Context Sensitive Graph Grammar and Top-Down/Bottom-up Inference

Song-Chun Zhu

Statistics and Computer Science
University of California, Los Angles

Joint work with four students: Hong Chen, Zijian Xu, Feng Han, Ziqiang Liu

Plan of this talk

Issue 1: Context-sensitive graph grammar to generate "composite graphical templates" for cloth modeling which are reconfigurable Markov random fields.

Issue 2: Bottom-up / top-down inference with graph grammars integrating discriminative and generative models.

Issue 3: Information scaling and entropy rate
What is the continuum spectrum for feature selection?

Issue I: Object Modeling with context-sensitive graph grammar

Motivation

Objects have large within-category variations in configurations

1. Vehicles --- sedan, hunchback, van, truck, SUV, …
2. Clothes --- jacket, T-shirt, sweater, ….
3. Furniture --- desk, chair, dresser, …

Configuration changes more in scene categories

A party scene, a living room, an office, a street scene, …

Each object should be represented by a set of graphical models or a graphical model that is "re-configurable".

Integrating Markov random fields with graph grammars

Objectives:

Not just for classification, but to understand the whole object.
Also for rendering, e.g. human portrait, sketches, and cartoon.

Why graph grammar?

1. They are known to generate a large set of configurations using a small number of primitives and production rules.
2. They can be context sensitive --- an essential property.

Review: grammars and syntactic recognition

A stochastic grammar is often a 5-tuple $\mathcal{G} = \langle V_N, V_T, R, p, \Sigma \rangle$

V_N --- non-terminal nodes,
V_T --- terminal nodes,
R --- production rules,
Σ --- the language (a set of valid sentences)
p --- the probability

$$\Sigma = \{ (w, p(w)) : \ s \xrightarrow{R^*} w, \quad w \in (V_N \cup V_T)^* \}$$

Example of graph grammar from K.S. Fu in 70-80s

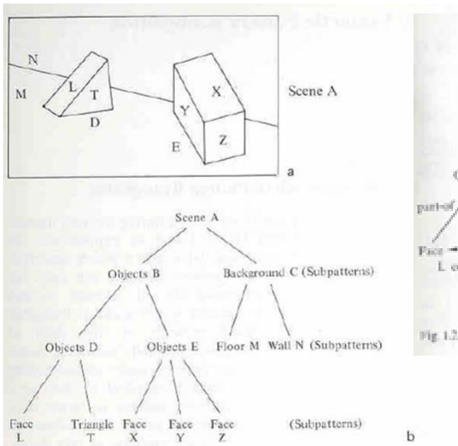


Fig. 1.1a and b. The pictorial pattern *A* and its hierarchical structural descriptions

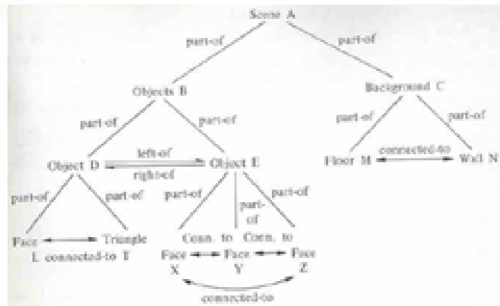


Fig. 1.2. A relational graph of scene *A*

Example of string grammar from K.S. Fu

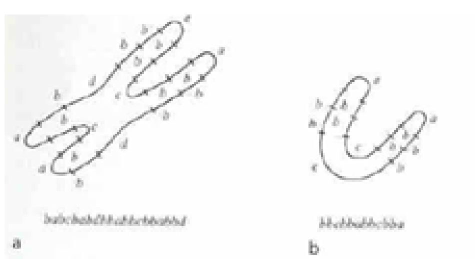


Fig. 1.7. (a) Submedian chromosome; (b) Telocentric chromosome

Examples of diagram grammar

(Rekers and Schurr 96)

Using a production rule to expand the graph.

The shaded vertices are the neighborhood Or environment.

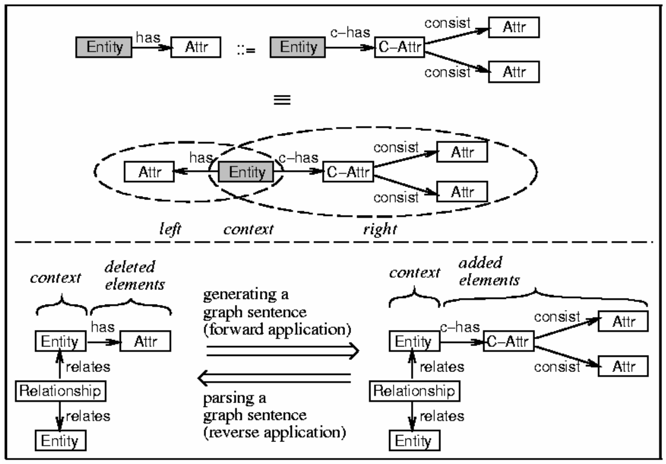


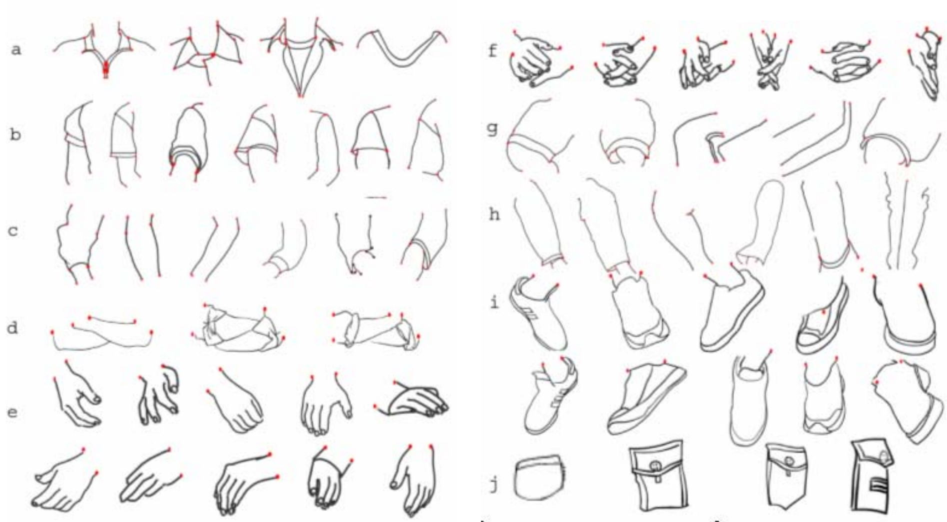
Figure 3: The application of a production

Data processing: decomposing the artist's sketch

(a). input image (b). artist sketch (c). folds/textures (d). structures (e). decomposition

The artist sketch is represented by a 2D attribute graph (like primal sketch Guo,Zhu,Wu iccv03) we decompose the graph into a number of 2D sub-graphs

Each part has a set of sub-templates (graphical sub-configurations)

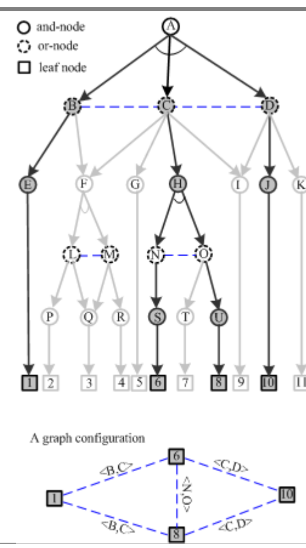


MSRI workshop on object recognition, March 2005, Song-Chun Zhu

An And-Or graph for composing template configurations

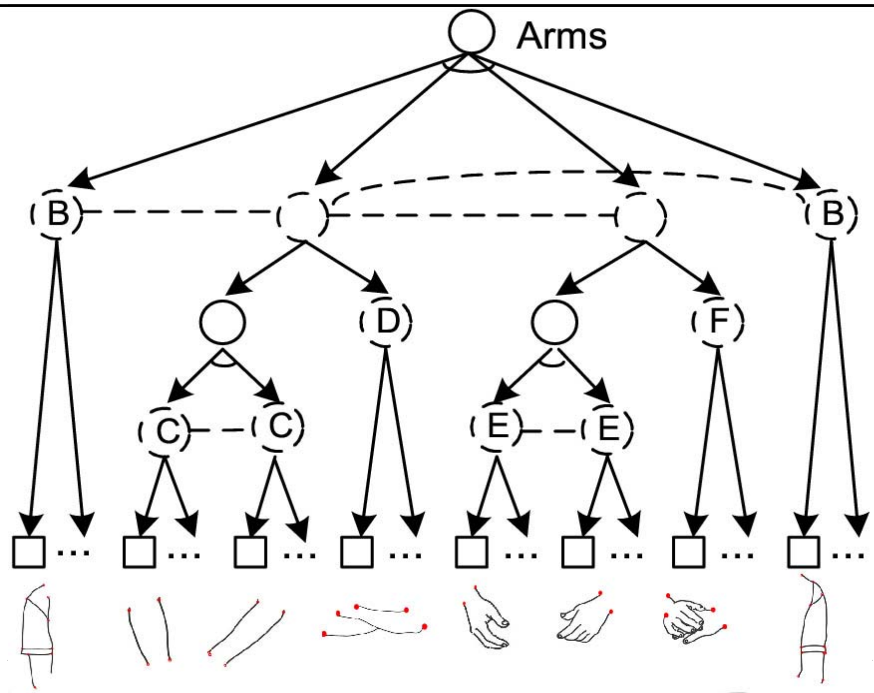
The And-Or graph was used in heuristic AI search (Pearl, 1984).
Like the 12-counterfeit coin problem.

It was not used for modeling, but for problem solving in a divide-and-conquer strategy. Pearl didn't use the horizontal links for context.



Chen, Xu, Liu, and Zhu, 2005

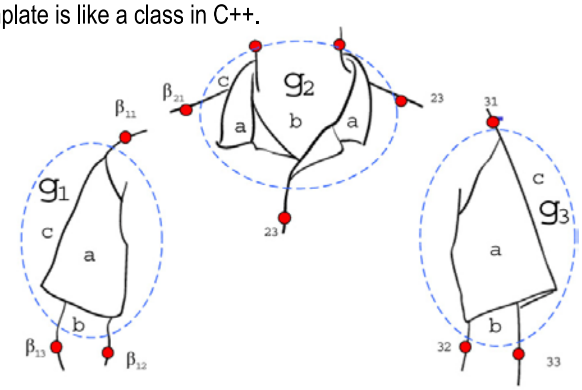
MSRI workshop on object recognition, March 2005, Song-Chun Zhu



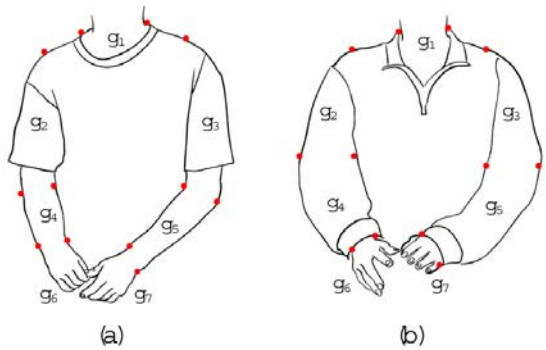
Composing the sub-templates

Each sub-template is a vertex in a composite template, and vertices are connected through “bonds”. The ideas of bonds and address variables were proposed by Grenander in his book and Fridman and Mumford.

Intuitively, each sub-template is like a class in C++.

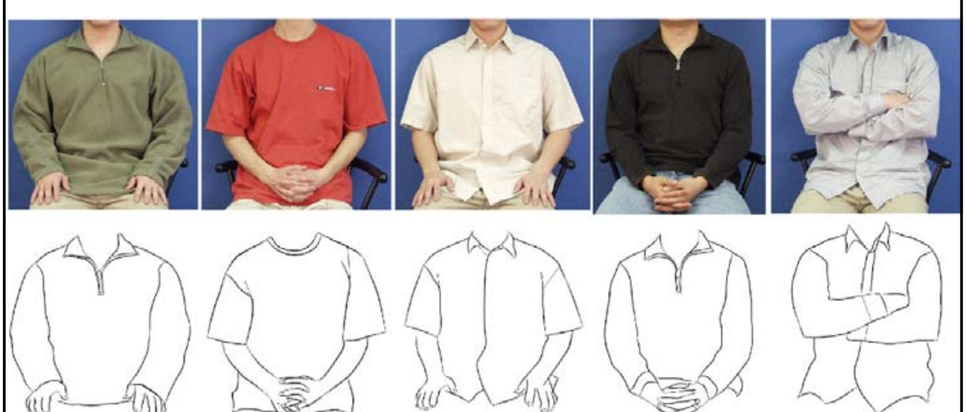


Examples of new configurations (synthesis)



Note that the number of sub-templates and their connections change.
Each is a possible "configuration".

Some results for recognition and sketch



Chen, Xu, Liu, and Zhu, 2005

Formulation

An And-Or graph represents a graph grammar for object class in a 5-tuple

$$\mathcal{G}_{\text{And-Or}} = < T, U \cup V, \Sigma, \mathcal{R}, \mathcal{A} > .$$

$T = \{t_1, ..., t_{m(T)}\}$ is a set of terminal nodes. A node is a subgraph.

$$t_i = (g_i, \text{env}(g_i)), i = 1, 2, ..., m(T)$$

$$g_i = (\{\mathbf{x}_{i1}, ..., \mathbf{x}_{ik(i)}\}, \{f_{mn} = < \mathbf{x}_{im}, \mathbf{x}_{in} >\}, \Lambda_i)$$

$$\text{env}(g_i) = \{\beta_{i1}, ..., \beta_{in(i)}\}$$

Formulation

$U = \{u_1, u_2, ..., u_{m(U)}\}$ is a set of AND-nodes

$$u \rightarrow t \in T; \quad \text{or}$$

$$u \rightarrow (v_1, ..., v_n) :: (r_1, ..., r_k), \ v_i \in V, r_j \in \mathcal{R}.$$

$$g_i(\mathcal{X}(u)) = f_i(\mathcal{X}(v_1), ..., \mathcal{X}(v_n)), \ \ i = 1, 2, ..., K$$

$V = \{v_1, v_2, ..., v_{m(V)}\}$ is a set of OR-nodes

$$v \rightarrow u_1 \,||\, u_2 \cdots ||\, u_n, \qquad u_1, ..., u_n \in U.$$

$$\omega(v) \in \{\emptyset, 1, 2, ..., n\}$$

Formulation

$$\Sigma = \{G_j = (g_{j,1}, ..., g_{j,m(j)}) : j = 1, 2, ..., M\}$$

is a set of valid configurations (Grenander called “global regularity”)

$$\Sigma = \{(1,6,8,10), (1,5,11), (2,4,6,7,9), ...\}$$

$$\mathcal{R} = \{r_{ij} =< v_i, v_j >: v_i, v_j \in V\}$$

$$r_{ij} = \{e_{k,l} = (\beta_{ik}, \beta_{jl}) : \beta_{ik} \in \text{env}(g_i), \beta_{jl} \in \text{env}(g_j)\}.$$

$$\mathcal{A} = \{(\mathcal{A}_i^{\text{pho}}, \mathcal{A}_i^{\text{geo}}) : i = 1, 2, ..., m(T)\}$$

$$\mathcal{A}_i^{\text{geo}} = (A_i, (\xi_{i1}, \eta_{i1}), ..., (\xi_{in(i)}, \eta_{ik(i)}))$$

Probability model

We can put a joint probability (prior) on the composite templates.
We imagine a physical system with dynamic connectivities.

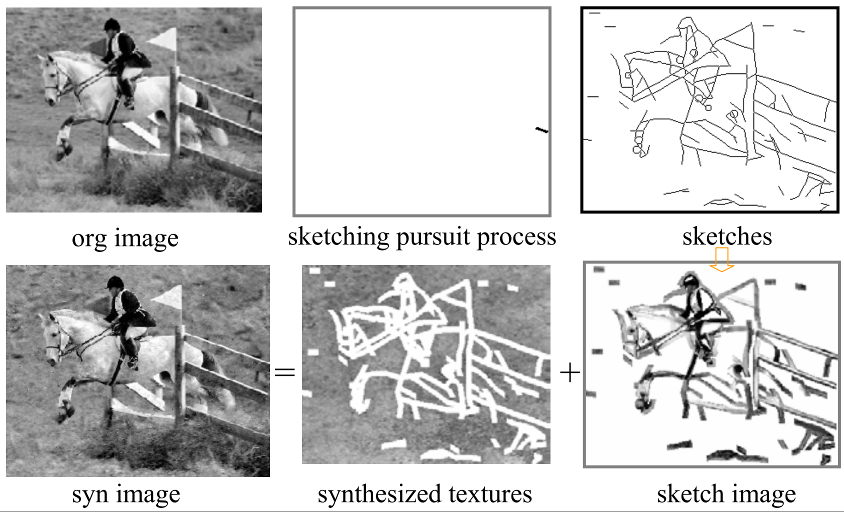
$$G \in \Sigma(\mathcal{G})$$

$$p(G; \mathcal{G}) = \frac{1}{Z(\mathcal{G})} \exp\{-\sum_{v \in V} E_v(\omega(v)) - \sum_{t_i \in T(G)} E(g_i) - \sum_{e_{ik,jl} \in \mathcal{R}(G)} E(\beta_{ik}, \beta_{jl})\}$$

The first term alone stands for a SCFG. The second and third terms are potentials on the graph G.

The likelihood model is the primal sketch model

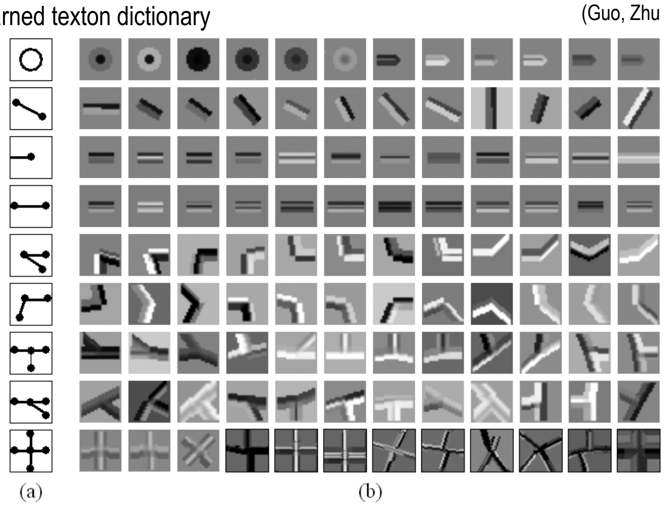
(Guo, Zhu and Wu, 2003)



Simple examples of the image primitive

Learned texton dictionary

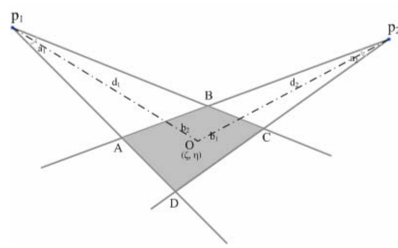
(Guo, Zhu and Wu, 2003)



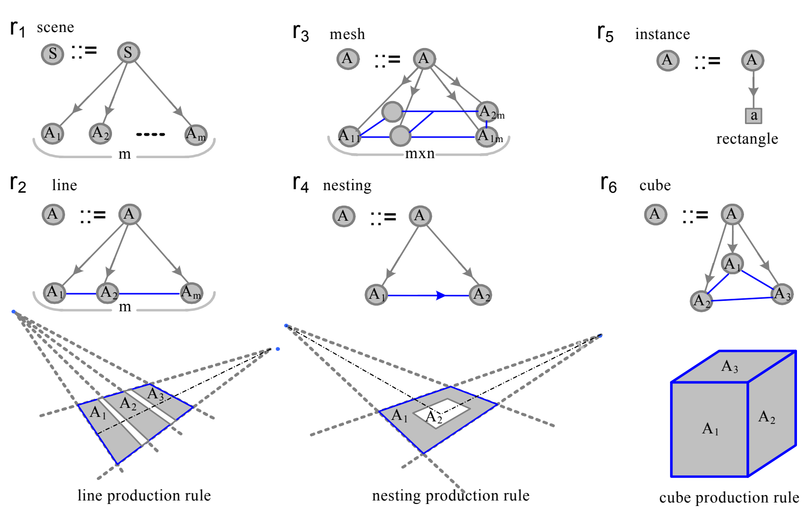
A simpler and more flexible graph grammar

Han and Zhu 2005

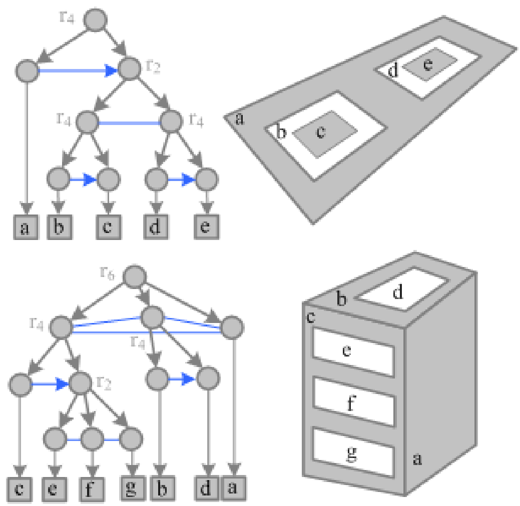
One terminal sub-template
--- a planar rectangle in 3-space



Six grammar rules which can be used recursively

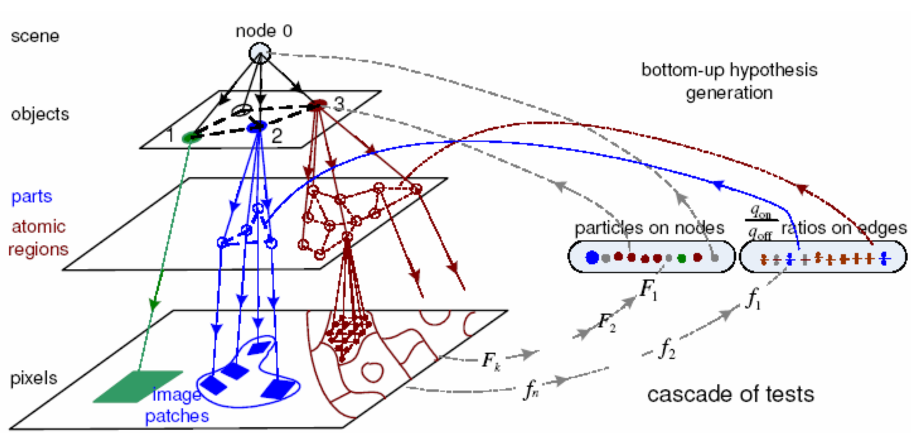


Two configuration examples

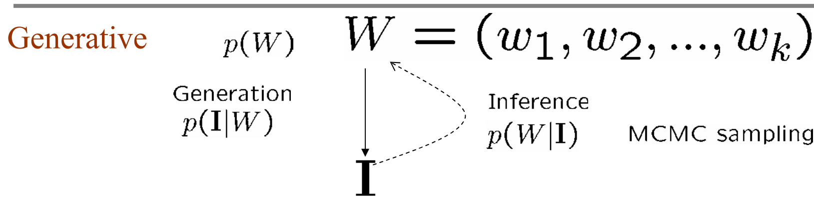


Issue 2: Top-down / Bottom-up Inference

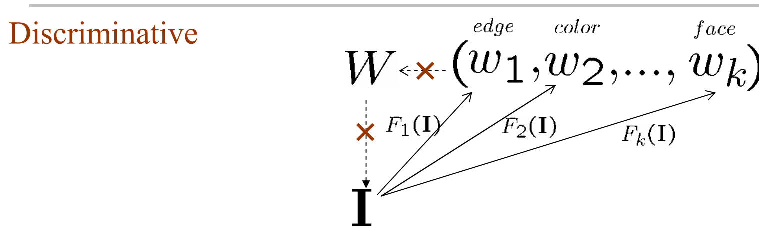
Integrating generative and discriminative methods



Generative vs. Discriminative Algorithms



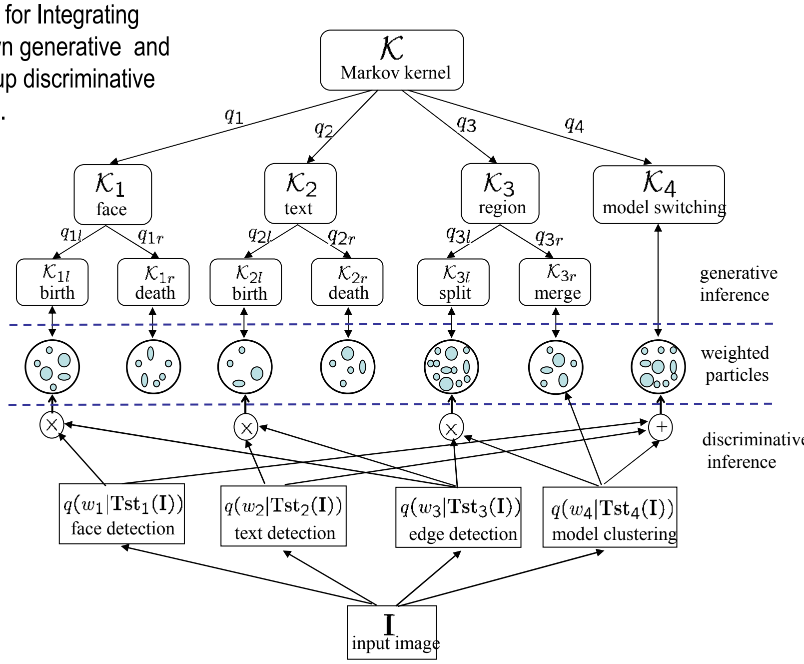
$$W^* = \arg\max\ p(W|\mathbf{I}) = \arg\max\ p(\mathbf{I}|W)p(W)$$



$$q(w_j|F_j(\mathbf{I})) \rightarrow p(w_j|\mathbf{I}),\ j = 1 \ldots k$$

marginal posterior

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



Bottom-up vs. Top-Down:
It is essentially an ordering problem

Both bottom-up tests and top-down kernels can be evaluated by the amount of information gains.

Measuring the power of a discriminative Test

$$\delta(w|F_+) = KL(p(w|\mathbf{I})||q(w|Tst_t(\mathbf{I}))) - KL(p(w|\mathbf{I})||q(w|Tst_t(\mathbf{I}), F_+))$$
$$= MI(w||Tst_t(\mathbf{I}, F_+) - MI(w||Tst_t(\mathbf{I})) = KL(q(w|Tst_t(\mathbf{I}), F_+)||q(w|Tst_t(\mathbf{I})))$$

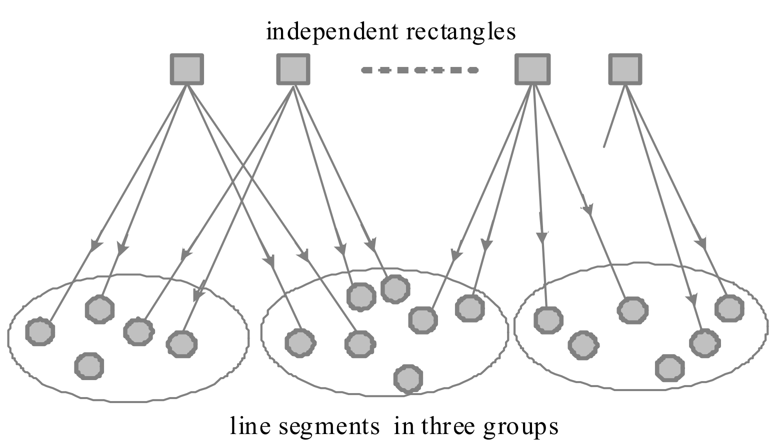
Measuring the power of sub-kernels

$$W_t \sim \mu_t(W) = \nu(W_0) \circ K_{a(1)} \circ K_{a(2)} \circ \cdots \circ K_{a(t)}$$

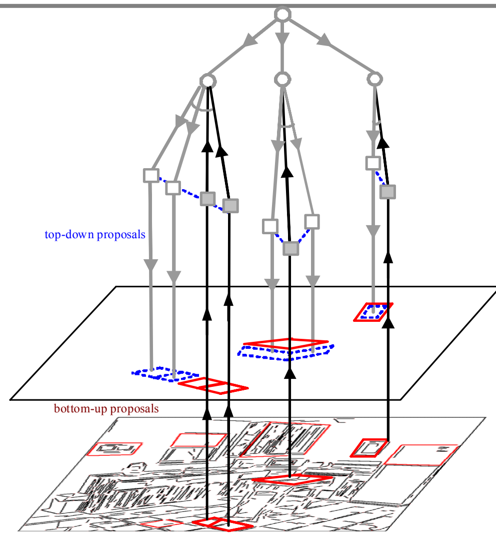
$$\delta_{a(t)} \stackrel{def}{=} KL(p(W|\mathbf{I})||\mu_t(W)) - KL(p(W|\mathbf{I})||\mu_{t+1}(W)) = KL(K_{a(t)}(W_t|W_{t+1})||p_{MC}(W_t|W_{t+1}))$$

Bottom-up detection (proposal) of rectangles

Each rectangle consists of two pairs of line segments that share a vanish point.



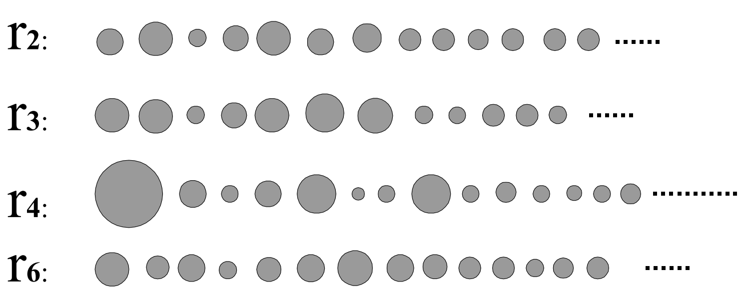
Bottom-up/top-down Inference with grammar



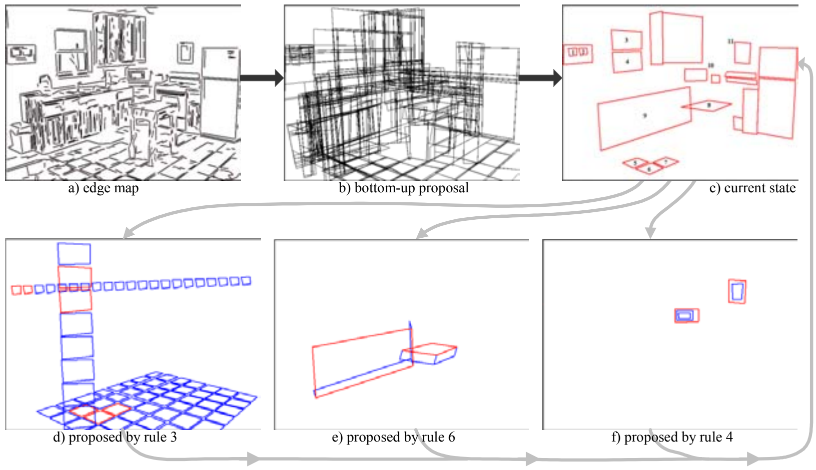
Each grammar rule has a list of particles

A particle is a production rule partially matched, its probability measures an approximated posterior probability ratio.

By analogy, a production rule is like a image base in matching pursuit.



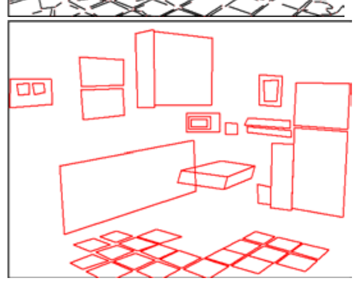
Example of top-down / bottom-up inference



Results

(Han and Zhu, 05)

Edge map



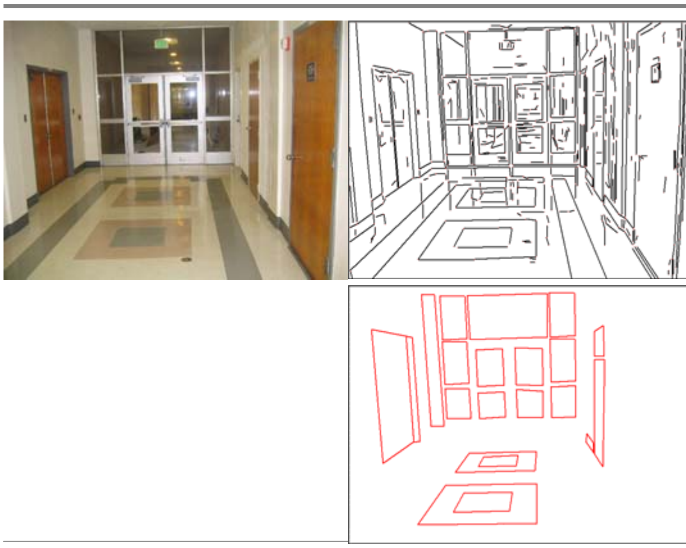
Rectangles inferred

Results

Edge map

Rectangles inferred

Results

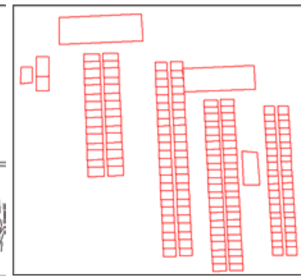


Edge map

Rectangles inferred

Results

Edge map



Rectangles inferred

Summary on issues 1-2

1. Context are modeled intrinsically by Markov random fields
   One can measure the context information quantitatively
   by the minimax entropy learning scheme.

2. Grammars (switches) and address variables (hooks) are ways to
   re-configure the random field structures and to generate
   composite graphical templates.

3. Top-down / bottom-up inference comes naturally with grammars
   The information gains of bottom-up tests and top-down kernels
   should be measured quantitatively. Then algorithm design is
   an ordering problem.