Object and Category Recognition Techniques

Professor Hager

http://www.cs.jhu.edu/~hager

Agenda

- Defining the problem(s) and the problem(s) with the problem(s)
- Specific object recognition
- Category recognition
- Face detection (if time)

An Acknowledgement:

Recognizing and Learning Object Categories

Li Fei-Fei, UIUC
Rob Fergus, MIT
Antonio Torralba, MIT

An Acknowledgement:

The Evolution of Object Categorization and the Challenge of Shape Abstraction

Sven J. Dickinson
Department of Computer Science
University of Toronto

Dagstuhl Form and Function,October 2009

Courtesy Sven Dickinson

What are objects?

Dictionary.com

object Search

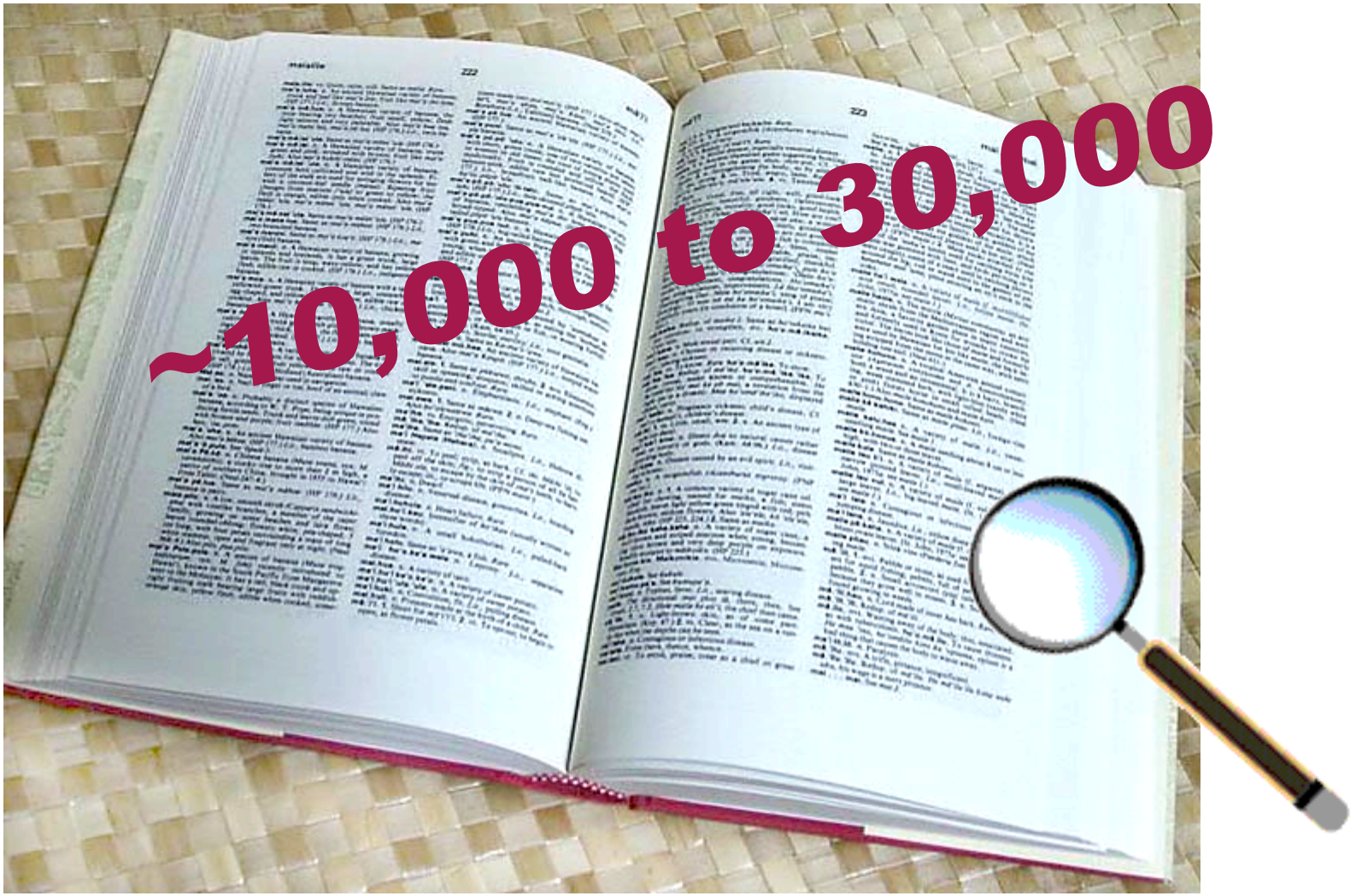
Dictionary Thesaurus Encyclopedia Web

ob·ject P Pr…on Key (ŏb′jĭkt, -jĕkt′)

n.

1. Somethi… perceptible …ne or more of the senses, especia… vision …or touch; a… material thing
2. A focus …g, thought, or action: *an object of co…*
3. The purpos… …l of a specific action or effort: *the object …game.*
4. *Grammar.*
   a. A noun, pronoun, …oun phrase that receives or is affected by the a…ion of a ve… within a sentence.
   b. A noun or substantive …verned by a preposition.
5. *Philosophy.* Something inte…ible or perceptible by the mind.
6. *Computer Science.* A discrete item that can be selected and maneuvered, such as an onscreen graphic. In object-oriented programming, objects include data and the procedures necessary to operate on that data.

How many object categories are there?



Biederman 1987

So what does object recognition involve?

Verification: is that a (the) bus?

Detection: are there cars?

Identification: is that a picture of Mao?

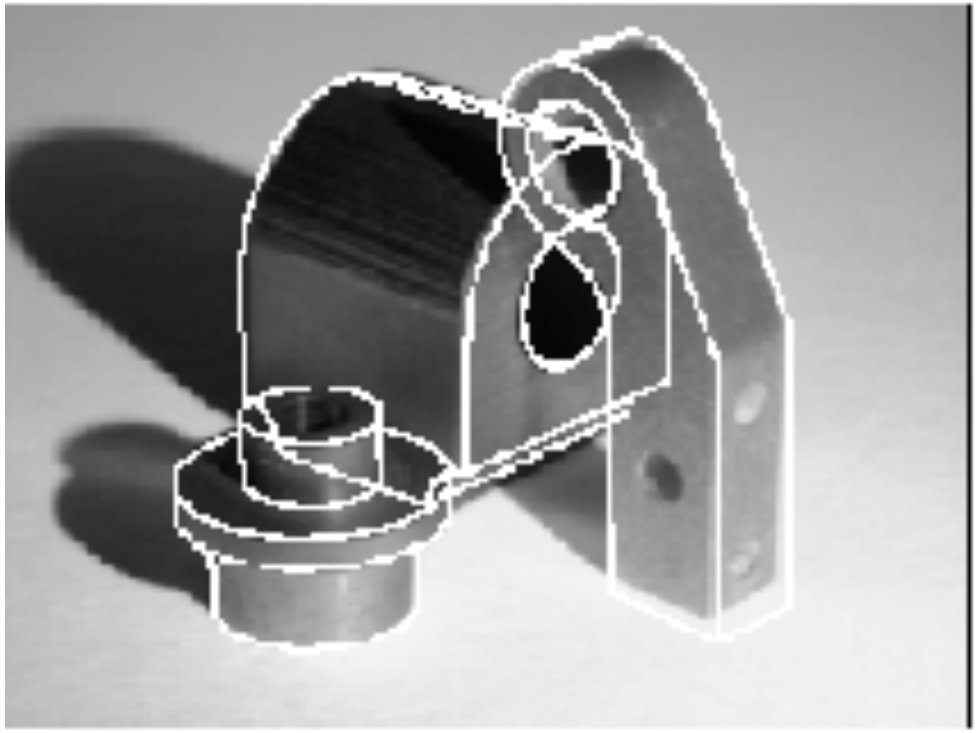
Object categorization

Scene and context categorization

Single Object Recognition

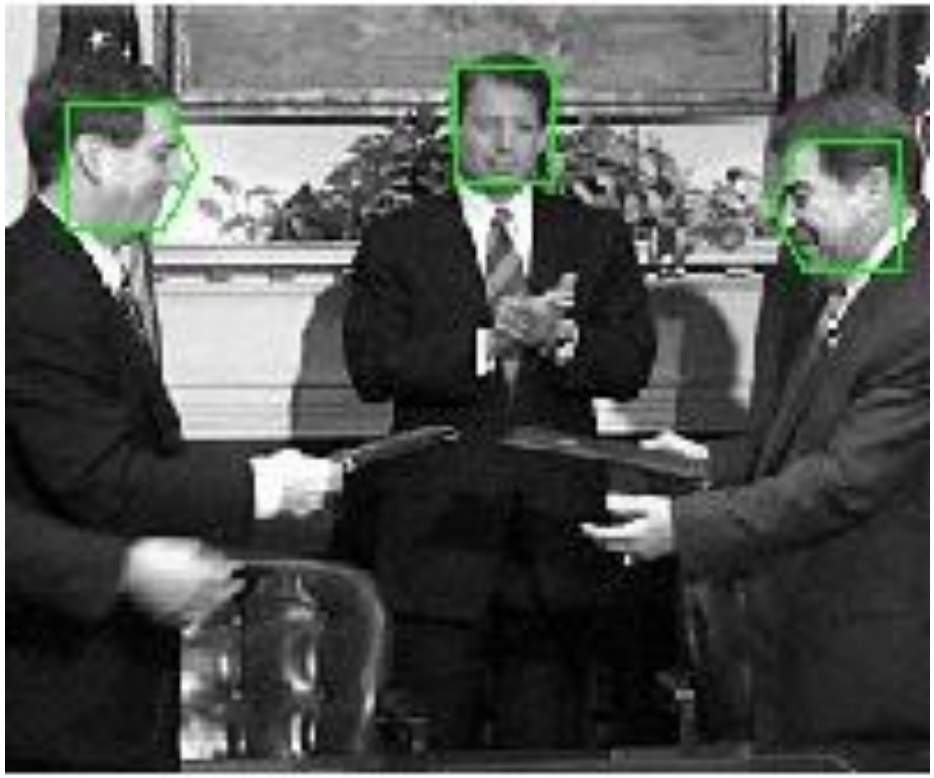
Given a database of **known objects** and an image determine what, if any of the objects are present in the image.

Single Object Recognition



Given a database of objects and an image determine what, if any of the objects are present in the image.

Single Object Recognition



Given a database of objects and an image determine what, if any of the objects are present in the image.

Challenges 1: view point variation

Michelangelo 1475-1564

Challenges 2: illumination

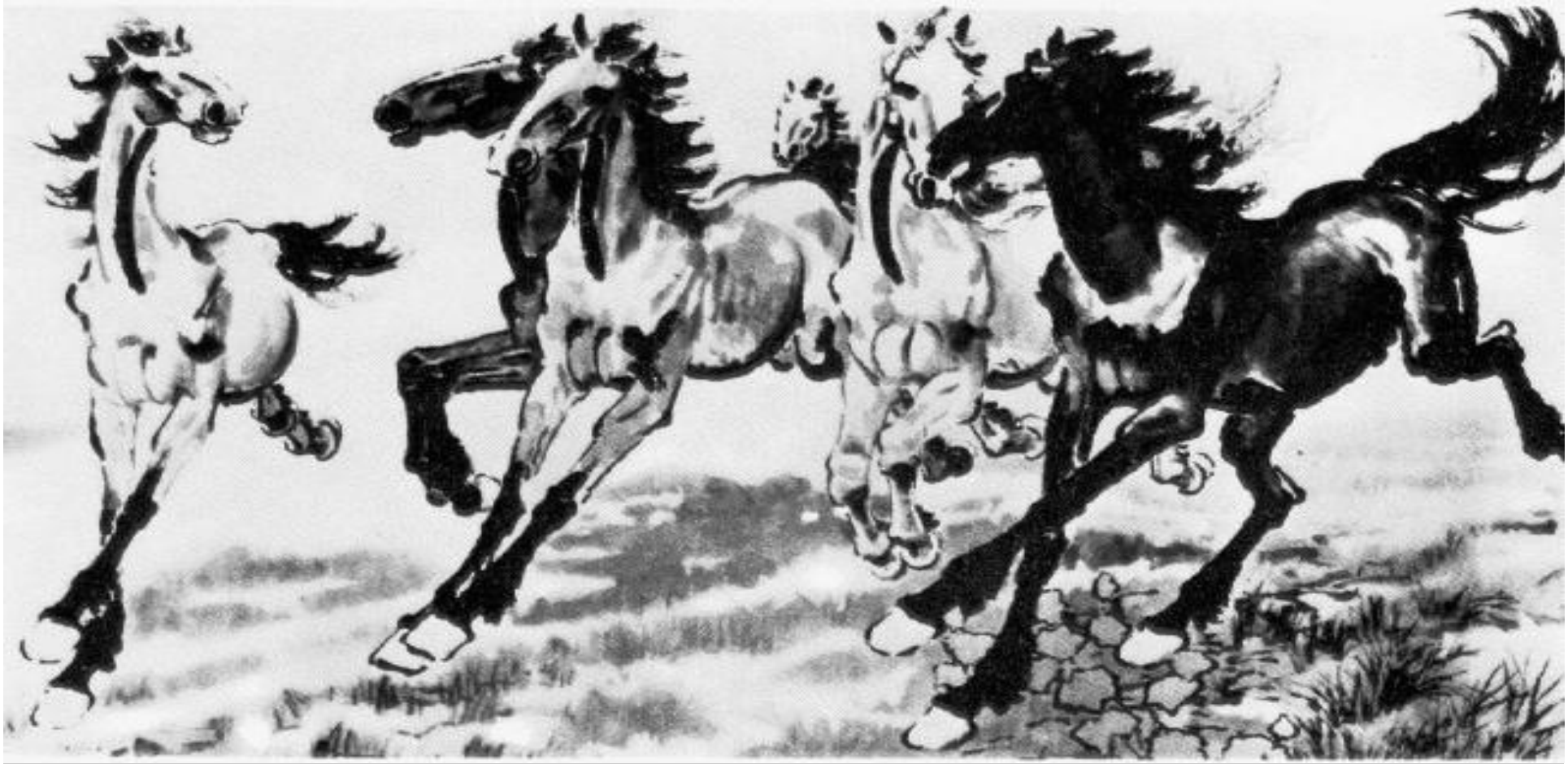
slide credit: S. Ullman

Challenges 3: occlusion

Magritte, 1957

Challenges 4: scale

Challenges 5: deformation



Xu, Beihong 1943

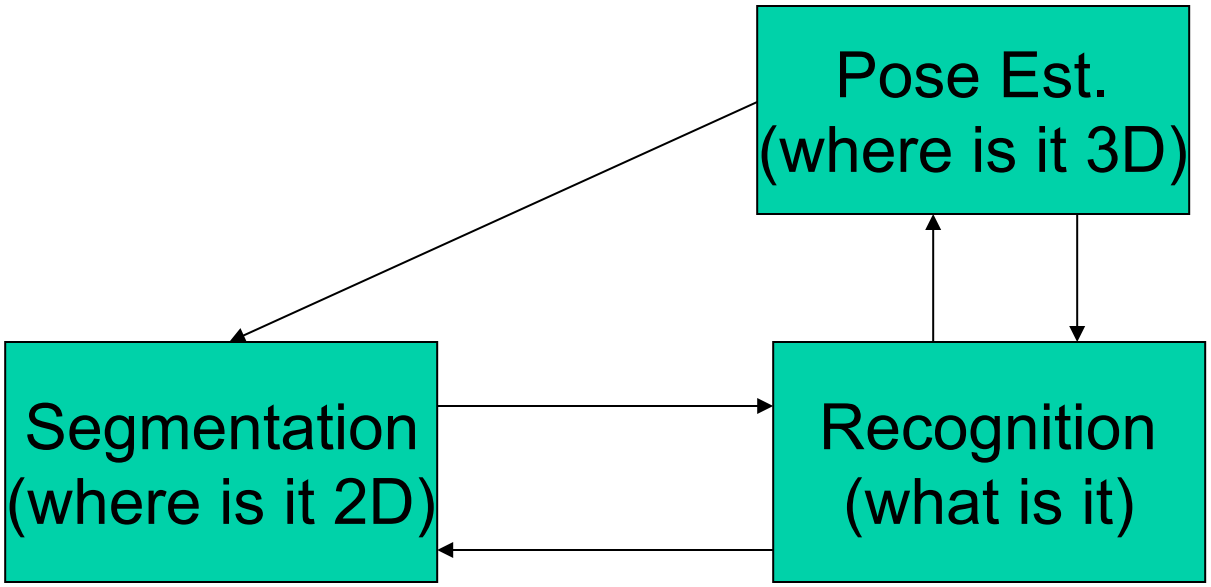
Challenges 6: background clutter

Klimt, 1913

Object Recognition: The Problem

Given: A database D of "known" objects and an image I:

1. Determine which (if any) objects in D appear in I
2. Determine the pose (rotation and translation) of the object

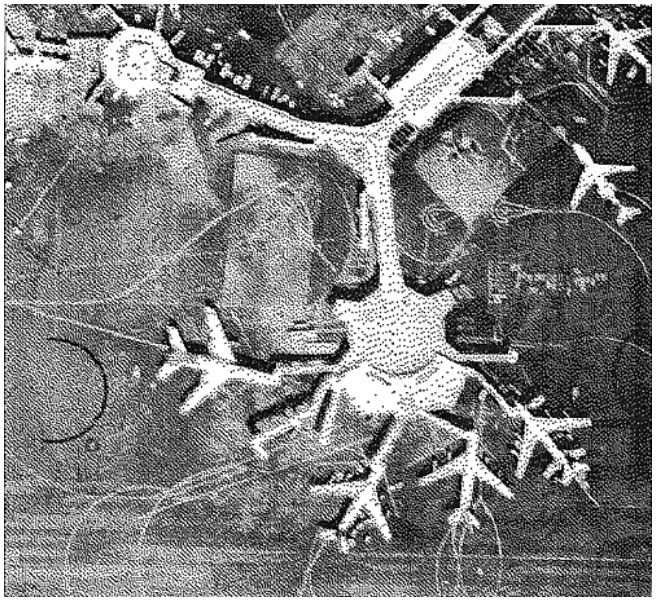


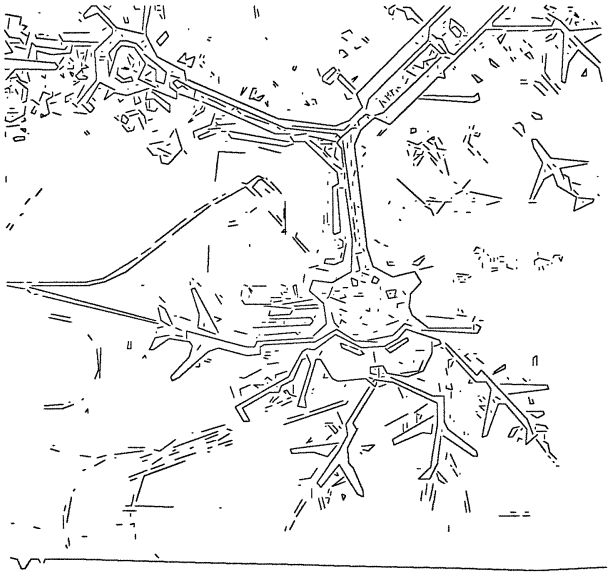
The object recognition conundrum

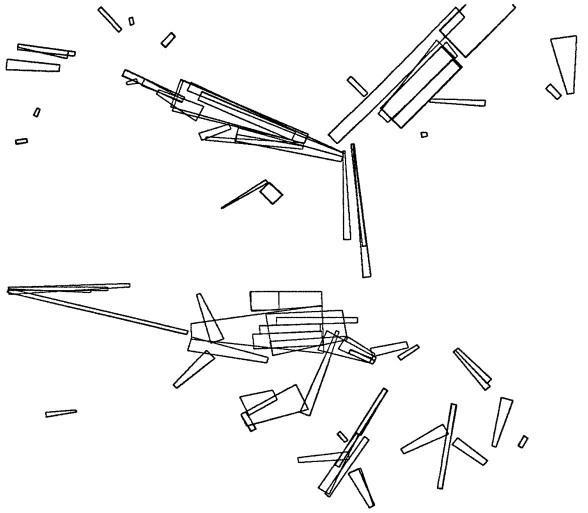
In the beginning (1970's) …

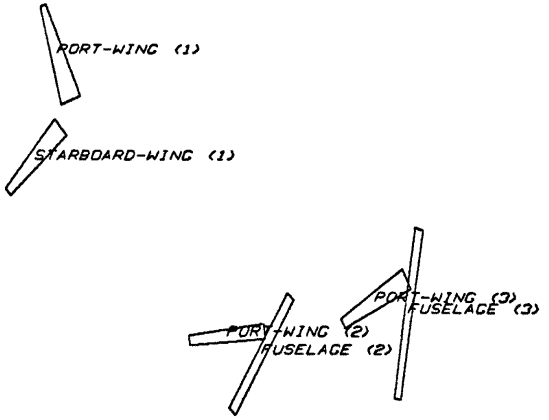
- Most notable recognition systems were categorical.
- Stanford University was the primary focal point for this research.
  - Recognition was typically based on recovering generic or parameterized volumetric parts from 2-D or 3-D images.
- Examples include: Binford, 1971; Agin and Binford, 1976; Nevatia and Binford, 1977; Marr and Nishihara, 1978; Brooks, 1981; etc.

Courtesy Sven Dickinson









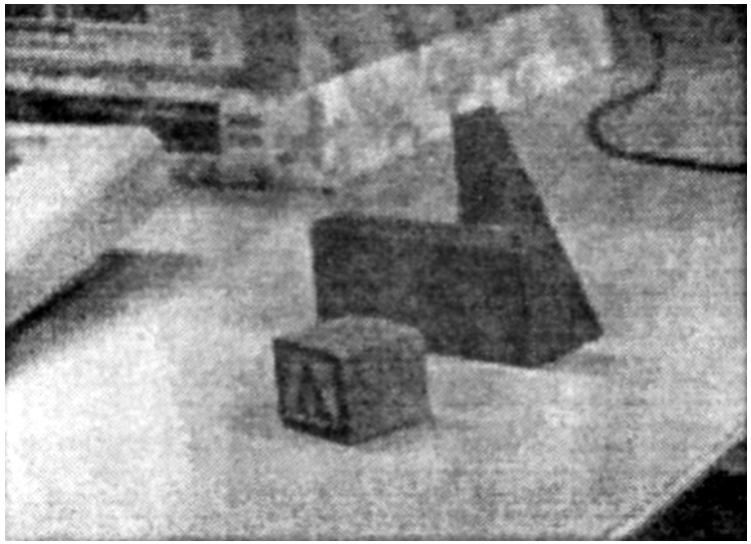
Courtesy Sven Dickinson

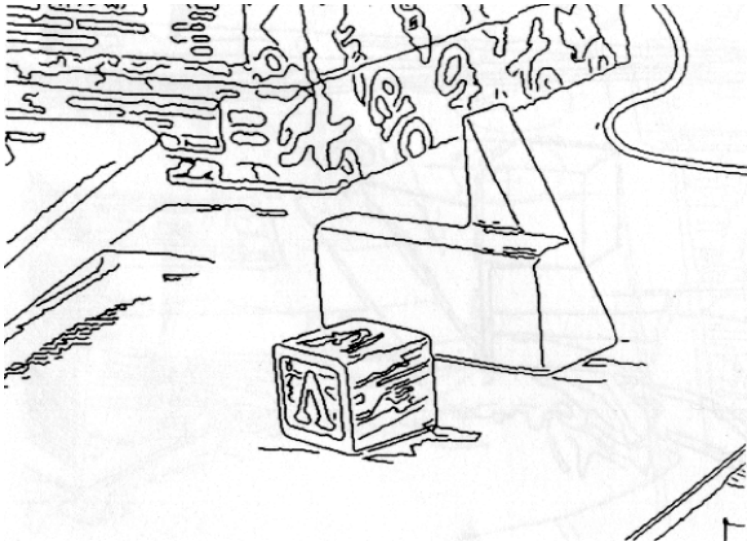
Courtesy of Rod Brooks

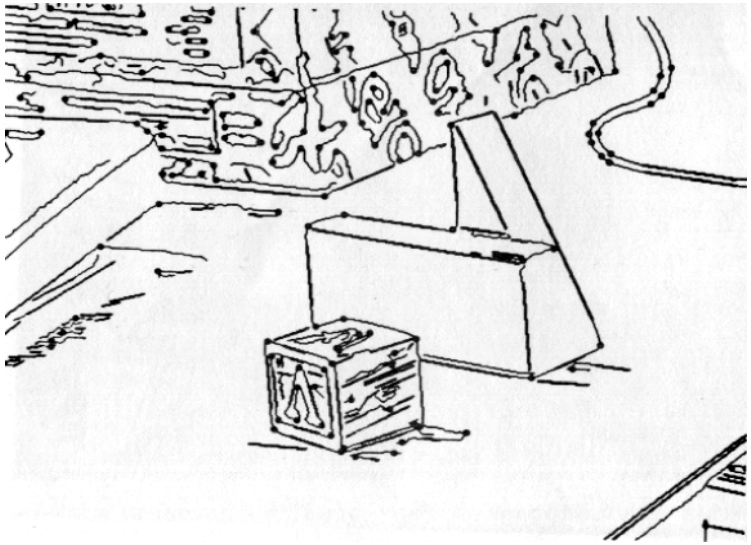
And then (mid 1980's) …

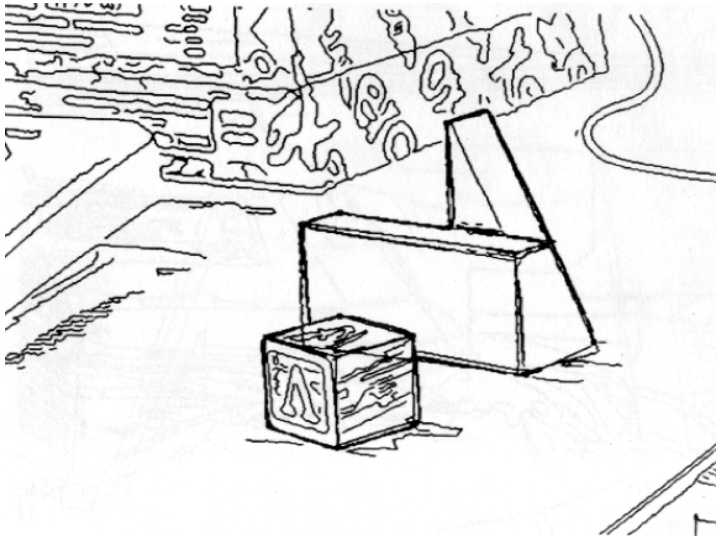
- Systems that exploit geometric constraints on polygonal/polyhedral models.
- MIT was the primary focal point for this research.
  - Exact object geometry was known, but some parameterization (e.g., part articulation) was possible.
  - Examples include: Grimson and Lozano-Perez, 1984; Lowe, 1985; Goad, 1986; Huttenlocher and Ullman, 1987; Clemens, 1991; Cass, 1992; etc.

Courtesy Sven Dickinson







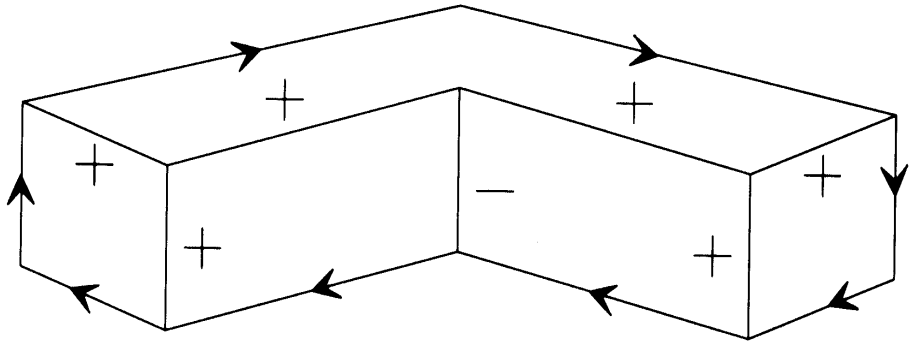


Courtesy Sven Dickinson

From Huttenlocher, 1988

We Interpret Line Drawings As 3D

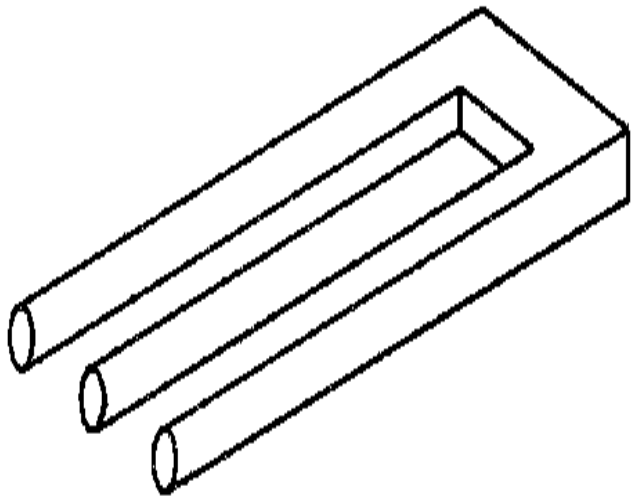
- We have strong intuitions about line drawings of simple geometric figures:
  - We can detect possible 3D objects (although our information is coming from a 2D line drawing).
  - We can detect the convexity or concavity of lines in the drawing.
  - If a line is convex, we have strong intuitions about whether it is an occluding edge.



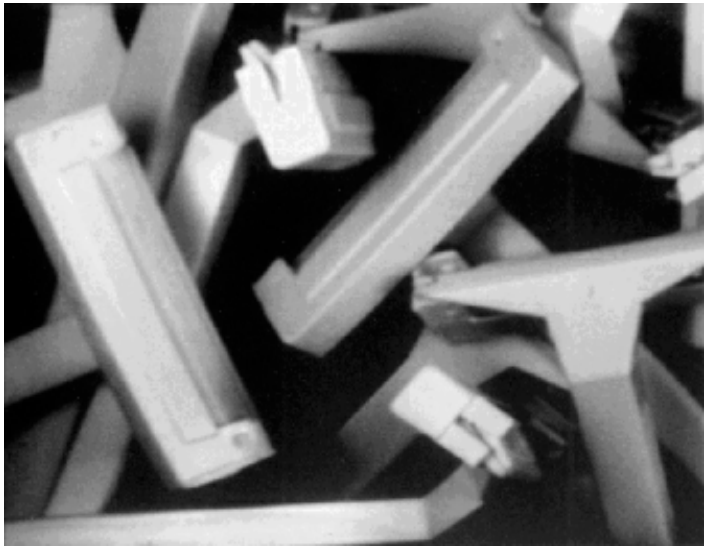
(liberally borrowed from Embick and Marcus)

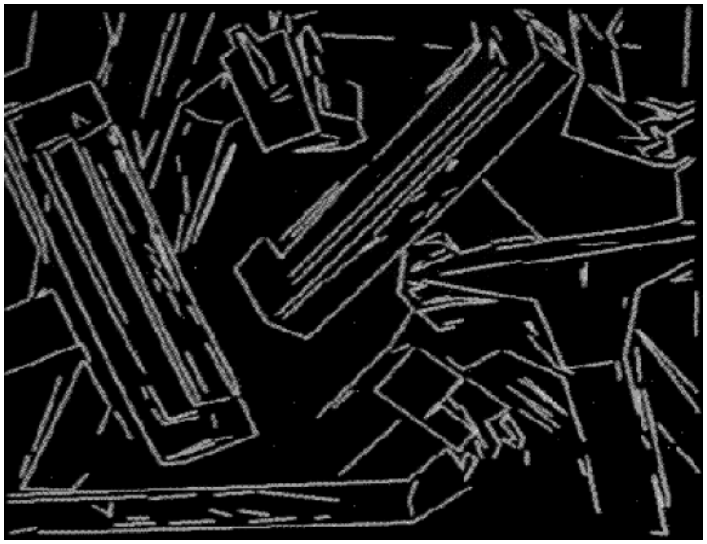
Is $O(4.5^N)$ Bad??

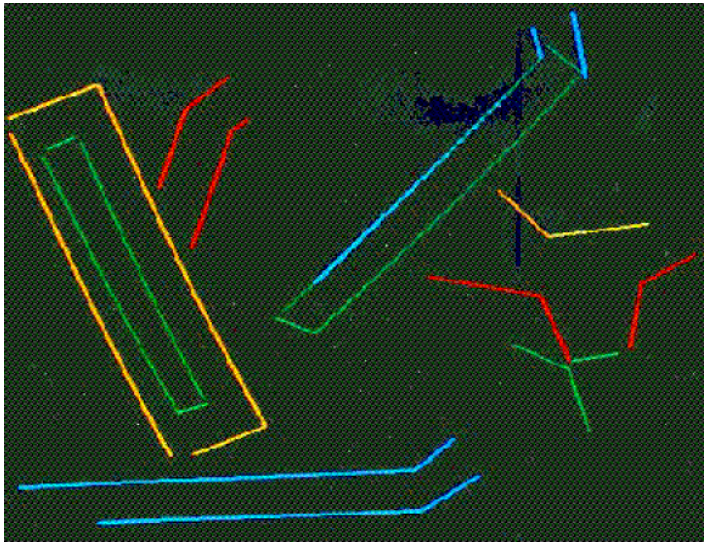
- A picture with 27 junctions, like the devil's trident) will not be rejected until all 4.5^{27} hypotheses have been rejected, leaving no interpretation.
- $4.5^{27} = 43324930223107382 4.244664378464222$
- A computer capable of checking for edge consistency at a rate of ***1 hypothesis per microsecond*** would take about ***1 million years*** to establish that the devil's trident has no consistent interpretation!

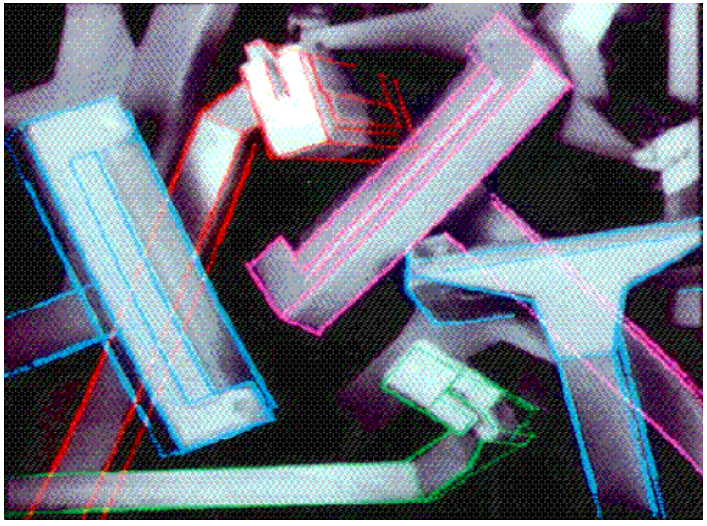


(liberally borrowed from Embick and Marcus)









Courtesy Sven Dickinson

Courtesy of David Lowe

Invariants

Basic definitions:

features f

transformations T

I is an *invariant* if I(f) = I(T f) for all T

Examples:

f are pairs of points, T is translation, $I(p_1,p_2) = p_1-p_2$

f are pairs of points, T is homogeneous transform $I(p_1,p_2) = p_1 \not\in p_2$

These are examples of *Euclidean* invariants

camera projection is *not* Euclidean!

If T is a projective transformation, then I is a *projective invariant*

camera projection is a special case

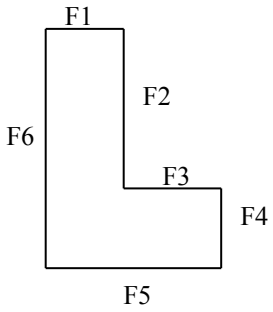
a much larger group ---> therefore fewer invariants!

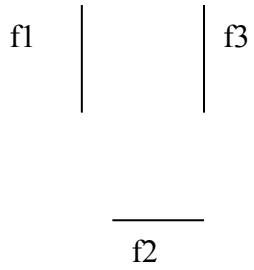
Interpretation Trees: Basic Idea

- Given:
  - A (usually 3D geometric) model, we a set of features and defined relationships (invariants) between features F_1, F_2, ... F_n
    - unary: e.g. length range
    - binary: e.g. distance range
    - trinary: e.g. angle between triples of points
  - An observed set of features f_1, f_2, ... f_m
- Compute:
  - all possible matches between model features and observed features which respect the given constraints
  - constraints are *object specific* rather than generic
  - constraints are quantitative (numbers) rather than qualitative properties

Grimson, W.E.L., Object Recognition by Computer: The Role of Geometric Constraints,Cambridge: MIT Press 1990.

angles	F1	F2	F3	F4	F5	F6
F1	0	3π/2	0	3π/2	π	π/2
F2	π/2	0	π/2	0	3π/2	π
F3	0	3π/2	0	3π/2	π	π/2
F4	π/2	0	π/2	0	3π/2	π
F5	π	π/2	π	π/2	0	3π/2
F6	3π/2	π	3π/2	π	π/2	0



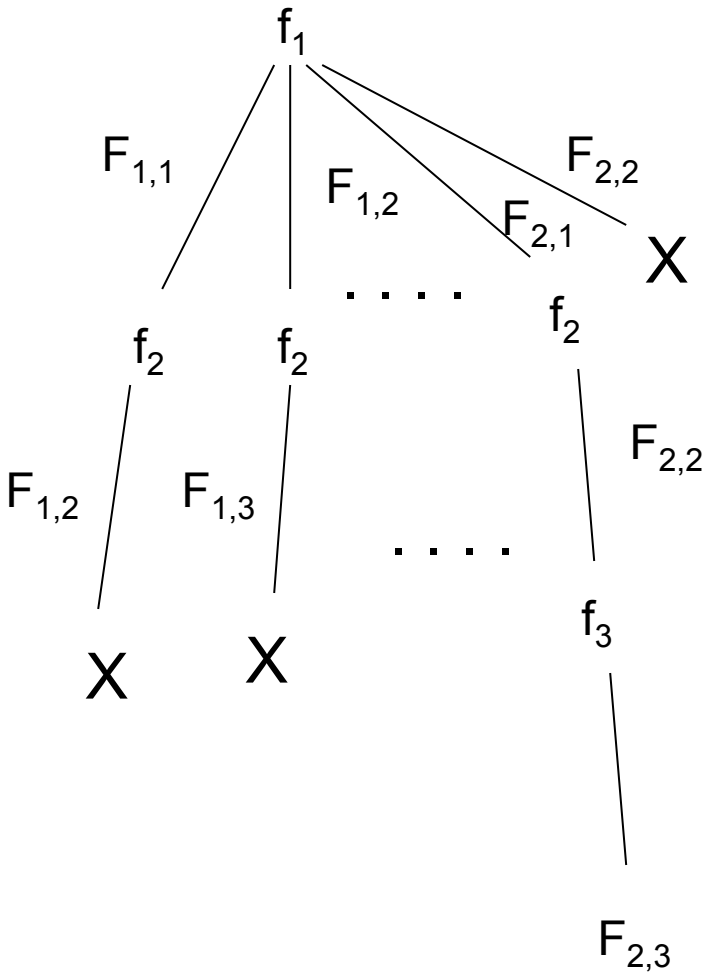


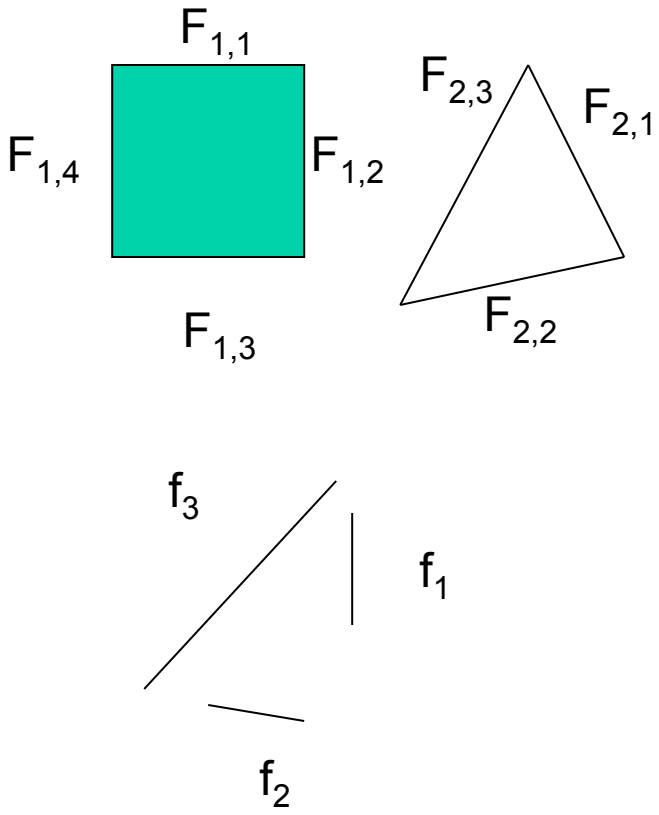
angles	f1	f2	f3
f1	0	π/2	π
f2	3π/2	0	π/2
f3	π	3π/2	0

distance	F1	F2	F3	F4	F5	F6
F1	[0,1]	[0,5]	[4,8]	[5,13]	[9,13]	[0,10]
F2	[0,5]	[0,4]	[0,5]	[1,10]	[1,10]	[1,10]
F3	[4,8]	[0,5]	[0,1]	[0,2]	[1,5]	[1,8]
F4	[5,13]	[1,10]	[0,2]	[0,1]	[0,5]	[4,13]
F5	[9,13]	[1,10]	[1,5]	[0,5]	[0,4]	[0,13]
F6	[0,10]	[1,10]	[1,8]	[4,13]	[0,13]	[0,9]

distance	f1	f2	f3
f1	[0,1]	[1,5]	[1,2]
f2	[1,5]	[0,1]	[1,5]
f3	[1,2]	[1,5]	[0,1]

Interpretation Tree: a 2D Example





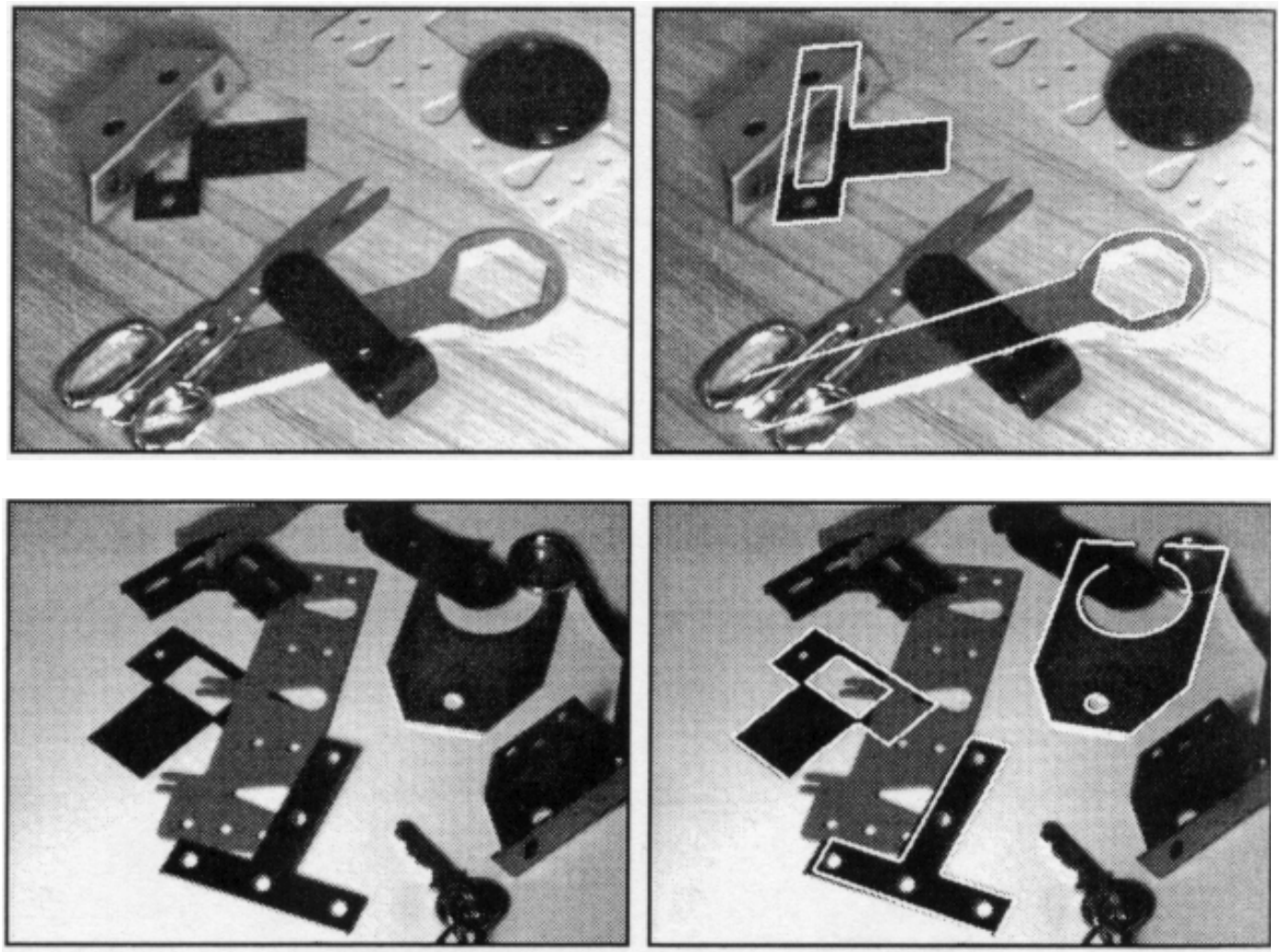
Limitations of ITs

- Fundamentally, a combinatorial approach to matching
- If *s are allowed (and they must be), increases the combinatorics, and also increases ambiguity
  - how many *'d features should we included in an interpretation?
  - is fewer *'d feature necessarily better?
- Unary or Binary constraints are not enough to always generate a unique or consistent match
- Depends on *Euclidean invariants (at least as presented)*

Next, there was (late 1980's) …

- Systems that exploit geometric invariants to facilitate efficient indexing into large databases.
- Oxford University was the primary focal point for this research.
- A priori knowledge of exact object geometry was essential.
- Examples include: Lamdan et al., 1988; Kriegman and Ponce, 1990; Forsyth et al., 1991; Rigoutsos and Hummel, 1993; etc.

Courtesy Sven Dickinson



From Rothwell et al., 1992

Courtesy Sven Dickinson

And then (the 1990's) …

- Appearance-based recognition.
- No segmentation, grouping, abstraction, or even 3-D modeling required.
- Recognition of complex exemplars (for the first time!), but exact object appearance must be known.
- Examples include: Turk and Pentland, 1991; Murase and Nayar, 1995; Leonardis and Bischoff, 1996; Camps et al, 1998, etc.

Courtesy Sven Dickinson

Image-based Object Recognition

An observation:

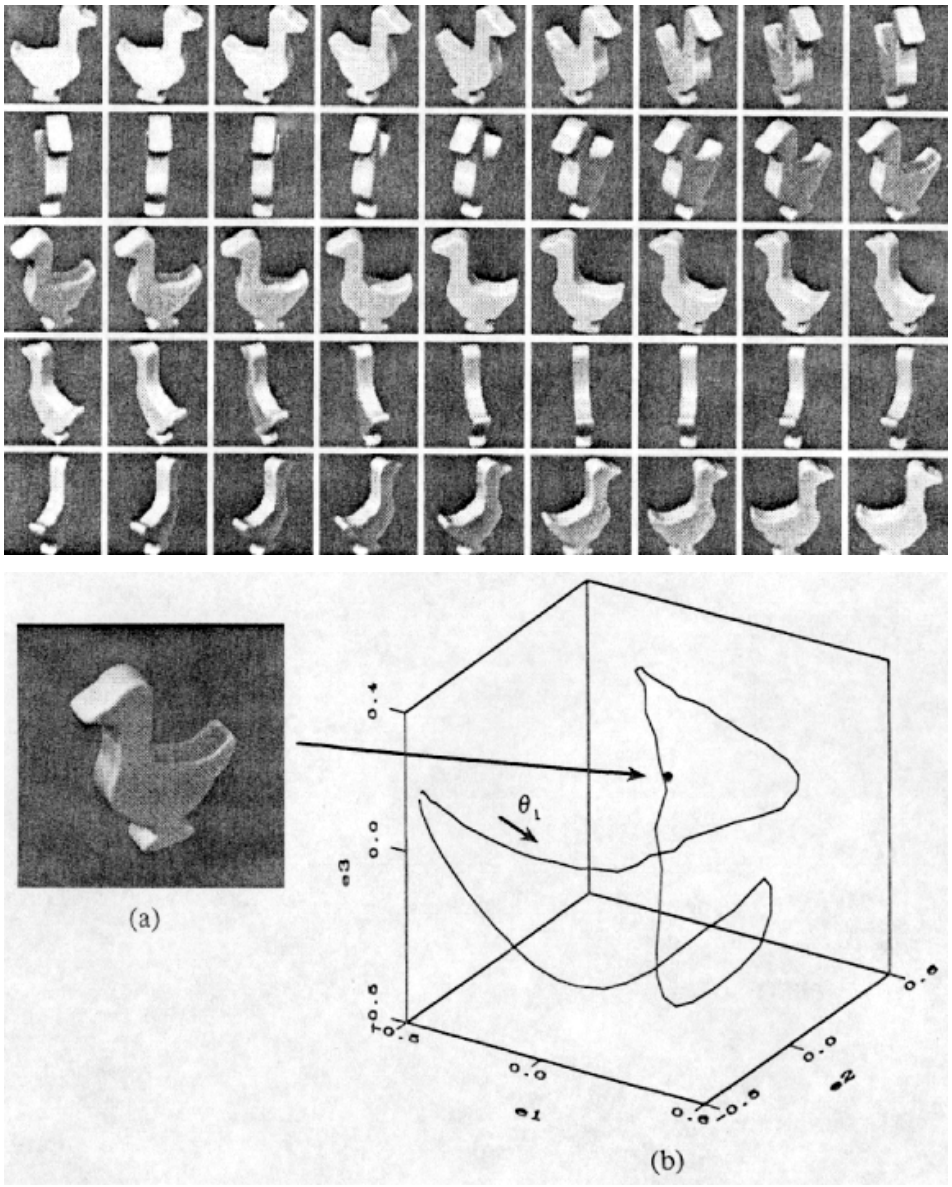
If we have seen an object from every viewpoint and under all lighting conditions, then object recognition is "simply" a table lookup in the space of 2D images

Another way to view it:

Consider an image as a point in a space
Consider now all points generated as above

Then, an object is some "surface" in the space of all images

"Visual Learning and Recognition of 3D Objects from Appearance," H. Murase and S.K. Nayar, International Journal on Computer Vision, Vol.14, No.1, pp.5-24, Jan, 1995.



Courtesy Sven Dickinson

From Murase and Nayar, 1995

Image-based Object Recognition: Learning

- Gather up all of the images of all objects under all viewing conditions:
  - segment to contain just the object; sample to common size
  - subtract the mean of the result from each image
  - normalize 0 mean images to unit norm
  - gather all resulting images into a matrix M (for models)

- Compute the eigenvalues and eigenvectors of $M M^t$
  - we can use SVD to do this!

- Retain the k eigenvectors with the largest associated eigenvalues
  - Usually, choose k such that $\sigma_k / \sigma_1 < \tau$ where τ is small (e.g. .05).
  - Call the resulting matrix E (for eigenvalue projection).

- Store a vectors $C_o = \{g^o_i = E^t I^0_i\}$ for each image i of object o

An Example

- Columbia SLAM system:
  - can handle databases of 100's of objects
  - single change in point of view
  - uniform lighting conditions

Courtesy Shree Nayar, Columbia U.

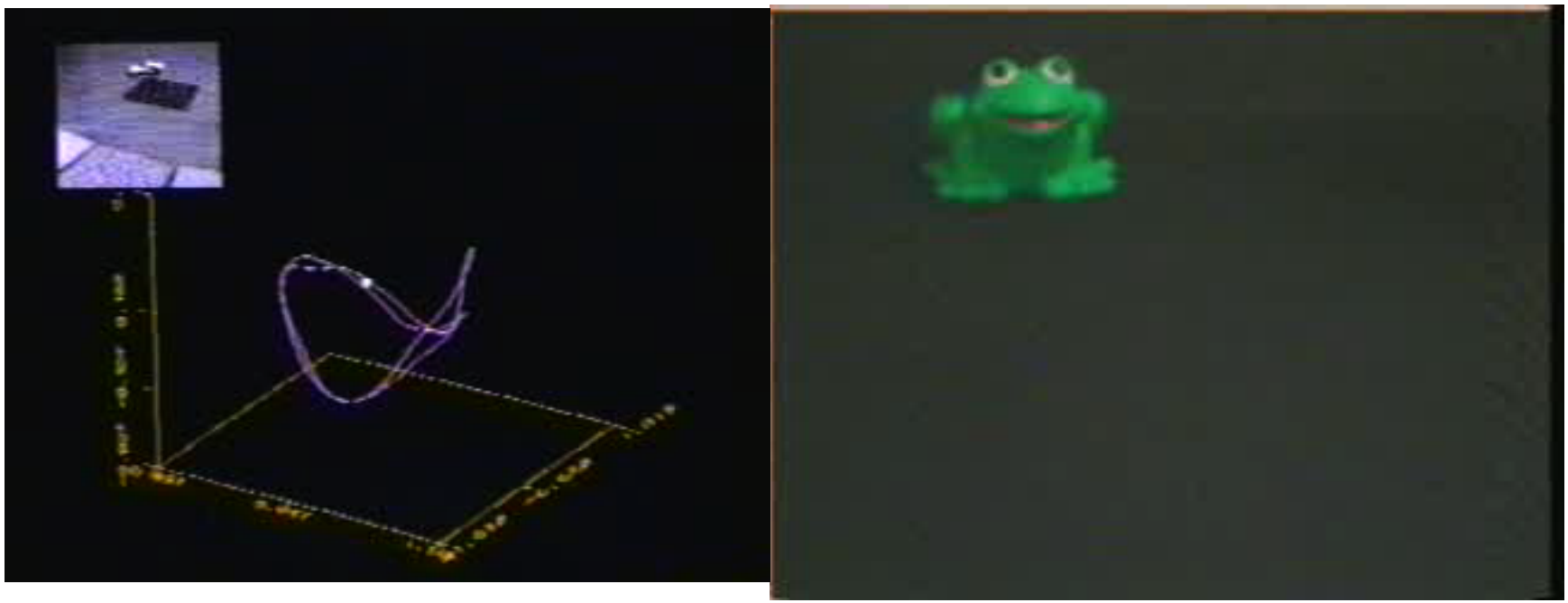


Image-based Object Recognition: Limitations

- Hard to get all of the samples needed
  - variations in pose
  - variations in lighting

- Better for Lambertian; less so for specular objects

- Assumes a constant background or good segmentation

- No occlusion!

And finally (the 2000's) …

- Robust descriptions of local image patches centered at interest points.
- Recognition of complex exemplars in the presence of occlusion, scale, rotation, articulation, etc.
- Categorization possible for restricted categories (e.g., faces, cars, animal species, motorcycles, etc.)
- Examples include: Schmid and Mohr, 1997; Lowe, 1999; Fergus et al., 2003; Rothganger et al., 2003; V. Ferrari et al., 2004, Carneiro and Jepson, 2005; etc.

Courtesy Sven Dickinson

Basic Ideas

- Use local features
  - feature = minimum or maximum in difference of Gaussian images; store location, scale (in DoG scale space) and orientation
  - feature location is blurred (equiv. chamfered) for matching purposes
  - a feature vector is stored by sampling gradient values in feature-defined coordinate system (128 values = 4x4 samples and 8 orientations)

- Use object views
  - view is a set of visible features
  - views that overlap contain links between common features
  - views are created automatically though clustering
  - views should work for around 20 degrees of out-of-plane rotation

Feature Matching

- Uses a Hough transform
  - parameters are position, orientation and scale for each training view
  - features are matched to closest Euclidean distance neighbor in database; each database feature indexed to object and view as well as location, orientation and scale
  - features are linked to adjacent model views; these links are also followed and accumulated
  - implemented using a hash table

Verification and Training

- Views are matched under affine transformations:
  - u' = s R u + d → leads to a linear system Ax = b
  - (geometric) match error e = sqrt(2 || A x* - b ||/(r-4)) where r is the # of matched features
  - in learning stage, use e to decide if a view should be clustered or create a new cluster; threshold T = 0.05 * max(r,c) where r,c is size of training image

- Training simply requires many images of objects, not necessarily organized in any way; three cases:
  - training image doesn't match an existing object model; new object model is formed with this image
  - training image matches an existing model view, but e > T;
    - new model view created and linked to three closes model views; overlapping features are linked.
  - training image matches an existing model view and e < T;
    - aggregate any new features into the existing model view

Final Probability Model

- There can still be many false positives and negatives
- Compute P(m | f) where f are the k matched features and m is a model view
- probability of false match for a single feature is
    - p = d l r s
    - d = fraction of database features in this model view
    - l = 0.2^2 = 0.04 (location ranges of 20% of model size)
    - r = 30/360 = 0.085
    - s = 0.5
    - P(f | : m) = binomial using p, n (# of features) and k (# of matches)
- P (m | f) = P(f|m) P(m) / (P(f|m) P(m) + P(f | : m) P(: m))
- Assume P(: m)=1 and P(f | m) = 1
- Thus P(m | f) = P(m) / (P(m) + P(f | : m))
- Assume P(m) is roughly constant and = 0.01
- Accept a model if P(m|f) > 0.95
- Requires 3-10 features depending on object and level of clutter

PDF of Matching

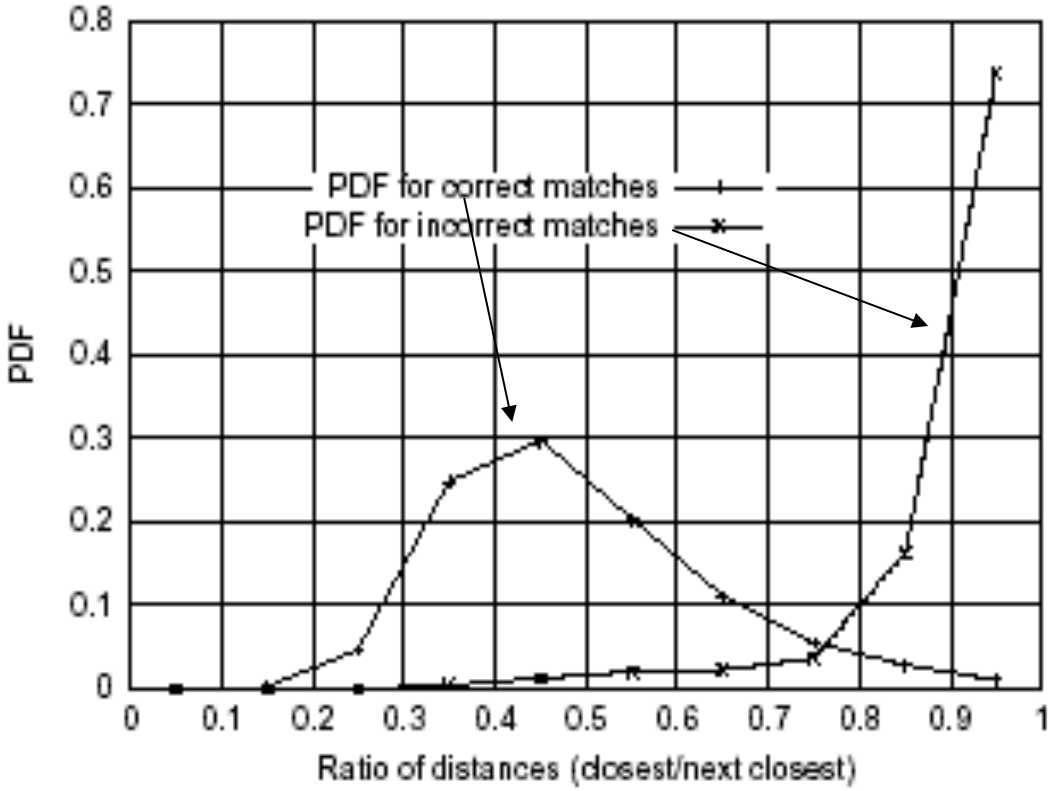


Figure 11: The probability that a match is correct can be determined by taking the ratio of distance from the closest neighbor to the distance of the second closest. Using a database of 40,000 key-points, the solid line shows the PDF of this ratio for correct matches, while the dotted line is for matches that were incorrect.

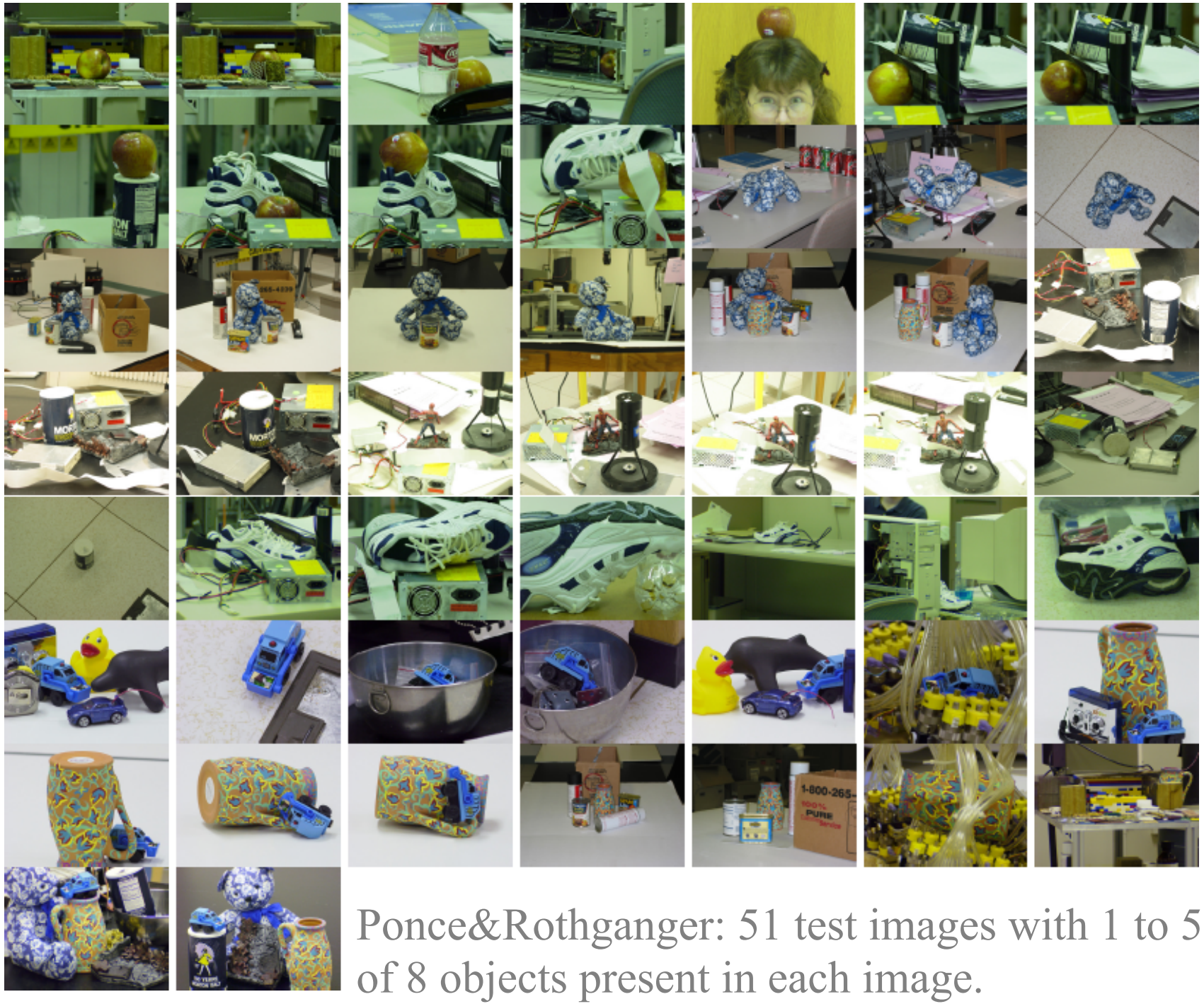
Results

- Matching requires histogramming followed by alignment

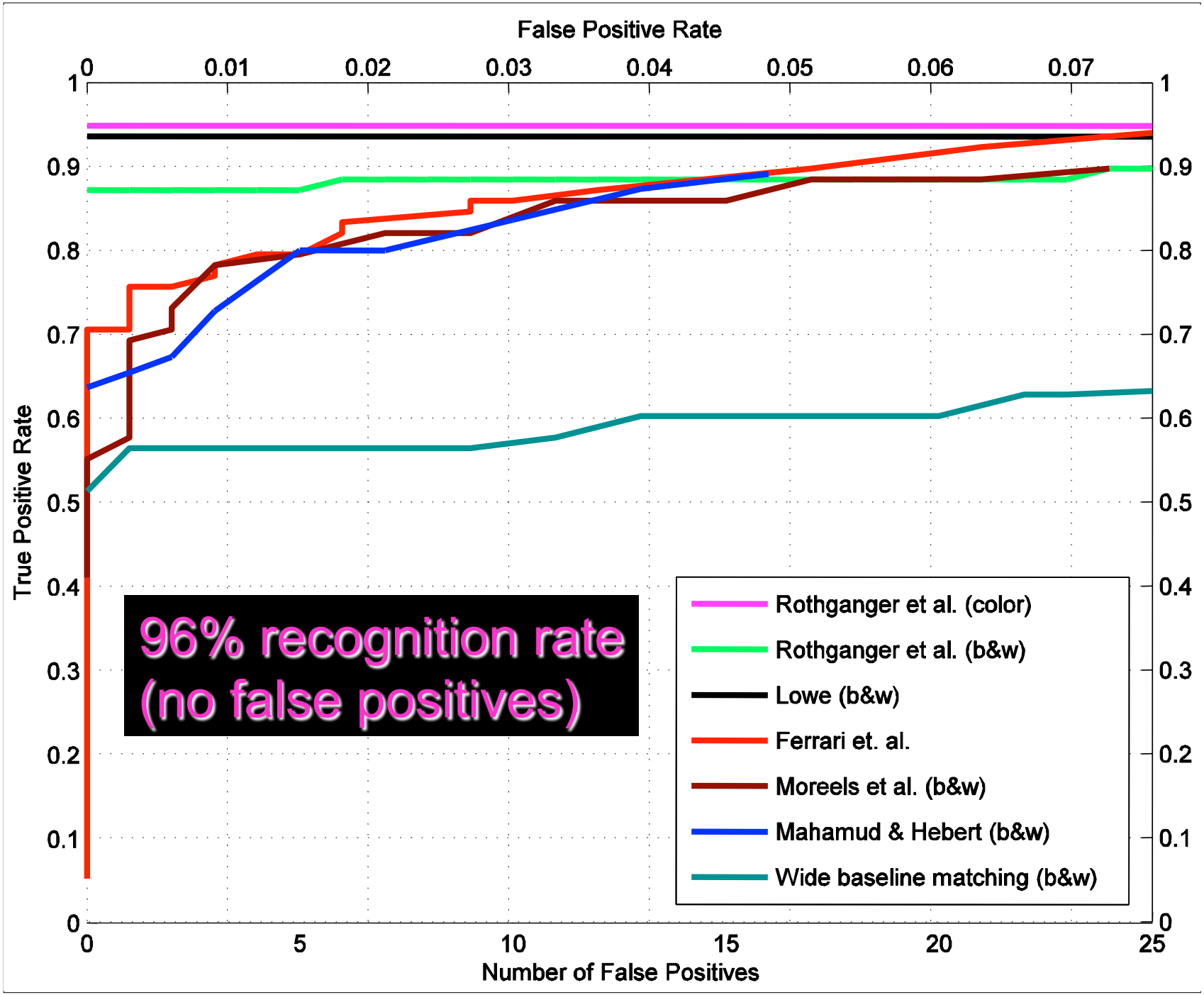
Figure 12: The training images for two objects are shown on the left. These can be recognized in a cluttered image with extensive occlusion, shown in the middle. The results of recognition are shown on the right overlaid on a reduced contrast version of the image. A parallelogram is drawn around each recognized object showing the boundaries of the original training image under the affine transformation solved for during recognition. Smaller squares indicate the keypoints that were used for recognition.

Results

Figure 13: This example shows location recognition within a complex scene. The training images for locations are shown at the upper left and the 640x315 pixel test image taken from a different viewpoint is on the upper right. The recognized regions are shown on the lower image, with key-points shown as squares and an outer parallelogram showing the boundaries of the training images under the affine transform used for recognition.



Ponce&Rothganger: 51 test images with 1 to 5 of 8 objects present in each image.


False Positive Rate
0
0.01
0.02
0.03
0.04
0.05
0.06
0.07
True Positive Rate
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
5
10
15
20
25
Number of False Positives
96% recognition rate
(no false positives)
Rothganger et al. (color)
Rothganger et al. (b&w)
Lowe (b&w)
Ferrari et. al.
Moreels et al. (b&w)
Mahamud & Hebert (b&w)
Wide baseline matching (b&w)

Extensions

- More recent work has focussed on improving
  - feature detection
    - high repeatibility for out-of-plane rotation
    - picking up “more” features per unit area
  - matching
    - Lowe uses nearest neighbor
    - Other options are thresholding, likelihood ratio ....

Comparing Features and Matching

from Mikolajczyk and Schmid: A Performance Evaluation of Local Descriptors

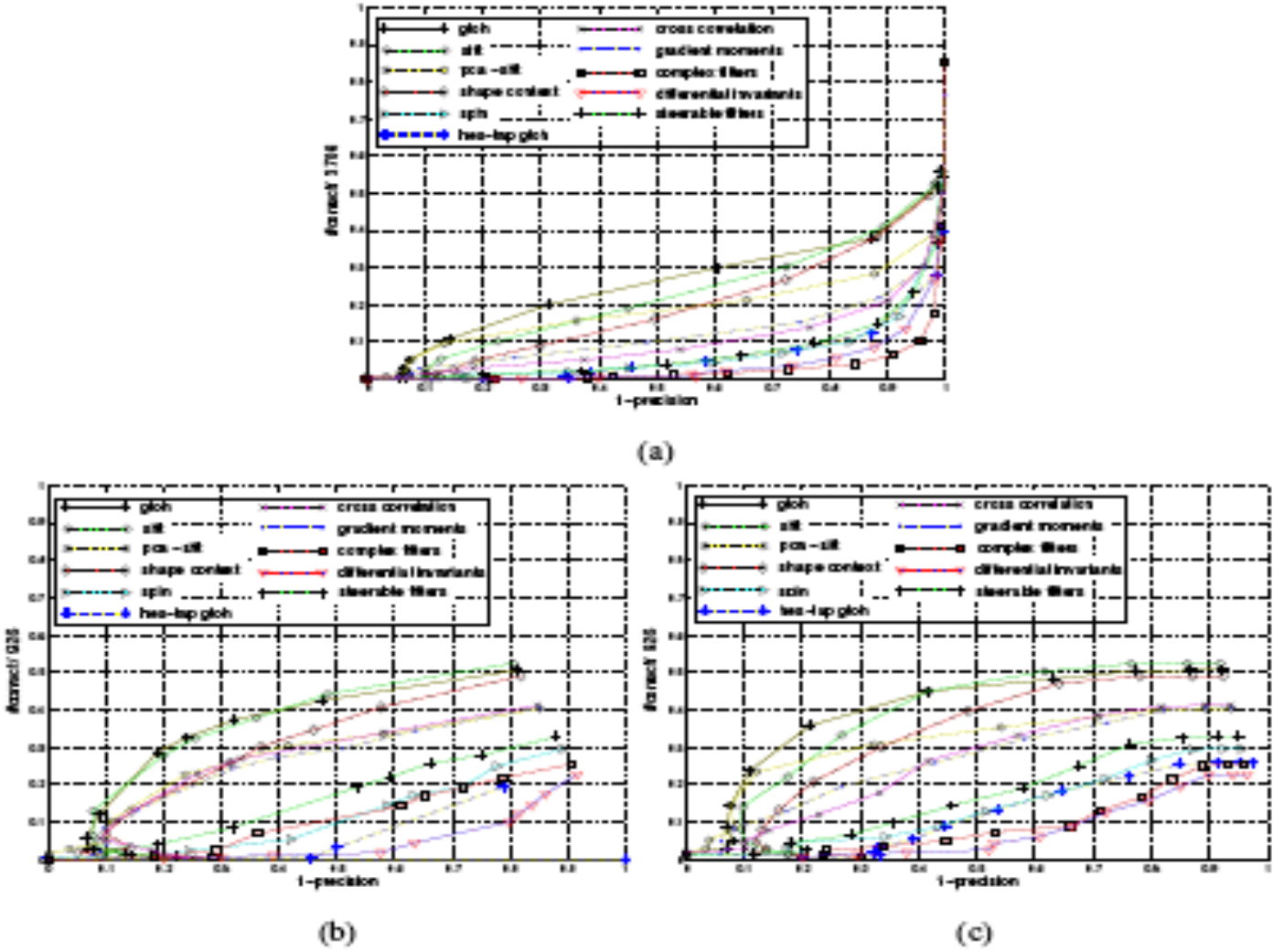


Fig. 4. Comparison of different matching strategies. Descriptors computed on Hessian-Affine regions for images from figure 3(e). (a) Threshold based matching. (b) Nearest neighbor matching. (c) Nearest neighbor distance ratio matching. `hes-lap gloh` is the GLOH descriptor computed for Hessian-Laplace regions (cf. section IV-A.4).

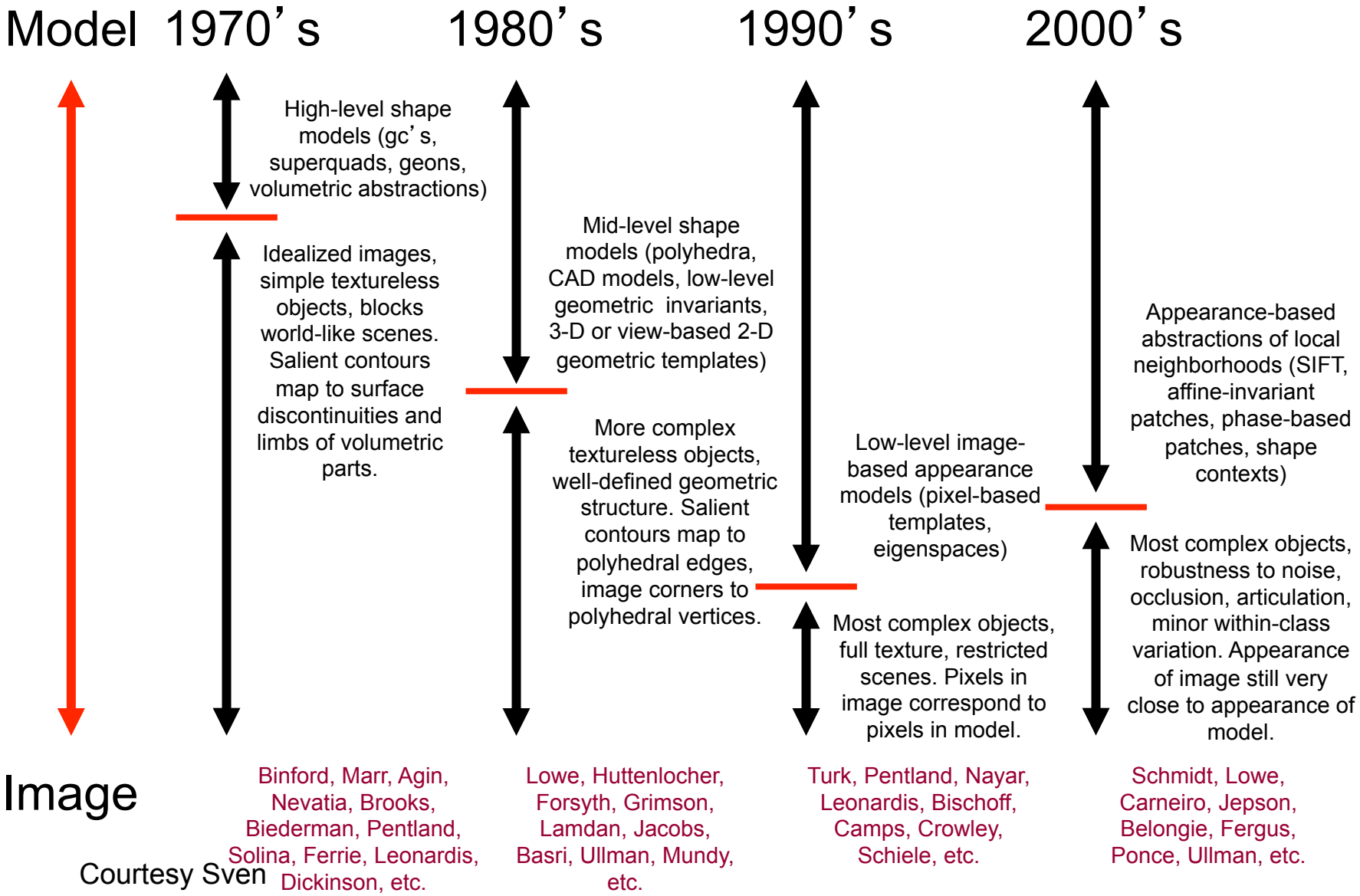
Affine Invariance Example

from Mikolajczyk and Schmid: A Performance Evaluation of Local Descriptors

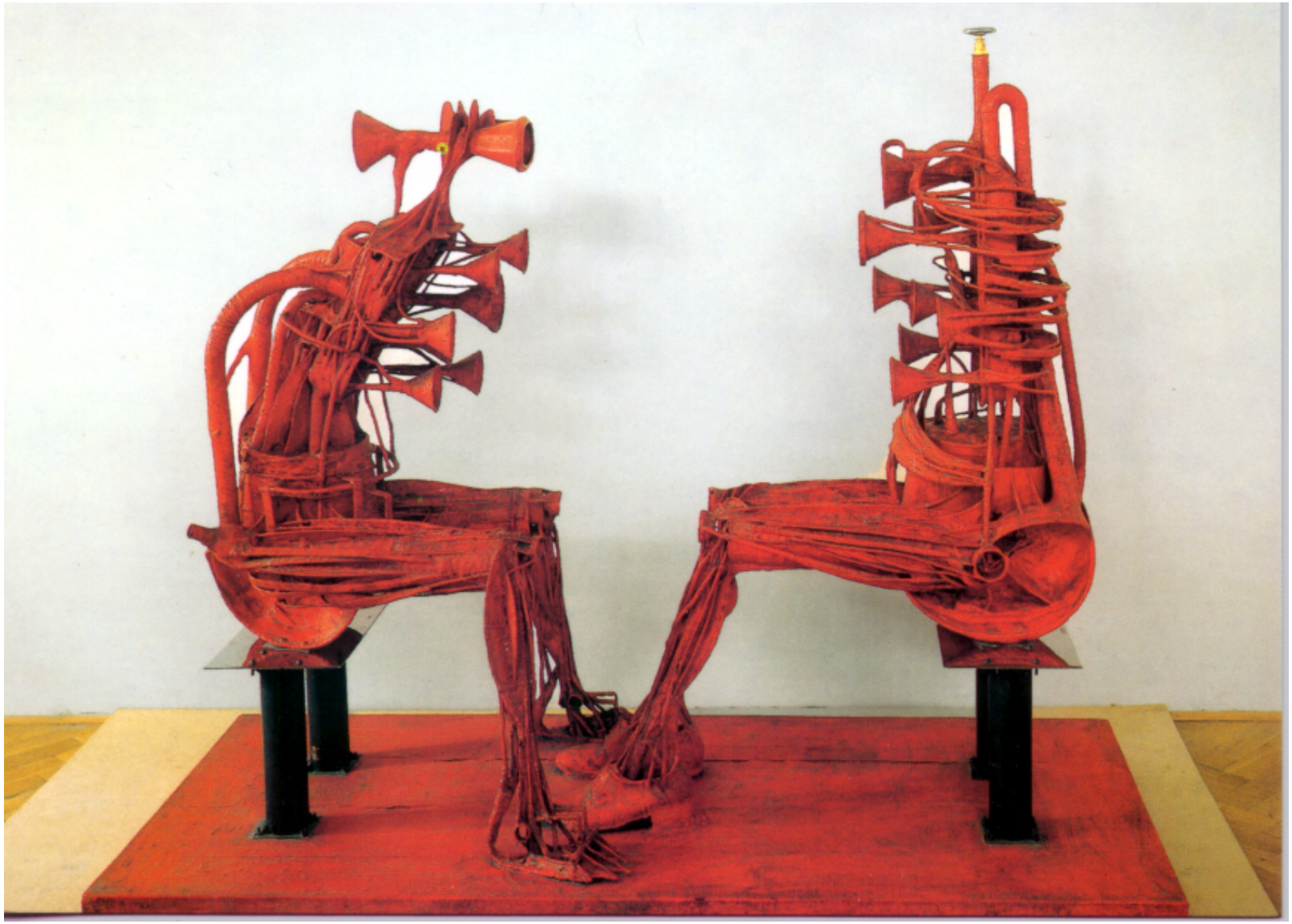
(a) (b)

Fig. 13. Matching example. There are 400 nearest neighbor matches obtained with the GLOH descriptor on Hessian-Affine regions. There are 192 correct matches (yellow) and 208 false matches (blue).

A Clear Trend



Courtesy Sven Dickinson



Great Dialog, Karel Nepras, 1966 (Prague National Gallery)

Courtesy Sven Dickinson

Great Dialog, Karel Nepras, 1966 (Prague National Gallery)
Memory Figure Sitting on a Stool, Akan Culture, Ghana
Courtesy Sven Dickinson

Challenges 7: intra-class variation

Category Recognition

- Categories share:
  - common parts (all cars have wheels; all planes have wings)
  - spatial relationships (two eyes above a nose above a mouth)
  - common appearance elements (all Campbell's soup say Campbell's somewhere
- Question: can local features be used to model these attributes
- A brief overview taken from a recent tutorial by Fei-Fei, Fergus, and Torralba
- http://people.csail.mit.edu/torralba/iccv2005/

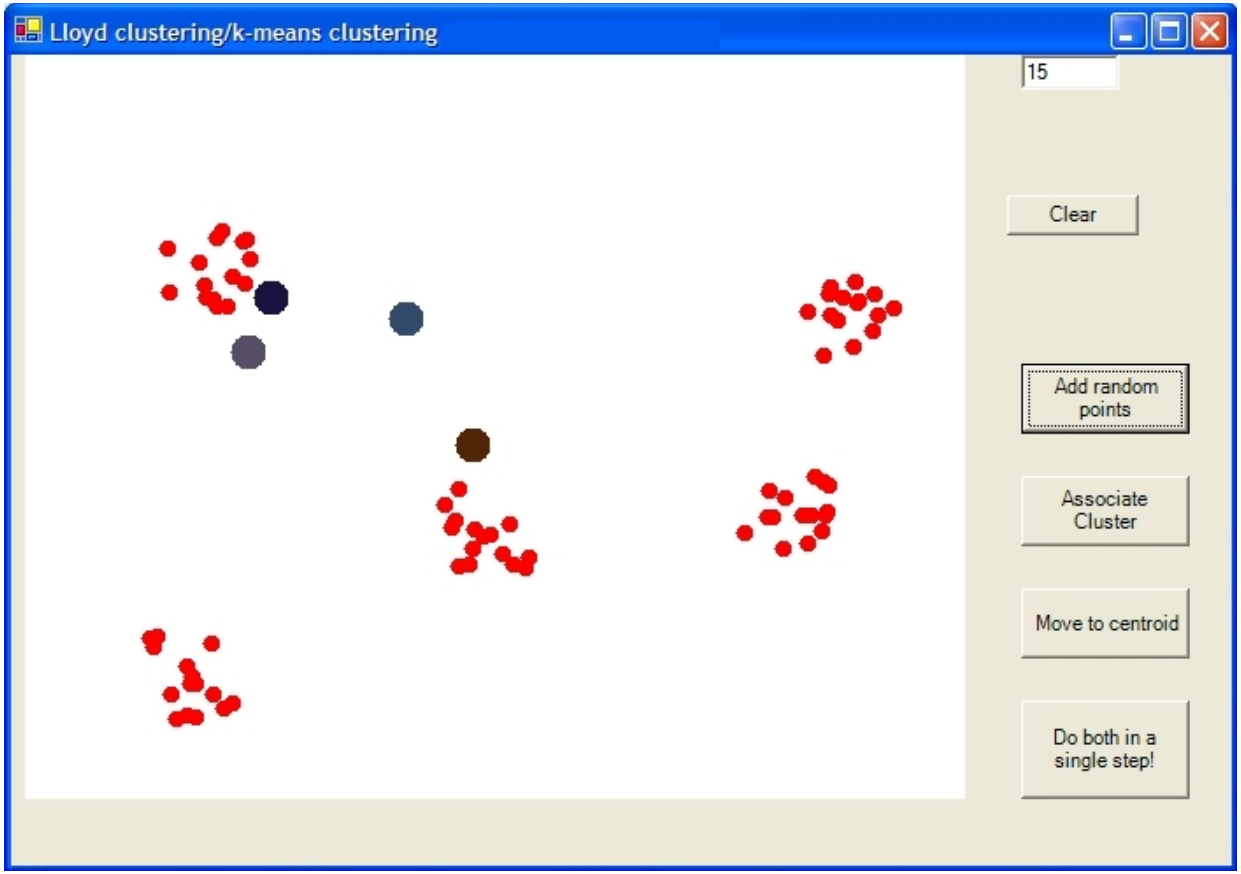
K-Means

- Choose a fixed number of clusters
- Choose cluster centers and point-cluster allocations to minimize error
- Can't do this by search, because there are too many possible allocations.

- Algorithm
  - fix cluster centers; allocate points to closest cluster
  - fix allocation; compute best cluster centers
- x could be any set of features for which we can compute a distance (careful about scaling)

$$\sum_{i \in \text{clusters}} \left\{ \sum_{j \in \text{elements of i'th cluster}} \left\| x_j - \mu_i \right\|^2 \right\}$$

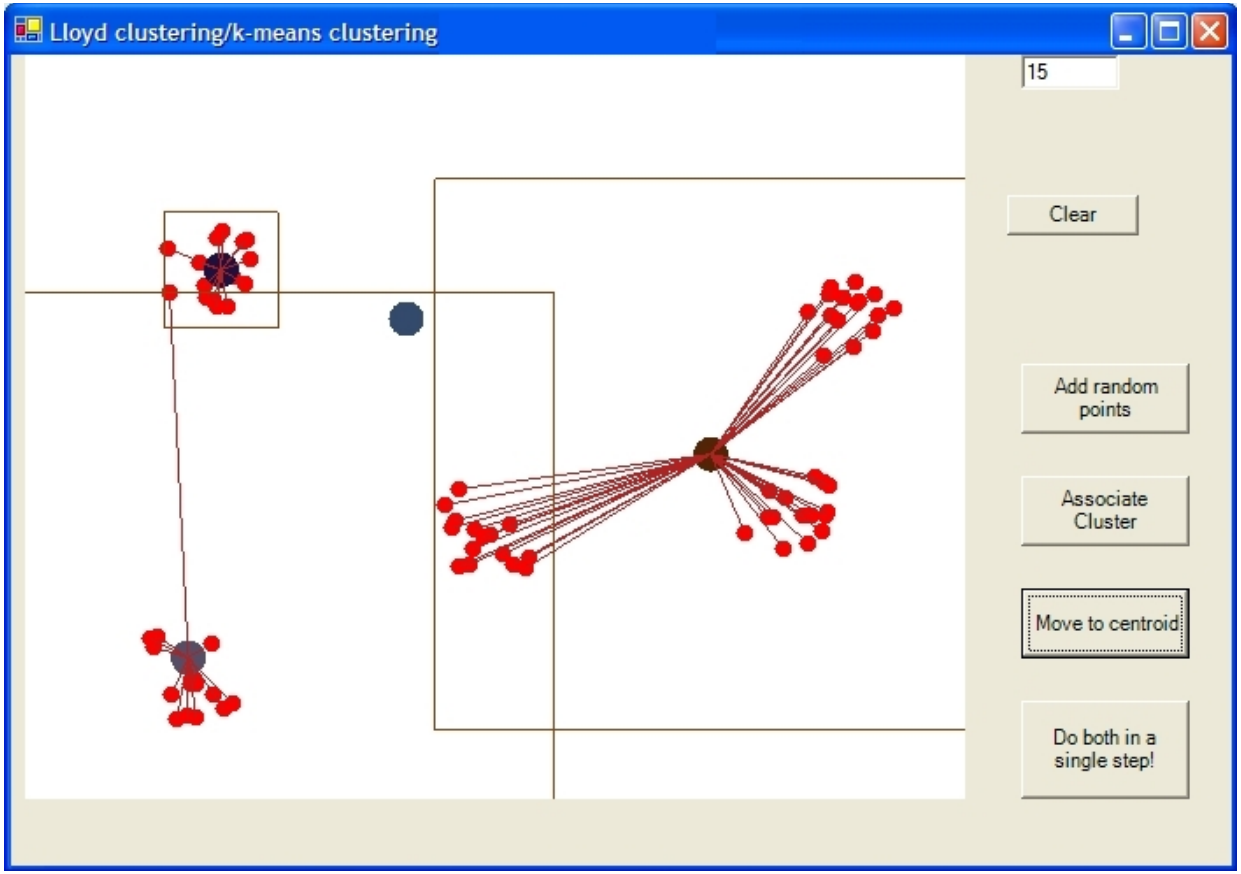
K-Means



The initial randomized centers and a number of points

Wikipedia

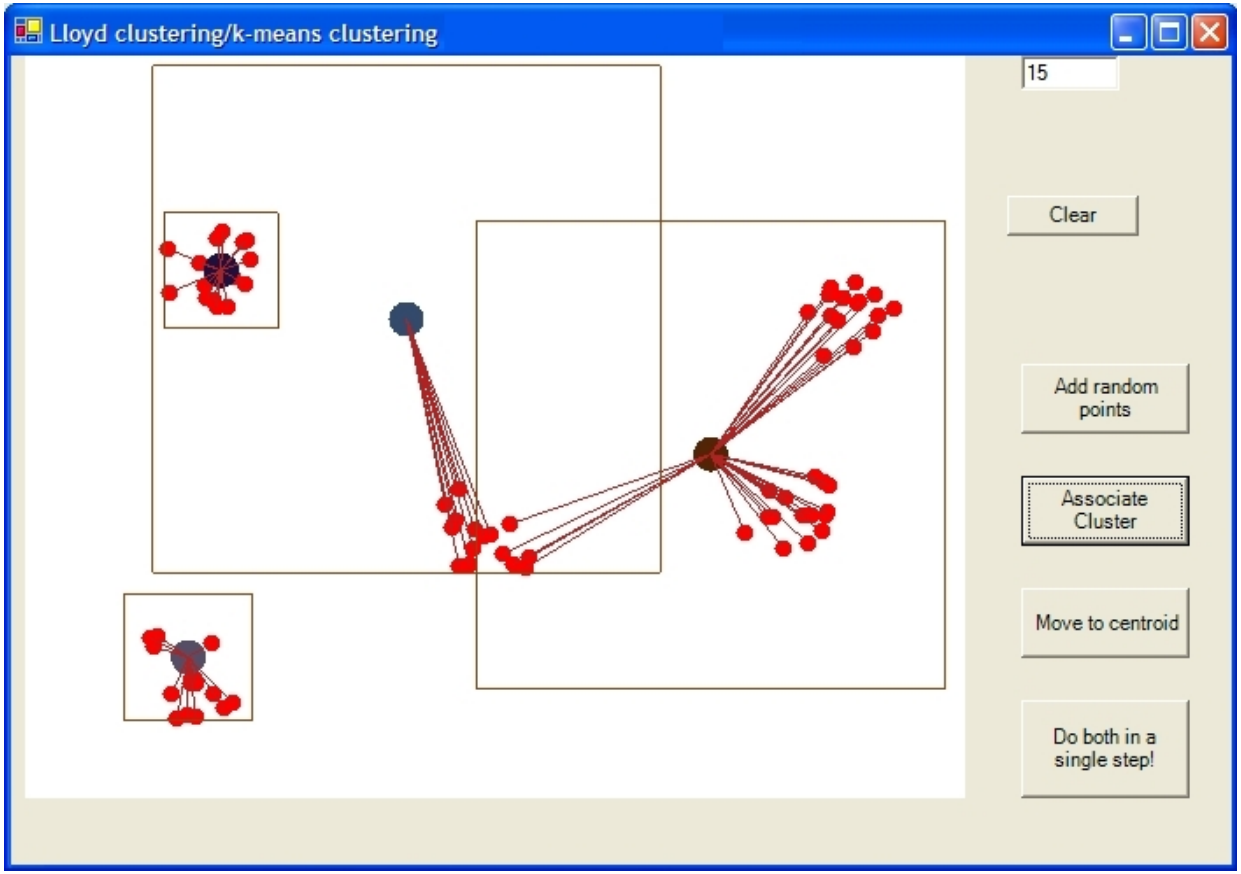
K-Means



Centers have been associated with the points and have been moved to the respective centroids

Wikipedia

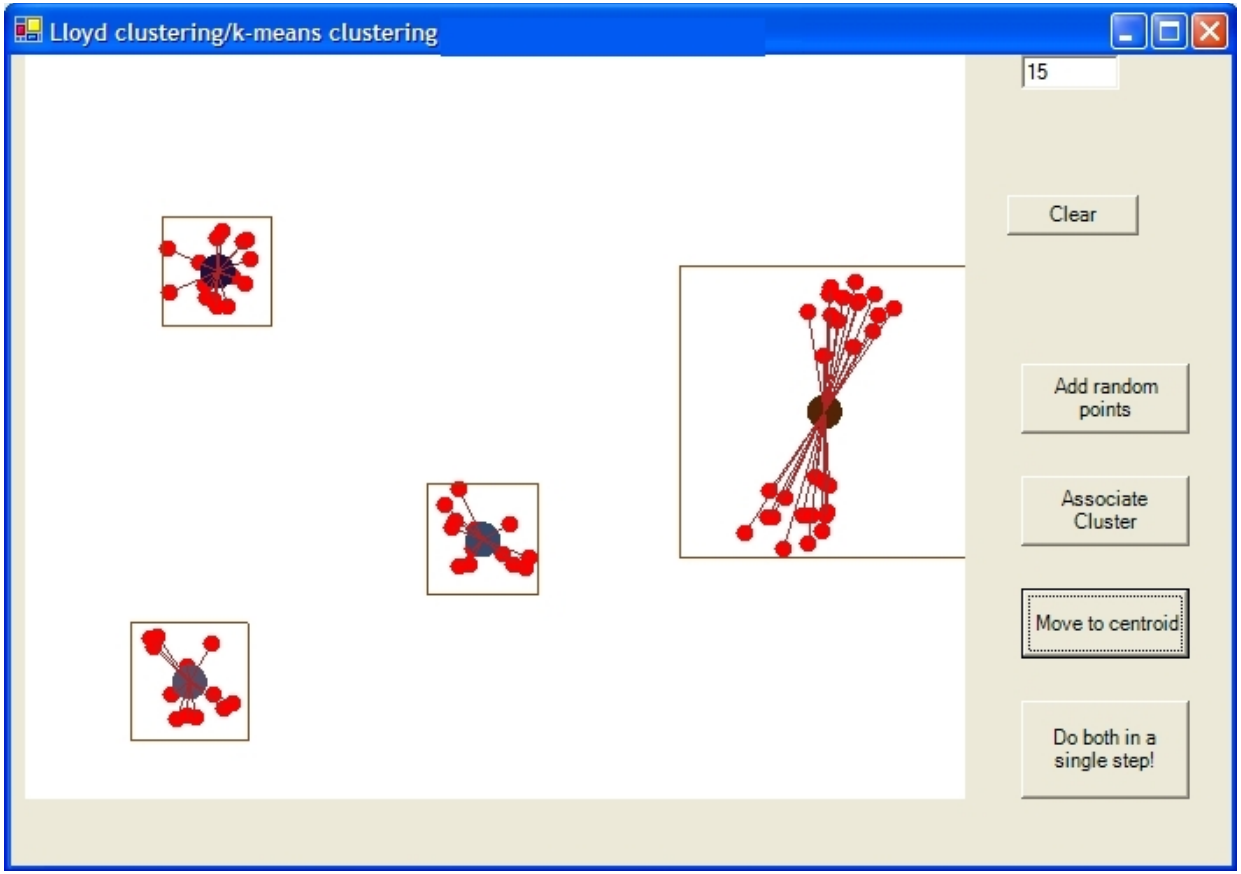
K-Means



Now, the association is shown in more detail, once the centroids have been moved.

Wikipedia

K-Means



Again, the centers are moved to the centroids of the corresponding associated points

Wikipedia

Image

Clusters on intensity

Clusters on color

K-means clustering using intensity alone and color alone

Expectation-Maximization

- Problems with K-means
  - "hard" association
  - No notion of "compactness" of cluster
  - No convergence proof

- EM is a general technique for inferring "missing data"
  - For us, the "missing data" is association

- A natural model is the Gaussian Mixture Model

  - $P(d) = \Sigma\, a_i\, p(d \mid o_i)$

The Algorithm for Clustering

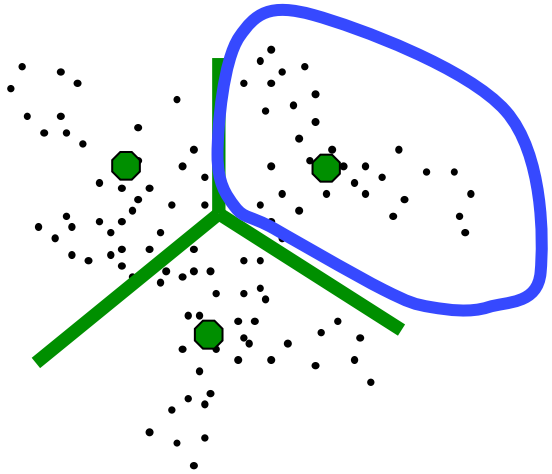
- The E- step (assuming we know the Gaussians)
  - $l_{j,k} = a_k P(d_j \mid m_k , s_k)$
  - Normalize to be a distribution for each j

- The M-step (assuming we know the association)

$$a_k = \text{sum}_j \; l_{j,k} \quad \text{<< normalize the a's after this}$$
$$m_k = \text{sum}_j \; l_{j,k} \, d_j / \text{sum}_j \; l_{j,k}$$
$$s_k = \text{sum}_j \; l_{j,k} \, d_j^2 - (m_k)^2 / \text{sum}_j \; l_{j,k}$$

Visual words

- Example: each group of patches belongs to the same visual word



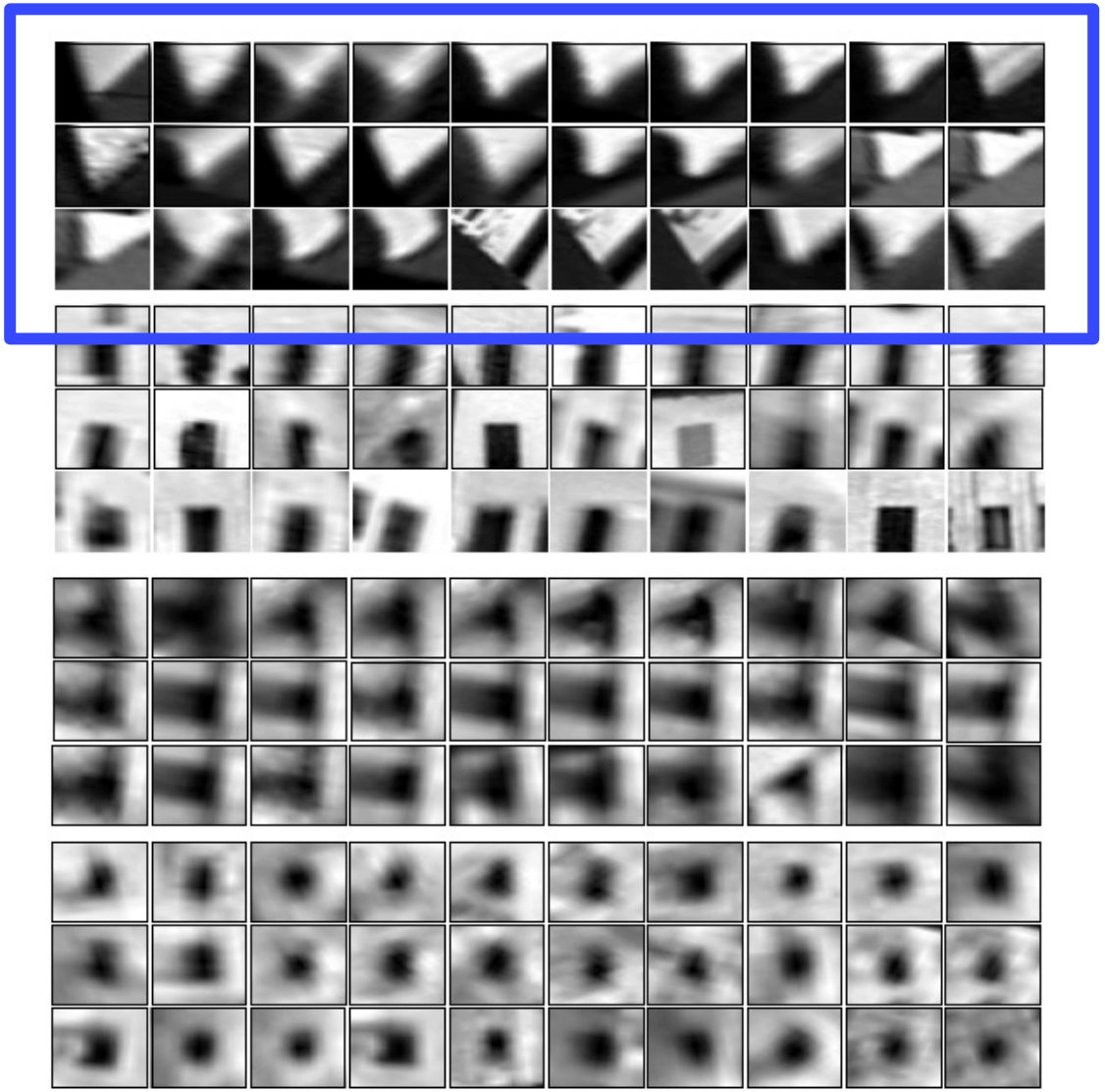
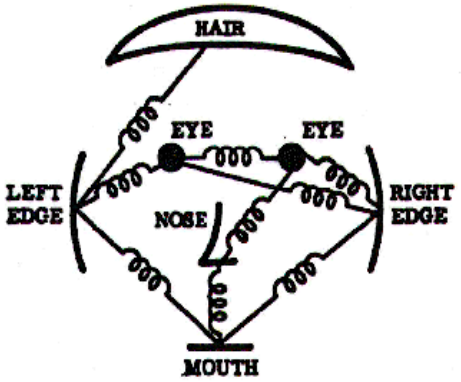
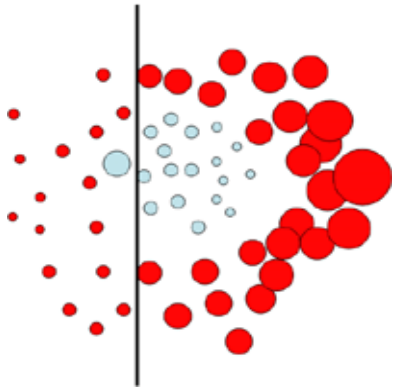


Figure from Sivic & Zisserman, ICCV 2003

Agenda

- Introduction
- **Bag of words models**
- Part-based models
- Discriminative methods
- Conclusions



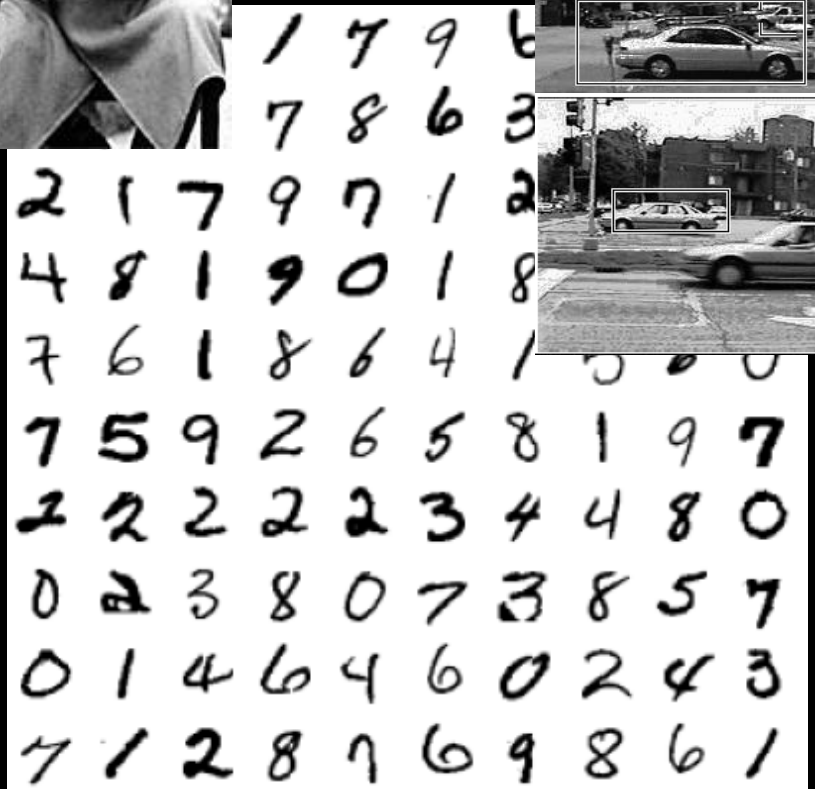


History: single object recognition

History: single object recognition

- Lowe, et al. 1999, 2003
- Mahamud and Herbert, 2000
- Ferrari, Tuytelaars, and Van Gool, 2004
- Rothganger, Lazebnik, and Ponce, 2004
- Moreels and Perona, 2005
- …

History: early object categorization



- Turk and Pentland, 1991
- Belhumeur et al. 1997
- Schneiderman et al. 2004
- Viola and Jones, 2000

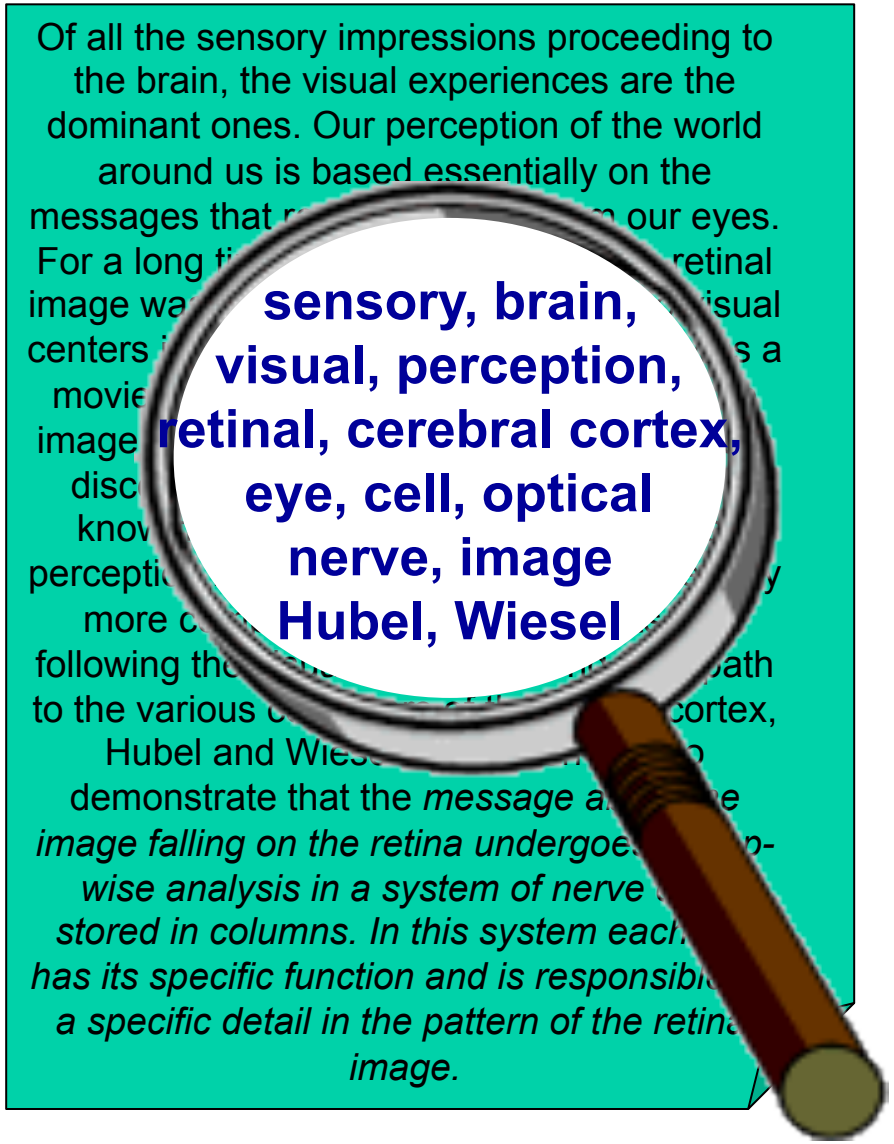
- Amit and Geman, 1999
- LeCun et al. 1998
- Belongie and Malik, 2002

- Schneiderman et al. 2004
- Argawal and Roth, 2002
- Poggio et al. 1993

Part 1: Bag-of-words models

by Li Fei-Fei (Stanford)

Analogy to documents



Related Work

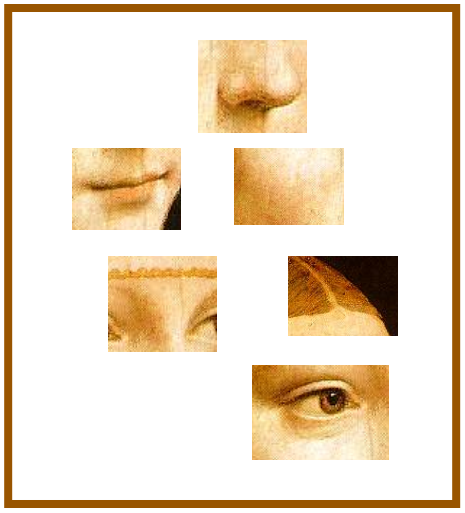
- Early "bag of words" models: mostly texture recognition
  - Cula & Dana, 2001; Leung & Malik 2001; Mori, Belongie & Malik, 2001; Schmid 2001; Varma & Zisserman, 2002, 2003; Lazebnik, Schmid & Ponce, 2003;
- Hierarchical Bayesian models for documents (pLSA, LDA, etc.)
  - Hoffman 1999; Blei, Ng & Jordan, 2004; Teh, Jordan, Beal & Blei, 2004
- Object categorization
  - Csurka, Bray, Dance & Fan, 2004; Sivic, Russell, Efros, Freeman & Zisserman, 2005; Sudderth, Torralba, Freeman & Willsky, 2005;
- Natural scene categorization
  - Vogel & Schiele, 2004; Fei-Fei & Perona, 2005; Bosch, Zisserman & Munoz, 2006

Object

Bag of ‘words’

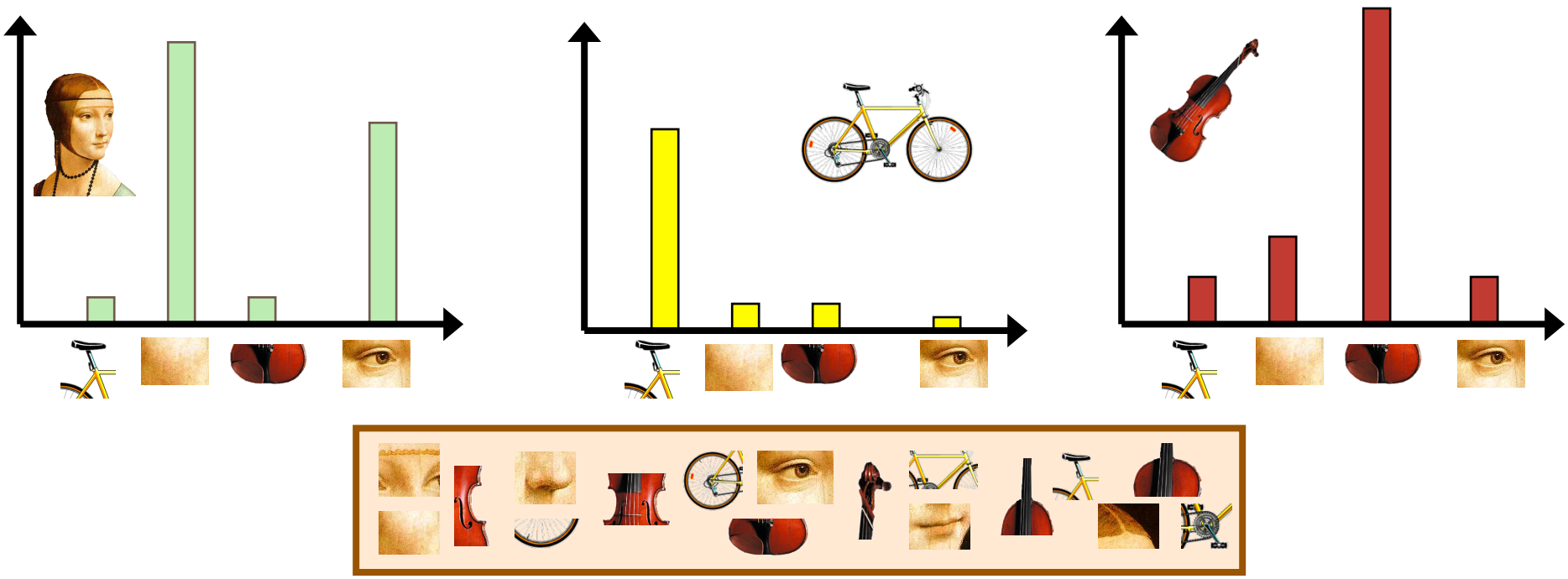
A clarification: definition of “BoW”

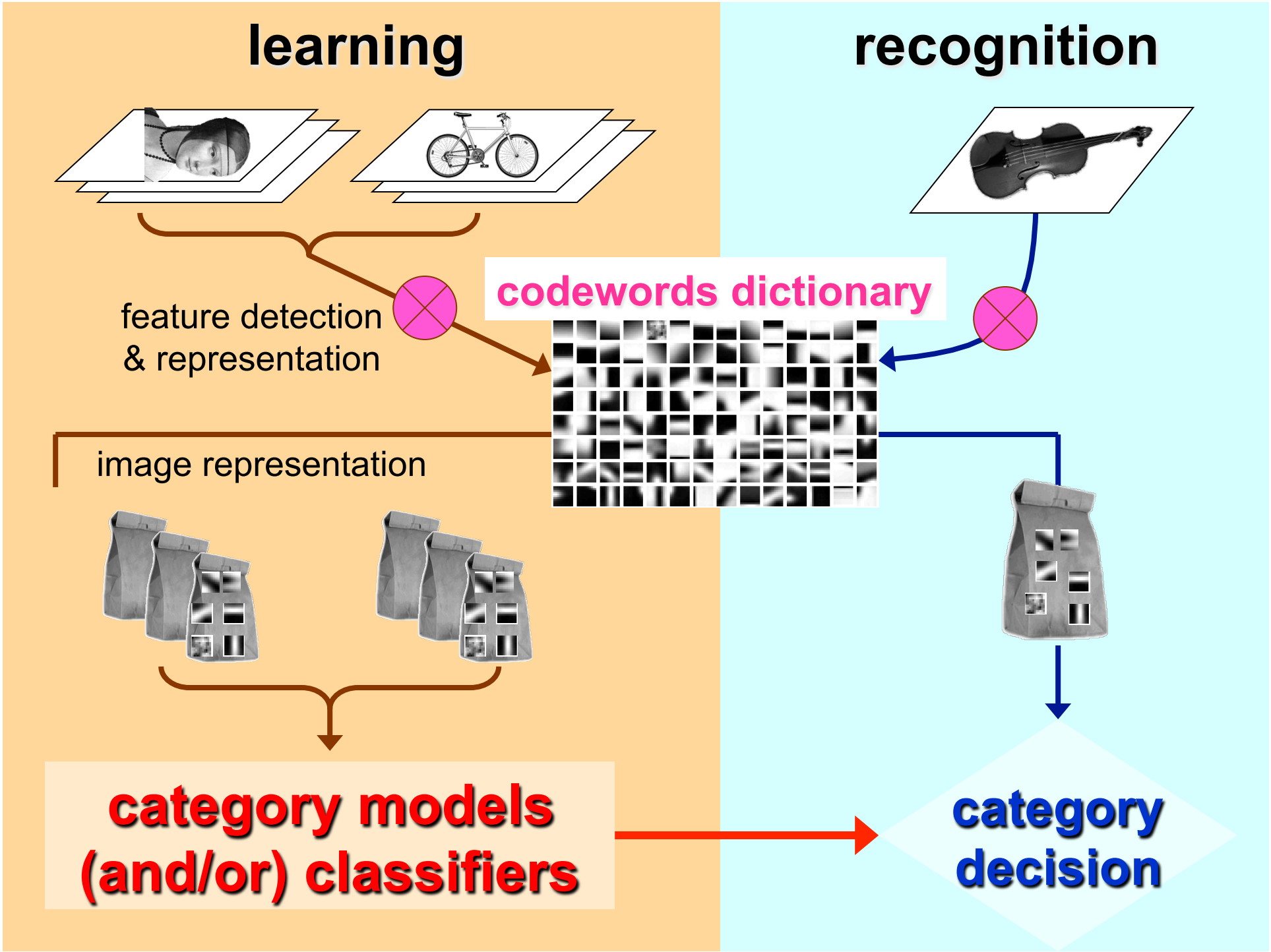
- Looser definition
  - Independent features



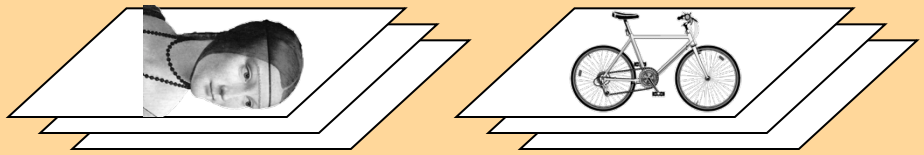
A clarification: definition of “BoW”

- Looser definition
  - Independent features
- Stricter definition
  - Independent features
  - histogram representation



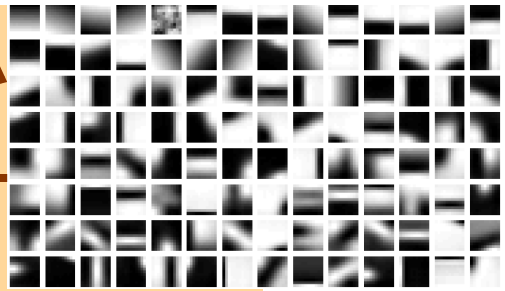

learning
recognition
feature detection
& representation
codewords dictionary
image representation
category models
(and/or) classifiers
category
decision

Representation



2.

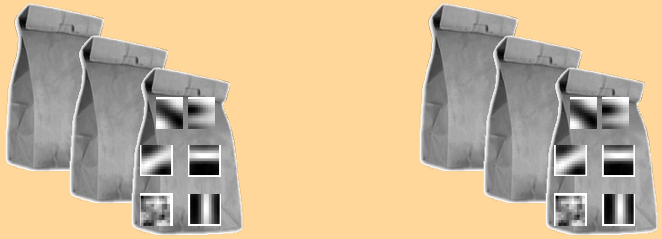
codewords dictionary



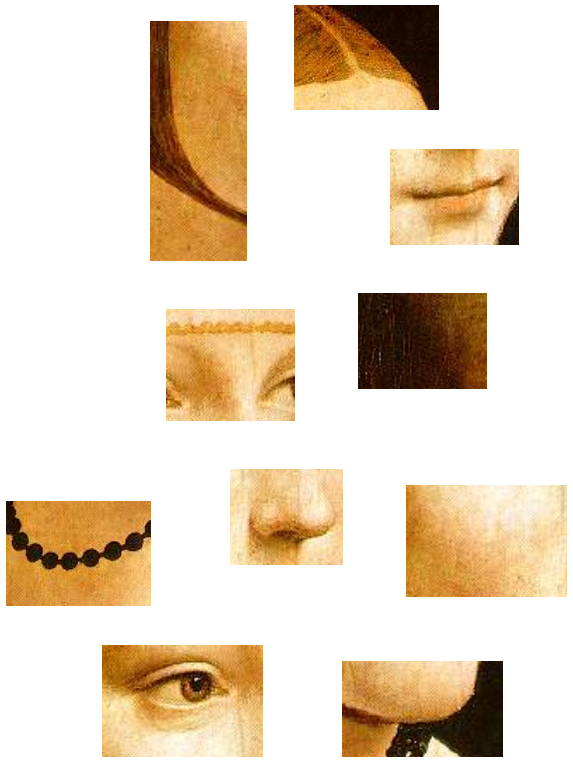
1. feature detection & representation

image representation

3.

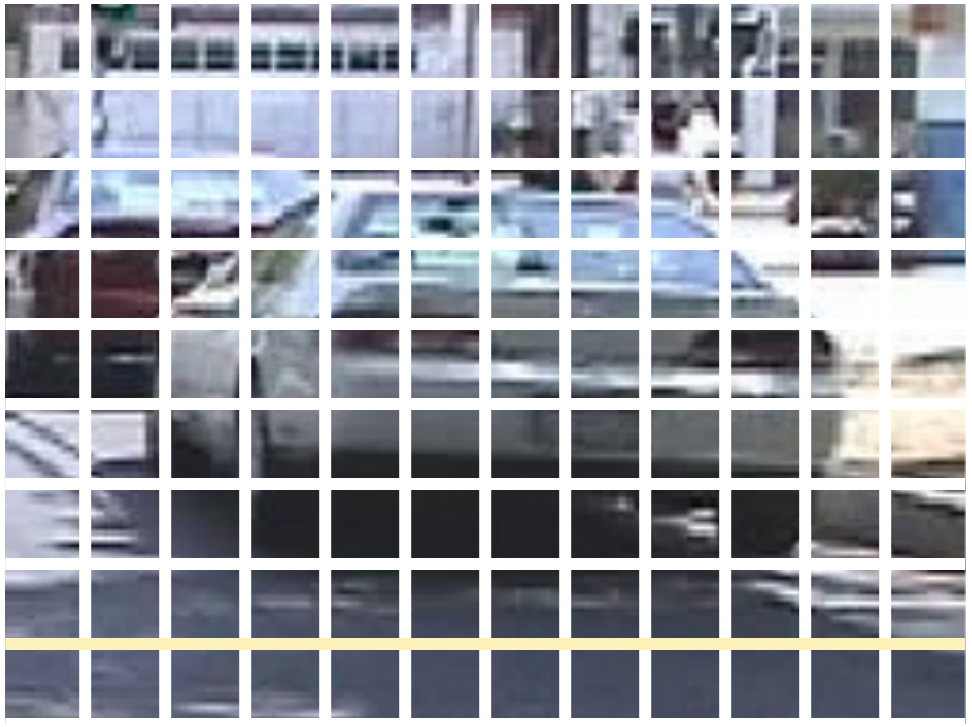


1.Feature detection and representation



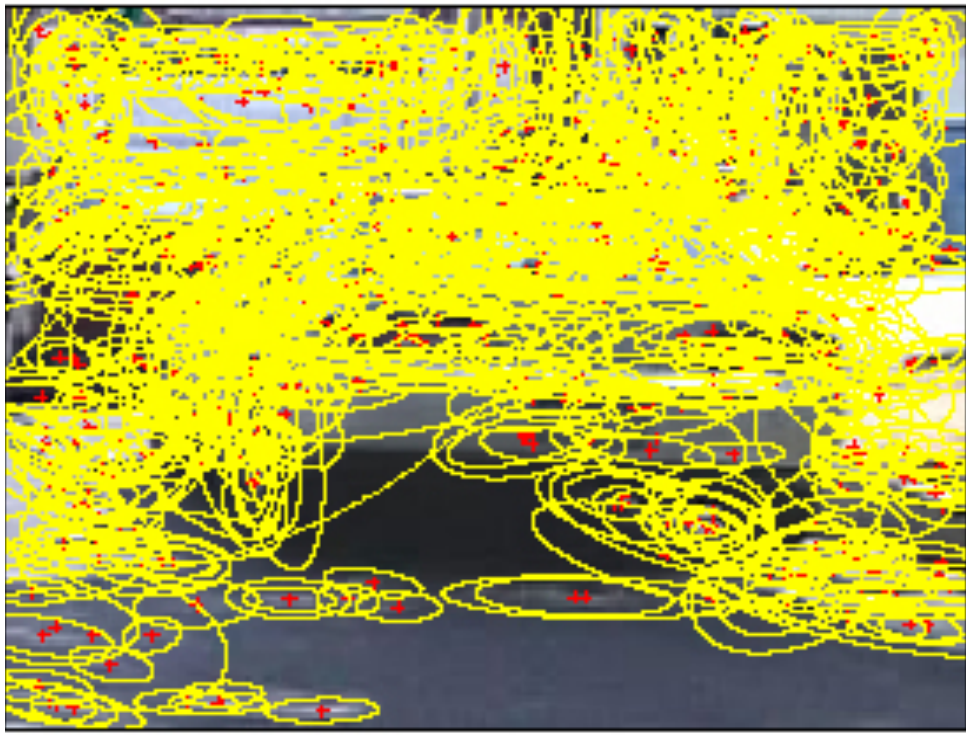
1.Feature detection and representation

- Regular grid
  - Vogel & Schiele, 2003
  - Fei-Fei & Perona, 2005



1.Feature detection and representation

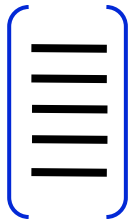
- Regular grid
  - Vogel & Schiele, 2003
  - Fei-Fei & Perona, 2005
- Interest point detector
  - Csurka, et al. 2004
  - Fei-Fei & Perona, 2005
  - Sivic, et al. 2005



1.Feature detection and representation

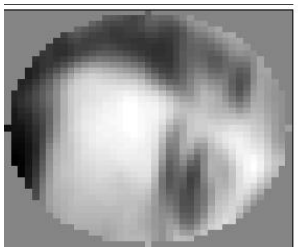
- Regular grid
  - Vogel & Schiele, 2003
  - Fei-Fei & Perona, 2005
- Interest point detector
  - Csurka, Bray, Dance & Fan, 2004
  - Fei-Fei & Perona, 2005
  - Sivic, Russell, Efros, Freeman & Zisserman, 2005
- Other methods
  - Random sampling (Vidal-Naquet & Ullman, 2002)
  - Segmentation based patches (Barnard, Duygulu, Forsyth, de Freitas, Blei, Jordan, 2003)

1.Feature detection and representation

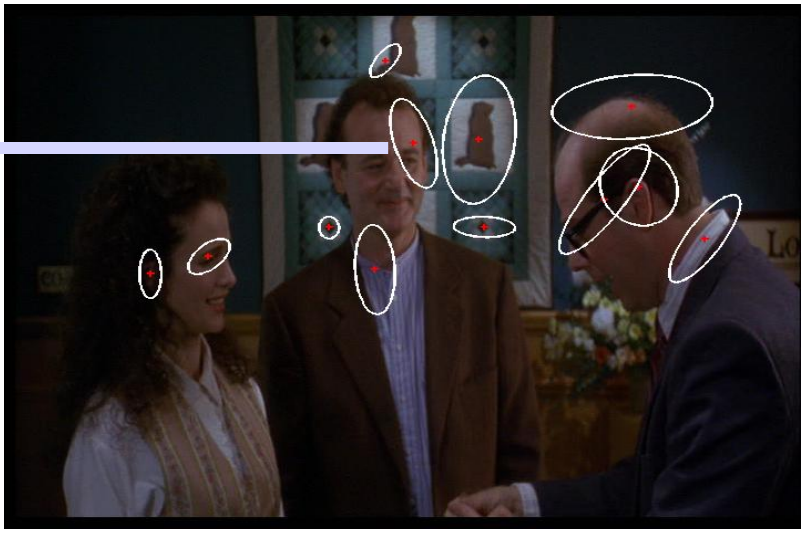


Compute SIFT descriptor

[Lowe'99]



Normalize patch



Detect patches

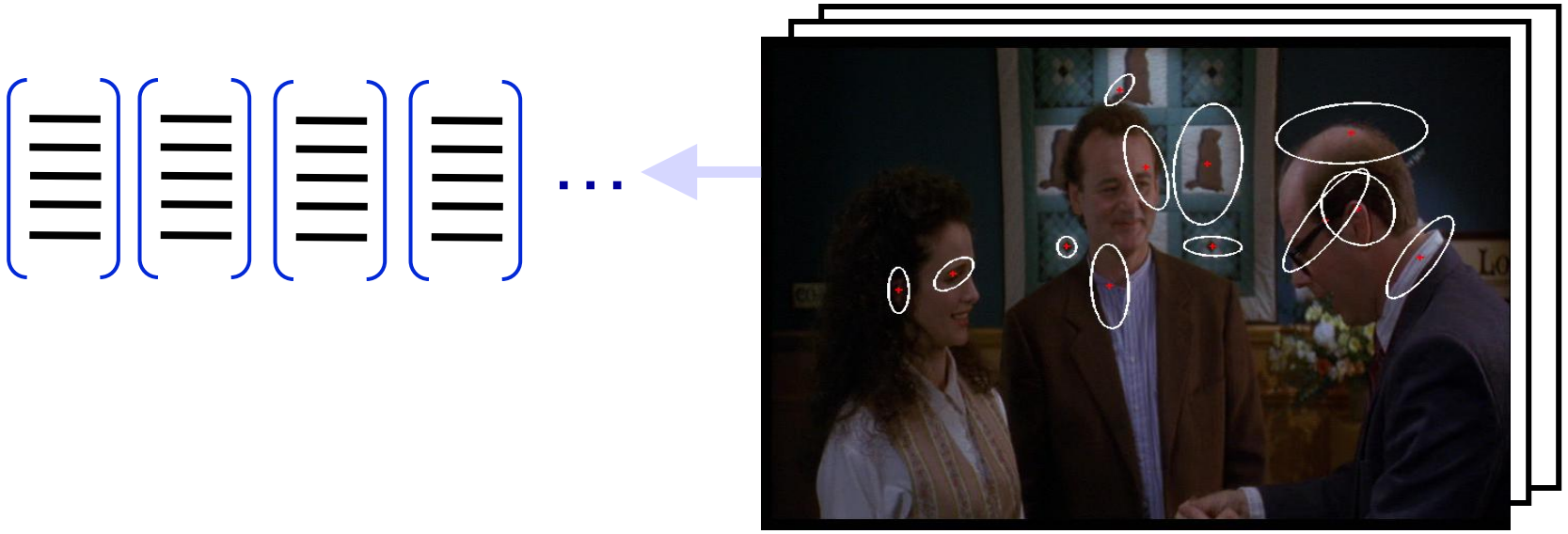
[Mikojaczyk and Schmid '02]

[Mata, Chum, Urban & Pajdla, '02]

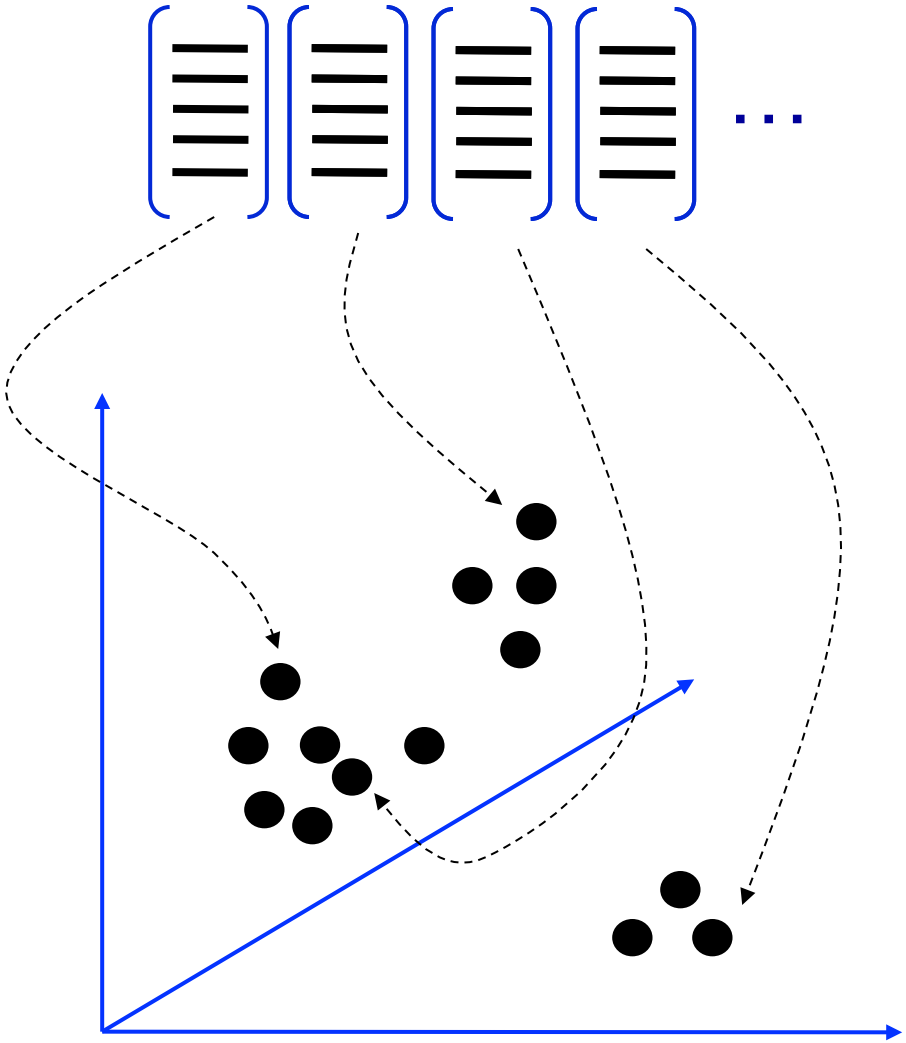
[Sivic & Zisserman, '03]

Slide credit: Josef Sivic

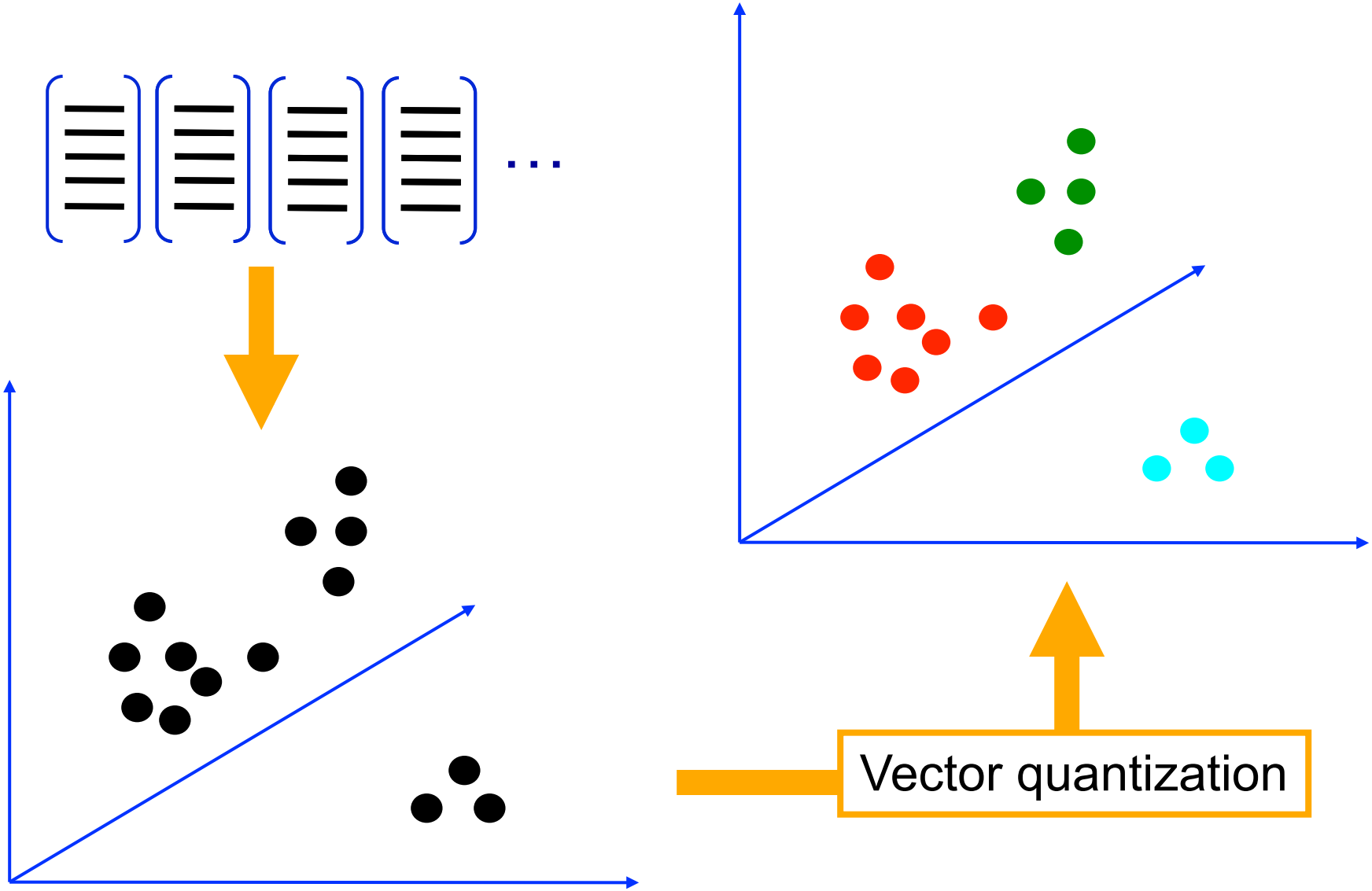
1.Feature detection and representation



2. Codewords dictionary formation



2. Codewords dictionary formation

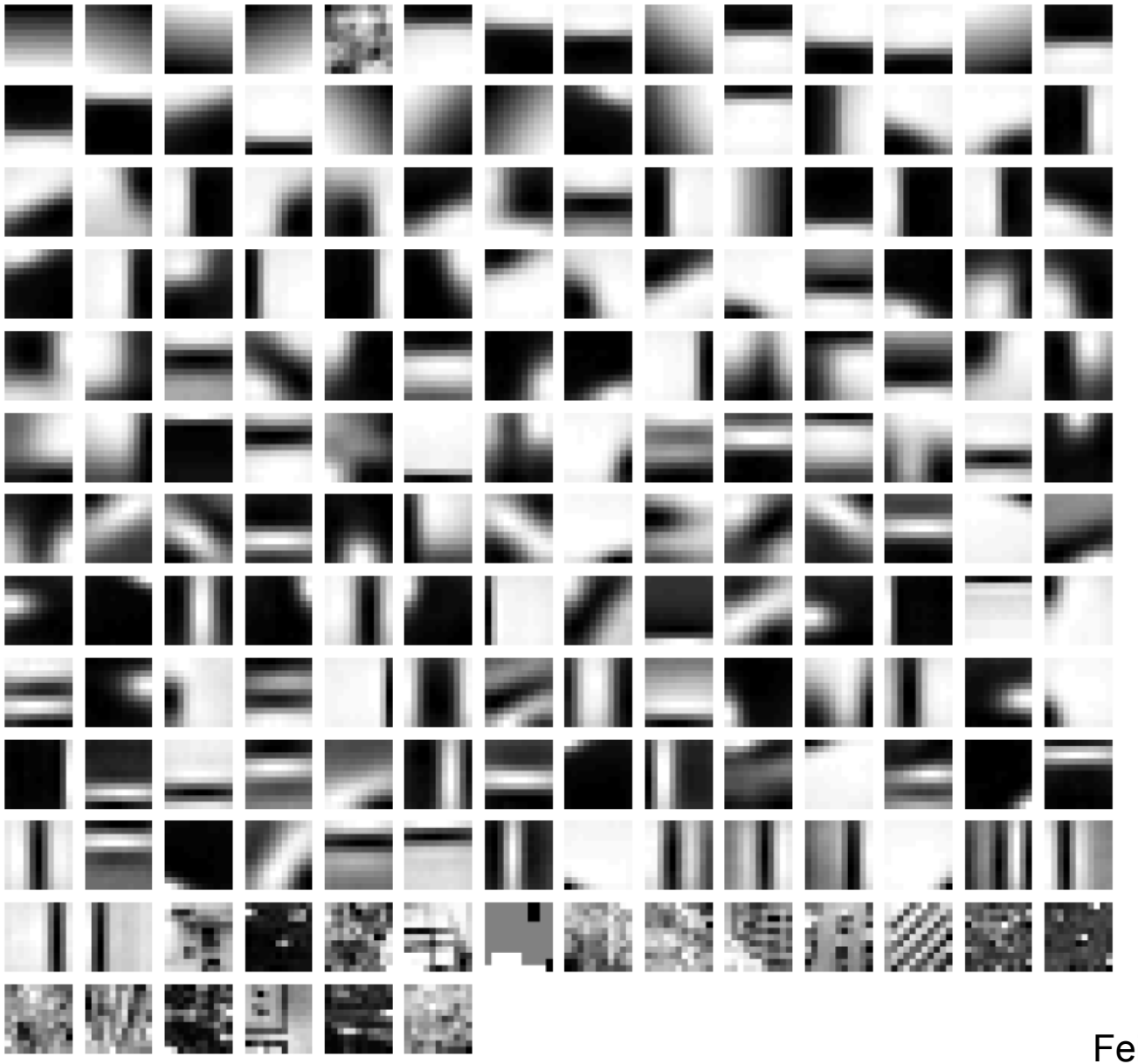


Slide credit: Josef Sivic

Building Code Words

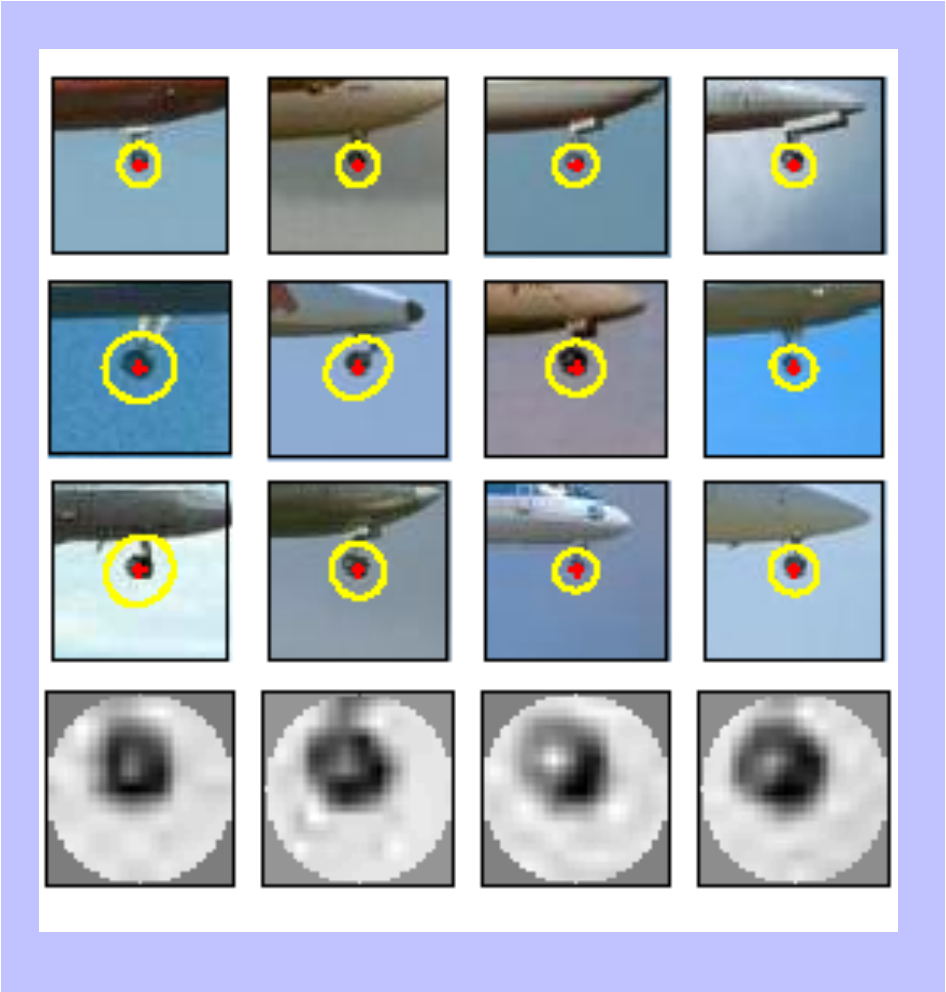
- Basic idea: common categories should have "clusters" of similar features
- Find the clusters, and then register which clusters tend to support which categories
- A common approach is to use K-means or EM to do this.

2. Codewords dictionary formation



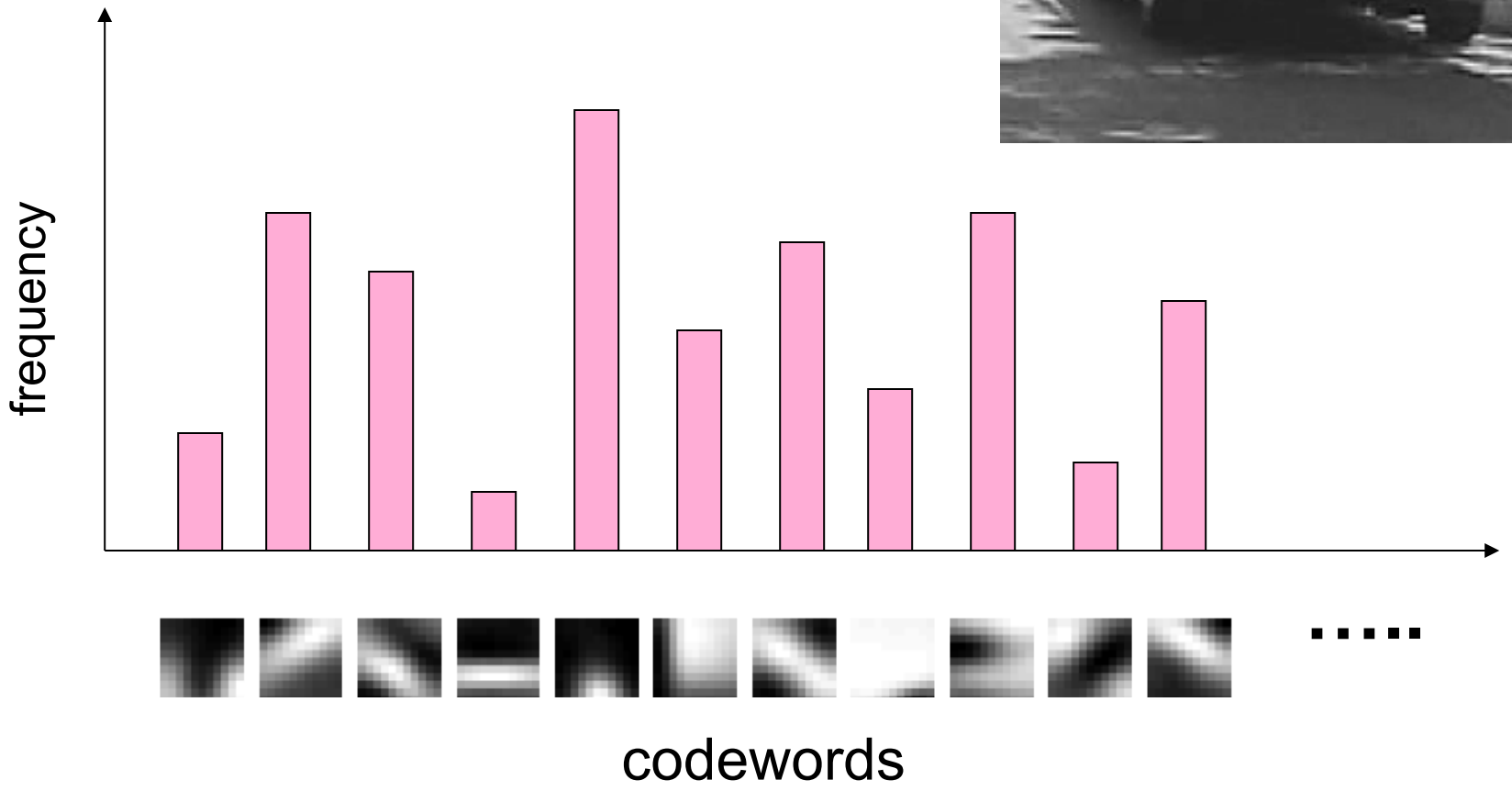
Fei-Fei et al. 2005

Image patch examples of codewords

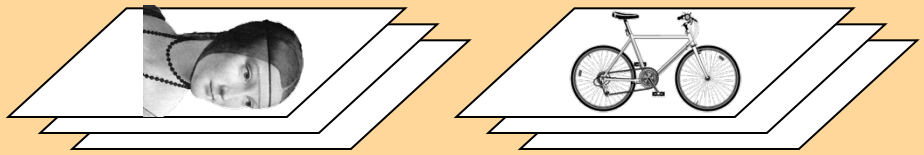


Sivic et al. 2005

3. Image representation

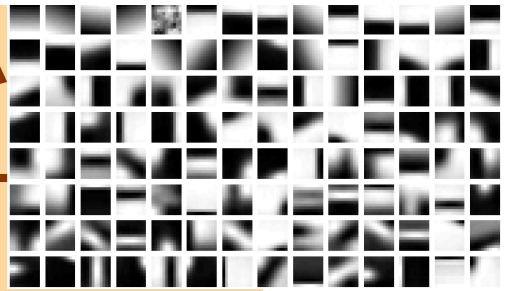


Representation



2.

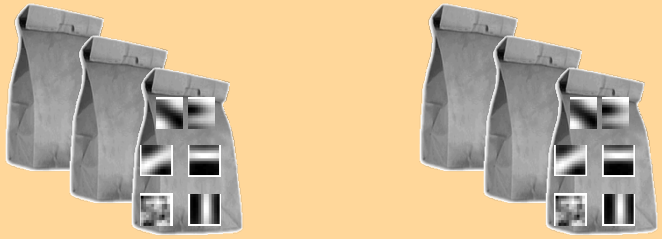
codewords dictionary



1. feature detection & representation

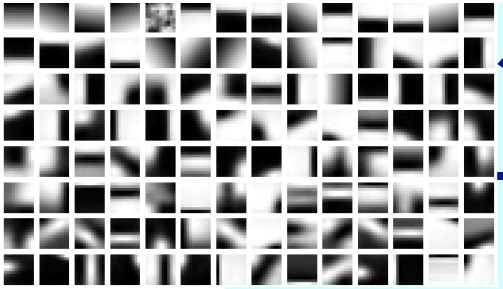
image representation

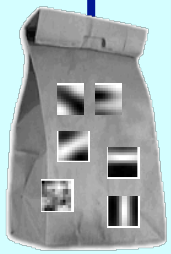
3.



Learning and Recognition

codewords dictionary



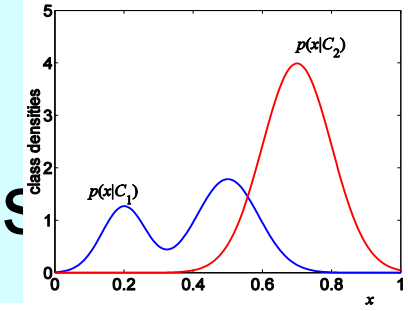


category models (and/or) classifiers

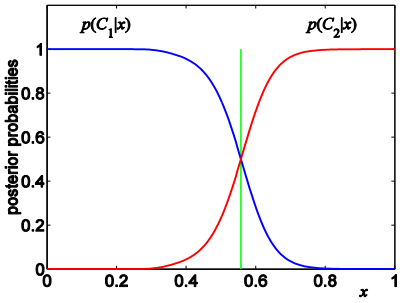
category decision

Learning and Recognition

1. Generative method:
   - graphical models

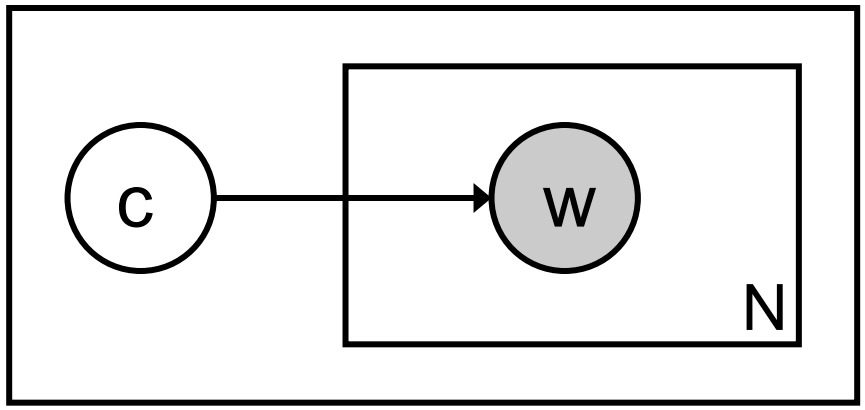


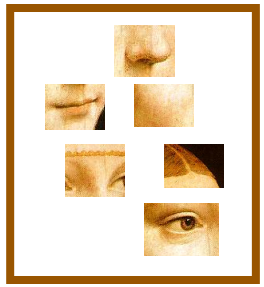
2. Discriminative method:
   - SVM



category models (and/or) classifiers

Case #1: the Naïve Bayes model





$$c^* = \arg\max_c p(c \mid w) \propto p(c)\, p(w \mid c) = p(c) \prod_{n=1}^{N} p(w_n \mid c)$$

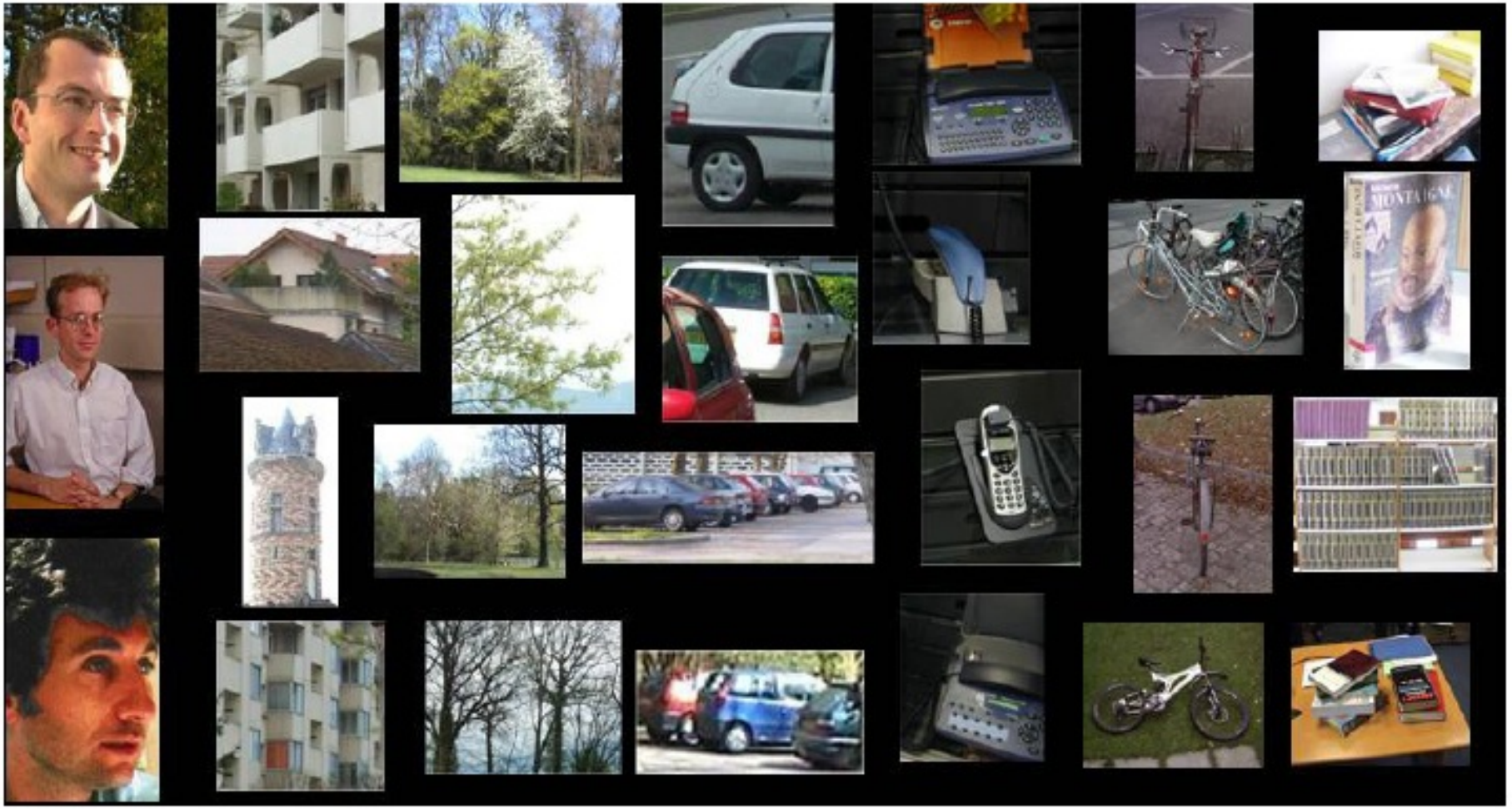
Object class decision

Prior prob. of the object classes

Image likelihood given the class

Csurka et al. 2004

Our in-house database contains 1776 images in seven classes[1]: faces, buildings, trees, cars, phones, bikes and books. Fig. 2 shows some examples from this dataset.



Csurka et al. 2004

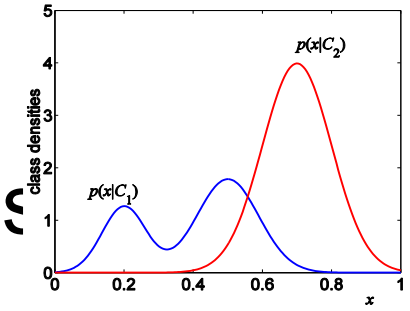
Table 1. Confusion matrix and the mean rank for the best vocabulary (*k=1000*).

True classes →	*faces*	*buildings*	*trees*	*cars*	*phones*	*bikes*	*books*
faces	**76**	4	2	3	4	4	13
buildings	2	**44**	5	0	5	1	3
trees	3	2	**80**	0	0	5	0
cars	4	1	0	**75**	3	1	4
phones	9	15	1	16	**70**	14	11
bikes	2	15	12	0	8	**73**	0
books	4	19	0	6	7	2	**69**
Mean ranks	1.49	1.88	1.33	1.33	1.63	1.57	1.57

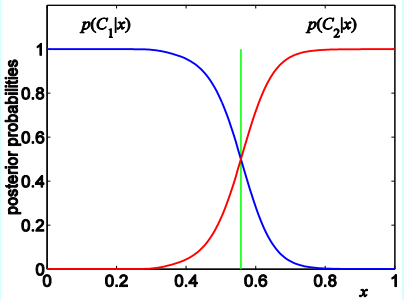
Csurka et al. 2004

Learning and Recognition

1. Generative method:
   - graphical models

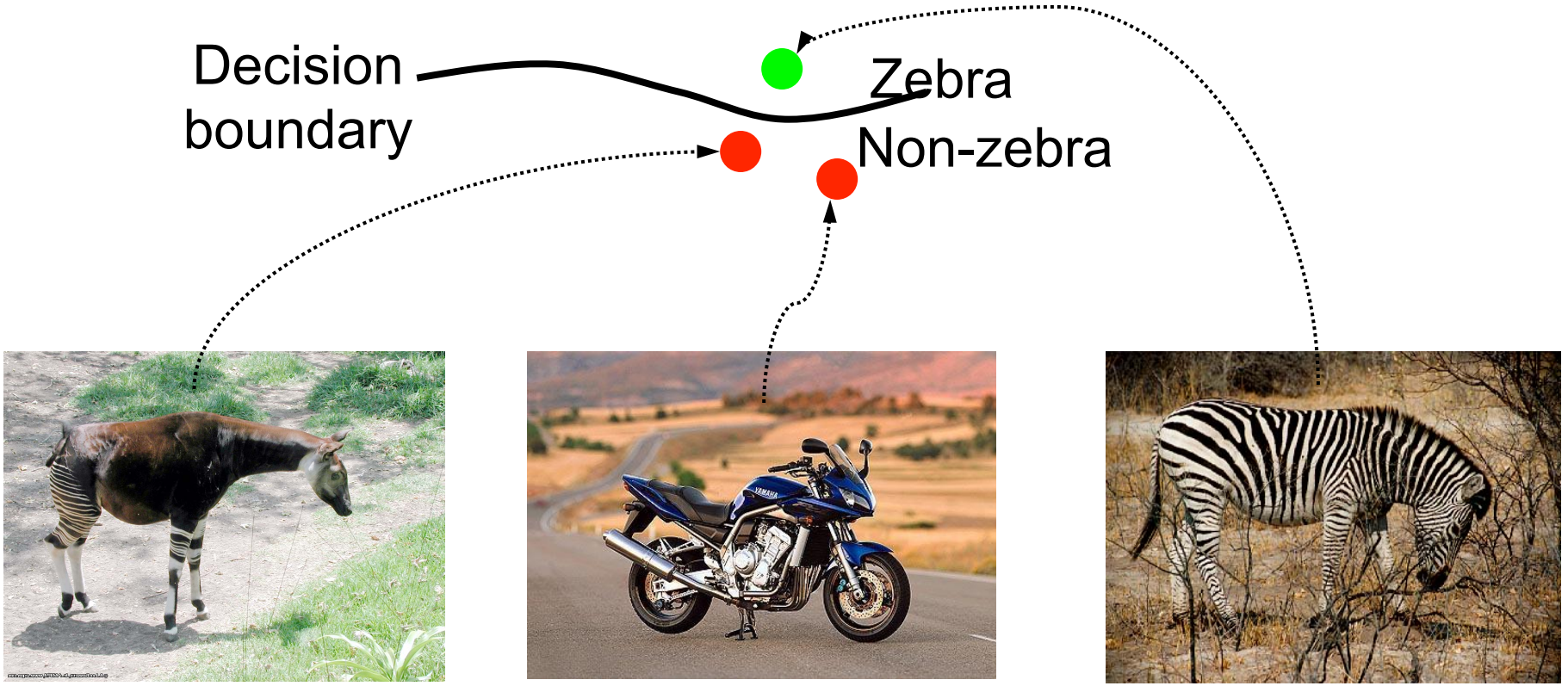


2. Discriminative method:
   - SVM



category models (and/or) classifiers

Discriminative methods based on ‘bag of words’ representation



Object recognition results

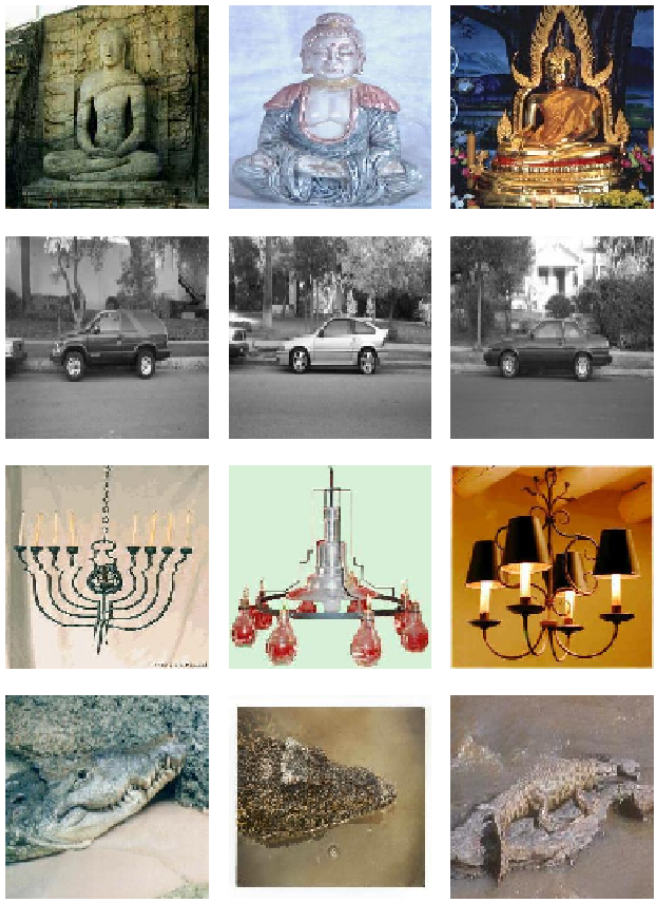
- ETH-80 database
  8 object classes
  (*Eichhorn and Chapelle 2004*)
- Features:
  - Harris detector
  - PCA-SIFT descriptor, *d*=10

Kernel	Complexity	Recognition rate
Match *[Wallraven et al.]*	$O(dm^2)$	84%
Bhattacharyya affinity *[Kondor & Jebara]*	$O(dm^3)$	85%
Pyramid match	$O(dmL)$	84%

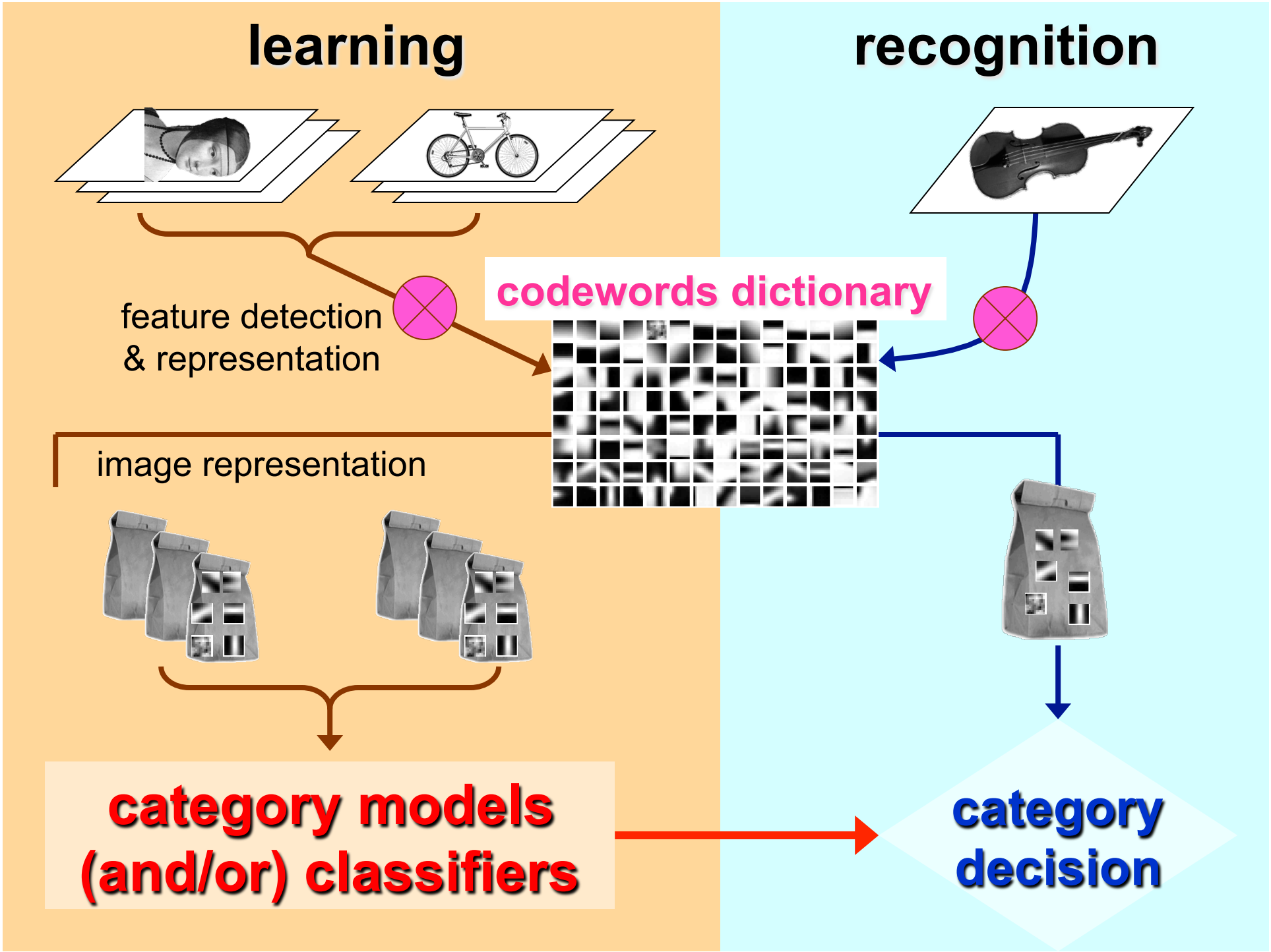
Slide credit: Kristen Grauman

Object recognition results

- Caltech objects database
  101 object classes
- Features:
  - SIFT detector
  - PCA-SIFT descriptor, *d*=10
- 30 training images / class
- 43% recognition rate
  (1% chance performance)
- 0.002 seconds per match



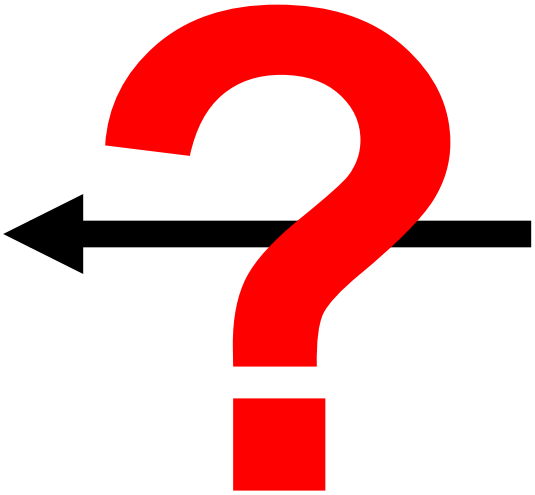
Slide credit: Kristen Grauman


learning
recognition
feature detection
& representation
codewords dictionary
image representation
category models
(and/or) classifiers
category
decision

CODE!

http://people.csail.mit.edu/
fergus/iccv2005/bagwords.html

What about spatial info?



Example: Fergus 2003

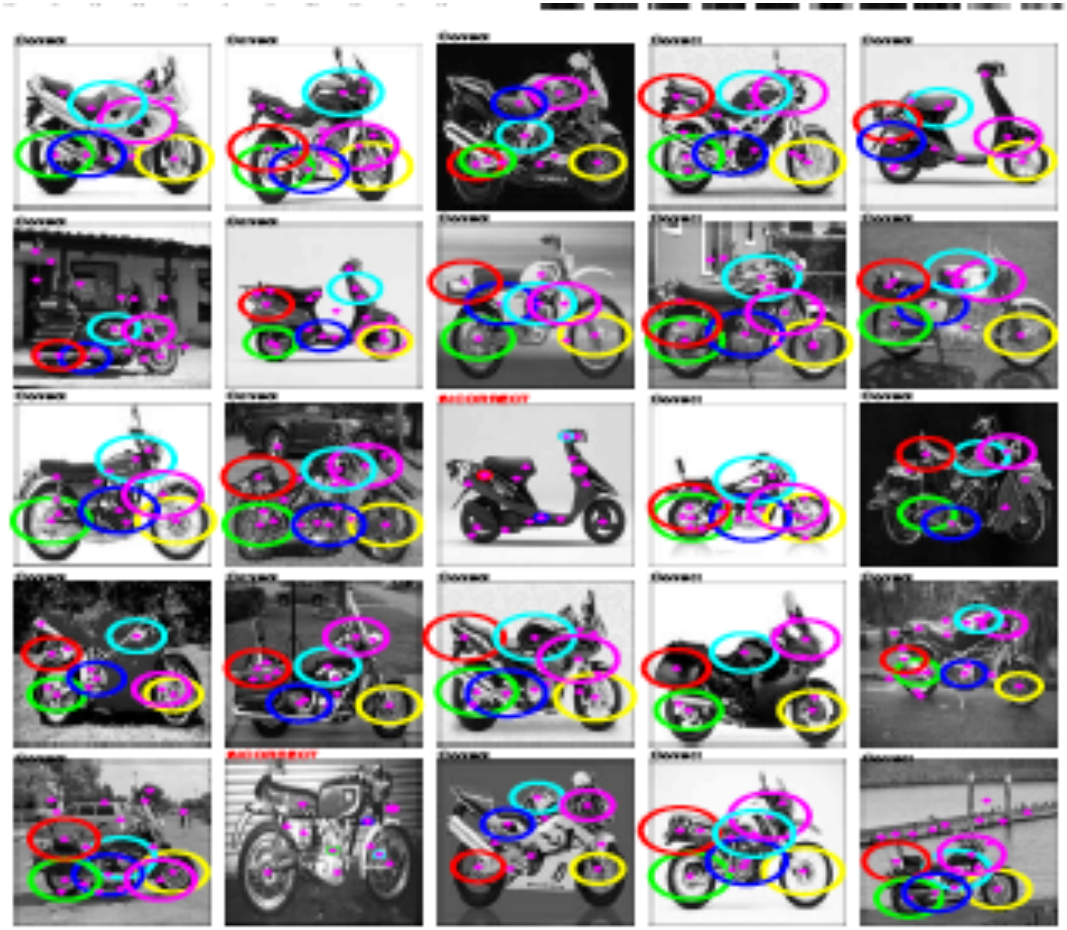


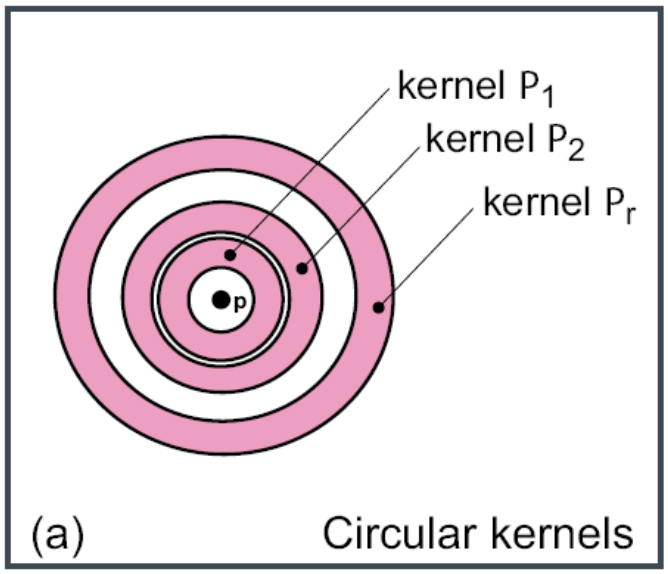
Figure 6: A typical motorbike model with 6 parts. Note the clear identification of the front and rear wheels, along with other parts such as the fuel tank.

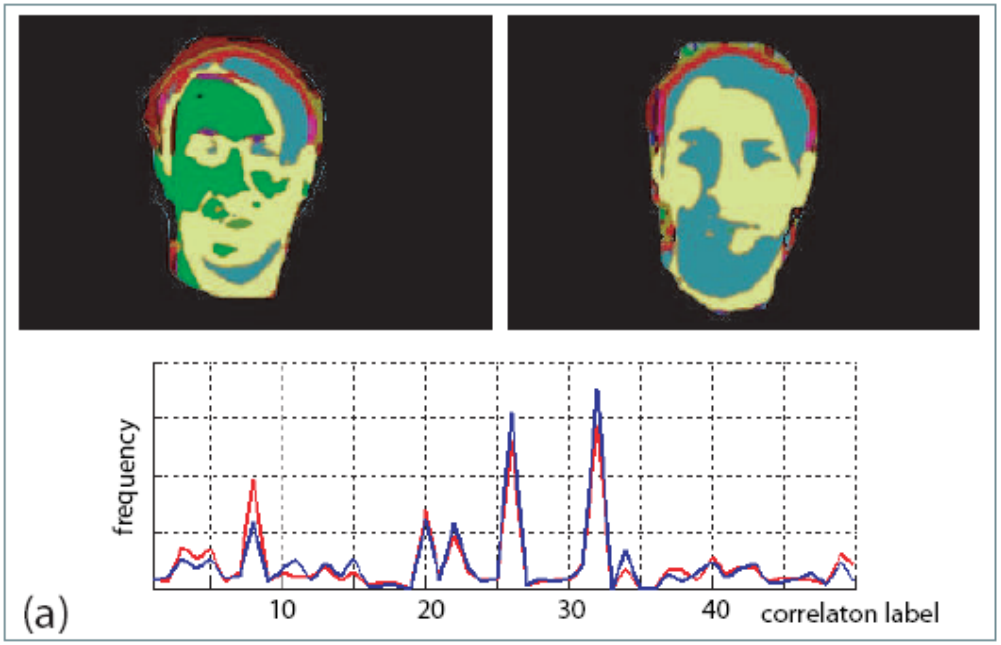
Results, Fergus 2003

Dataset	Total size of dataset	Object width (pixels)	Motorbike model	Face model	Airplane model	Cat model
Motorbikes	800	200	92.5	50	51	56
Faces	435	300	33	96.4	32	32
Airplanes	800	300	64	63	90.2	53
Spotted Cats	200	80	48	44	51	90 .0

What about spatial info?

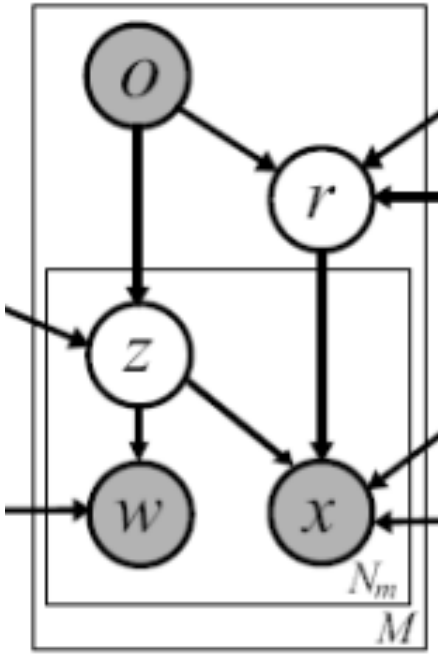
- Feature level
  - Spatial influence through correlogram features: Savarese, Winn and Criminisi, CVPR 2006

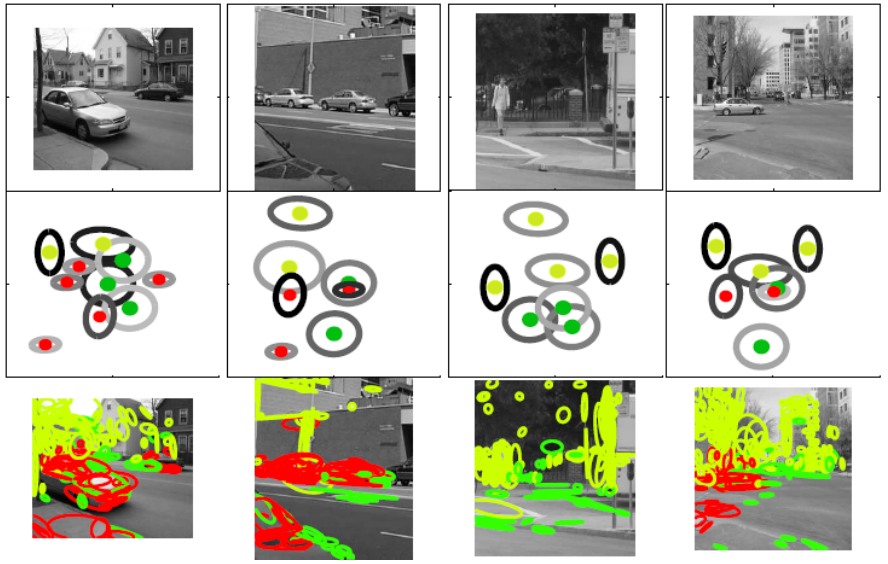




What about spatial info?

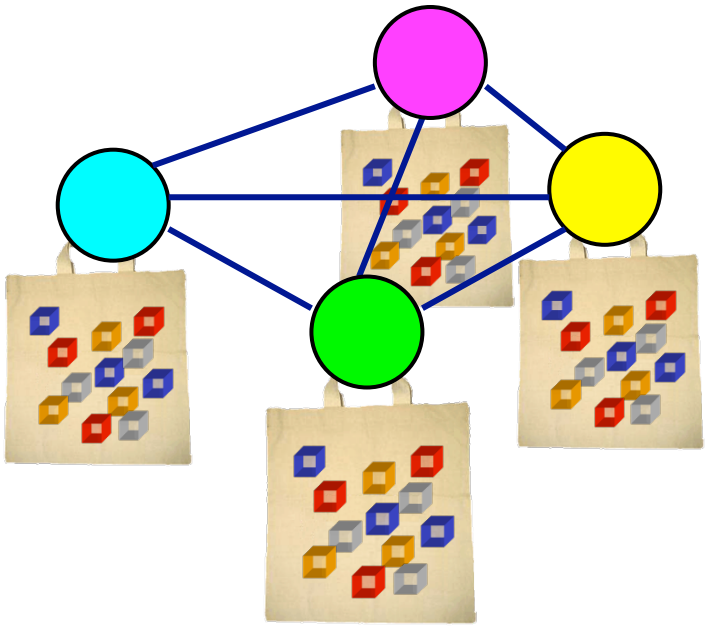
- Feature level
- Generative models
  - Sudderth, Torralba, Freeman & Willsky, 2005, 2006
  - Niebles & Fei-Fei, CVPR 2007

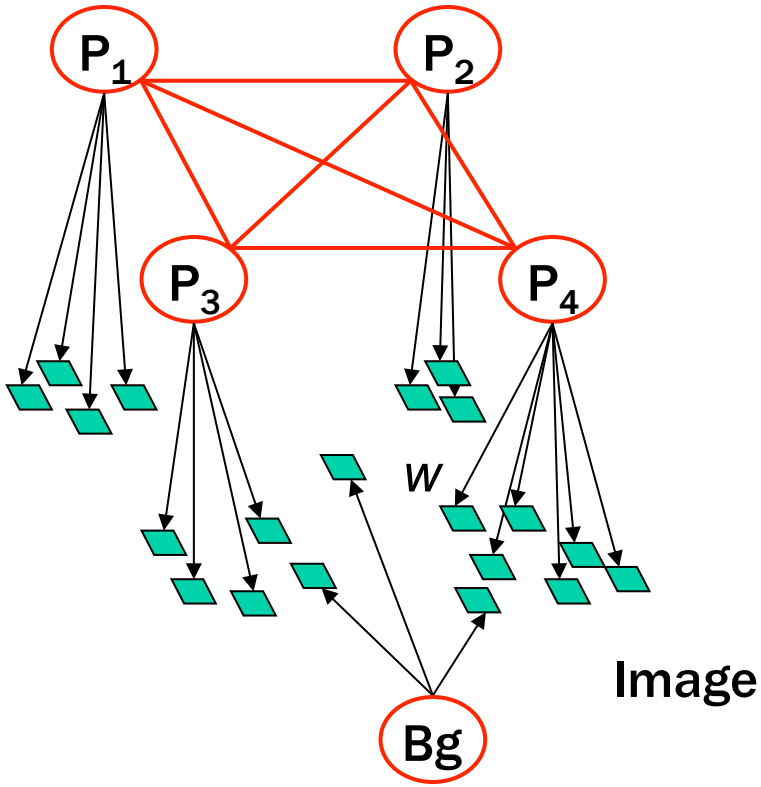




What about spatial info?

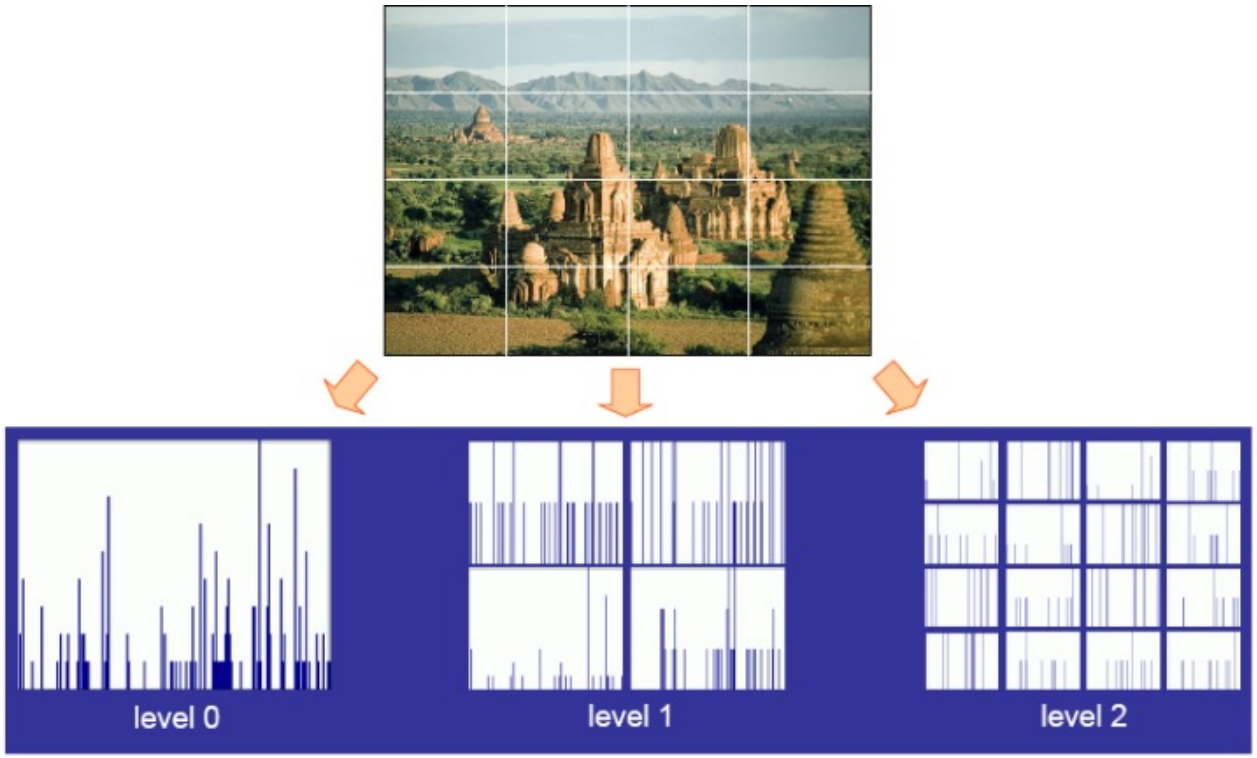
- Feature level
- Generative models
  - Sudderth, Torralba, Freeman & Willsky, 2005, 2006
  - Niebles & Fei-Fei, CVPR 2007





What about spatial info?

- Feature level
- Generative models
- Discriminative methods
  - Lazebnik, Schmid & Ponce, 2006



Summary

- Object recognition/categorization is a rapidly evolving area
- Current systems are getting to the point they may be useful in real applications.
- Much more remains to be done in understanding how to move to the next level of performance.