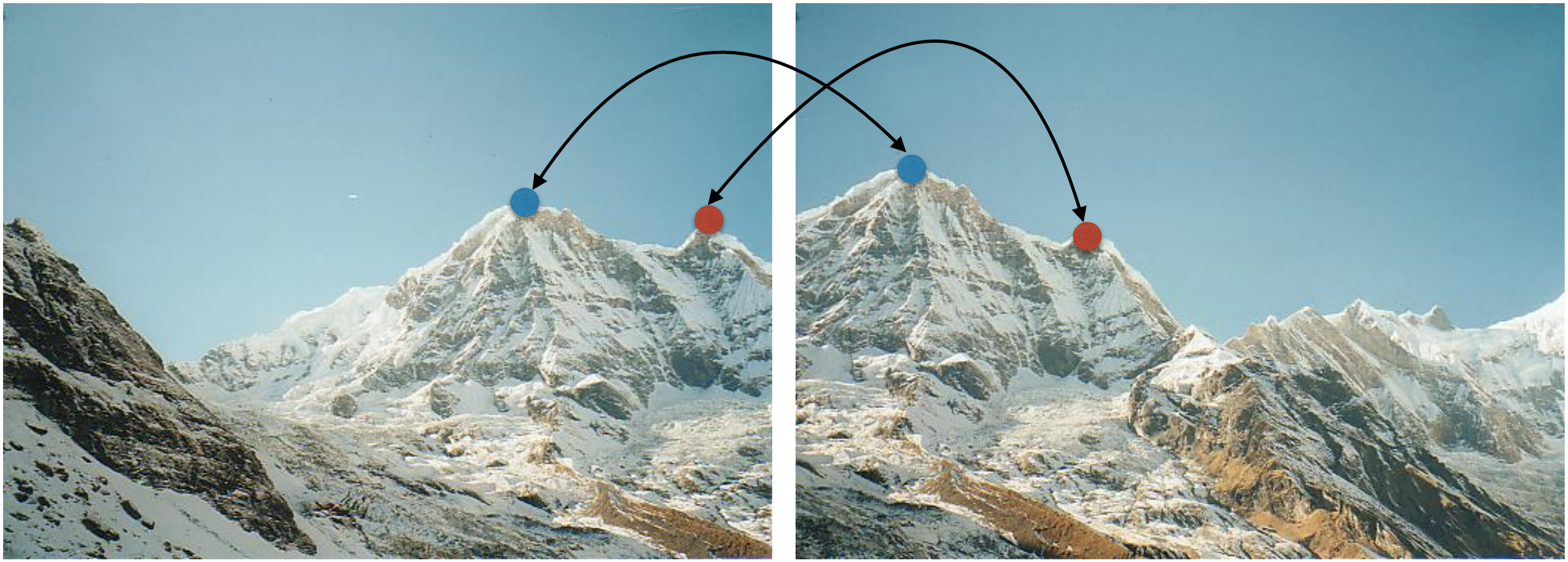
Local Feature Detection

Faisal Qureshi
faisal.qureshi@uoit.ca

Interest Points

Slides from Derek Hoiem, Svetlana Lazebnik, Antonio Torralba, Steve Seitz, David Forsyth, David Lowe, Fei-Fei Li, and James Hays.

Motivation for Feature Extraction



Panorama Stitching

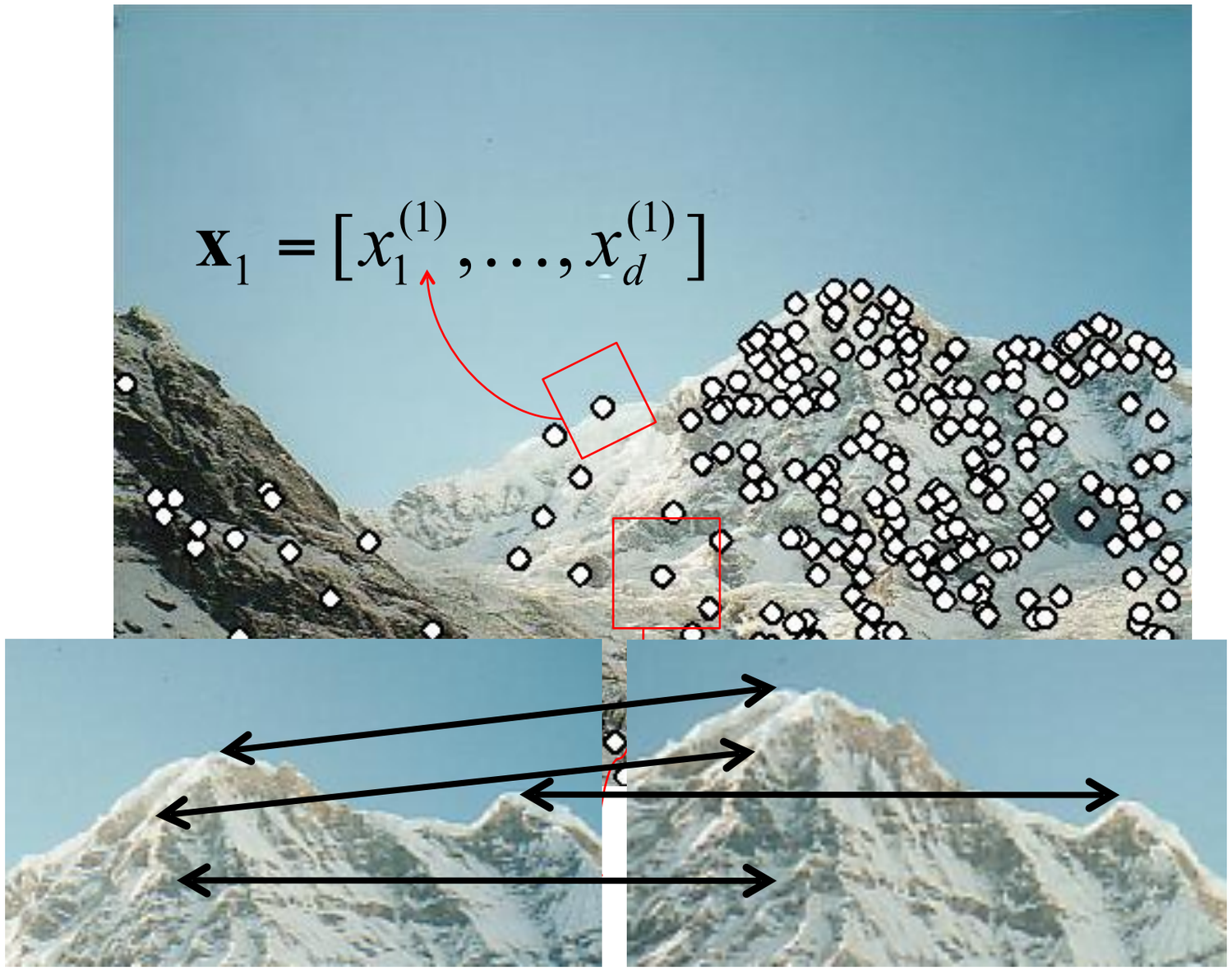
Requires that we find “corresponding locations” in two images to compute *homography*

How to find “corresponding locations” automatically?

Detection
Identify interest points

Description
Extract features around interest points

Matching
Match features in two images



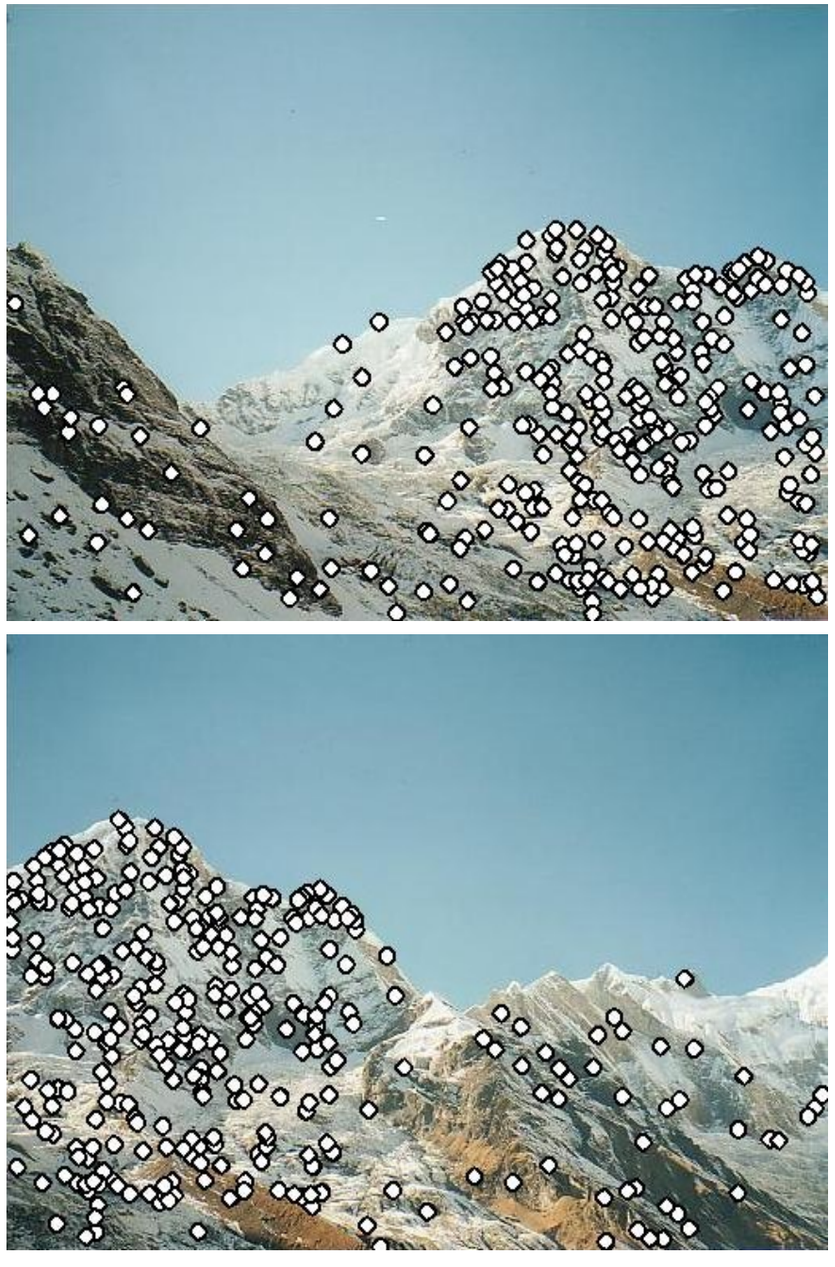
Characteristics of a Good Feature

- Repeatability — feature is invariant to geometric, lighting, etc. changes

- Saliency or distinctiveness

- Compactness — efficiency, many fewer features than the number of pixels in the image

- Locality — robustness to clutter and occlusion, a feature should only occupy a small area of an image



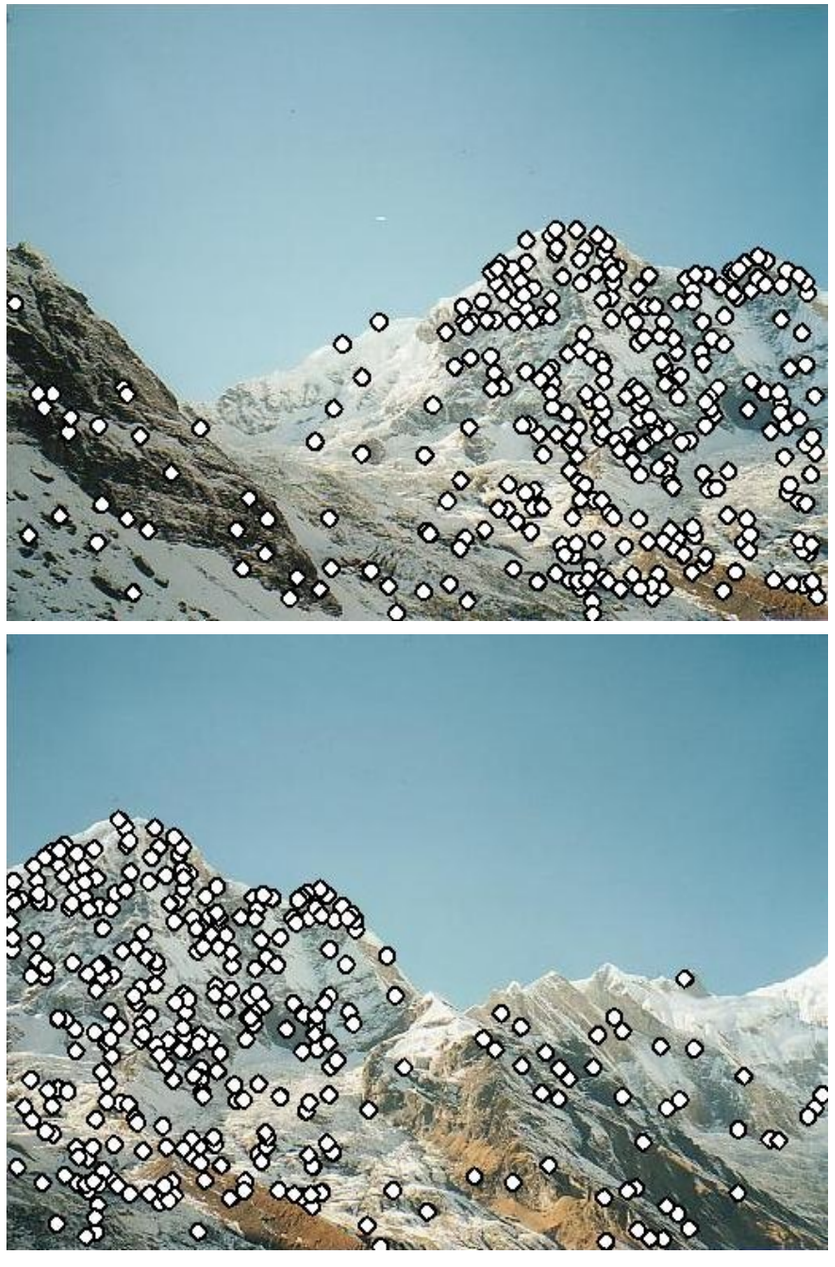
Characteristics of a Good Feature

- Repeatability — feature is invariant to geometric, lighting, etc. changes

- Saliency or distinctiveness

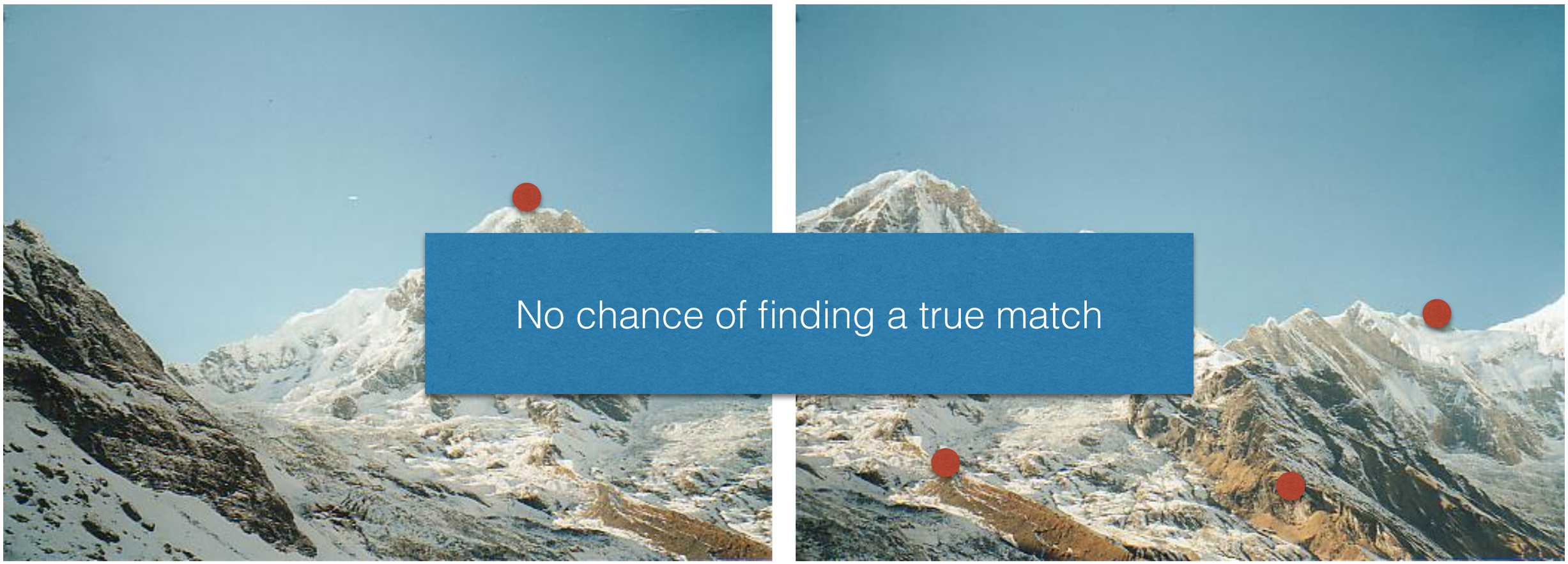
- Compactness — efficiency, many fewer features than the number of pixels in the image

- Locality — robustness to clutter and occlusion, a feature should only occupy a small area of an image



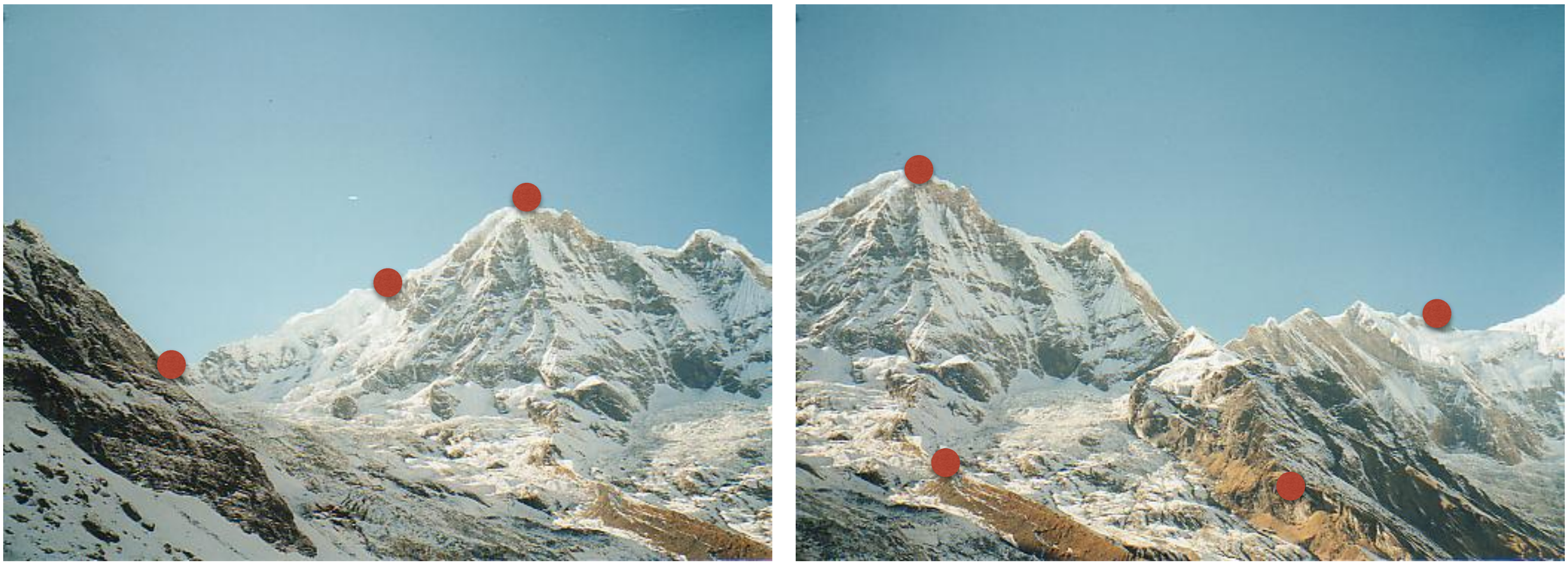
Repeatability

- We need to find at least some of the same points in two images to any chance of finding true matches



Repeatability

- We need to find at least some of the same points in two images to any chance of finding true matches



Detection process run independently on two images should return at least some of the corresponding locations

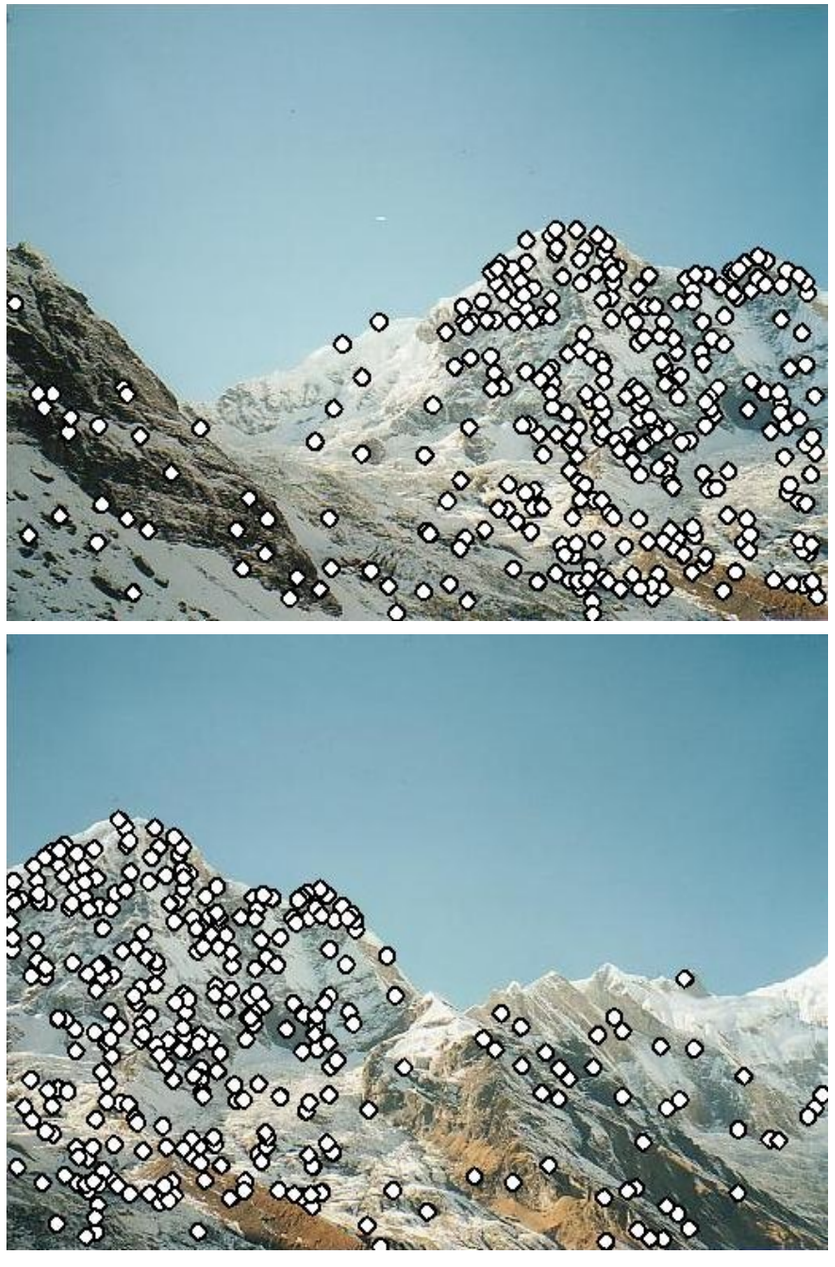
Characteristics of a Good Feature

- Repeatability — feature is invariant to geometric, lighting, etc. changes

- Saliency or distinctiveness

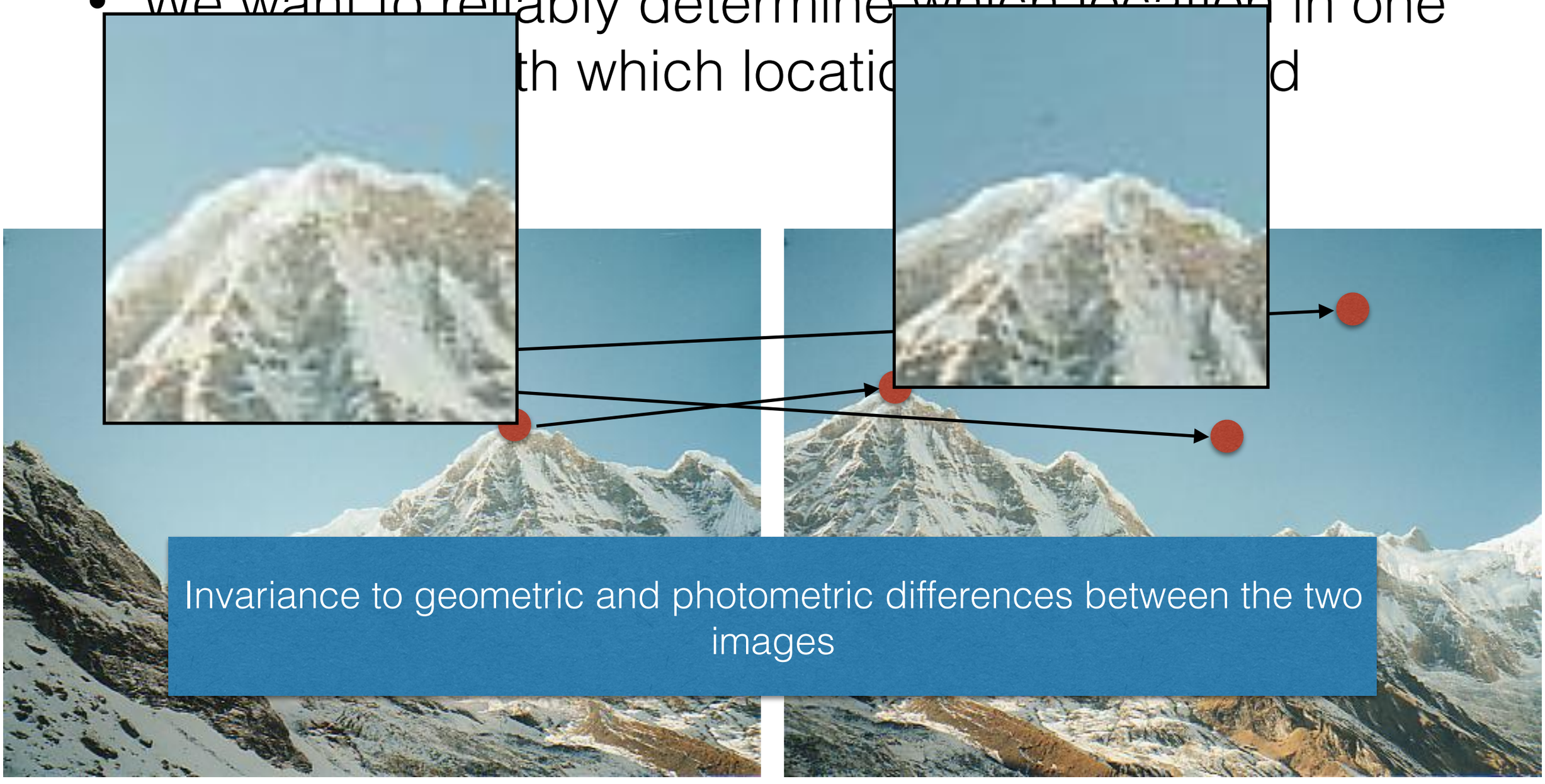
- Compactness — efficiency, many fewer features than the number of pixels in the image

- Locality — robustness to clutter and occlusion, a feature should only occupy a small area of an image



Distinctiveness

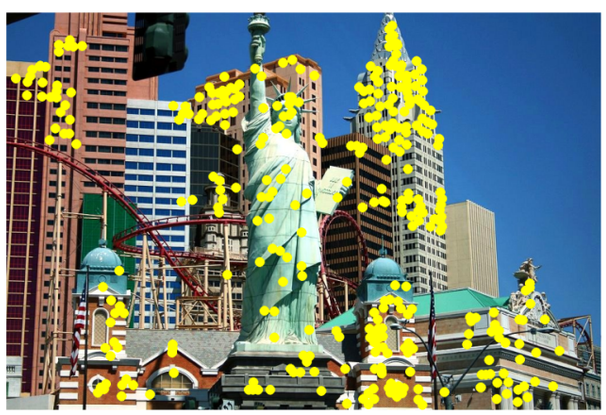
- We want to reliably determine which location in one [illegible] with which locatio[illegible]d



Invariance to geometric and photometric differences between the two images

Feature Points' Uses

- Image alignment
- 3D reconstruction
- Tracking
- Object recognition
- Image retrieval
- Navigation and mapping



Local Features

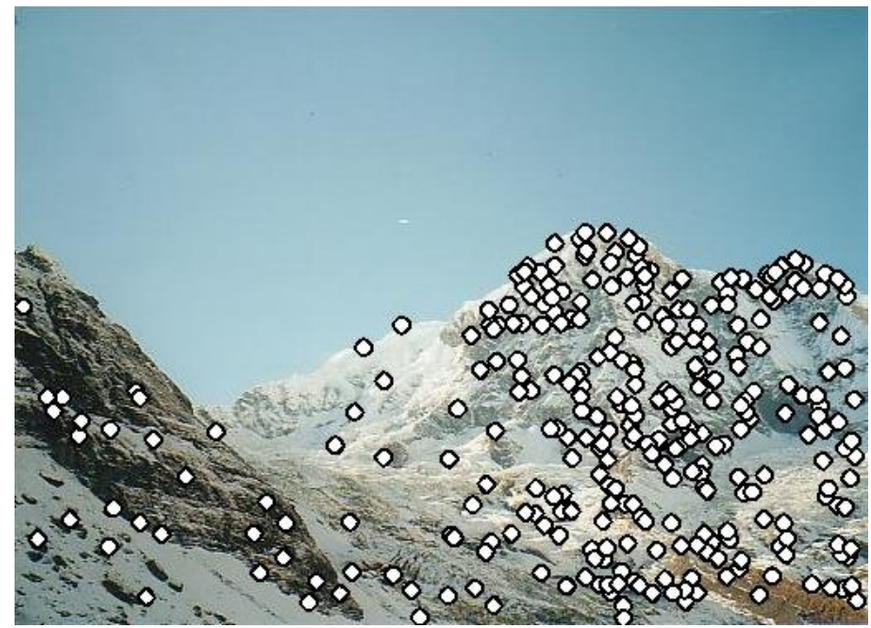
- Part1: detect interest points
- Part 2: compute descriptors that encode the area surrounding an interest point
- Part 3: use descriptors for finding matches (identifying corresponding locations, for example)

Local Features

- Part1: detect interest points

- Part 2: compute descriptors that encode the area surrounding an interest point

- Part 3: use descriptors for finding matches (identifying corresponding locations, for example)



Many Existing Detectors Available

Hessian & Harris	[Beaudet ‘78], [Harris ‘88]
Laplacian, DoG	[Lindeberg ‘98], [Lowe 1999]
Harris-/Hessian-Laplace	[Mikolajczyk & Schmid ‘01]
Harris-/Hessian-Affine	[Mikolajczyk & Schmid ‘04]
EBR and IBR	[Tuytelaars & Van Gool ‘04]
MSER	[Matas ‘02]
Salient Regions	[Kadir & Brady ‘01]
Others…	

Interest Point Detection

Available choices

- Hessian & Harris [Beaudet ‘78], [Harris ‘88]
- Laplacian, DoG [Lindeberg ‘98], [Lowe 1999]
- Harris-/Hessian-Laplace [Mikolajczyk & Schmid ‘01]
- Harris-/Hessian-Affine [Mikolajczyk & Schmid ‘04]
- EBR and IBR [Tuytelaars & Van Gool ‘04]
- MSER [Matas ‘02]
- Salient Regions [Kadir & Brady ‘01]
- and many others

From K. Grauman, B. Leibe

How to find an “interest point?”

What is better, A or B?

A

Corners!

B

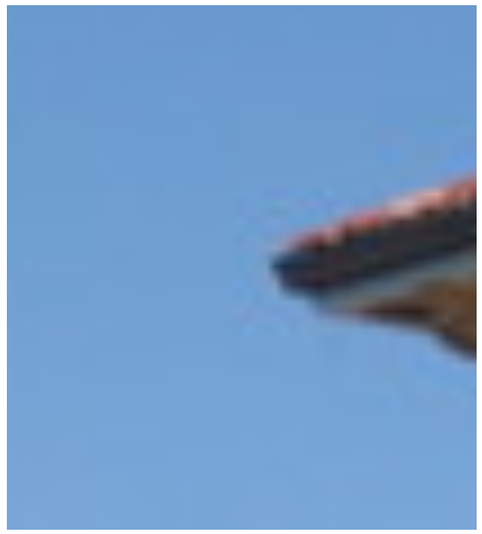
From Kristen Grauman

Corner Detection

- We should easily recognize the point looking through a small window

A

Flat area, difficult to localize

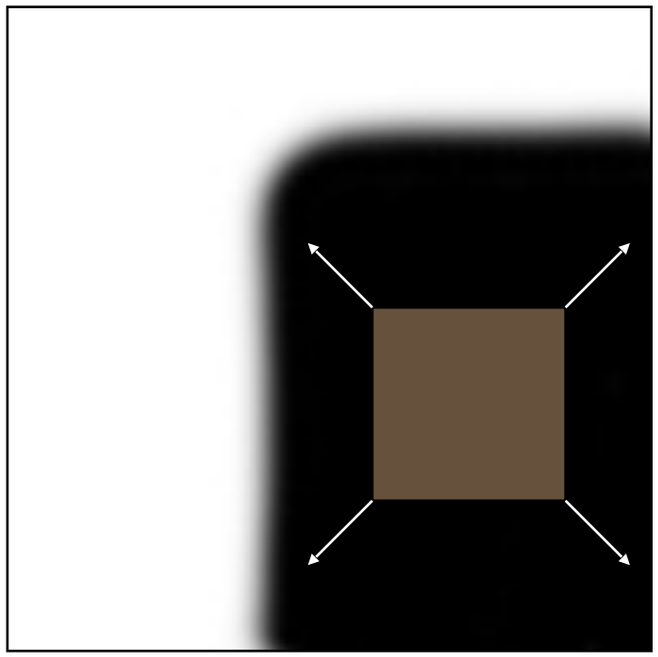


B

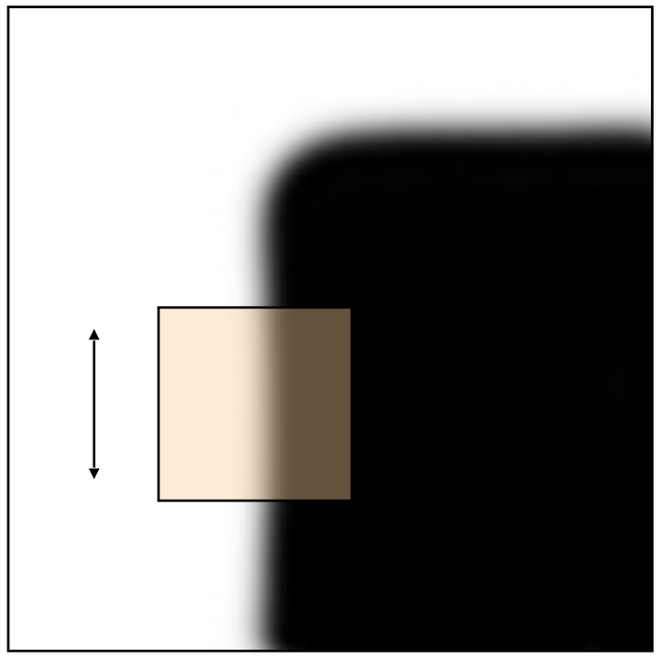
Corner, easy localization

Corner Detection

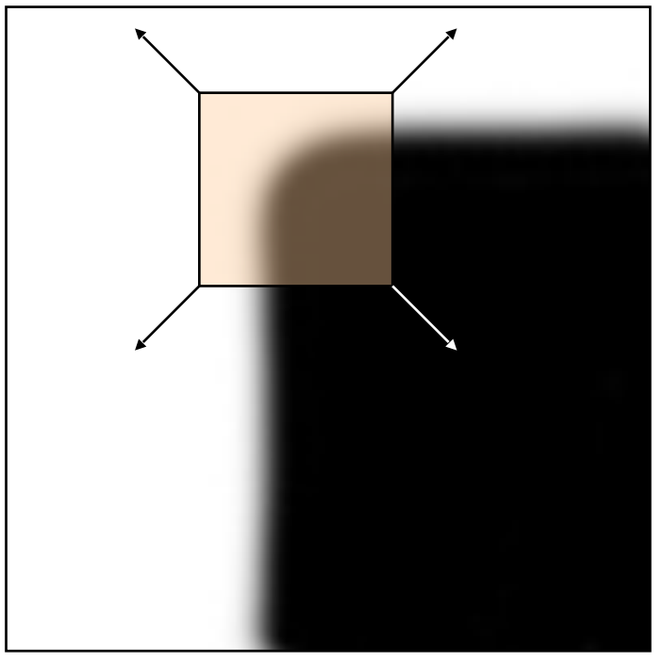
Corner Detection



“flat” region: no change in all directions



“edge”: no change along the edge direction



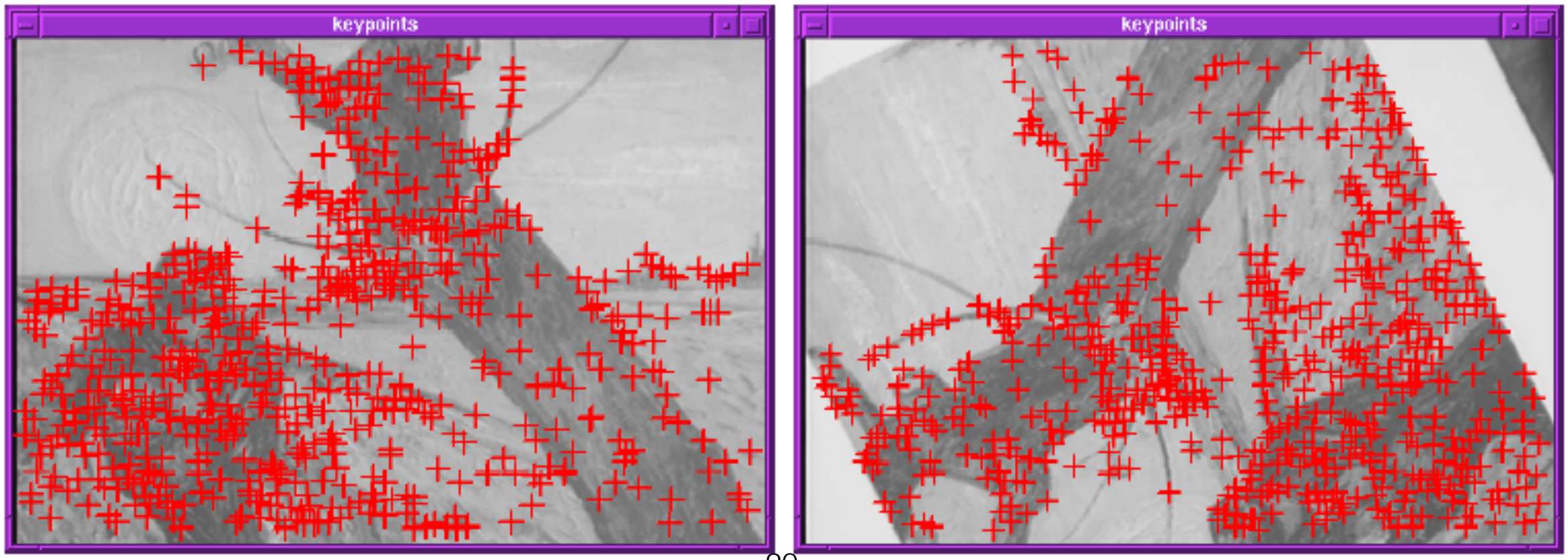
“corner”: significant change in all directions

Shifting a window in either direction should give a large change in intensity

Source: A. Efros

Harris Corner Detection

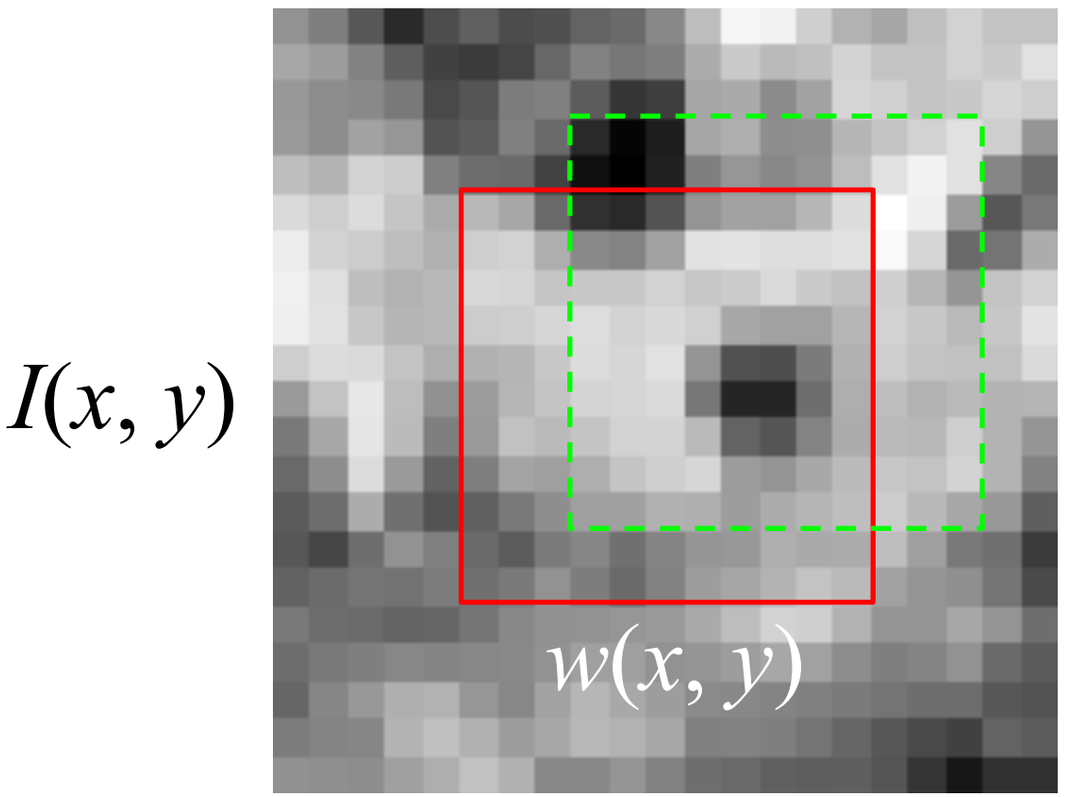
- Corners are repeatable and distinctive
- Image gradients change in both directions around corners

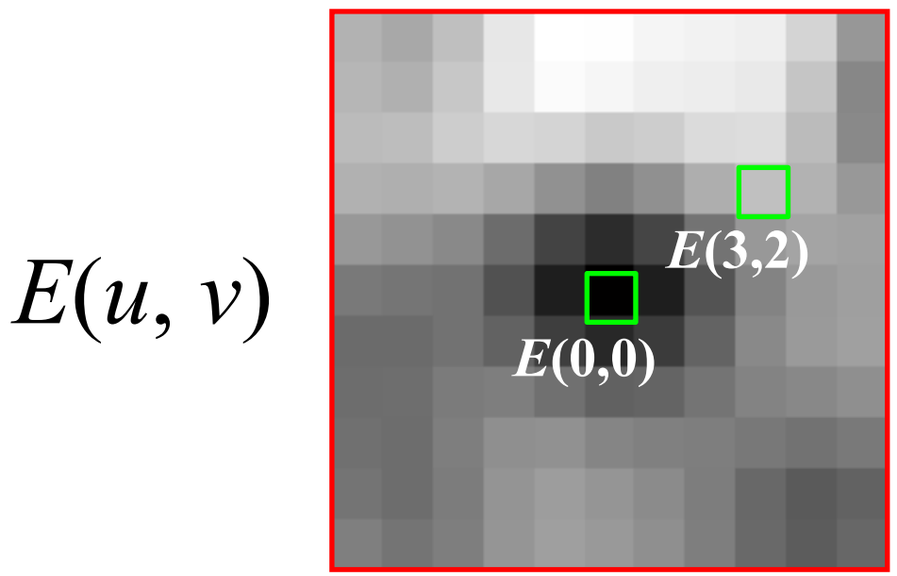


Corner Detection: Mathematics

Change in appearance of window w(x,y) for the shift [u,v]

$$E(u,v) = \sum_{x,y} w(x,y)\,[I(x+u,y+v) - I(x,y)]^2$$





Corner Detection: Mathematics

Change in appearance of window w(x,y) for the shift [u,v]

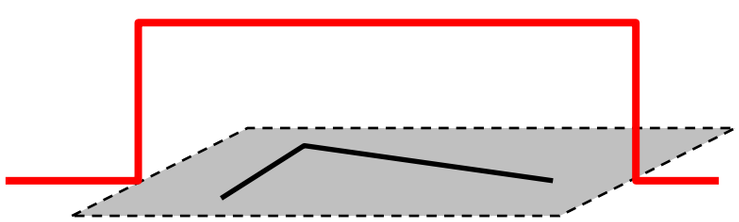
$$E(u,v) = \sum_{x,y} w(x,y)\,[I(x+u,y+v) - I(x,y)]^2$$

Window function

Shifted intensity

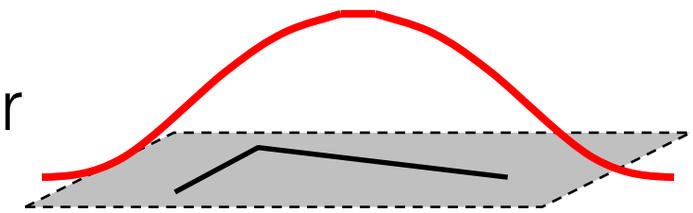
Intensity

Window function $w(x,y) =$



1 in window, 0 outside

or



Gaussian

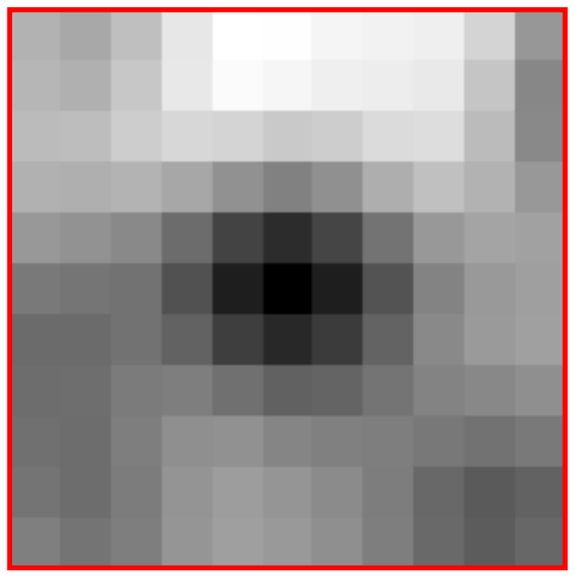
Source: R. Szeliski

Corner Detection: Mathematics

- How does E(u,v) behaves for small shifts [u,v]?

$$E(u,v) = \sum_{x,y} w(x,y)\,[I(x+u,y+v) - I(x,y)]^2$$

$E(u, v)$



Corner Detection: Mathematics

- How does E(u,v) behaves for small shifts [u,v]?

$$E(u,v) = \sum_{x,y} w(x,y)\left[I(x+u,y+v) - I(x,y)\right]^2$$

- Use Taylor Series for local quadratic approximation of E(u,v)

$$E(u,v) = E(0,0) + \begin{bmatrix} u & v \end{bmatrix} \begin{bmatrix} E_u(0,0) \\ E_v(0,0) \end{bmatrix} + \frac{1}{2} \begin{bmatrix} u & v \end{bmatrix} \begin{bmatrix} E_{uu}(0,0) & E_{uv}(0,0) \\ E_{vu}(0,0) & E_{vv}(0,0) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}$$

Corner Detection: Mathematics

$$E(u,v) = E(0,0) + \begin{bmatrix} u & v \end{bmatrix} \begin{bmatrix} E_u(0,0) \\ E_v(0,0) \end{bmatrix} + \frac{1}{2} \begin{bmatrix} u & v \end{bmatrix} \begin{bmatrix} E_{uu}(0,0) & E_{uv}(0,0) \\ E_{vu}(0,0) & E_{vv}(0,0) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}$$

$$E_u(u,v) = \sum_{x,y} 2w(x,y)\left[I(x+u,y+v) - I(x,y)\right] I_x(x+u,y+v)$$

$$E_{uu}(u,v) = \sum_{x,y} 2w(x,y) I_x(x+u,y+v) I_x(x+u,y+v)$$

$$+ \sum_{x,y} 2w(x,y)\left[I(x+u,y+v) - I(x,y)\right] I_{xx}(x+u,y+v)$$

$$E_{uv}(u,v) = \sum_{x,y} 2w(x,y) I_y(x+u,y+v) I_x(x+u,y+v)$$

$$+ \sum_{x,y} 2w(x,y)\left[I(x+u,y+v) - I(x,y)\right] I_{xy}(x+u,y+v)$$

Corner Detection: Mathematics

$$E(u,v) = E(0,0) + \begin{bmatrix} u & v \end{bmatrix} \begin{bmatrix} E_u(0,0) \\ E_v(0,0) \end{bmatrix} + \frac{1}{2} \begin{bmatrix} u & v \end{bmatrix} \begin{bmatrix} E_{uu}(0,0) & E_{uv}(0,0) \\ E_{vu}(0,0) & E_{vv}(0,0) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}$$

$$E(0,0) = 0$$

$$E_u(0,0) = 0$$

$$E_v(0,0) = 0$$

$$E_{uu}(0,0) = \sum_{x,y} 2w(x,y) I_x(x,y) I_x(x,y)$$

$$E_{vv}(0,0) = \sum_{x,y} 2w(x,y) I_y(x,y) I_y(x,y)$$

$$E_{uv}(0,0) = \sum_{x,y} 2w(x,y) I_x(x,y) I_y(x,y)$$

Corner Detection: Mathematics

- The quadratic approximation simplifies to

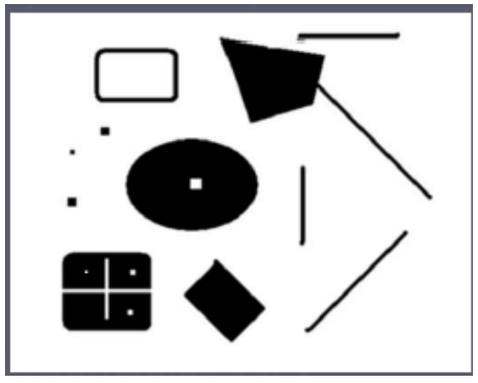
$$E(u,v) \approx \begin{bmatrix} u & v \end{bmatrix} M \begin{bmatrix} u \\ v \end{bmatrix}$$

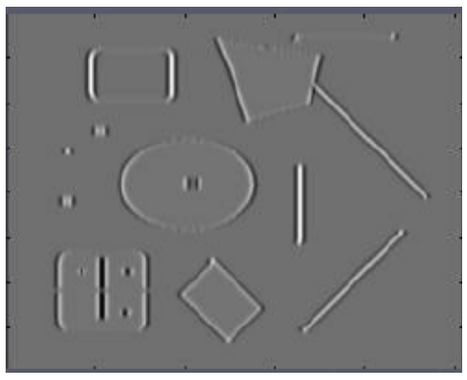
- M is the second moment matrix

$$M = \sum_{x,y} w(x,y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

Corners as Distinctive Interest Points

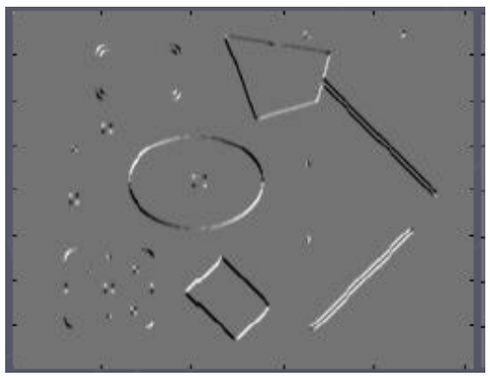
Image derivatives averaged in neighbourhood of a point





$$I_x \Leftrightarrow \frac{\partial I}{\partial x}$$

$$I_y \Leftrightarrow \frac{\partial I}{\partial y}$$



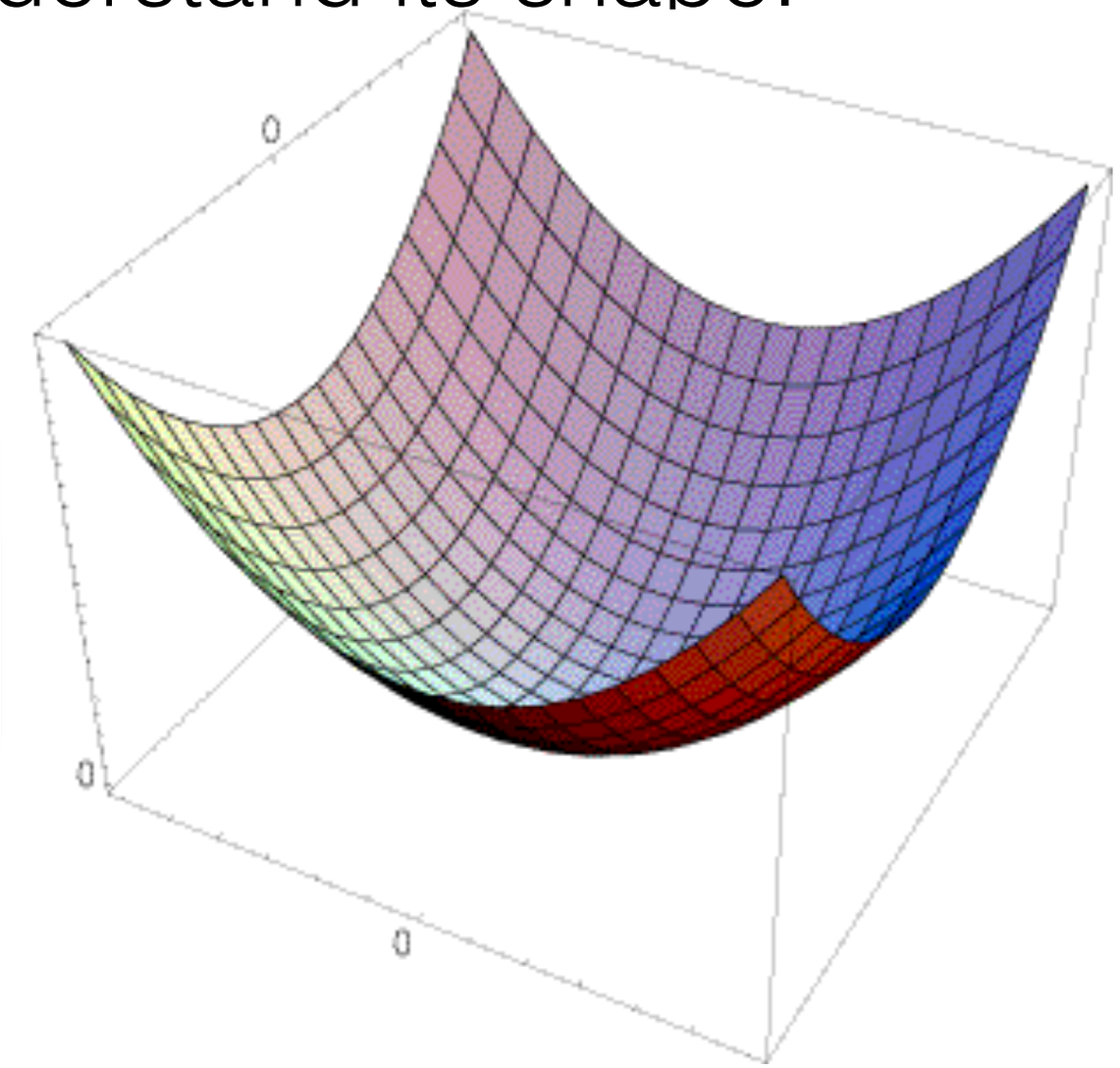
$$I_x I_y \Leftrightarrow \frac{\partial I}{\partial x}\frac{\partial I}{\partial y}$$

Interpreting the second moment matrix

The surface E(u,v) is locally approximated by a quadratic form. Let's try to understand its shape.

$$E(u,v) \approx \begin{bmatrix} u & v \end{bmatrix} M \begin{bmatrix} u \\ v \end{bmatrix}$$

$$M = \sum_{x,y} w(x,y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$



Interpreting the second moment matrix

First, consider the axis-aligned case
(gradients are either horizontal or vertical)

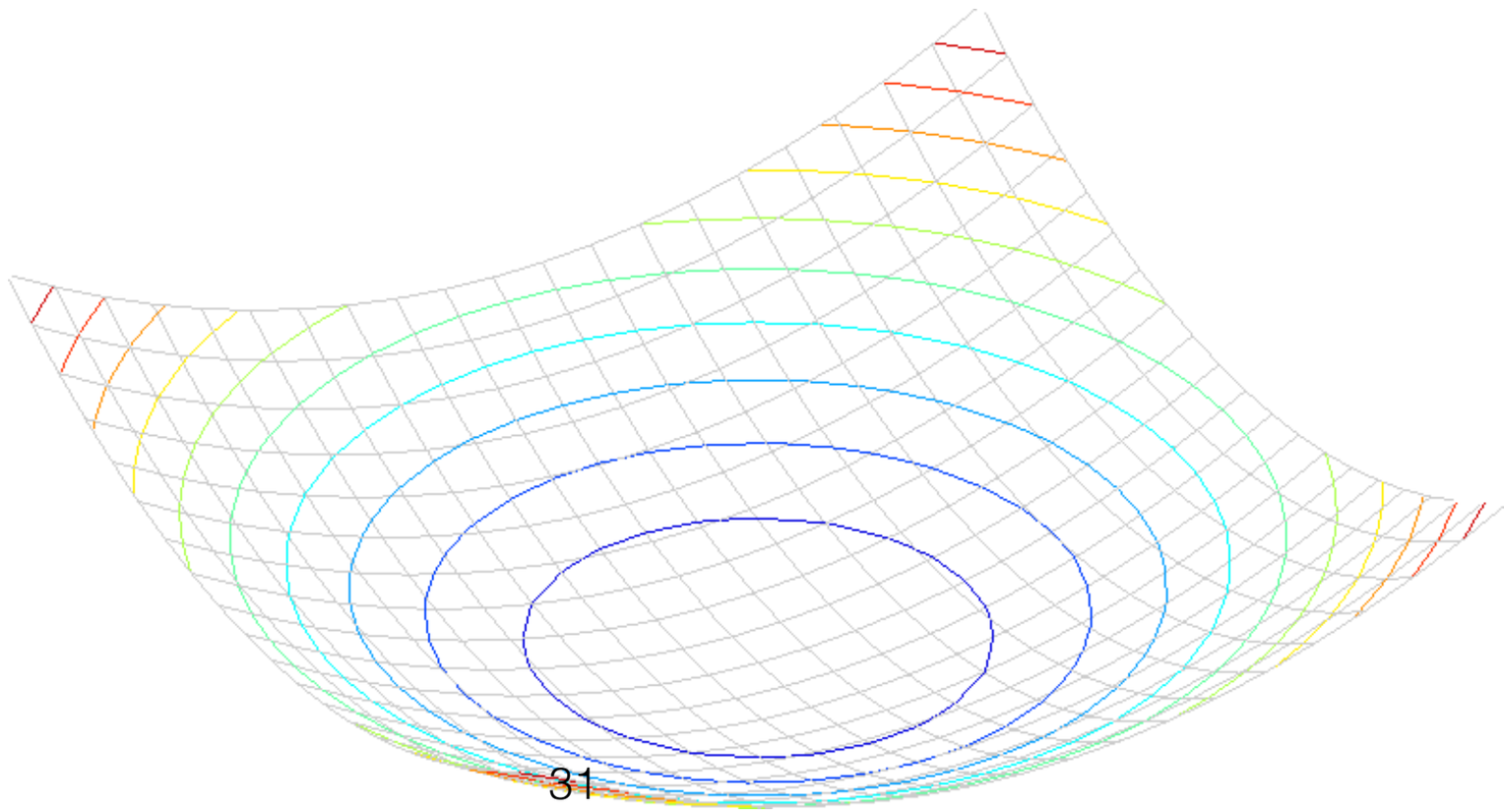
$$M = \sum_{x,y} w(x,y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$$

If either λ is close to 0, then this is not a corner, so look for locations where both are large.

Interpreting the second moment matrix

This is the equation of an ellipse: $[u \;\; v]\; M \begin{bmatrix} u \\ v \end{bmatrix} = \text{const}$

Consider a horizontal “slice” of E(u, v):

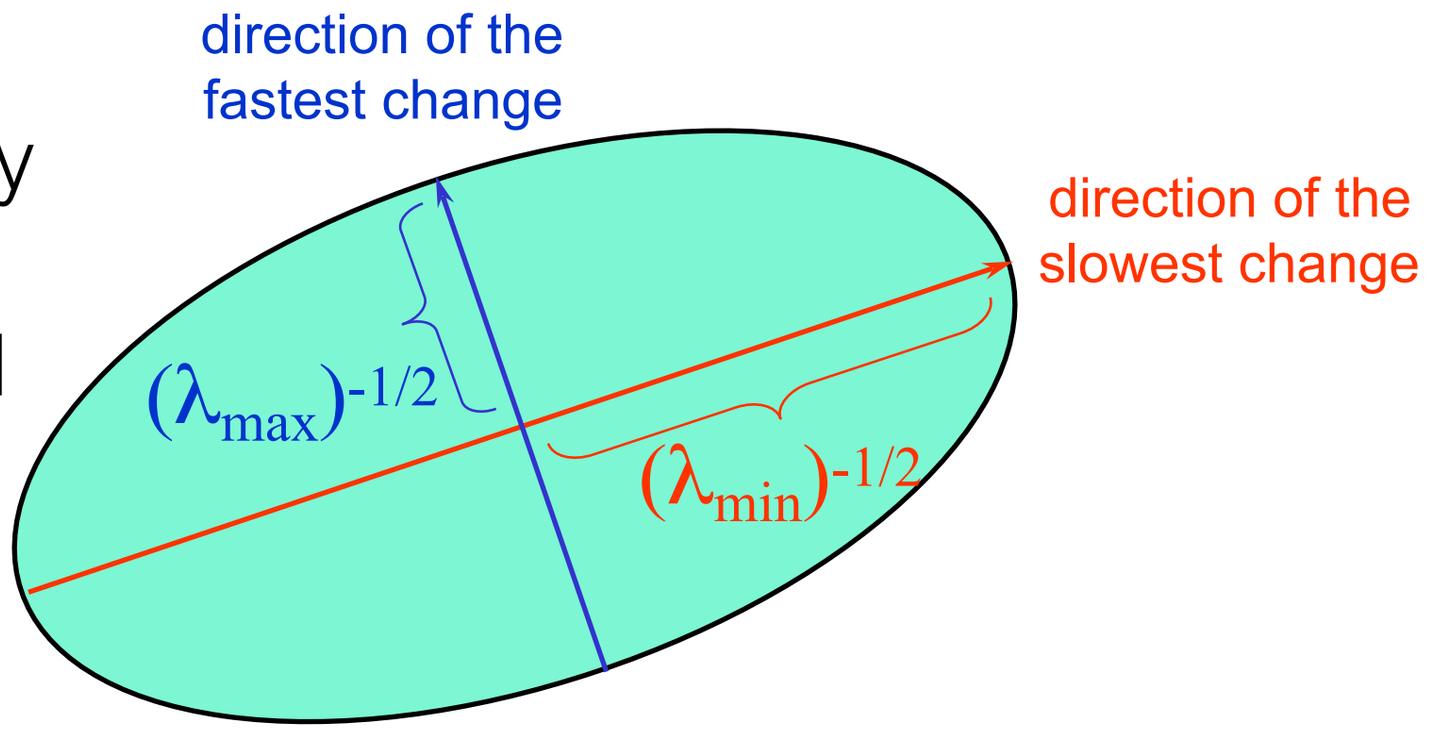


Interpreting the second moment matrix

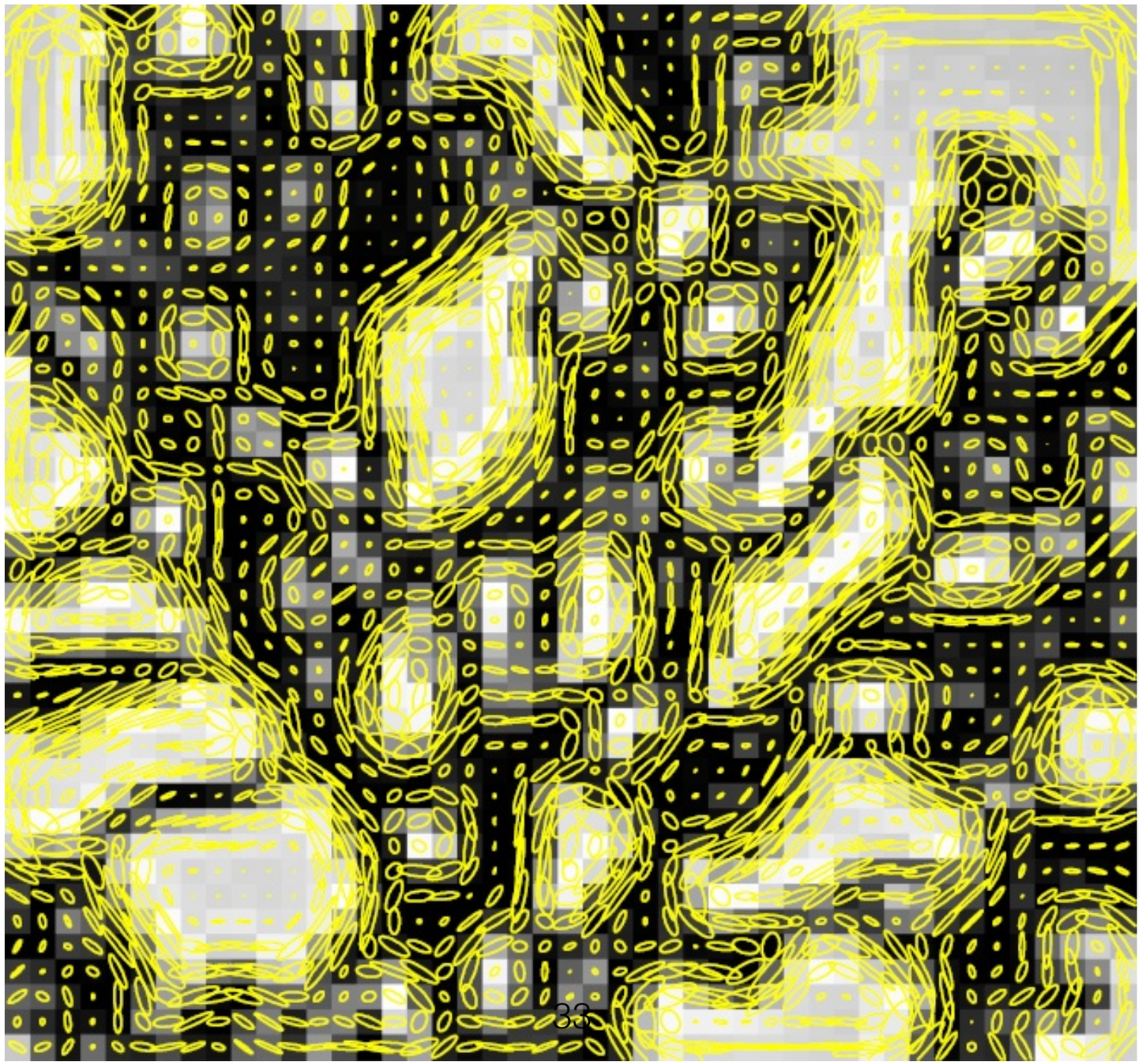
This is the equation of an ellipse. $[u \;\; v] \; M \begin{bmatrix} u \\ v \end{bmatrix} = \text{const}$

Diagonalization of M: $M = R^{-1} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} R$

The axis lengths of the ellipse are determined by the eigenvalues and the orientation is determined by R

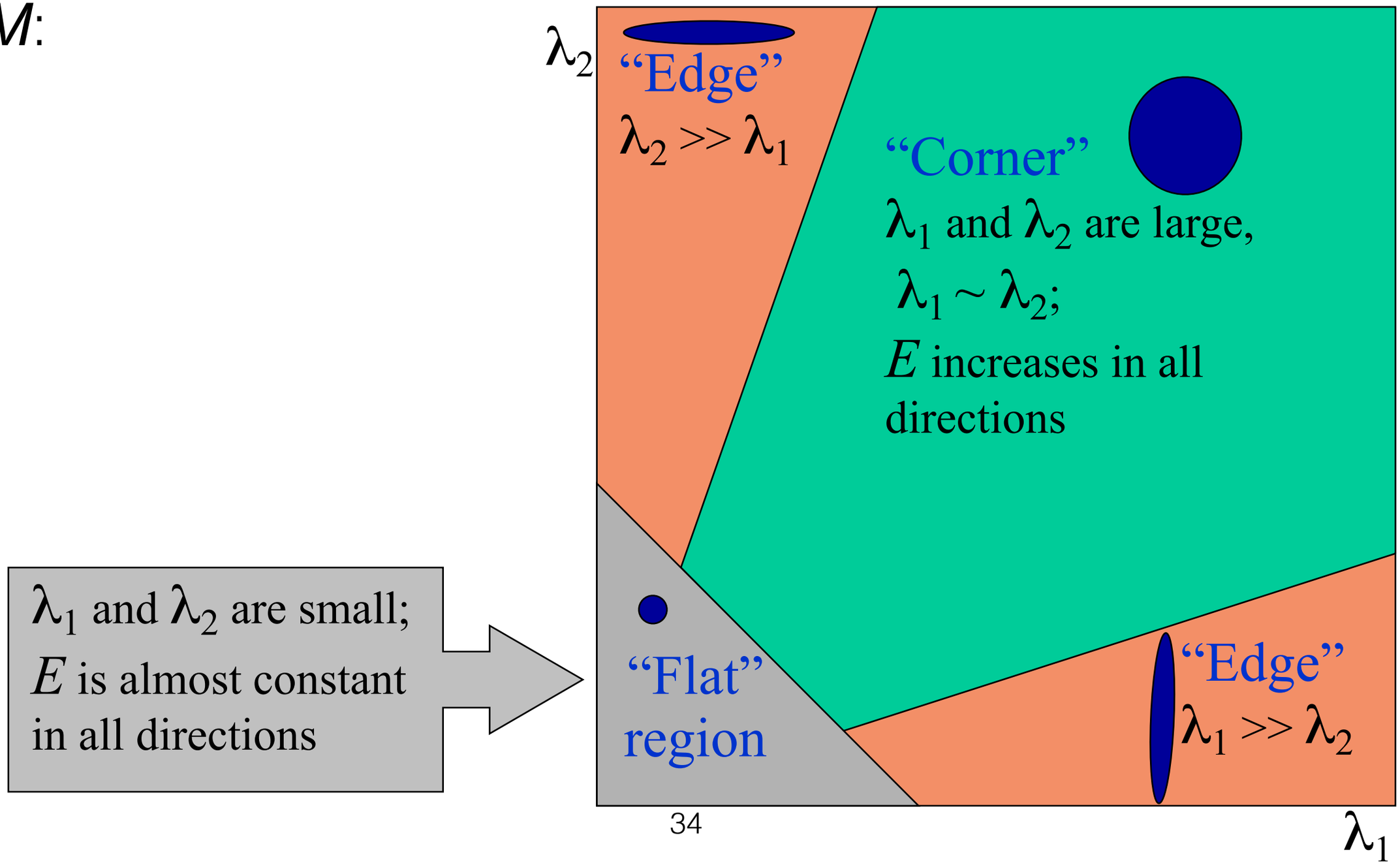


Visualization of second moment matrices



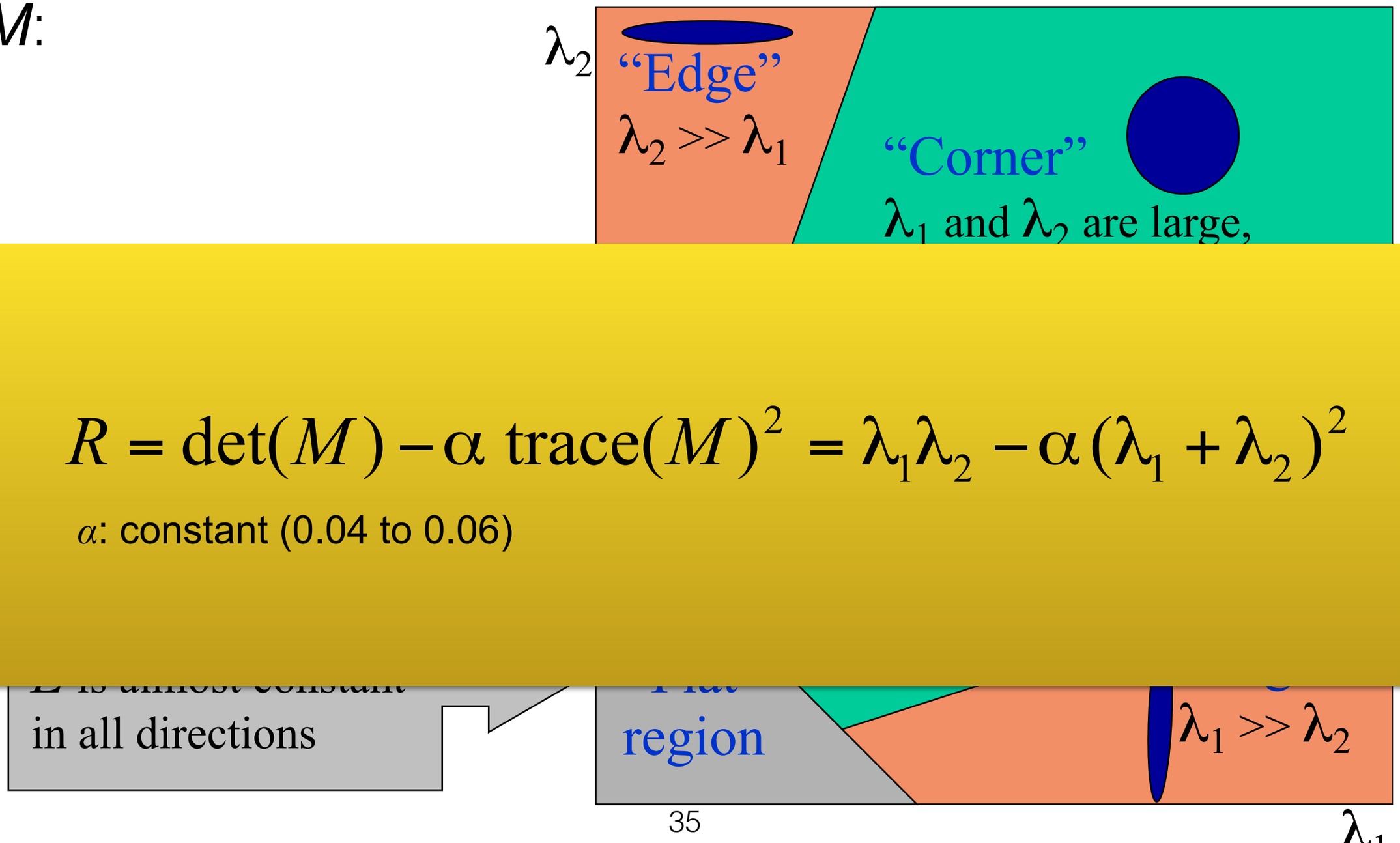
Interpreting the eigenvalues

Classification of image points using eigenvalues of *M*:



Interpreting the eigenvalues

Classification of image points using eigenvalues of *M*:



$$R = \det(M) - \alpha\ \mathrm{trace}(M)^2 = \lambda_1\lambda_2 - \alpha(\lambda_1 + \lambda_2)^2$$

α: constant (0.04 to 0.06)

Harris corner detector

1) Compute ***M*** matrix for each image window to get their ***cornerness*** scores.

2) Find points whose surrounding window gave large corner response (***f*** > threshold)

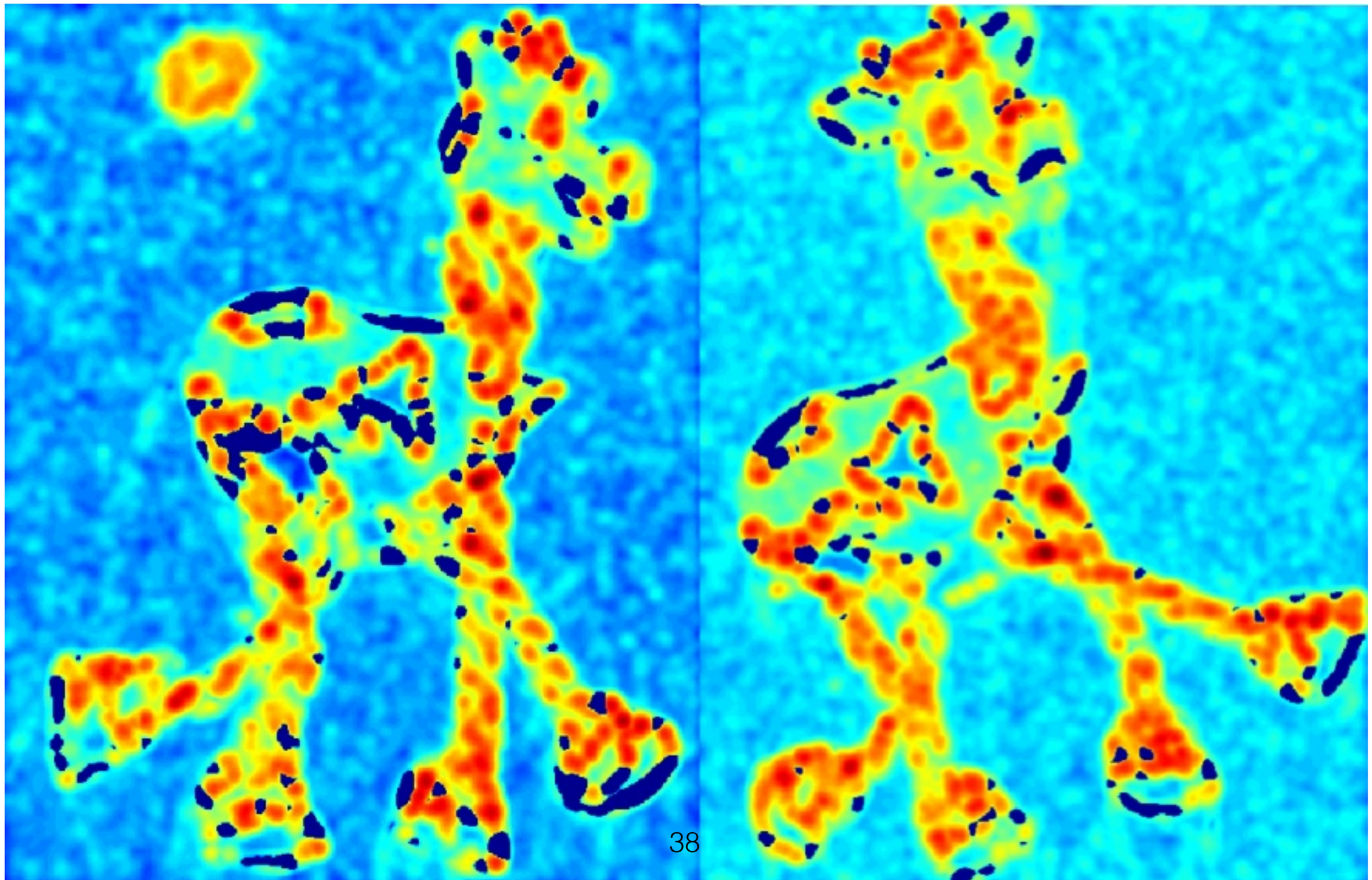
3) Take the points of local maxima, i.e., perform non-maximum suppression

C.Harris and M.Stephens. "A Combined Corner and Edge Detector." ***Proceedings of the 4th Alvey Vision Conference***: pages 147—151, 1988.

Harris Detector: Steps

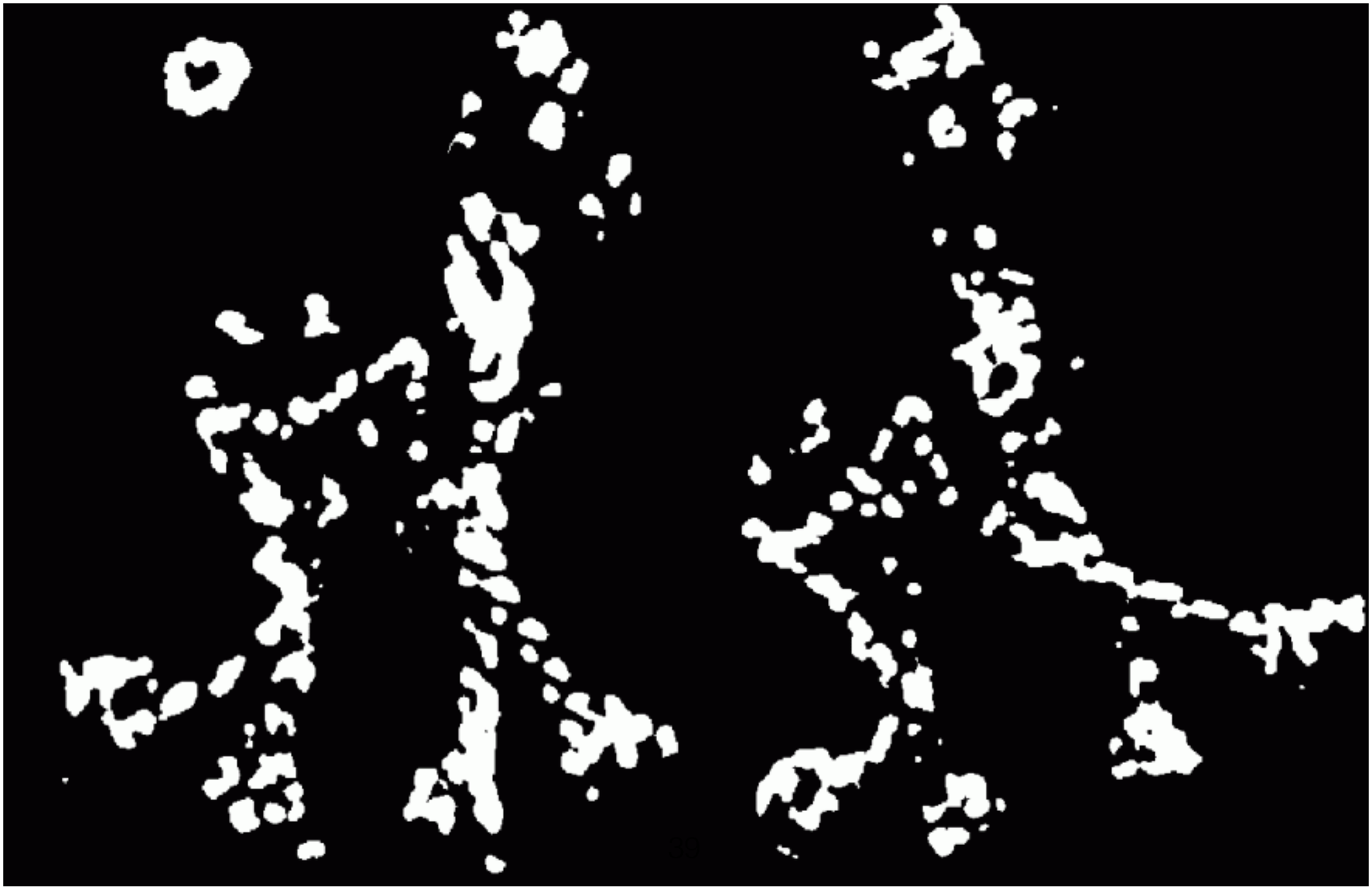
Harris Detector: Steps

Compute corner response *R*



Harris Detector: Steps

Find points with large corner response: *R*>threshold



Harris Detector: Steps

Take only the points of local maxima of *R*

Harris Detector: Steps

Invariance and covariance

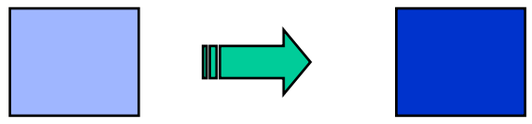
- We want corner locations to be ***invariant*** to photometric transformations and ***covariant*** to geometric transformations

- Invariance: image is transformed and corner locations do not change

- Covariance: if we have two transformed versions of the same image, features should be detected in corresponding locations

Invariance and covariance

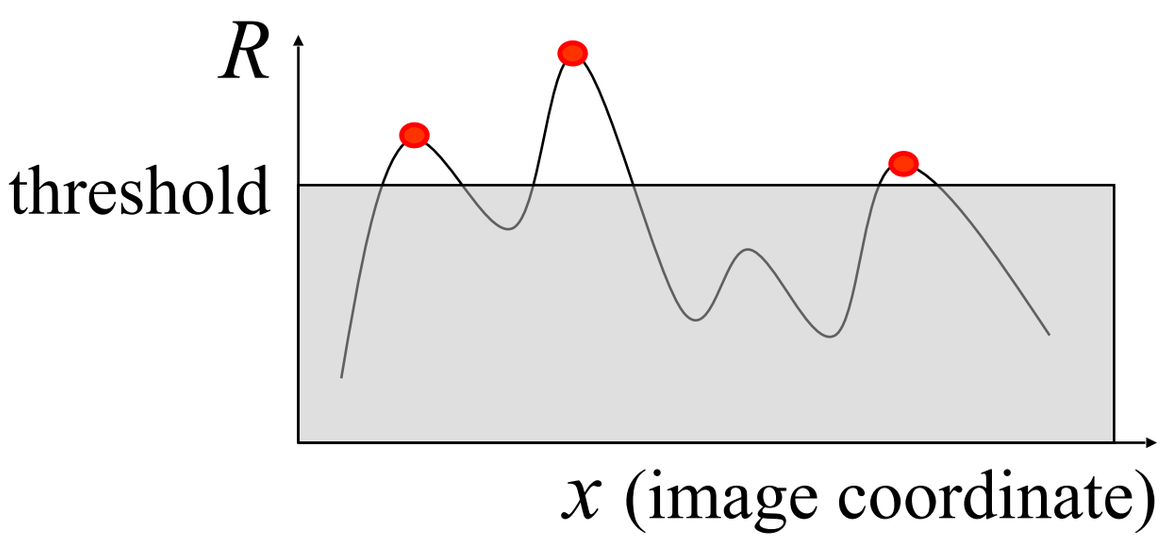
Affine intensity change

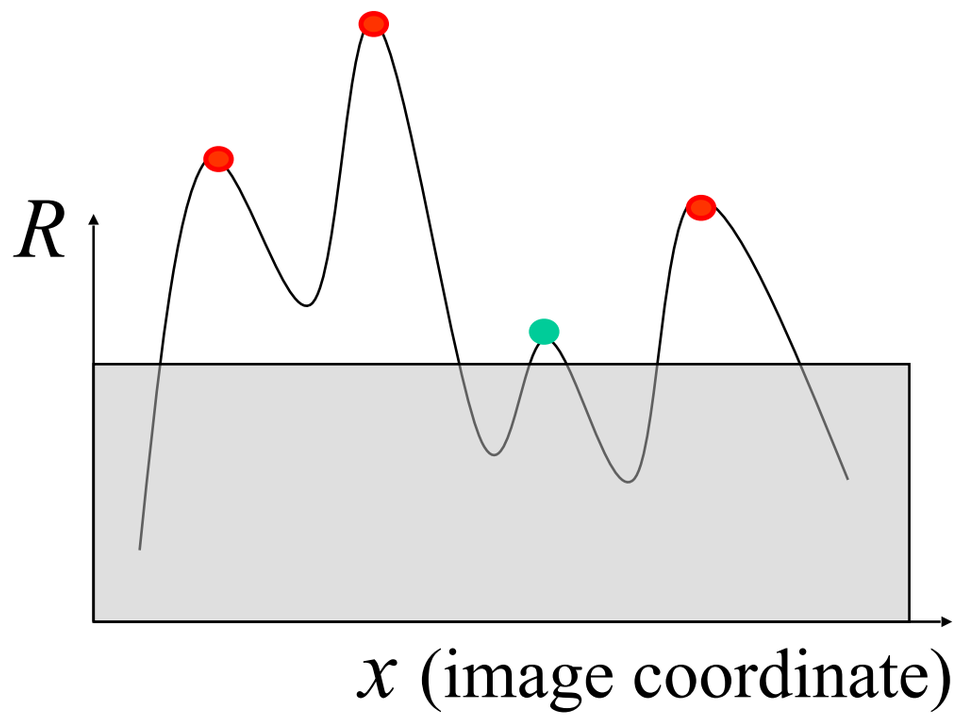


$$I \rightarrow a\,I + b$$

Only derivatives are used => invariance to intensity shift $I \rightarrow I + b$

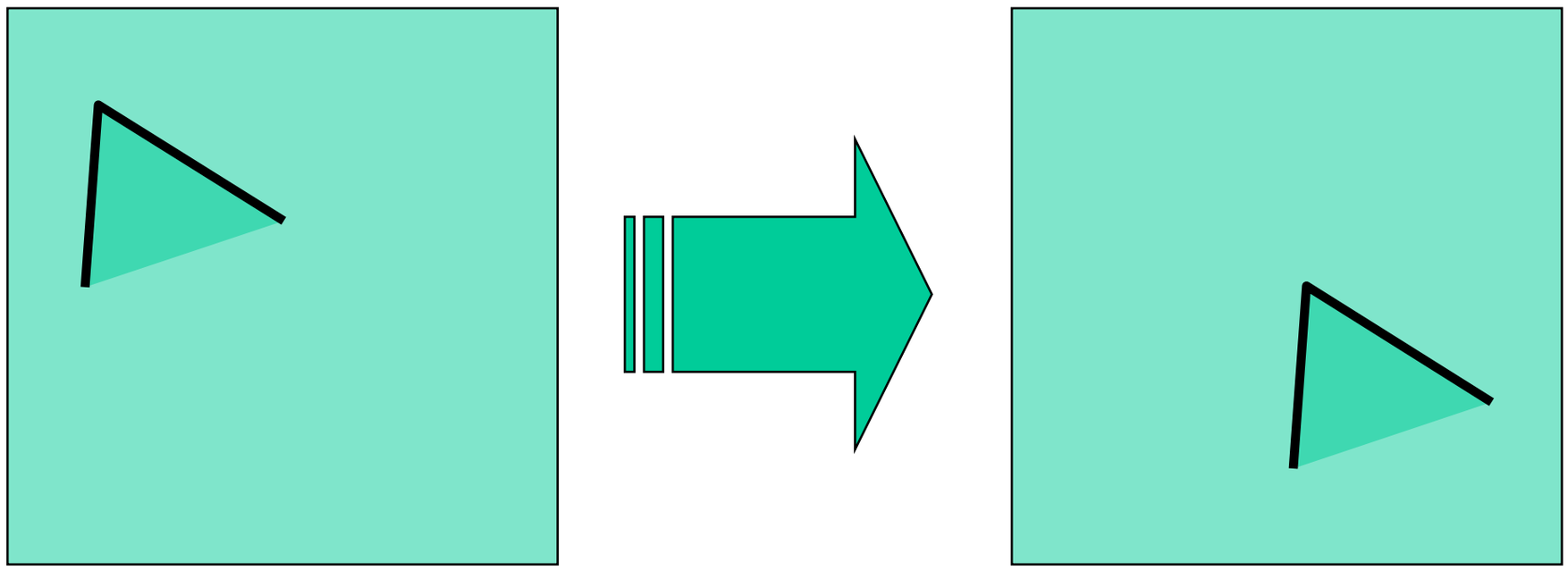
Intensity scaling: $I \rightarrow a\,I$





Partially invariant to affine intensity change

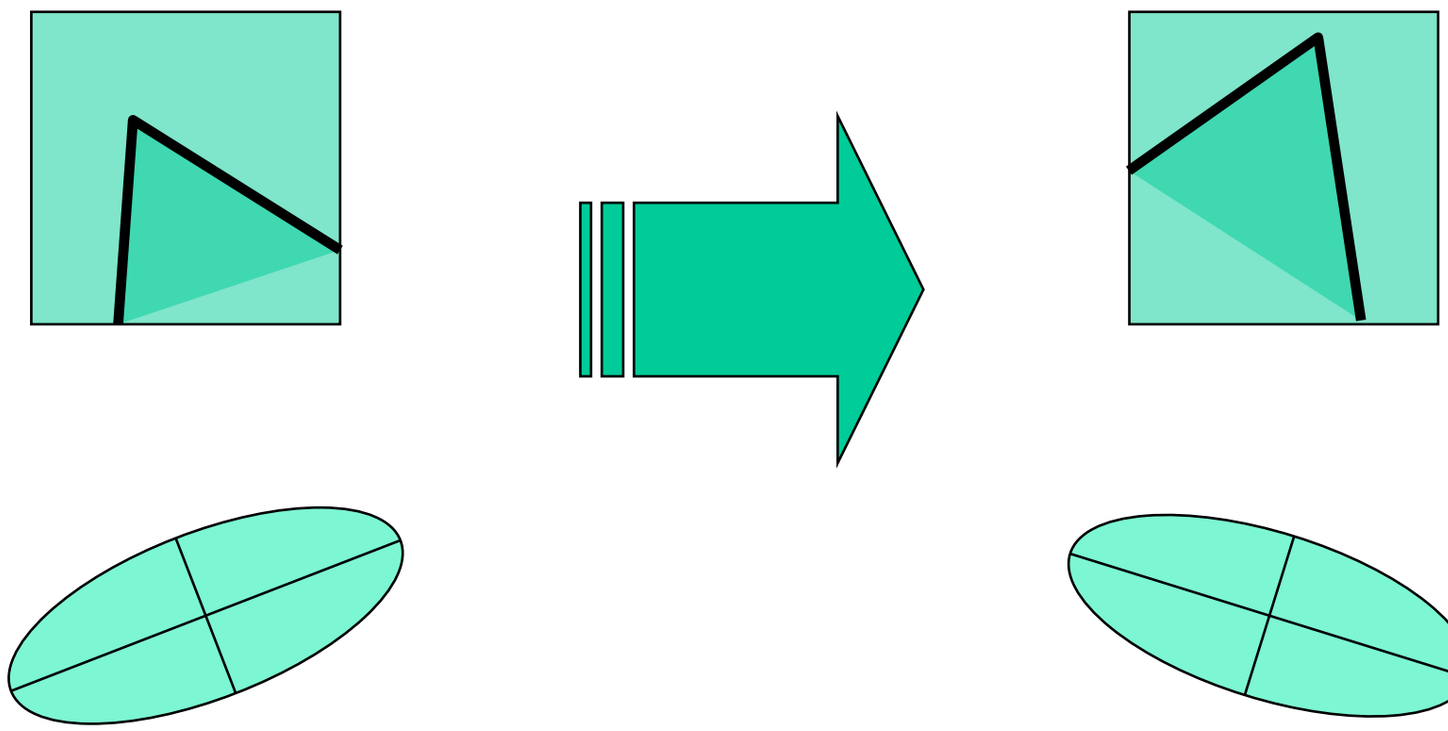
Image translation



- Derivatives and window function are shift-invariant

Corner location is covariant w.r.t. translation

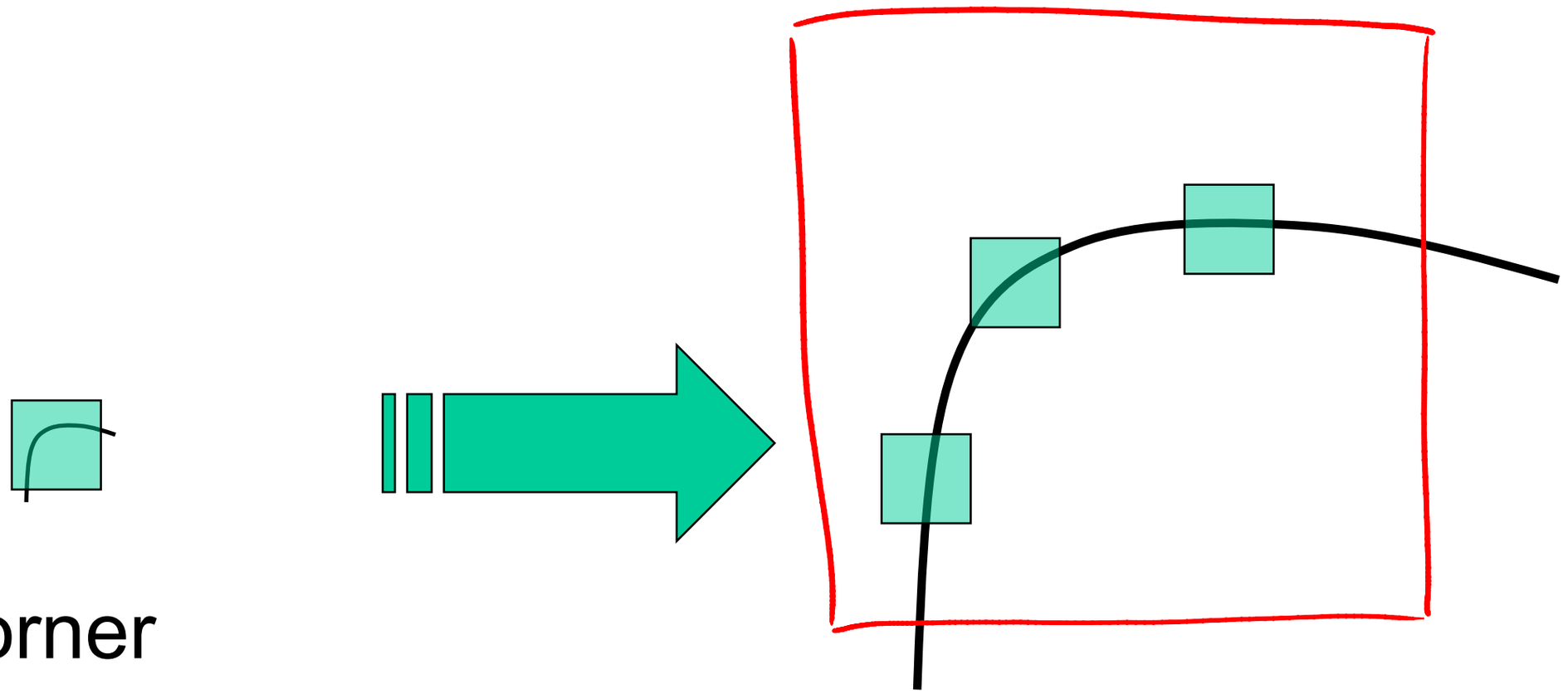
Image rotation



Second moment ellipse rotates but its shape (i.e. eigenvalues) remains the same

Corner location is covariant w.r.t. rotation

Scaling



Corner

All points will be classified as edges

Corner location is not covariant to scaling!

Feature extraction: Corners

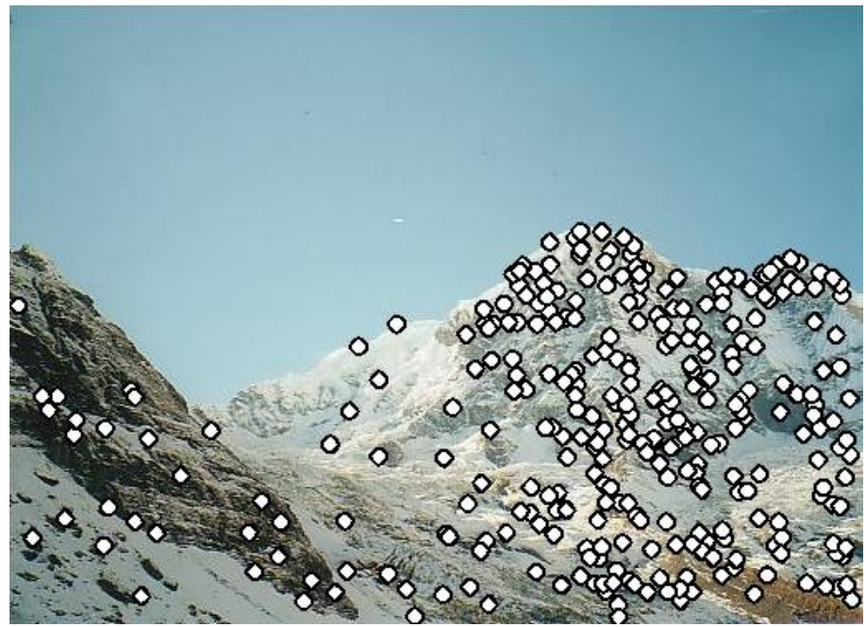
Slides from Rick Szeliski, Svetlana Lazebnik, and Kristin Grauman

Local Features

- Part1: detect interest points

- Part 2: compute descriptors that encode the area surrounding an interest point

- Part 3: use descriptors for finding matches (identifying corresponding locations, for example)



Interest Points

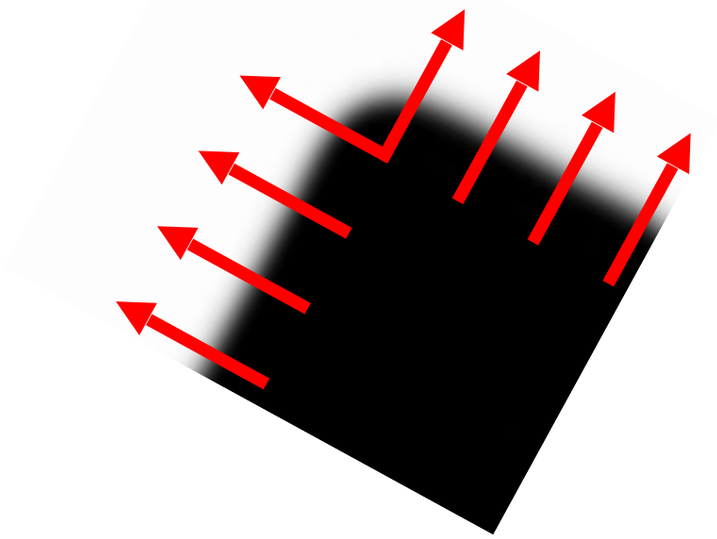
Slides from Derek Hoiem, Svetlana Lazebnik, Antonio Torralba, Steve Seitz, David Forsyth, David Lowe, Fei-Fei Li, and James Hays.

Local Invariant Features

- Detection of interest points
  - (Harris corner detection)
  - Scale invariant blob detection: LoG
- Description of local patches
  - SIFT pipeline for invariant local features

Recall: Corners as Distinctive Interest Points

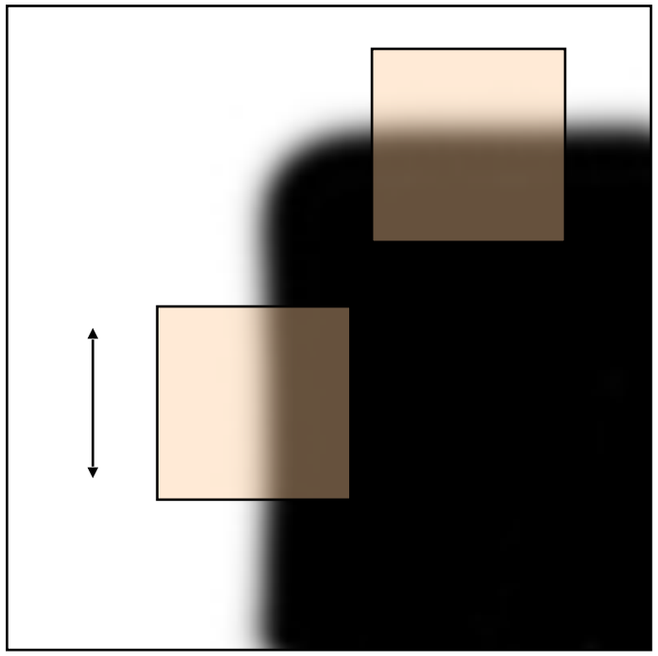
Since M is symmetric, we have $M = X \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} X^T$

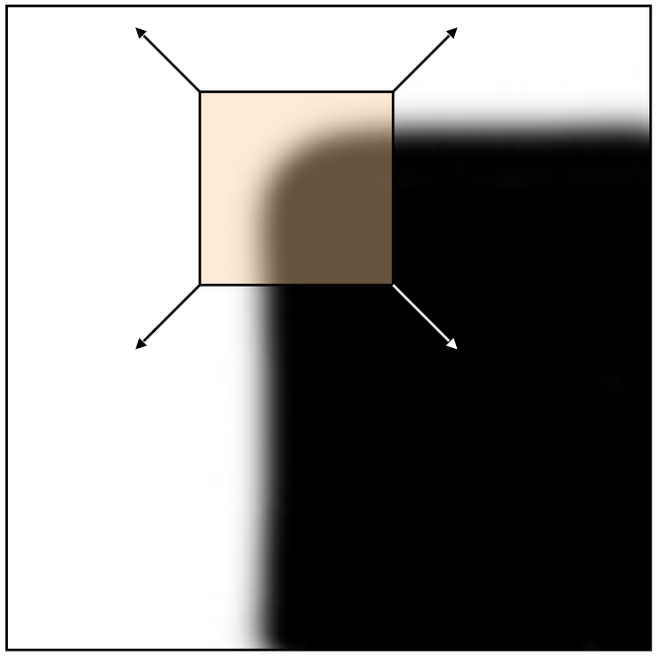


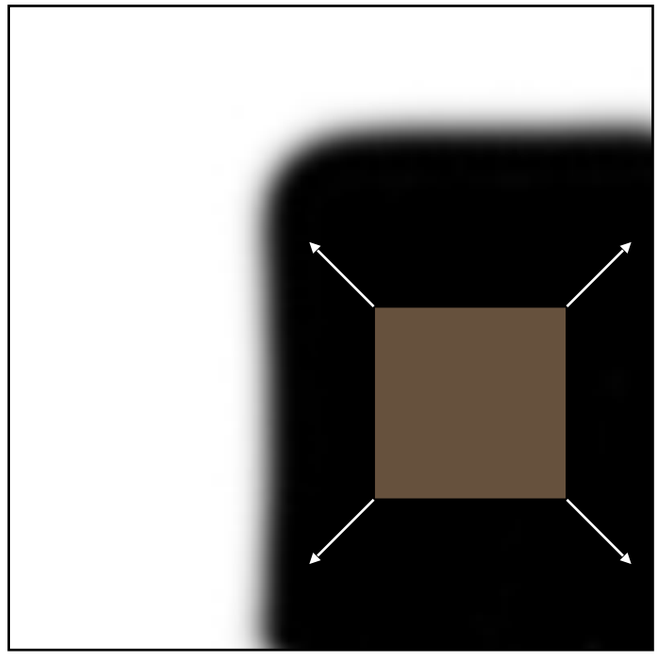
$$Mx_i = \lambda_i x_i$$

The *eigenvalues* of *M* reveal the amount of intensity change in the two principal orthogonal gradient directions in the window.

Recall: Corners as Distinctive Interest Points







"edge":

$\lambda_1 >> \lambda_2$

$\lambda_2 >> \lambda_1$

"corner":

λ_1 and λ_2 are large,

$\lambda_1 \sim \lambda_2$;

"flat" region

λ_1 and λ_2 are small;

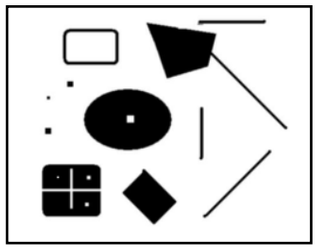
One way to score the cornerness:

$$f = \frac{\lambda_1 \lambda_2}{\lambda_1 + \lambda_2}$$

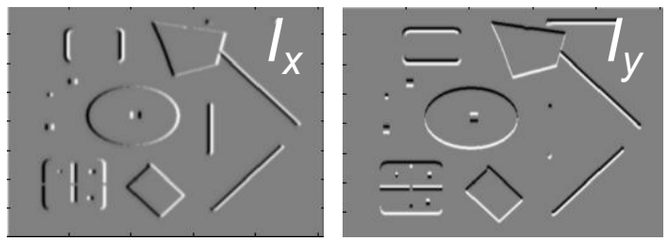
Harris Detector [Harris88]

- Second moment matrix (autocorrelation matrix)

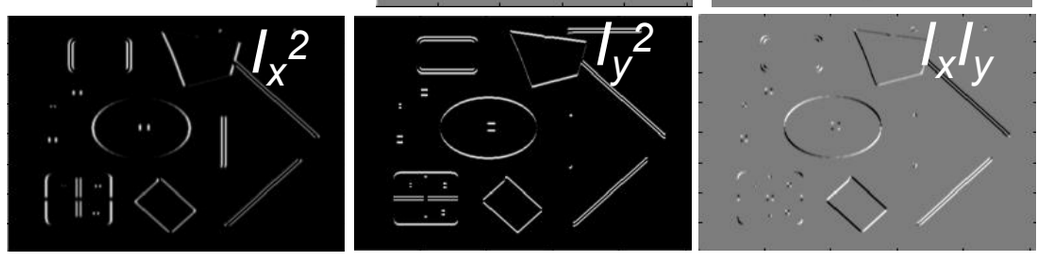
$$\mu(\sigma_I,\sigma_D) = g(\sigma_I) * \begin{bmatrix} I_x^2(\sigma_D) & I_x I_y(\sigma_D) \\ I_x I_y(\sigma_D) & I_y^2(\sigma_D) \end{bmatrix}$$



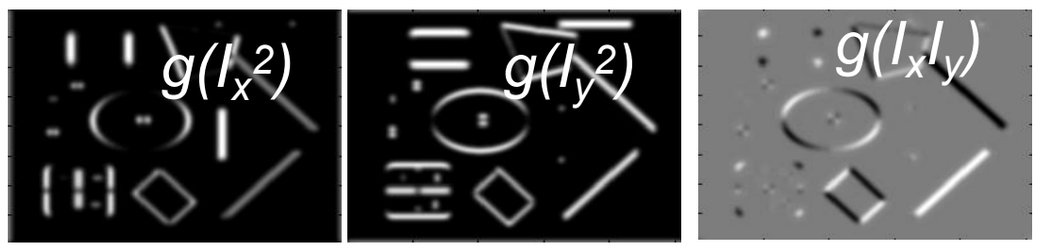
1. Image derivatives



2. Square of derivatives



3. Gaussian filter $g(\sigma_I)$



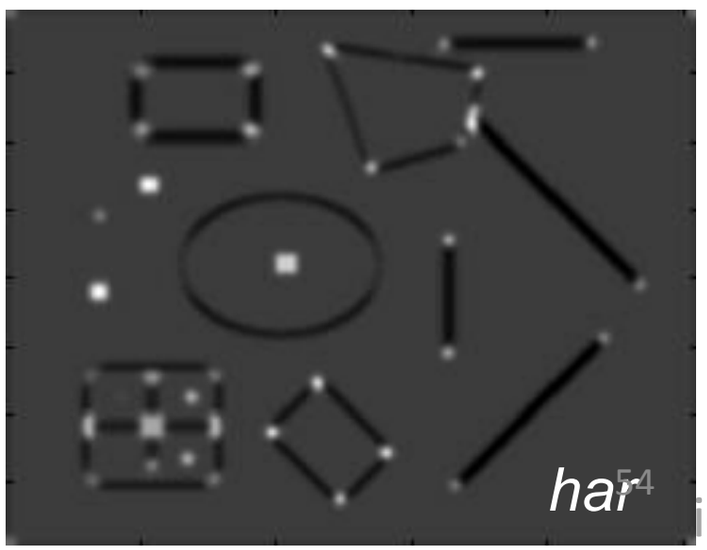
$$\det M = \lambda_1\lambda_2$$

$$\text{trace}\, M = \lambda_1 + \lambda_2$$

4. Cornerness function – both eigenvalues are strong

$$har = \det[\mu(\sigma_I,\sigma_D)] - \alpha[\text{trace}(\mu(\sigma_I,\sigma_D))^2] =$$

$$g(I_x^2)g(I_y^2) - [g(I_xI_y)]^2 - \alpha[g(I_x^2) + g(I_y^2)]^2$$



5. Non-maxima suppression

Properties of the Harris corner detector

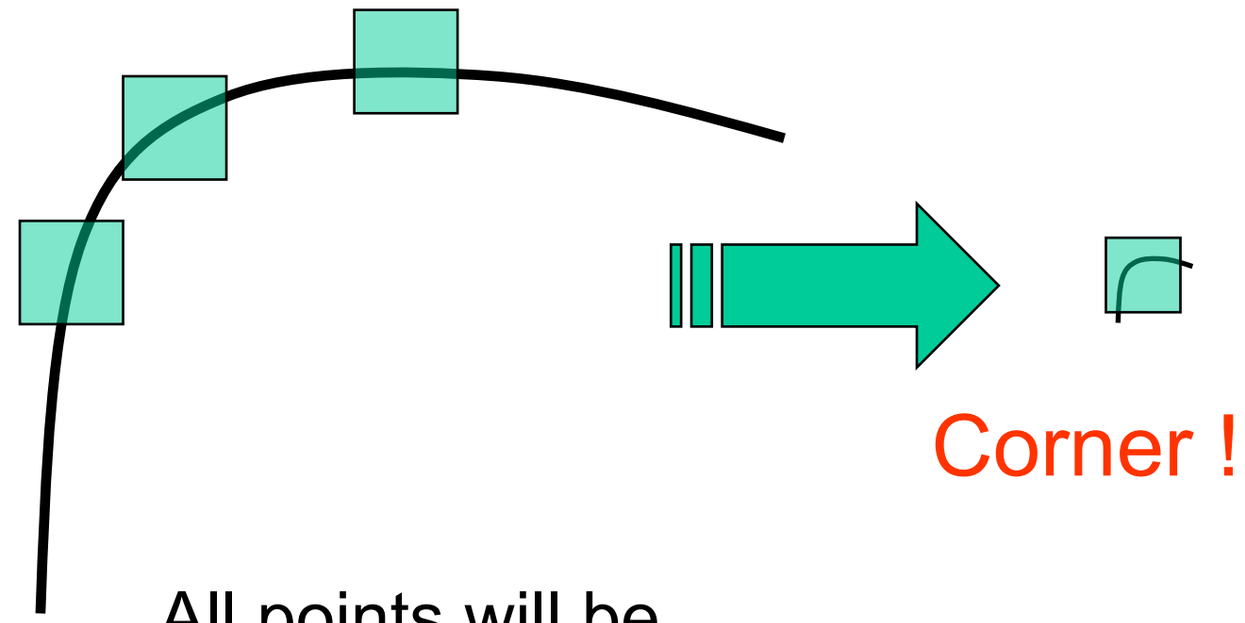
Rotation invariant? **Yes**

Scale invariant?

Properties of the Harris corner detector

Rotation invariant? **Yes**

Scale invariant? **No**



All points will be classified as edges

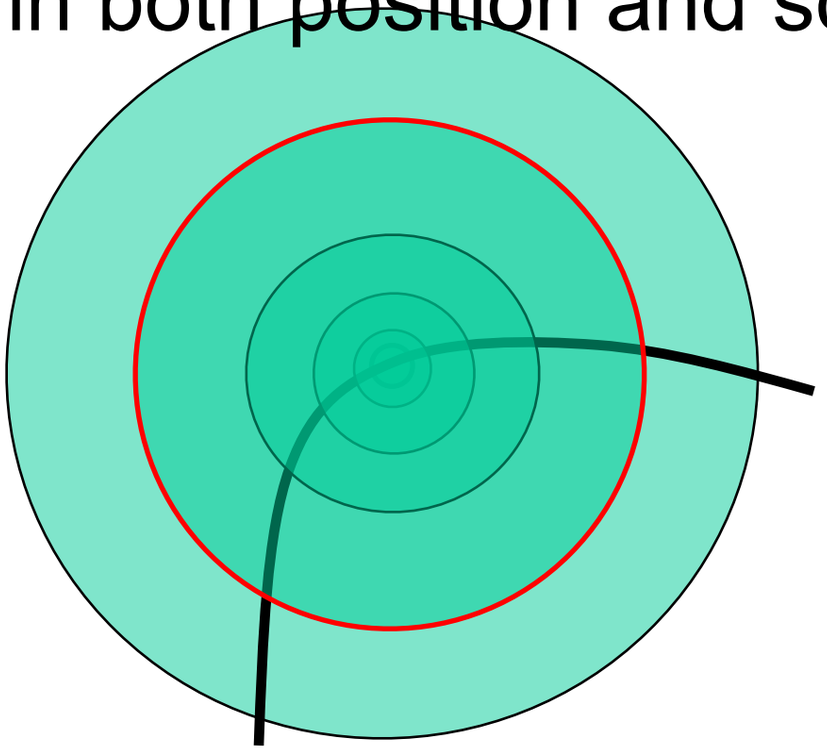
Scale invariant interest points

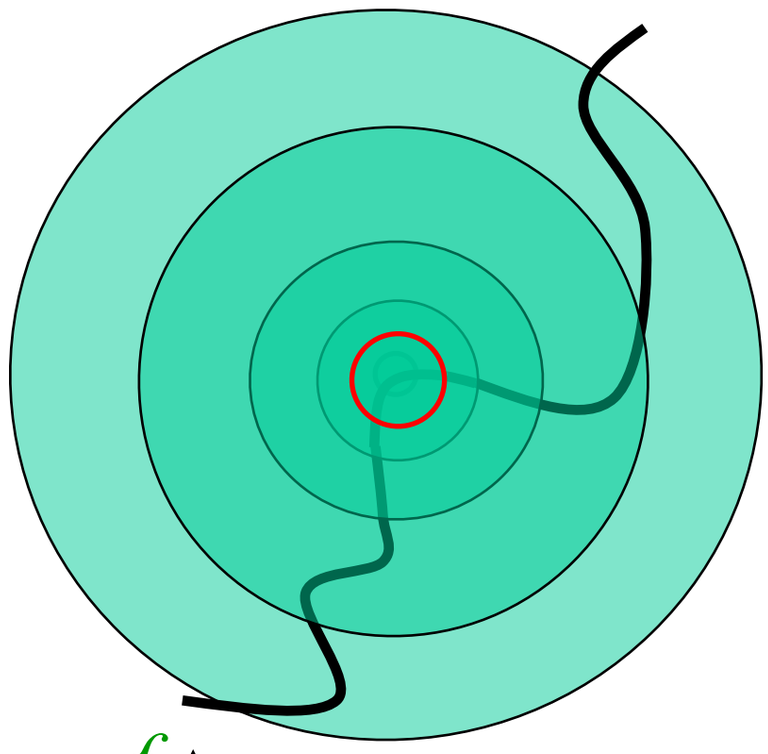
How can we independently select interest points in each image, such that the detections are repeatable across different scales?

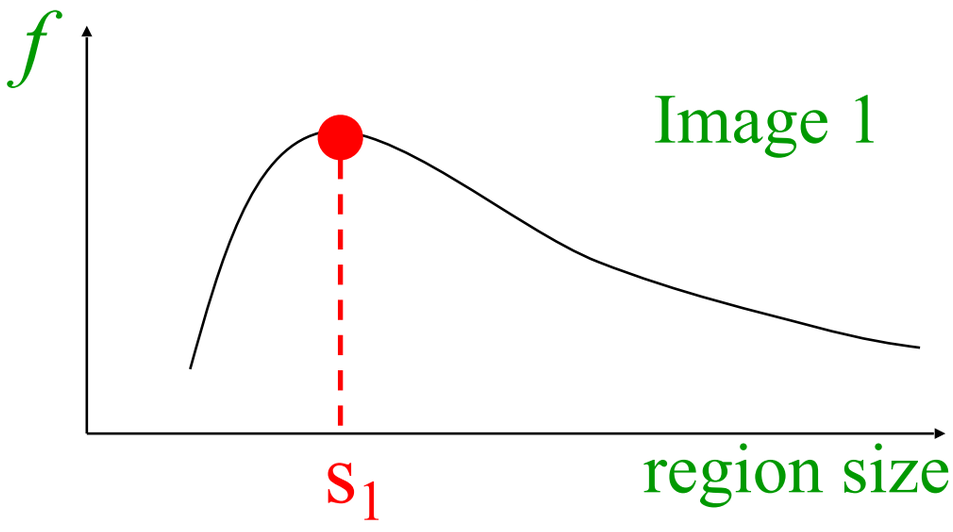
Automatic scale selection

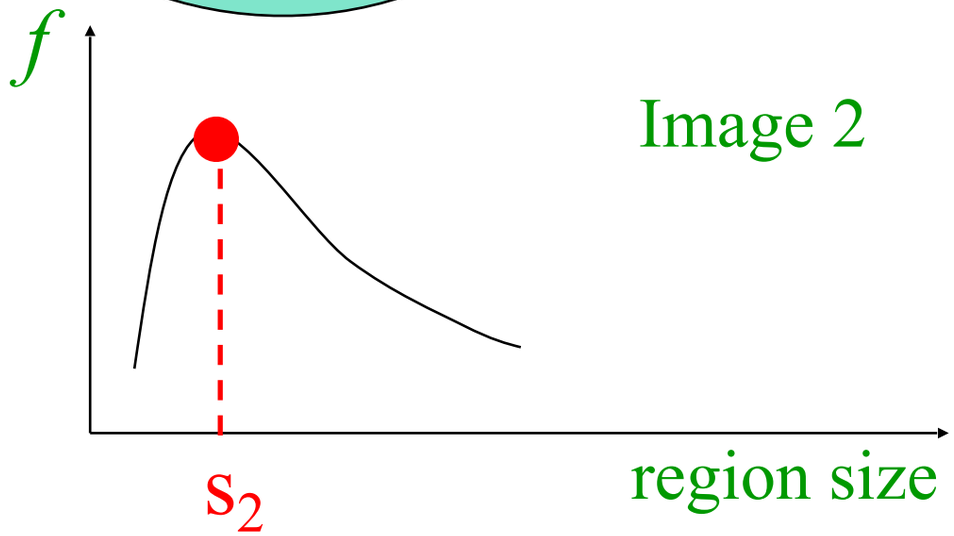
Intuition:

- Find scale that gives local maxima of some function *f* in both ~~position~~ and scale.



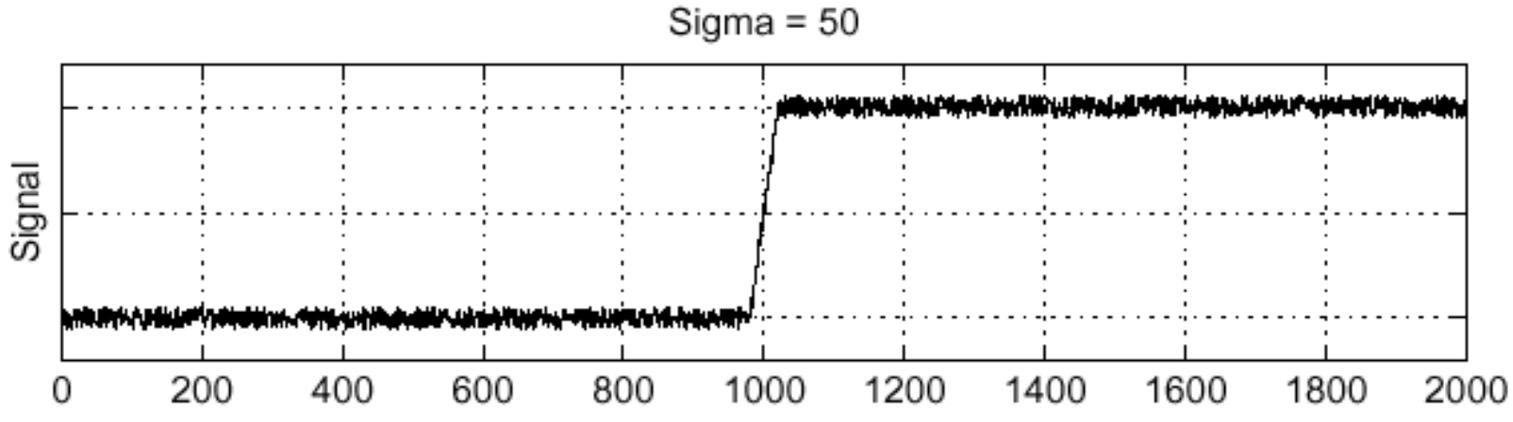






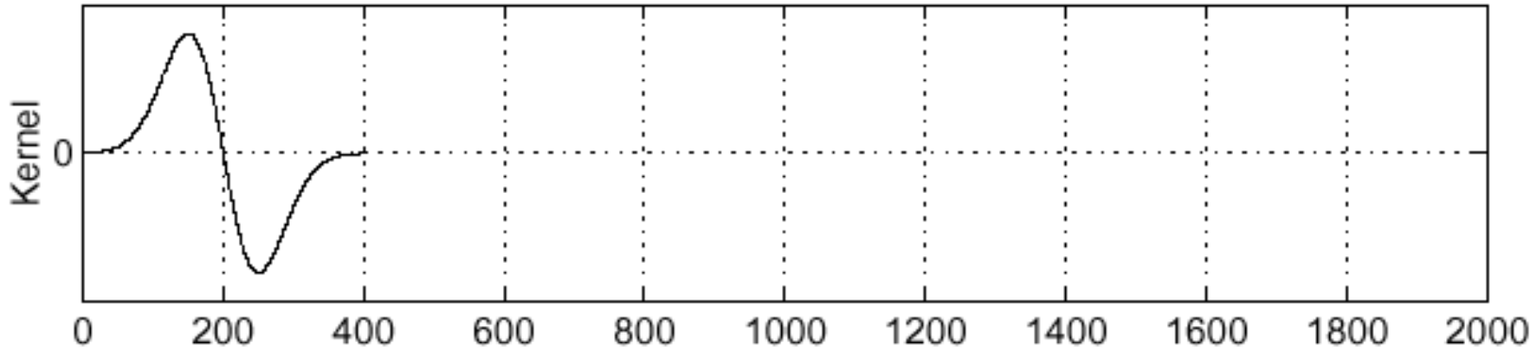
Recall: Edge detection

f



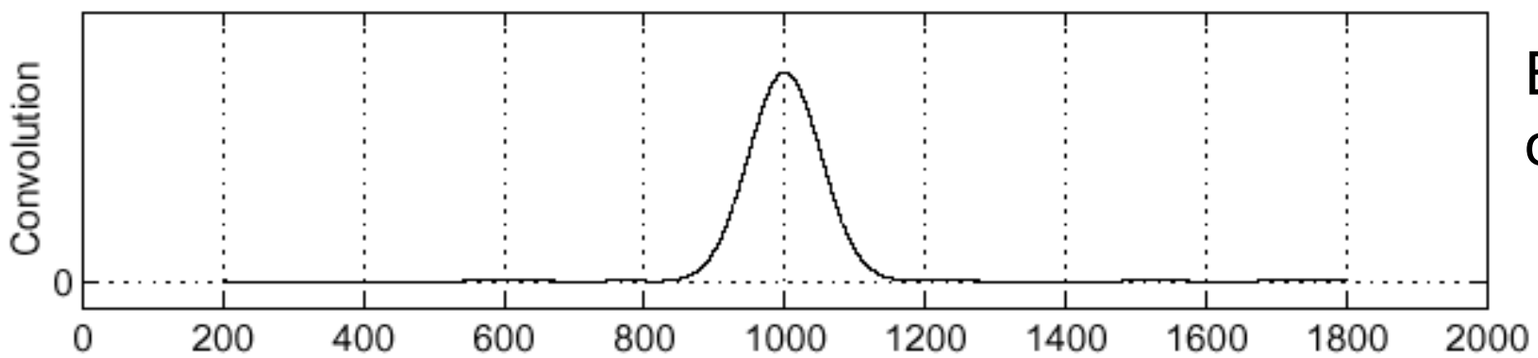
Edge

$\frac{d}{dx} g$



Derivative of Gaussian

$f * \frac{d}{dx} g$

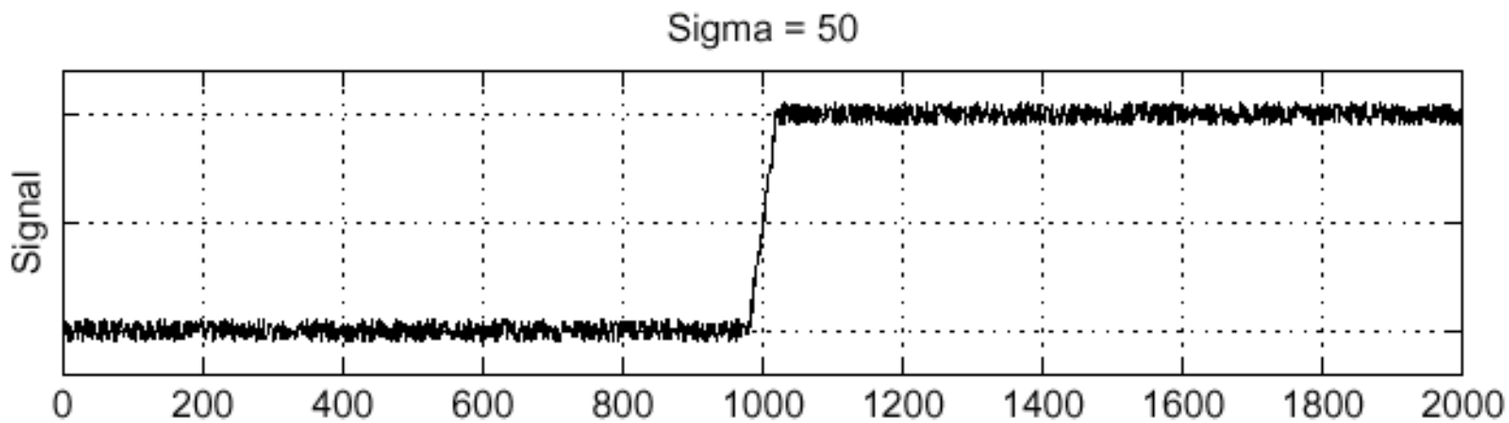


Edge = maximum of derivative

Source: S. Seitz

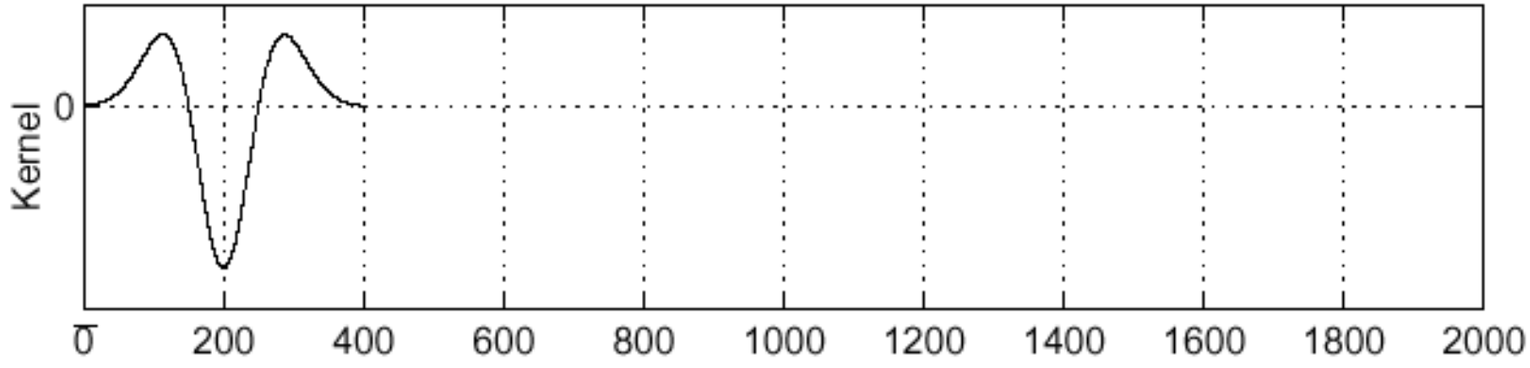
Recall: Edge detection

f



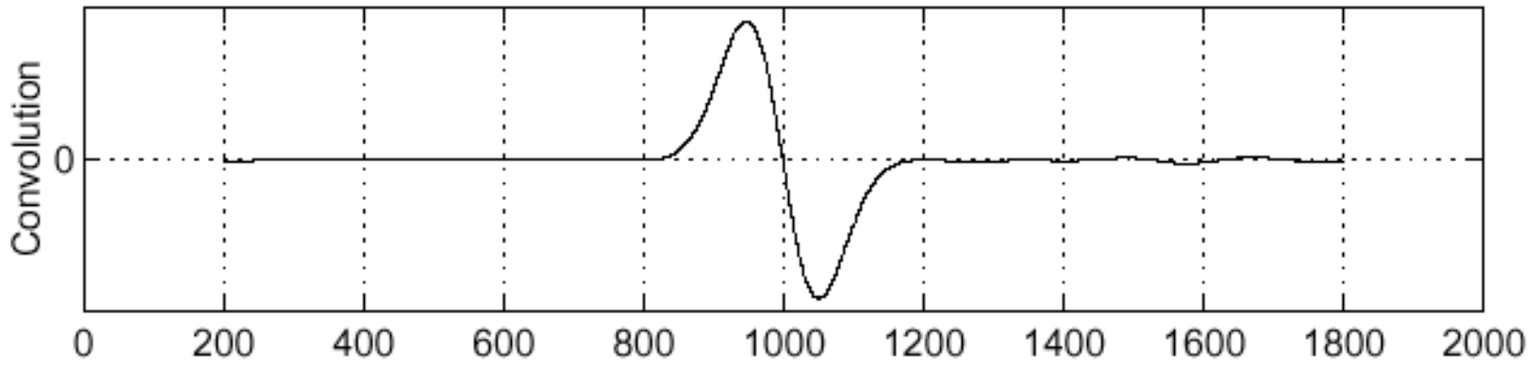
Edge

$\frac{d^2}{dx^2} g$



Second derivative of Gaussian (Laplacian)

$f * \frac{d^2}{dx^2} g$

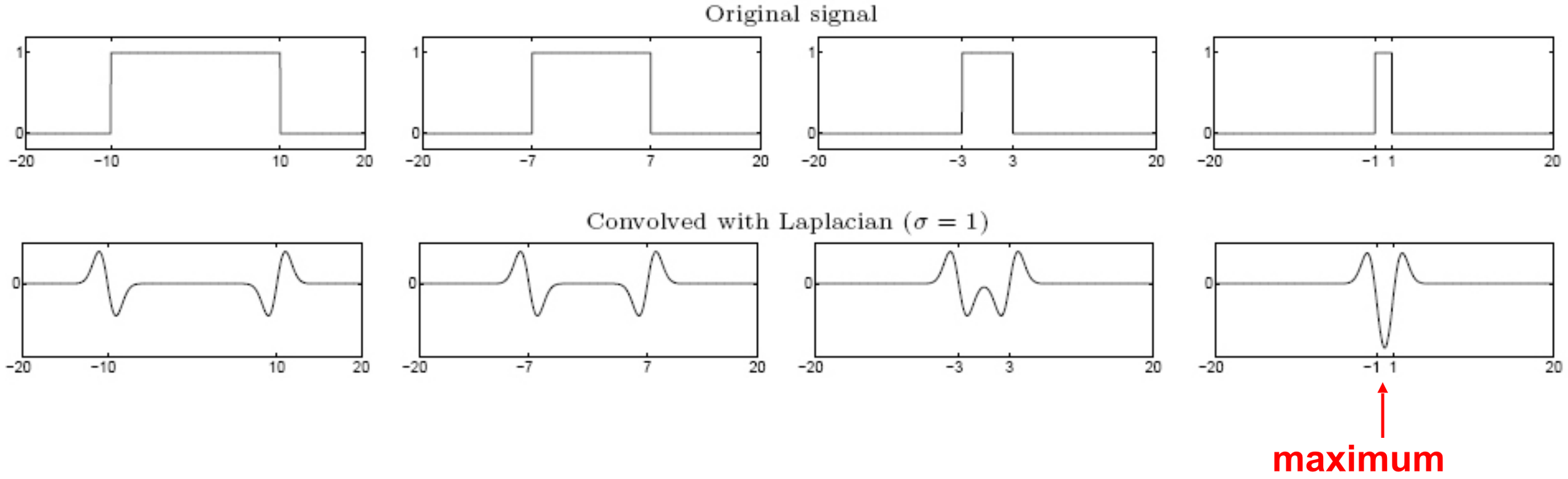


Edge = zero crossing of second derivative

Source: S. Seitz

From edges to blobs

- Edge = ripple
- Blob = superposition of two ripples

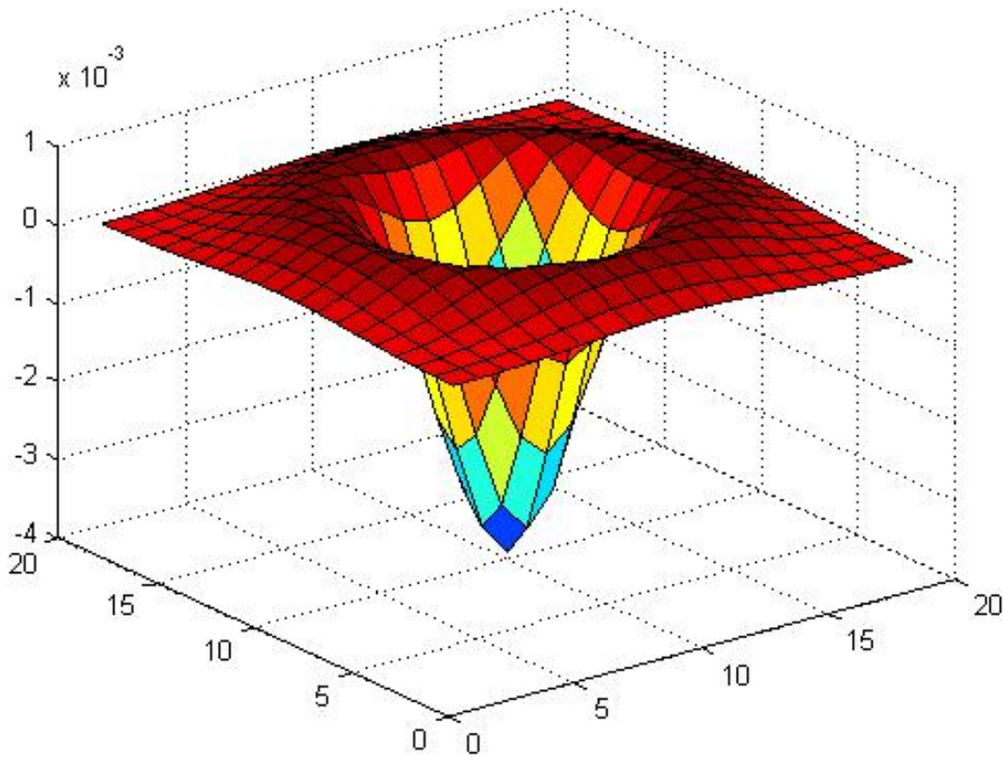


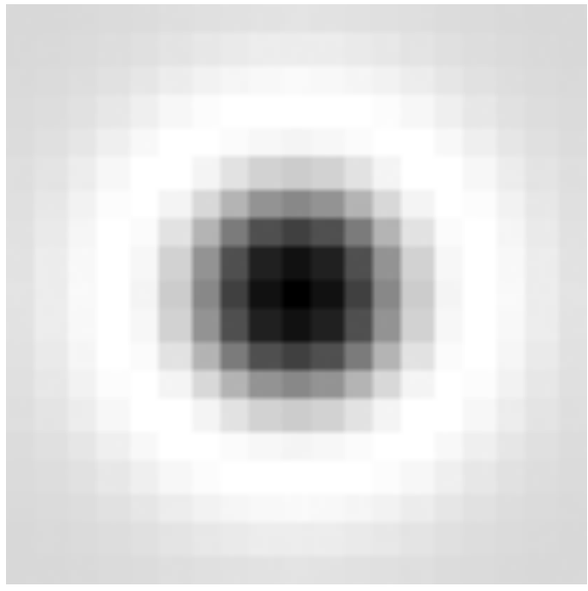
Spatial selection: the **magnitude** of the Laplacian response will achieve a maximum at the center of the blob, provided the scale of the Laplacian is "matched" to the scale of the blob

Slide credit: Lana Lazebnik

Blob detection in 2D

Laplacian of Gaussian: Circularly symmetric operator for blob detection in 2D

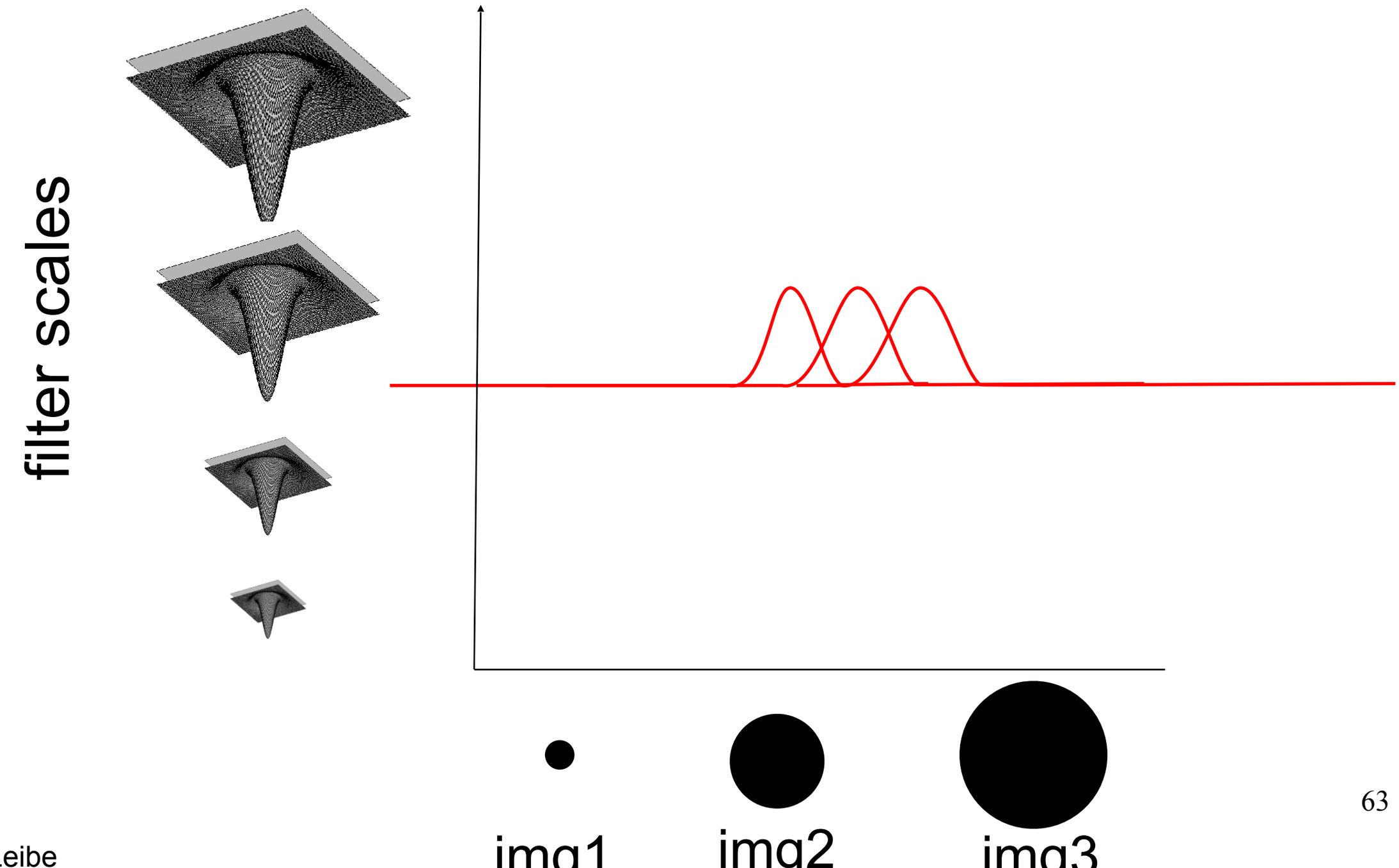




$$\nabla^2 g = \frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2}$$

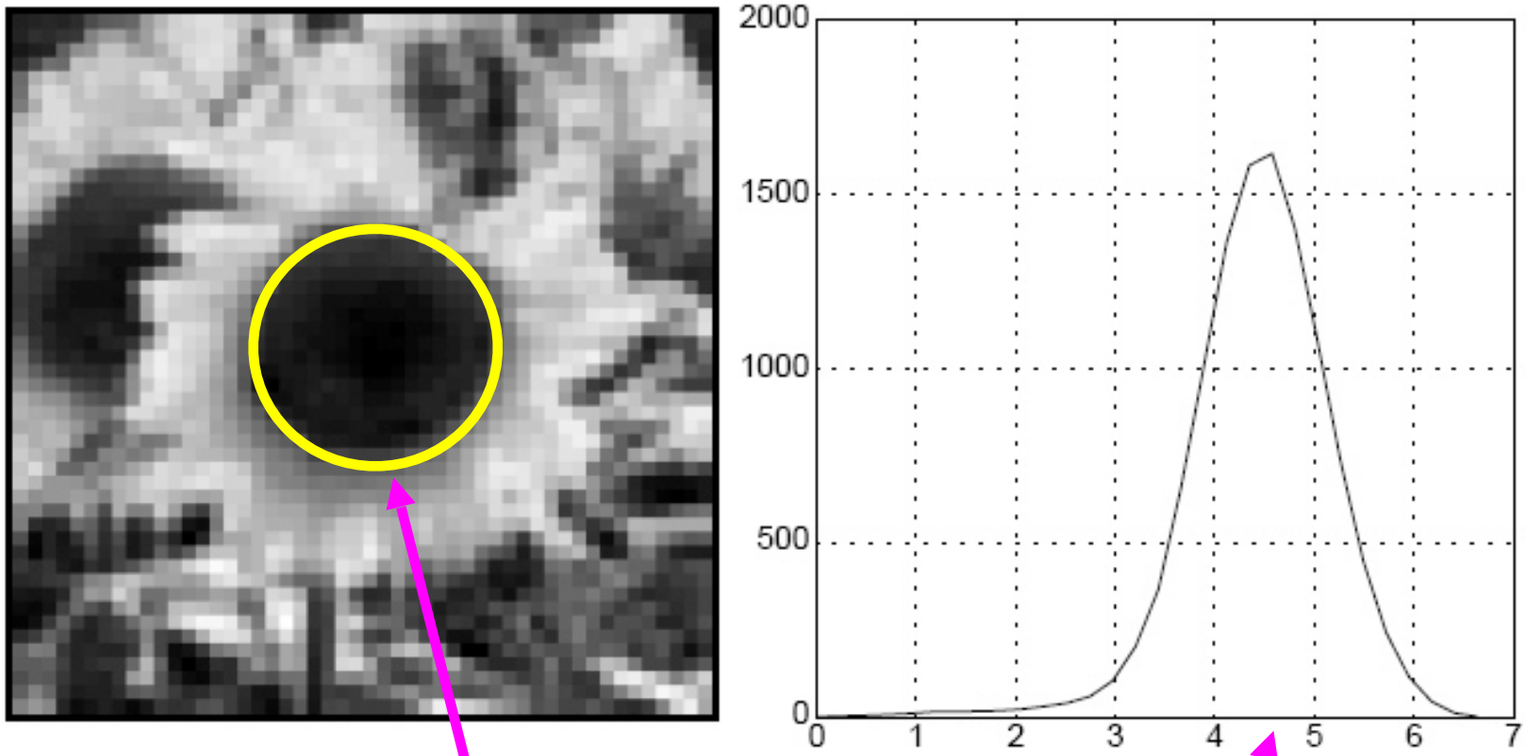
Blob detection in 2D: scale selection

Laplacian-of-Gaussian = "blob" detector $\nabla^2 g = \frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2}$



Blob detection in 2D

We define the *characteristic scale* as the scale that produces peak of Laplacian response

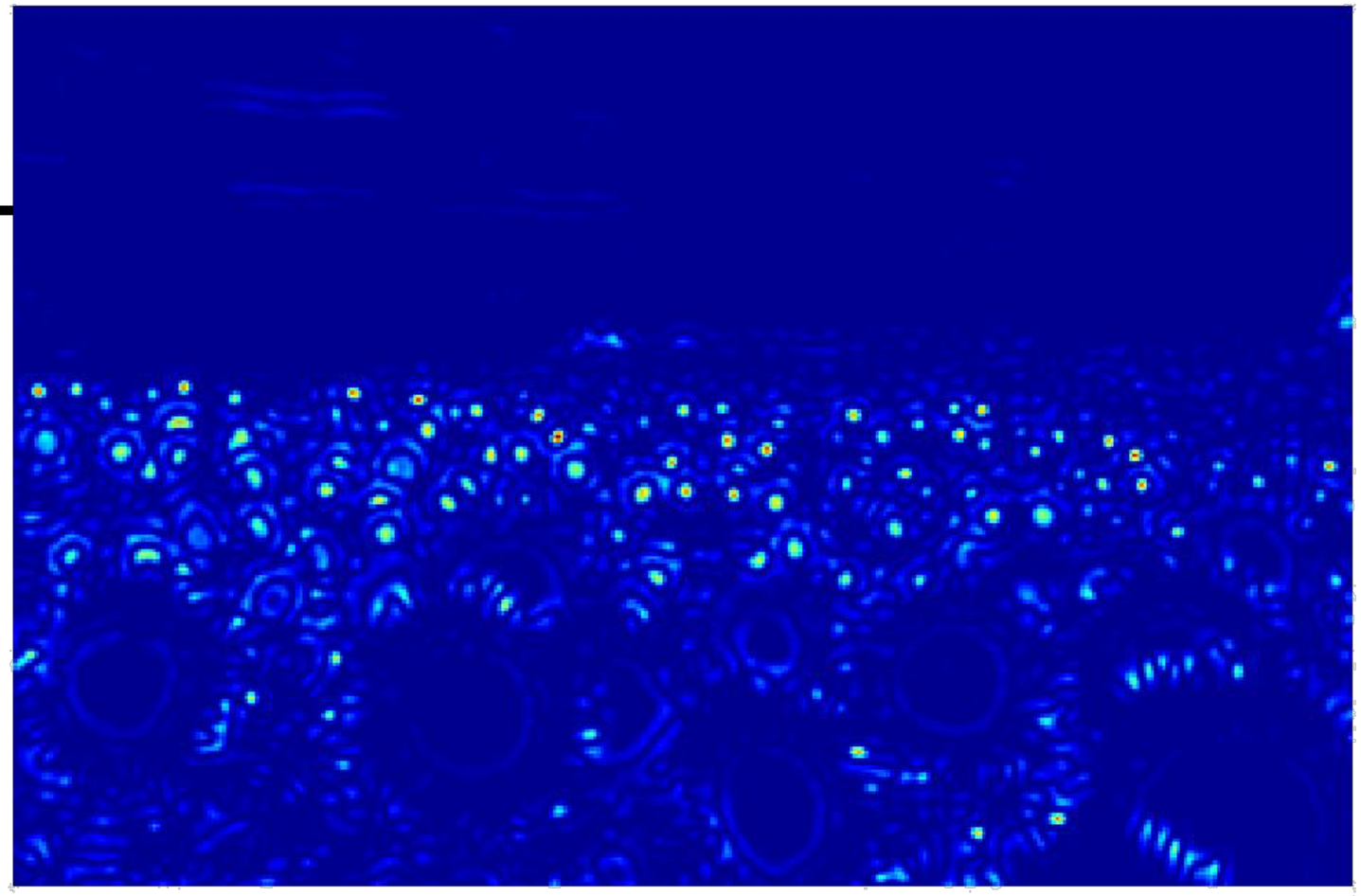


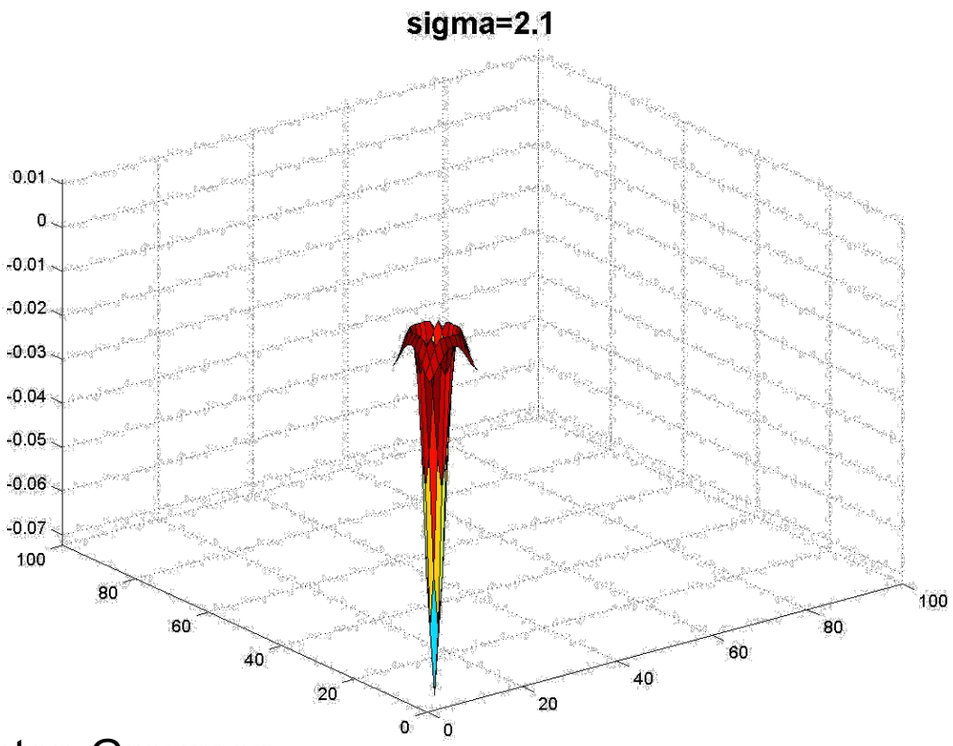
Slide credit: Lana Lazebnik

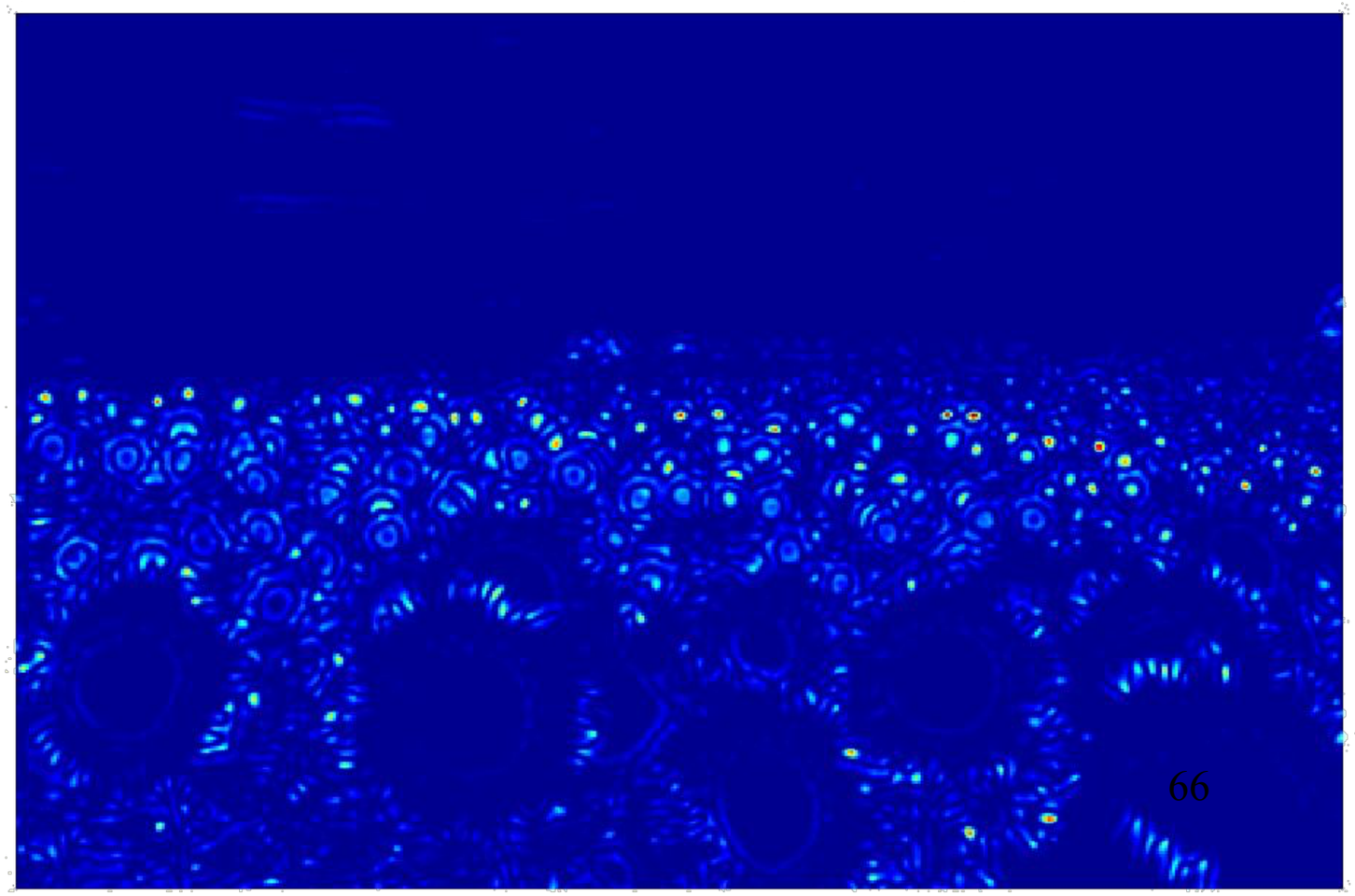
Example

Original image at
¾ the size

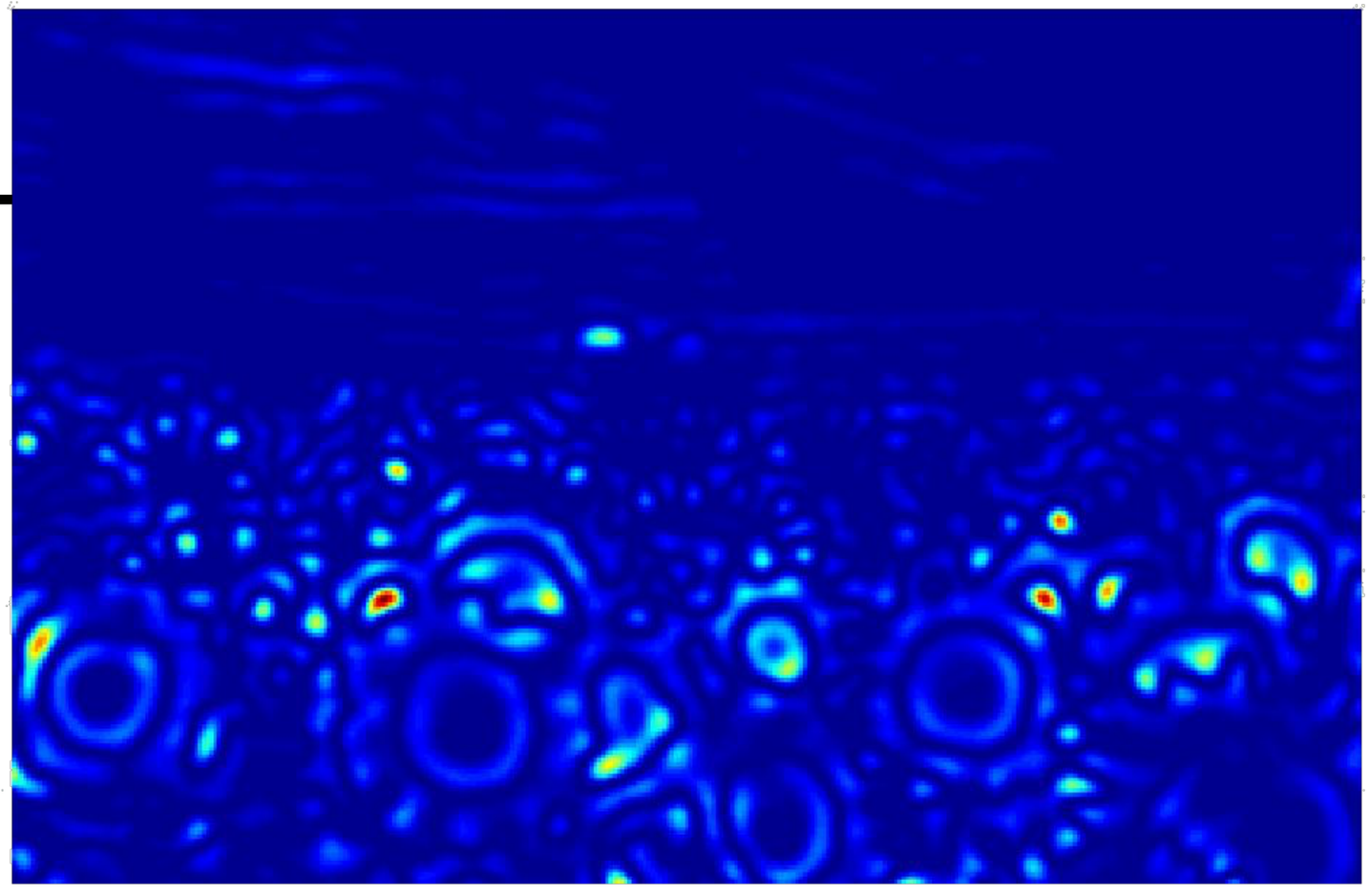
Original image at ¾ the size

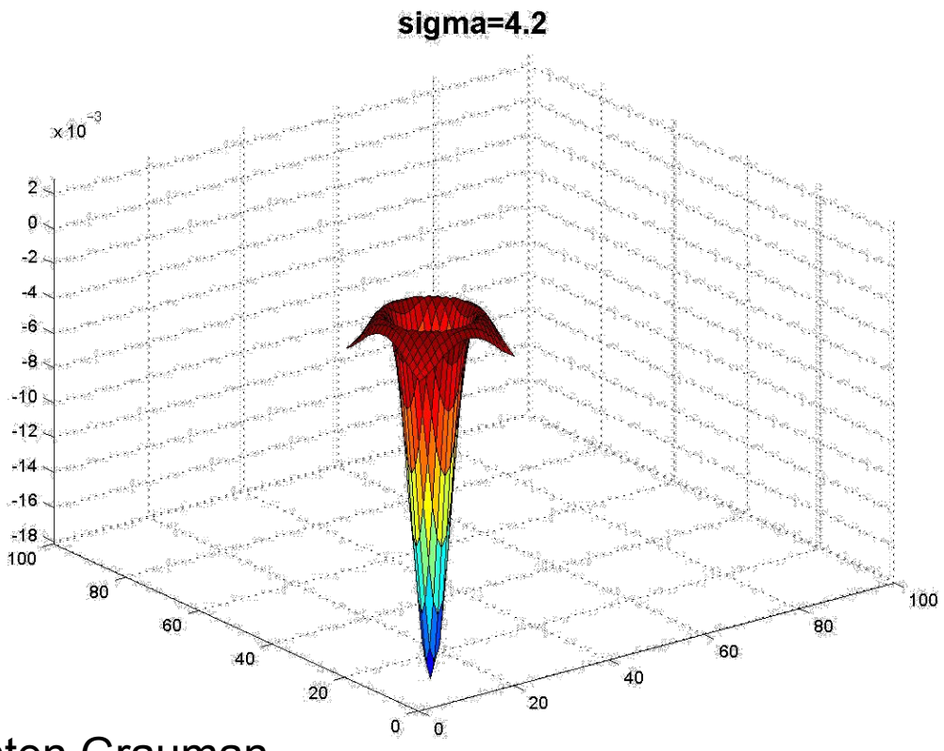


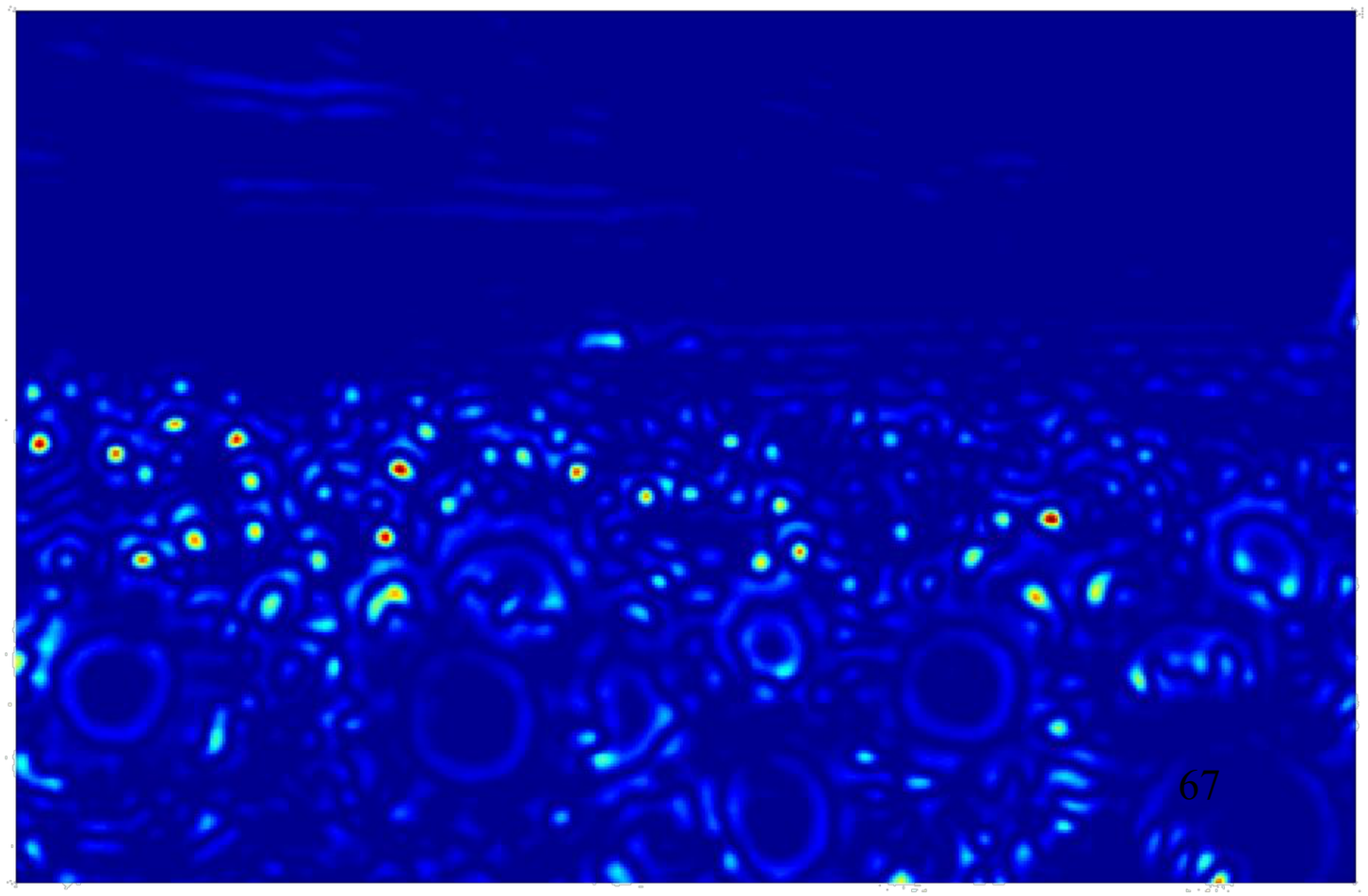




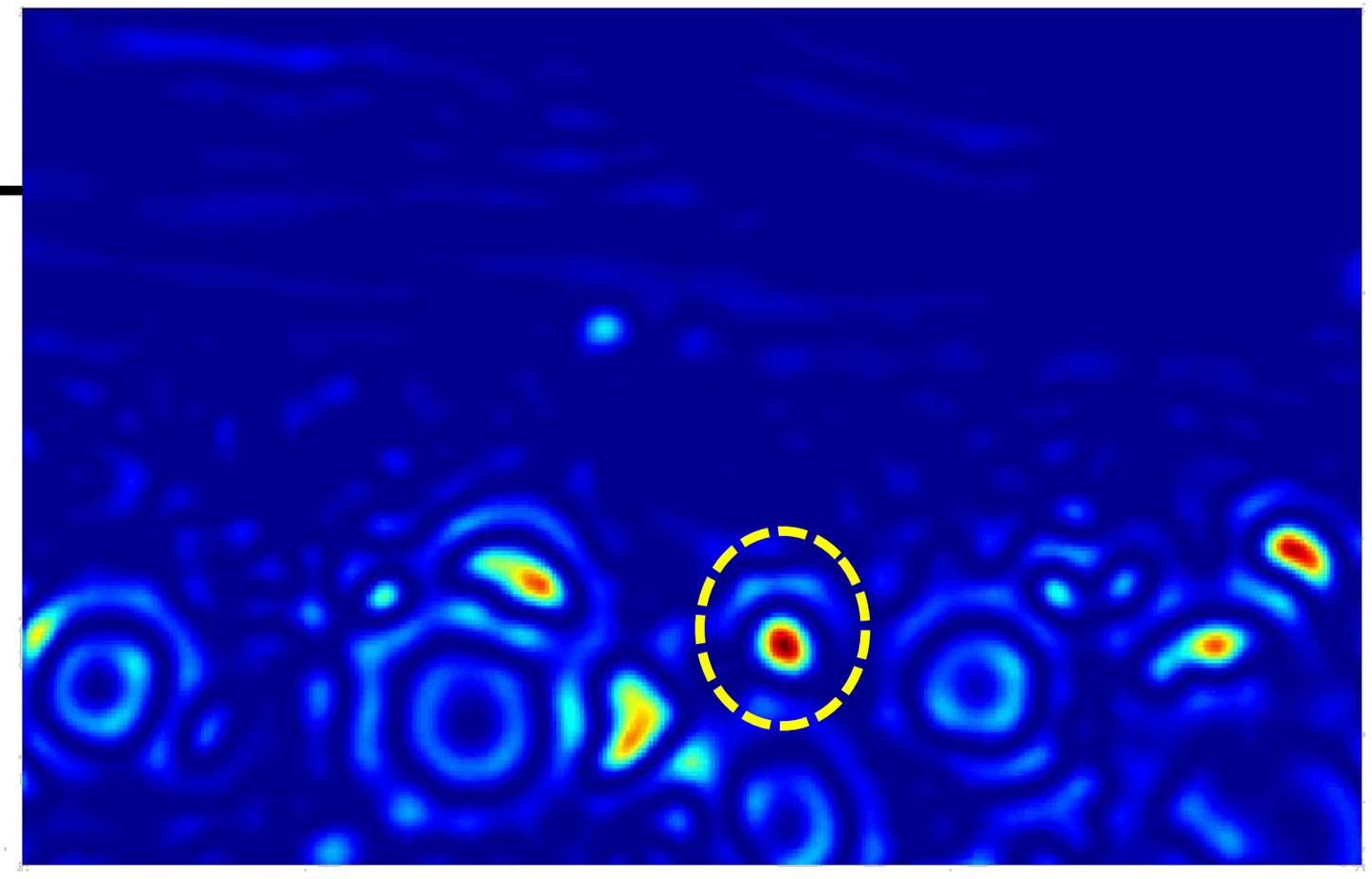
Kristen Grauman

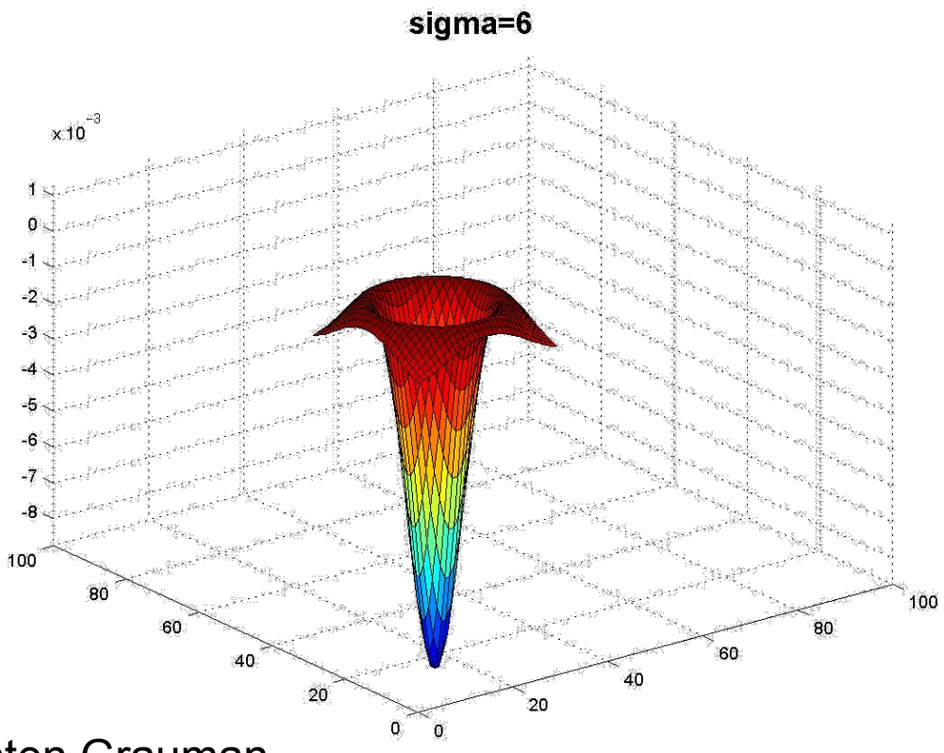


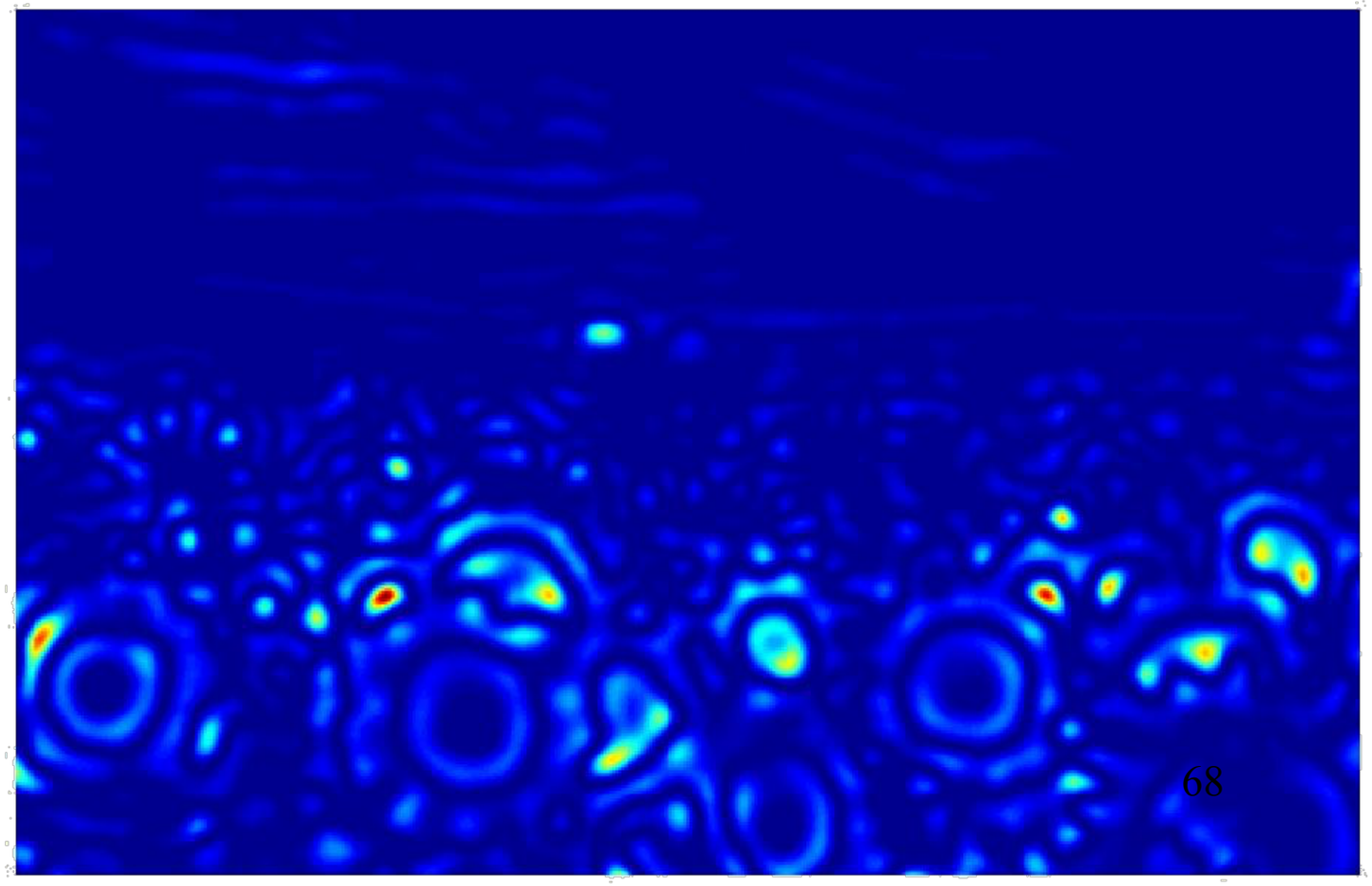


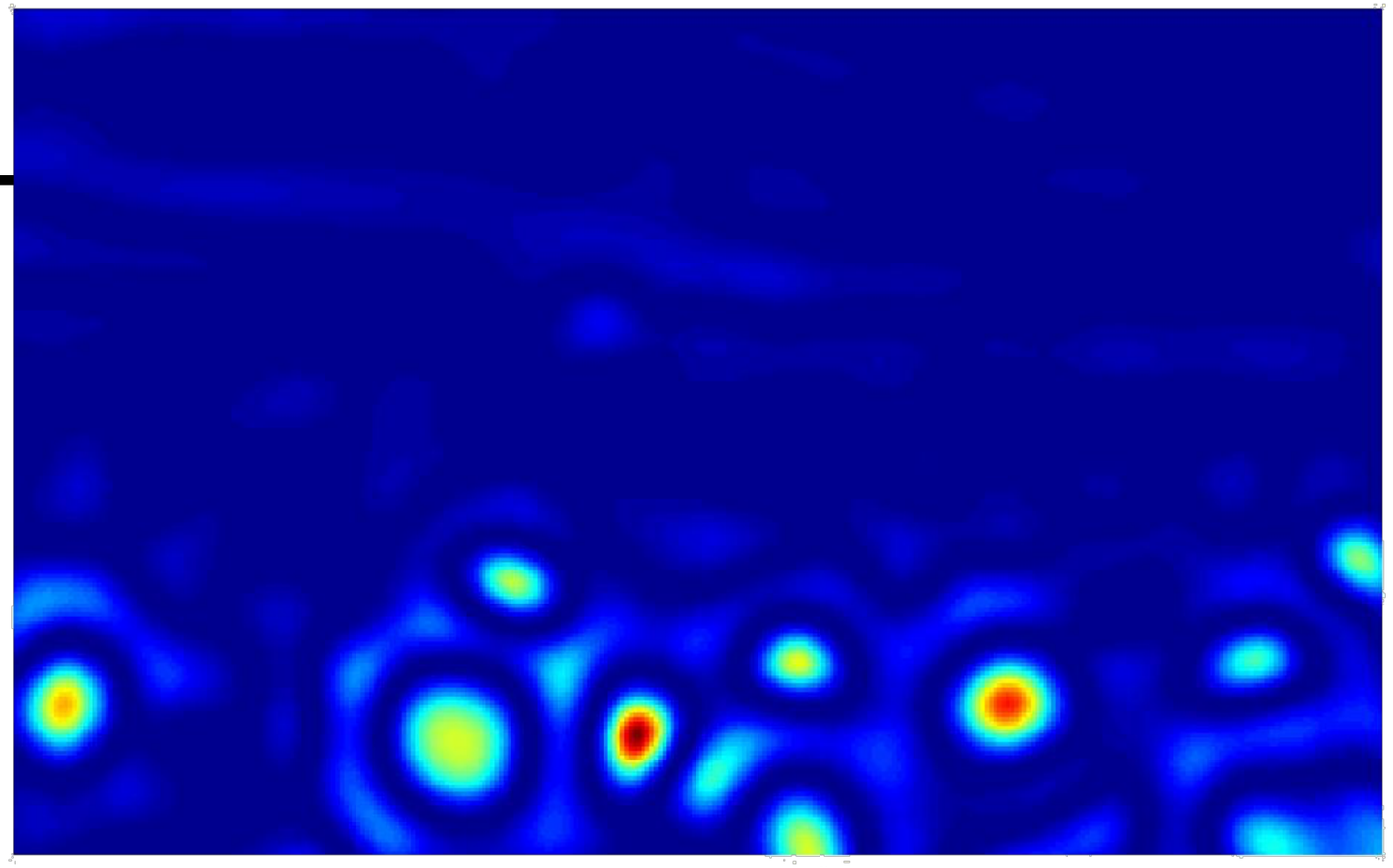


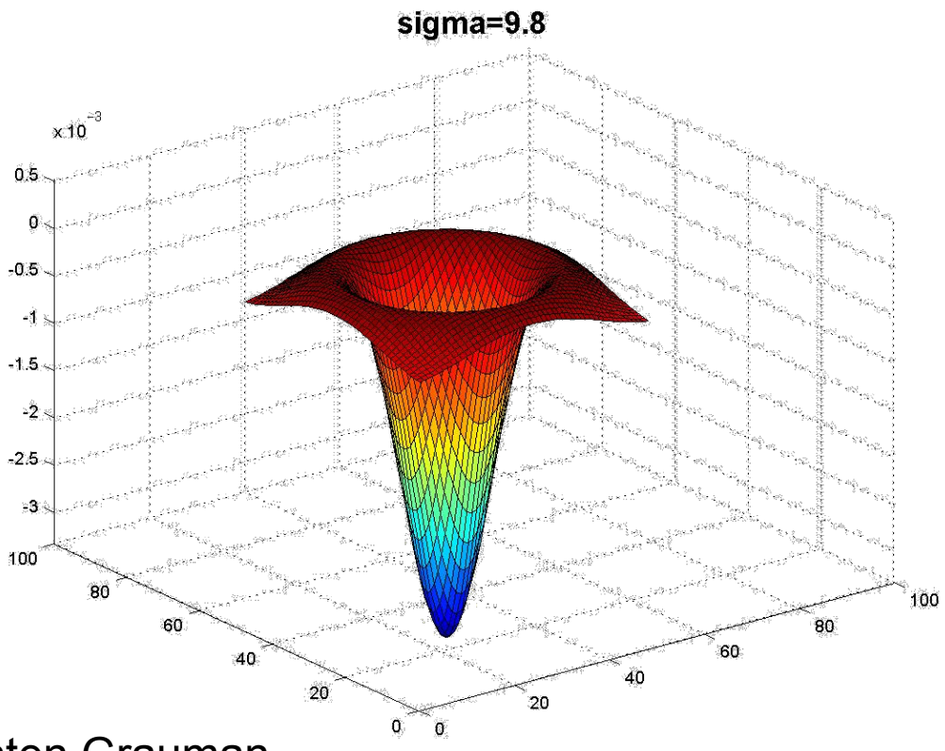
Kristen Grauman

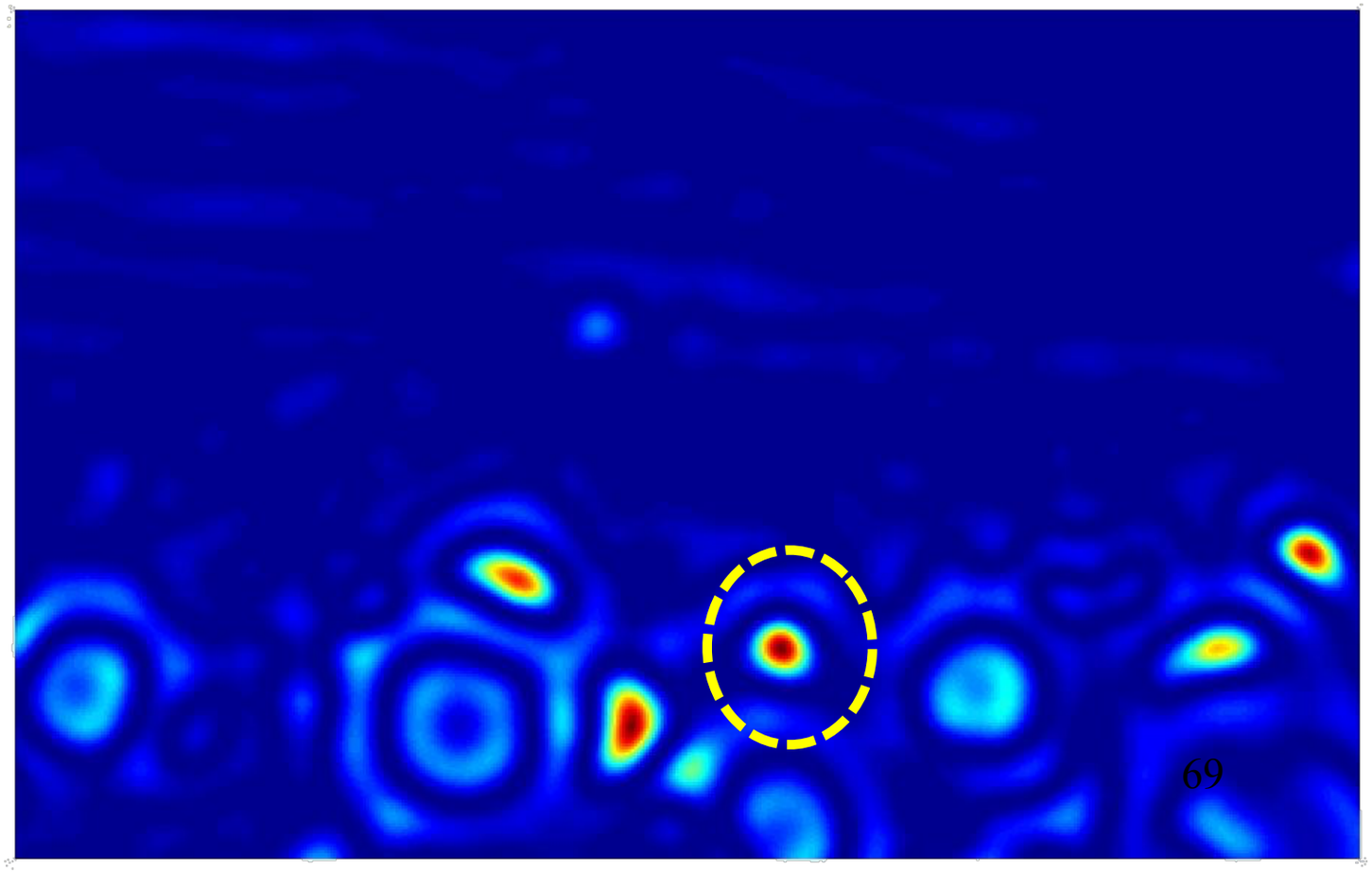


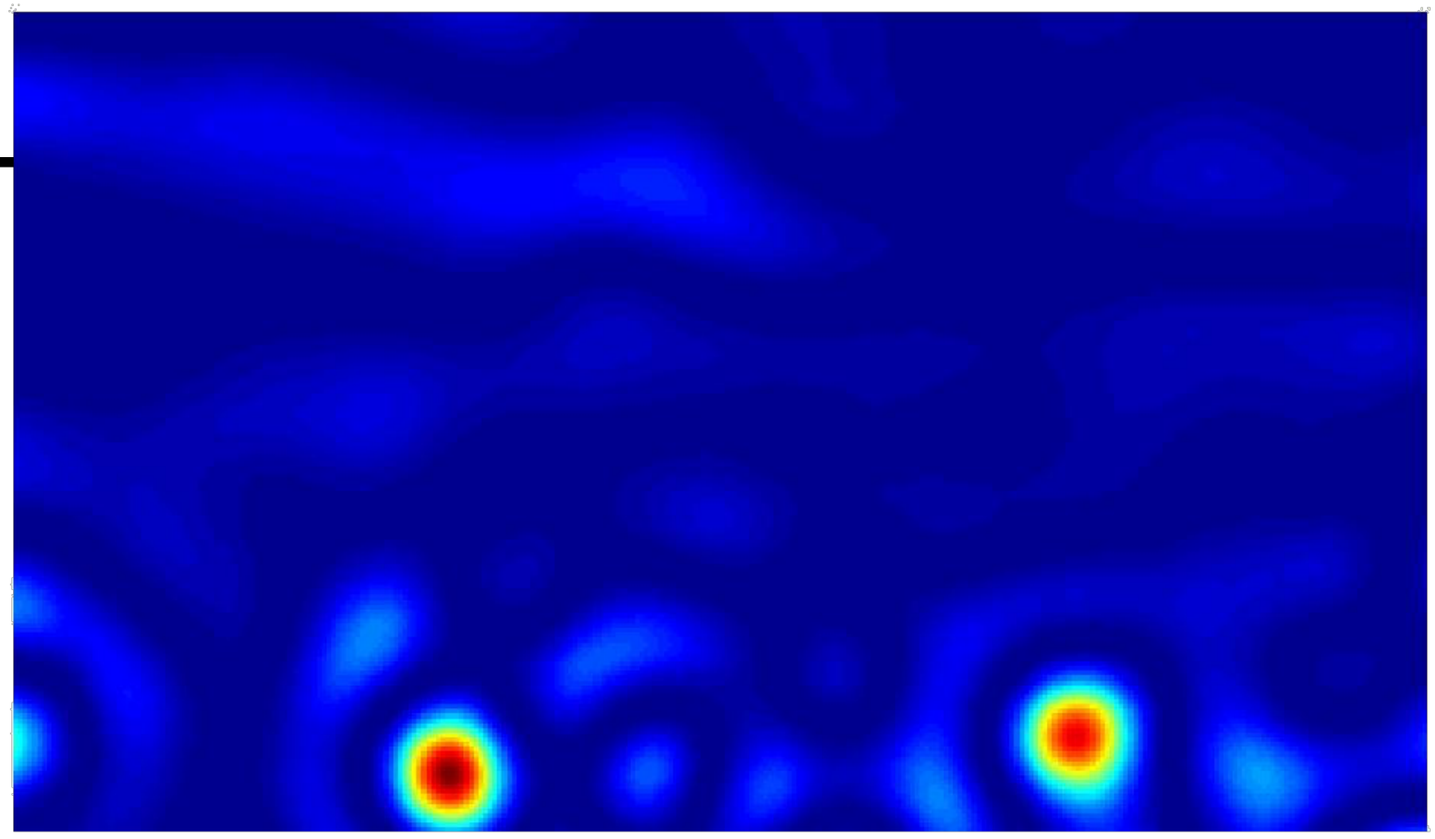

sigma=6
x 10^-3
1
0
-1
-2
-3
-4
-5
-6
-7
-8
100
80
60
40
20
0
0
20
40
60
80
100

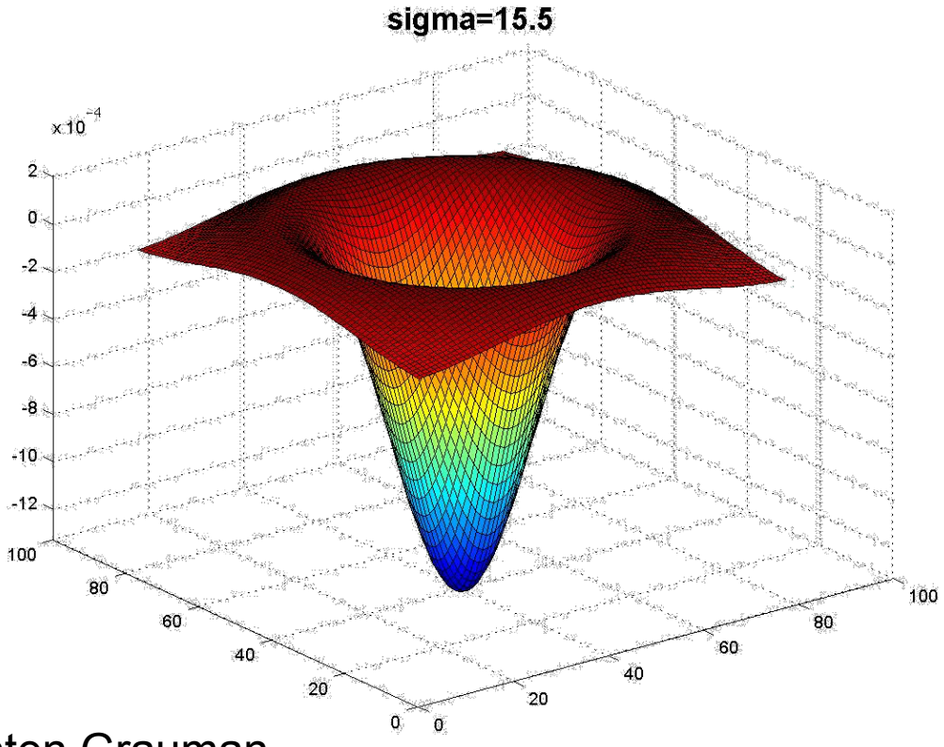


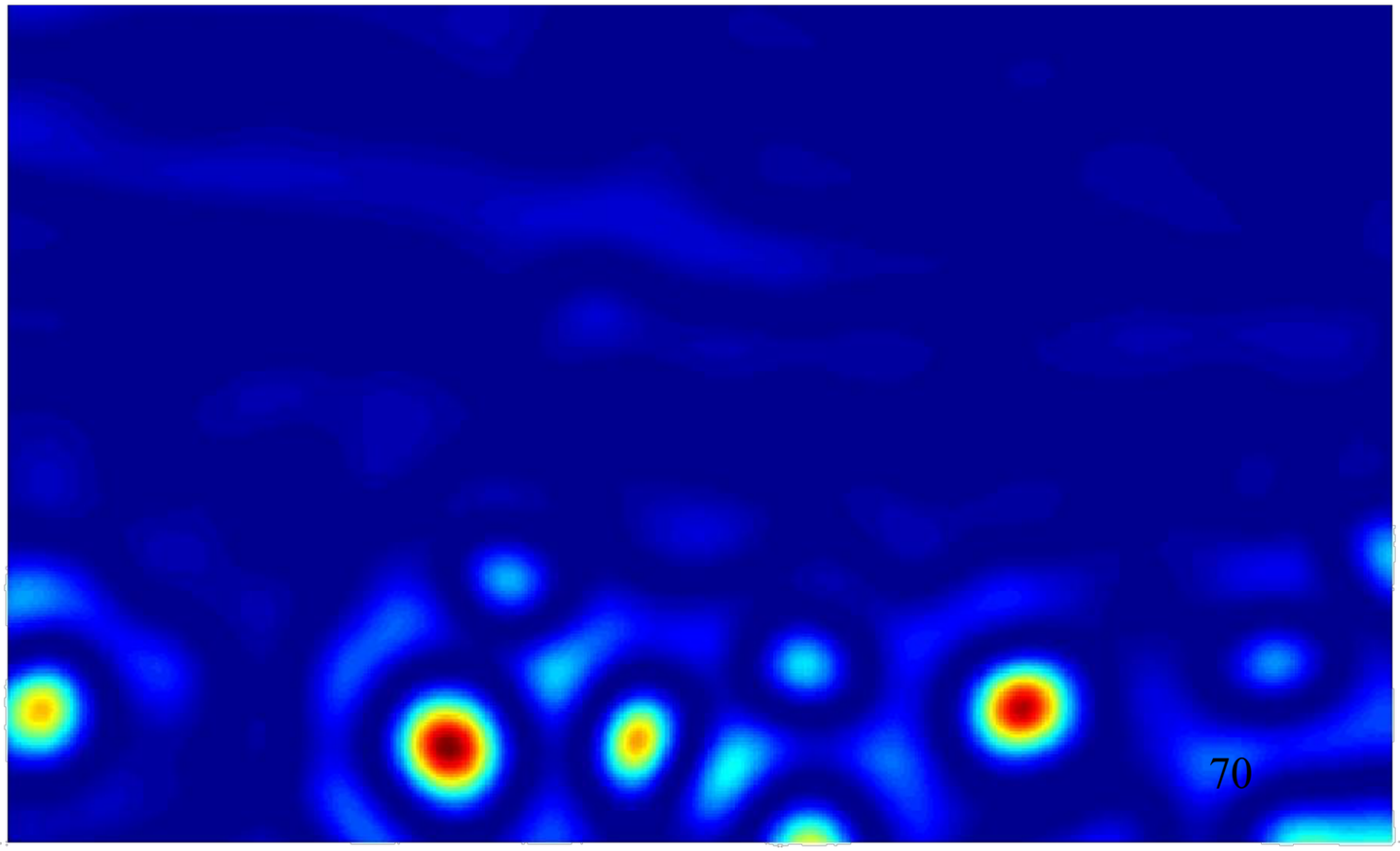



sigma=9.8
x 10^-3
0.5
0
-0.5
-1
-1.5
-2
-2.5
-3
100
80
60
40
20
0
0
20
40
60
80
100





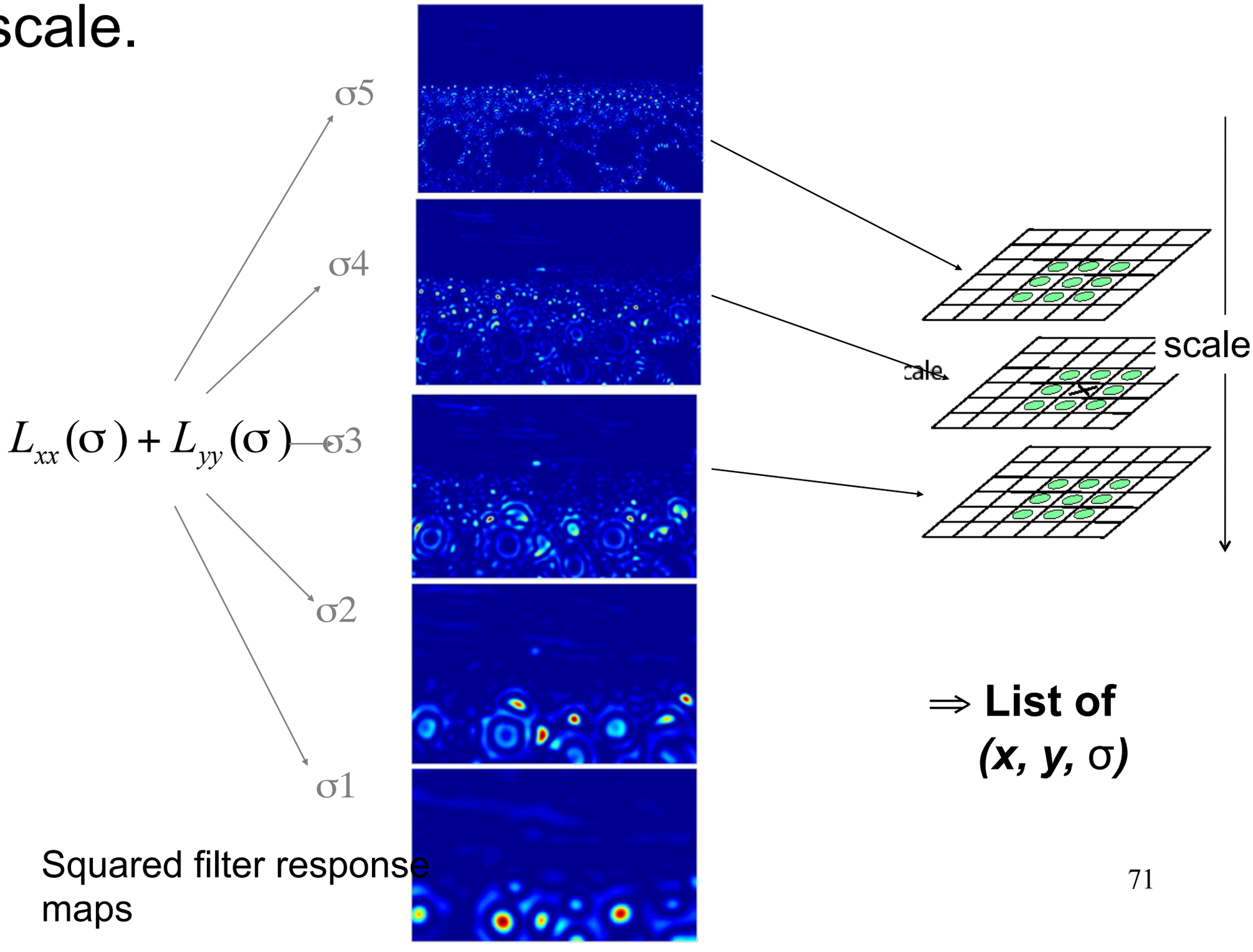




Kristen Grauman

Scale invariant interest points

Interest points are local maxima in both position and scale.



Squared filter response maps

⇒ **List of** ***(x, y,*** σ***)***

Scale-space blob detector: Example

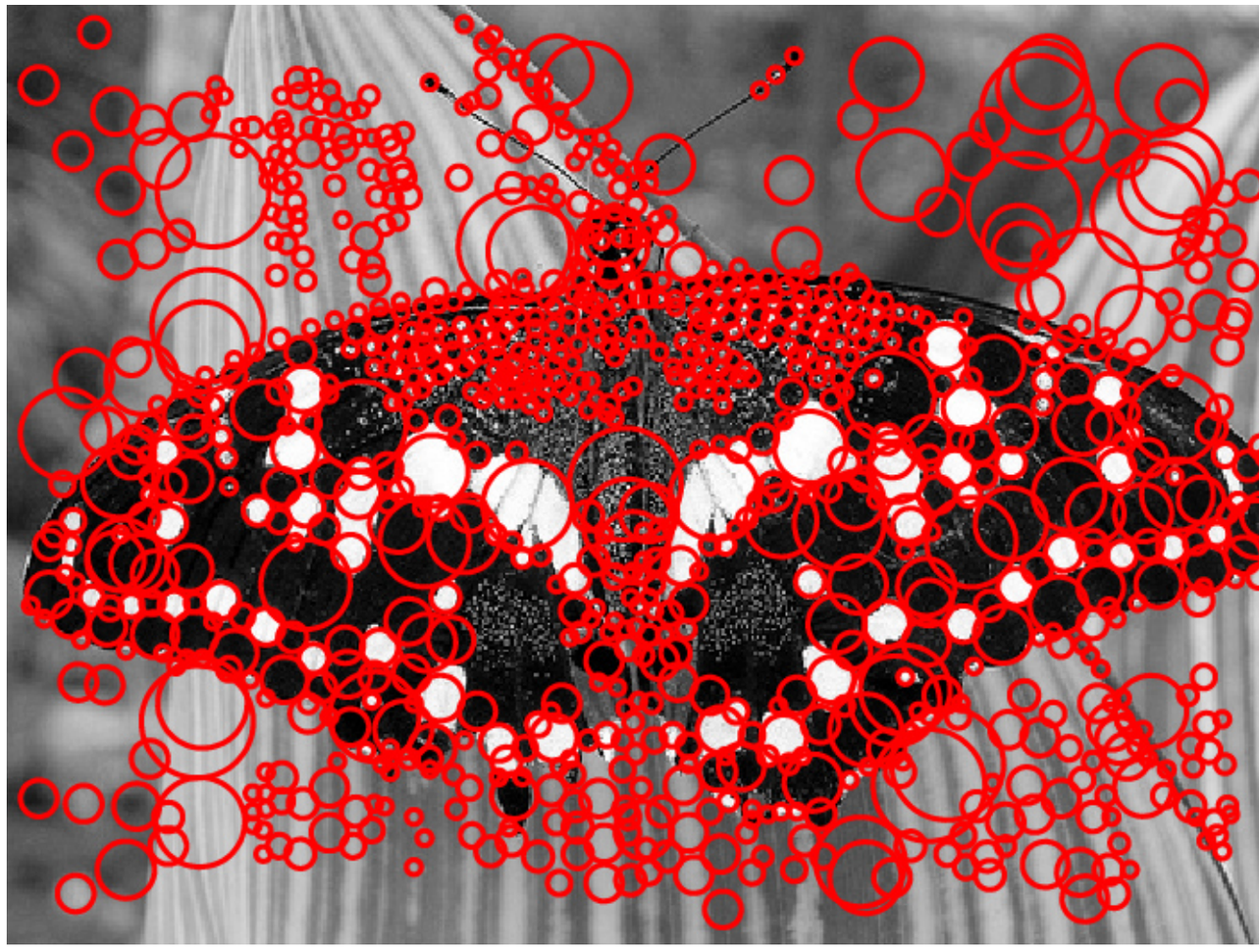


Image credit: Lana Lazebnik

Technical detail

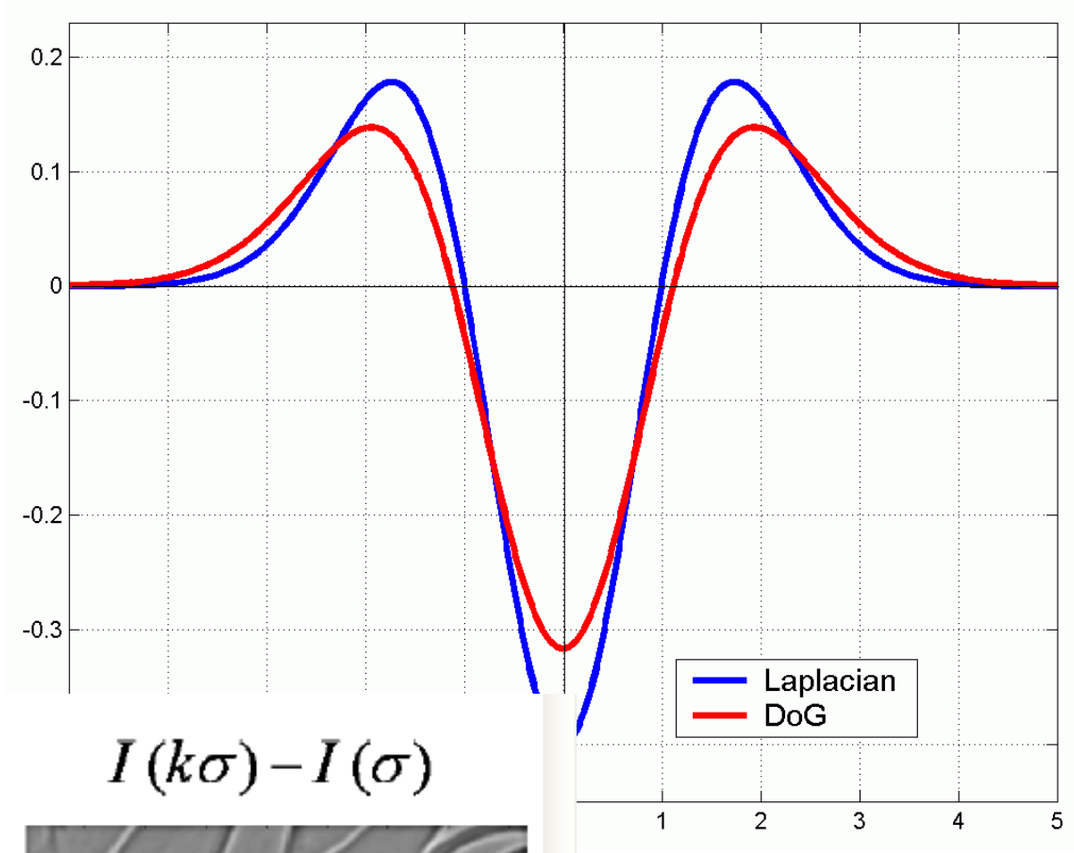
We can approximate the Laplacian with a difference of Gaussians; more efficient to implement.

$$L = \sigma^2 \left(G_{xx}(x, y, \sigma) + G_{yy}(x, y, \sigma) \right)$$

(Laplacian)

$$DoG = G(x, y, k\sigma) - G(x, y, \sigma)$$

(Difference of Gaussians)

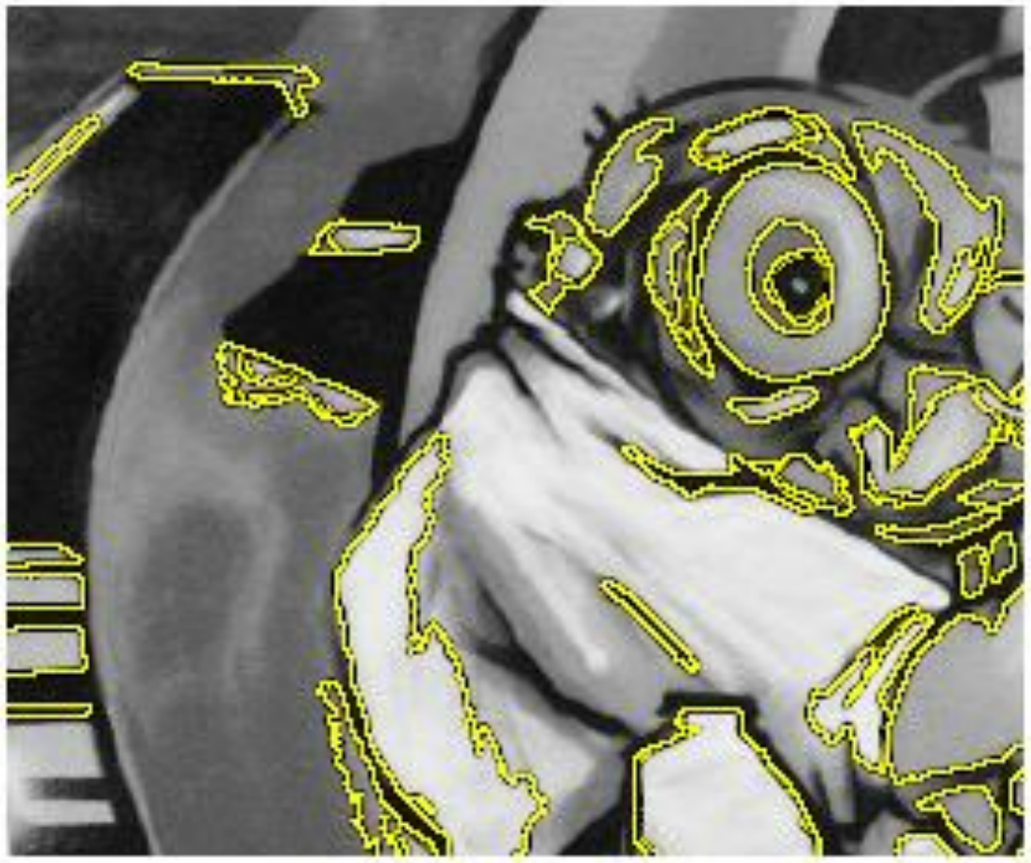


$I(k\sigma)$ $I(\sigma)$ $I(k\sigma) - I(\sigma)$

Maximally Stable Extremal Regions [Matas '02]

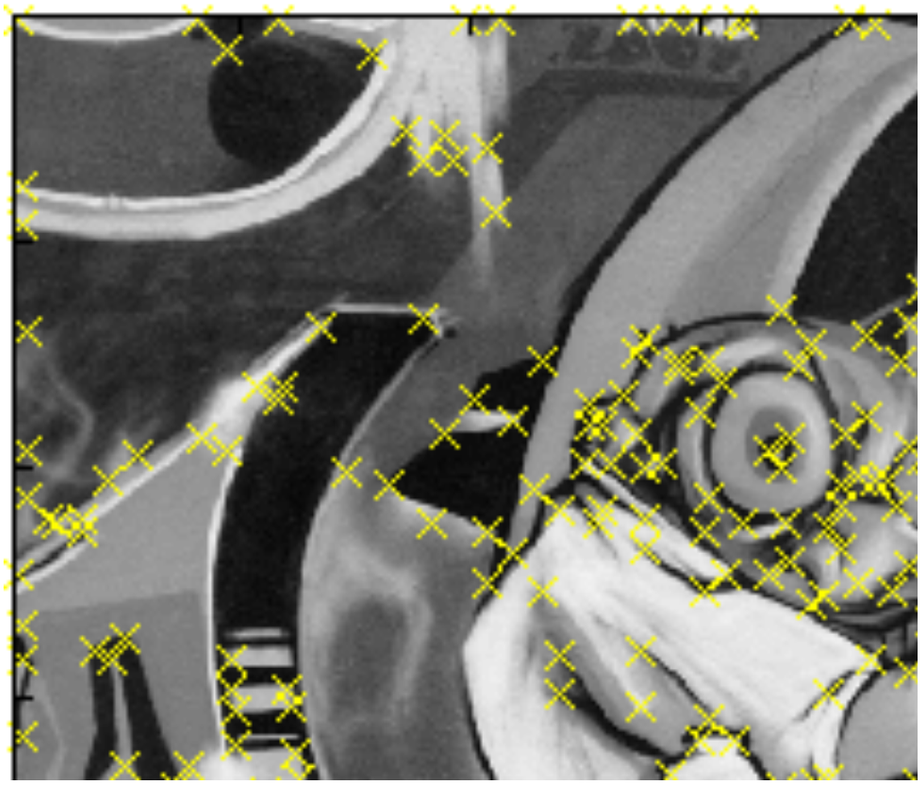
- Based on Watershed segmentation algorithm
- Select regions that stay stable over a large parameter range

Example Results: MSER



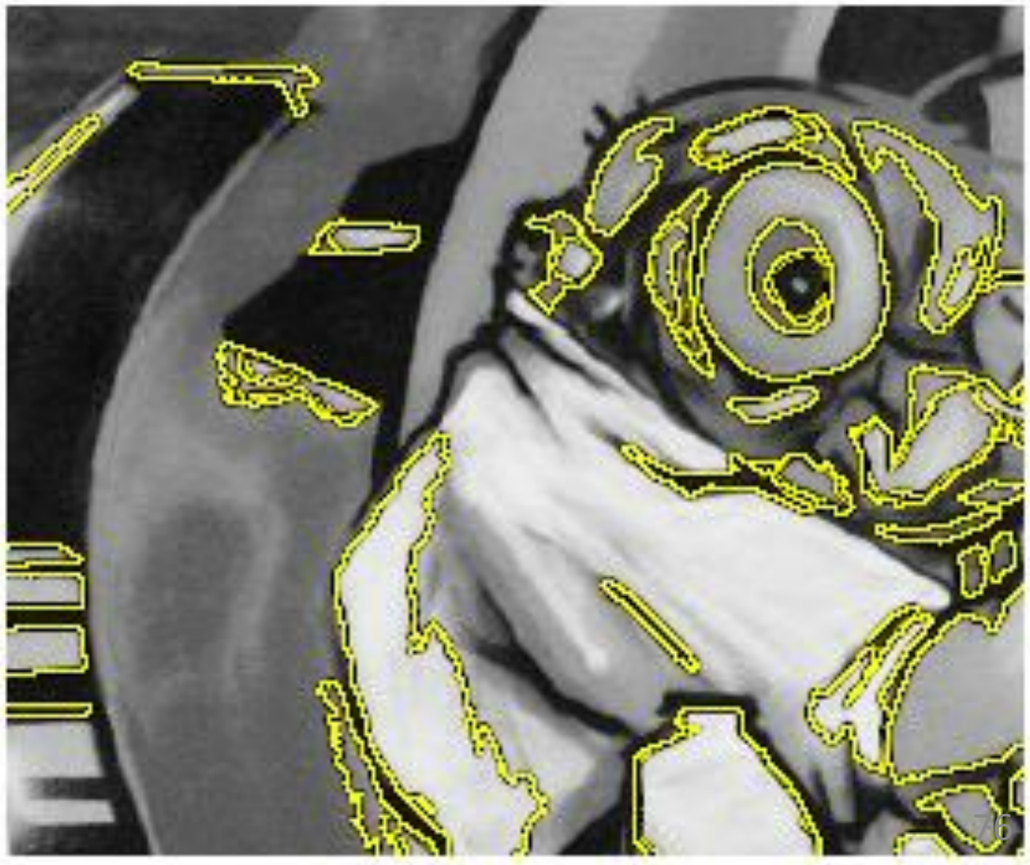
Comparison

Harris



LoG

MSER



Comparison of Keypoint Detectors

Table 7.1 Overview of feature detectors.

Feature Detector	Corner	Blob	Region	Rotation invariant	Scale invariant	Affine invariant	Repeatability	Localization accuracy	Robustness	Efficiency
Harris	√			√			+++	+++	+++	++
Hessian		√		√			++	++	++	+
SUSAN	√			√			++	++	++	+++
Harris-Laplace	√	(√)		√	√		+++	+++	++	+
Hessian-Laplace	(√)	√		√	√		+++	+++	+++	+
DoG	(√)	√		√	√		++	++	++	++
SURF	(√)	√		√	√		++	++	++	+++
Harris-Affine	√	(√)		√	√	√	+++	+++	++	++
Hessian-Affine	(√)	√		√	√	√	+++	+++	+++	++
Salient Regions	(√)	√		√	√	(√)	+	+	++	+
Edge-based	√			√	√	√	+++	+++	+	+
MSER			√	√	√	√	+++	+++	++	+++
Intensity-based			√	√	√	√	++	++	++	++
Superpixels			√	√	(√)	(√)	+	+	+	+

Tuytelaars Mikolajczyk 2008

Choosing a detector

- What do you want it for?
  - Precise localization in x-y: Harris
  - Good localization in scale: Difference of Gaussian
  - Flexible region shape: MSER
- Best choice often application dependent
  - Harris-/Hessian-Laplace/DoG work well for many natural categories
  - MSER works well for buildings and printed things
- Why choose?
  - Get more points with more detectors
- There have been extensive evaluations/comparisons
  - [Mikolajczyk et al., IJCV'05, PAMI'05]
  - All detectors/descriptors shown here work well

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

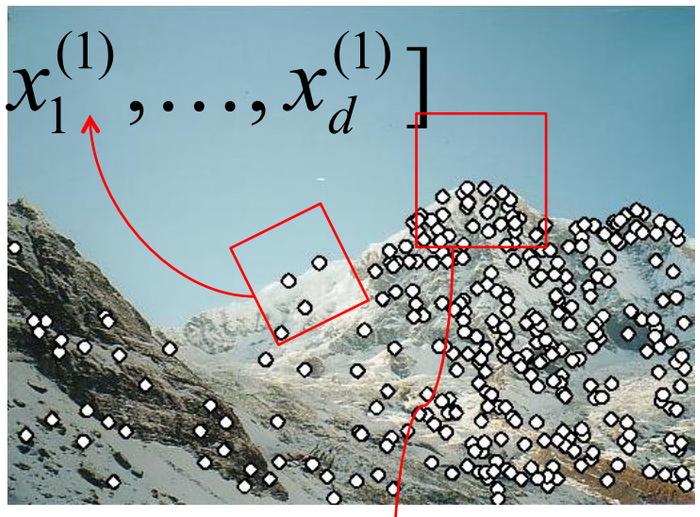
Descriptors

Local features: main components

1) Detection: Identify the interest points

2) Description:Extract vector feature descriptor surrounding each interest point.

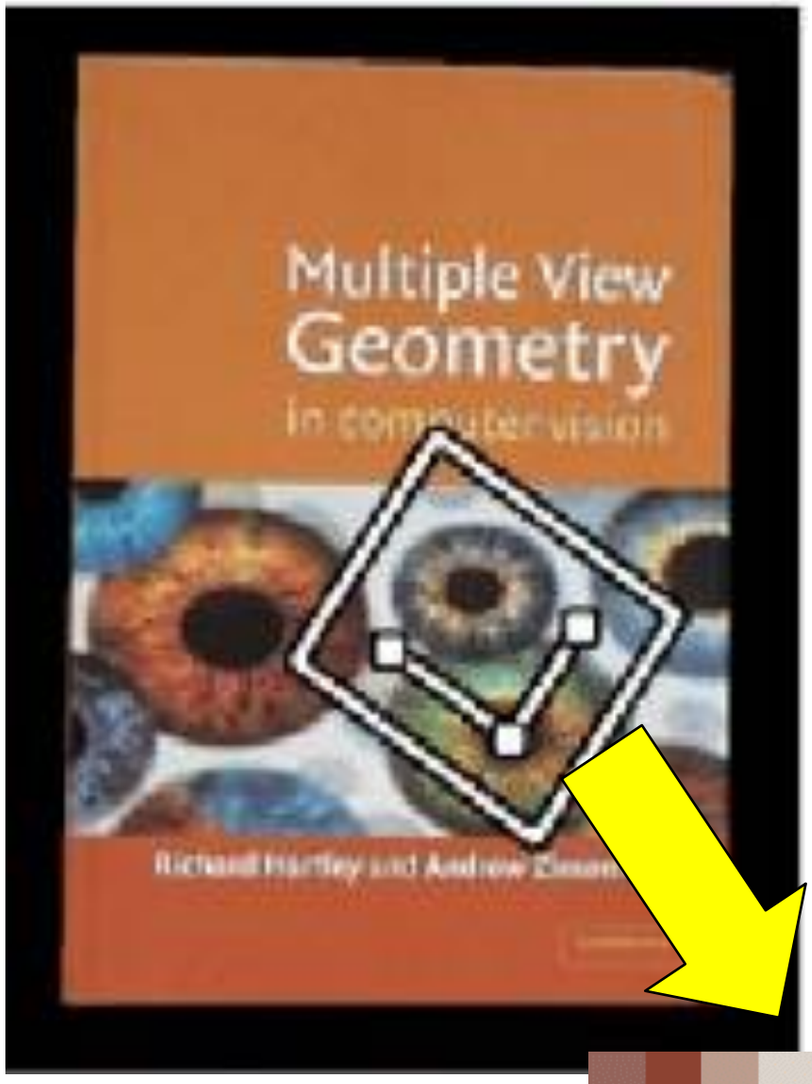
$$\mathbf{x}_1 = [x_1^{(1)}, \ldots, x_d^{(1)}]$$

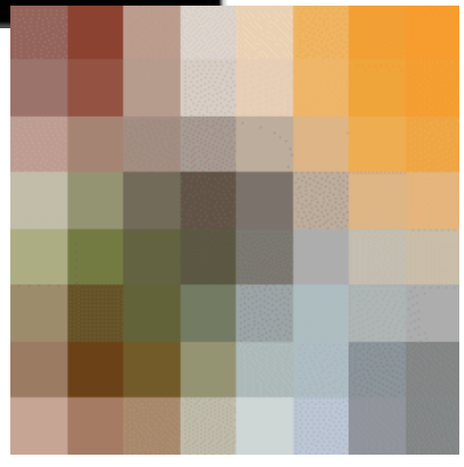


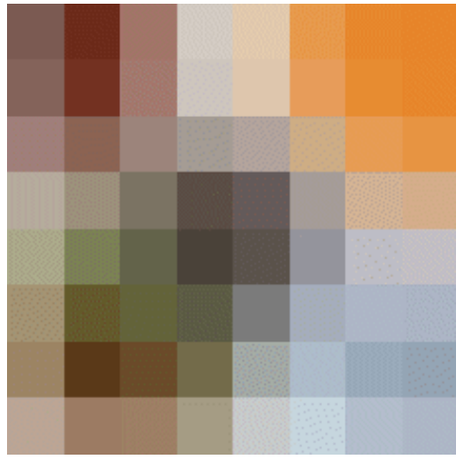
$$\mathbf{x}_2 = [x_1^{(2)}, \ldots, x_d^{(2)}]$$

3) Matching: Determine correspondence between descriptors in two views

Geometric transformations





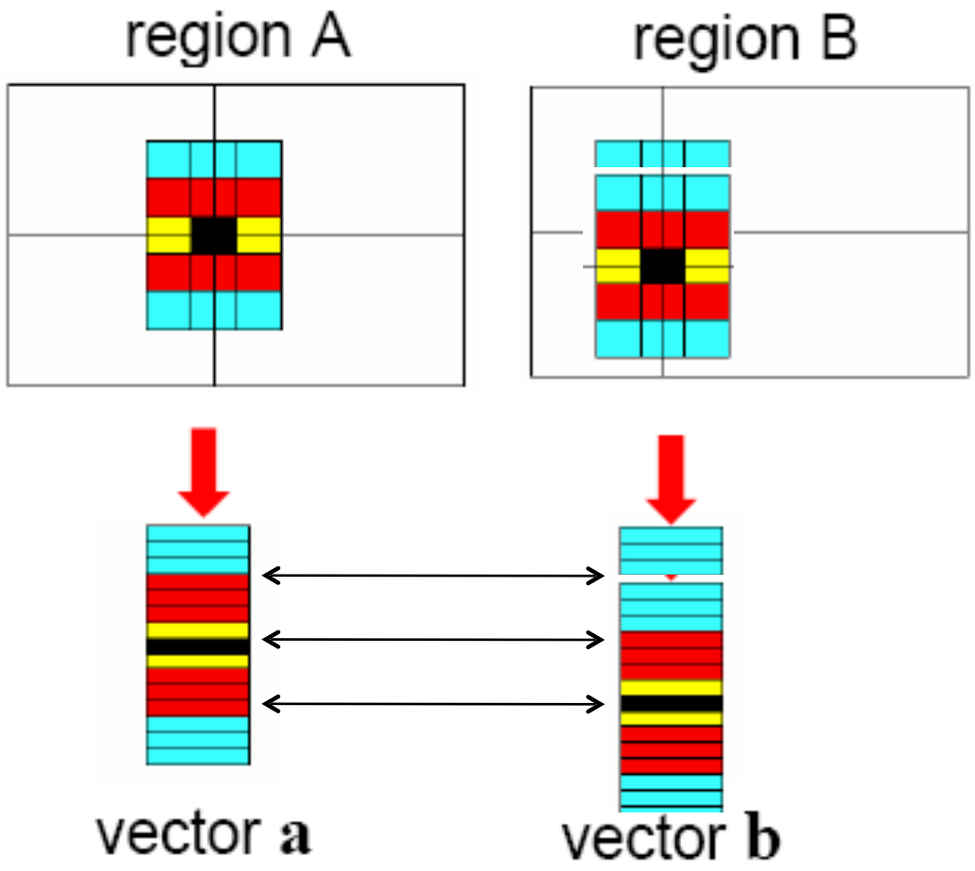


e.g. scale, translation, rotation

Photometric transformations

Figure from T. Tuytelaars ECCV 2006 tutorial

Raw patches as local descriptors

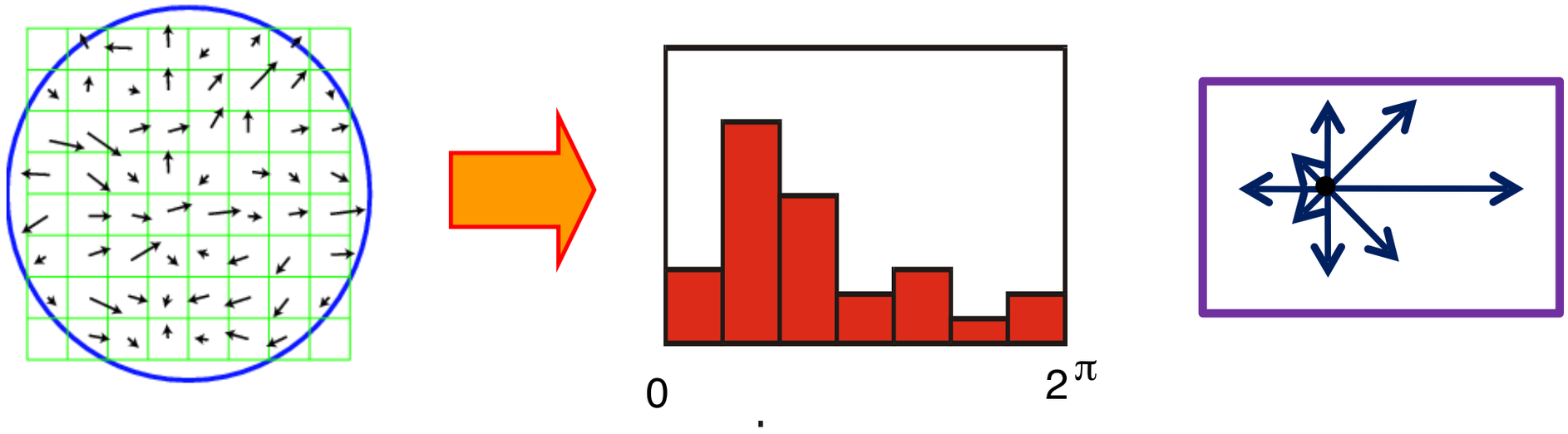


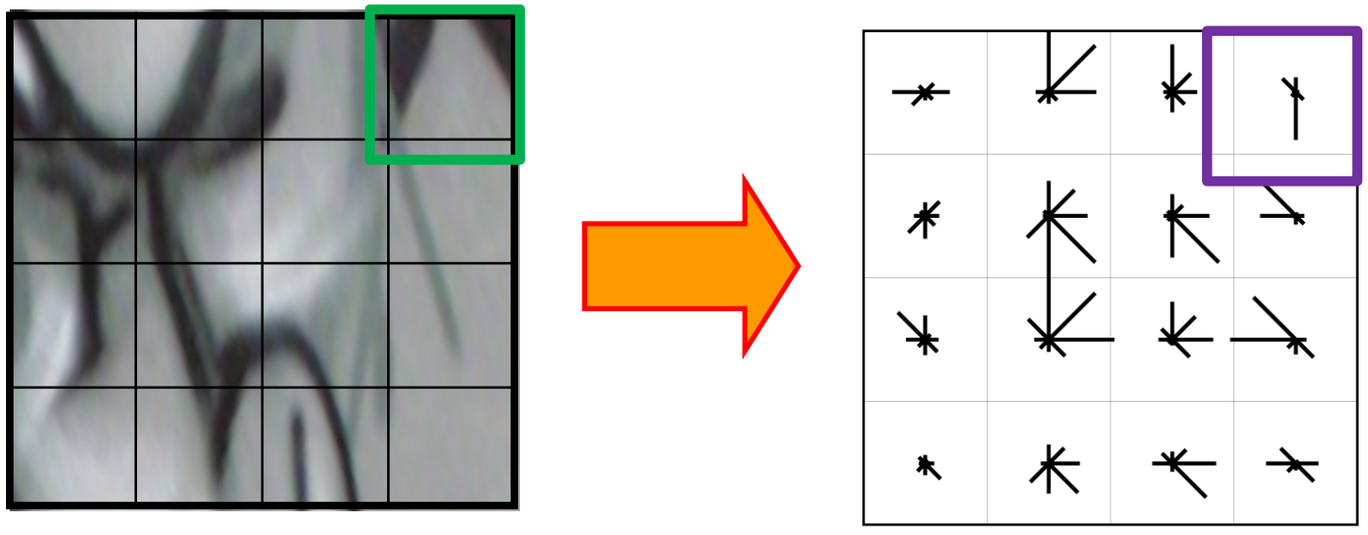
The simplest way to describe the neighborhood around an interest poin is to write down the list of intensities t form a feature vector.

But this is very sensitive to even sma shifts, rotations.

SIFT descriptor [Lowe 2004]

- Use histograms to bin pixels within sub-patches according to their orientation.

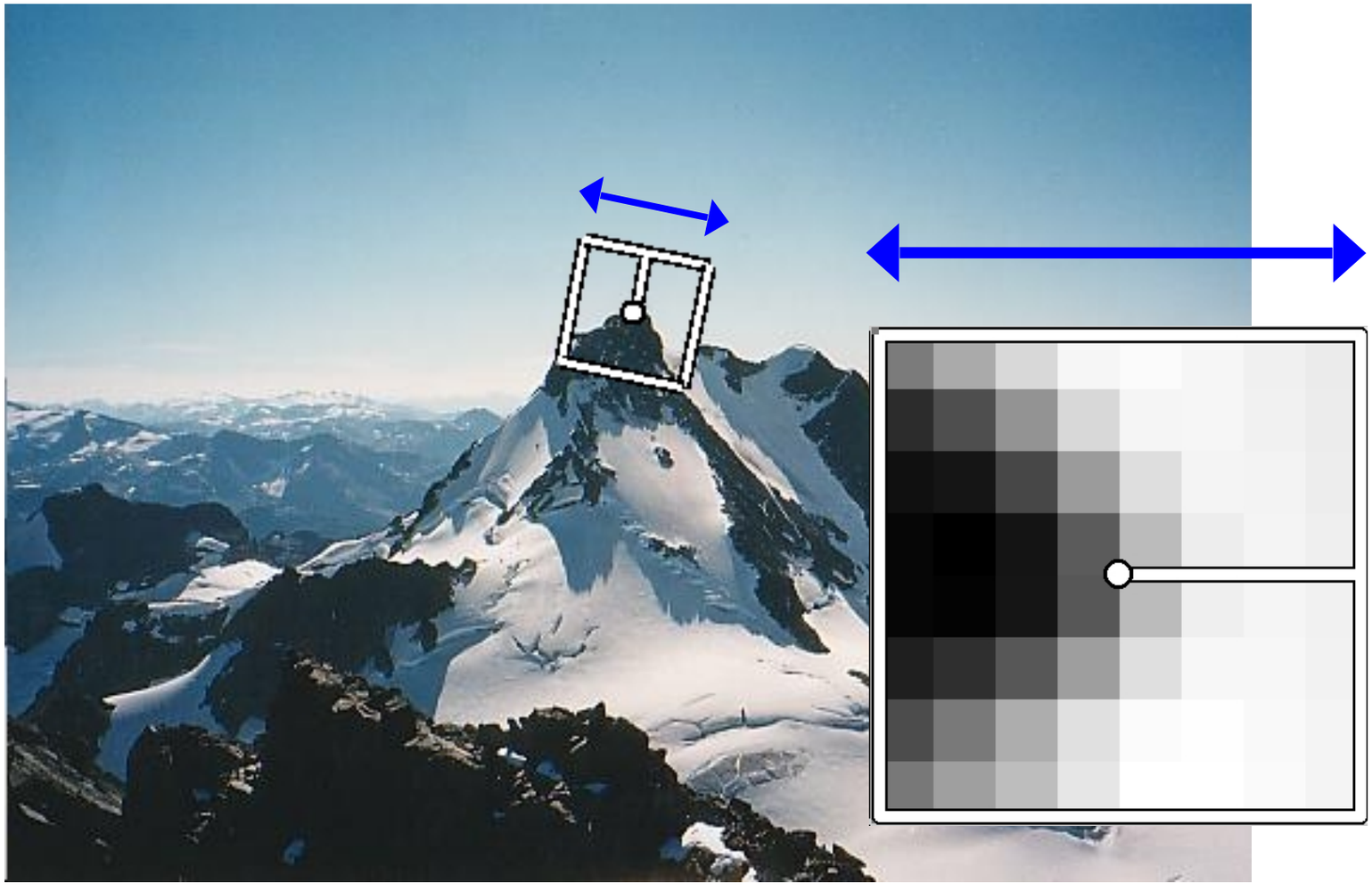




Why subpatches?

Why does SIFT have some illumination invariance?

Making descriptor rotation invariant



- Rotate patch according to its dominant gradient orientation
- This puts the patches into a canonical orientation.

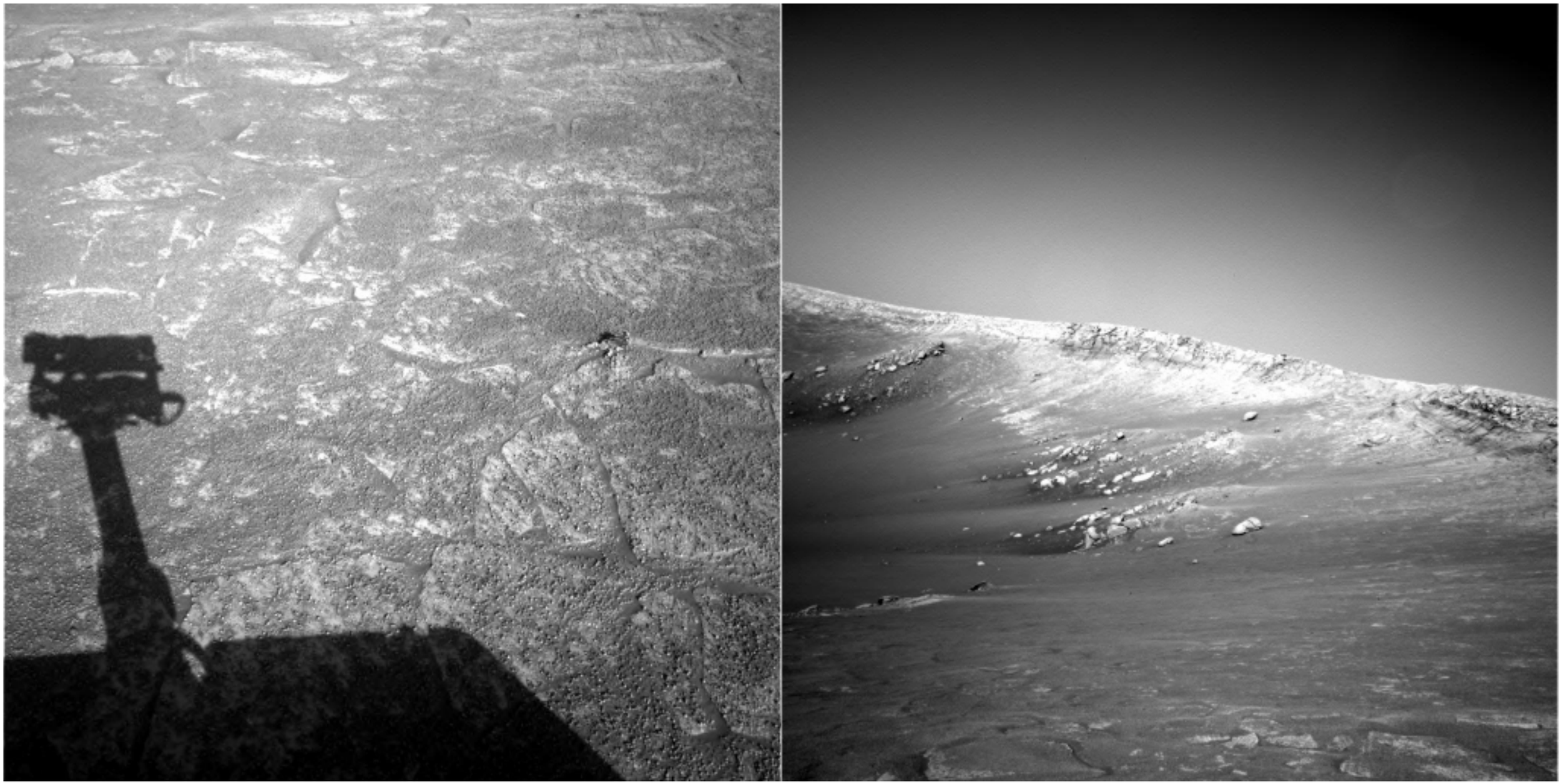
Image from Matthew Brown

SIFT descriptor [Lowe 2004]

- Extraordinarily robust matching technique
  - Can handle changes in viewpoint
    - Up to about 60 degree out of plane rotation
  - Can handle significant changes in illumination
    - Sometimes even day vs. night (below)
  - Fast and efficient—can run in real time
  - Lots of code available
    - http://people.csail.mit.edu/albert/ladypack/wiki/index.php/Known_implementations_of_SIFT

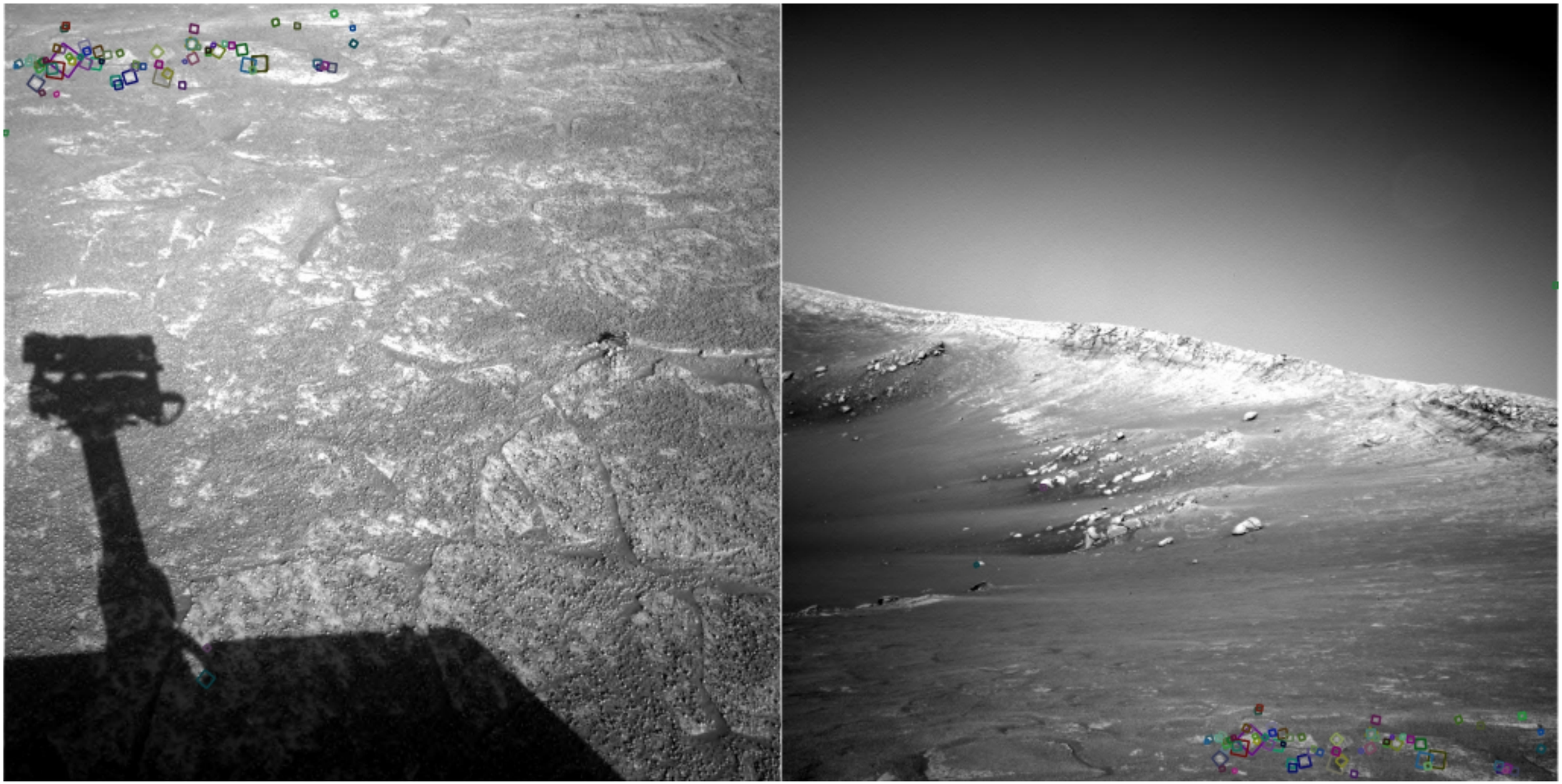
Steve Seitz

Example



NASA Mars Rover images

Example



NASA Mars Rover images
with SIFT feature matches
Figure by Noah Snavely

SIFT properties

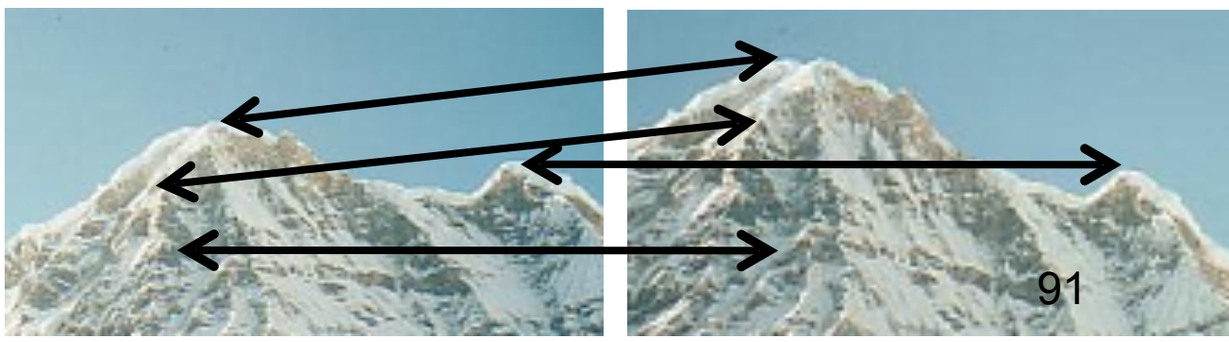
- Invariant to
  - Scale
  - Rotation
- Partially invariant to
  - Illumination changes
  - Camera viewpoint
  - Occlusion, clutter

Local features: main components

1) Detection: Identify the interest points

2) Description:Extract vector feature descriptor surrounding each interest point.

3) Matching: Determine correspondence between descriptors in two views



Matching local features

Matching local features

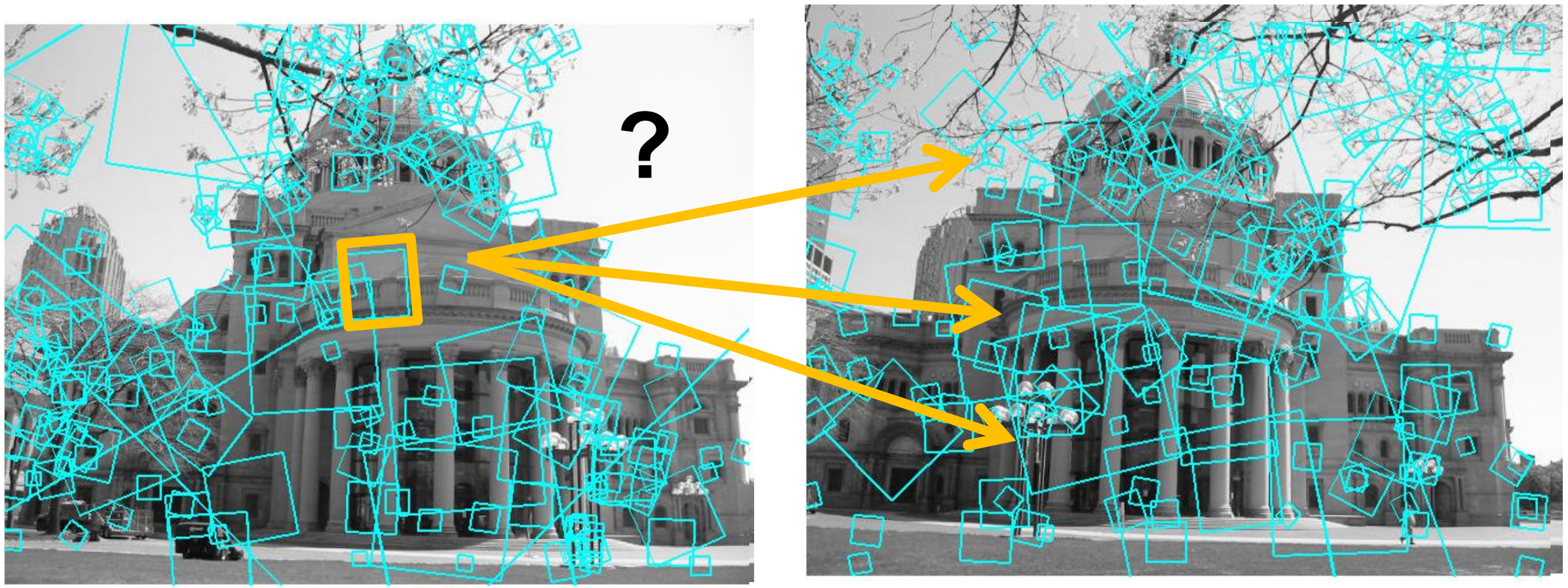


Image 1 Image 2

To generate **candidate matches**, find patches that have the most similar appearance (e.g., lowest SSD)

Simplest approach: compare them all, take the closest (or closest k, or within a thresholded distance)

Ambiguous matches

Image 1

Image 2

At what SSD value do we have a good match?

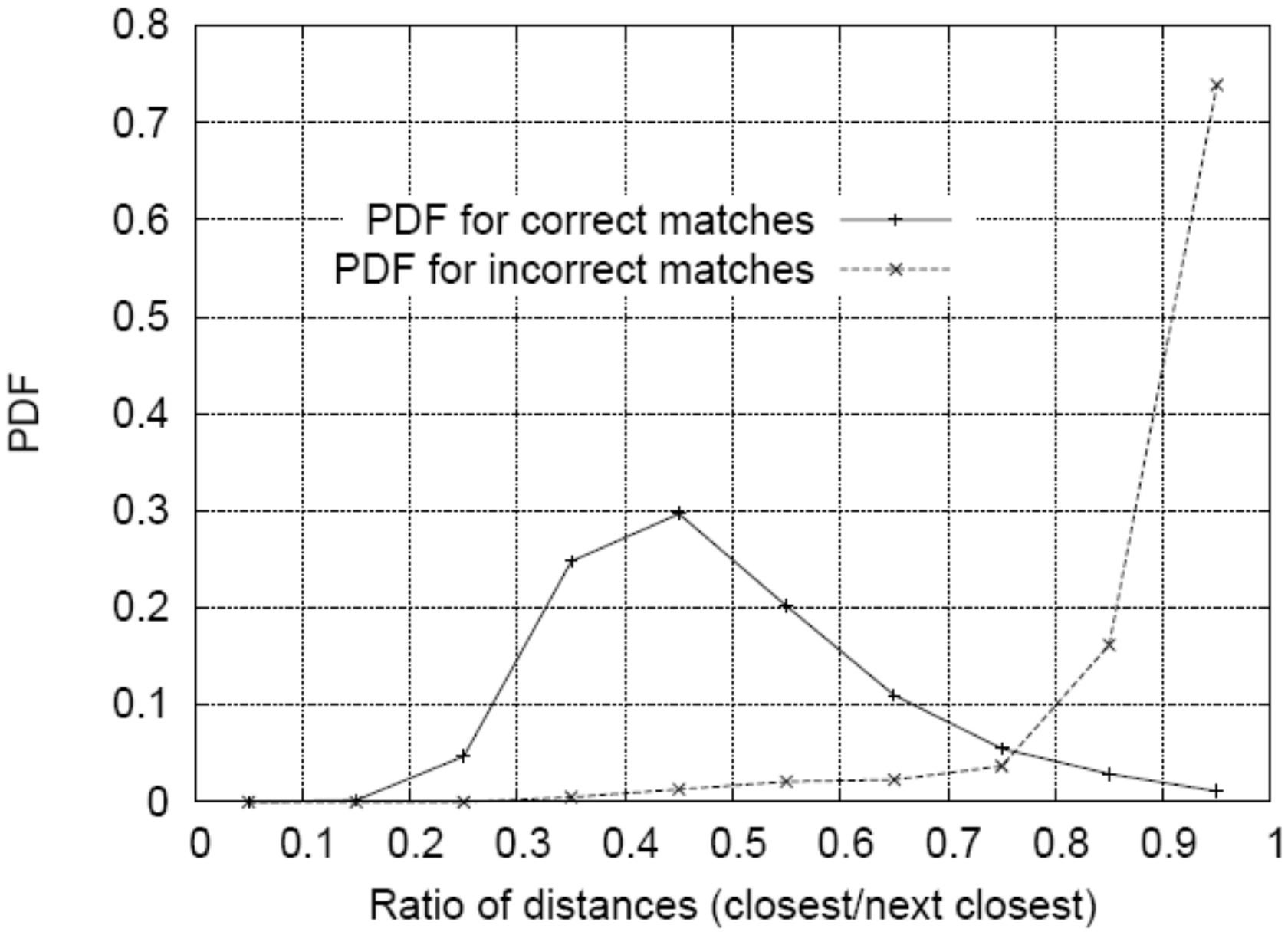
To add robustness to matching, can consider **ratio** : distance to best match / distance to second best match

If low, first match looks good.

If high, could be ambiguous match.

Matching SIFT Descriptors

- Nearest neighbor (Euclidean distance)
- Threshold ratio of nearest to 2nd nearest descriptor

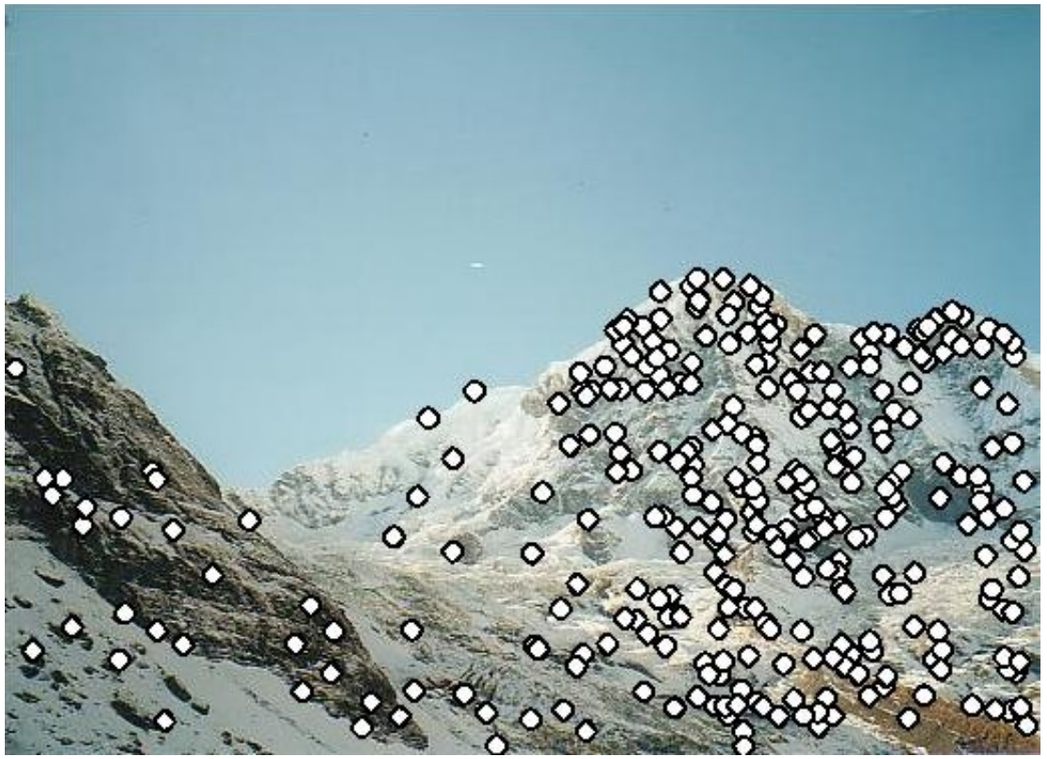


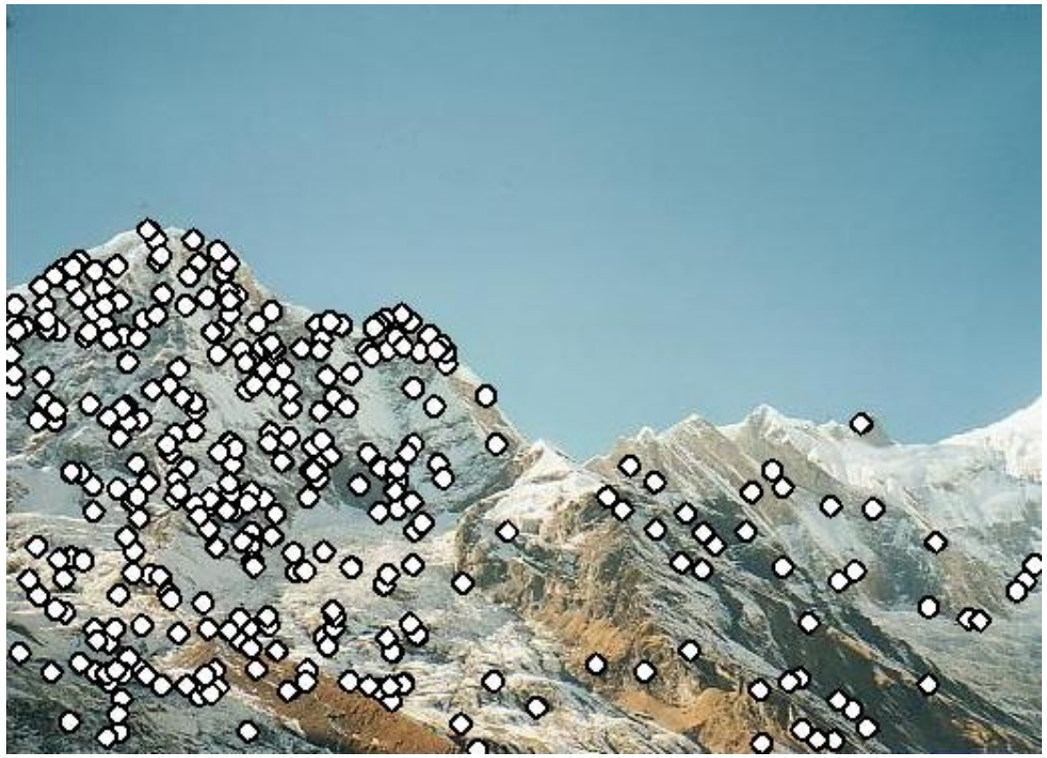
Lowe IJCV 2004

Recap: robust feature-based alignment

Source: L. Lazebnik

Recap: robust feature-based alignment

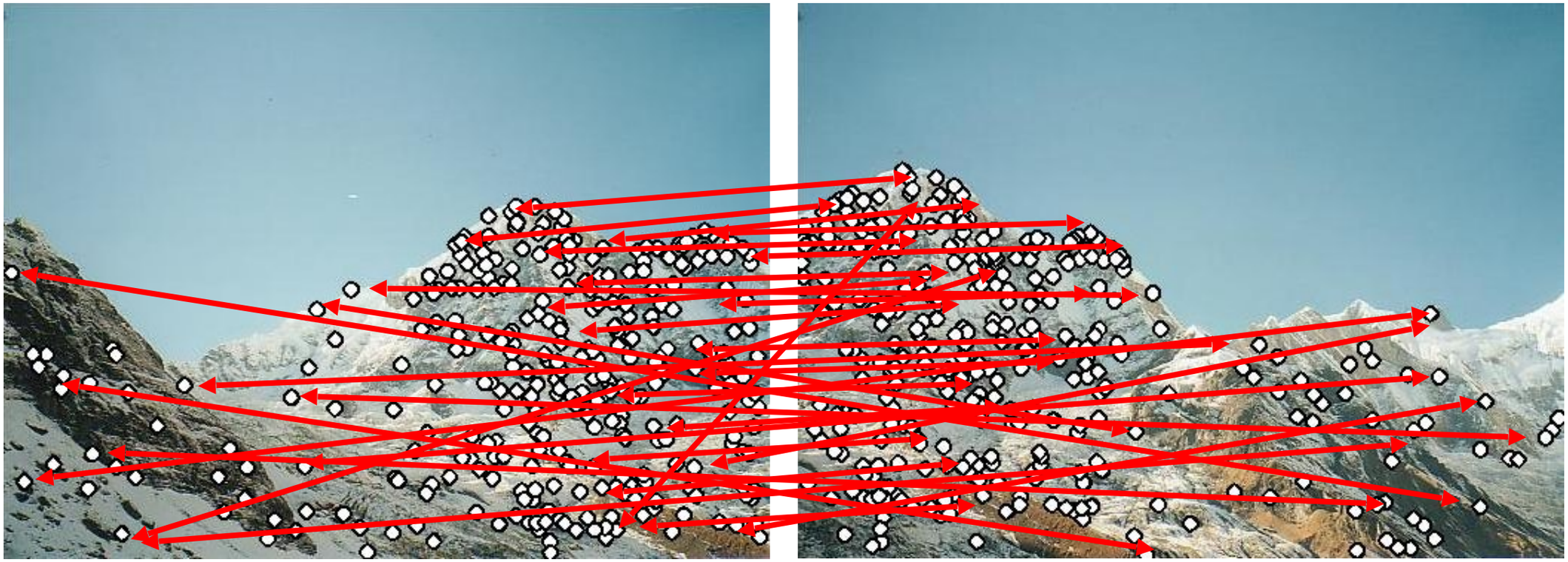




- Extract features

Source: L. Lazebnik

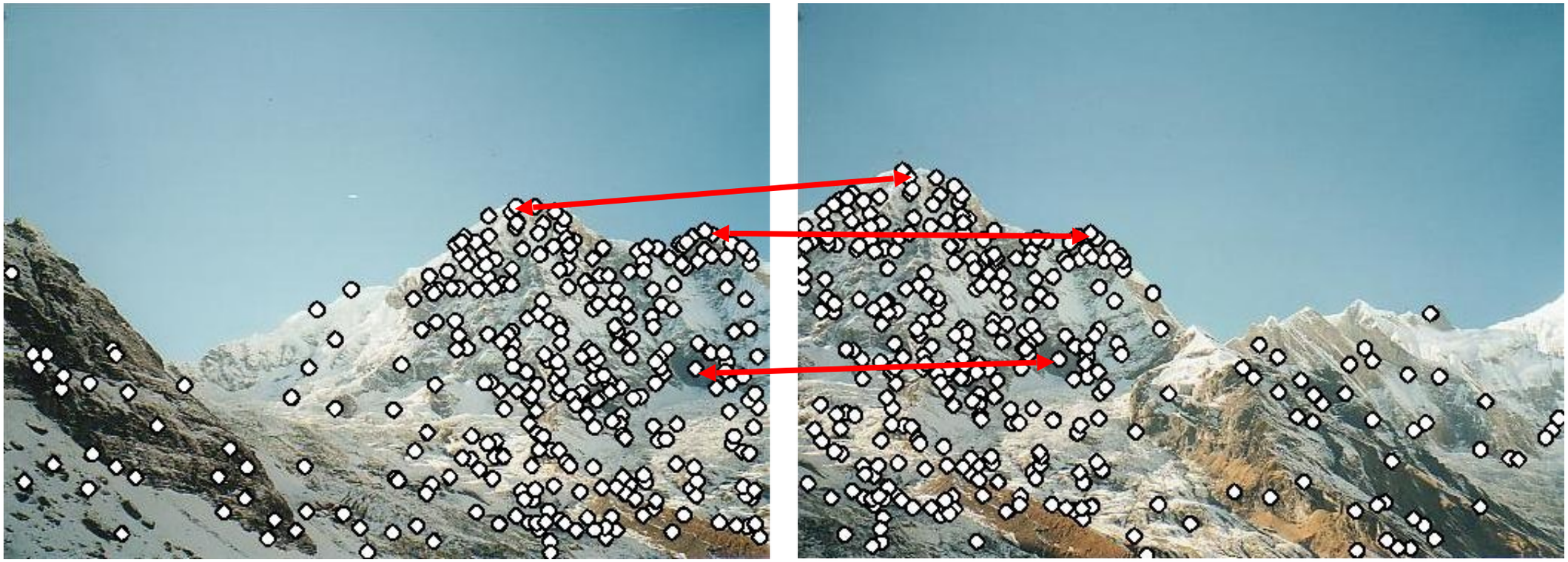
Recap: robust feature-based alignment



- Extract features
- Compute *putative matches*

Source: L. Lazebnik

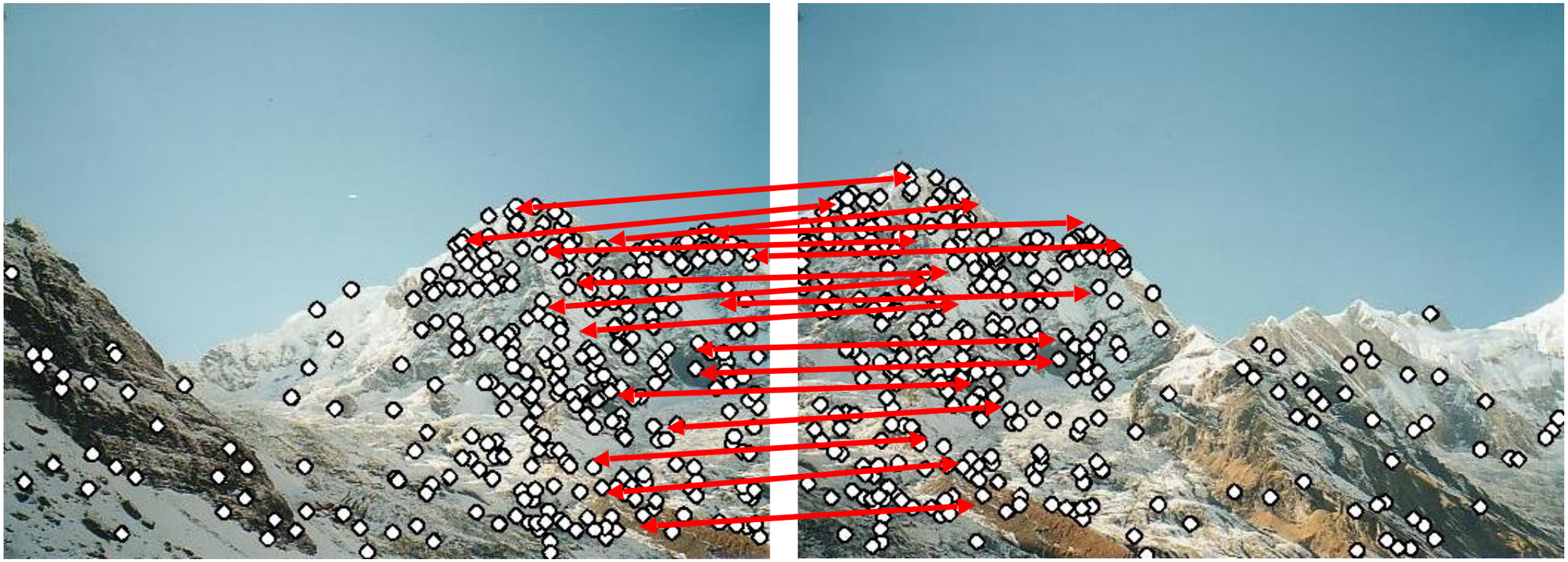
Recap: robust feature-based alignment



- Extract features
- Compute *putative matches*
- Loop:
  - *Hypothesize* transformation *T* (small group of putative matches that are related by *T*)

Source: L. Lazebnik

Recap: robust feature-based alignment



- Extract features
- Compute *putative matches*
- Loop:
  - *Hypothesize* transformation *T* (small group of putative matches that are related by *T*)
  - *Verify* transformation (search for other matches consistent with *T*)

Source: L. Lazebnik

Recap: robust feature-based alignment

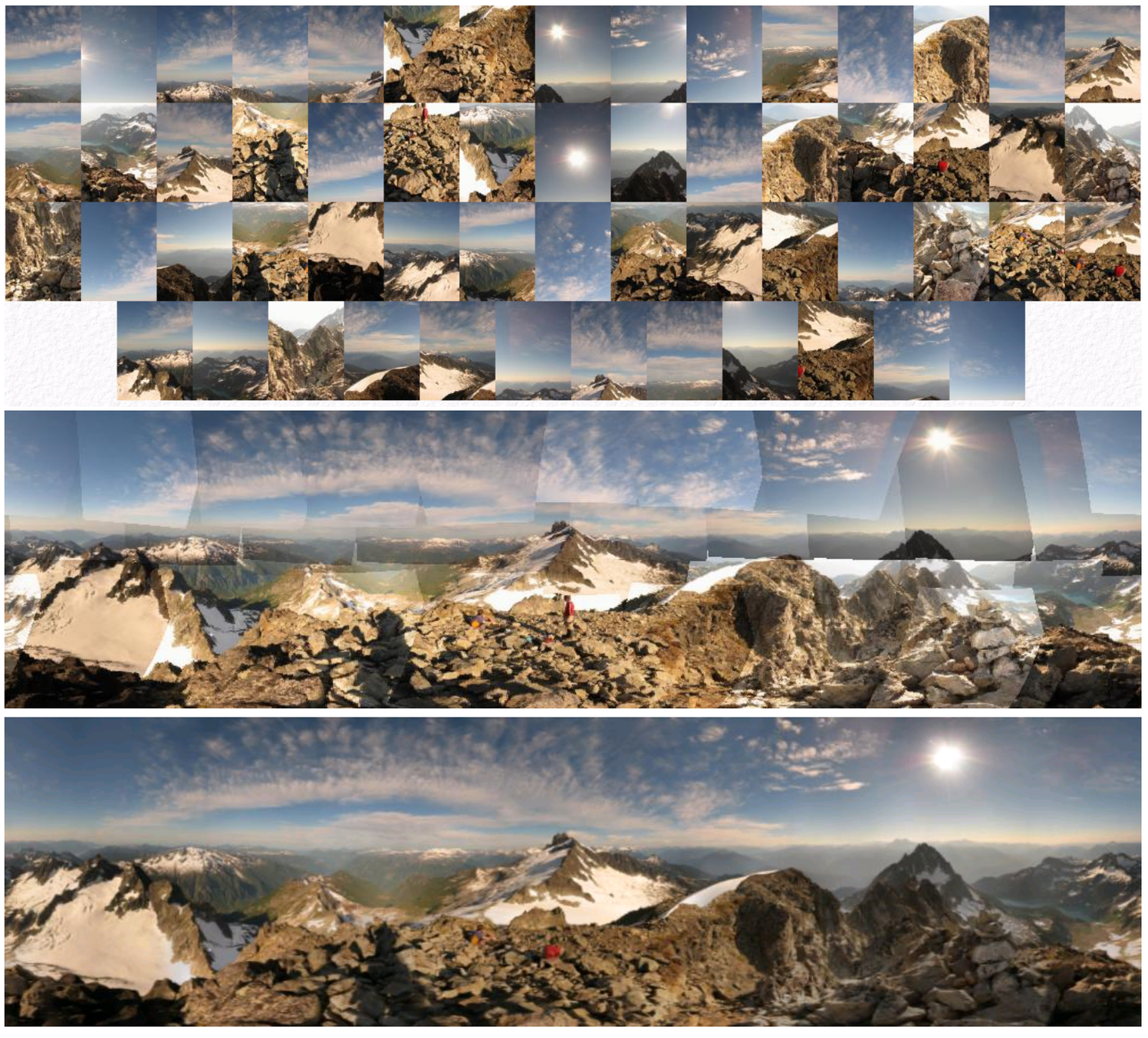
- Extract features
- Compute *putative matches*
- Loop:
  - *Hypothesize* transformation *T* (small group of putative matches that are related by *T*)
  - *Verify* transformation (search for other matches consistent with *T*)

Source: L. Lazebnik

Applications of local invariant features

- Wide baseline stereo
- Motion tracking
- Panoramas
- Mobile robot navigation
- 3D reconstruction
- Recognition
- …

Automatic mosaicing

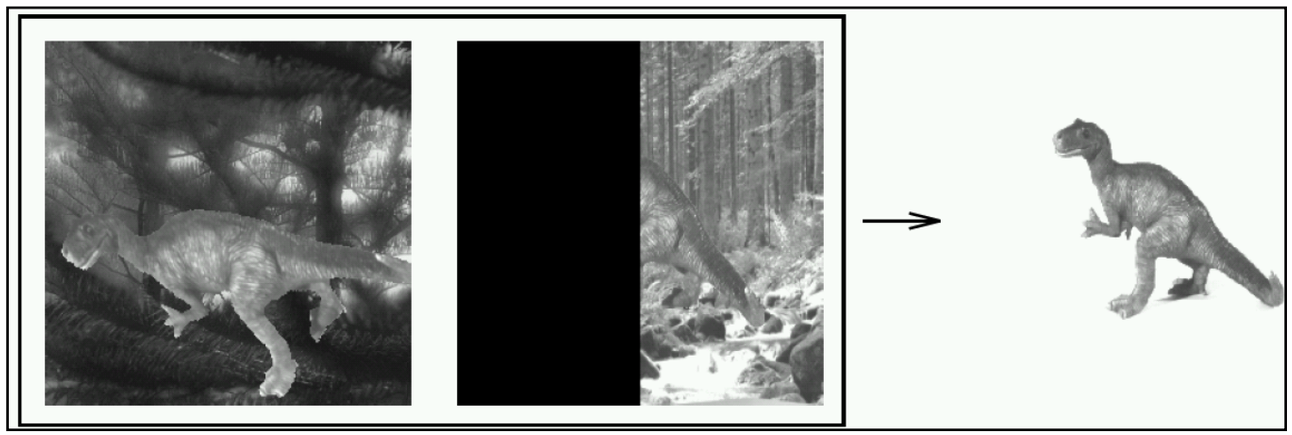


http://www.cs.ubc.ca/~mbrown/autostitch/autostitch.html

Wide baseline stereo

[Image from T. Tuytelaars ECCV 2006 tutorial]

Recognition of specific objects, scenes



Schmid and Mohr 1997

Sivic and Zisserman, 2003

Rothganger et al. 2003

Lowe 2002

Summary

- Interest point detection
  - Harris corner detector
  - Laplacian of Gaussian, automatic scale selection

- Invariant descriptors
  - Rotation according to dominant gradient direction
  - Histograms for robustness to small shifts and translations (SIFT descriptor)

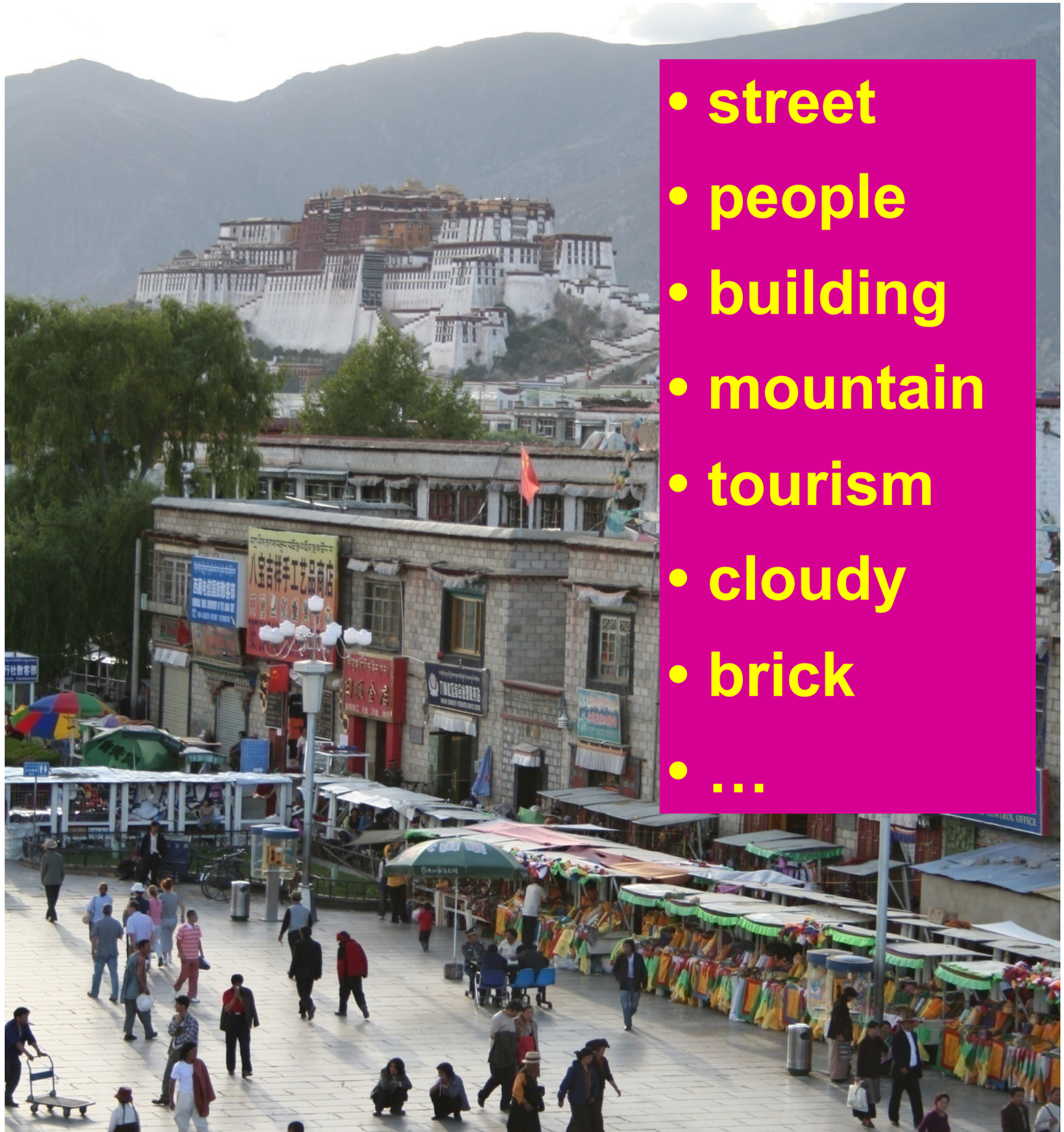
Local features

Specific recognition tasks

Scene categorization or classification

- **outdoor/indoor**
- **city/forest/factory/etc.**

Image annotation / tagging / attributes



Object detection

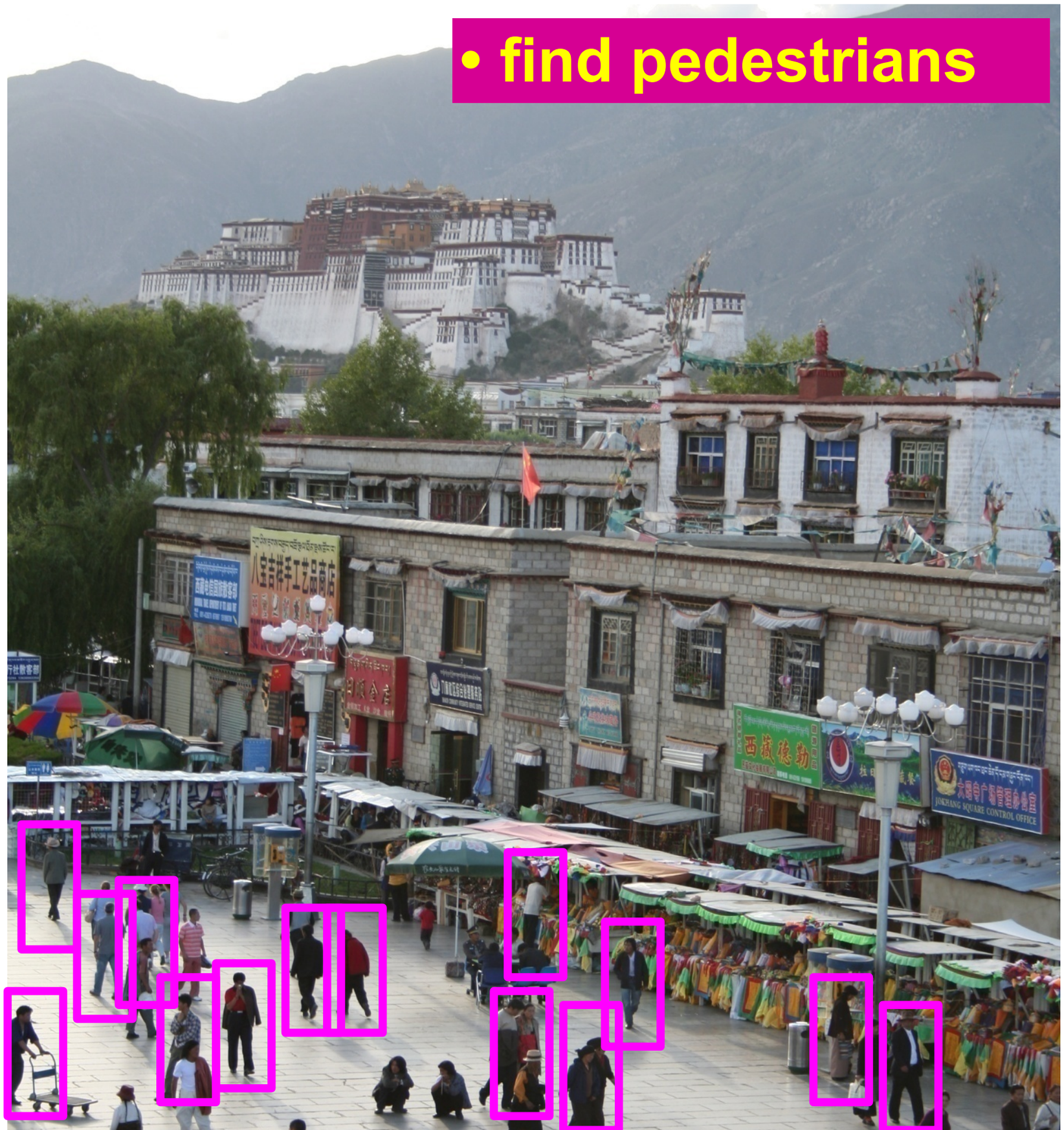


Image parsing

Local features and bag of words models

- Representation
  - Gist descriptor
  - Image histograms
  - Sift-like features
- Bag of Words models
  - Encoding methods

Image Categorization

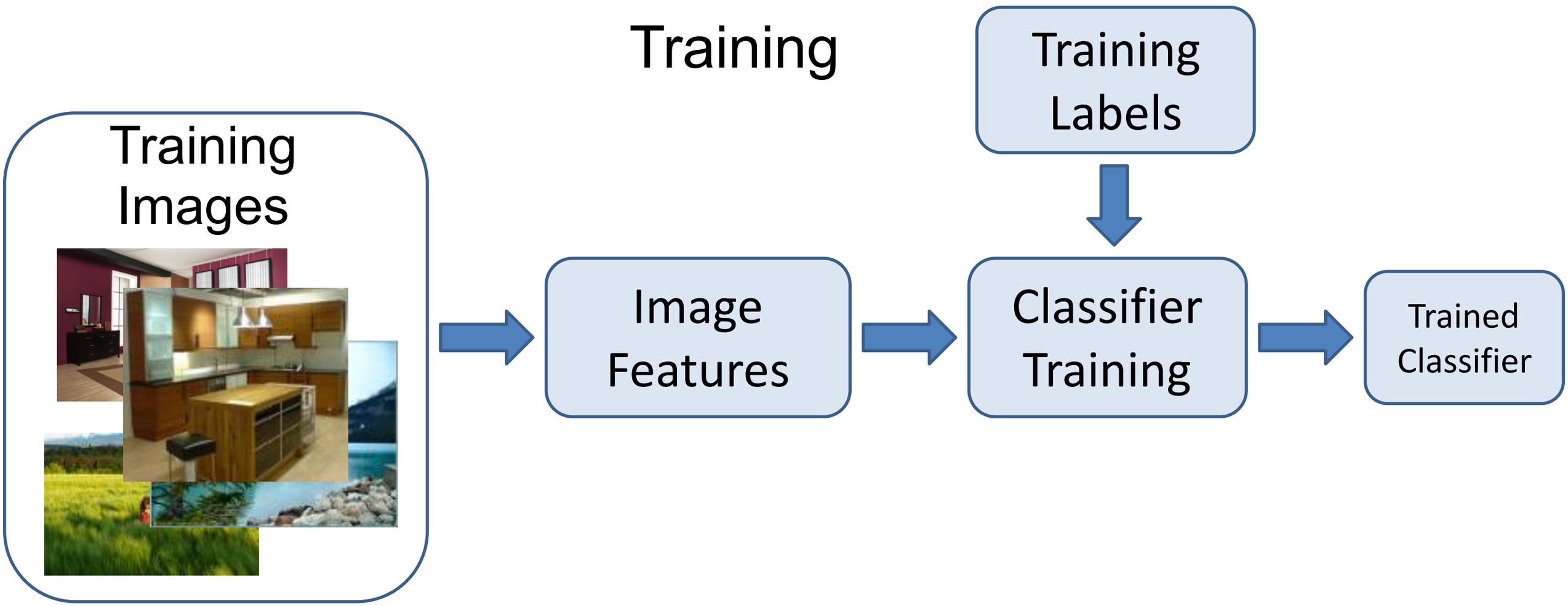


Image Categorization

Training

Training Images

Training Labels

Image Features

Classifier Training

Trained Classifier

Testing

Test Image

Image Features

Trained Classifier

Prediction

Outdoor

Part 1: Image features

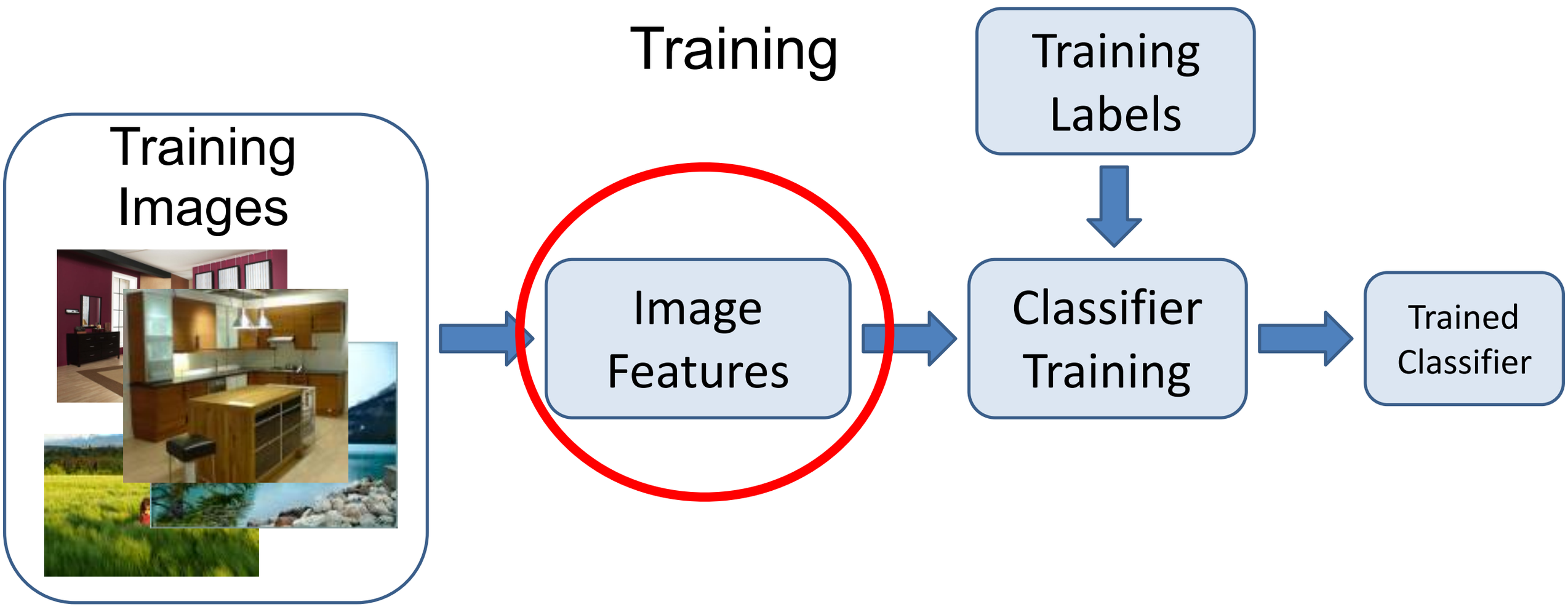
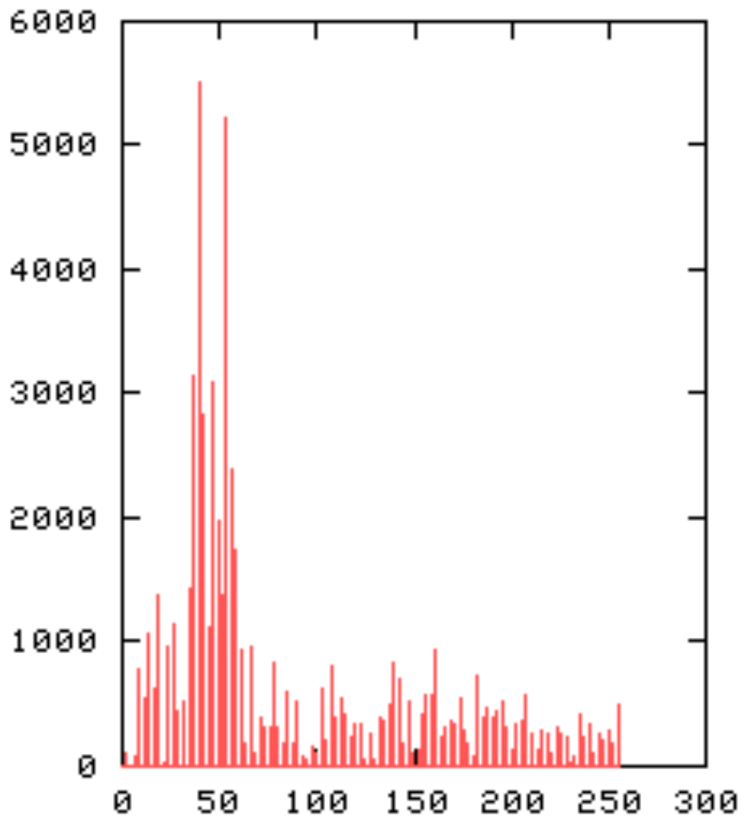
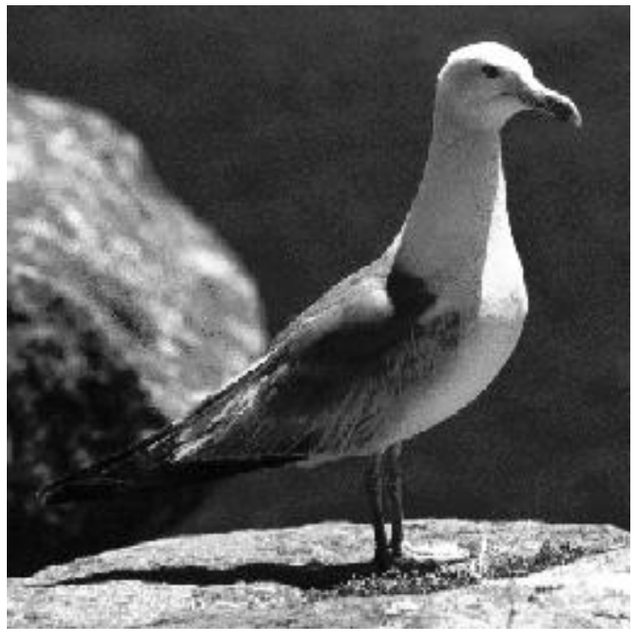


Image Representations: Histograms





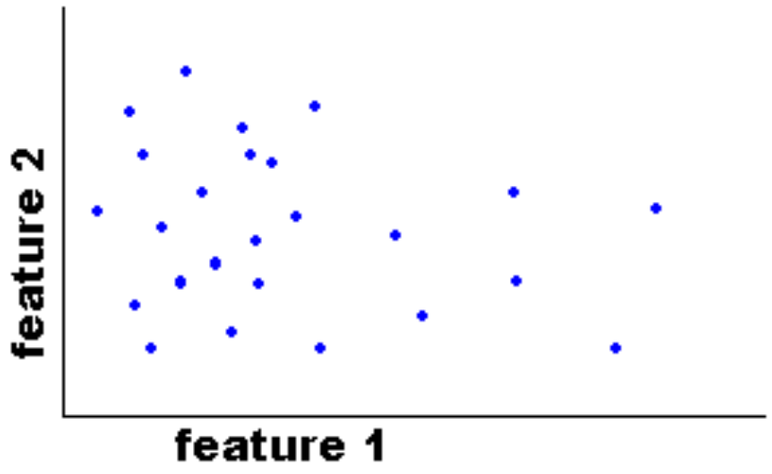
Global histogram

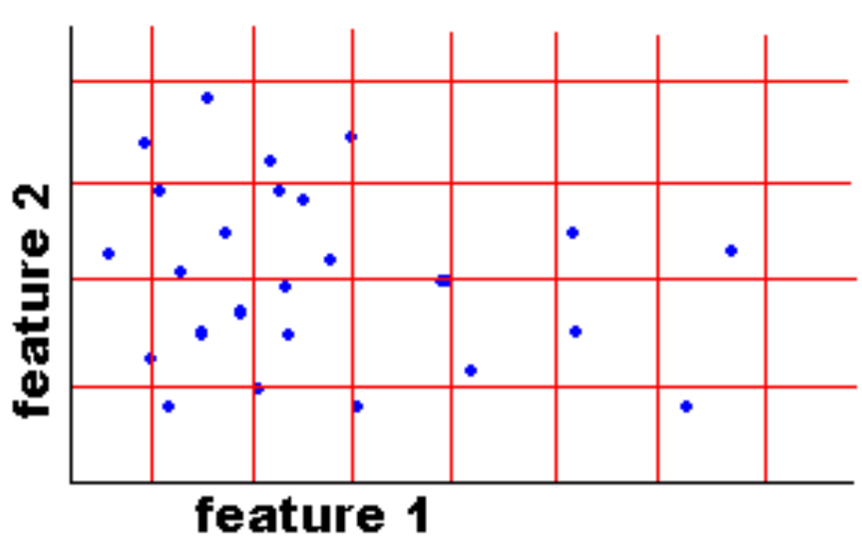
- Represent distribution of features
  - Color, texture, depth, …

Images from Dave Kauchak

Image Representations: Histograms

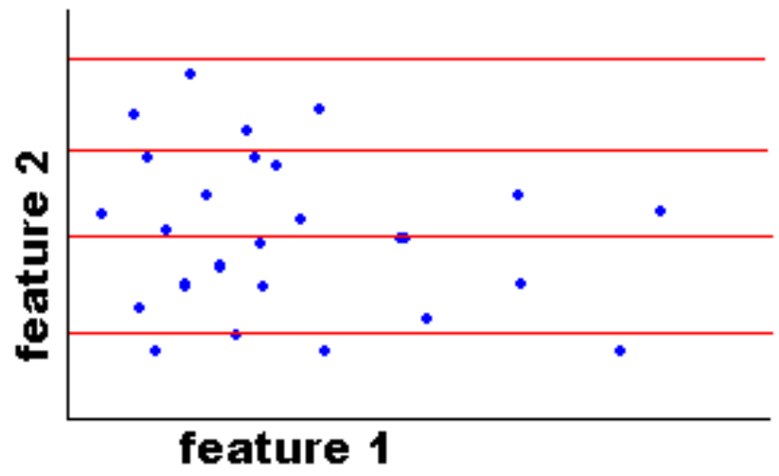
Histogram: Probability or count of data in each bin

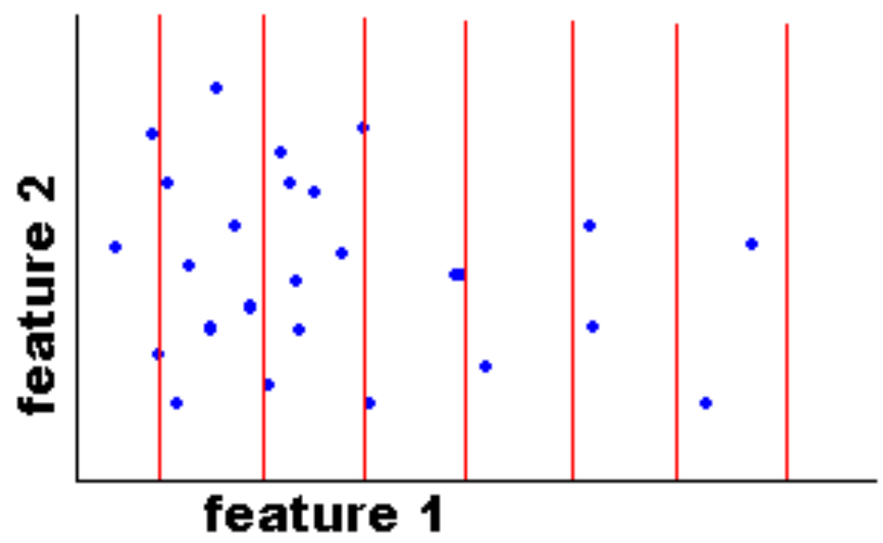




Joint histogram

- Requires lots of data
- Loss of resolution to avoid empty bins





Marginal histogram

- Requires independent features
- More data/bin than joint histogram

Images from Dave Kauchak

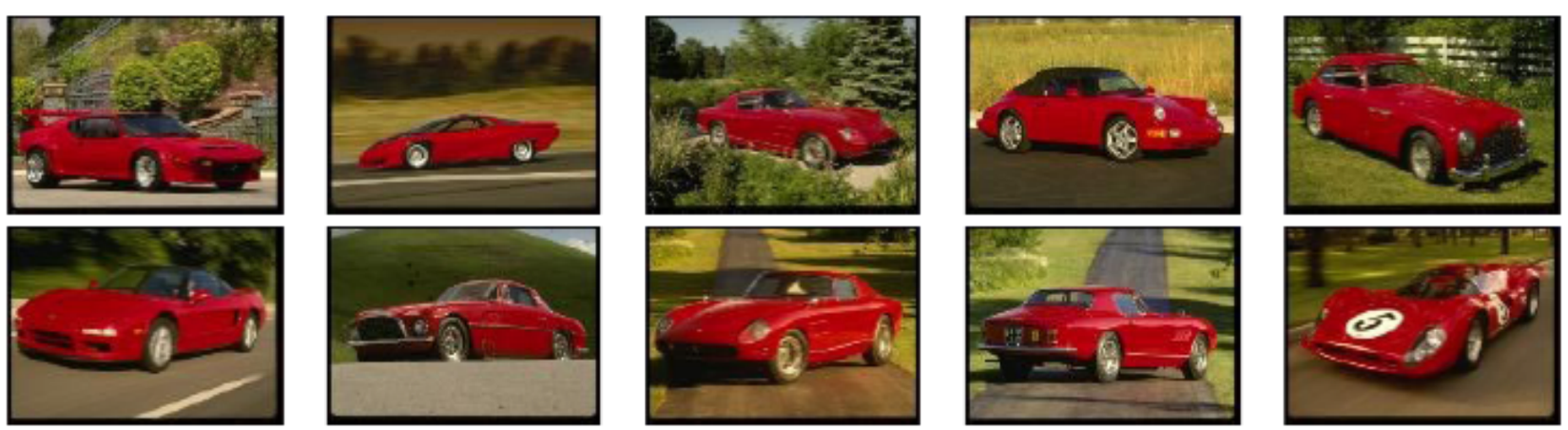
Computing histogram distance

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

Histogram intersection (assuming normalized histograms)

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

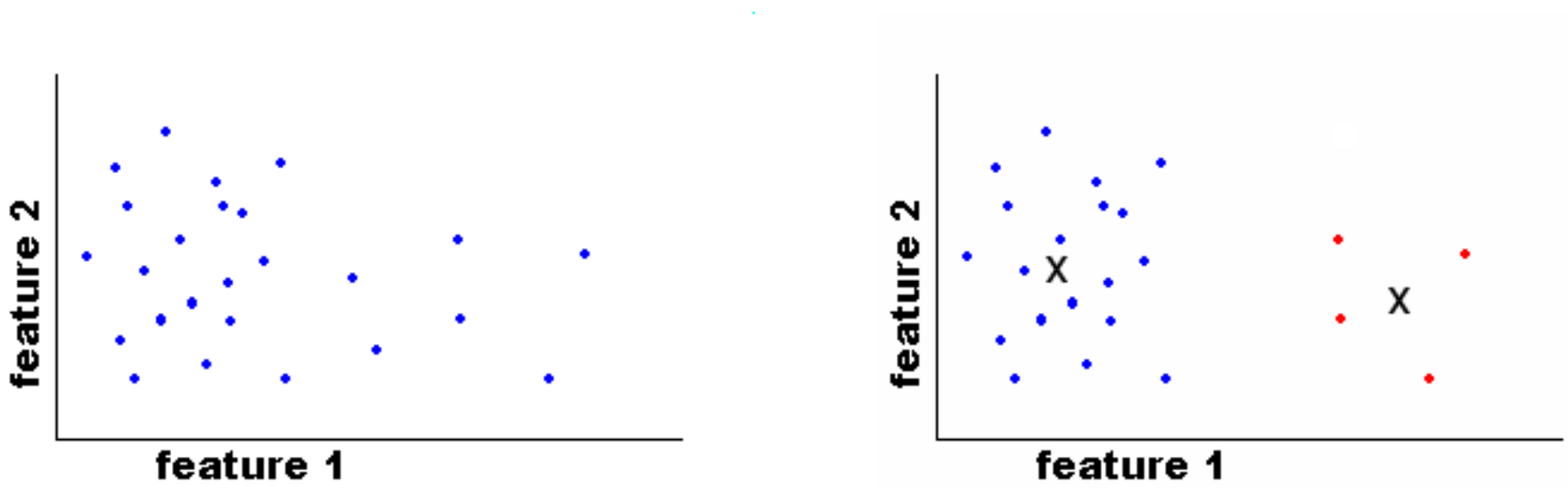
Chi-squared Histogram matching distance



Cars found by color histogram matching using chi-squared

Image Representations: Histograms

Clustering



Use the same cluster centers for all images

Images from Dave Kauchak

Histograms: Implementation issues

- Quantization
  - Grids: fast but applicable only with few dimensions
  - Clustering: slower but can quantize data in higher dimensions

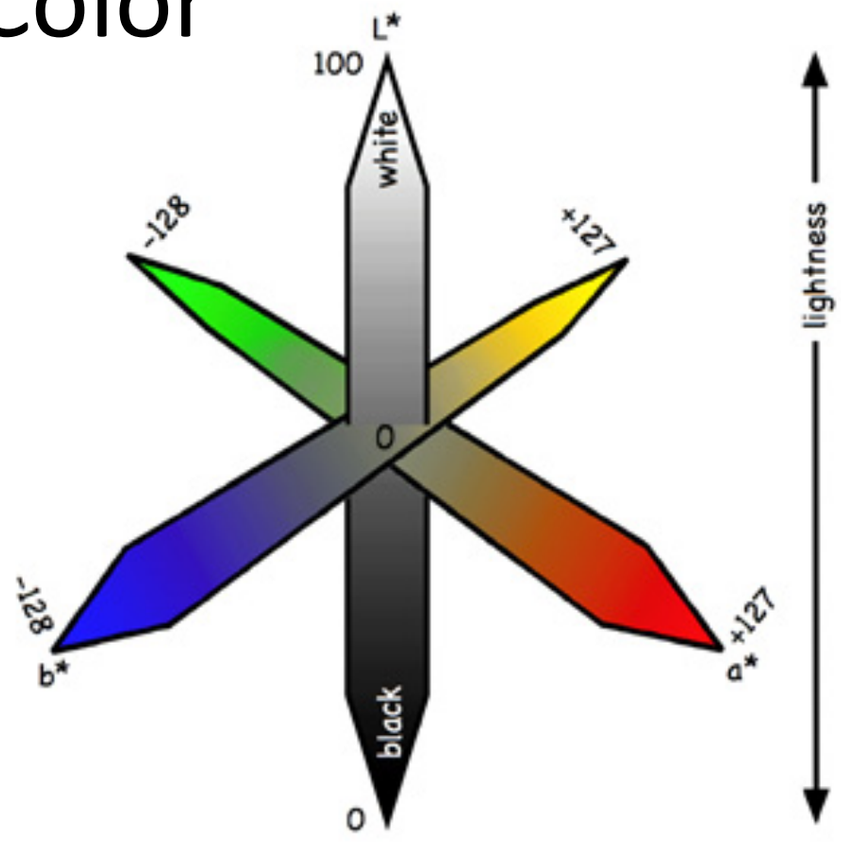
Few Bins
Need less data
Coarser representation

Many Bins
Need more data
Finer representation

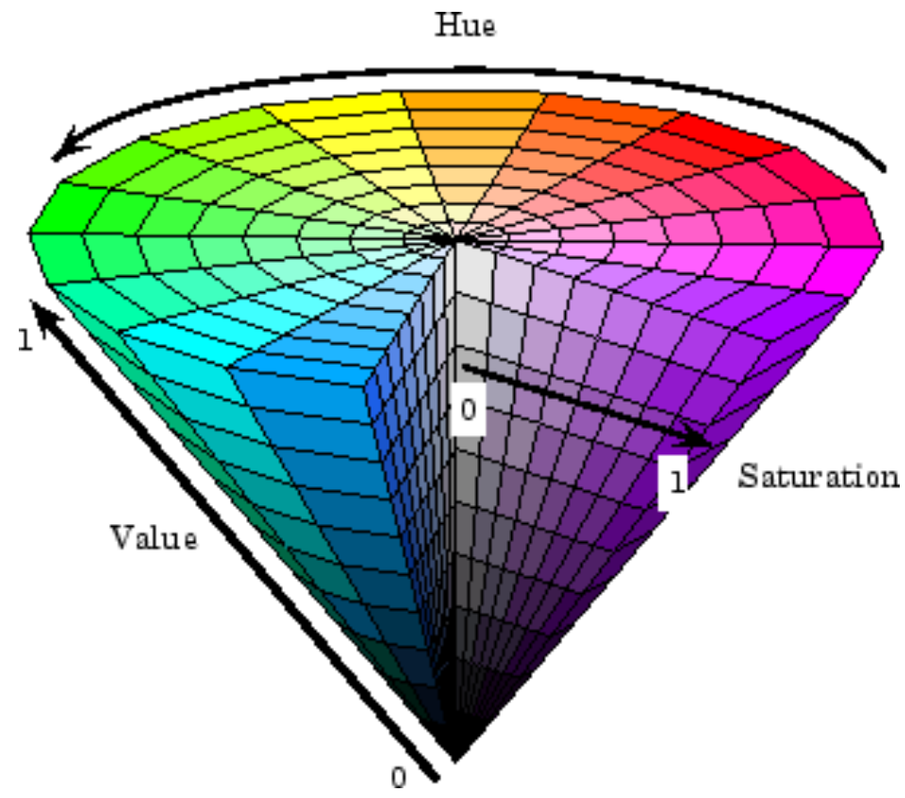
- Matching
  - Histogram intersection or Euclidean may be faster
  - Chi-squared often works better
  - Earth mover's distance is good for when nearby bins represent similar values

What kind of things do we compute histograms of?

- Color



L*a*b* color space

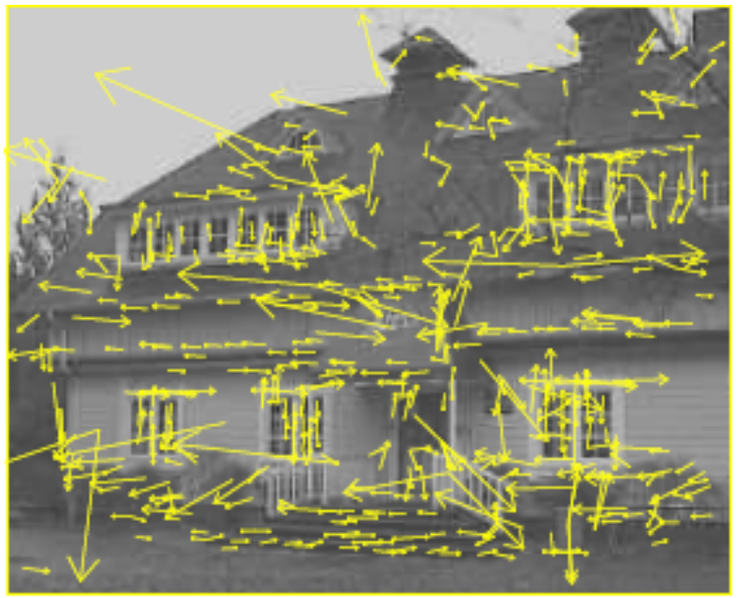


HSV color space

- Texture (filter banks or HOG over regions)

What kind of things do we compute histograms of?

- Histograms of oriented gradients



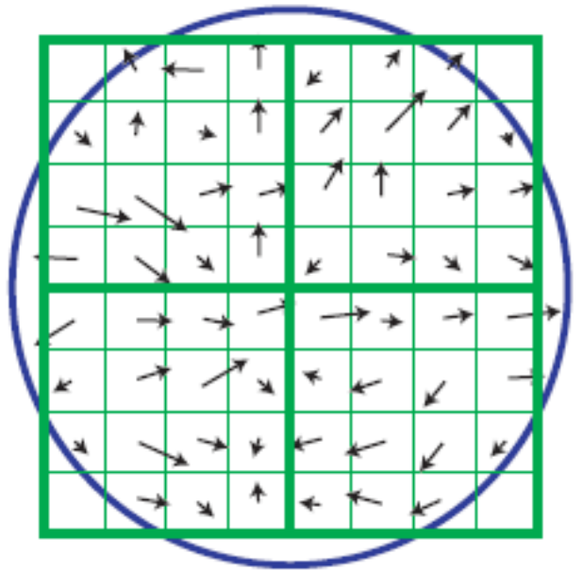
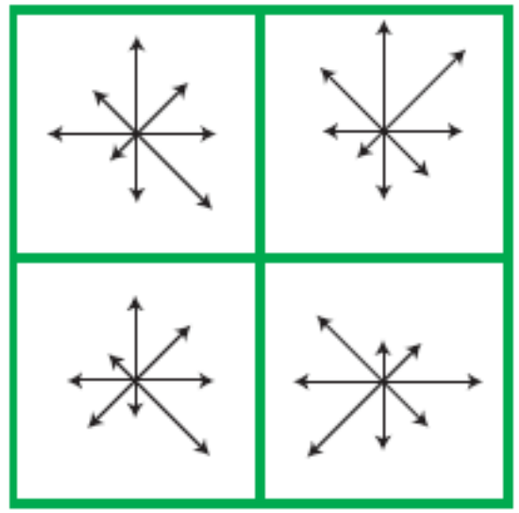


Image gradients

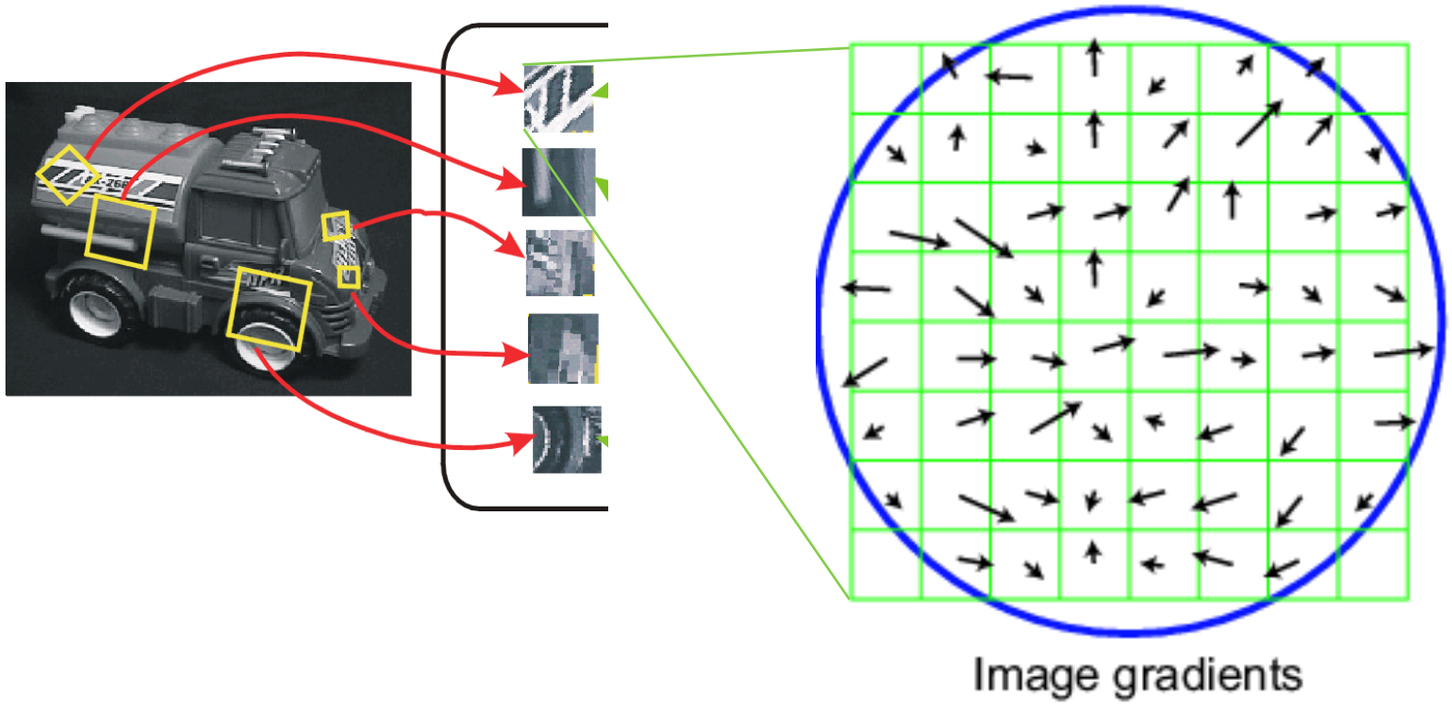


Keypoint descriptor

SIFT – Lowe IJCV 2004

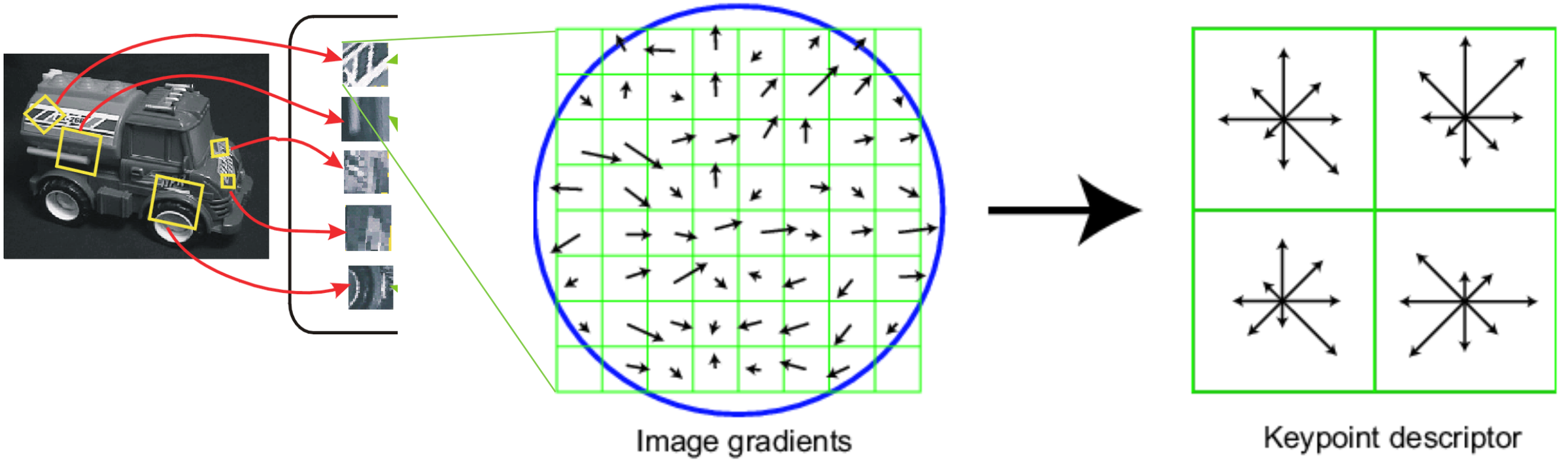
SIFT vector formation

- Computed on rotated and scaled version of window according to computed orientation & scale
  - resample the window
- Based on gradients weighted by a Gaussian of variance half the window (for smooth falloff)



SIFT vector formation

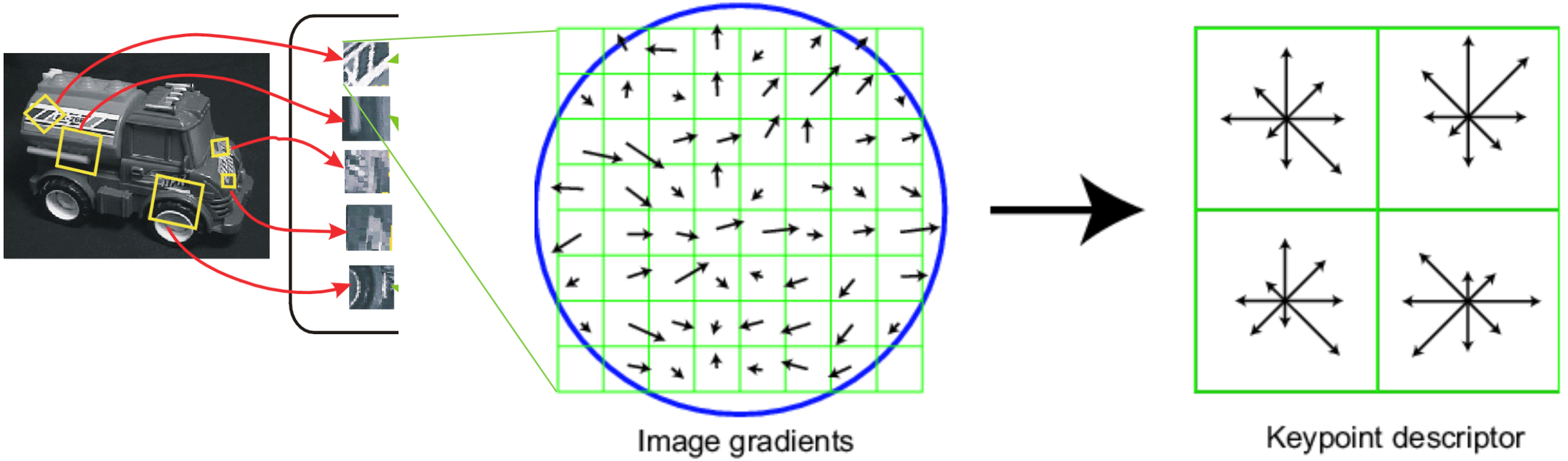
- 4x4 array of gradient orientation histograms
  – not really histogram, weighted by magnitude
- 8 orientations x 4x4 array = 128 dimensions
- Motivation: some sensitivity to spatial layout, but not too much.



showing only 2x2 here but is 4x4

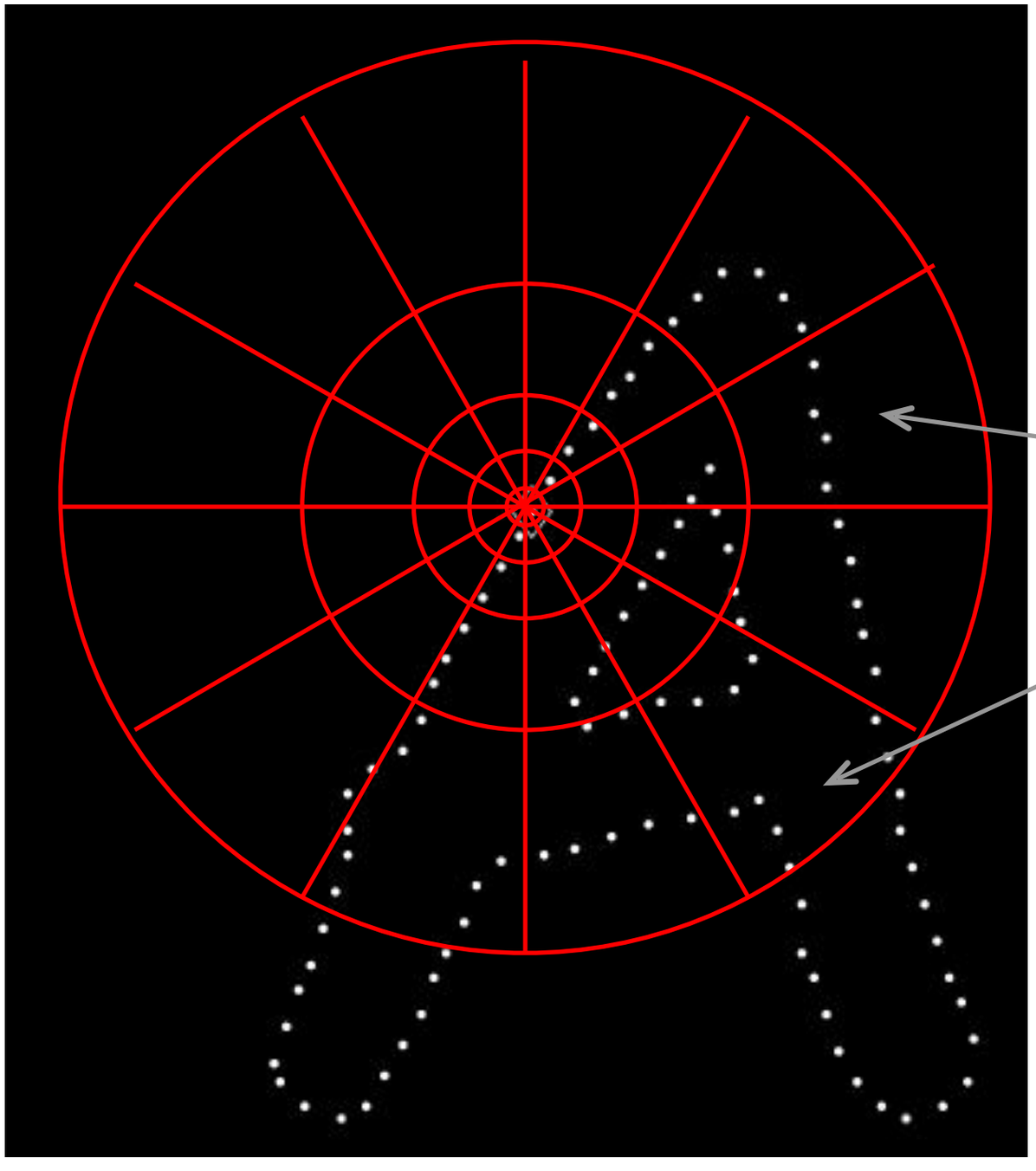
Reduce effect of illumination

- 128-dim vector normalized to 1
- Threshold gradient magnitudes to avoid excessive influence of high gradients
  - after normalization, clamp gradients >0.2
  - renormalize



James Hayes

Local Descriptors: Shape Context



Count the number of points inside each bin, e.g.:

Count = 4

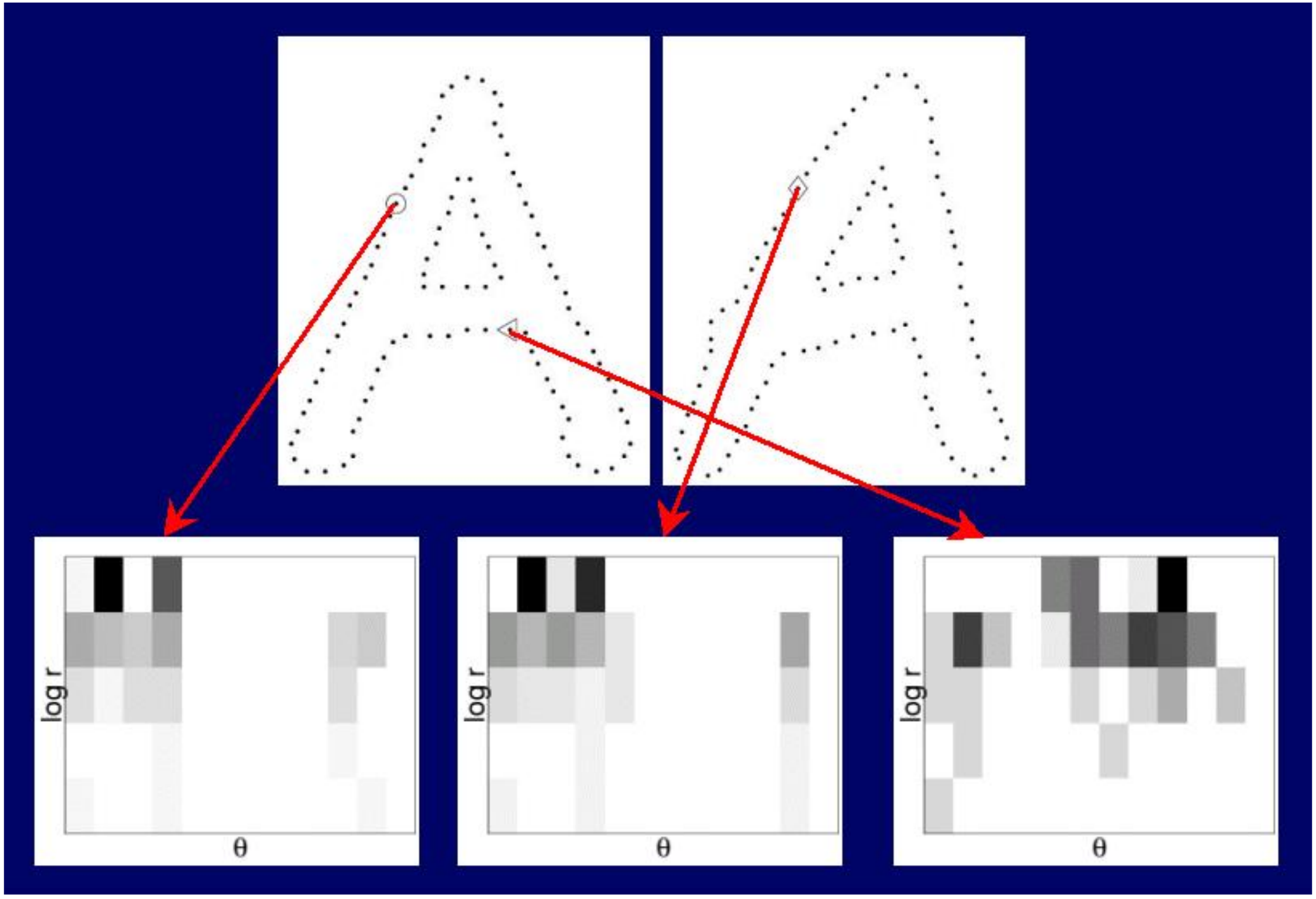
⋮

Count = 10

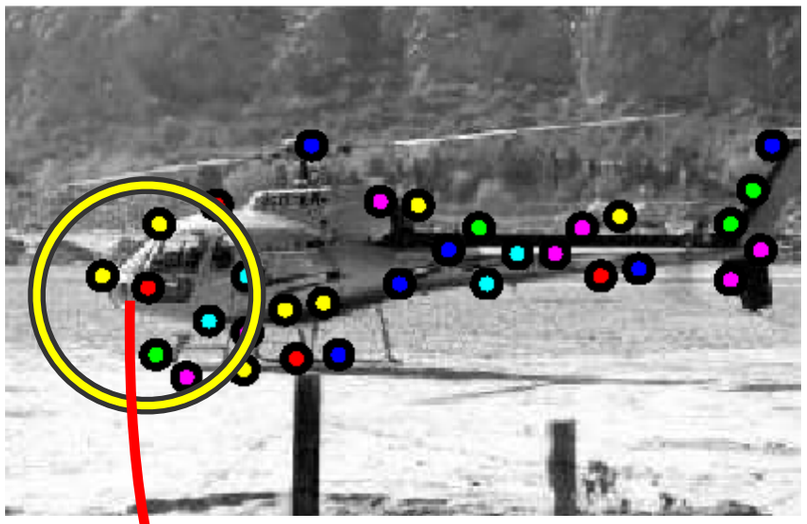
Log-polar binning: more precision for nearby points, more flexibility for farther points.

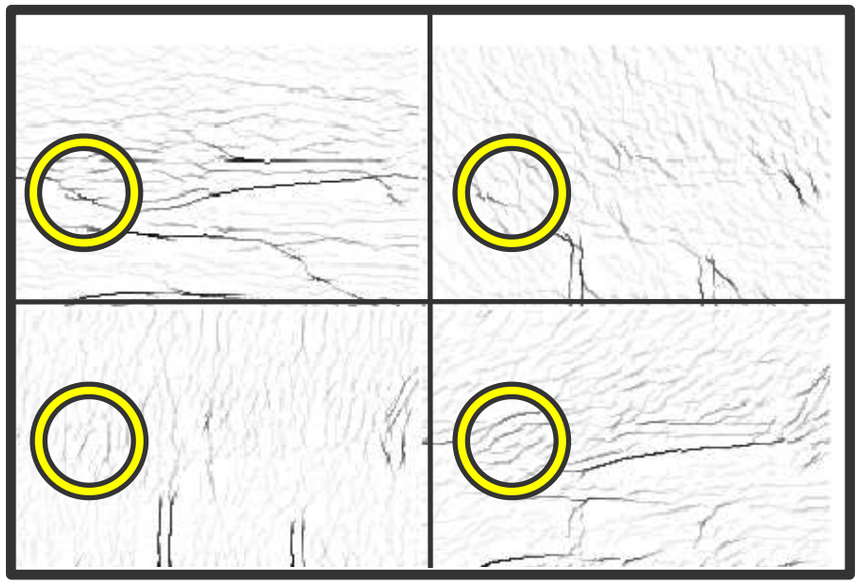
Belongie & Malik, ICCV 2001

Shape Context Descriptor



Local Descriptors: Geometric Blur

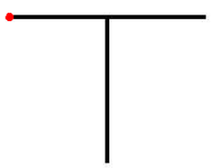




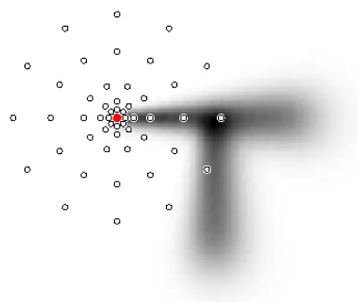
Compute edges at four orientations

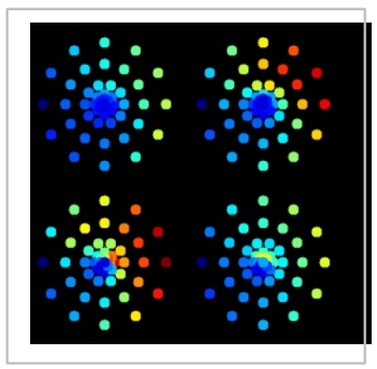
Extract a patch in each channel

Apply spatially varying blur and sub-sample



(Idealized signal)





Example descriptor

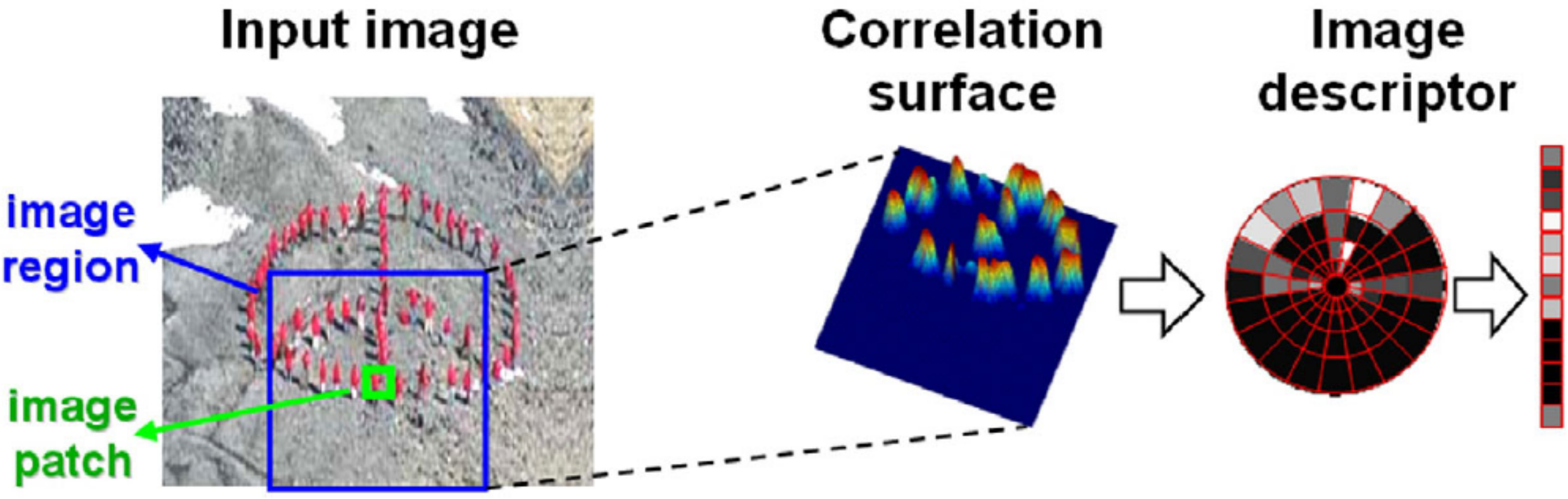
Berg & Malik, CVPR 2001

Self-similarity Descriptor

Figure 1. *These images of the same object (a heart) do NOT share common image properties (colors, textures, edges), but DO share a similar geometric layout of local internal self-similarities.*

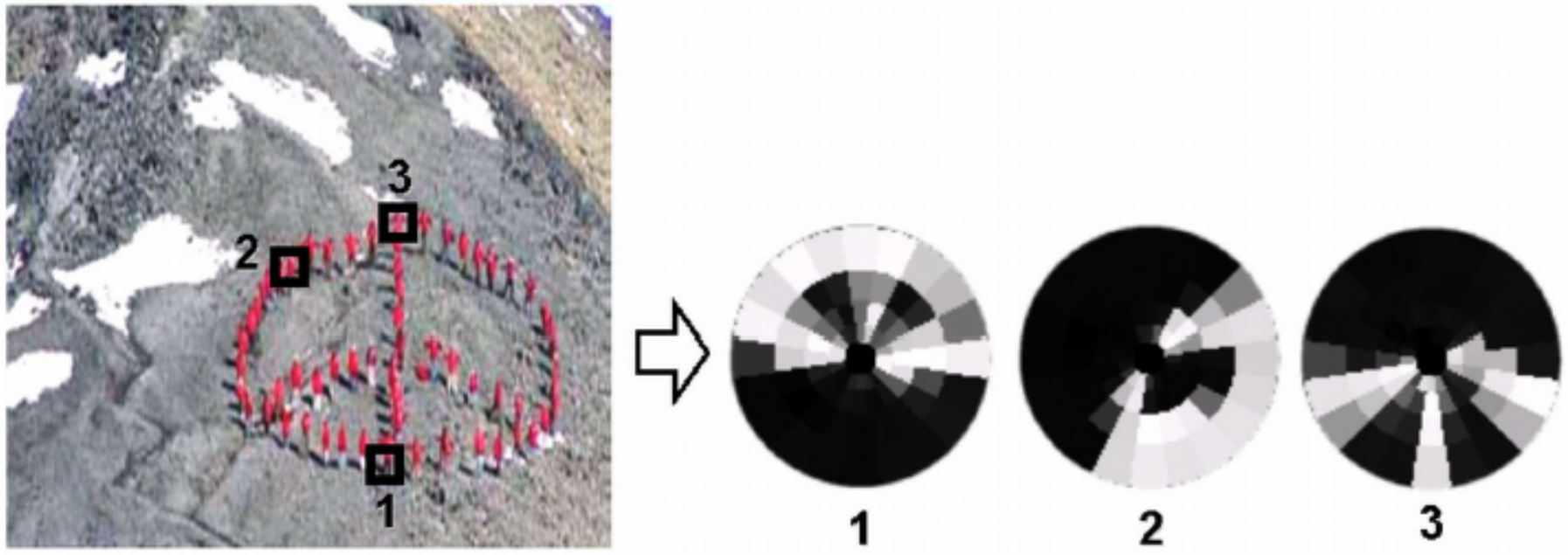
Matching Local Self-Similarities across Images and Videos, Shechtman and Irani, 2007

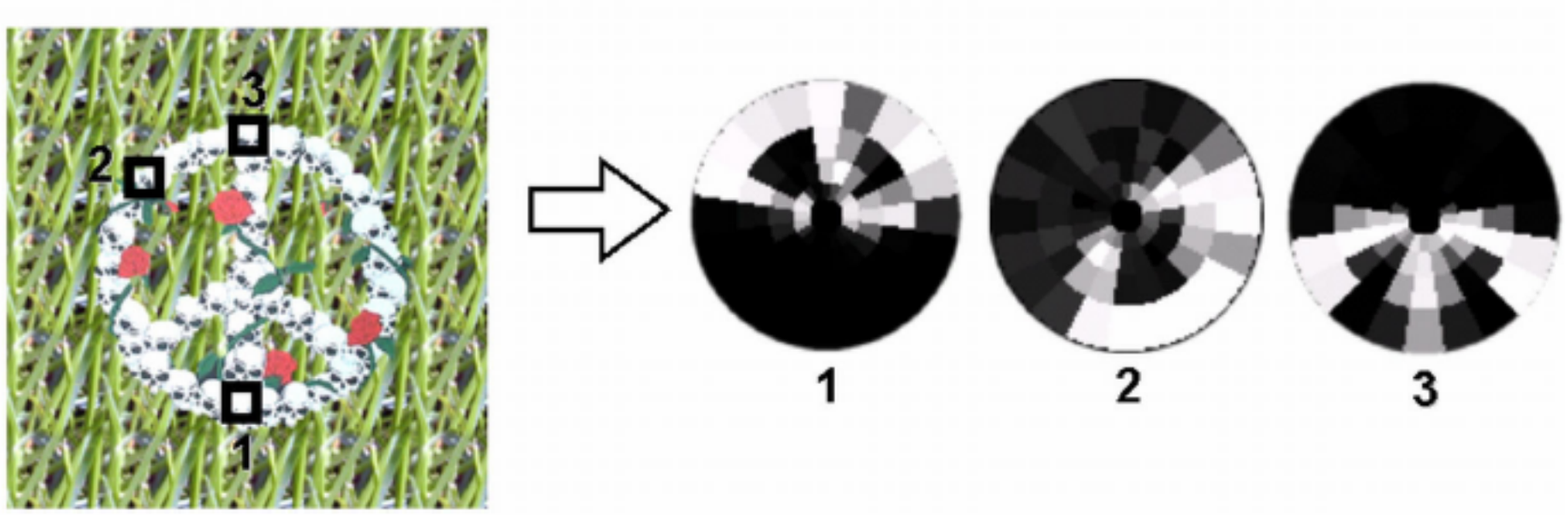
Self-similarity Descriptor



Matching Local Self-Similarities across Images and Videos, Shechtman and Irani, 2007

Self-similarity Descriptor





Matching Local Self-Similarities across Images and Videos, Shechtman and Irani, 2007

Right features depend on what you want to know

- Shape: scene-scale, object-scale, detail-scale
  - 2D form, shading, shadows, texture, linear perspective
- Material properties: albedo, feel, hardness, …
  - Color, texture
- Motion
  - Optical flow, tracked points
- Distance
  - Stereo, position, occlusion, scene shape
  - If known object: size, other objects

Things to remember about representation

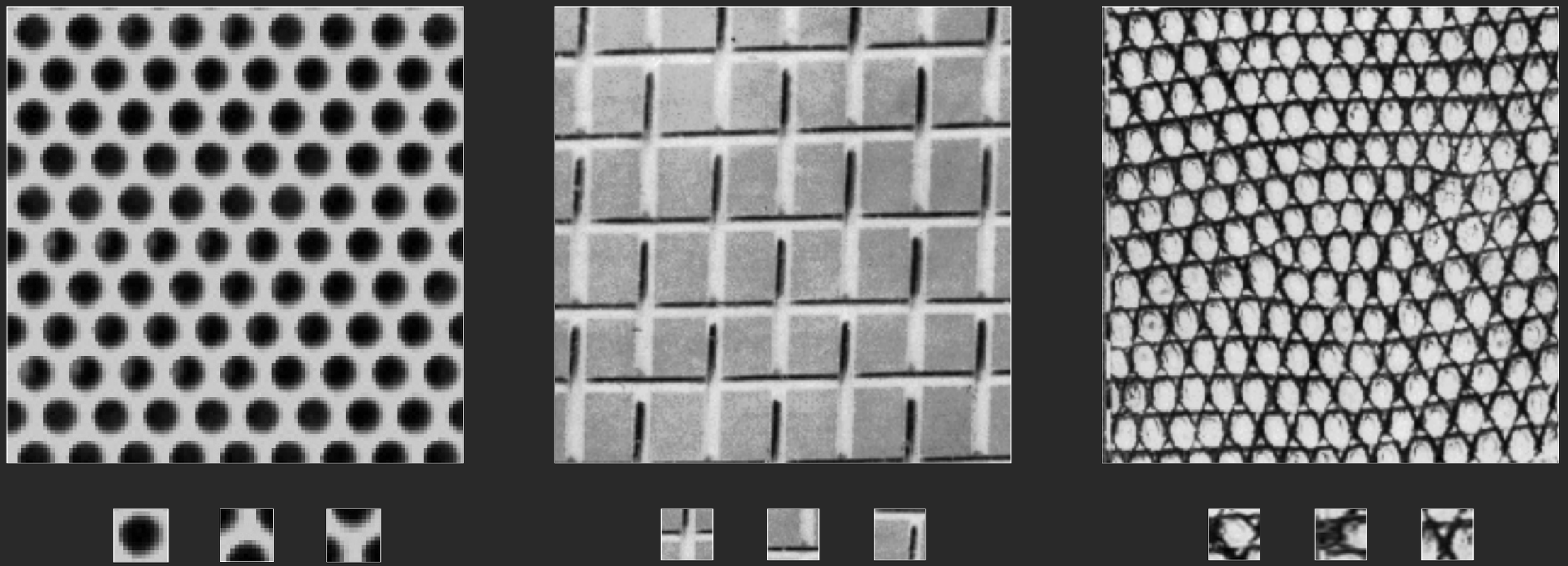
- Most features can be thought of as templates, histograms (counts), or combinations

- Think about the right features for the problem
  - Coverage
  - Concision
  - Directness

Bag-of-features models

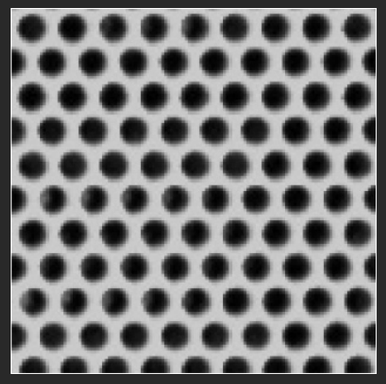
Origin 1: Texture recognition

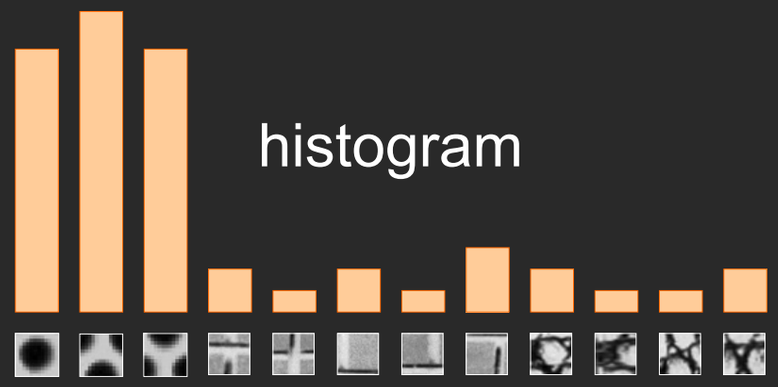
- Texture is characterized by the repetition of basic elements or *textons*
- For stochastic textures, it is the identity of the textons, not their spatial arrangement, that matters

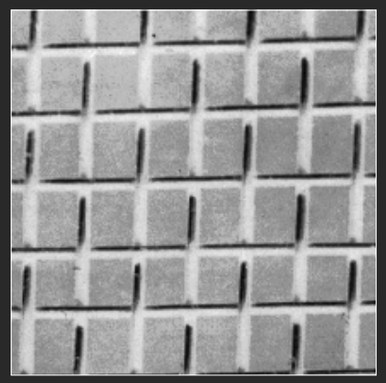


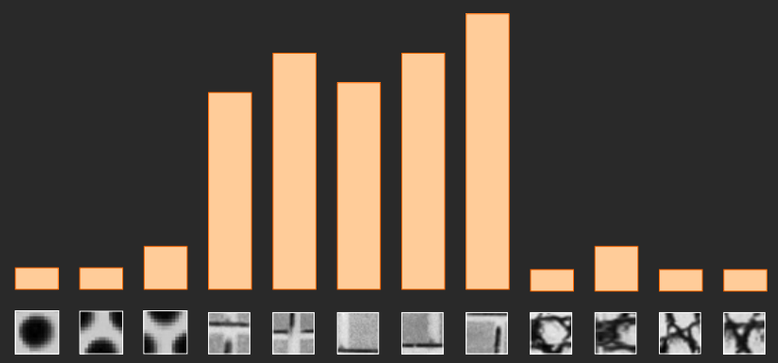
Julesz, 1981; Cula & Dana, 2001; Leung & Malik 2001; Mori, Belongie & Malik, 2001; Schmid 2001; Varma & Zisserman, 2002, 2003; Lazebnik, Schmid & Ponce, 2003

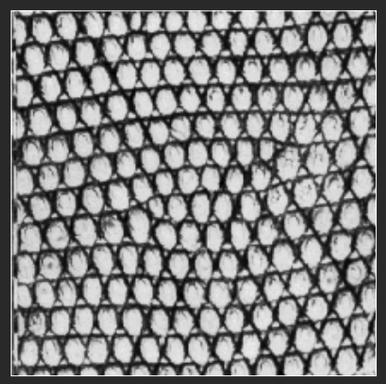
Origin 1: Texture recognition

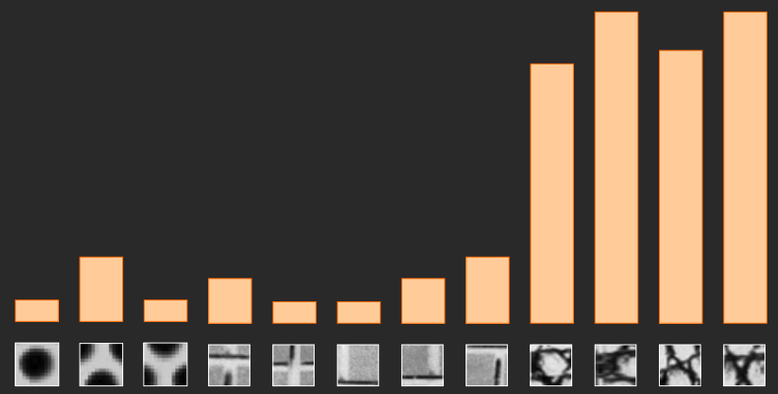












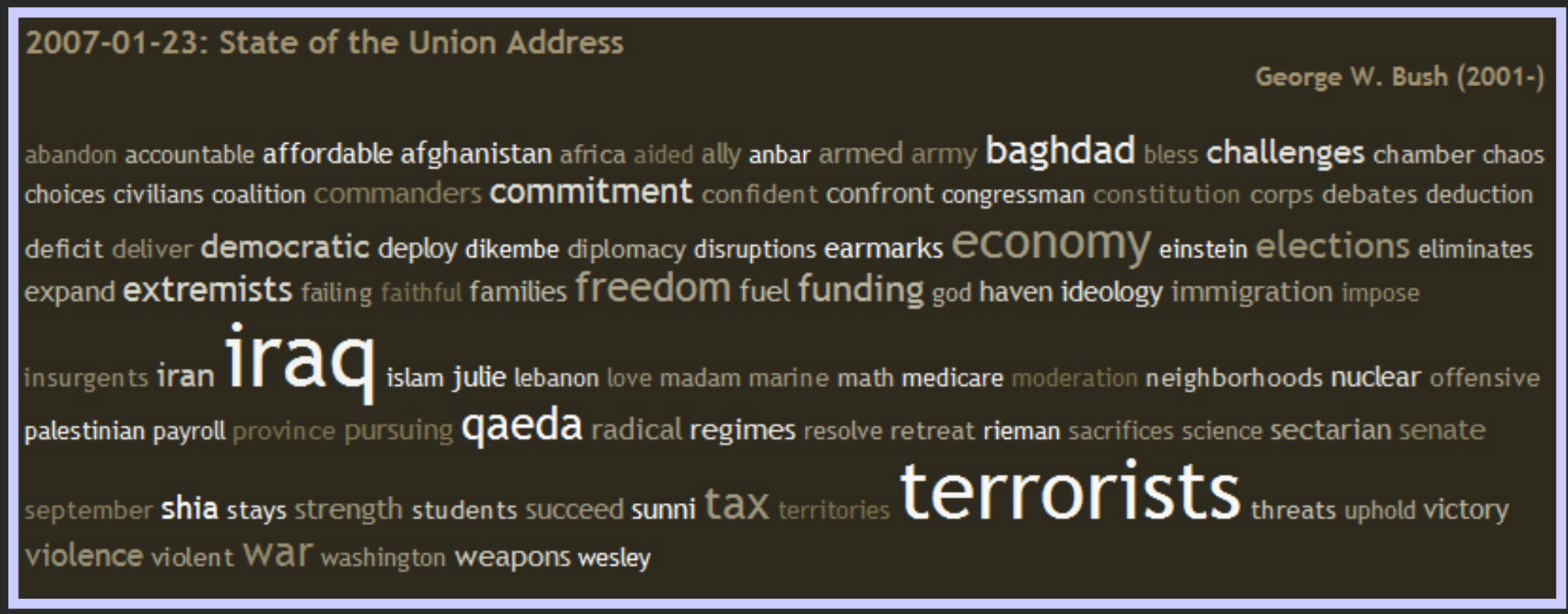
Julesz, 1981; Cula & Dana, 2001; Leung & Malik 2001; Mori, Belongie & Malik, 2001; Schmid 2001; Varma & Zisserman, 2002, 2003; Lazebnik, Schmid & Ponce, 2003

Origin 2: Bag-of-words models

- Orderless document representation: frequencies of words from a dictionary Salton & McGill (1983)

Origin 2: Bag-of-words models

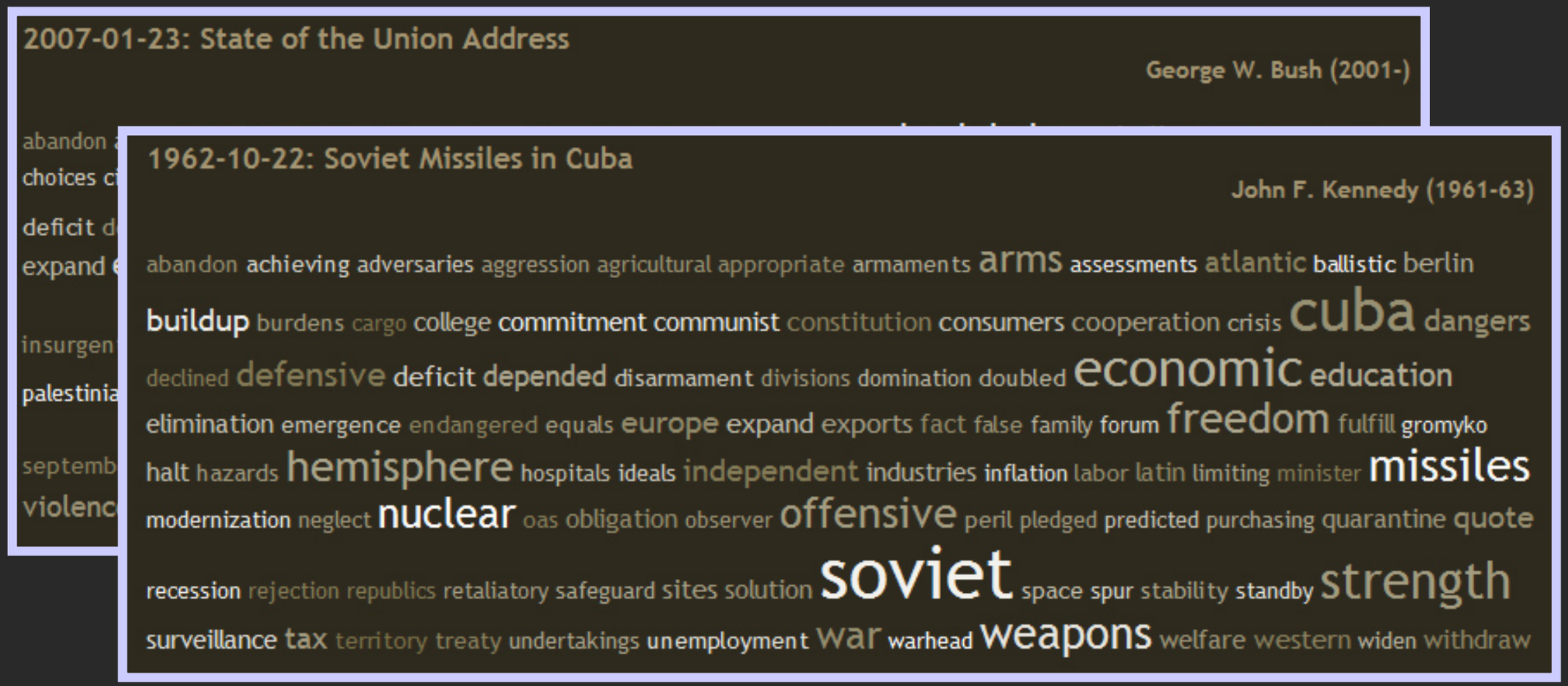
- Orderless document representation: frequencies of words from a dictionary Salton & McGill (1983)



US Presidential Speeches Tag Cloud
http://chir.ag/phernalia/preztags/

Origin 2: Bag-of-words models

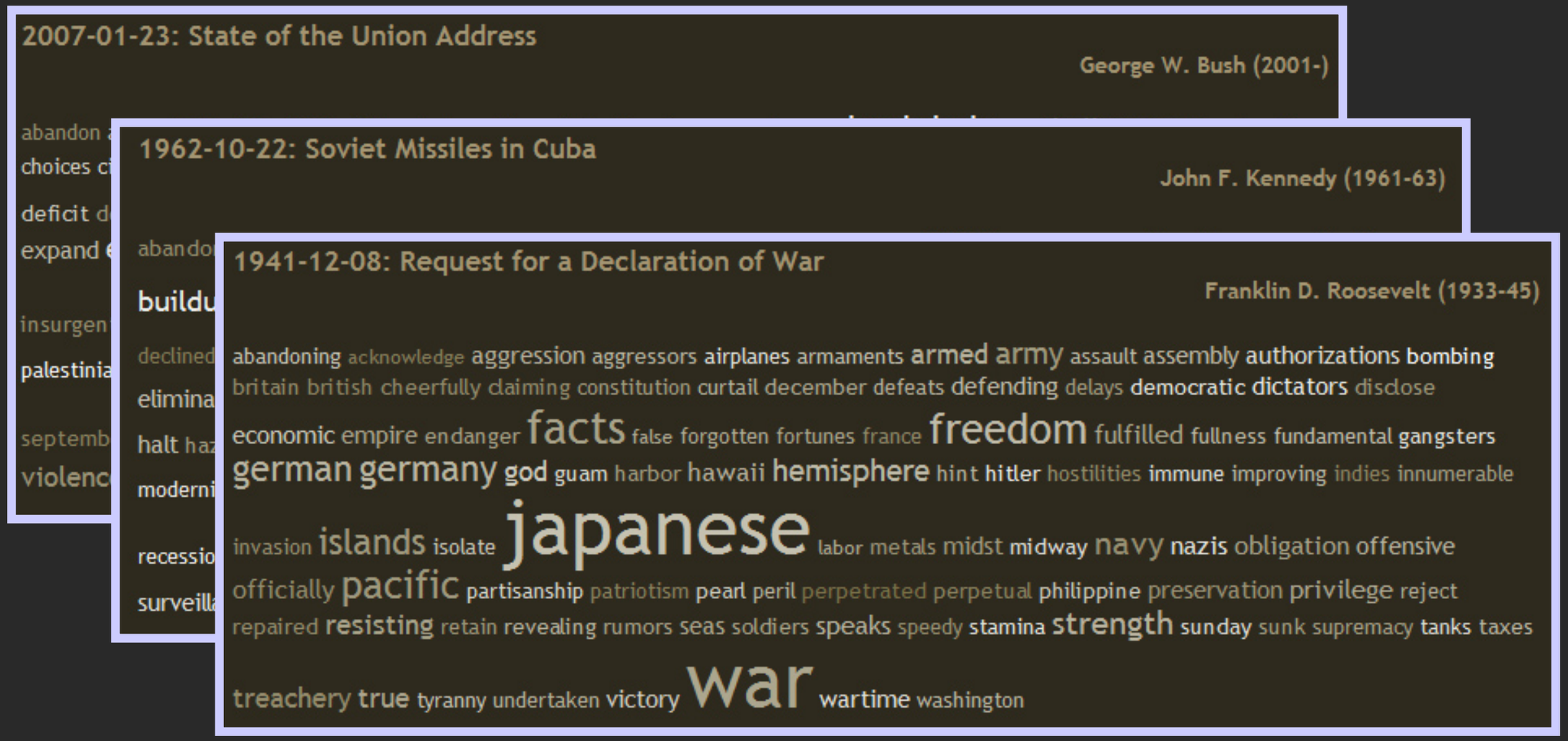
- Orderless document representation: frequencies of words from a dictionary Salton & McGill (1983)



US Presidential Speeches Tag Cloud
http://chir.ag/phernalia/preztags/

Origin 2: Bag-of-words models

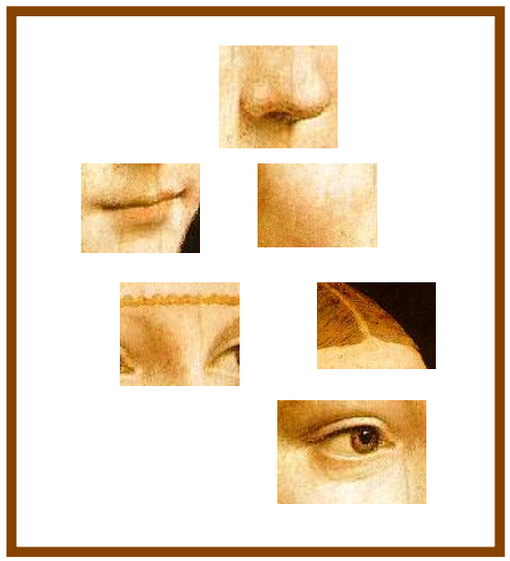
- Orderless document representation: frequencies of words from a dictionary Salton & McGill (1983)

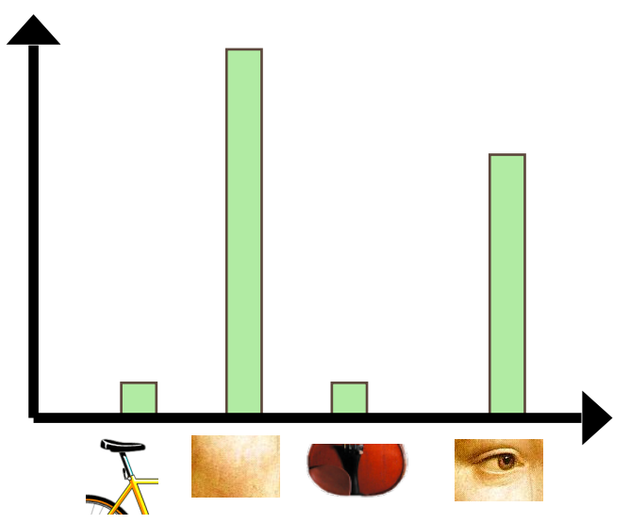


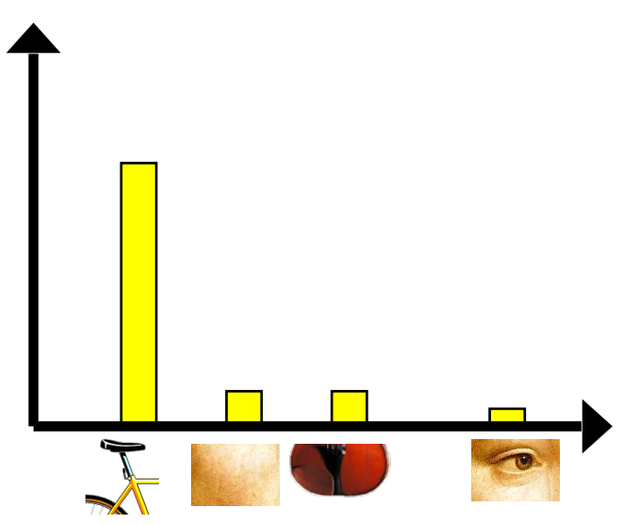
US Presidential Speeches Tag Cloud
http://chir.ag/phernalia/preztags/

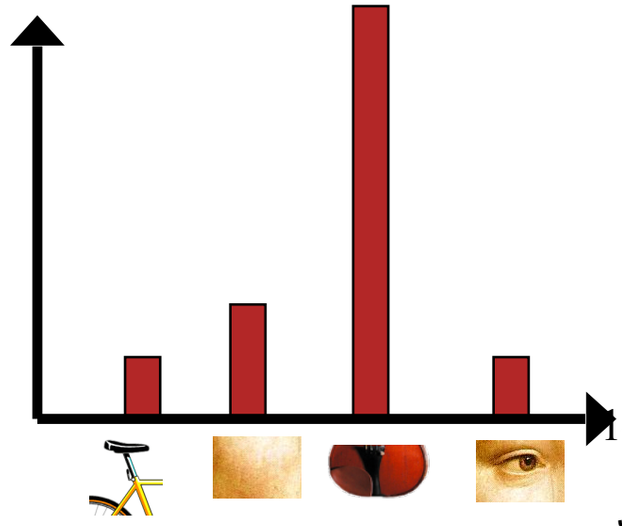
Bag-of-features steps

1. Extract features
2. Learn “visual vocabulary”
3. Quantize features using visual vocabulary
4. Represent images by frequencies of “visual words”





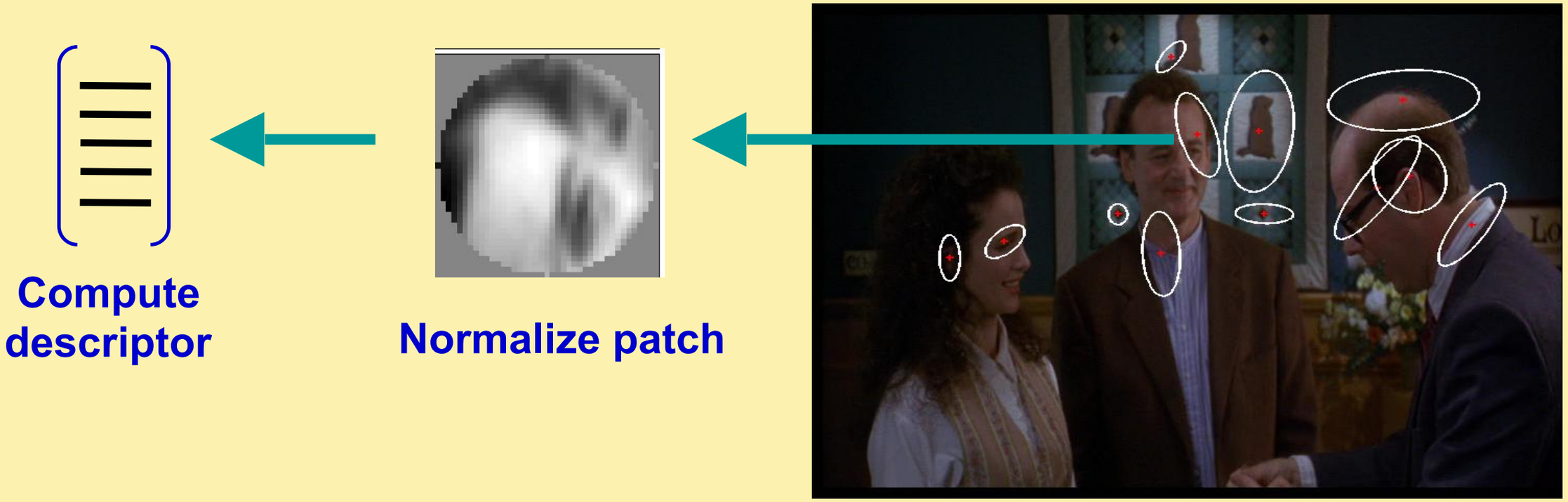




1. Feature extraction

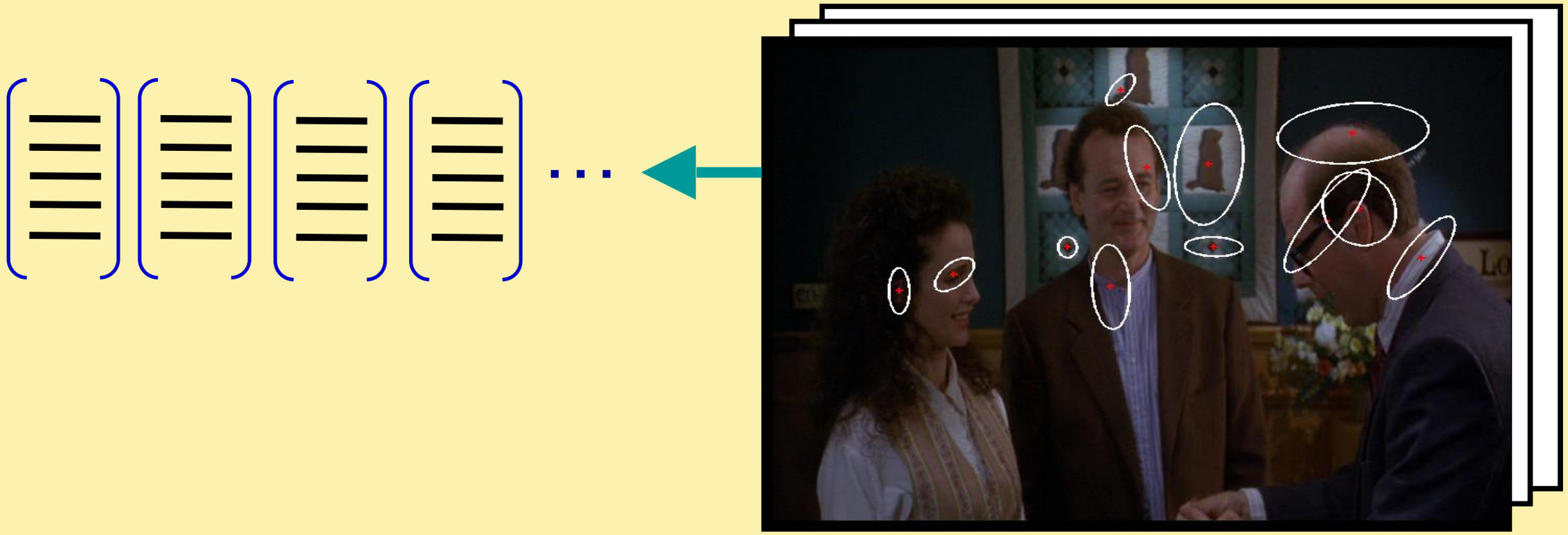
- Regular grid or interest regions

1. Feature extraction



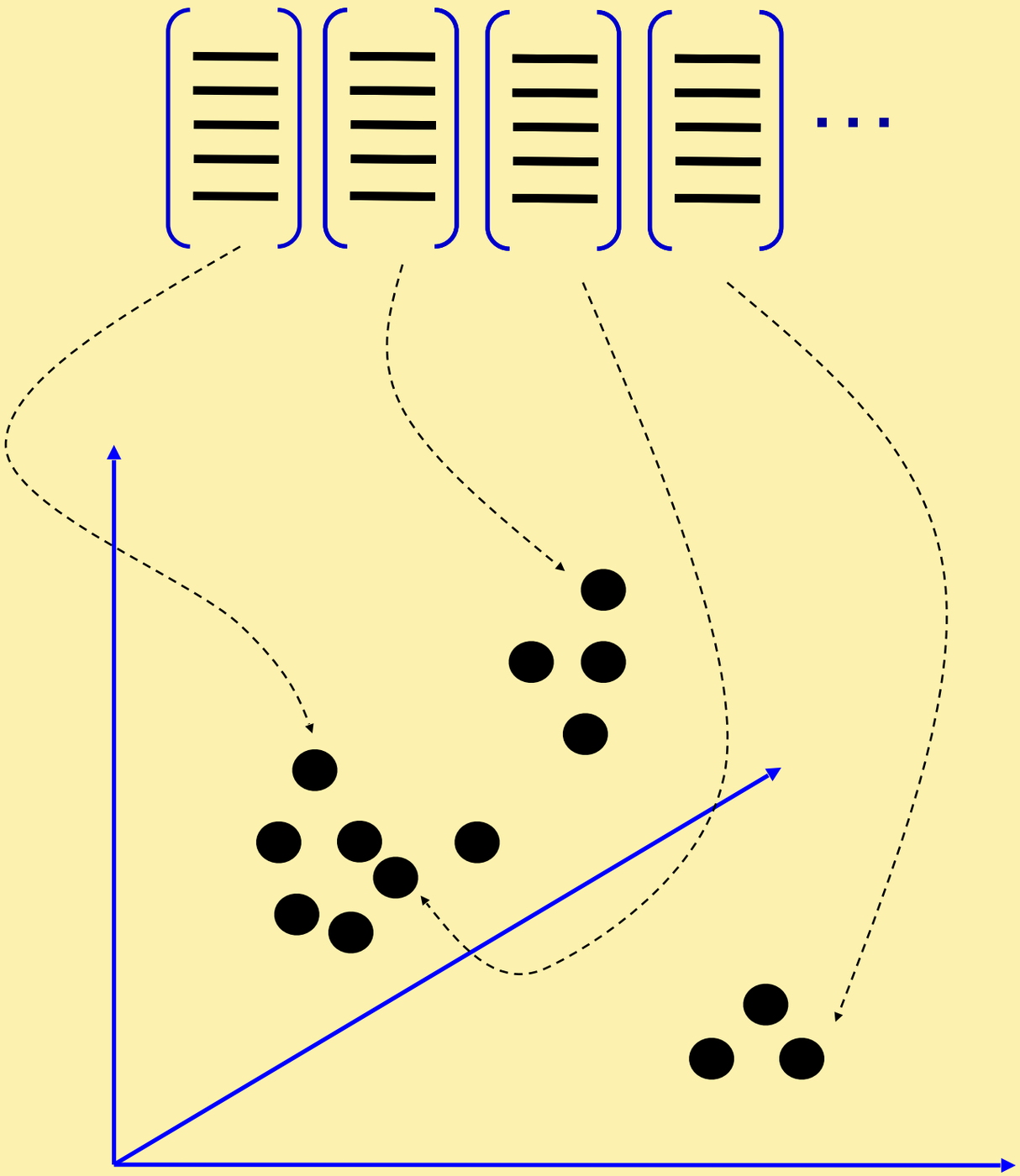
Slide credit: Josef Sivic

1. Feature extraction



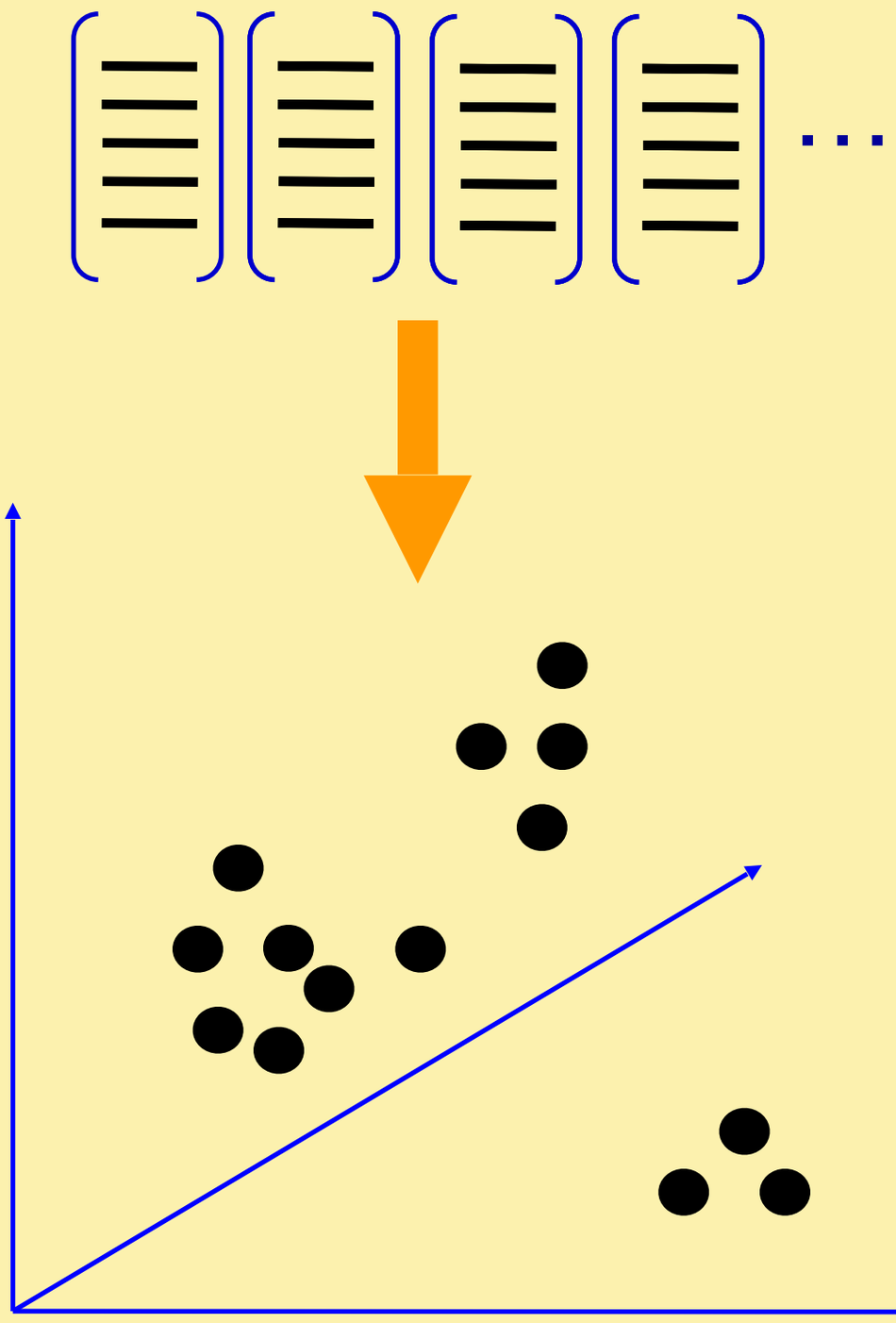
Slide credit: Josef Sivic

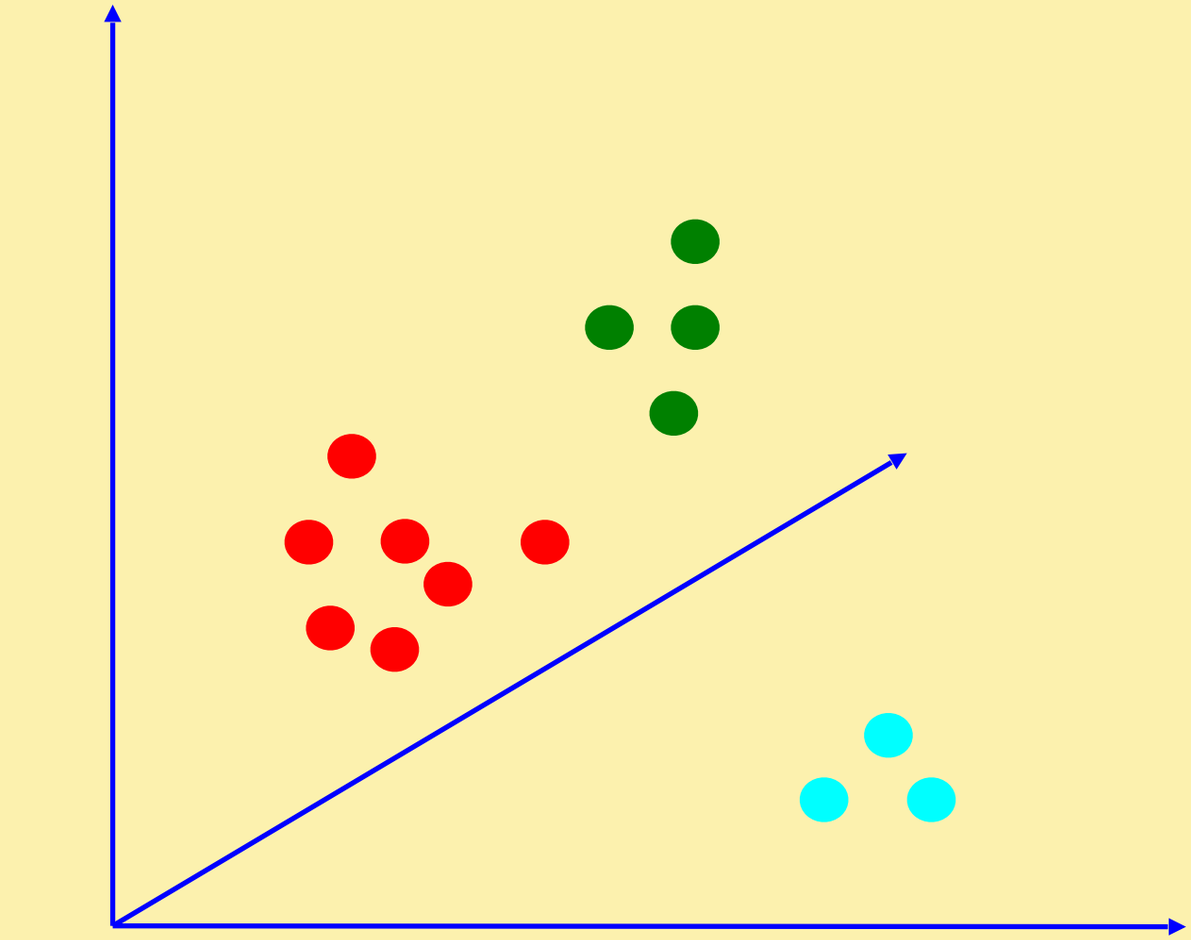
2. Learning the visual vocabulary



Slide credit: Josef Sivic

2. Learning the visual vocabulary

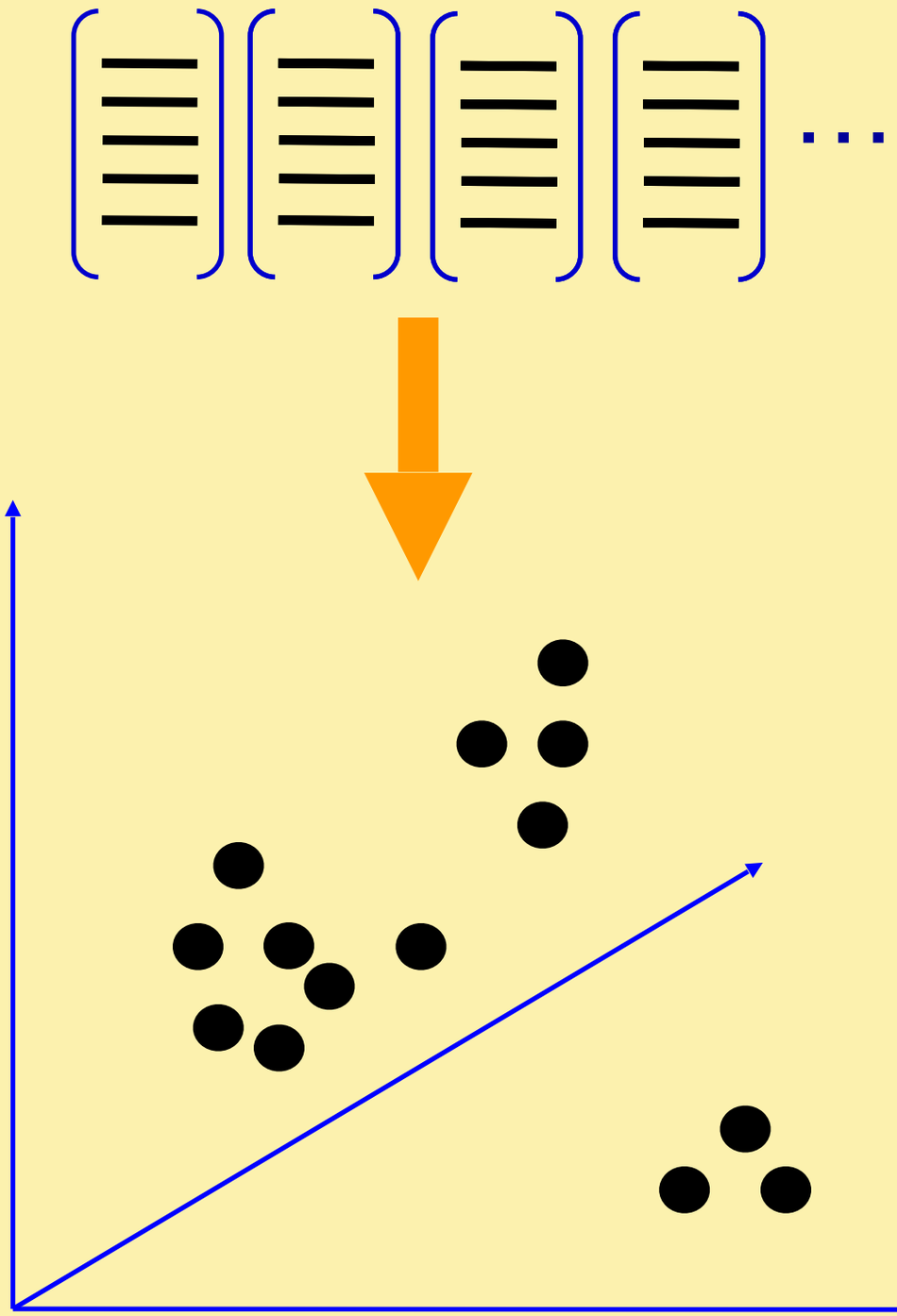


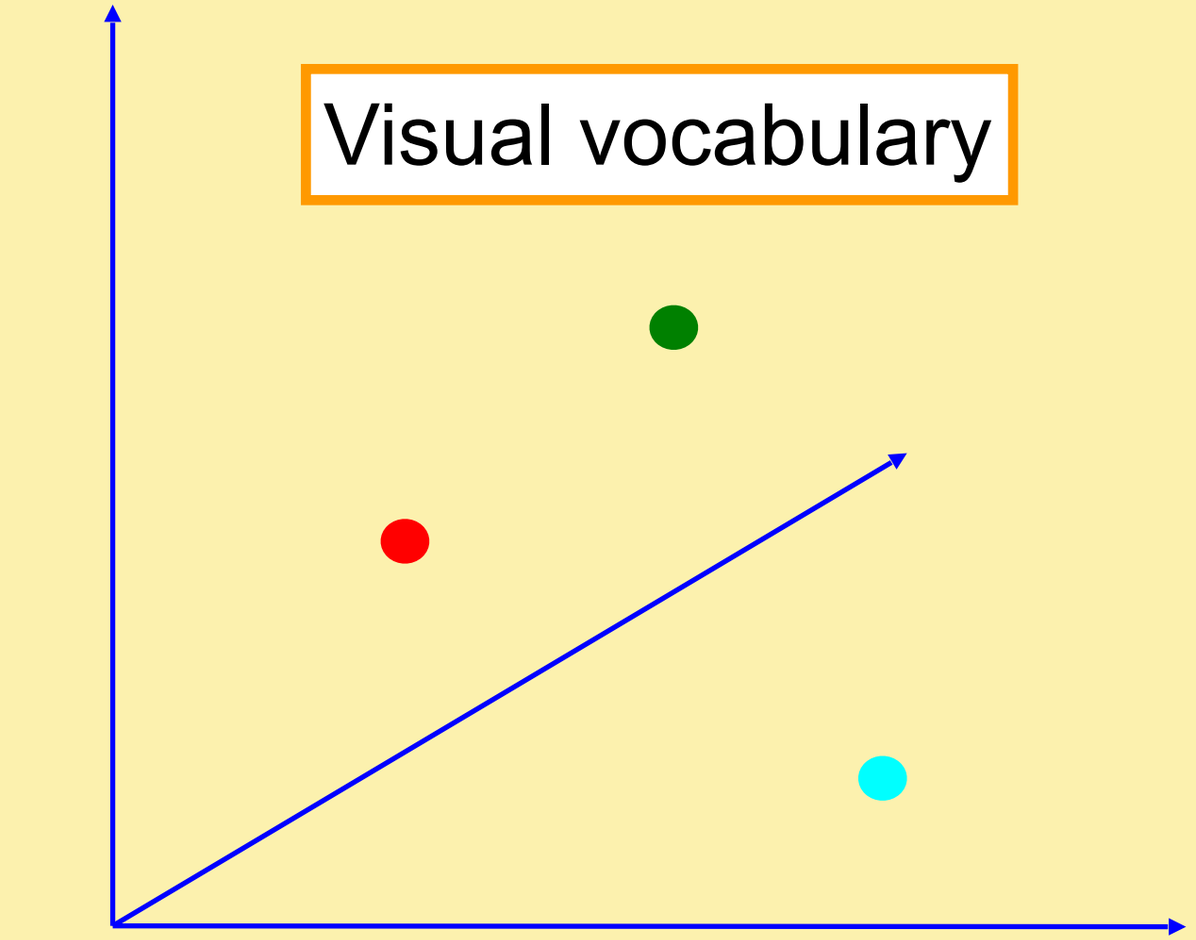


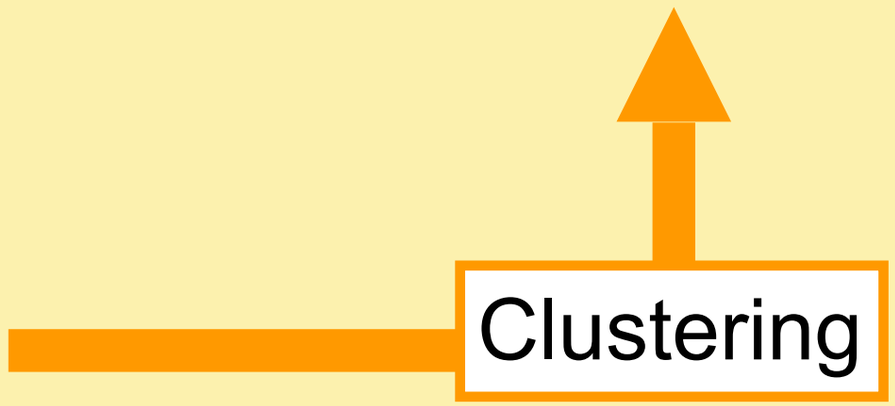
Clustering

Slide credit: Josef Sivic

2. Learning the visual vocabulary







Slide credit: Josef Sivic

K-means clustering

- Want to minimize sum of squared Euclidean distances between points x_i and their nearest cluster centers m_k

$$D(X,M) = \sum_{\text{cluster } k} \sum_{\text{point } i \text{ in cluster } k} (x_i - m_k)^2$$

Algorithm:

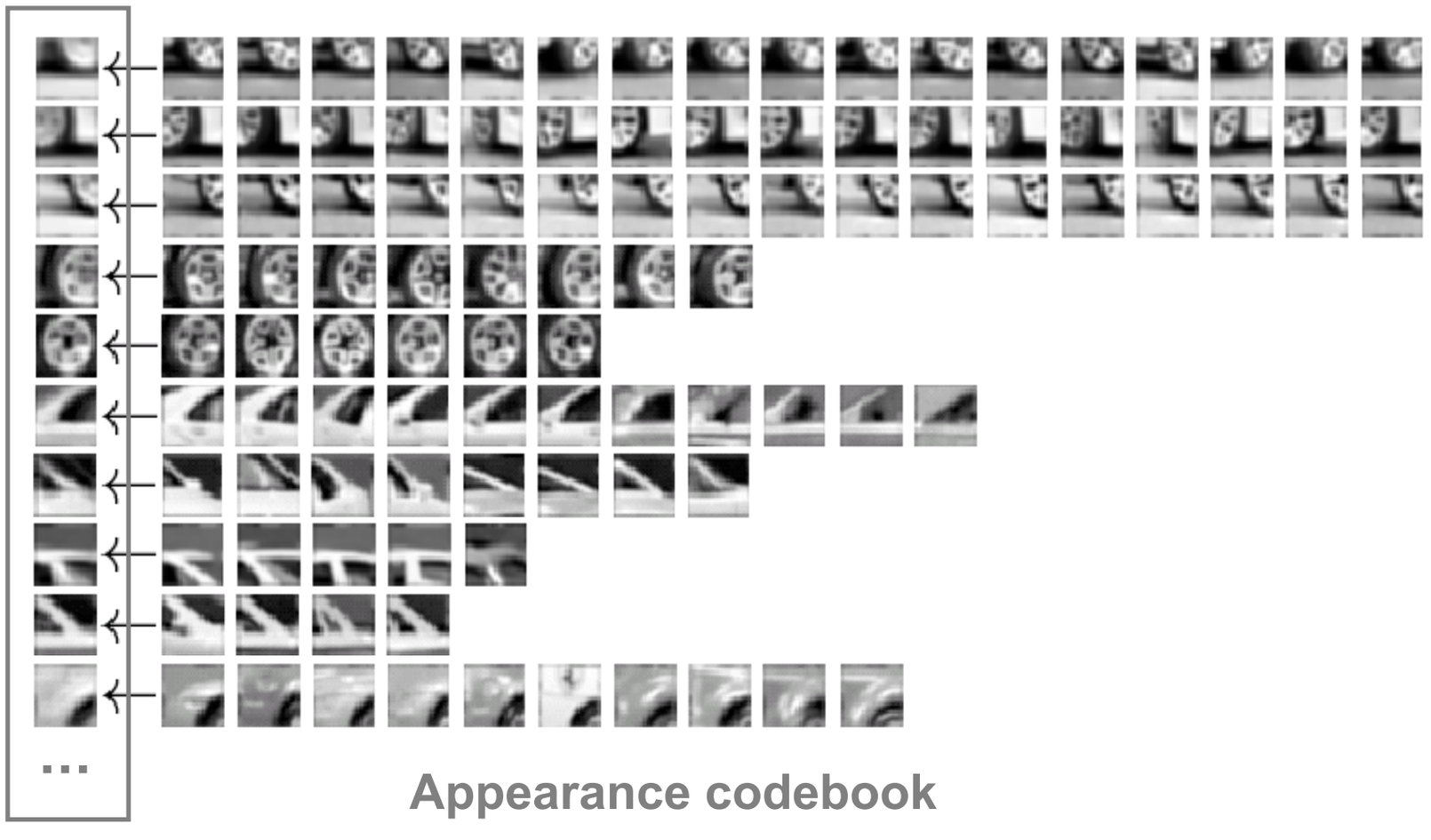
- Randomly initialize K cluster centers
- Iterate until convergence:
  - Assign each data point to the nearest center
  - Recompute each cluster center as the mean of all points assigned to it

Clustering and vector quantization

- Clustering is a common method for learning a visual vocabulary or codebook
  - Unsupervised learning process
  - Each cluster center produced by k-means becomes a codevector
  - Codebook can be learned on separate training set
  - Provided the training set is sufficiently representative, the codebook will be “universal”

- The codebook is used for quantizing features
  - A *vector quantizer* takes a feature vector and maps it to the index of the nearest codevector in a codebook
  - Codebook = visual vocabulary
  - Codevector = visual word

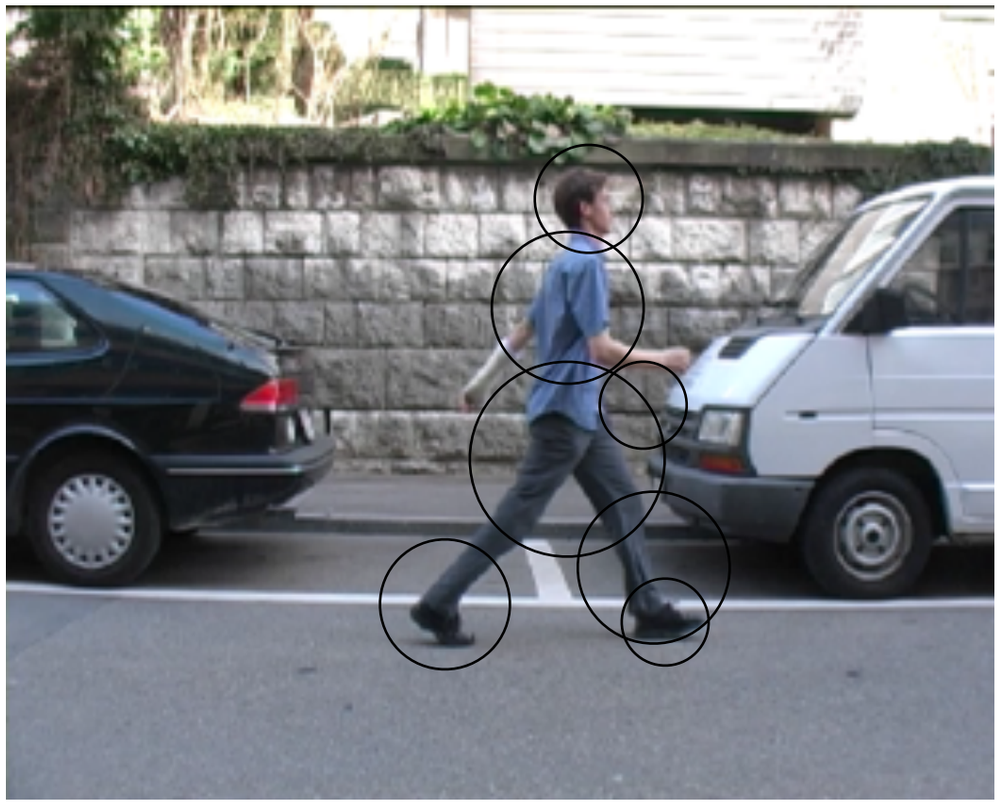
Example codebook

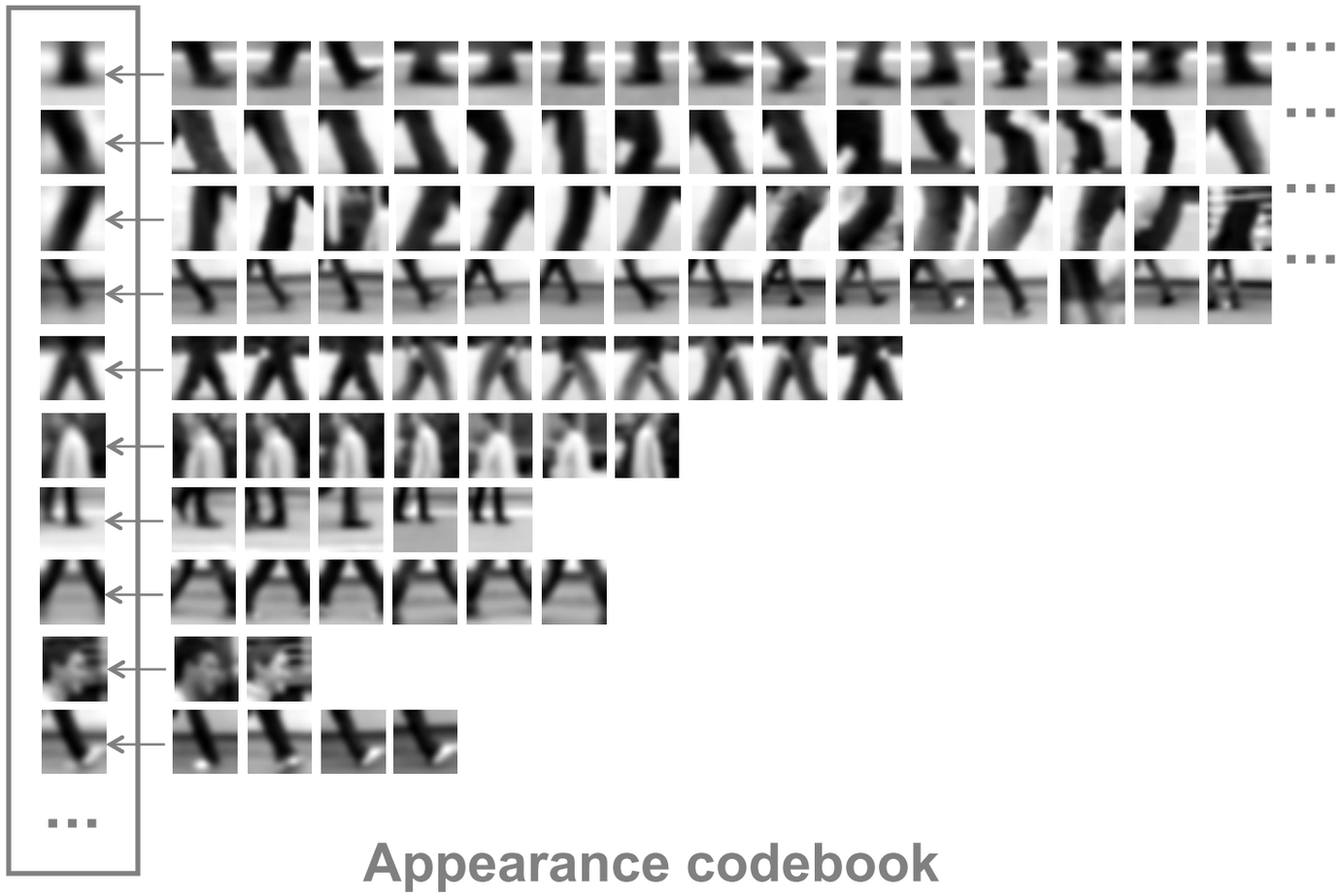


Appearance codebook

Source: B. Leibe

Another codebook



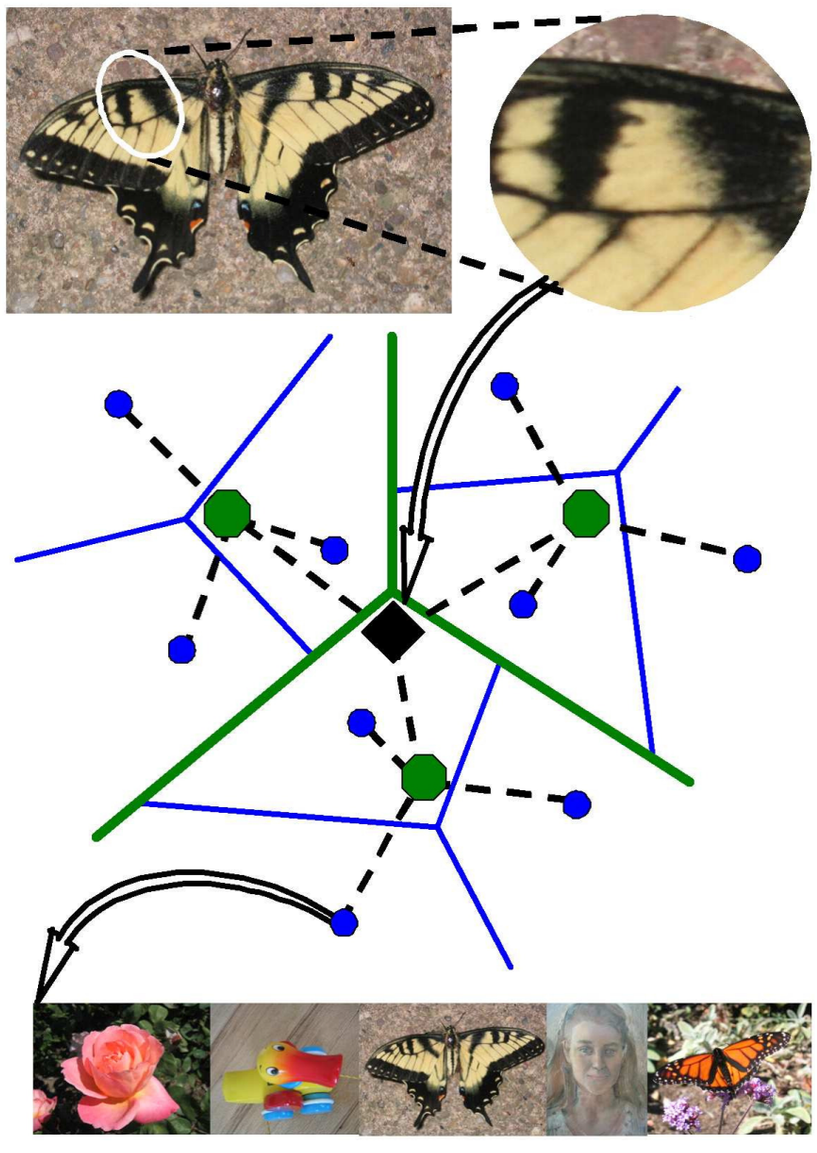


Appearance codebook

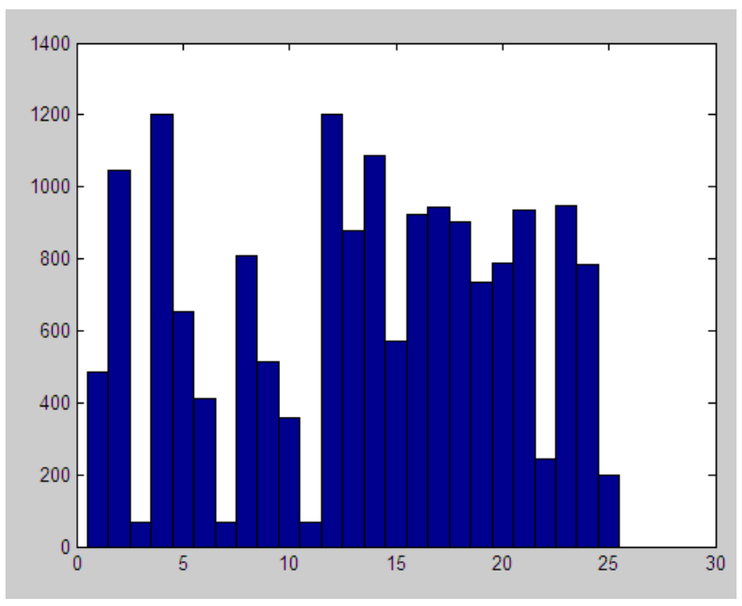
Source: B. Leibe

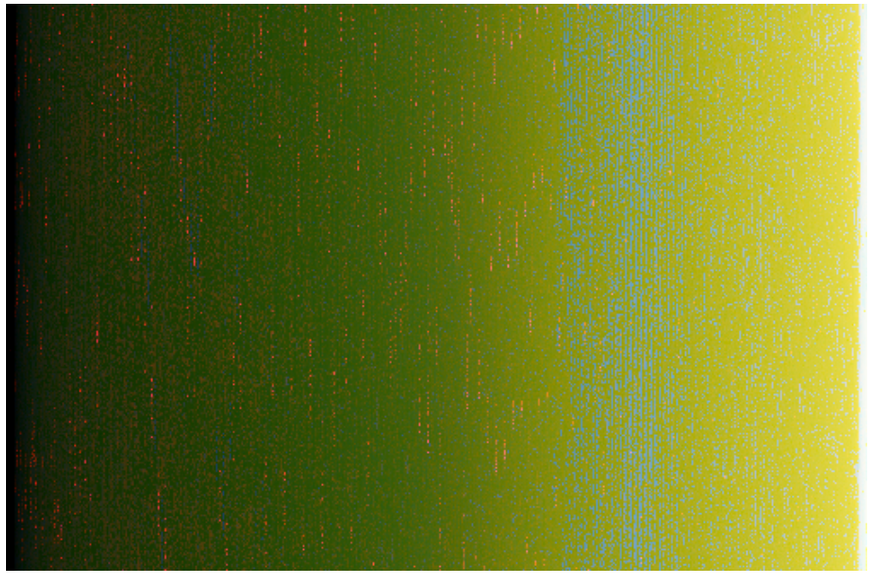
Visual vocabularies: Issues

- How to choose vocabulary size?
  - Too small: visual words not representative of all patches
  - Too large: quantization artifacts, overfitting
- Computational efficiency
  - Vocabulary trees
    (Nister & Stewenius, 2006)



But what about layout?





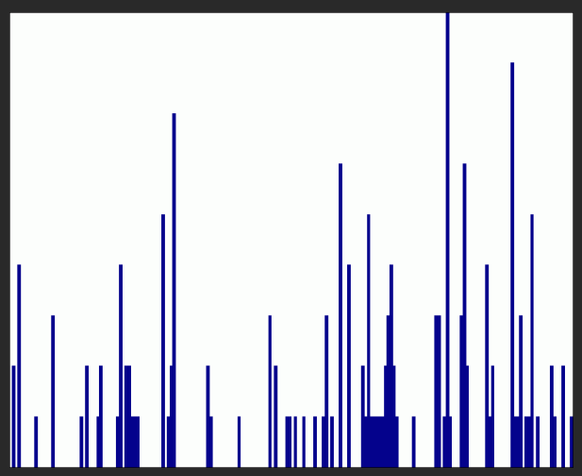
All of these images have the same color histogram

Spatial pyramid

Compute histogram in each spatial bin

Spatial pyramid representation

- Extension of a bag of features
- Locally orderless representation at several levels of resolution

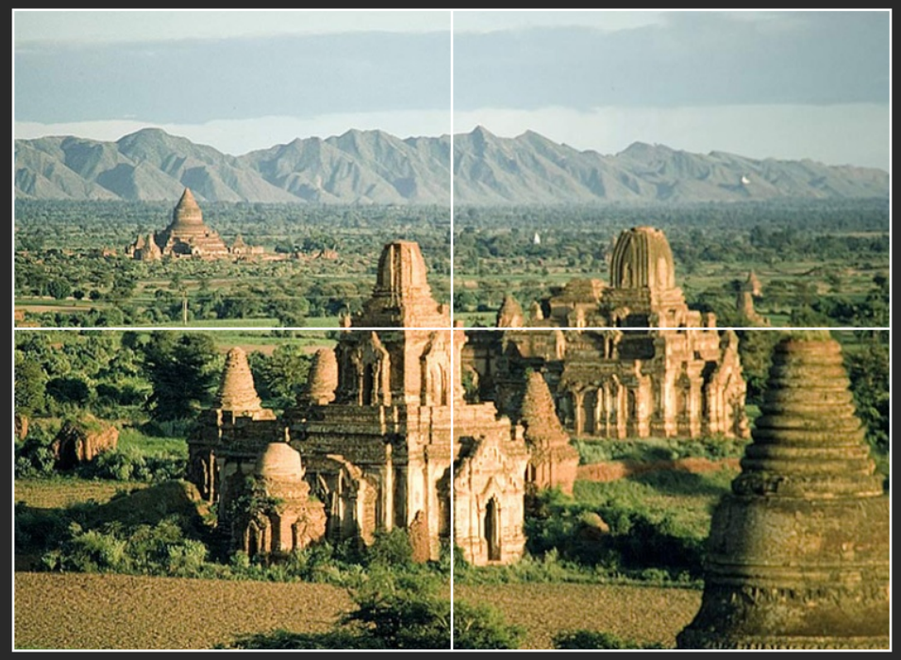


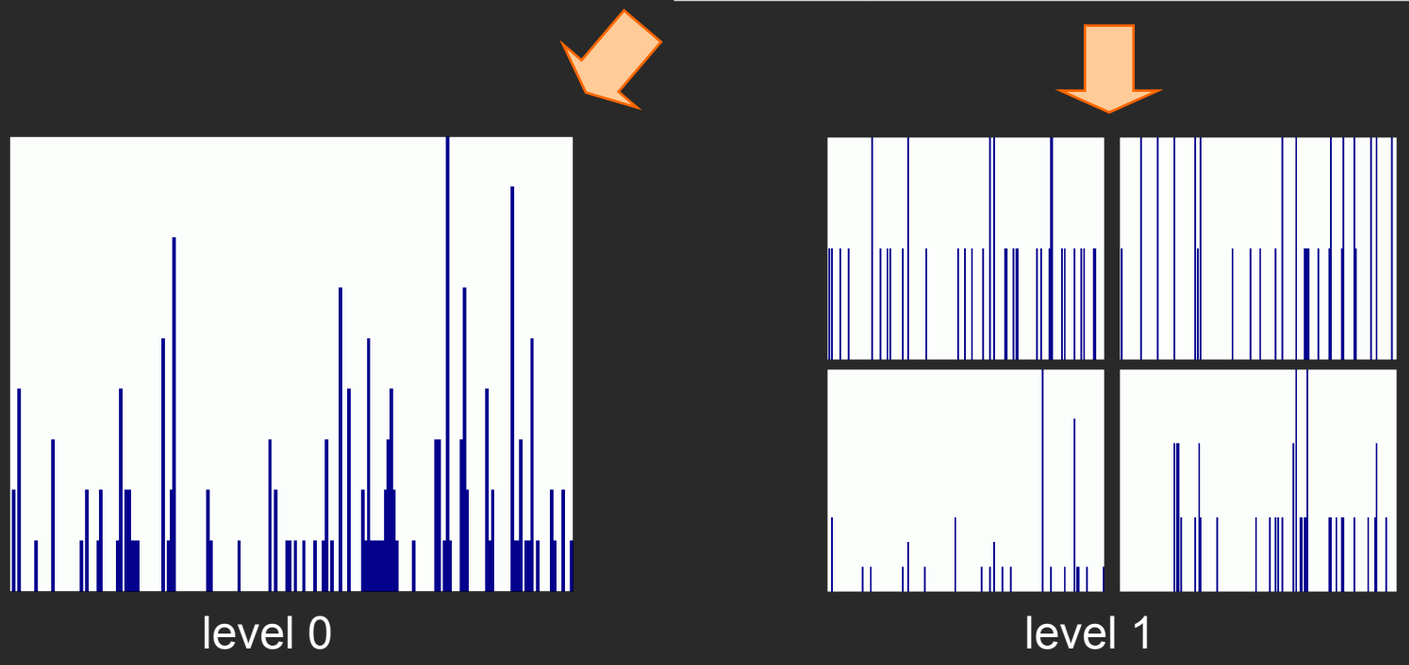
level 0

Lazebnik, Schmid & Ponce (CVPR 2006)

Spatial pyramid representation

- Extension of a bag of features
- Locally orderless representation at several levels of resolution

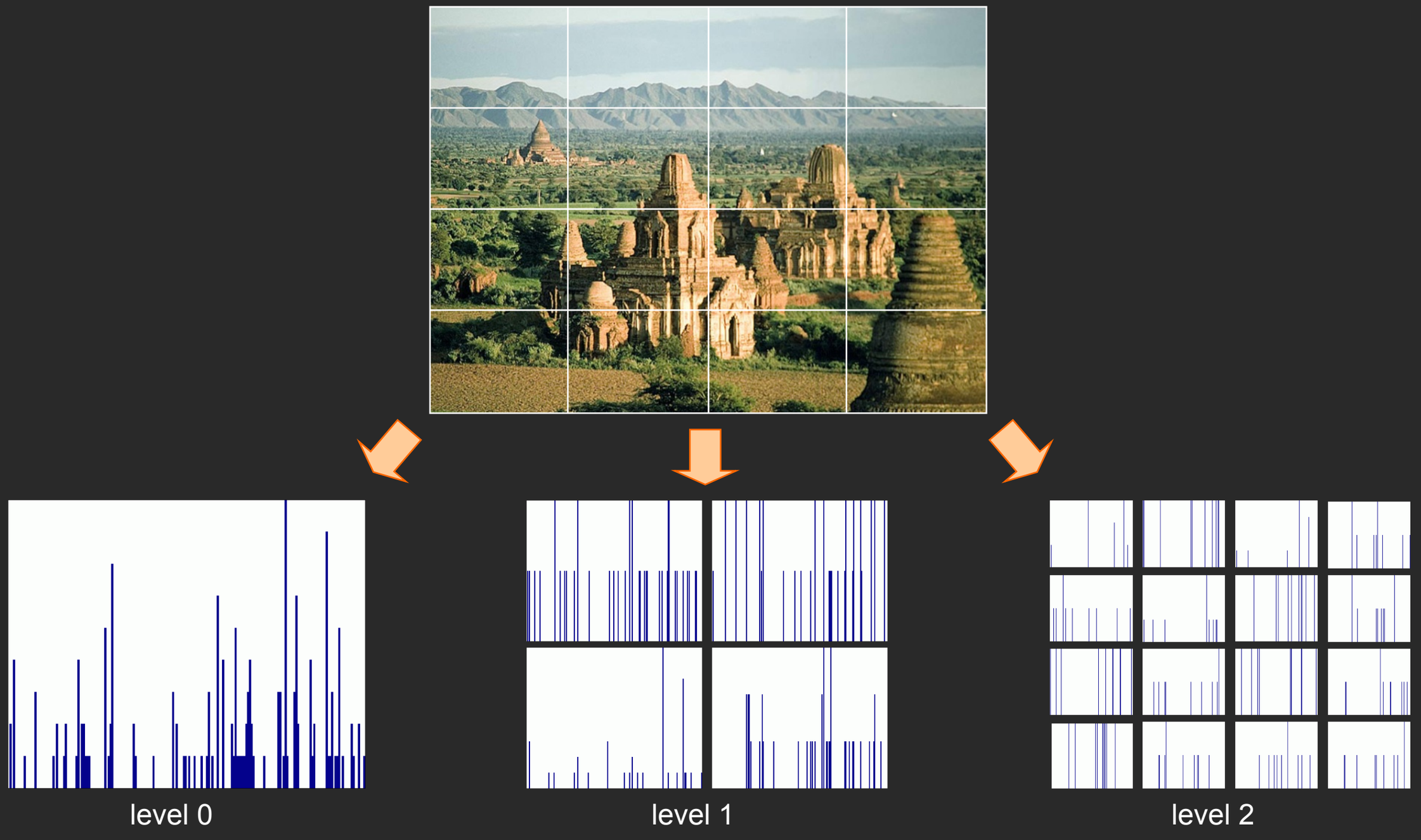




Lazebnik, Schmid & Ponce (CVPR 2006)

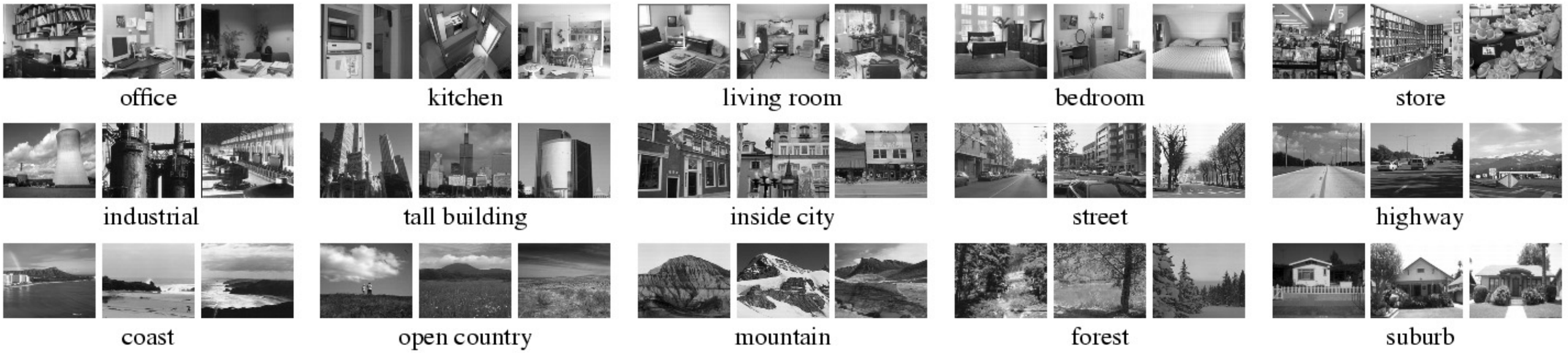
Spatial pyramid representation

- Extension of a bag of features
- Locally orderless representation at several levels of resolution



Lazebnik, Schmid & Ponce (CVPR 2006)

Scene category dataset

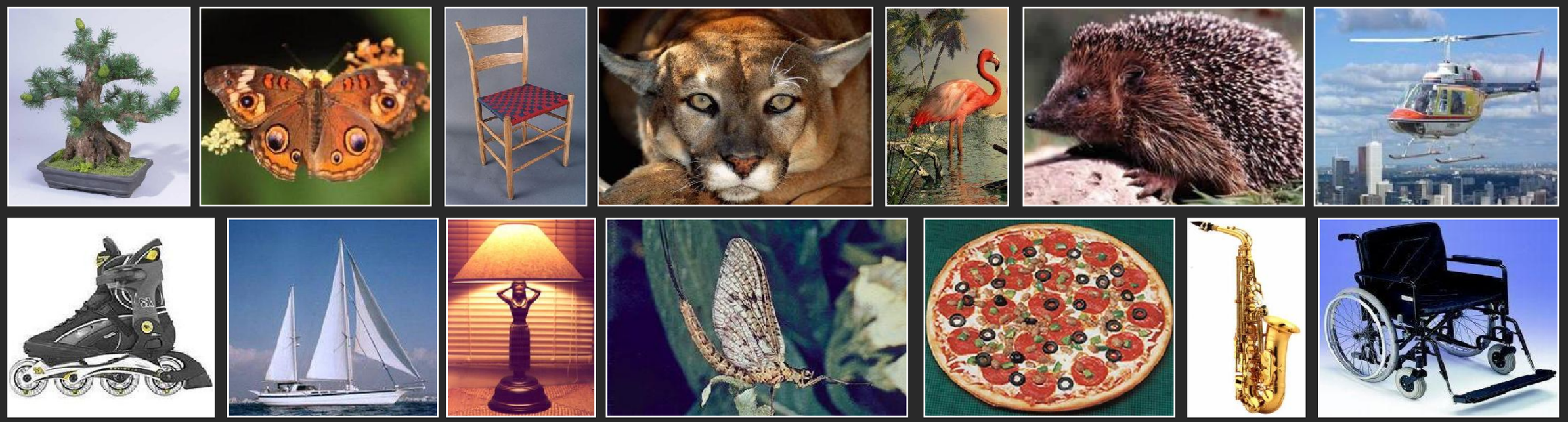


Multi-class classification results
(100 training images per class)

	Weak features (vocabulary size: 16)		Strong features (vocabulary size: 200)	
Level	Single-level	Pyramid	Single-level	Pyramid
0 (1×1)	45.3 ±0.5		72.2 ±0.6	
1 (2×2)	53.6 ±0.3	56.2 ±0.6	77.9 ±0.6	79.0 ±0.5
2 (4×4)	61.7 ±0.6	64.7 ±0.7	79.4 ±0.3	**81.1** ±0.3
3 (8×8)	63.3 ±0.8	**66.8** ±0.6	77.2 ±0.4	80.7 ±0.3

Caltech101 dataset

http://www.vision.caltech.edu/Image_Datasets/Caltech101/Caltech101.html

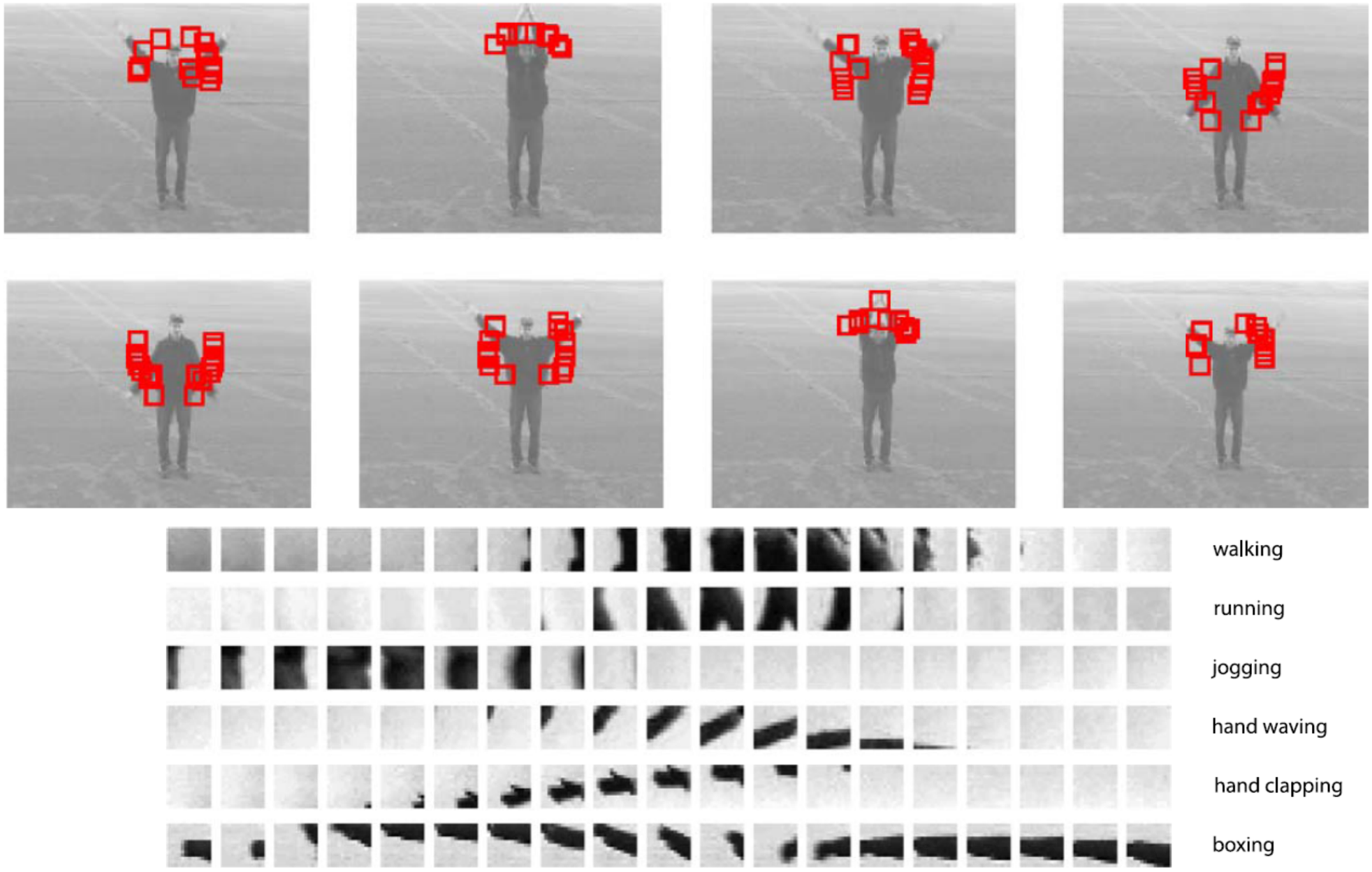


Multi-class classification results (30 training images per class)

	Weak features (16)		Strong features (200)	
Level	Single-level	Pyramid	Single-level	Pyramid
0	15.5 ±0.9		41.2 ±1.2	
1	31.4 ±1.2	32.8 ±1.3	55.9 ±0.9	57.0 ±0.8
2	47.2 ±1.1	49.3 ±1.4	63.6 ±0.9	**64.6** ±0.8
3	52.2 ±0.8	**54.0** ±1.1	60.3 ±0.9	64.6 ±0.7

Bags of features for action recognition

Space-time interest points



Juan Carlos Niebles, Hongcheng Wang and Li Fei-Fei, **Unsupervised Learning of Human Action Categories Using Spatial-Temporal Words**, IJCV 2008.

Summary

- Local features
- Bag of Word Models