Compositional Models
(Part I: Unsupervised Learning)

A.L. Yuille

Dept. Statistics, UCLA

Plan of Talk

- Part I:
- Compositional Models: Unsupervised Learning.
- Part II:
- Compositional Models: Complexity of Representation and Inference.
- Note: Compositional Models relate closely to Grammatical Models (see tomorrow).

Compositional Models of Objects

- *Compositional Models represent objects in terms of object parts and their spatial relations.*
- These parts are represented recursively in terms of subparts (with spatial relations), and sub-subparts,…
- *Detecting an object also estimates the positions of its parts and subparts automatically.*
- Composition allows explicit part-sharing, yielding big gains in computational efficiency (**2nd part**).
- This talk describes *unsupervised learning algorithms which learn representation of objects*.

Compositional Models: Examples

- Examples: Models of Baseball Players and Horses.
- *Executive Summary*: High-level nodes encode coarse descriptions of object. E.g., centroid position
- Details (e.g., leg positions) are specified by lower-level nodes.

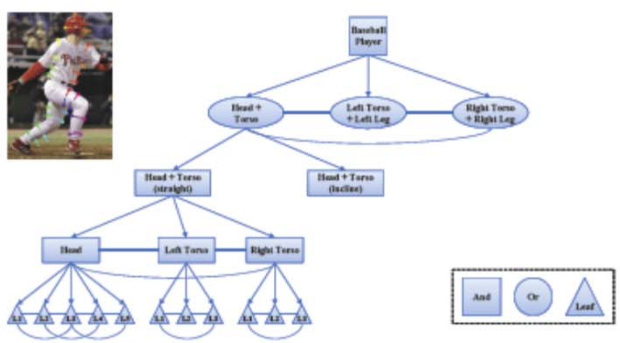
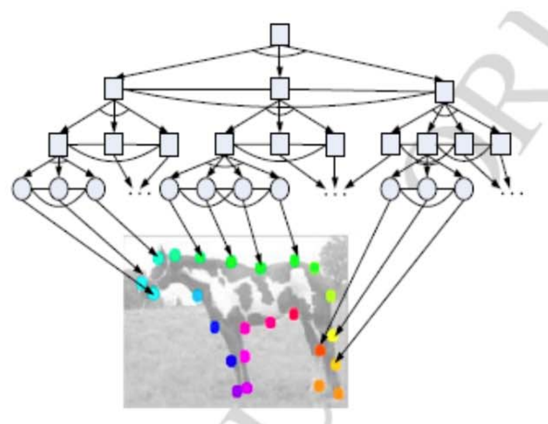
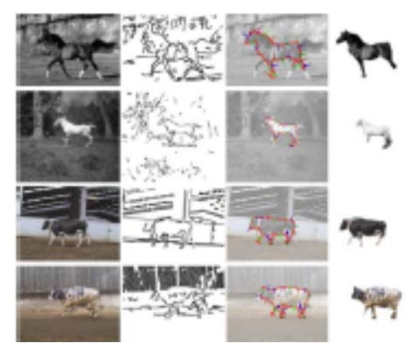


Figure 4. The AND/OR Graph Model (Zhu, Chen, Lin, & Yuille, 2010). The Baseball player is an AND of the head and torso, and left and right legs, but the head is an OR of straight head and torso or an inclined head and torso (top left).





Prior and Related Work

- Prior work on compositional and grammatical models of vision: typically hand-specifies the graphical and grammatical structures of the models. Although the parameters are learnt.
- S. Geman, S. Todorovic, SC Zhu, D.B. Mumford, L. Zhu, A.L. Yuille, P. Felzenzswalb, C. Williams.
- It is desirable to learn the structure of these models automatically.

Advantages of Explicit Representations

Compositionality

Construct models by composing smaller elements.

This enables:

(1). Ability to transfer between contexts and generalize or extrapolate (e.g. , from Cow to Yak).

(2). Ability to reason about the system, intervene, do diagnostics.

(3). Allows the system to answer many different questions based on the same underlying knowledge structure.

"An embodiment of faith that the world is knowable, that one can tease things apart, comprehend them, and mentally recompose them at will." K. Holyoak.

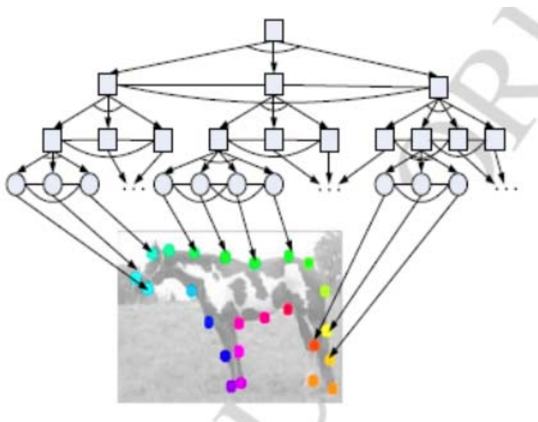
"The world is compositional or God exists". S. Geman.

Mathematics of Compositional Models

Part 2: Compositionality

- Build models for elementary components.
- What is a Compositional Model?
- A probability distribution defined over a graph specified by parent-child relations:

$$\prod_{\nu} \mathrm{P}(\vec{\mathrm{x}}_{ch(\nu)}|\mathrm{x}_{\nu})\mathrm{P}_{\mathrm{p}}(\mathrm{x}_{\mathrm{root}})$$



Key Property: Modularity

- The probability distribution of an object is composed from parts composed of subparts.
- *This enables you to make new distributions – by extracting one part of the object and replacing it by a different parts. Or by altering the parameters of the parts (e.g., making a leg thicker).*
- These changes can be done in a modular manner.
- *More generally, construct a distribution by building it from elementary parent-child components.*
- Modularity enables us to learn the distributions, one parent-child clique at a time.

Parent-Child components: basic building blocks

- Parent-Child determinism:

$$\mathbf{P}(\vec{\mathbf{x}}_{ch(\nu)}|\mathbf{x}_\nu, \boldsymbol{\lambda}_\nu) = \boldsymbol{\delta}(\mathbf{x}_\nu - \mathbf{f}(\vec{\mathbf{x}}_{ch(\nu)}))\mathbf{h}(\vec{\mathbf{x}}_\nu; \boldsymbol{\lambda}_\nu)$$

- f() is a deterministic function
- Executive summary – e.g., parent node encodes average position.
- h() spatial relations between child nodes. Specified by parameter lambda.
- Prior propagation: If object has uniform prior position, then subpart has uniform prior.

$$\mathbf{P}_{\mathbf{p}}(\mathbf{x}_{\nu_i}) = \sum_{\vec{\mathbf{x}}_{ch(\nu)}/\mathbf{x}_{\nu_i}} \boldsymbol{\delta}(\mathbf{x}_\nu - \mathbf{f}(\vec{\mathbf{x}}_{ch(\nu)}))\mathbf{h}(\vec{\mathbf{x}}_{ch(\nu)}; \boldsymbol{\lambda}_\nu)\mathbf{P}_{\mathbf{p}}(\mathbf{x}_\nu)$$

Parent-Child Example:

- Executive summary: parent node take mean position of child nodes.
- Spatial relations between parts are specified by Gaussian distribution on relative positions.

$$(\mathrm{x}_{\nu}, \mathrm{x}_{\nu_1}, \mathrm{x}_{\nu_2}) = (\mathrm{z}_{\nu}, \mathrm{z}_{\nu_1}, \mathrm{z}_{\nu_2}), \ \textbf{spatial position}$$

$$\mathrm{z}_{\nu} = \mathbf{f}(\mathrm{z}_{\nu_1}, \mathrm{z}_{\nu_2}) = \mathbf{1/2}(\mathrm{z}_{\nu_1} + \mathrm{z}_{\nu_2})$$

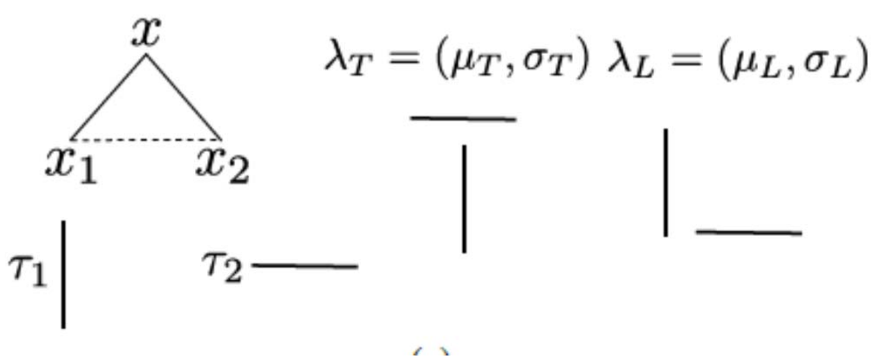
$$\mathrm{h}(\mathrm{z}_{\nu_1}, \mathrm{z}_{\nu_2}; \lambda_{\nu}) = \mathrm{N}(\mathrm{z}_{\nu_1} - \mathrm{z}_{\nu_2}; \mathbf{m}, \sigma), \ \textbf{Gaussian}$$

$$\mathrm{P}_{\mathrm{p}}(\mathrm{z}_{\nu}) = \mathrm{U}(\mathrm{z}_{\nu}), \ \textbf{the uniform distribution}$$

$$\textbf{G translation group}$$

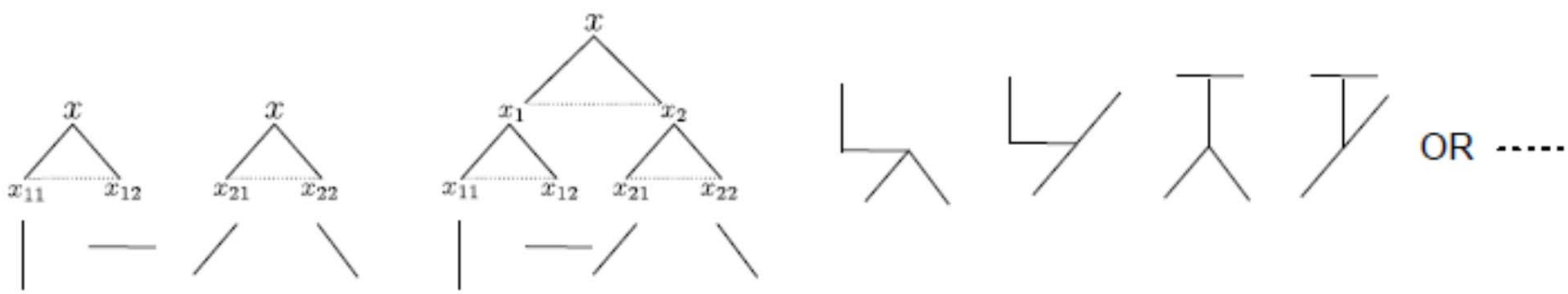
Compositional Models for T and L

- How to make a T or an L?
- *Dictionary* of Level-0 models:
- E.g., horizontal or vertical bars.
- Level-1 model – T or L – is a composition of two Level-0 models plus spatial relations..
- Child nodes: horizontal or vertical bars.



Compositional Learning: dictionaries.

- Start with a dictionary of Level-0 models.
- Learn a Dictionary of Level-1 models by combining models from the Level-0 dictionary.
- Repeat to build Level-2 dictionaries and high-level dictionaries.

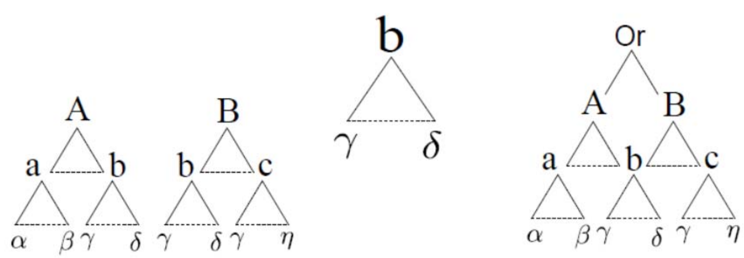


Examples of dictionaries for 120 objects.

- The mean shapes of elements of dictionaries at: Level-0, Level-1, Level-2 Level-3, Level-4.
- Note: the dictionaries are probability distribution, but we only show their mean shapes.

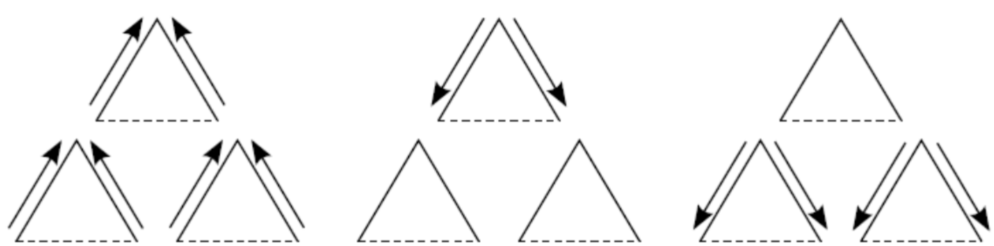
Multiple Objects:

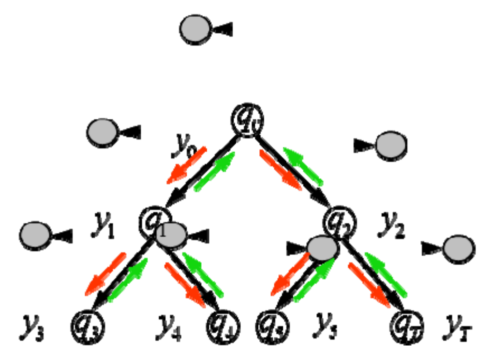
- Multiple objects can be represented in terms of these hierarchical dictionaries.
- This enables part-sharing between objects – dictionary elements used in several objects.
- *Part-Sharing enables efficient learning, representation and inference.* (**2nd part**).



Inference on Compositional Models.

- We perform inference using Dynamic Programming (message passing).
- Bottom-Up propagates local hypotheses to obtain consistent top-level interpretations.
- Top-down disambiguates local hypotheses.





- Discussed in detail in **2nd part**. Inference can be parallelized.

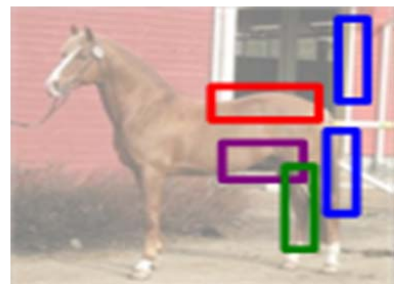
Unsupervised Learning

- Automatically learns a hierarchical set of dictionaries.
- Method: clustering, efficient encoding.
- Theory: parallel search through set of possible generative models of the data.

- ***Number of levels is determined automatically by the algorithm.***

How to Learn Compositional Models?

- Cocktail party problem – object in cluttered background.



- Hard Learning Problems: (unsupervised)
- Do not know the graph structure of the model (e.g., no. of levels)
- (ii) Do not know the assignment of leaf nodes of the model to the data.
- (iii) Do not know the model parameters (lambdas).

Strategy: Exploit Modularity

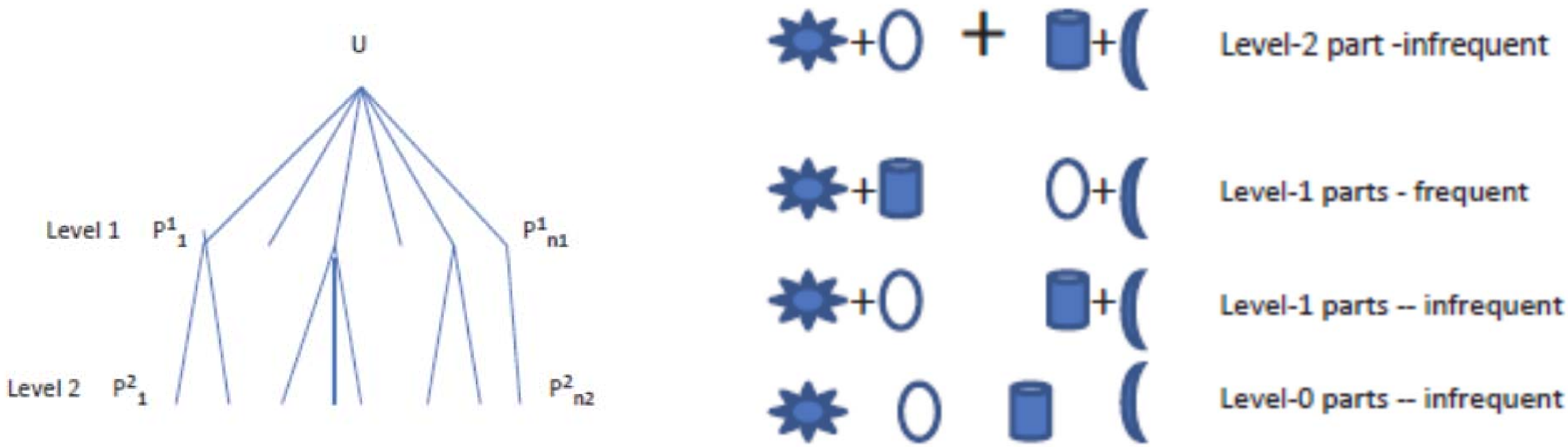
Start by learning the lowest levels of the dictionaries – i.e. the smallest parts.

Learn these dictionary elements separately. Allow for overlap – we can enforce consistency later.

Each dictionary element gives an encoding of the data which is better than the encoding by the root model (uniform distribution).

Proceed level by level. Build new models by composition from models at lower levels. Impose consistency of assignments during composition.

Parallel Search in Model Space

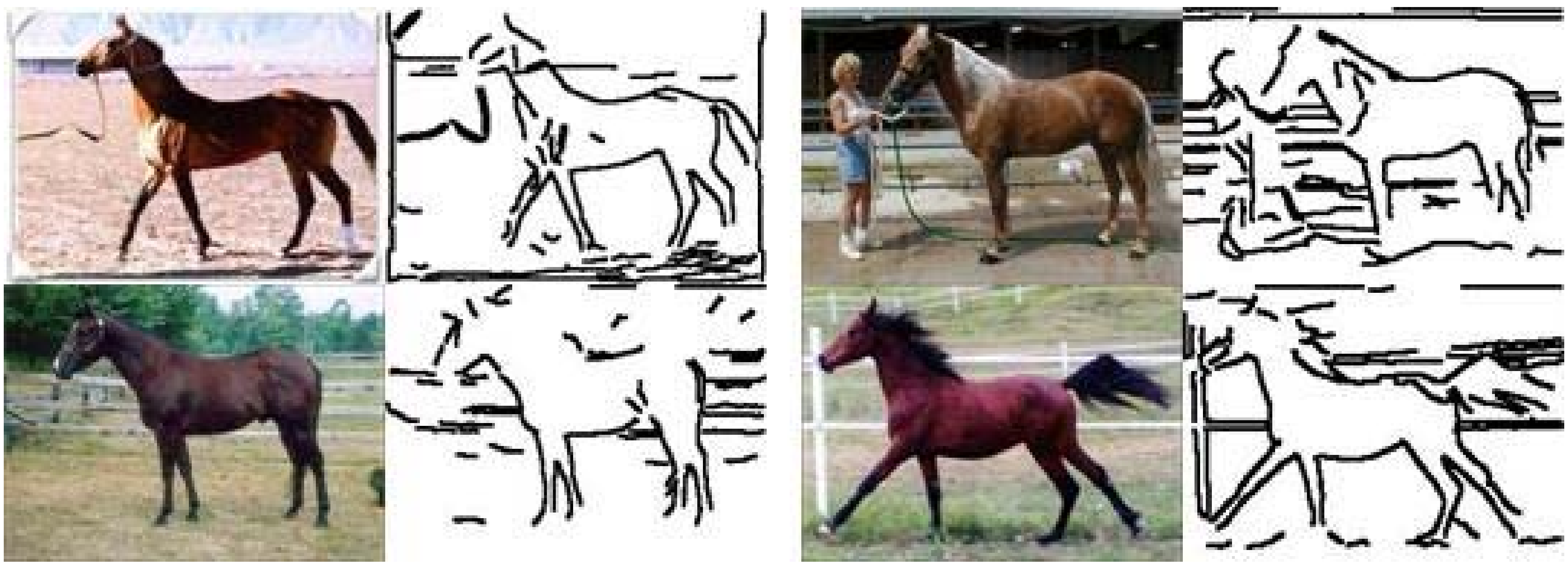


Low-level models may perform poorly by themselves, but may combine well to form good high-level compositions.

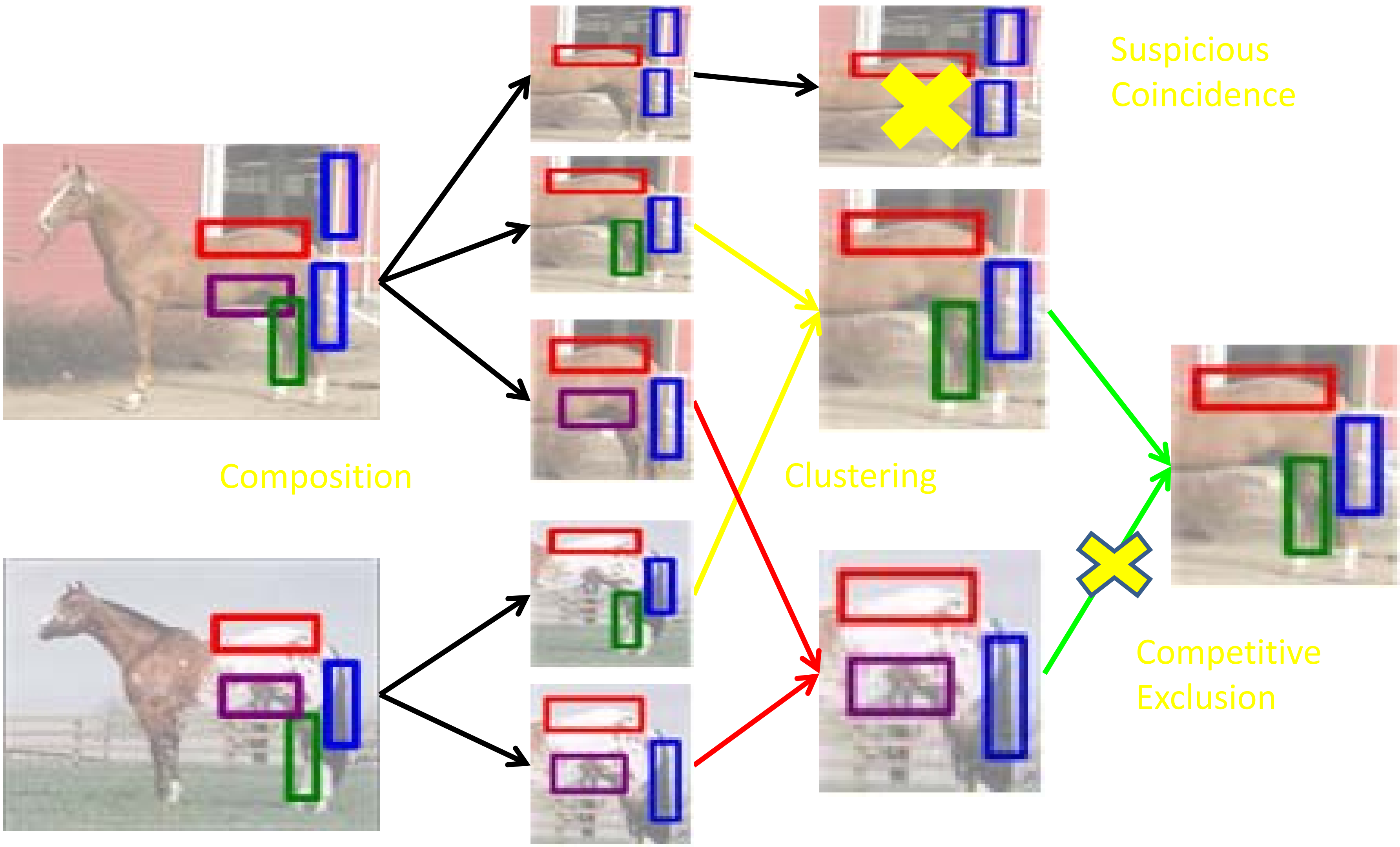
Do not reject weak models too soon.

Horse Dataset: L.Zhu et al. 2008.

- Input Images: Horse Dataset. 10 images used for training. 300 for testing.

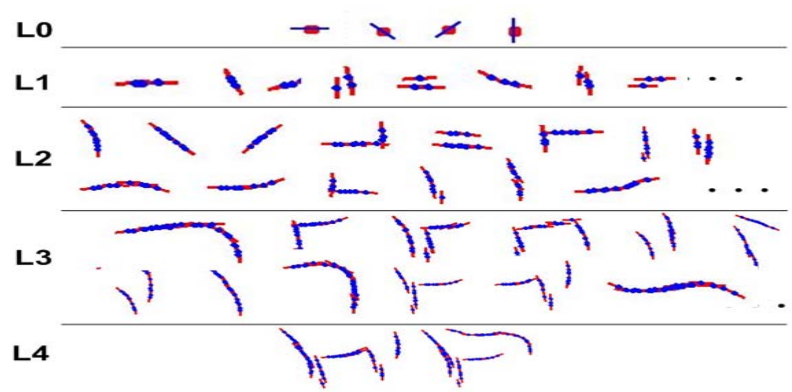


Compositional Learning



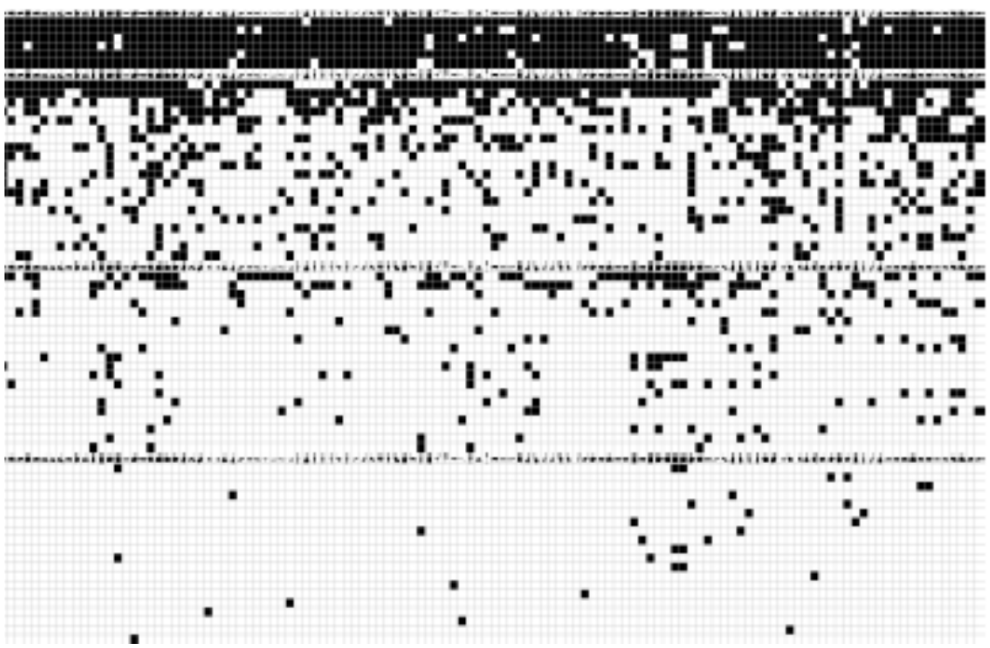
Generic Parts to Object Structures

- As we go up the hierarchy, the dictionaries mimic the features used in low-level, mid-level, and high-level vision.
- *E.g., Mid-level gives 'gestalt rules'*
- High-level is specific to the object. Low-, and mid-level are more generic.



Dictionaries and part sharing.

- Sharing of parts between 120 objects (horizontal)
- Vertical – Level-1, :Level-2,..
- Part sharing is very frequent at low levels. But less sharing at higher levels.



Brief Mathematical Descriptions

- The input to computational learning are a set of images. We assume a set M_0 of level-0 dictionary models, which are pre-specified – e.g., edge detectors.
- For each image, we obtain a set of points with their corresponding types: $(x_i; \ tau(x_i))$.
- The type – tau – indicates the element of the level-0 dictionary (e.g., horizontal or vertical bar).

Brief Mathematics: Better encoding.

To create the level-1 dictionaries we cluster sets of r points from $\{(x_i, \tau(x_i))\}$ which have fixed $\vec{\tau} = (\tau_1, ..., \tau_r)$ to find frequently occurring spatial relations (e.g., the spatial relations between the horizontal and vertical bars for the T and L). Hence we search for examples $\{(x_1^\mu, \tau_1), ..., (x_r^\mu, \tau_r) : \mu = 1, .., n\}$ and parameters λ such that:

$$\log \frac{P(\vec{x}^\mu, \vec{\hat{x}}^\mu, \vec{\tau}, \lambda))}{\prod_{i=1}^r P_D(x_i^\mu, \tau_i))} > K_1, \ \forall \mu \tag{8}$$

where $\hat{x}^\mu$ is the optimal estimate of the parent node – i.e. $\hat{x}^\mu = \arg\max_x P(\vec{x}^\mu, \vec{\tau} | \hat{x}^\mu, \vec{\tau}, \lambda)$ – $P_D(x_i)$ is a default distribution for the positions of the points (e.g., the uniform distribution), and K_1 is a threshold.

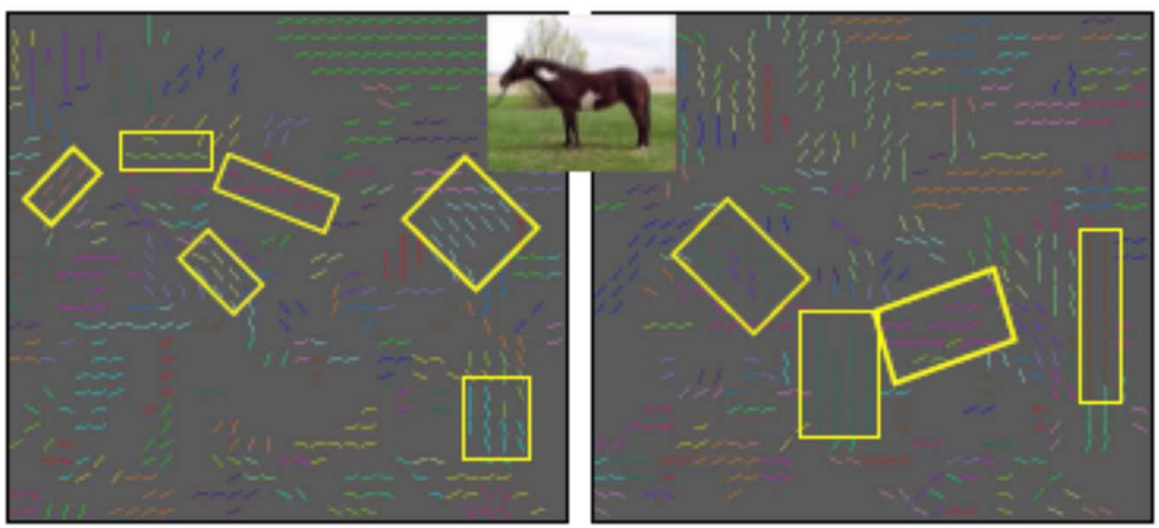
We take the local maxima over the value λ to obtain a level-1 dictionary $\mathcal{M}_1$. Each dictionary element is indexed by its type $\tau^1 = (\vec{\tau}, \lambda^1)$, where $\vec{\tau}$ are the types of the r children, and λ_1 parameterizes the spatial relations. We do *not* impose consistency so a pair (x, τ) can be used in many different clusters. This lack of consistency is desirable because we do not want to make premature decisions. It gives an over-complete representation of the image in terms of level-1 models. Note that this equation is similar to thresholding the local evidence for a part, with the main difference being the lack of the data terms $\log \frac{P(x|\tau(x))}{P(I(x)|\tau_0)}$.

What Inputs to Use?

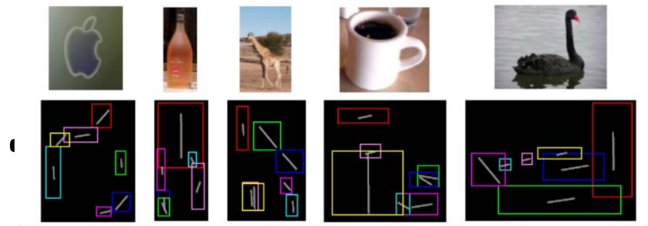
- The work described above uses edges as inputs. But alternative features can be used.
- For example, we can use HOG-Bundles (Mottaghi and Yuille 2011). These are built from HOG features by local spatial grouping.
- Note: edges have disadvantages because there are many of them and have similar properties. HOG-bundles are fewer and easier to differentiate.

HOG-Bundles

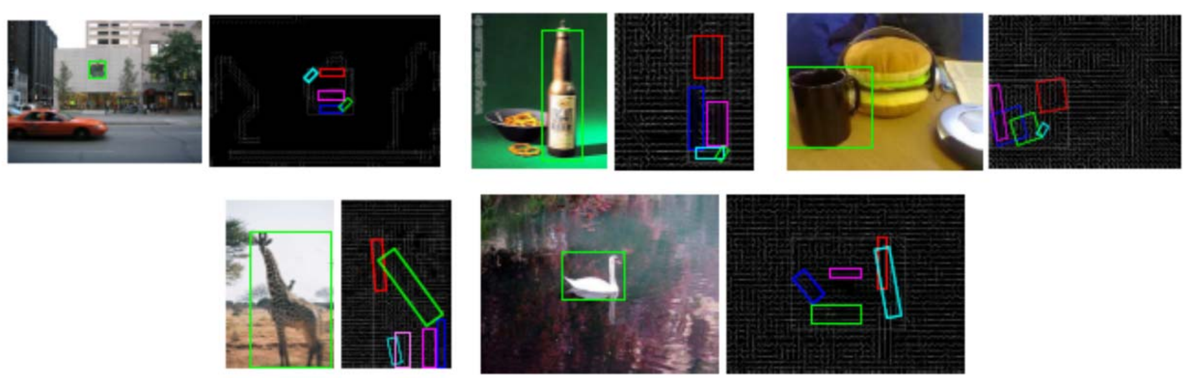
- Start with HoG-bundle representation of images.
- Hog-bundles: HOGs detect edges – HOG-bundles group by proximity and collinearity.
- HOG bundles often correspond (roughly) to parts of object.



ETHZ dataset

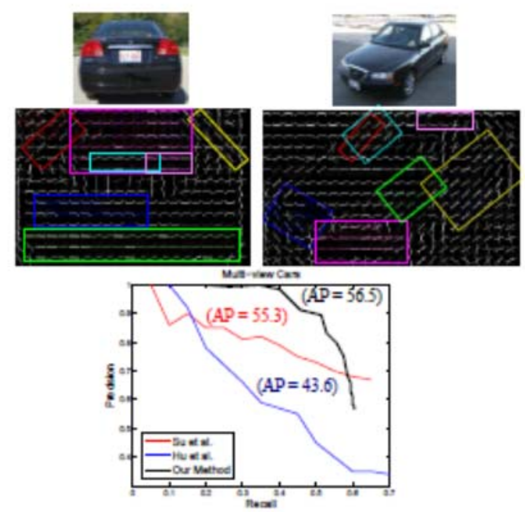


- Learnt models for each category of the ETHZ dataset. Rectangles represent the HOG bundles.
- No. of parts and relative position/orientation is learnt automatically.



Multiview Car Dataset

- Learns models for different viewpoints (automatically). Test on Car dataset (Su, Sun, Fei Fei, Savarase 2009).
- Performance was best – expect for methods with explicit 3D car models.



Any Relation to Neurons?

- There are some interesting relations to work by L. Valiant in Circuits of the Mind.
- Valiant studies how sets of “neuroids” could automatically store memories of conjunctions,.
- His more recent work considers memorizing conjunctions of conjunctions – analogous to higher level compositions.
- His interest was in Random Access Memory models. But the same ideas could be used for compositional models.

Summary of Part 1.

- Compositional Models represent objects explicitly in terms of parts, subparts, and spatial relations.
- This explicitness enables diagnostics and transfer.
- Unsupervised learning – learns dictionaries bottom-up exploiting modularity.
- Part-sharing – makes learning efficient.
- Efficiency of Inference and representation and parallel implementation (**2nd talk**).
- But will they work on Pascal or ImageNet?

References:

- S. Geman et al.. Composition Systems. Quarterly of Applied Mathematics, 60. 2002.
- S.C. Zhu and D.M. Mumford. A Stochastic Grammar of Images. 2006.
- D.M. Mumford and A. Desolneux. Pattern Theory. 2010
- L. Zhu et al. Unsupervised Structure Learning. ECCV. 2008.
- L. Zhu et al. Part and Appearance Sharing. CVPR 2010.
- R. Mottaghi and A.L. Yuille. A compositional

Compositional Models
Part II: Complexity of Representation and Inference

A.L. Yuille UCLA

"Compositional Models and the Fundamental Problem of Vision"?

Hierarchical Models

- One of the hopes, and expectations, of hierarchical models is that they can represent complex structures in terms of compositions of elementary components – *shared parts.*
- This should yield big gains in the complexity of representation and inference.
- *But how can we analyze and quantify this?*

A Fundamental Problem of Vision

- **Complexity:**
- Set of images is almost infinite (Kersten 1987).
- No. of objects is big 30,000 (Biederman 1984).

- *But the human brain can detect objects and understand scenes within 150 msecs.*
- And we want computer vision systems to do the same.

The Fundamental Problem

- This lecture explores this fundamental problem from the perspective of compositional models.
- *Quantify the gains of part sharing a*nd *executive summary. (Recall objects have a hierarchical distributed representation).*
- *(I):* We analyze compositional models and show they can yield exponential gains in efficiency.
- .(II) We perform a similar analysis for a novel parallel implementation of compositional models.
- *(III) Speculations about the Visual Cortex.*

Compositional Models:

- Examples: Graphical Models for Horses and Players.
- *Executive Summary*: High-level nodes encode coarse descriptions of object. E.g. centroid position
- Details (e.g. leg positions) are specified by lower-level nodes.

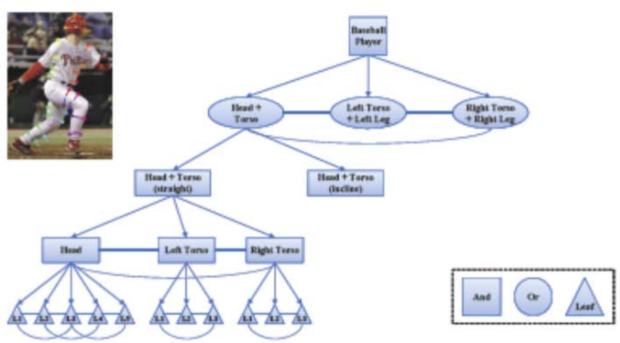
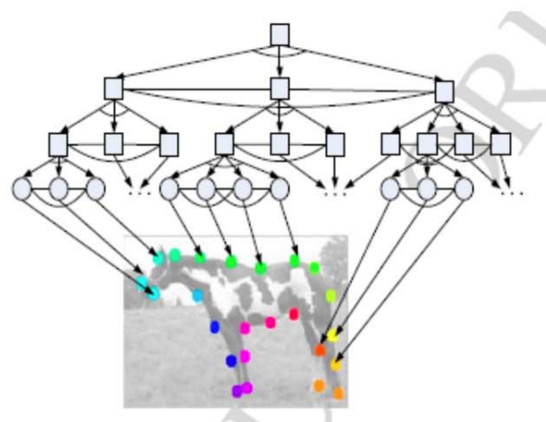
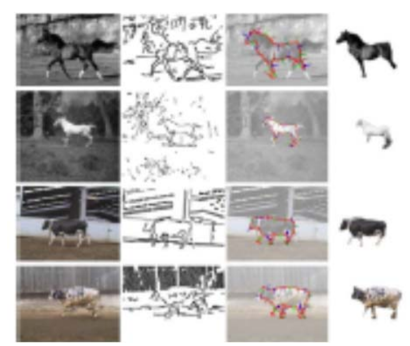


Figure 4. The AND/OR Graph Model (Zhu, Chen, Lin, & Yuille, 2010). The Baseball player is an AND of the head and torso, and left and right legs, but the head is an OR of straight head and torso or an inclined head and torso (top left).





Compositional Model of a Single Object

- Each Object is represented by a graphical model.
- Generative for positions of parts.

$$P(\vec{x}) = P(x_H) \prod_{\nu} P(\vec{x}_{ch(\nu)} | x_\nu; \tau_\nu),$$

- Basic Building Block: Child-Parent Models:

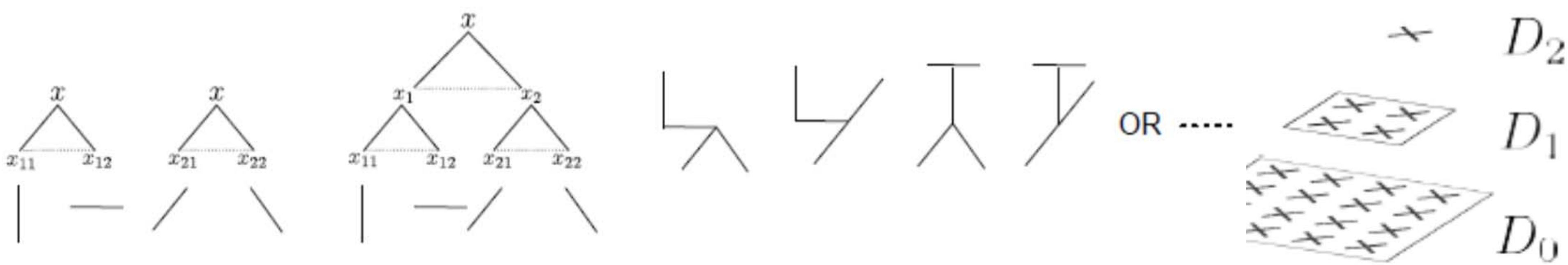
$$\mathbf{P}(\vec{\mathbf{x}}_{ch(\nu)} | \mathbf{x}_\nu, \boldsymbol{\lambda}_\nu) = \boldsymbol{\delta}(\mathbf{x}_\nu - \mathbf{f}(\vec{\mathbf{x}}_{ch(\nu)}))\mathbf{h}(\vec{\mathbf{x}}_\nu; \boldsymbol{\lambda}_\nu)$$

- Generative model for data.

$$P(\mathbf{I} | \{x_l : l \in \mathcal{L}\}) = \prod_{x \in \{x_l\}} P(I(x) | \tau(x)) \times \prod_{x \notin \{x_l\}} P(I(x) | \tau_0),$$

Examples

- Left: T's, L's, and their compositions.
- Right: *Executive summary* – quantified by a *Spatial decay factor q* – lower resolution needed for higher-levels of the hierarchy.

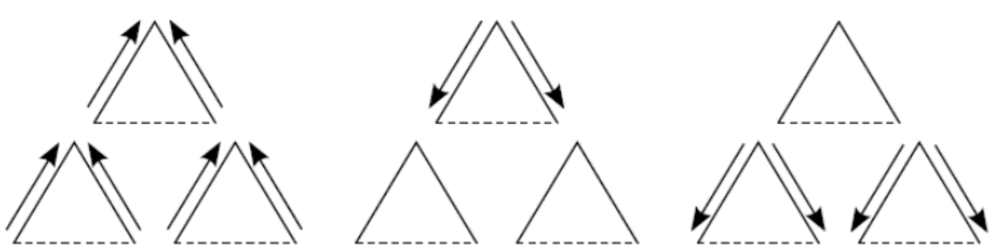


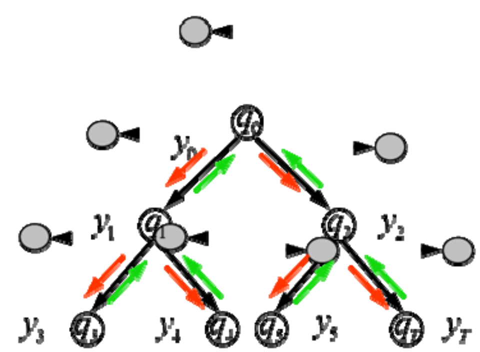
(a) (b)

Inference for a Single Object

- For each object, we can perform inference using Dynamic Programming (message passing):
- Bottom-Up and Top-Down pass (c.f. inside/outside algorithm).

$$\vec{x}^* = \arg\max_{\vec{x}} \{ \sum_{x \in \mathcal{L}} \log \frac{P(I(x)|\tau(x))}{P(I(x)|\tau_0)} + \sum_{\nu} \log P(\vec{x}_{Ch(\nu)}|x_\nu; \tau_\nu) + \log U(x_{\mathcal{H}}) \}.$$





Compositional Inference: Bottom-Up

- DP Example: Level-2 state: $\vec{x} = (x, x_1, x_2, x_{11}, x_{12}, x_{21}, x_{22})$.
- Inference Task is to maximize:

$$\log P(x_1, x_2|x) + \log P(x_{11}, x_{12}|x_1) + \log P(x_{21}, x_{22}|x_2)$$
$$+ \log \frac{P(I(x_{11})|\tau(x_{11}))}{P(I(x_{11})|\tau_0)} + \log \frac{P(I(x_{12})|\tau(x_{12}))}{P(I(x_{12})|\tau_0)} + \log \frac{P(I(x_{21})|\tau(x_{21}))}{P(I(x_{21})|\tau_0)} + \frac{\log P(I(x_{22})|\tau(x_{22}))}{P(I(x_{22})|\tau_0)}.$$

- DP: bottom-up (first step) Computes set

$\{x_1, \phi(x_1)\}$ and $\{x_2, \phi(x_2)\}$

- By

$$\phi(x_1) = \max_{x_{11}, x_{12}} \{\log P(x_{11}, x_{12}|x_1) + \log \frac{P(I(x_{11})|\tau(x_{11}))}{P(x_{11}|\tau_0)} + \log \frac{P(I(x_{12})|\tau(x_{12}))}{P(I(x_{12})|\tau_0)}$$
$$\phi(x_2) = \max_{x_{21}, x_{22}} \{\log P(x_{21}, x_{22}|x_2) + \log \frac{P(I(x_{21})|\tau(x_{21}))}{P(I(x_{12})|\tau_0)} + \log \frac{P(I(x_{22})|\tau(x_{22}))}{P(I(x_{22})|\tau_0)}$$

- Repeat: $\phi(x) = \max_{x_1, x_2} \{\log P(x_1, x_2|x) + \phi_1(x_1) + \phi_2(x_2)\}$

Compositional Inference: Top-Down

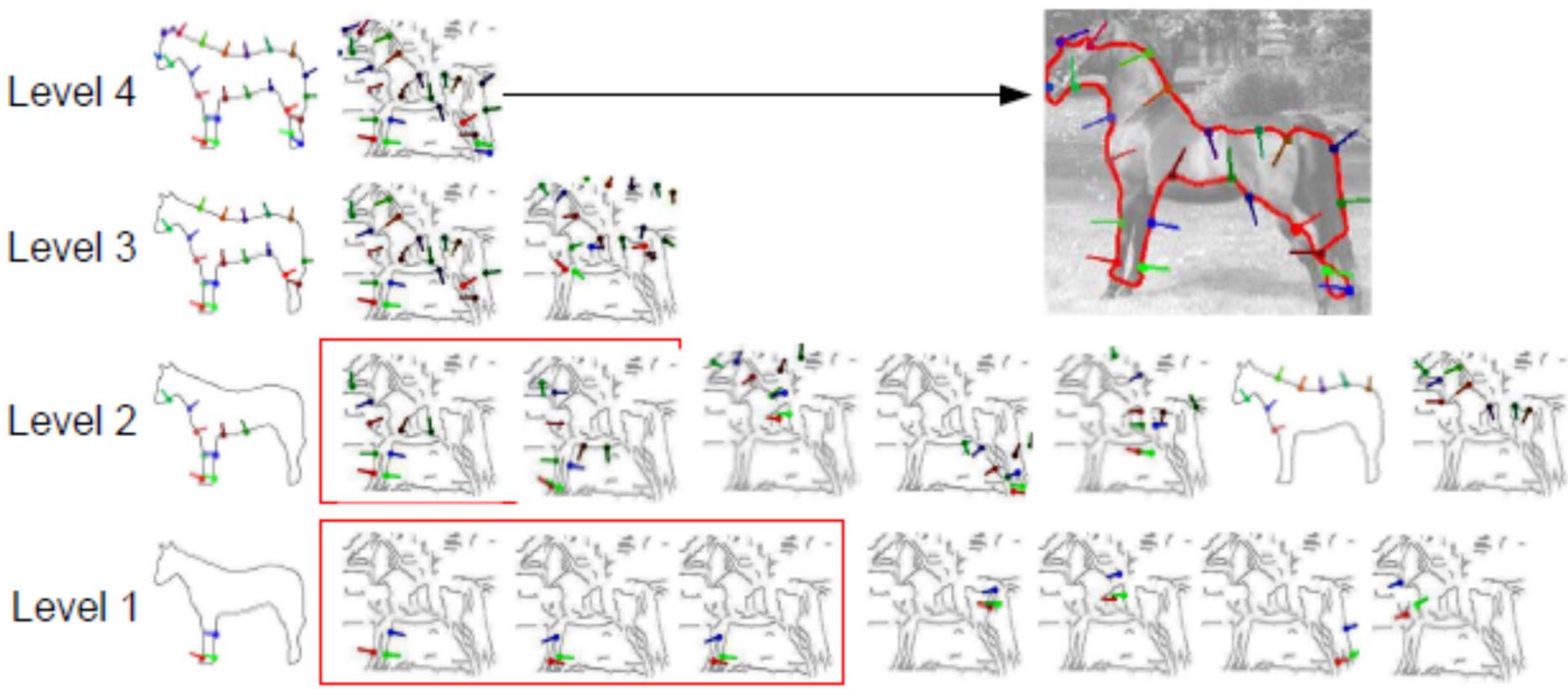
- Top-Down: Estimate $x^* = \arg\max \phi(x).$
- Repeat:

$$(x_1^*, x_2^*) = \arg\max_{(x_1,x_2)} \{\log P(x_1, x_2|x^*) + \phi_1(x_1) + \phi_2(x_2)\}$$

- And so on to obtain: $x_{11}^*, x_{12}^*, x_{21}^*, x_{22}^*.$
- *Intuition:* propagate up hypotheses about the states of subparts of the object. *Increased context as you rise up the hierarchy, less ambiguity.* Estimate coarse structure first --- executive summary. *Top-down uses high-level context to resolve low-levels ambiguities.*

Inference: Illustration

- Bottom-Up



Theories of the Visual Cortex

- Most theories of the visual cortex assume bottom-up/feedforward processing – but some advocate top-down generative approaches.
- Compositional models have aspects of both. They are generative (e.g., synthesis and attention). But allow rapid inference.
- *Inference is done by propagating hypotheses upward in a feedforward pass,* followed by a top-down pass to remove low-level ambiguities.
- *“High-level vision tells low-level to stop gossiping”.Murray, Kersten et al.’s fMRI study.*

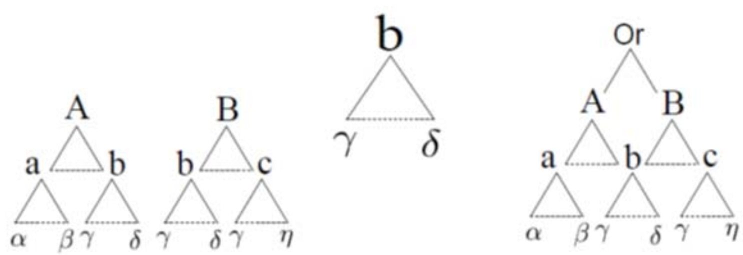
Complexity of Inference for a Single Object

- We can analyze the complexity of inference for a single object – standard analysis of DP.
- Factors:
- (i) No. of Layers -- H.
- (ii) No of children in parent-child --- r.
- (iii) No. of parent-child configurations – C_r
- (iv) Spatial decay factor (ex. summary.) -- q

- Assumed to be the same at all levels of the hierarchy.

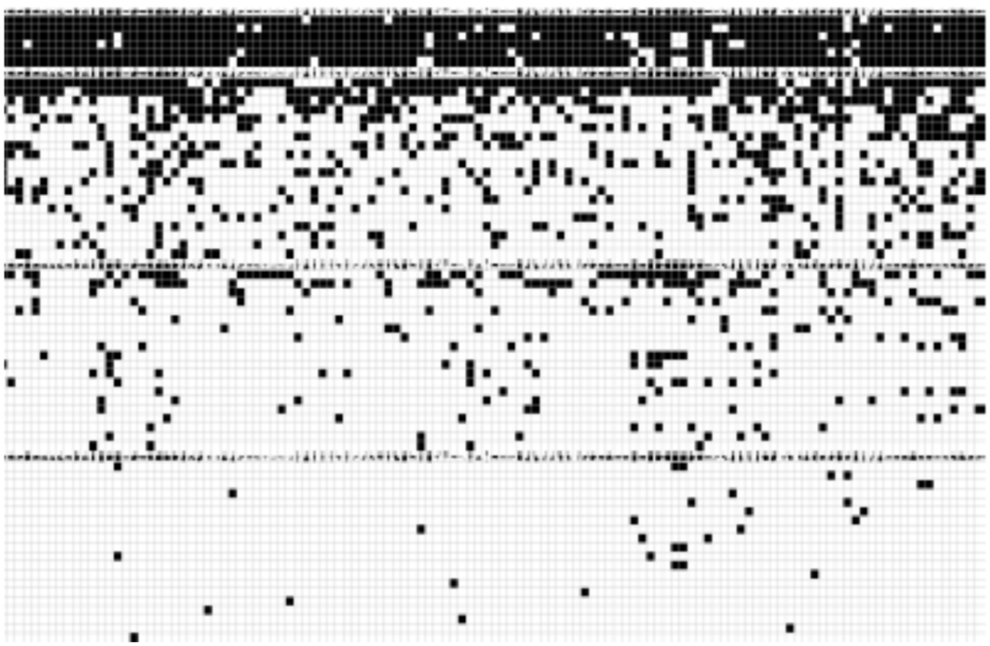
Multiple Objects: Part Sharing

- *If parts and shared between objects we can share the computation between many objects – or many instances of the same object.*
- Captured by hierarchical dictionaries:a,b,c,A,B.
- *Model competition – at top-level – determines which object is present (if any),*
- *No need to train a final classification stage! (Rev.)*



Part Sharing Example: L.Zhu et al. CVPR 2010

- Sharing of parts between 120 objects (horizontal)
- Left: Part Sharing (black)
- Right: Dictionaries – mean shapes only.

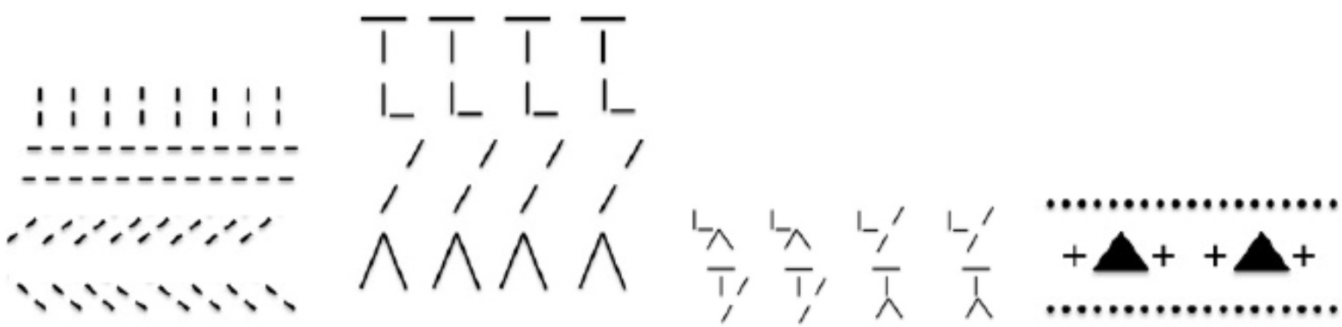


Multiple Objects: Inferences

- Inference is performed on the dictionaries with model competition at top-level.
- Recall that a dictionary element at level l is composed (by parent-child relations) of dictionary elements at level l-1
- The complexity of inference depends on the number of dictionary elements.
- Exact inference – relations to UAI work on techniques for speeding up inference on graphs? (E.g., Darwiche and Choi).

Parallel Implementation.: Convolutional Compositions?

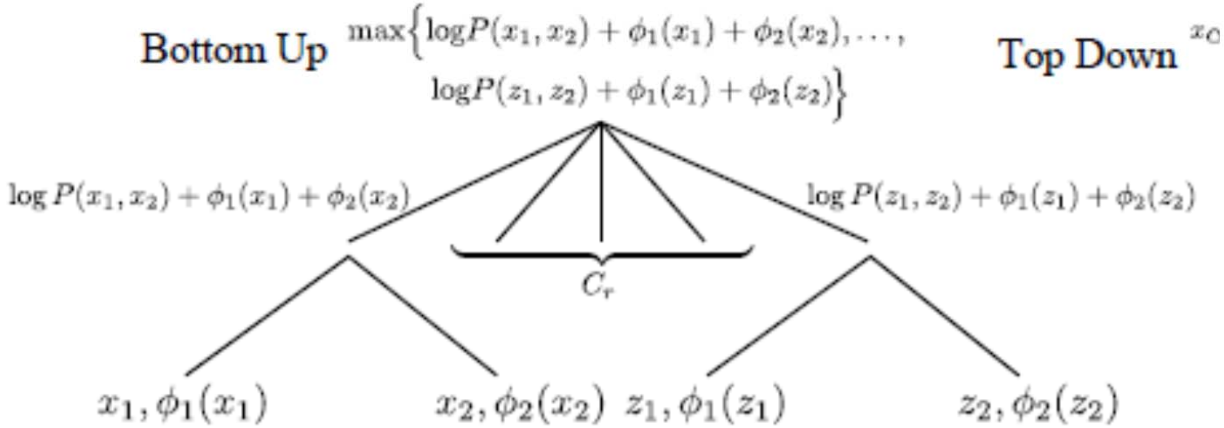
- Dynamic Programming is naturally parallelized.
- Make copies of the dictionaries at different spatial positions.
- Fewer copies at high-levels (executive summary).
- Non-Linear "Receptive Fields":



Parallel Implementation of DP

- The bottom-up pass is an AND-like operation followed by an OR-like operation.
- The top-down pass selects the child configuration with maximum score.

$$x_{Ch} = \arg\max\Big\{\log P(x_1, x_2) + \phi_1(x_1) + \phi_2(x_2), \ldots, \log P(z_1, z_2) + \phi_1(z_1) + \phi_2(z_2)\Big\}$$



Complexity for Single Objects.

- The complexity of DP – bottom-up pass is:
- D_0 size of image lattice
- C_r no. child-parent configurations.
- H no. of levels
- r no. of children (e.g. r=3)
- q scale decrease factor (executive summary).

$$N_{bu} = \sum_{h=1}^{\mathcal{H}} |\mathcal{D}_0| C_r r^{\mathcal{H}} (q/r)^h = |\mathcal{D}_0| C_r r^{\mathcal{H}} \sum_{h=1}^{\mathcal{H}} (q/r)^h = |\mathcal{D}_0| C_r \frac{q r^{\mathcal{H}-1}}{1 - q/r} \{1 - (q/r)^{\mathcal{H}}\}.$$

Serial and Parallel Impl. with Part Sharing

- If we do not share parts, then computation scales by the no. M_H of objects.
- For serial Impl. – with part sharing – the complexity depends on the dictionary size M_h at levels h:

$$N_{p_s} = |\mathcal{D}_0| C_r \sum_{h=1}^{\mathcal{H}} |\mathcal{M}_h| q^h.$$

- Parallel Impl – comp. time linear in no. level H.
- But requires no. “neurons”. Copies of dictionaries.
- Trade-off – speed neurons

$$N_n = \sum_{h=1}^{\mathcal{H}} |\mathcal{M}_h| q^h |\mathcal{D}_0|.$$

Analysis: Inference Regimes

- The complexity gains depends on the no. of shared parts: M_h at level h.
- Three Regimes:
- (i) The exponential growth regime (shape?)
- (ii) The empirical regime (CVPR 2010)
- (iii) The exponential decrease regime (appearance?)

Exponential Growth Regime

- This regime is natural for shapes (at the low levels, at least).
- Dictionary elements at one level can be composed with most other dictionary elements to form the dictionary at the next level.

Result 1: If the number of shared parts scales exponentially by $|\mathcal{M}_h| \propto \frac{1}{q^h}$ then we can perform inference for order $q^{\mathcal{H}}$ objects using part sharing in time linear in $\mathcal{H}$, or with a number of neurons linear in $\mathcal{H}$ for parallel implementation. By contrast, inference without part-sharing requires exponential complexity.

Empirical Regime

- This regime was learnt by the unsupervised algorithm (**1st part).** L.. Zhu et al. CVPR 2010.
- Note: similar to the exponential growth regime for the first few levels, then size of dictionaries decays quickly.

Result 2: If $|\mathcal{M}_h|$ grows slower than $1/q^h$ and if $|\mathcal{M}_h| < r^{\mathcal{H}-h}$ then there are gains due to part sharing using serial and parallel computers. This is illustrated in figure (7)(center panel) based on the dictionaries found by unsupervised computational learning [19]. In parallel implementations, computation is linear in $\mathcal{H}$ while requiring a limited number of nodes ("neurons").

3^{rd} Regime: Exponential Decay

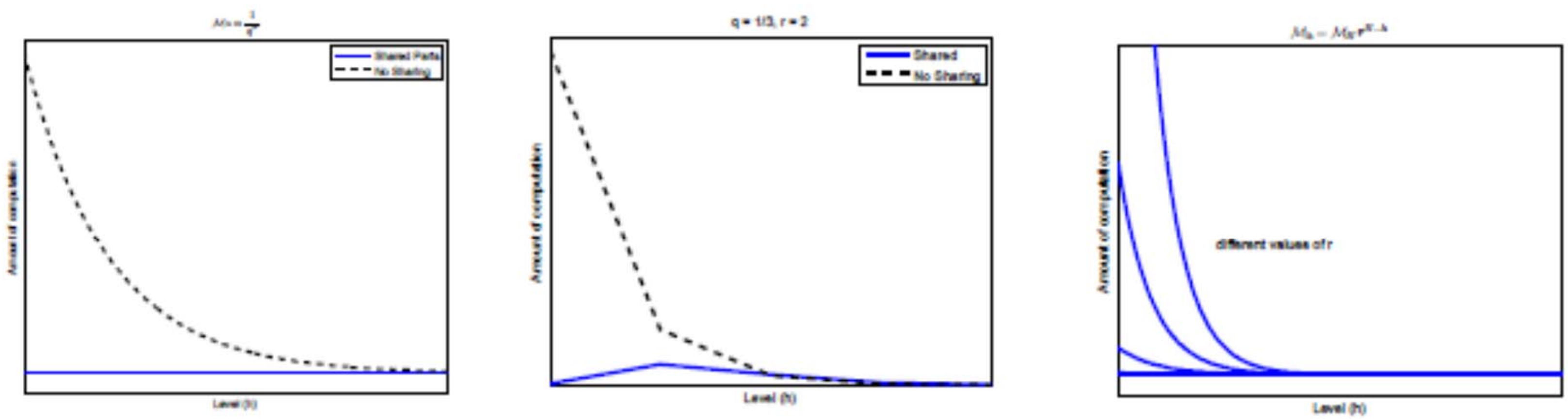
- M_h decreases exponentially with h.
- *This is the "appearance" regime?*
- Intuition: low-level give detailed description:
- (i) Siamese cat fur, (ii) Cat fur, (iii) fur,.
- Executive summary in appearance.

the advantages of parallel computing.

Result 3: If $|\mathcal{M}_h| = r^{\mathcal{H}-h}$ then there is no gain for part sharing if serial computers are used, see figure (7)(right panel). Parallel implementations can do inference in time which is linear in $\mathcal{H}$ but require an exponential number of nodes ("neurons").

Complexity in Figures.

- These illustrate complexity for the three domains.



(a) (b) (c)

• Exponential Decay Regime

- This regime is intriguing. It may corresponds to representing the full appearance of objects, and not just their edges.
- Low-level dictionaries represent local appearance patterns.
- In the parallel impl, it requires a very large no. of “neurons” at the lowest levels.
- *Implications for the brain? It suggests that there should be many low-level dictionaries with many local copies.*
- Note: 70% of neurons in the visual cortex are in the low levels. V1 and V2? 30% of the cortex.

Summary

- Complexity Analysis of Compositional Models.
- Serial and Parallel Implementations.
- Gains due to part sharing – compositionality – depend on how the part dictionaries scale with level. Three regimes.
- Visual Cortex speculations: can we derive the structure of the cortex from first principles – as a hierarchical pattern recognition device which is efficient for representation and inference?

References:

- A.L. Yuille and R. Mottaghi. Int. Conf. Learning Representations (ICML). 2013. Archive.

- L. Zhu et al. Unsupervised Structure Learning. ECCV. 2008.
- L. Zhu et al. Part and Appearance Sharing. CVPR 2010.