# Visual Recognition and Search

January 25, 2008

## Today

- Some logistics
- Overview lecture on recognition models
- Discussion of bag-of-words and constellation model approaches

## Schedule

| Date | Topic | Presenter | Demo | Papers | Items due |
|---|---|---|---|---|---|
| 18-Jan | Course overview | | | | |
| 25-Jan | Background in recognition, local feature models | | | Sivic et al. 2003<br>Fergus et al. 2003 | |
| 1-Feb | Visual vocabularies | Joseph | Xin | Nowak et al. 2006<br>Nister et al. 2006<br>Mossmann et al. 2006 | |
| 8-Feb | Learning about images from keyword-based Web search | David | Dongliang | | |
| 15-Feb | Image and video retargeting | Harshdeep | Adrian | | |
| 22-Feb | Fast indexing methods | Dongliang | Marc (index), Maysam (datasets) | | |
| 29-Feb | Faces; initial proposal discussions | Newton | Harshdeep | | |
| 7-Mar | Text/speech and images/video | Joo Hyun | Sonal | | Project proposal |
| 14-Mar | *Spring break, no class* | | | | |
| 21-Mar | Exploring images in 3d | Maysam | Jaechul | | |
| 28-Mar | Context and background knowledge in recognition | Adrian | Joseph | | |
| 4-Apr | Learning distance functions | Xin | David | | |
| 11-Apr | Detecting abnormal events | Jaechul | *[none]* | | |
| 18-Apr | Place recognition and kidnapped robots | Sonal | Joo Hyun | | Project rough drafts due |
| 25-Apr | Shape matching; discussion of rough draft reviews | Marc | Newton | | Reviews due on the drafts |
| 2-May | Last day of class: project presentations | | | | Final papers |

*Paper reviews are due each week on Thursday by 10 PM.*

## Demo guidelines

Implement/download code for a core idea in the paper and show us toy examples:

- Experiment with different types of (mini) training/testing data sets
- Evaluate sensitivity to parameter settings
- Show (on a small scale) an example in practice that highlights a strength/weakness of the approach
- Want to consider illustrative example, not a system

## Demo presentation format

- Give algorithm, relevant technical details
- Describe scope of experiments
- Present the experiments, explain rationale for outcomes
- Conclude with a summary of the messages

## Timetable for presenters

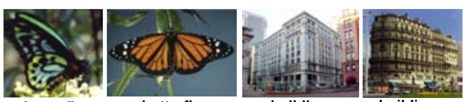
- By the Wednesday the week before:
  - email slides to me, schedule time to meet and discuss.
- Week of:
  - refine slides, practice presentation, know about how long each part requires.
- Day of:
  - send me final slides as PDF file

For Feb 1 and Feb 8 presenters: by upcoming Wednesday and Friday

# Reviews

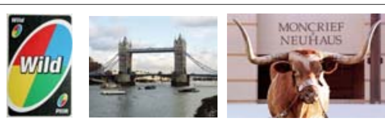
- Submit **one** review per week unless you are presenting (but read all assigned papers)
- Evaluation:
  - 0 none
  - 1 “check –”: little effort/reflection
  - 2 “check”, good review
  - 3 “check+”, very good review

# Possible levels of recognition



butterfly butterfly building building

Categories




Wild card Tower Bridge Bevo

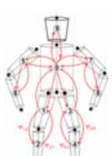
Specific objects

Functional

# Recognition questions

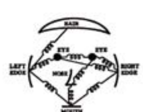
- How to **represent** a category or object
- How to perform the **recognition** (classification, detection) with that representation
- How to **learn** models, new categories/objects

# Representations

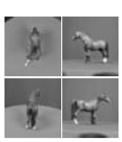


Model-based

Appearance-based



Parts + structure



Multi-view

Bag of features

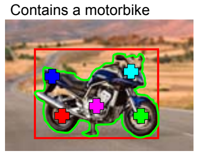
# Learning

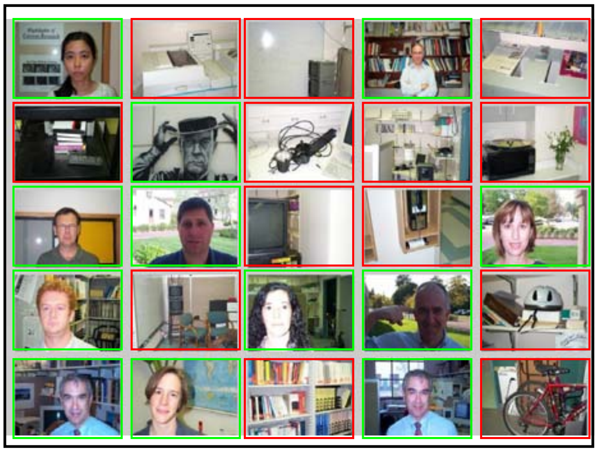
- What defines a category/class?
- What distinguishes classes from one another?
- How to understand the connection between the real world and what we observe?
- What features are most informative?
- What can we do without human intervention?
- Does previous learning experience help learn the next category?

# Learning situations

- Varying levels of supervision
  - Unsupervised
  - Image labels
  - Object centroid/bounding box
  - Segmented object
  - Manual correspondence (typically sub-optimal)

Contains a motorbike





# Inputs/outputs/assumptions

- What **input** is available?
  - Static grayscale image
  - 3D range data
  - Video sequence
  - Multiple calibrated cameras
  - Segmented data, unsegmented data
  - CAD model
  - Labeled data, unlabeled data, partially labeled data

# Inputs/outputs/assumptions

- What is the **goal**?
  - Say yes/no as to whether an object present in image
  - Determine pose of an object, e.g. for robot to grasp it
  - Categorize all objects
  - Forced choice from pool of categories
  - Bounding box on object
  - Full segmentation
  - Build a model of an object category

# Outline

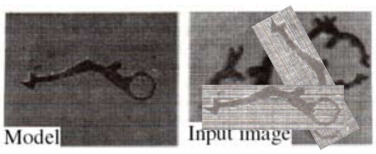
- Overview of recognition background
  - Model-based
  - Appearance-based
  - Local feature-based
    - Features and interest operators
    - Bags of words
    - Constellation models/part-based models

# Model-based recognition

- Which image features correspond to which features on which object model in the “modelbase”?
- If enough match, *and* they match well with a particular transformation for given camera model, then
  - Identify the object as being there
  - Estimate pose relative to camera

## 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.




## How to form a hypothesis?

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

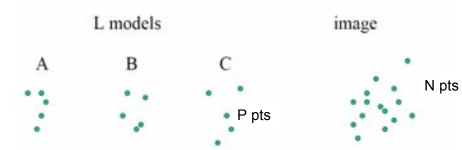
Use correspondence to estimate camera pose relative to object coordinate frame

## 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 P features, N features found in the image




## Generating hypotheses

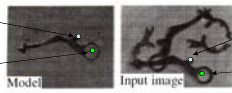
We want a good correspondence between model features and image features.
- Brute force?
- Prune search via geometric or relational constraints: interpretation tree
- Pose consistency: use subsets of features to estimate larger correspondence
- Voting, pose clustering

## 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 (how many depends on camera type)
  - Backproject: transform all model features to image features
  - Verify

# 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).

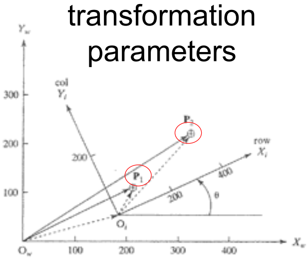
# 2d affine mappings

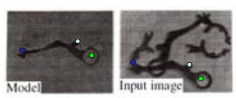
In non-homogenous coordinates

$$\underbrace{\begin{bmatrix} u \\ v \end{bmatrix}}_{\text{[image point]}} = \underbrace{\begin{bmatrix} m_1 & m_2 \\ m_3 & m_4 \end{bmatrix}}_{\text{[scale, rotation, shear]}} \underbrace{\begin{bmatrix} x \\ y \end{bmatrix}}_{\text{[model point]}} + \underbrace{\begin{bmatrix} t_x \\ t_y \end{bmatrix}}_{\text{[translation]}}$$

# Solving for the transformation parameters

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} m_1 & m_2 \\ m_3 & m_4 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$




$\mathbf{P}_1^{(model)} = [200,100]$ $\quad \mathbf{P}_1^{(image)} = [100,60]$

$\mathbf{P}_2^{(model)} = [300,200]$ $\quad \mathbf{P}_2^{(image)} = [380,120]$

$\vdots \qquad \vdots$

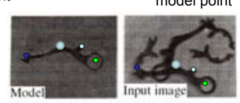
$$\begin{bmatrix} x & y & 0 & 0 & 1 & 0 \\ 0 & 0 & x & y & 0 & 1 \\ & & \cdots & & & \\ & & \cdots & & & \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \\ t_x \\ t_y \end{bmatrix} = \begin{bmatrix} u \\ v \\ \vdots \end{bmatrix}$$

Rewrite in terms of unknown parameters

# Alignment: backprojection

- Having solved for this transformation from some number of detected matches (3+ here), can compute (hypothesized) location of any *other* model points in the image space.

$$\underbrace{\begin{bmatrix} u \\ v \end{bmatrix}}_{\text{image point}} = \begin{bmatrix} m_1 & m_2 \\ m_3 & m_4 \end{bmatrix} \underbrace{\begin{bmatrix} x \\ y \end{bmatrix}}_{\text{model point}} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$




# Alignment: backprojection

Similar ideas for camera models (3d->2d)

- Perspective camera

$$\underbrace{\overline{\mathbf{p}}}_{\text{image coordinates}} = \mathbf{M}\underbrace{\mathbf{P}_w}_{\text{model coordinates}}$$

$$x_{im} = \frac{\mathbf{M}_1 \cdot \mathbf{P}_w}{\mathbf{M}_3 \cdot \mathbf{P}_w} \qquad y_{im} = \frac{\mathbf{M}_2 \cdot \mathbf{P}_w}{\mathbf{M}_3 \cdot \mathbf{P}_w}$$

- Simpler calibration possible with simpler camera models

# Alignment: verification

- Given the backprojected 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?

## Alignment: verification

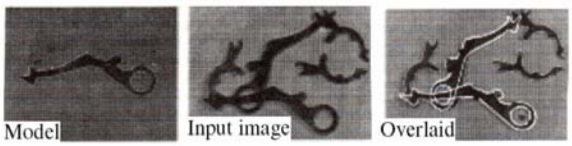
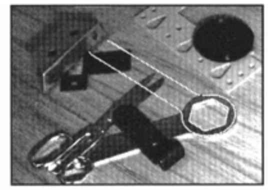



Figure from "Object recognition using alignment," D.P. Huttenlocher and S. Ullman, Proc. Int. Conf. Computer Vision, 1986, copyright IEEE, 1986

## Alignment: verification



Edge-based verification can be brittle

## Pose clustering (voting)

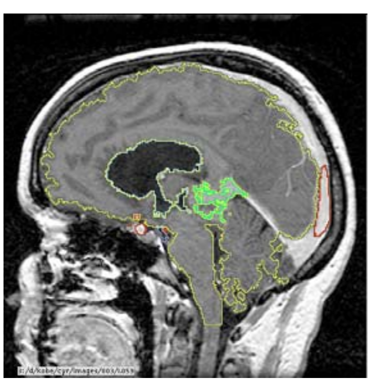
- Narrow down the number of hypotheses to verify: identify those model poses that a lot of features agree on.
  - Use each group's correspondence to estimate pose
  - Vote for that object pose in accumulator array (one array per object if we have multiple models)

## Application: Surgery

- To minimize damage by operation planning
- To reduce number of operations by planning surgery
- To remove only affected tissue
- Problem
  - ensure that the model with the operations planned on it and the information about the affected tissue lines up with the patient
  - display model information supervised on view of patient
  - **Big Issue**: coordinate alignment, as above

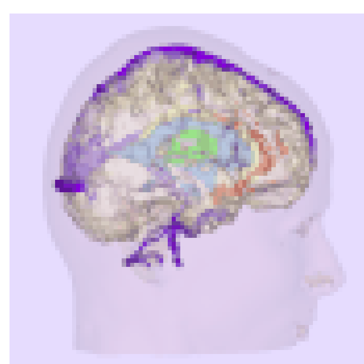


Segmentation used to break single MRI slice into regions.

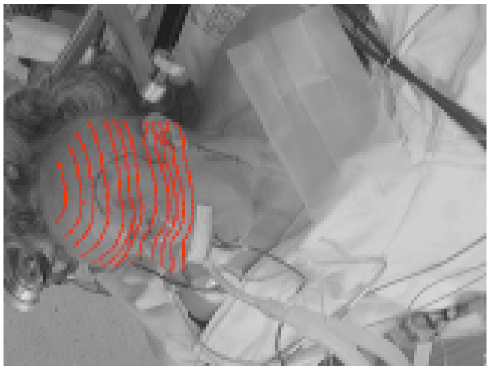


Figures by kind permission of Eric Grimson; http://www.ai.mit.edu/people/welg/welg.html.

Regions assembled into 3d model

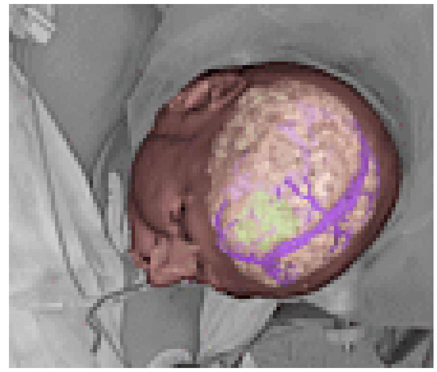


Figures by kind permission of Eric Grimson; http://www.ai.mit.edu/people/welg/welg.html.



Figures by kind permission of Eric Grimson; http://www.ai.mit.edu/people/welg/welg.html.

Patient with model superimposed. Note that view of model is registered to patient's pose here.



Figures by kind permission of Eric Grimson; http://www.ai.mit.edu/people/welg/welg.html.

Figures by kind permission of Eric Grimson; http://www.ai.mit.edu/people/welg/welg.html.

## Summary: model-based recognition

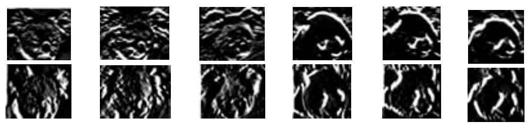
- Hypothesize and test: looking for object and pose that fits well with image
  - Use good correspondences to designate hypotheses
  - Limit verifications performed by voting
- Requires model for the specific objects
  - Searching a modelbase
  - Registration tasks
- Requires camera model selection

*Limits of model-based recognition?*

## Outline

- Overview of recognition background
  - Model-based
  - Appearance-based
  - Local feature-based
    - Features and interest operators
    - Bags of words
    - Constellation models

## Global measure of appearance



– vector of pixel intensities
– grayscale / color histogram
– bank of filter responses ,…

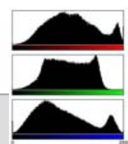
## Global measure of appearance

- e.g., Color histogram

[Swain and Ballard, IJCV 1991]


Slide credit: Stan Sclaroff: http://www.ai.mit.edu/courses/6.801/Fall2002/lect/lect24.pdf

## Color Histograms



Off-line, for each image
    create histogram with a bin for each color
    initialize each bin counter = 0
        for each pixel in image:
            increment bin counter corresponding to pixel color
        end

On-line, use histograms in image similarity measure:
    Euclidean, dot product, histogram intersection, etc.

Slide credit: Stan Sclaroff: http://www.ai.mit.edu/courses/6.801/Fall2002/lect/lect24.pdf

## Images Classified as Sunsets using Overall Color Histograms

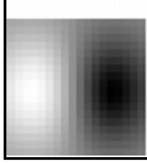

Slide credit: Stan Sclaroff: http://www.ai.mit.edu/courses/6.801/Fall2002/lect/lect24.pdf

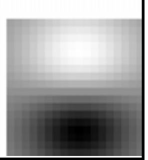
## Global measure of appearance

e.g., responses to linear filters

- Applying filter = taking a dot-product between image and some vector
- Filtering the image is a set of dot products

- Insight
  – filters look like the effects they are intended to find
  – filters find effects they look like

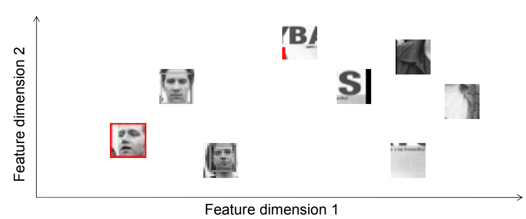




Slide credit: David Forsyth

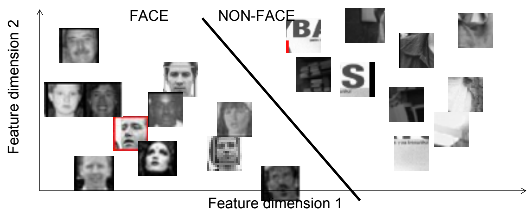
## Learning with global representations

- In addition to sorting images based on nearness in feature space, can learn classifiers

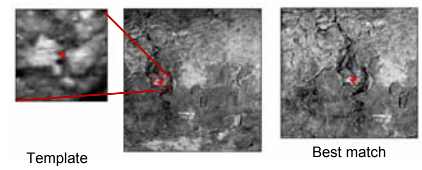


## Learning with global representations

- In addition to sorting images based on nearness in feature space, can learn classifiers



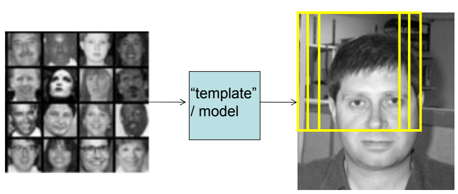

## Windowed search




- Windowed correlation search: to find a fixed scale pattern

## Windowed search

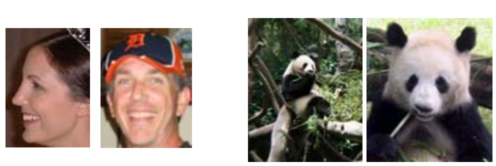
- In general, simple way to check the global measure of appearance when the test image has clutter; search over scales, orientations,…




*When are “global” representations (and window-based detection) appropriate?*

## Limitations of global representations

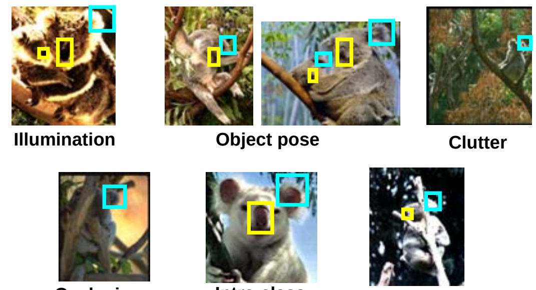
- Success may rely on alignment
- All parts of image impact description



## Outline

- Overview of recognition background
  - – Model-based
  - – Appearance-based
  - – Local feature-based
    - Features and interest operators
    - Bags of words
    - Constellation models

# Local image features

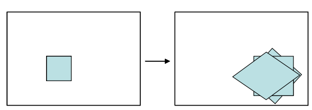


Illumination | Object pose | Clutter

Occlusions | Intra-class appearance | Viewpoint

# Classes of transformations

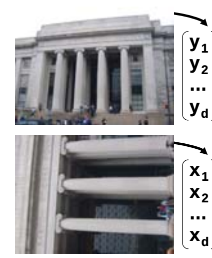
- **Euclidean/rigid**: Translation + rotation
- **Similarity**: Translation + rotation + uniform scale
- **Affine**: Similarity + shear
  – Valid for orthographic camera, locally planar object
- **Photometric**: affine intensity change
  – I -> aI + b



# Invariant local features

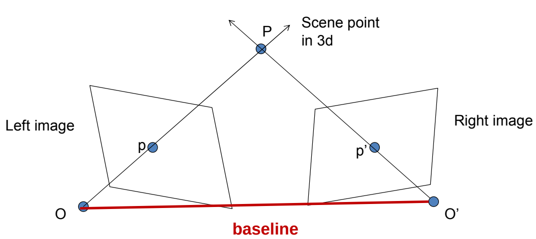
Subset of local feature types designed to be invariant to
- – Scale
- – Translation
- – Rotation
- – Affine transformations
- – Illumination

1) Detect distinctive interest points
2) Extract invariant descriptors




*[Mikolajczyk & Schmid, Matas et al., Tuytelaars & Van Gool, Lowe, Kadir et al.,… ]*

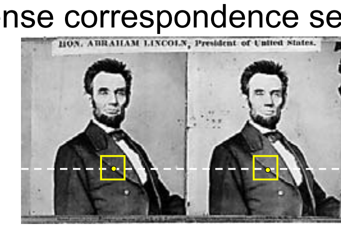
## History of local invariant features…




Estimate scene point based on camera relationships and correspondence.

## History of local invariant features…

# Dense correspondence search




For each epipolar line

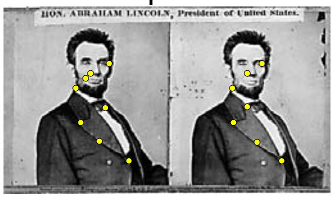
For each pixel / window in the left image

- compare with every pixel / window on same epipolar line in right image
- pick position with minimum match cost (e.g., SSD, correlation)

Adapted from Li Zhang

History of local invariant features…

## Sparse correspondence search




- Restrict search to sparse set of detected features
- Rather than pixel values (or lists of pixel values) use *feature descriptor* and an associated *feature distance*
- Still narrow search further by epipolar geometry

History of local invariant features…

## Wide baseline stereo

- 3d reconstruction depends on finding good correspondences
- Especially with wide-baseline views, local image deformations not well-approximated with rigid transformations
- Cannot simply compare regions of fixed shape (circles, rectangles) – shape is not preserved under affine transformations

## Wide baseline stereo

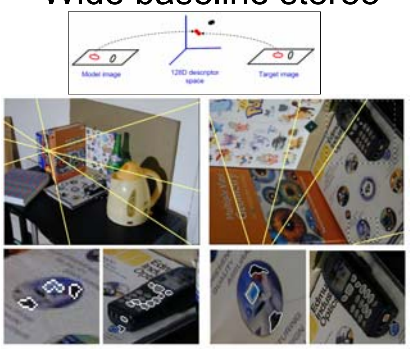



Figure 1: BOOKSHELF: Estimated epipolar geometry on indoor scene with significant scale change. In the cutouts the change in the resolution of detected DRs is clearly visible.

J. Matas, O. Chum, M. Urban, T. Pajdla. Robust Wide Baseline Stereo From Maximally Stable Extremal Regions, BMVC 2002.

## Wide baseline stereo

Figure 2: VALBONNE: Estimated epipolar geometry and points associated to the matched regions are shown in the first row. Cutouts in the second row show matched bricks.

J. Matas, O. Chum, M. Urban, T. Pajdla. Robust Wide Baseline Stereo From Maximally Stable Extremal Regions, BMVC 2002.

## Wide baseline stereo


Figure 3: WASH: Epipolar geometry and dense matched regions with fully affine distortion.

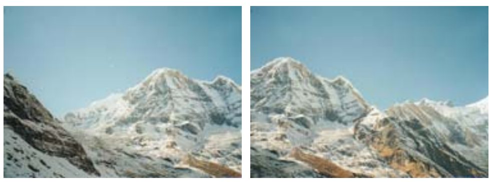
J. Matas, O. Chum, M. Urban, T. Pajdla. Robust Wide Baseline Stereo From Maximally Stable Extremal Regions, BMVC 2002.

## Interest points:
## From stereo to recognition

- Feature detectors previously used for stereo, motion tracking
- Now also for recognition
  - Schmid & Mohr 1997
    - Harris corners to select interest points
    - Rotationally invariant descriptor of local image regions
    - Identify consistent clusters of matched features to do recognition

## Matching with features

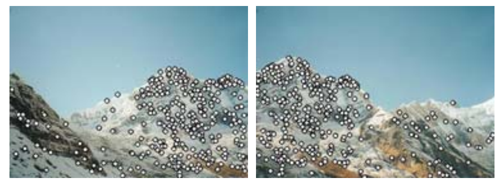
- We need to match (align) images



[These slides are from Darya Frolova and Denis Simakov]

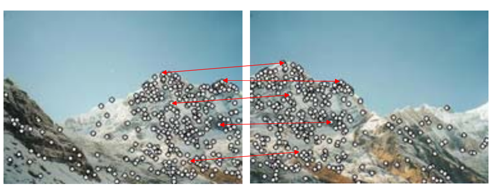
## Matching with Features

- Detect feature points in both images



## Matching with Features

- Detect feature points in both images
- Find corresponding pairs

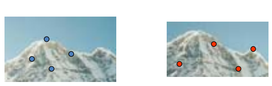


## Matching with Features

- Detect feature points in both images
- Find corresponding pairs

## Matching with Features

- Problem 1:
  - Detect the *same* point *independently* in both images

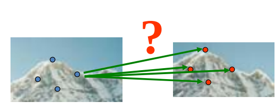


no chance to match!

We need a repeatable detector

## Matching with Features

- Problem 2:
  - For each point correctly recognize the corresponding one




We need a reliable and distinctive descriptor

## (Good) invariant local features

- Reliably detected
- Distinctive
- Robust to noise, blur, etc.
- Description normalized properly

## Exhaustive search

A multi-scale approach

Slide from T. Tuytelaars ECCV 2006 tutorial

## Exhaustive search

A multi-scale approach

Slide from T. Tuytelaars ECCV 2006 tutorial

## Exhaustive search

A multi-scale approach

Slide from T. Tuytelaars ECCV 2006 tutorial

## Exhaustive search

A multi-scale approach

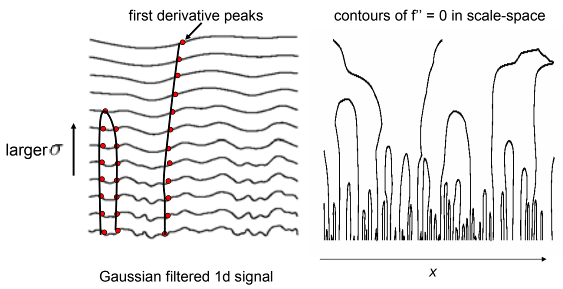
Slide from T. Tuytelaars ECCV 2006 tutorial

## Key idea of invariance

We want to extract the patches from each image *independently*: features should adapt their shape, *covariant* with the affine transformation relating them.

Slide adapted from T. Tuytelaars ECCV 2006 tutorial

## Scale space (Witkin 83)




Adapted from Steve Seitz, UW

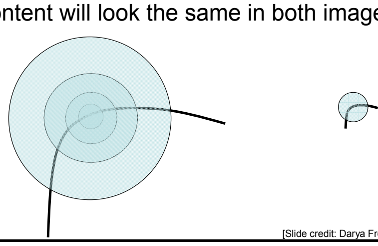
## Scale space

Scale space insights:

- edge position may shift with increasing scale ($\sigma$)
- two edges may merge with increasing scale (edges can disappear)
- an edge may ***not*** split into two with increasing scale (new edges do not appear)

## Scale Invariant Detection

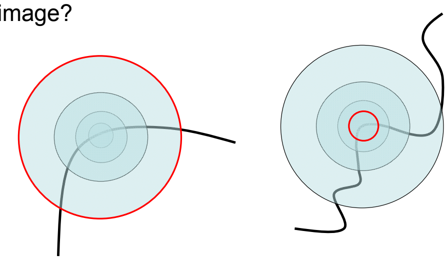
- Consider regions of different sizes around a point
- At the right scale, regions of corresponding content will look the same in both images



[Slide credit: Darya Frolova and Denis Simakov]

## Scale Invariant Detection

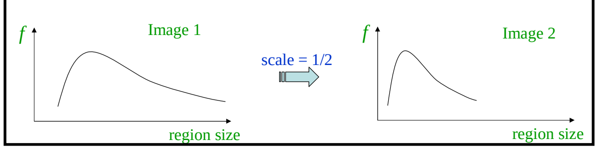
- The problem: how do we choose corresponding circles ***independently*** in each image?



## Scale Invariant Detection

- Solution:
  - Design a function on the region (circle), which is "scale invariant" (*the same for corresponding regions, even if they are at different scales*)

    Example: average intensity. For corresponding regions (even of different sizes) it will be the same.
  - For a point in one image, we can consider it as a function of region size (circle radius)



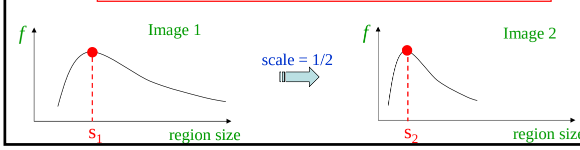

## Scale Invariant Detection

- Common approach:

  Take a local maximum of this function

  Observation: region size, for which the maximum is achieved, should be *invariant* to image scale.

Important: this scale invariant region size is found in each image independently!



## Scale Invariant Detection

[Images from T. Tuytelaars]

## Automatic scale selection

Lindeberg et al., 1996


**Following example was created by T. Tuytelaars, ECCV 2006 tutorial**

## Automatic scale selection


## Automatic scale selection

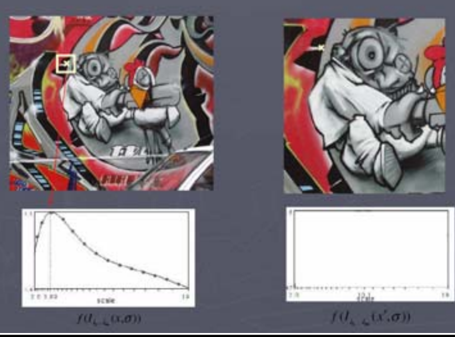

## Automatic scale selection


## Automatic scale selection

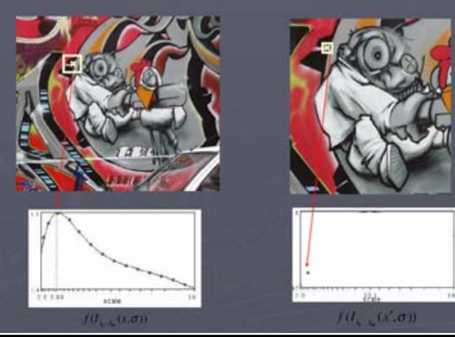
## Automatic scale selection


## Automatic scale selection

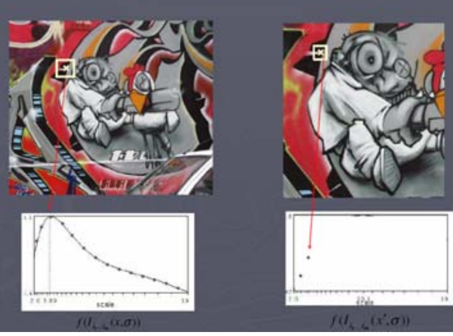

## Automatic scale selection



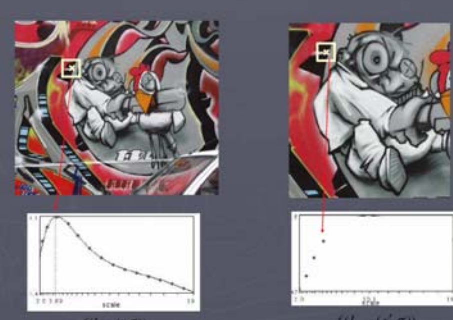

## Automatic scale selection

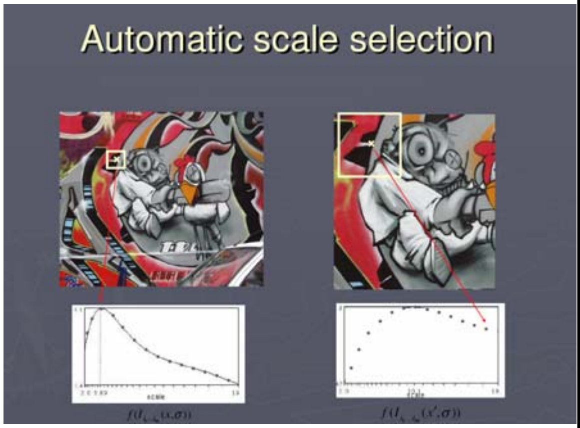


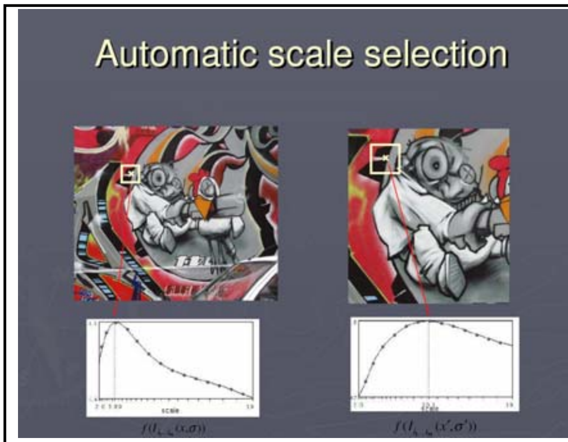

## Automatic scale selection




## Automatic scale selection

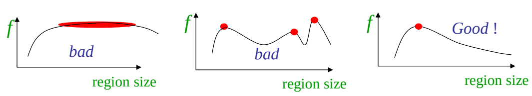





## Scale Invariant Detection

- A “good” function for scale detection:
  has one stable sharp peak




- For usual images: a good function would be a one which responds to contrast (sharp local intensity change)

## Scale selection principle

- Intrinsic scale is the scale at which normalized derivative assumes a maximum -- marks a feature containing interesting structure. (T. Lindeberg ’94)

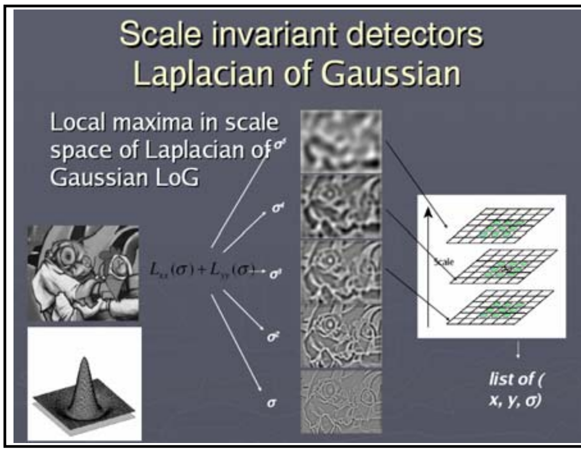
➔ Maxima/minima of Laplacian

## Scale invariant detection

Requires a method to repeatably select points in location and scale:

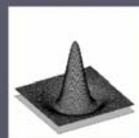
- Only reasonable scale-space kernel is a Gaussian (Koenderink, 1984; Lindeberg, 1994)
- An efficient choice is to detect peaks in the difference of Gaussian pyramid (Burt & Adelson, 1983; Crowley & Parker, 1984)
- Difference-of-Gaussian is a close approximation to Laplacian

Slide adapted from
David Lowe



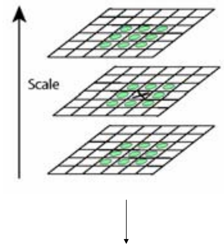
## Lowe's DoG

Difference of Gaussians as approximation of the Laplacian of Gaussian



## SIFT: Key point localization

- Detect maxima and minima of difference-of-Gaussian in scale space
- Then reject points with low contrast (threshold)
- Eliminate edge responses (use ratio of principal curvatures)




Candidate keypoints: list of (x,y,σ)

## SIFT: Example of keypoint detection

Threshold on value at DOG peak and on ratio of principle curvatures

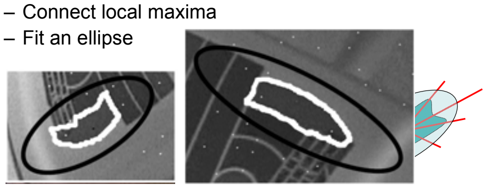

**(a)** 233x189 image
**(b)** 832 DOG extrema
**(c)** 729 left after peak value threshold
**(d)** 536 left after testing ratio of principle curvatures

## Scale Invariant Detection: Summary

- Given: two images of the same scene with a large *scale difference* between them
- Goal: find *the same* interest points *independently* in each image
- Solution: search for *maxima* of suitable functions in *scale* and in *space* (over the image)

## Affine Invariant Detection

- Intensity-based regions (IBR):
  - Start from a local intensity extrema
  - Consider intensity profile along rays
  - Select maximum of invariant function f(t) along each ray
  - Connect local maxima
  - Fit an ellipse



T.Tuytelaars, L.V.Gool. "Wide Baseline Stereo Matching Based on Local, Affinely Invariant Regions". BMVC 2000.

## Affine Invariant Detection

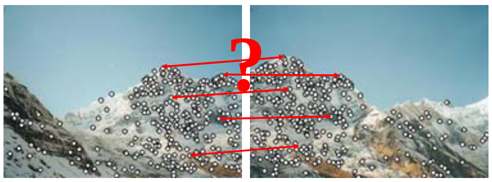
- Maximally Stable Extremal Regions (MSER)
  - *Threshold* image intensities: $I > I_0$
  - Extract *connected components* ("Extremal Regions")
  - Seek extremal regions that remain "Maximally Stable" under range of thresholds

Matas et al. Robust Wide Baseline Stereo from Maximally Stable Extremal Regions. BMVC 2002.

# Point Descriptors

- We know how to detect points
- Next question:

**How to *describe* them for matching?**

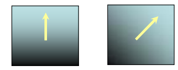



Point descriptor should be:
1. Invariant
2. Distinctive

# Rotation Invariant Descriptors

- Find local orientation

Dominant direction of gradient

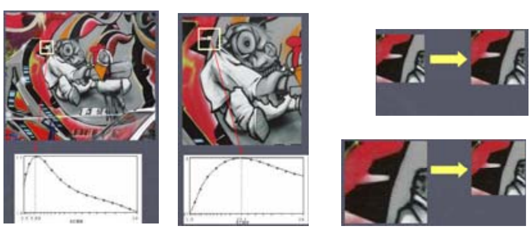


- Rotate description relative to dominant orientation

[1] K.Mikolajczyk, C.Schmid. "Indexing Based on Scale Invariant Interest Points". ICCV 2001
[2] D.Lowe. "Distinctive Image Features from Scale-Invariant Keypoints". Accepted to IJCV 2004

# Scale Invariant Descriptors

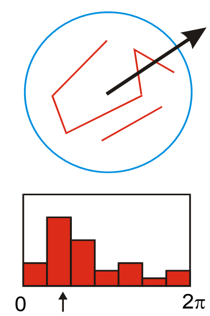
- Use the scale determined by detector to compute descriptor in a normalized frame



[Images from T. Tuytelaars]

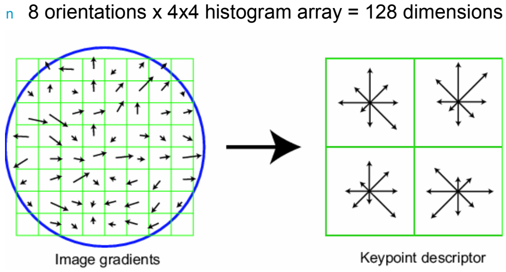
## SIFT descriptors: Select canonical orientation

- Create histogram of local gradient directions computed at selected scale
- Assign canonical orientation at peak of smoothed histogram
- Each key specifies stable 2D coordinates (x, y, scale, orientation)




## SIFT descriptors: vector formation

- Thresholded image gradients are sampled over 16x16 array of locations in scale space
- Create array of orientation histograms
- 8 orientations x 4x4 histogram array = 128 dimensions




# SIFT properties

- Invariant to
  - Scale
  - Rotation
- Partially invariant to
  - Illumination changes
  - Camera viewpoint
  - Occlusion, clutter

*Revisiting model-based recognition with more powerful features:*

## Recognition with SIFT [Lowe]

1) Index descriptors (distinctive features narrow possible matches)
2) 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 (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.

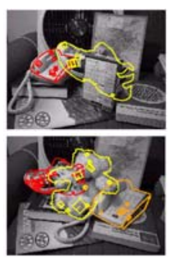
Recognition result

[Lowe]

## 3d objects

Background subtract for model boundaries

Objects recognized, though affine model not as accurate.



Recognition in spite of occlusion

[Lowe]

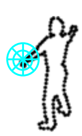
## Value of local (invariant) features

- Complexity reduction via selection of distinctive points
- Describe images, objects, parts without requiring segmentation
  - Local character means robustness to clutter, occlusion
- Robustness: similar descriptors in spite of noise, blur, etc.

## Local representations

Describe component regions or patches separately

SIFT [Lowe]



Shape context [Belongie et al.]

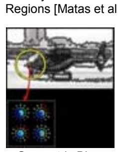
Superpixels [Ren et al.]

Maximally Stable Extremal Regions [Matas et al.]

Salient regions [Kadir et al.]

Harris-Affine [Schmid et al.]

Spin images [Johnson and Hebert]



Geometric Blur [Berg et al.]

*Local features will be something we can match across images…*

*What possible models for objects and categories can be formed with local descriptors as the basis?*

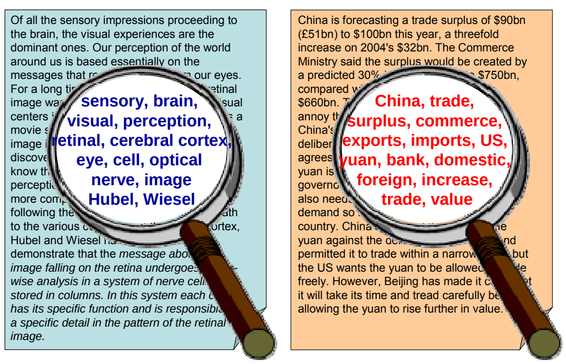
# Outline

- Overview of recognition background
  - Model-based
  - Appearance-based
  - Local feature-based
    - Features and interest operators
    - Bags of words
    - Constellation models

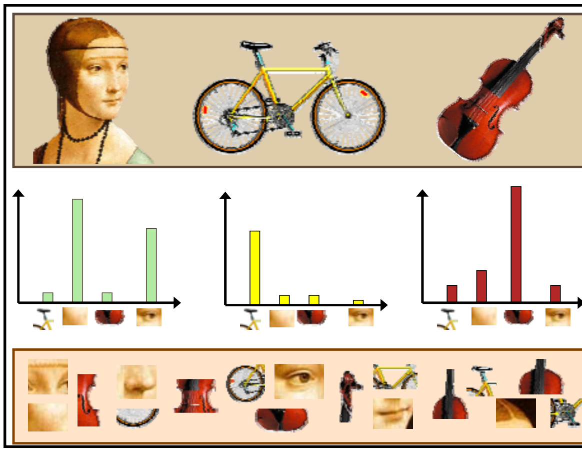
**Object** → **Bag of 'words'**

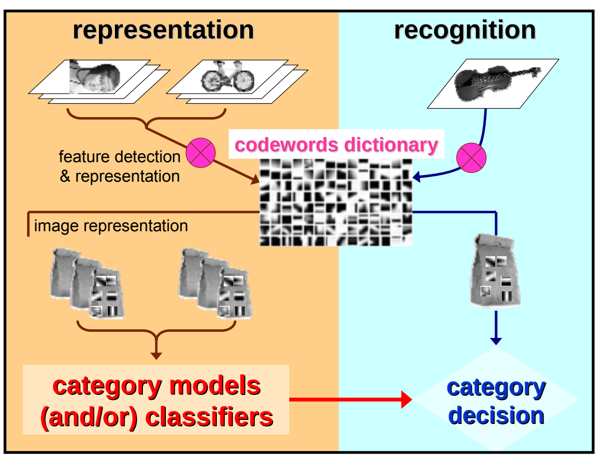
ICCV 2005 short course, L. Fei-Fei

## Analogy to documents



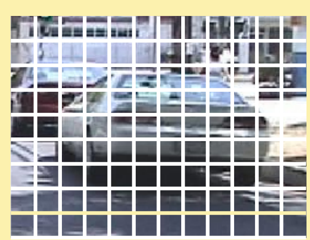

ICCV 2005 short course, L. Fei-Fei





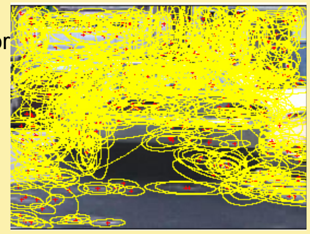

## 1.Feature detection and representation

- Regular grid



## 1.Feature detection and representation

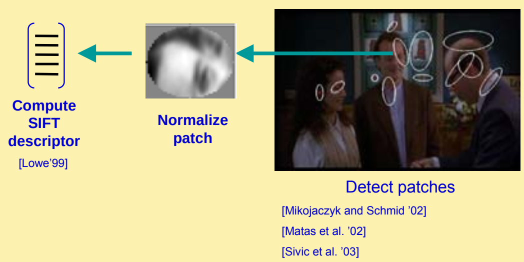
- Regular grid
- Interest point detector



## 1.Feature detection and representation

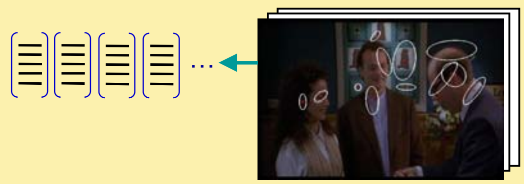
- Regular grid
- Interest point detector
- Other methods
  - Random sampling
  - Segmentation based patches

## 1.Feature detection and representation

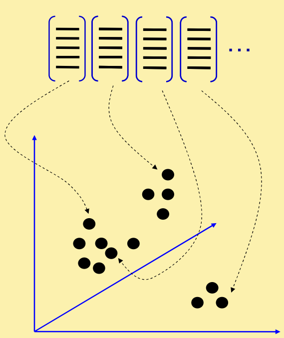



Slide credit: Josef Sivic

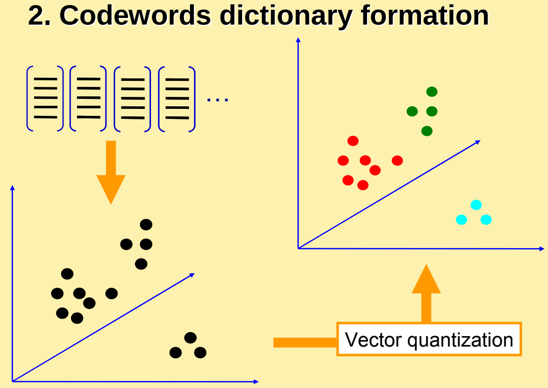
## 1.Feature detection and representation



## 2. Codewords dictionary formation

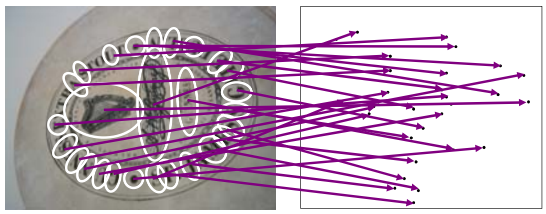


## 2. Codewords dictionary formation



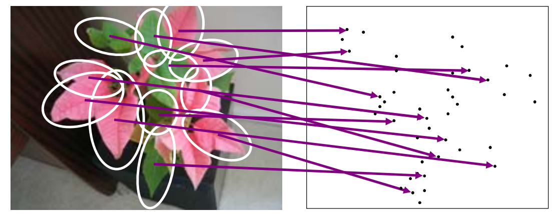

Slide credit: Josef Sivic

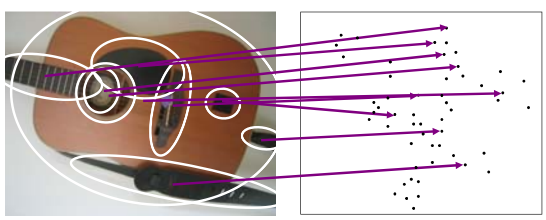
Extract some local features from a number of images …

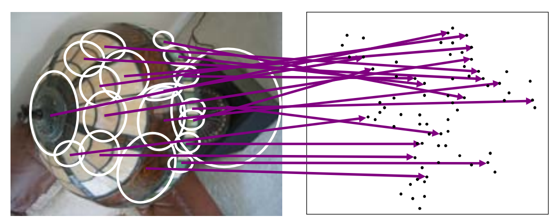


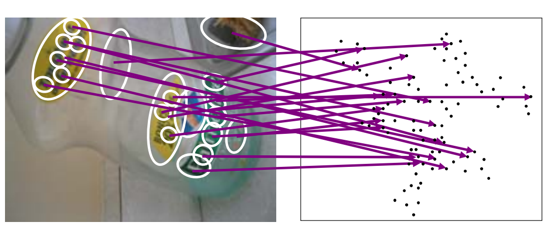
SIFT descriptor space: each point is 128-dimensional

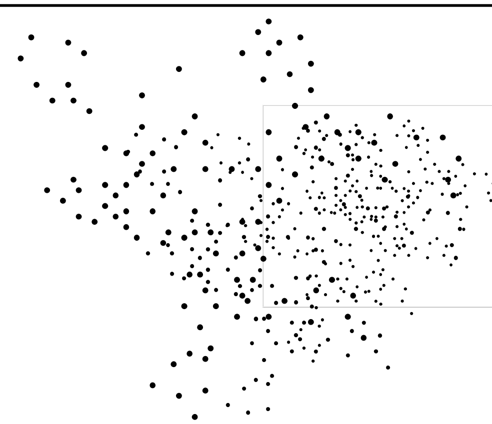
Slides from D. Nister

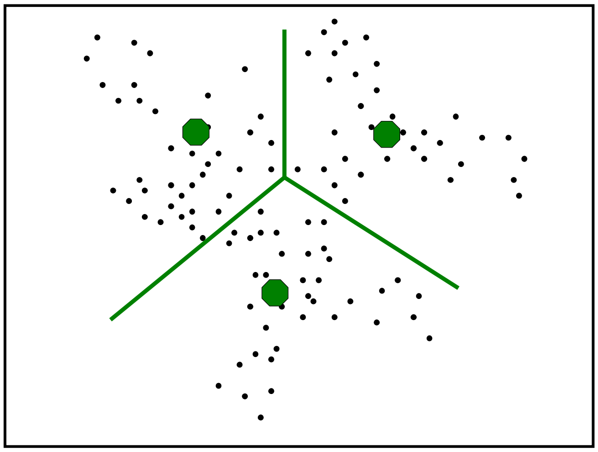




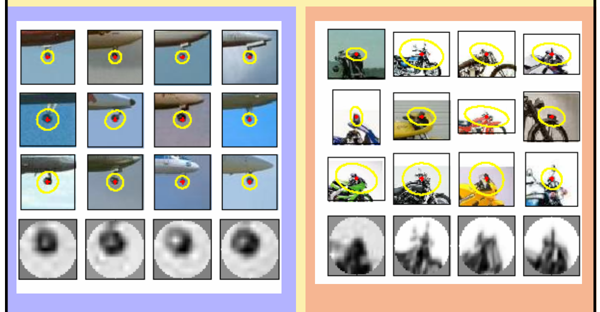






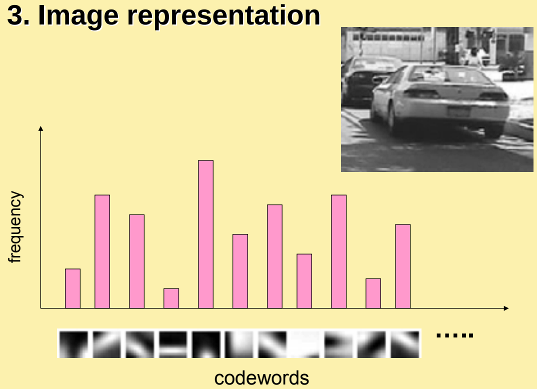


## Image patch examples of codewords



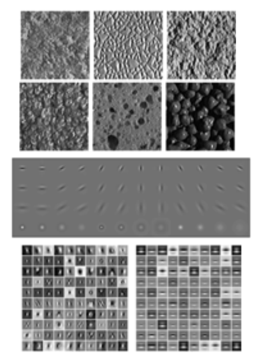
Sivic et al. 2005

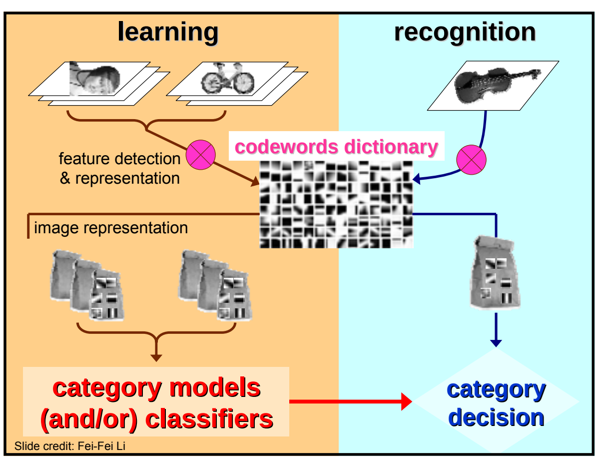
## 3. Image representation




## Visual words = textons

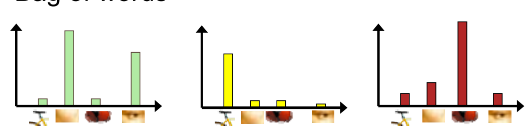
- Previous use of local feature quantization:
- *Texton* = cluster center of filter responses over collection of images [Leung & Malik, 1999; Varma & Zisserman 2002]
- Represent texture or material with histogram of texton occurrences (or prototypes of whatever feature type employed)



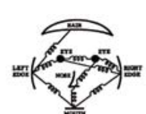



## Today's papers: two general ways to build a representation from local features

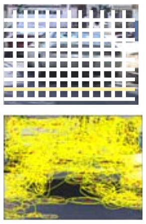
- Bag of words



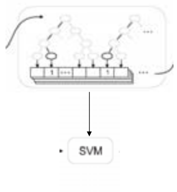
- Constellation models



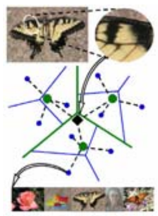
## Next time: visual vocabularies



Interest operators, sampling strategy




Quantization strategy



Search, indexing structures

## Next time

- Topic: visual vocabularies
- Presenter: Joseph
- Demo: Xin
- Papers to read (review one):
  - Sampling Strategies for Bag-of-Features Image Classification. E. Nowak, F. Jurie, and B. Triggs. ECCV, 2006.
  - Fast Discriminative Visual Codebooks using Randomized Clustering Forests, by A. Moosmann, B. Triggs and F. Jurie. NIPS, 2006.
  - Scalable Recognition with a Vocabulary Tree, by D. Nister and H. Stewenius. CVPR, 2006.