Compositional (Semantic) Models and Unsupervised Graph Structure Learning.

Alan Yuille

JHU

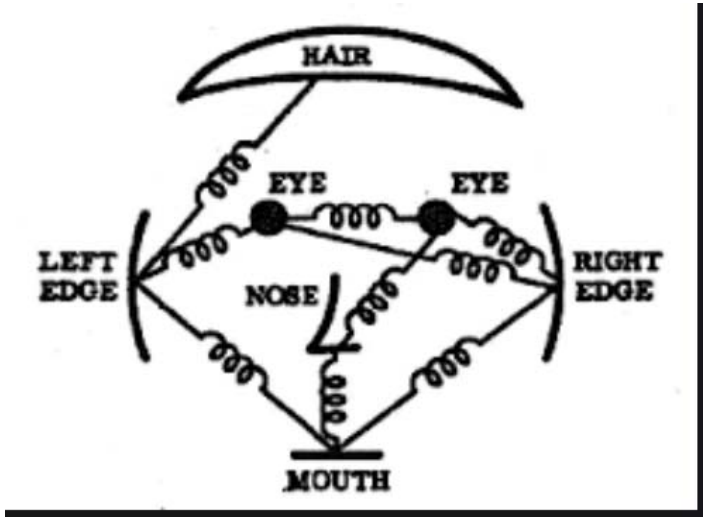
Plan of Lecture

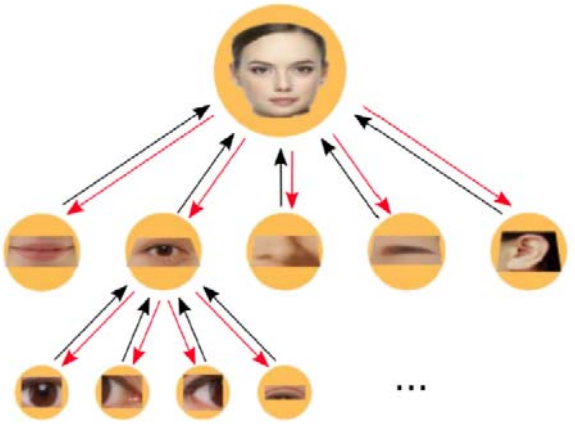
- (1) Representing Objects by Hierarchical (Semantic) Compositional Models.
- (2) Unsupervised Structure Learning.

- Note: “semantic” is used to mean that the parts are interpretable (i.e. not just top-right of object) and “semantic composition” means that an object, or part, is composed of interpretable parts or subparts (i.e. not just mathematical composition).

(1). Objects and Semantic Parts

- Objects can be represented by parts and their spatial relations (Pictorial Structures: left).
- Hierarchical semantic compositional models: objects, parts, subparts (right).





Hierarchical Compositional Models

- Properties:
- The models are explainable/interpretable. If an object is detected you can explain which parts and subparts are present.
- The model performs multiple tasks – detect/recognize/localize objects, detect/recognize/localize parts/subparts, detect/localize the object boundaries.
- *Many advantages – but many challenges.*

HCMs

- Formally, HCMs can be represented by hierarchical graph structures.
- Relations to Deep Networks.
- The parts/subparts and relationships between them need to be learnt.
- Inference is bottom-up and top-down. And requires sideways/lateral reasoning.
- Explicit/interpretable – advantages for out-of-distribution learning and domain adaptation.
- Cognitive Science and Neuroscience justification.

Hierarchical Models

- Why Hierarchies?
- Mimics the structure of the human/primate visual ventral system.
- Follows the low-, middle-, high-level nature of vision.
- Low-level vision is ambiguous. High-level vision exploits context and is un-ambiguous.
- Optimal design for representing, learning, and retrieving image patterns?

Grammars/Compositional Models

- **Explicit Representations** – ability to perform multiple tasks.
- **Sharing** – efficiency of inference, efficiency of learning.
- Relates to Stochastic Grammers used in Natural Language Processing.

A Probabilistic Model is defined by four elements

- (i) **Graph Structure** – Nodes/Edges -- *Representation*
- (ii) **State Variables** – W – input I. --*Representation*
- (ii) **Potentials** – Phi -- *Probability*
- (iii) **Parameters/Weights** – Lambda – *Probability*

- *The state variables are defined at the graph nodes.*
- *The potentials and parameters are defined over the graph edges – and relate the model to the image I.*

The Mathematics

- The mathematical formulation.
- Exponential models.

$$\text{Graph} : (\mathcal{V}, \mathcal{E}) : \ \mathcal{V} \text{ nodes}, \ \mathcal{E} \text{ edges}. \ \ \mathcal{V}^l : \ \text{nodes level } l.$$
$$\text{Children } ch(\mu) \subset \mathcal{V}^{l-1}, \text{siblings } sib(\mu) \subset \mathcal{V}^l.$$
$$\text{State variables} : w_\mu \quad w_{ch(\mu)}, w_{sib(\mu)} \text{ states of children and siblings.}$$

$$\text{Vertical Potentials } \phi^V(w_\mu, w_{ch(\mu)}) : \ \text{Weights } \boldsymbol{\lambda}^V_\mu.$$
$$\text{Horizontal Potentials } \phi^H(w_\mu, w_{sib(\mu)}) : \ \text{Weights } \boldsymbol{\lambda}^H_\mu.$$
$$\text{Data Potentials } \phi^D(w_\mu, \mathbf{I}) : \ \text{Weights } \boldsymbol{\lambda}^D_\mu.$$

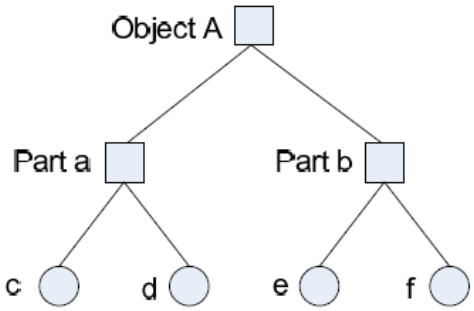
$$P(\mathbf{W}|\mathbf{I}) = \frac{1}{Z[\boldsymbol{\lambda}^D, \mathbf{I}]} \times \exp\{\sum_{\mu \in \mathcal{G}} \boldsymbol{\lambda}^D_\mu \cdot \phi^D(w_\mu, \mathbf{I}) + \sum_{\mu \in \mathcal{V}} \boldsymbol{\lambda}^V_\mu \cdot \phi^V(w_\mu, w_{ch(\mu)}) + \sum_{\mu \in \mathcal{V}} \boldsymbol{\lambda}^H_\mu \cdot \phi^H(w_\mu, w_{sib(\mu)})\}.$$

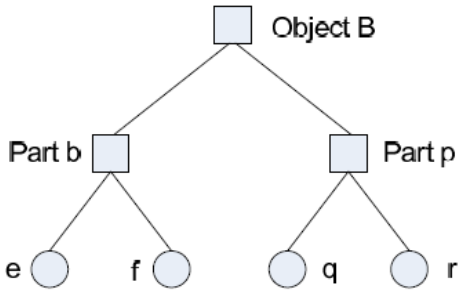
Tasks:

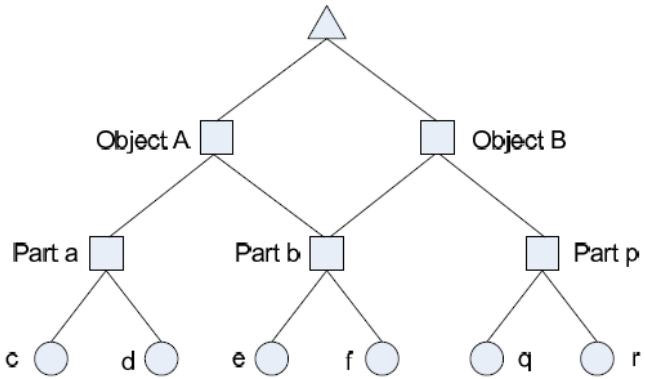
- (I) **Inference** – estimate the state W from input I – assuming known Graph Structure, Potentials and Parameters. (*Intuitively: propagate hypotheses up the hierarchy and validate them top-down – dynamic programming as a special case).*
- (II) **Learning Parameters/Potentials** – assuming known Graph Structure. (*Straightforward if inference can be done*).
- (III) **Structure Induction** – learn the Graph Structure. (*Second half of Lecture*)

Key Idea: Compositionality

- Objects and Images are constructed by compositions of parts – ANDs and ORs.
- The probability models for are built by combining elementary models by composition.
- Efficient Inference and Learning.







Why compositionality?

(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.

(4). Scale up to multiple objects by part-sharing.

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

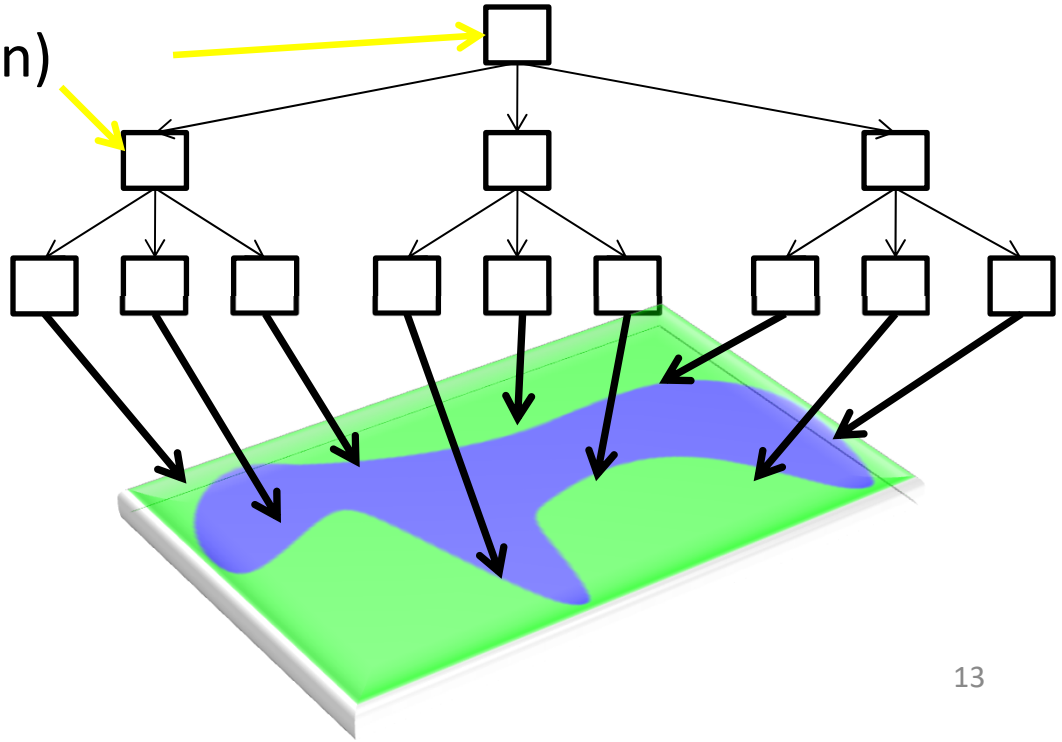
"The world is compositional or God exists".

Horse Model (ANDs only).

Nodes of the Graph represents parts of the object.
Lower level parts are edges and edge-groupings.

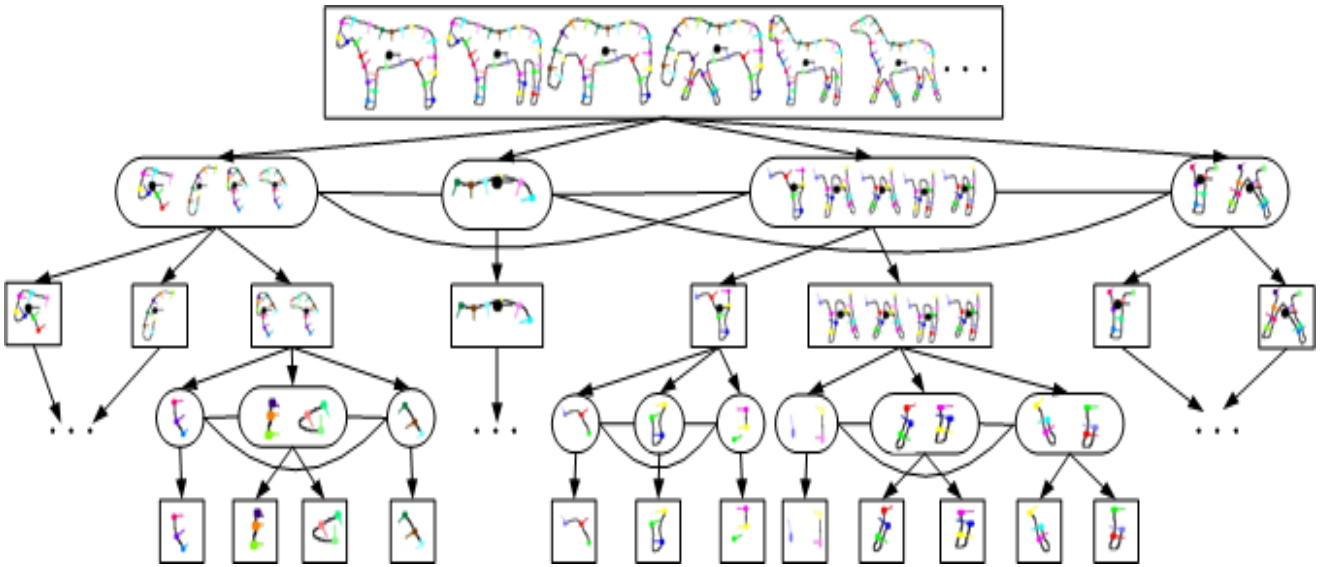
Parts can move and deform.

y: (position, scale, orientation)



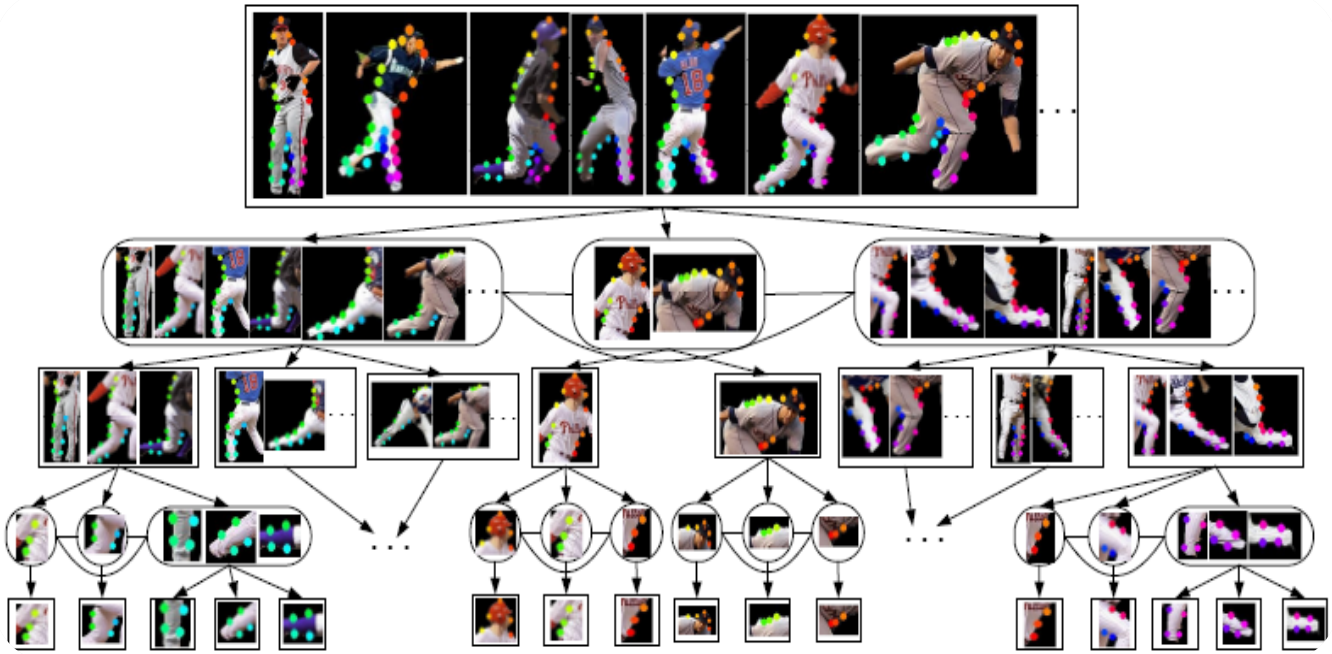
AND/OR Graphs for Horses

- Introduce OR nodes and switch variables.
- Settings of switch variables alters graph topology – *allows different parts for different viewpoints/poses*:
- Mixtures of models – with shared parts.



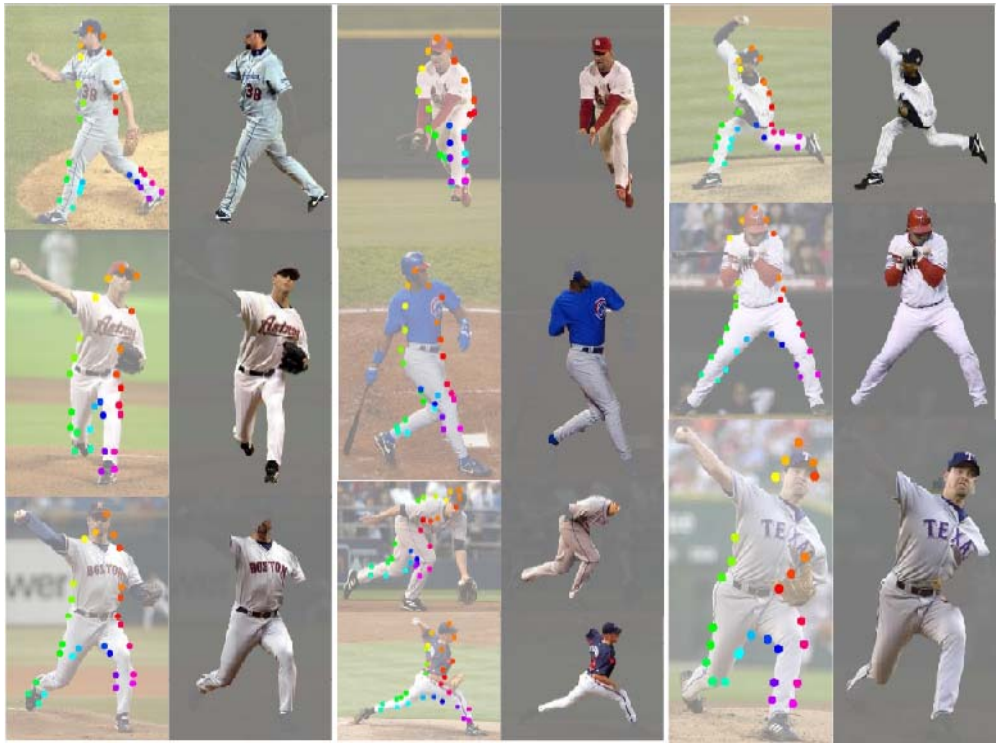
AND/OR Graphs for Baseball

- Enables RCMs to deal with objects with multiple poses and viewpoints (~100).
- Inference and Learning by bottom-up and top-down processing:



Results on Baseball Players:

- Performed well on benchmarked datasets.
- Zhu, Chen, Lin, Lin, Yuille CVPR 2008, 2010.



Conclusion (1): Challenges of HCMs?

- HCMs are significantly more complex than Deep Networks, in terms of inference and learning algorithms. Structure learning is particularly difficult (second half of lecture).
- But their advantages – interpretability, multi-tasking – and their potential – ability to deal with domain transfer and sophisticated attacks – are so strong that this class of models should be pursued.
- Later lectures will discuss recent developments which combine HCMs with Deep Network Features.

(2) Unsupervised Structure Learning

- This is an extremely challenging task. The work described is by two extremely strong students/postdocs – Long (Leo) Zhu and Yuanhao Chen (who are the driving forces behind the AI company YiTu).
- Intuition for structure learning: Clustering.

Generative Models and Images

- Learning Generative Models of entire images is too hard at present – cf. special cases.
- Structure Induction is very hard.
- To simplify: use generalize models for simple features.
- (i) Interest Points (IPs). Described by SIFT.
- (ii) Edgelets.
- Learn models for objects (not images).

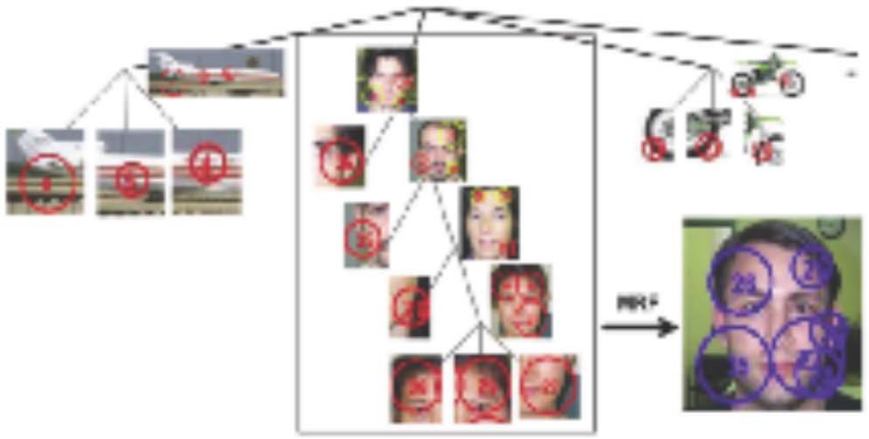
Unsupervised Structure Induction.

- The Challenge:
- We do not know the graph structure.
- We do not know if an object is present in the image.
- We do not know how many types of objects can be present in the image.
- We do not know what IPs are `object' or `background'.
- We do not know the correspondence between image IP's and the graphical model.

Probabilistic Graphical Mixture Model (1)

- L. Zhu, Y. Chen, and A.L. Yuille. PAMI. Jan. 2009.
- Dataset – Caltech.

- The input data is a set of natural images;
- The output of model a structure like following



PGMM 2.

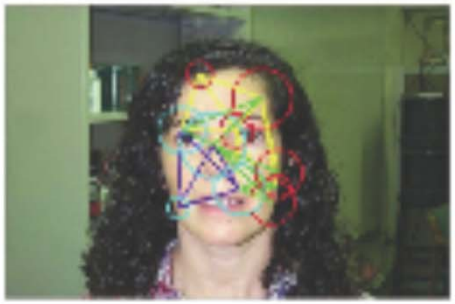
- The object has a cluttered/noisy background. We do not know what is object and what is background.

- The cocktail party effect describes the ability to focus one's listening attention on a single talker among a mixture of conversations and background noises, ignoring other conversations.
- A single talker: Interest Points
- Other conversations: background

PGMM 3.

- This method is based on Interest Points (IPs).
- Why? Because there are few IP's (sparse).
- They capture important (interesting) parts of the object.

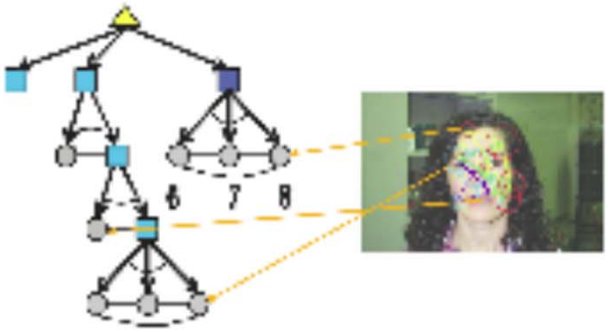
- An interest point is a point in the image which is
  - well-defined position, include an attribute of scale
  - robust to photometric and geometric transformations
- Interest operator
  - Kadir-Brady operator
  - SIFT operator



PGMM 4:

- Correspondence problem.
- Some interest-points (IPs) are background
- Others are from the object,
- But from which part of the object?

- Why we need inference to solve correspondence problem?
  - Data used by Orban is clean and vocabulary known;
  - PGMMs extract IP's and clustering them into a vocabulary from natural images;
  - PGMMs needs to extract IP's and match them to words already known from a new image.



PGMM 5:

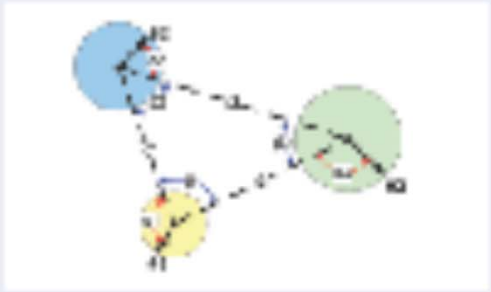
- The Basic Idea:

  - The basic idea of PGMMs is to search over model structure to find optimal structure.
  - The whole procedure is a greedy search,
    1. Initially, all of the data are assumed to be generated by a background model, without any spatial relationship between them;
    2. Expand the structure by using AND/OR graph grammar, and the grammar will be demonstrated below;
    3. For each extension, use the model evaluation method to evaluate it and get a score. Accept the extension with the highest score and update the structure;
    4. Repeat 2 & 3 until the score almost doesn't change, exit with graph structure then.

PGMM 6:

- The model is built by a Grammars.
- The basic elements are triplets of IP's.

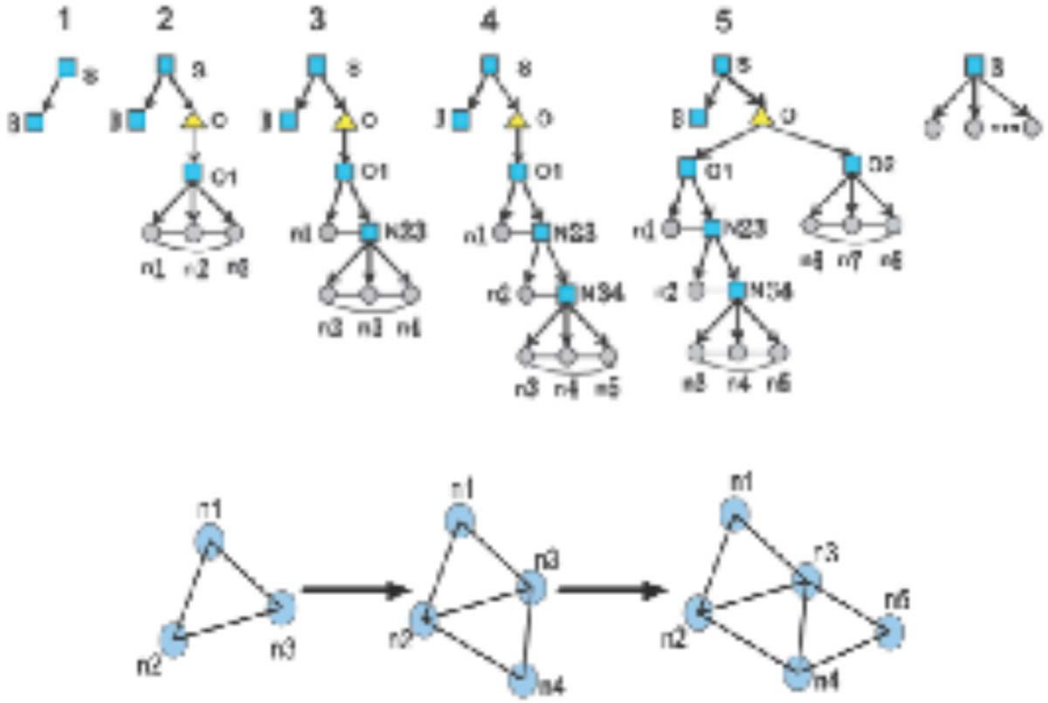
- Triplets is based on the three related nodes' position z_i, scale l_i and orientation θ_i. $r_i = \{z_i, l_i, \theta_i\}$ denote the position feature for a point, and $l = \{r_a, r_b, r_c\}$ denotes a triplet



- The graph grammars are,
  1. AND extension. Combine the new triplet and old one.
  2. OR extension. Connect the new triplet with an old one.

PGMM 7:

- Grammars – and how to grow them.
- Start with a triplet – and another triplet – if the resulting model fits the data better.
- Model selection – choose between models.



PGMM 8:

- Model section is performed by evaluating the probability that they model generates the data.
- In practice, we make a standard approximation (Laplace).

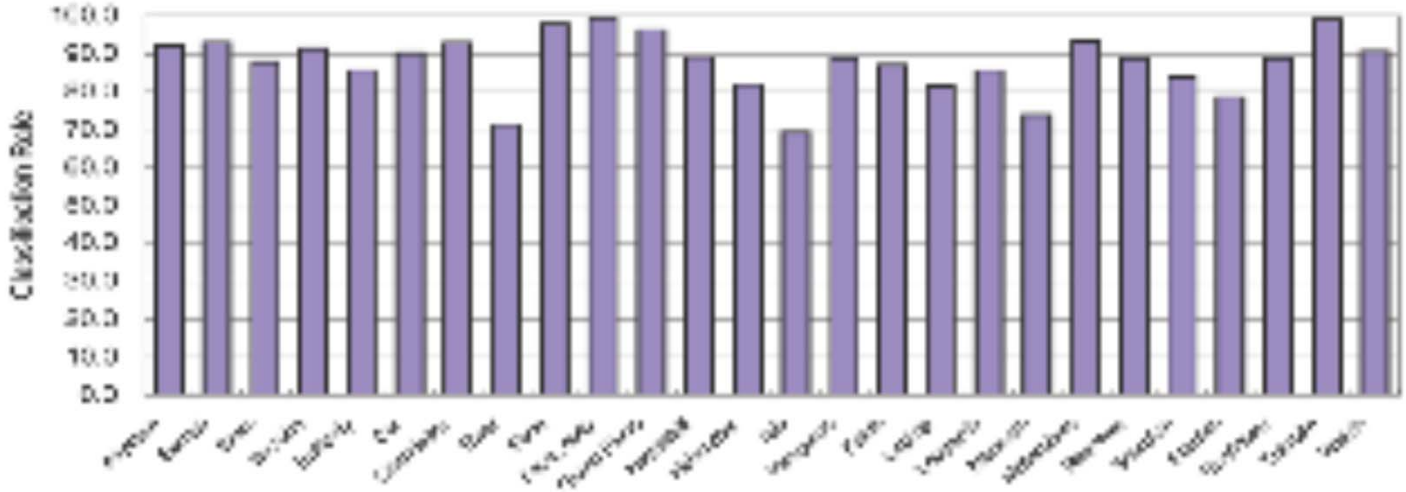
- For model selection, to integrate θ is a big challenge, Leo used the maximum value of θ to replace the results of integral,

$$P(D|I) = \prod_i \sum_H P\left(D^i|H,\theta^*,I\right) P\left(H|\theta^*,I\right) P\left(\theta^*|I\right) P\left(I\right) \quad (14)$$

- Using Laplace approximation.

PGMM 9:

- Some experimental results on Caltech 101:
- Could only use a limited number of model because this approach needs a lot of data.
- Unusual to do unsupervised learning for Caltech.



The classification performance for 26 classes that have at least 80 images. The average classification rate is 87.6 percent.

PGMM 10.

- Summary:
- *Could learn one, two, three or more models if the dataset required it (e.g. plane, face, bike).*
- *Could learn object models even when half the data was random background.*
- Performance of models was as good as alternative (supervised methods) for the set of objects with sufficient data (in 2006).

PGMM 11.

- **Limitations of PGMM**: this model only uses image features defined at interest points.
- How to improve?
- Use this model to learn a ‘skeleton structure” of the object.
- **Then use the skeleton to train a model which uses more cues – edges and appearance.**
- *Eureka Moment? – when the simple IP model is powerful enough to train a model with more cues.*
- POM’s Paper. Y. Chen, L. Zhu, A.L. Yuille, and HJ Zhang. PAMI. Oct. 2009.

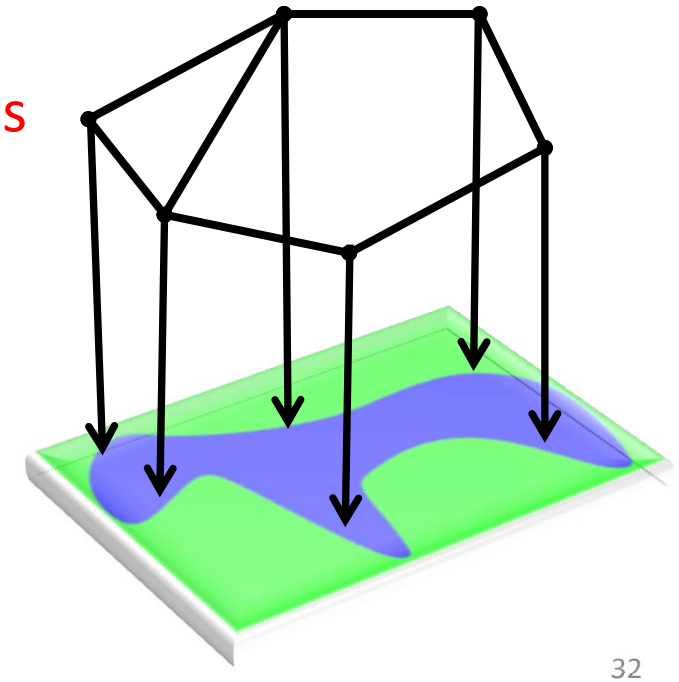
Unsupervised Hierarchical Structure Learning

- Task: given 10 training images, no labeling, no alignment, highly ambiguous features.
  - Estimate Graph structure (nodes and edges)
  - Estimate the parameters.

Correspondence is unknown

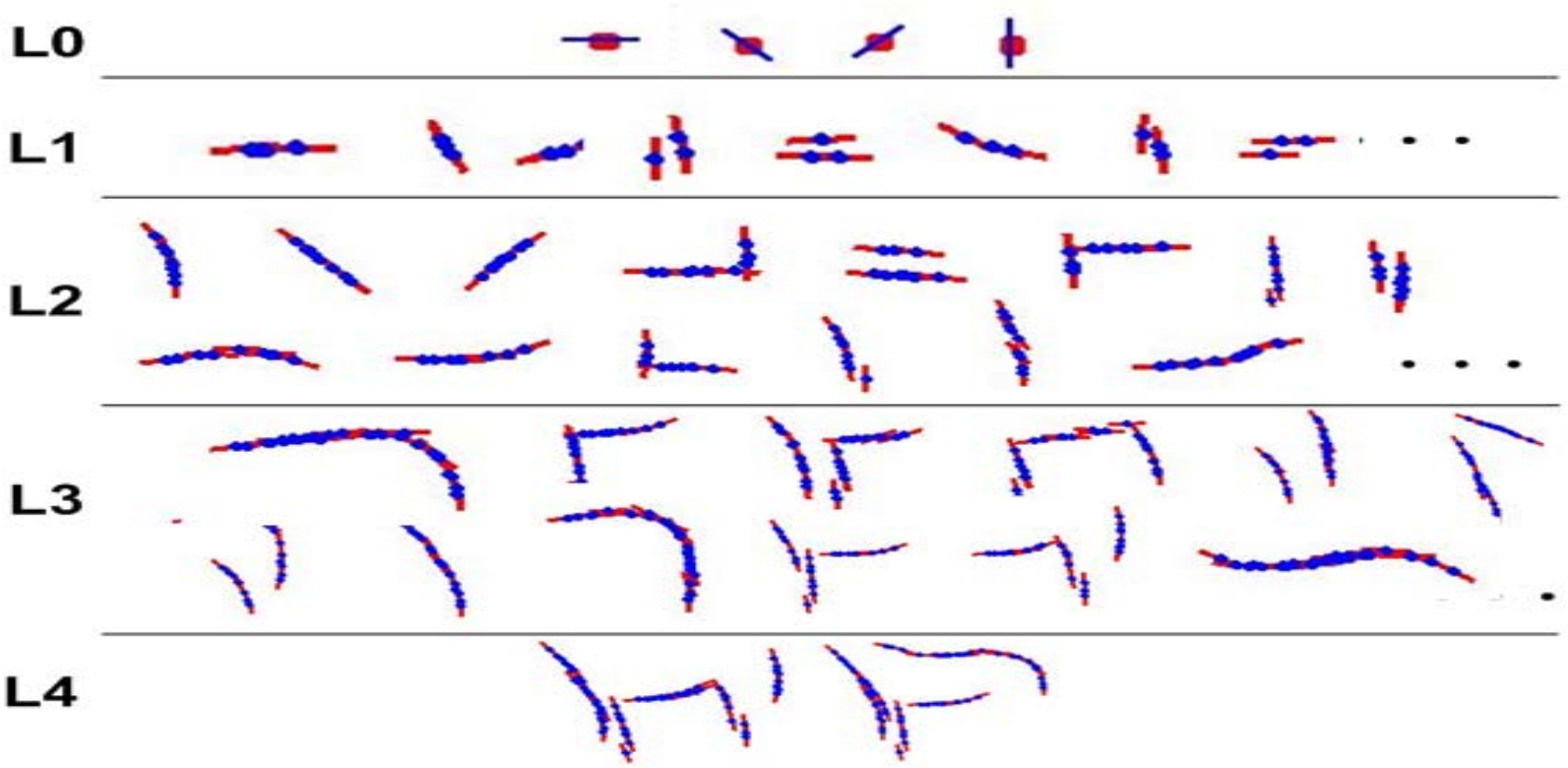
?

Combinatorial Explosion problem

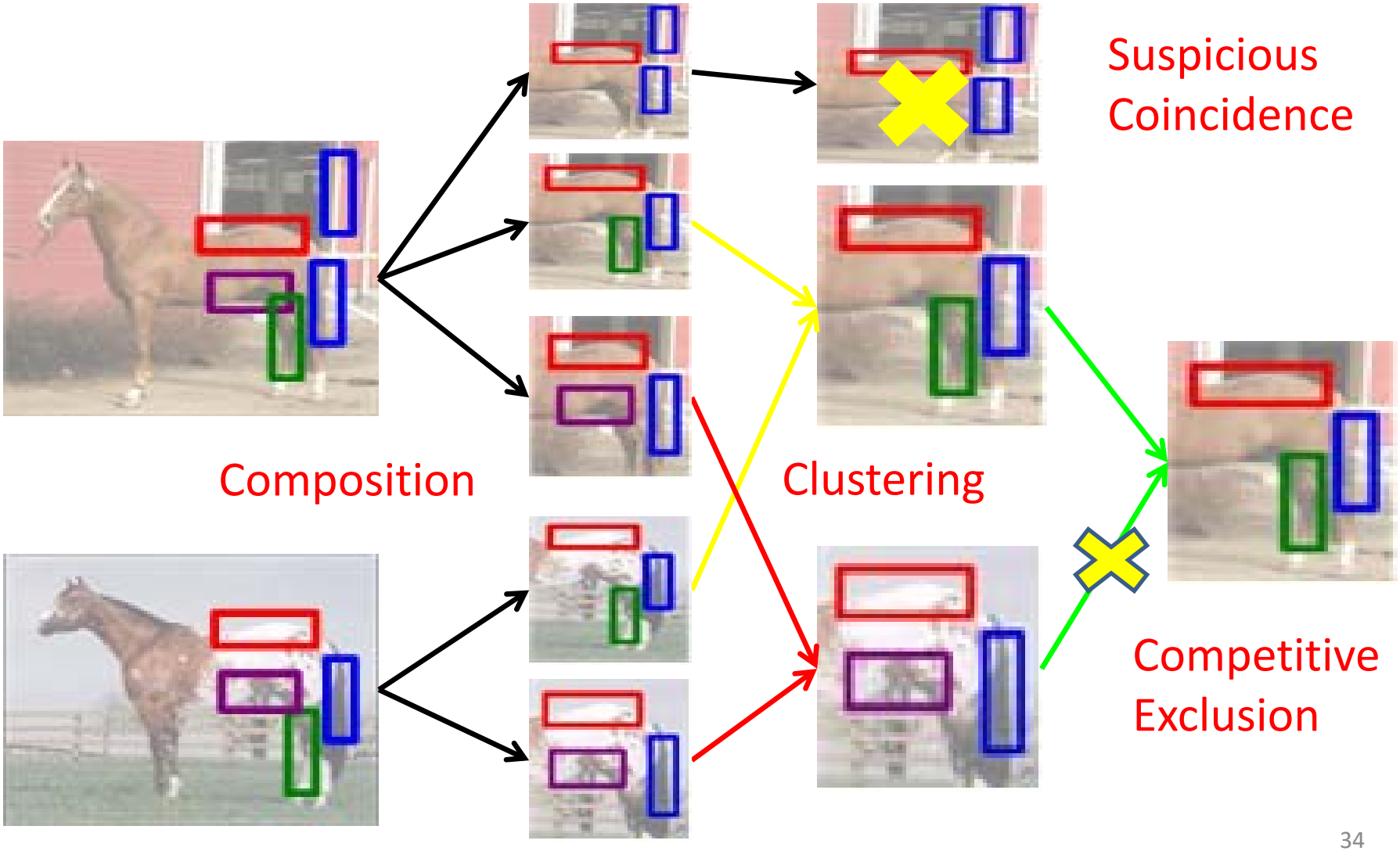


The Dictionary: From Generic Parts to Object Structures

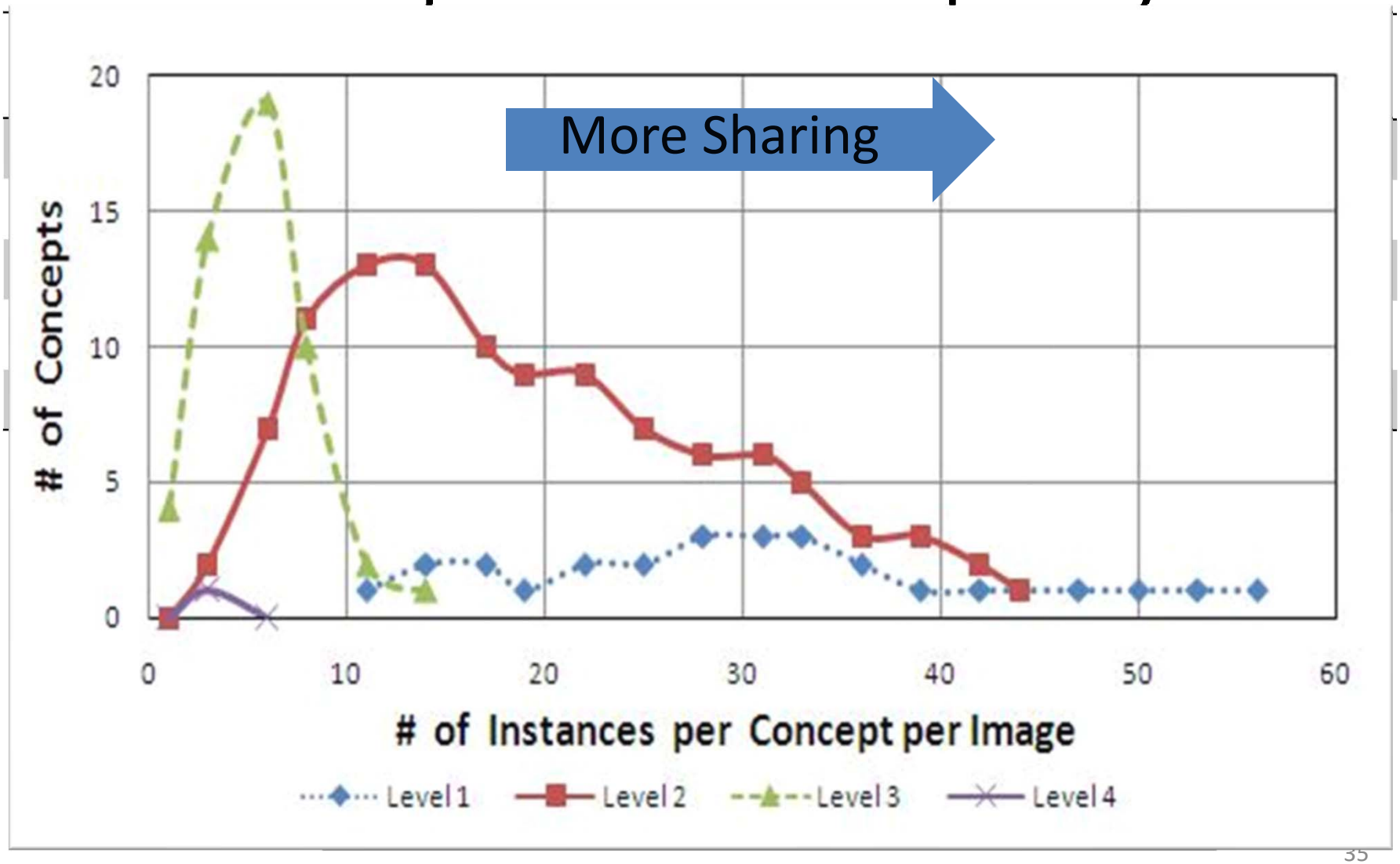
- Unified representation (RCMs) and learning
- Bridge the gap between the generic features and specific object structures



Bottom-up Learning

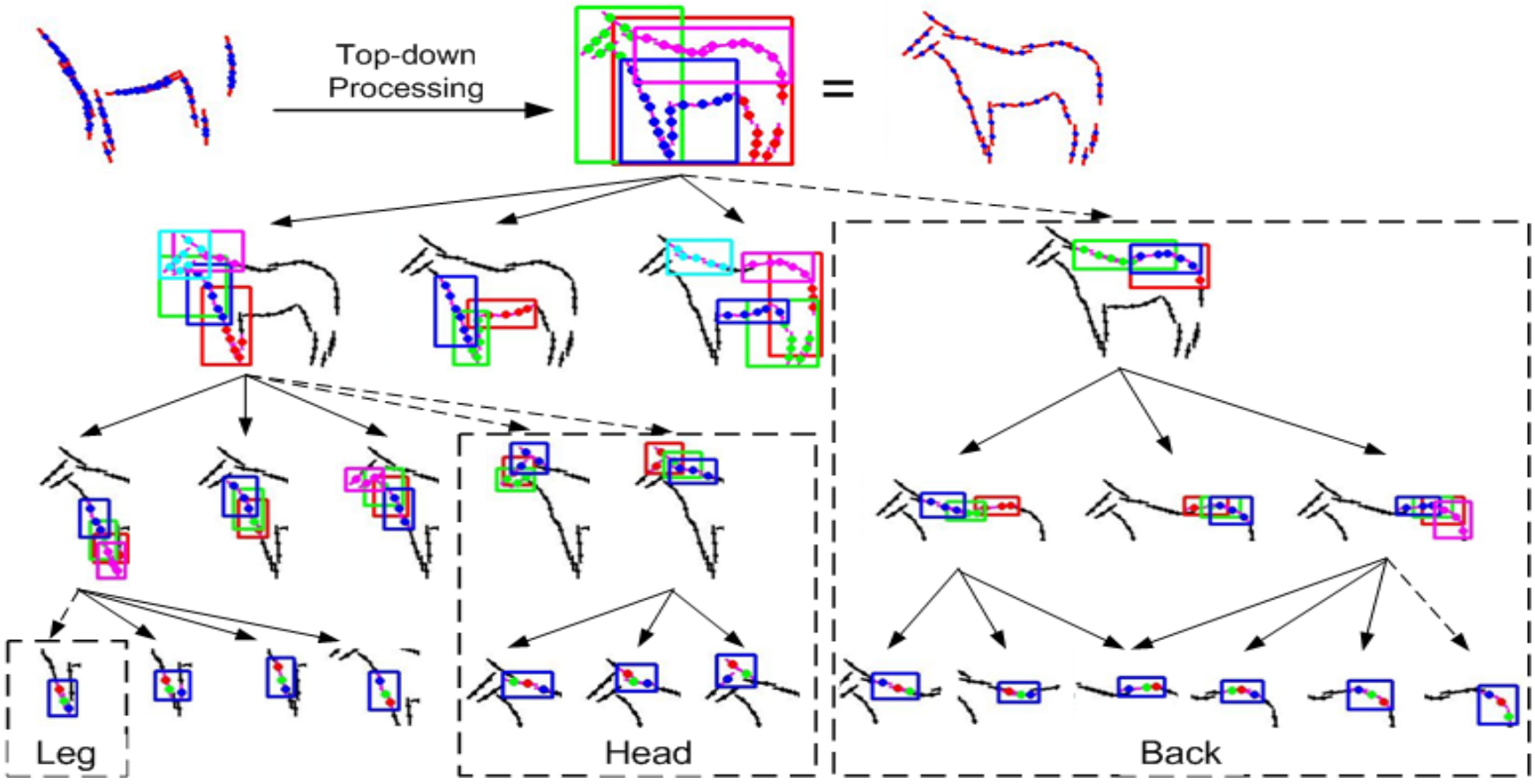


Dictionary Size, Part Sharing and Computational Complexity



Top-down refinement

- Fill in missing parts
- Examine every node from top to bottom



Part Sharing for multiple objects

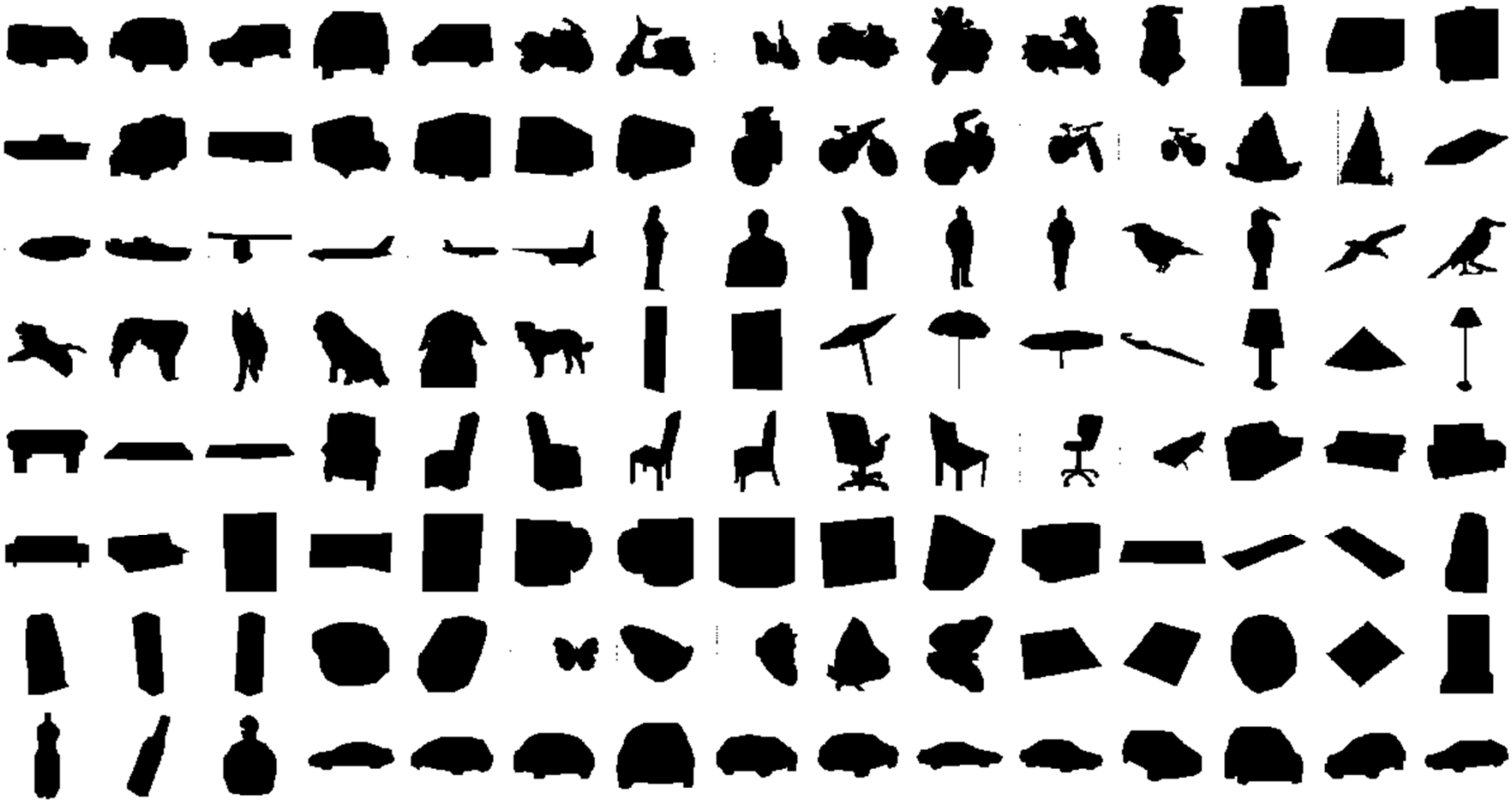
Strategy: share parts between different objects and viewpoints.

Learning Shared Parts

- Unsupervised learning algorithm to learn parts shared between different objects.
- Zhu, Chen, Freeman, Torralba, Yuille 2010.
- Structure Induction – learning the graph structures and learning the parameters.
- Supplemented by supervised learning of masks.

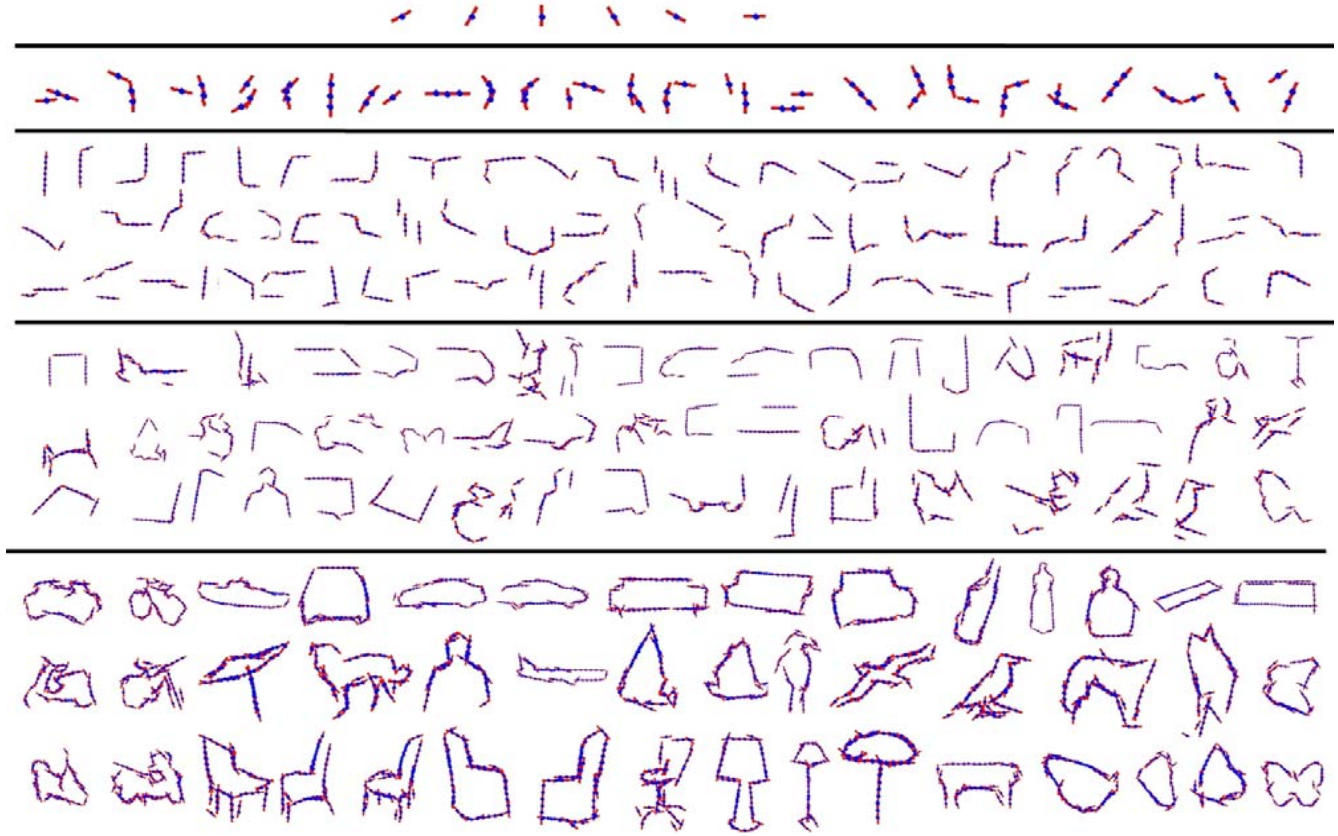
Many Objects/Viewpoints

- 120 templates: 5 viewpoints & 26 classes

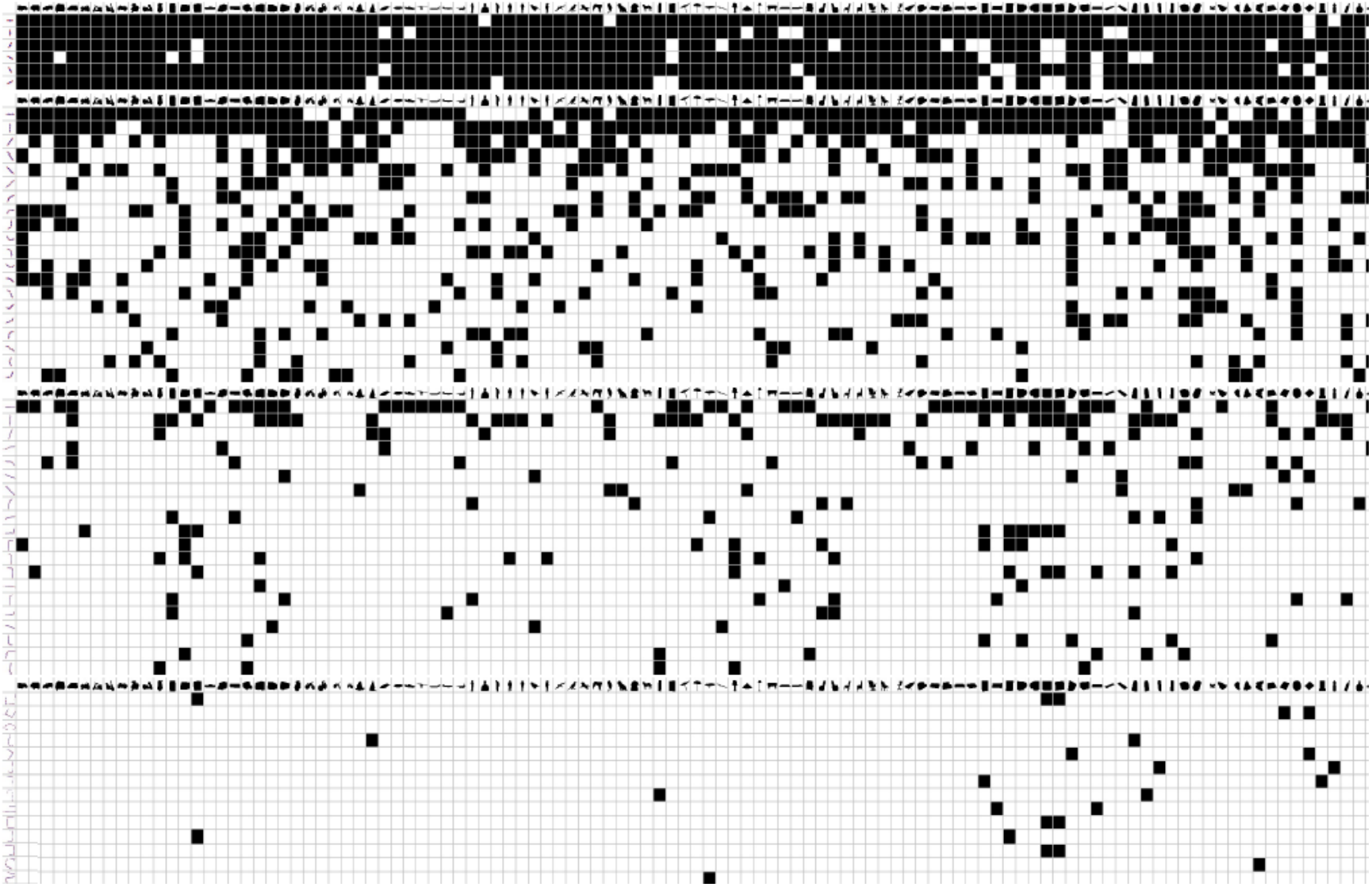


Learn Hierarchical Dictionary.

- Low-level to Mid-level to High-level.
- Learn by suspicious coincidences.

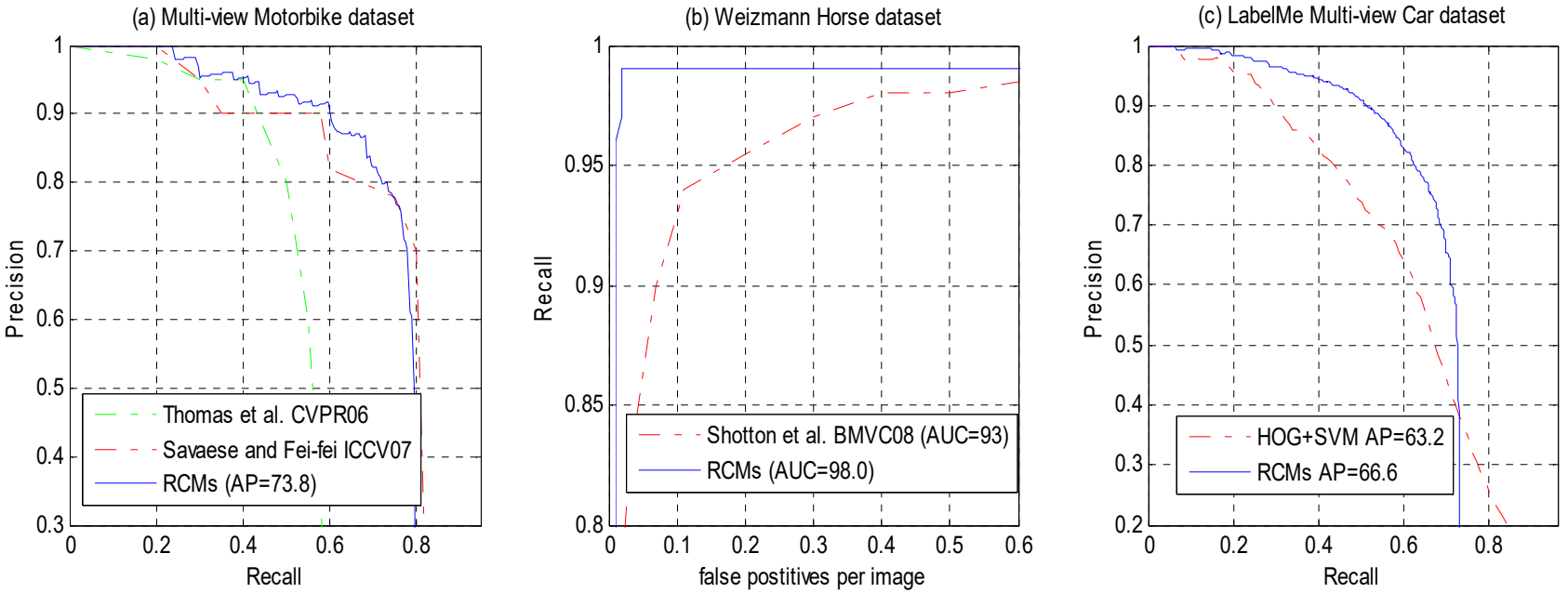


Part Sharing decreases with Levels



Multi-View Single Class Performance

- Comparable to State of the Art.



(2) Conclusion

- Unsupervised Structure Learning is possible, but very difficult.
- Question: can (limited) additional information make structure learning much easier (e.g., one annotated example of the object?).
- How do humans learn object structure? Humans learn in a life-long manner since infancy. Infants learn by interacting with the world, touching and playing with objects, not simply by seeing many images.

Conclusion

- Compositional Models are challenging, but their potential advantages are enormous – multi-task, interpretable/explainable, domain adaptation, out-of-distribution learning.