C280, Computer Vision

Prof. Trevor Darrell

trevor@eecs.berkeley.edu

Lecture 12: Introduction to Recognition; Boosting, HOG, and Bag-of-Word Models

Last few lectures…

- Feature-based Alignment
  - Stitching images together
  - Homographies, RANSAC, Warping, Blending
  - Global alignment of planar models
- Dense Motion Models
  - Local motion / feature displacement
  - Parametric optic flow
- Stereo / ‘Multi-view’: Estimating depth with known inter-camera pose
- ‘Structure-from-motion’: Estimation of pose and 3D structure
  - Factorization approaches
  - Global alignment with 3D point models

Recognition Challenges / Overview

Object Categorization

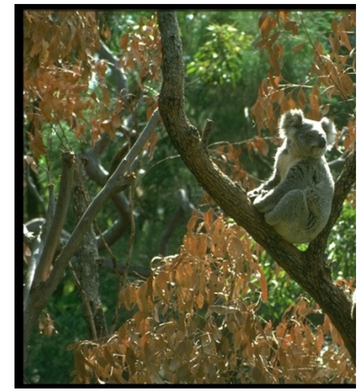
- How to recognize ANY car

- How to recognize ANY cow

Challenges: robustness

Illumination

Object pose



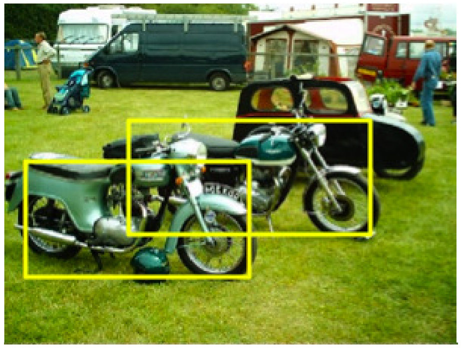
Clutter

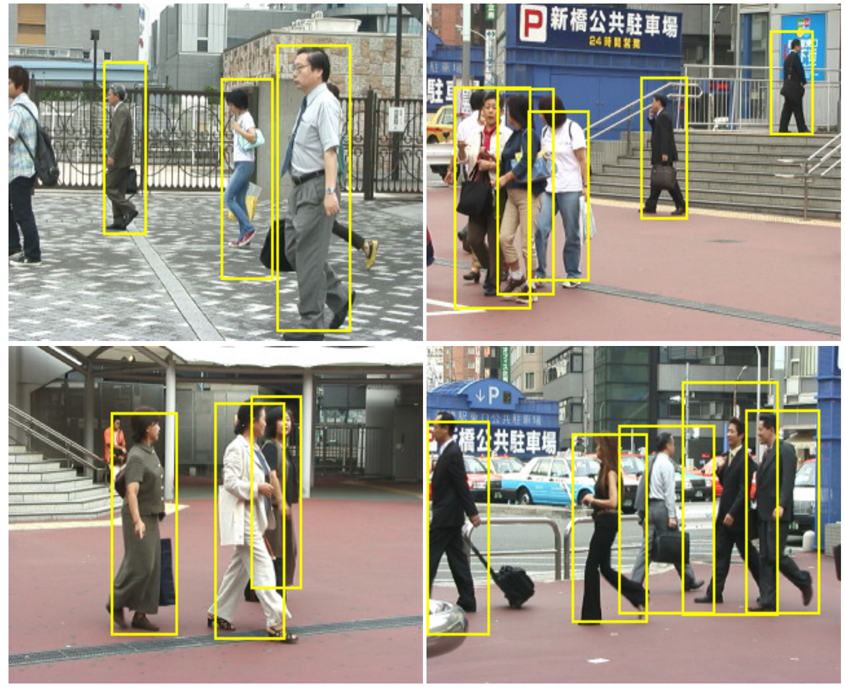
Occlusions

Intra-class appearance

Viewpoint

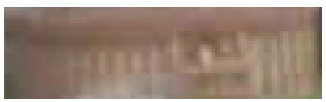
Challenges: robustness



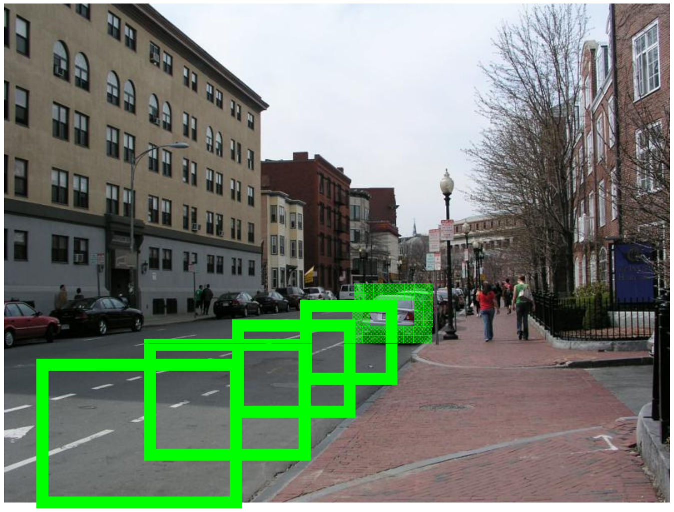


- **Detection in Crowded Scenes**
  - Learn object variability
    - Changes in appearance, scale, and articulation
  - Compensate for clutter, overlap, and occlusion

Challenges: context and human experience



Challenges: context and human experience

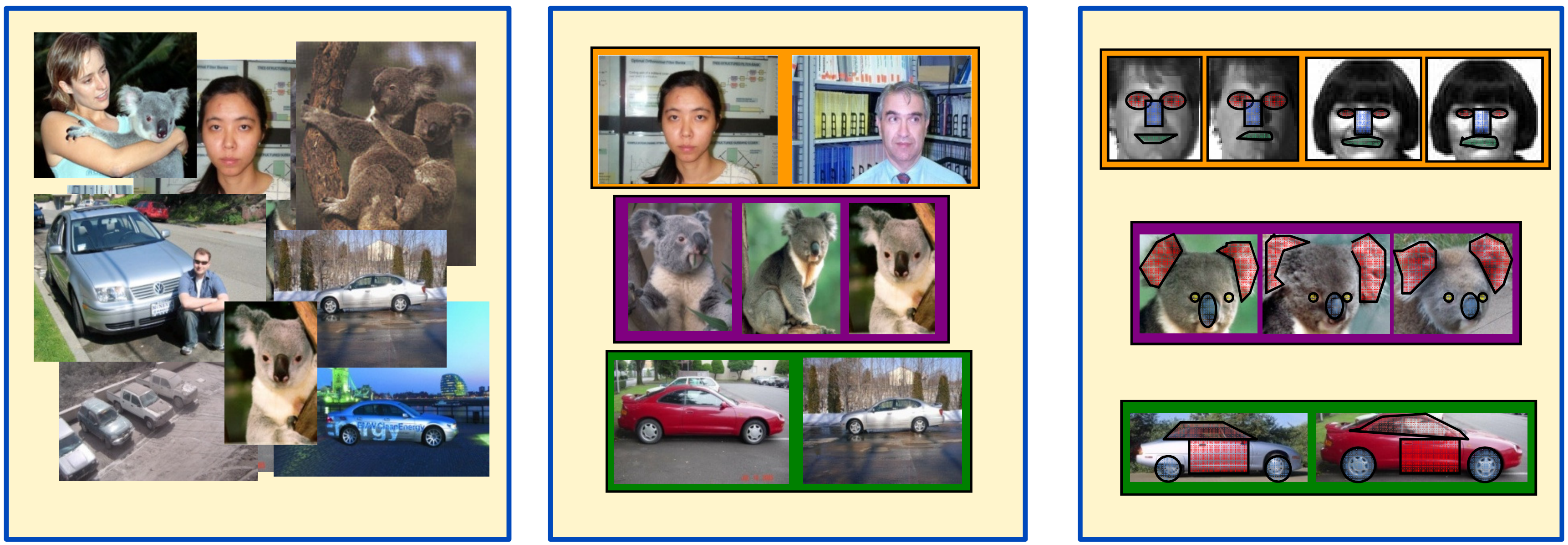


Context cues

Image credit: D. Hoeim

Challenges: learning with minimal supervision

Less More



Unlabeled, multiple objects

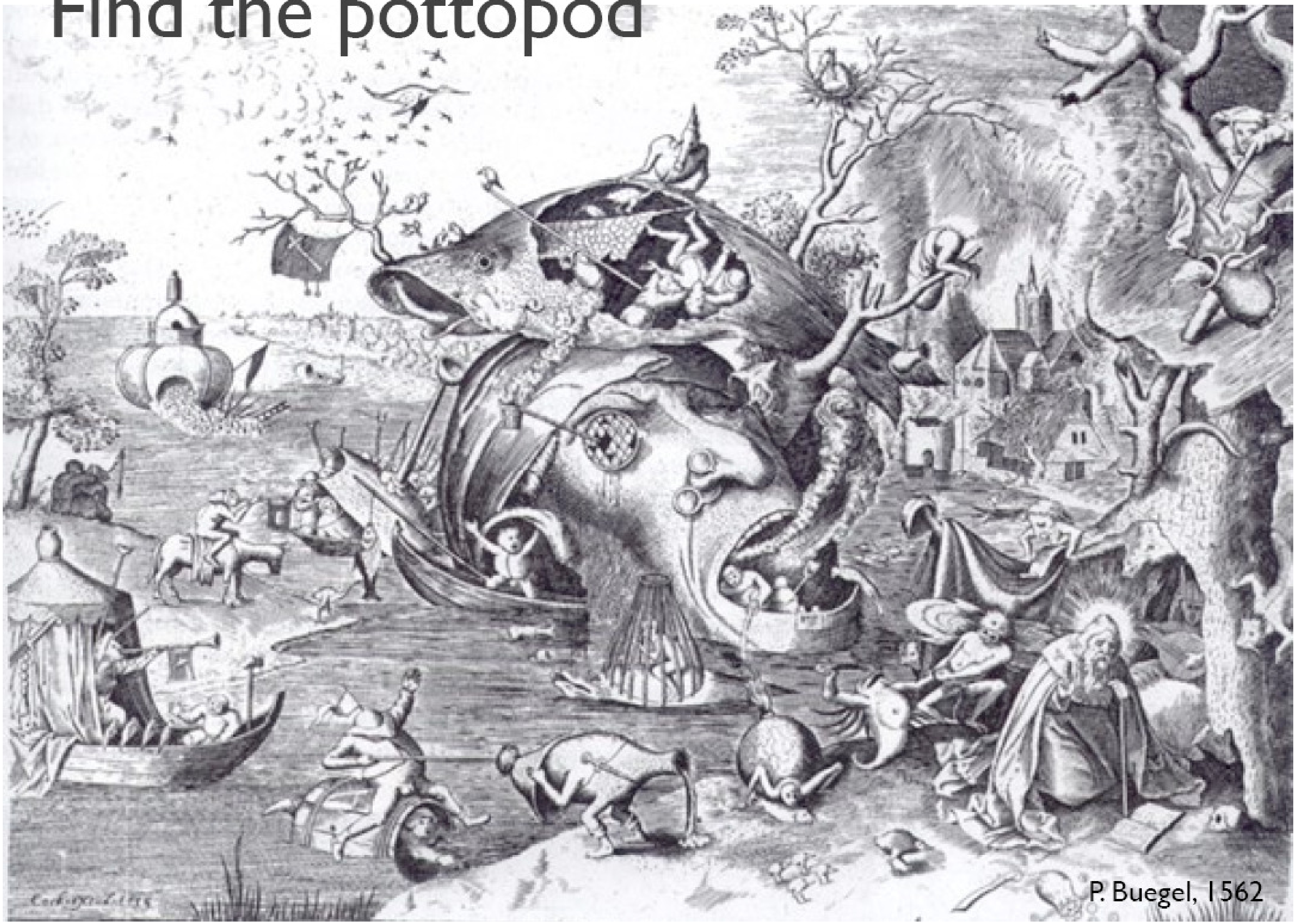
Classes labeled, some clutter

Cropped to object, parts and classes labeled

Slide from Pietro Perona, 2004 Object Recognition workshop

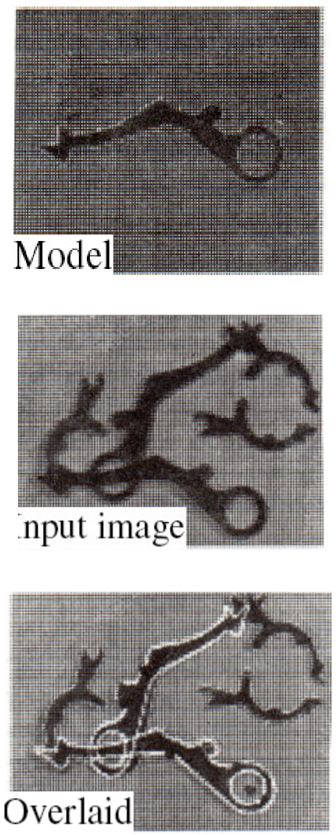
This is a pottopod

Find the pottopod



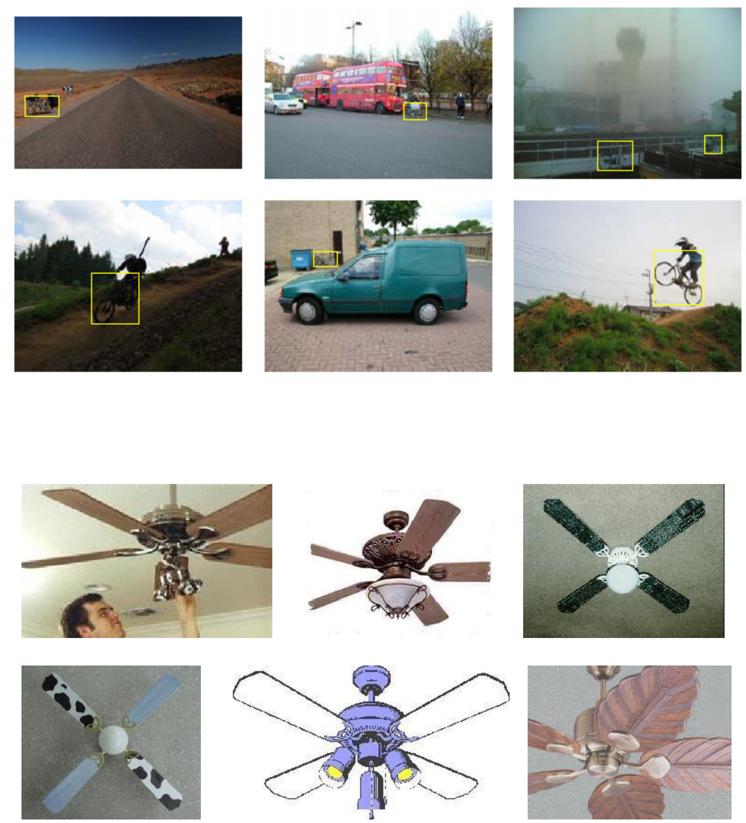
Slide from Pietro Perona, 2004 Object Recognition workshop

Rough evolution of focus in recognition research



1980s

1990s to early 2000s



2000-2010...

Inputs/outputs/assumptions

- What is the **goal**?
  - Say yes/no as to whether an object present in image

  And/or:
  - Determine pose of an object, e.g. for robot to grasp
  - Categorize all objects
  - Forced choice from pool of categories
  - Bounding box on object
  - Full segmentation
  - Build a model of an object category

Today

- Scanning window paradigm
- GIST
- HOG
- Boosted Face Detection
- Local-feature Alignment; from Roberts to Lowe...
- BOW Indexing

Next three lectures

- Thursday: learning object categories from the web
  - LSA and LDA models
  - Harvesting training data from the web
  - Exploiting image and text
- Tues. Oct. 20th: Generative models
  - Condensation
  - ISM
  - Transformed-HDPs
  - More Context…
- Thurs. Oct. 22nd: Advanced BOW kernels
  - Pyramid and spatial-pyramid match
  - Multi-kernel learning
  - Latent-part SVM models

Scanning windows…

Detection via classification: Main idea

Basic component: a binary classifier

→ Car/non-car Classifier

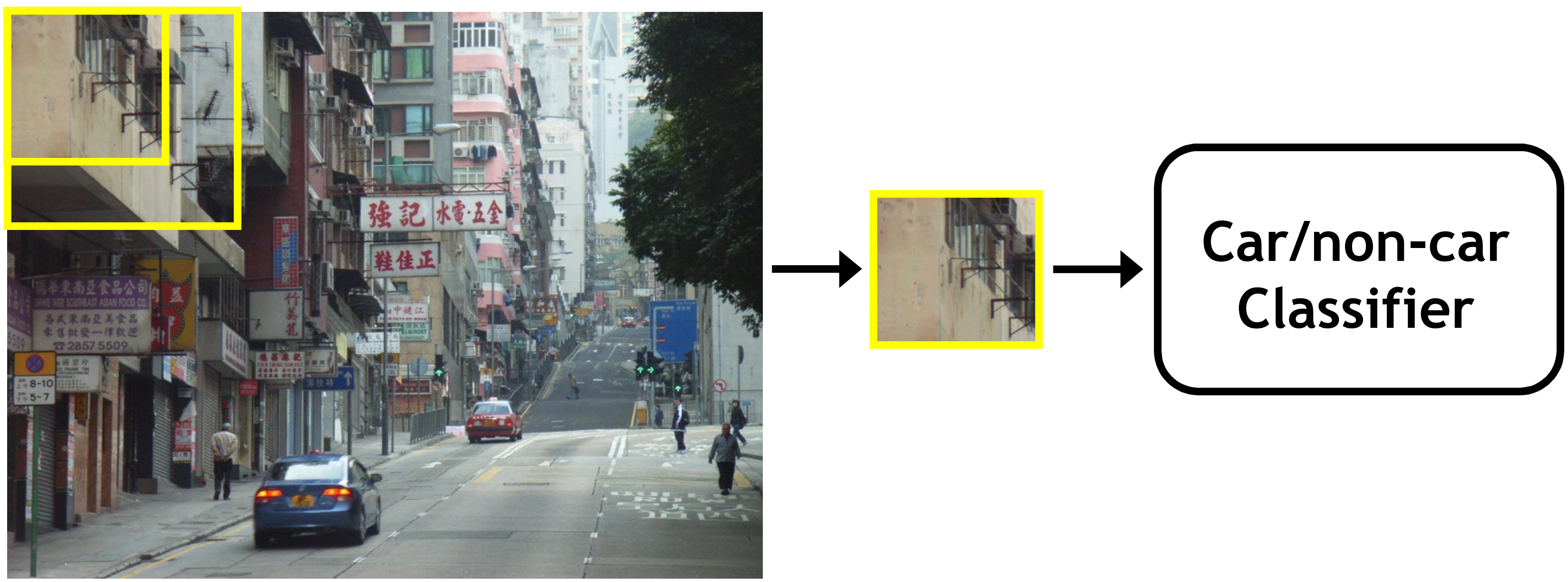
↓

No, not a car.

Yes, car.

Detection via classification: Main idea

If object may be in a cluttered scene, slide a window around looking for it.

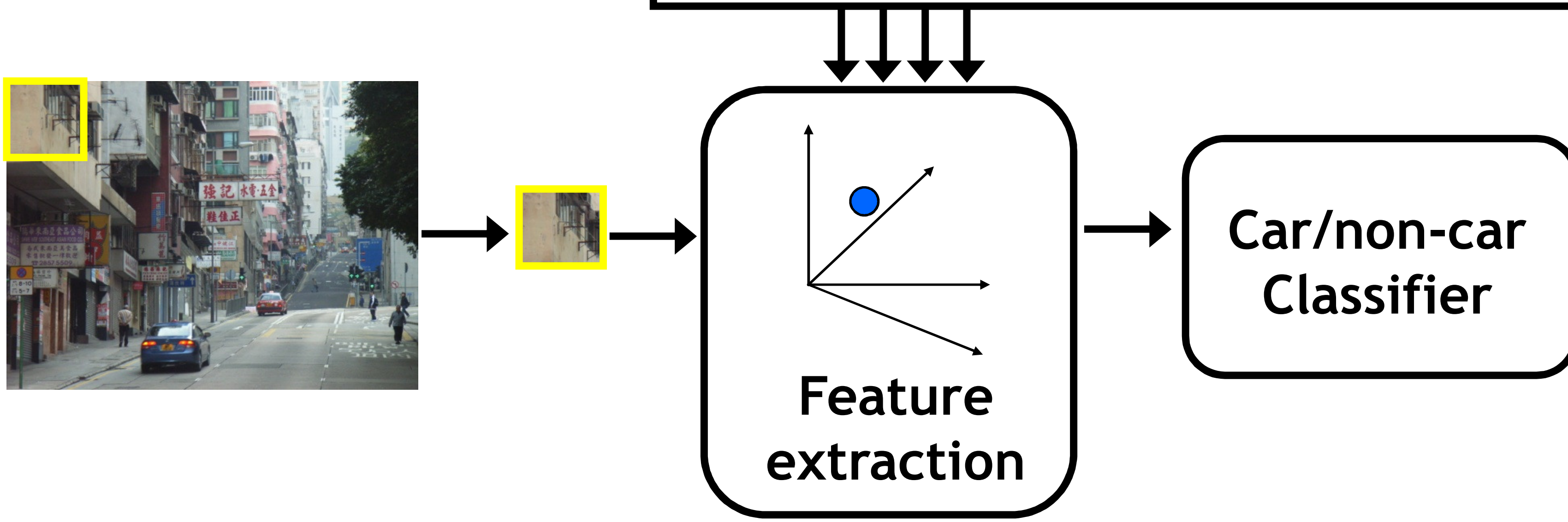


(Essentially, our skin detector was doing this, with a window that was one pixel big.)

Detection via classification: Main idea

Fleshing out this pipeline a bit more, we need to:

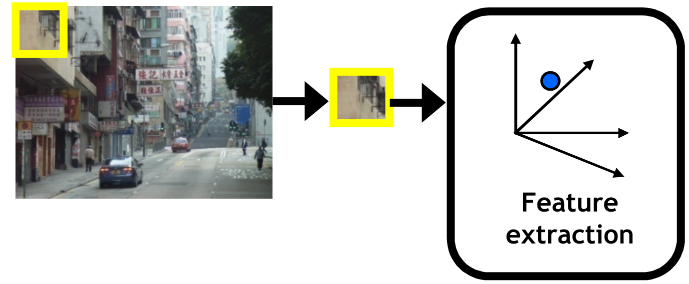
1. Obtain training data
2. Define features
3. Define classifier

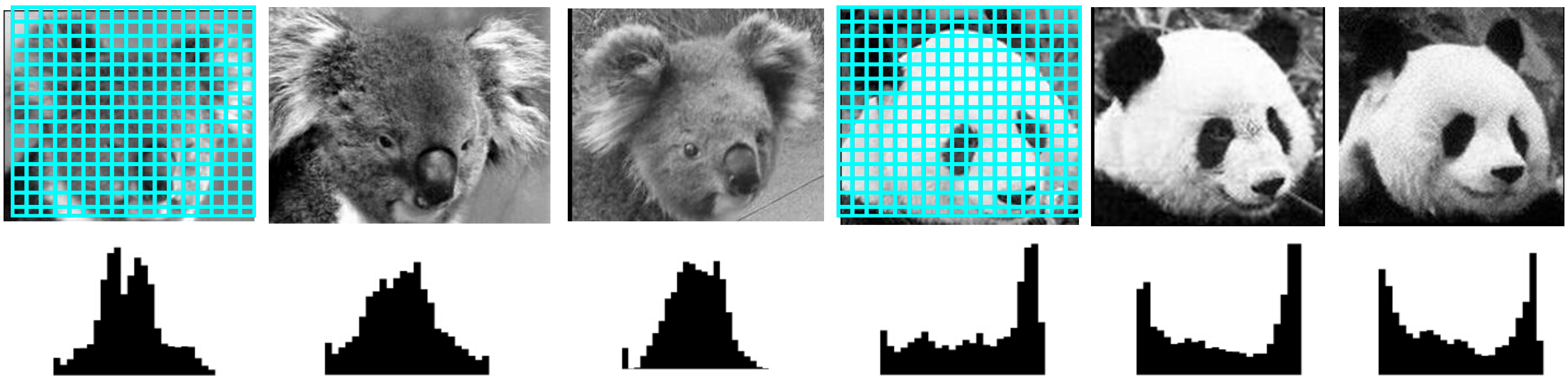


Detection via classification: Main idea

- Consider all subwindows in an image
  - Sample at multiple scales and positions (and orientations)
- Make a decision per window:
  - "Does this contain object category X or not?"

Feature extraction: global appearance





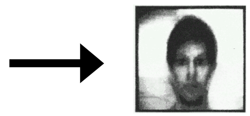
Simple holistic descriptions of image content

- grayscale / color histogram
- vector of pixel intensities

Eigenfaces: global appearance description

An early appearance-based approach to face recognition

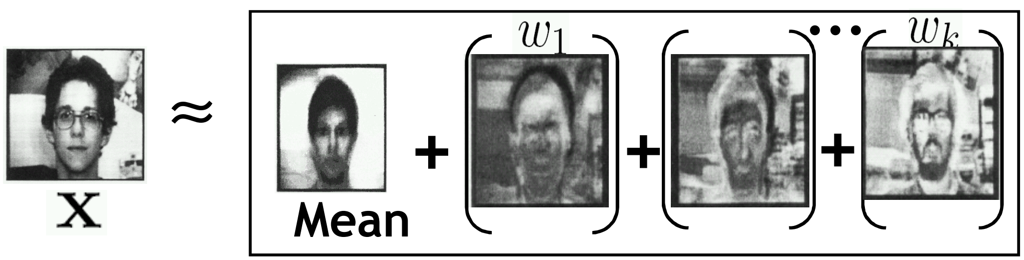
Training images



Mean

Eigenvectors computed from covariance matrix

Generate low-dimensional representation of appearance with a linear subspace.



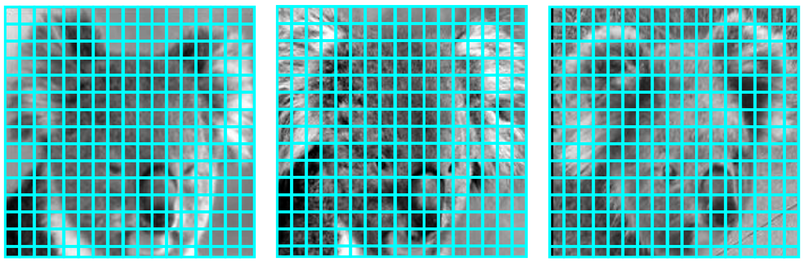
Project new images to "face space".

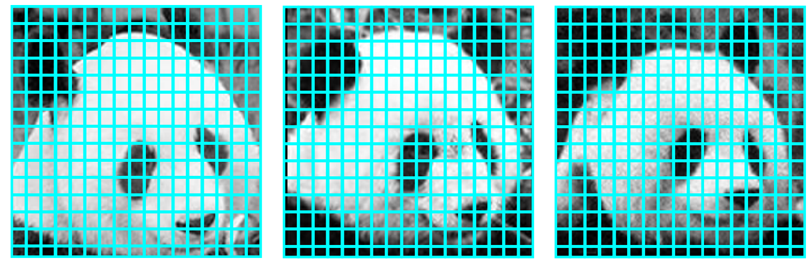
Recognition via nearest neighbors in face space

Turk & Pentland, 1991

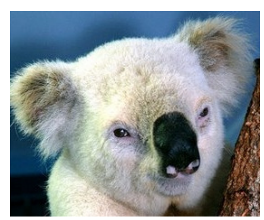
Feature extraction: global appearance

- Pixel-based representations sensitive to small shifts





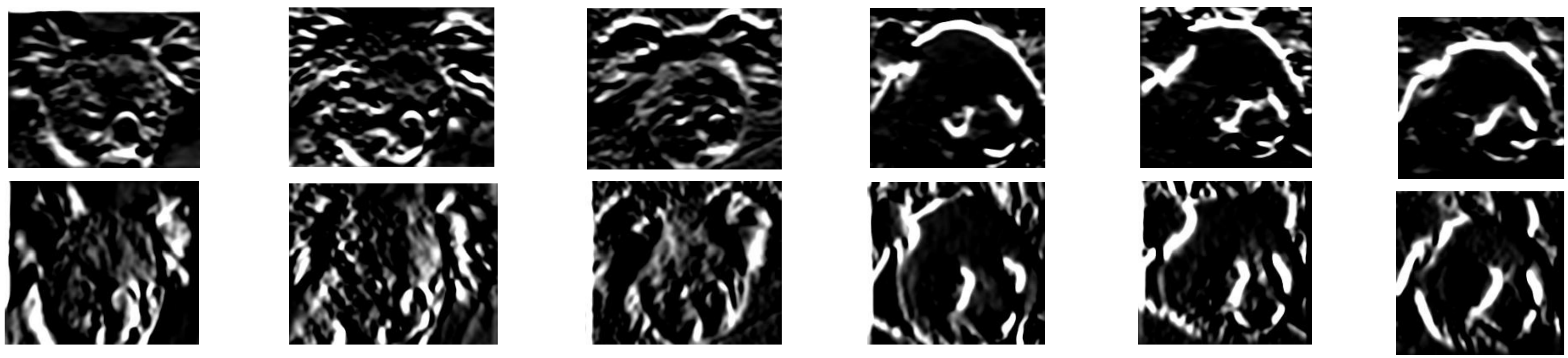
- Color or grayscale-based appearance description can be sensitive to illumination and intra-class appearance variation



Cartoon example:
an albino koala

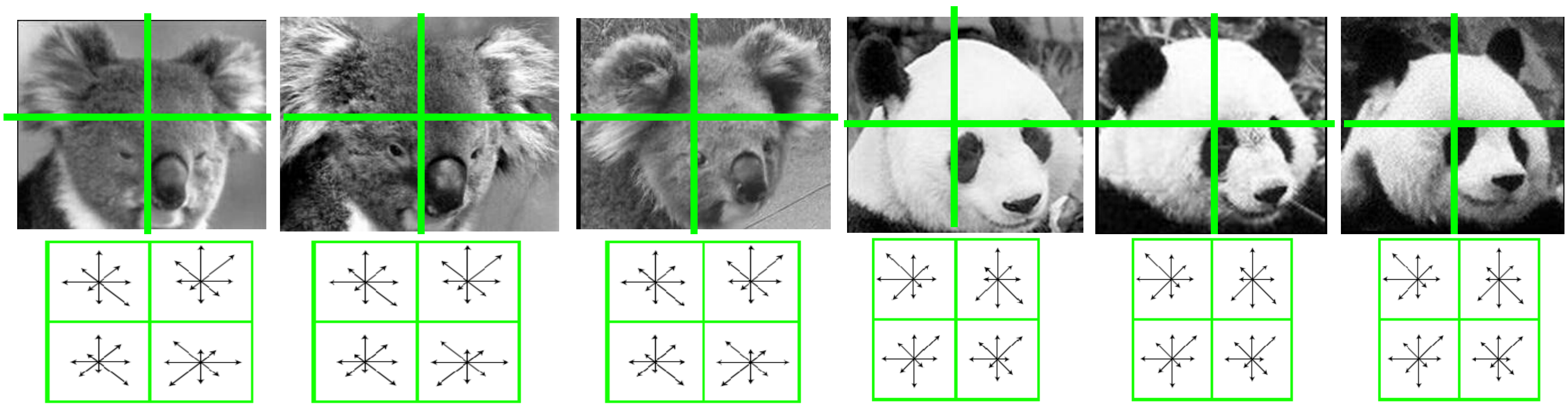
Gradient-based representations

- **Consider edges, contours, and (oriented) intensity gradients**



Gradient-based representations

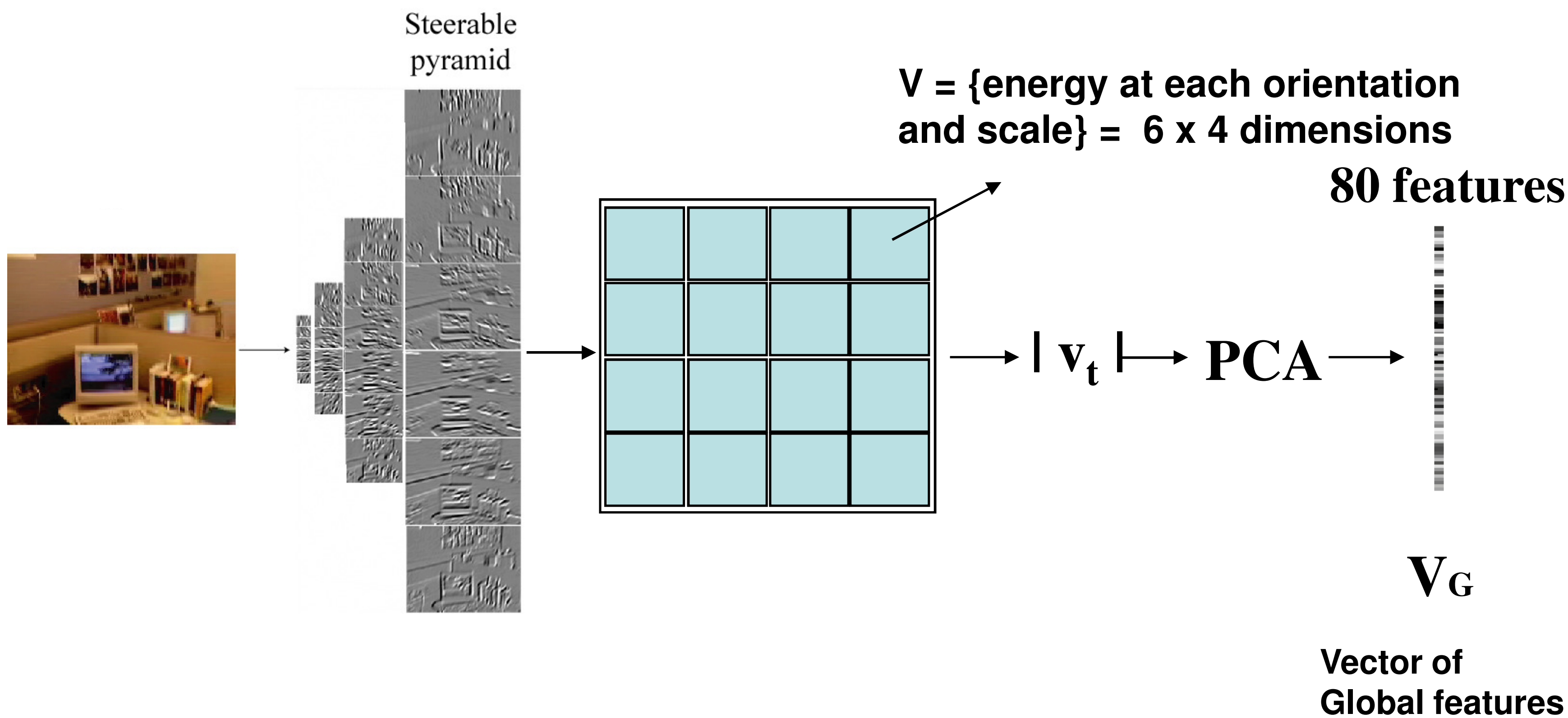
- Consider edges, contours, and (oriented) intensity gradients



- Summarize local distribution of gradients with histogram
  - Locally orderless: offers invariance to small shifts and rotations
  - Contrast-normalization: try to correct for variable illumination

GIST

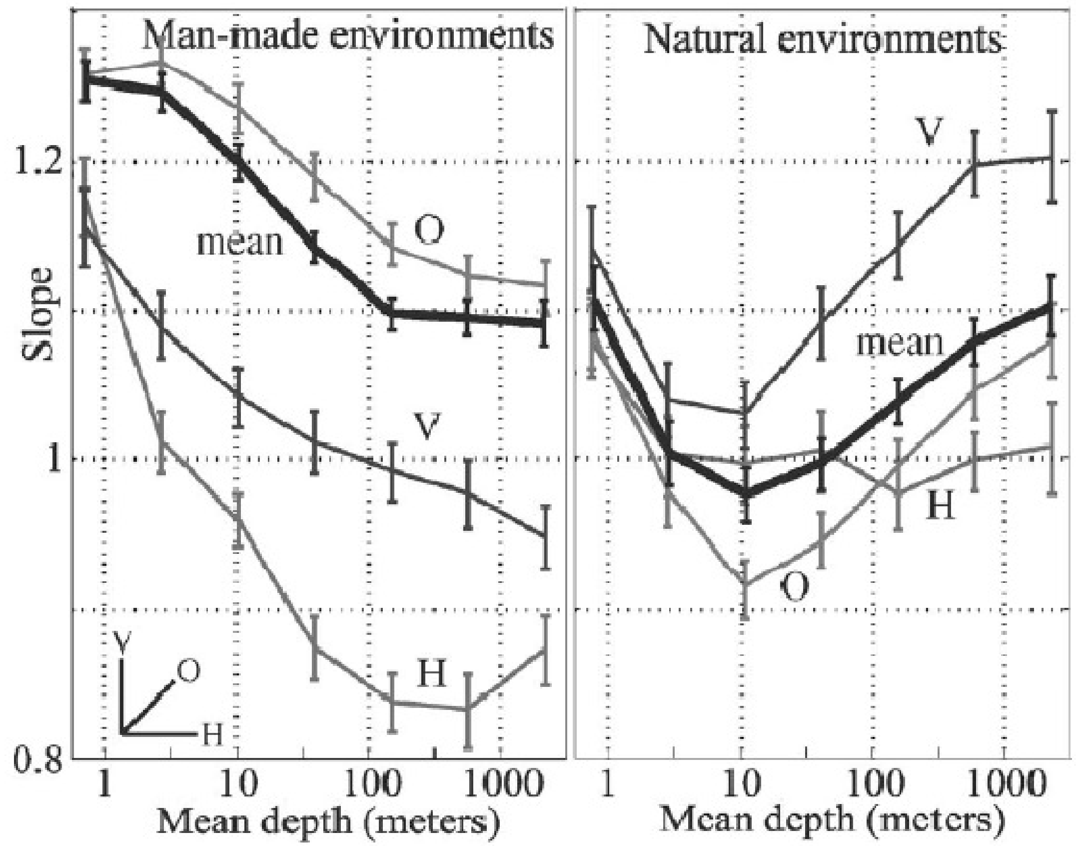
Representing Image Structure with "GIST"



Slide Credit: Olivia

Oliva & Torralba (2001,2002, 2006)

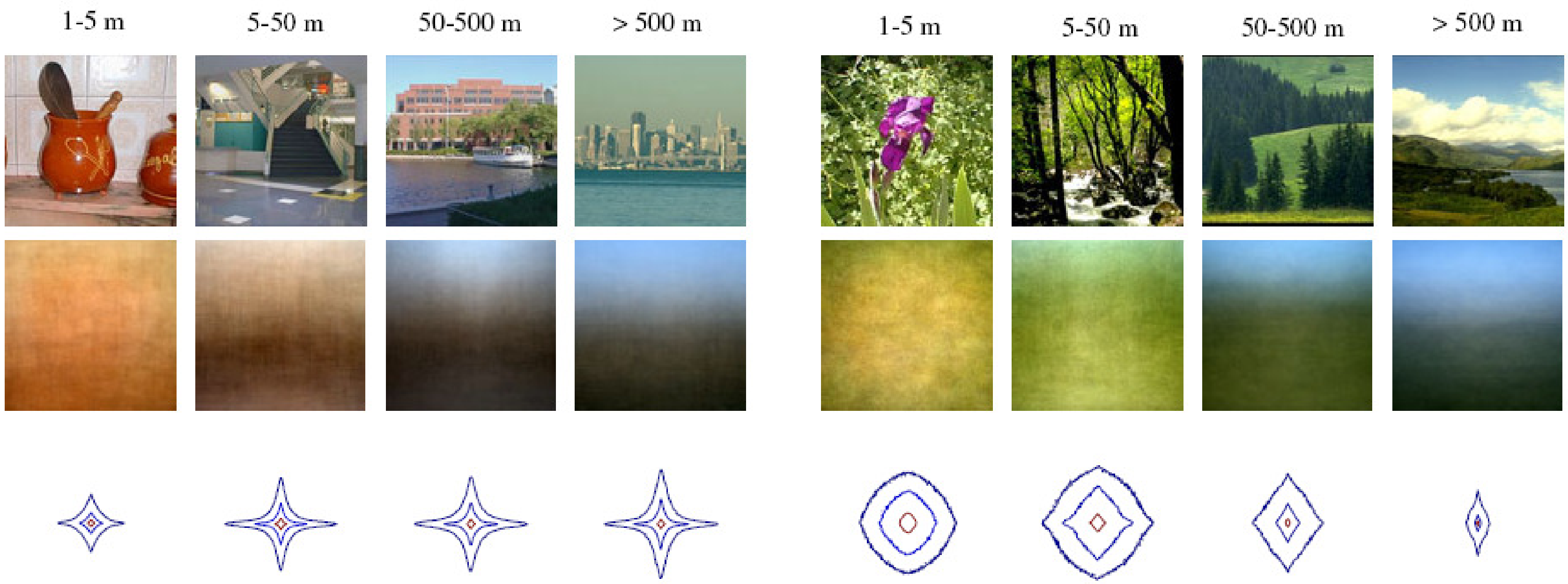
What do Images Statistics say about Depth?



V: Vertical
H: Horizontal
O: Oblique

Slide Credit: Torralba, Olivia, J. Huang

Scene Scale



- “The point of view that any given observer adopts on a specific scene is constrained by the volume of the scene.”
- How does the amount of clutter vary against scene scale in man-made environments? In natural environments?

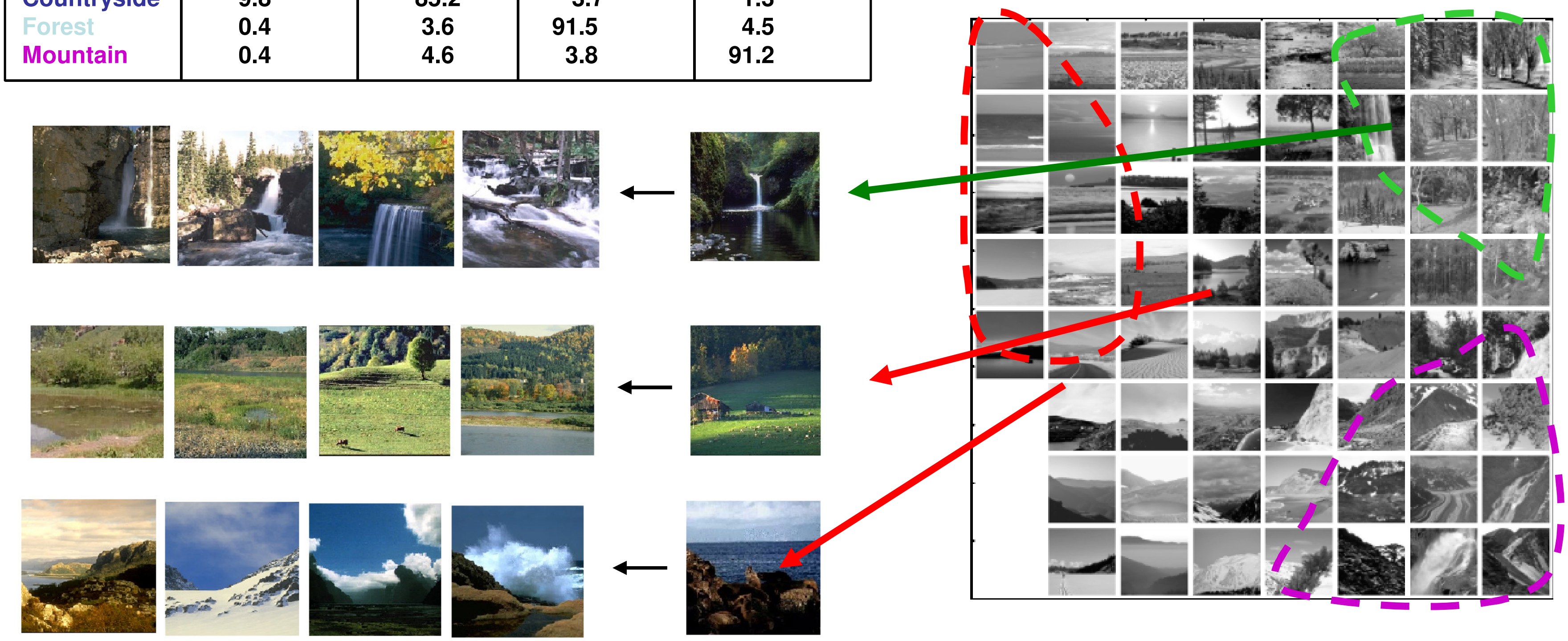
Slide Credit: Torralba, Olivia, J. Huang

Categorization of Natural Scenes

Confusion Matrix (in % using Layout template) :
Classification of prototypical scenes (400 / category)

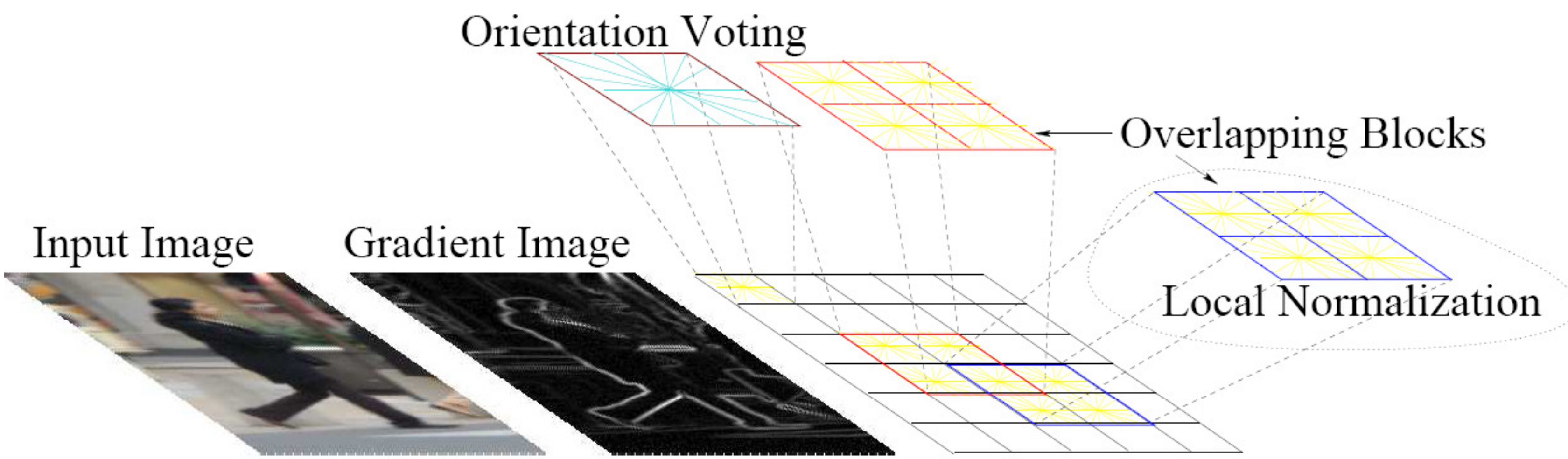
	Coast	Countryside	Forest	Mountain
Coast	88.6	8.9	1.2	1.3
Countryside	9.8	85.2	3.7	1.3
Forest	0.4	3.6	91.5	4.5
Mountain	0.4	4.6	3.8	91.2

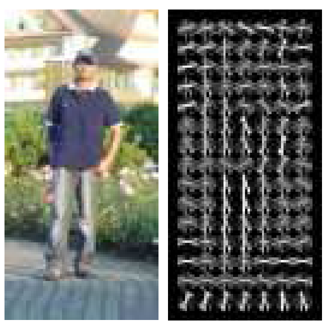
Local organization:
correct for 92 % images
(4 similar images on 7 K-NN)



HOG

Gradient-based representations: Histograms of oriented gradients (HoG)



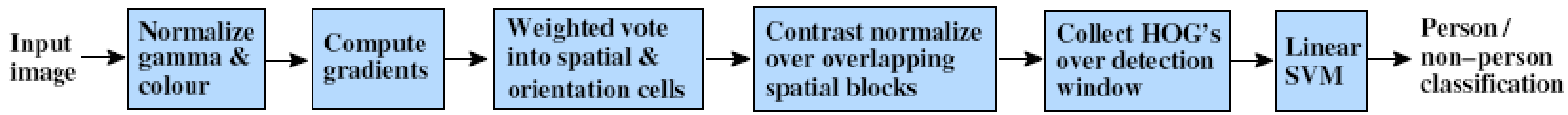


Map each grid cell in the input window to a histogram counting the gradients per orientation.

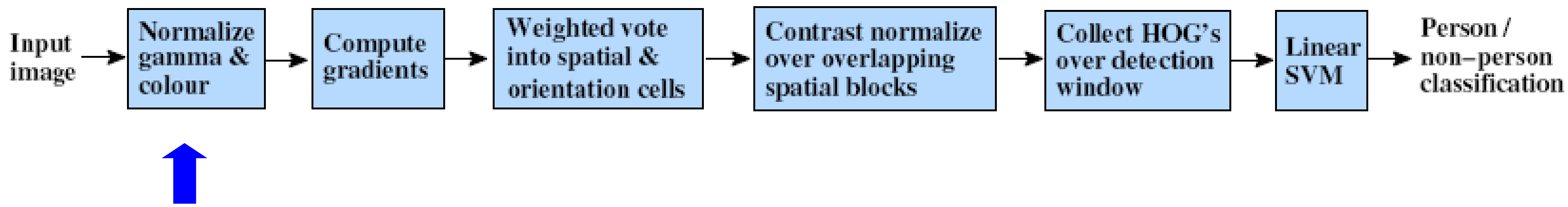
Code available:
http://pascal.inrialpes.fr/soft/olt/

Dalal & Triggs, CVPR 2005

Slide credit: Dalal, Triggs, P. Barnum



Slide credit: Dalal, Triggs, P. Barnum



- Tested with
  - RGB
  - LAB
  - Grayscale
- Gamma Normalization and Compression
  - Square root
  - Log

Slide credit: Dalal, Triggs, P. Barnum

Input image → Normalize gamma & colour → Compute gradients → Weighted vote into spatial & orientation cells → Contrast normalize over overlapping spatial blocks → Collect HOG's over detection window → Linear SVM → Person / non–person classification

-1	0	1

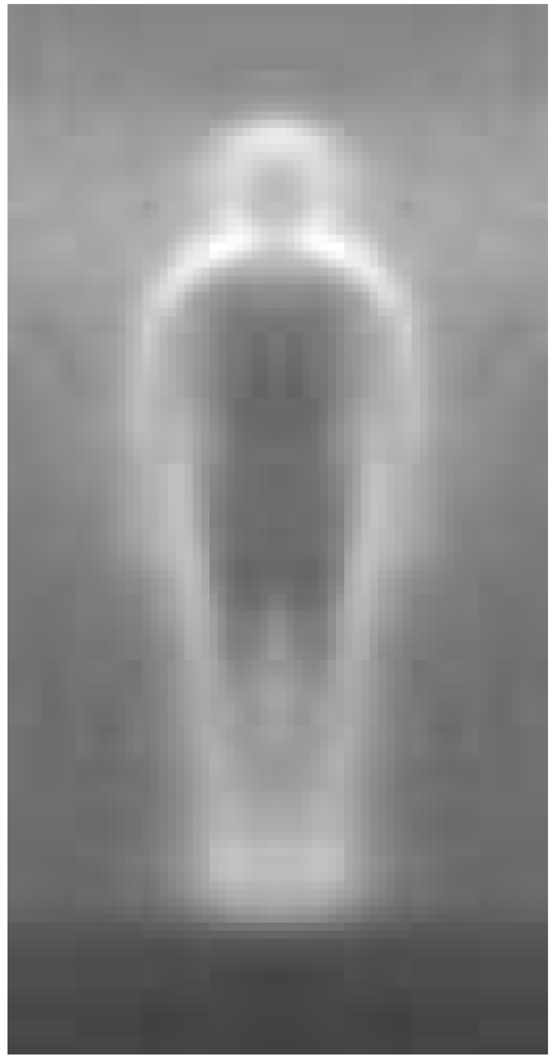
centered

-1	1

uncentered

1	-8	0	8	-1

cubic-corrected



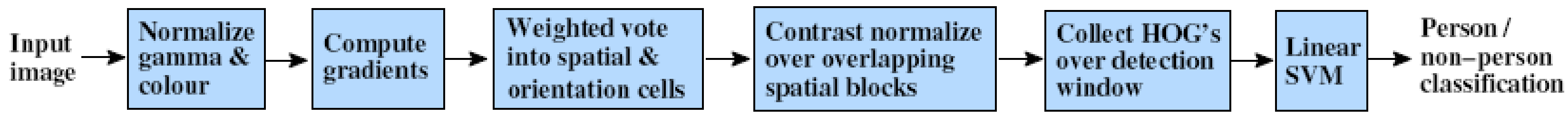
0	1
-1	0

diagonal

-1	0	1
-2	0	2
-1	0	1

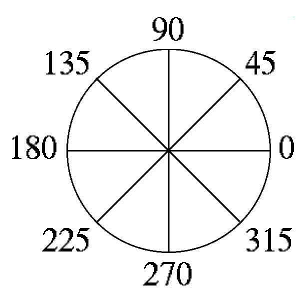
Sobel

Slide credit: Dalal, Triggs, P. Barnum

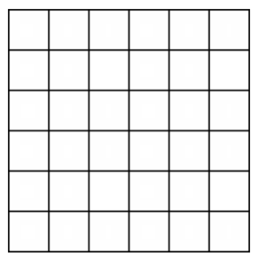


- Histogram of gradient orientations

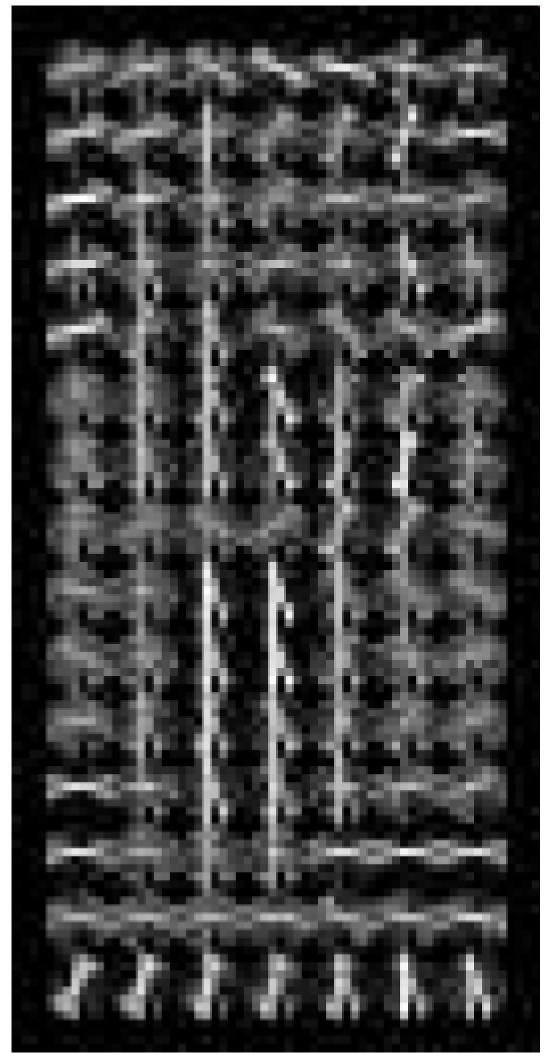
  -Orientation



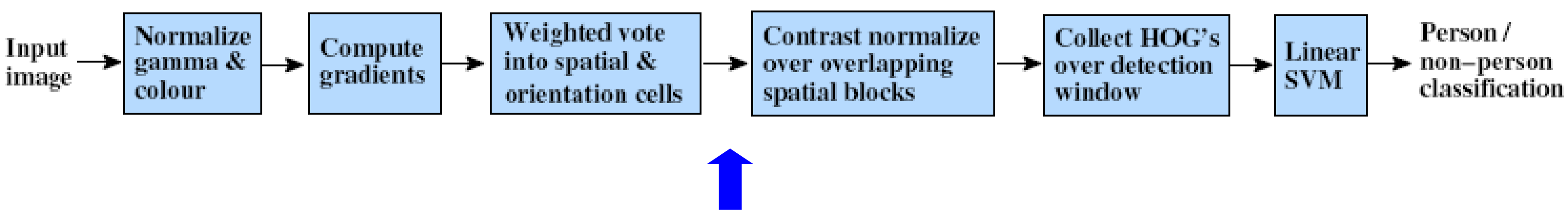
  -Position

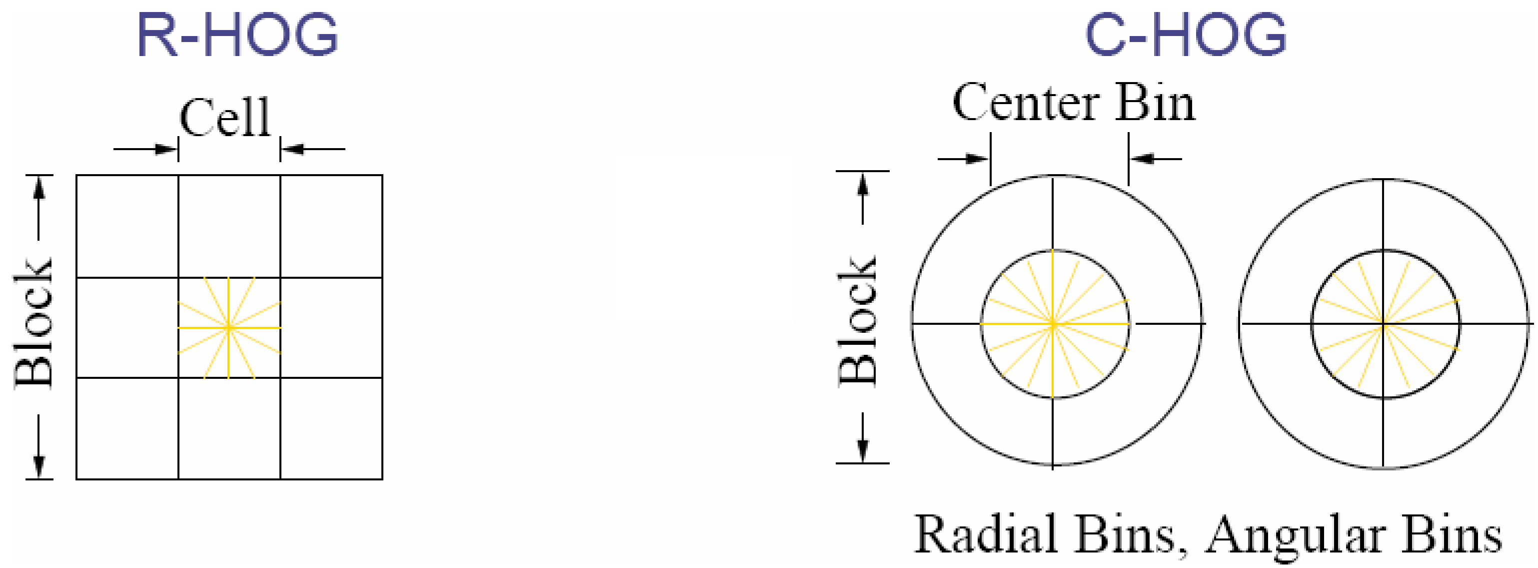


  – Weighted by magnitude

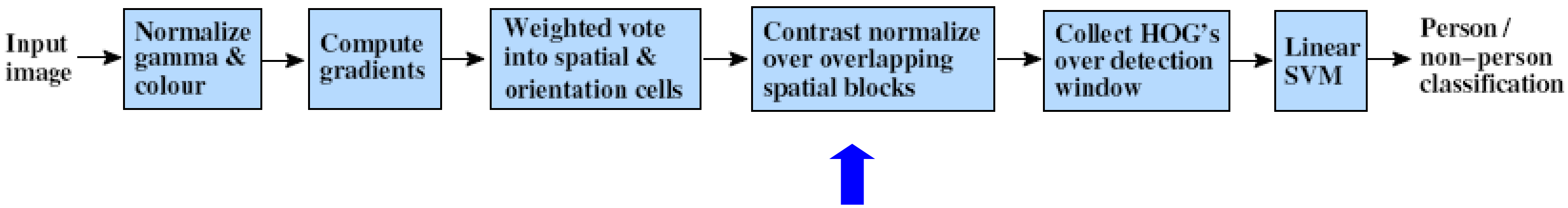


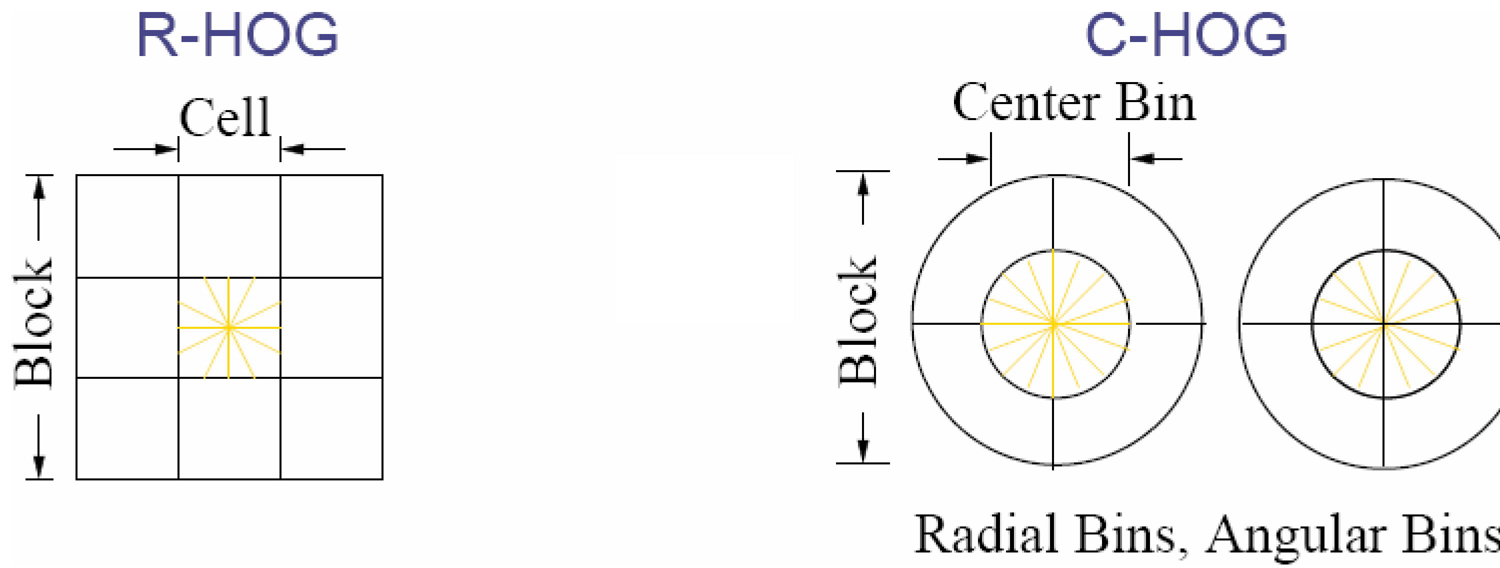
Slide credit: Dalal, Triggs, P. Barnum





Slide credit: Dalal, Triggs, P. Barnum





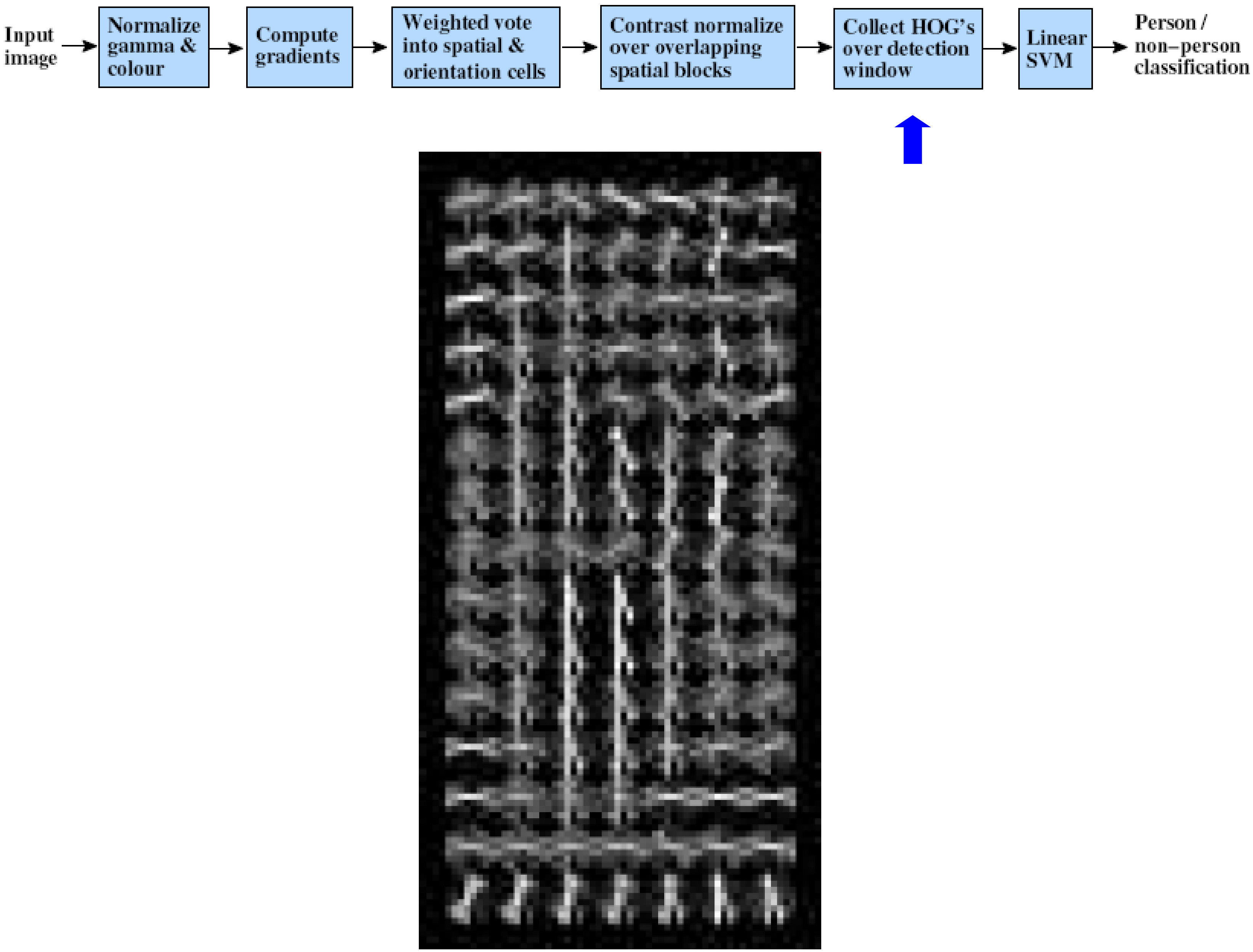
$L1-norm: v \longrightarrow v/(||v||_1+\epsilon)$

$L1-sqrt: v \longrightarrow \sqrt{v/(||v||_1+\epsilon)}$

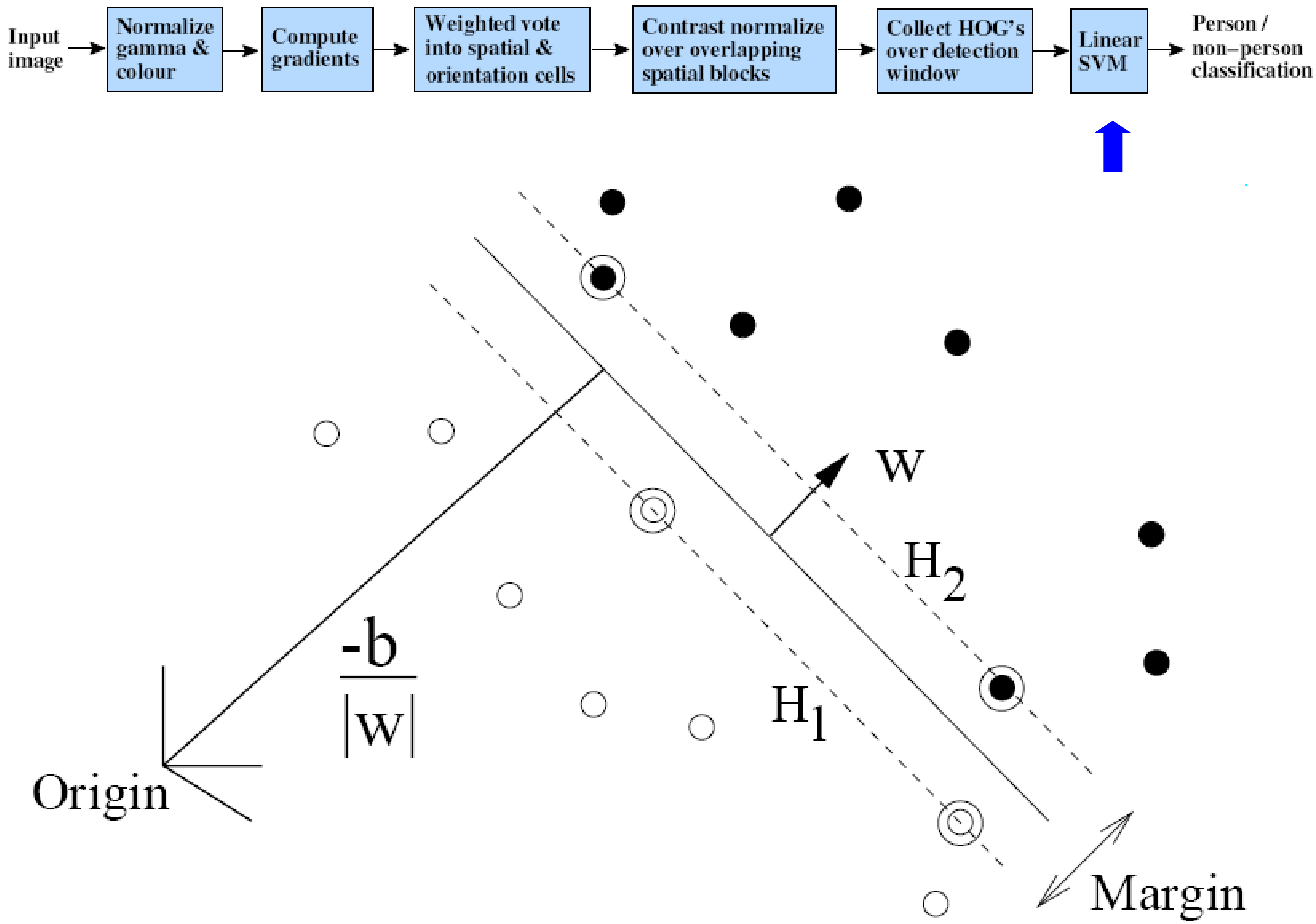
$L2-norm: v \longrightarrow v/\sqrt{||v||_2^2+\epsilon^2}$

$L2-hys$: L2-norm, plus clipping at .2 and renomalizing

Slide credit: Dalal, Triggs, P. Barnum

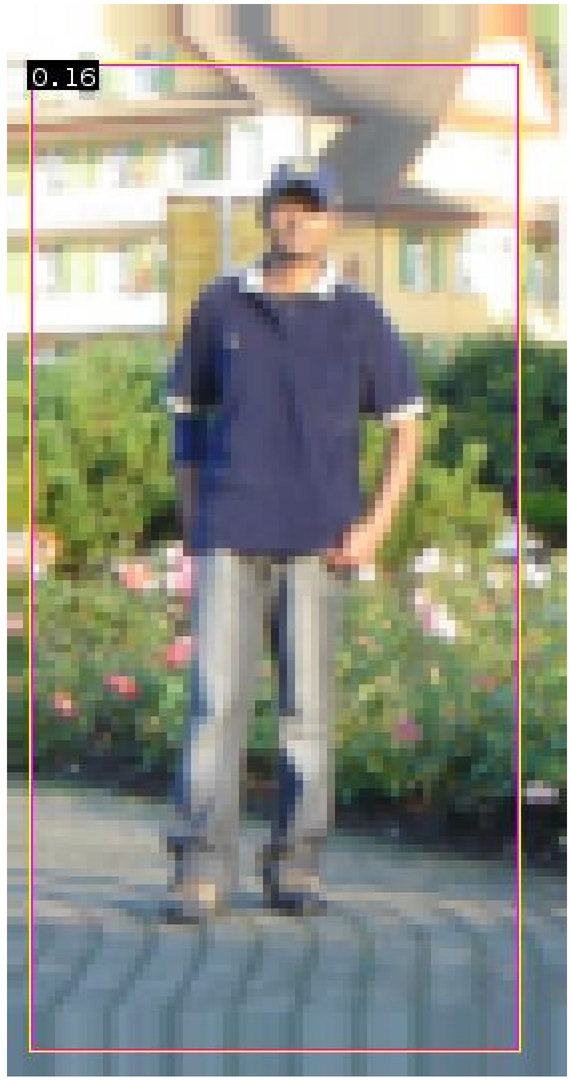


Slide credit: Dalal, Triggs, P. Barnum



Slide credit: Dalal, Triggs, P. Barnum

Input image → Normalize gamma & colour → Compute gradients → Weighted vote into spatial & orientation cells → Contrast normalize over overlapping spatial blocks → Collect HOG's over detection window → Linear SVM → Person / non–person classification

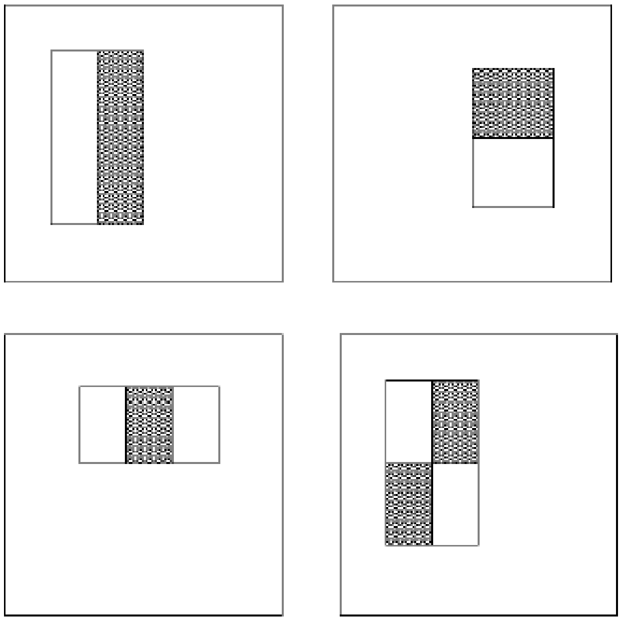


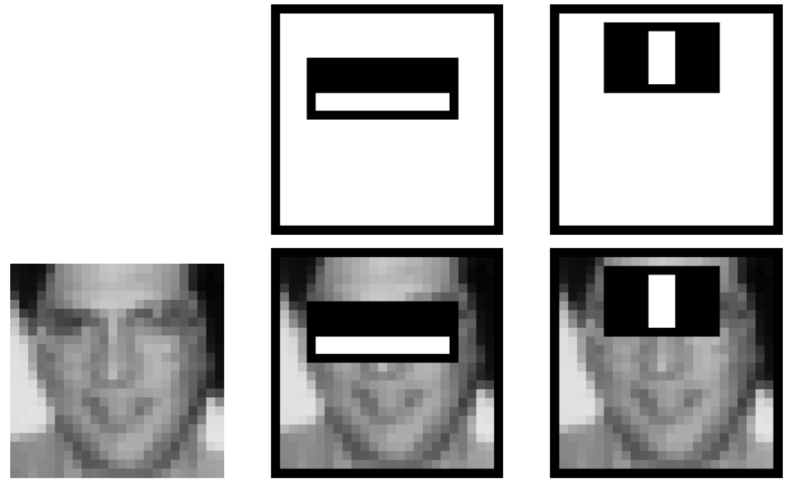
Slide credit: Dalal, Triggs, P. Barnum

Slide credit: Dalal, Triggs, P. Barnum

Boosted Face Detection with Gradient Features

Gradient-based representations: Rectangular features





Compute differences between sums of pixels in rectangles

Captures contrast in adjacent spatial regions, efficient to compute

Each feature parameterized by scale, position, type.

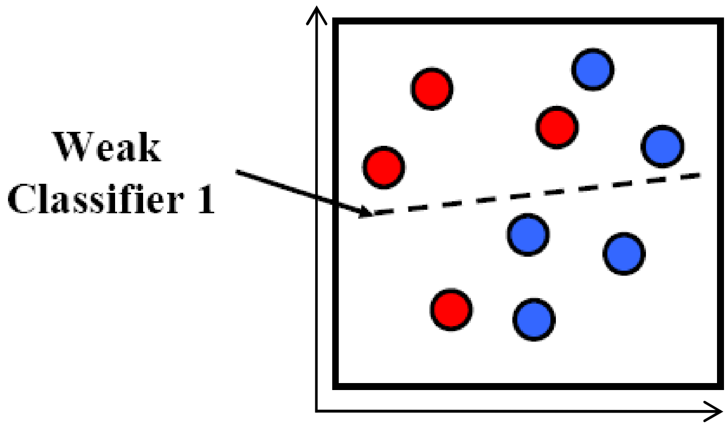
Viola & Jones, CVPR 2001

Boosting

- Build a strong classifier by combining number of “weak classifiers”, which need only be better than chance
- Sequential learning process: at each iteration, add a weak classifier
- Flexible to choice of weak learner
  - including fast simple classifiers that alone may be inaccurate

- We’ll look at Freund & Schapire’s AdaBoost algorithm
  - Easy to implement
  - Base learning algorithm for Viola-Jones face detector

AdaBoost: Intuition



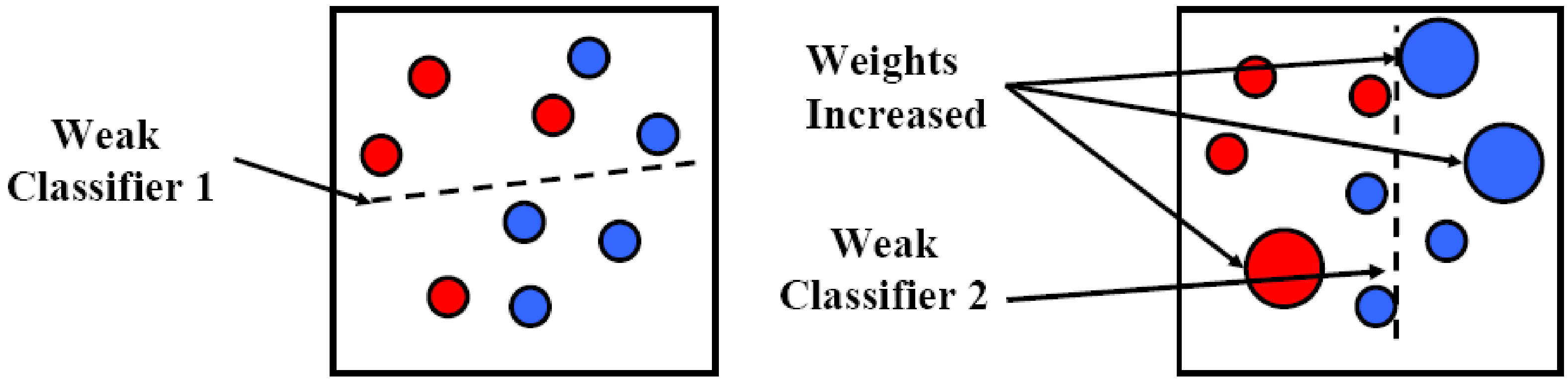
Consider a 2-d feature space with positive and negative examples.

Each weak classifier splits the training examples with at least 50% accuracy.

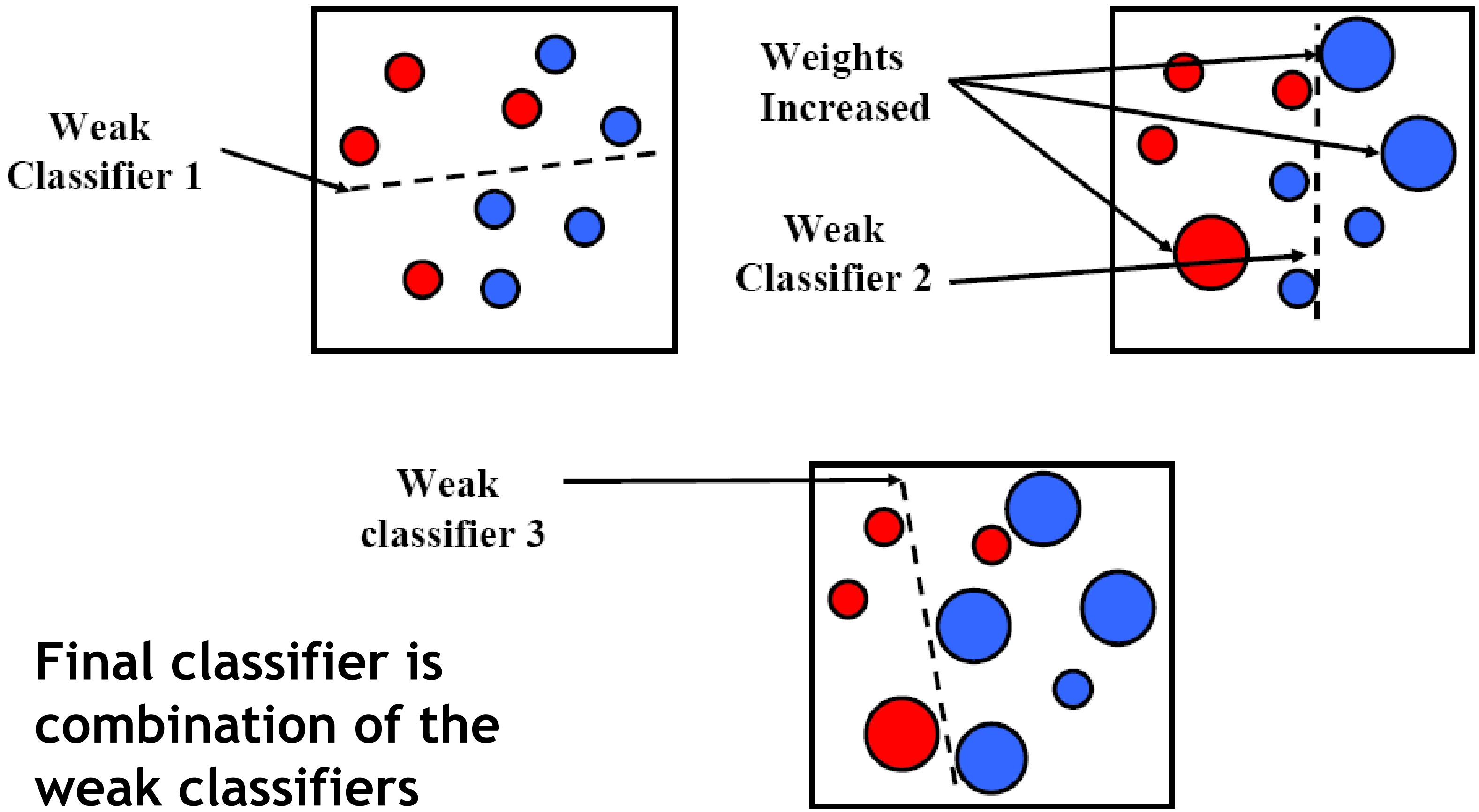
Examples misclassified by a previous weak learner are given more emphasis at future rounds.

Figure adapted from Freund and Schapire

AdaBoost: Intuition



AdaBoost: Intuition

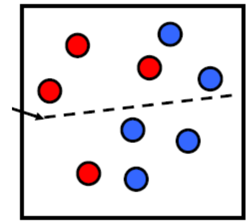


Final classifier is combination of the weak classifiers

AdaBoost Algorithm

- Given example images $(x_1, y_1), \ldots, (x_n, y_n)$ where $y_i = 0, 1$ for negative and positive examples respectively.
- Initialize weights $w_{1,i} = \frac{1}{2m}, \frac{1}{2l}$ for $y_i = 0, 1$ respectively, where m and l are the number of negatives and positives respectively.
- For $t = 1, \ldots, T$:
  1. Normalize the weights,
  $$w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{j=1}^{n} w_{t,j}}$$
  so that w_t is a probability distribution.
  2. For each feature, j, train a classifier h_j which is restricted to using a single feature. The error is evaluated with respect to w_t, $\epsilon_j = \sum_i w_i \, |h_j(x_i) - y_i|$.
  3. Choose the classifier, h_t, with the lowest error ϵ_t.
  4. Update the weights:
  $$w_{t+1,i} = w_{t,i}\beta_t^{1-e_i}$$
  where $e_i = 0$ if example x_i is classified correctly, $e_i = 1$ otherwise, and $\beta_t = \frac{\epsilon_t}{1-\epsilon_t}$.
- The final strong classifier is:
$$h(x) = \begin{cases} 1 & \sum_{t=1}^{T} \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^{T} \alpha_t \\ 0 & \text{otherwise} \end{cases}$$
where $\alpha_t = \log \frac{1}{\beta_t}$

Start with uniform weights on training examples



$\{x_1, \ldots x_n\}$

For T rounds

Evaluate *weighted* error for each feature, pick best.

Re-weight the examples:
Incorrectly classified -> more weight
Correctly classified -> less weight

Final classifier is combination of the weak ones, weighted according to error they had.

Freund & Schapire 1995

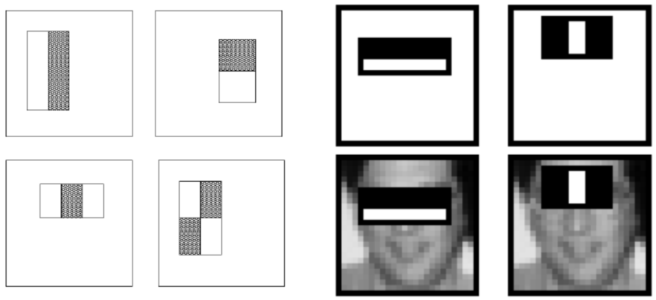
Example: Face detection

- Frontal faces are a good example of a class where global appearance models + a sliding window detection approach fit well:
  - Regular 2D structure
  - Center of face almost shaped like a "patch"/window

- Now we'll take AdaBoost and see how the Viola-Jones face detector works

Feature extraction

"Rectangular" filters

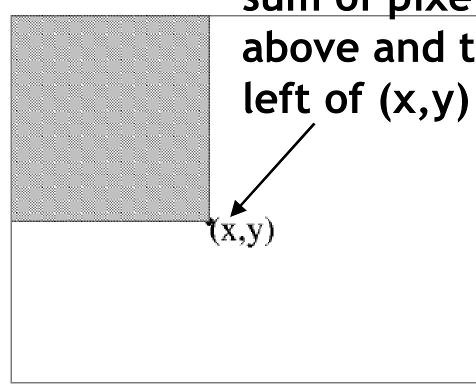


Feature output is difference between adjacent regions

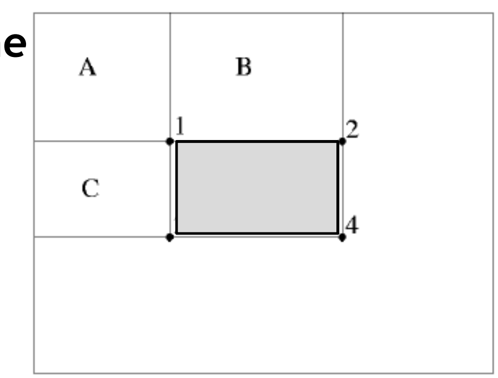
Efficiently computable with integral image: any sum can be computed in constant time

Avoid scaling images → scale features directly for same cost

Value at (x,y) is sum of pixels above and to the left of (x,y)



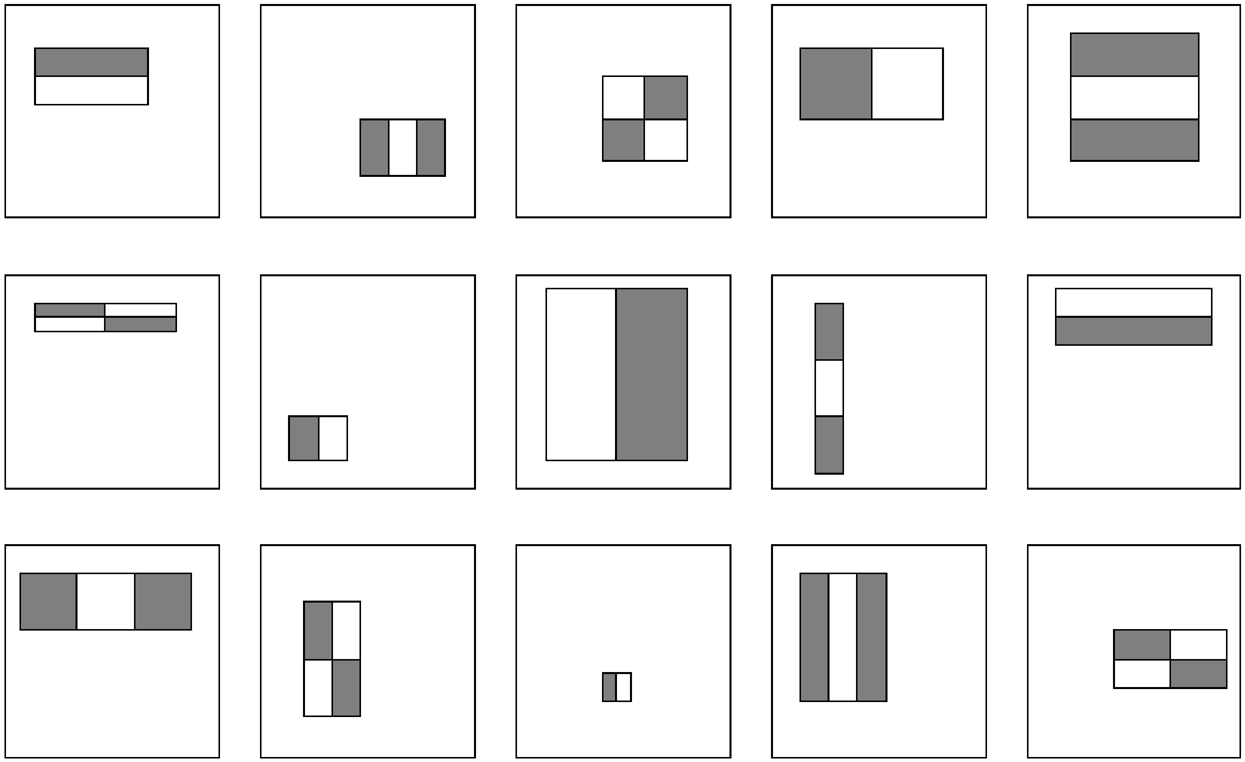
Integral image



$$
\begin{aligned}
D &= 1+4-(2+3) \\
&= A+(A+B+C+D)-(A+C+A+B) \\
&= D
\end{aligned}
$$

Viola & Jones, CVPR 2001

Large library of filters



Considering all possible filter parameters: position, scale, and type:

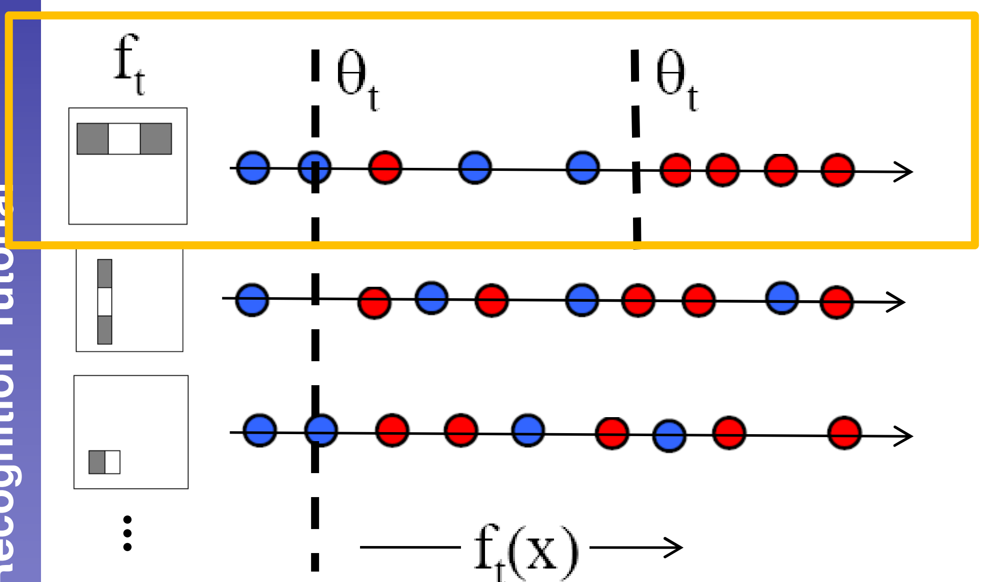
180,000+ possible features associated with each 24 x 24 window

Use AdaBoost both to select the informative features and to form the classifier

Viola & Jones, CVPR 2001

AdaBoost for feature+classifier selection

- **Want to select the single rectangle feature and threshold that best separates positive (faces) and negative (non-faces) training examples, in terms of *weighted* error.**



Outputs of a possible rectangle feature on faces and non-faces.

Resulting weak classifier:

$$h_t(x) = \begin{cases} +1 & \text{if } f_t(x) > \theta_t \\ -1 & \text{otherwise} \end{cases}$$

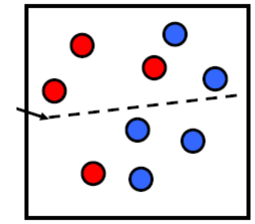
For next round, reweight the examples according to errors, choose another filter/threshold combo.

Viola & Jones, CVPR 2001

AdaBoost Algorithm

- Given example images $(x_1, y_1), \ldots, (x_n, y_n)$ where $y_i = 0, 1$ for negative and positive examples respectively.
- Initialize weights $w_{1,i} = \frac{1}{2m}, \frac{1}{2l}$ for $y_i = 0, 1$ respectively, where m and l are the number of negatives and positives respectively.
- For $t = 1, \ldots, T$:
  1. Normalize the weights,
  $$w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{j=1}^{n} w_{t,j}}$$
  so that w_t is a probability distribution.
  2. For each feature, j, train a classifier h_j which is restricted to using a single feature. The error is evaluated with respect to w_t, $\epsilon_j = \sum_i w_i \, |h_j(x_i) - y_i|$.
  3. Choose the classifier, h_t, with the lowest error ϵ_t.
  4. Update the weights:
  $$w_{t+1,i} = w_{t,i} \beta_t^{1-e_i}$$
  where $e_i = 0$ if example x_i is classified correctly, $e_i = 1$ otherwise, and $\beta_t = \frac{\epsilon_t}{1-\epsilon_t}$.
- The final strong classifier is:
$$h(x) = \begin{cases} 1 & \sum_{t=1}^{T} \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^{T} \alpha_t \\ 0 & \text{otherwise} \end{cases}$$
where $\alpha_t = \log \frac{1}{\beta_t}$

Start with uniform weights on training examples



$\{x_1, \ldots x_n\}$

For T rounds

Evaluate *weighted* error for each feature, pick best.

Re-weight the examples:
Incorrectly classified -> more weight
Correctly classified -> less weight

Final classifier is combination of the weak ones, weighted according to error they had.

Freund & Schapire 1995

AdaBoost for Efficient Feature Selection

- Image Features = Weak Classifiers
- For each round of boosting:
  - Evaluate each rectangle filter on each example
  - Sort examples by filter values
  - Select best threshold for each filter (min error)
    - Sorted list can be quickly scanned for the optimal threshold
  - Select best filter/threshold combination
  - Weight on this feature is a simple function of error rate
  - Reweight examples

Viola and Jones, Robust object detection using a boosted cascade of simple features, CVPR 2001

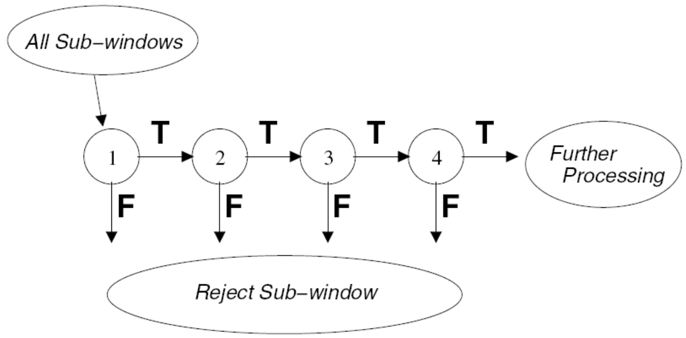
- Even if the filters are fast to compute, each new image has a lot of possible windows to search.

- How to make the detection more efficient?

Cascading classifiers for detection

For efficiency, apply less accurate but faster classifiers first to immediately discard windows that clearly appear to be negative; e.g.,

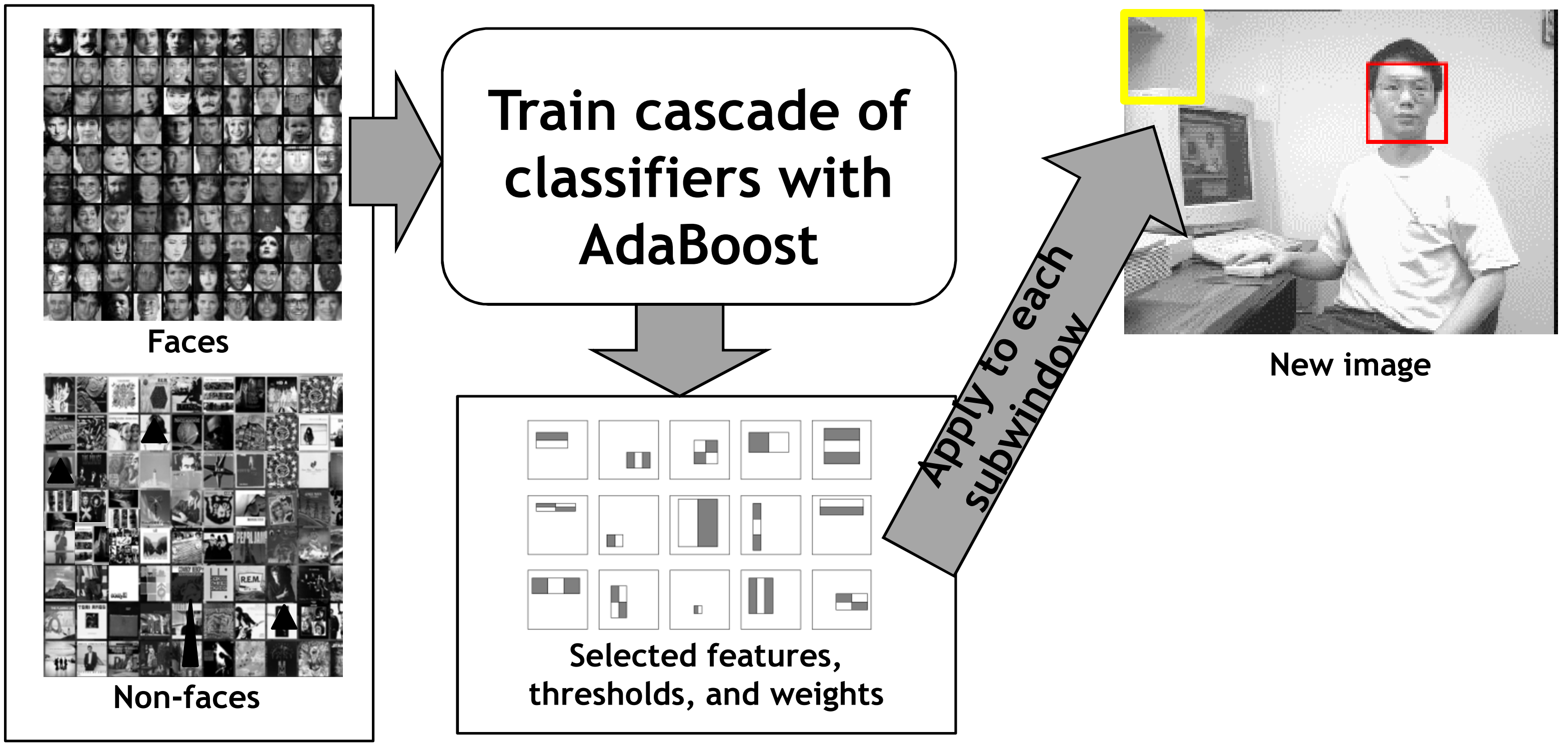
- Filter for promising regions with an initial inexpensive classifier
- Build a chain of classifiers, choosing cheap ones with low false negative rates early in the chain



Fleuret & Geman, IJCV 2001
Rowley et al., PAMI 1998
Viola & Jones, CVPR 2001

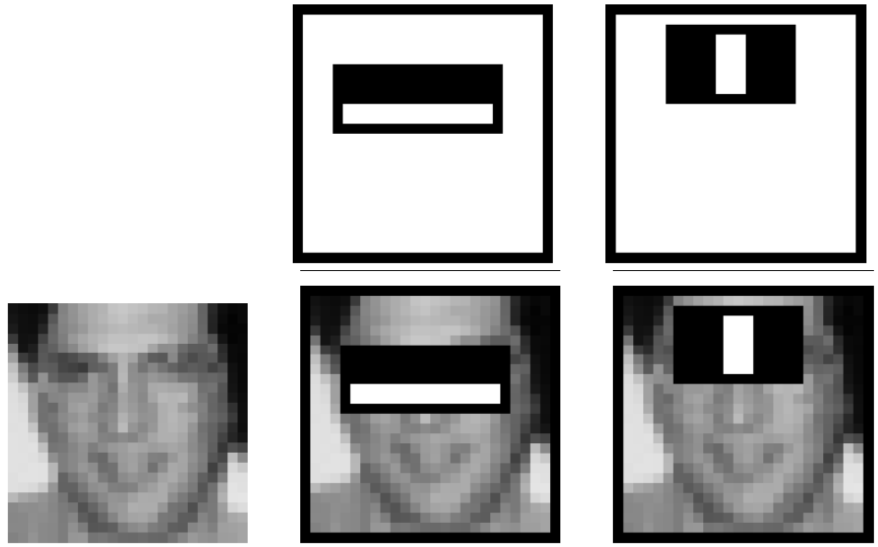
Figure from Viola & Jones CVPR 2001

Viola-Jones Face Detector: Summary



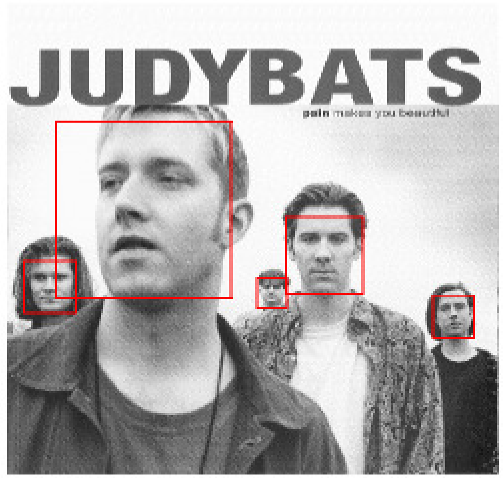
- Train with 5K positives, 350M negatives
- Real-time detector using 38 layer cascade
- 6061 features in final layer
- [Implementation available in OpenCV: http://www.intel.com/technology/computing/opencv/]

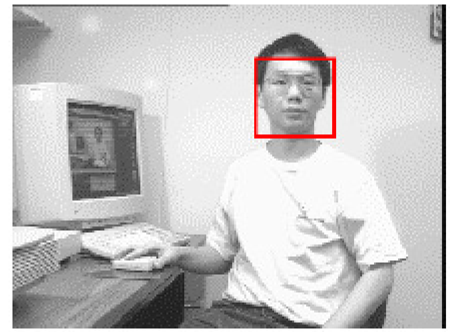
Viola-Jones Face Detector: Results

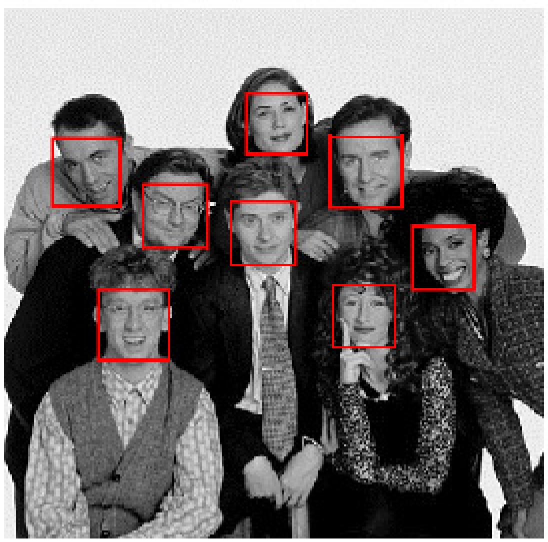


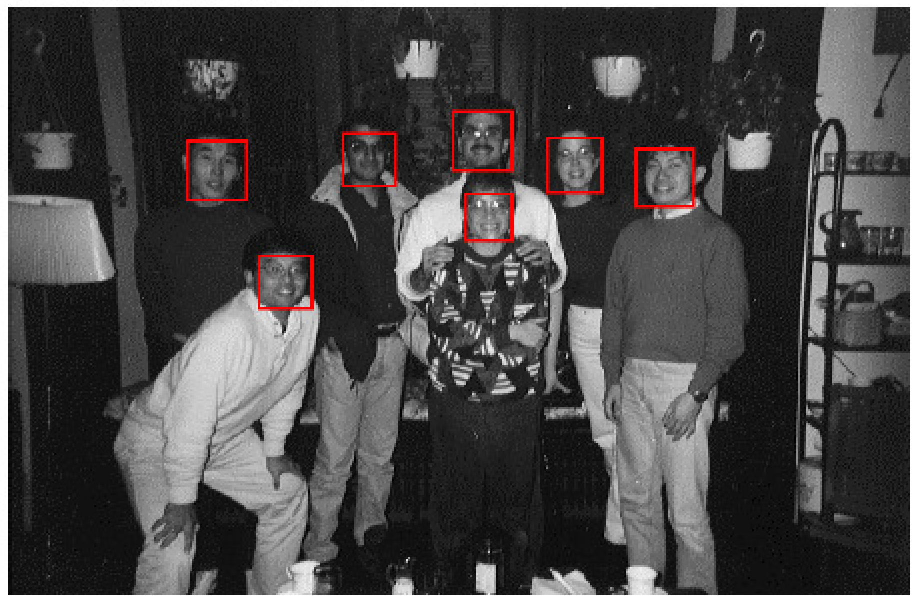
First two features selected

Viola-Jones Face Detector: Results

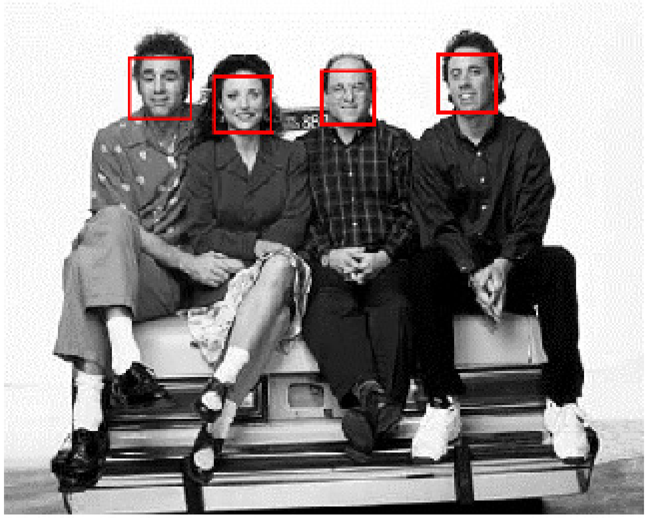


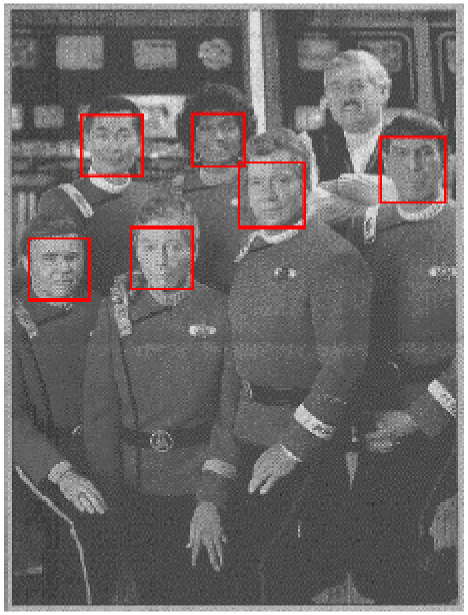


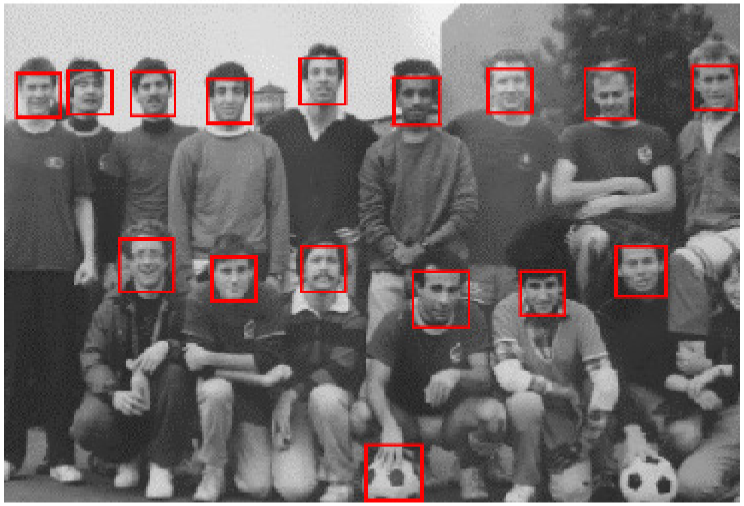


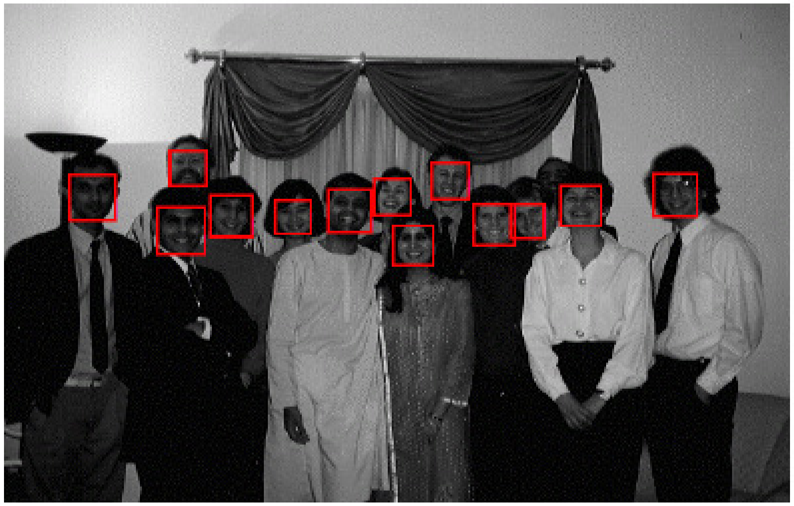


Viola-Jones Face Detector: Results





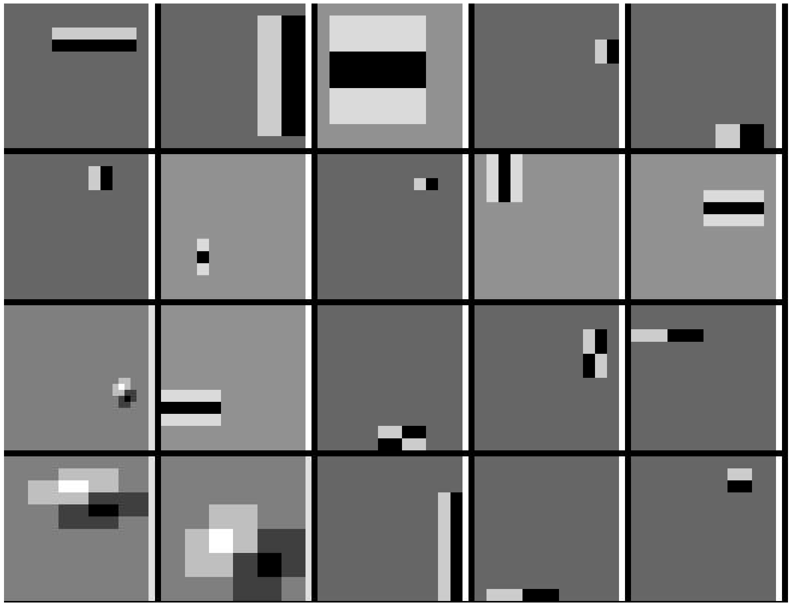




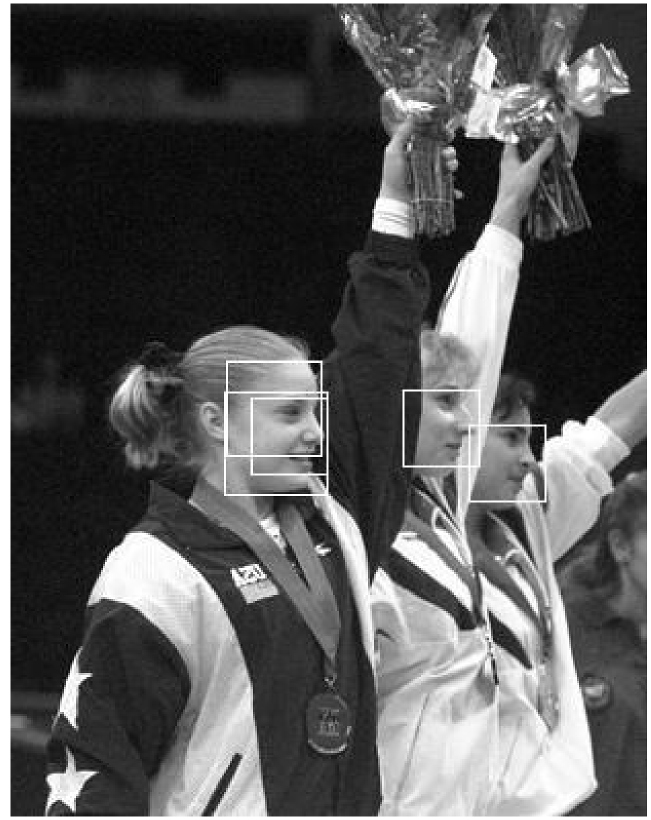
Viola-Jones Face Detector: Results

Detecting profile faces?

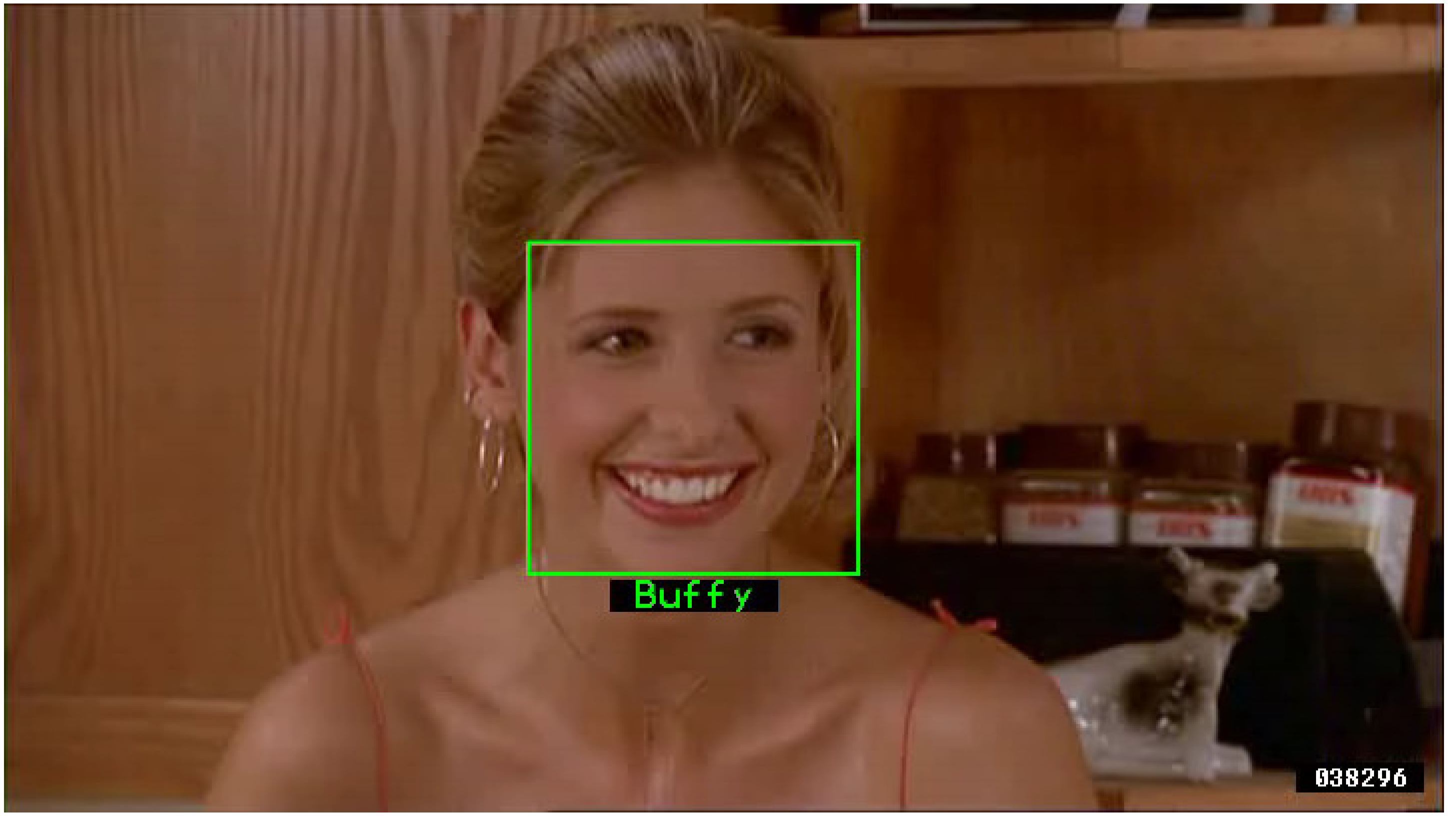
Detecting profile faces requires training separate detector with profile examples.



Viola-Jones Face Detector: Results



Example application



Frontal faces detected and then tracked, character names inferred with alignment of script and subtitles.

Everingham, M., Sivic, J. and Zisserman, A.
"Hello! My name is... Buffy" - Automatic naming of characters in TV video, BMVC 2006.
http://www.robots.ox.ac.uk/~vgg/research/nface/index.html

Example application: faces in photos

All Web People Objects Tags **My Photos**

SEARCH

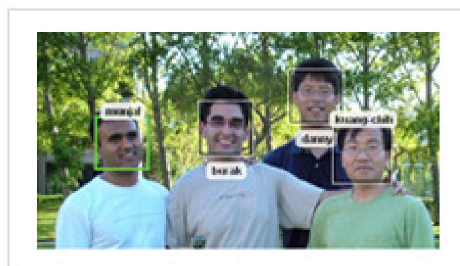
Advanced

Riya Personal Search

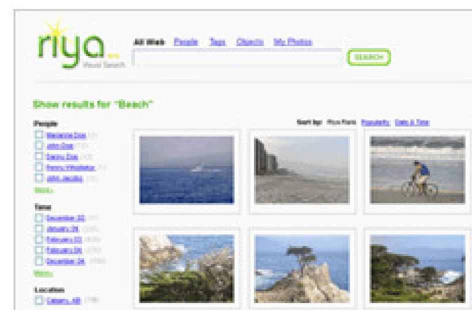
Use our face recognition and text recognition, to search your personal photos

Upload your personal photos (public or privately)

Use our face and text recognition to auto tag your photos



Search & share photos with your friends



Riya's Personal Search lets you upload and search your own photos by name. You can keep them private or make them public and share with all Riya searchers. We allow you to use face and text recognition to search your own photos.

Highlights

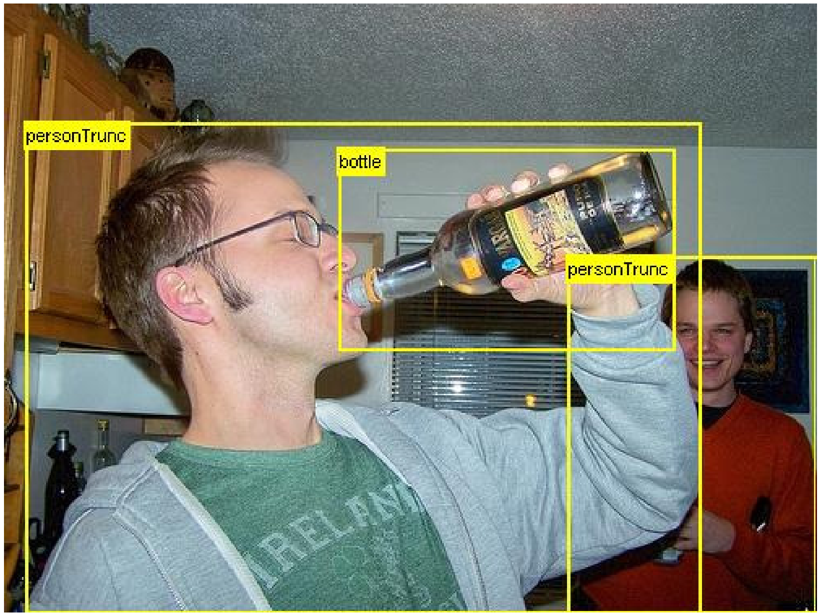
- Sliding window detection and global appearance descriptors:
  - Simple detection protocol to implement
  - Good feature choices critical
  - Past successes for certain classes

Limitations

- High computational complexity
  - For example: 250,000 locations x 30 orientations x 4 scales = 30,000,000 evaluations!
  - If training binary detectors independently, means cost increases linearly with number of classes
- With so many windows, false positive rate better be low

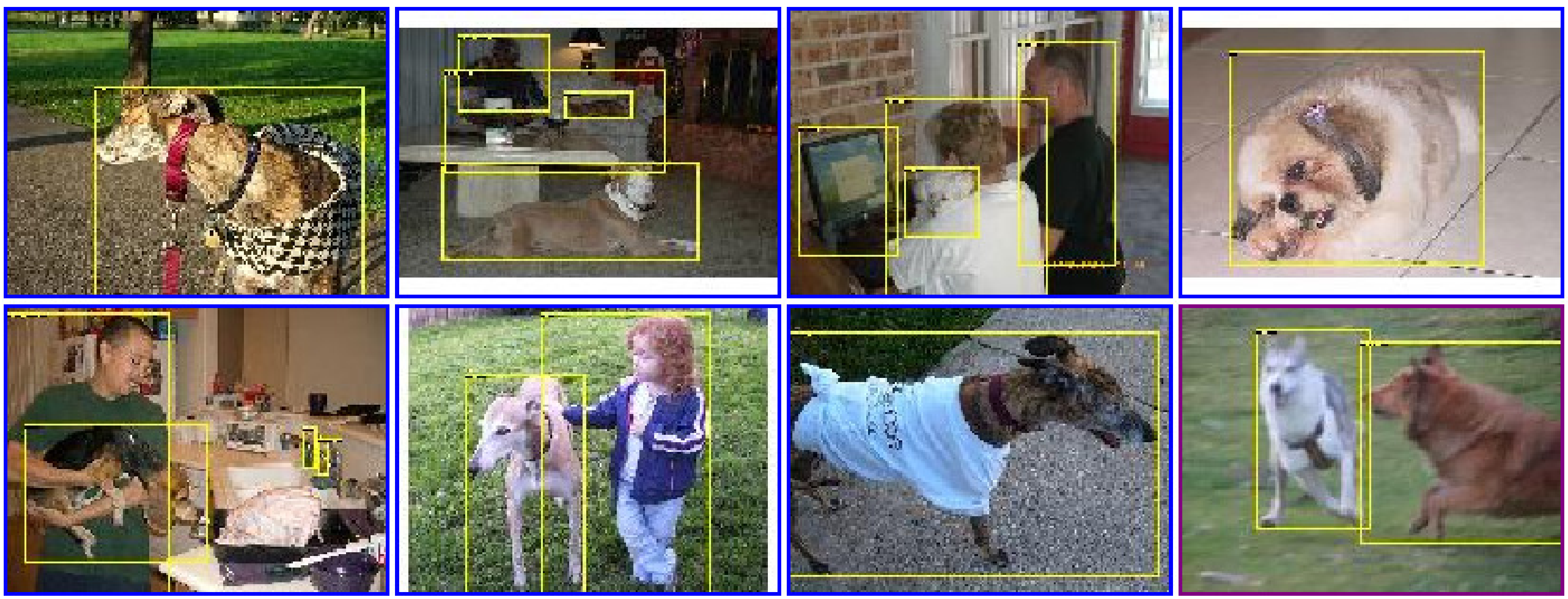
Limitations (continued)

- **Not all objects are "box" shaped**



Limitations (continued)

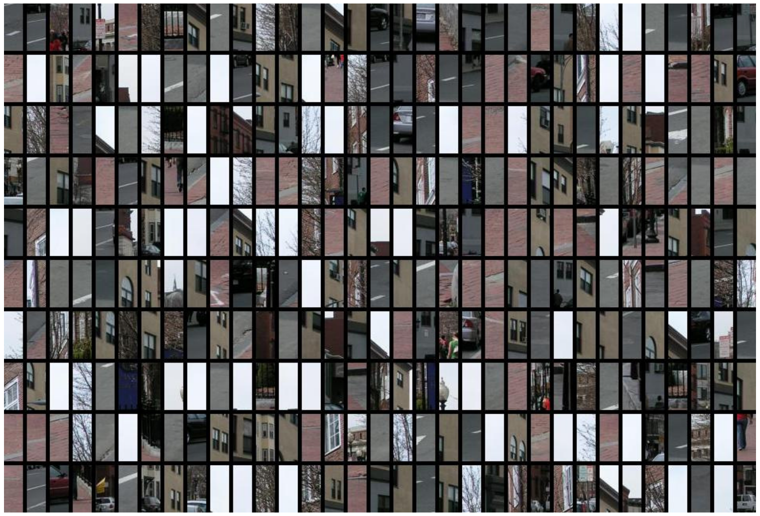
- **Non-rigid, deformable objects not captured well with representations assuming a fixed 2d structure; or must assume fixed viewpoint**
- **Objects with less-regular textures not captured well with holistic appearance-based descriptions**



Limitations (continued)

- If considering windows in isolation, context is lost

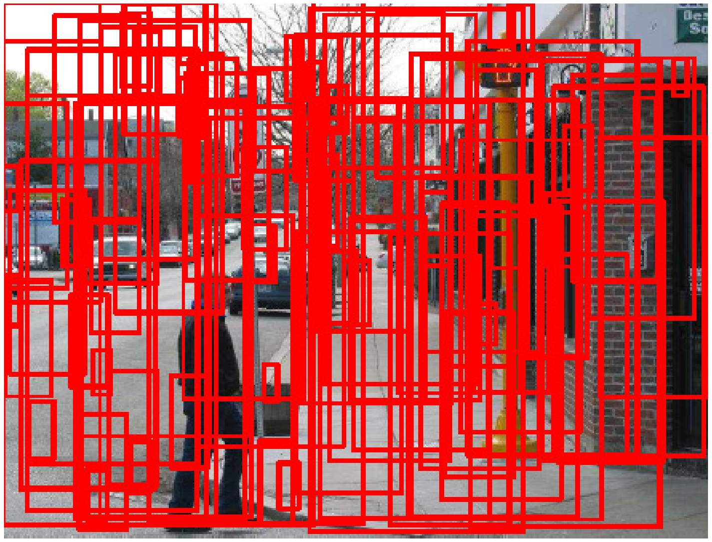
Sliding window



Detector's view

Figure credit: Derek Hoiem

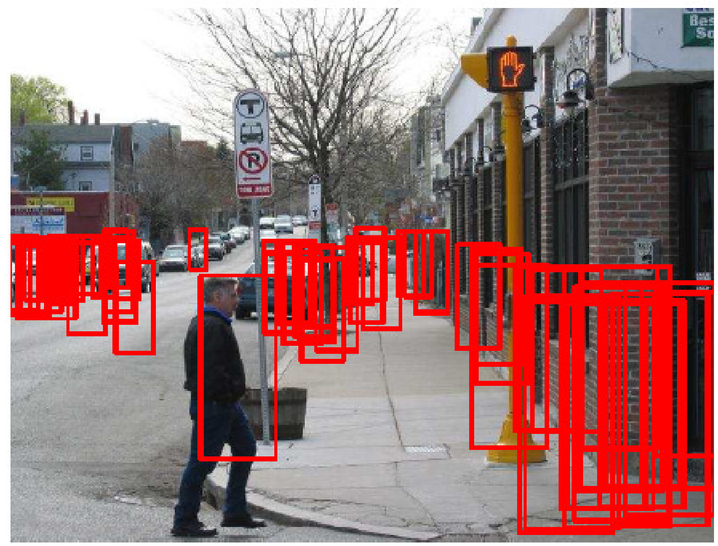
Context can constrain a sliding window search



(b) P(person) = uniform

(d) P(person | geometry)

(f) P(person | viewpoint)



(g) P(person|viewpoint,geometry)

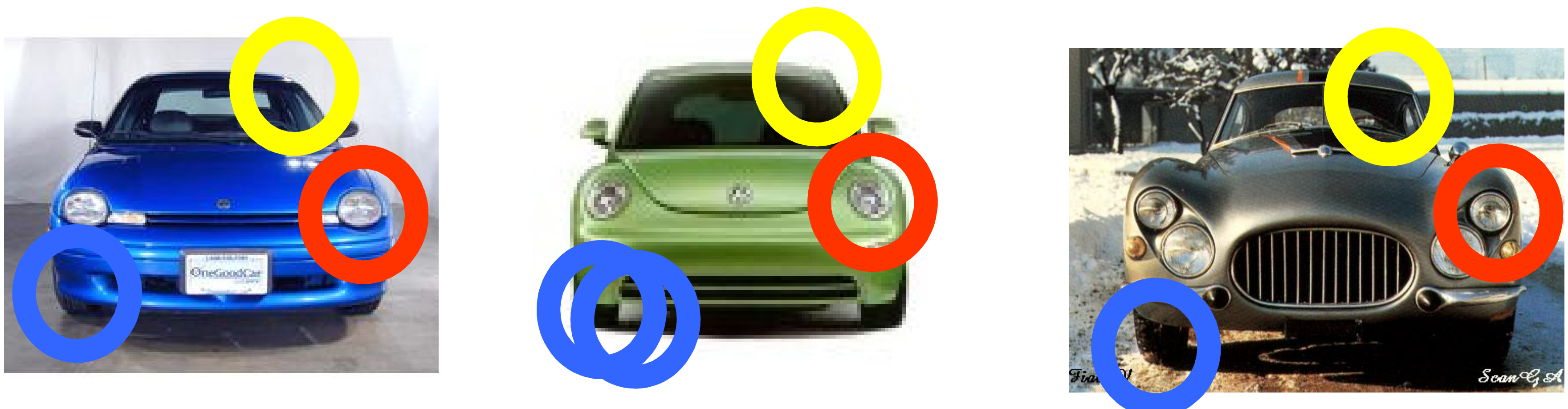
Hoiem, Efros, Herbert, 2006

Limitations (continued)

- In practice, often entails large, cropped training set (expensive)
- Requiring good match to a global appearance description can lead to sensitivity to partial occlusions

Image credit: Adam, Rivlin, & Shimshoni

Models based on local features will alleviate some of these limitations…

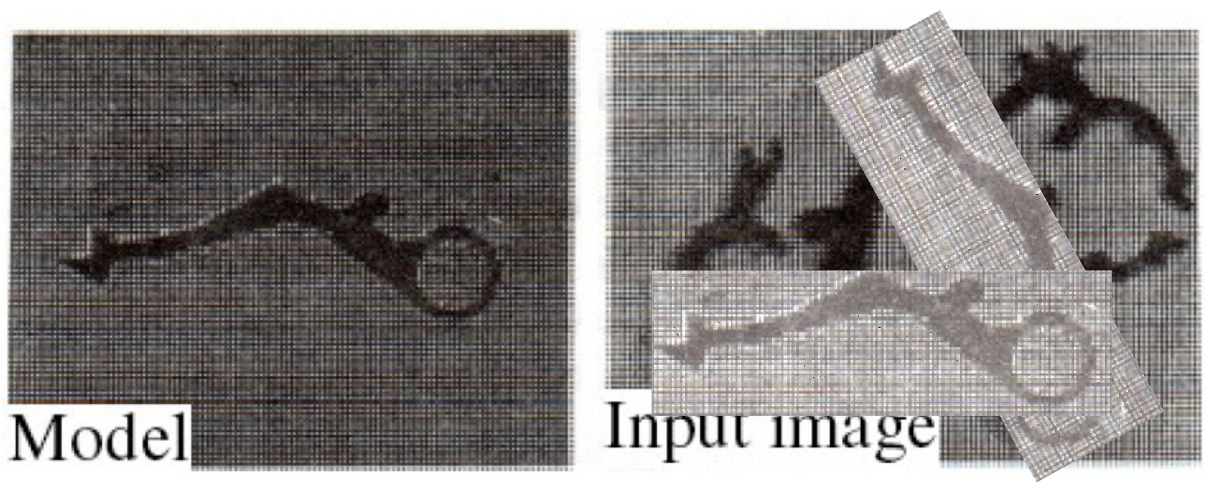


K. Grauman, B. Leibe

Local-feature Alignment

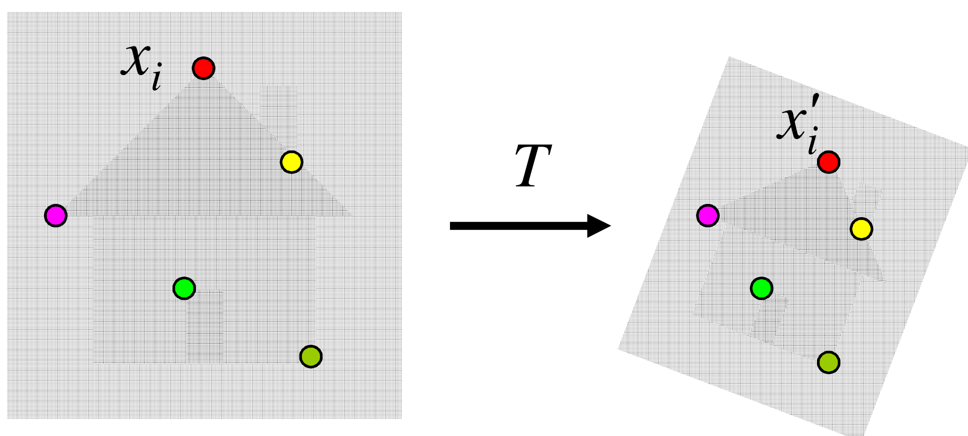
Hypothesize and test: main idea

- Given model of object
- New image: hypothesize object identity and pose
- Render object in camera
- Compare rendering to actual image: if close, good hypothesis.



Recall: Alignment

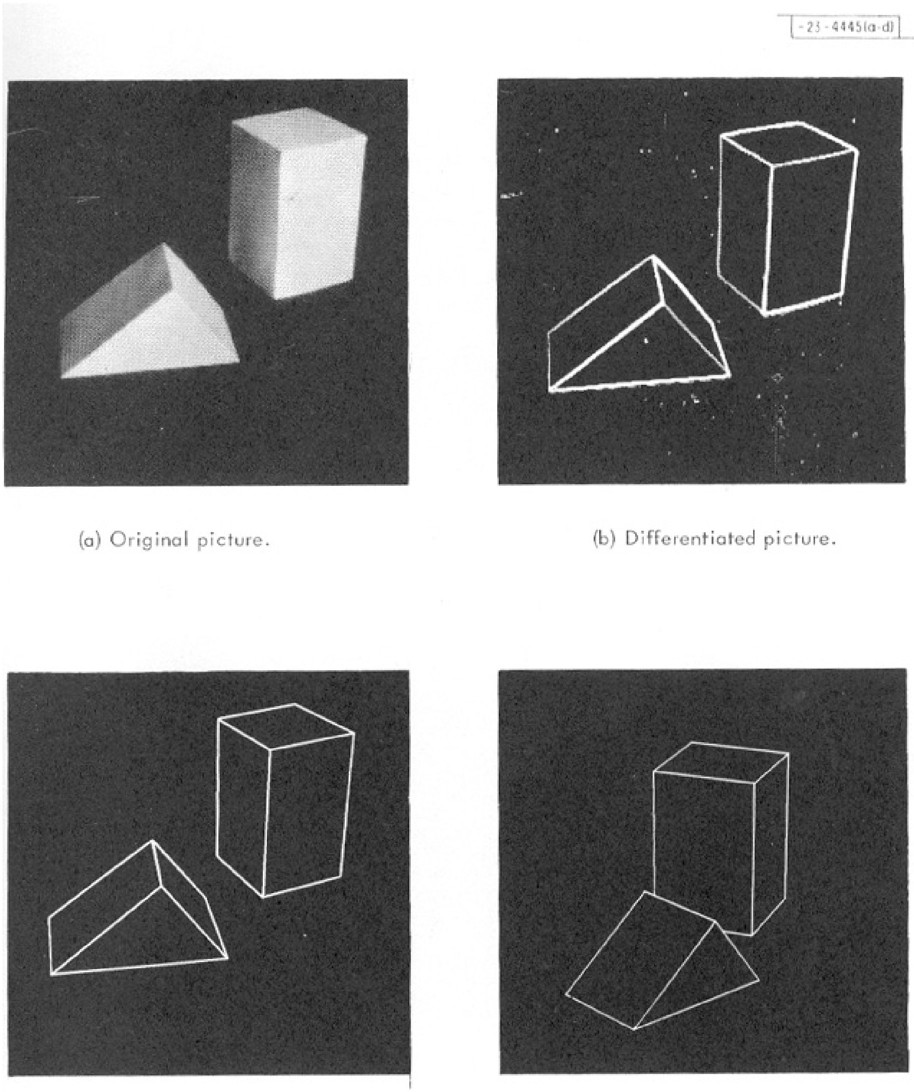
- Alignment: fitting a model to a transformation between pairs of features (*matches*) in two images



Find transformation T that minimizes

$$\sum_i \text{residual}(T(x_i), x_i')$$

Alignment-based

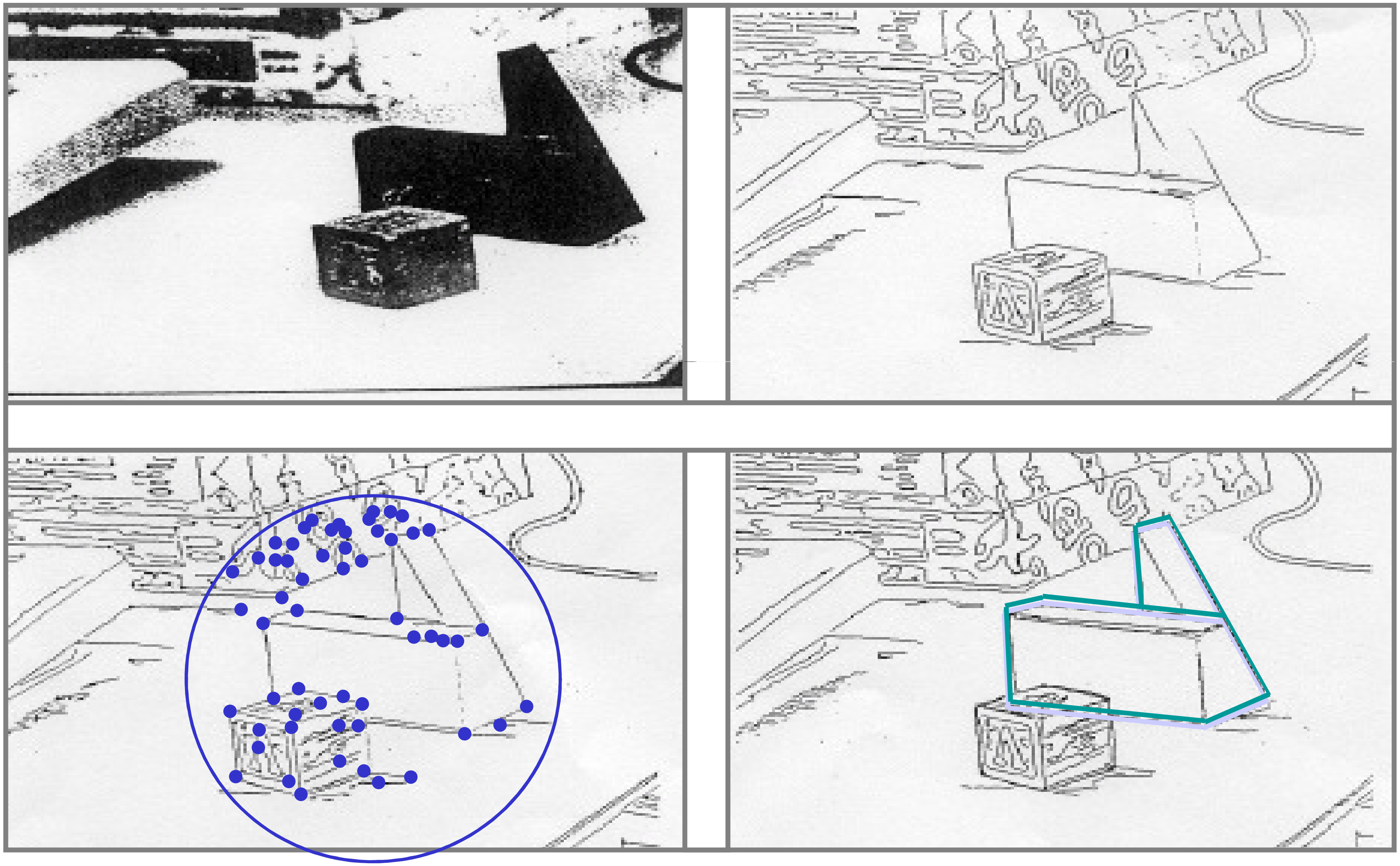


(a) Original picture. (b) Differentiated picture.

(c) Line drawing. (d) Rotated view.

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

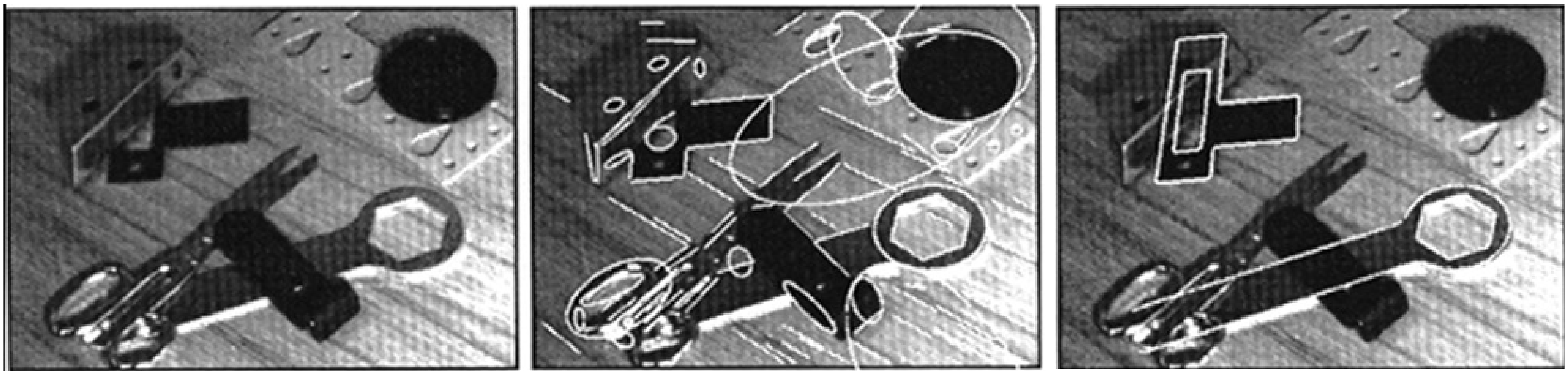
Alignment-based

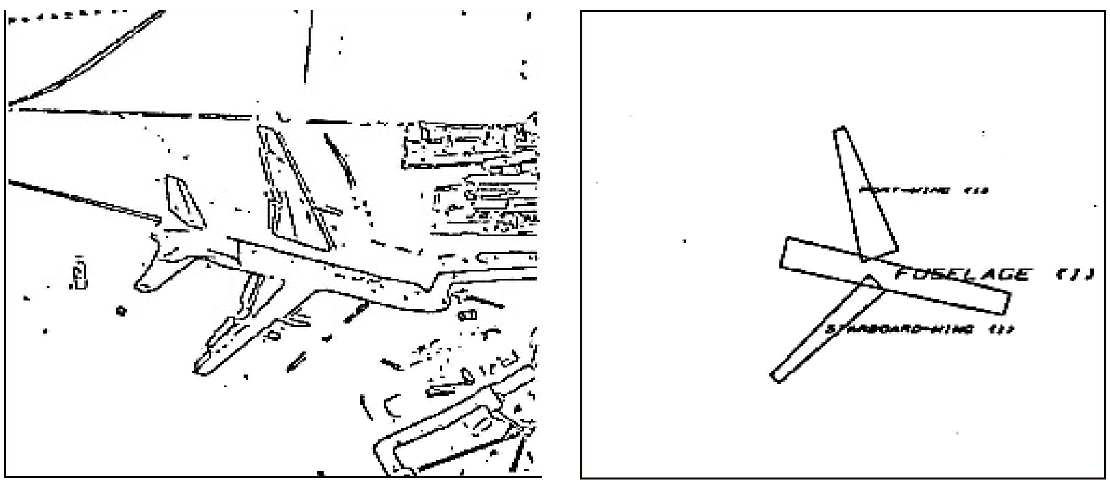


Huttenlocher & Ullman (1987)

Source: Lana Lazebnik

Alignment-based





ACRONYM (Brooks and Binford, 1981)

How to form a hypothesis?

Given a particular model object, we can estimate the *correspondences* between image and model features

Use correspondence to estimate model pose relative to object coordinate frame

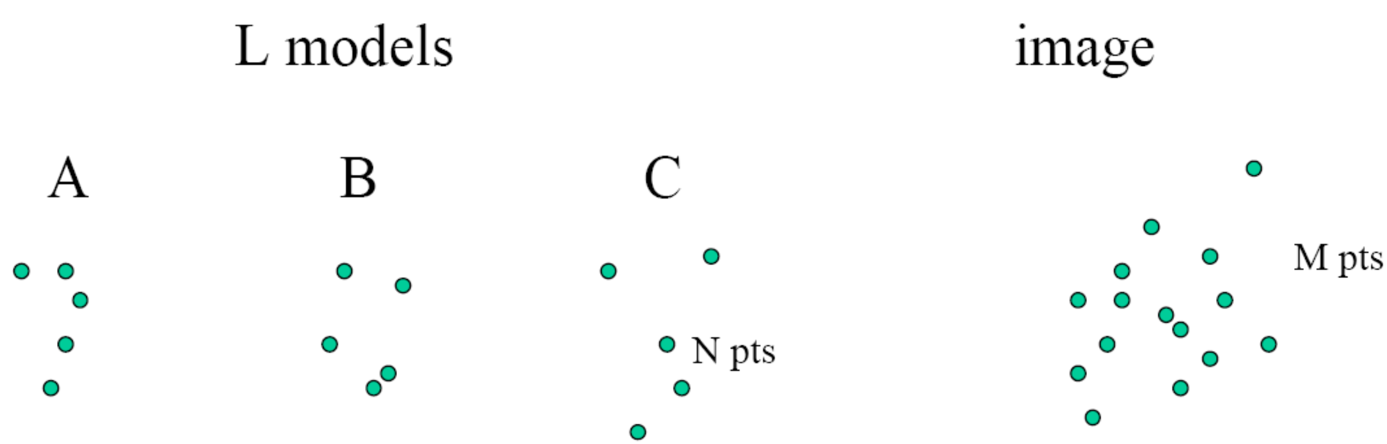
Generating hypotheses

We want a good correspondence between model features and image features.

– Brute force?

Brute force hypothesis generation

- For every possible model, try every possible subset of image points as matches for that model's points.
- Say we have L objects with N features, M features in image



Generating hypotheses

We want a good correspondence between model features and image features.

- Brute force?

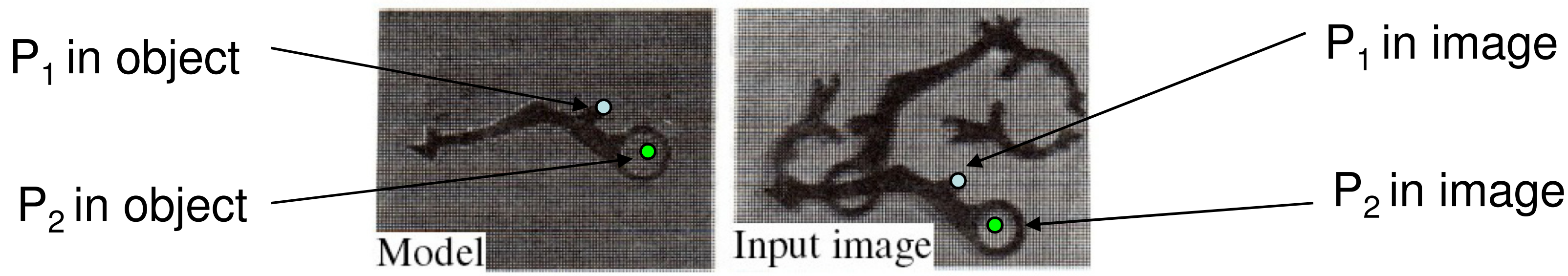
- **Pose consistency**, alignment: use subsets of features to estimate larger correspondence
- **Voting**, pose clustering

Pose consistency / alignment

- Key idea:
  - If we find good correspondences for a small set of features, it is easy to obtain correspondences for a much larger set.
- Strategy:
  - Generate hypotheses using small numbers of correspondences
  - Backproject: transform *all* model features to image features
  - Verify

Example: 2d affine mappings

- Say camera is looking down perpendicularly on planar surface



- We have two coordinate systems (object and image), and they are related by some affine mapping (rotation, scale, translation, shear).

Alignment: verification

- Given the back-projected model in the image:
  - Check if image edges coincide with predicted model edges
  - May be more robust if also require edges to have the same orientation
  - Consider texture in corresponding regions

- Possible issues?

Alignment: verification

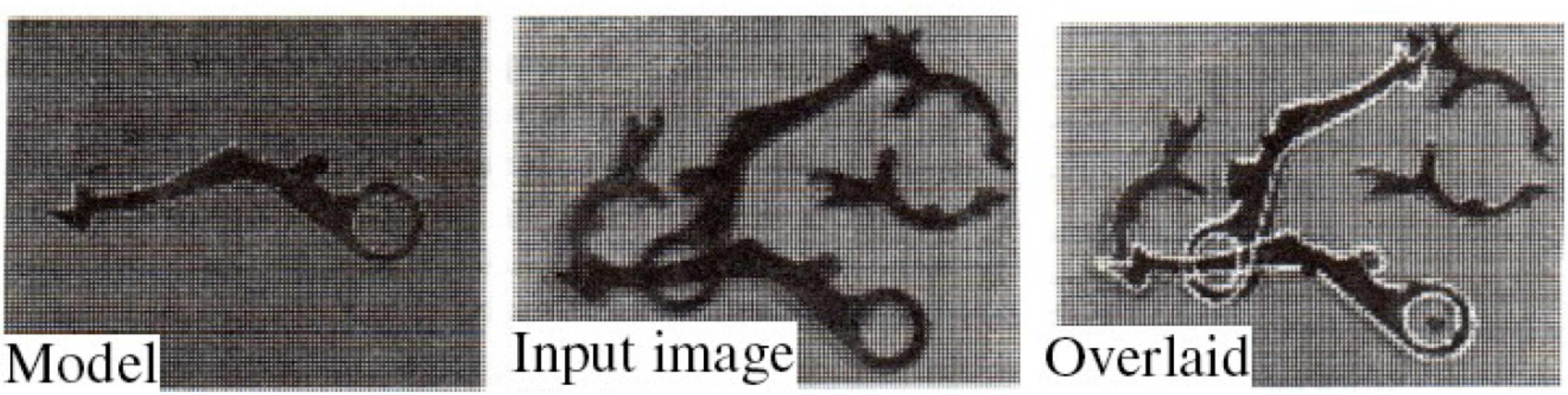
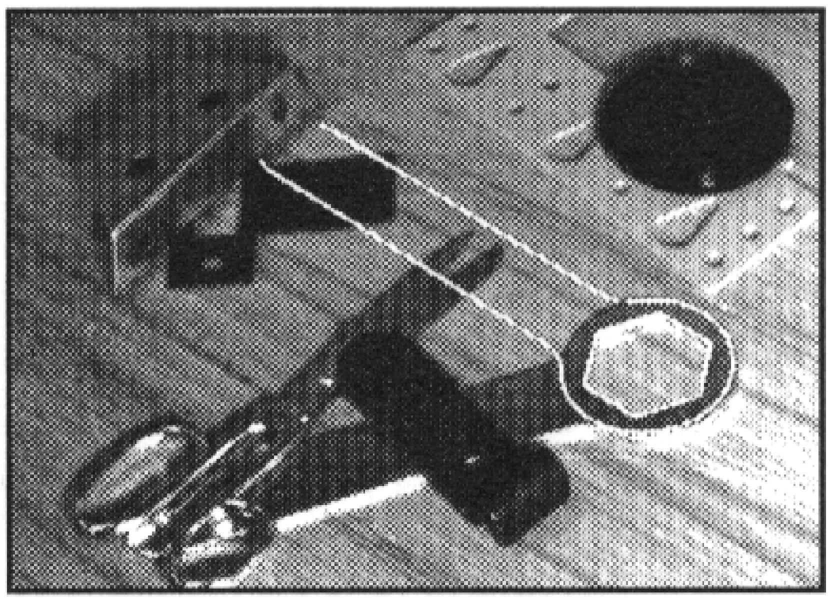


Figure from "Object recognition using alignment," D.P. Huttenlocher and S. Ullman, Proc. Int. Conf. Computer Vision, 1986, copyright IEEE, 1986

Alignment: verification



Issue with hypothesis & test approach

- May have false matches
  - We want *reliable* features to form the matches
    - **Local invariant features** useful to find matches, and to verify hypothesis

      (SIFT, etc.)

- May be too many hypotheses to consider
  - We want to look at the *most likely* hypotheses first
    - **Pose clustering (voting):** Narrow down number of hypotheses to verify by letting features *vote* on model parameters.

Pose clustering (voting)

- Narrow down the number of hypotheses to verify: identify those model poses that a lot of features agree on.
  - Use each group’s correspondence to estimate pose
  - Vote for that object pose in accumulator array (one array per object if we have multiple models)

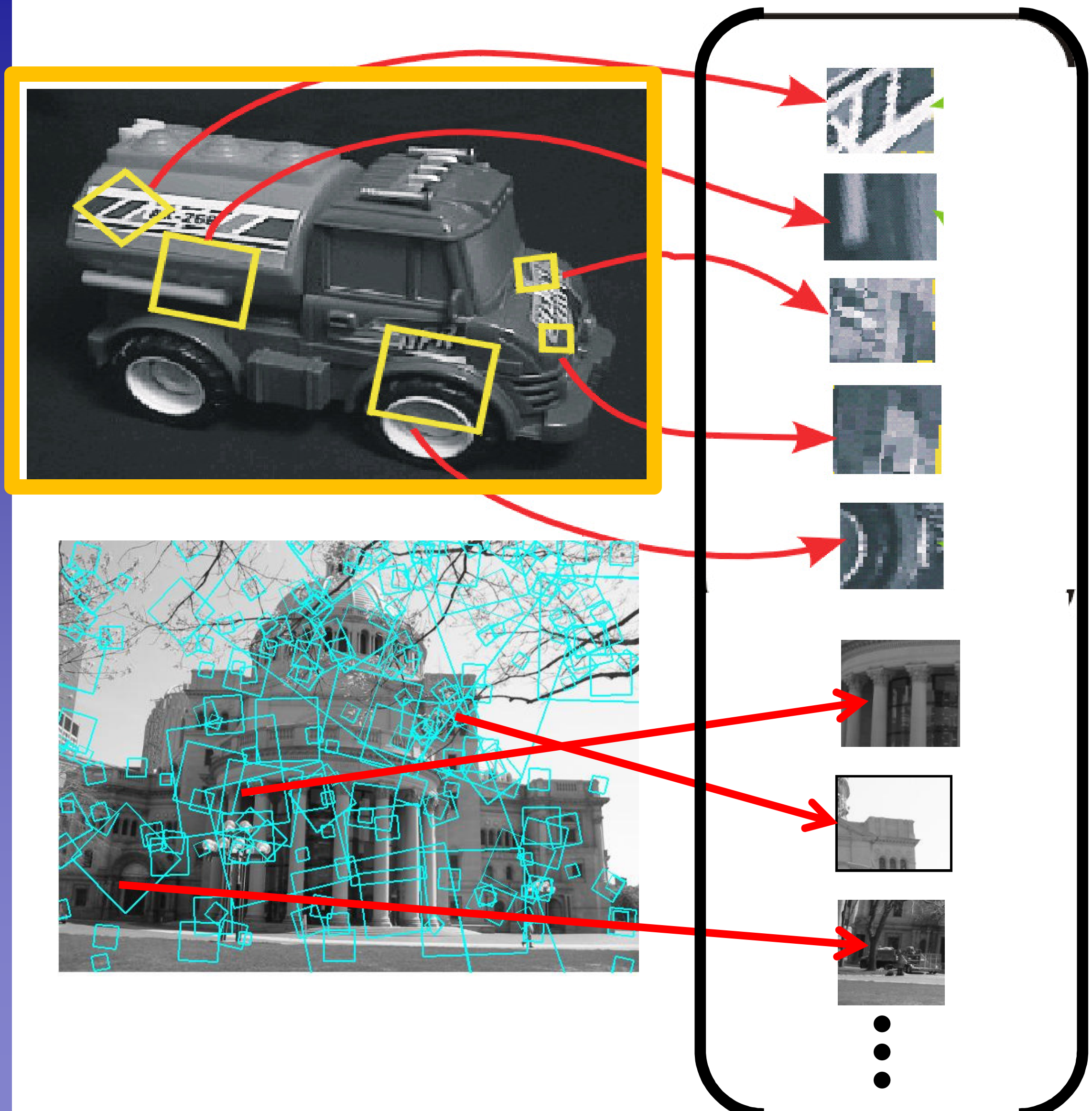
- Local invariant features can give more reliable matches and means of verification

Pose clustering and verification with SIFT [Lowe]

To detect **instances** of objects from a model base:

1) Index descriptors (distinctive features narrow possible matches)

Indexing local features



Pose clustering and verification with SIFT [Lowe]

To detect **instances** of objects from a model base:

1) Index descriptors (distinctive features narrow possible matches)

2) Generalized Hough transform to vote for poses (keypoints have record of parameters relative to model coordinate system)

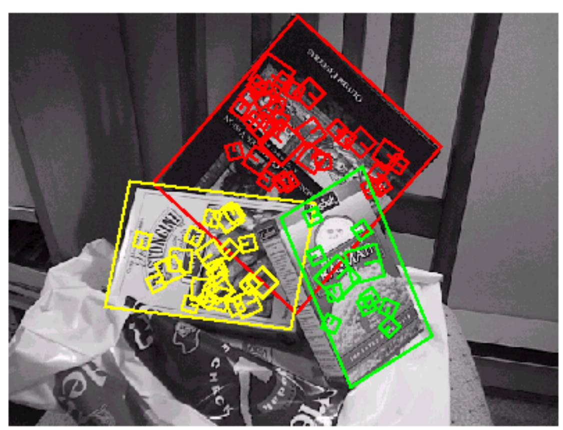
3) Affine fit to check for agreement between model and image features (approximates perspective projection for planar objects)

Planar objects

Model images and their SIFT keypoints

Input image

Model keypoints that were used to recognize, get least squares solution.



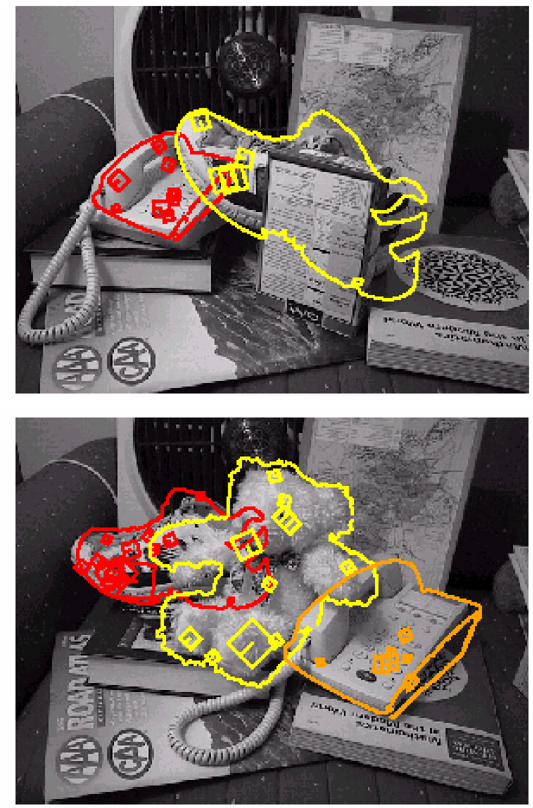
Recognition result

[Lowe]

3d objects

Background subtract for model boundaries

Objects recognized, though affine model not as accurate.



Recognition in spite of occlusion

[Lowe]

Recall: difficulties of voting

- Noise/clutter can lead to as many votes as true target
- Bin size for the accumulator array must be chosen carefully
- (Recall Hough Transform)

- In practice, good idea to make broad bins and spread votes to nearby bins, since verification stage can prune bad vote peaks.

A probabilistic interpretation (and re-tuning) of Lowe's system:

P. Moreels and P. Perona, **"A probabilistic cascade of detectors for individual object recognition,"** European Conference on Computer Vision, 2008.

Coarse-to-Fine detection

- Progressively narrow down focus on correct region of hypothesis space
- Reject with little computation cost irrelevant regions of search space
- Use first information that is easy to obtain
- Simple building blocks organized in a cascade
- Probabilistic interpretation of each step

Score of an extended hypothesis

Features assignments

Hypothesis: model + position

observed features geometry + appearance

database of models

$$P(H,V|F,M) = \frac{P(F,H,V|M)}{P(F|M)}$$

constant

Votes per model

Votes per model pose bin (Hough transform)

$$P(F,H,V|M) = P(H|M)\cdot P(\bar{N}|H,M)\cdot P(\tilde{N}|\bar{N},H,M)\cdot P(V|\tilde{N},\bar{N},H,M)\cdot P(F|V,\tilde{N},\bar{N},H,M)$$

Prior on model and poses

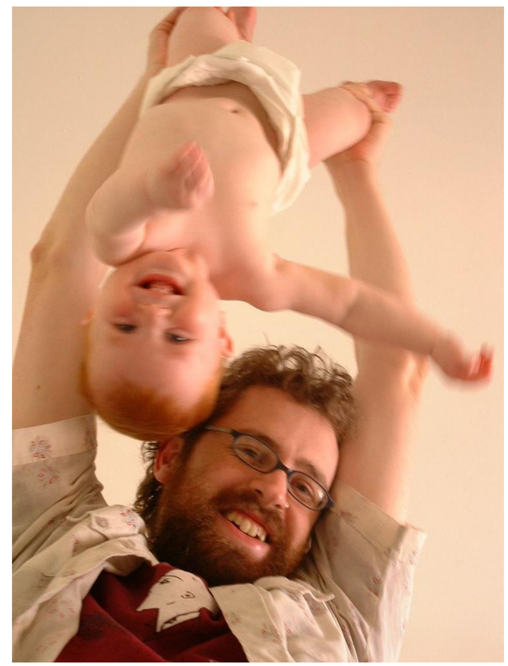
Prior on assignments (before actual observations)

Consistency (after PROSAC)

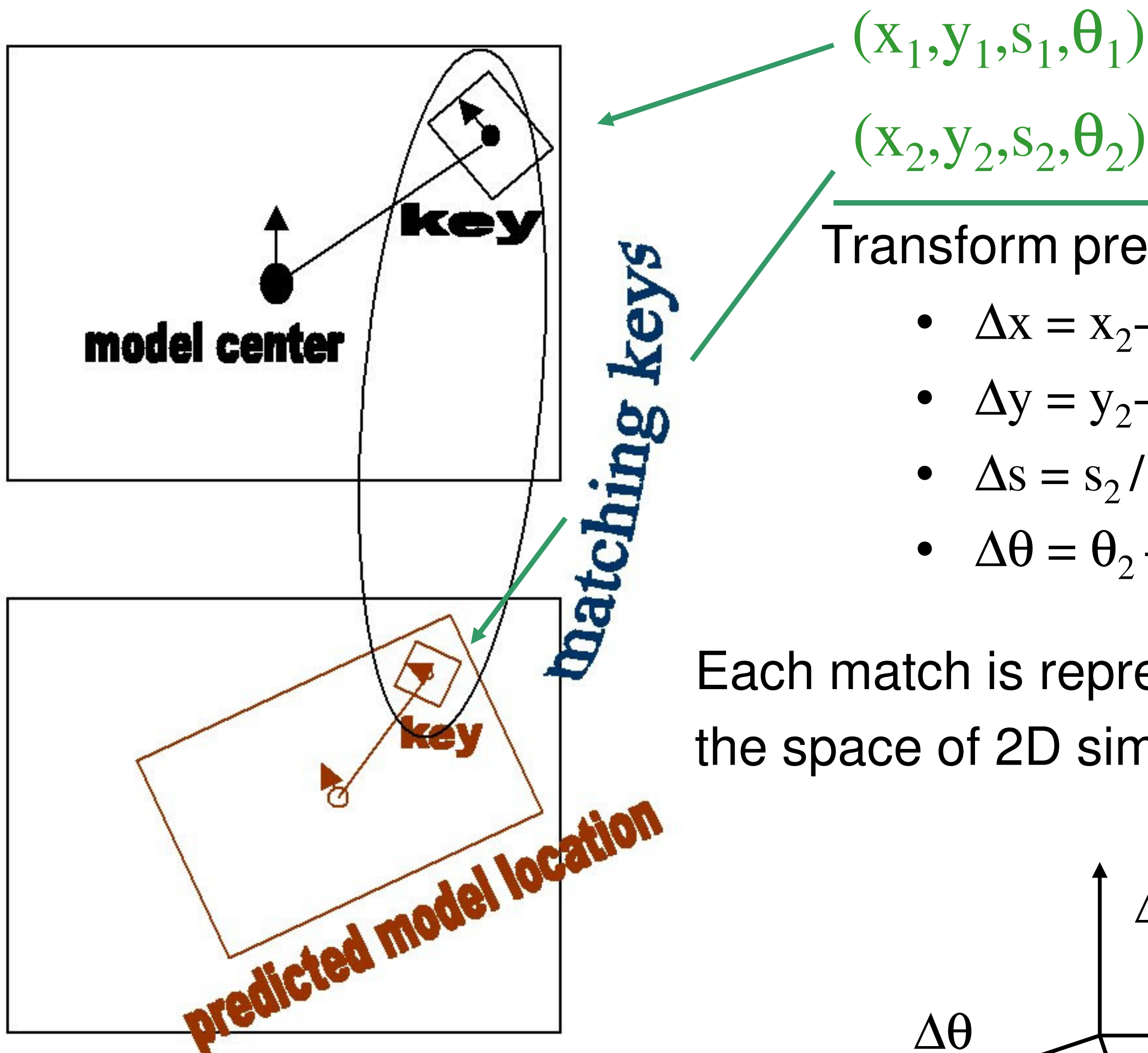
Coarse data : prior knowledge

- Which objects are likely to be there, which pose are they likely to have ?

unlikely
situations



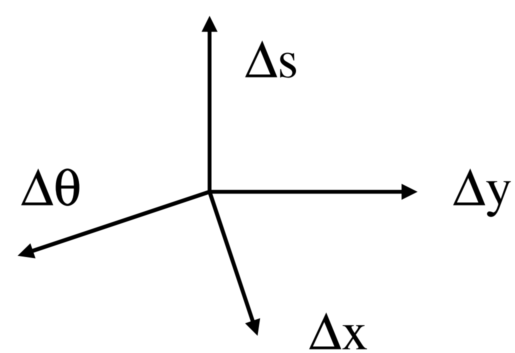
Coarse Hough transform



Transform predicted by this match:

- $\Delta x = x_2 - x_1$
- $\Delta y = y_2 - y_1$
- $\Delta s = s_2 / s_1$
- $\Delta\theta = \theta_2 - \theta_1$

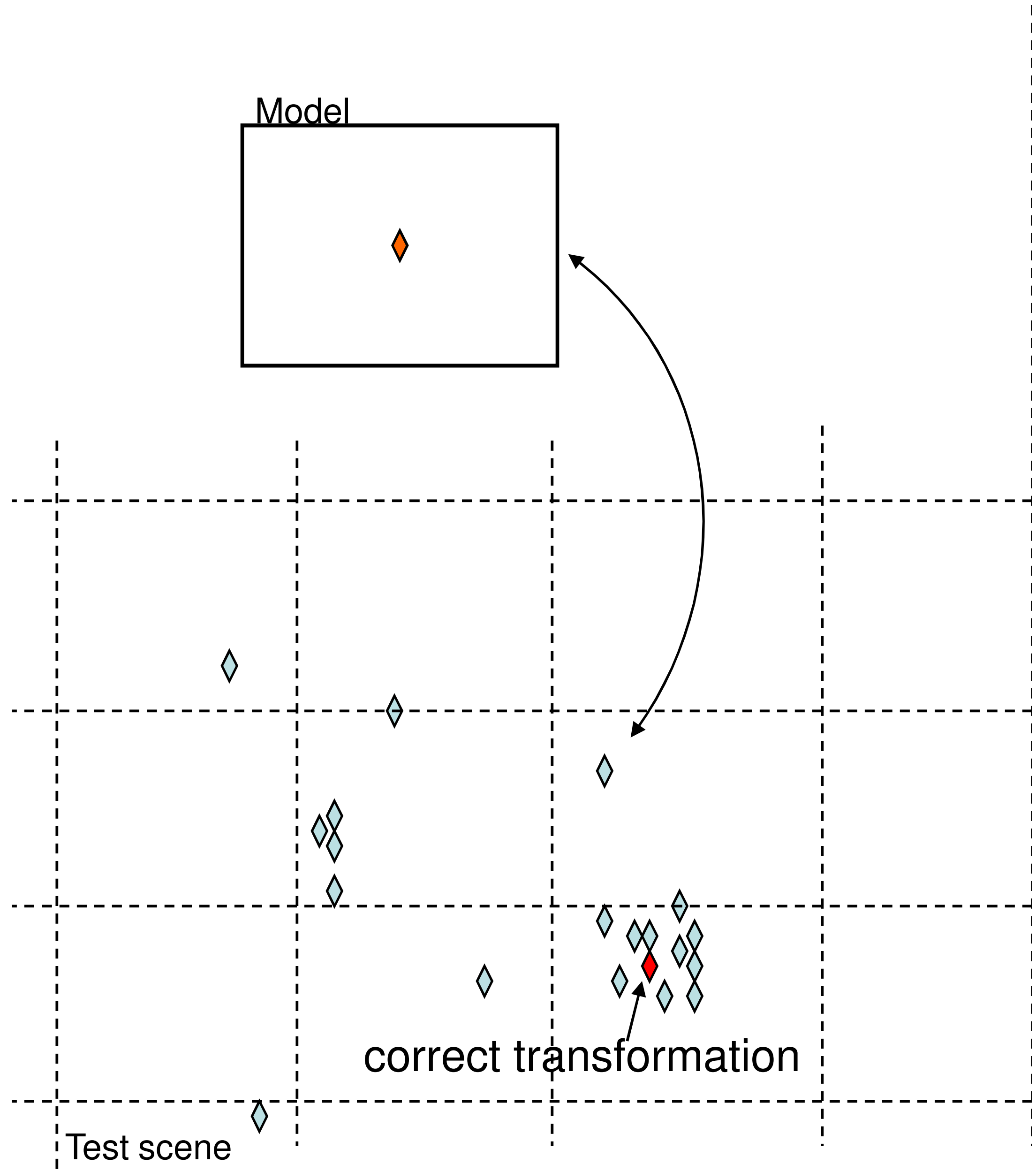
Each match is represented by a dot in the space of 2D similarities (Hough space)



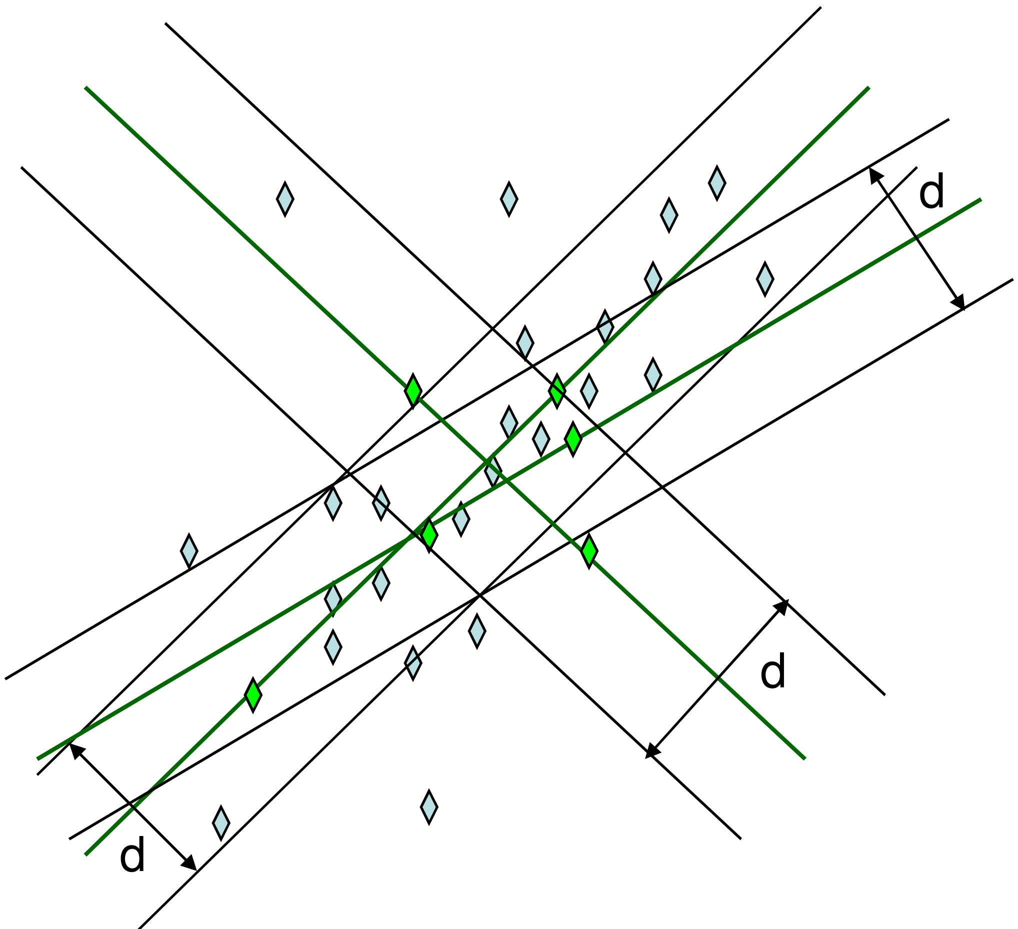
[Lowe1999,2004]

Coarse Hough transform

- Prediction of position of model center after transform
- The space of transform parameters is discretized into 'bins'
- Coarse bins to limit boundary issues and have a low false-alarm rate for this stage
- We count the number $\tilde{N}$ of votes collected by each bin.



Correspondence or clutter ? PROSAC



- Similar to RANSAC – robust statistic for parameter estimation
- Priority to candidates with good **quality** of appearance match
- 2D affine transform : 6 parameters
  $\Rightarrow$ each sample contains 3 candidate correspondences.

Output of PROSAC : pose transformation
+ set of features correspondences

[Fischler 1973]
[Chum&Matas 2005]

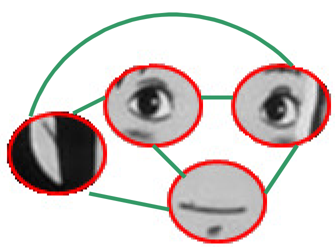
Consistency

Consistency between observations and predictions from hypothesis

$$P(F|V,\tilde{N},\bar{N},H,M) = \prod_{V(i)\neq 0} p_{fg}(f_i|H,f_{V(i)}) \cdot \prod_{V(i)=0} p_{bg}(f_i)$$

Common-frame approximation : parts are conditionally independent once reference position of the object is fixed. [Lowe1999,Huttenlocher90,Moreels04]

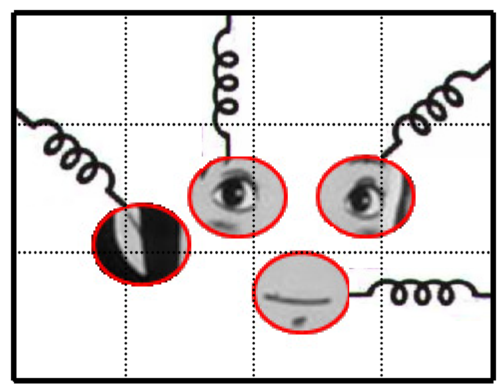
Constellation model



$$p_m(x_1 \ldots x_N)$$

$$D.O.F. = O(Parts^2)$$

Common-frame



$$\prod_i p_m(x_i|\Theta_m)$$

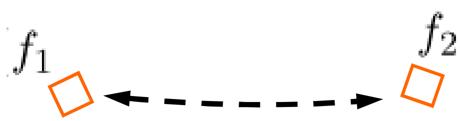
position of model m

$$D.O.F. = O(Parts)$$

Consistency

Consistency between observations and predictions from hypothesis

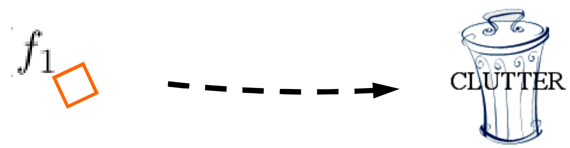
$$P(F|V,\tilde{N},\bar{N},H,M) = \prod_{V(i)\neq 0} p_{fg}(f_i|H,f_{V(i)}) \cdot \prod_{V(i)=0} p_{bg}(f_i)$$



foreground features

appearance

geometry



'null' assignments

appearance

geometry

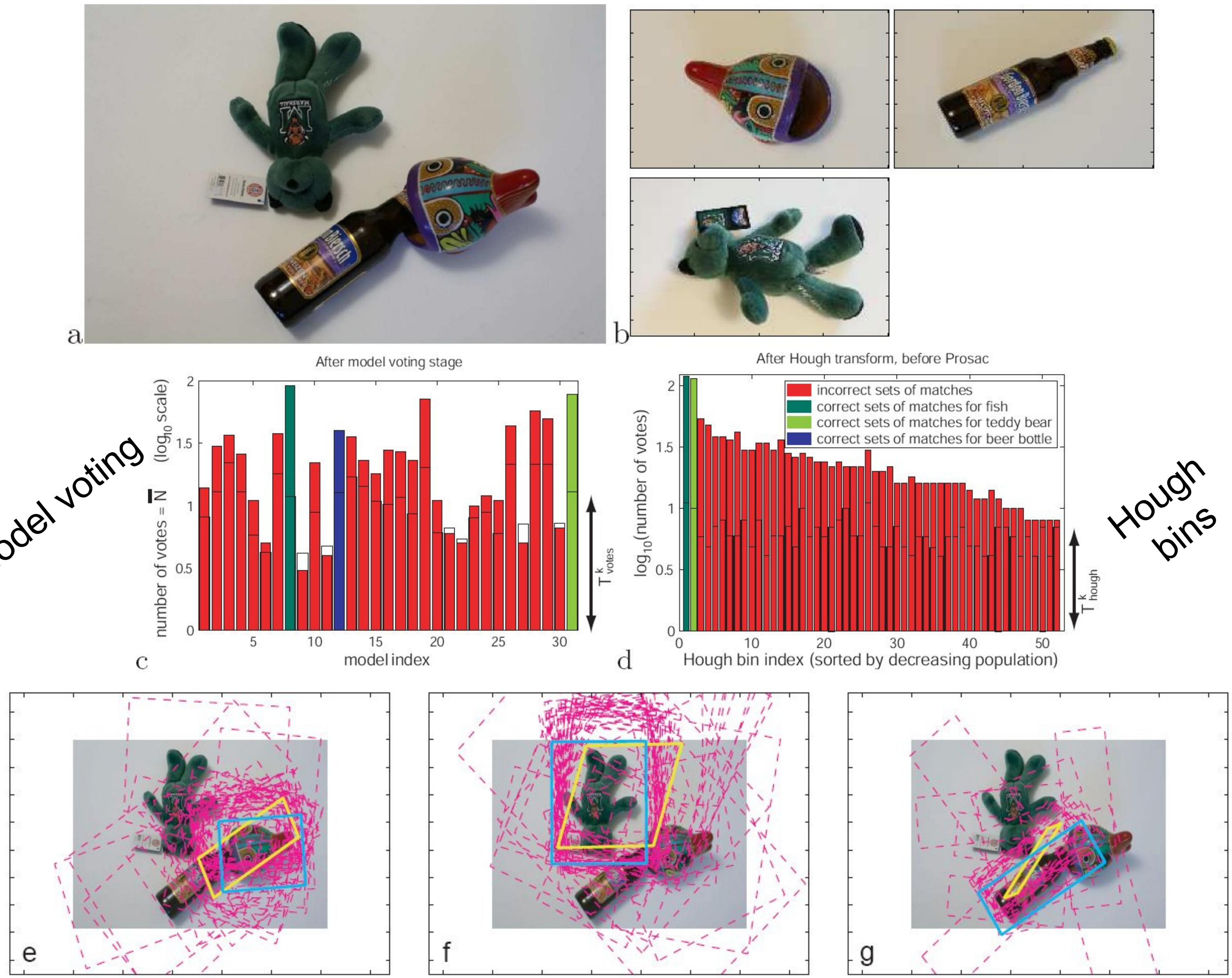
$$p_{fg}(f_i|H,f_{V(i)}) = p_{fg,\mathcal{A}}(\mathcal{A}|H,\mathcal{A}_{V(i)}) \cdot p_{fg,\mathcal{X}}(\mathcal{X}|H,\mathcal{X}_{V(i)})$$

Consistency - appearance

Consistency - geometry

An example

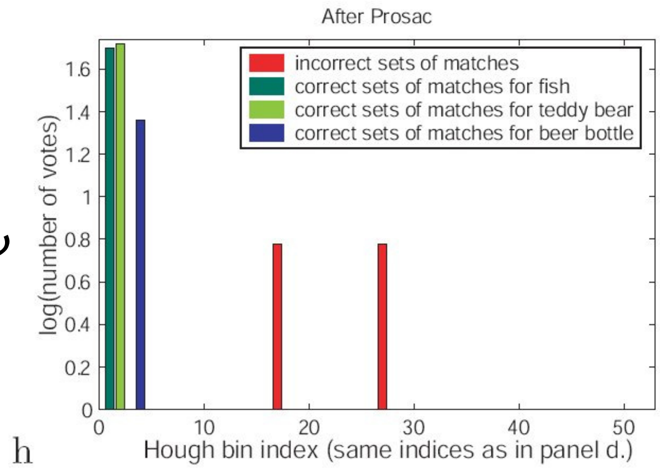
Model voting

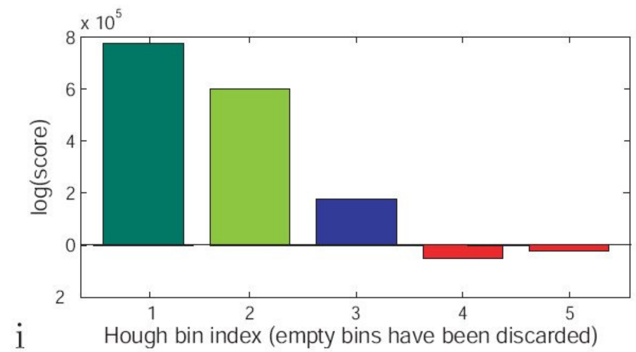


Hough bins

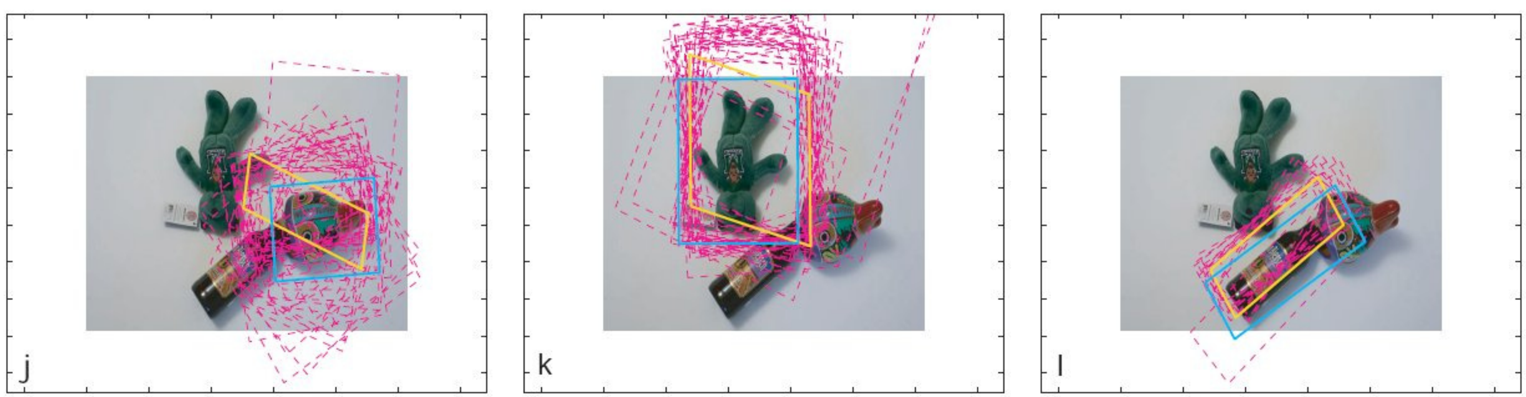
An example

After PROSAC

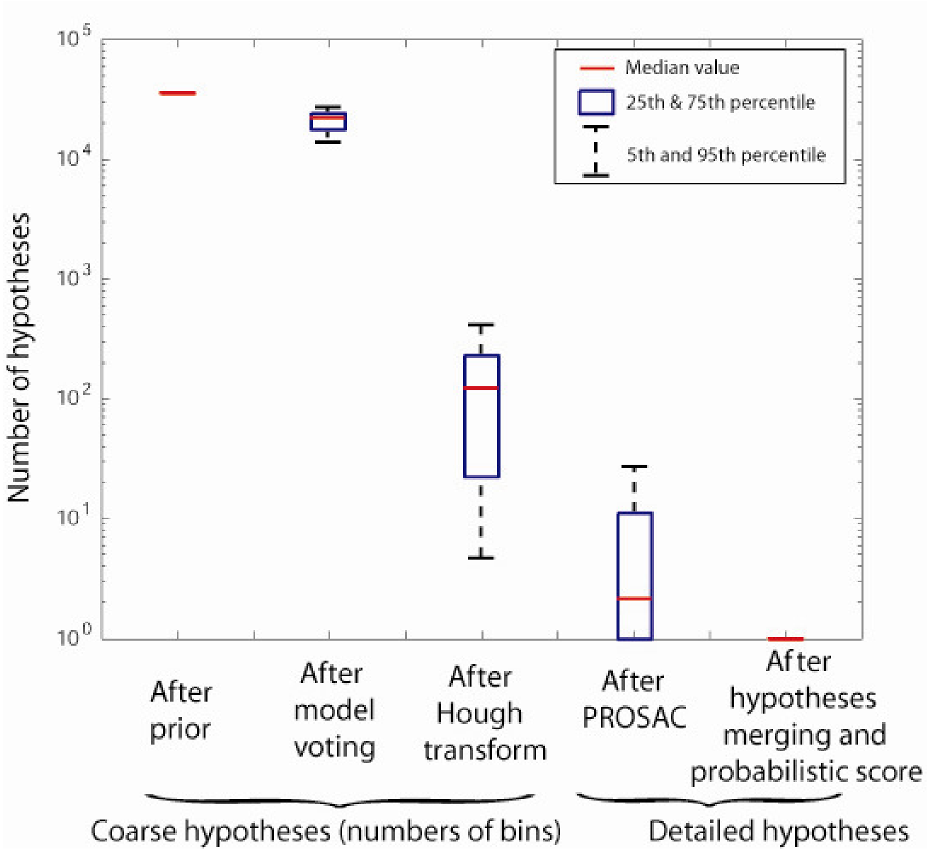


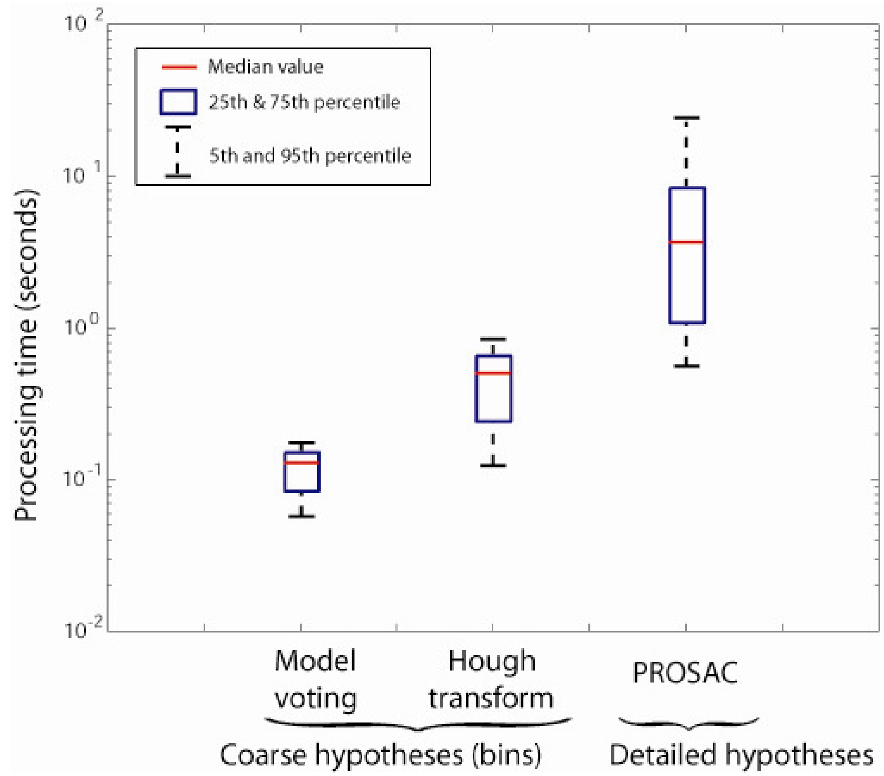


Probabilistic scores



Efficiency of coarse-to-fine processing





Giuseppe Toys database – Models

61 objects, 1-2 views/object

Giuseppe Toys database – Test scenes

141 test scenes

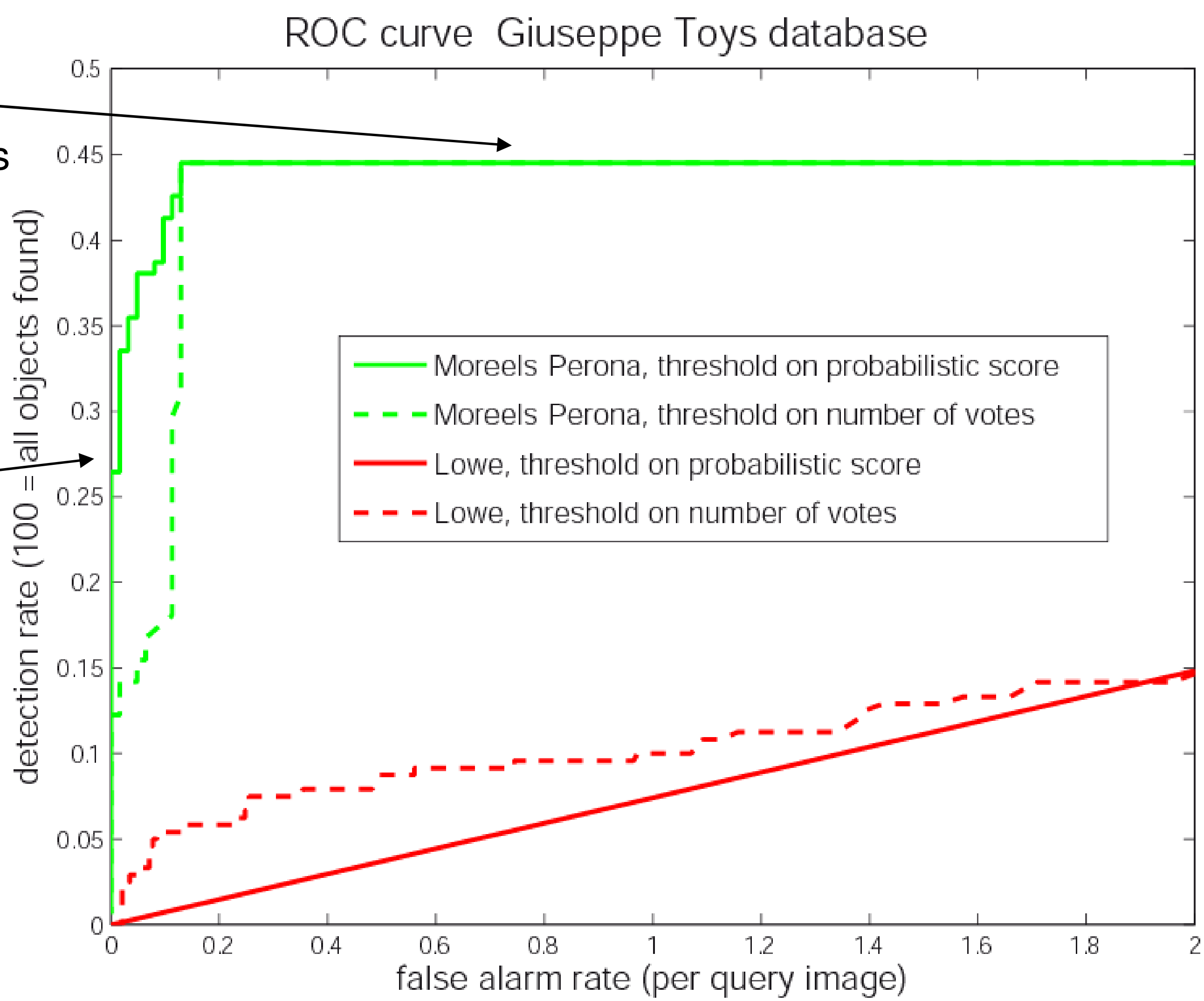
Results – Giuseppe Toys database

-

undetected objects:
features with poor appearance distinctiveness index to incorrect models

+

Lower false alarm rate
- more systematic verification of geometry consistency
- more consistent verification of geometric consistency



Lowe'99,'04

Conclusions – Moreels and Perona

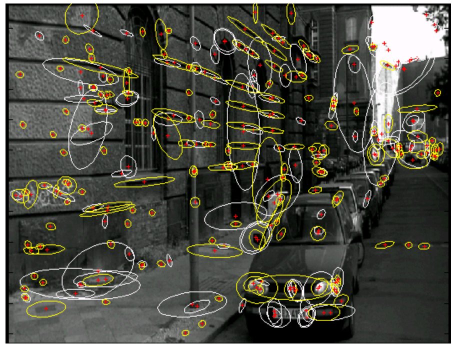
- Coarse-to-fine strategy prunes irrelevant search branches at early stages.
- Probabilistic interpretation of each step.
- Higher performance than Lowe, especially in cluttered environment.
- Front end (features) needs more work for smooth or shiny surfaces.

Scaling up:
BOW Indexing

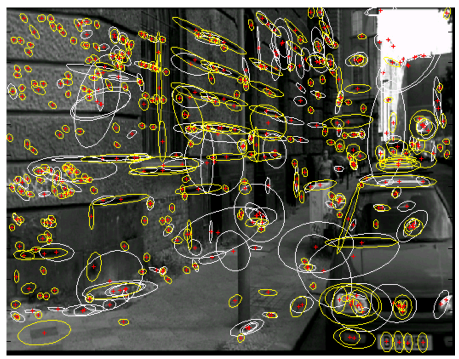
Outline of a large-scale retrieval strategy

regions

frames



invariant descriptor vectors

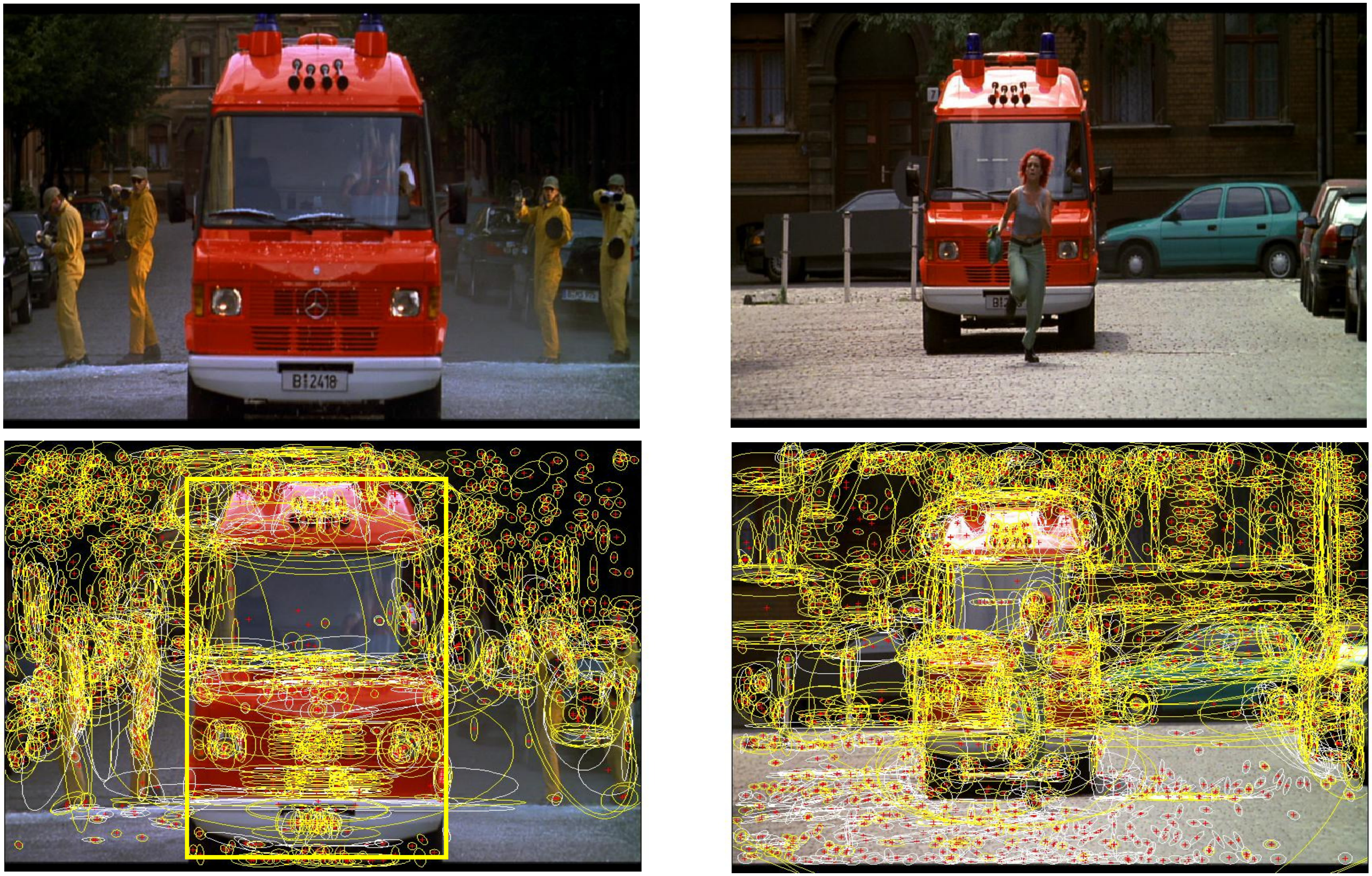


invariant descriptor vectors

1. Compute affine covariant regions in each frame independently
2. “Label” each region by a vector of descriptors based on its intensity
3. Finding corresponding regions is transformed to finding nearest neighbour vectors
4. Rank retrieved frames by number of corresponding regions
5. Verify retrieved frame based on spatial consistency

Slide credit: J. Sivic

Example of object recognition



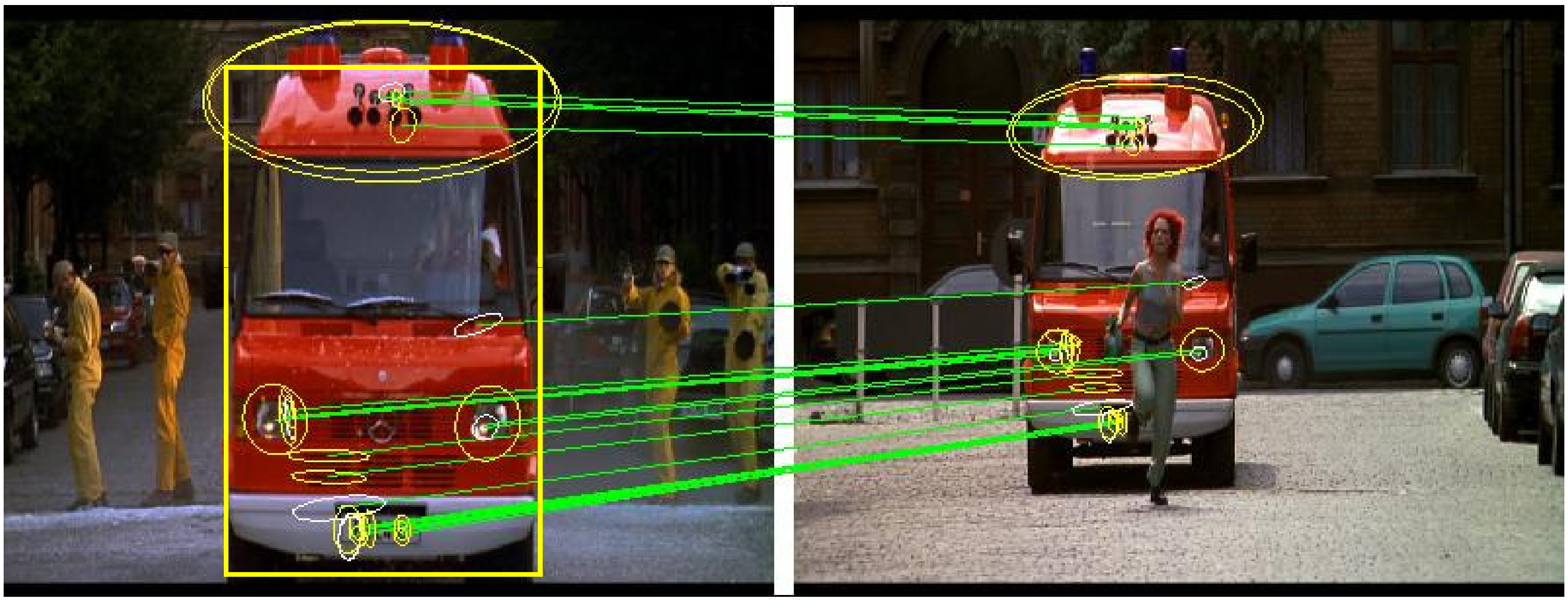
1000+ descriptors per frame

Shape adapted regions

Maximally stable regions

Slide credit: J. Sivic

Match regions between frames using SIFT descriptors and spatial consistency



Multiple regions overcome problem of partial occlusion

Shape adapted regions

Maximally stable regions

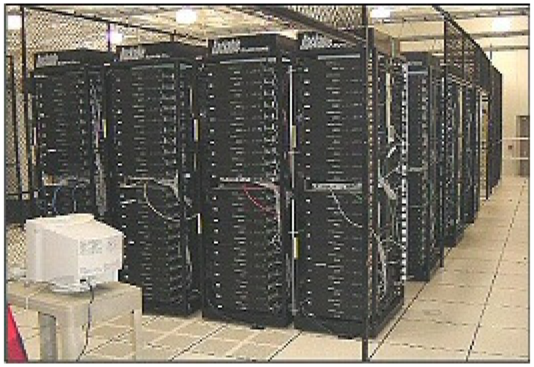
Slide credit: J. Sivic

Visual search using local regions

Schmid and Mohr '97 – 1k images
Sivic and Zisserman'03 – 5k images
Nister and Stewenius'06 – 50k images (1M)
Philbin et al.'07 – 100k images
Chum et al.'07 + Jegou and Schmid'07 – 1M images
Chum et al.'08 – 5M images

Index 1 billion (10^9) images

– 200 servers each indexing 5M images?



Slide credit: J. Sivic

Beyond Nearest Neighbors…
Indexing local features using inverted file index

Index

For text documents, an efficient way to find all *pages* on which a *word* occurs is to use an index…

We want to find all *images* in which a *feature* occurs.

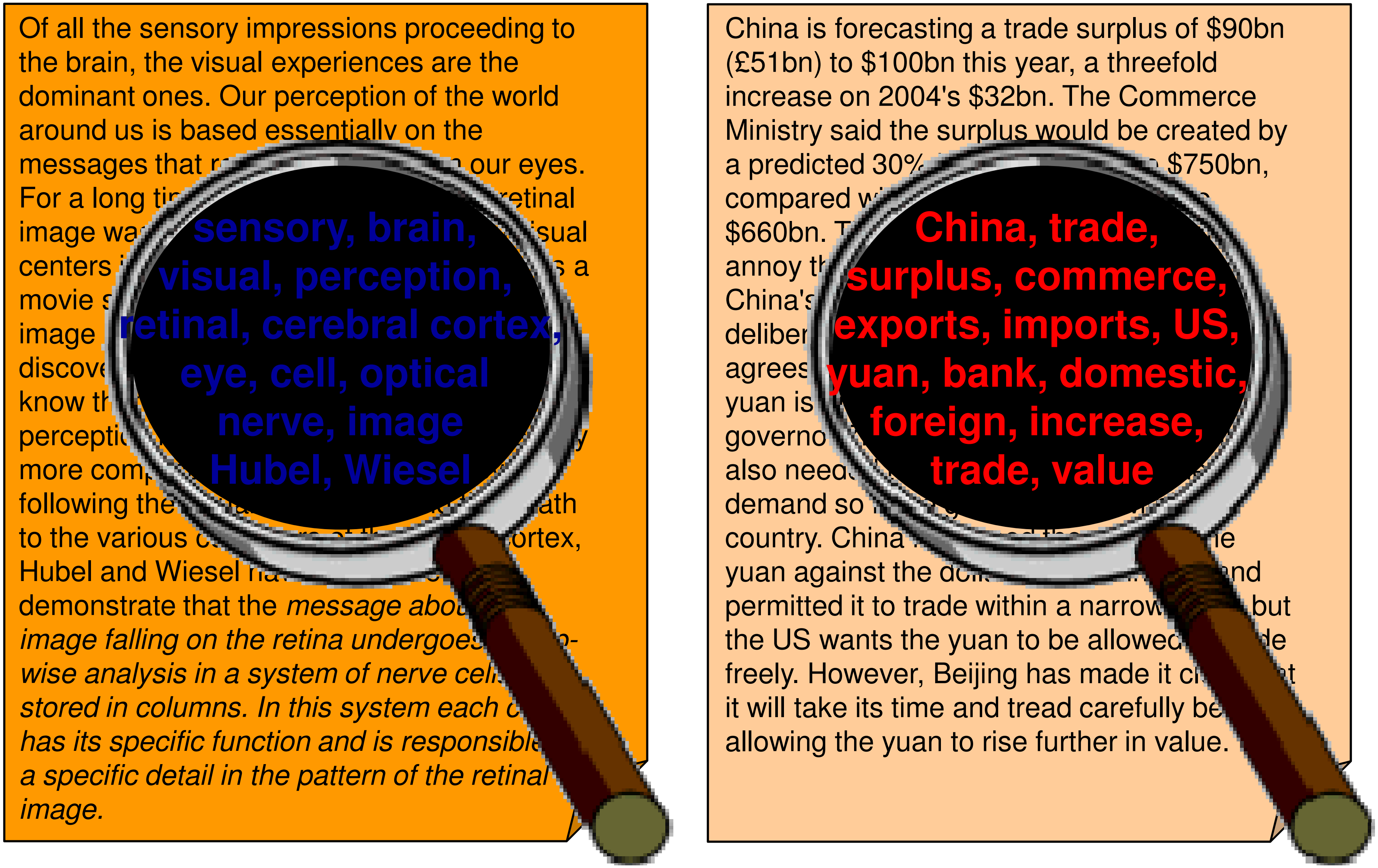
To use this idea, we'll need to map our features to "visual words".

Slide credit: J. Sivic

Object → **Bag of ‘words’**

Slide credit L. Fei-Fei

Analogy to documents

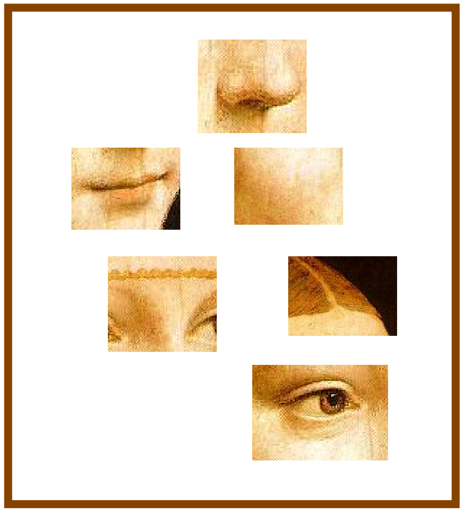


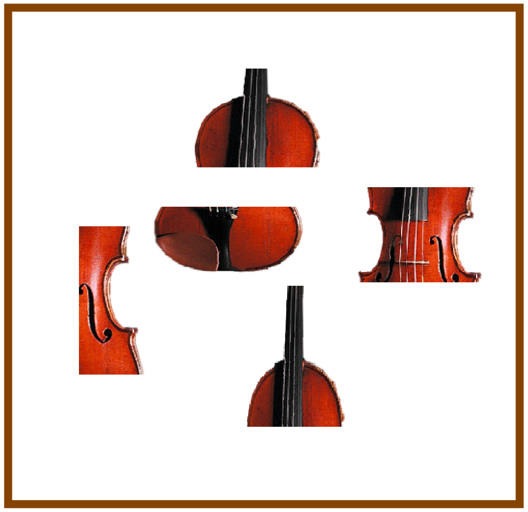
Slide credit L. Fei-Fei

A clarification: definition of “BoW”

Looser definition

- Independent features





Slide credit L. Fei-Fei

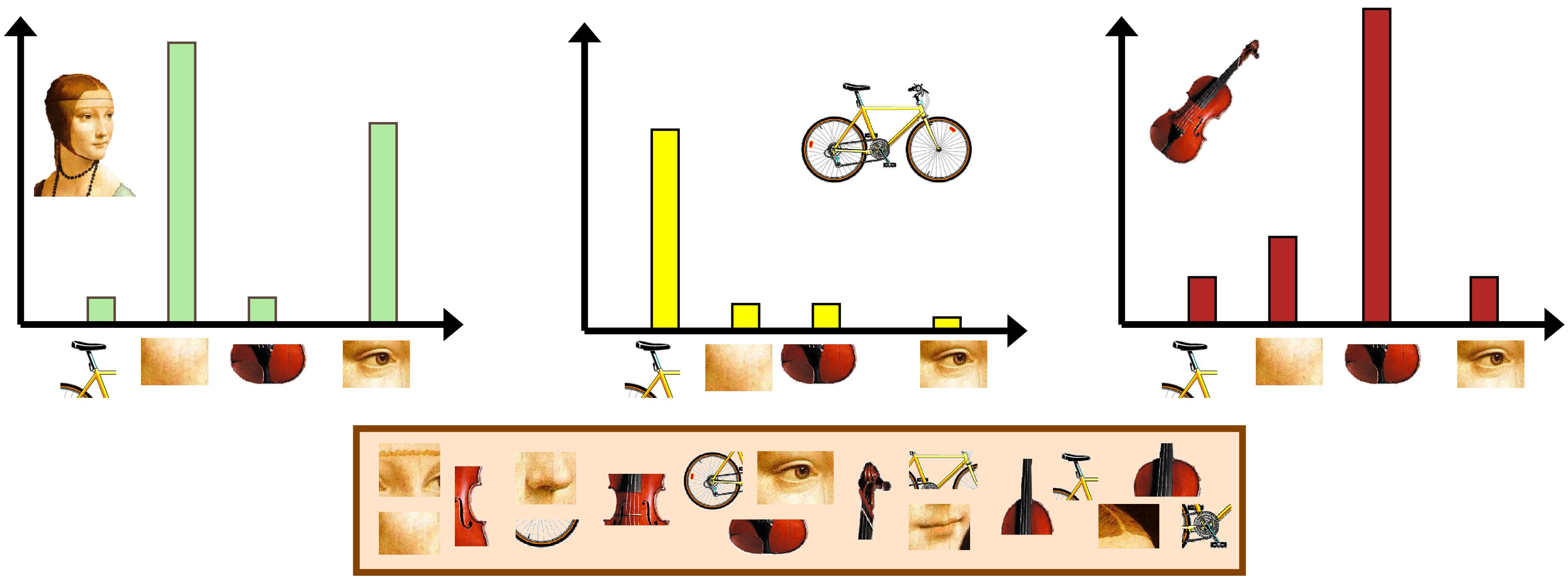
A clarification: definition of "BoW"

Looser definition

- Independent features

Stricter definition

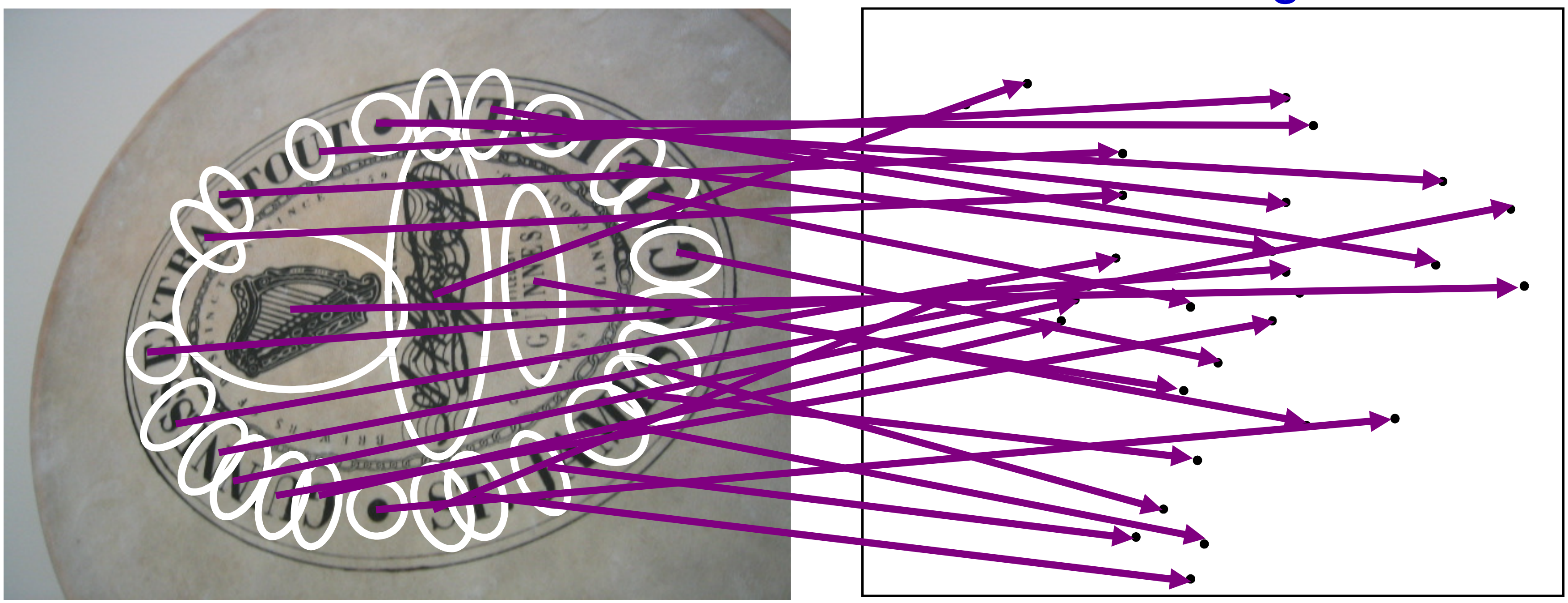
- Independent features
- histogram representation



Slide credit L. Fei-Fei

Visual words: main idea

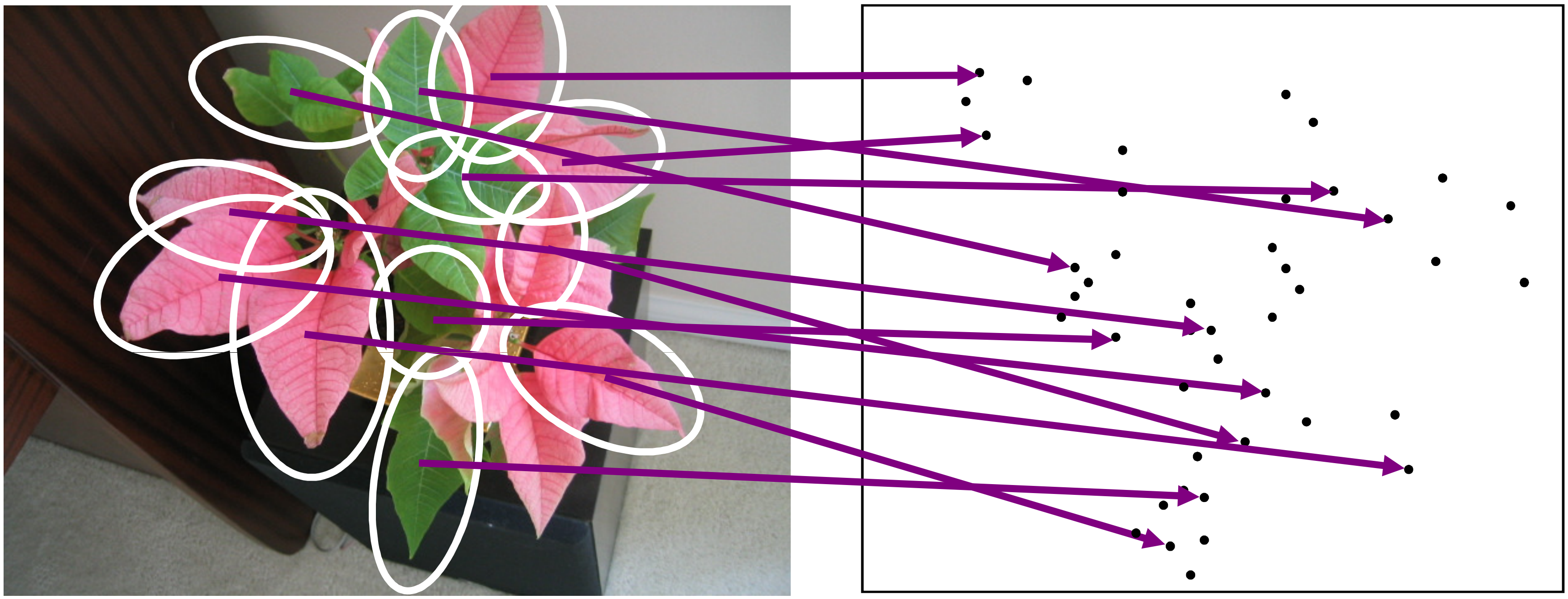
Extract some local features from a number of images …



e.g., SIFT descriptor space: each point is 128-dimensional

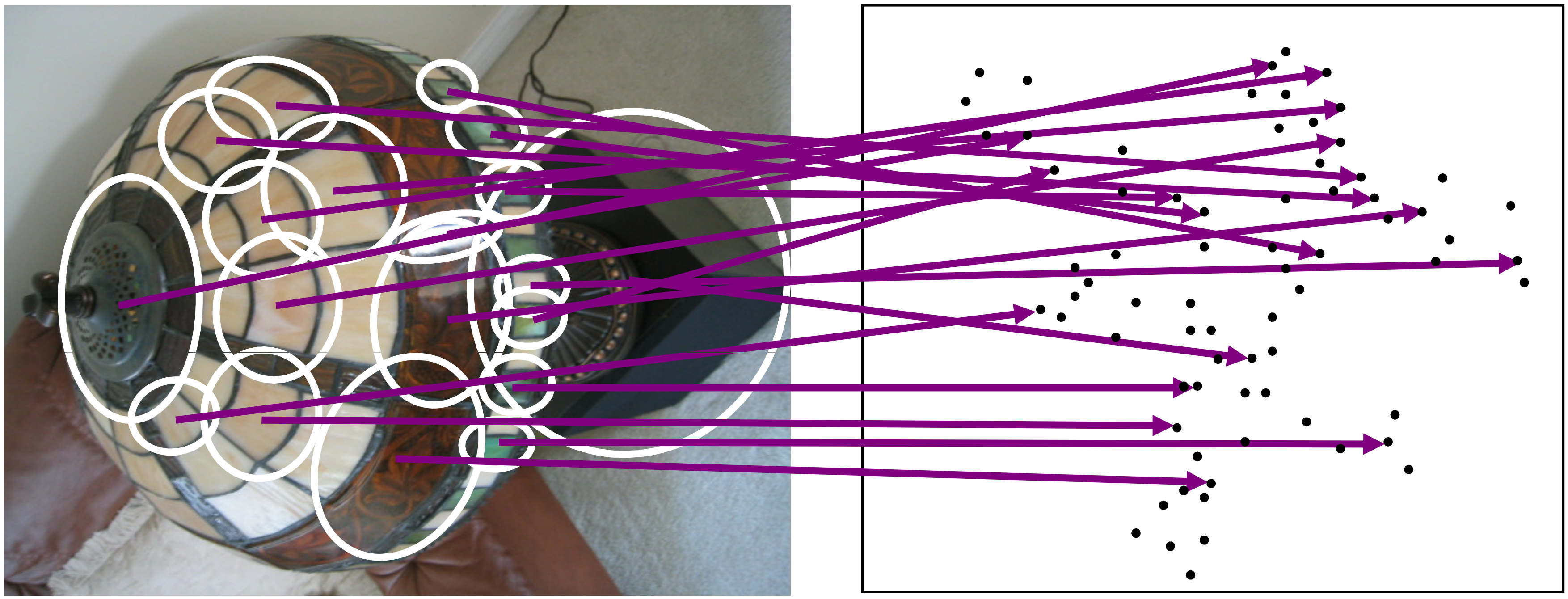
Slide credit: D. Nister

Visual words: main idea



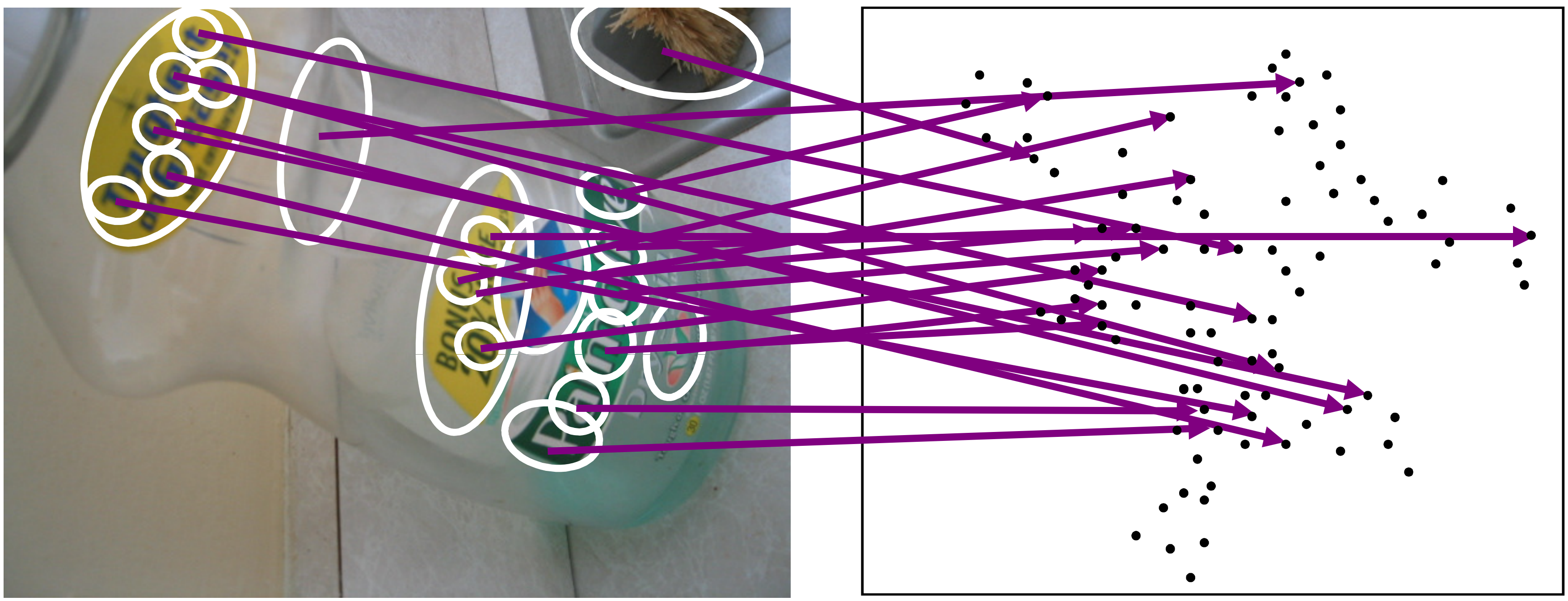
Slide credit: D. Nister

Visual words: main idea

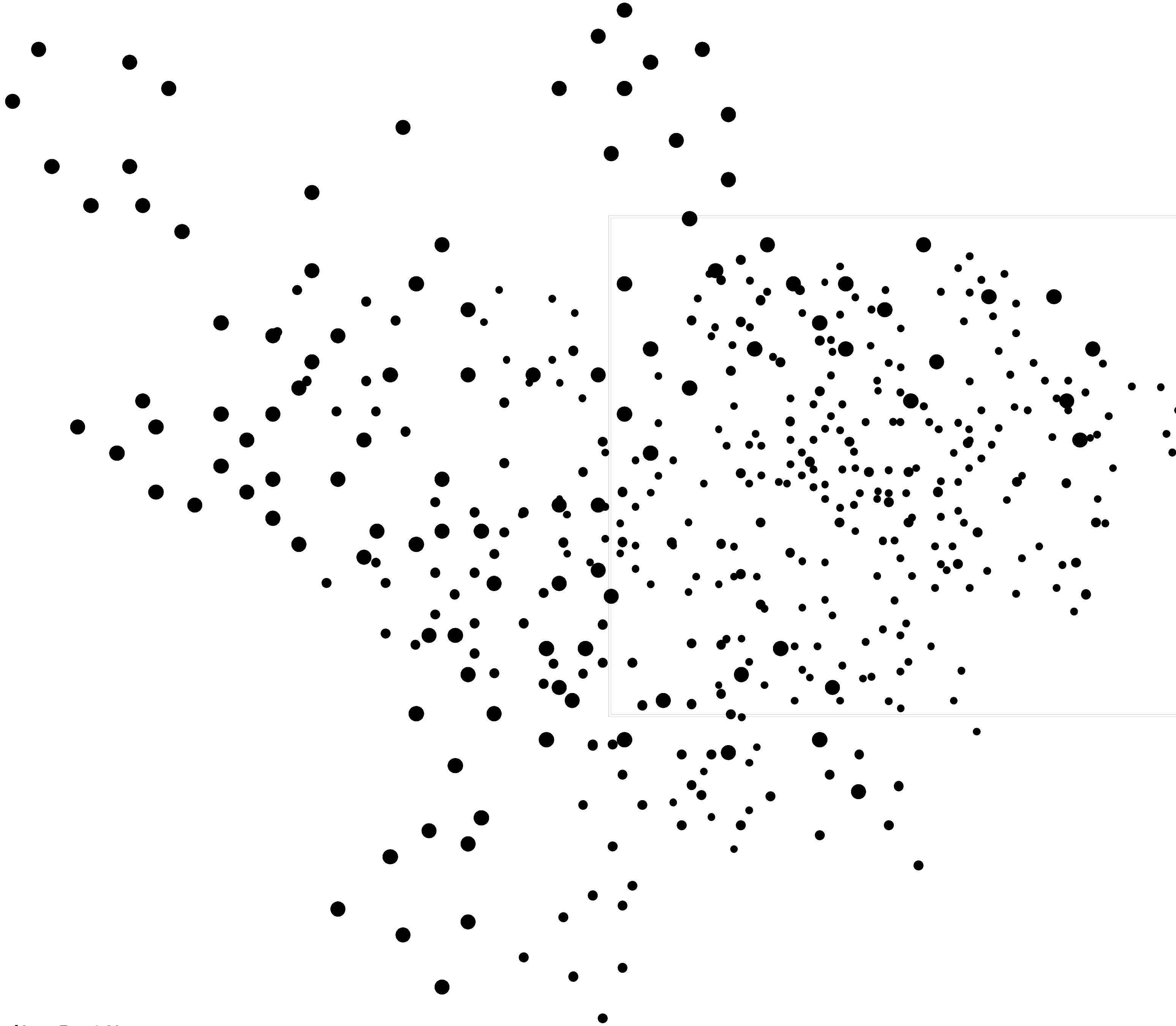


Slide credit: D. Nister

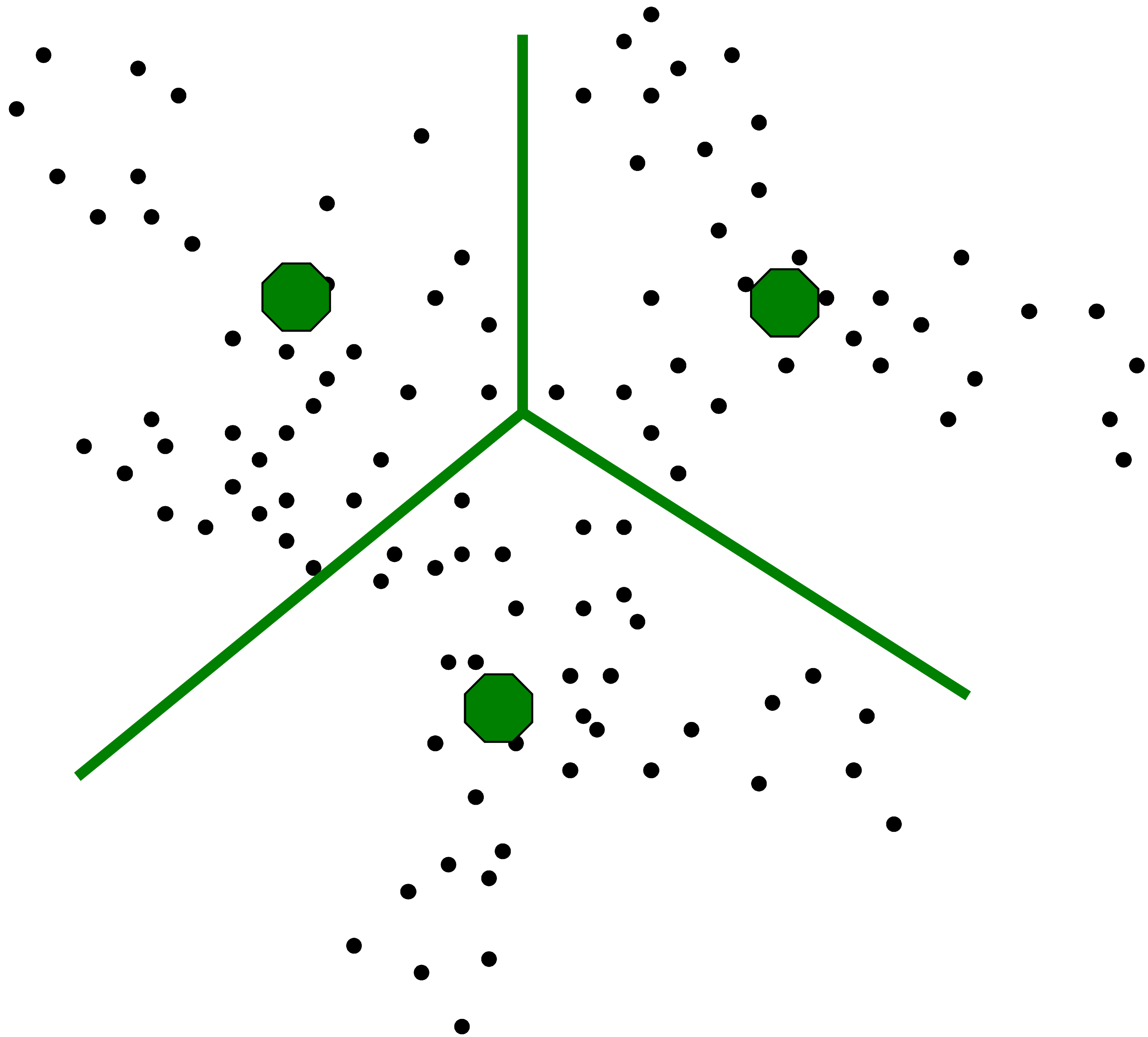
Visual words: main idea



Slide credit: D. Nister



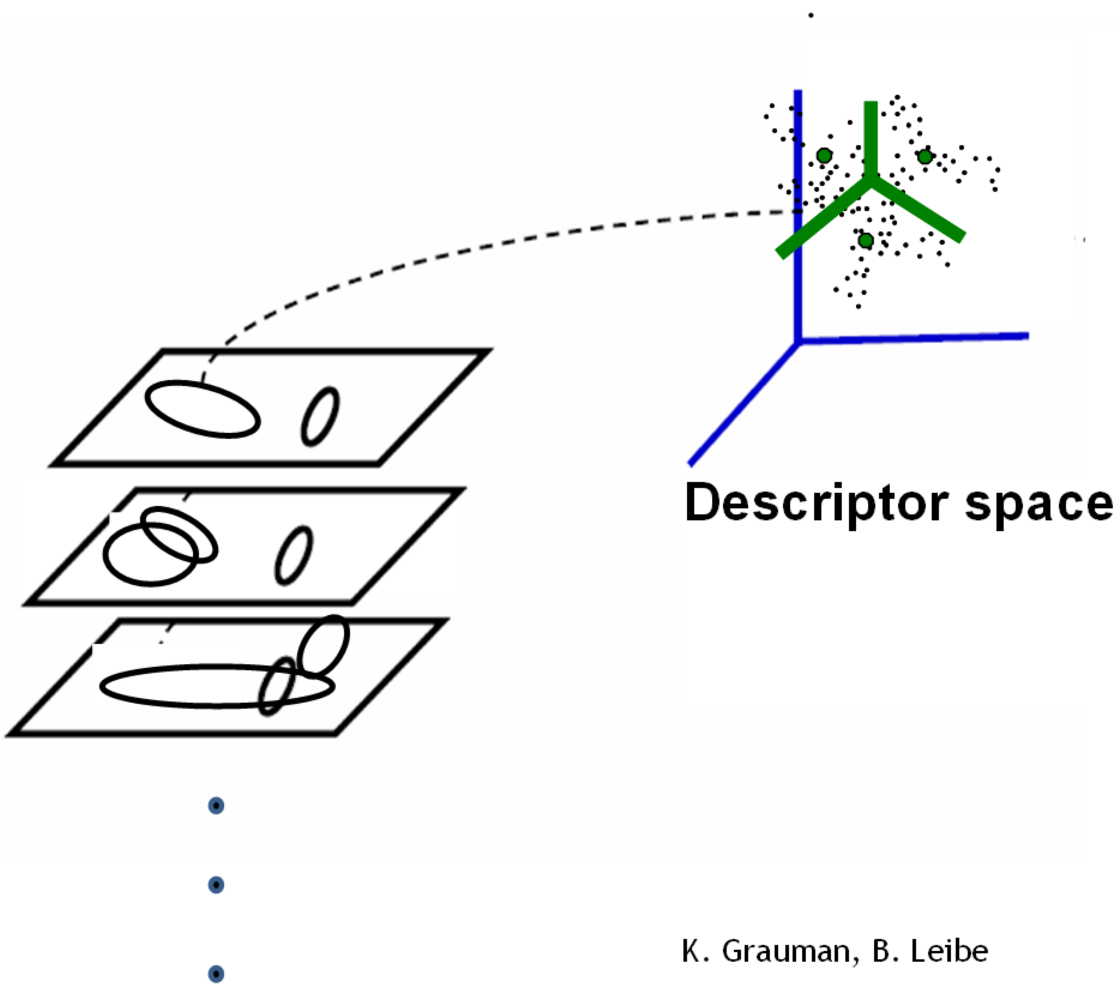
Slide credit: D. Nister



Slide credit: D. Nister

Visual words: main idea

Map high-dimensional descriptors to tokens/words by quantizing the feature space

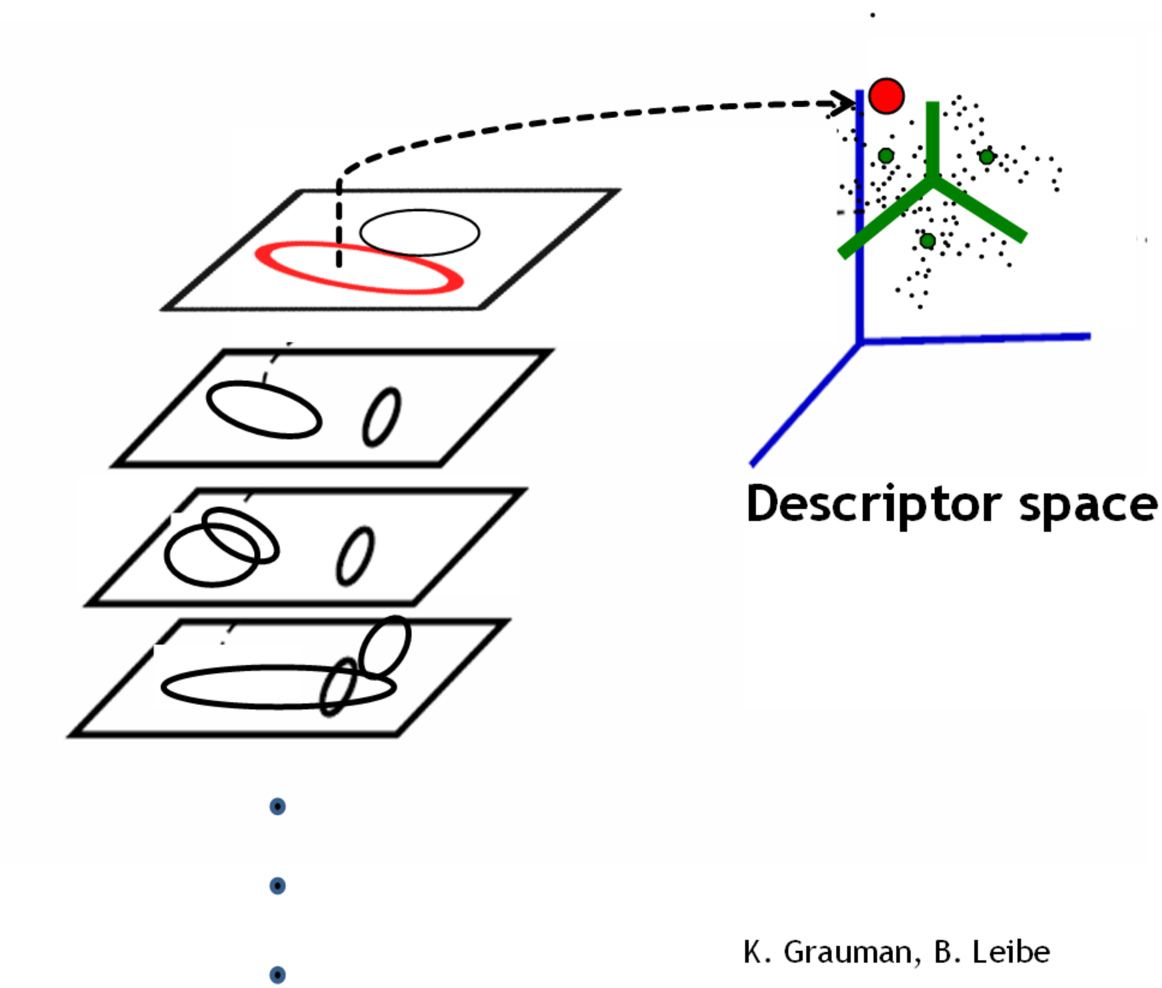


- Quantize via clustering, let cluster centers be the prototype "words"

K. Grauman, B. Leibe

Visual words: main idea

Map high-dimensional descriptors to tokens/words by quantizing the feature space

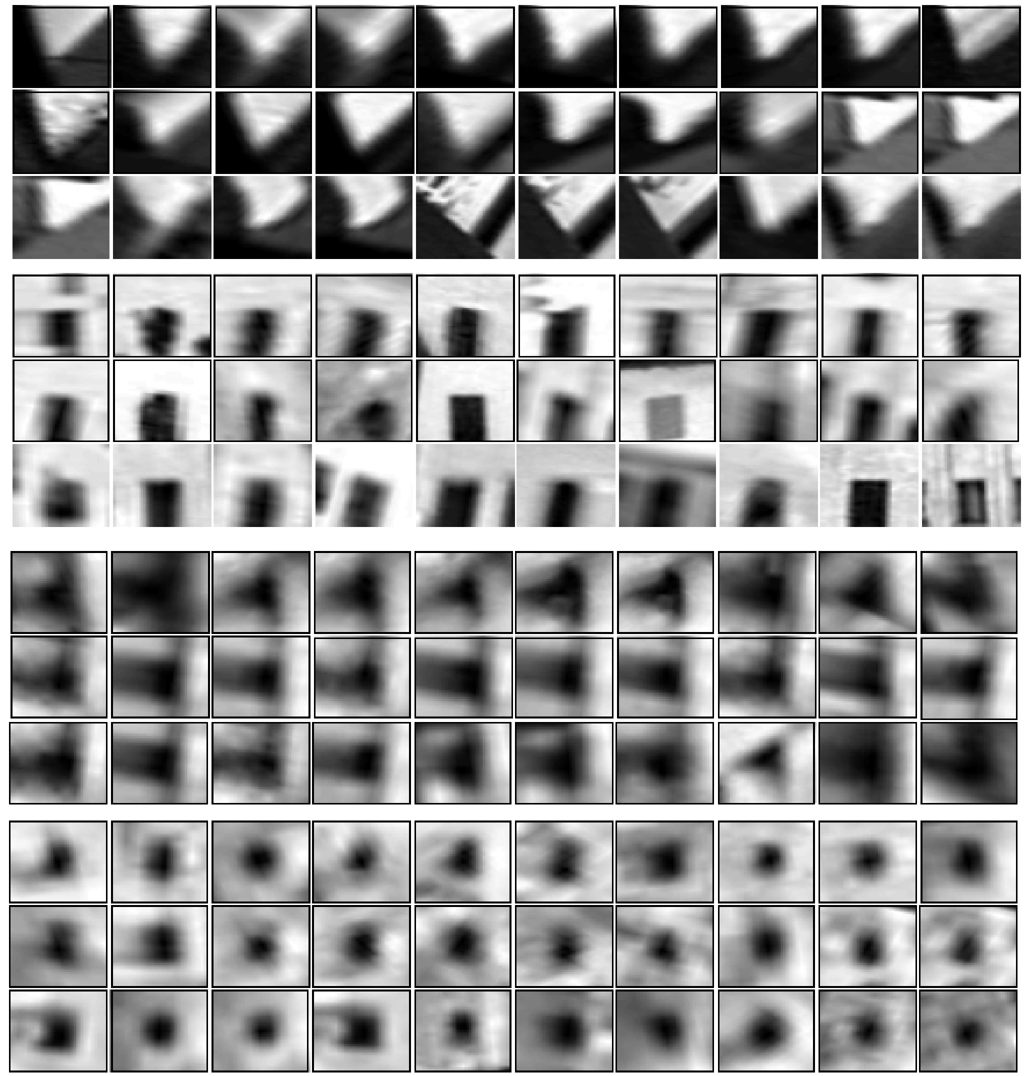


- Determine which word to assign to each new image region by finding the closest cluster center.

K. Grauman, B. Leibe

Visual words

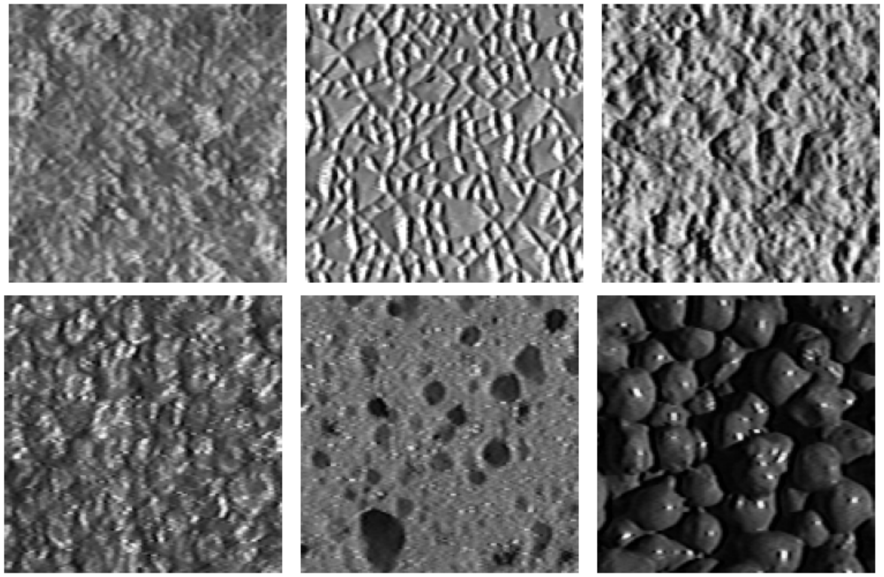
Example: each group of patches belongs to the same visual word



Slide credit: J. Sivic

Visual words

- First explored for texture and material representations
- *Texton* = cluster center of filter responses over collection of images
- Describe textures and materials based on distribution of prototypical texture elements.



Leung & Malik 1999; Varma & Zisserman, 2002; Lazebnik, Schmid & Ponce, 2003;

Slide credit: J. Sivic

Inverted file index for images comprised of visual words

frame #5

frame #10

Word number	List of image numbers
1	→ 5,10, ...
2	→ 10,...
...	...

- Score each image by the number of common visual words (tentative correspondences)
- But: does not take into account spatial layout of regions

Slide credit: J. Sivic

Clustering / quantization methods

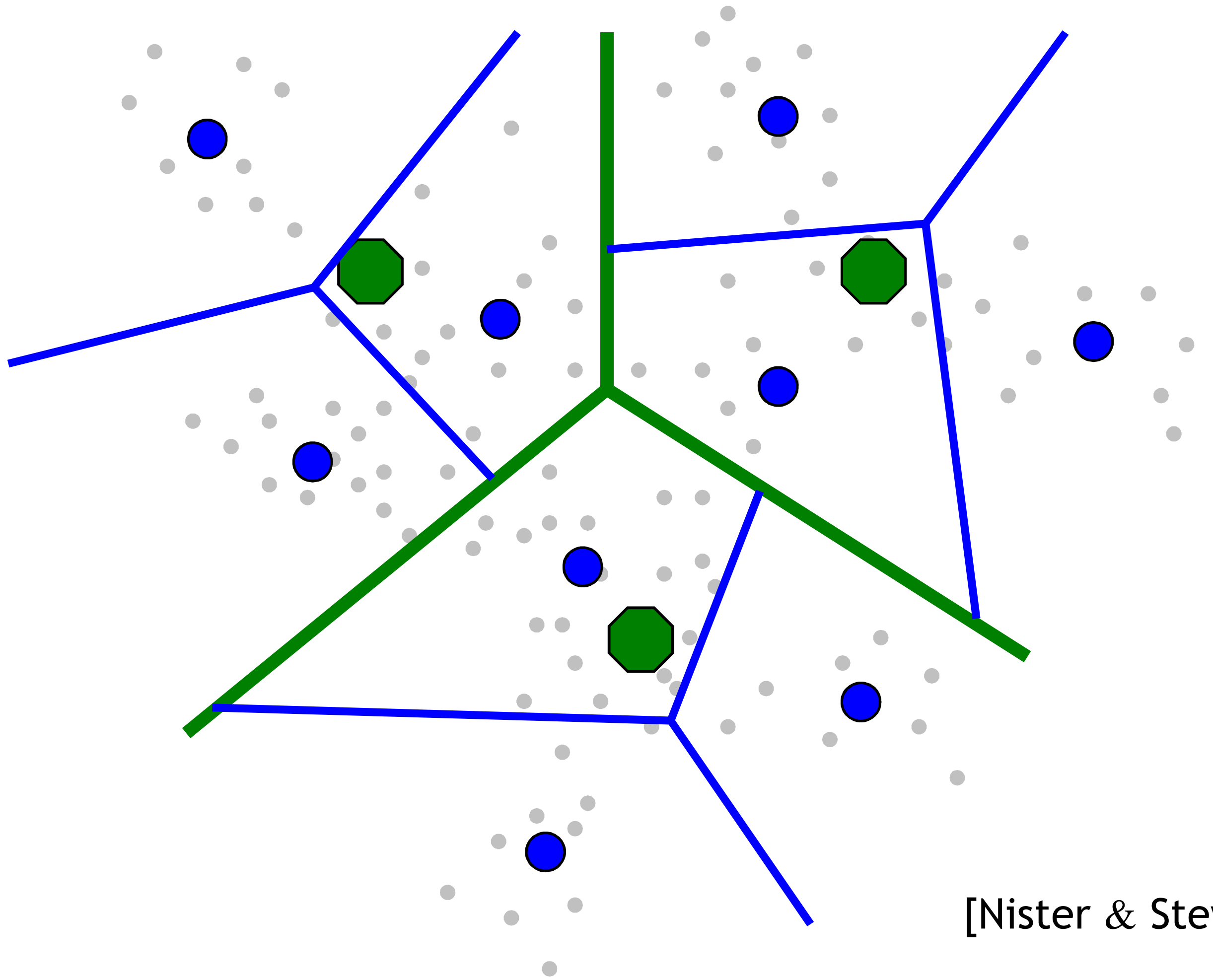
- k-means (typical choice), agglomerative clustering, mean-shift,…

- **Hierarchical clustering: allows faster insertion / word assignment while still allowing large vocabularies**
  - **Vocabulary tree [Nister & Stewenius, CVPR 2006]**

Slide credit: J. Sivic

Example: Recognition with Vocabulary Tree

Tree construction:

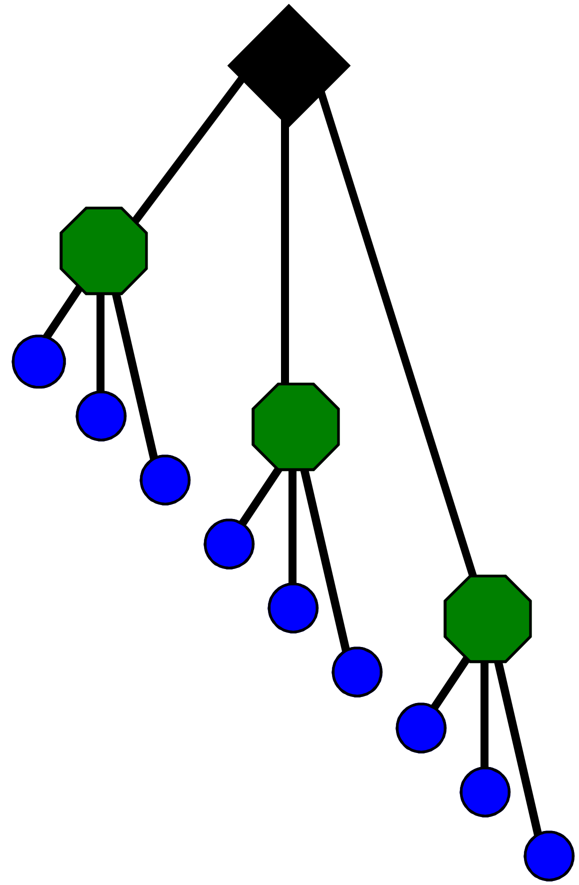


[Nister & Stewenius, CVPR'06]

Slide credit: David Nister

Vocabulary Tree

Training: Filling the tree

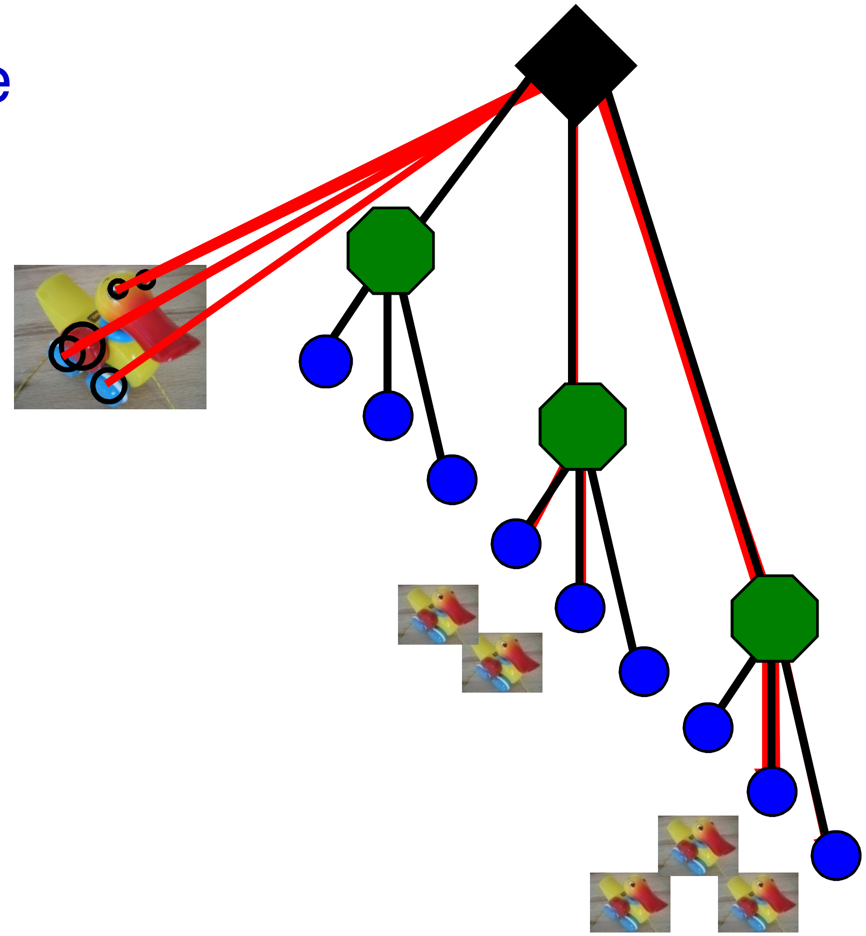


[Nister & Stewenius, CVPR'06]

Slide credit: David Nister

Vocabulary Tree

Training: Filling the tree

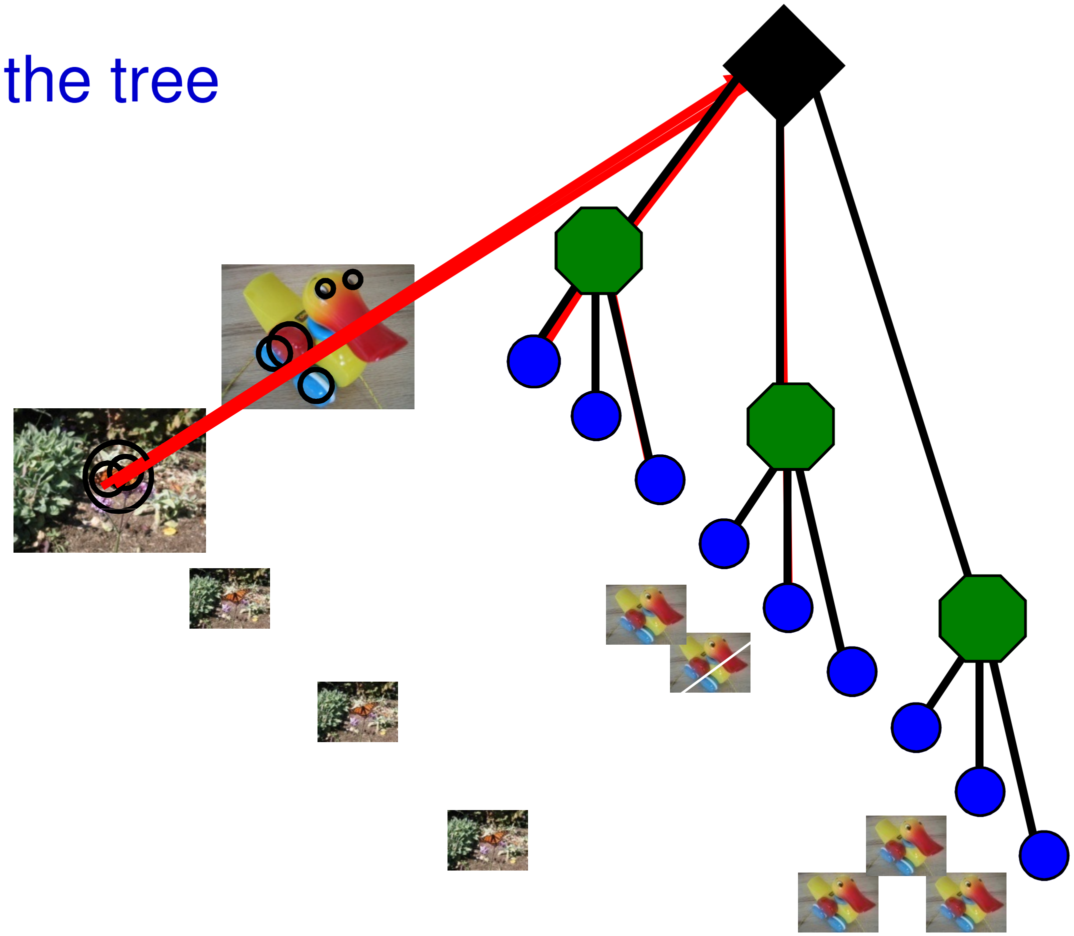


[Nister & Stewenius, CVPR'06]

Slide credit: David Nister

Vocabulary Tree

Training: Filling the tree

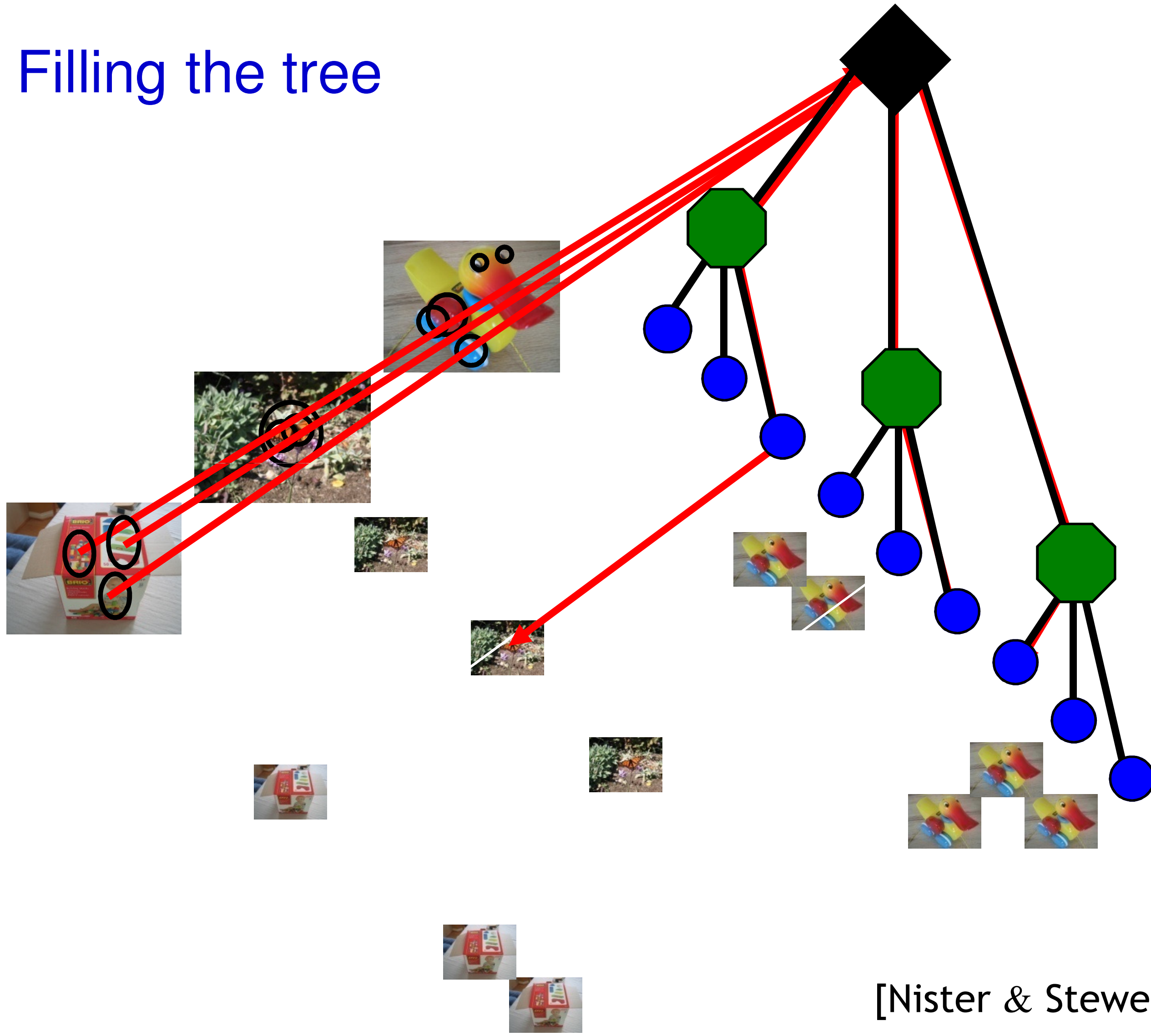


[Nister & Stewenius, CVPR'06]

Slide credit: David Nister

Vocabulary Tree

Training: Filling the tree

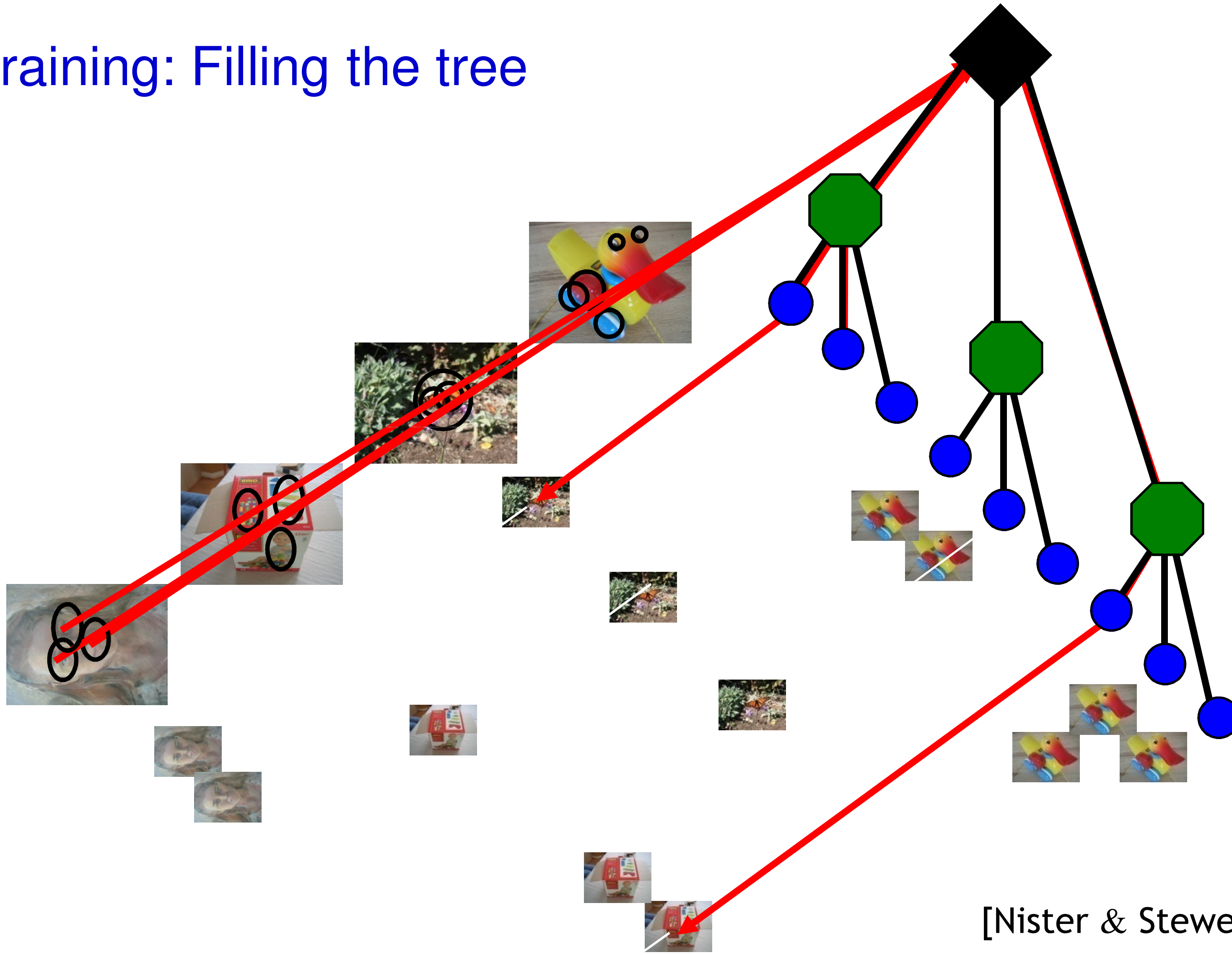


[Nister & Stewenius, CVPR'06]

Slide credit: David Nister

Vocabulary Tree

Training: Filling the tree

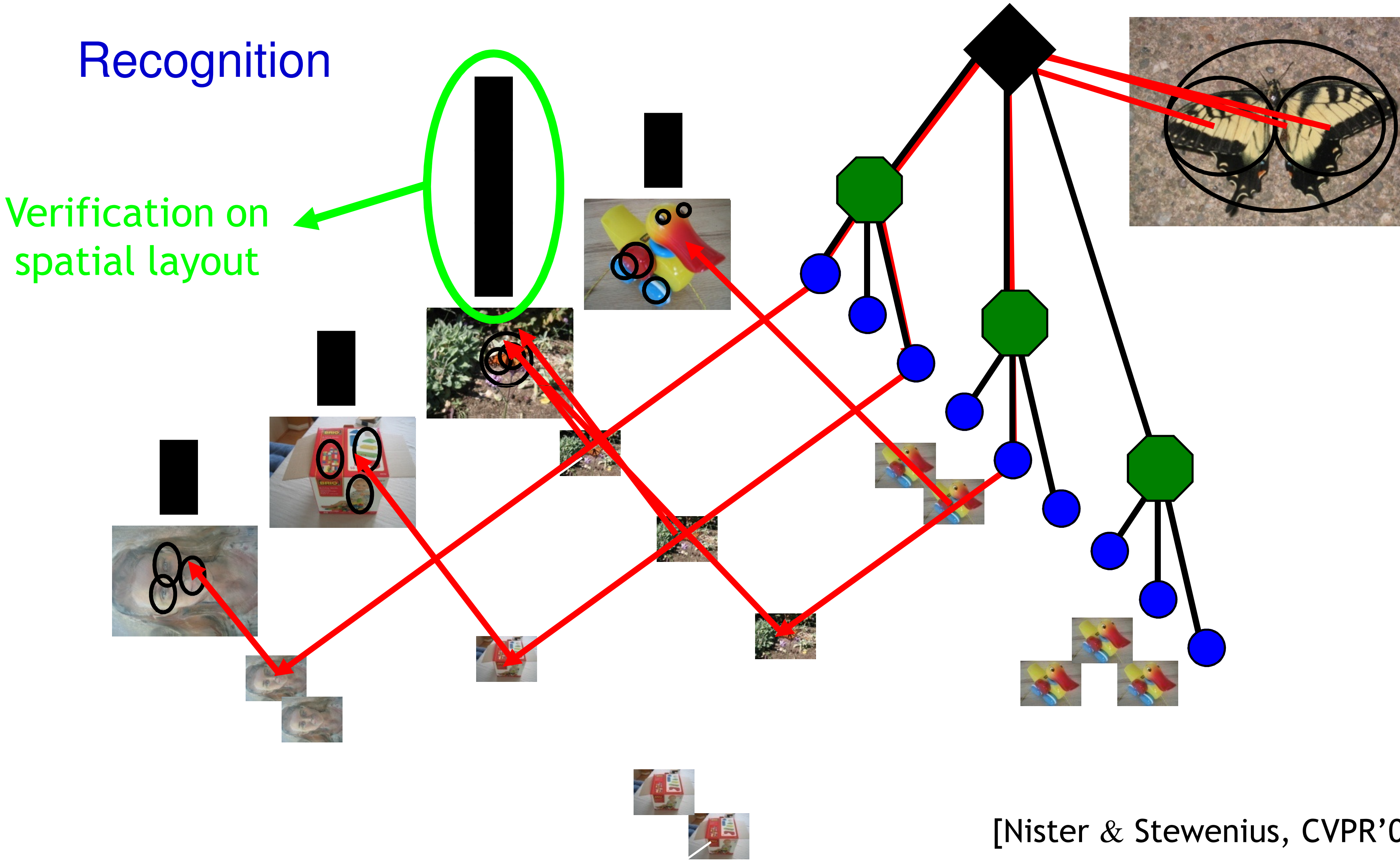


[Nister & Stewenius, CVPR’06]

Slide credit: David Nister

Vocabulary Tree

Recognition



[Nister & Stewenius, CVPR'06]

Slide credit: David Nister

Vocabulary Tree: Performance

Evaluated on large databases

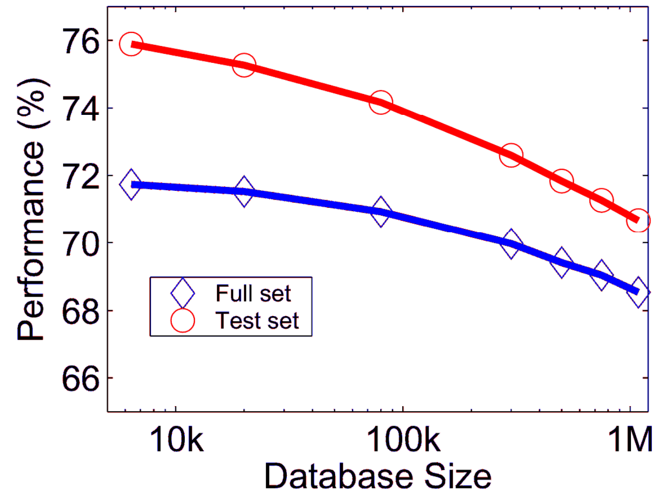
- Indexing with up to 1M images

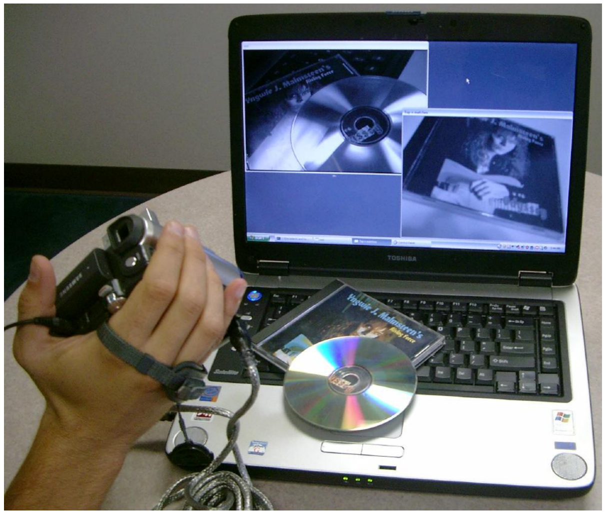
Online recognition for database of 50,000 CD covers

- Retrieval in ~1s

Find experimentally that large vocabularies can be beneficial for recognition

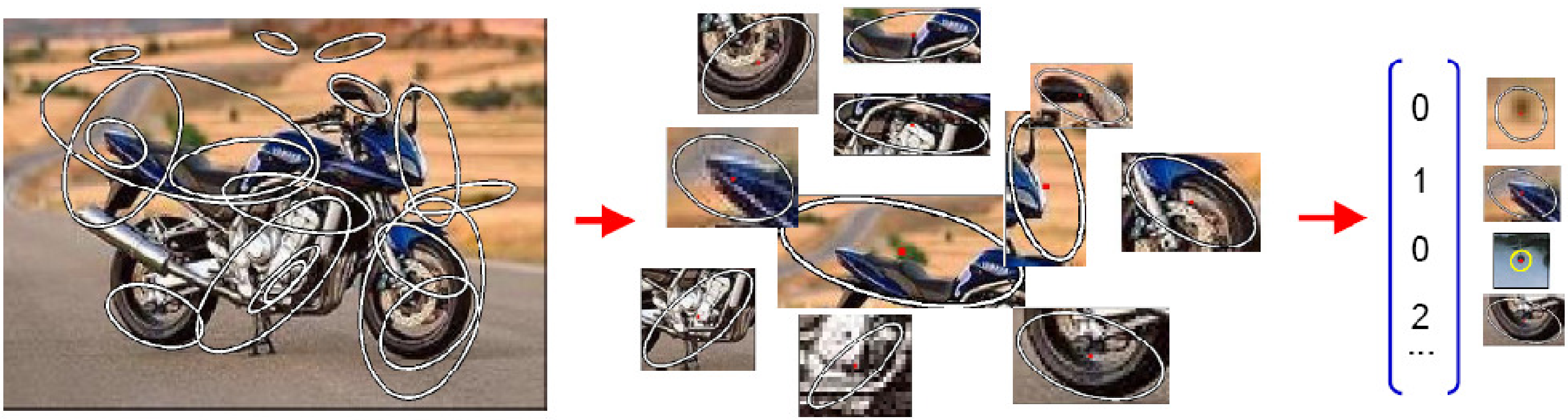
[Nister & Stewenius, CVPR’06]





Slide credit: J. Sivic

“Bag of visual words”



Slide credit: J. Sivic

Today

- Scanning window paradigm
- GIST
- HOG
- Boosted Face Detection
- Local-feature Alignment; from Roberts to Lowe...
- BOW Indexing

Next three lectures

- Thursday: learning object categories from the web
  - LSA and LDA models
  - Harvesting training data from the web
  - Exploiting image and text
- Tues. Oct. 20th: Generative models
  - Condensation
  - ISM
  - Transformed-HDPs
  - More Context…
- Thurs. Oct. 22nd: Advanced BOW kernels
  - Pyramid and spatial-pyramid match
  - Multi-kernel learning
  - Latent-part SVM models

Slide Credits

- As attributed…