Lecture 1 Introduction to Stat 232B- CS266B

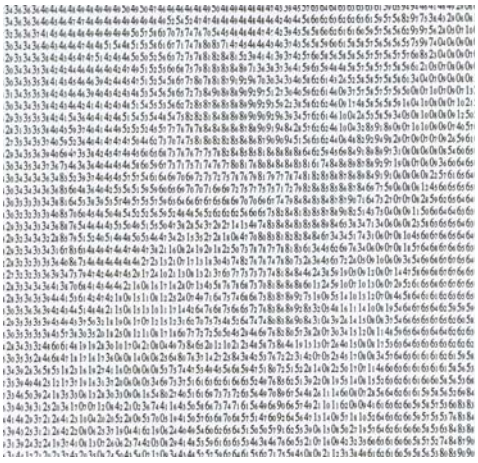
Statistical Computing and Inference in Vision and Image Science

Vision: understanding the contents of images

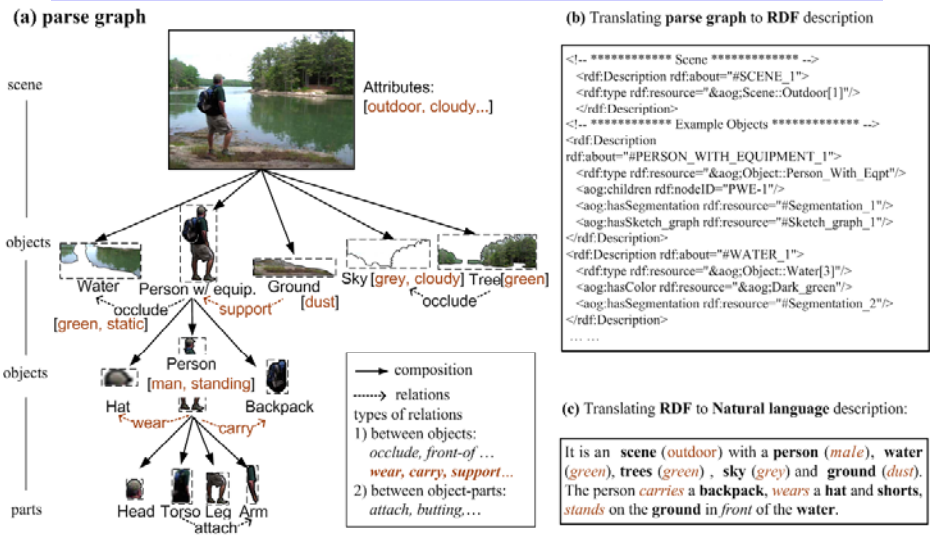
If we view images as signals in a "language" spoken by natural environments, then vision is essentially to parse images, like "natural language understanding":

What are the keywords and grammar structures of images?

What are the suitable computer architectures for this parsing task?

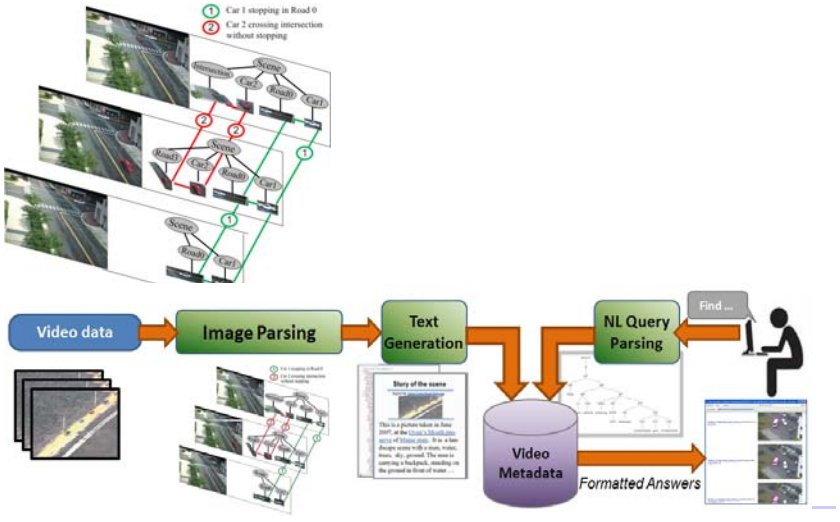


An example of image parsing and text description



Benjamin Yao et al 2009.

An example of Video parsing and text generation



A prototype system for automated text generation

	Land_vehicle_359 approaches intersection_0 along road_0 at 57:27. It stops at 57.29. Land_vehicle_360 approaches intersection_0 along road_3 at 57:31.
	Land_vehicle_360 moves at an above-than-normal average speed of 26.5 mph in zone_4 (approach of road_3 to intersection_0) at 57:32. It enters intersection_0 at 57:32. It leaves intersection_0 at 57:34. There is a possible failure-to-yield violation between 57:27 to 57:36 by Land_vehicle_360.
	Land_vehicle_359 enters intersection_0 at 57:35. It turns right at 57:39. It leaves intersection_0 at 57:36. It exits the scene at the top-left of the image at 57:18.

Ref: Benjamin Yao et al "From image parsing to text generation", 2009.
In collaboration with Mun Wai Lee at ObjectVideo Inc.

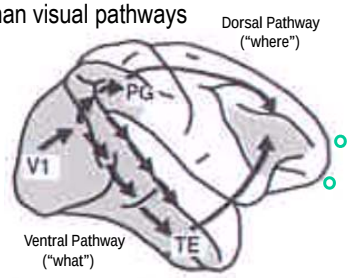
Observations of the vision system

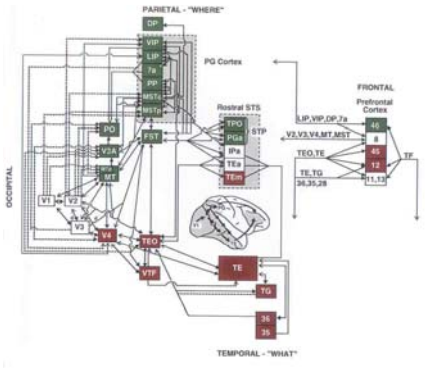
1, Understanding an image needs a vast amount of prior knowledge about the world !

We are hardwired for image understanding

In the visual pathways, there are more downward (top-down) and lateral connections than forward (bottom-up) connections (10 :1)

Human visual pathways





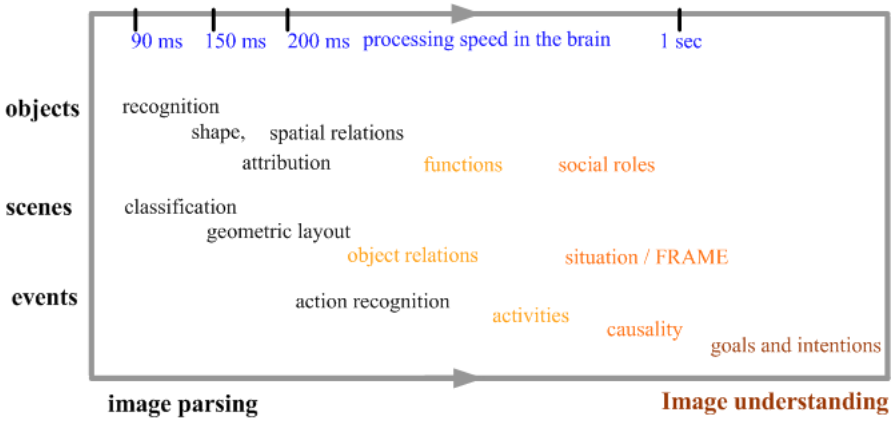
Observations of the vision system

2: Vision seems to be a continuous a computational process:
---- the more you look, the more you see.

Image shown to subjects	40ms	80ms	107ms	500ms
	"Possibly outdoor scene, maybe a farm. I could not tell for sure."	"There seem to be two people in the center of the scene."	" People playing rugby. Two persons in close contact, wrestling, on grass. Another man more distant. Goal in sight."	"Some kind of game or fight. Two groups of two men. One in the foreground was getting a fist in the face. Outdoors, because I see grass and maybe lines on the grass? That is why I think of a game, rough game though, more like rugby than football because they weren't in pads and helmets..."

Human subjects reporting on what he/she saw in an image shown for different presentation durations (PD=27, 40, 67, 80, 107, 500ms). from L. Fei-Fei and P. Perona 2007

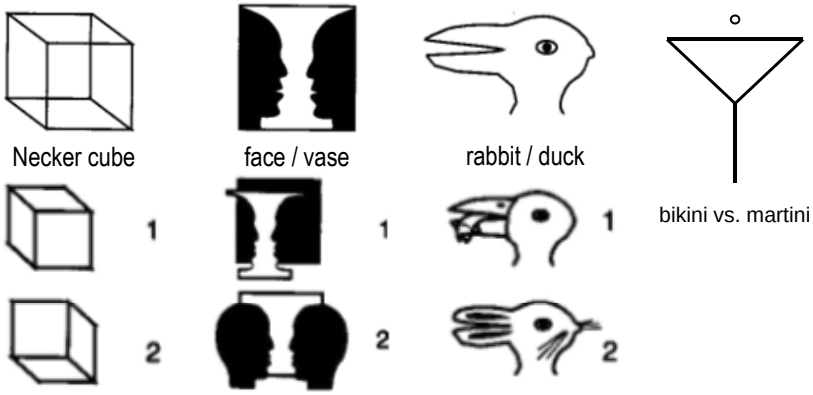
Vision is a continuous (literally infinite) computing process



Observations of the vision system

3: Vision can resolve ambiguities.

In mathematical terms, our perception can switch or jump in some structured state space.



A common property is that the individual elements are strongly coupled and those strongly coupled elements must change their labels together. It is very hard to implement.

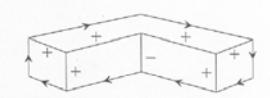
Here are two more challenging examples

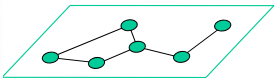
Can computers find and switch between these solutions ?

Zanforlin.mov

Underlying models: A tale of three kingdoms

Waltz, 1960s constrain - satisfaction



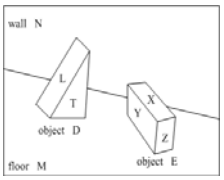


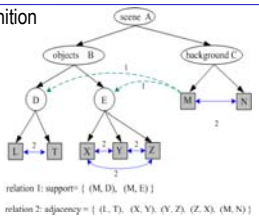
Flat descriptive models

Markov random fields
Graphical models
FRAME, Mixed Random fields

--- contexts at all levels

Fu, 1970s, syntactic pattern recognition



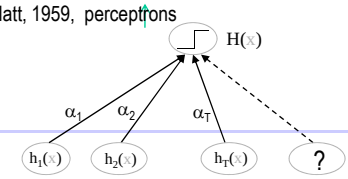


Hierarchic generative models

Stochastic context free grammar
Sparse coding
Wavelets / harmonic analysis
image grammars

--- vocabulary at all levels

Rosenblatt, 1959, perceptrons



Discriminative models

Adaboosting

--- features at all levels

Algorithms for the three types of models

We organize these algorithms in three methods according to the underlying representation.

1. *Descriptive methods*: algorithms working on various graphs where the nodes/vertices represent states of the same semantic level. E.g, constraint-satisfaction, relaxation-labeling, dynamic programming, Viterbi, belief propagation, Gibbs sampler, Sequential Monte Carlo, Swendsen-Wang cuts.

2. *Generative methods:* algorithms working on hierarchic graph representations where one level of vertices semantically generate the nodes at the level below as parts/components. E.g. heuristic search, search on And-Or graphs, matching pursuit, various parsing algorithms (inside-outside, Earley), Metropolis-Hastings, Markov chain Monte Carlo, reversible jumps.

3. *Discriminative methods*: algorithm working on selecting features for discriminating various classes. taught in stat231, e.g. clustering, decision tree, boosting, SVM.

Bayesian View

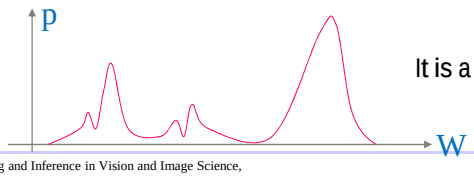
A basic assumption, since Helmholtz (1860), is that biologic and machine vision compute the most probable interpretation(s) from input images.

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 sample from the posterior.

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



It is a sampling problem !

Terminology

In computer vision, the target probability $\pi(x)$ is often defined on a graph representation $G=<V, E>$. We divide G in two types of graph structures, and thus the Markov chains are designed accordingly.

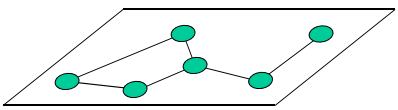
1. Descriptive models on a plat graph where all vertices are semantically at the same level, e.g. various Markov random fields

2. Generative models on a hierarchic And-Or graph with multiple levels of vertices where a high level vertex is divided into various components at the low level. e.g. Markov trees, sparse coding,
   object recognition, *image parsing, etc*

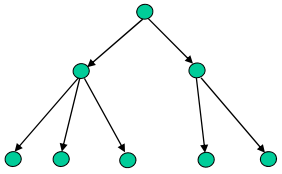
In advanced models, these two structures are integrated because the vertices at each level of a generative model are connected by contextual horizontal links which Represent various relations among the vertices

The terminology

Descriptive or declarative
(Constraint-satisfaction, Markov random fields, Gibbs, Julesz ensemble)



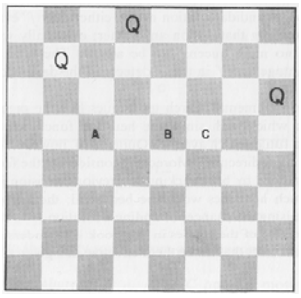
Generative (+ Descriptive)
(hidden Markov, hierarchic model decomposing whole to parts)



Variants of Descriptive
(Causal Markov Models,
Markov chain, Markov tree, DAG etc)

Ex 1: 8-Queen problem

Put 8 queens in a 8 x 8 chess board so that they are safe: i.e. no two queens occupy the same row, column, or diagonal lines.



Inference 1: 8-Queen problem

This is a *constraint-satisfaction* problem on a 8 x 8 grid.

Let's define s be a solution, s could be a binary 8x8 matrix or a list of the coordinates for the 8 queens.

Define the solution in a set:

$$\Omega* = \{\ s :\quad h_i(s) <= 1,\quad i=1,2,\ldots,46\ \}$$

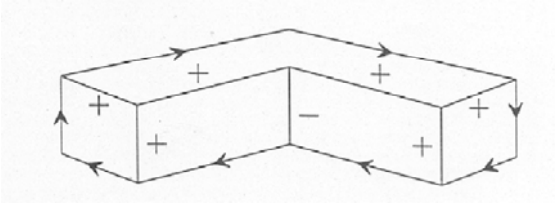
$h_i(s)$ is a hard (logic) constraints respectively for the 8 row, 8 column, 30 diagonal lines.

The computational problem is

$$find \qquad s \in \Omega *$$

Ex 2: Line drawing interpretation

Label the edges of a line drawing (graph) so that they are consistent



This is also *constraint-satisfaction* problem on a graph G=<V,E>.

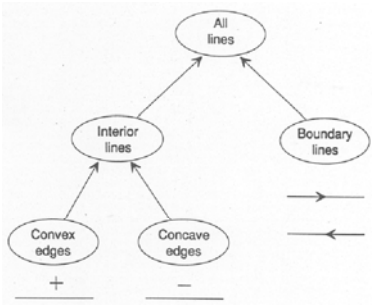
Define the solution in a set:

$$\Omega* = \{\ s : \quad h_i(s)=1, \quad i=1,2,\ldots,|V|\ \}$$

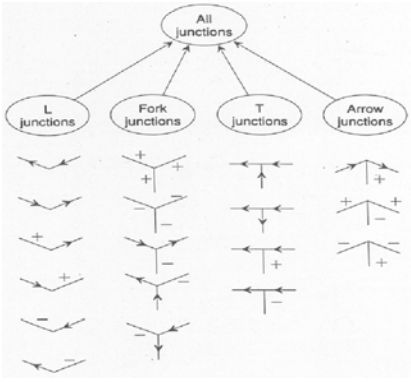
$h_i(s)$ is a hard (logic) constraints respectively for consistence at each vertex.

Ex 2: Line drawing interpretation

allowed edge labels



allowed junction labels



Example: Channel codes

Binary-channel codes can be seen as a set of bits that must satisfy a number of constraints

$$g(x_1, x_2, \ldots x_n) \in \{ X: x_1 \otimes x_3 \otimes x_5 = 0; x_1 \otimes x_2 \otimes x_3 \otimes x_4 = 0 \}$$

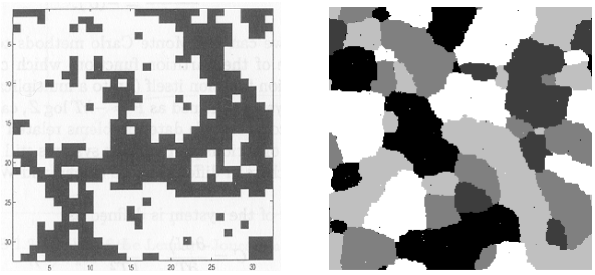
	x_1	x_2	x_3	x_4	x_5
c_1	1	0	1	0	1
c_2	1	1	1	1	0
c_3	0	1	1	1	1

A real example for large scope aerial image understanding (Porway et al 2008)

Original Image | Bottom-Up Proposals | Top-down verified

Ex 4: simulating Gibbs models

Simulating the Ising/Potts models,
("*Simulation*" extends "*relaxation*", "*Gibbs model*" extends "*Constraints*")



This such high dimensional space, the concept of a "solution" is extended to the *typical configurations*, the set of typical configurations is often huge !

Ex 6: Simulating typical protein structures

This such high dimensional space, the concept of a "solution" is extended to the *typical configurations*, the set of typical configurations is often huge !

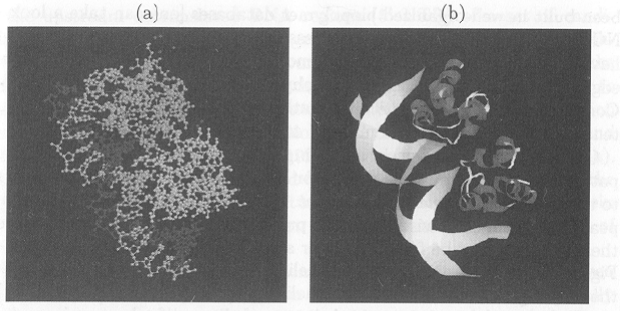
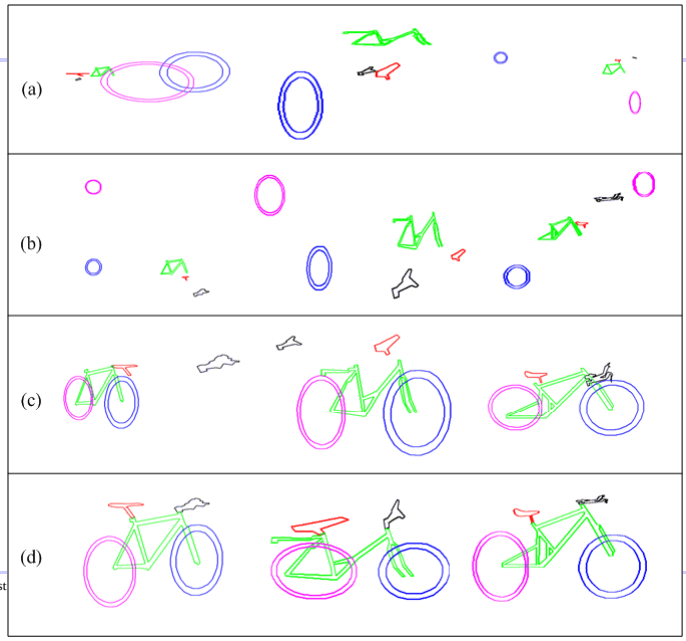


FIGURE 1.4. (a) A ball-and-stick plot of the interaction between a regulatory protein in yeast, 3CRO, and the DNA segment to which it binds. (b) The same structure as in (a), but expressed by a ribbon representation widely used in the protein structure modeling community. [From ref book by Liu]

Simulating a bike model



More examples

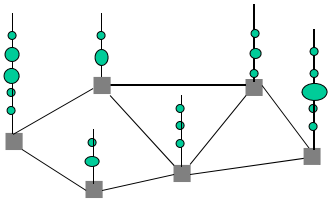
There are many more similar examples, e.g.

image restoration, image segmentation, graph partition/coloring, shape from stereo/motion/shading …

Common properties:

1. A graph representation G=<V, E>.
   G could be directed, undirected, such as chain, tree, DAG, lattice, etc.

2. hard constraints or soft "energy" preference between adjacent vertices.



Descriptive methods: summary

These problems belong to the descriptive family. The computing algorithms includes:

Relaxation-Labeling, Dynamic programming (I consider HMM as descriptive model not generative),
Belief propagation,
Gibbs sampler, Swendsen-Wang, Swendsen-Wang cut.

Issues in algorithm design:

1. Visiting scheme design and message passing.
   which step is more informative, relax more constraints (like line-drawing). In general, the ordering of Gibbs kernels
2. Computing joint solution or marginal belief.
   the marginal believe may be conflicting to each other.
3. Clustering strongly-coupled sub-graphs for effective moves.
   the Swendson-Wang ideas.
4. Computing multiple solutions.

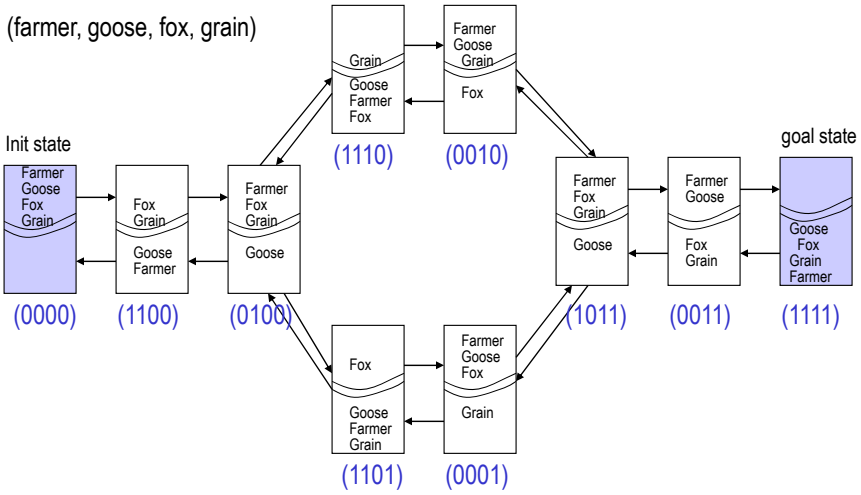
Ex 7: farmer, goose, fox, and grain

A *farmer* wants to move himself, a silver *fox*, a fat *goose*, and some Tasty *grain* across a river. Unfortunately, his *boat* is so tiny he can Take only one of his possessions across on any trip. Worse yet, an Unattended fox will eat a goose, and an unattended goose will eat Grain.

How can he cross the river without losing his possessions?

This can be formulated as finding a path in the state-space graph (next page).
In the coin example, the path is further extended to and-or graph.

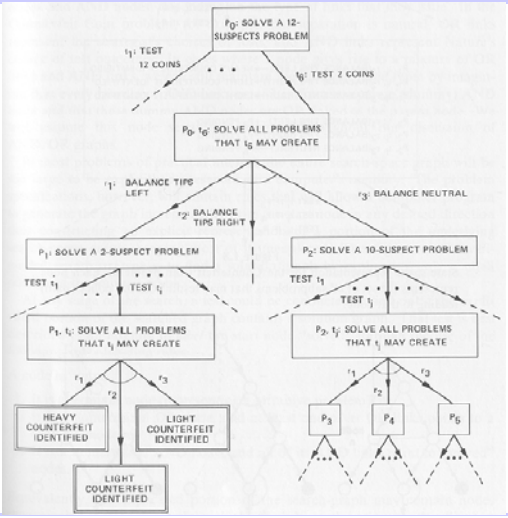
The State Space Graph



Ex 8: 12 Counterfeit coin problem

Given 12 coins, one is known to be heavier or lighter than the others. Find that coin with no more than 3 tests using a two-pan scale.

This generates the And-Or graph representation.



And-Or Graph is also called “hyper-graph”

The and-Or graph represents the decomposition of task into sub-tasks recursively.

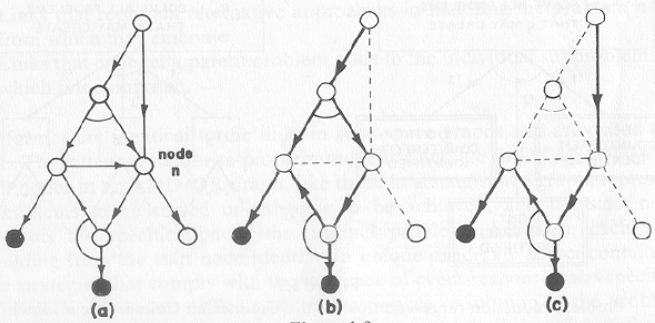
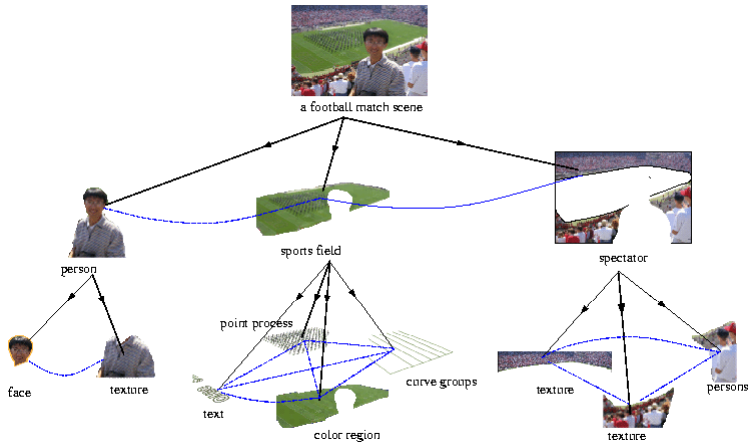


Figure 1.9
An AND/OR graph (a) and two of its solution graphs (b) and (c). Terminal nodes are marked as black dots.

Ex 9: Images parsing

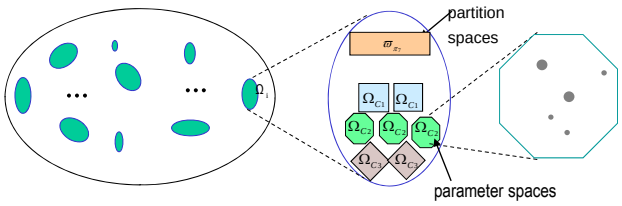
Tu et al 2002-05

Parsing an image into its constituent visual patterns. The parsing graph below is a solution graph with AND-nodes



State space decomposition

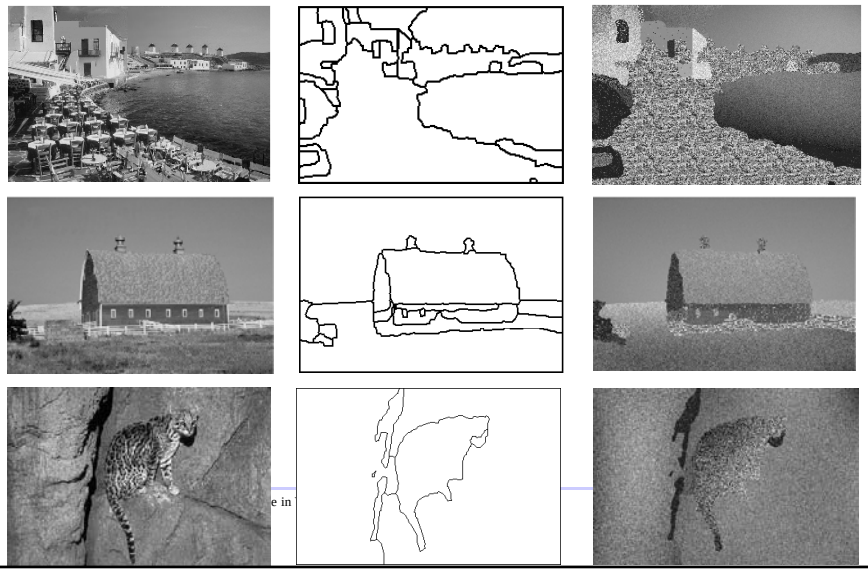
A key concept in vision is composition that complex visual patterns, such as scene, objects are composed of simple elements. This leads to product state spaces. Anatomize the state space is a crucial aspect towards effective algorithm design.



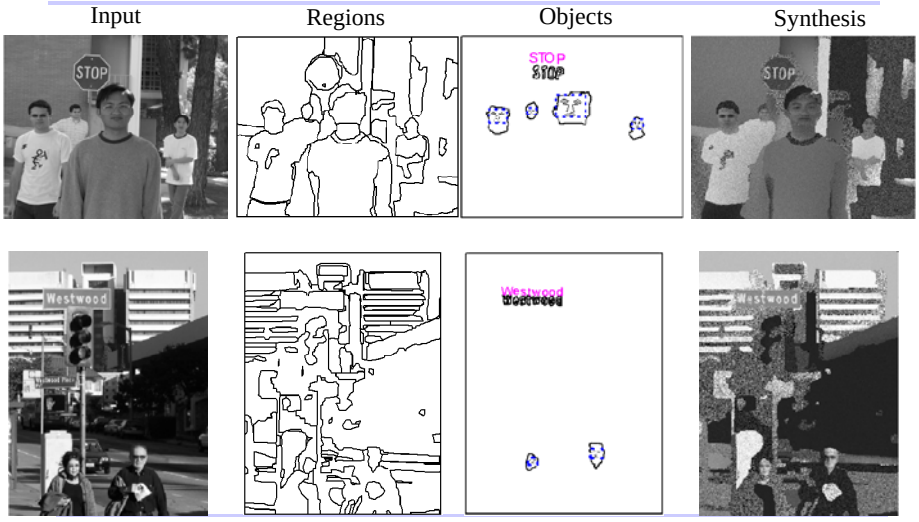
Example: Image segmentation by Data Driven Markov Chain Monte Carlo

(Tu and Zhu, 01)

a. Input image b. segmented regions c. synthesis I ~ *p*(I | W*)



Example: Image Parsing



Tu, Chen, Yuille, and Zhu, iccv2003

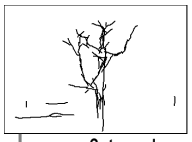
Example: image parsing

输入图像

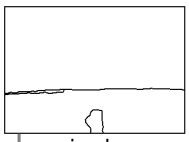
Example: 3D Sketch from a single image

Example II: 3D reconstruction (Han and Zhu, 2003)

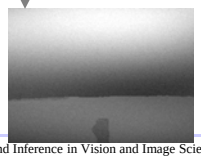
input I



curve & tree layer

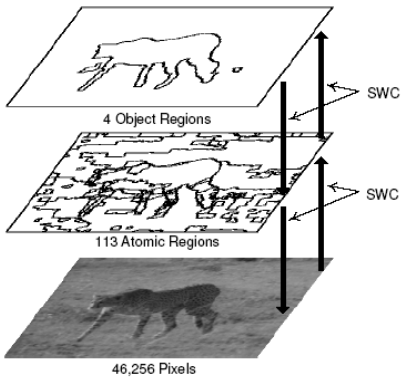


region layer

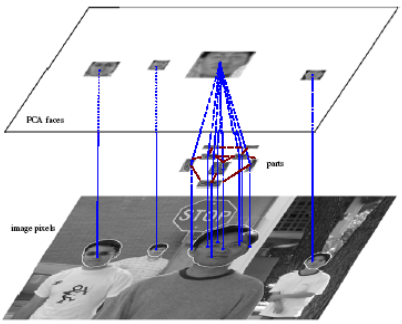


3D reconstruction and rendering

Two Computing Mechanisms

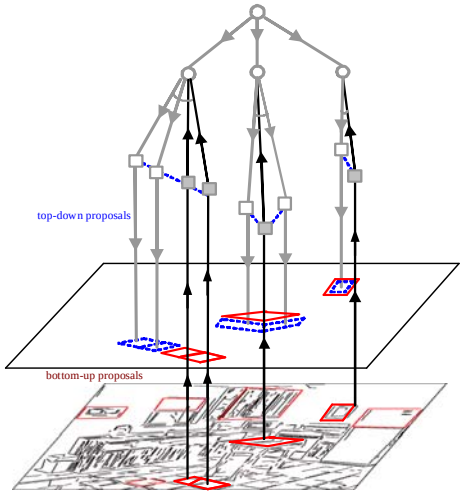


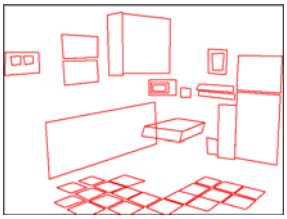
(a) bottom-up graph construction



(b) Top-down graph construction

Top-down and Bottom-up Search





Integrating generative and discriminative methods

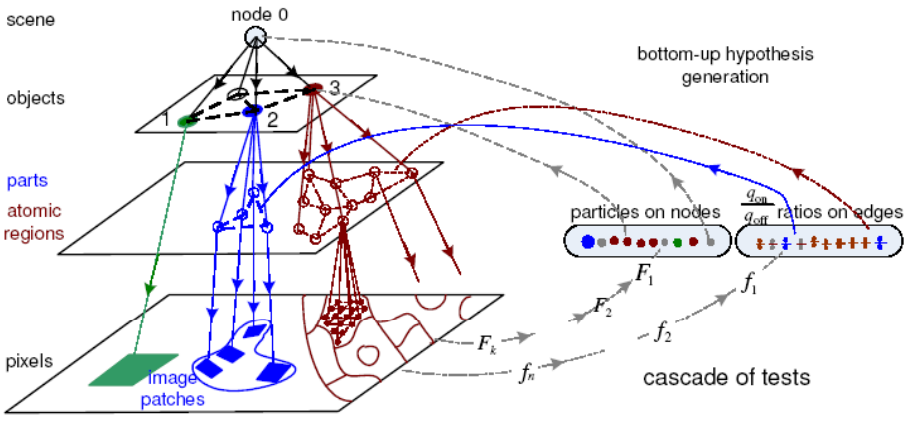
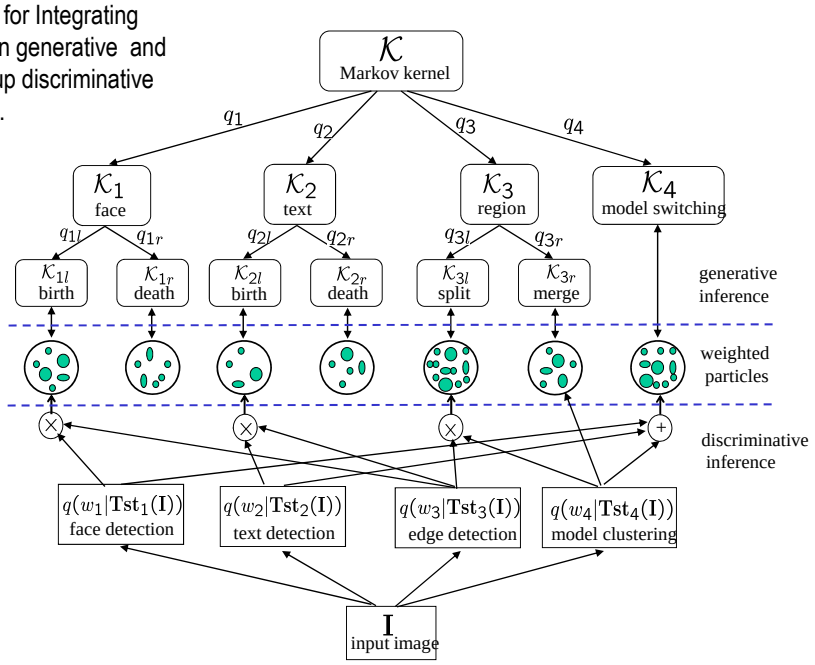


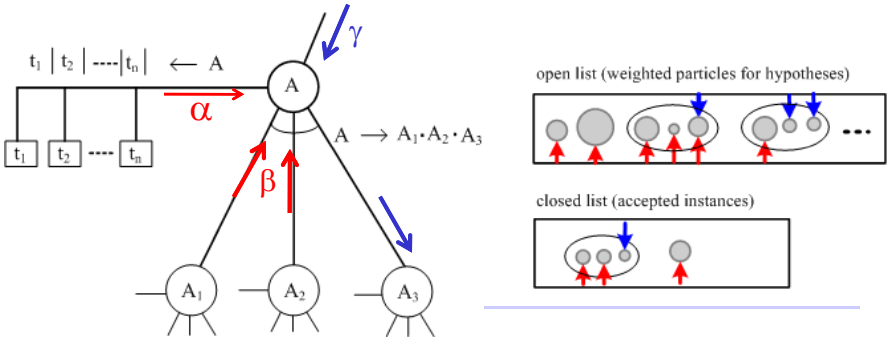
Diagram for Integrating
Top-down generative and
Bottom-up discriminative
Methods.



Recursive computing and parsing

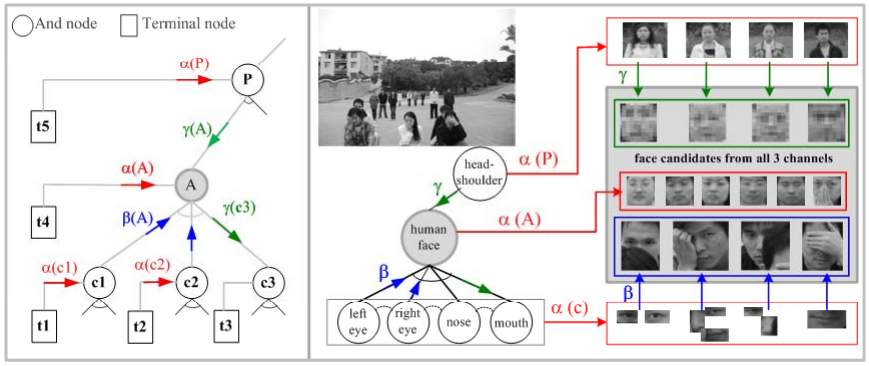
In the And-Or graph --- a recursive structure.
we only need to consider a single node A.

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



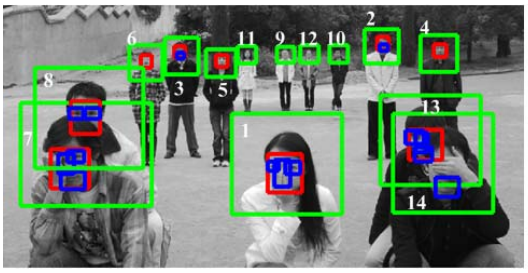
Compositional boosting, T.F. Wu et al, CVPR 07

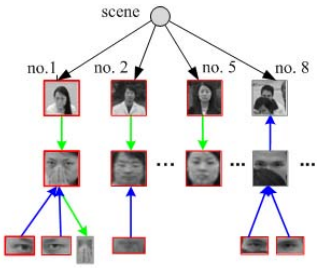
Recursive parsing: the α, β, γ -processes



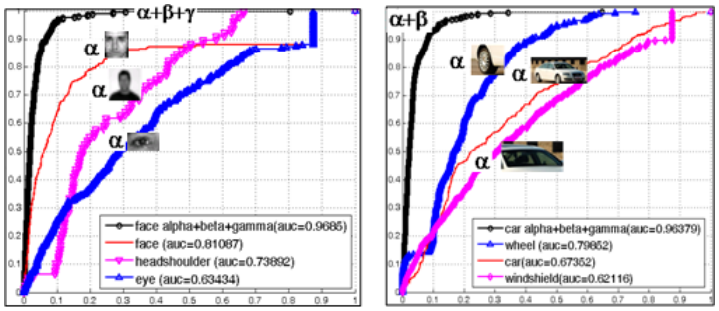
T.F. Wu et al 2009

Ordering the α, β, γ -processes



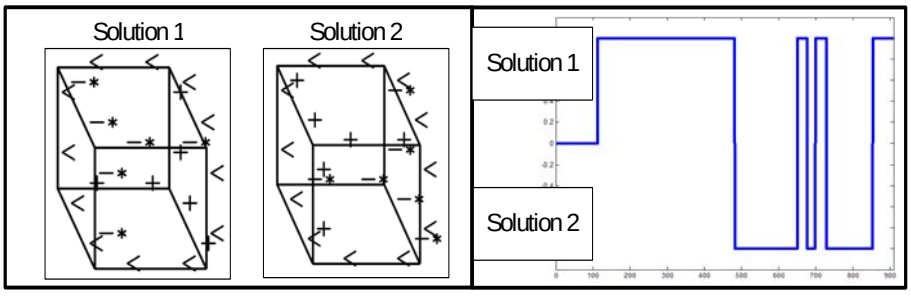


Ordering the α, β, γ -processes



(a) head/shoulder---face---eyes (b) car---parts

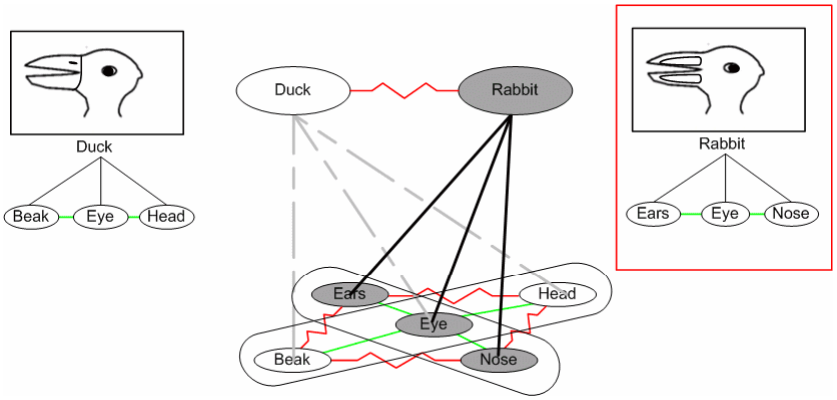
Solving ambiguities: the Necker Cube



From Porway et al 09

Solving ambiguities: the duck-rabbit ambiguity

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.

Motivation for Integration

It turns out that we need to integrate all three methods for complex inference tasks for example, image parsing in computer vision.

1. Construct the parsing graph by generative methods.
   (Heursitic search in AI, grammar parsing, matching pursuit, Markov chain Monte Carlo, reversible jumps)

2. Passing messages in a parsing graph by descriptive methods.
   (relaxation labeling, belief propagation, dynamic programming, MCMC simulation)

3. Making hypotheses and proposals by discriminative methods.
   (clustering, adaboosting, edge detection, RANSAC etc.)

To see the connections between these methods, we need to trace literature in computer science, statistics, computer vision, language understanding etc. Make connections between them. Then pool a global picture.

Various Criteria for Algorithm Design

For simple problems, like 8-queen, we may seek exact solutions, but for very complex problems, like TSP, the exact solution needs a lift-time search, and we have to search for nearly optimal solutions.

Let *A* be an algorithm that take external input I, and I follows a probability f(I).

For example, I could be an input image, an adversary chess player, a city map in a TSP problem etc.
f(I) characterizes the *ensemble* of problems.

Definition 1. *A* is *optimal* if it can always find an exact solution *s* for any I.

$$s \in \Omega^*(\mathrm{I}) \qquad \forall \mathrm{I}$$

Various Criteria for Algorithm Design

Definition 2. *A* is *near optimal* if it can always find a solution *s* within ε-distance to the exact solution for any I. e.g. we often only care about near optimal for TSP.

$$s \in \Omega^*_{\varepsilon}(\mathrm{I}) \qquad \forall \mathrm{I}$$

Definition 3. *A* is *approximate optimal* if it can probably find a solution *s* within ε-distance to the exact solution for any I.

$$p(s \in \Omega^*_{\varepsilon}(\mathrm{I})) > 1-\delta \qquad \forall \mathrm{I}$$

Definition 4. *A* is *approximate optimal for ensemble f(I)* if it can probably find a solution *s* within ε-distance to the exact solution for the ensemble. This relaxes the worst case to average cases.

$$p(s \in \Omega^*_{\varepsilon}(\mathrm{I})) > 1-\delta$$

MCMC as a common framework

To summarize, we have several types of problems, and MCMC is a common framework

1. *Simulation, i.e. draw fair (typical) samples from a pdf.*

$$s \sim p(s), \ s \text{ is a configuration.}$$

2. *Integration/computing in very high dimensions, i.e. to compute*

$$c = \int p(s) f(s) ds$$

3. *Optimization*

$$s^* = \arg\max p(s)$$

4. *Learning with hidden variables*

Many Open Questions about MCMC

1. The necessity of MCMC inference
   Under what condition does a stochastic algorithm beat a deterministic one?
2. Integrating the heuristic search mechanisms in MCMC design.
   MCMC only remember one past state, while AI search opens many solutions.
3. Optimal control strategy, and mechanisms for constructing the solution graphs.
4. Developing a mathematical foundation for the use of heuristics, or we may have to reformulate heuristics.
5. Balancing computational power with costs for each types of moves.
6. The integration of three types of methods: descriptive, generative, discriminative.

A road map for vision