
BP has trouble with loops and never converges.

Mixing rate: Correlation plot

solution states over time



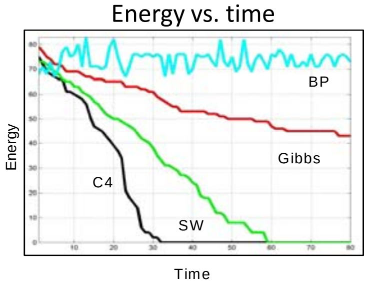
auto-correlation over time



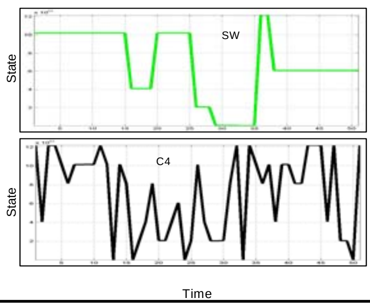
C4 has
- Low burn-in time (converges quickly).
- High mixing rate (samples remaing unbiased over short run).

Simulating the Potts model

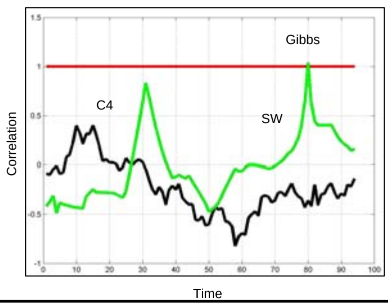
Energy vs. time



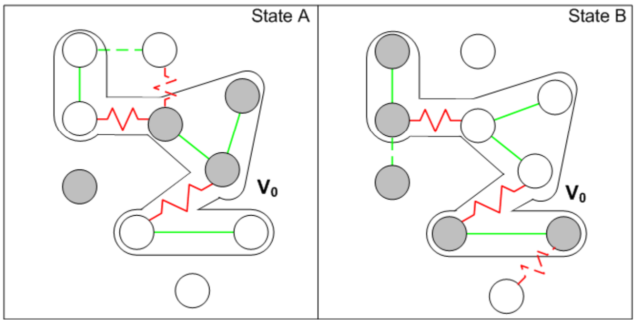
State vs. time



Correlation vs. time



C4: Generalization to Arbitrary Posteriors



Same *protocol with Potts model*, different *dynamics.*

1. p(U|X) → Data driven edge probabilities. Learn distributions on features of node cliques.
2. p(X|U) → Random sampling or CSP.

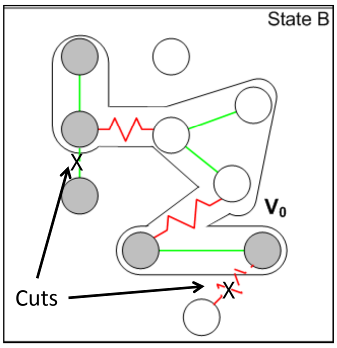
Metropolis-Hastings Design

$$\alpha(A \to B) = \min\left(1, \frac{q(B \to A)}{q(A \to B)} \cdot \frac{p(X_B \mid F, B)}{p(X_A \mid F, A)}\right)$$

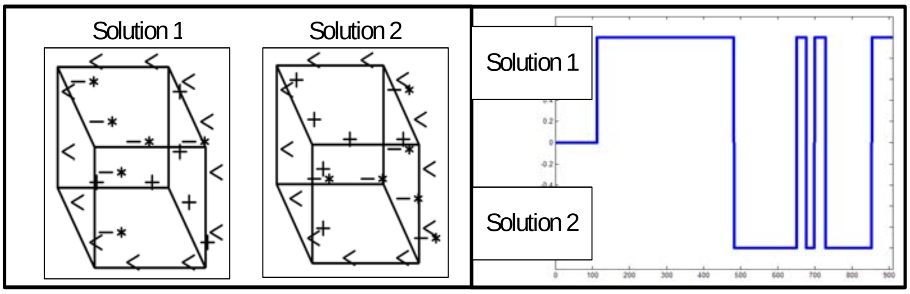
$$\frac{q(B \to A)}{q(A \to B)} = \frac{q(V_0 \mid B)}{q(V_0 \mid A)} \cdot \frac{q(l_{V_0} = \lambda' \mid V_0, B)}{q(l_{V_0} = \lambda \mid V_0, A)}$$

$q(V_0 \mid B)$ depends only on its *cuts*! Ratio simplifies!

$$\frac{q(V_0 \mid B)}{q(V_0 \mid A)} = \frac{\prod_{e \in cut(V_0, B)} (1 - q_e)}{\prod_{e \in cut(V_0, A)} (1 - q_e)}$$



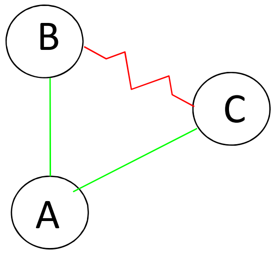
Solving the Necker Cube problem



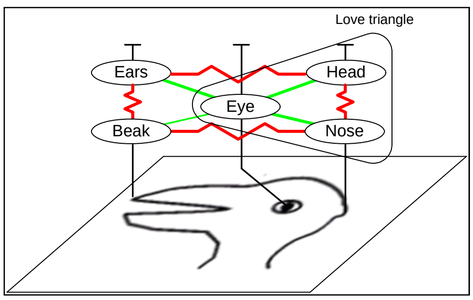
C4 finds both solutions, swaps corner labels continuously.

Problem with "Flat"--C4

The "love triangles":
create inconsistent clusters

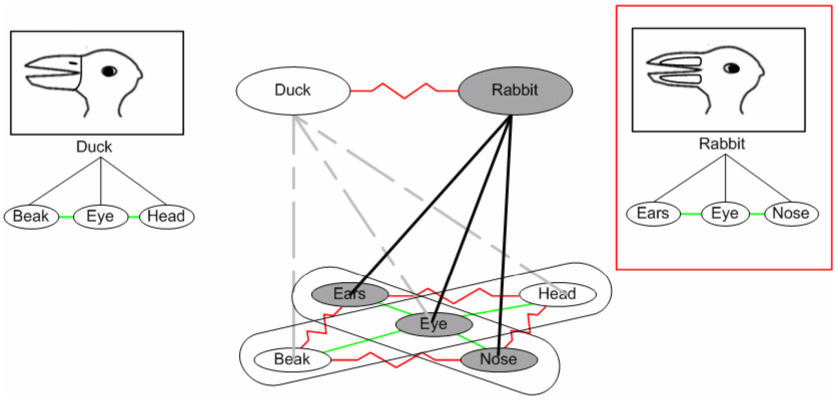


e.g. Love triangles in the duck / rabbit illusion.



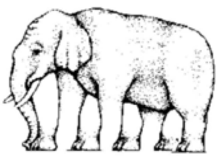
Hierarchical C4

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

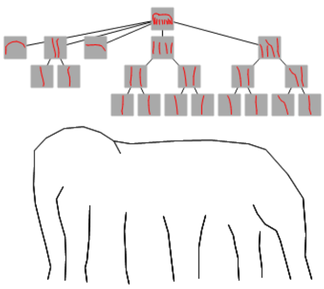


System will now flip between duck and rabbit without love triangle issue.

Solving the Elephant Illusion



Hierarchical part model



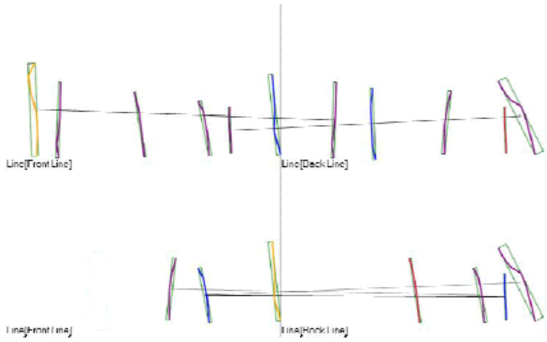
Layered representation of hierarchy

Elephant
Leg Pair, Head, Back
Leg, Trunk
Line

Composing by the β-channels

Trunk compatibilities

Leg compatibilities



Elephant

Leg Pair, Head, Back

Leg, Trunk

Line 

Part binding for next layer

Trunk bindings 

Leg bindings 

Elephant

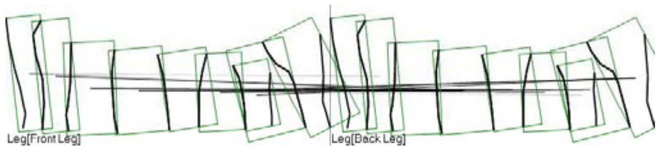
Leg Pair, Head, Back

Leg, Trunk 

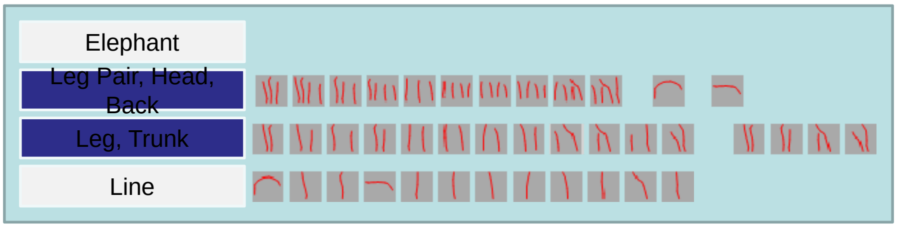
Line 

Continue sampling / binding for each layer

Leg Pair compatibilities

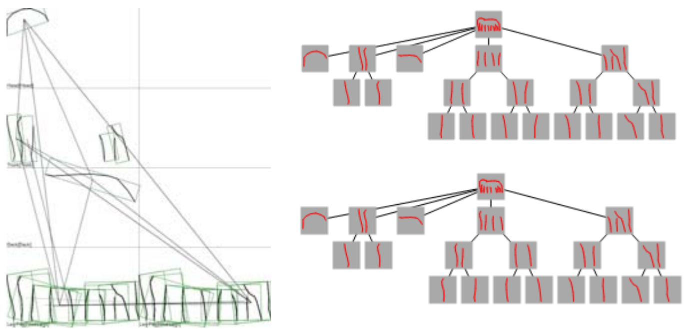


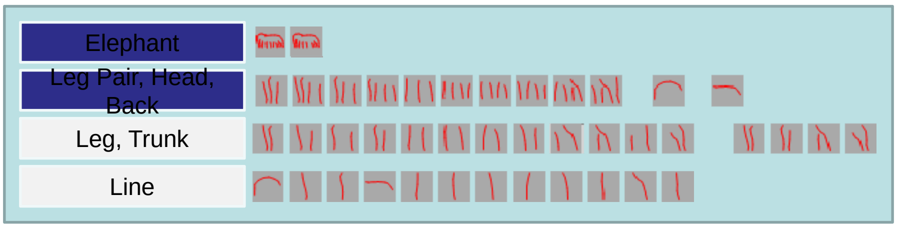
Leg Pair bindings



Top Level Bindings and sampling

Elephant compatibilities





Long standing debates in vision

They have to be resolved by numeric answers

Structure	vs.	Appearance
Hierarchy	vs.	Context
Bottom-up	vs.	Top-down
Generative	vs.	Discriminative
View-centered	vs.	Object-centered