# Recognition:

# A Little Bit of History

# Flying Through the History of Recognition

- We will do a quick fast-forward through the history of recognition
- For every type of approach, try to factor out the time when it was done. Why?
  - Because in the old days people didn't have enough computational resources
  - They didn't have enough or even any data
  - Machine Learning techniques weren't as powerful yet, or at least the Vision researchers haven't learned them yet
- What makes a good researcher:
  - Recognizing good ideas
  - Figuring out why something doesn't work and what has the potential of making it to work
  - Taking risks
- As we go through history, try to spot good ideas!

# Textbook

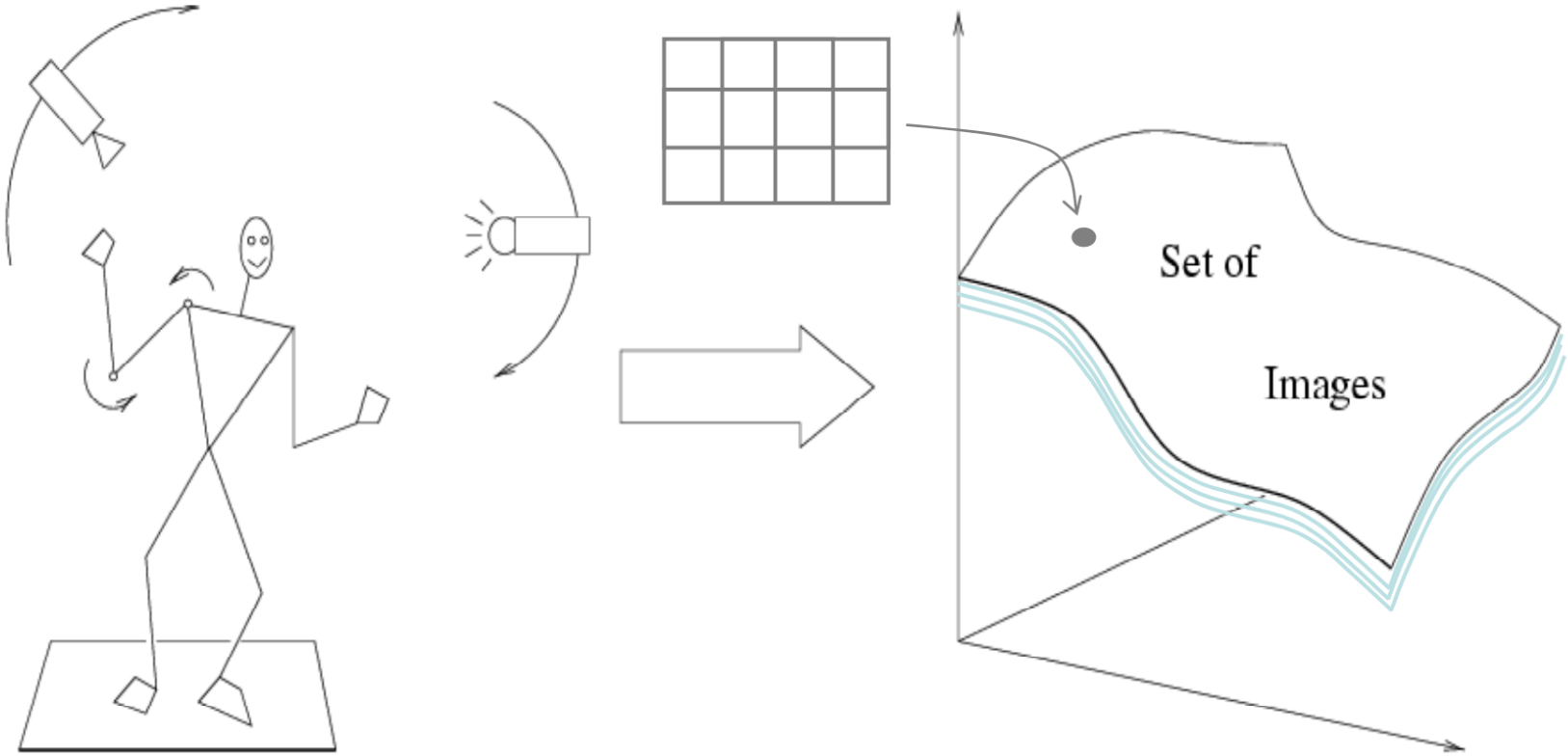
This paper has a lot of old-age material:

J. L. Mundy

*Object Recognition in the Geometric Era: a Retrospective*

Paper: `http://www.di.ens.fr/~ponce/mundy.pdf`

# The Challenge of Recognition Is Modeling Variability




Variability: Camera position
Illumination
Shape parameters

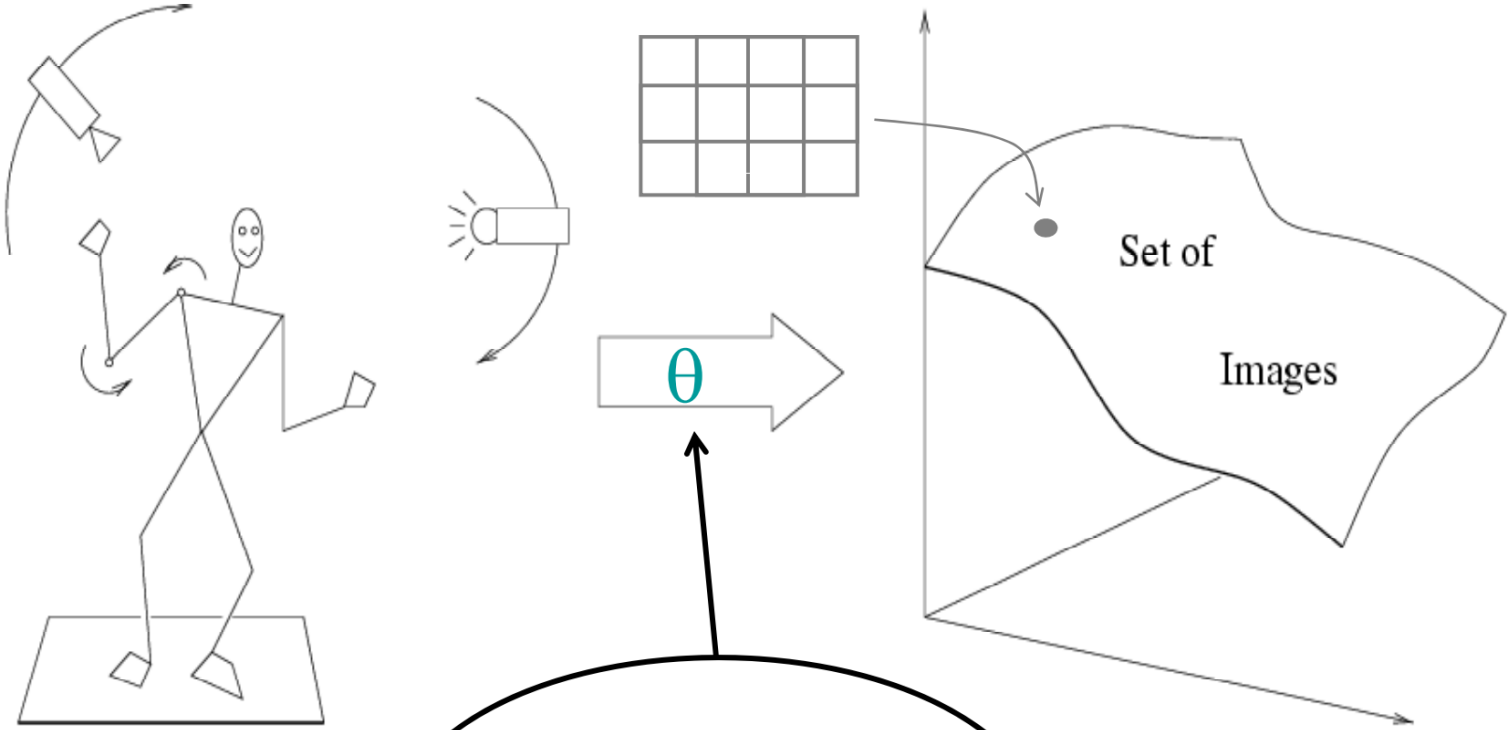
Within-class variations?

[Source S. Lazebnik]

# Recognition Ideas Through History

- 1960s – early 1990s: the geometric era

# 3D Shape Assumed Known




Variability:

Camera position
Illumination

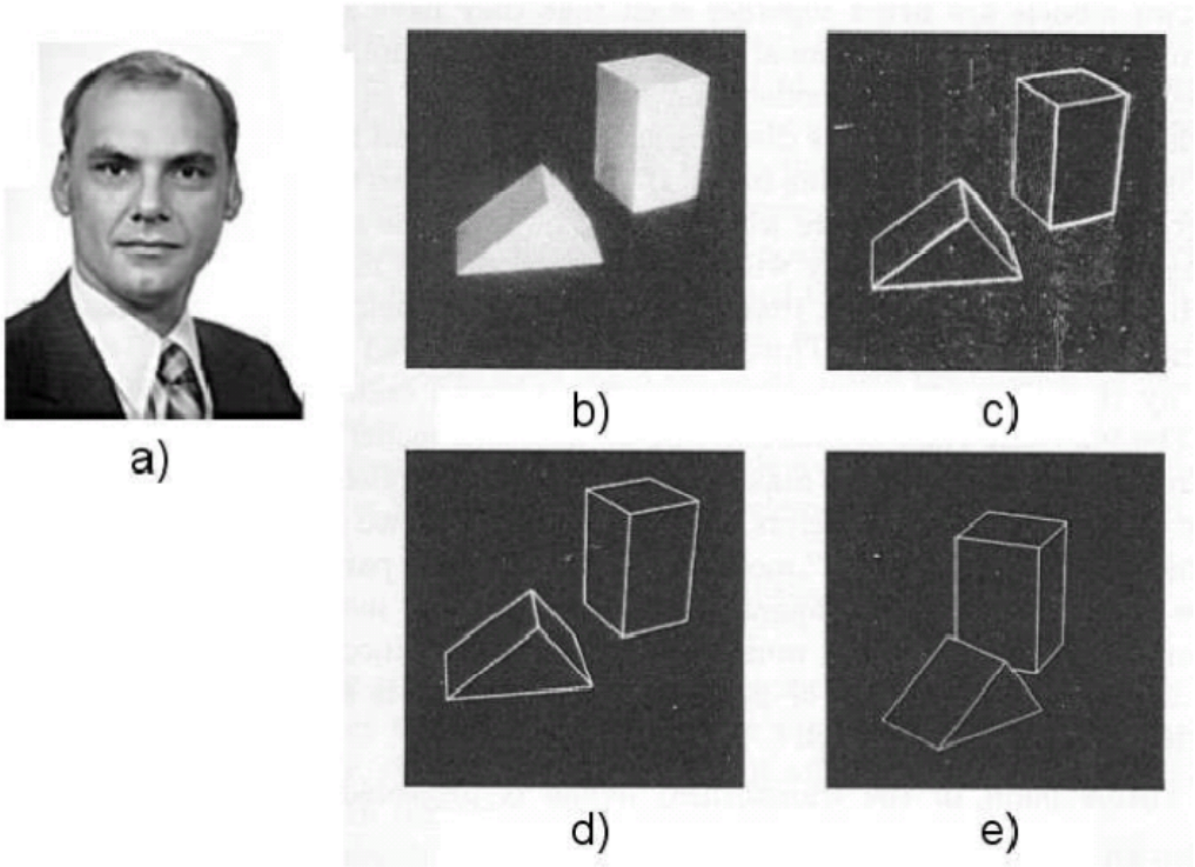
**Alignment**

Shape: assumed known

Roberts (1965); Lowe (1987); Faugeras & Hebert (1986); Grimson & Lozano-Perez (1986); Huttenlocher & Ullman (1987)

[Source S. Lazebnik]

# Blocks World

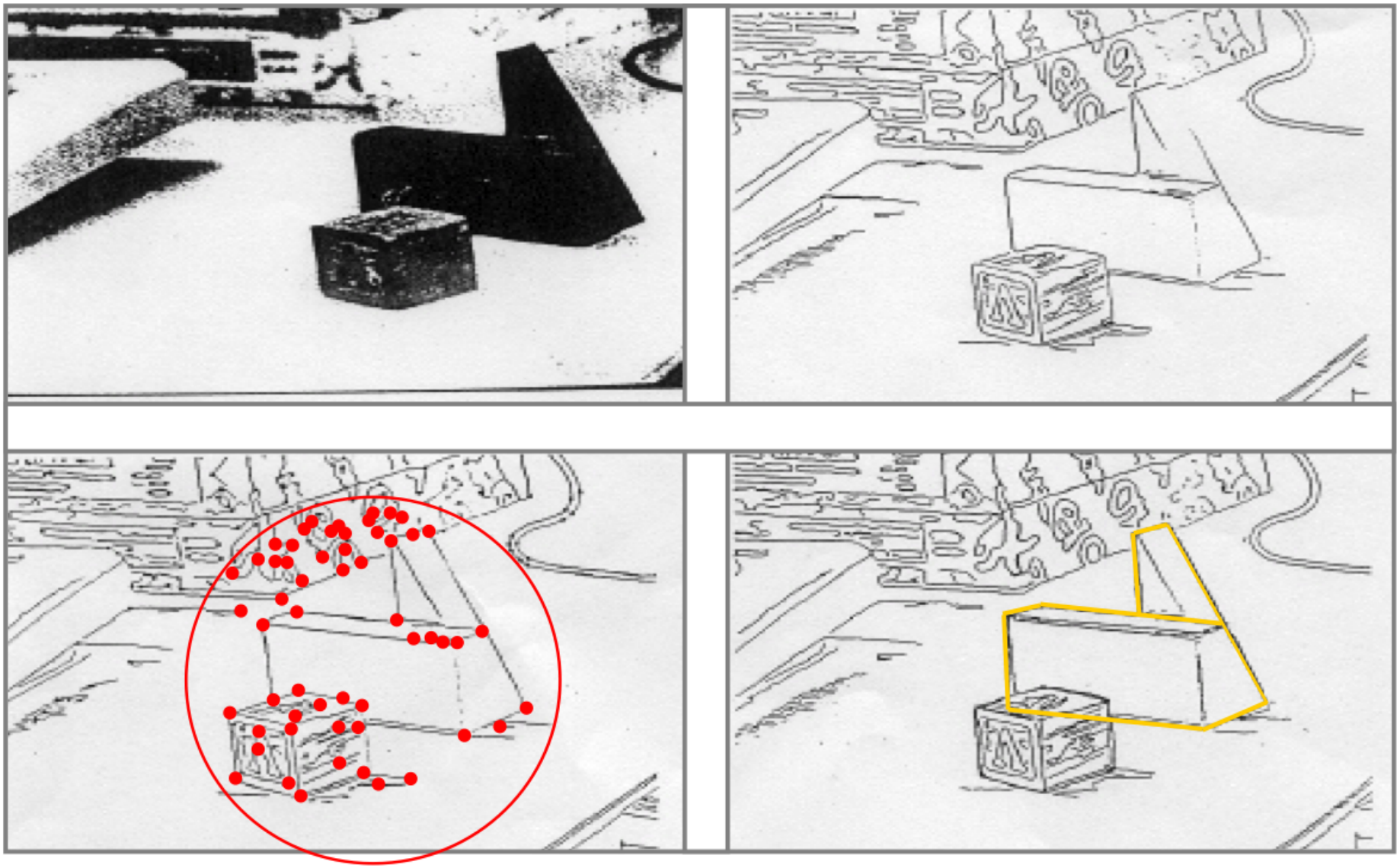



L. G. Roberts, *Machine Perception of Three Dimensional Solids*, Ph.D. thesis, MIT Department of Electrical Engineering, 1963.

**Fig. 1.** A system for recognizing 3-d polyhedral scenes. a) L.G. Roberts. b)A blocks world scene. c)Detected edges using a 2x2 gradient operator. d) A 3-d polyhedral description of the scene, formed automatically from the single image. e) The 3-d scene displayed with a viewpoint different from the original image to demonstrate its accuracy and completeness. (b) - e) are taken from [64] with permission MIT Press.)

[Source S. Lazebnik]

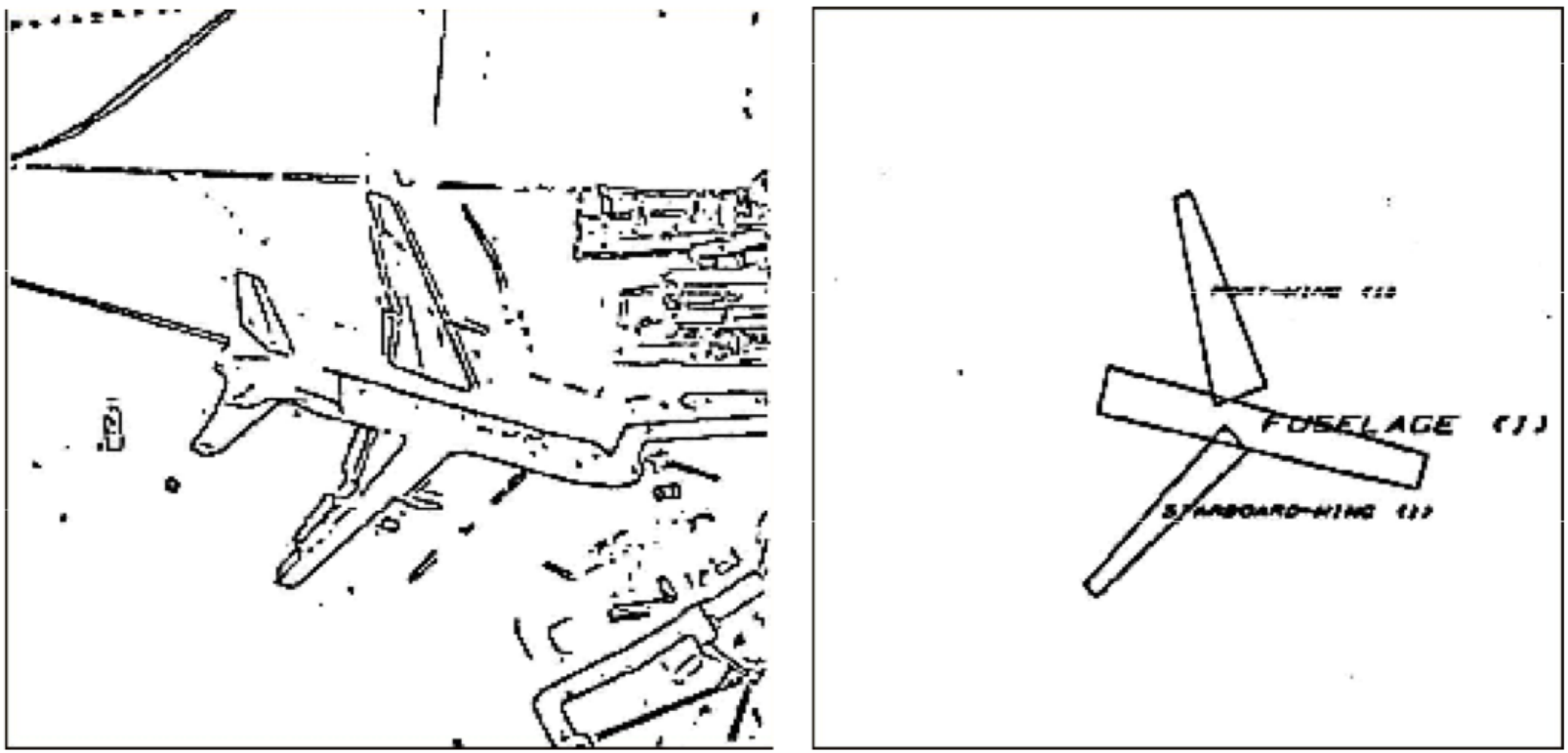
# Alignment



Huttenlocher and Ullman, 1987

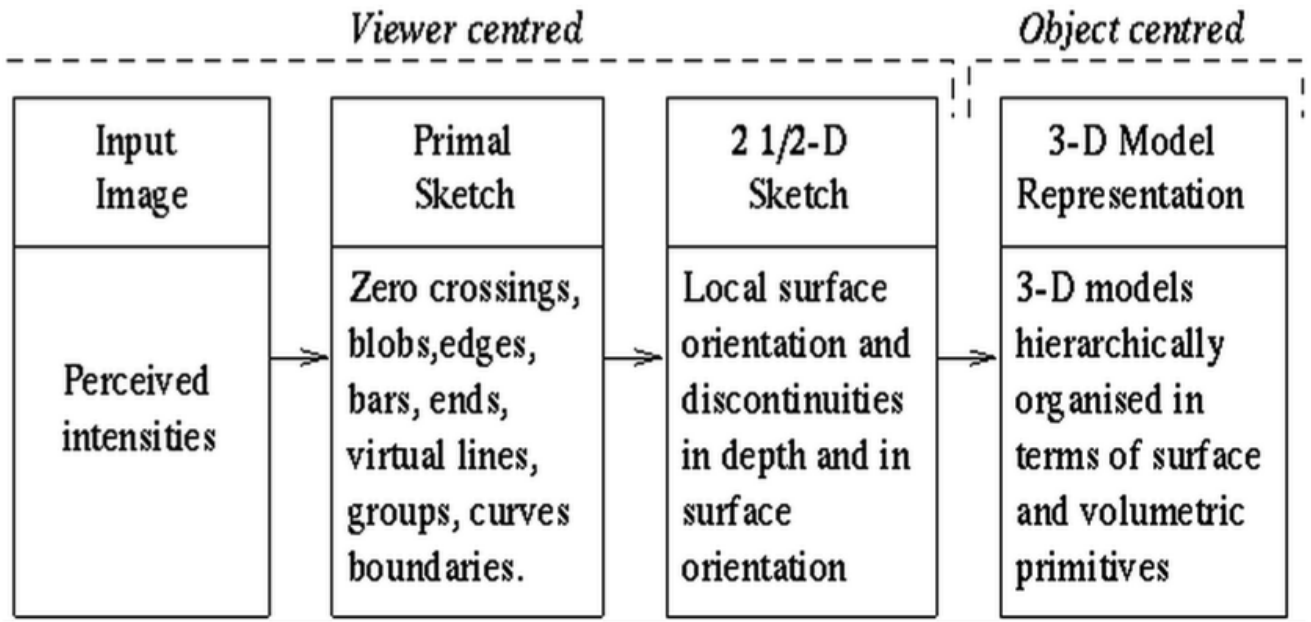
# What About Modeling an Object Class?

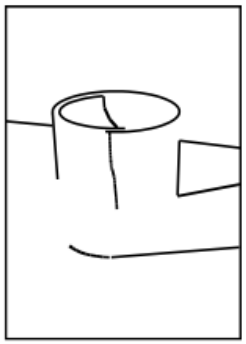
- Modeling the shape across the full object class is difficult
- The idea is to come up with some sort of abstraction: object decomposed into generic parts

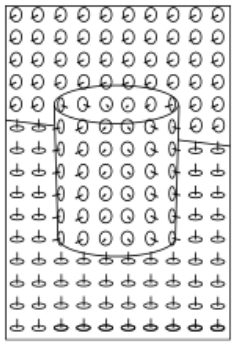


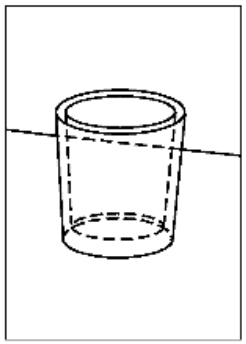

**ACRONYM (Brooks & Binford, 1981)**

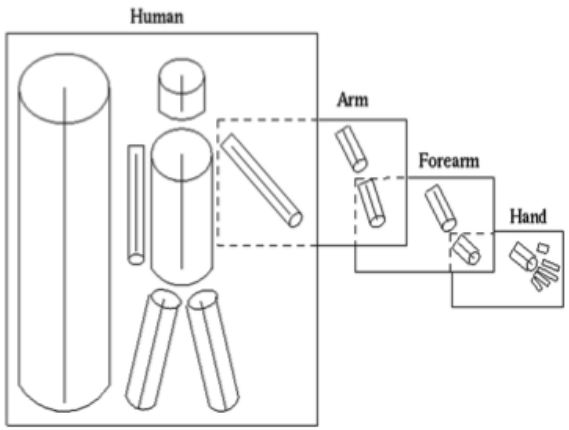
# Marr's Primal Sketch Theory









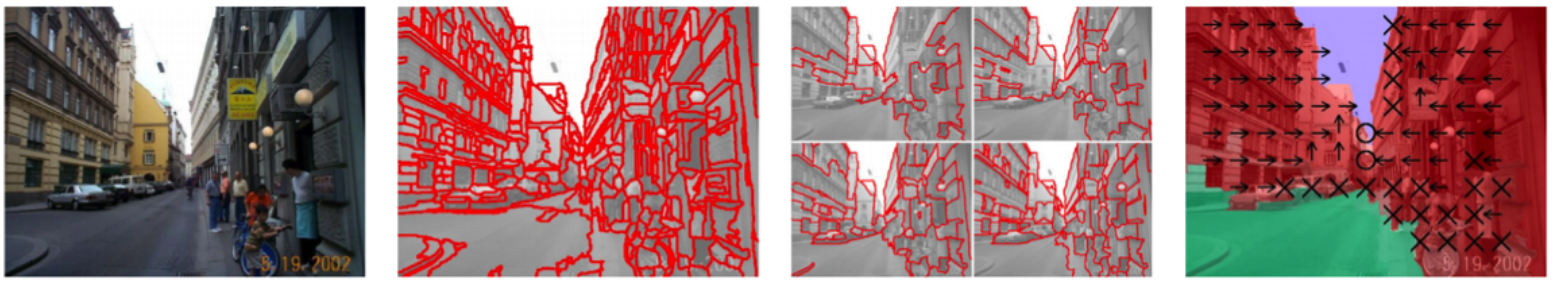

D. Marr, Primal Sketch, 1982

# Surface Normals Estimation – Today

- The idea of surface estimation from single image can be made to work...

| SURFACE CUES |
|---|
| **Location and Shape** |
| L1. Location: normalized x and y, mean |
| L2. Location: normalized x and y, $10^{th}$ and $90^{th}$ pctl |
| L3. Location: normalized y wrt estimated horizon, $10^{th}$, $90^{th}$ pctl |
| L4. Location: whether segment is above, below, or straddles estimated horizon |
| L5. Shape: number of superpixels in segment |
| L6. Shape: normalized area in image |
| **Color** |
| C1. RGB values: mean |
| C2. HSV values: C1 in HSV space |
| C3. Hue: histogram (5 bins) |
| C4. Saturation: histogram (3 bins) |
| **Texture** |
| T1. LM filters: mean absolute response (15 filters) |
| T2. LM filters: histogram of maximum responses (15 bins) |
| **Perspective** |
| P1. Long Lines: (number of line pixels)/sqrt(area) |
| P2. Long Lines: percent of nearly parallel pairs of lines |
| P3. Line Intersections: histogram over 8 orientations, entropy |
| P4. Line Intersections: percent right of image center |
| P5. Line Intersections: percent above image center |
| P6. Line Intersections: percent far from image center at 8 orientations |
| P7. Line Intersections: percent very far from image center at 8 orientations |
| P8. Vanishing Points: (num line pixels with vertical VP membership)/sqrt(area) |
| P9. Vanishing Points: (num line pixels with horizontal VP membership)/sqrt(area) |
| P10. Vanishing Points: percent of total line pixels with vertical VP membership |
| P11. Vanishing Points: x-pos of horizontal VP - segment center (0 if none) |
| P12. Vanishing Points: y-pos of highest/lowest vertical VP wrt segment center |
| P13. Vanishing Points: segment bounds wrt horizontal VP |
| P14. Gradient: x, y center of mass of gradient magnitude wrt segment center |

- Compute superpixels
- For each superpixel compute several interesting features that make use of vanishing points, color, texture, lines...
- Train classifiers to predict several geometric classes: support, vertical sky



Input | Superpixels | Multiple Segmentations | Surface Layout

Figure: D. Hoiem, A.A. Efros, and M. Hebert, Recovering Surface Layout from an Image, 2007

# Surface Normals Estimation – Today

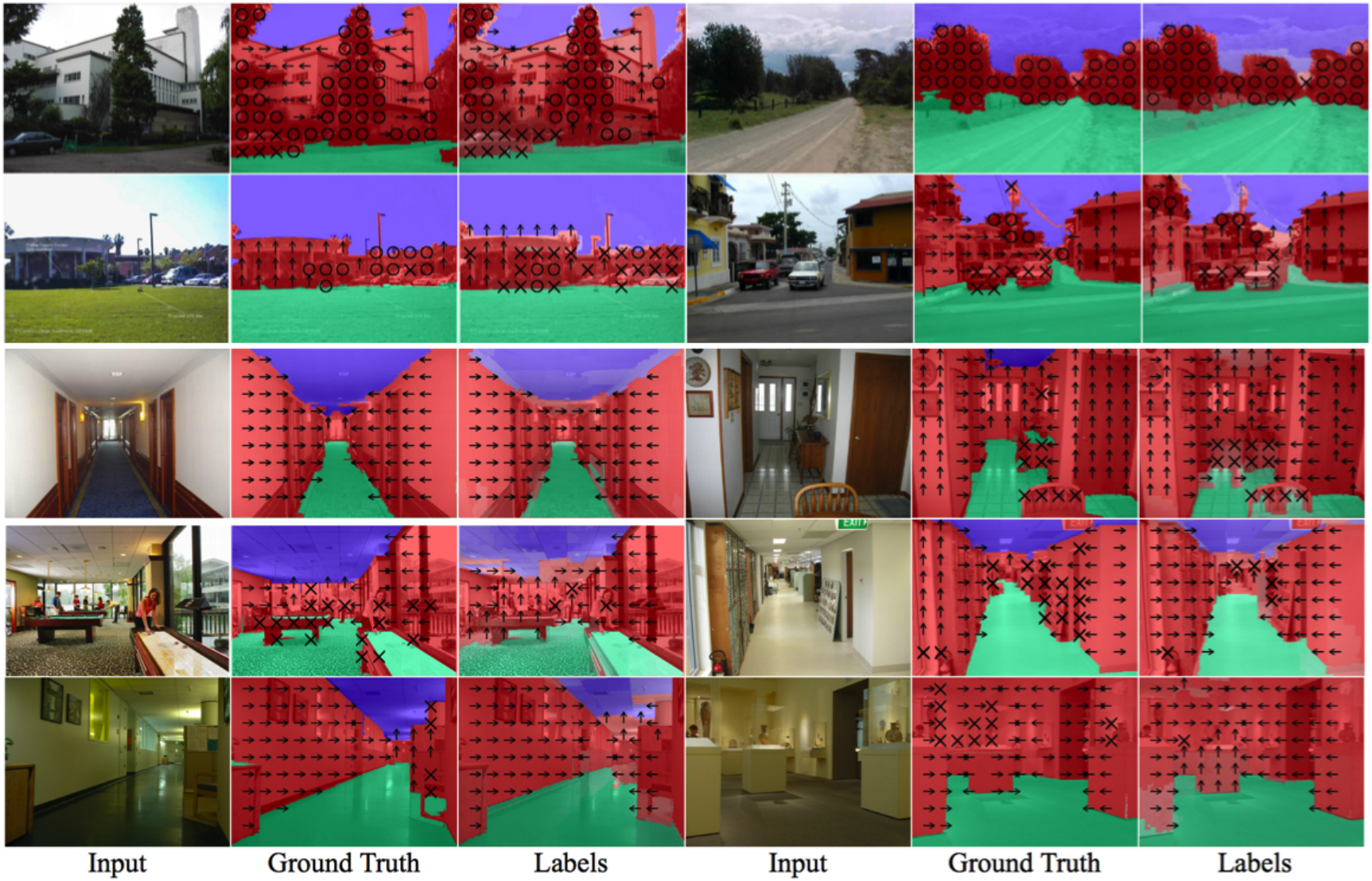



Figure: D. Hoiem, A.A. Efros, and M. Hebert, Recovering Surface Layout from an Image, 2007

# Useful Information for Recogniton

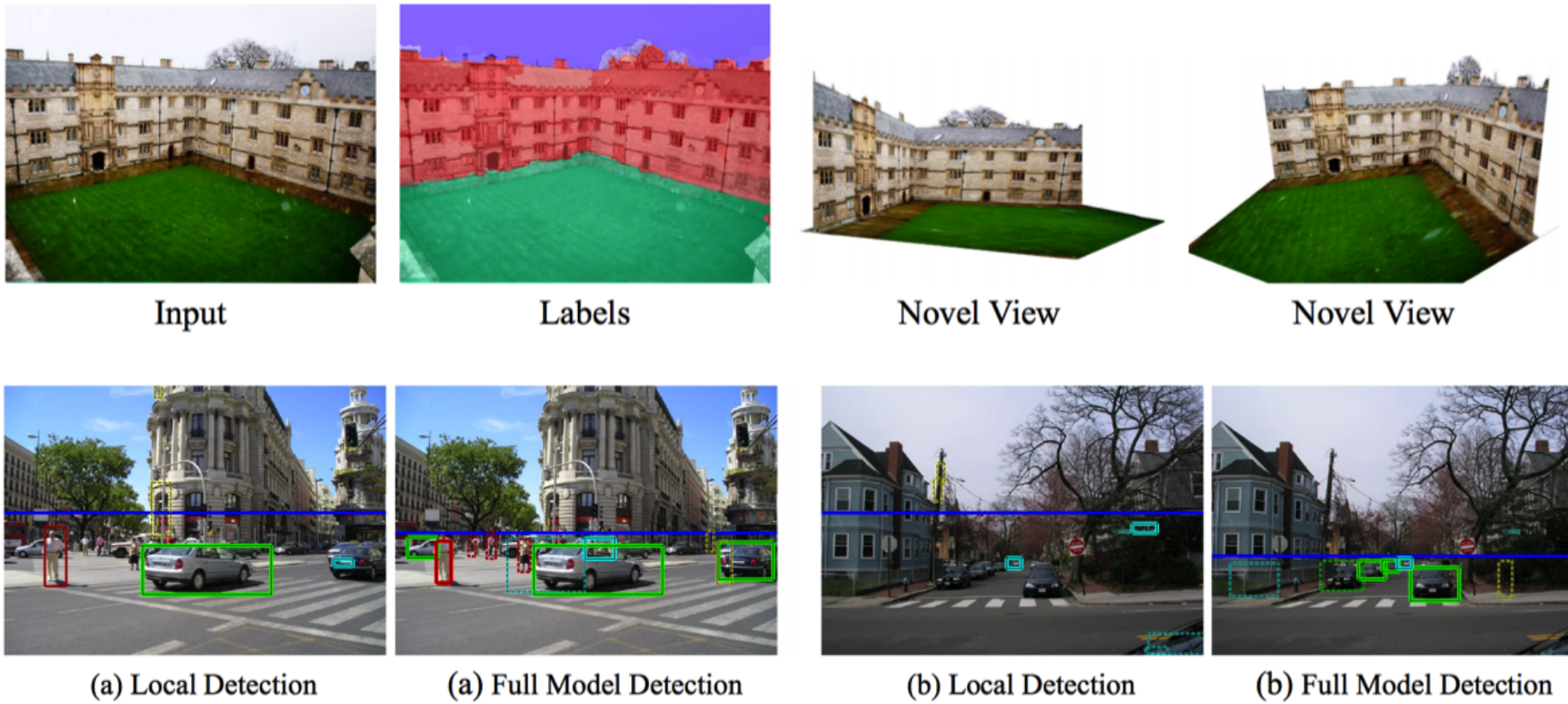



Figure: D. Hoiem, A.A. Efros, and M. Hebert, Recovering Surface Layout from an Image, 2007

# Binford's Generalized Cylinders

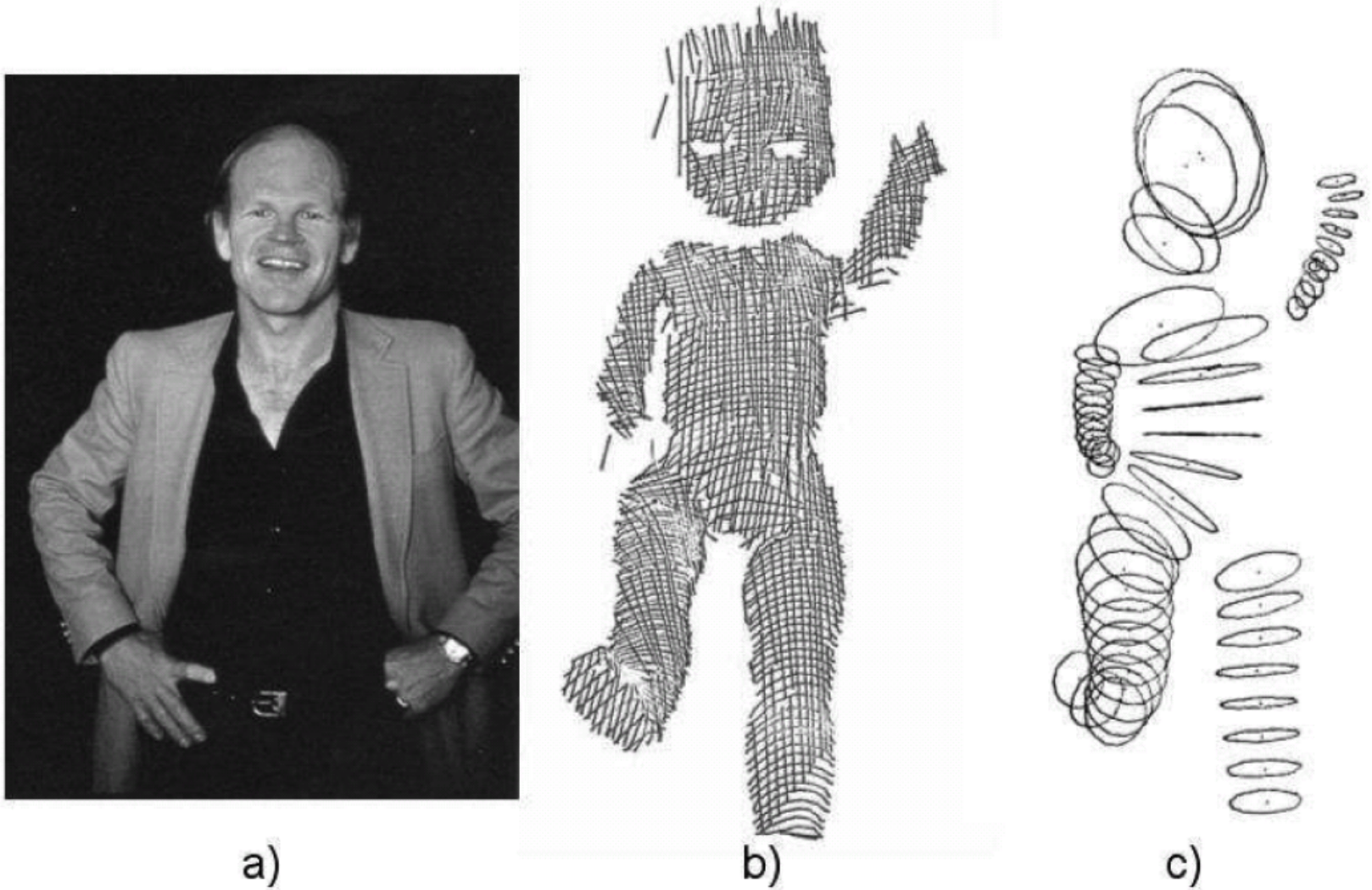
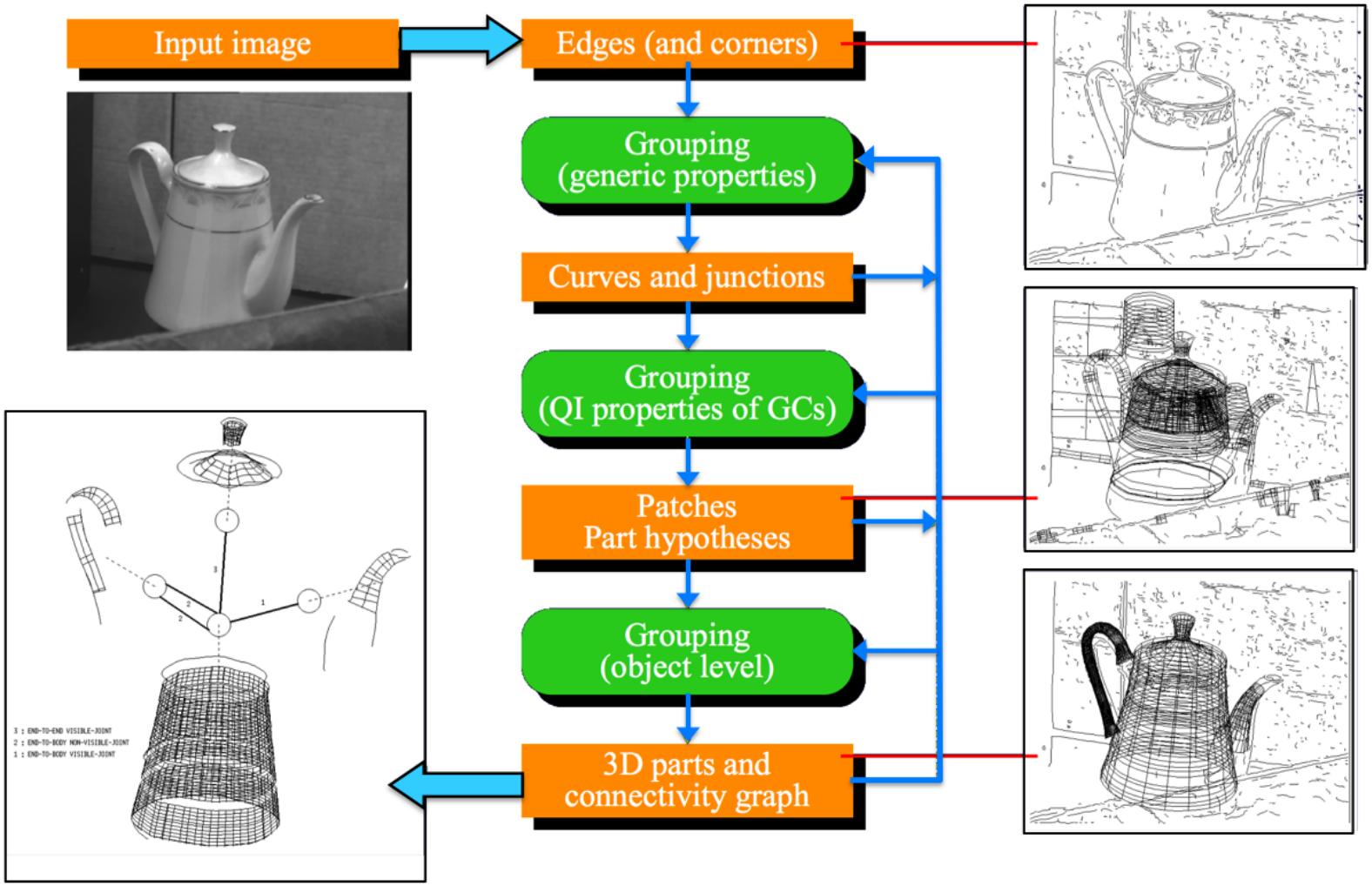



**Fig. 3.** The representation of objects by assemblies of generalized cylinders. a) Thomas Binford. b) A range image of a doll. c) The resulting set of generalized cylinders. ( b) and c) are taken from Agin [1] with permission.)

# Nevatia's Generalized Cylinders

- Binford's student Ram Nevatia continued to push the GC theory. With limited success.




G. Medioni, *Generic shape learning and recognition,* Workshop on Generic Object Recognition and Categorization, CVPR 2004

# From Cylinders to Geons

Biederman, Recognition by Components, 1987

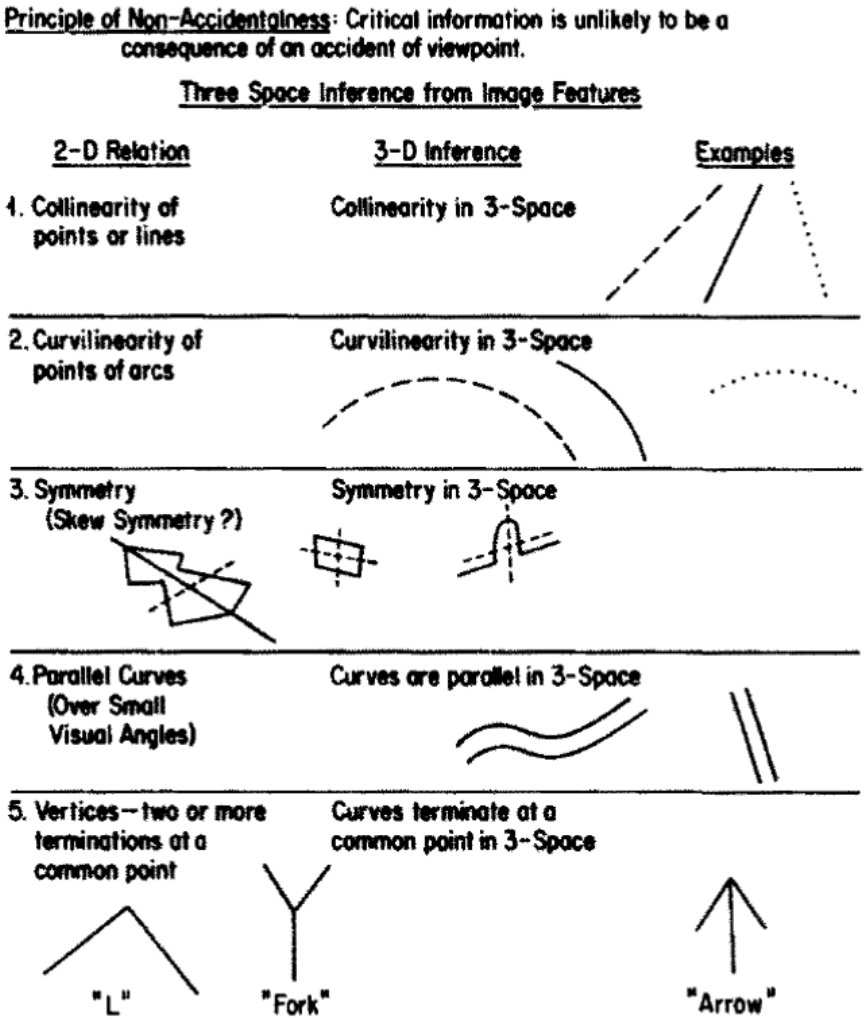
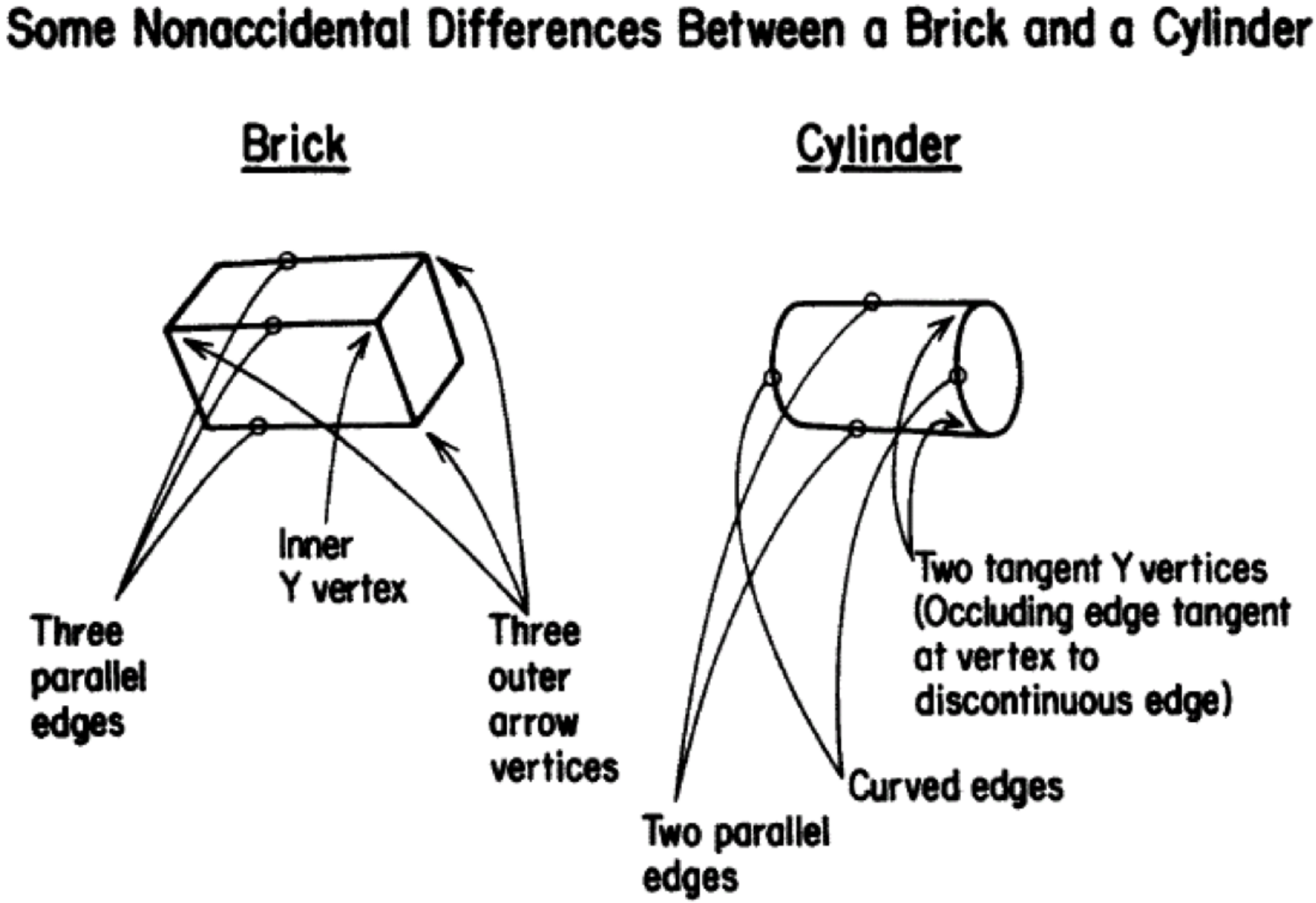



*Figure 4.* Five nonaccidental relations. (From Figure 5.2, *Perceptual organization and visual recognition* [p. 77] by David Lowe. Unpublished doctorial dissertation, Stanford University. Adapted by permission.)

[Source: A. Torralba]

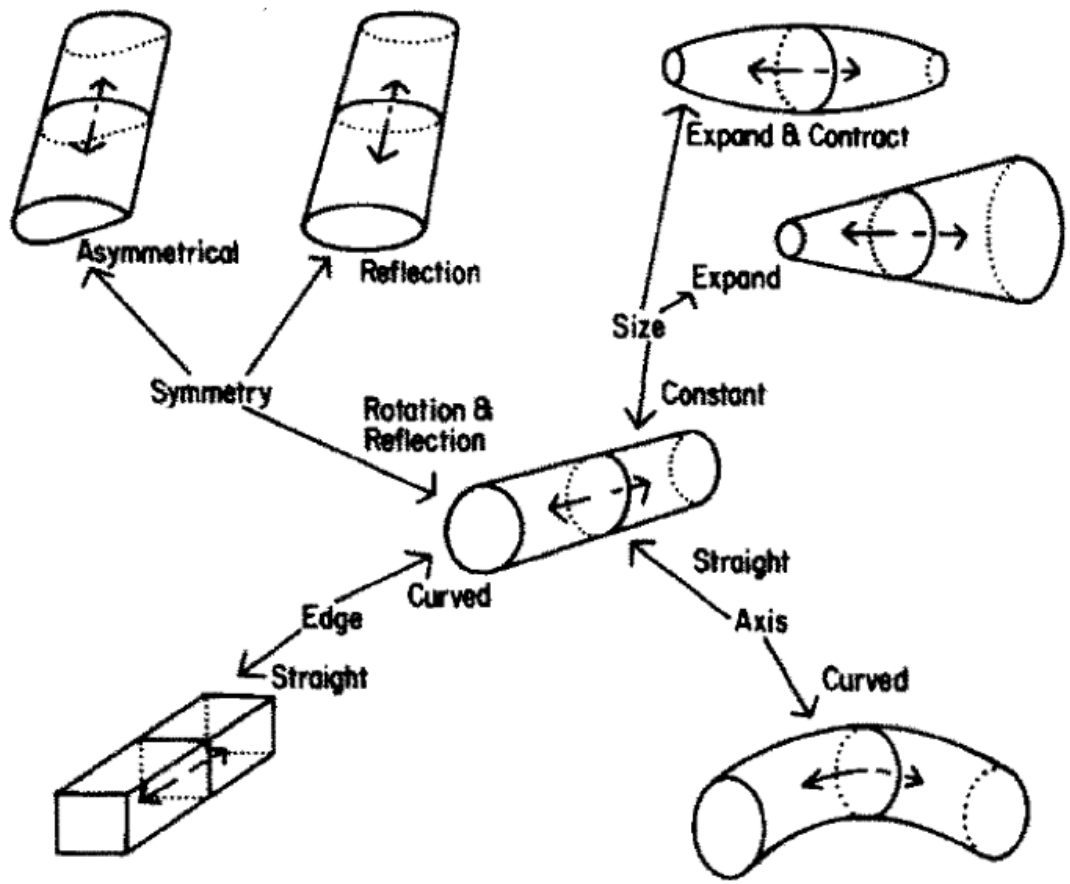
# From Cylinders to Geons




[Source: A. Torralba]

# From Generalized Cylinders to Geons

- From variation over only two or three levels in the non-accidental relations of four attributes of generalized cylinders, a set of 36 GEONS can be generated.




[Source: A. Torralba]

# The Geons

CROSS SECTION

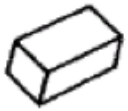
| Geon | Edge<br>Straight S<br>Curved C | Symmetry<br>Rot & Ref ++<br>Ref +<br>Asymm – | Size<br>Constant ++<br>Expanded –<br>Exp & Cont – – | Axis<br>Straight +<br>Curved – |
|---|---|---|---|---|
|  | S | + + | + + | + |
|  | C | + + | + + | + |
|  | S | + | – | + |
|  | S | + + | + | – |
|  | C | + + | – | + |
|  | S | + | + | + |

*Figure 7.* Proposed partial set of volumetric primitives (geons) derived from differences in nonaccidental properties.

CROSS SECTION

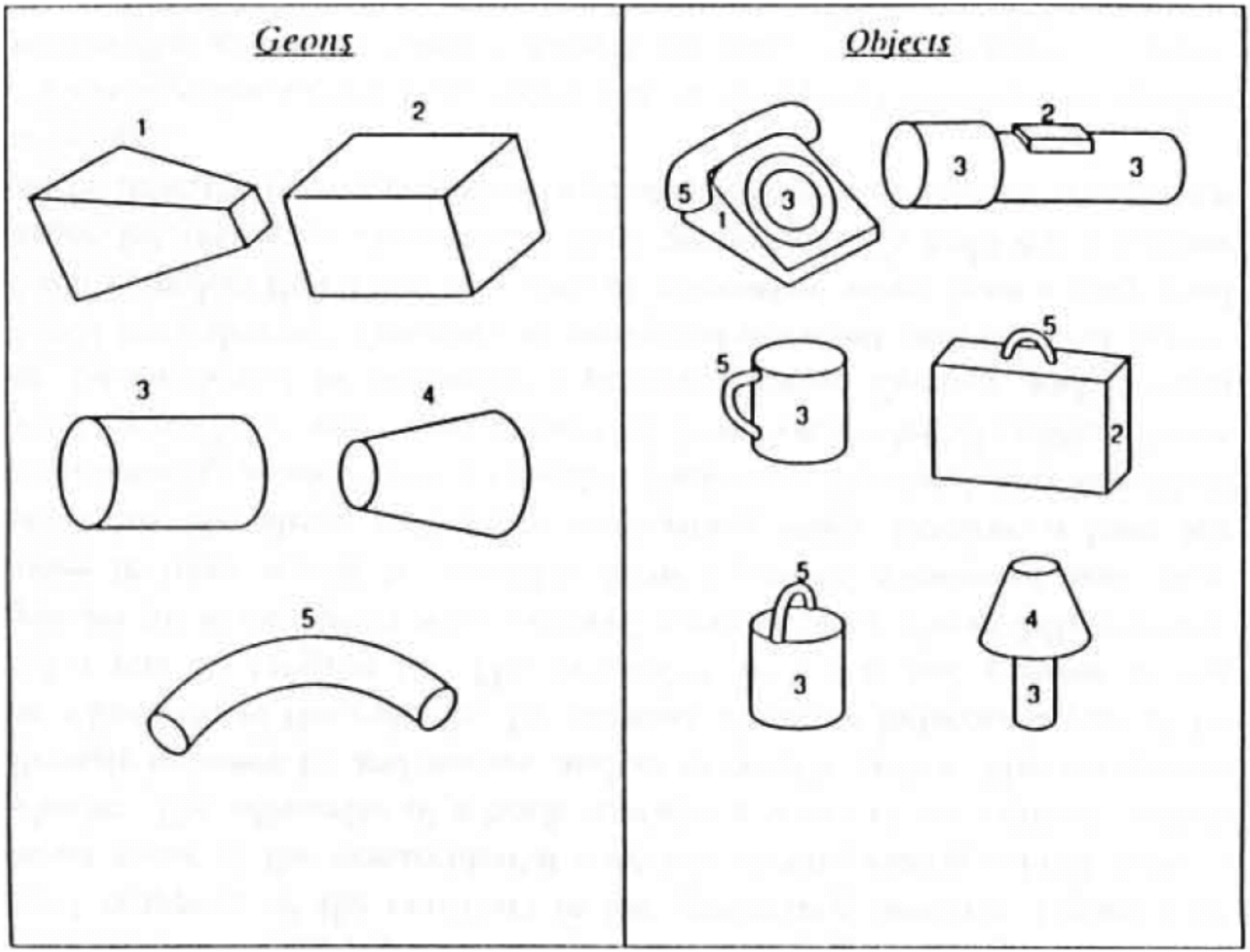
| Geon | Edge<br>Straight S<br>Curved C | Symmetry<br>Rot & Ref ++<br>Ref +<br>Asymm – | Size<br>Constant ++<br>Expanded –<br>Exp & Cont – – | Axis<br>Straight +<br>Curved – |
|---|---|---|---|---|
|  | S | + | + + | – |
|  | C | + | + + | – |
|  | S | + + | – | – |
|  | C | + + | – | – |
|  | S | + | – | – |
|  | C | + | – | – |

*Figure 9.* Geons with curved axis and straight or curved cross sections. (Determining the shape of the cross section, particularly if straight, might require attention.)

[Source: A. Torralba]

# Geons: Lego for Objects

- Any object can be represented with the set of 36 geons




[Source: A. Torralba]

# Objects As Geons

- Spatial arrangements of parts matters!

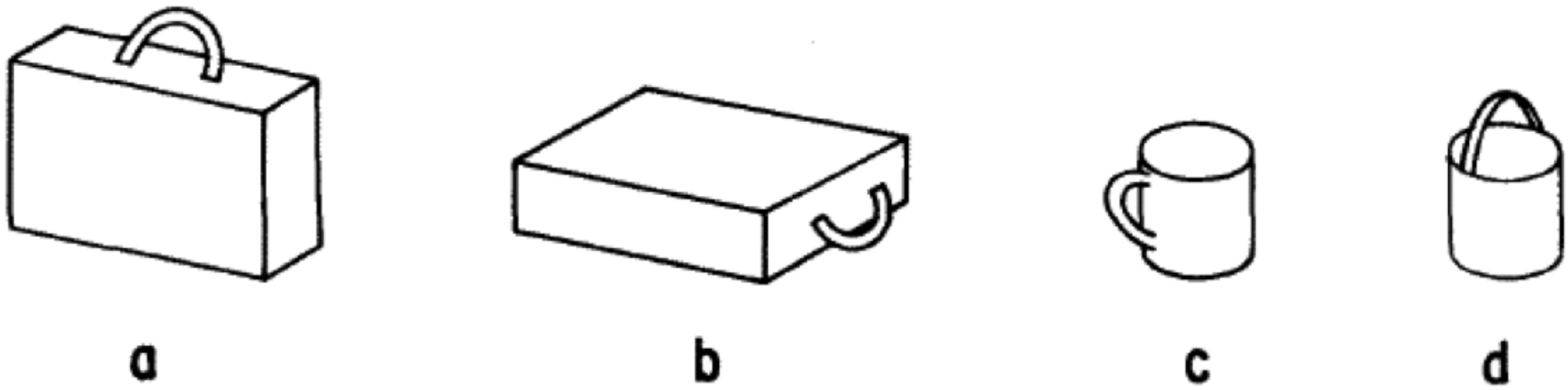
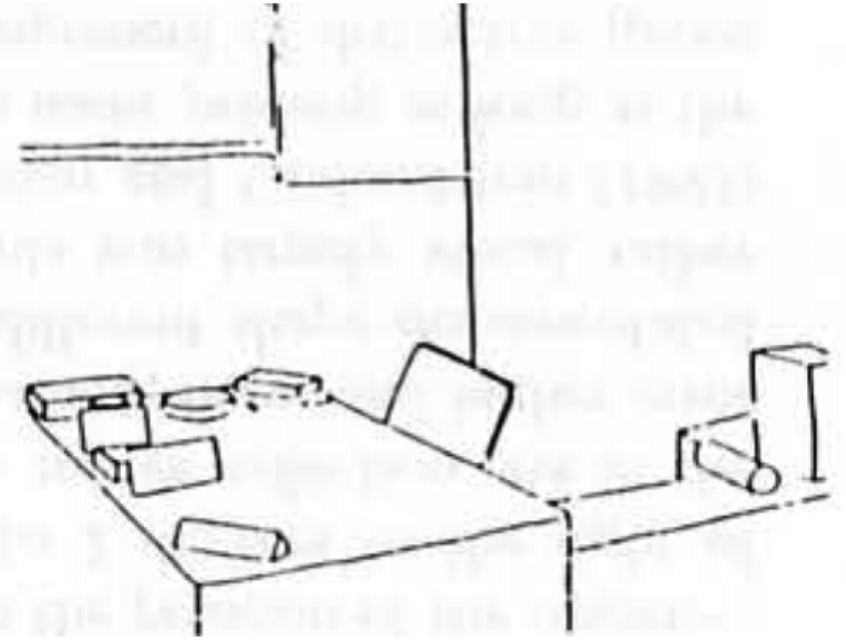


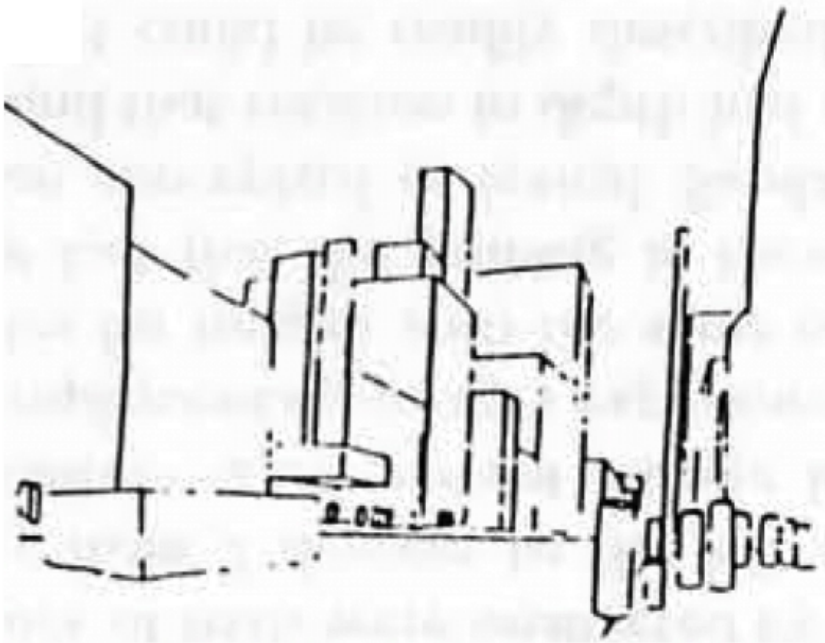

*Figure 3.* Different arrangements of the same components can produce different objects.

[Source: A. Torralba]

# The World is Made of Geons

- Why stop at the object. A scene is a composition of objects and objects are compositions of geons.

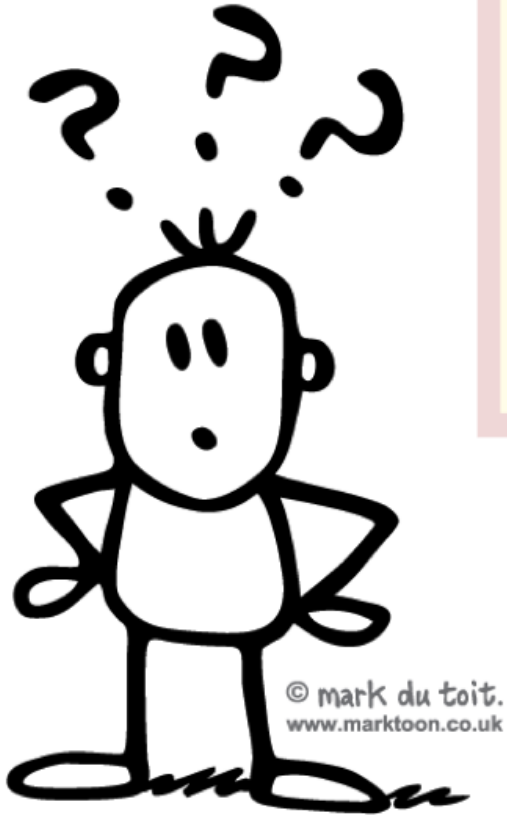


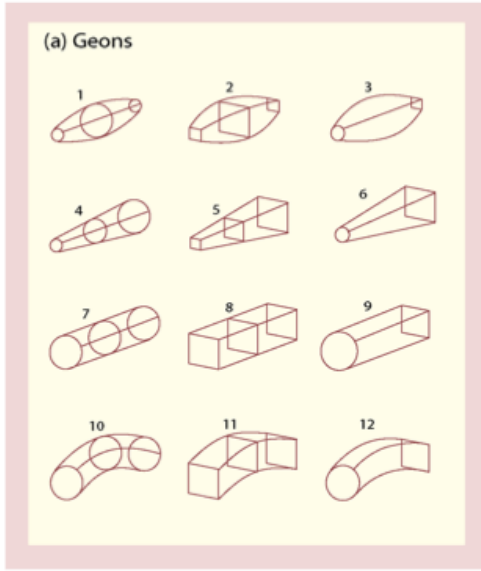


[Source: A. Torralba]

# Geons

- Nice theory. But how would I extract geons from an image?



# Superquadrics

- Following the idea of geons, let's find a set of **parametrizable** simple volumes. Why is this important?

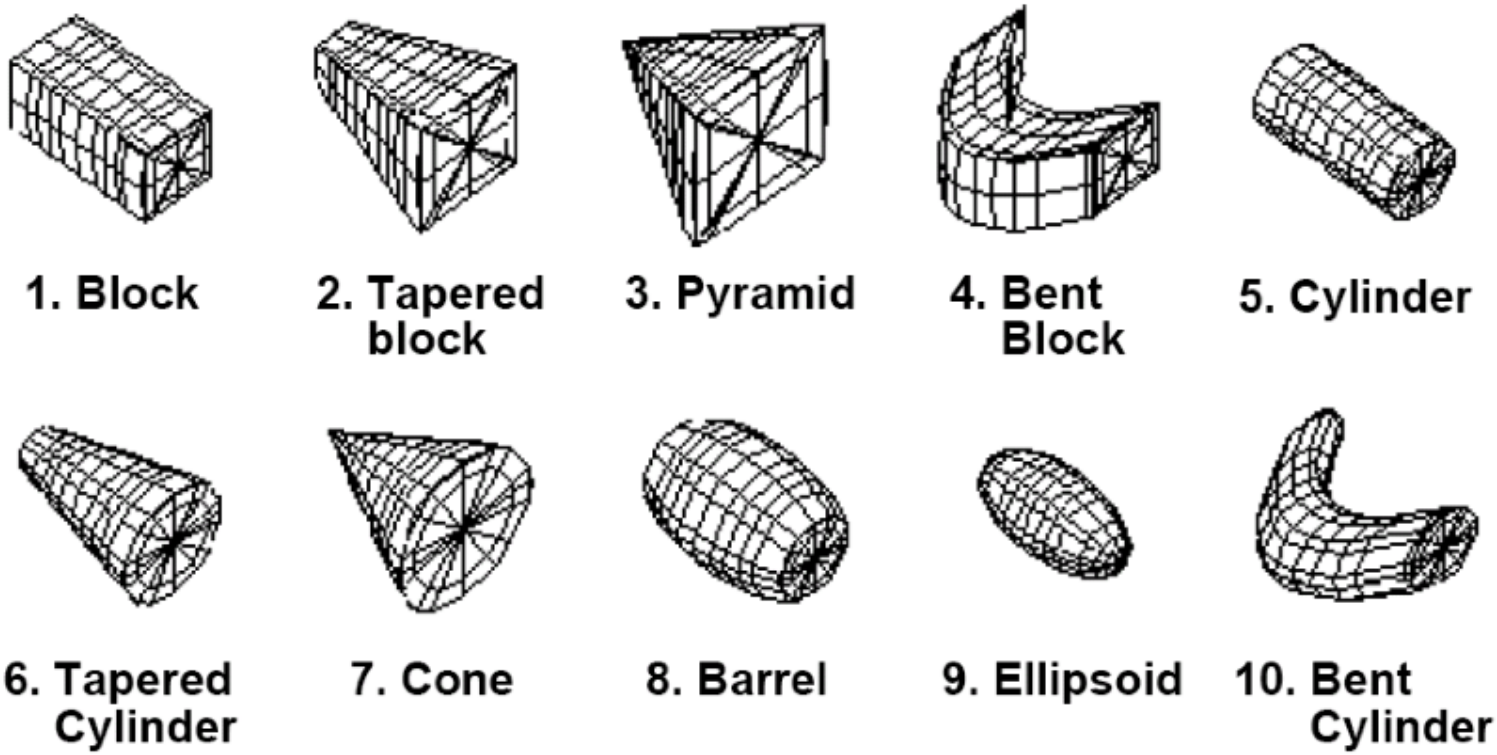



Figure: Introduced in computer vision by A. Pentland, 1986

[Adopted from: A. Torralba]

# Superquadrics

- It was possible to fit superquadrics to the data. Where data means range images (image + depth).

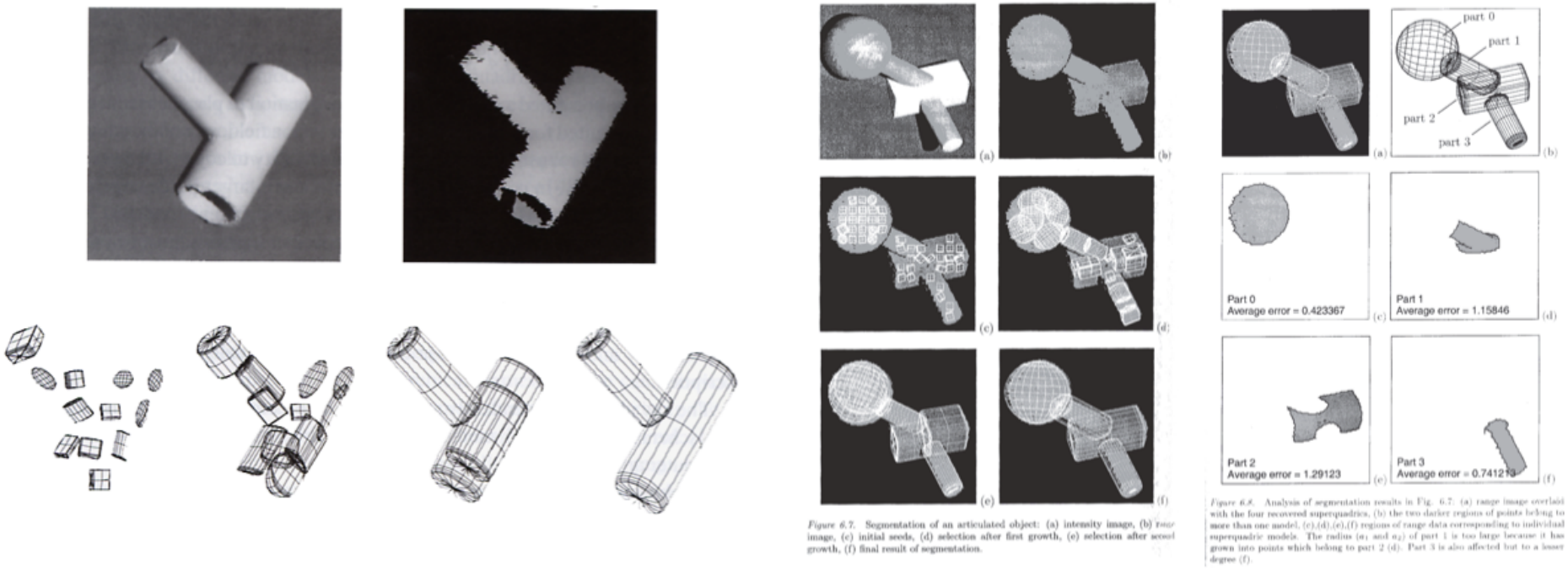



Figure: A. Leonardis, A. Jaklic, and F. Solina, 1997.

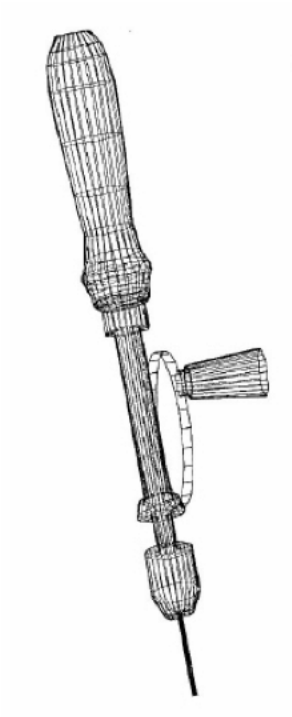
# Nothing Worked (Well)

- Nothing really worked
- Why? What was the problem?
- What were some of the good ideas of this era?
- Do you think we could make some of these ideas work now, with e.g., training data and Machine Learning?

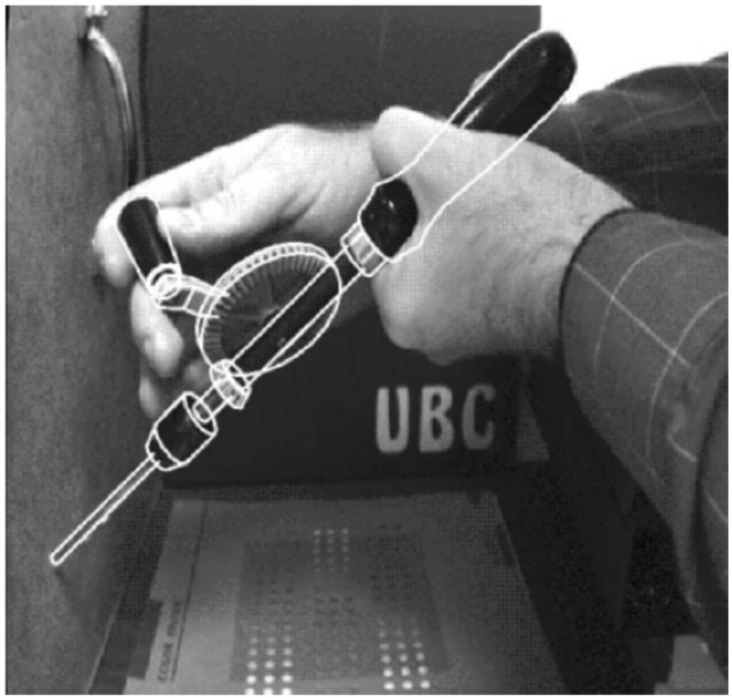
# Old Ideas With New Data and Technology

**Goal: Match known shape to image:**

- Before: Do some grouping on the image side to get corners, lines, etc
- Before: match **one** known 3D model to the image evidence



**3D Model**




**Alignment**

# Old Ideas With New Data and Technology

- Now: 3D Warehouse (`https://3dwarehouse.sketchup.com/`) has millions of accurate CAD models of objects. 8,375 search results for query "IKEA". We can have models for all our furniture!


Figure: `http://ikea.csail.mit.edu/`

# Old Ideas With New Data and Technology

- Now: 3D Warehouse (`https://3dwarehouse.sketchup.com/`) has millions of accurate CAD models of objects. 8,375 search results for query "IKEA". We can have models for all our furniture!
- Now: Forget about bottom-up grouping and geons. Train classifiers and learn what local patches can be reliably detected for each 3D model.

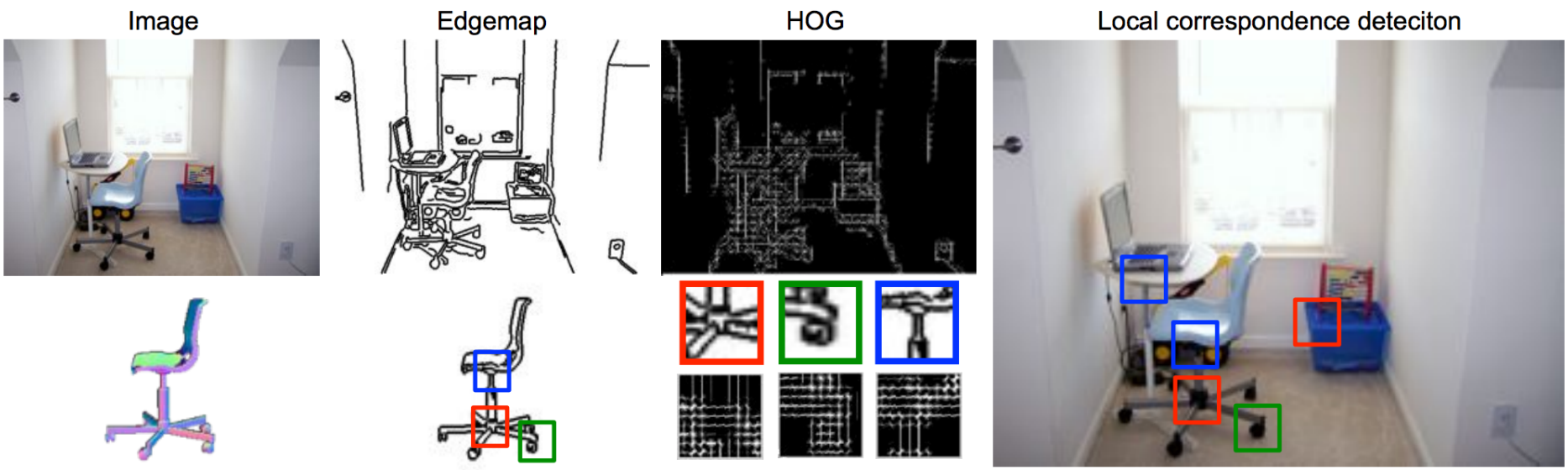



Figure 2. **Local correspondence:** for each 3D interest point $\mathbf{X}_i$ (red, green, and blue), we train an LDA patch detector on an edgemap and use its response as part of our cost function. We compute HOG on edgemaps to ensure a real image and our model share the modality.

Figure: J. J. Lim, H. Pirsiavash, Antonio Torralba. Parsing IKEA Objects: Fine Pose Estimation. ICCV'13

# Old Ideas With New Data and Technology

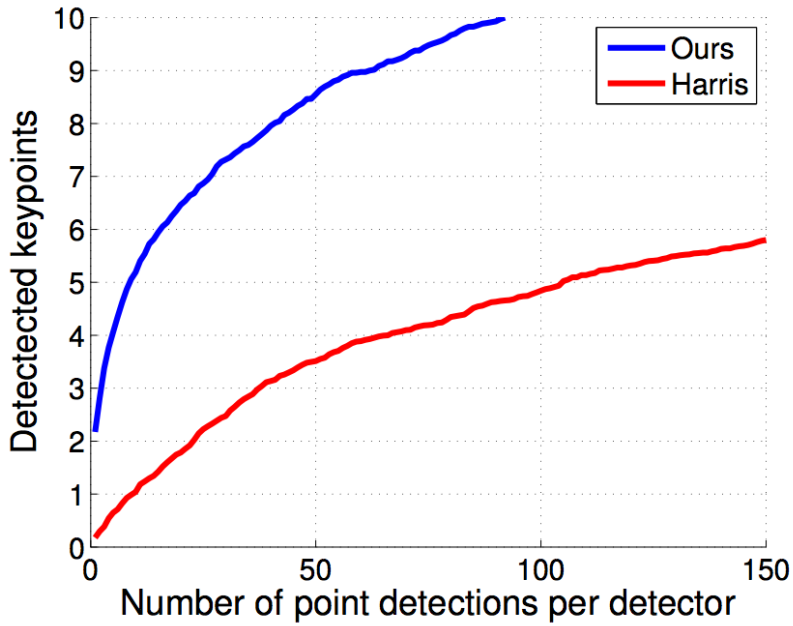



Figure 6. **Correspondence evaluation:** we are comparing correspondences between our interest point detector and Harris detector. The minimum number of interest points we need for reliable pose estimation is 5. Ours can recall 5 correct correspondences by considering only the top 10 detections per 3D interest point, whereas the Harris detector requires 100 per point. This results in effectively $10^5$ times fewer search iterations in RANSAC.

Figure: Learned discriminative patches vs Harris corners

[J. J. Lim, H. Pirsiavash, Antonio Torralba. Parsing IKEA Objects: Fine Pose Estimation. ICCV'13]

# Old Ideas With New Data and Technology

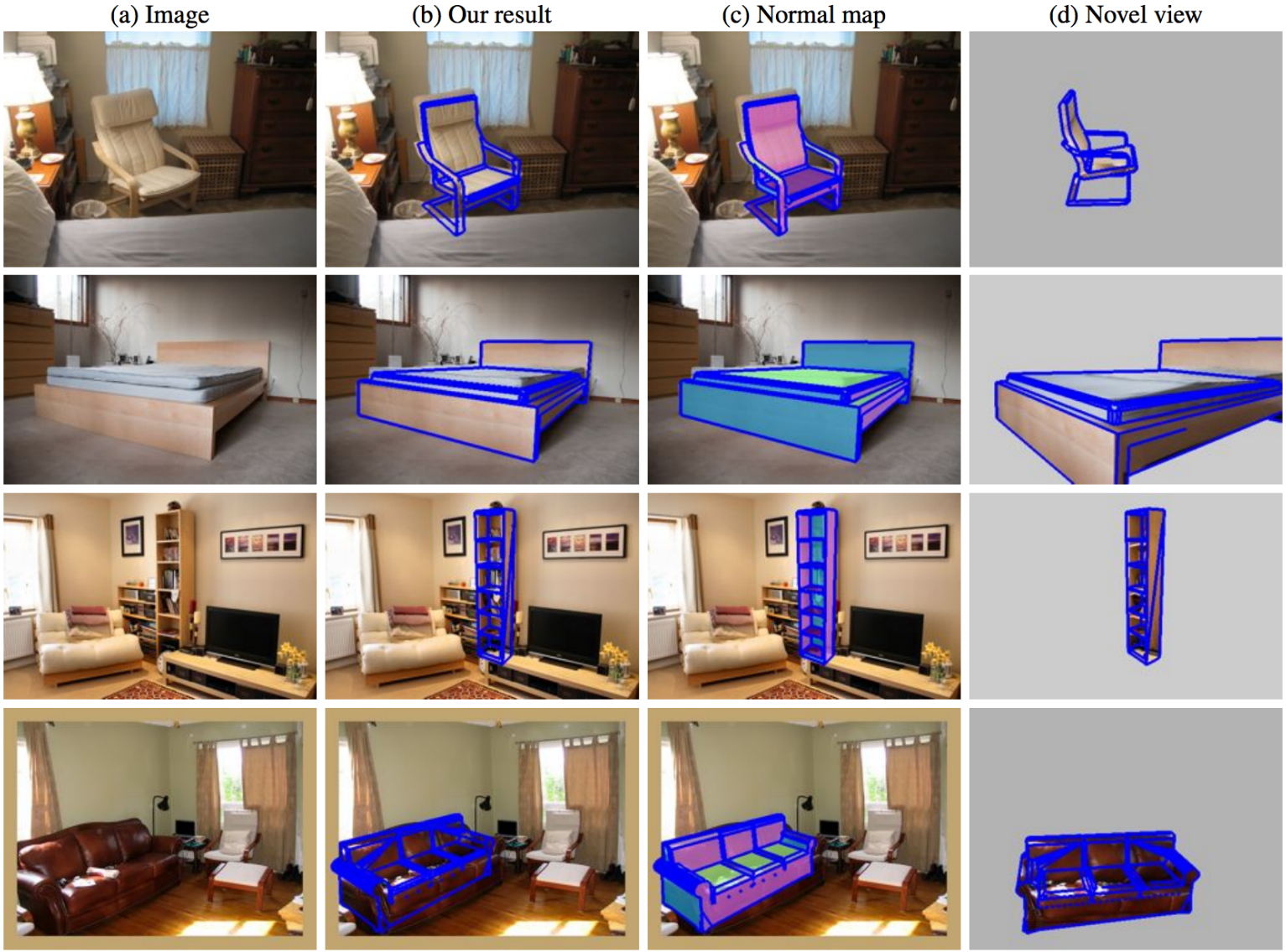



Figure: Results

[J. J. Lim, H. Pirsiavash, Antonio Torralba. Parsing IKEA Objects: Fine Pose Estimation. ICCV'13]

# Old Ideas With New Data and Technology

Figure: Results: Still some failure modes

[J. J. Lim, H. Pirsiavash, Antonio Torralba. Parsing IKEA Objects: Fine Pose Estimation. ICCV'13]

# Old Ideas With New Data and Technology

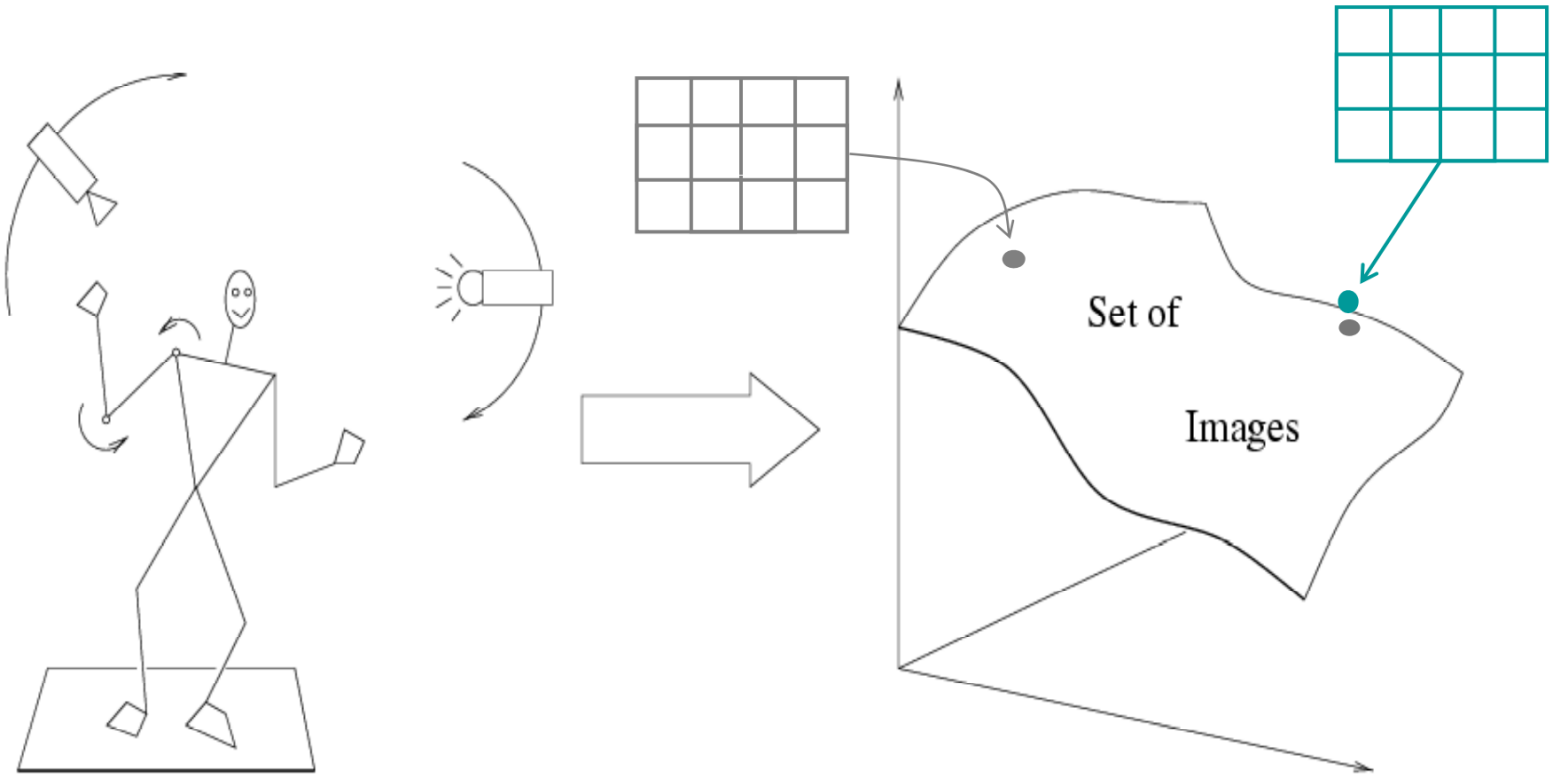
- If you want to be safe from computer vision detectors, don't buy stuff in IKEA ;)

[J. J. Lim, H. Pirsiavash, Antonio Torralba. Parsing IKEA Objects: Fine Pose Estimation. ICCV'13]

# Recognition Ideas Through History

- 1960s – early 1990s: the geometric era
- **1990s: appearance-based models**

# Forget About 3D, Think Only About Image




Empirical models of image variability

**Appearance-based techniques**

Figure: Turk & Pentland, 1991; Murase & Nayar, 1995, etc

[Source: S. Lazebnik]

# "Eigenfaces"

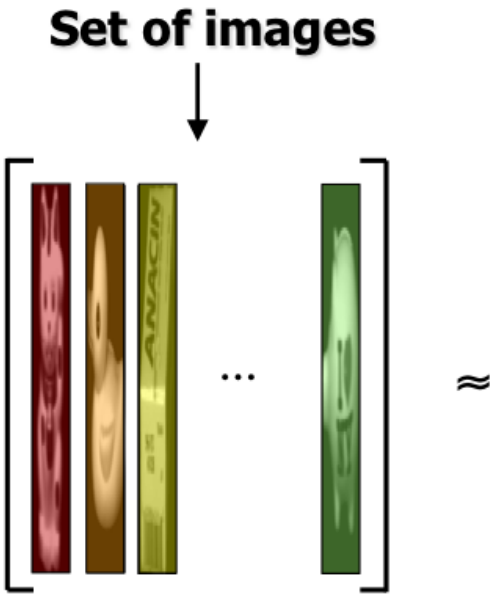
- Work with pixels. Align all the "training" images, and subtract the average image. Vectorize.

- Take pictures of different faces, lots of pictures for each person.
- Align the images (positions of eyes, nose, mouth should roughly match across images)
- Compute the average face image
- Subtract the average face from each image
- Vectorize each image (e.g., image(:) in Matlab)

Figure: Turk & Pentland, 1991; Murase & Nayar, 1995, etc

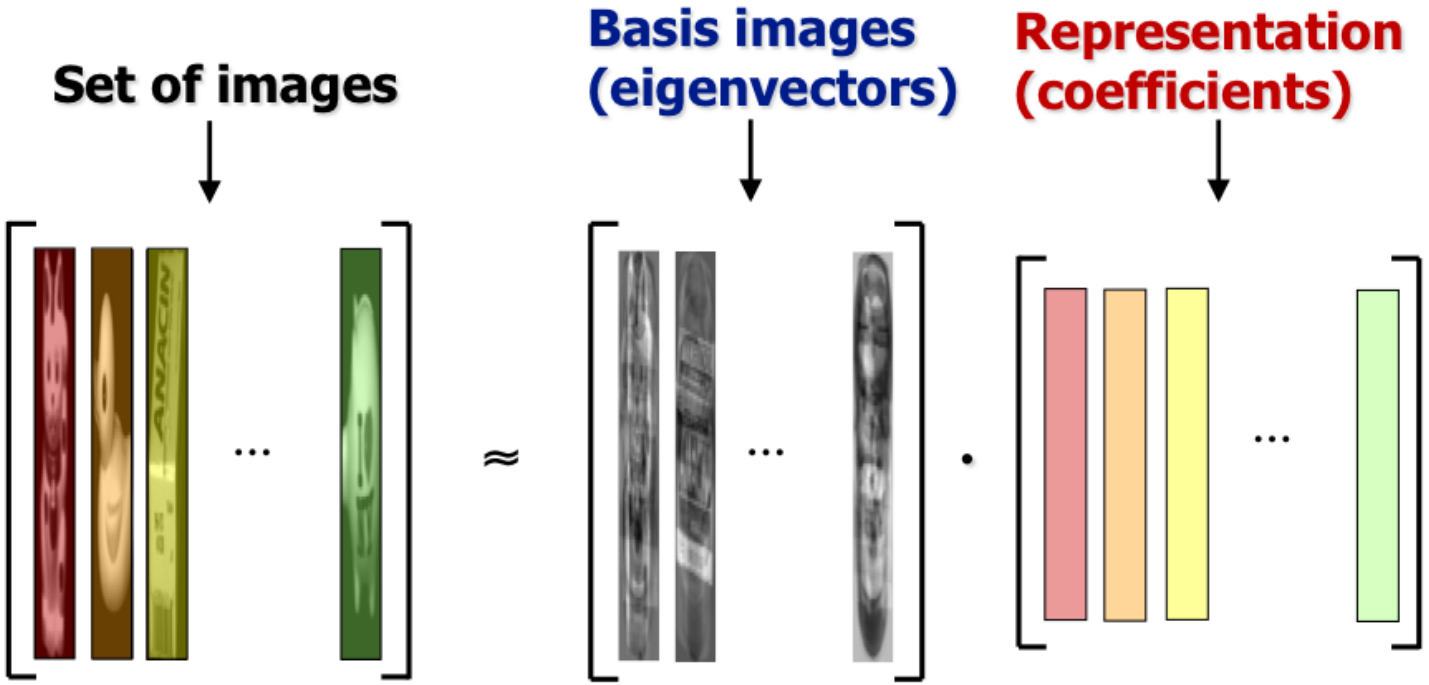
# “Eigenfaces”

- Stack the training image vectors in a matrix $X$



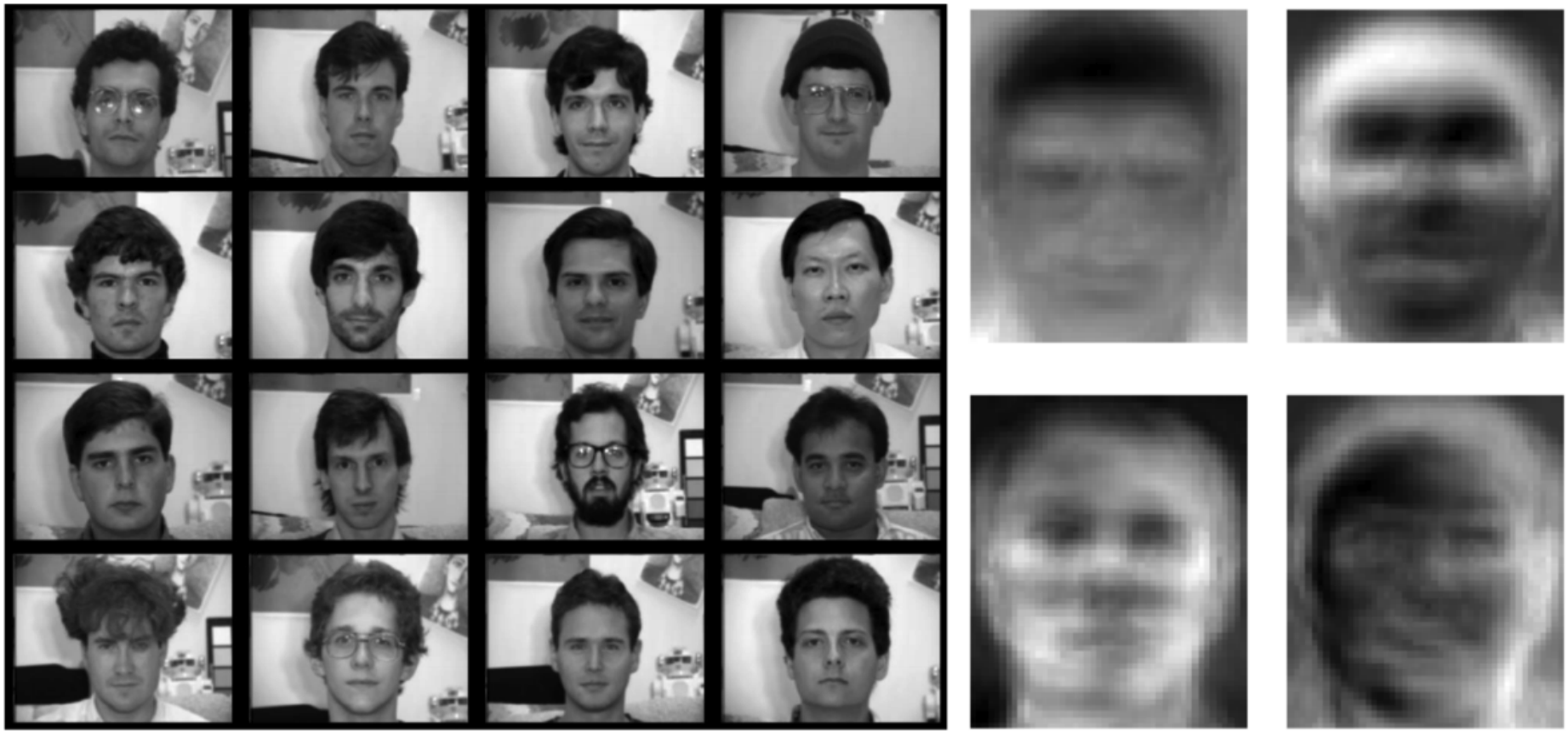
# "Eigenfaces"

- Stack the training image vectors in a matrix $X$
- Perform PCA. This is nothing but finding the eigenvectors and eigenvalues of the covariance matrix: $cov(X) = X \cdot X^T$. In Matlab: $[\mathrm{U,D}] = \mathrm{EIG}(X \cdot X')$;. $U$ contains the eigenvectors
- We can now represent the images with this new "basis". The coefficients are easily computed as: $A = U^T \cdot X$.



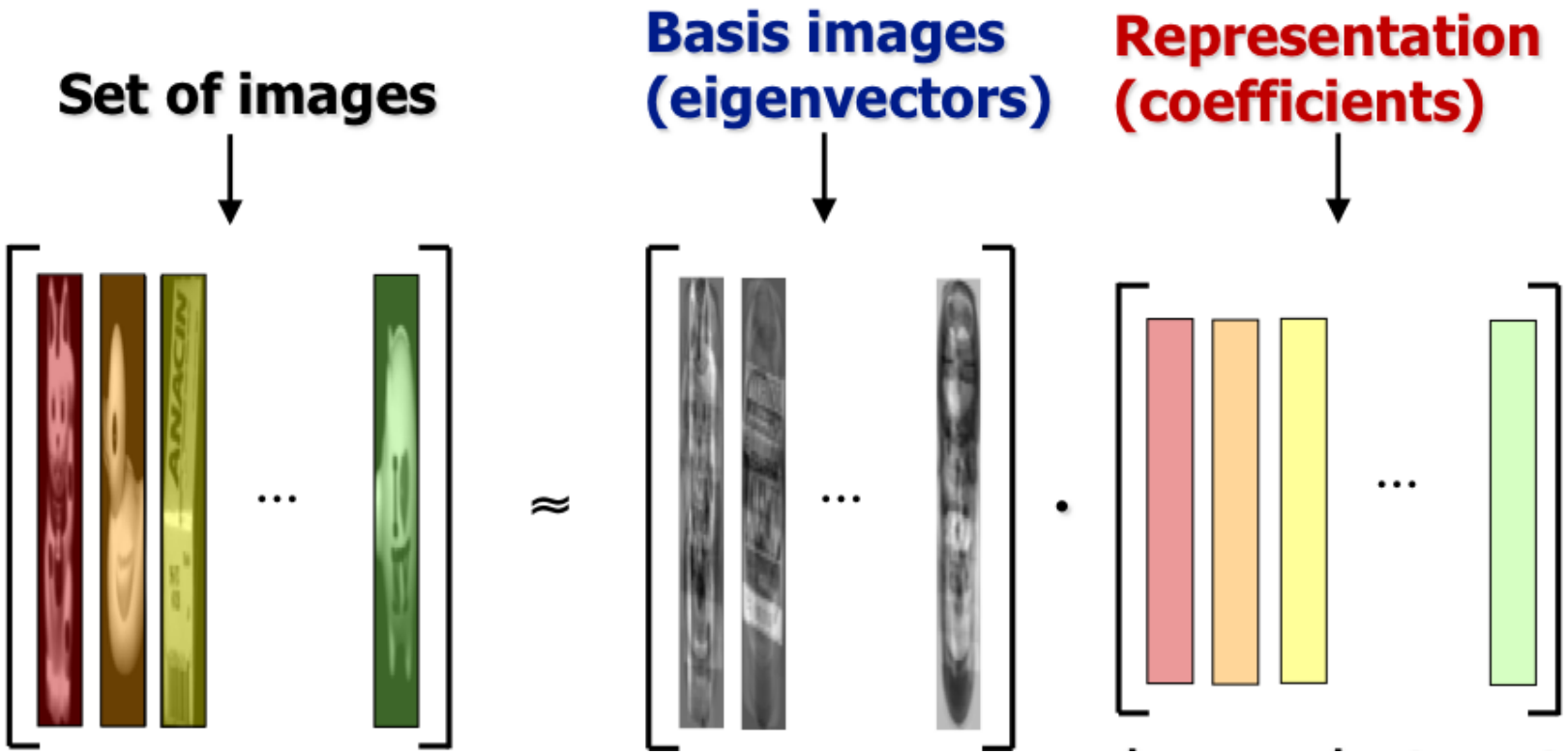
# “Eigenfaces”

- The eigenvectors look like faces. Scary faces.



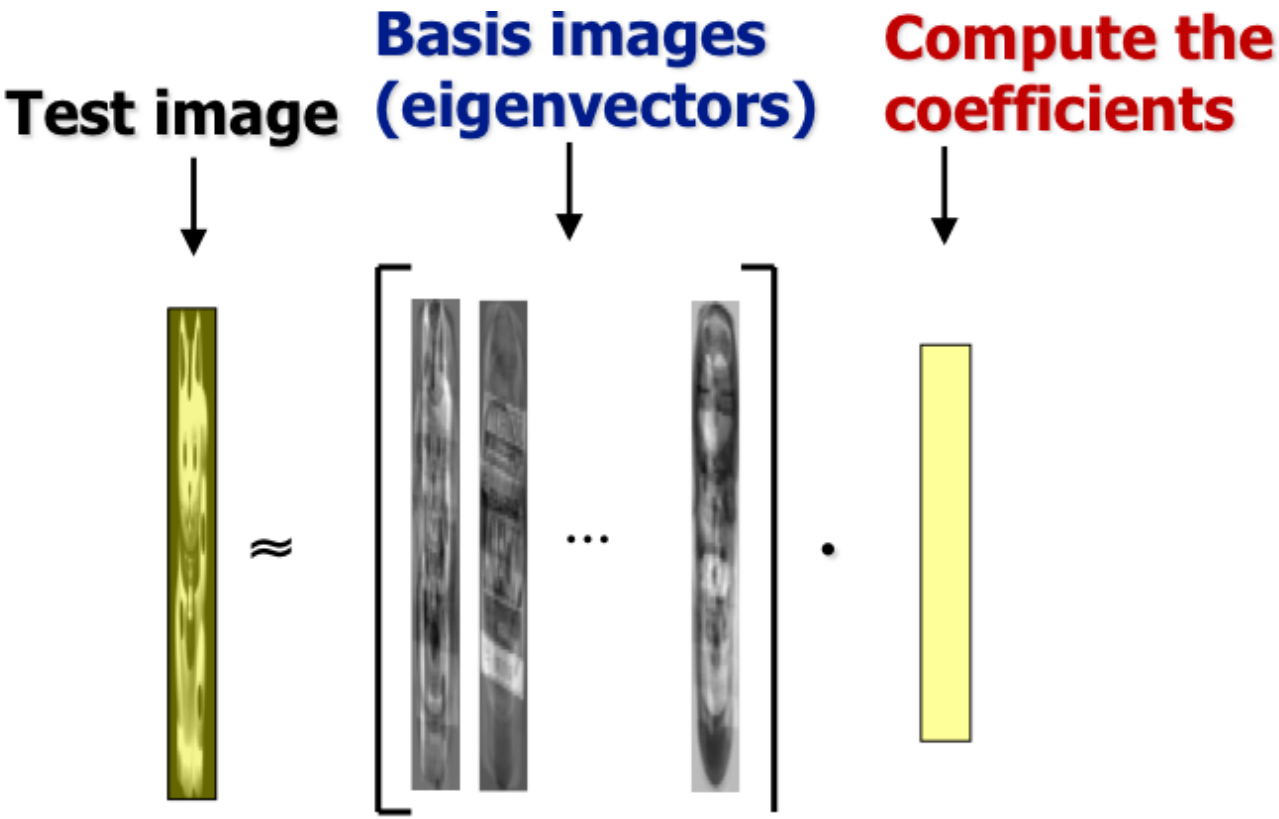
# "Eigenfaces"

- Remember the coefficients for each training "class" (the person the face image belongs to). This is our representation of the class.



# "Eigenfaces"

- Now we want to classify a new test image.
- We subtract the average face, vectorize and compute the coefficients. Easy. The coefficients can be computed as before: $\mathbf{a} = U^T \cdot \mathbf{x}$, where $\mathbf{x}$ is the new vectorized test image.



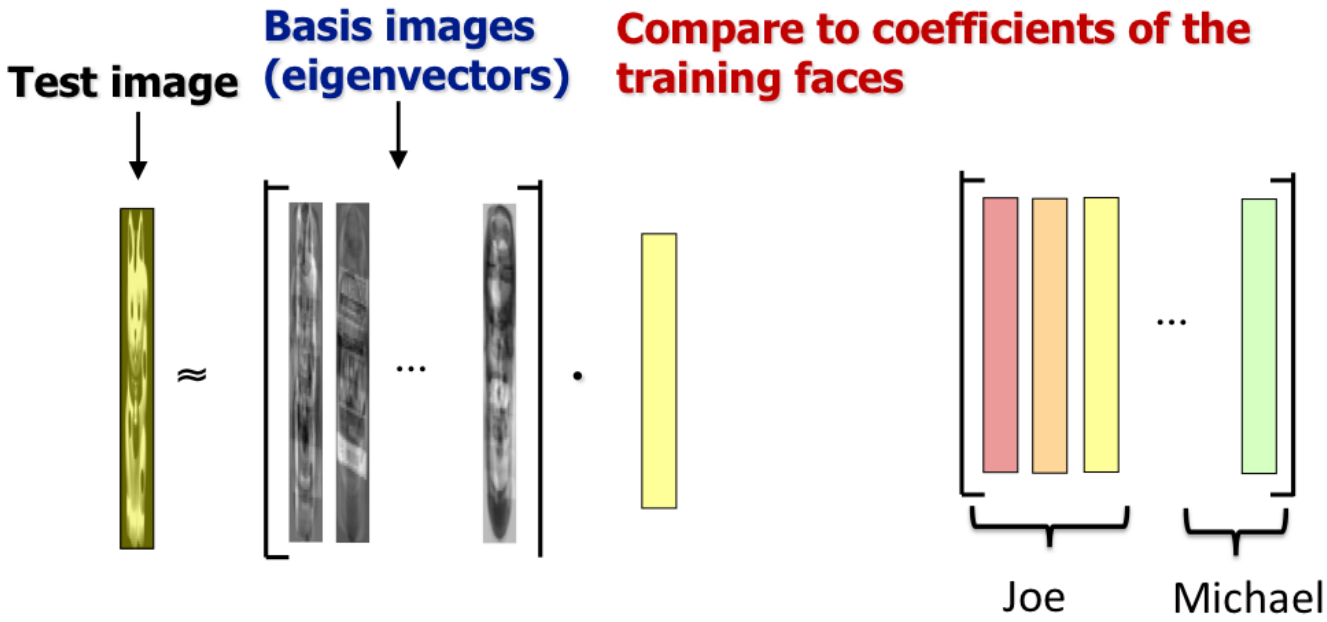
# "Eigenfaces"

- To classify test image, find the training image which has the most similar coefficients. If distance between two coefficient vectors is above threshold, say test image belongs to the winning class, otherwise "Unknown".

$$d = \min_c ||\text{coeff}_{test} - \text{coeff}_{train,c}||$$

$d < \text{thresh}$ → Unknown

$d > \text{thresh}$ → $c^*$ (e.g., Joe)



# Problem?

- Math was easy in those days... And the approach seemed to work pretty well. At least enough to stop thinking about 3D and more intense math.
- Can you see any problems with this approach?
- Can you think of cases for which this approach doesn't work?
- Can you do detection with this approach?

# Problem?

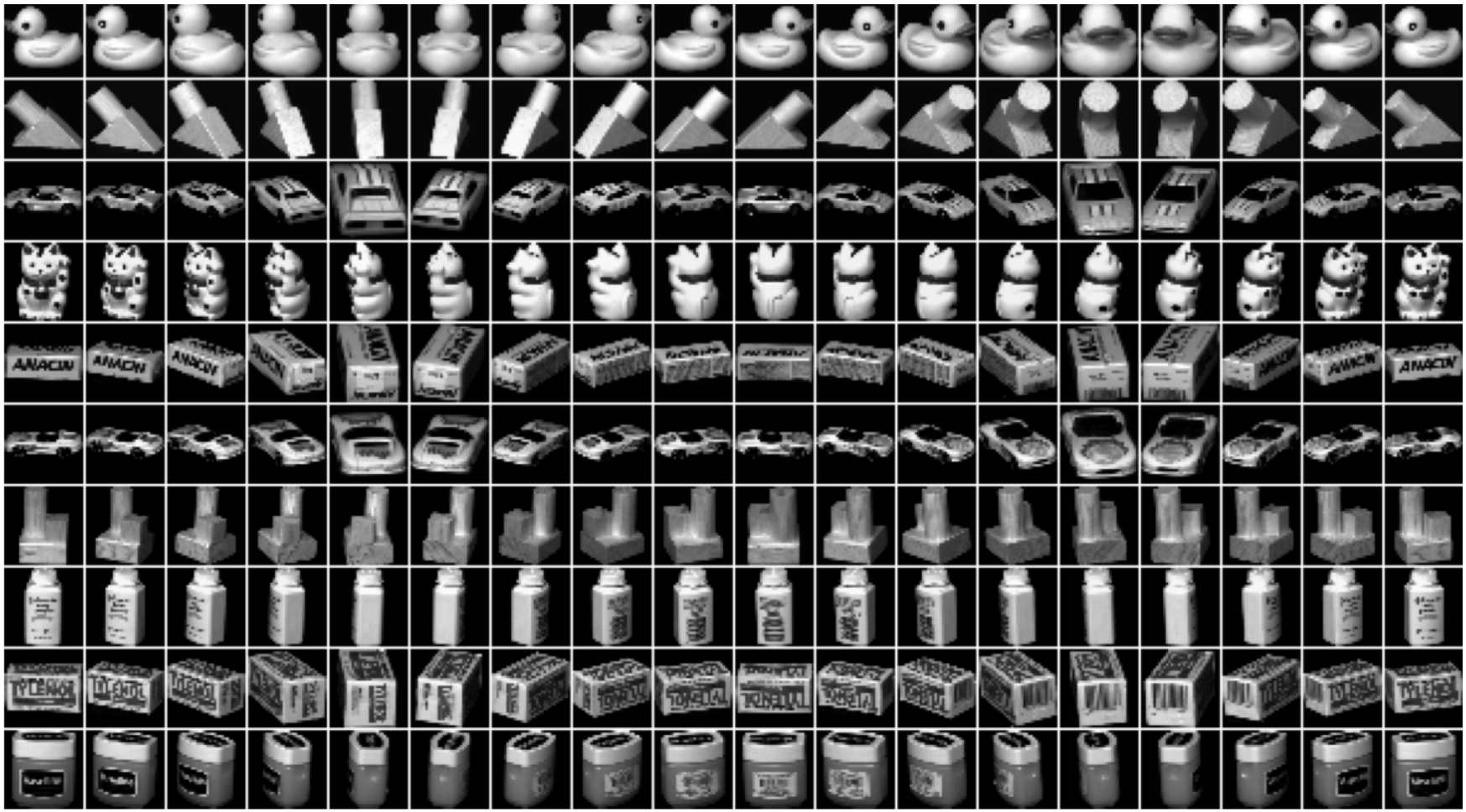
- Requires global registration of patterns (maybe possible for faces, what about other objects?)
- Not robust to clutter, occlusion, geometric transformations. Why?

# Not To Be Unfair

- People did think about 3D in those days.
- Any idea how you could estimate an accurate 3D viewpoint of the depicted object with this kind of approach?

# 3D Without Thinking In 3D

- Generate images of objects in all possible viewpoints. Then just apply the same PCA approach and hope for the best.
- This was one of the first datasets in computer vision. It was called COIL.



# The Appearance Era vs Today

- The PCA approach slightly resembles some of the most successful approaches today. For example Neural Networks train on full images (global representation) and they don't care about 3D; "Give me data and I'll memorize it. And pray it will work.".
- How come the PCA approach doesn't work very well but NNs do?

# Recognition Ideas Through History

- 1960s – early 1990s: the geometric era
- 1990s: appearance-based models
- **early 2000: local features**

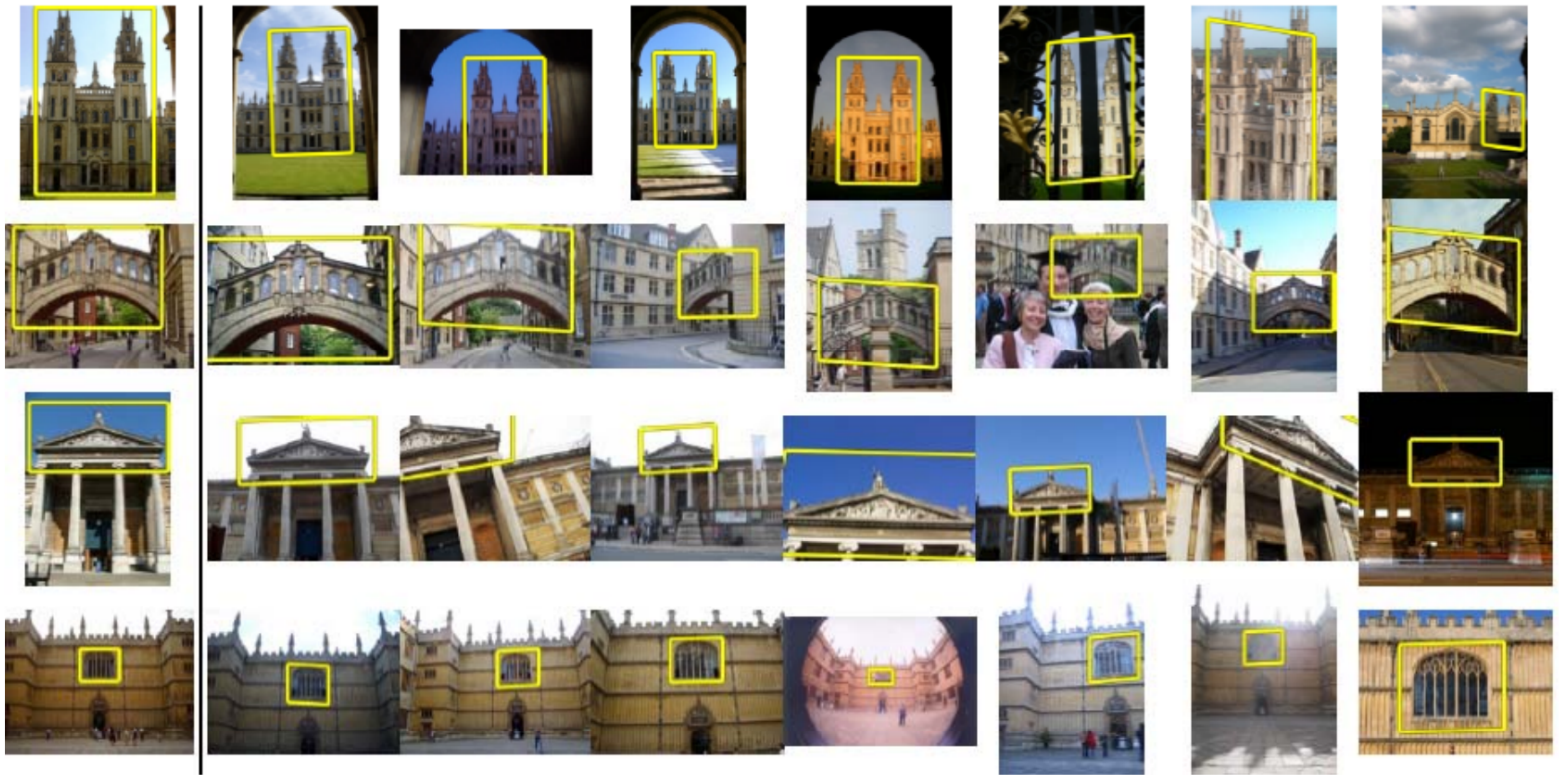
# Local Features

- Back to 3D, this time with the powerful local features (SIFT)
- Forget about object class, focus on instance recognition (e.g. a specific CD/DVD/object vs a generic class such as car or cat)


D. Lowe (1999, 2004)

# Fast Retrieval

- Via clustering and document-like indexing, people could now do super fast image retrieval



Philbin et al. '07

# Problem of SIFT for Class Recognition

- It was shown that SIFT doesn't work very well for object class recognition. Any idea why not?
- But the idea of local features is great. And with this people start revisiting the very old work which said that objects need to be represented with components, parts

# Problem of SIFT for Class Recognition

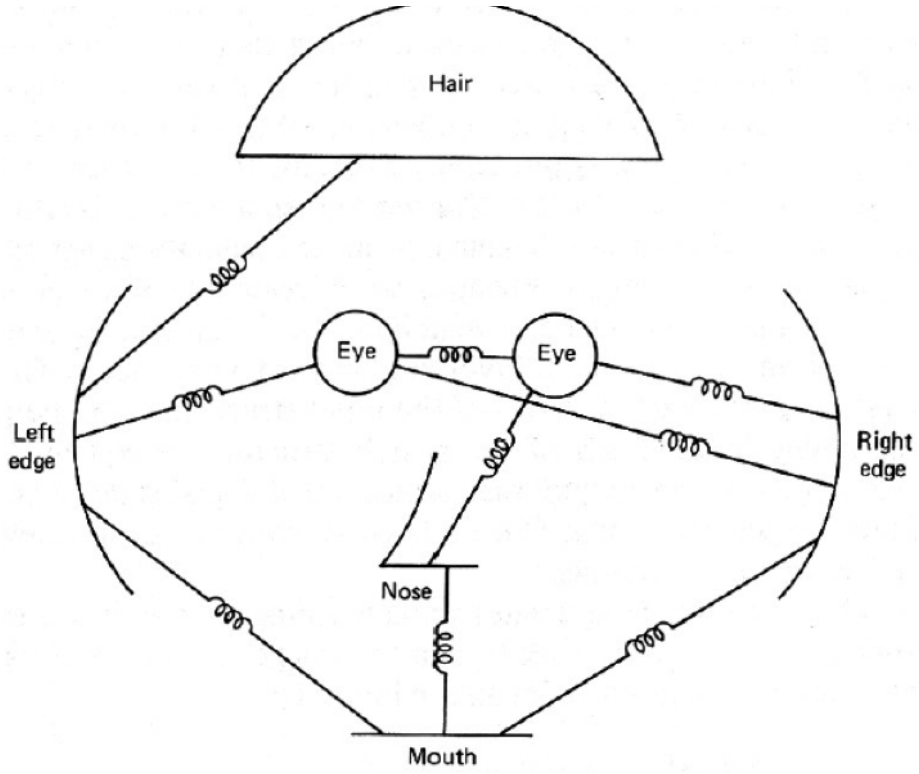
- It was shown that SIFT doesn't work very well for object class recognition. Any idea why not?
- But the idea of local features is great. And with this people start revisiting the very old work which said that objects need to be represented with components, parts

# Recognition Ideas Through History

- 1960s – early 1990s: the geometric era
- 1990s: appearance-based models
- early 2000: local features
- **slightly less early 2000s: parts-based models**

# Parts Are Back

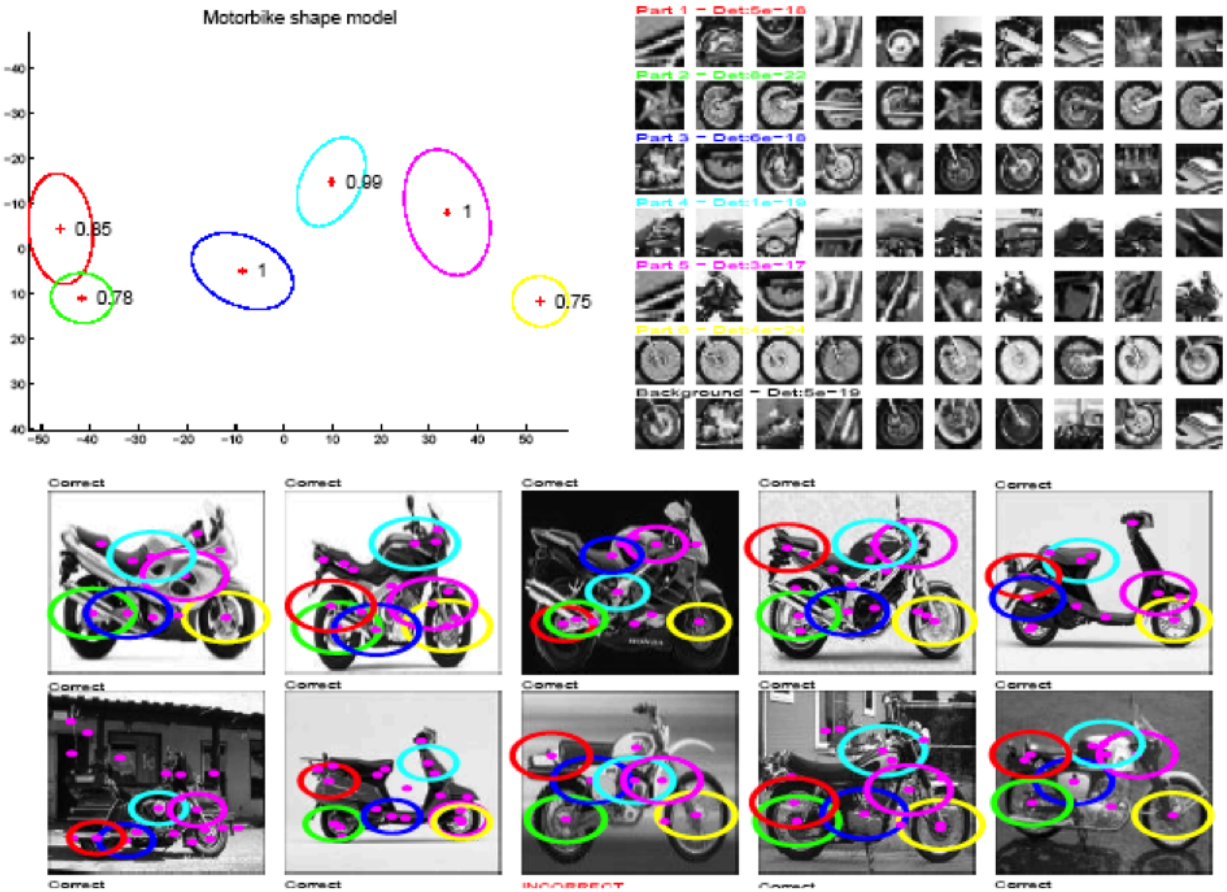
- Object is represented with a set of (meaningful) parts
- We need to model relative locations between parts
- Main difference with old approaches: This time around we are also modeling the **appearance** of object parts




Fischler & Elschlager, 1973

# The Constellation Model

- Parts are represented with clusters of local patches
- Relative locations between parts are modeled with Gaussians

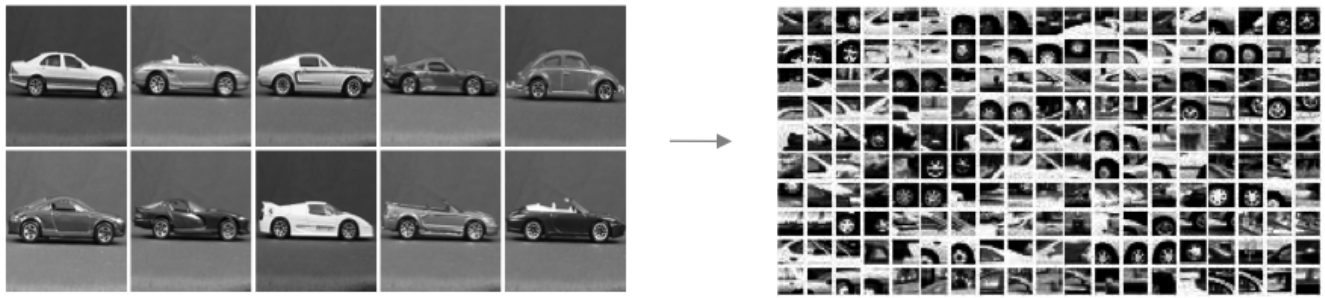



Fergus, Perona, Zisserman, 2003

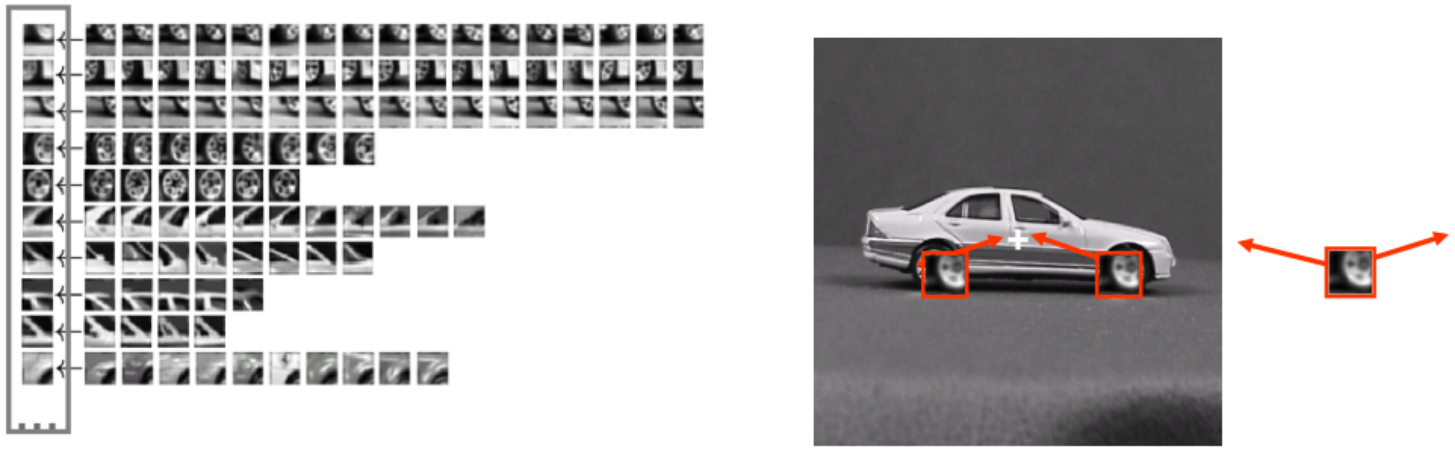
# The Implicit Shape Model

- A Hough-voting based approach

Collect patches from whole training set



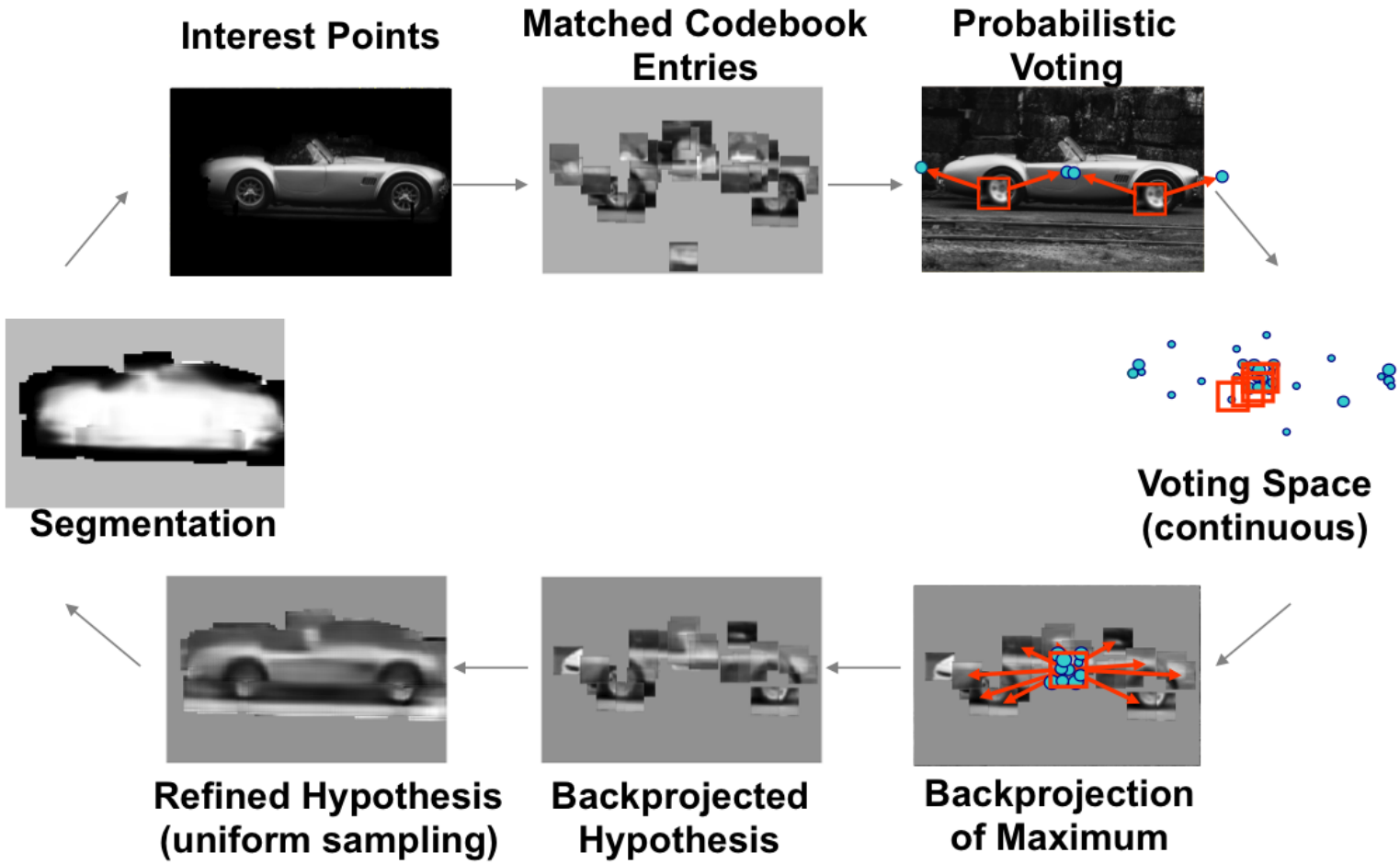
Appearance codebook



Leibe & Schiele, 2004

# The Implicit Shape Model

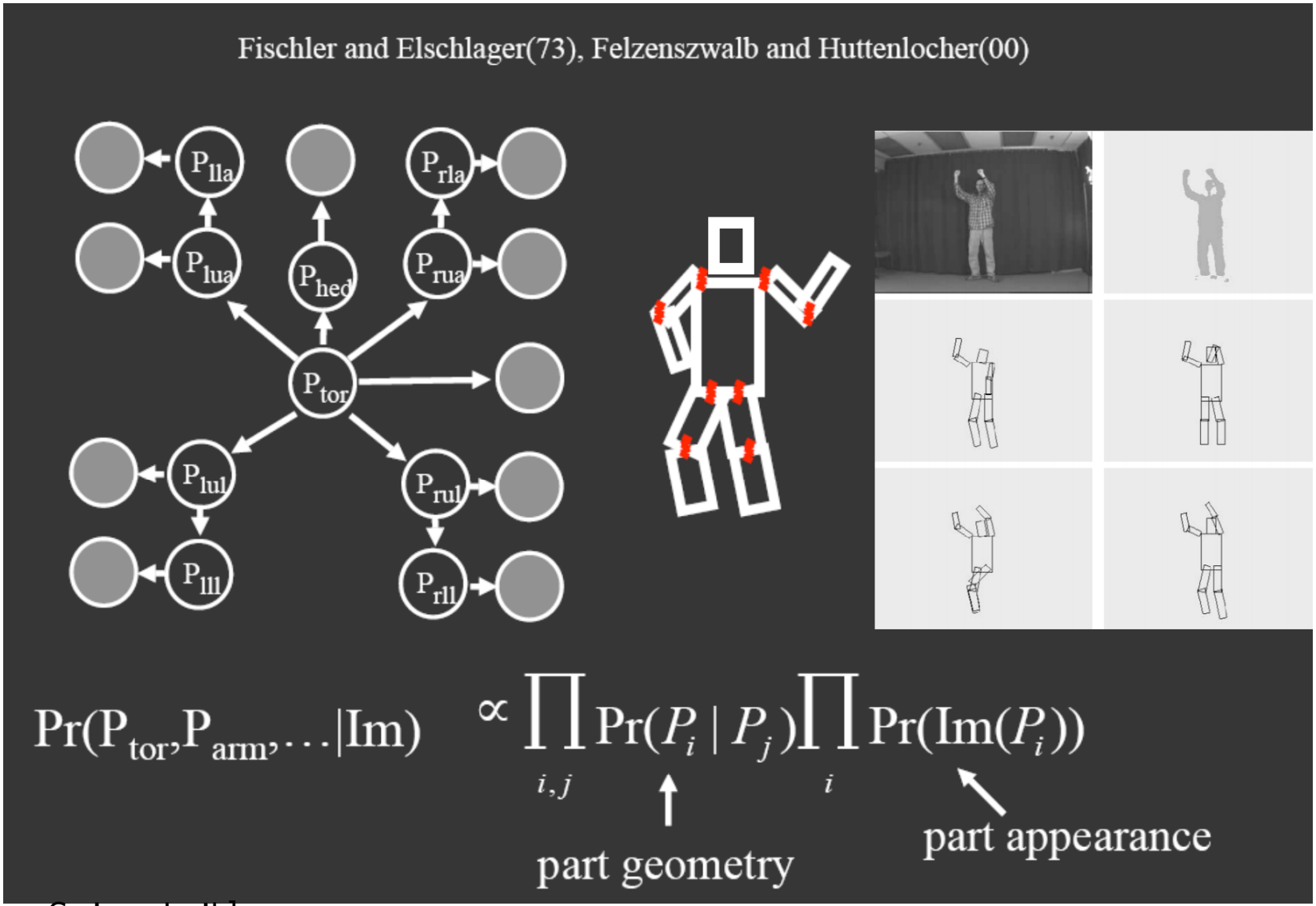
- We will talk more about this approach next time. It has some nice ideas.




Leibe & Schiele, 2004

# Pictorial Structure Model

- Models dependencies between parts as a tree. Good for representing humans.



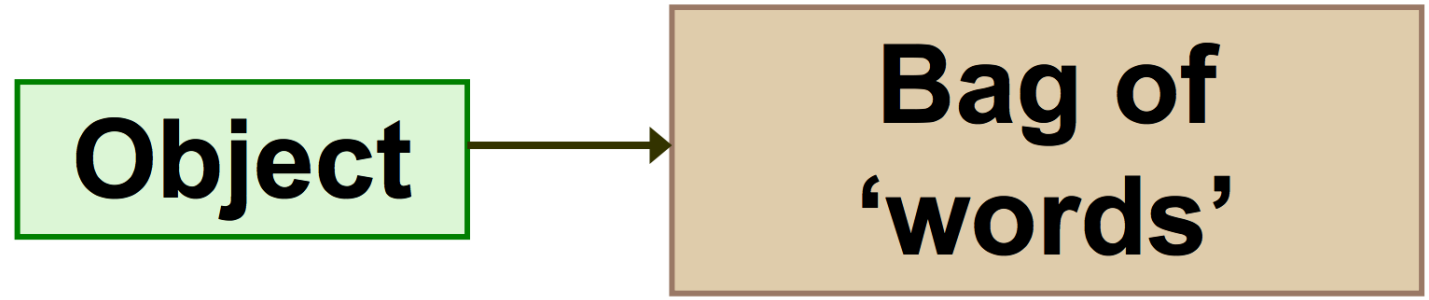

[Source: S. Lazebnik]

# Recognition Ideas Through History

- 1960s – early 1990s: the geometric era
- 1990s: appearance-based models
- early 2000: local features
- slightly less early 2000s: parts-based models
- **mid-2000s: bags of features**

# Bags-of-words Models

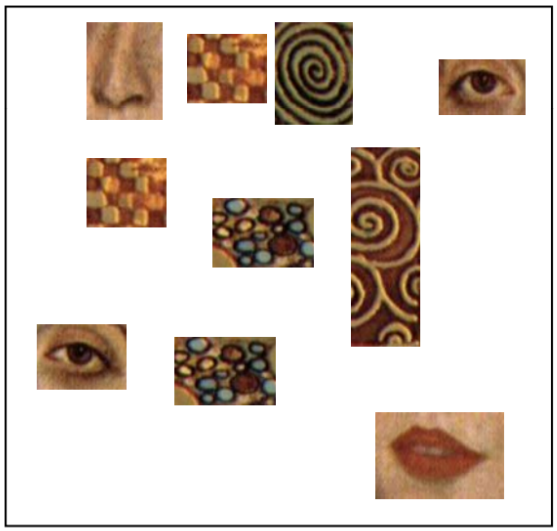
- Since parts (local features) work so well, people got a crazy idea: let's just forget about spatial relations altogether.

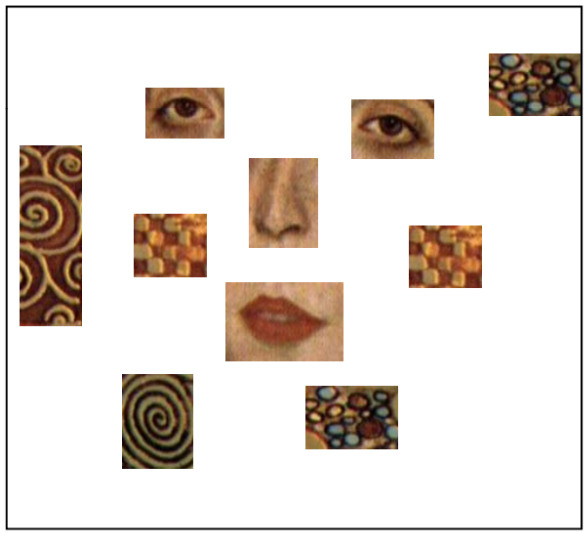



[Source: S. Lazebnik]

# Bags-of-words Models

- Let's just represents object with orderless features. A histogram of features. We have seen how this works for object retrieval, remember?





[Pic from: S. Lazebnik]

# Bags-of-words Models

- Take image, extract features. Cluster them across dataset $\rightarrow$ visual words.
- Assign each feature in image to visual word. Form a histogram of visual words over the full image. This is the descriptor of the image.
- Train a classifier on the BoW descriptors.

# Bags-of-words Models

- Take image, extract features. Cluster them across dataset $\rightarrow$ visual words.
- Assign each feature in image to visual word. Form a histogram of visual words over the full image. This is the descriptor of the image.
- Train a classifier on the BoW descriptors.
- Worked surprisingly well despite the lack of meaningful representation

# Bags-of-words Models

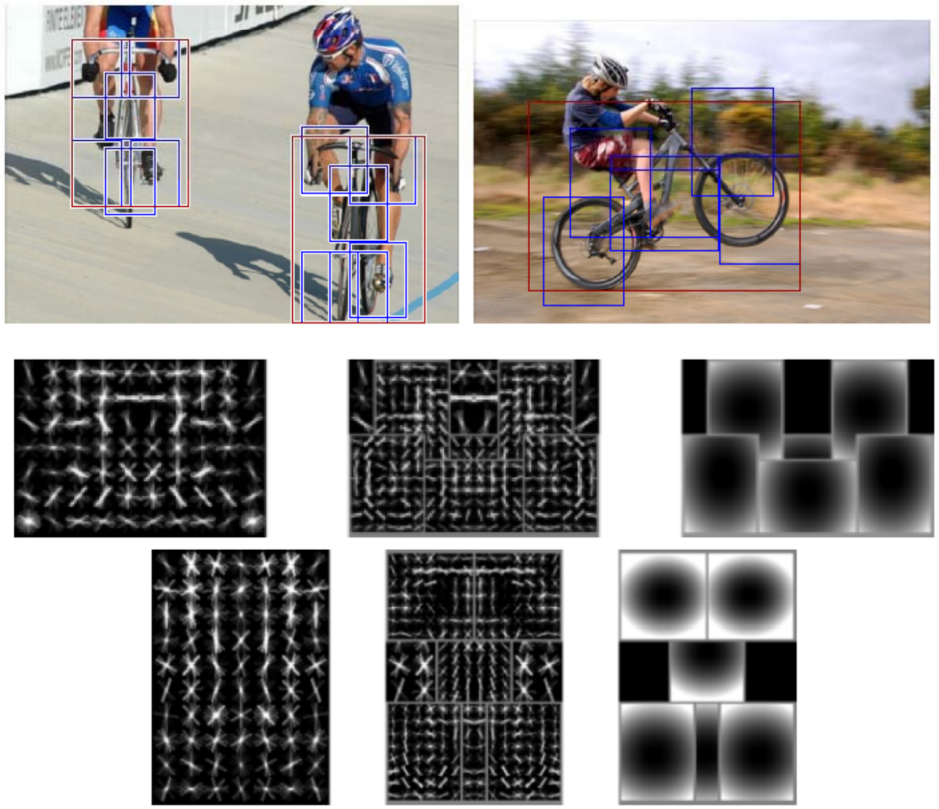
- Take image, extract features. Cluster them across dataset $\rightarrow$ visual words.
- Assign each feature in image to visual word. Form a histogram of visual words over the full image. This is the descriptor of the image.
- Train a classifier on the BoW descriptors.
- Worked surprisingly well despite the lack of meaningful representation

# Recognition Ideas Through History

- 1960s – early 1990s: the geometric era
- 1990s: appearance-based models
- early 2000: local features
- slightly less early 2000s: parts-based models
- mid-2000s: bags of features
- **2007-2013: deformable part models**

# Deformable Part-based Model

- Parts are back yet once again. This time equipped with a powerful Machine Learning technique (latent SVM) and a great feature (HOG)
- The detector is a sliding window. It explores each window in an image, extracts features and classifies it object-no object with an SVM classifier.



P. Felzenszwalb, R. Girshick, D. McAllester, D. Ramanan, "Object Detection with Discriminatively Trained Part-Based Models," PAMI 2009

[Adopted from: S. Lazebnik]

# Recognition Ideas Through History

- 1960s – early 1990s: the geometric era
- 1990s: appearance-based models
- early 2000: local features
- slightly less early 2000s: parts-based models
- mid-2000s: bags of features
- 2007-2013: deformable part models
- **and we know what comes after 2013**