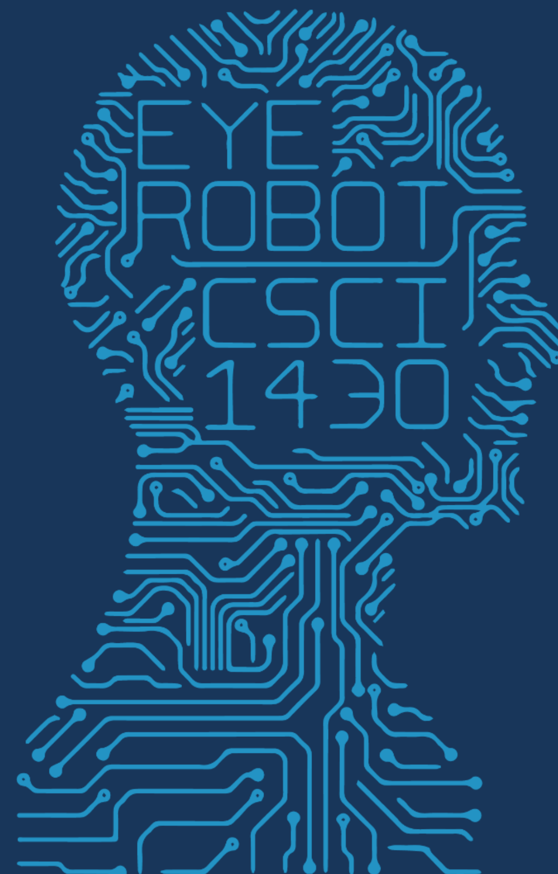
1950

Future Vision

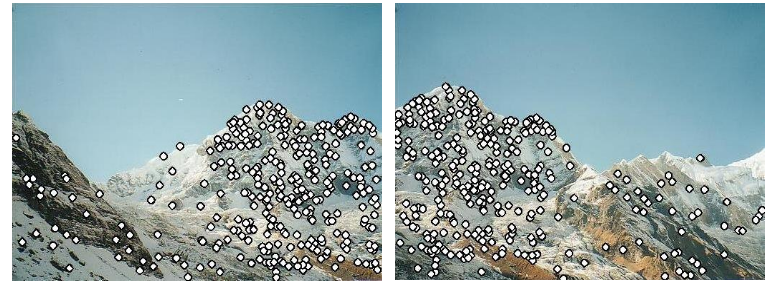


2017 MWF 1pm 368

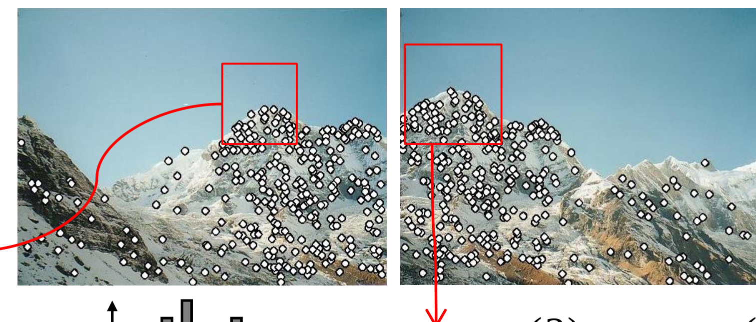
Computer Vision

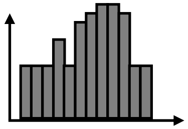
Local features: main components

1) **Detection:**
Find a set of distinctive key points.

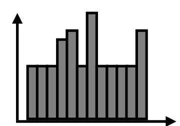


2) **Description:**
Extract feature descriptor around each interest point as vector.





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



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

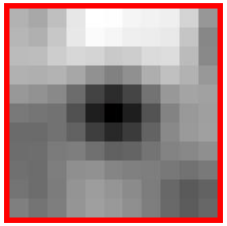
3) **Matching:**
Compute distance between feature vectors to find correspondence.

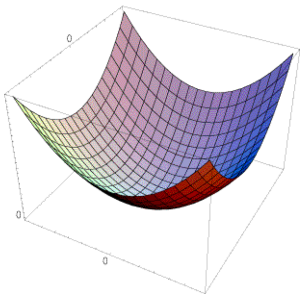
$$d(\mathrm{x}_1, \mathrm{x}_2) < T$$

K. Grauman, B. Leibe

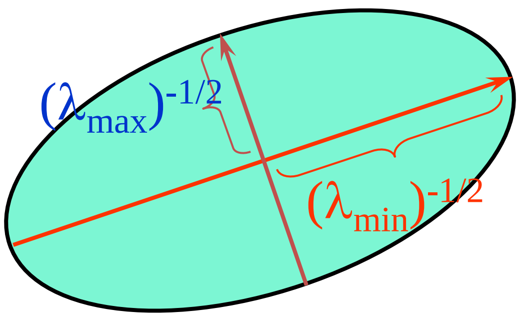
Review: Harris corner detector

$E(u, v)$





- Approximate distinctiveness by local auto-correlation.
- Approximate local auto-correlation by second moment matrix **M**.
- Distinctiveness (or cornerness) relates to the eigenvalues of **M**.
- Instead of computing eigenvalues directly, we can use determinant and trace of **M**.



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

James Hays

Trace / determinant and eigenvalues

- Given n x n matrix A with eigenvalues $\lambda_{1 \ldots n}$

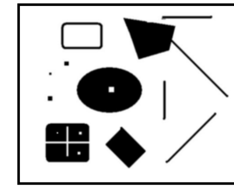
$$\operatorname{tr}(A) = \sum_{i=1}^{n} A_{ii} = \sum_{i=1}^{n} \lambda_i = \lambda_1 + \lambda_2 + \cdots + \lambda_n .$$

$$\det(A) = \prod_{i=1}^{n} \lambda_i = \lambda_1 \lambda_2 \cdots \lambda_n .$$

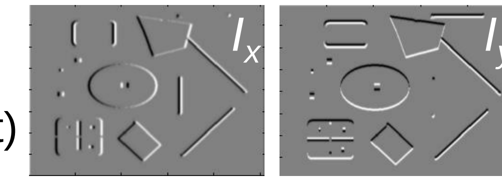
- $R = \lambda_1\lambda_2 - \alpha(\lambda_1+\lambda_2)^2 = \det(M) - \alpha \operatorname{trace}(M)^2$

Harris Detector [Harris88]

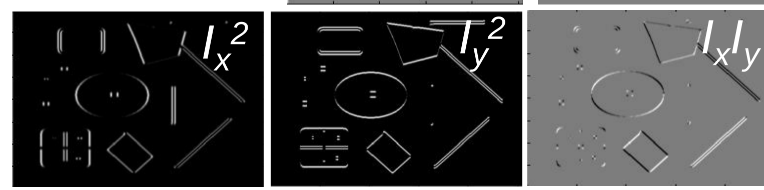
$$M(\sigma_I,\sigma_D) = g(\sigma_I) * \begin{bmatrix} I_x^2(\sigma_D) & I_x I_y(\sigma_D) \\ I_x I_y(\sigma_D) & I_y^2(\sigma_D) \end{bmatrix}$$



1. Image derivatives (optionally, blur first)



2. Square of derivatives



$$\det M = \lambda_1 \lambda_2$$
$$\operatorname{trace} M = \lambda_1 + \lambda_2$$

3. Gaussian filter $g(\sigma_I)$

4. Cornerness function – both eigenvalues are strong

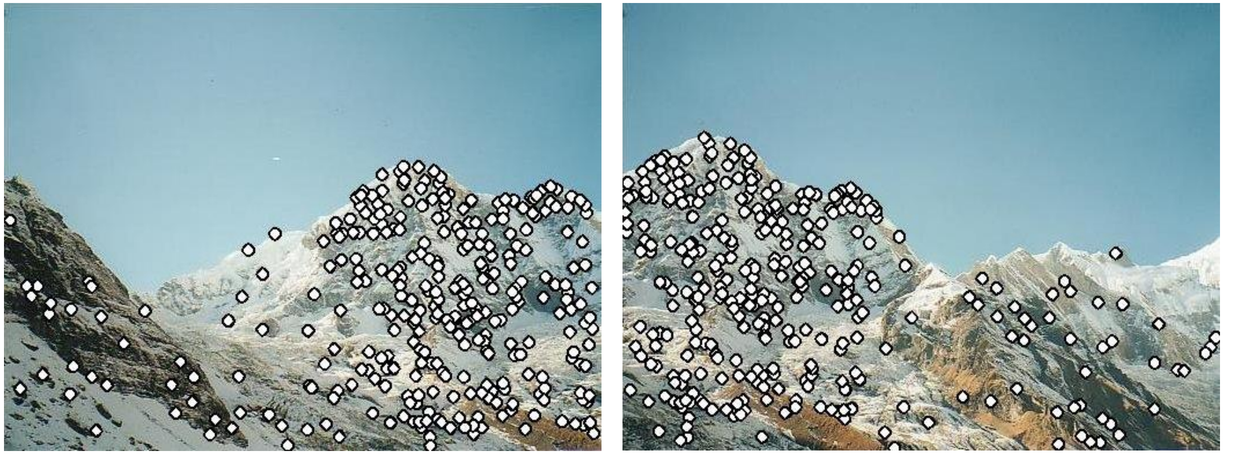
$$har = \det[M(\sigma_I,\sigma_D)] - \alpha[\operatorname{trace}(M(\sigma_I,\sigma_D))^2]$$

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

5. Non-maxima suppression

James Hays

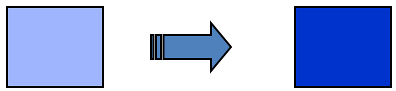
Characteristics of good features



- Repeatability
  - The same feature can be found in several images despite geometric and photometric transformations
- Saliency
  - Each feature is distinctive
- Compactness and efficiency
  - Many fewer features than image pixels
- Locality
  - A feature occupies a relatively small area of the image; robust to clutter and occlusion

Kristen Grauman

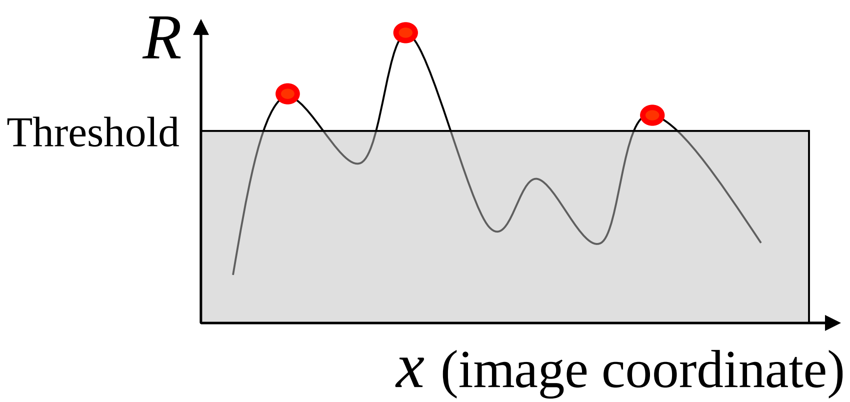
Affine intensity change

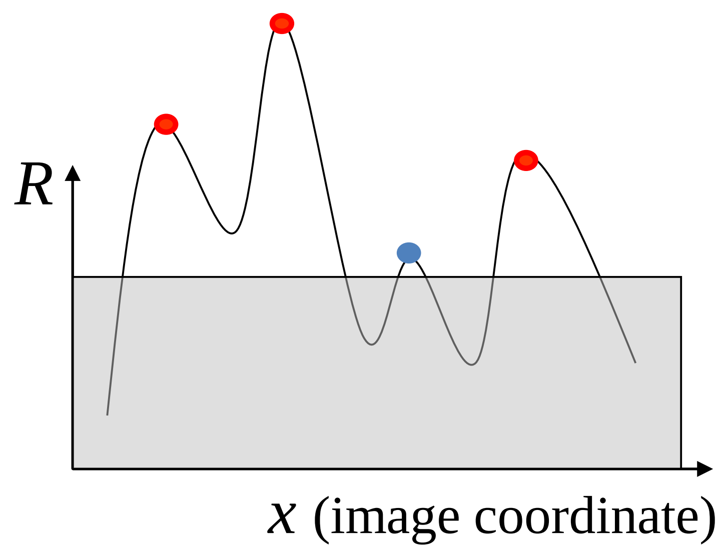


$$I \rightarrow a\,I + b$$

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

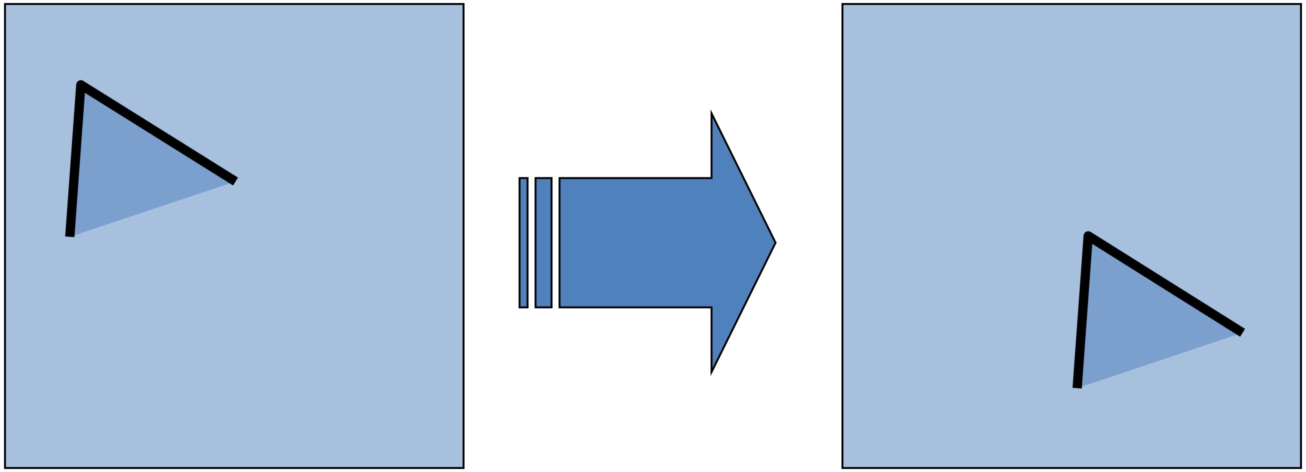
- Intensity scaling: $I \rightarrow a\,I$





Partially invariant to affine intensity change

Image translation

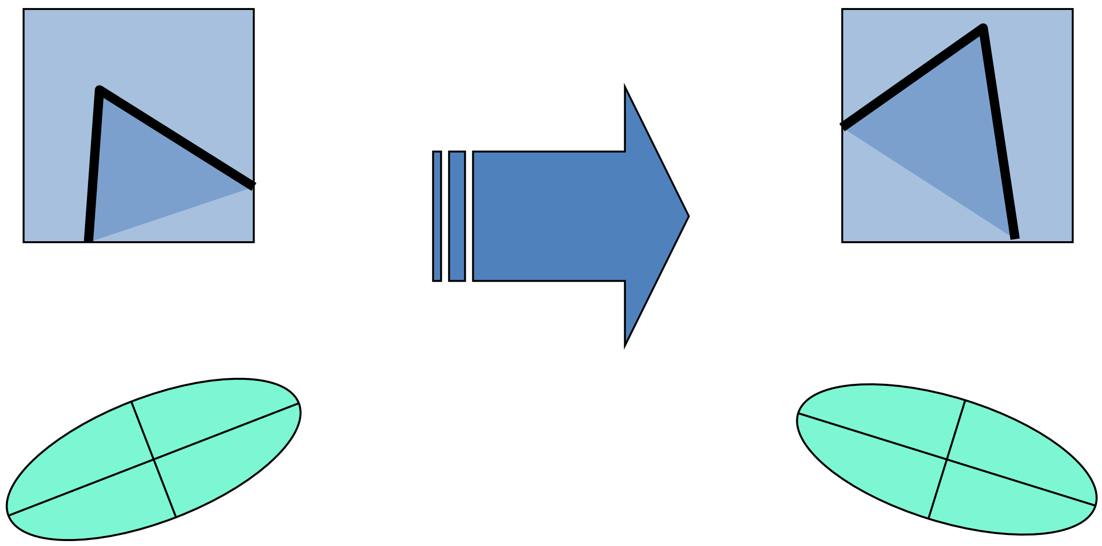


- Derivatives and window function are shift-invariant.

Corner location is covariant w.r.t. translation

James Hays

Image rotation

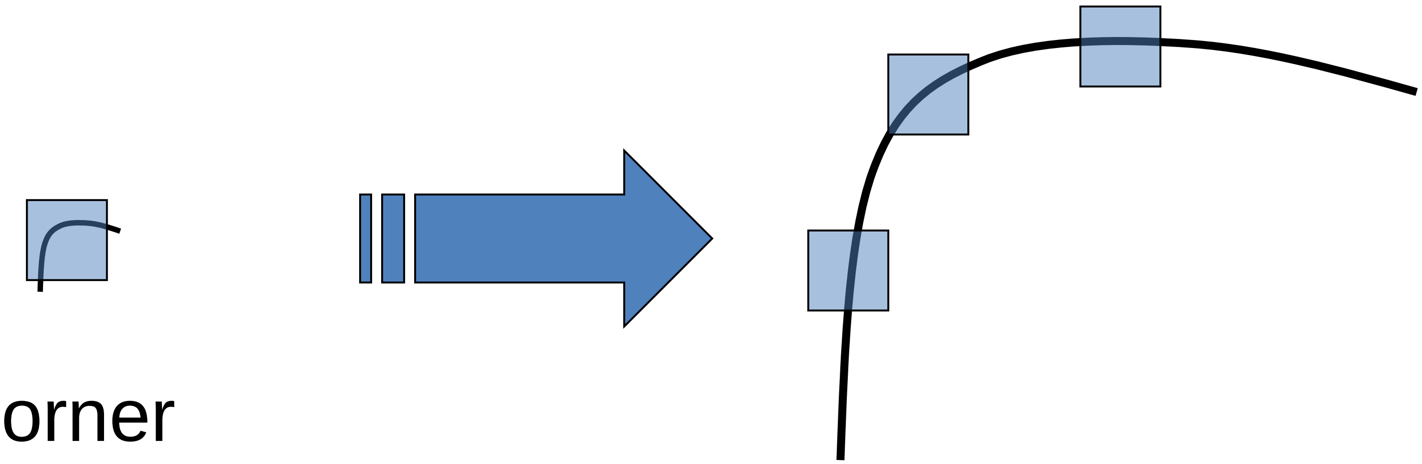


Second moment ellipse rotates but its shape (i.e., eigenvalues) remains the same.

Corner location is covariant w.r.t. rotation

James Hays

Scaling



Corner

All points will be classified as edges

Corner location is not covariant to scaling!

James Hays

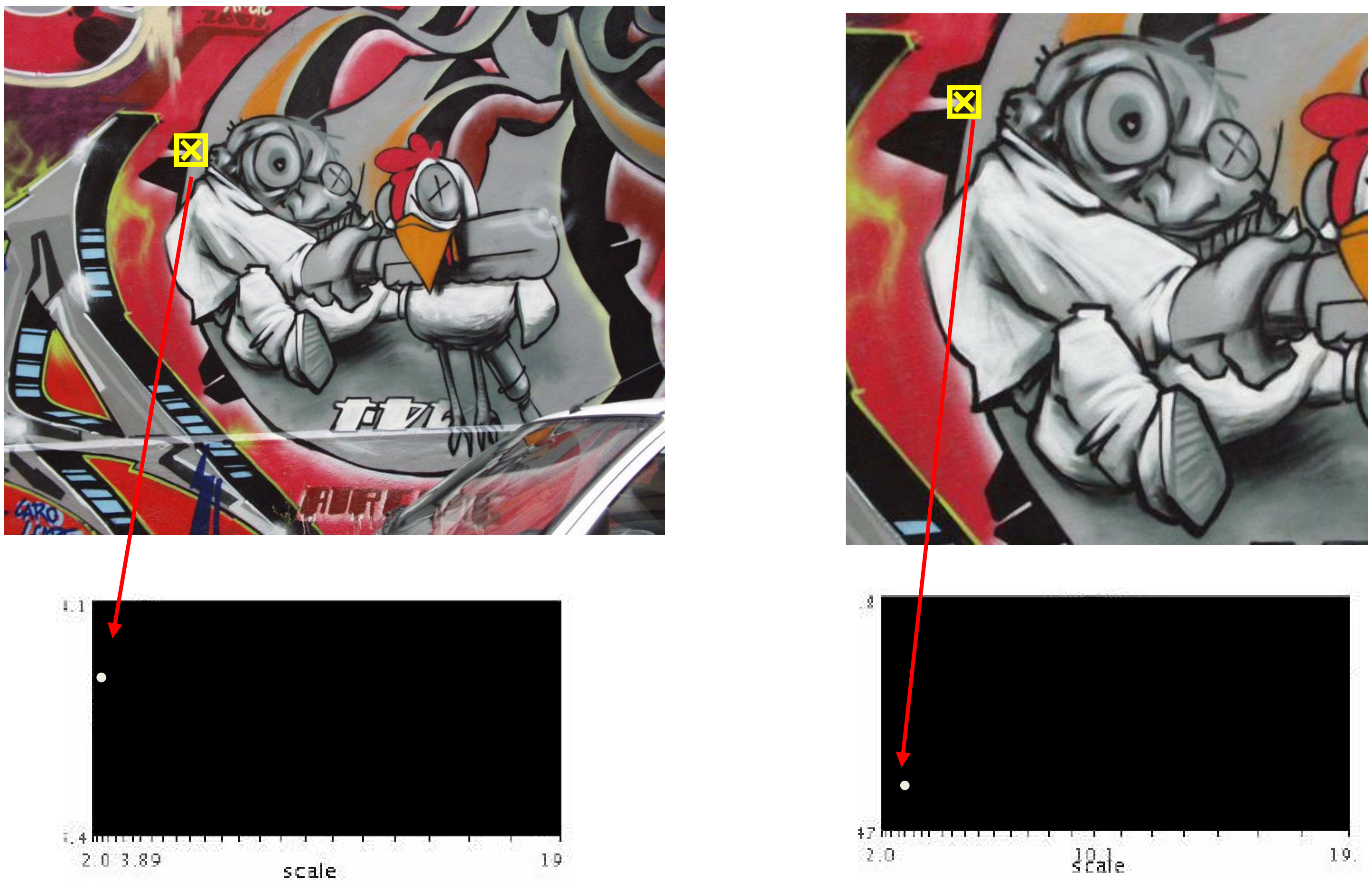
Automatic Scale Selection

$$f(I_{i_1 \ldots i_m}(x, \sigma)) = f(I_{i_1 \ldots i_m}(x', \sigma'))$$

How to find corresponding patch sizes?

Automatic Scale Selection

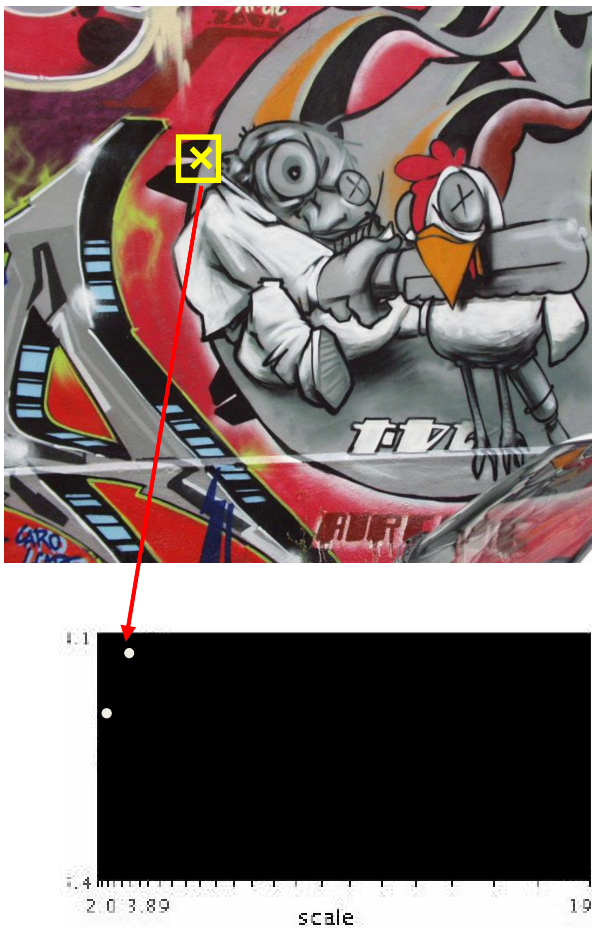
- Function responses for increasing scale (scale signature)



$f(I_{i_1 \dots i_m}(x, \sigma))$ $\qquad$ $f(I_{i_1 \dots i_m}(x', \sigma'))$

Automatic Scale Selection

- Function responses for increasing scale (scale signature)



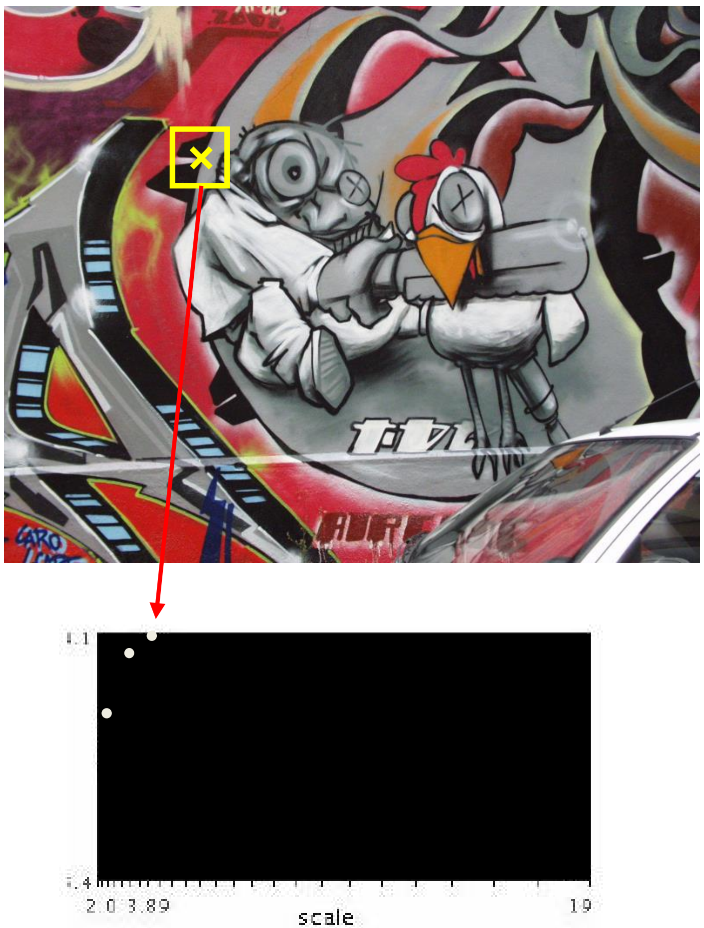
$f(I_{i_1 \dots i_m}(x, \sigma))$

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

K. Grauman, B. Leibe

Automatic Scale Selection

- Function responses for increasing scale (scale signature)



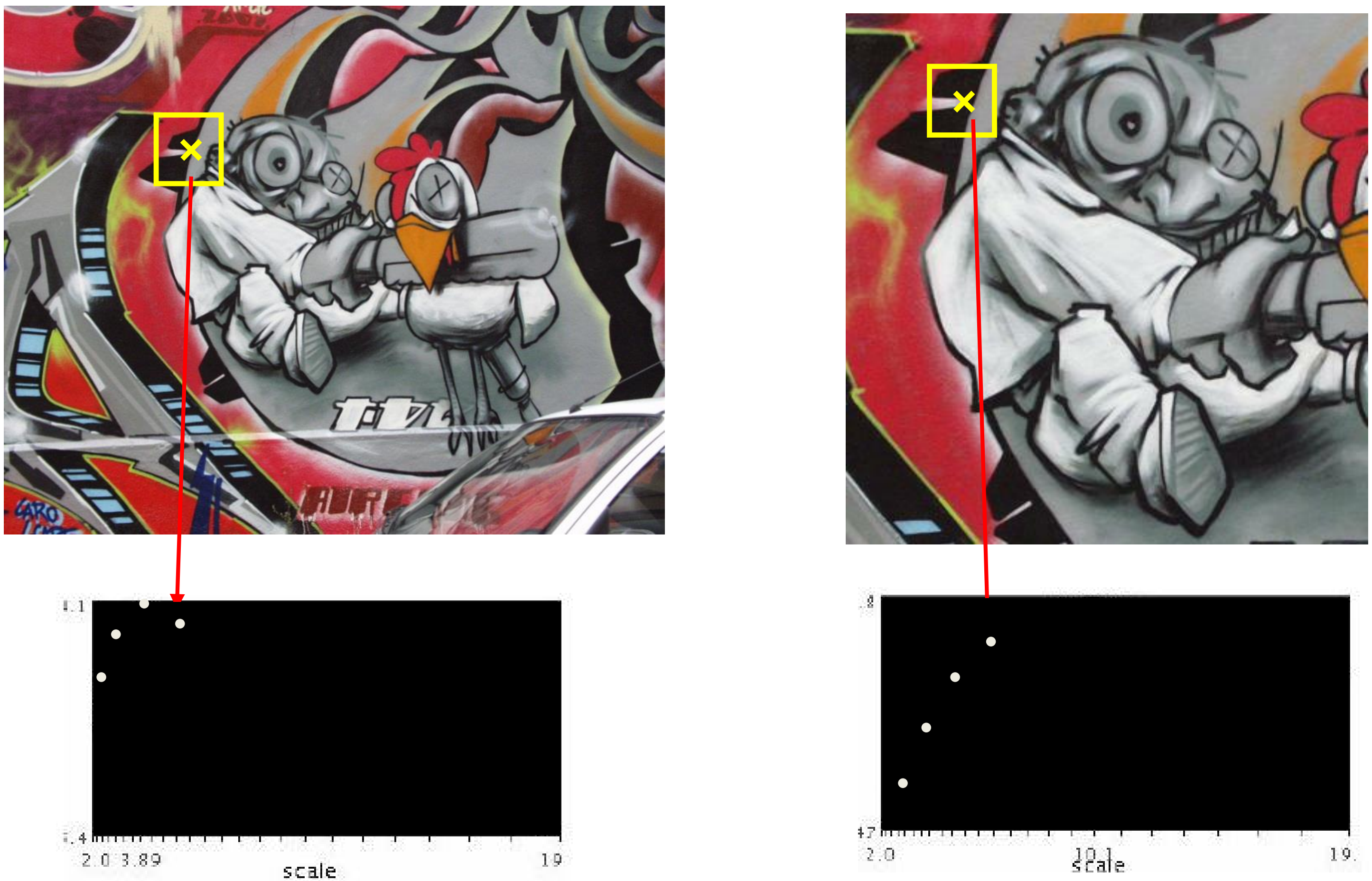
$f(I_{i_1 \ldots i_m}(x, \sigma))$

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

K. Grauman, B. Leibe

Automatic Scale Selection

- Function responses for increasing scale (scale signature)

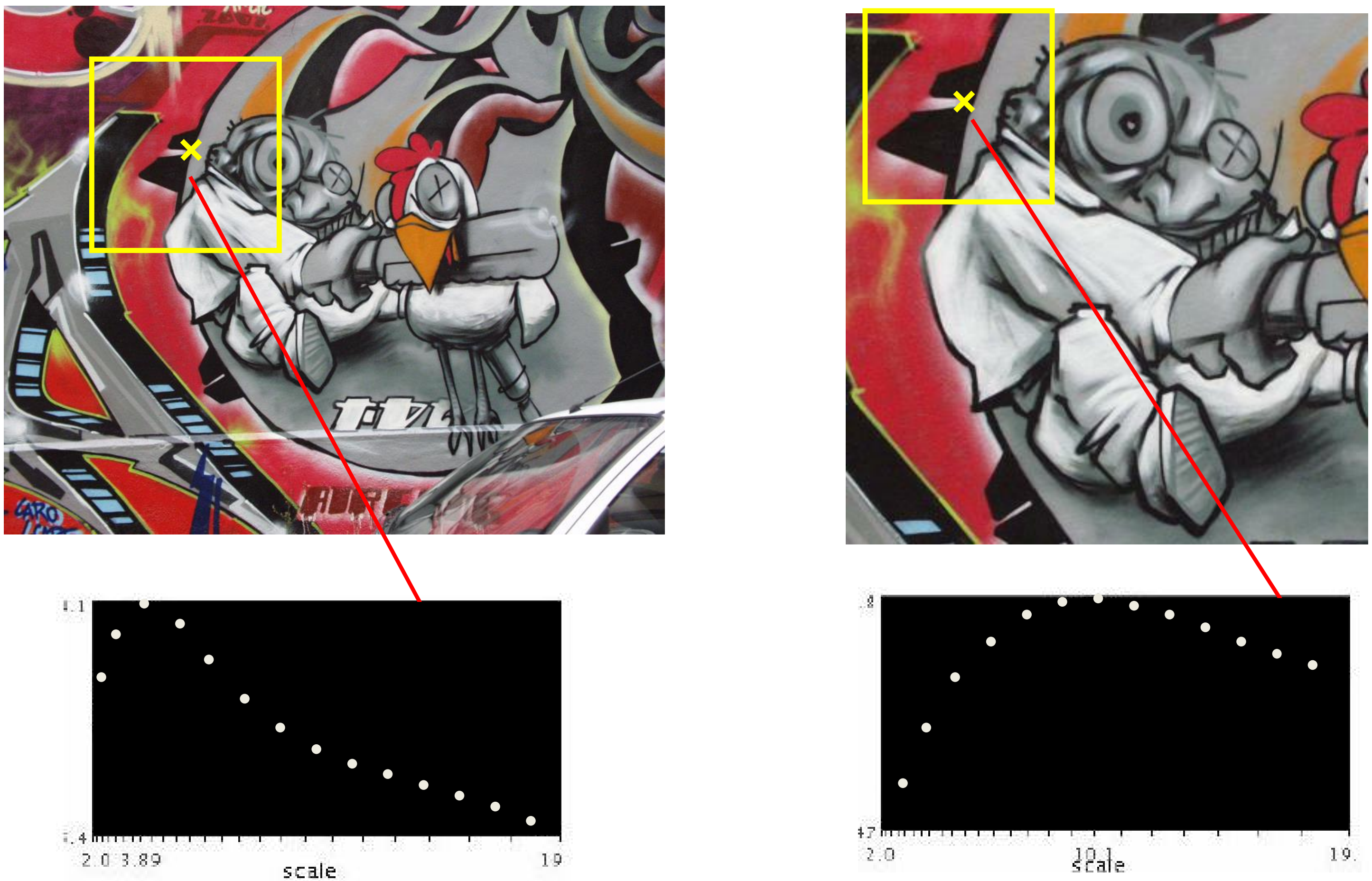


$f(I_{i_1 \ldots i_m}(x, \sigma))$ $\qquad$ $f(I_{i_1 \ldots i_m}(x', \sigma'))$

K. Grauman, B. Leibe

Automatic Scale Selection

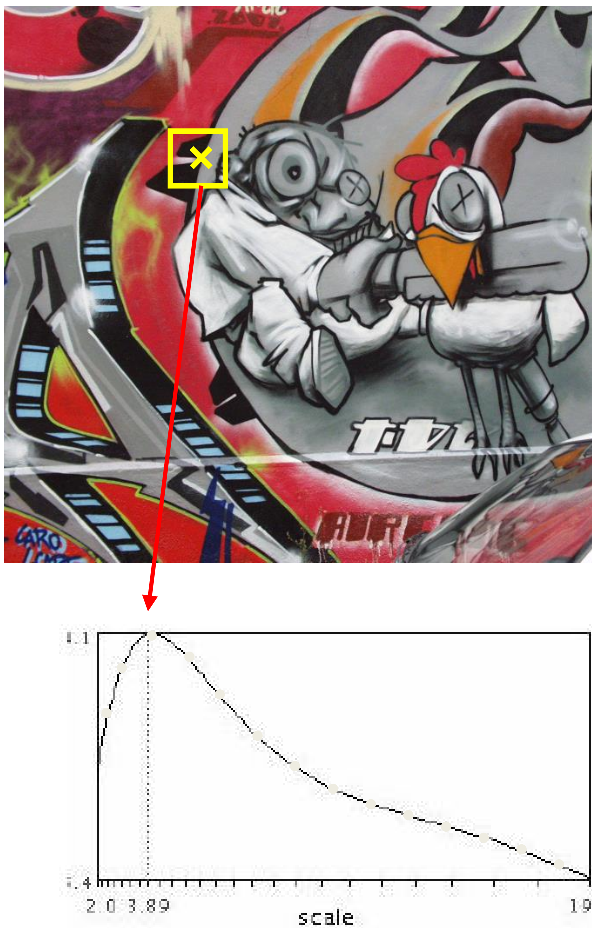
- Function responses for increasing scale (scale signature)



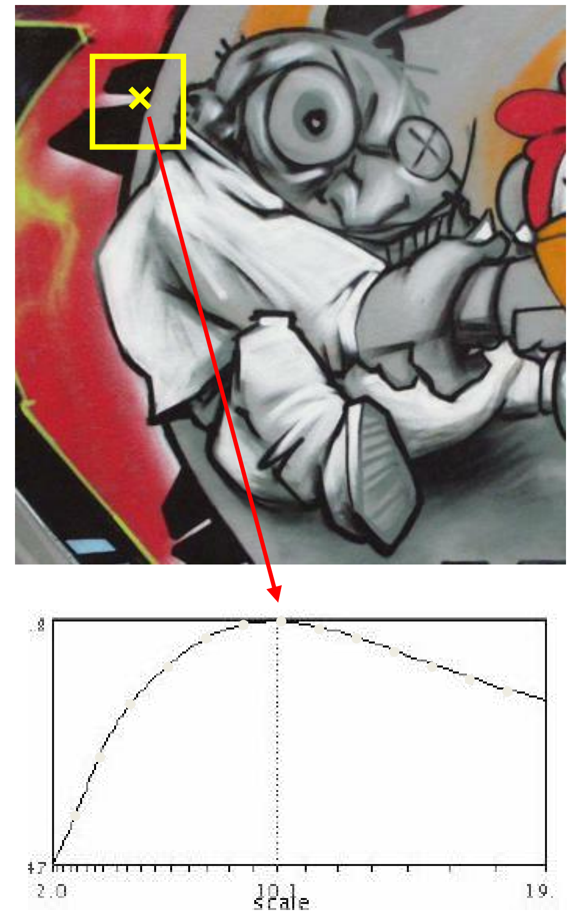
$f(I_{i_1 \ldots i_m}(x, \sigma))$ $\qquad$ $f(I_{i_1 \ldots i_m}(x', \sigma'))$

Automatic Scale Selection

- Function responses for increasing scale (scale signature)



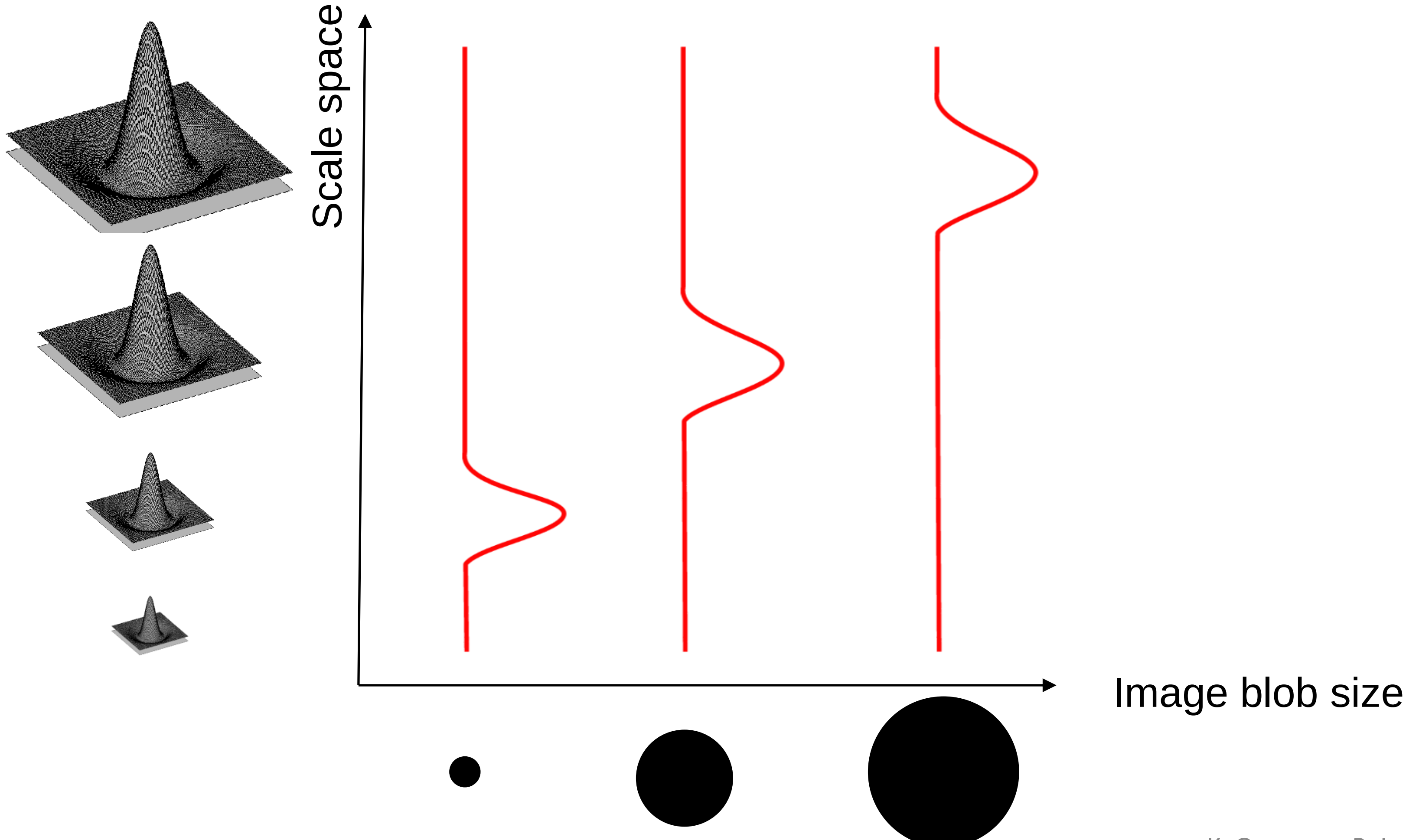
$f(I_{i_1 \ldots i_m}(x, \sigma))$



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

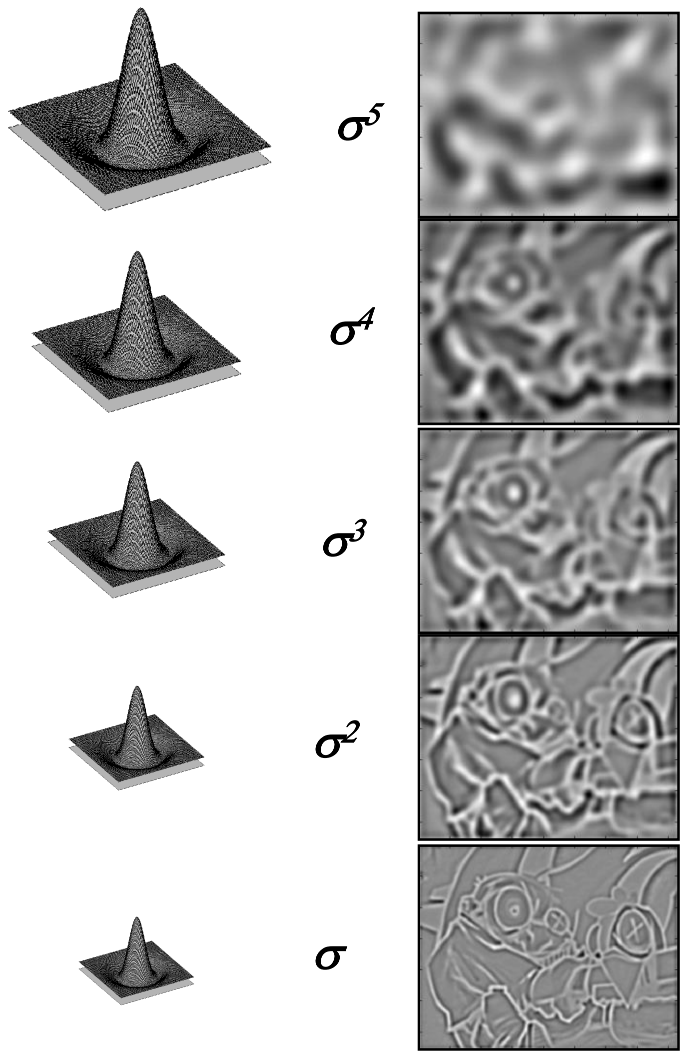
What Is A Useful Signature Function?

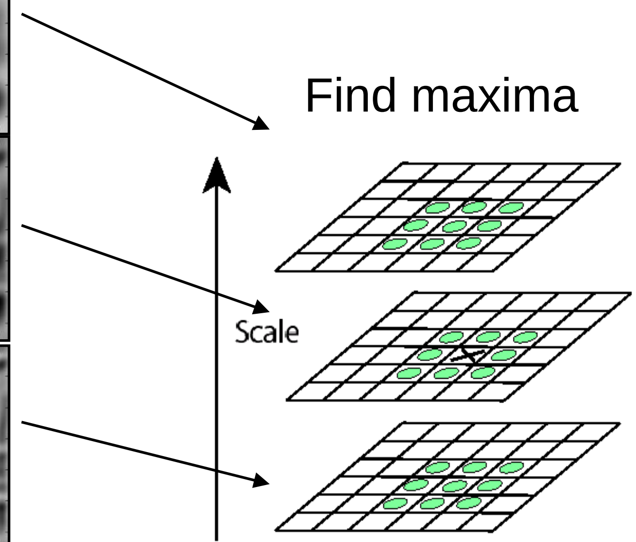
- "Blob" detector
  - Laplacian (2nd derivative) of Gaussian (LoG)



K. Grauman, B. Leibe

Find local maxima in position-scale space

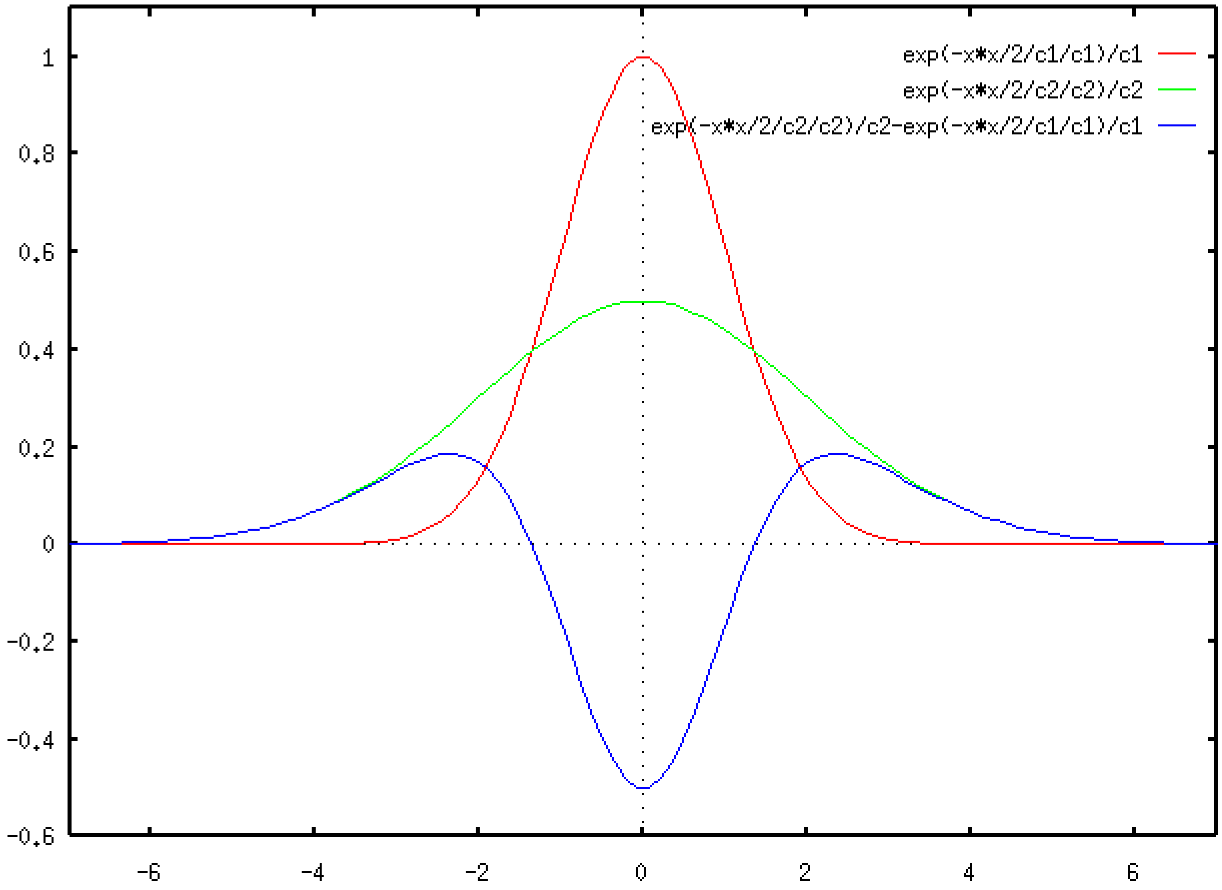




$\Rightarrow$ **List of**
(x, y, s)

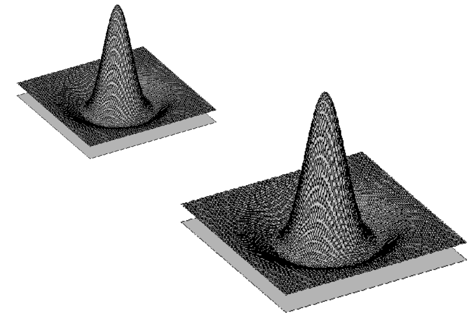
Difference-of-Gaussian (DoG)

Approximate LoG with DoG (!)



Difference-of-Gaussian (DoG)

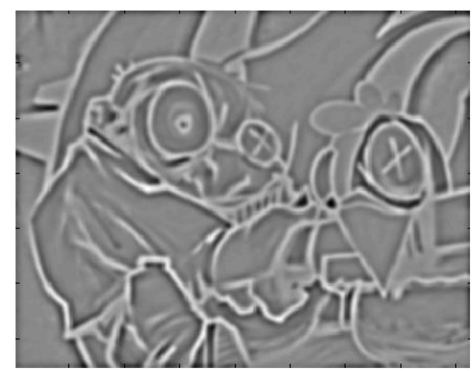
Approximate LoG with DoG (!)



1. Blur image with σ Gaussian kernel
2. Blur image with *k*σ Gaussian kernel
3. Subtract 2. from 1.

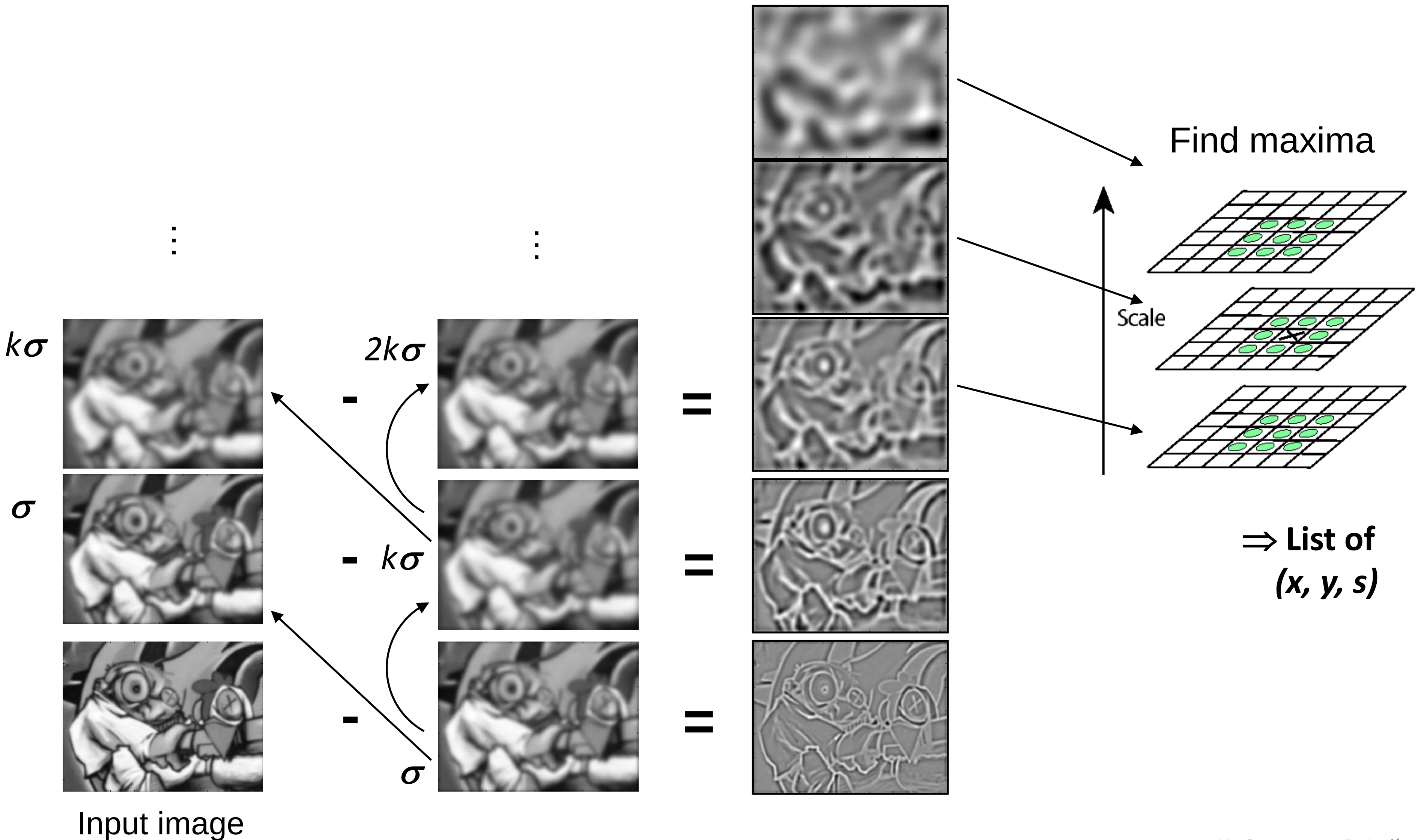
-

=



K. Grauman, B. Leibe

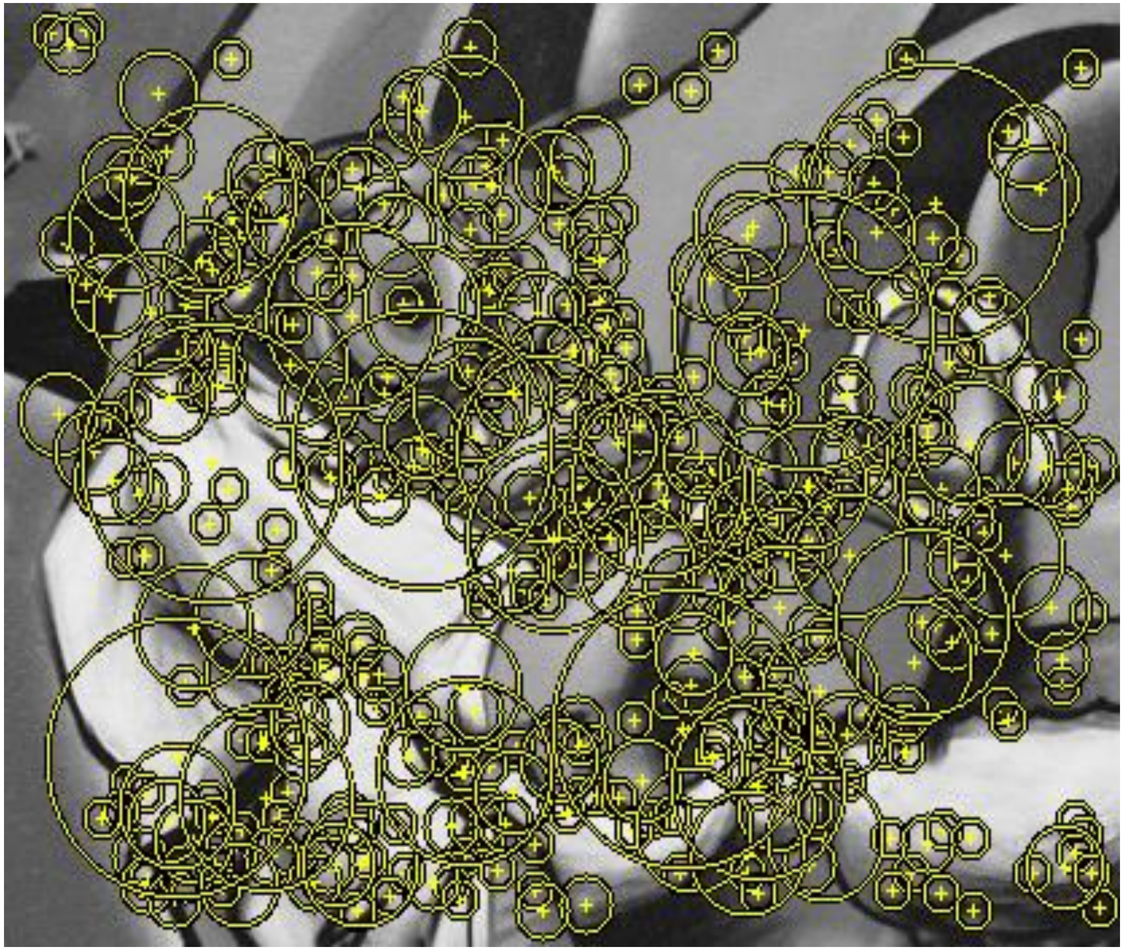
Find local maxima in position-scale space of Difference-of-Gaussian



K. Grauman, B. Leibe

Results: Difference-of-Gaussian

- Larger circles = larger scale
- Descriptors with maximal scale response

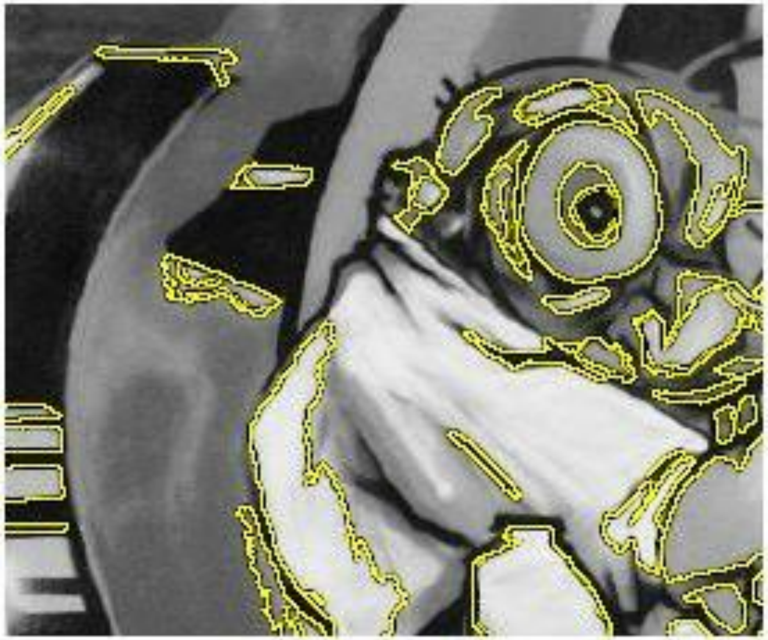


K. Grauman, B. Leibe

Maximally Stable Extremal Regions [Matas '02]

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

Example Results: MSER



Local Image Descriptors

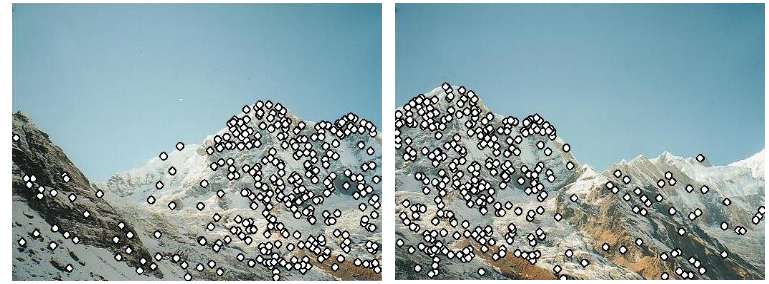
Read Szeliski 4.1

Acknowledgment: Many slides from James Hays, Derek Hoiem and Grauman & Leibe 2008 AAAI Tutorial

Local features: main components

1) Detection:

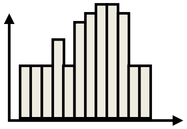
Find a set of distinctive key points.



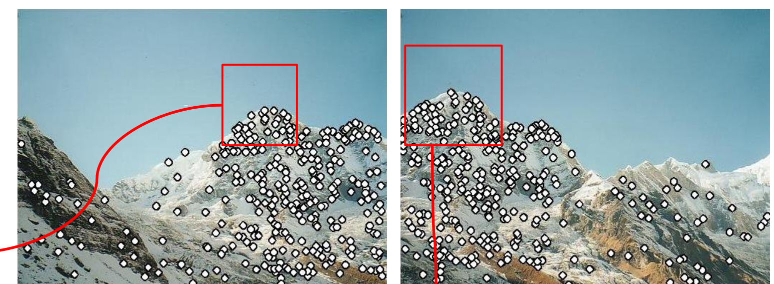
2) Description:

Extract feature descriptor around each interest point as vector.

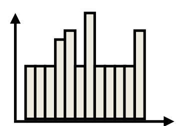
x_1



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



x_2

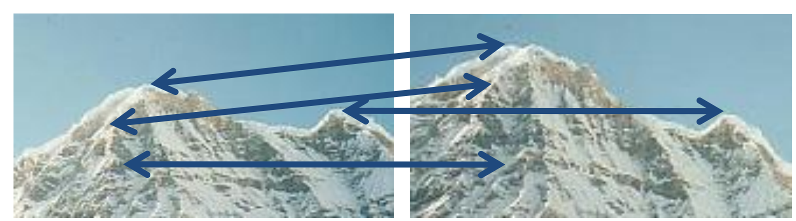


$$\mathbf{x}_2 = [x_1^{(2)}, \ldots, x_d^{(2)}]$$

3) Matching:

Compute distance between feature vectors to find correspondence.

$$d(\mathrm{x}_1, \mathrm{x}_2) < T$$



K. Grauman, B. Leibe

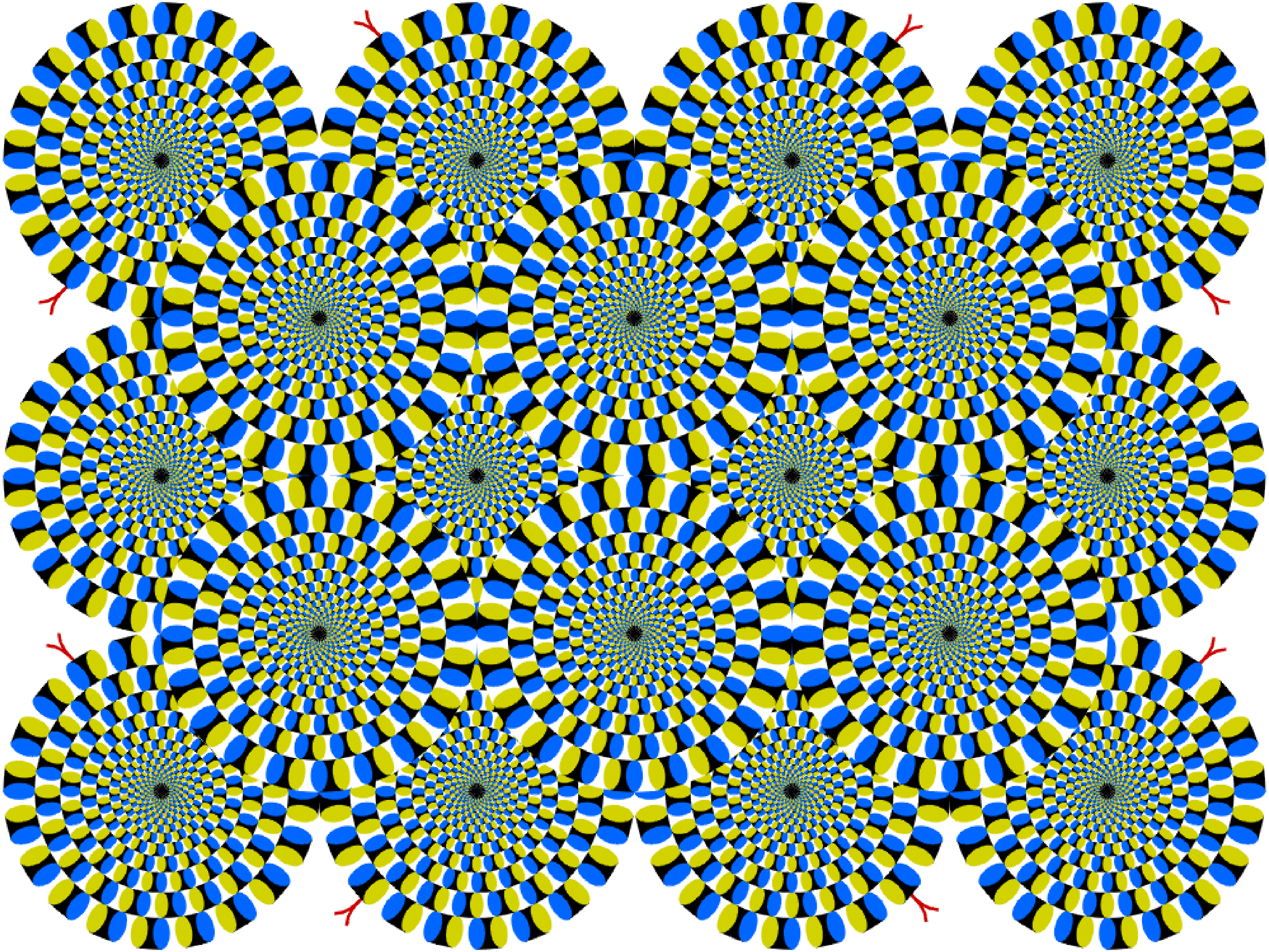
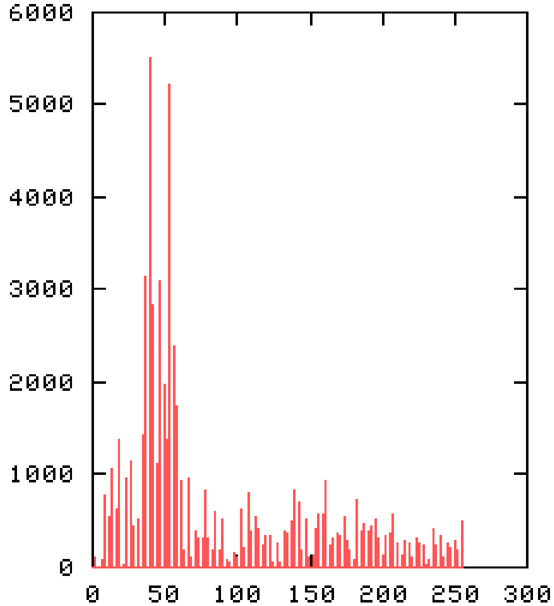
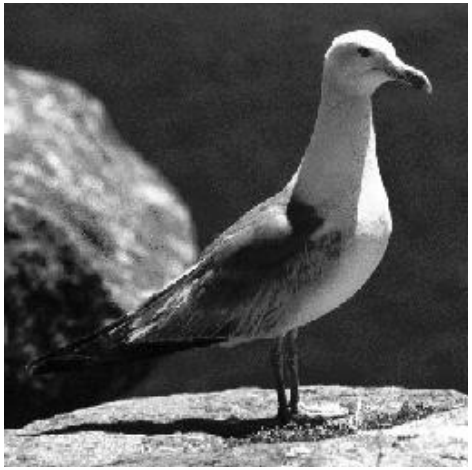


Image Representations: Histograms





Global histogram to represent distribution of features

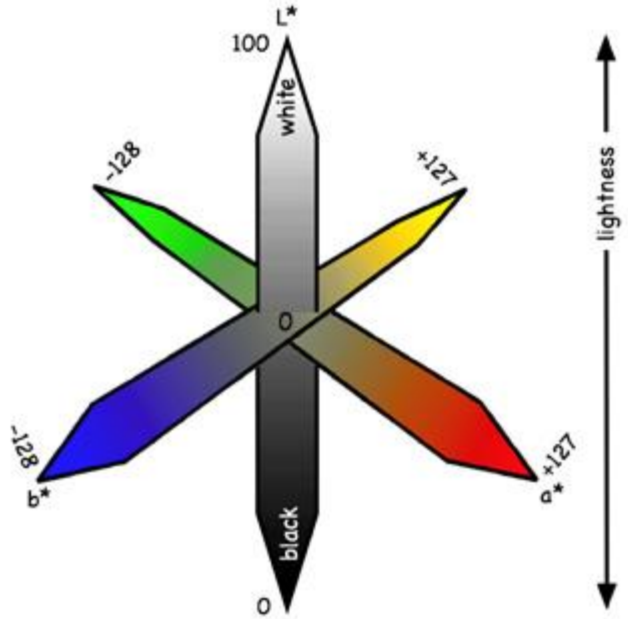
- Color, texture, depth, …

Local histogram per detected point

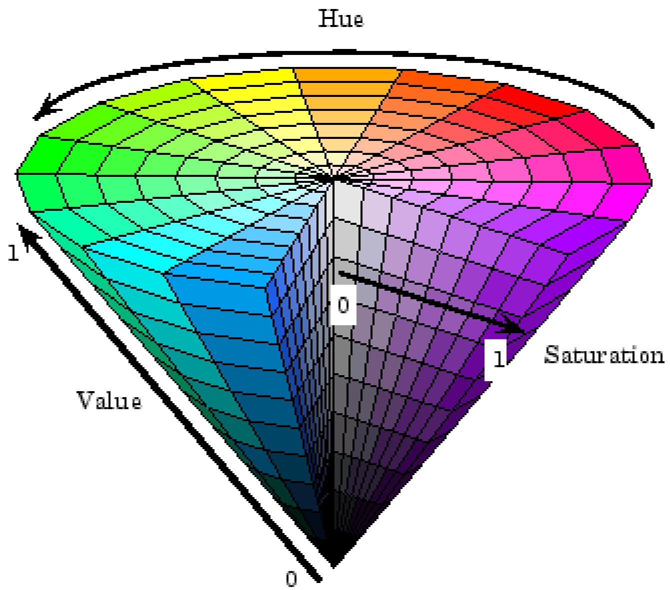
Images from Dave Kauchak

For what things do we compute histograms?

- Color



L*a*b* color space

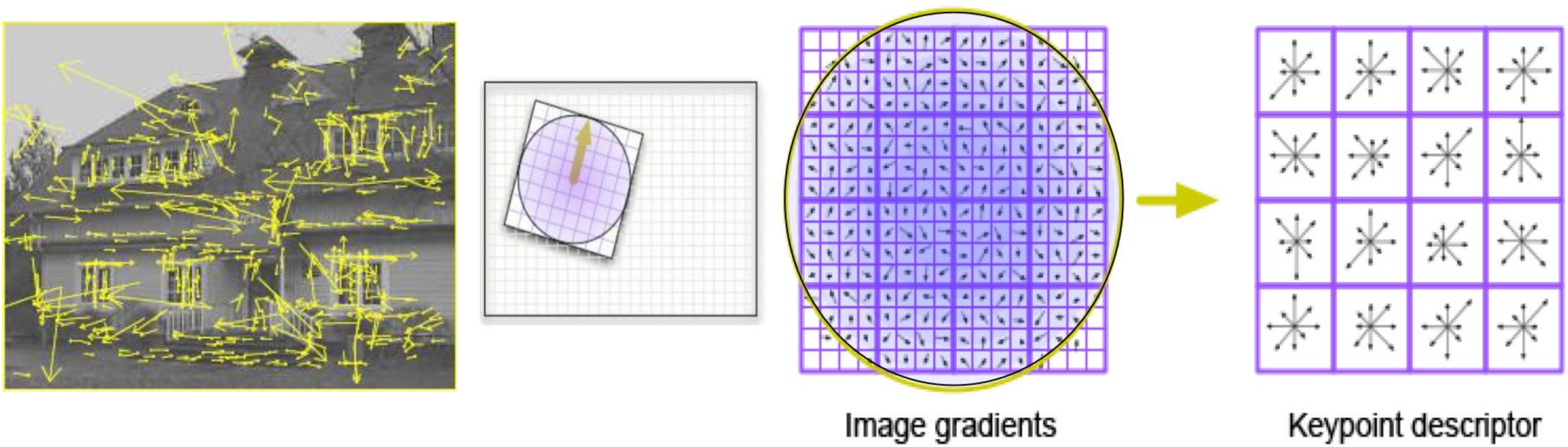


HSV color space

- Model local appearance

For what things do we compute histograms?

- Texture
- Local histograms of oriented gradients
- SIFT: Scale Invariant Feature Transform
  - Extremely popular (40k citations)



SIFT – Lowe IJCV 2004

James Hays

SIFT

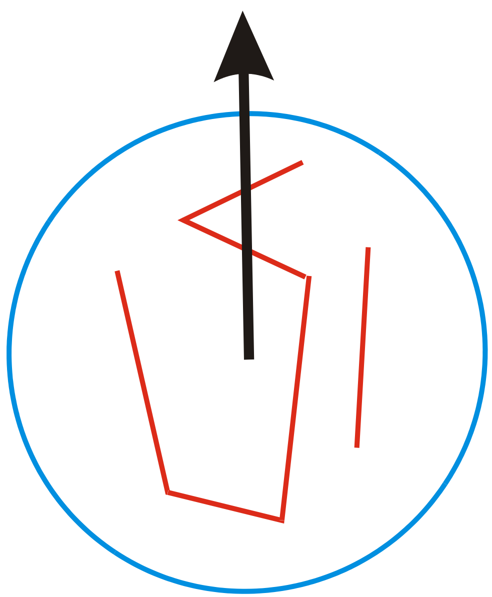
- Find Difference of Gaussian scale-space extrema
- Post-processing
  - Position interpolation
  - Discard low-contrast points
  - Eliminate points along edges

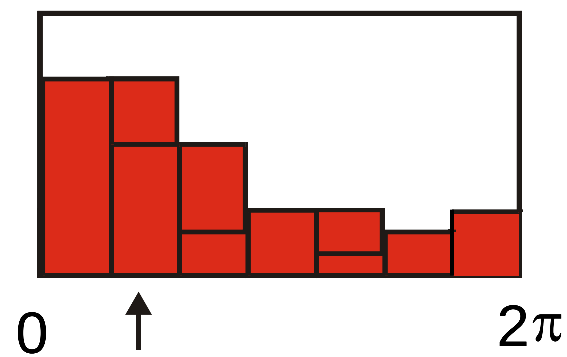
SIFT

- Find Difference of Gaussian scale-space extrema
- Post-processing
  - Position interpolation
  - Discard low-contrast points
  - Eliminate points along edges
- Orientation estimation

SIFT Orientation Normalization

- Compute orientation histogram
- Select dominant orientation Θ
- Normalize: rotate to fixed orientation





[Lowe, SIFT, 1999]

SIFT

- Find Difference of Gaussian scale-space extrema
- Post-processing
  - Position interpolation
  - Discard low-contrast points
  - Eliminate points along edges
- Orientation estimation
- Descriptor extraction
  - Motivation: We want some sensitivity to spatial layout, but not too much, so blocks of histograms give us that.

SIFT descriptor formation

- Compute on local 16 x 16 window around detection.
- Rotate and scale window according to discovered orientation Θ and scale σ (gain invariance).
- Compute gradients weighted by a Gaussian of variance half the window (for smooth falloff).

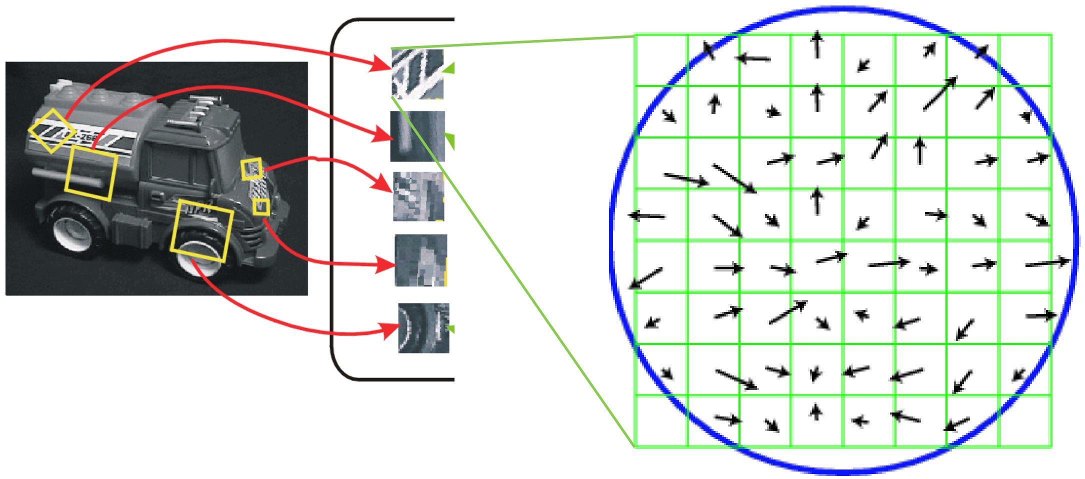


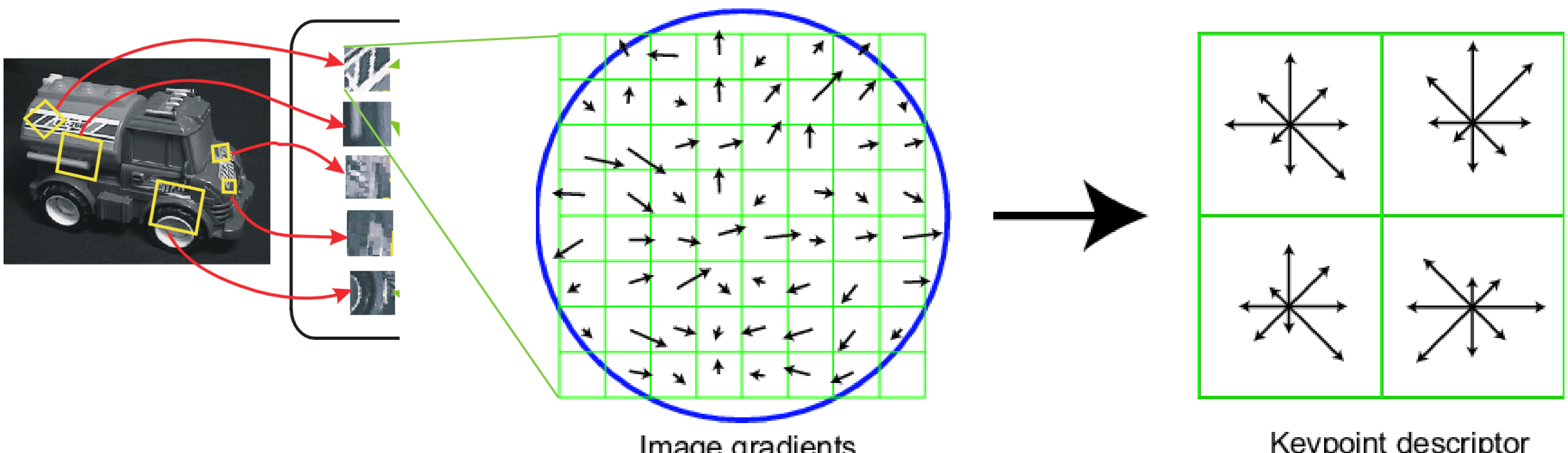
Image gradients

Actually 16x16, only showing 8x8

James Hays

SIFT vector formation

- 4x4 array of gradient orientation histograms weighted by gradient magnitude.
- Bin into 8 orientations x 4x4 array = 128 dimensions.

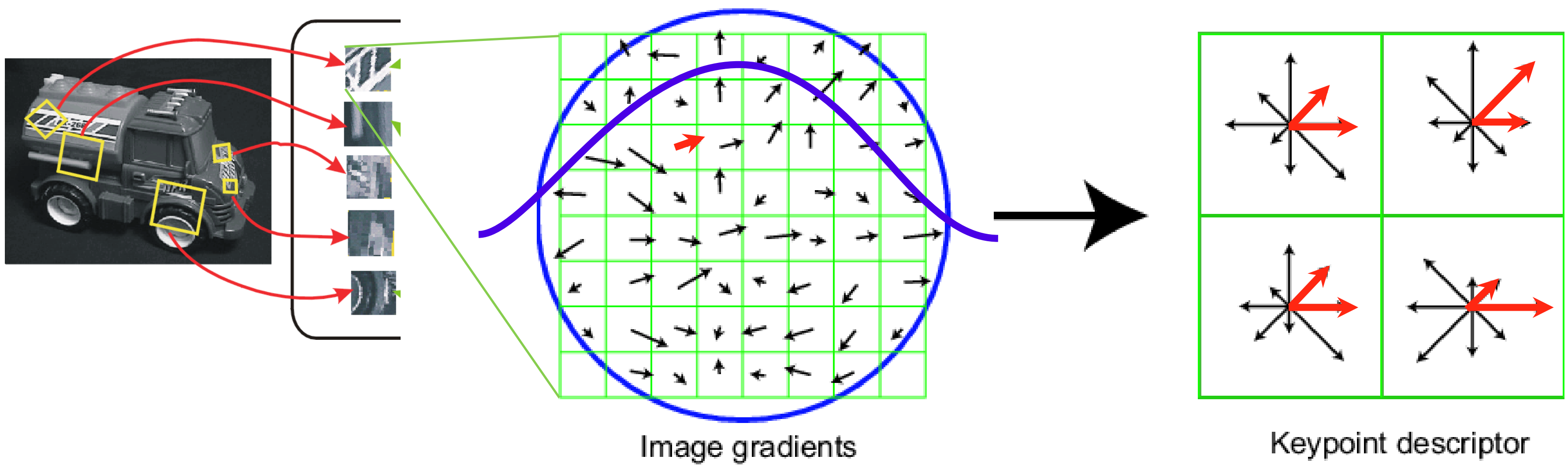


Showing only 2x2 here but is 4x4

James Hays

Ensure smoothness

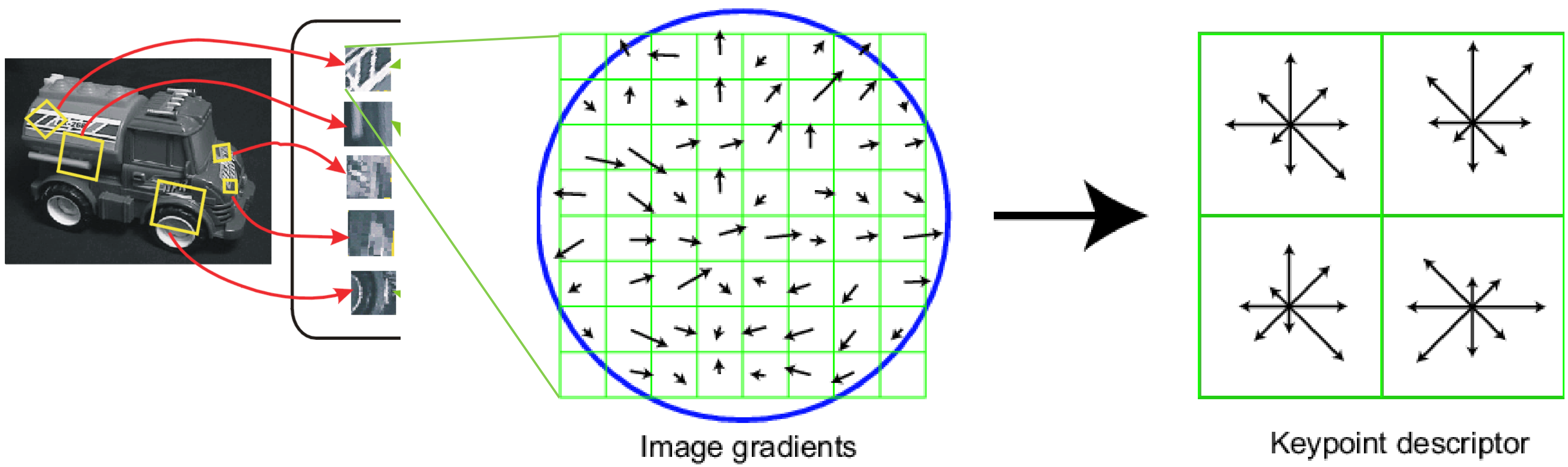
- Gaussian weight the magnitudes
- Trilinear interpolation
  - A given gradient contributes to 8 bins: 4 in space x 2 in orientation



James Hays

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



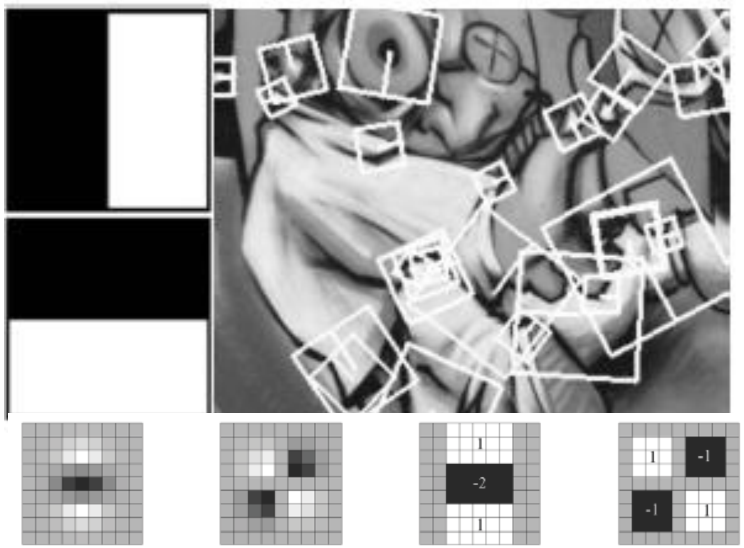
James Hays

SIFT-like descriptor in Project 2

- SIFT is hand designed based on intuition
- You implement your own SIFT-like descriptor

- May be some trial and error
- Feel free to look at papers / resources for inspiration

Local Descriptors: SURF



Fast approximation of SIFT idea

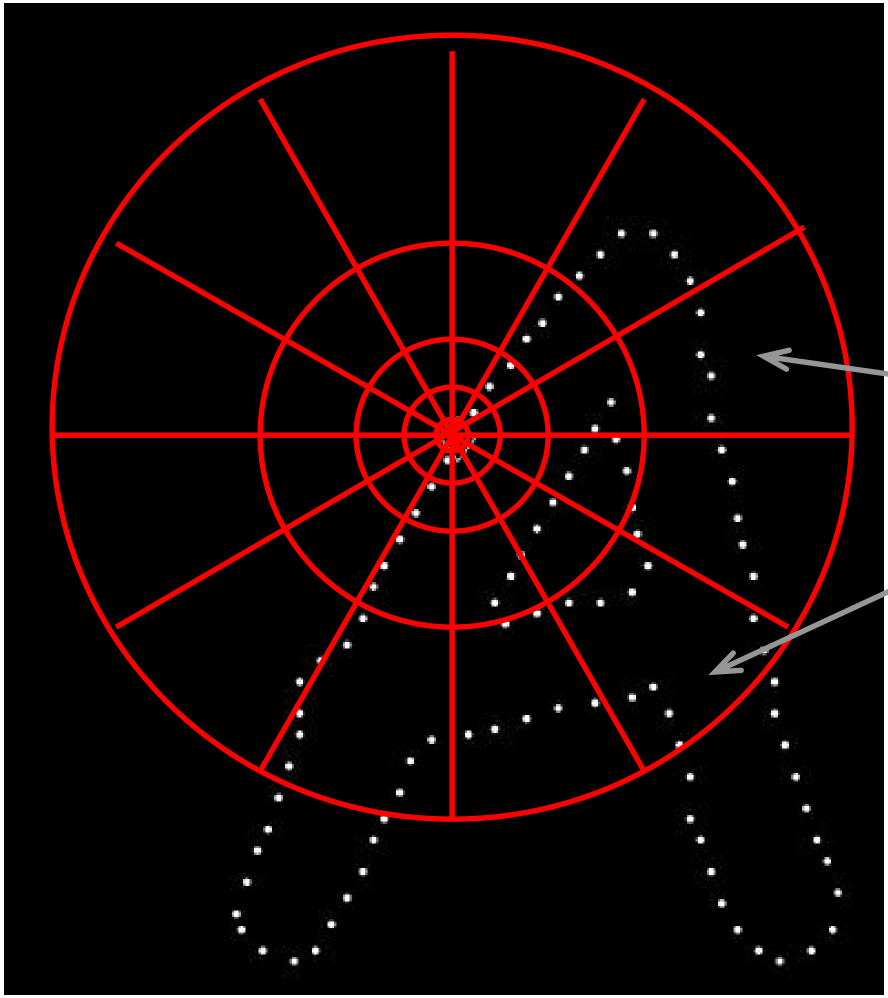
- **Efficient computation by 2D box filters & integral images**
  $\Rightarrow$ 6 times faster than SIFT
- **Equivalent quality for object identification**

GPU implementation available

- **Feature extraction @ 200Hz (detector + descriptor, 640×480 img)**
- http://www.vision.ee.ethz.ch/~surf

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

Local Descriptors: Shape Context



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

Count = 4

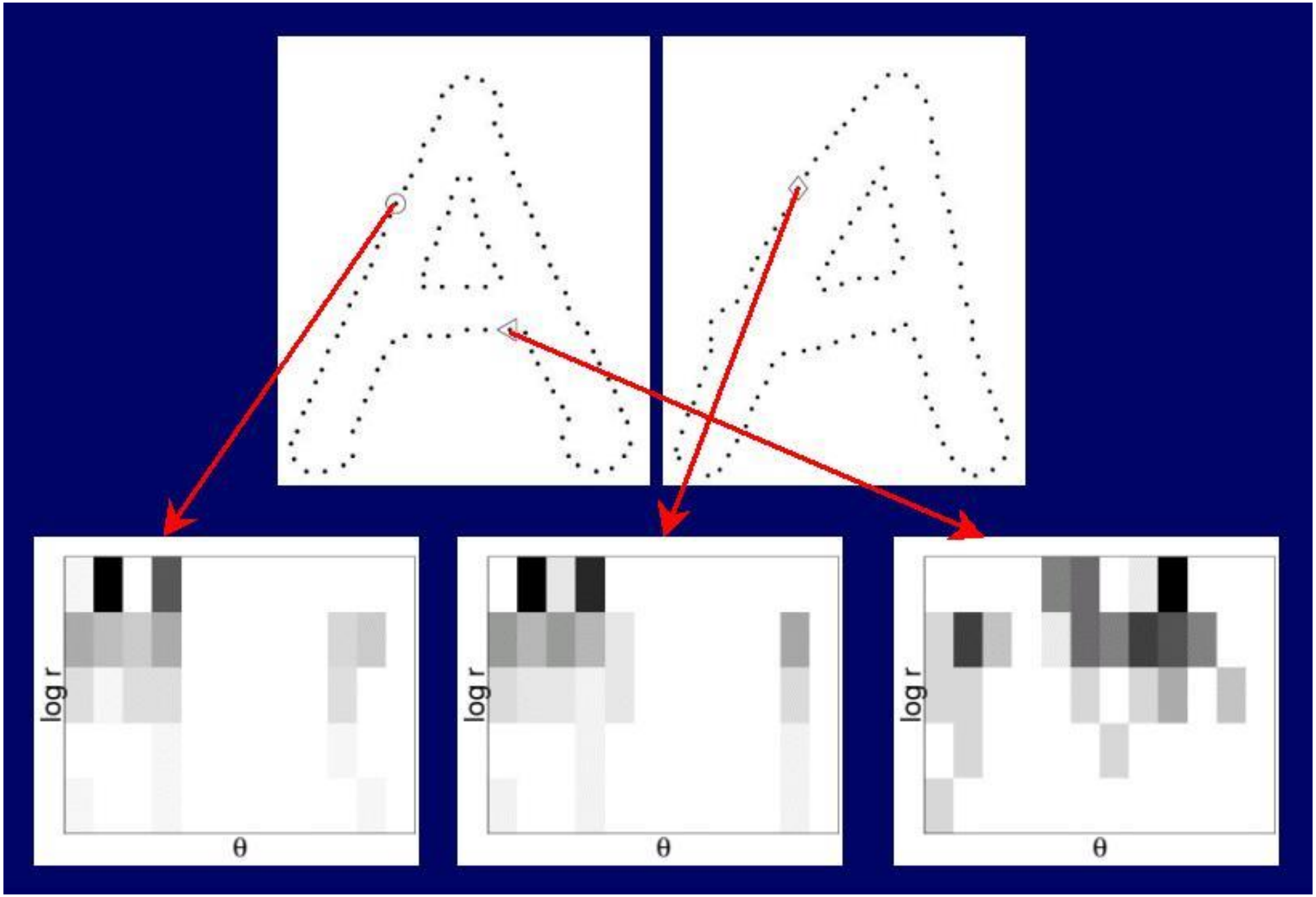
⋮

Count = 10

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

Belongie & Malik, ICCV 2001

Shape Context Descriptor



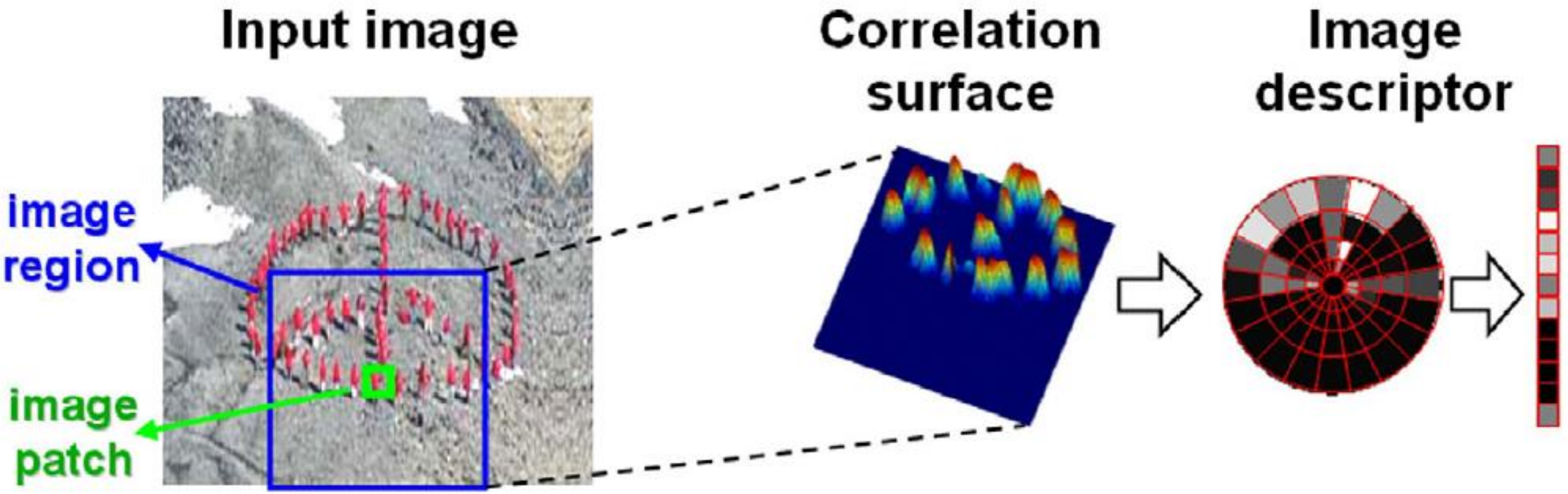
Self-similarity Descriptor

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

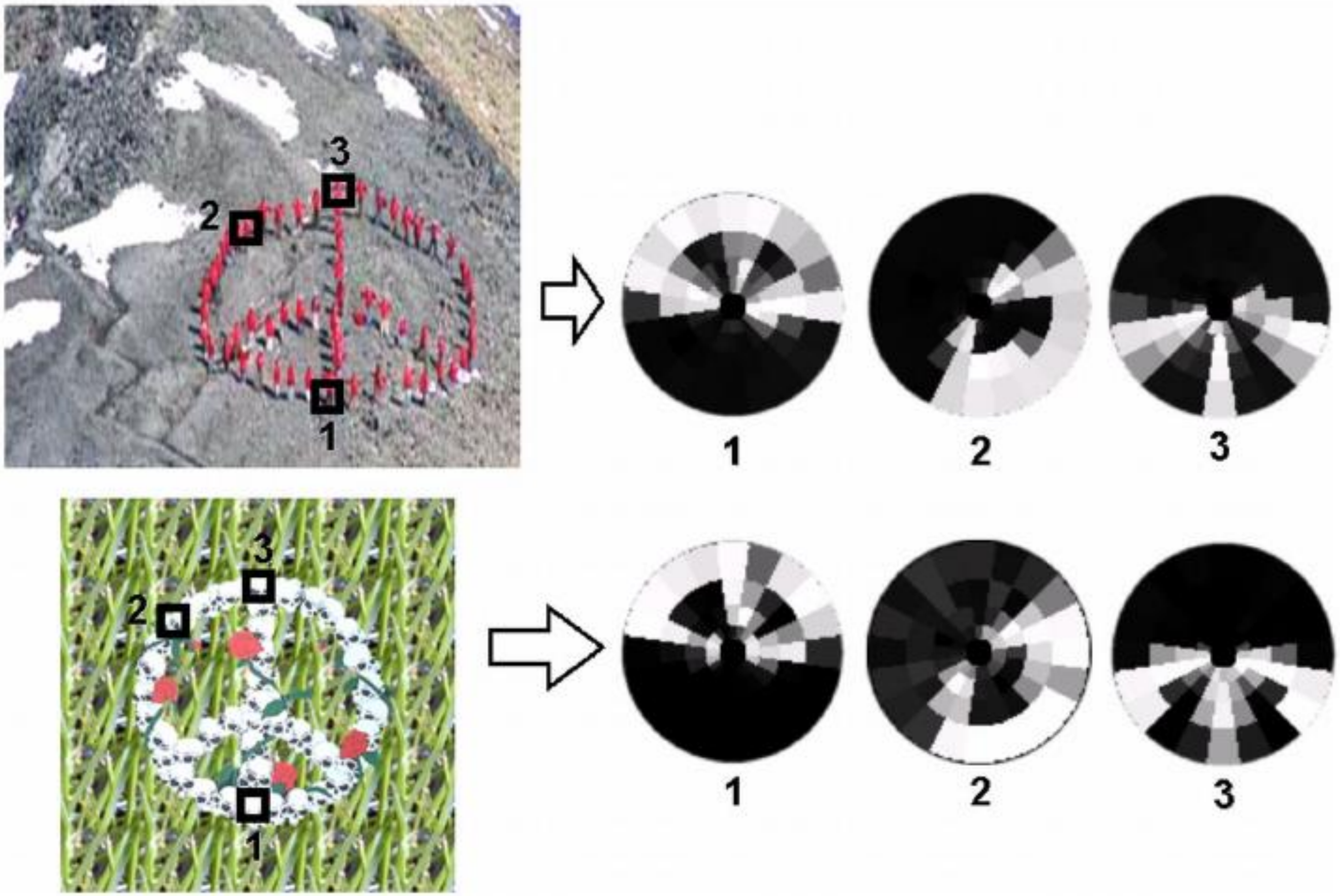
James Hays

Self-similarity Descriptor



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

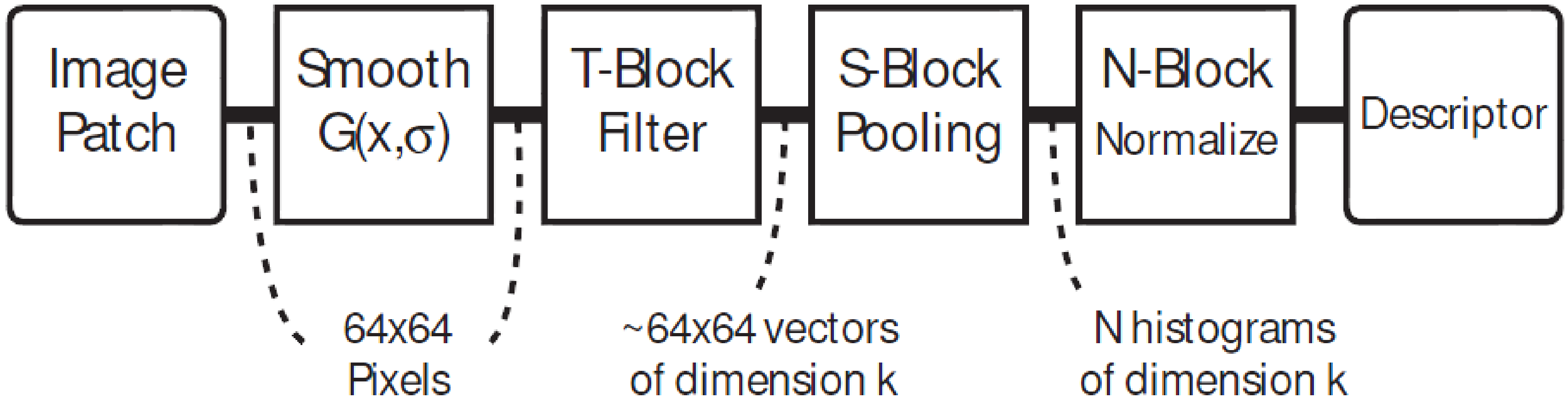
Self-similarity Descriptor

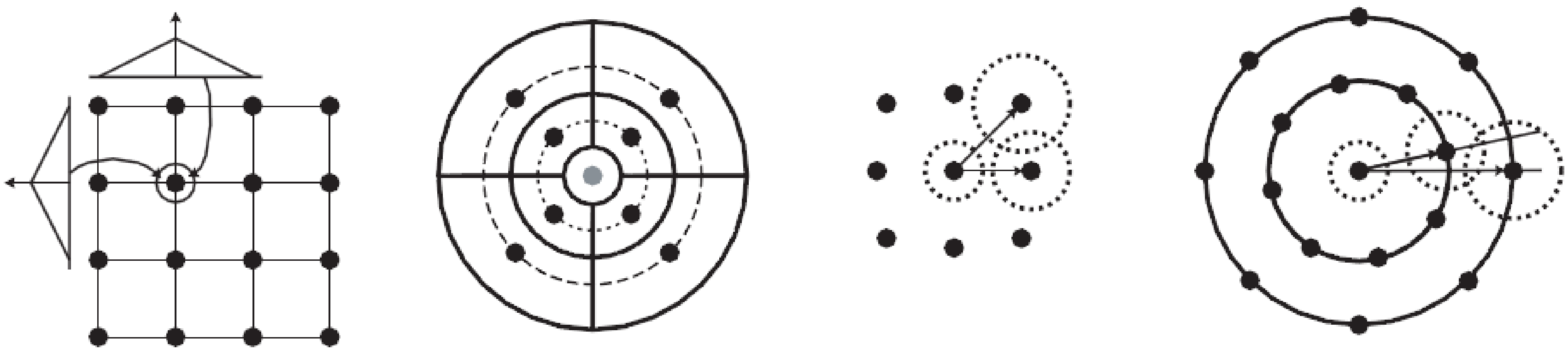


Matching Local Self-Similarities across Images and Videos, Shechtman and Irani, 2007

James Hays

Learning Local Image Descriptors
Winder and Brown, 2007





S1: SIFT grid with bilinear weights

S2: GLOH polar grid with bilinear radial and angular weights

S3: 3x3 grid with Gaussian weights

S4: 17 polar samples with Gaussian weights

Local Descriptors

- Most features can be thought of as templates, histograms (counts), or combinations
- The ideal descriptor should be
  - Robust
  - Distinctive
  - Compact
  - Efficient
- Most available descriptors focus on edge/gradient information
  - Capture texture information
  - Color rarely used

Available at a web site near you…

- Many local feature detectors have executables 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

Next time

- Feature matching, Hough Transform