Lecture 5:
Hierarchical Compositional Representations of Object Structure

Aleš Leonardis

University of Ljubljana
Faculty of Computer and Information Science
Visual Cognitive Systems Laboratory

University of Birmingham
School of Computer Science
Centre for Computational Neuroscience & Cognitive Robotics

At CVPR, Providence, Rhode Island
June 16, 2012

Hierarchical Compositional Representations of Object Structure

Aleš Leonardis

Contributors: Marko Boben, Matej Kristan, Sanja Fidler,
Domen Tabernik

University of Ljubljana
Faculty of Computer and Information Science
Visual Cognitive Systems Laboratory

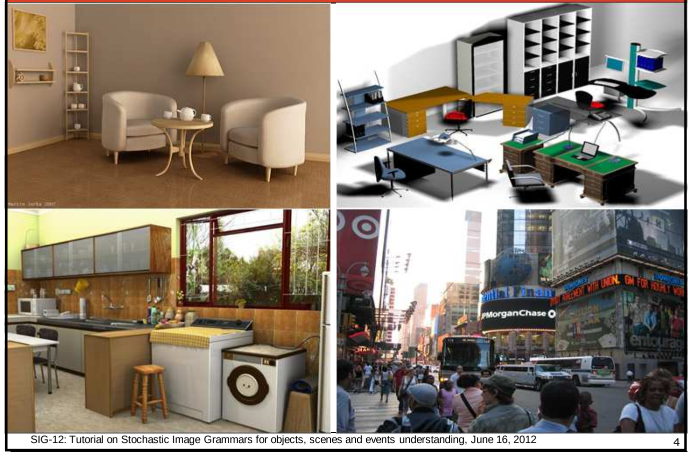
University of Birmingham
School of Computer Science
Centre for Computational Neuroscience & Cognitive Robotics

Univerza v Ljubljani

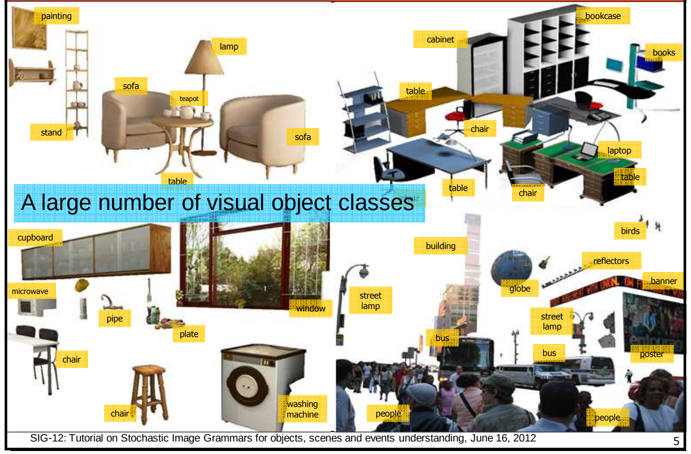
Outline

- Motivation (large number of object categories)
- Requirements
- Representation (And-Or Graphs)
- Inference
- Learning
- Experiments (Videos)
- Extensions
  - Flexible object structure
  - Adding discriminative information
- Conclusions

Large number of visual object classes

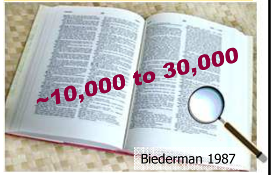


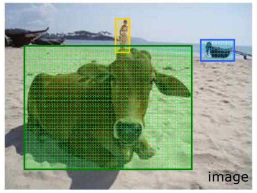
Large number of visual object classes



Intra-class variability, articulations,…

- A large number of object classes
- Significant intra/inter-class variation
- Multiple articulations
- Multiple 3D poses
- Varying illuminations
- Objects can appear at any position in an image, any scale, orientation…





Tasks

- Recognition of exemplars

- Categorization
  - Subordinate-
  - Basic-
  - Super-ordinate-level categories

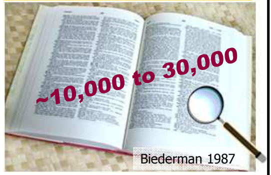
Tasks

- Grasping
- Manipulation
- Talking and reasoning about objects

Central issues

Central issues:

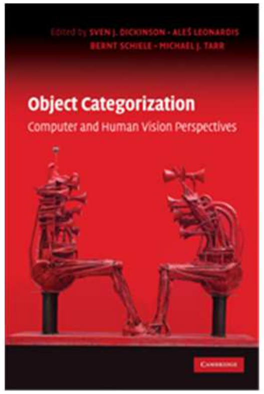
- **Representation**
- **Inference**
- **Learning**



How to tackle the problem?

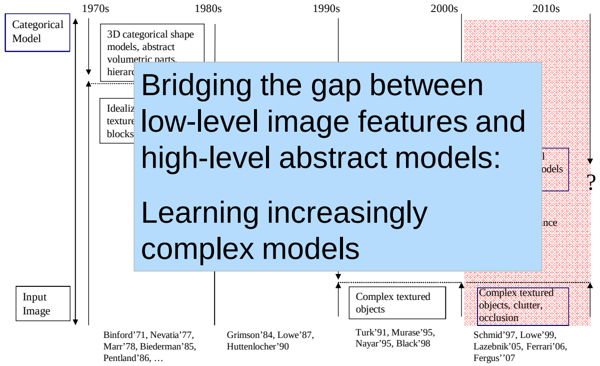
- A variety of different representations S. Dickinson, "The evolution of object categorization and the challenge of image abstraction",

In Object categorisation: Computer and Human Vision Perspectives; S. J. Dickinson, A. Leonardis, B. Schiele, M. J. Tarr, Eds., Cambridge University Press 2009.


Edited by SVEN J. DICKINSON · ALEŠ LEONARDIS
BERNT SCHIELE · MICHAEL J. TARR
Object Categorization
Computer and Human Vision Perspectives
CAMBRIDGE

Evolution of object models

Adapted from S. Dickinson, *The evolution of object categorization and the challenge of image abstraction*, Object categorization: Computer and Human Vision Perspectives, Cambridge University Press 2009.


1970s
1980s
1990s
2000s
2010s
Categorical Model
3D categorical shape models, abstract volumetric parts, hierarc
Idealiz texture blocks
odels
?
nce
Input Image
Complex textured objects
Complex textured objects, clutter, occlusion
Binford'71, Nevatia'77, Marr'78, Biederman'85, Pentland'86, …
Grimson'84, Lowe'87, Huttenlocher'90
Turk'91, Murase'95, Nayar'95, Black'98
Schmid'97, Lowe'99, Lazebnik'05, Ferrari'06, Fergus''07

Bridging the gap between low-level image features and high-level abstract models:

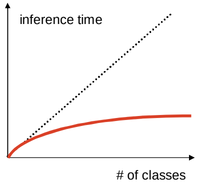
Learning increasingly complex models

Bridging the gap

- Representations and learning: the key issues
- Object categorization (2D shape)
- Requirements:
  - A representation should:
    - Support a variety of tasks
    - Enable fast and robust object detection/segmentation/parsing
    - Scale with the number of classes (modest increase in memory)
    - Accommodate exponential variability of objects
    - Enable efficient learning

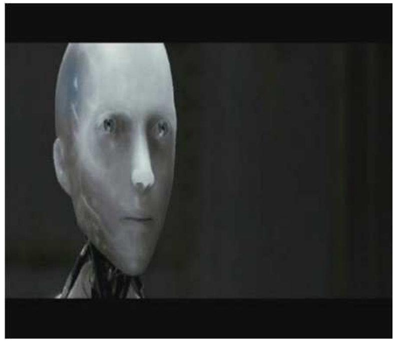
Requirements

- Inference
  - Sub-linear in the number of classes
  - Coping with noisy or missing information



- Learning should:
  - Require minimal human effort
  - Be done incrementally (no need for re-training the complete representation)
  - Share-ability (in terms of representation and processing)
  - Transfer of knowledge (learning time getting shorter)
  - Scaffolding (gradual increase of knowledge)

Hierarchical Compositional Model



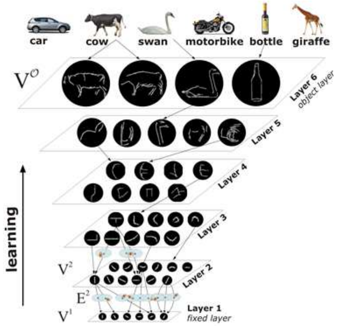
Related work

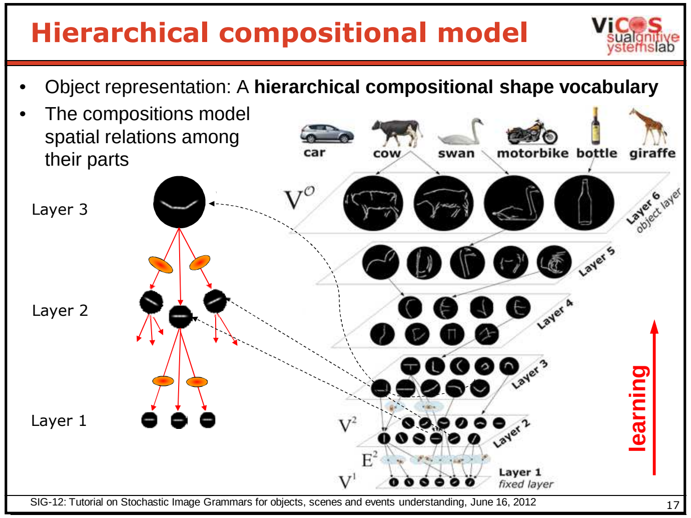
- Hierarchical representations
  - Fukushima, Sarkar & Boyer, Riesenhuber & Serre & Poggio (HMAX), Mutch & Lowe, Lecun et al. (convolutional nets), Amit & D. Geman, S. Geman, Torralba, Borenstein & Epstein & Ullman, Scalzo & Piater, Bouchard & Triggs, Ahuja & Todorovic, S.C. Zhu & Mumford, L. Zhu & Yuille, Hinton, …
- Compositionality
  - S. Geman & Bienenstock, Amit & D. Geman, Dickinson, Ettinger, S.C. Zhu & Mumford, Yuille et al., Todorovic & Ahuja, Ullman et al., Felzenswalb
- Unsupervised learning
  - Utans, Serre & Riesenhuber & Poggio, Scalzo & Piater, Lecun, Hinton, Ommer & Buhmann, Yuille et al.
- Incremental learning
  - Hinton, Krempp & Amit & Geman, Opelt & Pinz & Zisserman, Fei Fei & Fergus & Perona

⇨ unsupervised learning of hierarchical compositional shape hierarchy

Hierarchical compositional model

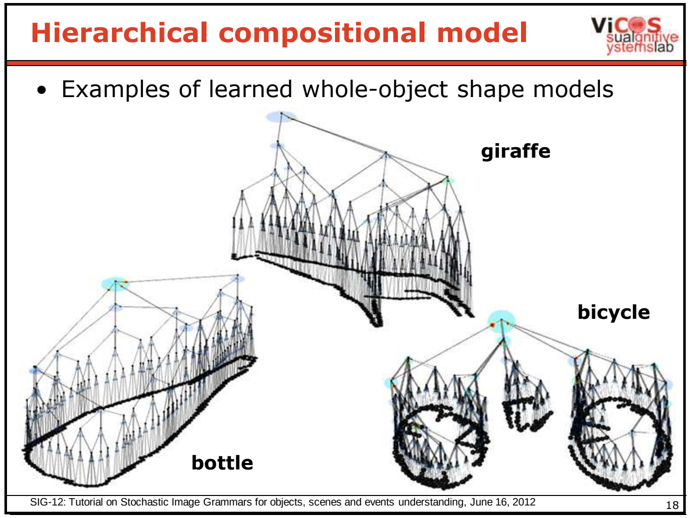
- Architecture of the hierarchical vocabulary:
  - at each *layer* the vocabulary contains a set of hierarchical deformable models called *compositions*.
  - Each composition is defined recursively, i.e. is built from compositions from the previous layer.
  - Compositions can be *grouped* (*OR-ed*) together based on their properties, e.g. geometric similarity.
  - Compositions on the first layer are simple image features (e.g. Gabor feature vectors).





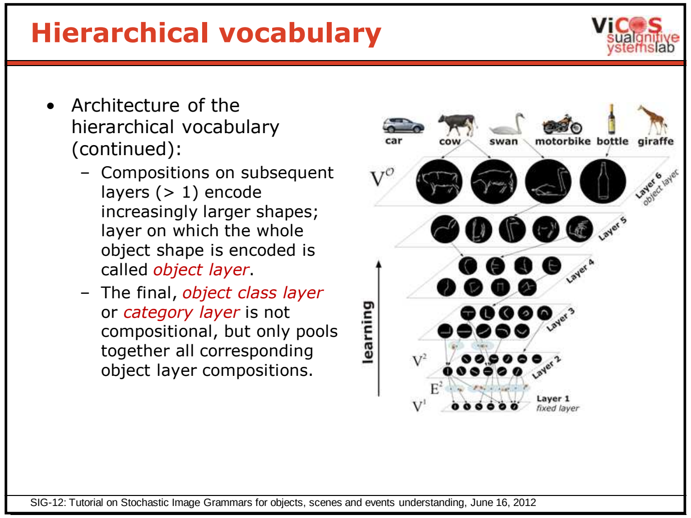
Hierarchical compositional model

- Object representation: A **hierarchical compositional shape vocabulary**
- The compositions model spatial relations among their parts



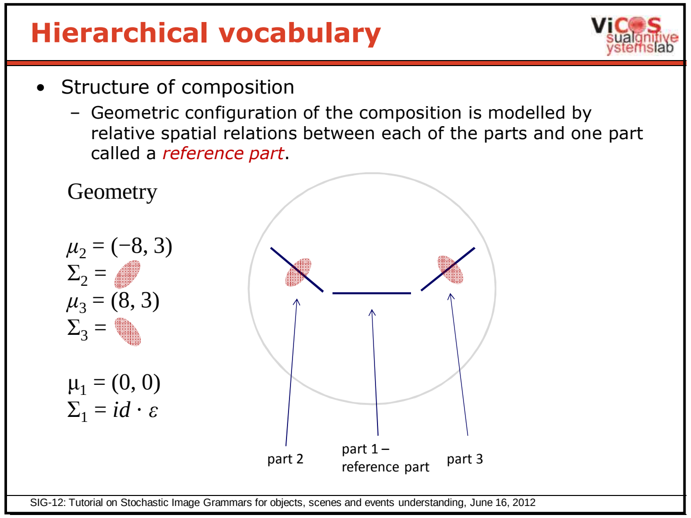
Hierarchical compositional model

- Examples of learned whole-object shape models



Hierarchical vocabulary

- Architecture of the hierarchical vocabulary (continued):
  - Compositions on subsequent layers (> 1) encode increasingly larger shapes; layer on which the whole object shape is encoded is called *object layer*.
  - The final, *object class layer* or *category layer* is not compositional, but only pools together all corresponding object layer compositions.



Hierarchical vocabulary

- Structure of composition
  - Geometric configuration of the composition is modelled by relative spatial relations between each of the parts and one part called a *reference part*.

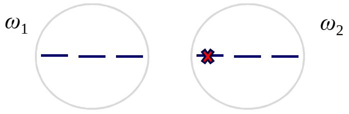
Geometry

$\mu_2 = (-8, 3)$
$\Sigma_2 =$
$\mu_3 = (8, 3)$
$\Sigma_3 =$

$\mu_1 = (0, 0)$
$\Sigma_1 = id \cdot \varepsilon$

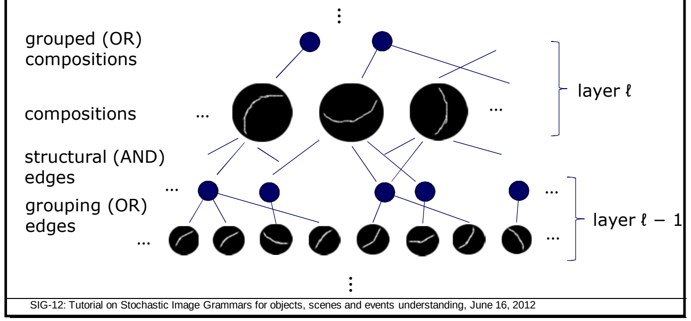
Hierarchical vocabulary

- Structure of composition
  - Geometric configuration of the composition is modelled by relative spatial relations between each of the parts and one part called a *reference part*.
  - We allow for *repulsive* (or "forbidden") parts. These are the parts that the composition cannot consist of. We need them to deal with compositions that are supersets of one another.



Hierarchical vocabulary

- We represent vocabulary as an AND-OR graph
  - Nodes of the graph represent compositions (AND) and grouped compositions (OR).
  - Edges of the graph represent relations between them: compositional relations (AND) or grouping relations (OR)



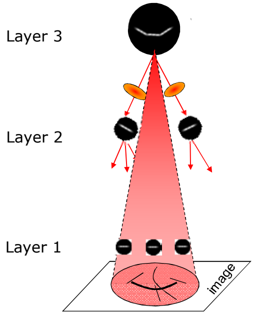
Hierarchical vocabulary – notation

Note: not to overload the notation, we will explain the inference and learning process under the assumption that we do not have OR compositions!

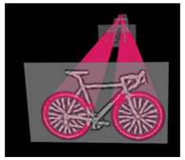
- Let Ω denote set and structure of all compositions; due to its hierarchical structure we write it as $\Omega = \Omega^1 \cup \Omega^2 \cup \ldots \cup \Omega^O \cup \Omega^C$ where $\Omega^\ell = \{\omega_i^\ell\}_i$, $i = 1,\ldots,N^\ell$, is a set of compositions at layer ℓ.
- Composition structure:
  - A composition ω^ℓ, $\ell > 1$ consists of P parts. (Note that P can be different for different compositions)
  - Geometric relation of part p relative to the reference part is modelled by 2D Gaussians and denoted by $\theta_{g\,p} = (\mu_p, \Sigma_p)$

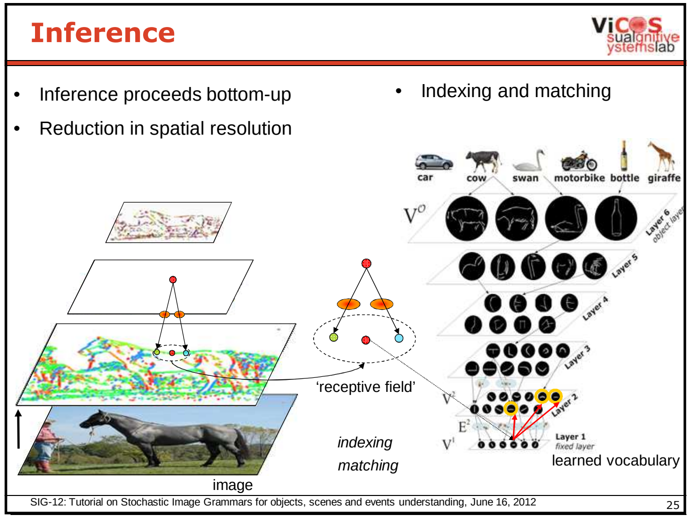
Hierarchical vocabulary

- Object representation: A **hierarchical compositional shape vocabulary**
- The compositions model spatial relations among their parts



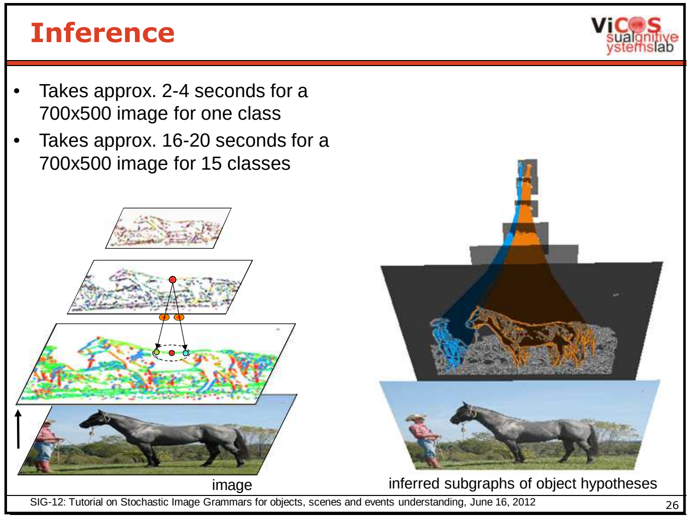
- Invariance to local deformations
- Exponential flexibility
- Robustness to clutter
- Fast inference





Inference

- Inference proceeds bottom-up
- Reduction in spatial resolution
- Indexing and matching



Inference

- Takes approx. 2-4 seconds for a 700x500 image for one class
- Takes approx. 16-20 seconds for a 700x500 image for 15 classes

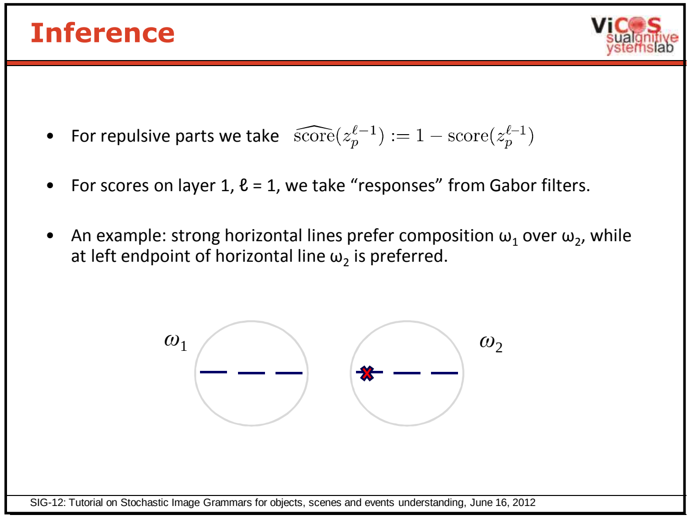
Inference

- With a given vocabulary we infer a hierarchy of *hidden states*.
- Hidden states of the 1st layer receive input from observations, states on other layers receive input only from the layer below.
- We denote hidden state on layer ℓ by $z^\ell = (\omega^\ell, x^\ell)$ where ω^ℓ is a vocabulary composition and x^ℓ is a location in the image.
- We assign to each hidden state z^ℓ, $\ell > 1$, its *score* which is computed as

$$\mathrm{score}(z^\ell) = \prod_{p=1}^{P(\omega^\ell)} \max_{z_p^{\ell-1}} \left(\widehat{\mathrm{score}}(z_p^{\ell-1}) \cdot D(x_j^{\ell-1} - x^\ell \mid \mu_j^\ell, \Sigma_j^\ell) \right)$$

- in general, $\widehat{\mathrm{score}}(z_p^{\ell-1}) := \mathrm{score}(z_p^{\ell-1})$, except for repulsive parts
- D represents a deformation score function and we define it as

$$D(x \mid \mu, \Sigma) = \exp\left(-0.5 \cdot (x-\mu)^T \Sigma^{-1} (x-\mu)\right)$$



Inference

- For repulsive parts we take $\widehat{\mathrm{score}}(z_p^{\ell-1}) := 1 - \mathrm{score}(z_p^{\ell-1})$
- For scores on layer 1, ℓ = 1, we take “responses” from Gabor filters.
- An example: strong horizontal lines prefer composition ω_1 over ω_2, while at left endpoint of horizontal line ω_2 is preferred.

Inference graph

- In the inference process we build the inference graph G = (Z, E). Nodes $z^\ell \in Z$ are hypotheses (hidden states). Like vocabulary, G has also hierarchical structure and we write

$$Z = Z^0 \cup Z^1 \cup \ldots \cup Z^O .$$

- Computation of G is recursive.
  - Assume that hypotheses $Z^{\ell-1}$ have been computed.
  - To get Z^ℓ we visit each hypothesis $z^{\ell-1} = (\omega^{\ell-1}, x^{\ell-1})$ and find all compositions $R(\omega^{\ell-1})$ having $\omega^{\ell-1}$ for their reference part.
  - For each composition $\omega^\ell \in R(\omega^{\ell-1})$ we make a hypothesis $z^\ell = (\omega^\ell, x^\ell)$, $x^\ell = x^{\ell-1}$, and calculate its score.
  - We perform reduction in spatial resolution, i.e., locations x^ℓ are down-sampled by factor $\rho^\ell \leq 1$, (usually we take $\rho^\ell = 0.5$).
    - We bring far-away (location-wise) hidden states closer and indirectly (through learning) we keep scales of the Gaussians approximately the same over all layers (faster inference).

Inference graph

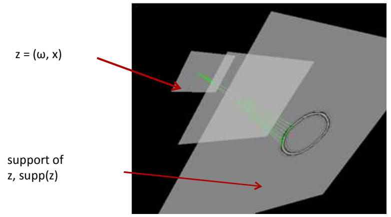
- Computation of G, continued:
  - If the score(z^ℓ) is greater than a threshold τ^ℓ, then we add z^ℓ to Z^ℓ.
  - we add edges from z^ℓ to nodes in $Z^{\ell-1}$ yielding "max" value in score calculation.

$$\mathrm{score}(z^\ell) = \prod_{p=1}^{P(\omega^\ell)} \max_{z_p^{\ell-1}} \left(\widehat{\mathrm{score}}(z_p^{\ell-1}) \cdot D(x_j^{\ell-1} - x^\ell \mid \mu_j^\ell, \Sigma_j^\ell) \right)$$

  - Note also: At the same position x^ℓ we allow only one state with a particular composition. If we get two states $z = (\omega^\ell, x^\ell)$ and $z = (\omega^\ell, x^\ell)$ with $\omega'^\ell = \omega^\ell$ and $x'^\ell = x^\ell$, then we keep the one with larger score. (This can happen due to spatial contraction.)

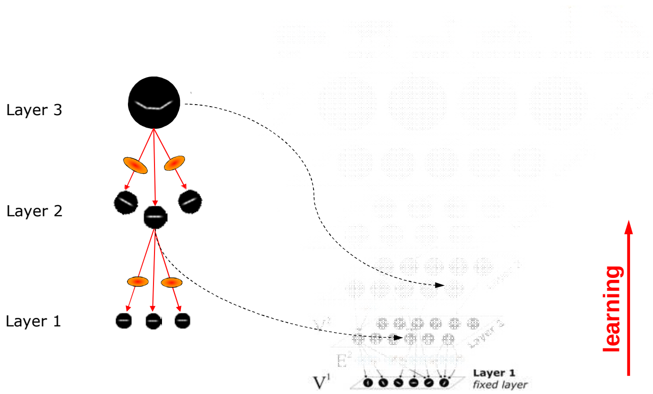
Inference graph

- Support of the graph:
  - Nodes which can be reached from z^ℓ via edges added in the inference process form a subgraph in G we denote by $G(z^\ell)$.
  - Layer 1 compositions of $G(z^\ell)$ are called a support of hypothesis z^ℓ and is denoted by $\mathrm{supp}(z^\ell)$.
  - Example: Graph G(z) of 3rd layer detection z of composition ω =



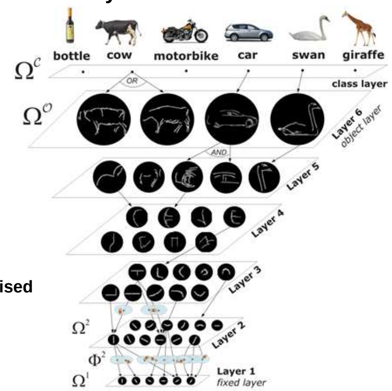
Learning

- **Bottom-up**



Learning

- **Learning the hierarchical vocabulary**
  - Learn the **number** of compositions at each layer
  - Learn the **structure** of each composition (the number of parts and the parameters of the distributions)

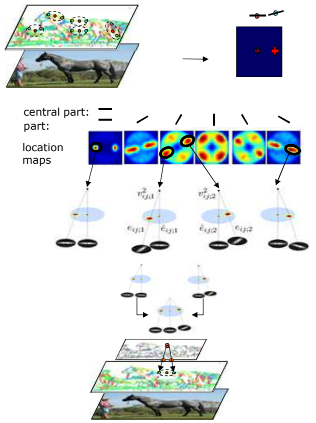


Learning of **structure** is **unsupervised**

Learning of **classes** is **supervised**

Learning

- Learning is performed bottom-up, by combining simple features into increasingly more complex compositions
- Learning steps
  - •Learning pair-wise spatial relations
  - •Cluster into 'duplet' compositions
  - •Learn higher-order compositions by tracking frequent co-occurrences of duplets
- Only statistically most significant compositions define a certain layer
- We learn layer by layer

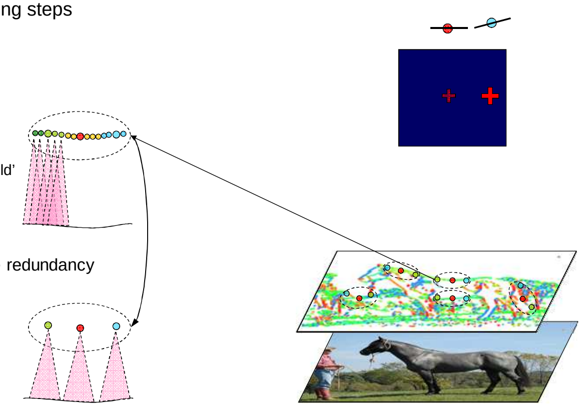


Learning

- Learning steps

'receptive field'

→ reduce redundancy



Learning

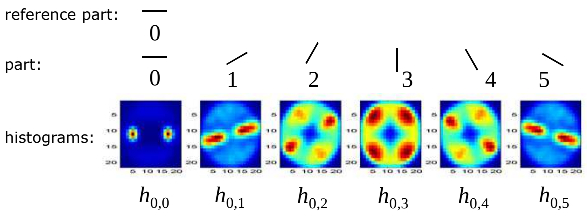
- Learning of the structure consists from:
  1. Learning spatial correlations between parts
  2. Learning compositions of parts
  3. Learning the parameters
- Assumptions for learning layer $\ell > 1$:
  - For each training image I we have the inference graph $G = (Z^1 \cup Z^2 \cup \dots \cup Z^{\ell-1}, E)$ built up to layer $\ell - 1$.

Learning spatial correlations between parts

- We learn geometry distributions between all possible pairs of compositions from layer ℓ − 1.
  - Let $h^{\ell}_{i,j} : [-r^{\ell}, r^{\ell}] \to \mathbb{R}$ be a histogram of occurrences of a composition $\omega_j^{\ell-1}$ relative to composition $\omega_i^{\ell-1}$ (which plays a role of a reference)
  - During training, $h^{\ell}_{i,j}$ is updated at $x'^{\ell-1} - x^{\ell-1}$ for each pair of hidden states $(z^{\ell-1}, z'^{\ell-1})$ where $z^{\ell-1} = (\omega_i^{\ell-1}, x^{\ell-1})$ and $z'^{\ell-1} = (\omega_j^{\ell-1}, x'^{\ell-1})$ such that:
    - $|x^{\ell-1} - x'^{\ell-1}| \le r^{\ell}$
    - supports of $z^{\ell-1}$ and $z'^{\ell-1}$ are "sufficiently" disjoint (overlap of their supports is small).
- Histograms are updated for all inference graphs of training images

Learning spatial correlations between parts

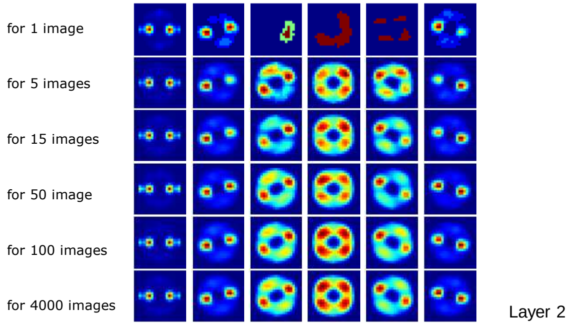
- An example:



(for 7000 images, $r = 8$)

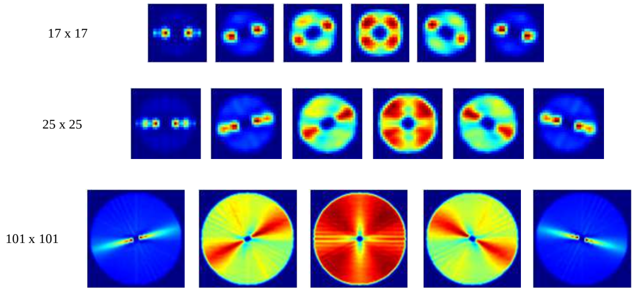
Learning spatial correlations between parts

- 'Convergence' of distributions



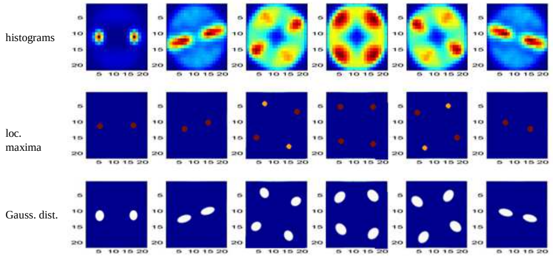
Learning spatial correlations between parts

- Statistical confirmation for highly correlated parts in small neighborhoods (⇒ local is better)



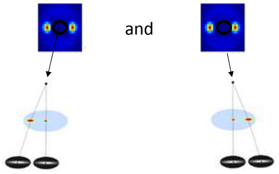
Learning spatial correlations between parts

- For each histogram, local maxima are determined
- For each local maximum we estimate the mean µ and variance Σ. by fitting a Gaussian distribution around it.



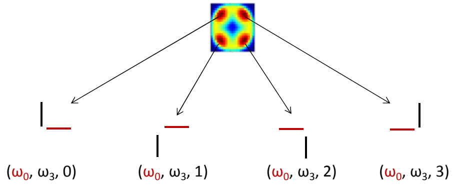
Learning spatial correlations between parts

- Local maxima define two-part compositions called duplets. In this way we “sparsify” (or “discretize”) geometric positions between two compositions.
- Example 1:
  - There are just two “statistically significant” positions of composition ω_0 = – relative to reference composition ω_0 = – , i.e. there are two duplets with reference composition ω_0 and the other composition ω_0.



Learning spatial correlations between parts

- Notation: duplet with composition ω relative to reference part ω_R at position i is denoted by (ω_R, ω, i)
  Example 2:
  - There are four significant positions of composition ω_3 = | relative to reference composition ω_0 = –, i.e. there are four duplets with reference composition ω_0 and the other composition ω_3.


$(\omega_0, \omega_3, 0)$
$(\omega_0, \omega_3, 1)$
$(\omega_0, \omega_3, 2)$
$(\omega_0, \omega_3, 3)$

Learning compositions of parts

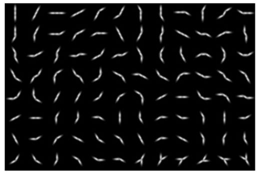
- We find a set of compositions: each composition is a set of frequently co-occurring duplets (two-part compositions):

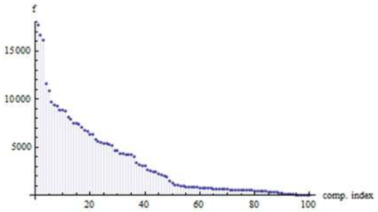
1. Inference is performed on training images with all obtained duplets as described in the inference section.
2. For each training image neighborhood I_k we find a set of disjoint duplets with the same reference part ω_R, $(\omega_R, \omega_p, i_1)$, $(\omega_R, \omega_q, i_1)$, … which best explain I_k.

   This set forms a composition with:
   - reference part ω_R,
   - subparts ω_p, ω_q, … , and
   - geometric parameters (µ, Σ) corresponding to the estimated Gaussians of duplets $(\omega_R, \omega_p, i_1)$, $(\omega_R, \omega_q, i_1)$, …

Learning compositions of parts

- For each obtained composition ω we update its “count” value f(ω). It is taken to be the sum of its scores, i.e. each time we get ω, we set f(ω) = f(ω) + score (x_k, ω).
- Example of the first 100 second layer compositions sorted by decreasing value of f:



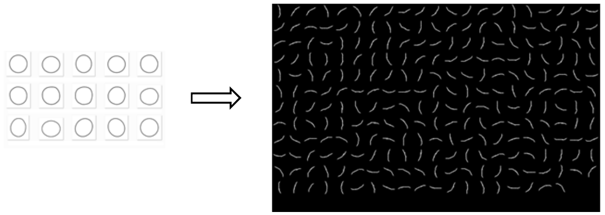


Object layer learning

- For learning the last, object layer, we use a similar approach as for lower layers, but additionally:
  - For each training image (receptive field) we produce many (redundant) compositions with different number of subparts and different combinations of subparts.
  - We validate these compositions on validation image set and keep only those which have good performance: low false negative/false positive ratio.

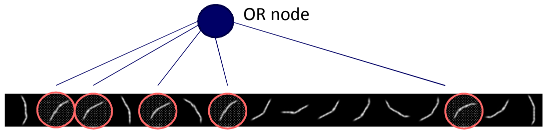
Learning compositions of parts

- On higher layers we can easily get an “explosion” of parts due to many possible combinations of compositions.
- Example: For the set of “circles” we obtain the following 3rd layer:



Learning compositions of parts

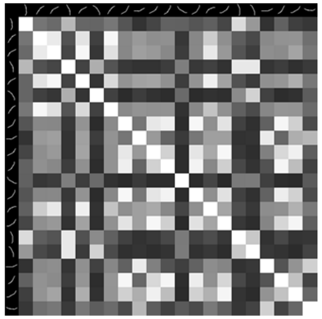
- To reduce the number of parts we:
  - Merge similar compositions and giving them the same label (adding “OR” nodes to the vocabulary)



  - Reduce redundancy by selecting a subset of all (merged) compositions which already describe the training set sufficiently well (e.g., in this way we remove parts modelling texture).

Merging compositions of parts

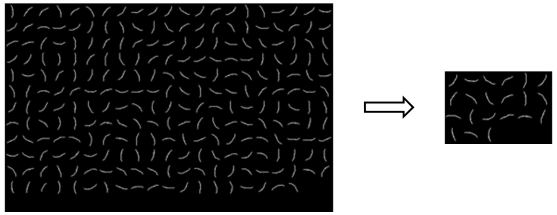
- Example: Distances between layer 3 compositions.



(White color: distance = 0)

Reducing the set of compositions

- Further, we select only a subset of compositions which approximately maintains the description power of the full set.



Reducing the set of compositions

- Formally, we set the following optimization problem:
  - We have vocabulary built up to layer ℓ, and Ω_0^ℓ is (current) set of compositions on layer ℓ.
  - For image I_k define

$$\text{loss}_k^\ell(\Omega) := 1 - \frac{|\text{supp} Z_k^\ell(\Omega)|}{|\text{supp} Z_k^{\ell-1}(\Omega^{\ell-1})|}$$

which measures how well image I_k is covered, relative to the covering with layer $\ell - 1$, if on layer ℓ we only take compositions $\Omega \subseteq \Omega^\ell$.

Learning thresholds

- Detections of compositions in the inference process are accepted if their scores are above a threshold.
- Thresholds are determined for each particular composition and are based on the performance of the object layer detections.
- For each object layer composition we learn a 2-class SVM classifier which accepts or rejects a detection:
  - For each detection $z^O = (x^O, \omega^O)$ we make a vector composed of its score and scores of its subpart detections $z_1^{O-1}, \ldots, z_P^{O-1}$: $(\text{score}(z^{O-1}), \text{score}(z_1^{O-1}), \ldots, \text{score}(z_P^{O-1}))$
  - SVM classifier is trained on the vectors obtained from true positive and false positive detections on validation images.

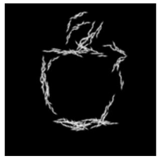
Learning thresholds

- On other layers we learn the thresholds in a way that nothing is lost with respect to the accuracy of object detection while at the same time optimizing for the efficiency of inference.
- For each composition ω^ℓ we find the smallest score it produces in any of the parse graphs of positive object detections over all train images I_k. Threshold for its score is then:

$$\tau_{\omega^\ell} = \min_{k} \min_{(\omega^\ell, x^\ell) \in \mathcal{T}_{I_k}(z^{\mathcal{O}})} \mathrm{score}(\omega^\ell, x^\ell)$$

Shape consistency and deformations

- Due to spatial deformations we allow for each subpart (Gaussian "distributions" (μ_p, Σ_p)), the support shape of detections on higher layers (5) and particularly on object layer can significantly **deviate** from the shape that composition represented during the learning phase.
- For example, if we "sample" subparts of each composition representing an apple according to (μ_p, Σ_p) recursively, we get:

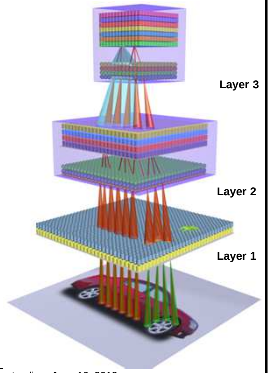


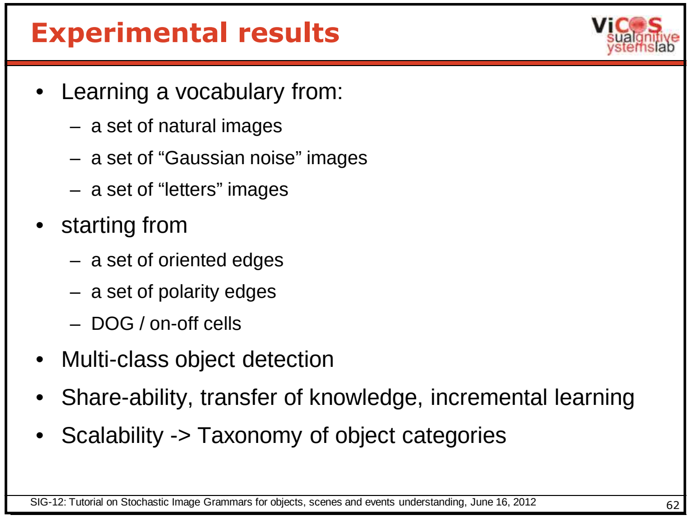
Shape consistency and deformations

- Therefore we keep track of **average shapes (= average supports)** of compositions obtained in the learning process.
- In the inference process we calculate distance of the inferred shape to the learned average shape and use it as an additional "score" which can be used to accept or reject an object layer detection.
- We add this **shape consistency score** to the vector of the SVM classifier.

Summary: Compositional shape hierarchy

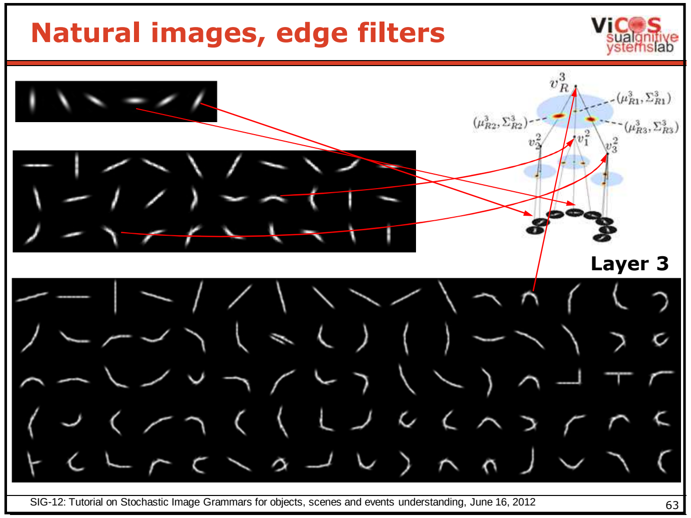
- A Computational Model for Learning a Multi-Level Compositional Representation of Visual Structure
  - Computational plausibility
    - Hierarchical representation
    - Compositionality *(parts composed of parts)*
    - Indexing & matching recognition scheme
  - Statistics driven learning (unsupervised learning)
  - Fast, incremental (continuous) learning





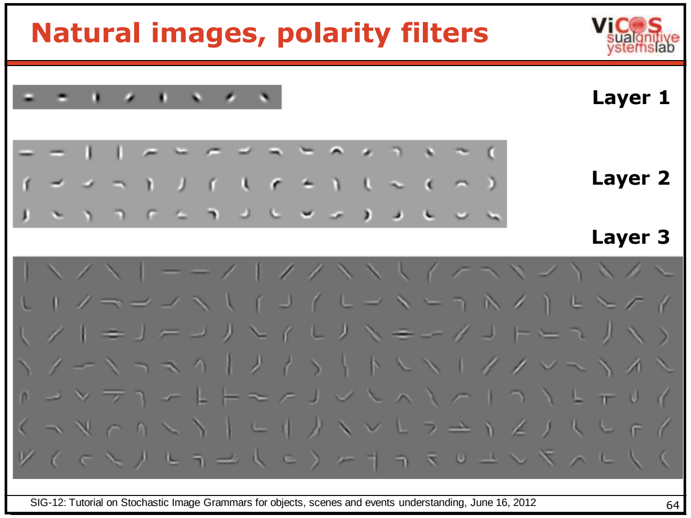
Experimental results

- Learning a vocabulary from:
  - a set of natural images
  - a set of “Gaussian noise” images
  - a set of “letters” images
- starting from
  - a set of oriented edges
  - a set of polarity edges
  - DOG / on-off cells
- Multi-class object detection
- Share-ability, transfer of knowledge, incremental learning
- Scalability -> Taxonomy of object categories

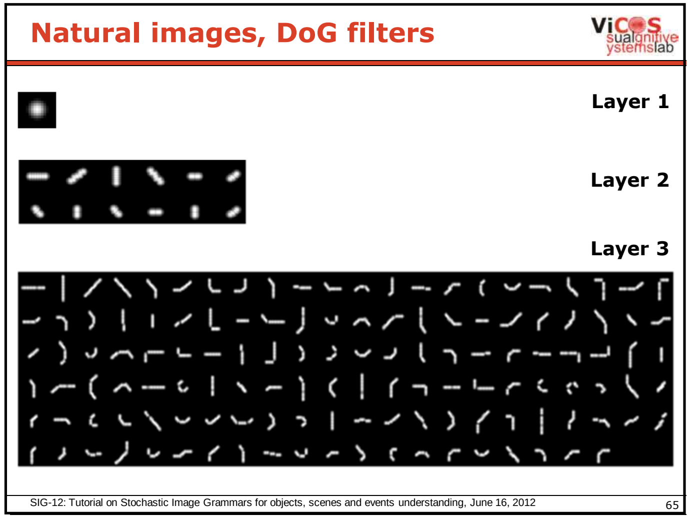


Natural images, edge filters

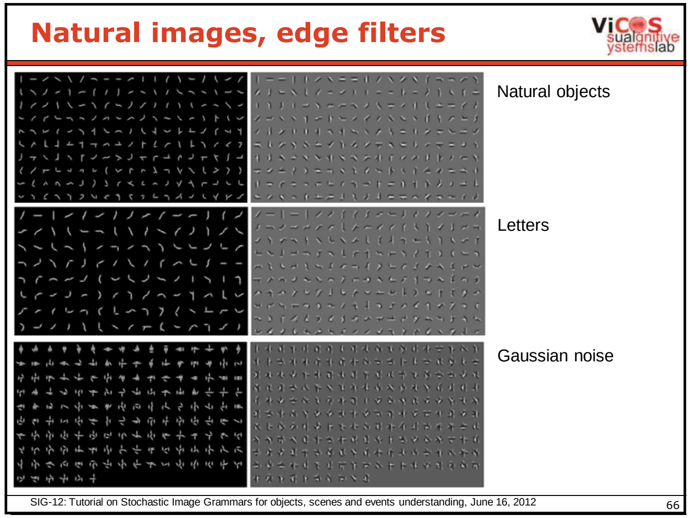
v_R^3
$(\mu_{R1}^3, \Sigma_{R1}^3)$
$(\mu_{R2}^3, \Sigma_{R2}^3)$
$(\mu_{R3}^3, \Sigma_{R3}^3)$
v_2^2
v_1^2
v_3^2
Layer 3



Natural images, polarity filters



Natural images, DoG filters

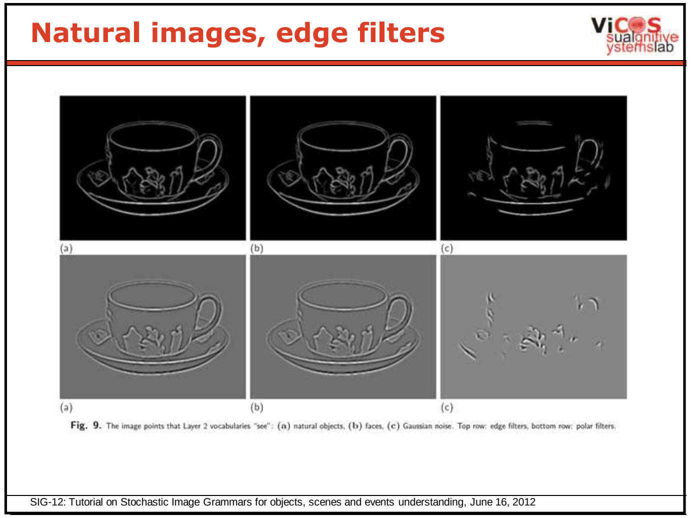


Natural images, edge filters

Natural objects

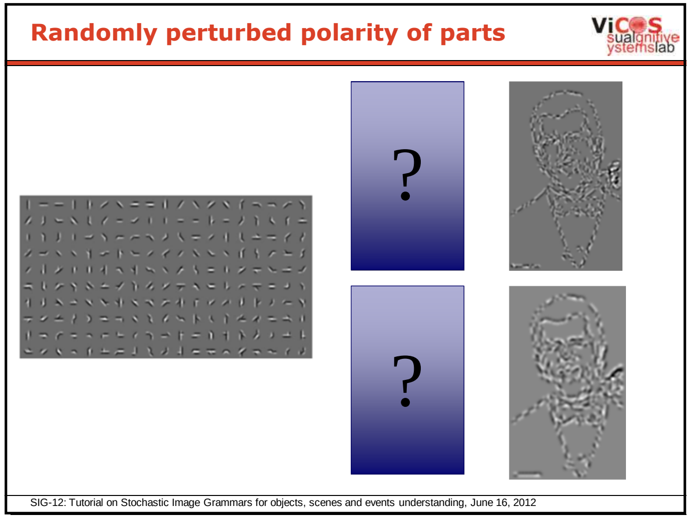
Letters

Gaussian noise

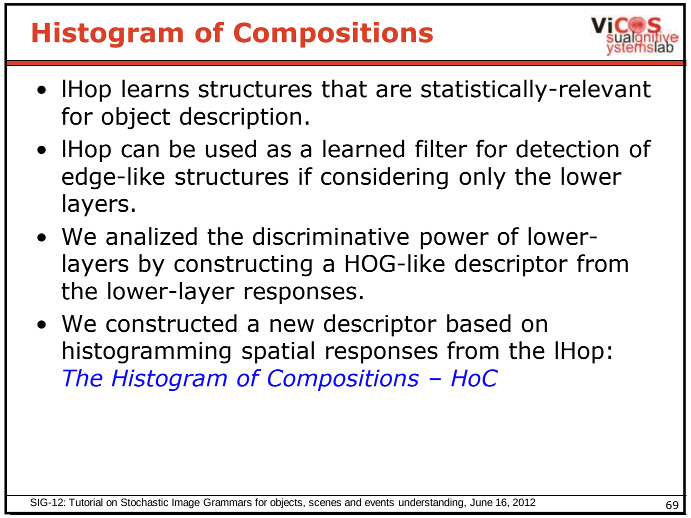


Natural images, edge filters

Fig. 9. The image points that Layer 2 vocabularies "see": (**a**) natural objects, (**b**) faces, (**c**) Gaussian noise. Top row: edge filters, bottom row: polar filters.

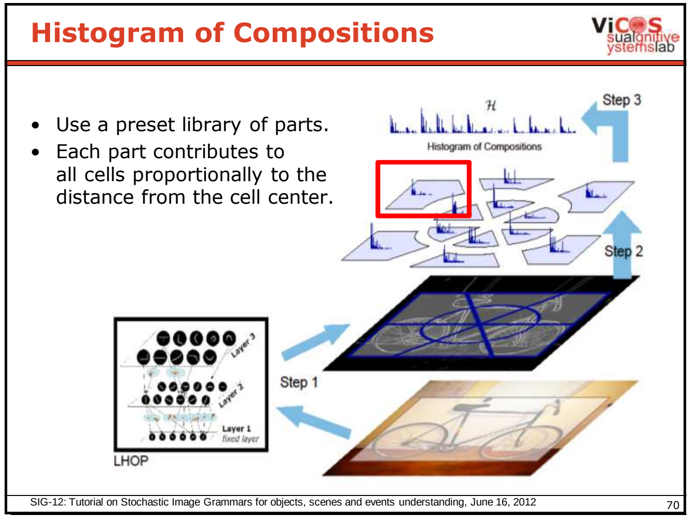


Randomly perturbed polarity of parts



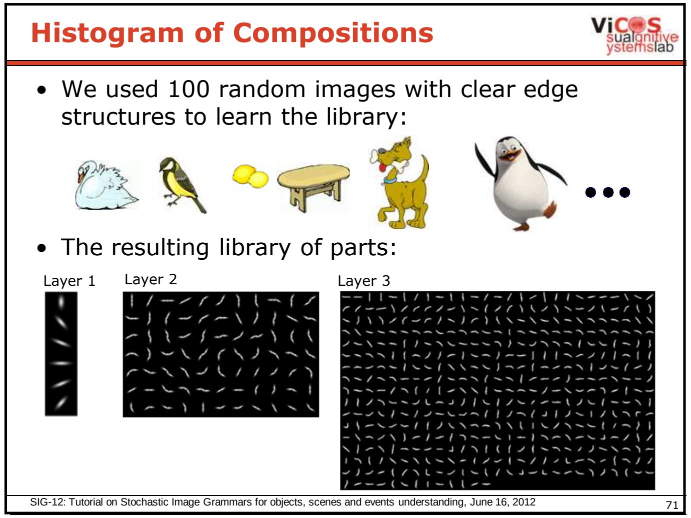
Histogram of Compositions

- lHop learns structures that are statistically-relevant for object description.
- lHop can be used as a learned filter for detection of edge-like structures if considering only the lower layers.
- We analized the discriminative power of lower-layers by constructing a HOG-like descriptor from the lower-layer responses.
- We constructed a new descriptor based on histogramming spatial responses from the lHop: *The Histogram of Compositions – HoC*



Histogram of Compositions

- Use a preset library of parts.
- Each part contributes to all cells proportionally to the distance from the cell center.



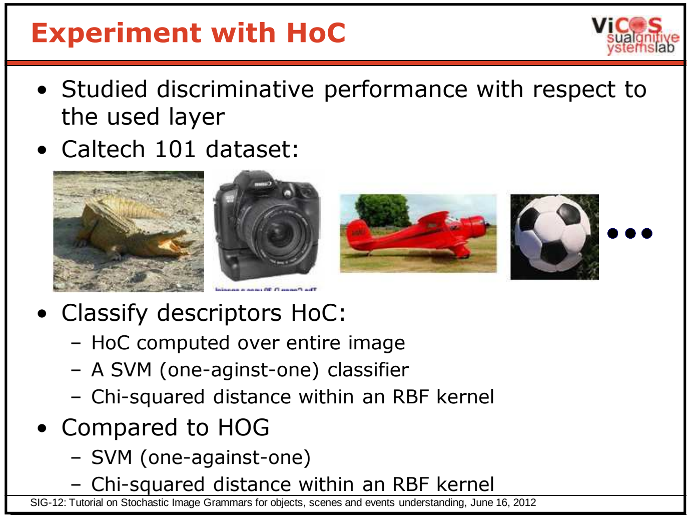
Histogram of Compositions

- We used 100 random images with clear edge structures to learn the library:

•••

- The resulting library of parts:

Layer 1 Layer 2 Layer 3

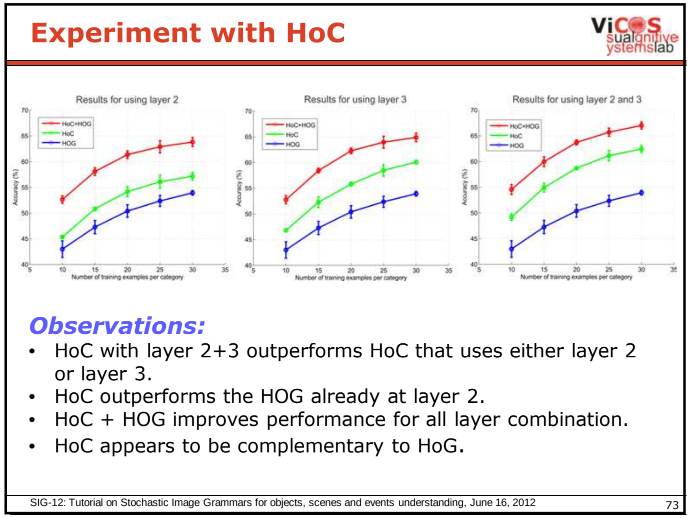


Experiment with HoC

- Studied discriminative performance with respect to the used layer
- Caltech 101 dataset:

•••

- Classify descriptors HoC:
  - HoC computed over entire image
  - A SVM (one-aginst-one) classifier
  - Chi-squared distance within an RBF kernel
- Compared to HOG
  - SVM (one-against-one)
  - Chi-squared distance within an RBF kernel



Experiment with HoC

Observations:

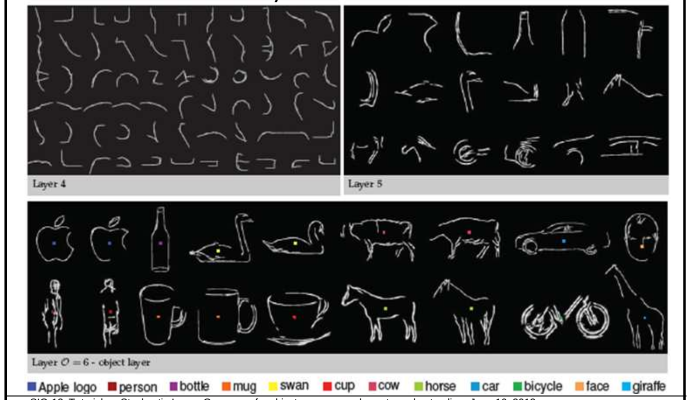
- HoC with layer 2+3 outperforms HoC that uses either layer 2 or layer 3.
- HoC outperforms the HOG already at layer 2.
- HoC + HOG improves performance for all layer combination.
- HoC appears to be complementary to HoG.

Multi-class learning and detection

- Learning a vocabulary from simple Gabor feature to whole-object class shapes
- Learning a representation of 15 object categories (cup, mug, bottle, cow, giraffe, swan, horse, person, face car_front, car_rear, car_side, motorbike, bicycle, apple logo)
- Learning of the first 3 layers on natural images (or jointly on images of all classes), while learning the higher layers *incrementally* (one class after another)

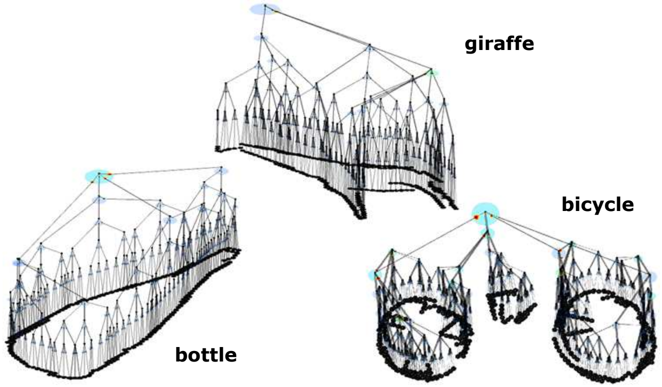
Multi-class learning and detection

- Learned vocabulary

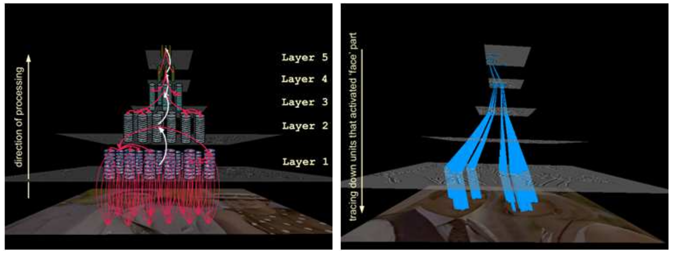


Multi-class learning and detection

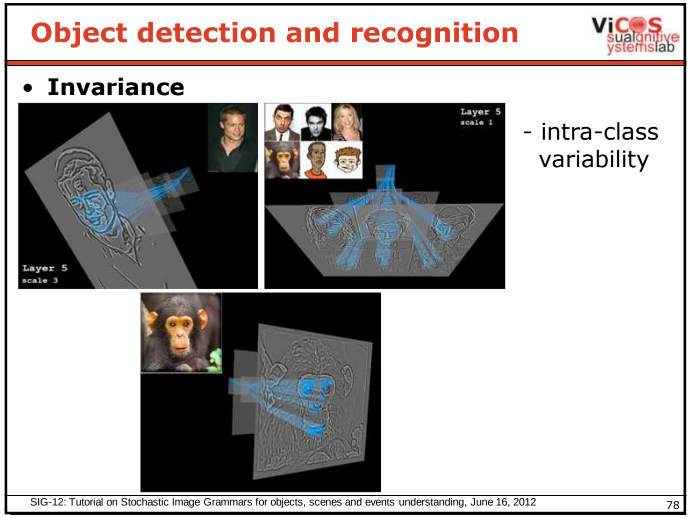
- Examples of learned whole-object shape models



Detection



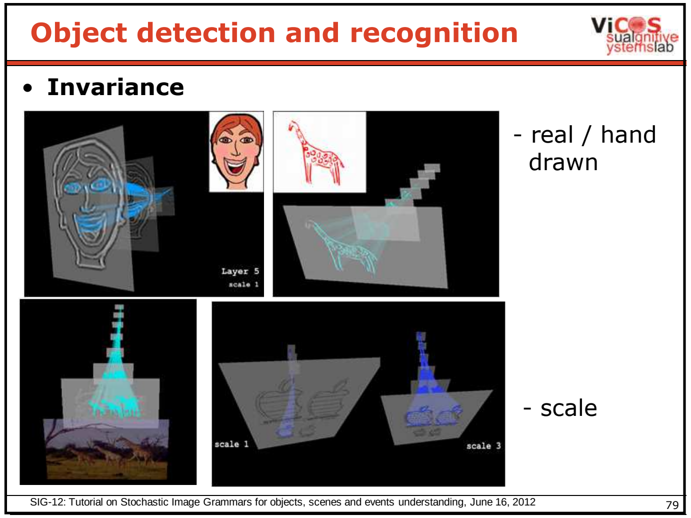
Inference proceeds bottom-up. Active parts can easily be “traced” down to the image.



Object detection and recognition

- **Invariance**

- intra-class variability

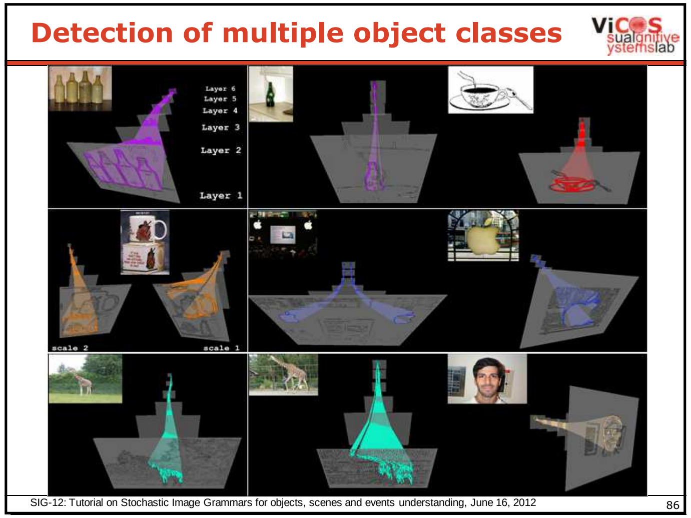


Object detection and recognition

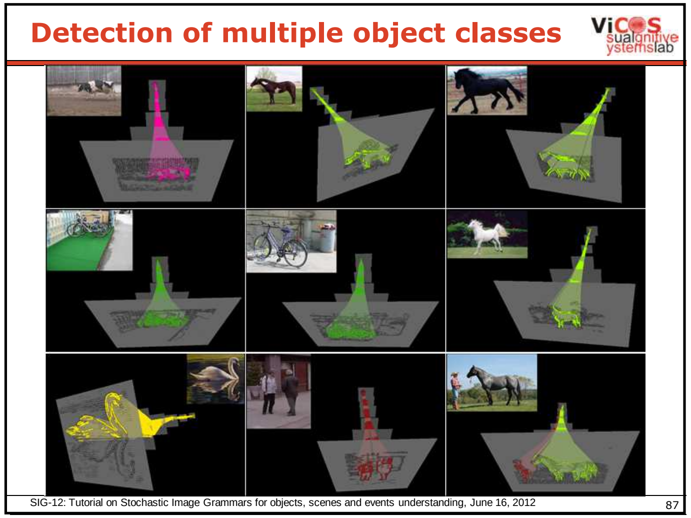
- **Invariance**

- real / hand drawn

- scale

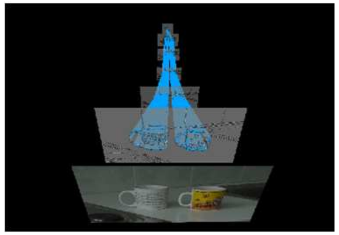


Detection of multiple object classes

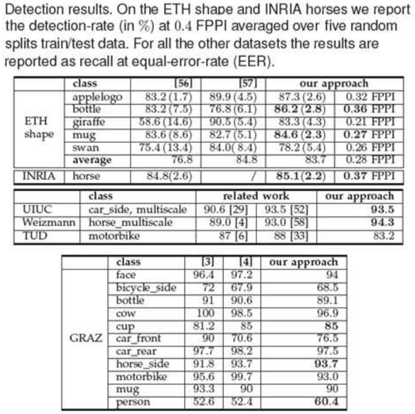


Detection of multiple object classes

Detection of object classes, cups



Detection



Detection results. On the ETH shape and INRIA horses we report the detection-rate (in %) at 0.4 FPPI averaged over five random splits train/test data. For all the other datasets the results are reported as recall at equal-error-rate (EER).

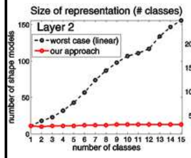
	class	[56]	[57]	our approach	
ETH shape	applelogo	83.2 (1.7)	89.9 (4.5)	87.3 (2.6)	0.32 FPPI
	bottle	83.2 (7.5)	76.8 (6.1)	**86.2 (2.8)**	**0.36 FPPI**
	giraffe	58.6 (14.6)	90.5 (5.4)	83.3 (4.3)	0.21 FPPI
	mug	83.6 (8.6)	82.7 (5.1)	**84.6 (2.3)**	**0.27 FPPI**
	swan	75.4 (13.4)	84.0(8.4)	78.2 (5.4)	0.26 FPPI
	average	76.8	84.8	83.7	0.28 FPPI
INRIA	horse	84.8(2.6)	/	**85.1(2.2)**	**0.37 FPPI**

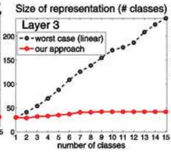
	class	related work		our approach
UIUC	car_side, multiscale	90.6 [29]	93.5 [52]	**93.5**
Weizmann	horse_multiscale	89.0 [4]	93.0 [58]	**94.3**
TUD	motorbike	87 [6]	88 [33]	83.2

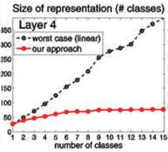
	class	[3]	[4]	our approach
GRAZ	face	96.4	97.2	94
	bicycle_side	72	67.9	68.5
	bottle	91	90.6	89.1
	cow	100	98.5	96.9
	cup	81.2	85	**85**
	car_front	90	70.6	76.5
	car_rear	97.7	98.2	97.5
	horse_side	91.8	93.7	**93.7**
	motorbike	95.6	99.7	93.0
	mug	93.3	90	90
	person	52.6	52.4	**60.4**

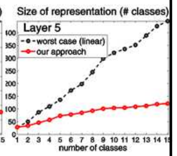
Size of the vocabulary

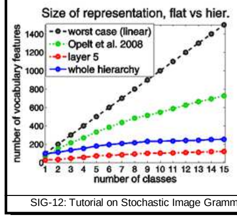
- **Size of the vocabulary** as a function of the number of learned class

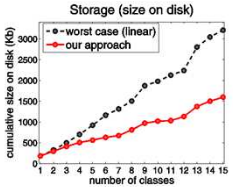








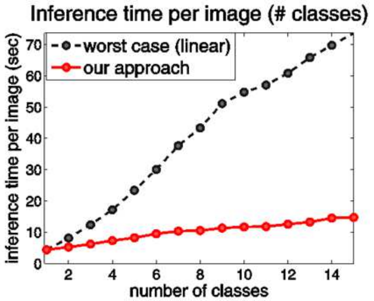




- Sublinear scaling
- Scales better than Opelt et al.
- Only 1.6Mb to store a 15-class vocabulary on disk

Inference time

- **Inference time** (average per image) as a function of the number of learned class



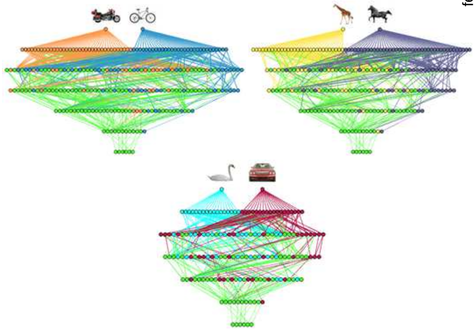
Hardware information:

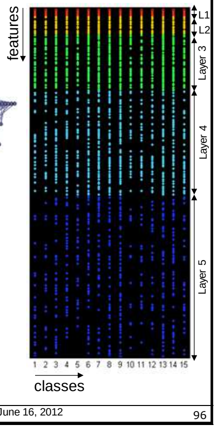
- Intel Xeon-4 CPU 2:66 Ghz computer (one core used)
- implemented in C++

- Only 16 seconds per image (approx. 500x700) for 15-class object detection

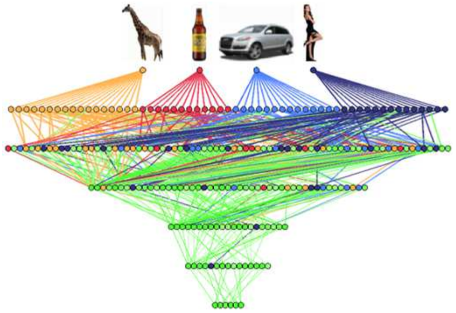
Sharing of features

- **Sharing** of features between classes





Sharing of features

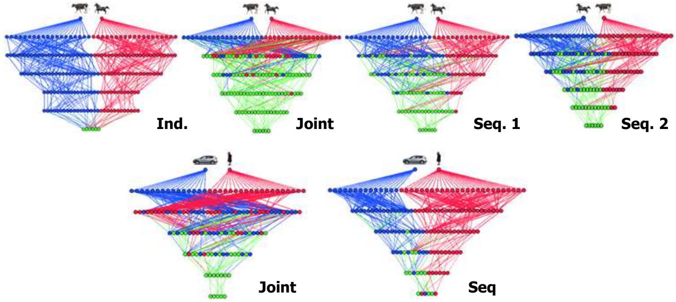


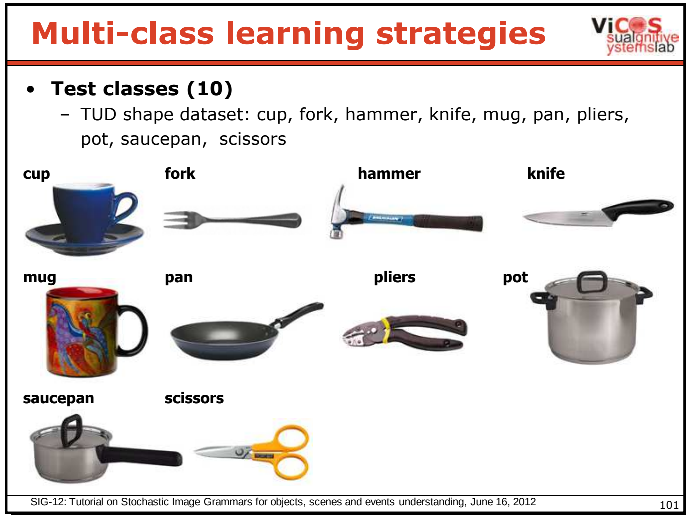
Multi-class learning strategies

- We evaluate 3 different strategies for learning a hierarchical multi-class object vocabulary for object detection:
  - **independent**
  - **joint**
  - **sequential training**
- We show that sequential learning of object classes attains the best tradeoff between the complexity of learning and detection and the accuracy of performance

Multi-class learning strategies

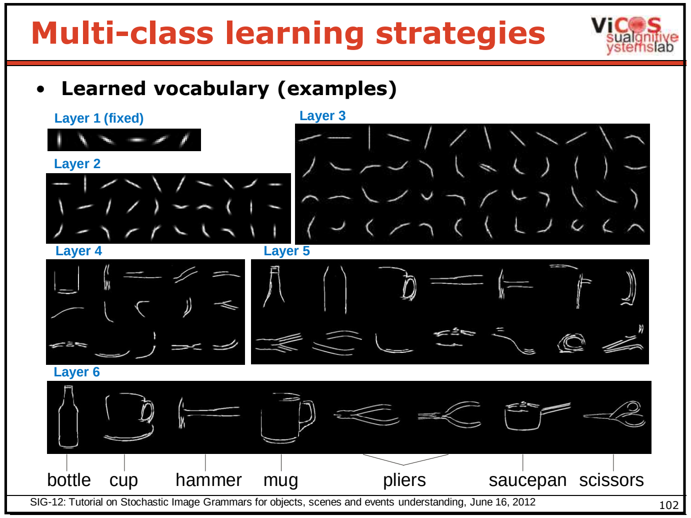
- **Feature sharing** among similar and dissimilar classes
  - Joint achieves the best sharing of features. Sequential is comparable.
  - Sharing is also present for visually dissimilar objects (lower layers)





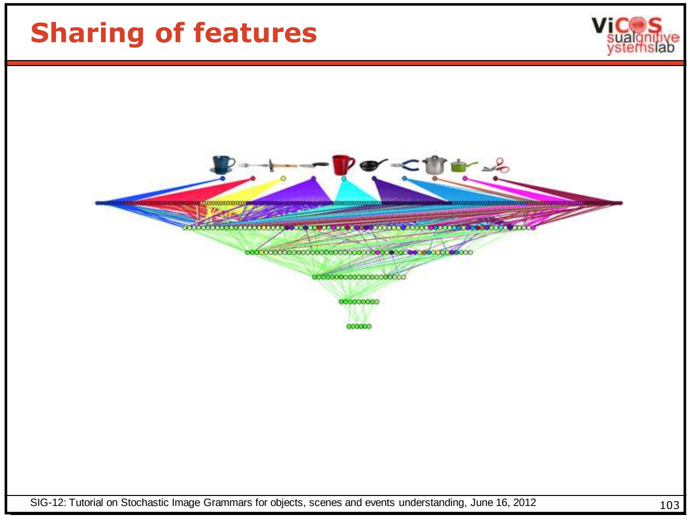
Multi-class learning strategies

- **Test classes (10)**
  - TUD shape dataset: cup, fork, hammer, knife, mug, pan, pliers, pot, saucepan, scissors

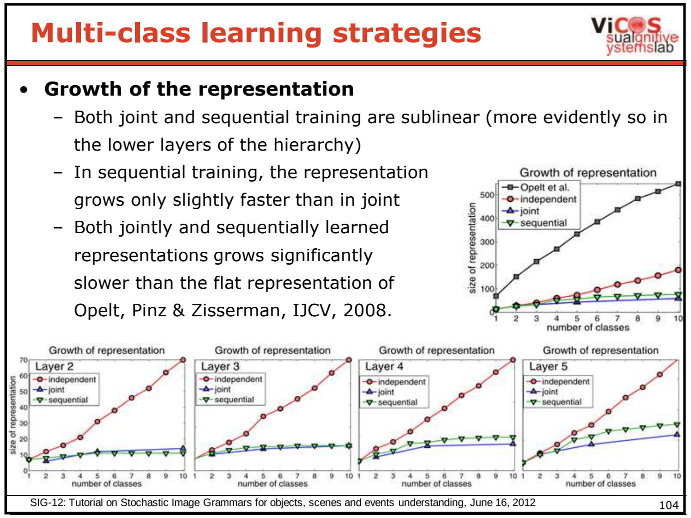


Multi-class learning strategies

- **Learned vocabulary (examples)**



Sharing of features

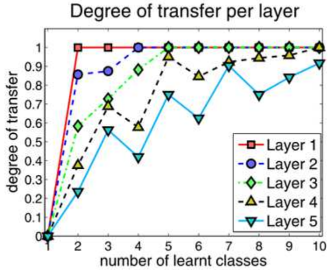


Multi-class learning strategies

- **Growth of the representation**
  - Both joint and sequential training are sublinear (more evidently so in the lower layers of the hierarchy)
  - In sequential training, the representation grows only slightly faster than in joint
  - Both jointly and sequentially learned representations grows significantly slower than the flat representation of Opelt, Pinz & Zisserman, IJCV, 2008.

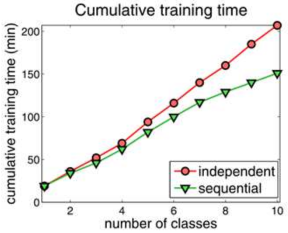
Transfer of features

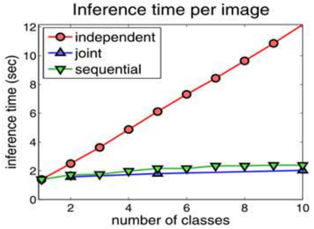
- **Transfer** of features in incremental learning



Multi-class learning strategies

- **Complexity of learning and inference**
  - Due to re-use of features, sequential training runs faster when learning each novel class (up to a constant time)
  - Inference time is best for joint, but only slightly worse for sequential training



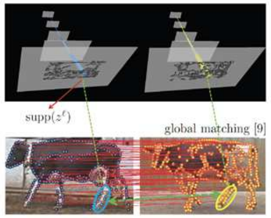


Deformations and articulations

- Ability of a composition to allow for deformations is desirable and crucial for the robustness of the algorithm. To some extent we are able to code the variations due to spatial deformation parameters (μ, Σ), but we can go further.
- The idea is to "OR" those compositions which represent some functional parts (e.g. legs of cows, necks of swans, etc.)
- We choose to do such functional OR-ing based on global matching of train images (we could also use correspondences given by motion, ...).

Shape consistency and deformations

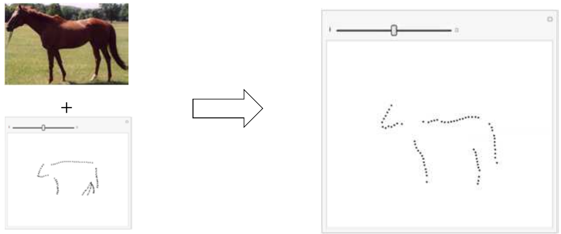
- Example: putting two compositions (blue and yellow) representing a leg into correspondence by global matching of two cows.



Two compositions are matched, if the global matching maps supports of the two compositions one to another (significant portion of them).

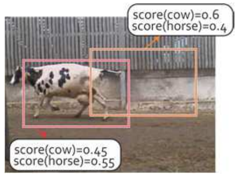
Transfer of deformations

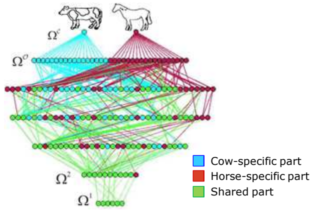
- Transfer of deformations to novel classes:
  - Example: transfer of variation of cow parts to one horse training image



Adding discriminative power to lHOP

- Part sharing causes problems when differentiating between:
  - Visually-similar categories
  - Category and a visually-similar structure on the background



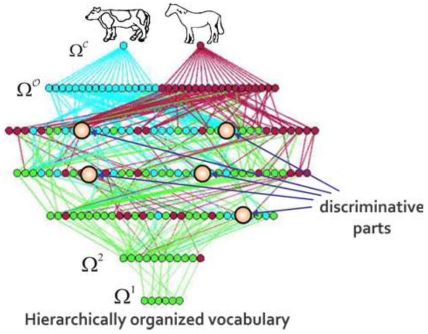


(reproduced from:Fidler et al., NIPS2009)

- Observation:
  - Visually-similar categories share many parts
  - Visually-similar categories differ in a small subset of parts

Adding discriminative power

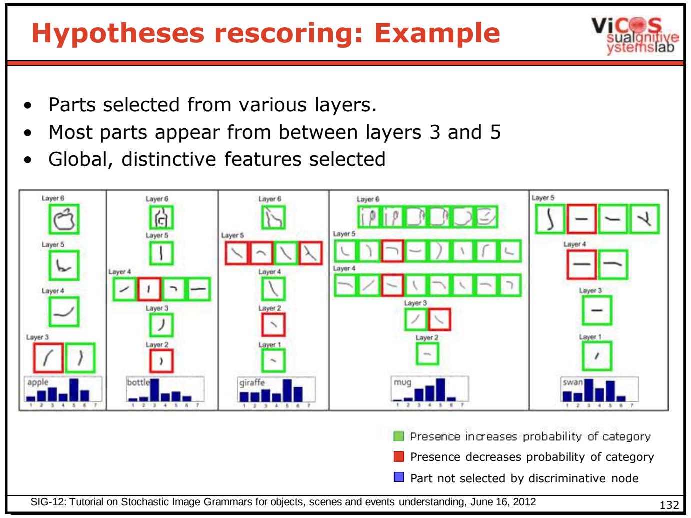
- Goal: Identify the subset of discriminative nodes to improve discrimination



Hypotheses rescoring

- Experiment: Discrimination between category and similar „background" structures (Lhop hypotheses rescoring).
- ETHZ dataset with 5 categories:

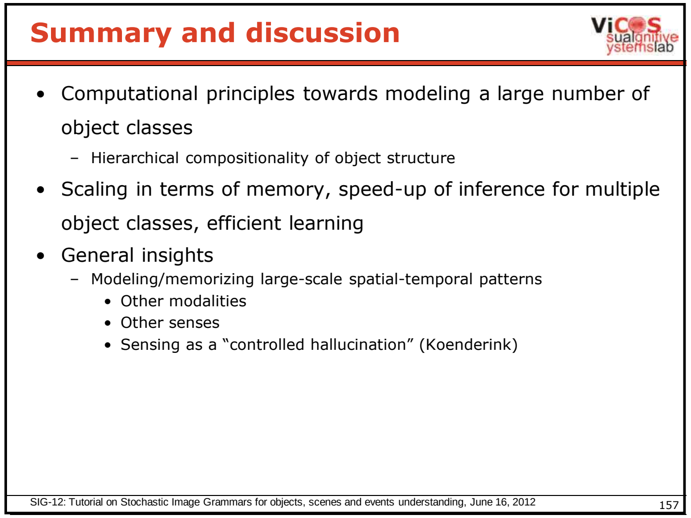
- Standard setup:
  - Half images of category for training and half + images of remaining categories for testing, (5 random splits).
- LHop trained from training images:
  - Hierarchy with 7 layers.
  - Average vocabulary consisted of 525 parts.
- dLHop trained from lHop detections on training images:
  - Lhop detections on five scales.
  - Detections that overlaped with GT by at least 40% taken as positive examples, other as negative (background).
  - On average 9 nodes per category automatically selected (~17% of all nodes).



Hypotheses rescoring: Example

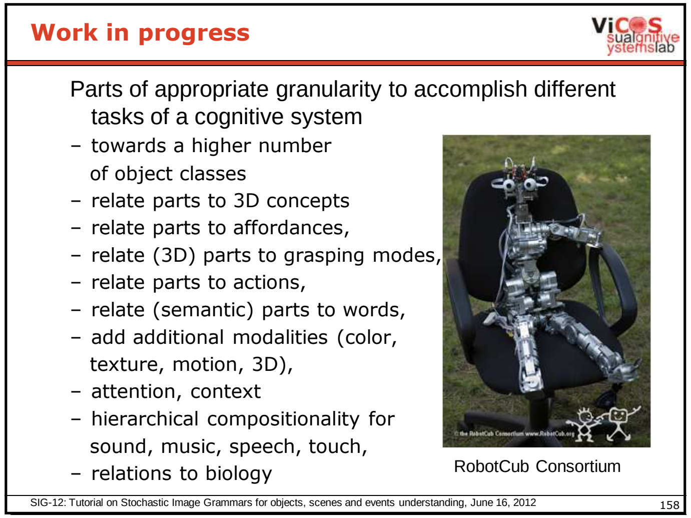
- Parts selected from various layers.
- Most parts appear from between layers 3 and 5
- Global, distinctive features selected

🟩 Presence increases probability of category
🟥 Presence decreases probability of category
🟦 Part not selected by discriminative node



Summary and discussion

- Computational principles towards modeling a large number of object classes
  - Hierarchical compositionality of object structure
- Scaling in terms of memory, speed-up of inference for multiple object classes, efficient learning
- General insights
  - Modeling/memorizing large-scale spatial-temporal patterns
    - Other modalities
    - Other senses
    - Sensing as a "controlled hallucination" (Koenderink)



Work in progress

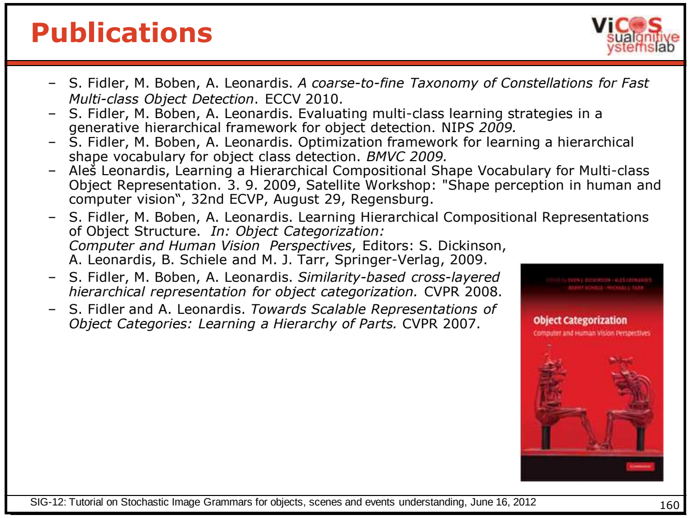
Parts of appropriate granularity to accomplish different tasks of a cognitive system

- towards a higher number of object classes
- relate parts to 3D concepts
- relate parts to affordances,
- relate (3D) parts to grasping modes,
- relate parts to actions,
- relate (semantic) parts to words,
- add additional modalities (color, texture, motion, 3D),
- attention, context
- hierarchical compositionality for sound, music, speech, touch,
- relations to biology

RobotCub Consortium

Thank you

The work was supported in part by EU projects: Cosy, Poeticon, Mobvis, CogX, USA DARPA project: Neovision2; National projects: ARRS.



Publications

- S. Fidler, M. Boben, A. Leonardis. *A coarse-to-fine Taxonomy of Constellations for Fast Multi-class Object Detection*. ECCV 2010.
- S. Fidler, M. Boben, A. Leonardis. Evaluating multi-class learning strategies in a generative hierarchical framework for object detection. NIP*S 2009.*
- S. Fidler, M. Boben, A. Leonardis. Optimization framework for learning a hierarchical shape vocabulary for object class detection. *BMVC 2009.*
- Aleš Leonardis, Learning a Hierarchical Compositional Shape Vocabulary for Multi-class Object Representation. 3. 9. 2009, Satellite Workshop: "Shape perception in human and computer vision", 32nd ECVP, August 29, Regensburg.
- S. Fidler, M. Boben, A. Leonardis. Learning Hierarchical Compositional Representations of Object Structure. *In: Object Categorization: Computer and Human Vision Perspectives*, Editors: S. Dickinson, A. Leonardis, B. Schiele and M. J. Tarr, Springer-Verlag, 2009.
- S. Fidler, M. Boben, A. Leonardis. *Similarity-based cross-layered hierarchical representation for object categorization.* CVPR 2008.
- S. Fidler and A. Leonardis. *Towards Scalable Representations of Object Categories: Learning a Hierarchy of Parts.* CVPR 2007.