Deep Learning for Computer Vision

Spring 2018

http://vllab.ee.ntu.edu.tw/dlcv.html (primary)

https://ceiba.ntu.edu.tw/1062DLCV (grade, etc.)

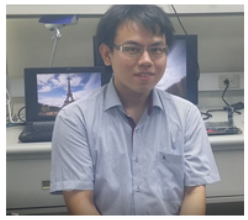
FB: DLCV Spring 2018

Yu-Chiang Frank Wang 王鈺強, Associate Professor

Dept. Electrical Engineering, National Taiwan University

2018/03/21

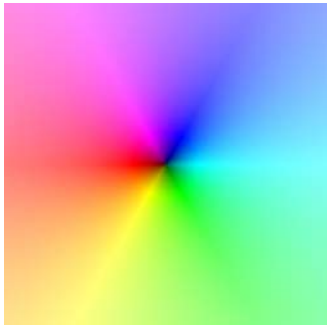
Updates of TA Hours/Locations

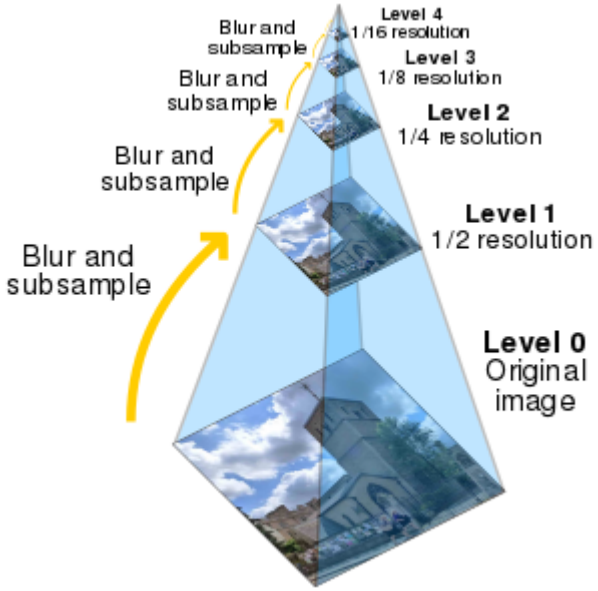
Mon	Tue	Wed	Thu	Fri
 Yu-Jhe Li 11:00-12:00 (BL-527)	 Yan-Bo Lin 14:20-15:10 (BL-527)	 Alex Liu 13:20-14:20 (BL-527)	 Chi-Hsin Lo 10:00-11:00 (BL-B1)	 Keng-Sen Tseng 13:20-14:20 (BL-B1)
 Huai-Jin Peng 13:20-14:20 (BL-B1)	 Hsuan-I Ho 15:30-17:20 (BL-527)	 Yao-Cheng Yang 14:30-15:30 (BL-527)	 Cheng-Yen Yang 13:30-14:30 (BL-B1)	
 Jia-Wei Yan 15:10-16:10 (BL-527)	 Frank Liao 18:30-19:30 (BL-527)	 Jason Kuo 18:00-19:00 (BL-B1)	 Yu-Ying Yeh 14:30-15:30 (BL-527)	

- FB: **DLCV Spring 2018**; feedback are welcome (to TA/me).

What Hasn't Be Covered Yet…

- Image Representation
  - Color Space
  - Image Filtering
  - Image Pyramid

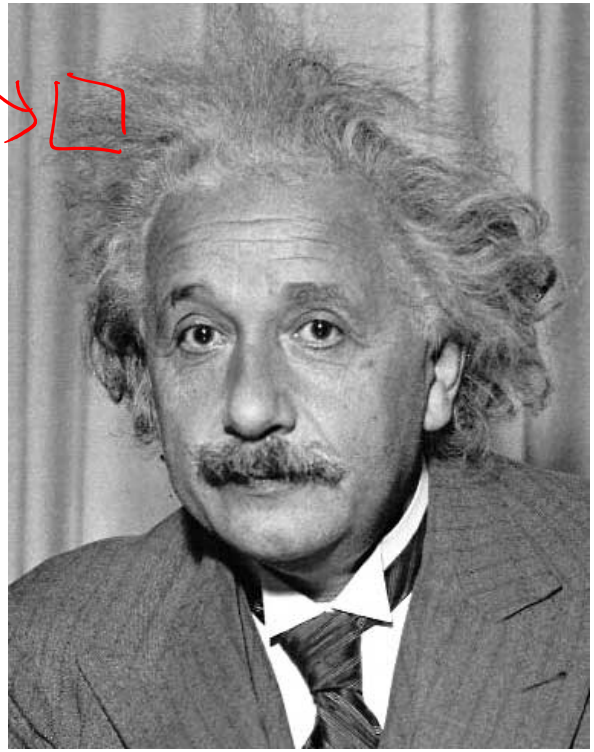




Many slides from J.-B. Huang, J. Hayes, N Snavely, K. Grauman, and D. Hoiem

From Template Matching to Image Pyramid

- Goal: find  in an image
- Main challenge: what is a good similarity or distance metric for measuring between two patches? D( , )
  - Cross-Correlation?
  - Normalized cross correlation?
  - Zero-mean correlation?
  - Sum of Square Difference (SSD)?



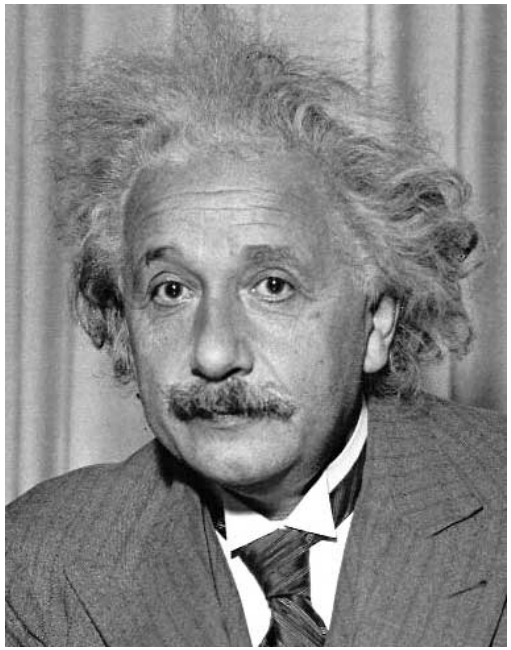
From Template Matching to Image Pyramid

- Goal: find  in an image
- Method #1: filter the image with the eye patch

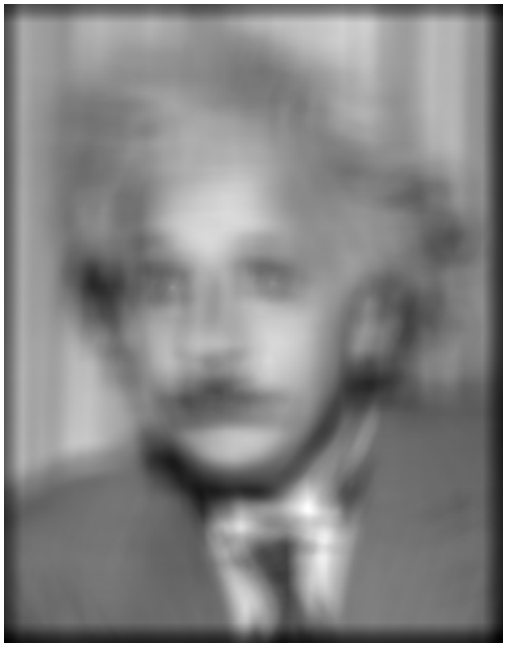
$$h[m,n] = \sum_{k,l} g[k,l]\, f[m+k, n+l]$$

g = filter f = image

- What went wrong?



Input



Filtered Image

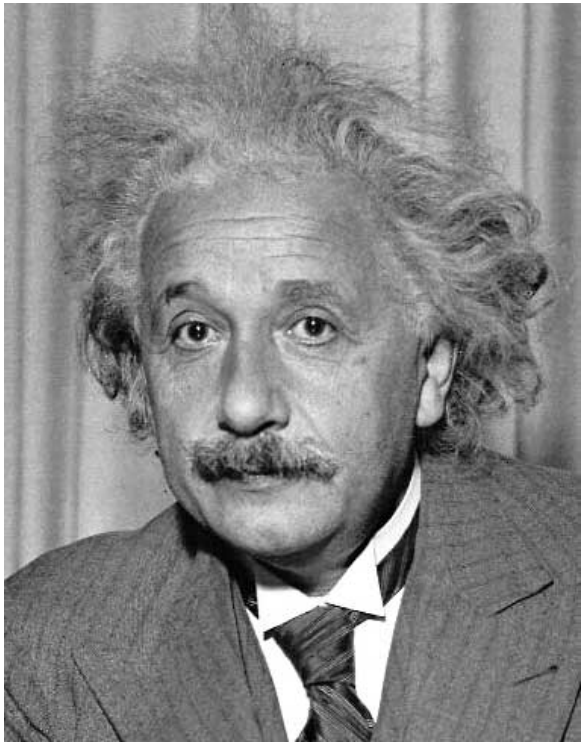
From Template Matching to Image Pyramid

- Goal: find  in an image
- Method #2: filter the image with the zero-mean eye patch

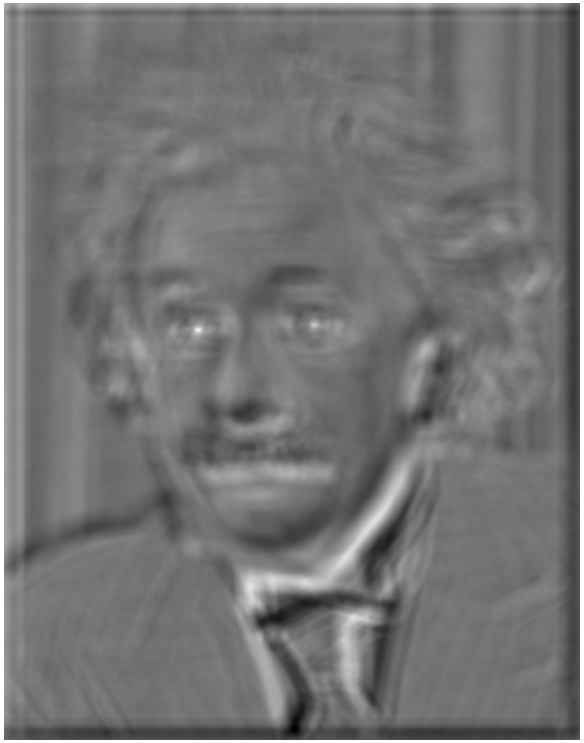
$$h[m,n] = \sum_{k,l} (g[k,l] - \bar{g})\,(f[m+k, n+l])$$

g = filter f = image

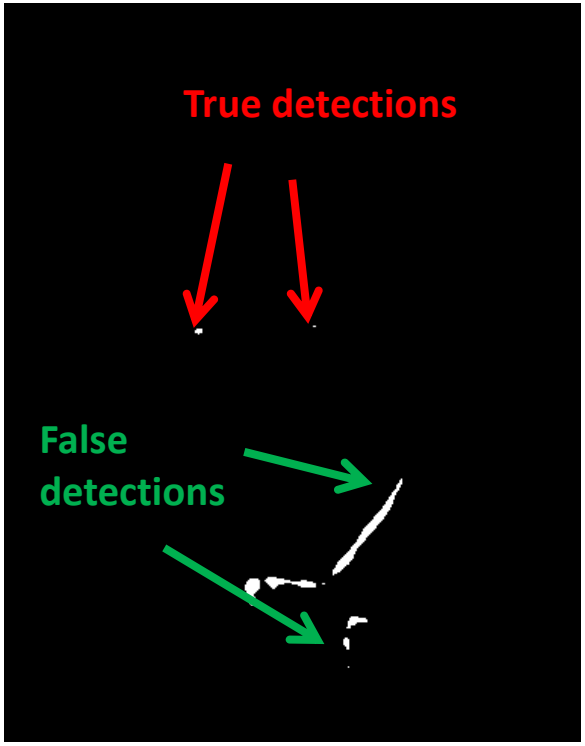
- A little bit better…



Input



Filtered Image (scaled)

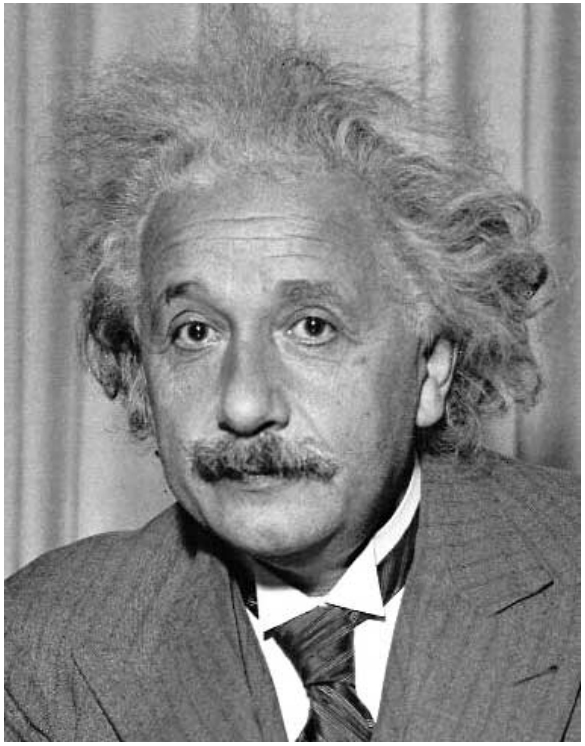


Thresholded Image

From Template Matching to Image Pyramid

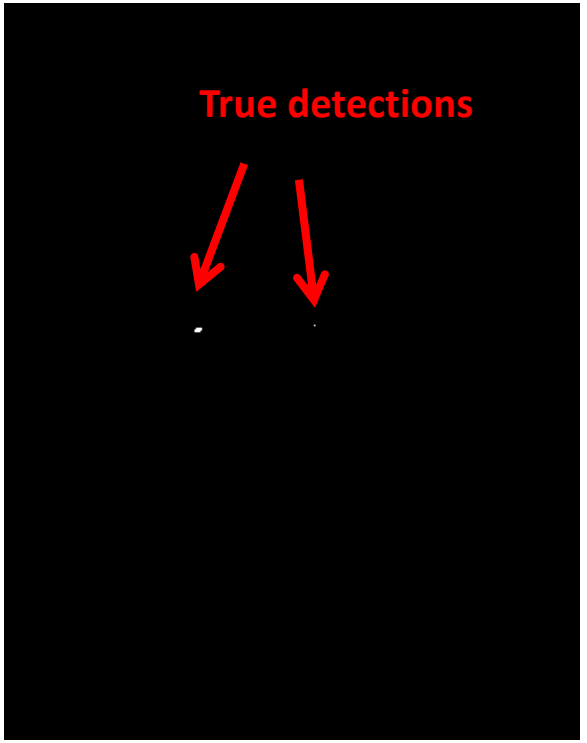
- Goal: find  in an image
- Method #3: SSD $$h[m,n] = \sum_{k,l} (g[k,l] - f[m+k,n+l])^2$$

  g = filter f = image
- Much better!?



Input

1- sqrt(SSD)



Thresholded Image

Matching with Filters

- Can SSD be implemented with linear filters? And why?

$$h[m,n] = \sum_{k,l} (g[k,l] - f[m+k,n+l])^2$$

$$h[m,n] = \sum_{k,l} (g[k,l])^2 - 2\sum_{k,l} g[k,l] f[m+k,n+l] + \sum_{k,l} (f[m+k,n+l])^2$$

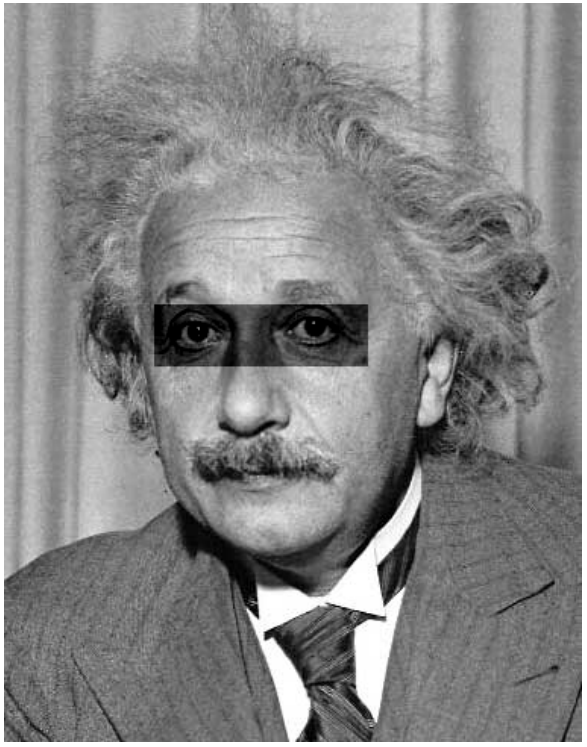
Constant

Filtering with g

Filtering with box filter

Matching with Filters

- Can SSD be implemented with linear filters? And why?
- Method #3: SSD using  …What's the potential problem?



Input

SSD Output

From Template Matching to Image Pyramid

- Goal: find  in an image
- Method #4: Normalized cross-correlation

mean template

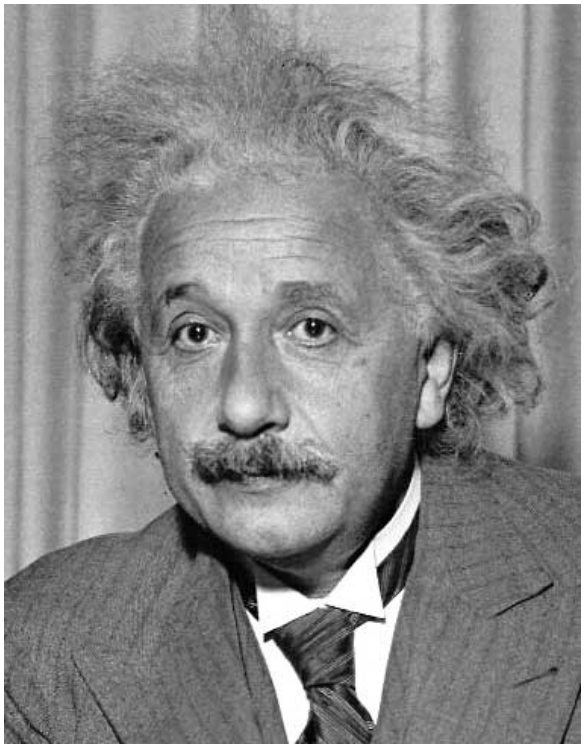
mean image patch

$$h[m,n] = \frac{\sum_{k,l}(g[k,l]-\bar{g})(f[m+k,n+l]-\bar{f}_{m,n})}{\left(\sum_{k,l}(g[k,l]-\bar{g})^2 \sum_{k,l}(f[m+k,n+l]-\bar{f}_{m,n})^2\right)^{0.5}}$$

Matlab: `normxcorr2(template, im)`

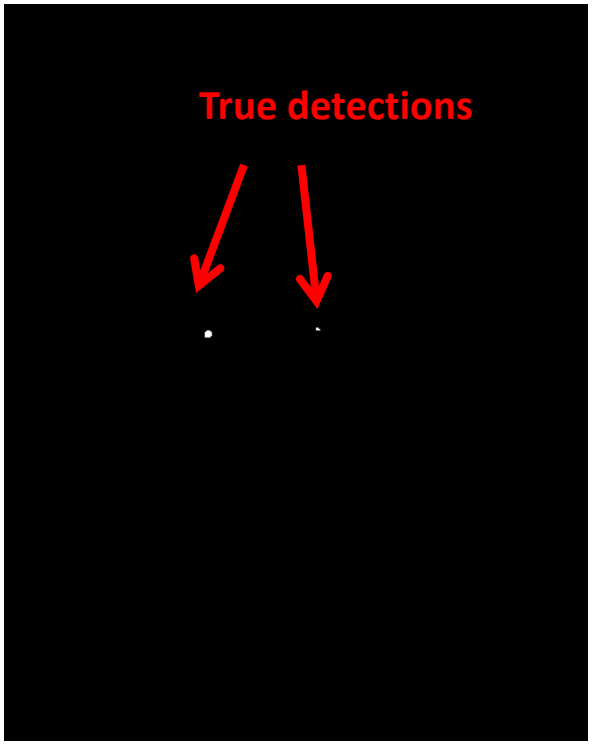
From Template Matching to Image Pyramid

- Goal: find  in an image
- Method #4: Normalized cross-correlation



Input

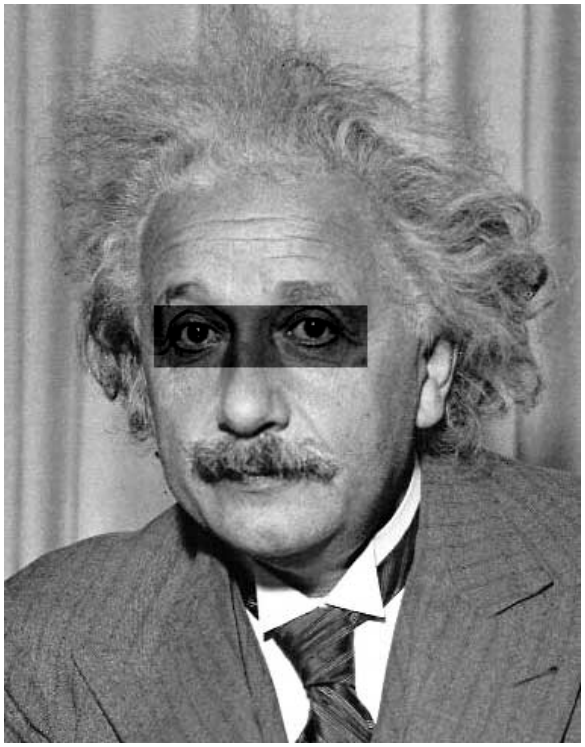
Normalized X-Correlation



Thresholded Image

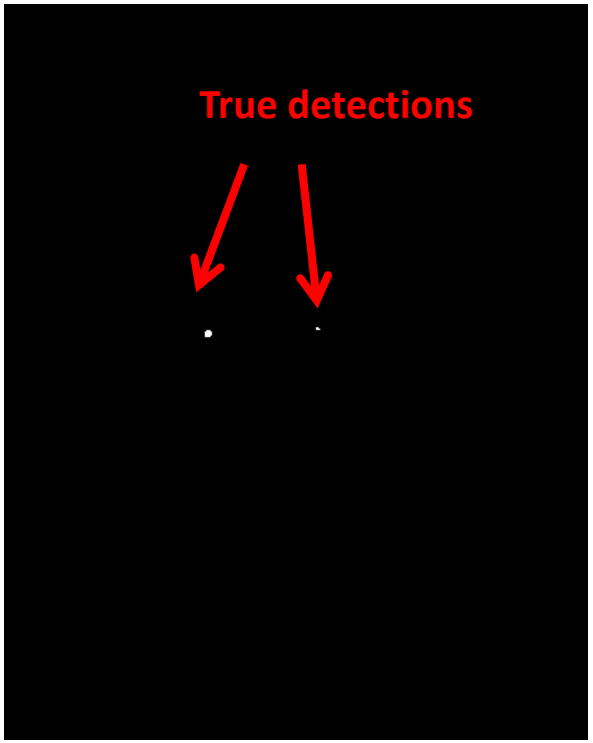
From Template Matching to Image Pyramid

- Goal: find  in an image
- Method #4: Normalized cross-correlation



Input

Normalized X-Correlation



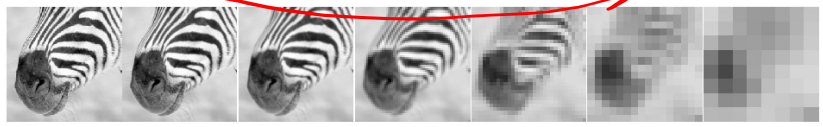
Thresholded Image

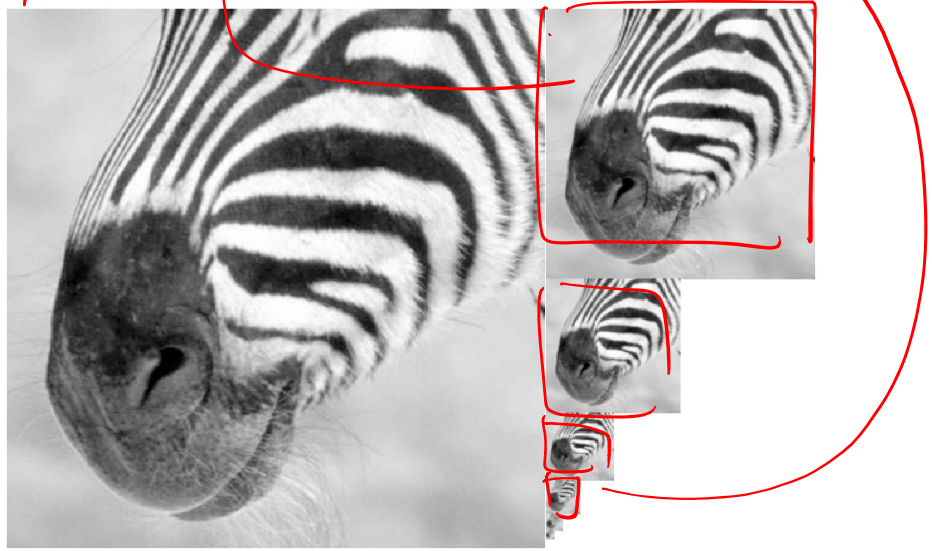
Who is the Winner?

- Answer: It depends…
- Remarks:
  - Fastest but not robust: zero-mean cross-correlation
  - Next fastest but sensitive to overall intensity: SSD
  - Slowest though invariant to local average intensity/contrast: norm cross-correlation

Size Matters…

- Question: What if we need to find larger or smaller eyes?
- Answer: Image Pyramid! Why?
- Example: Gaussian pyramid (Why LPF?)

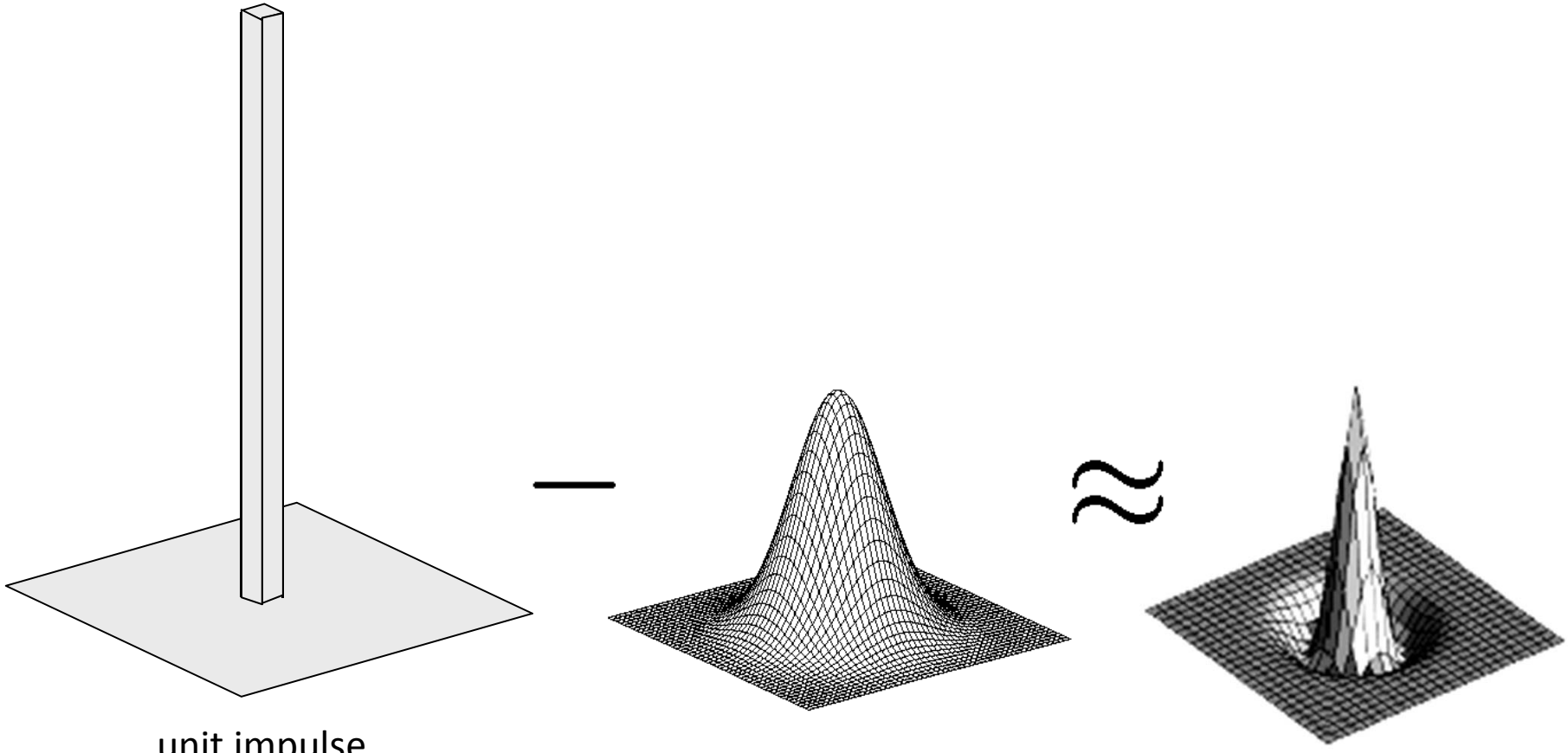




Wagon-Wheel Effect

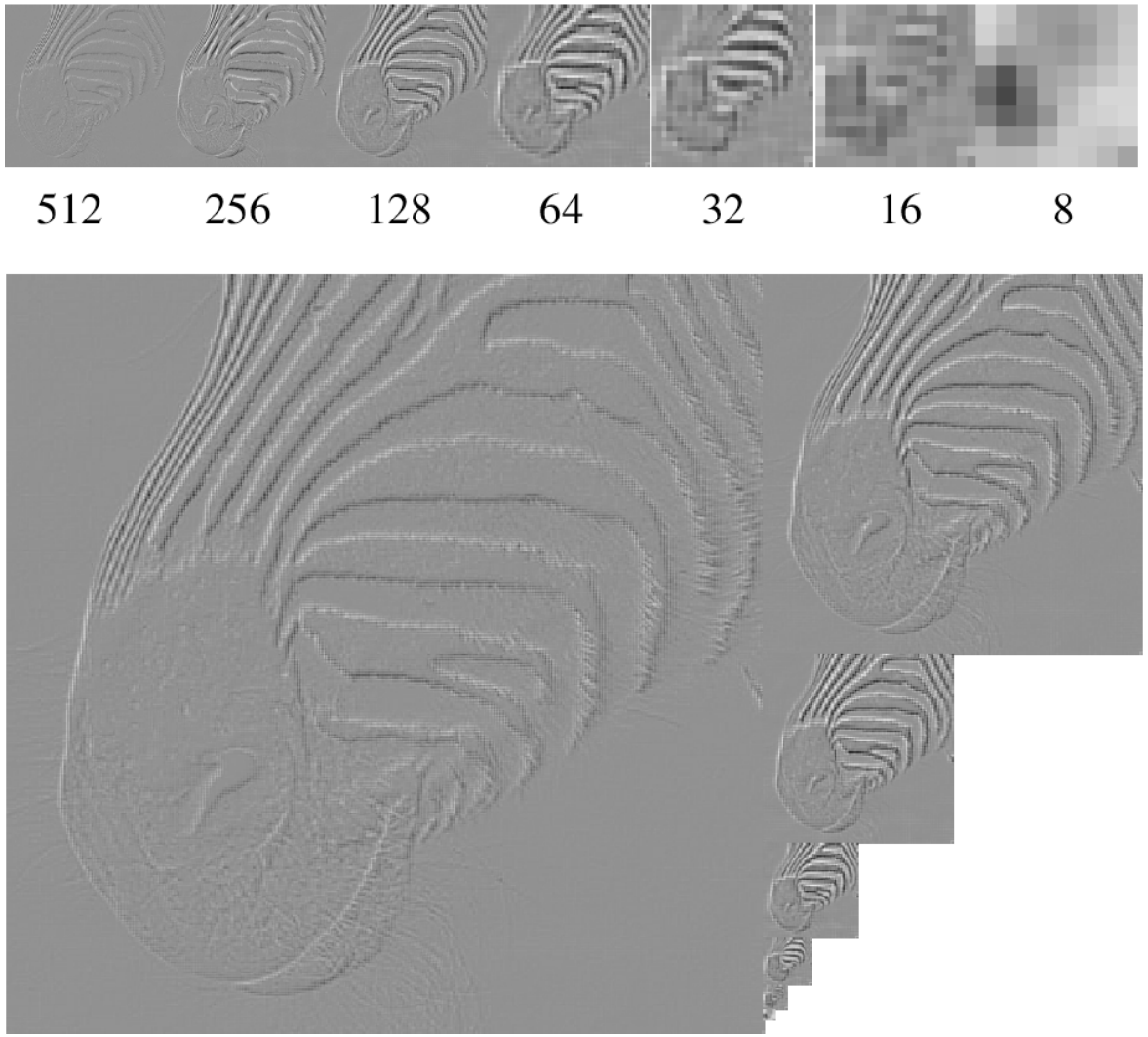
https://www.youtube.com/watch?v=QOwzkND_ooU

Example: Laplacian filter



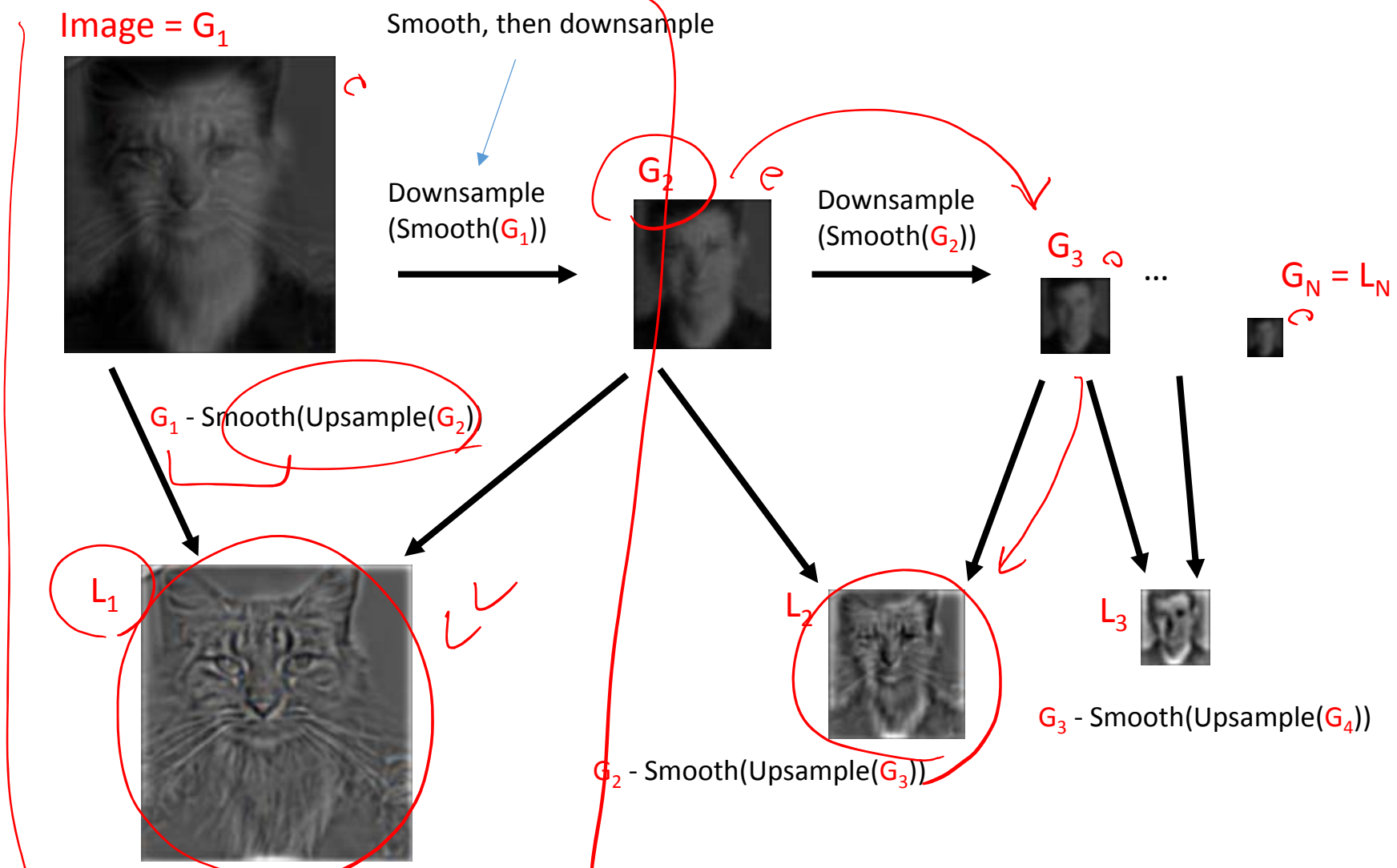
Source: Lazebnik

Laplacian Pyramid



Source: Forsyth

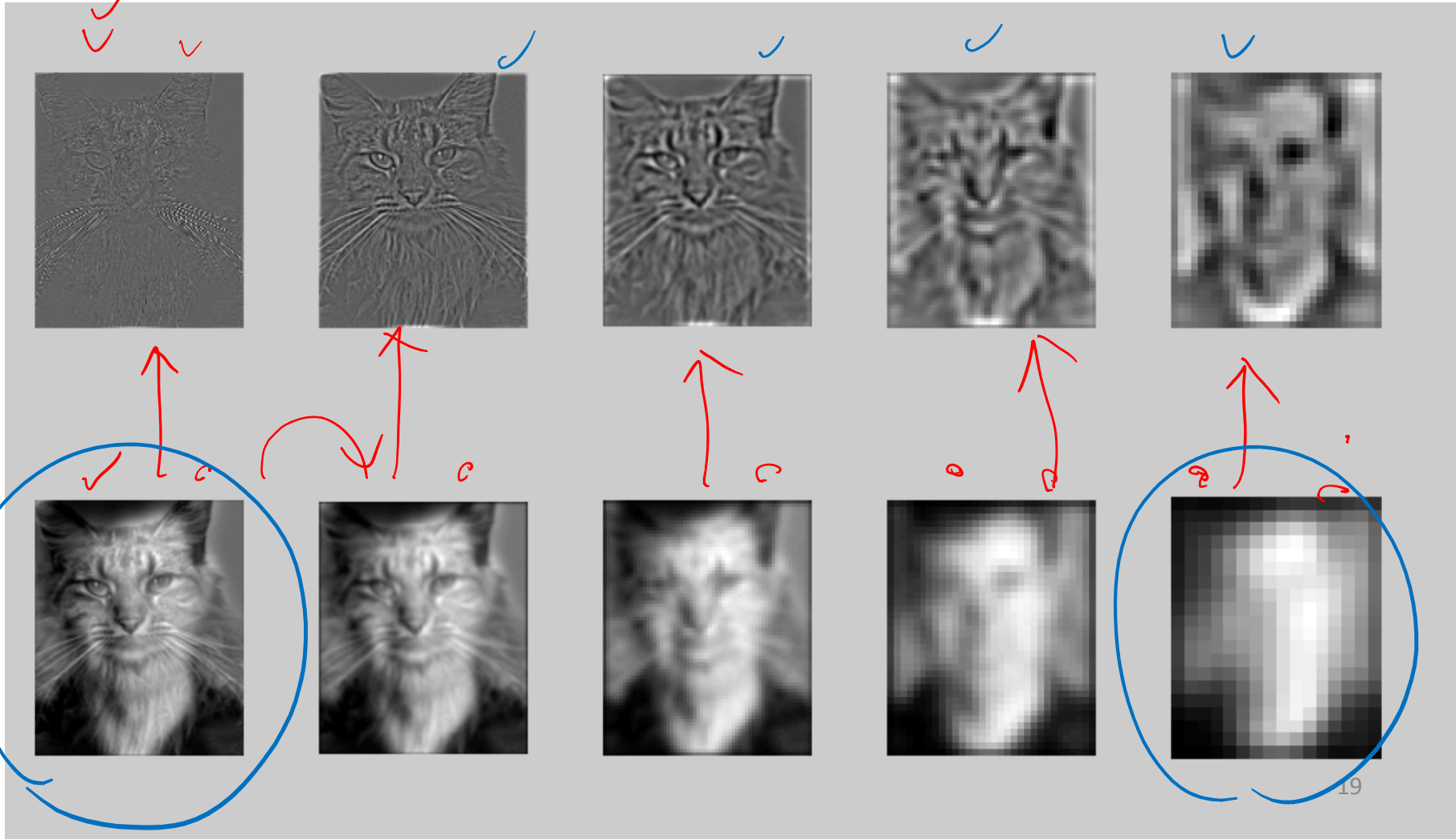
Creating the Gaussian/Laplacian Pyramid



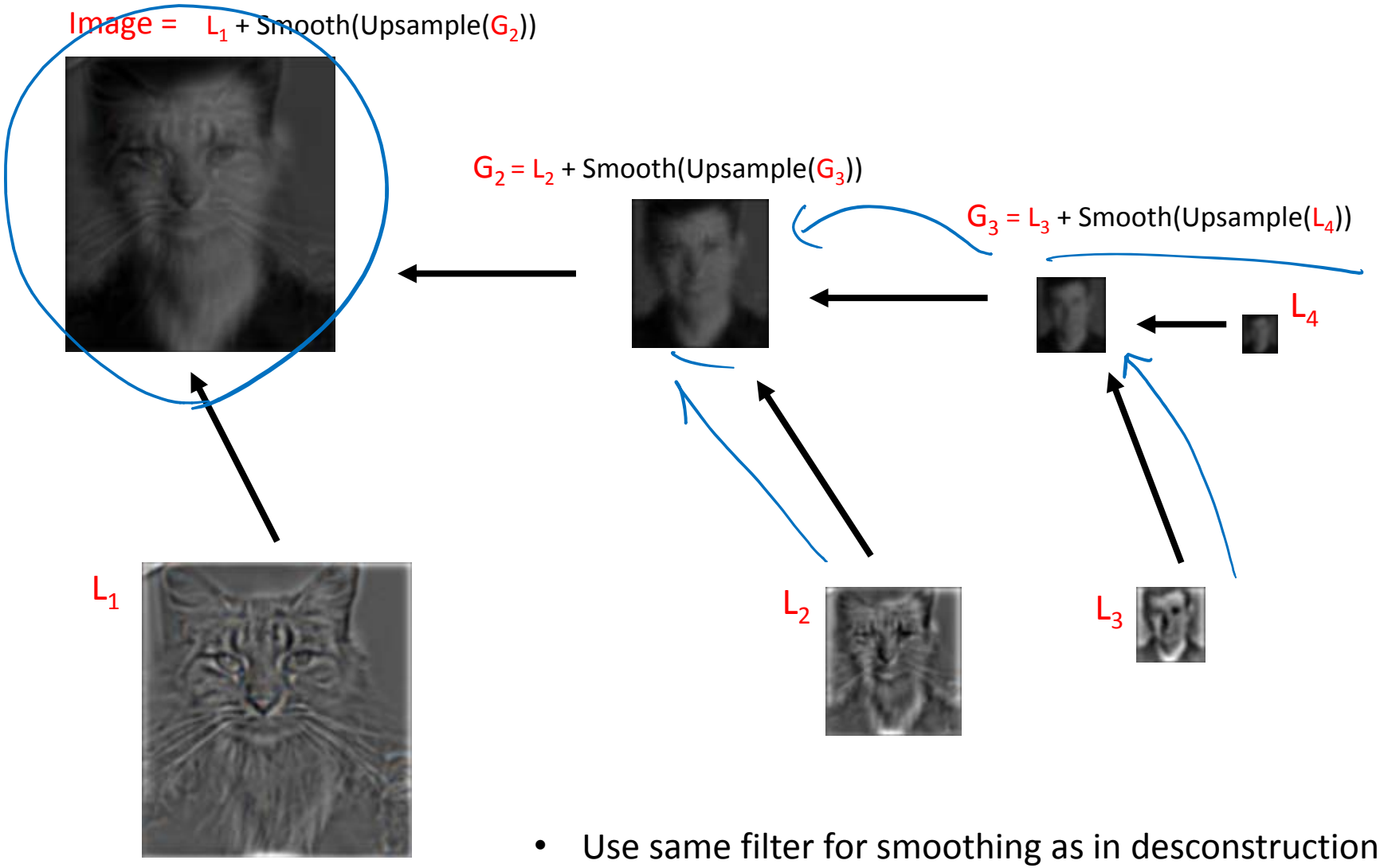
- Use same filter for smoothing in each step (e.g., Gaussian with $\sigma = 2$)
- Downsample/upsample with "nearest" interpolation

Hybrid Image in Laplacian Pyramid

High frequency → Low frequency



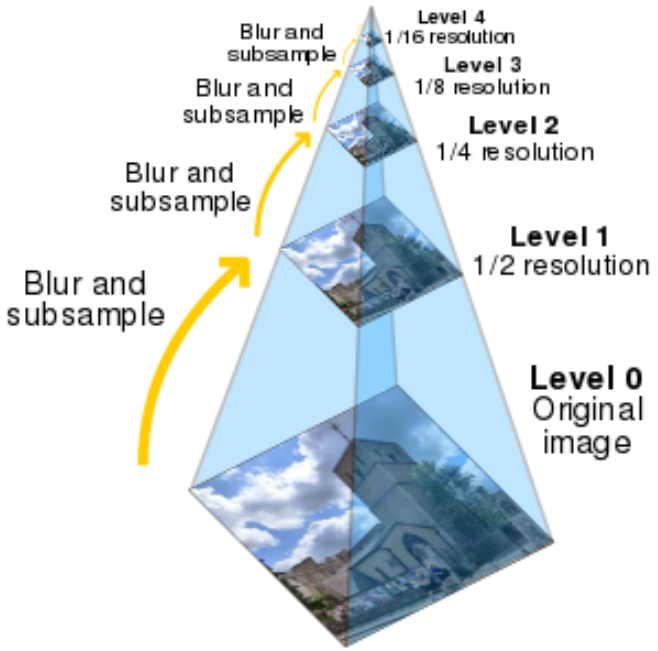
Reconstructing Image from Laplacian Pyramid



- Use same filter for smoothing as in desconstruction
- Upsample with “nearest” interpolation
- Reconstruction will be lossless

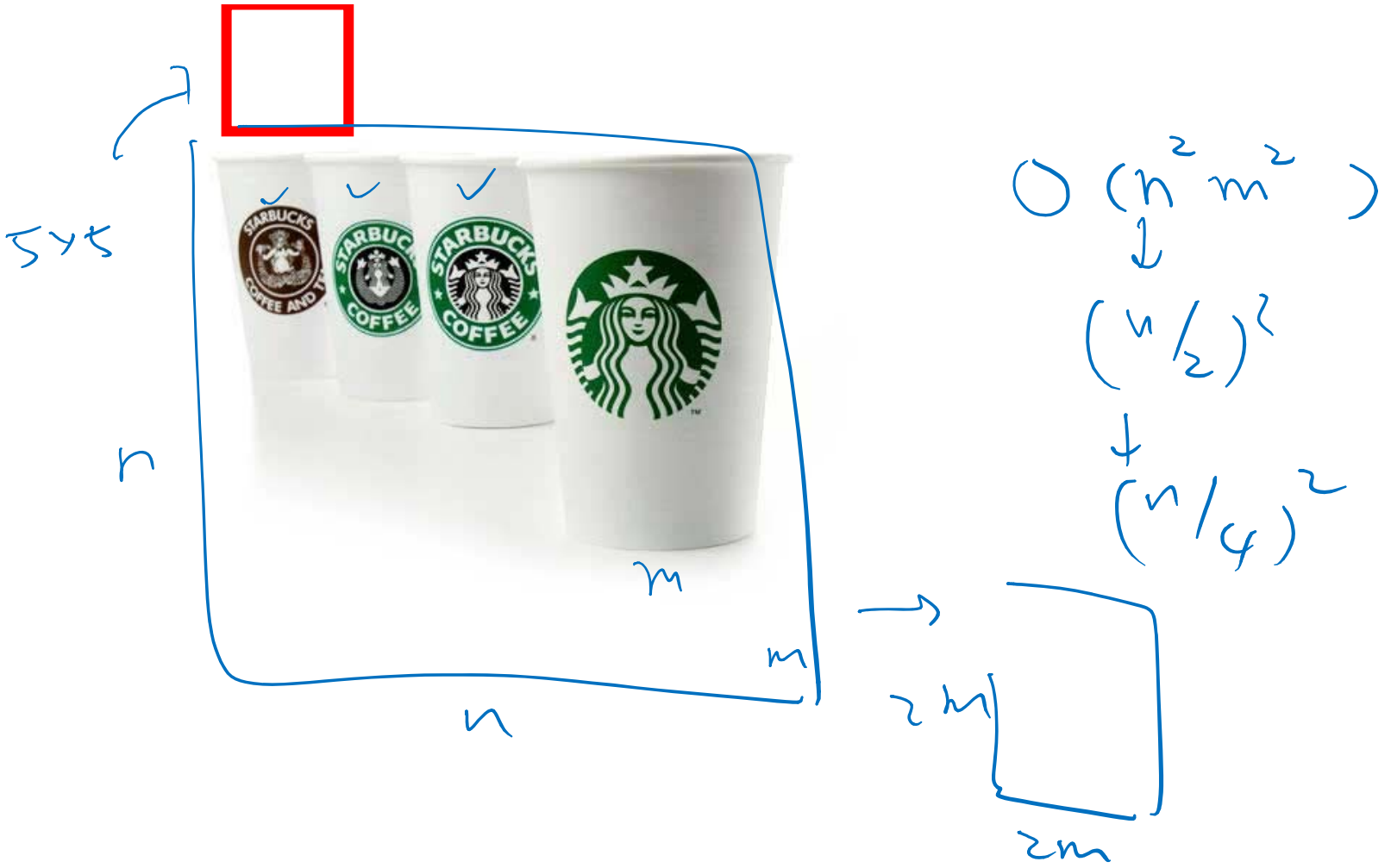
Example Uses of Image Pyramids

- Object detection
  - Scale search
  - Features
- Detecting stable interest points
- Course-to-fine registration
- Compression




5 x 5
STARBUCKS
COFFEE AND
STARBUCKS
COFFEE
STARBUCKS
COFFEE

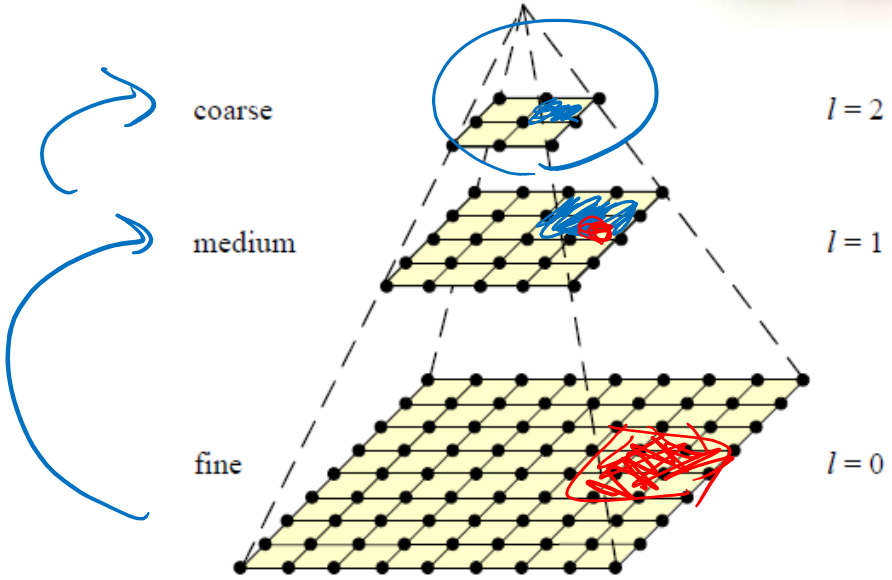

STARBUCKS
COFFEE
STARBUCKS
COFFEE
STARBUCKS
COFFEE


STARBUCKS
COFFEE
STARBUCKS
COFFEE
STARBUCKS
COFFEE
n
m
n
m
2m
2m
O(n² m²)
(n/2)²
(n/4)²

Coarse-to-Fine Image Registration

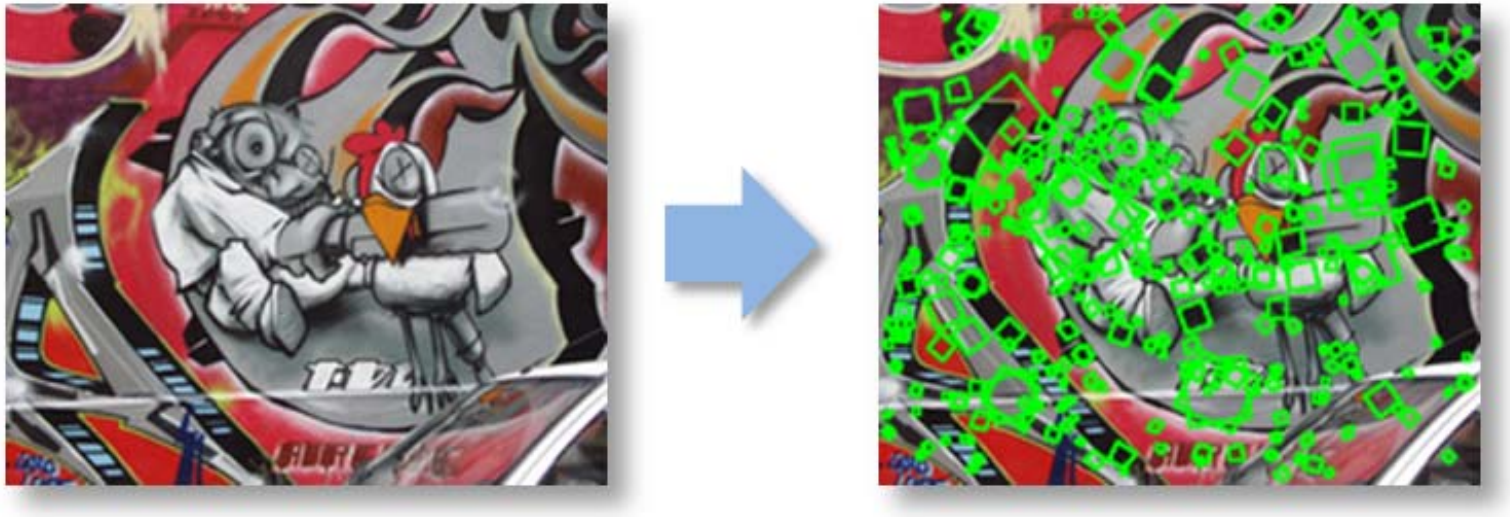
1. Compute Gaussian pyramid
2. Align with coarse pyramid
3. Successively align with finer pyramids
   - Search smaller range

Are we guaranteed to get the same result?



What's to Be Covered Today…

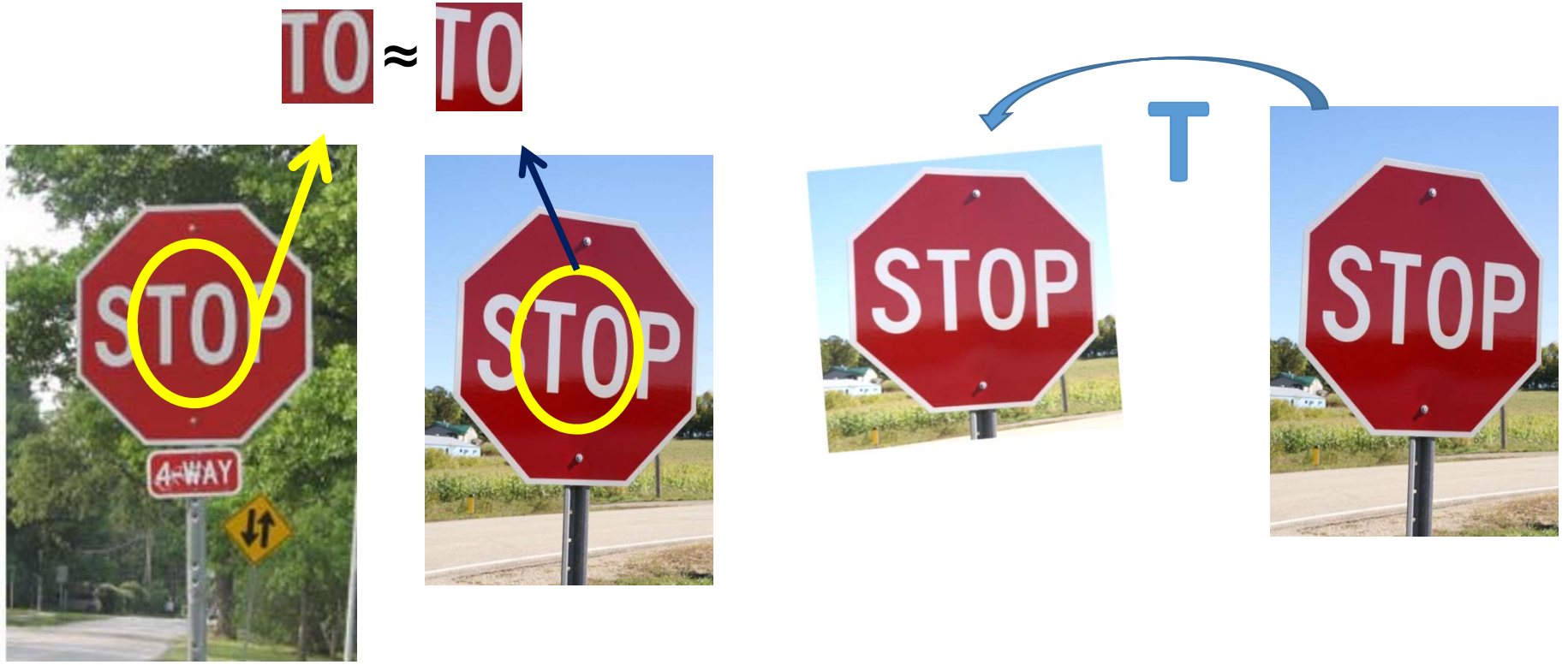
- Interest Points
  - Keypoint Detection & Description
  - Feature Tracking & Optical Flow
  - Image Recognition



Many slides from J.-B. Huang, J. Hayes, N Snavely, K. Grauman, and D. Hoiem

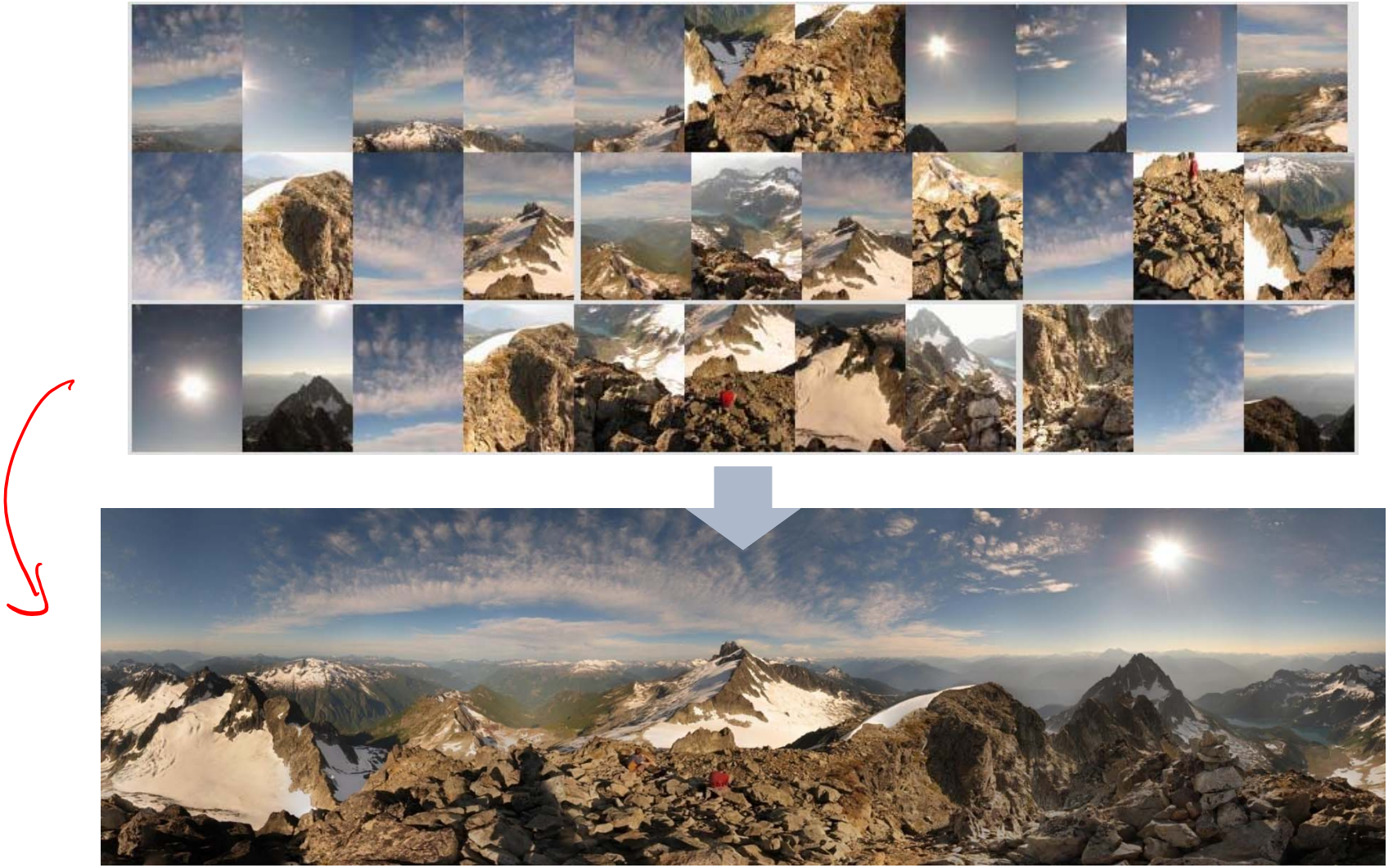
Why Interest Points?

- Registration & Correspondence
  - Identifying corresponding points/patches/regions across images
  - Apps: matching, alignment, stitching, etc.



Why Interest Points? (cont'd)

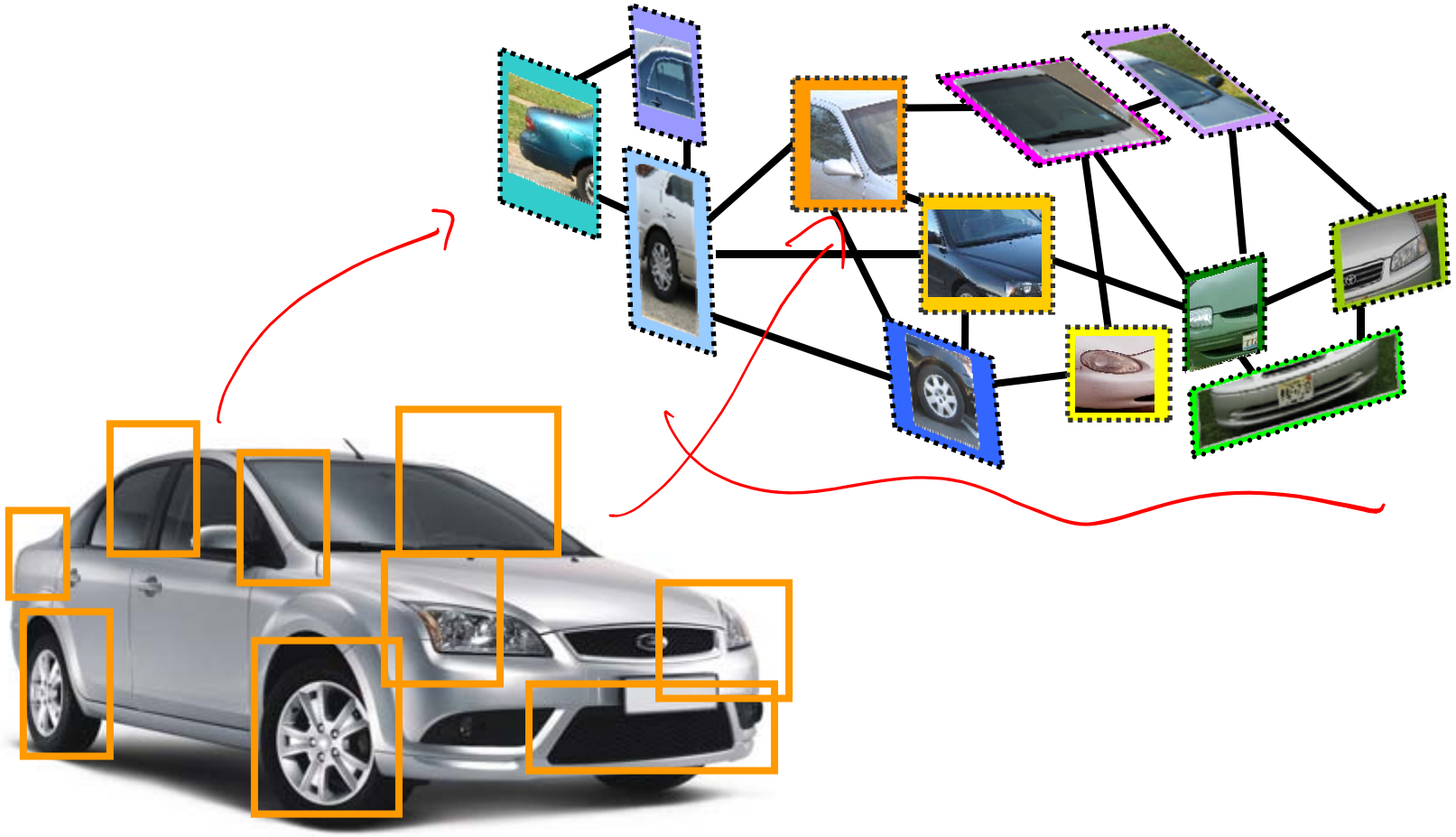
- Example: panorama



Credit: Matt Brown

Why Interest Points? (cont'd)

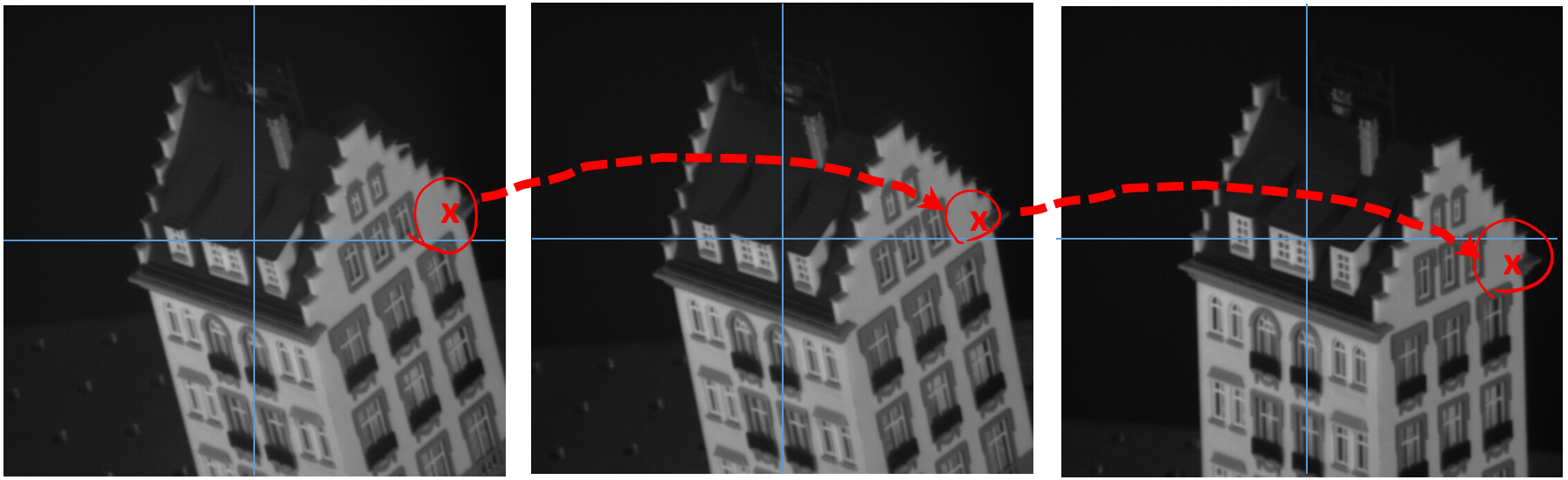
- Example: fitting 3D models



Credit: Silvio Savarese

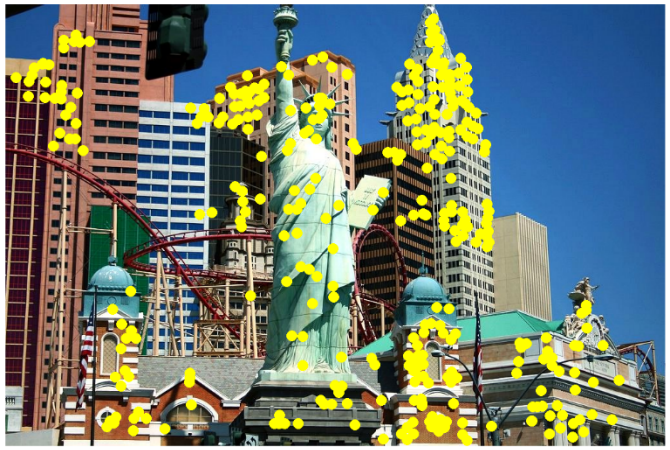
Why Interest Points? (cont'd)

- Example: tracking



Why Interest Points? (cont'd)

- Examples
  - Image alignment
  - 3D reconstruction
  - Motion tracking
  - Object recognition
  - Robot navigation
  - Indexing and database retrieval



About Interest Points

- Desirable properties for local features
  - **Locality**

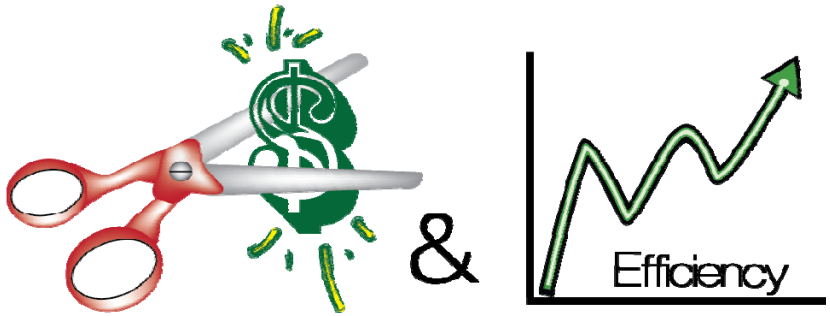
    Features are local, and robust to noise
  - **Quantity**

    A large number would be expected.
  - **Distinctiveness**

    Differentiate a variety of images of interest (e.g., objects, etc.)
  - **Repeatability**

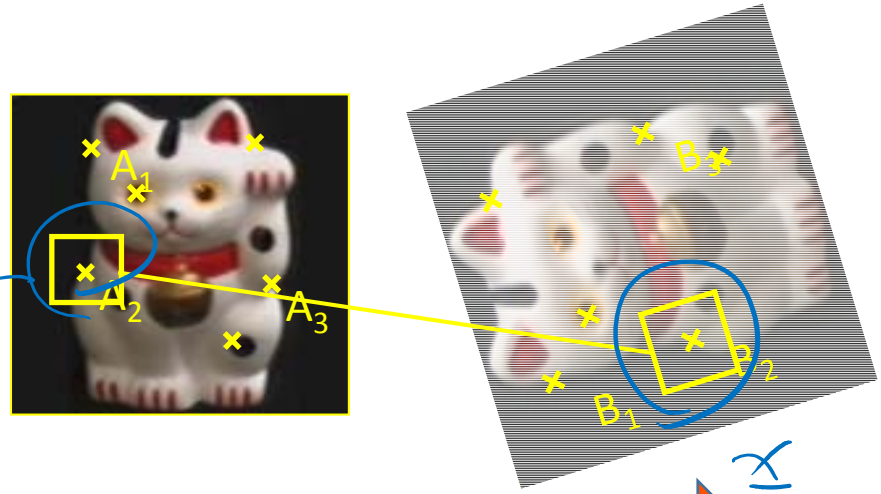
    Able to detect the same interest points of interest
  - **Compact & Efficiency**

    Real-time performance would be desirable.



About Interest Points

- Key Trade-offs



Detection

Few Points

More distinctive representation
Robust detection
Precise localization

More Points

Robust to occlusion
Works with less texture

Description

More Distinctive

Minimize wrong matches

More Flexible

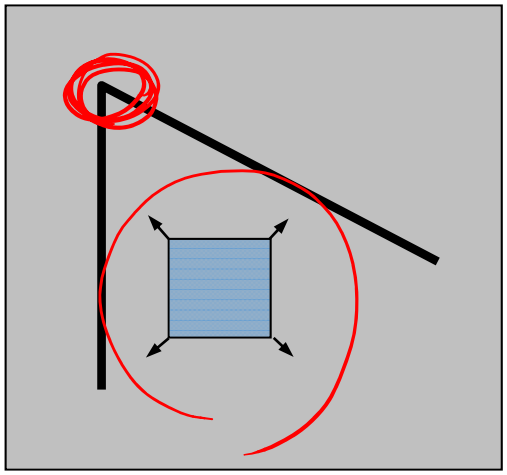
Robust to expected variations
Maximize correct matches

Choosing Interest Points

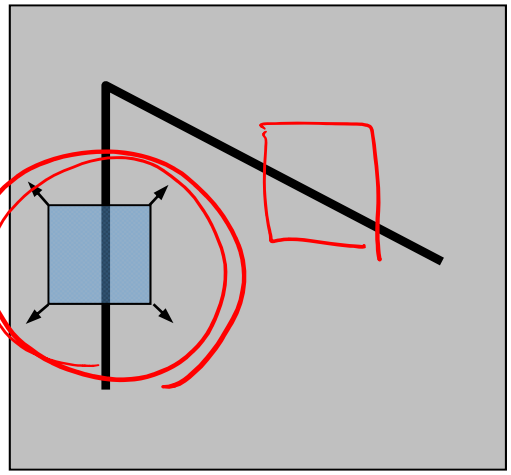
- Where would you tell your friend to meet you?

Corner Detection

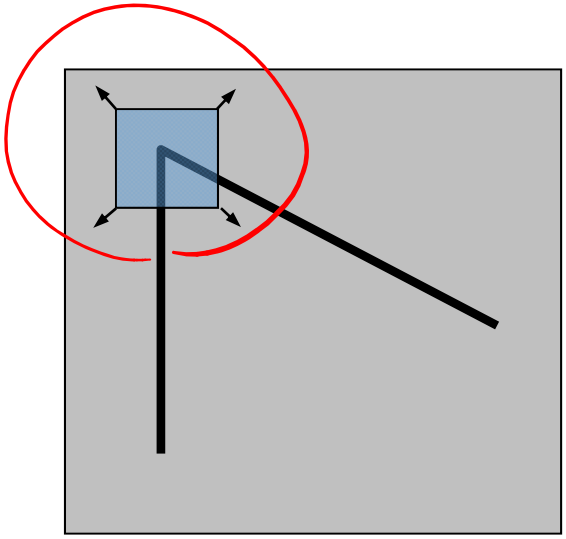
- Basic Idea
  - How to expect from the window output when shifting?



"flat" region:
no change in all directions



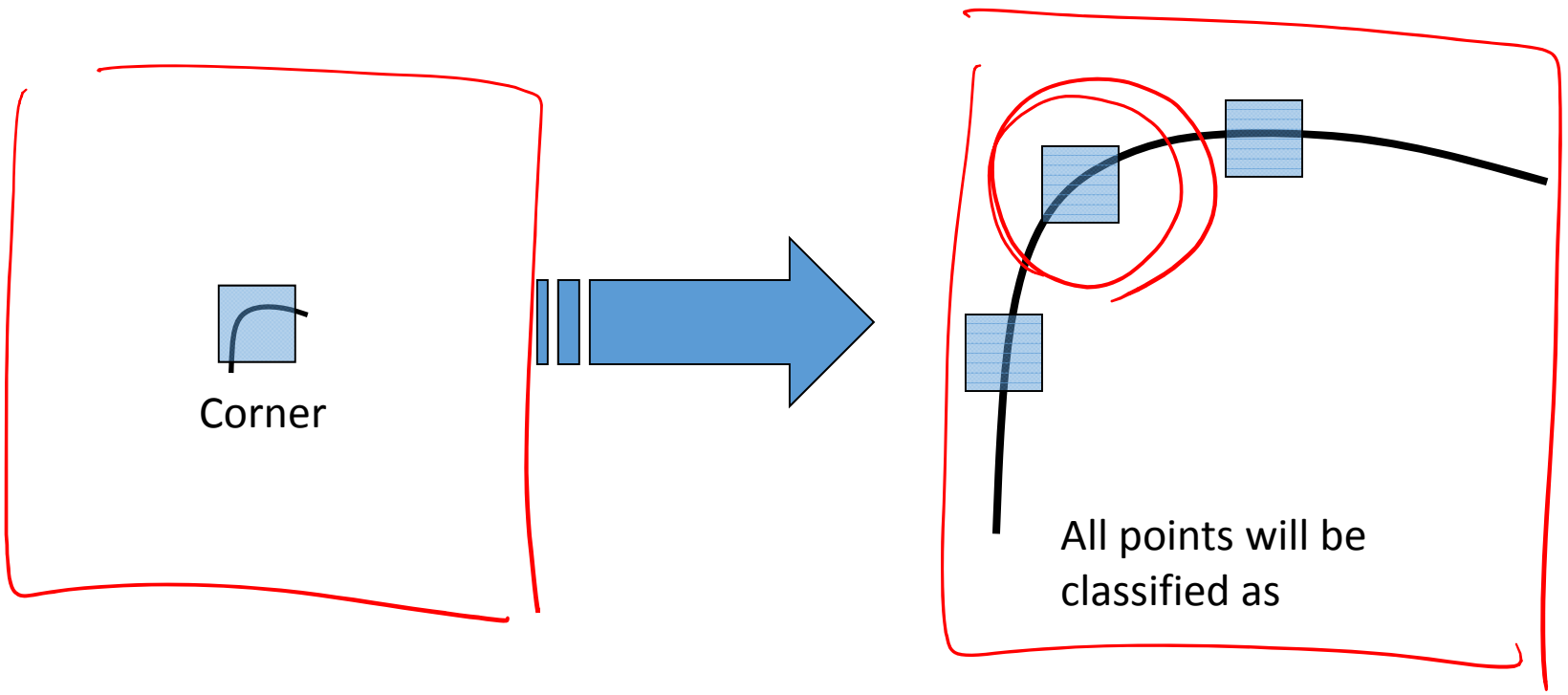
"edge":
no change along the edge direction



"corner":
significant change in all directions

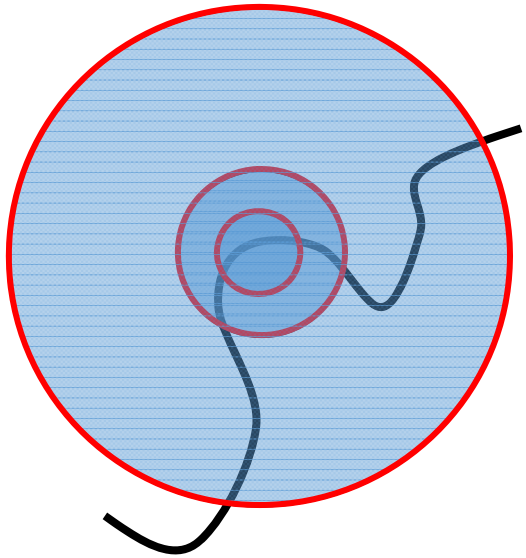
Corner Detection

- What's the limitation?
  - Not invariant to… scale.



Scale-Invariant Detection

- Key Idea
  - Find scale that gives local maximum of detector function f
  - Example of f: Harris operator (not covered in this course)

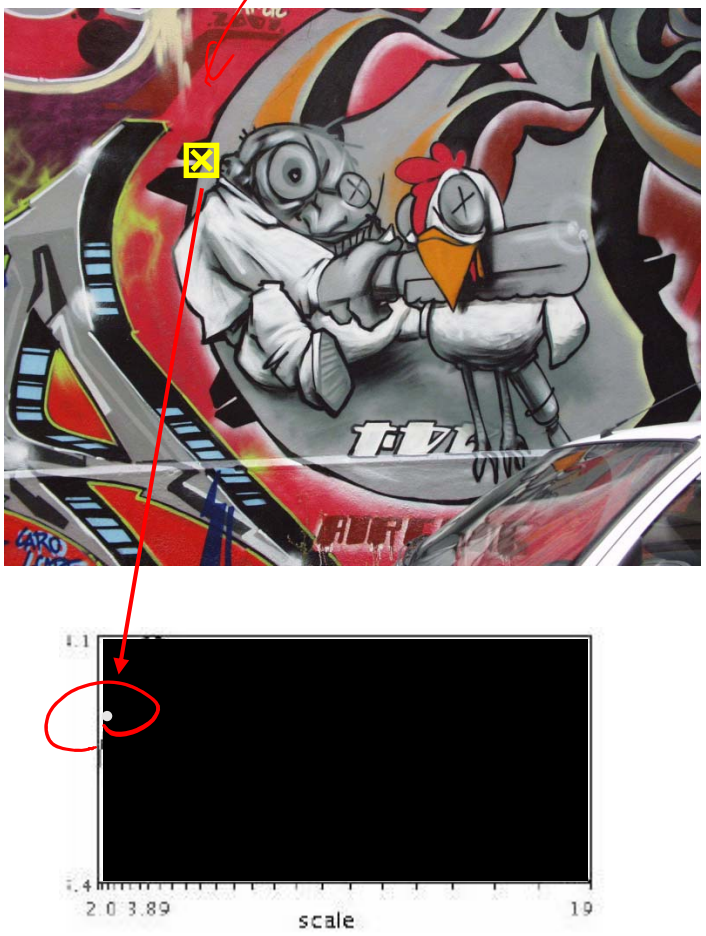


Automatic Scale Selection

$$f(I_{i_1 \mathrm{K}\, i_m}(x,\sigma)) \quad = \quad f(I_{i_1 \mathrm{K}\, i_m}(x',\sigma'))$$

How to find corresponding patch sizes?

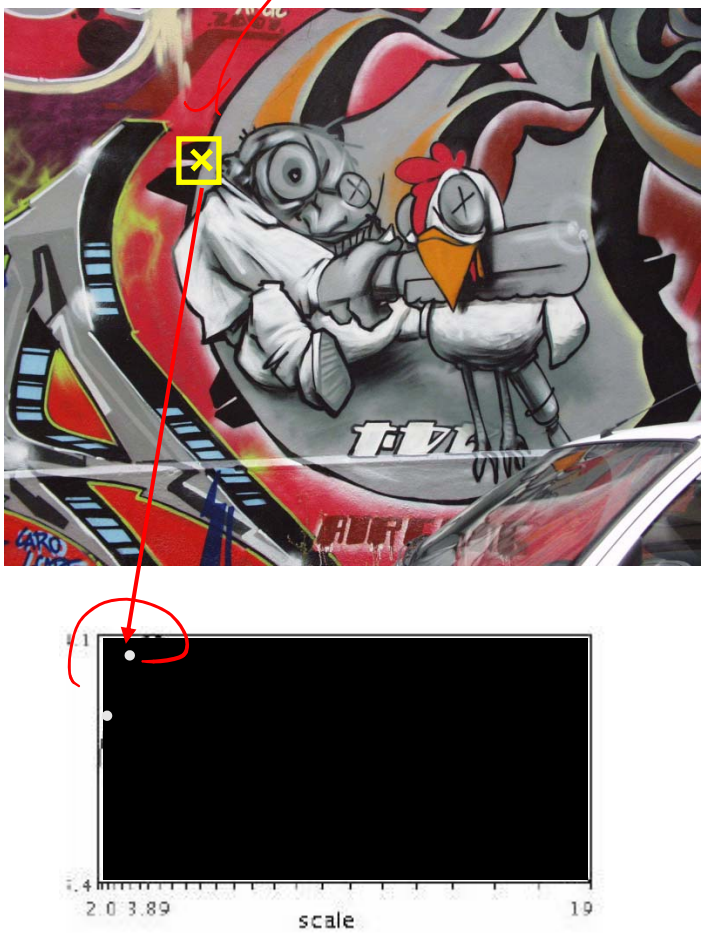
Automatic Scale Selection



$f(I_{i_1 \mathrm{K}\, i_m}(x,\sigma))$

$f(I_{i_1 \mathrm{K}\, i_m}(x',\sigma))$

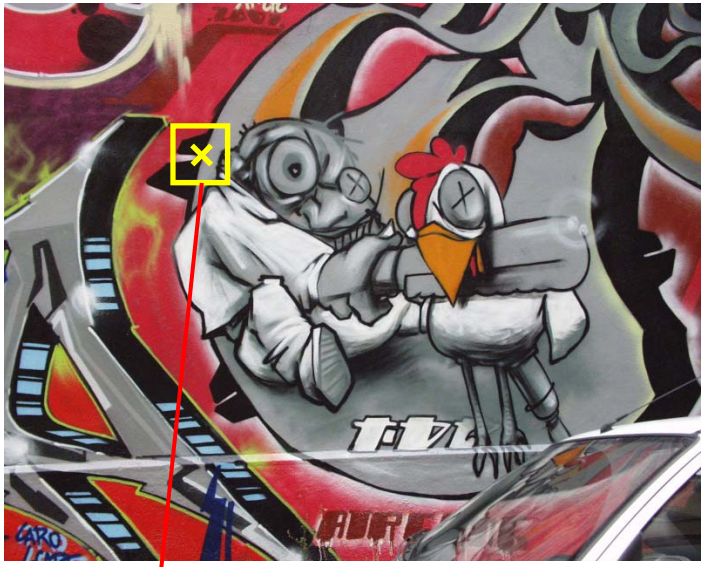
Automatic Scale Selection

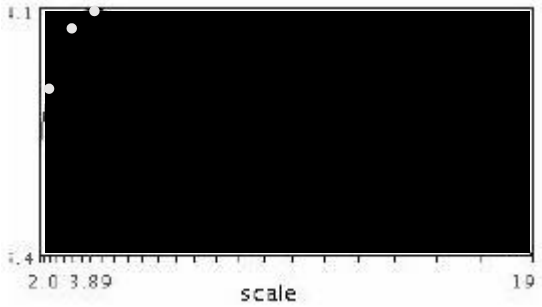


$f(I_{i_1 \mathrm{K}\, i_m}(x,\sigma))$

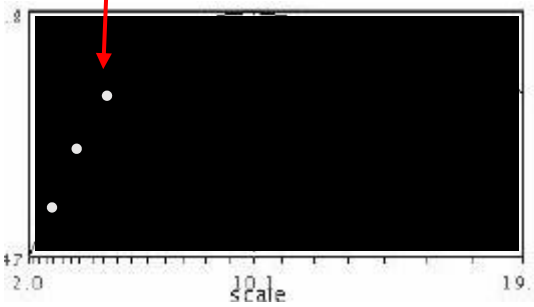
$f(I_{i_1 \mathrm{K}\, i_m}(x',\sigma))$

Automatic Scale Selection



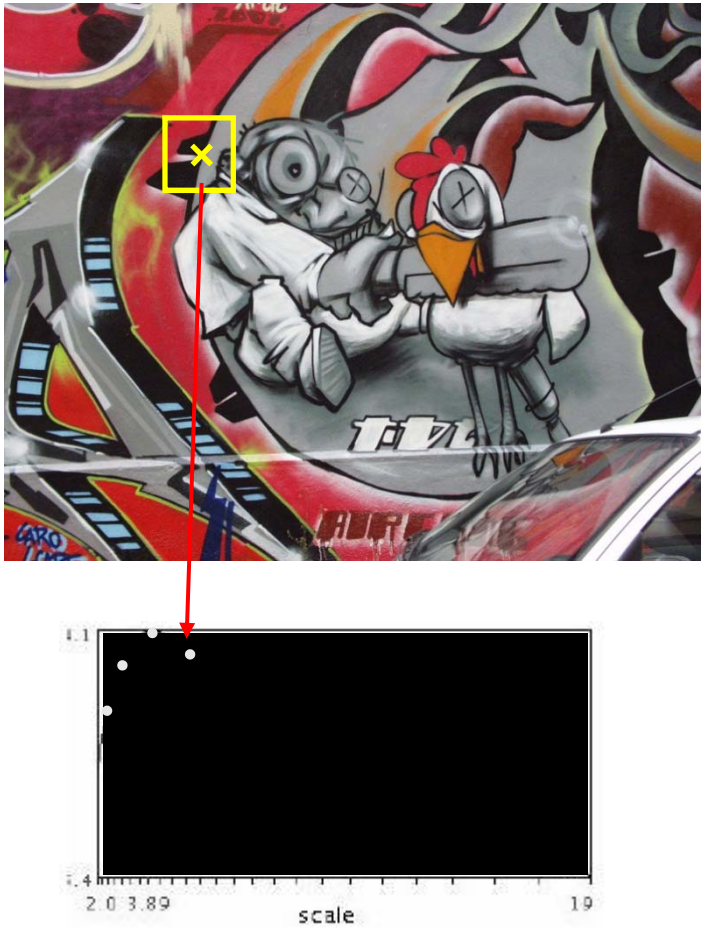


$f(I_{i_1 \mathrm{K}\, i_m}(x,\sigma))$



$f(I_{i_1 \mathrm{K}\, i_m}(x',\sigma))$

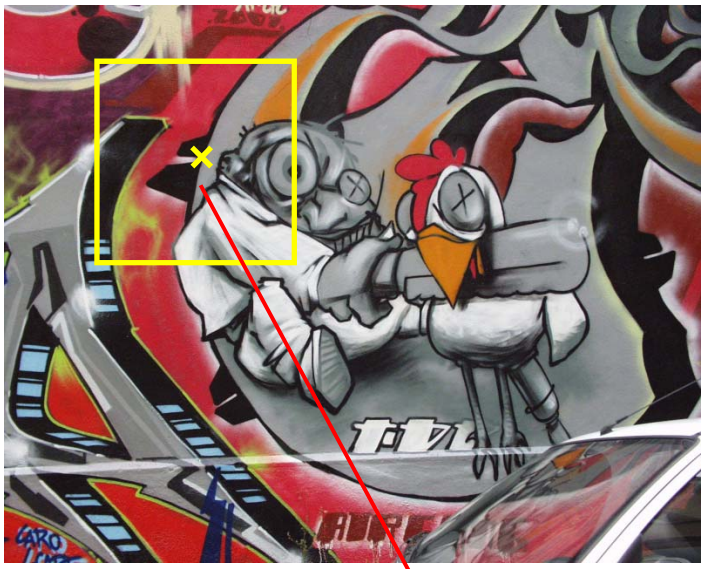
Automatic Scale Selection

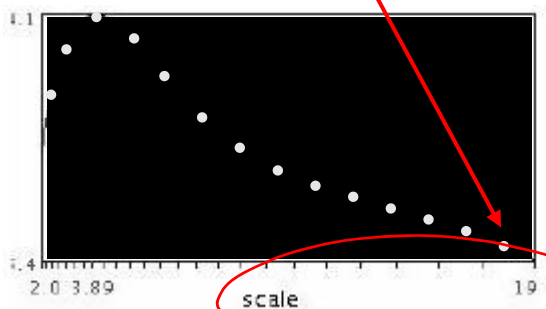


$f(I_{i_1 \mathrm{K}\, i_m}(x,\sigma))$

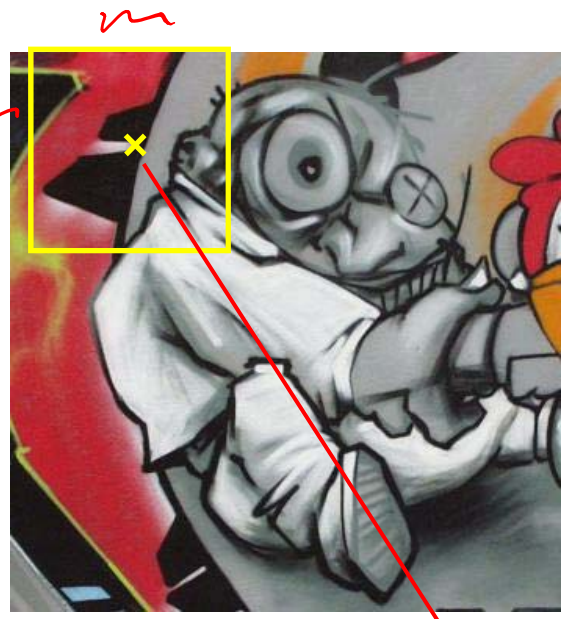
$f(I_{i_1 \mathrm{K}\, i_m}(x',\sigma))$

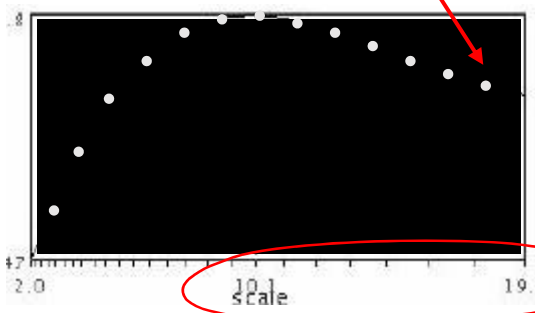
Automatic Scale Selection





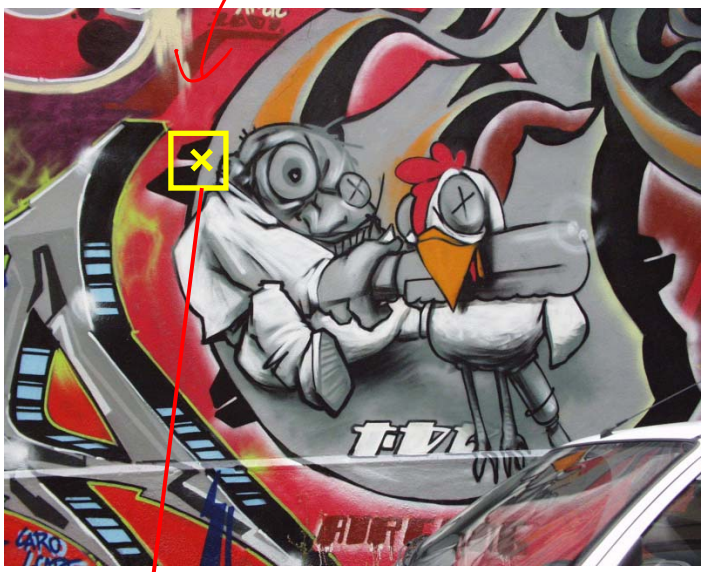
$f(I_{i_1 \mathrm{K}\, i_m}(x,\sigma))$

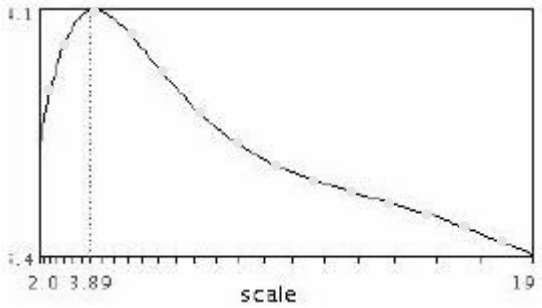




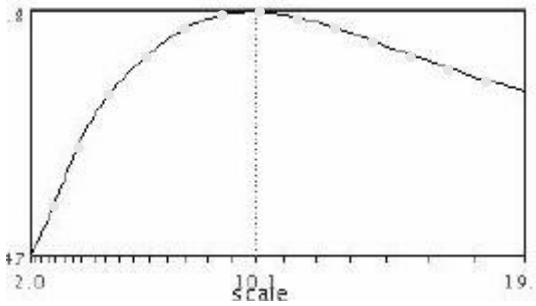
$f(I_{i_1 \mathrm{K}\, i_m}(x',\sigma))$

Automatic Scale Selection





$f(I_{i_1 \mathrm{K}\, i_m}(x,\sigma))$



$f(I_{i_1 \mathrm{K}\, i_m}(x',\sigma'))$

Implementation of Automatic Scale Selection

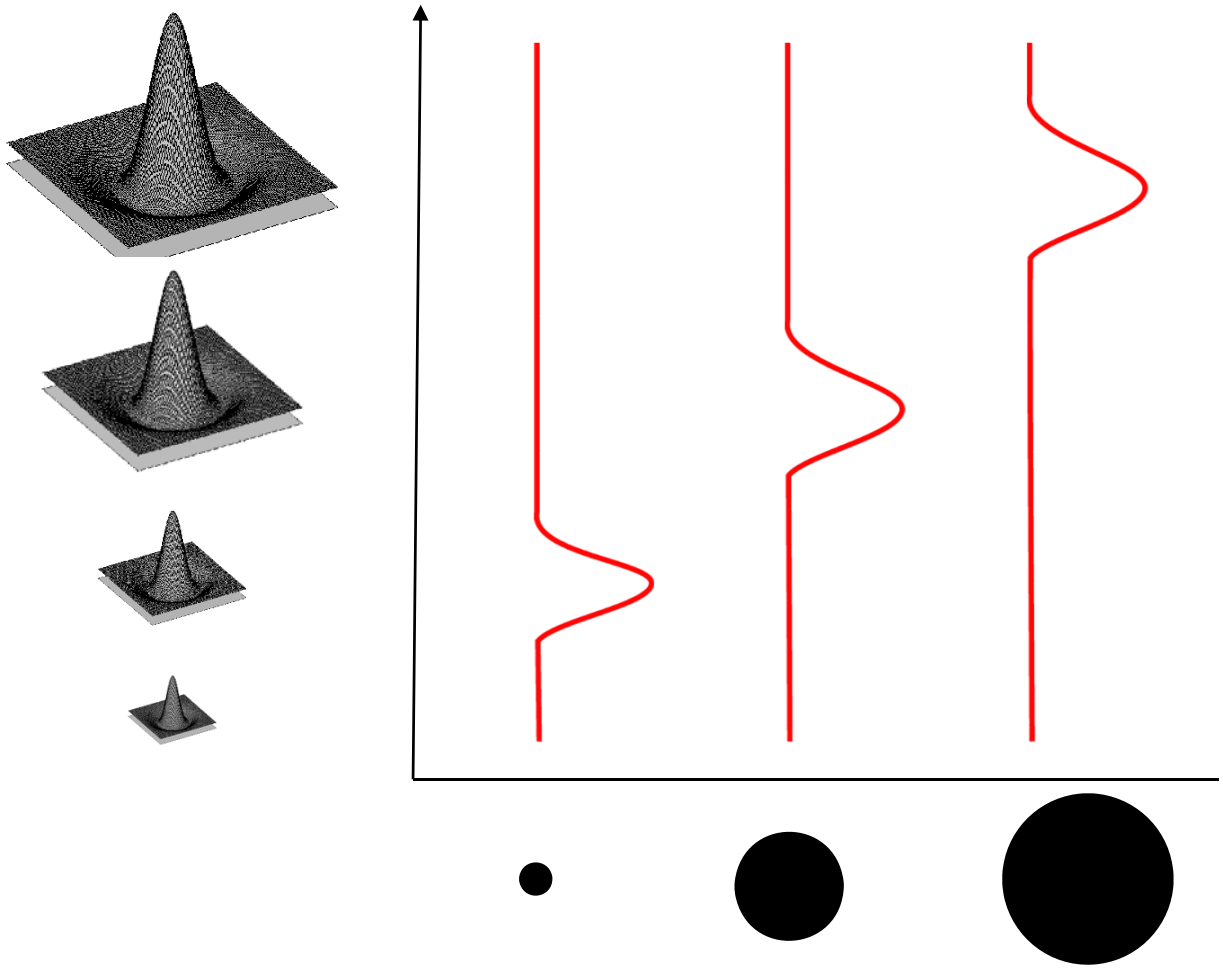
- Instead of computing *f* for larger and larger windows, we can implement using a fixed window size with a Gaussian pyramid.

$O(n^2 m^2)$

(sometimes need to create in-between levels, e.g., a ¾-size image)

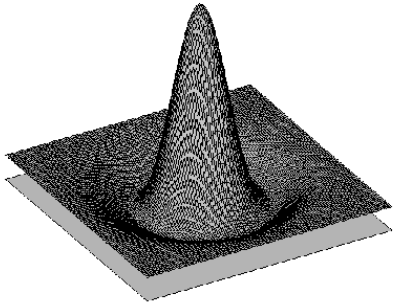
What's a Desirable Signature Function?

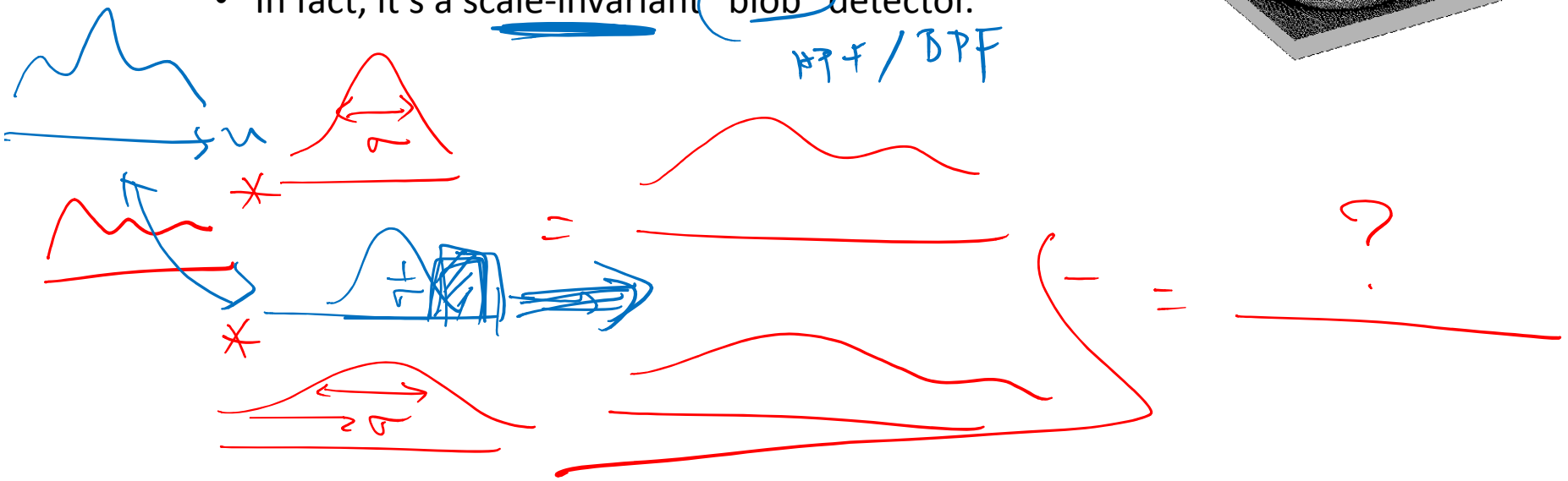
- Difference-of-Gaussian (DoG)
  - It's a "blob" detector. Why?



Difference of Gaussian (DoG)

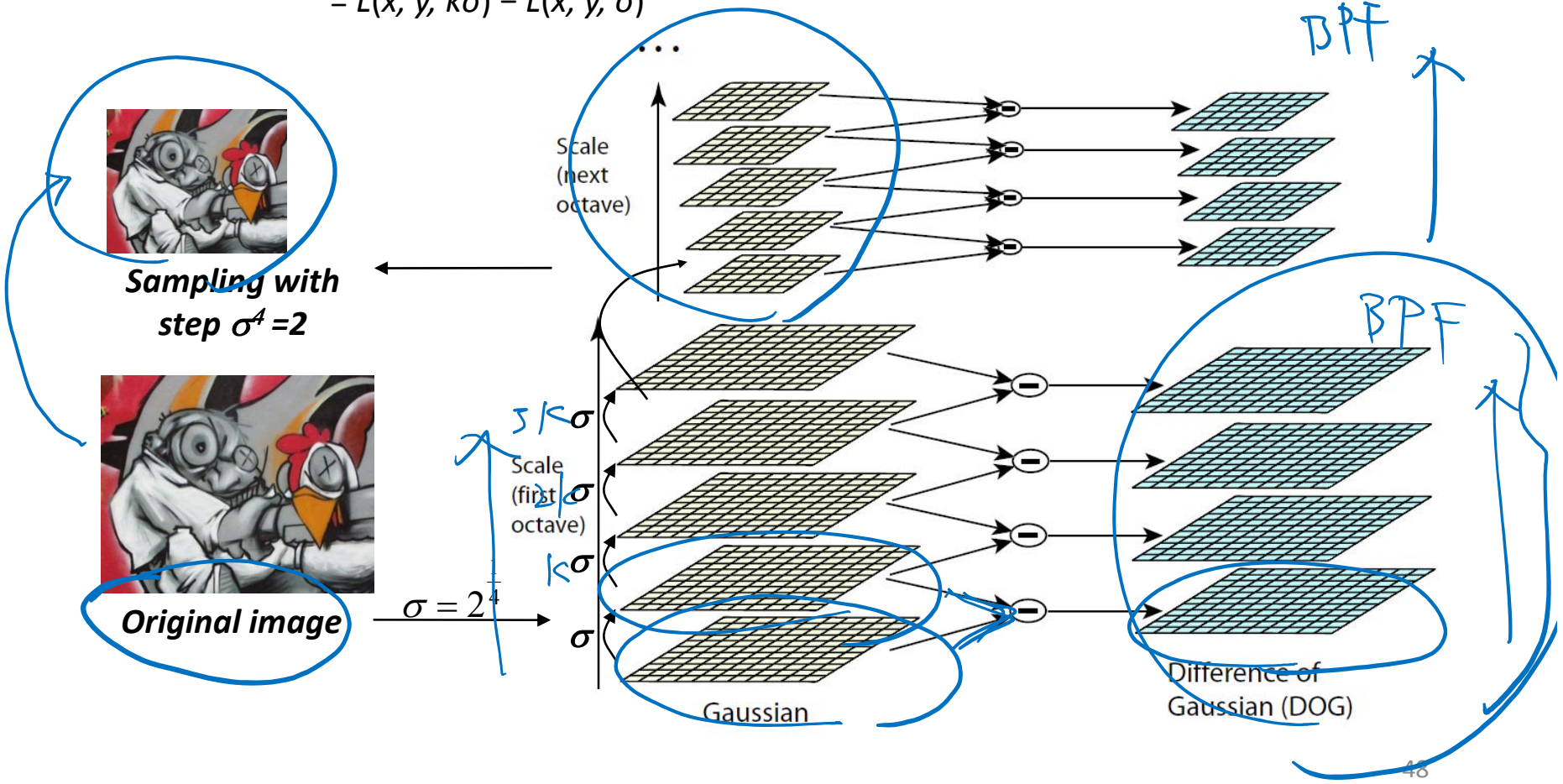
- Remarks
  - In fact, it's a scale-invariant "blob" detector.





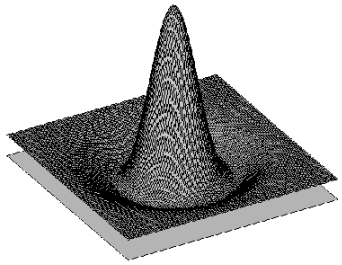
Difference of Gaussian (DoG)

- Remarks
  - Efficient to compute (via Gaussian scale pyramid)
  - $D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y)$
    $= L(x, y, k\sigma) - L(x, y, \sigma)$

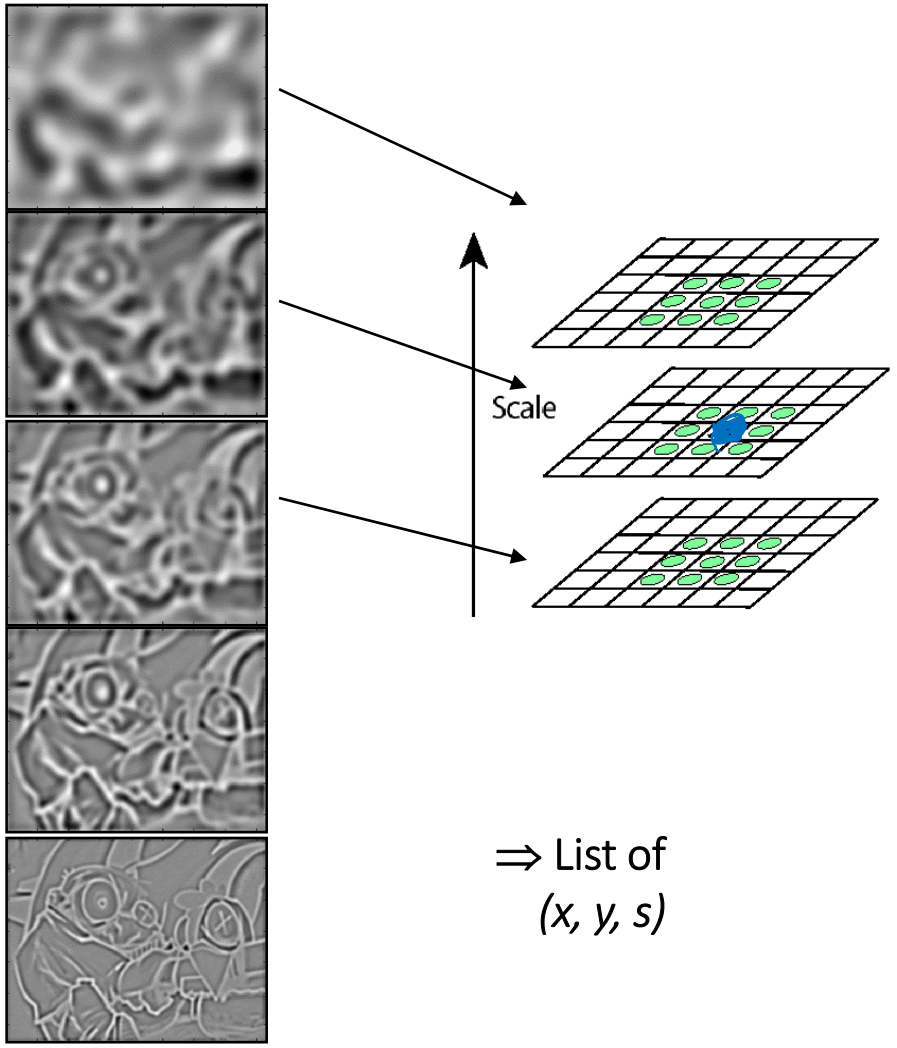


Difference of Gaussian (DoG)

- Remarks
  - Find local optimum in DoG



$$L_{xx}(\sigma) + L_{yy}(\sigma)$$

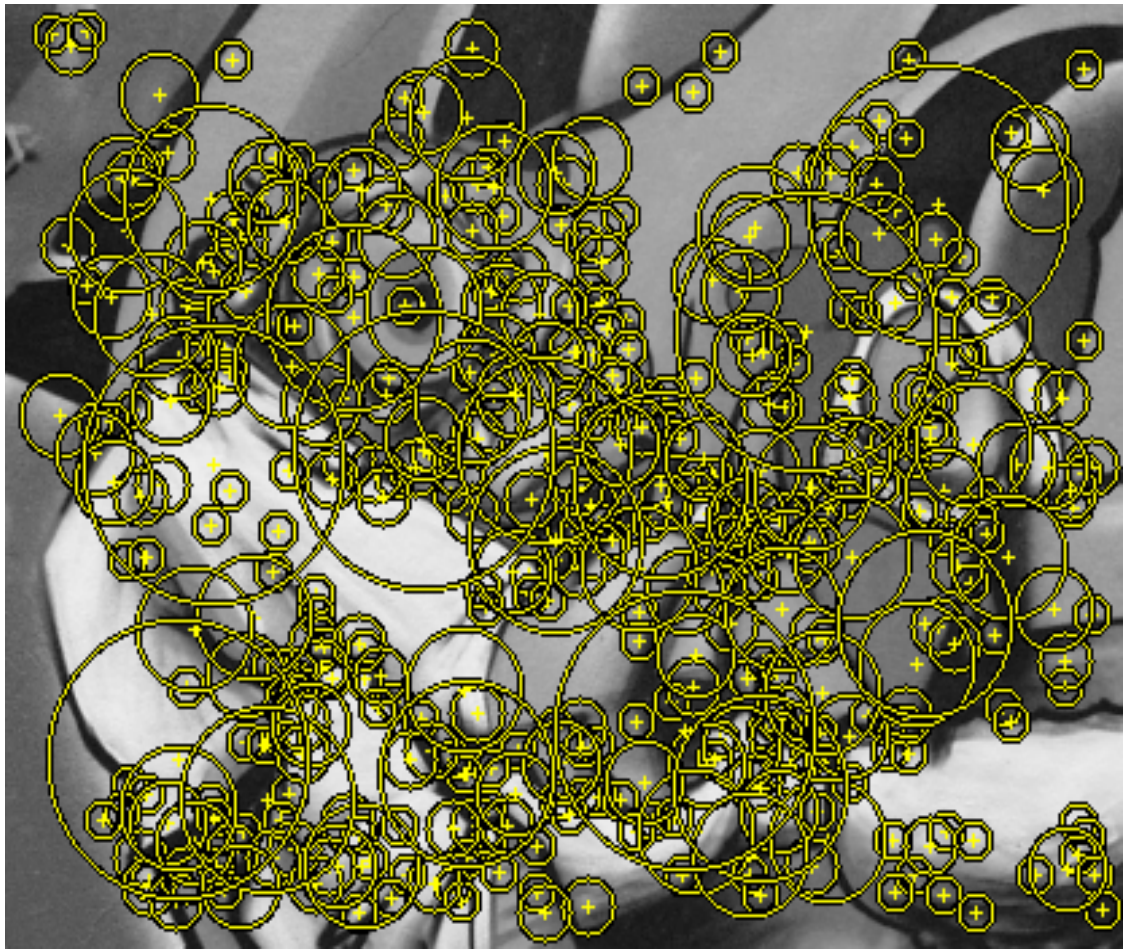


⇒ List of
(x, y, s)

For each location, compare to its 26 nearest neighbors & retain only minima/maxima.

Difference of Gaussian (DoG)

- Remarks
  - Example results

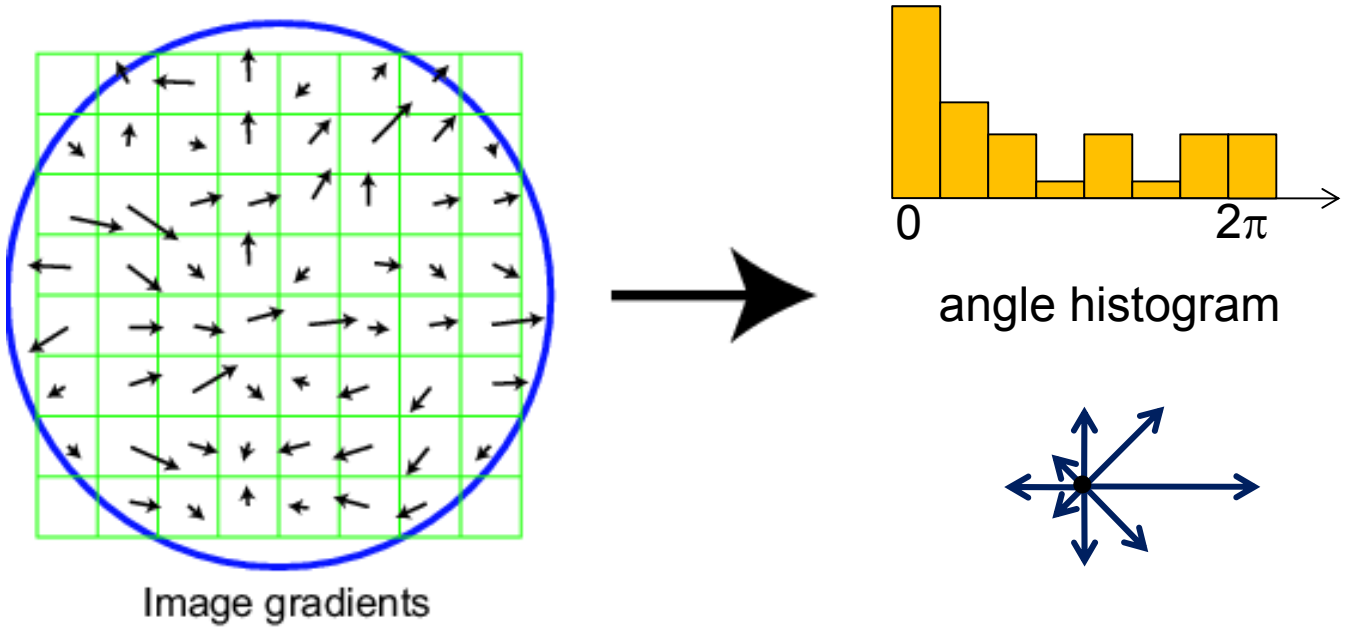


More Resources Available

- For most local feature detectors, executables are available online:
  - http://www.robots.ox.ac.uk/~vgg/research/affine
  - http://www.cs.ubc.ca/~lowe/keypoints/
  - http://www.vision.ee.ethz.ch/~surf

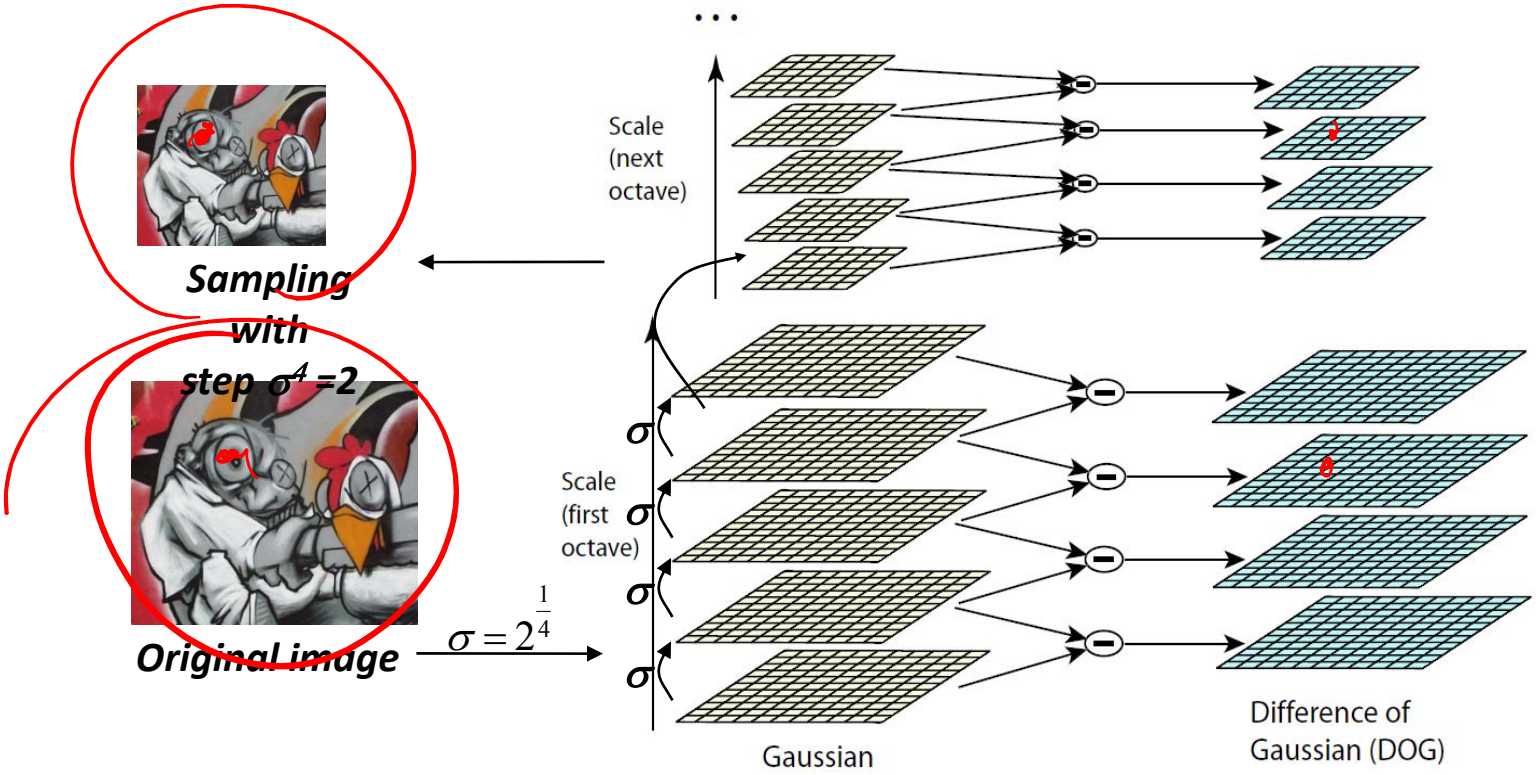
Scale Invariant Feature Transform (SIFT)

- Key Ideas
  - Take a 4 x 4 (= 16 grids) square window around each detected keypoint
  - Compute edge orientation (angle of the gradient - 90°) for each pixel in it
  - Throw out weak edges (threshold gradient magnitude)
  - Create histogram of surviving edge orientations

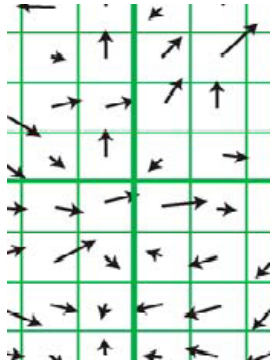


Scale Invariant Feature Transform (SIFT)

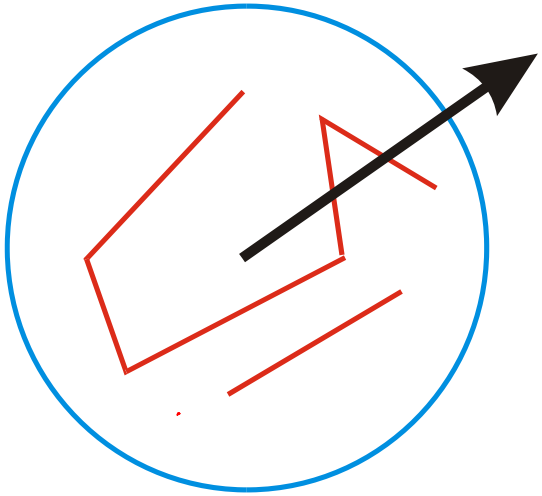
- Step 1. Keypoint Detection/Localization
  - Eliminate edge responses

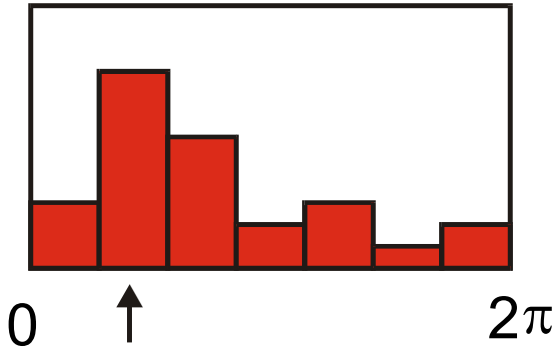


Scale Invariant Feature Transform (SIFT)



- Step 2. Orientation Normalization
  - Caluculate orientation and magnitude of gradients in each pixel
  - Histogram of orientations of sample points near keypoint.



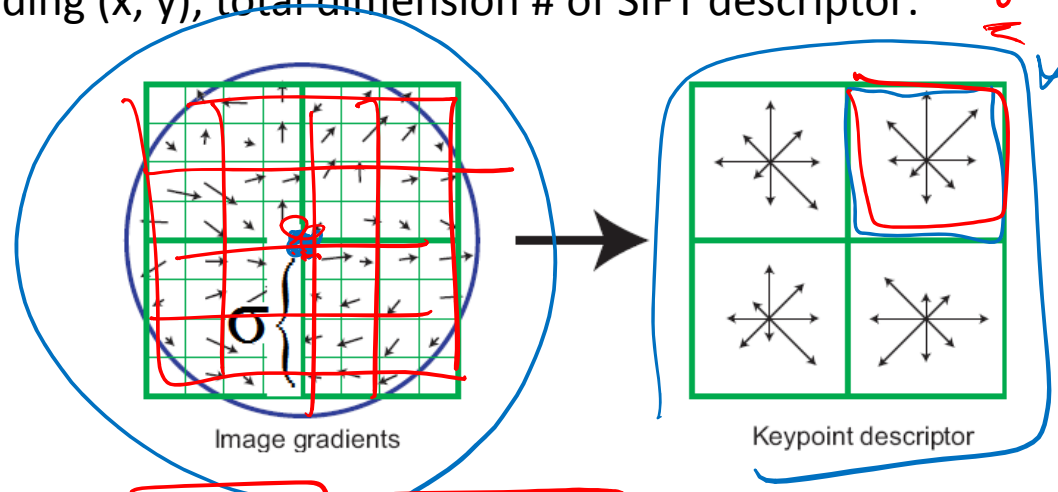


$$m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2}$$

$$\theta(x, y) = \tan^{-1}((L(x, y+1) - L(x, y-1))/(L(x+1, y) - L(x-1, y)))$$

Scale Invariant Feature Transform (SIFT)

- Step 3. Keypoint Descriptor
  - Recall that orienation has been normalized.
  - **3-1**. Divide sample points around keypoint in 4 x 4 = 16 regions
    (4 regions shown in the bottom-right figure)
  - **3-2**. Calculate histogram of orientations with 8 bins for each region
    (followed by Trilinear interpolation + Vector normalization)
  - Excluding (x, y), total dimension # of SIFT descriptor:

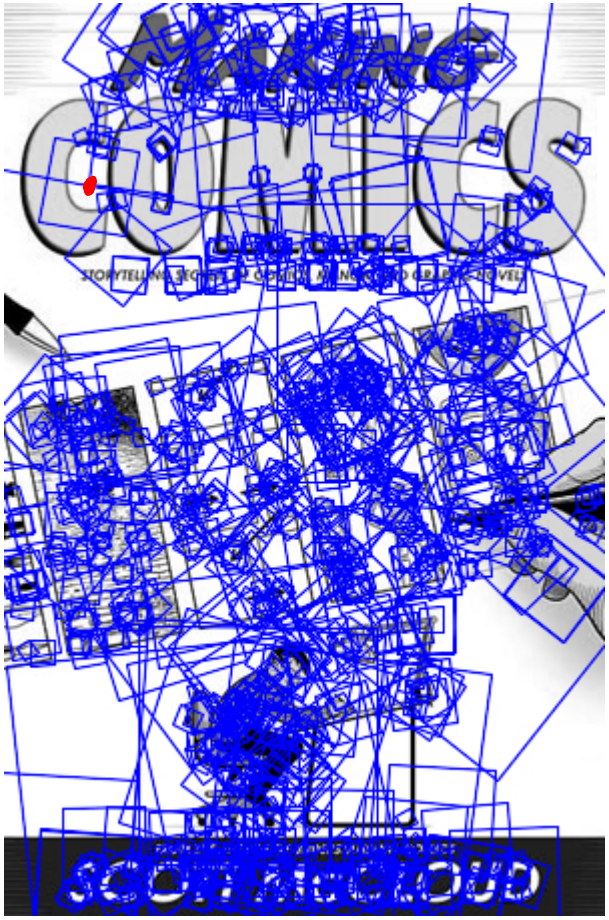


- Invariant to local scale & orientation. What about out-of-plane rotation?
- Invariant for illumination and 3D viewpoint changes?

Scale Invariant Feature Transform (SIFT)

- Example

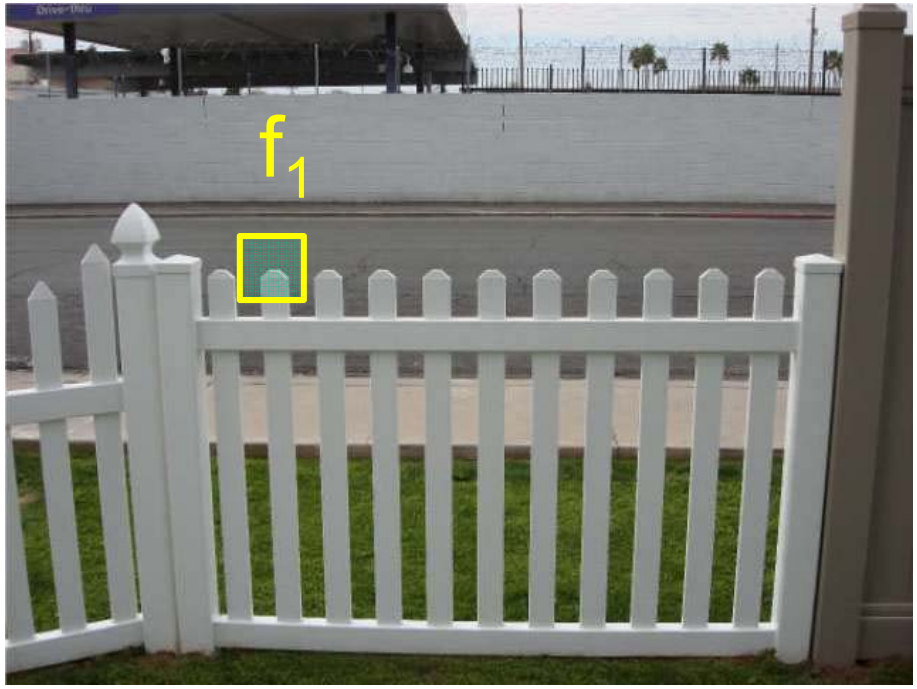
sift



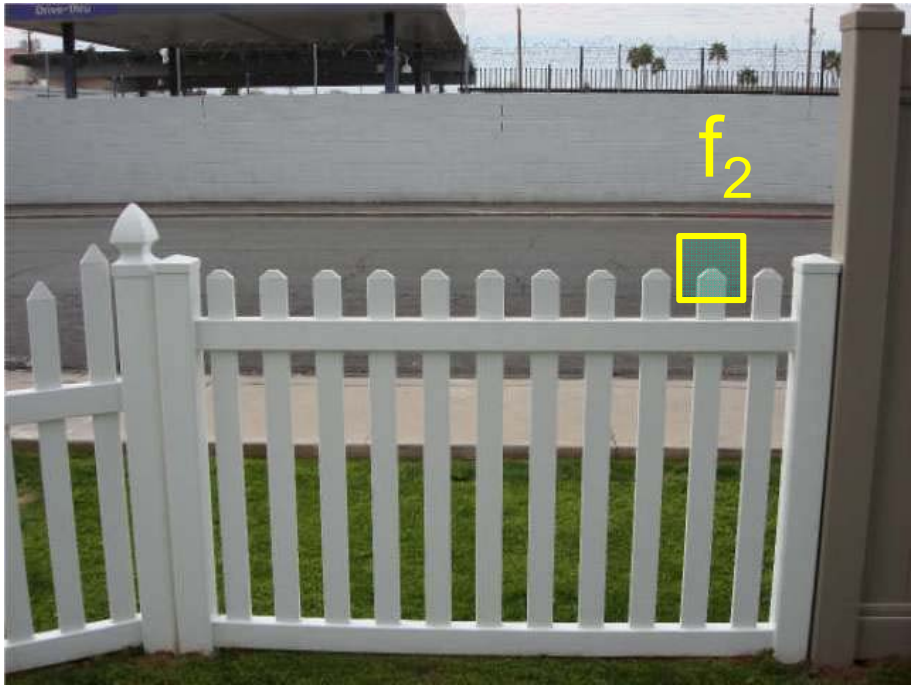
868 SIFT features

Interest Points for Image Matching

- Given a region of interest f_1 in I_1, how to find the best match f_2 in I_2?
  1. Define distance function that compares two descriptors f_1 & f_2
  2. Test all the features in I_2, find the one with min distance



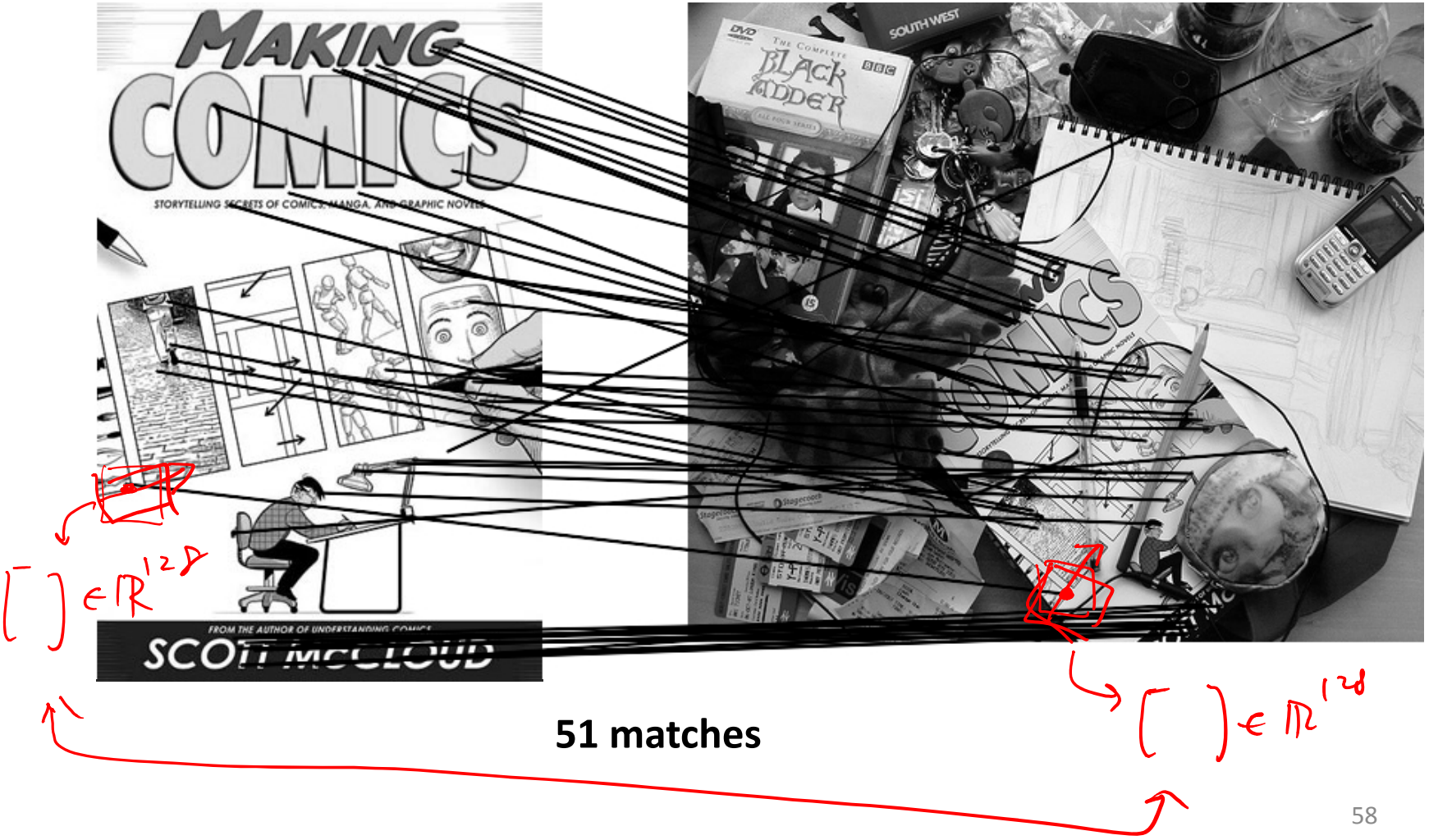
I_1



I_2

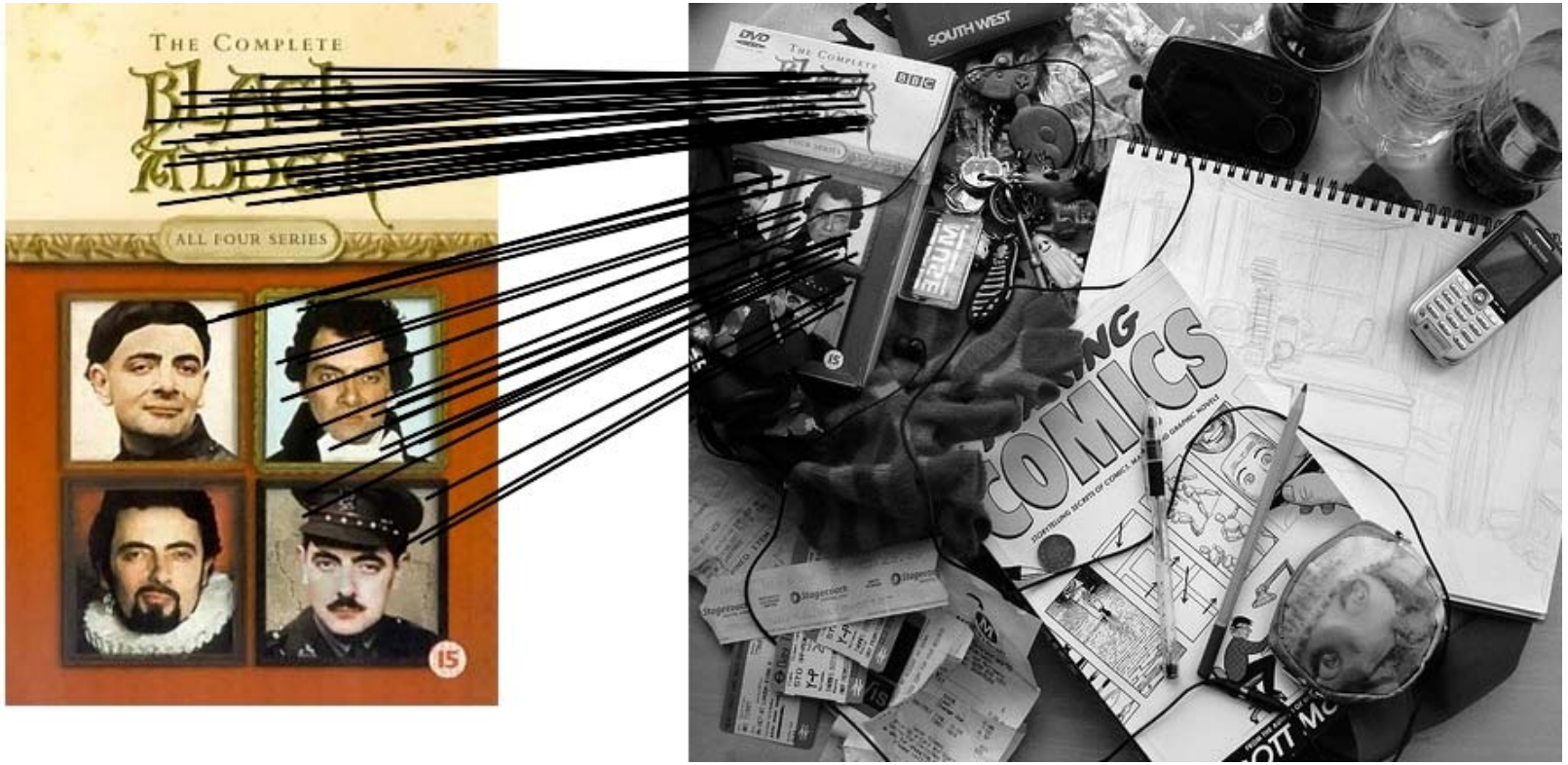
Interest Points for Image Matching

- Examples



Interest Points for Image Matching

- Examples

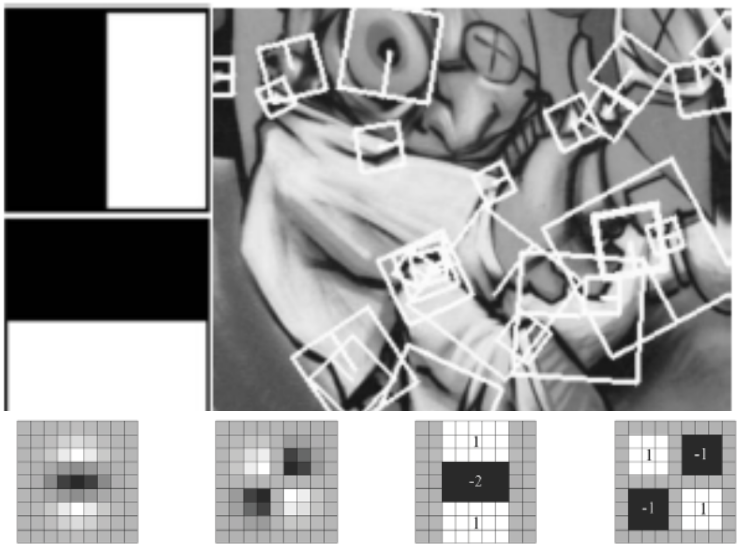


58 matches

✓ Disregard outlier pairs by RANdom SAmple Consensus (RANSAC) algorithm

Recent Advances in Interest Points

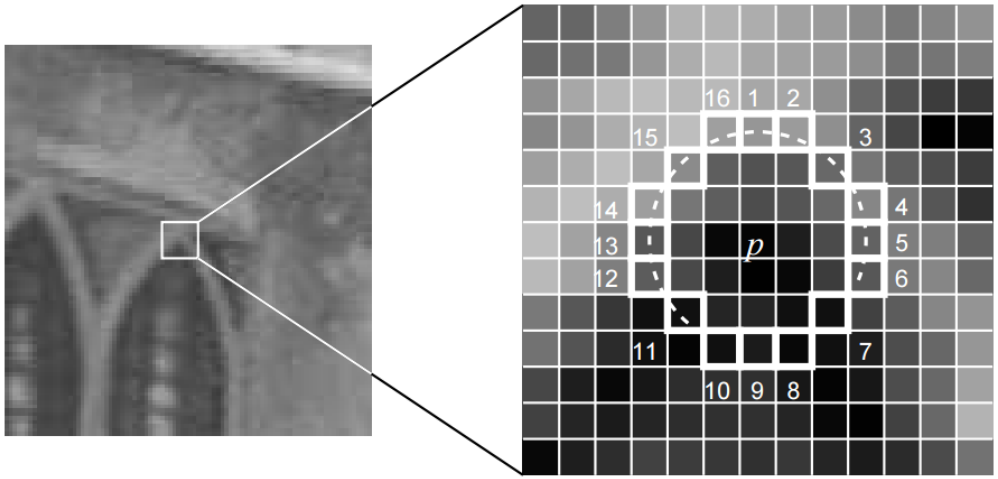
- Speeded Up Robust Features (SURF)
  - Fast approximation of SIFT
  - Efficient computation by 2D box filters & integral images
  - Equivalent quality for object identification
  - GPU implementation available
    - Feature extraction @ 200Hz
      (detector + descriptor, 640×480 img) http://www.vision.ee.ethz.ch/~surf



[Bay, ECCV'06], [Cornelis, CVGPU'08]

Recent Advances in Interest Points

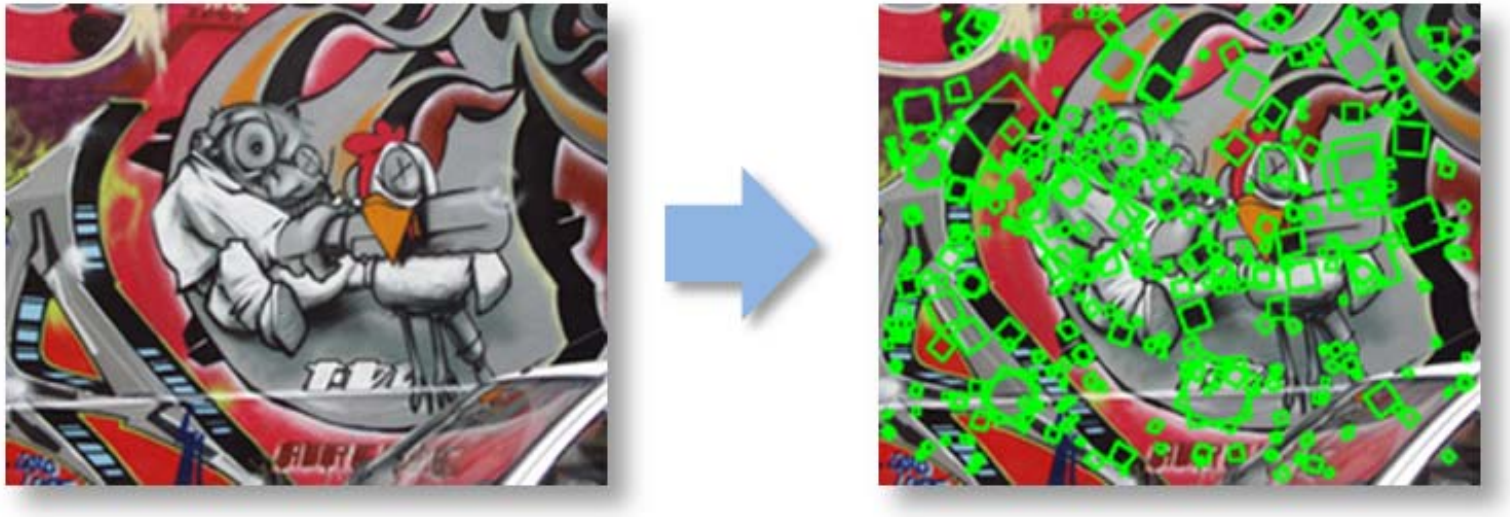
- Binary Descriptors
  - BRIEF: Binary Robust Independent Elementary Features, ECCV 10
  - ORB (Oriented FAST and Rotated BRIEF), CVPR 11
  - BRISK: Binary robust invariant scalable keypoints, ICCV 11
  - Freak: Fast retina keypoint, CVPR 12
  - LIFT: Learned Invariant Feature Transform, ECCV 16



Features from Accelerated Segment Test, ECCV 06

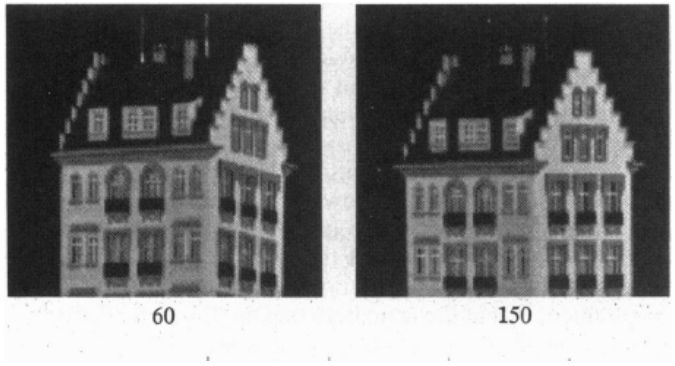
What's to Be Covered Today…

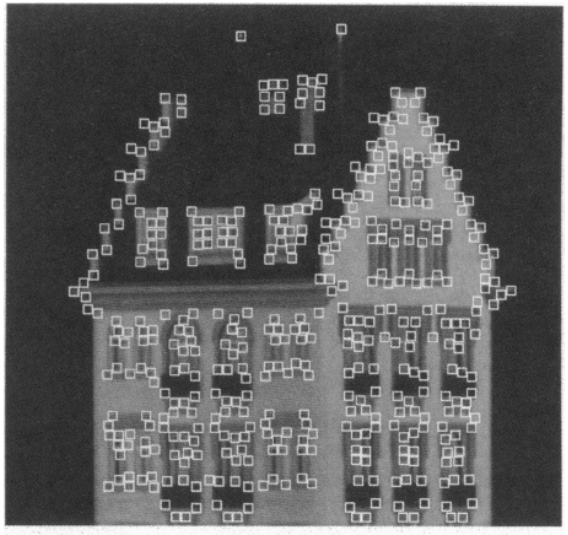
- Interest Points
  - Keypoint Detection & Description
  - Feature Tracking & Optical Flow
  - Image Recognition



Feature Tracking

- Why? E.g., Tracking moving objects, structure from motion, etc.
- Challenges
  - If the amount of motion is small, tracking is relatively easy.
  - Robust interest points (invariant to appearance changes over time)
  - Might need to add/delete interest points over time
  - Efficient tracking across frames
  - Drifting/accumulated errors





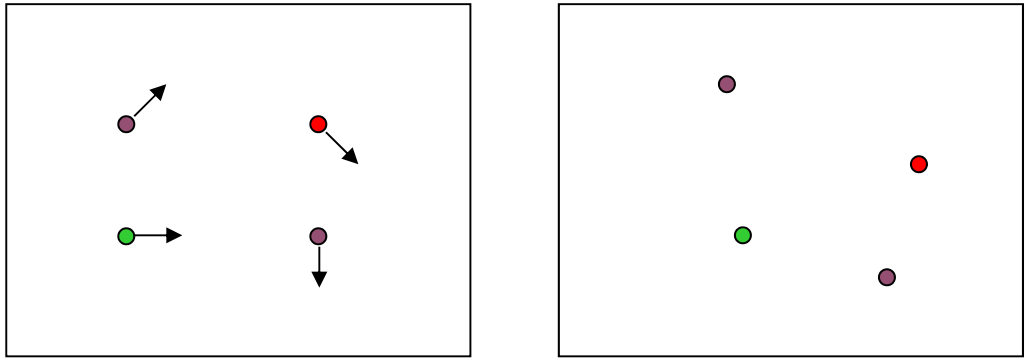
Feature Tracking

- Definition
  - Given two consecutive frames, estimate the translation between points
- Example: Luckas-Kanade Tracker
  - Brightness constancy

    Projection of the same point looks the same in each frame
  - Small motion

    Points do not move very fast.
  - Spatial coherence

    Neighboring points have similar moving patterns.



$I(x,y,t)$ $I(x,y,t+1)$

About Brightness Constancy Constraint

- Equation

$$I(x,y,t) = I(x+u, y+v, t+1)$$

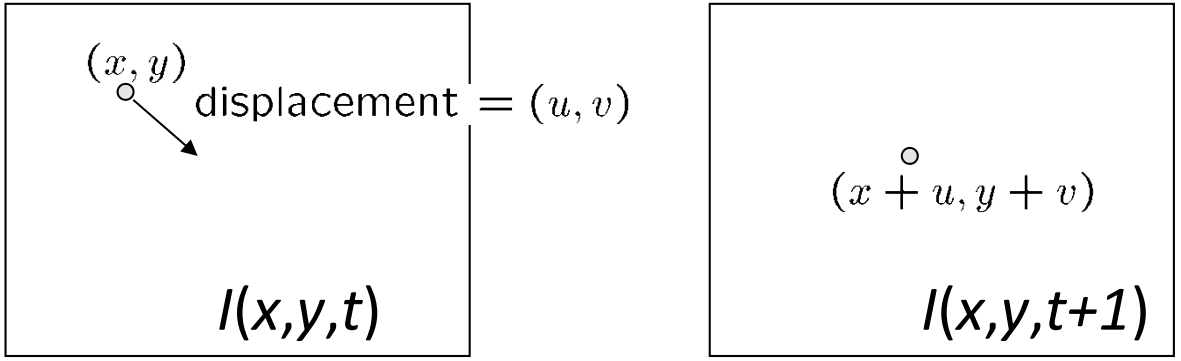
- Take Taylor expansion of $I(x+u, y+v, t+1)$ at (x,y,t):

Image derivative along x Difference over frames

$$I(x+u,y+v,t+1) \approx I(x,y,t) + I_x \cdot u + I_y \cdot v + I_t$$

$$I(x+u,y+v,t+1) - I(x,y,t) = I_x \cdot u + I_y \cdot v + I_t$$

Thus $I_x \cdot u + I_y \cdot v + I_t \approx 0 \qquad \rightarrow \nabla \mathrm{I} \cdot \left[\mathrm{u}\ \ \mathrm{v}\right]^{\mathrm{T}} + \mathrm{I_t} = 0$



About Brightness Constancy Constraint

- Can we use this equation to recover image motion (u, v) at point (x, y)?

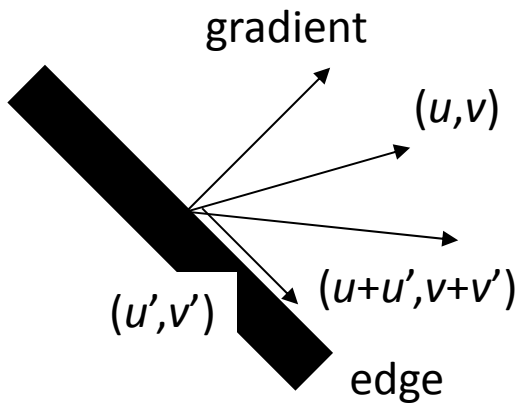
$$\nabla I \cdot \begin{bmatrix} u & v \end{bmatrix}^T + I_t = 0$$

- How many equations & unknowns are in the above equation?

- The component of the motion perpendicular to the gradient (i.e., parallel to the edge) cannot be measured.

If (*u*, *v*) satisfies the equation, so does (*u+u'*, *v+v'*) if

$$\nabla I \cdot \begin{bmatrix} u' & v' \end{bmatrix}^T = 0$$



Solving the Ambiguity

- Can we use this equation to recover image motion (u, v) at point (x, y)?

$$\nabla \mathrm{I} \cdot \left[\mathrm{u}\ \ \mathrm{v}\right]^{\mathrm{T}} + \mathrm{I}_{\mathrm{t}} = 0$$

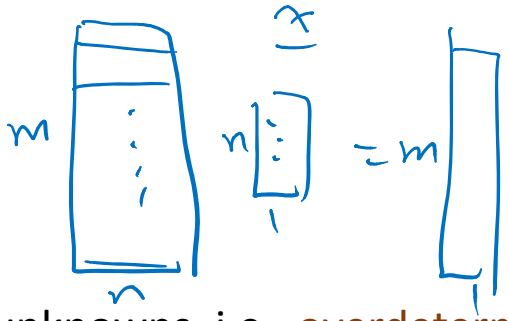
- How to get more equations for a point?
- Spatial coherence constraint
  - Assume the point's neighbors have the same (u,v).
  - If we use a window of 5 x 5 pixels, that gives us 25 equations per point and results in a least-squares problem (**Ax** = **b**).

$$0 = I_t(\mathbf{p_i}) + \nabla I(\mathbf{p_i}) \cdot [u\ \ v]$$

$$\begin{bmatrix} I_x(\mathbf{p_1}) & I_y(\mathbf{p_1}) \\ I_x(\mathbf{p_2}) & I_y(\mathbf{p_2}) \\ \vdots & \vdots \\ I_x(\mathbf{p_{25}}) & I_y(\mathbf{p_{25}}) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} I_t(\mathbf{p_1}) \\ I_t(\mathbf{p_2}) \\ \vdots \\ I_t(\mathbf{p_{25}}) \end{bmatrix}$$

B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. IJCAI, 1981.

Recall that (in Week 2)…



- Given **A x** = **b**, if m > n & rank(**A**) = n…
  - We have more # of equations than # of unknowns, i.e., overdetermined.
  - How to get a desirable **x**?

$$\underline{e} = \underline{b} - \underline{A}\,\underline{x}$$

$$J(\underline{x}) = \frac{1}{2}\cdot(\underline{e}^T\cdot\underline{e}) = \frac{1}{2}(\underline{b}^T - \underline{x}^T\cdot\underline{A}^T)(\underline{b} - \underline{A}\,\underline{x})$$

$$= \frac{1}{2}(\underline{b}^T\cdot\underline{b} - 2\underline{b}^T\underline{A}\cdot\underline{x} + \underline{x}^T\underline{A}^T\underline{A}\,\underline{x})$$

1×m m×n n×1

$$\frac{\partial J}{\partial \underline{x}} = -\underline{A}^T\cdot\underline{b} + \underline{A}^T\cdot\underline{A}\cdot\underline{x} = 0$$

$$(\underline{A}^T\underline{A})\,\underline{x} = \underline{A}^T\underline{b} \Rightarrow \underline{x} = \boxed{(\underline{A}^T\underline{A})^{-1}\cdot\underline{A}^T}\,\underline{b}$$

n×n

full rank

pinv of $\underline{A}$

Conditions for Solvability

$$\begin{bmatrix} I_x(\mathbf{p_1}) & I_y(\mathbf{p_1}) \\ I_x(\mathbf{p_2}) & I_y(\mathbf{p_2}) \\ \vdots & \vdots \\ I_x(\mathbf{p_{25}}) & I_y(\mathbf{p_{25}}) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} I_t(\mathbf{p_1}) \\ I_t(\mathbf{p_2}) \\ \vdots \\ I_t(\mathbf{p_{25}}) \end{bmatrix}$$

- Optimal (u, v) satisfies Lucas-Kanade equation

$$\begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} \sum I_x I_t \\ \sum I_y I_t \end{bmatrix}$$

$A^T A$ $\qquad$ $A^T b$

- When is this solvable? i.e., what are the good points to track?
  - $\mathbf{A}^T\mathbf{A}$ (2nd moment matrix) should be

  - We will *not* cover the following details in this course that…
    - $\mathbf{A}^T\mathbf{A}$ should not be too small due to noise

    - eigenvalues λ_1 and λ_2 of $\mathbf{A}^T\mathbf{A}$ should not be too small
    - $\mathbf{A}^T\mathbf{A}$ should be well-conditioned

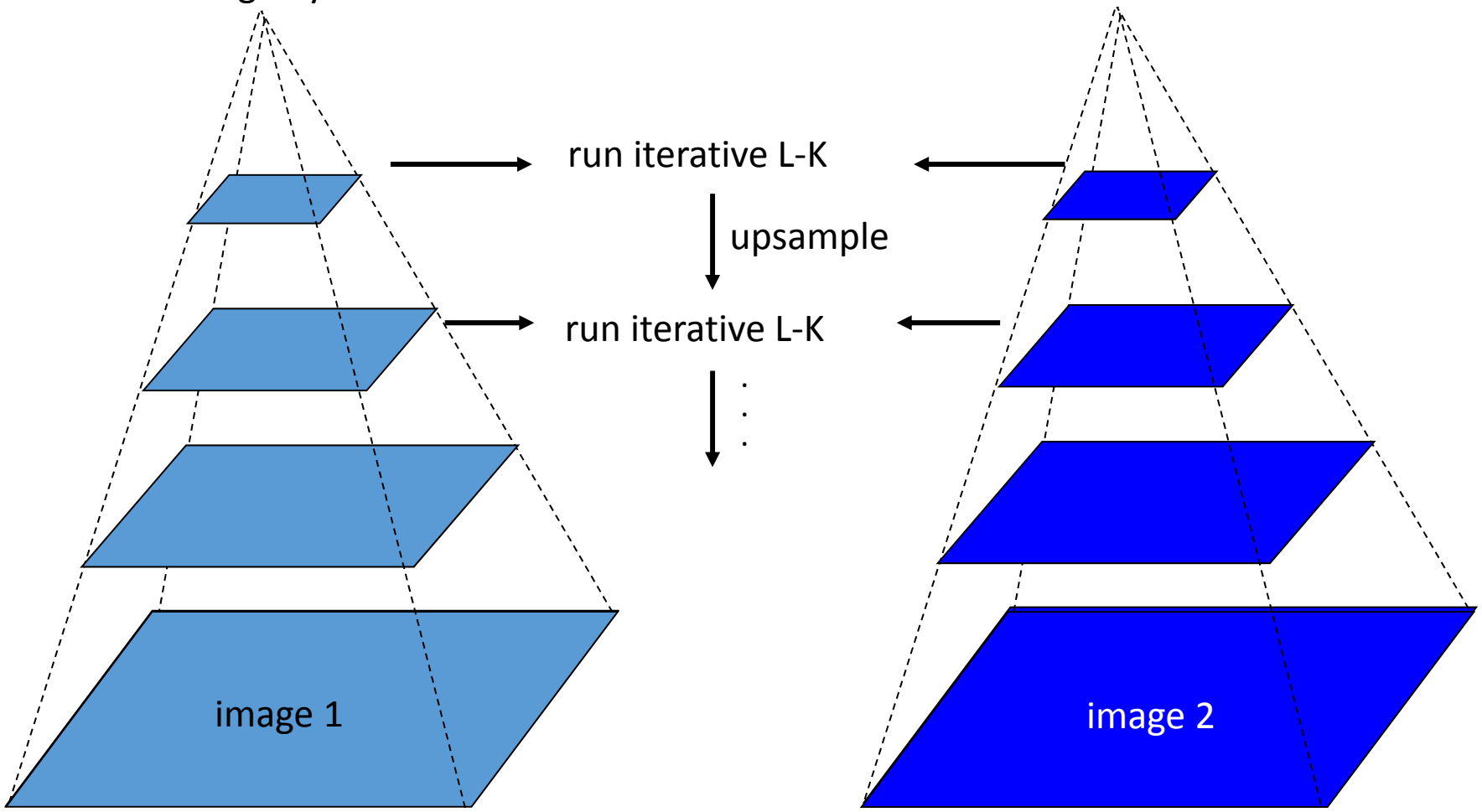
    - λ_1 / λ_2 should not be too large (λ_1 = larger eigenvalue)

    (This is exactly the criteria of Harris corner detector.)

B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. IJCAI, 1981.

Dealing with Larger Movements

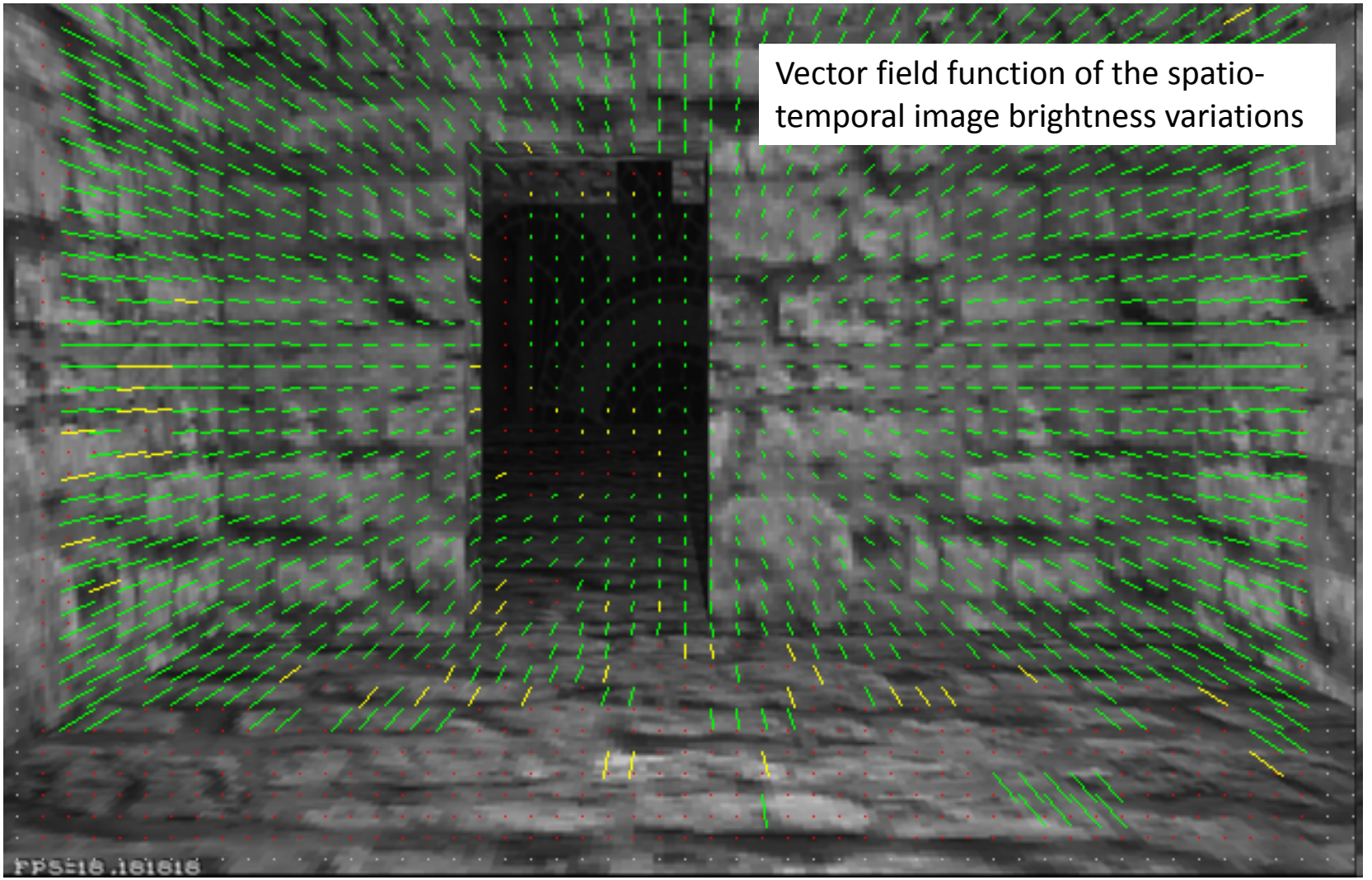
- Coarse-to-fine registration/search
  - Image Pyramid



Gaussian pyramid of image frame t

Gaussian pyramid of image frame t+1

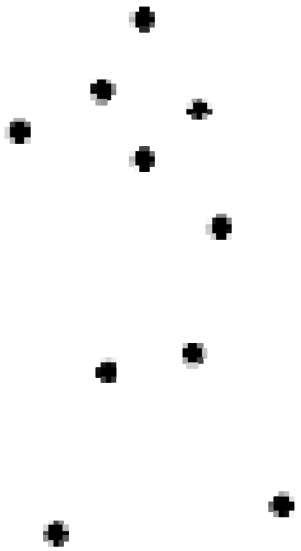
Optical Flow



Picture courtesy of Selim Temizer - Learning and Intelligent Systems (LIS) Group, MIT

Motion and Perceptual Organization

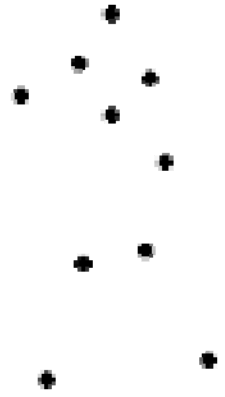
- Even “impoverished” motion data can evoke a strong percept

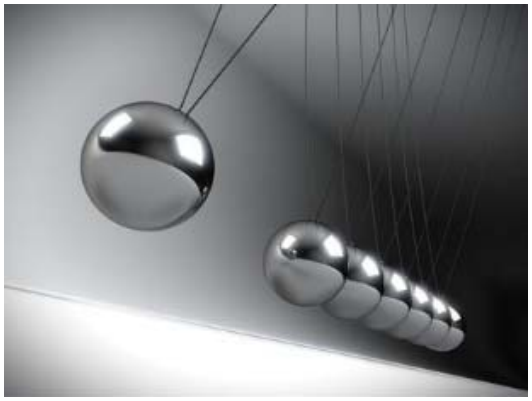


G. Johansson, “Visual Perception of Biological Motion and a Model For Its Analysis",
Perception and Psychophysics 14, 201-211, 1973.

Why Motion?

- Estimating 3D structure
- Segmenting objects based on motion cues
- Learning and tracking dynamical models
- Recognizing events and activities
- Improving video quality (motion stabilization)





Optical Flow

- Definition
  - Optical flow is the apparent motion of brightness patterns in an image.
- Ideally…
  - Optical flow would be the same as the motion field.
- Be careful!
  - Apparent motion might be due to lighting changes w/o actual motion.
  - E.g., a uniform rotating sphere under fixed lighting vs. a stationary sphere under moving illumination

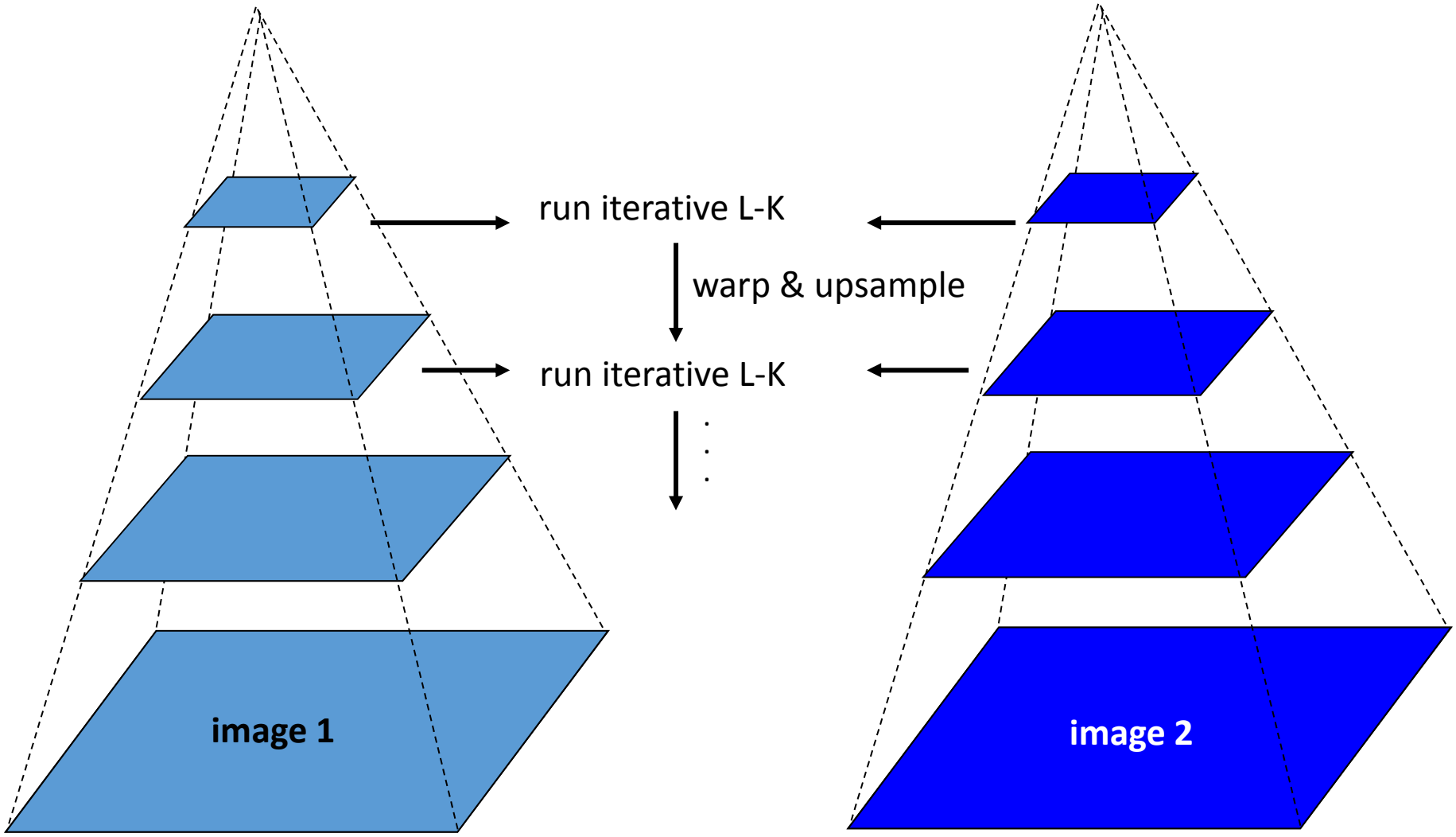
Lukas-Kanade Optical Flow

- Same as Lucas-Kanade feature tracking, but for each pixel
  - As we saw, works better for textured pixels.
- Operations can be done one frame at a time, rather than pixel by pixel
  - Efficient!

Iterative Refinement

- Iterative Lukas-Kanade Algorithm
  1. Estimate displacement at each pixel by solving Lucas-Kanade eqns
  2. Warp I(t) towards I(t+1) using the estimated flow field
     - Basically, just interpolation
  3. Repeat until convergence!

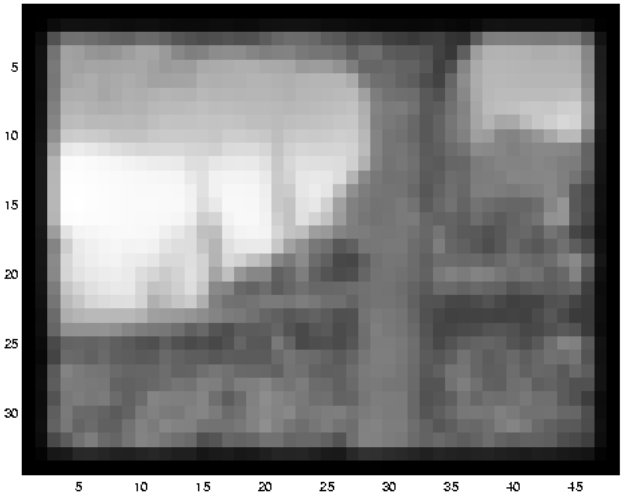
Coarse-to-Fine Lukas-Kanade Optical Flow

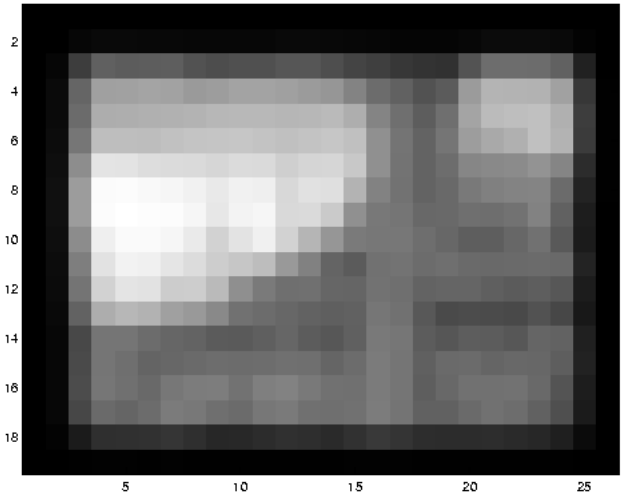


Example

* From Khurram Hassan-Shafique CAP5415 Computer Vision 2003

Example





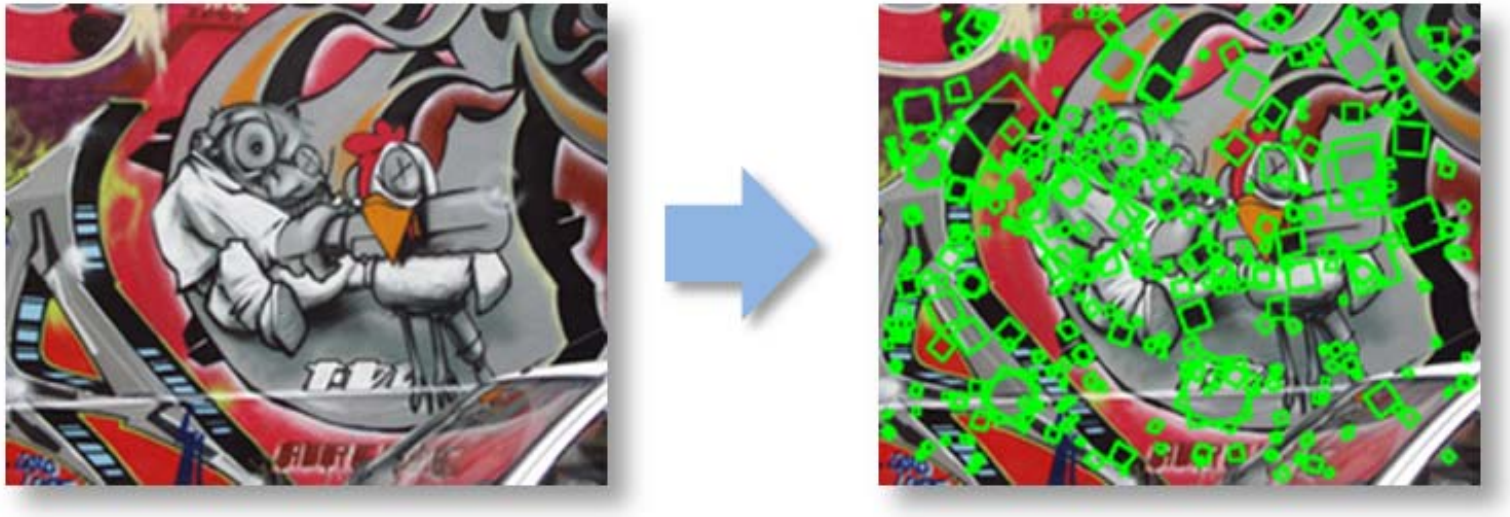
* From Khurram Hassan-Shafique CAP5415 Computer Vision 2003

Errors in LK

- The motion is large.
  - Possible solution: keypoint matching
- A point does not move like its neighbors.
  - Possible solution: region-based matching
- Brightness constancy does not hold.
  - Possible solution: Gradient constancy

What's to Be Covered Today…

- Interest Points
  - Keypoint Detection & Description
  - Feature Tracking & Optical Flow
  - Image Recognition



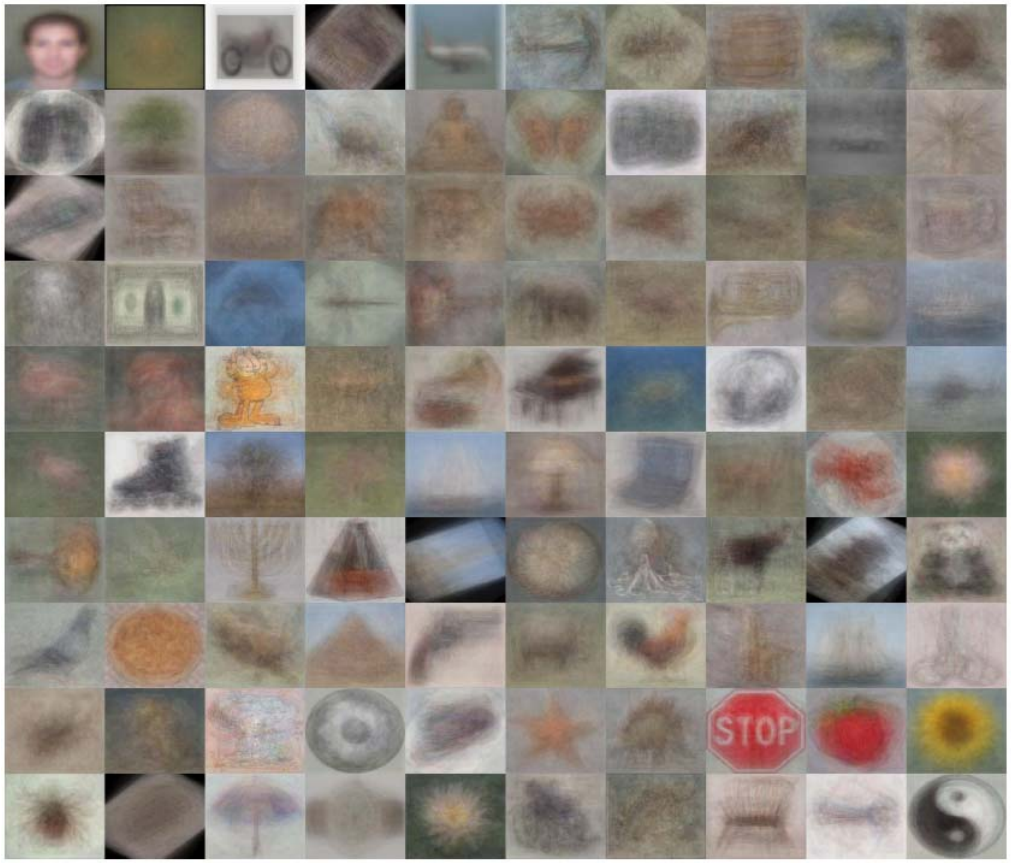
Many slides from J.-B. Huang, J. Hayes, N Snavely, K. Grauman, and D. Hoiem

Image Categorization

- ~~Fine-grained image classification~~

obj

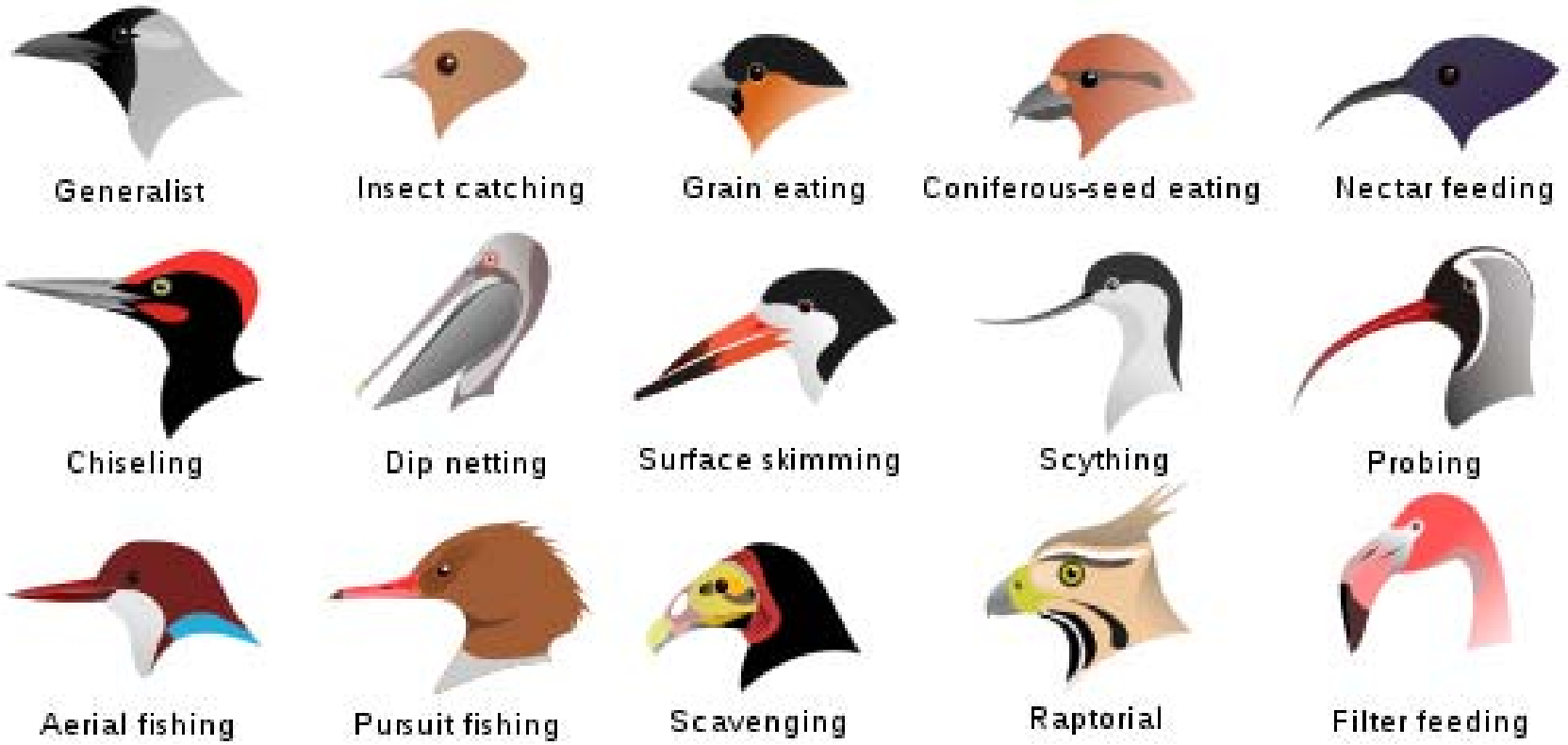
recog.



Average Object Images of Caltech 101

Image Categorization

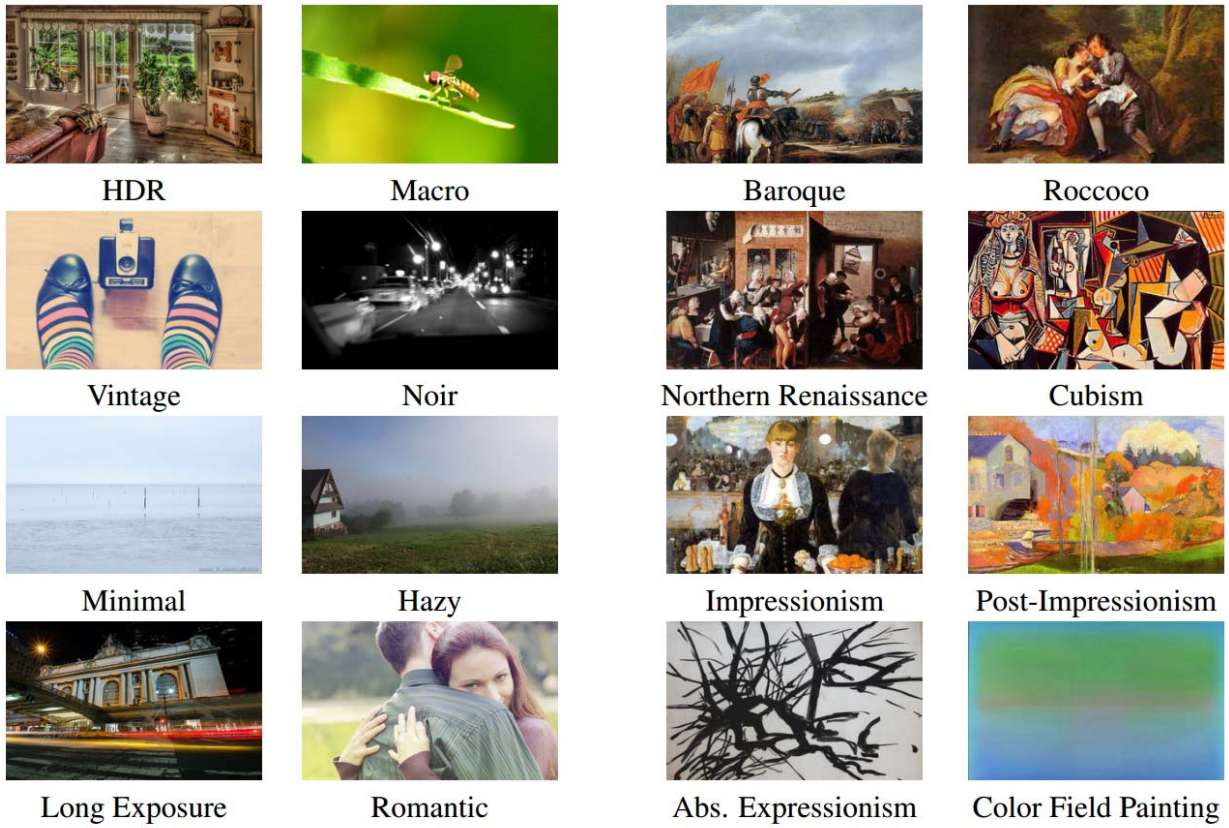
- Object recognition fine-grained recg.



Visipedia Project

Image Categorization

- Image style recognition



[Karayev et al. BMVC 2014]

Image Categorization

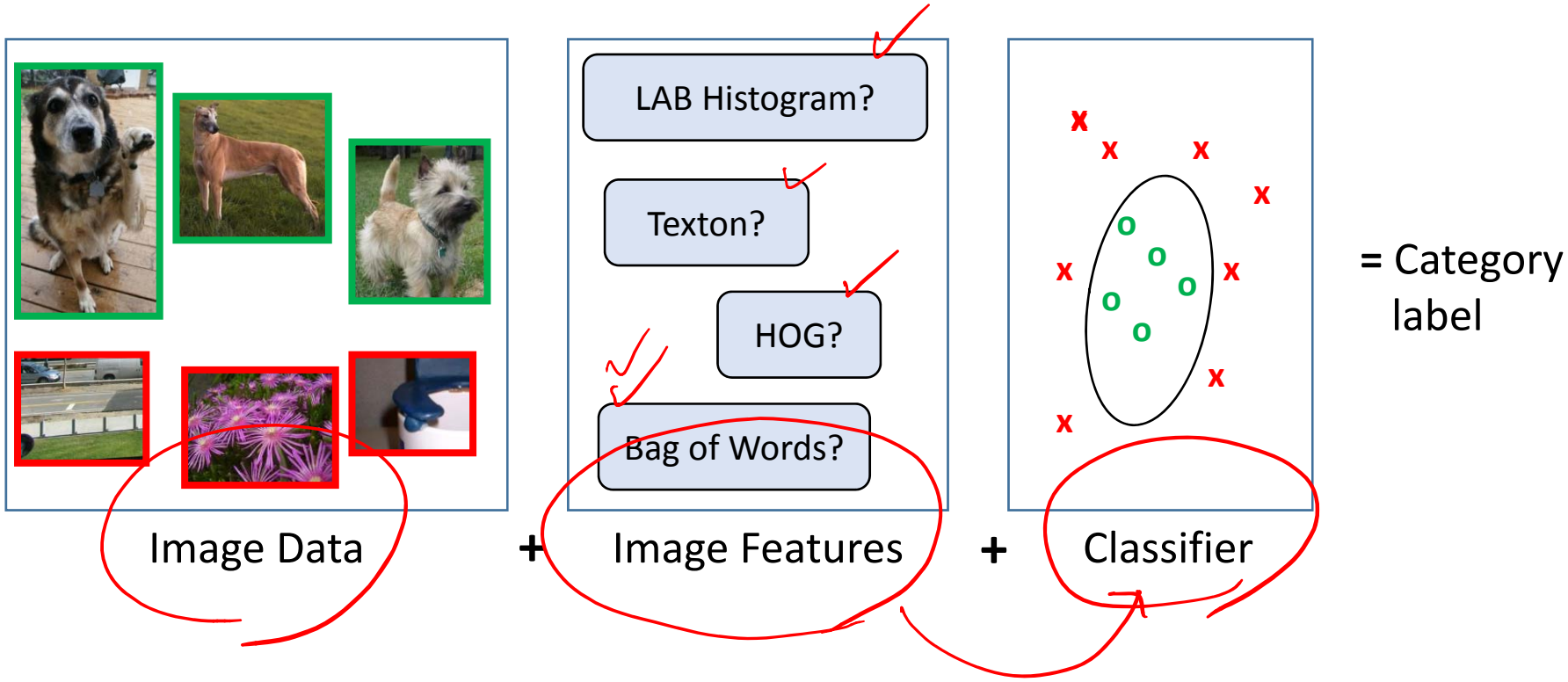
- Dating historical photos

1940 1953 1966 1977

[Palermo et al. ECCV 2012]

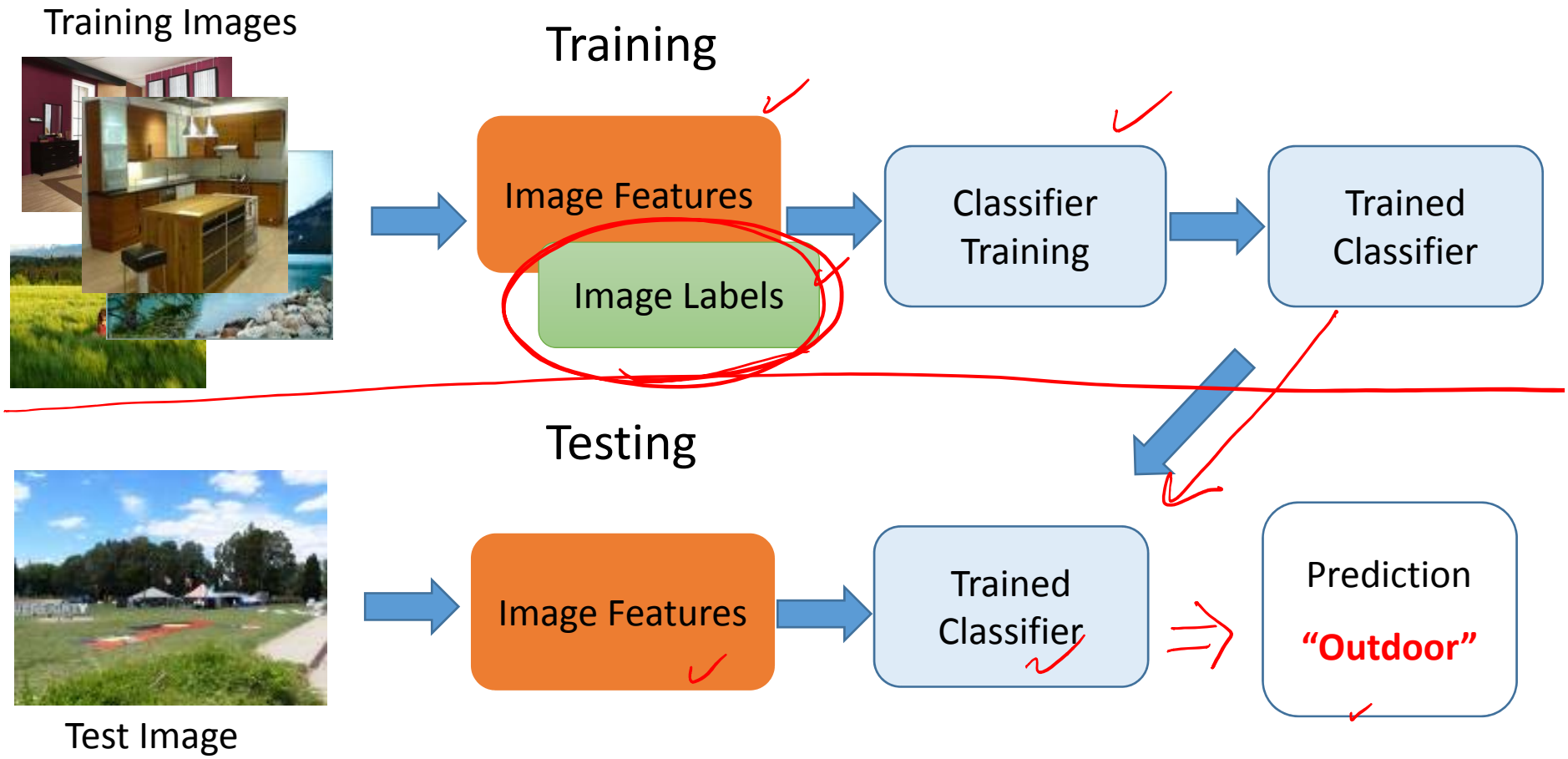
Supervised Learning for Visual Classification

- General framework



Supervised Learning for Visual Classification

- Training vs. Testing Phases

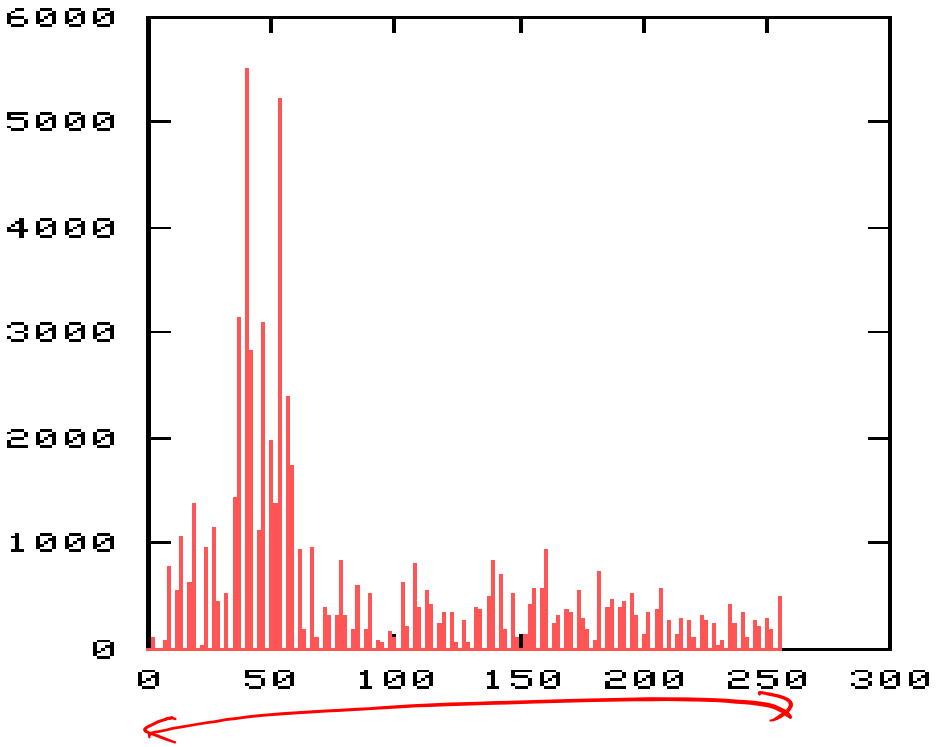


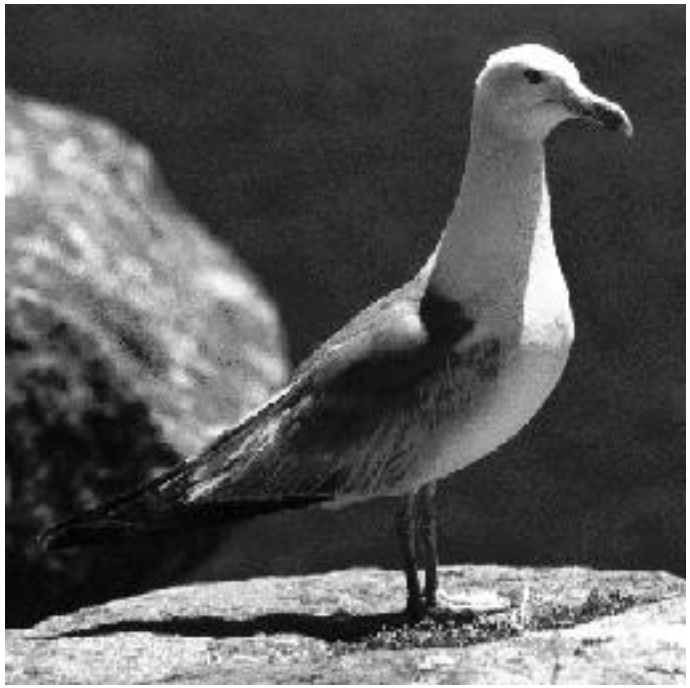
What Are the Right Features?

- Depending on the task of interest!
- Possible choices
  - Object: shape
    - Local shape info, shading, shadows, texture
  - Scene : geometric layout
    - linear perspective, gradients, line segments
  - Material properties: albedo, feel, hardness
    - Color, texture
  - Action: motion
    - Optical flow, tracked points

Image Representation: Histograms

- Global histogram
  - Possible to describe color, texture, depth, or even interest points!

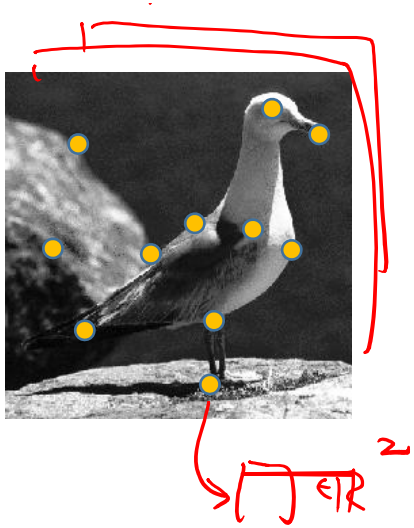




Images from Dave Kauchak

Image Representation: Histograms

- Take images with 2D features/descriptors as an example



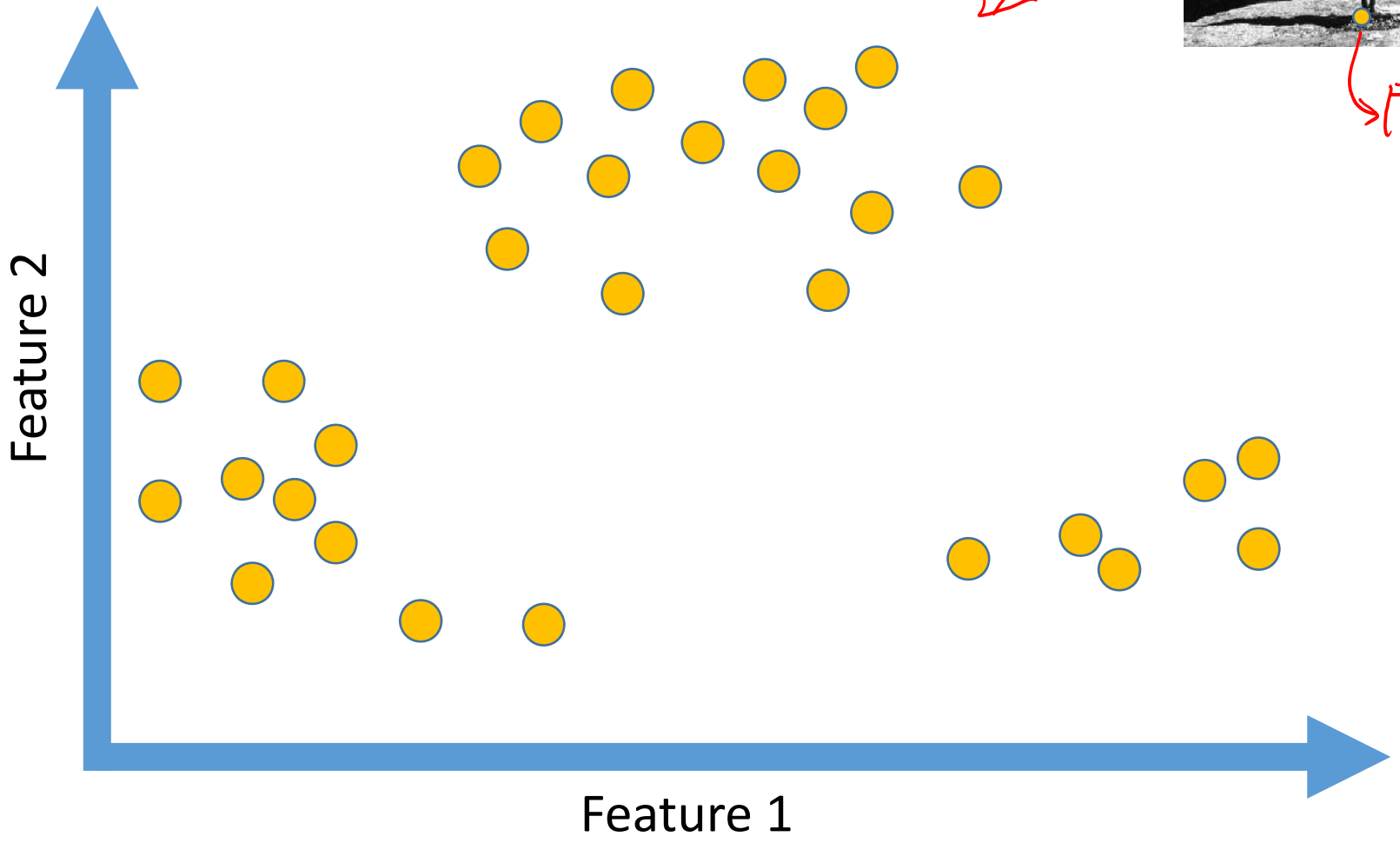


Image Representation: Histograms

- # of occurrence of data in each bin
- Marginal histogram of feature 1

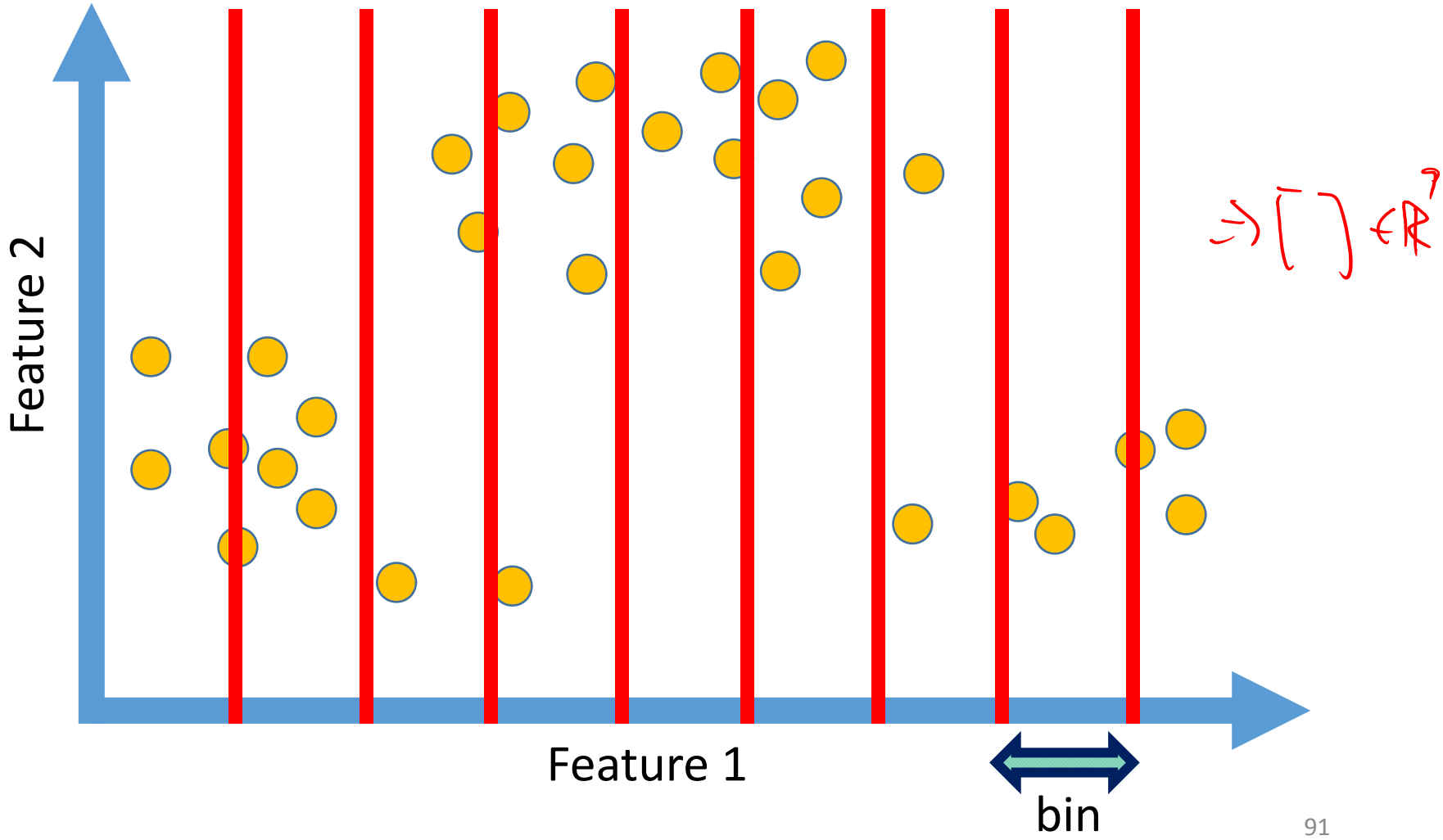


Image Representation: Histograms

- # of occurrence of data in each bin
- Marginal histogram of feature 2

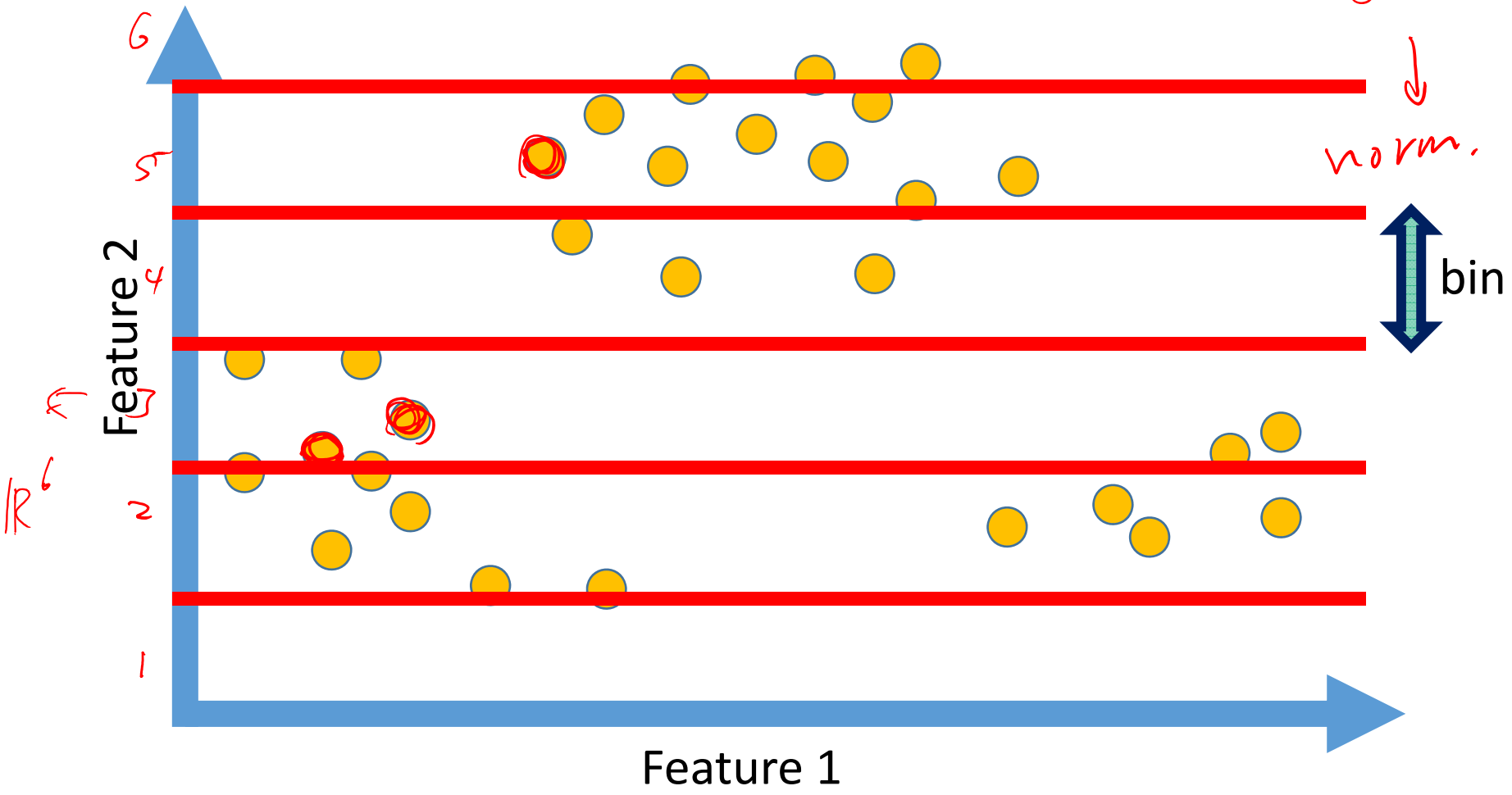


Image Representation: Histograms

- Better modeling (quantization) of multi-dimensional data
- Clustering

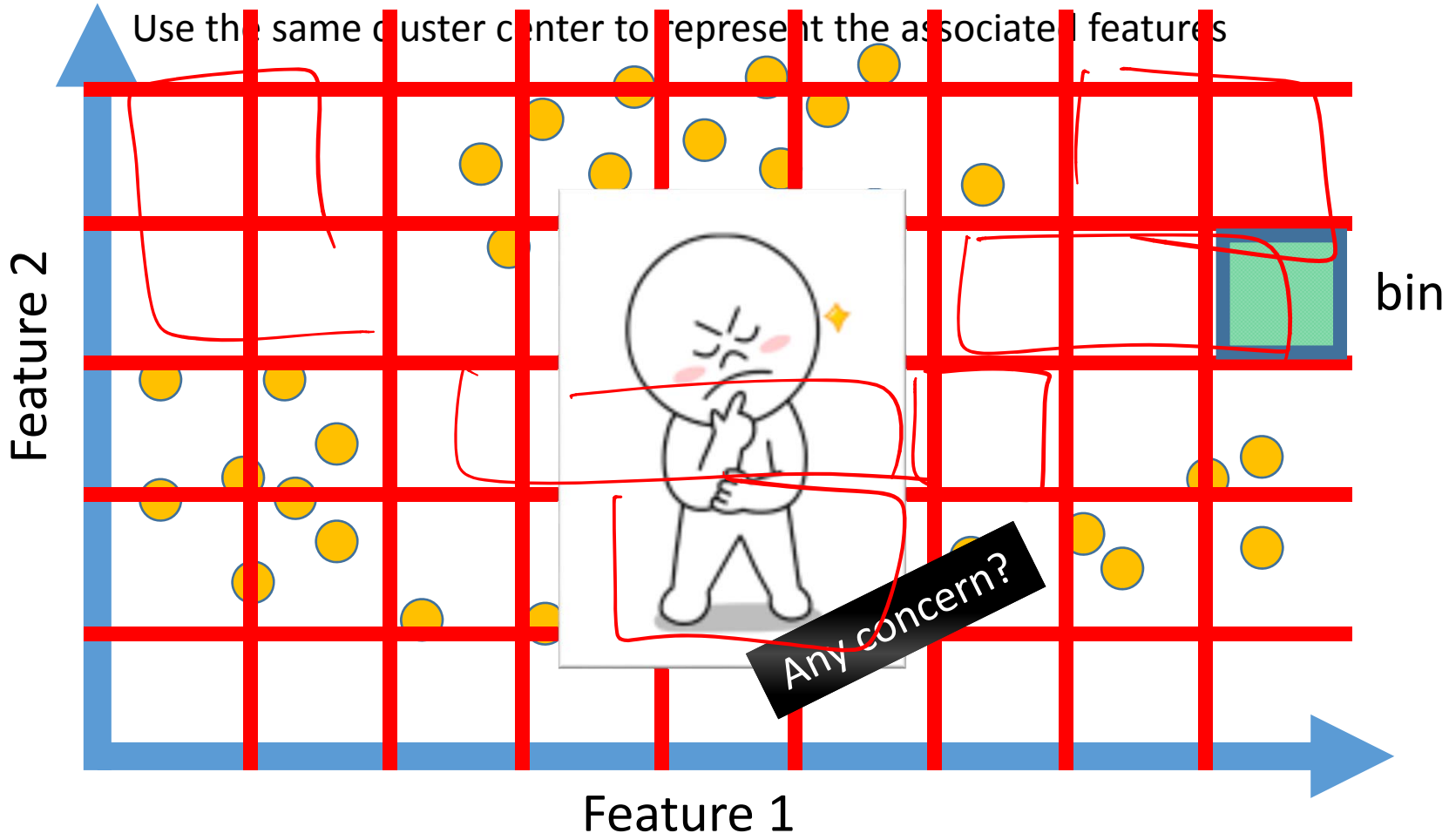
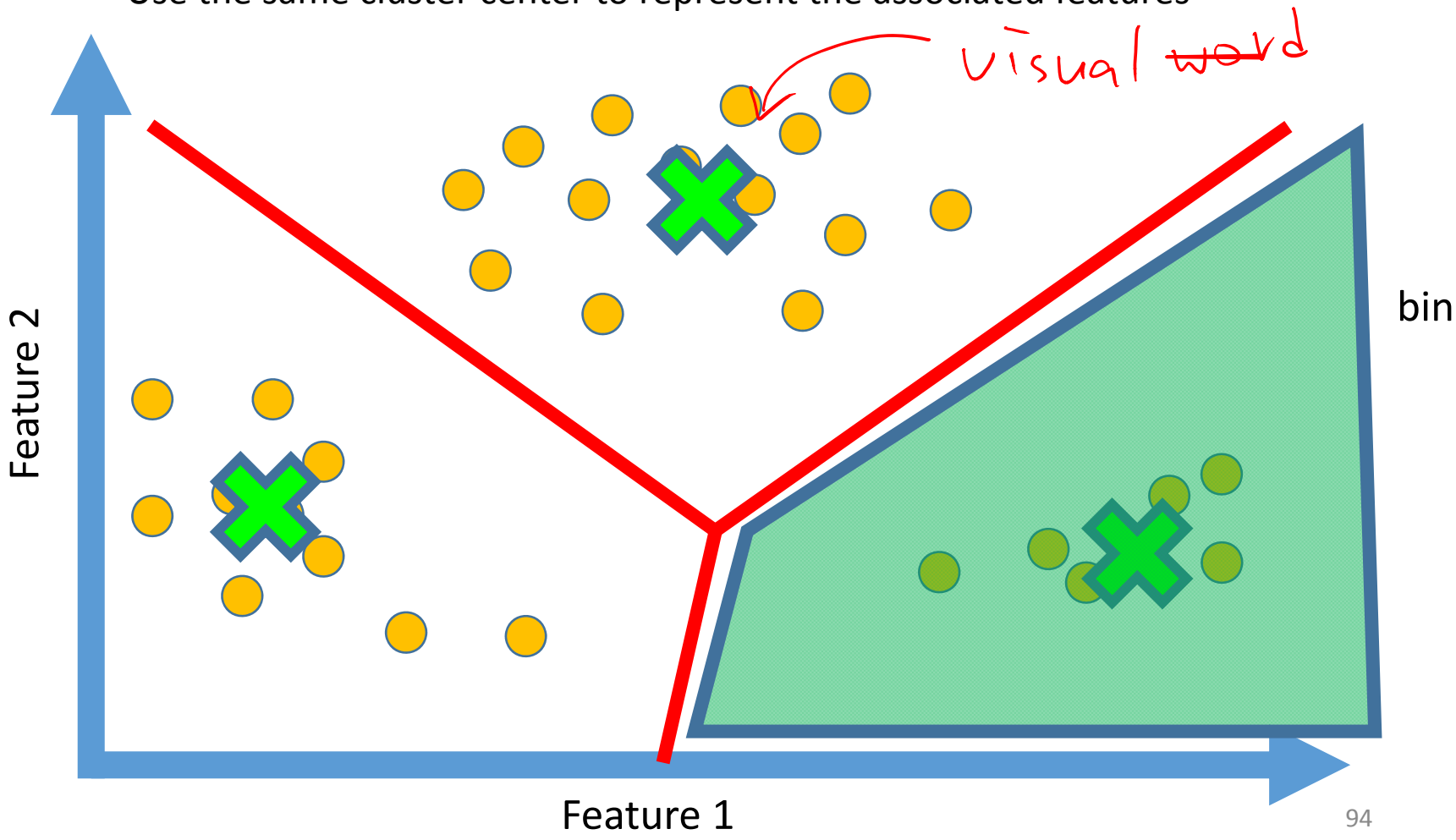


Image Representation: Histograms

- Better modeling (quantization) of multi-dimensional data
- Clustering
  - Use the same cluster center to represent the associated features



Remarks on Histogram-Based Image Representation

- Quantization
  - Grids vs. clusters

Fewer Bins
Need less data
Coarser representation

More Bins
Need more data
Finer representation

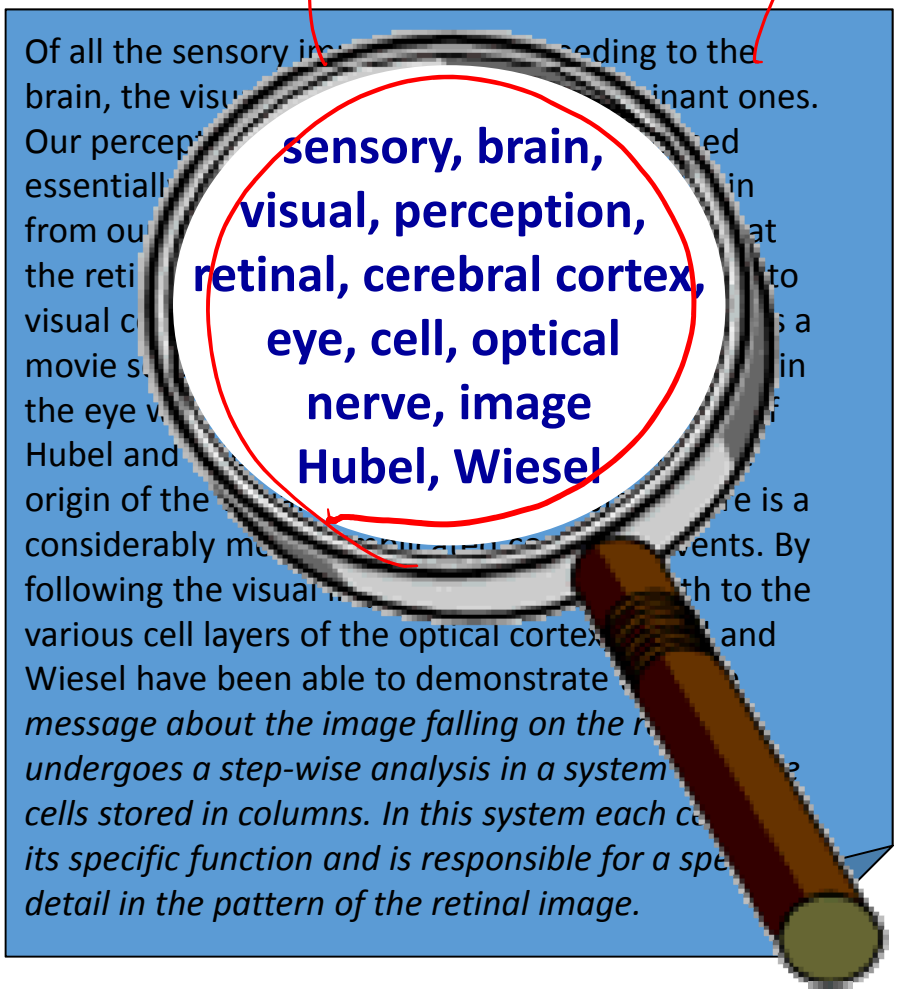
- Possible distance metrics
  - Euclidean distance
  - Histogram intersection kernel
  - Chi-squared distance
  - Earth mover's distance
    (min cost to transform one distribution to another)

$$\text{histint}(h_i, h_j) = 1 - \sum_{m=1}^{K} \min\left(h_i(m), h_j(m)\right)$$

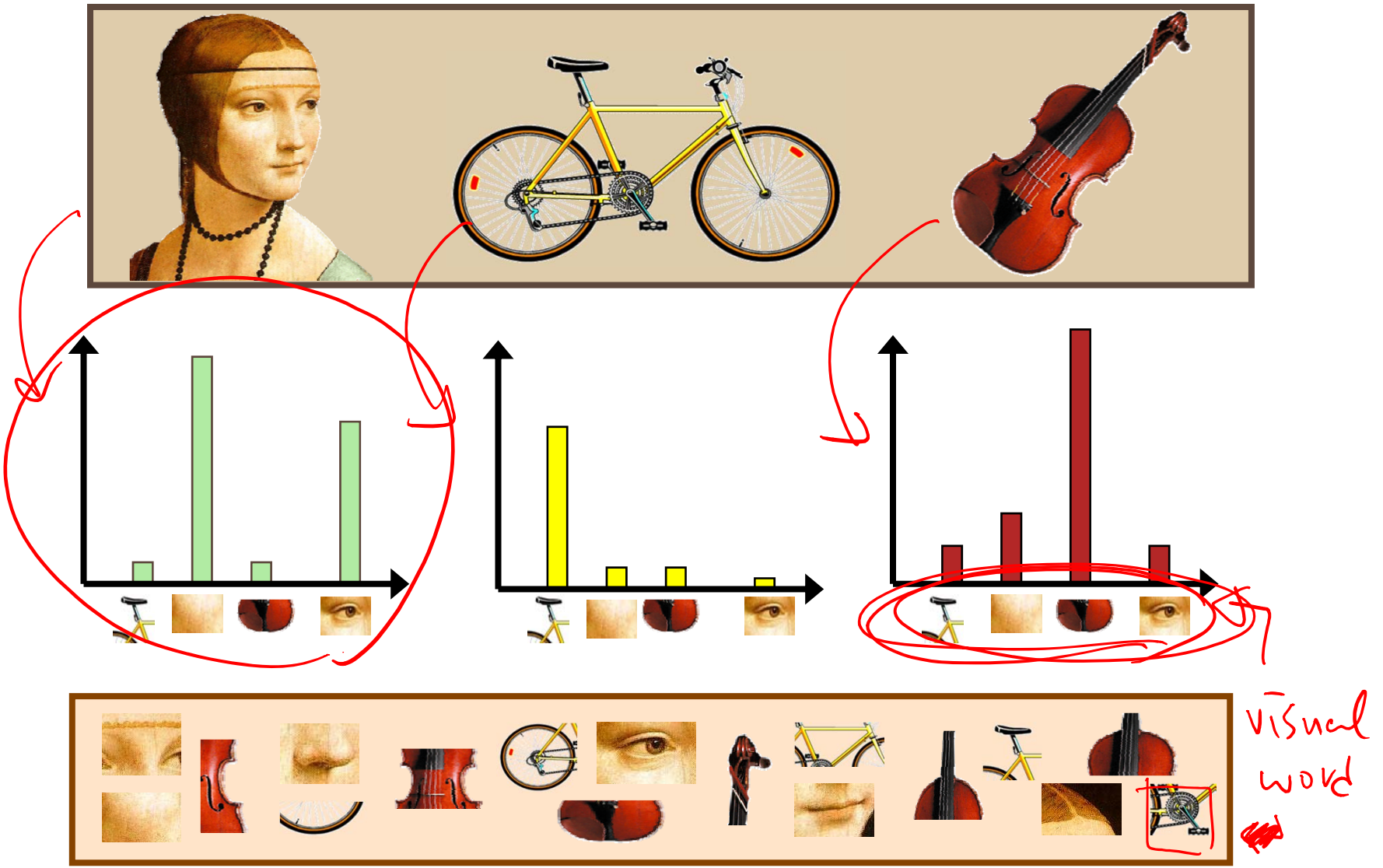
$$\chi^2(h_i, h_j) = \frac{1}{2} \sum_{m=1}^{K} \frac{[h_i(m) - h_j(m)]^2}{h_i(m) + h_j(m)}$$

Bag-of-Words Models for Image Classification

- Analogy to document categorization



Bag of Words (or Visual Words)



Bag-of-Words for Image Classification

- Training

Interest point detection

Training images

Clustering (dictionary learning)

k = 1

k = 2

k = 3

visual word

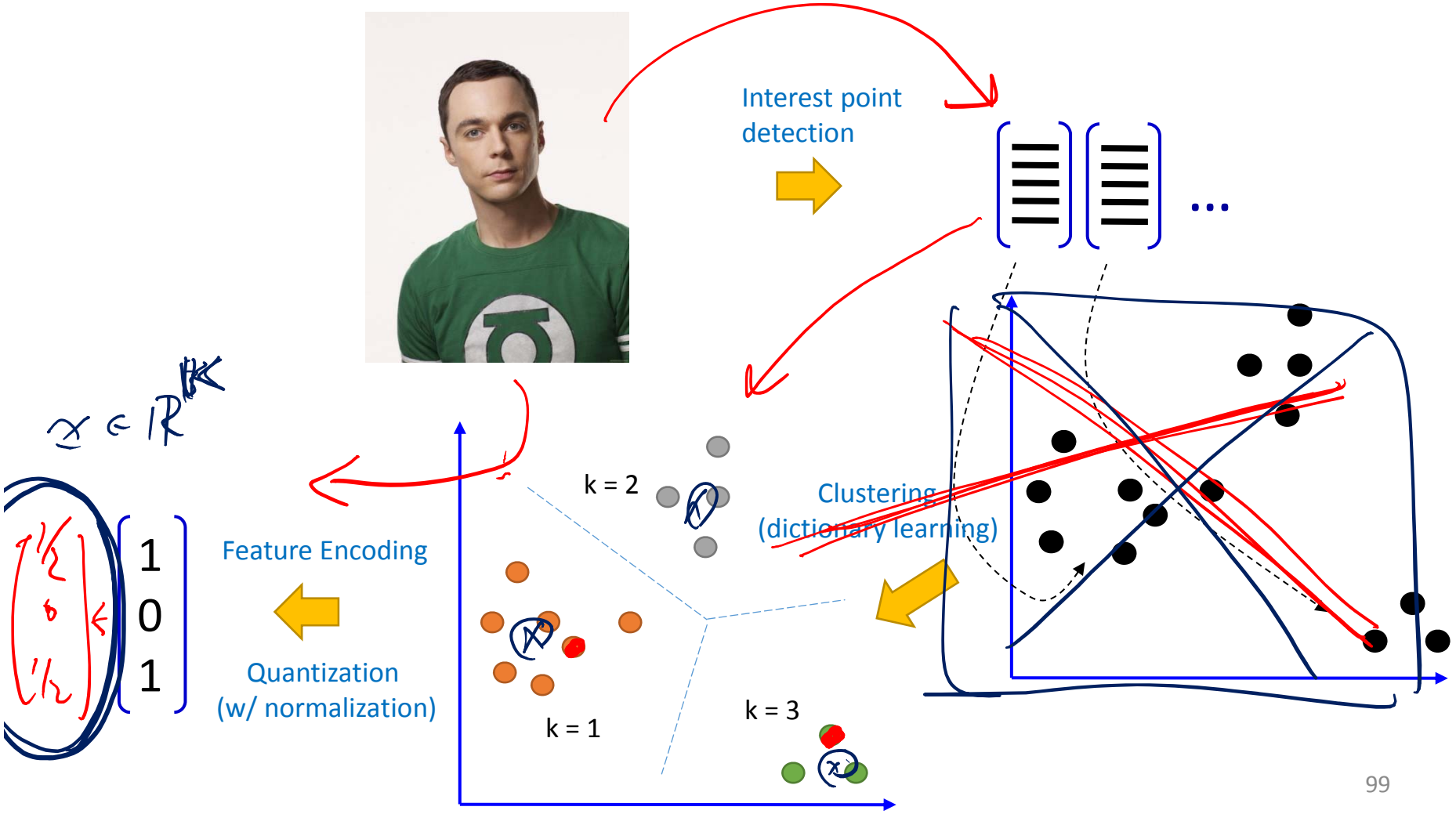
Feature Encoding

Quantization (w/ normalization)

2
1
0

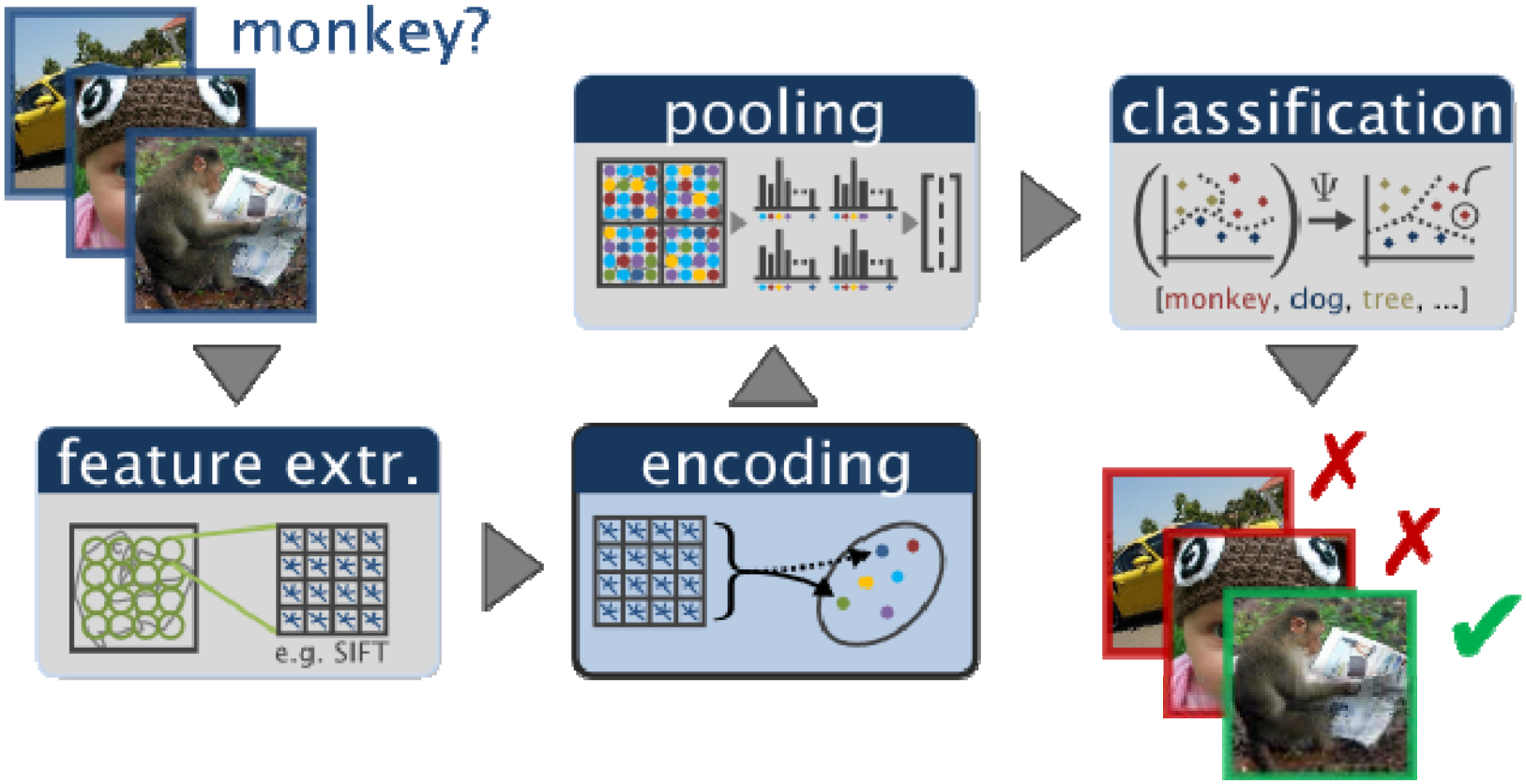
Bag-of-Words for Image Classification

- Testing



Bag-of-Words for Image Classification

- Overview

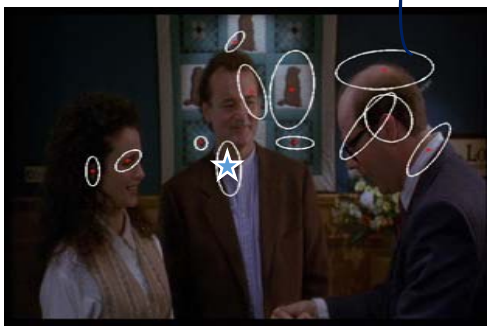


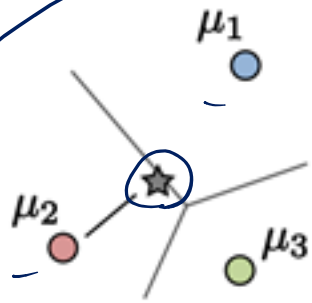
[Chatfieldet al. BMVC 2011]

About Feature Encoding for Bag-of-Words

- Hard vs. soft assignments to clusters

(testing)

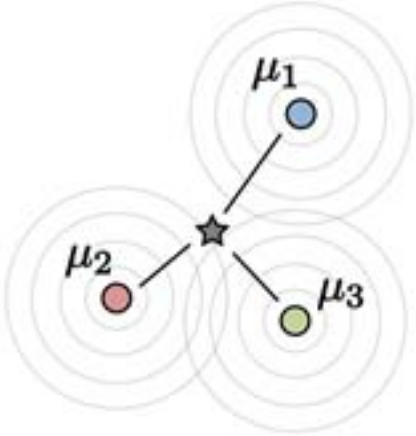




$$\min_{\alpha_t} \| x_t - A\alpha_t \|_2^2 \quad \text{s.t. } \|\alpha_t\|_1 = 1,\ \|\alpha_t\|_0 = 1$$

$\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$

100

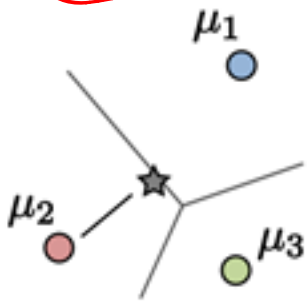


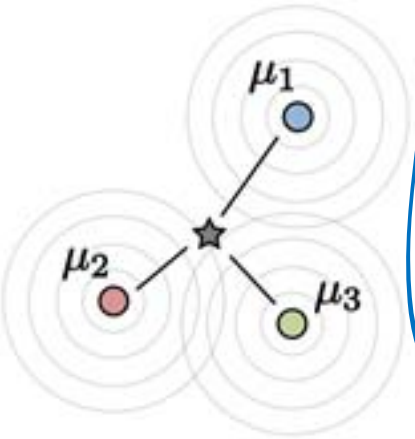
$$\min_{\alpha'_t} \| x_t - A\alpha'_t \|_2^2 \quad \text{s.t. } \|\alpha'_t\|_1 = 1$$

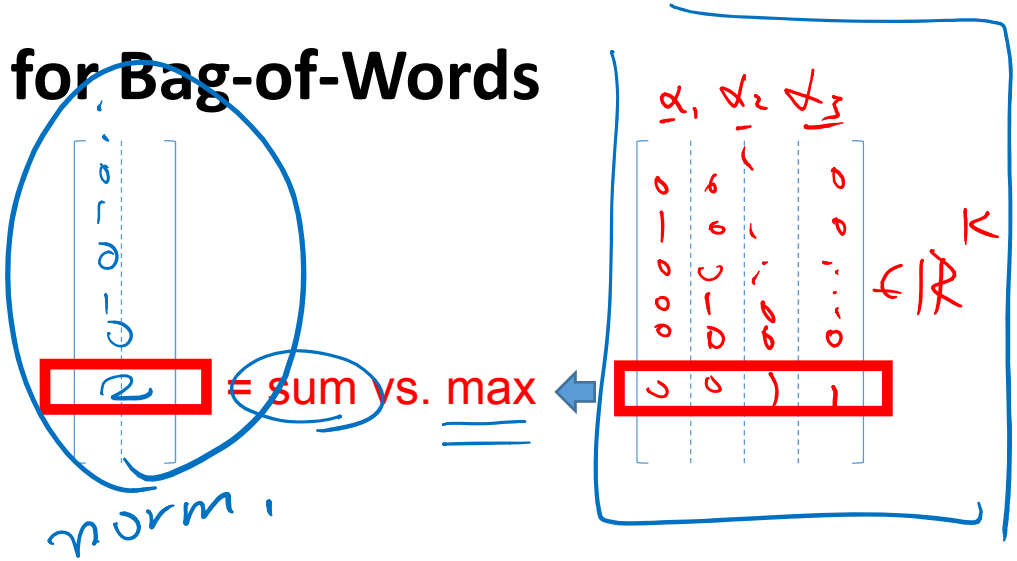
$\begin{bmatrix} 0.2 \\ 0.5 \\ 0.3 \end{bmatrix}$

About Feature Encoding for Bag-of-Words

- Sum vs. max pooling

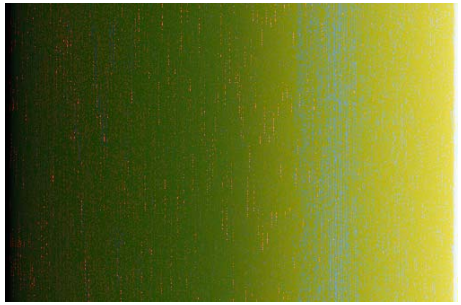


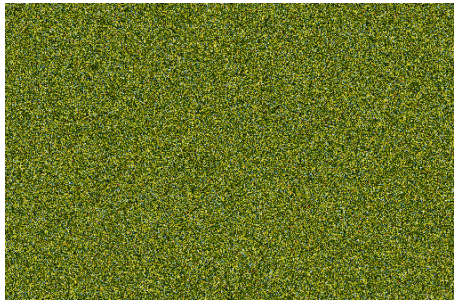


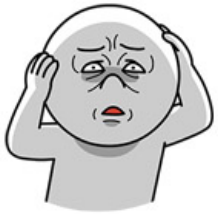


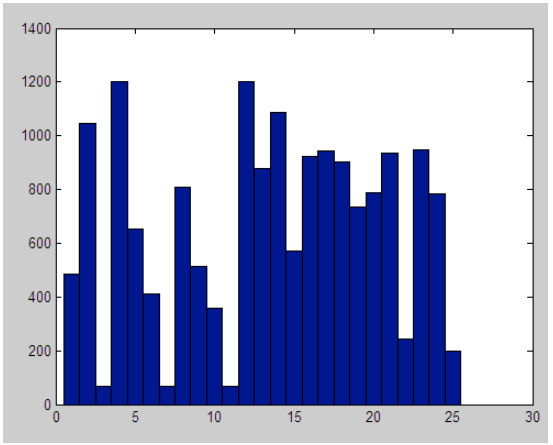
Final Remarks on BoW

- What's the limitation?
  - Loss of... spatial info.
  - What's the possible solution?



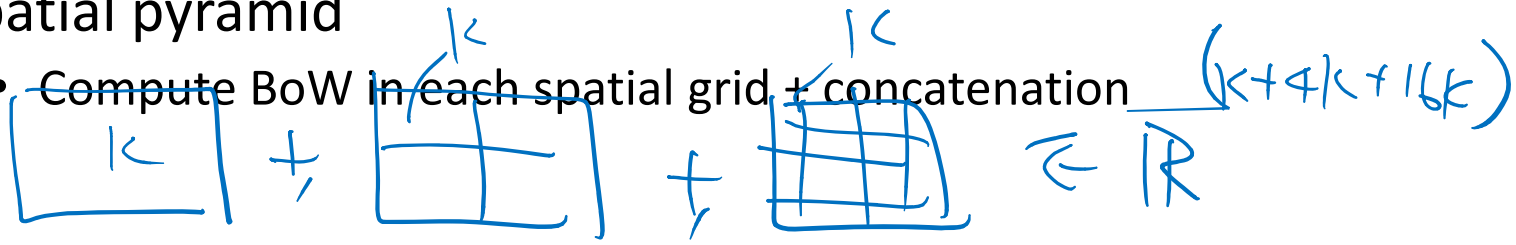


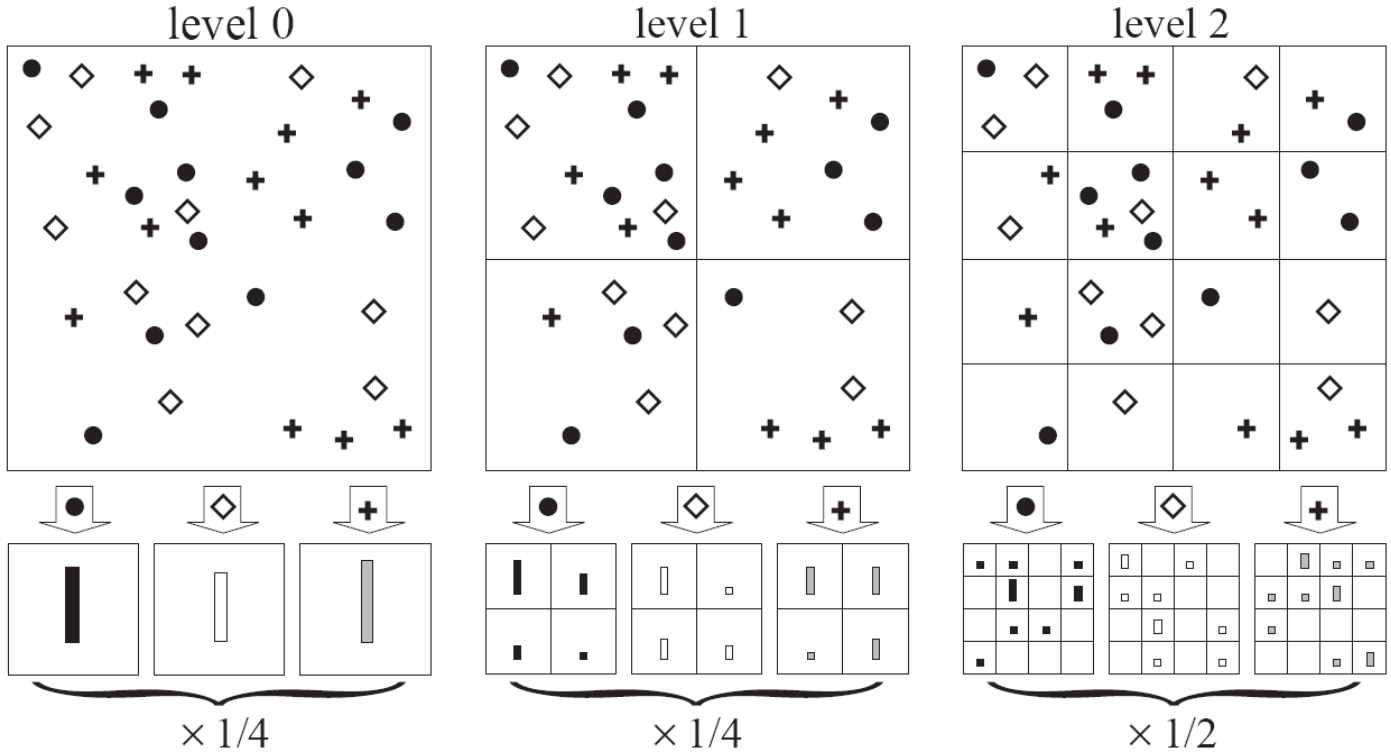




Final Remarks on BoW

- Spatial pyramid
  - ~~Compute~~ BoW ~~in each~~ spatial grid + concatenation

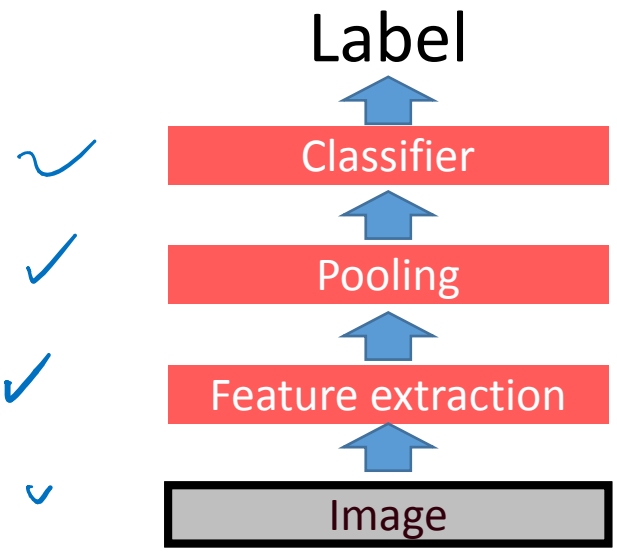


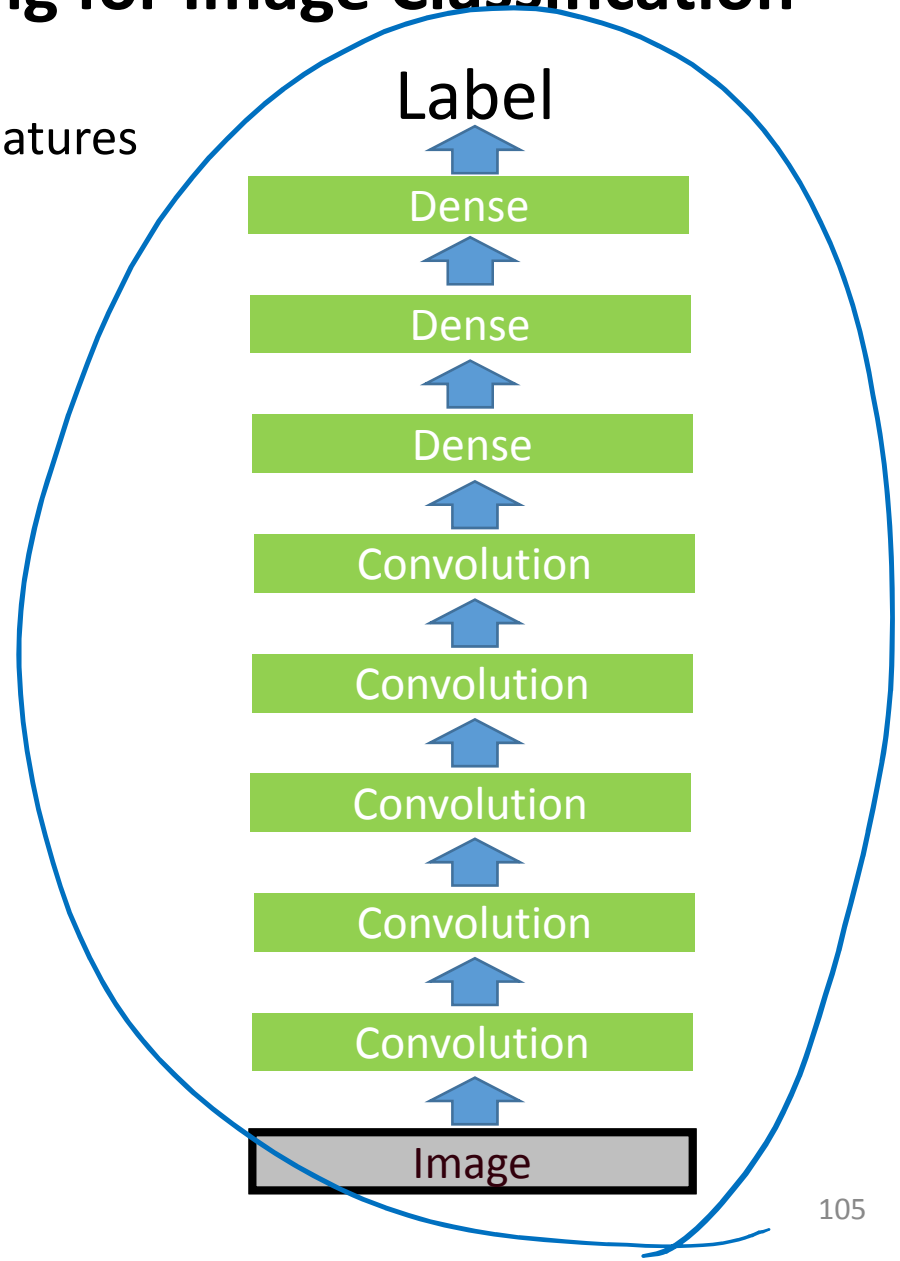


[Lazebnik et al. CVPR 2006]

Shallow vs. Deep Learning for Image Classification

- Engineered vs. deeply learned features
  - Lots of data + GPU





What We Learned Today…

- Image Representation
  - Image Pyramid
- Interest Points
  - Keypoint Detection & Description
  - Feature Tracking & Optical Flow
  - Image Recognition
- HW 2 is out and due 4/11 Wed 1am.
- Next time, we're finally going deep!