Local Image Features

Read Szeliski 4.1

Computer Vision

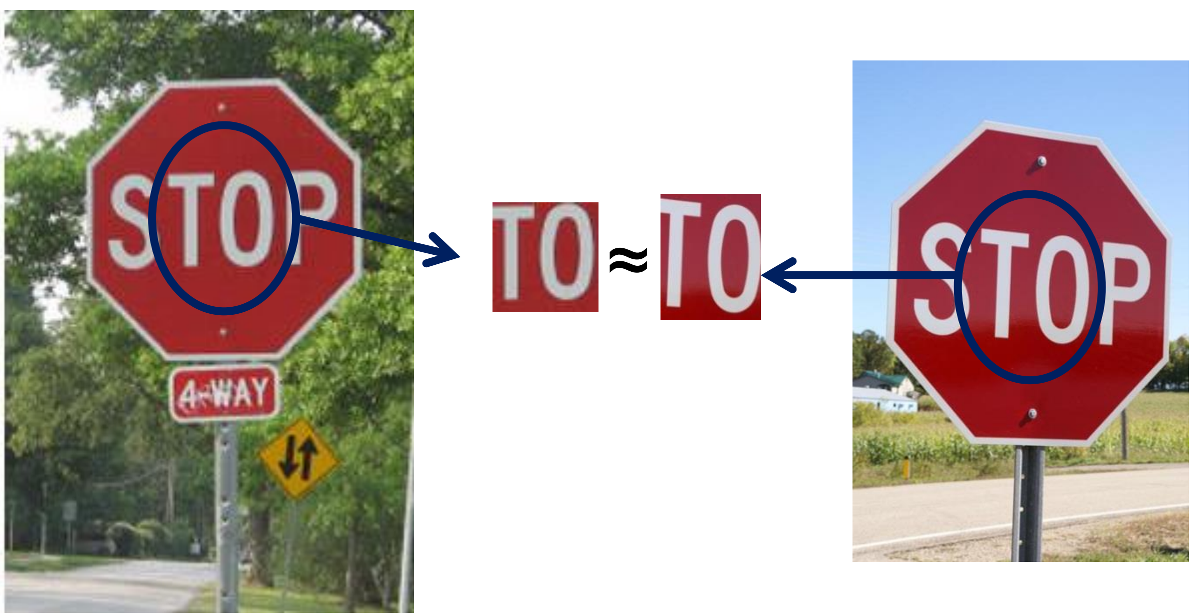
CS 143, Brown

James Hays

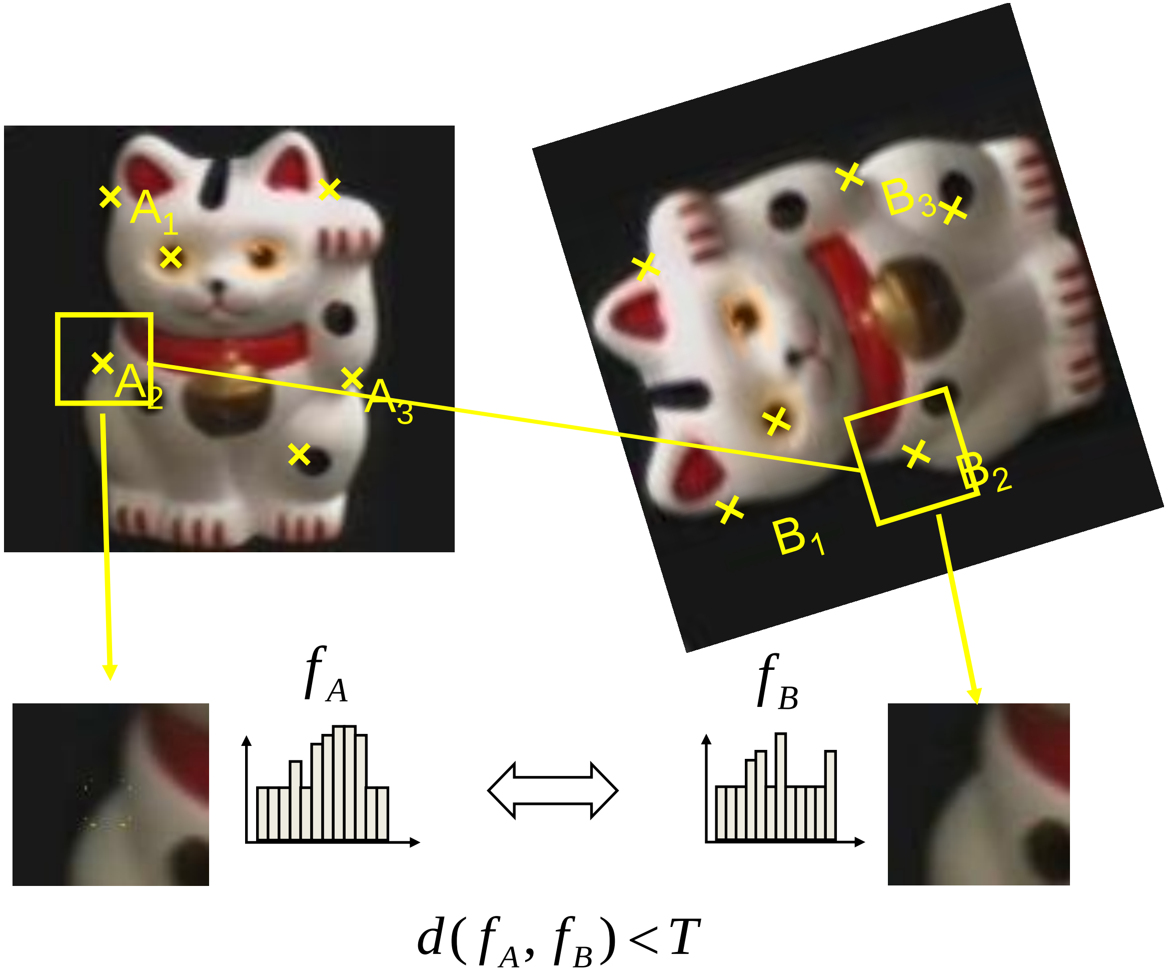
Acknowledgment: Many slides from Derek Hoiem and Grauman&Leibe 2008 AAAI Tutorial

This section: correspondence and alignment

- Correspondence: matching points, patches, edges, or regions across images



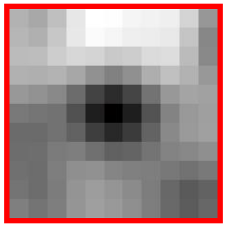
Overview of Keypoint Matching

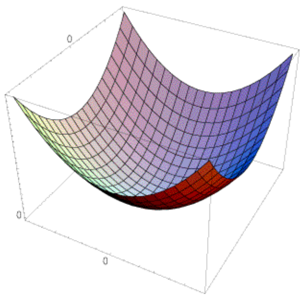


1. **Find a set of distinctive key-points**
2. **Define a region around each keypoint**
3. **Extract and normalize the region content**
4. **Compute a local descriptor from the normalized region**
5. **Match local descriptors**

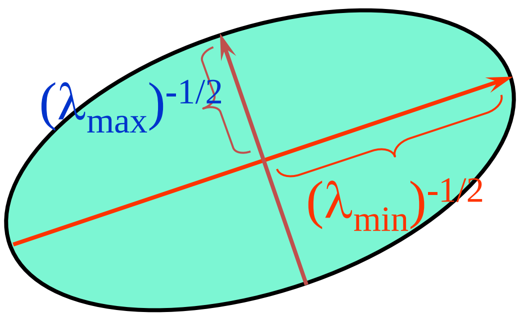
Review: Harris corner detector

$E(u, v)$



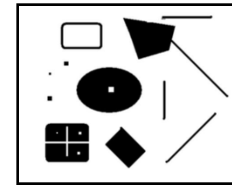


- Approximate distinctiveness by local auto-correlation.
- Approximate local auto-correlation by second moment matrix
- Quantify distinctiveness (or cornerness) as function of the eigenvalues of the second moment matrix.
- But we don't actually need to compute the eigenvalues by using the determinant and trace of the second moment matrix.



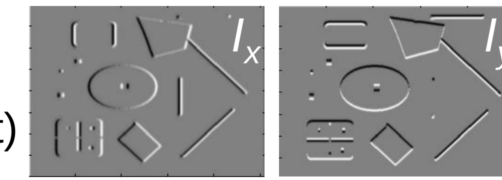
Harris Detector [Harris88]

- Second moment matrix



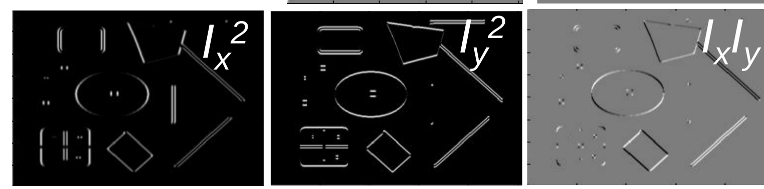
$$\mu(\sigma_I,\sigma_D)=g(\sigma_I)*\begin{bmatrix} I_x^2(\sigma_D) & I_xI_y(\sigma_D) \\ I_xI_y(\sigma_D) & I_y^2(\sigma_D)\end{bmatrix}$$

1. Image derivatives (optionally, blur first)



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

2. Square of derivatives



3. Gaussian filter $g(\sigma_I)$

4. Cornerness function – both eigenvalues are strong

$$har=\det[\mu(\sigma_I,\sigma_D)]-\alpha[\operatorname{trace}(\mu(\sigma_I,\sigma_D))^2]=$$
$$g(I_x^2)g(I_y^2)-[g(I_xI_y)]^2-\alpha[g(I_x^2)+g(I_y^2)]^2$$

5. Non-maxima suppression

So far: can localize in x-y, but not scale

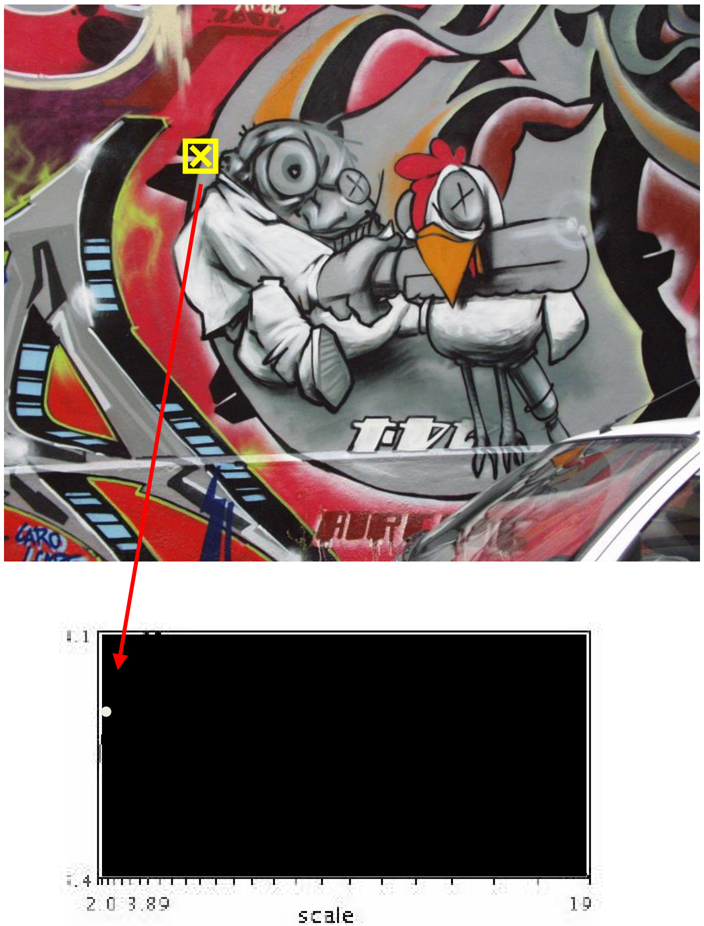
Automatic Scale Selection

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

How to find corresponding patch sizes?

Automatic Scale Selection

- Function responses for increasing scale (scale signature)

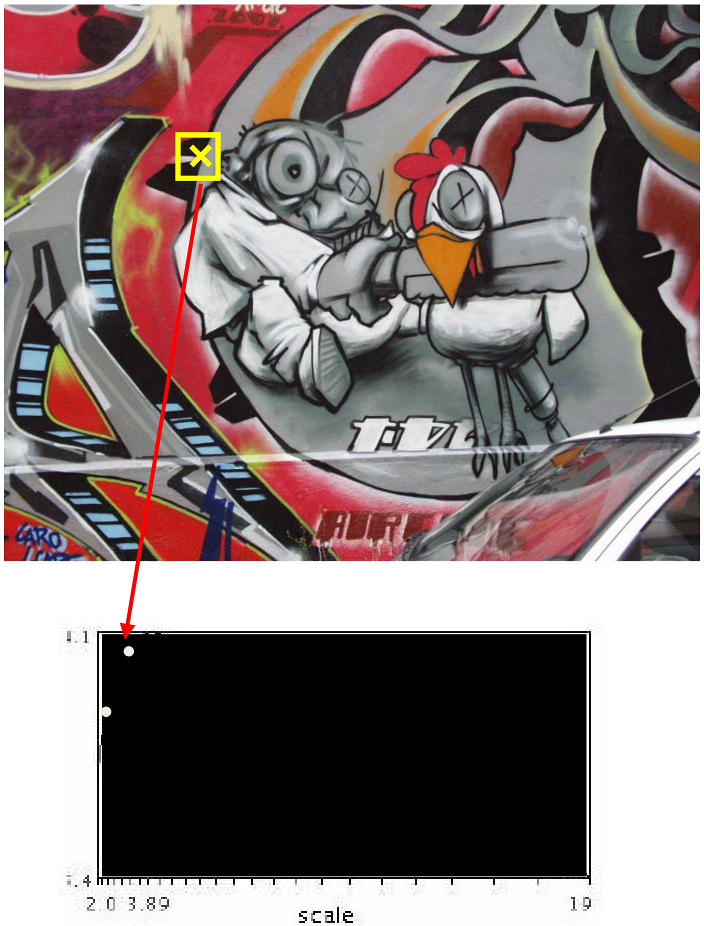


$f(I_{i_1 \dots i_m}(x, \sigma))$

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

Automatic Scale Selection

- Function responses for increasing scale (scale signature)



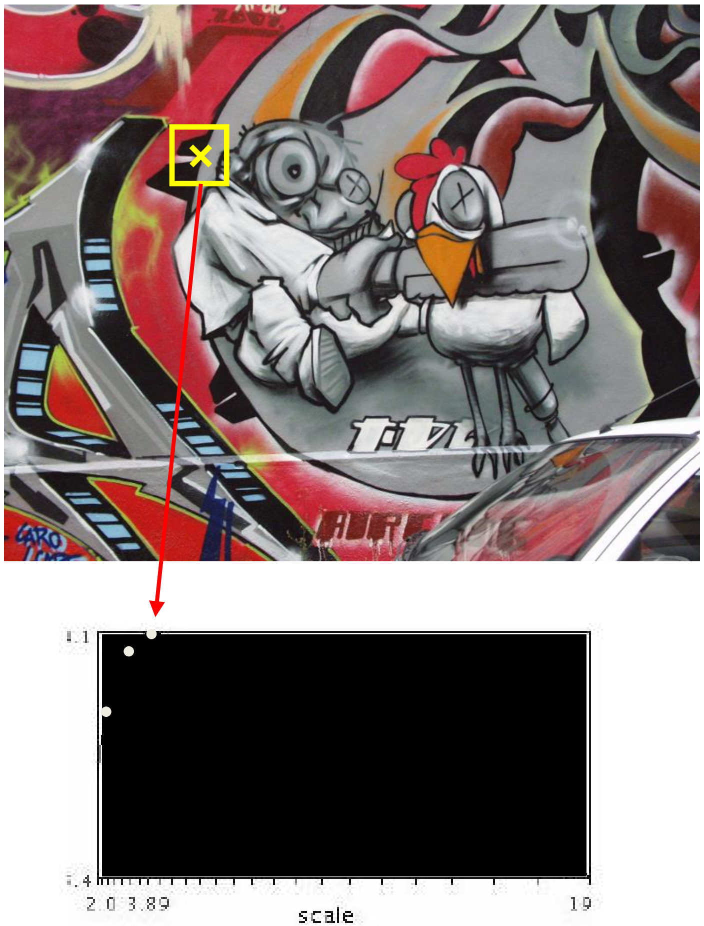
$f(I_{i_1 \ldots i_m}(x,\sigma))$

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

K. Grauman, B. Leibe

Automatic Scale Selection

- Function responses for increasing scale (scale signature)

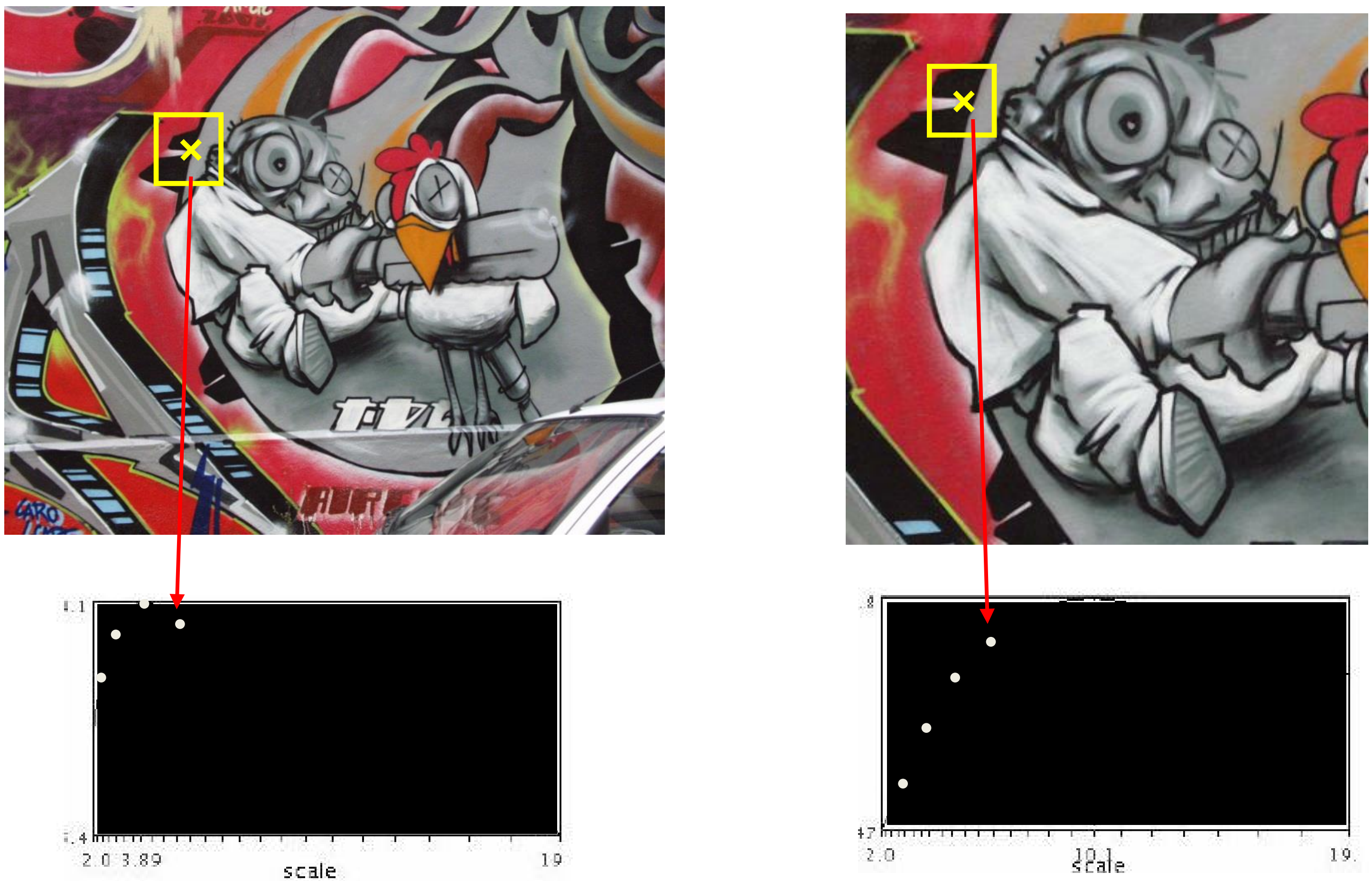


$f(I_{i_1 \ldots i_m}(x,\sigma))$

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

Automatic Scale Selection

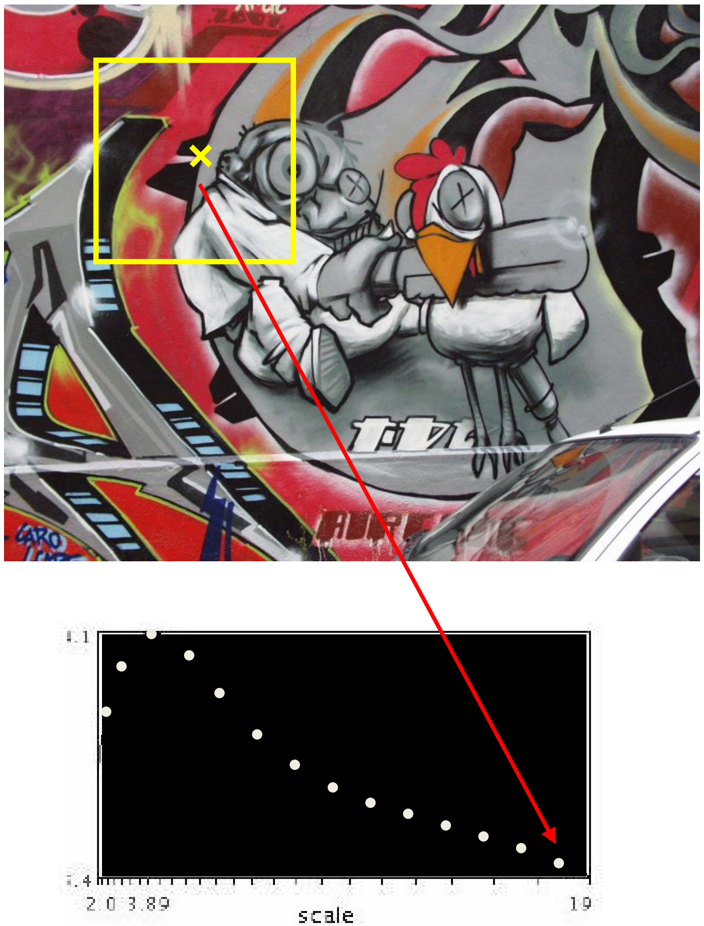
- Function responses for increasing scale (scale signature)



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

Automatic Scale Selection

- Function responses for increasing scale (scale signature)



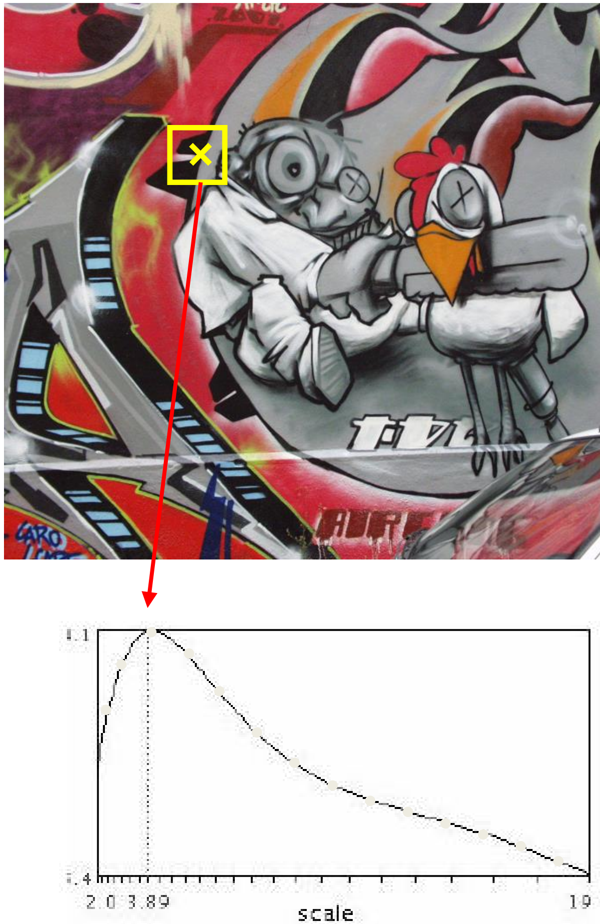
$$f(I_{i_1 \ldots i_m}(x,\sigma))$$

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

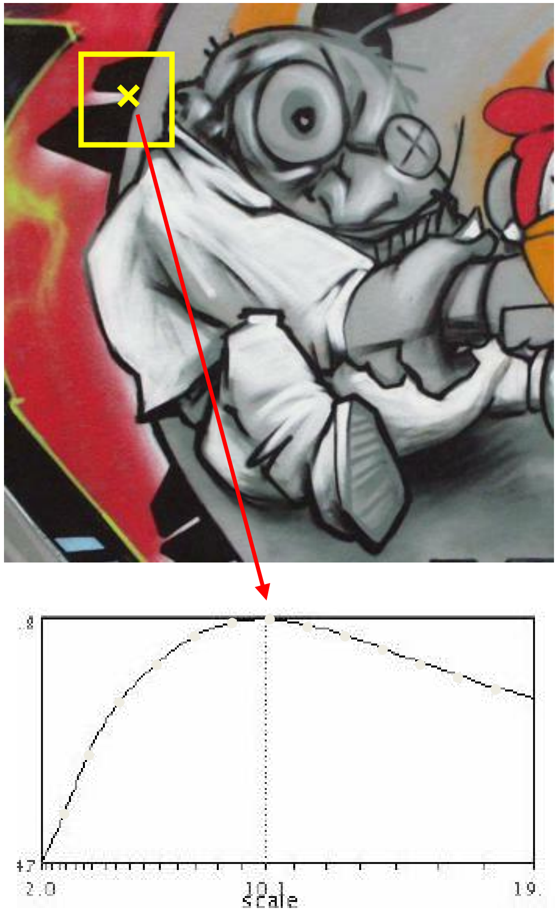
K. Grauman, B. Leibe

Automatic Scale Selection

- Function responses for increasing scale (scale signature)



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

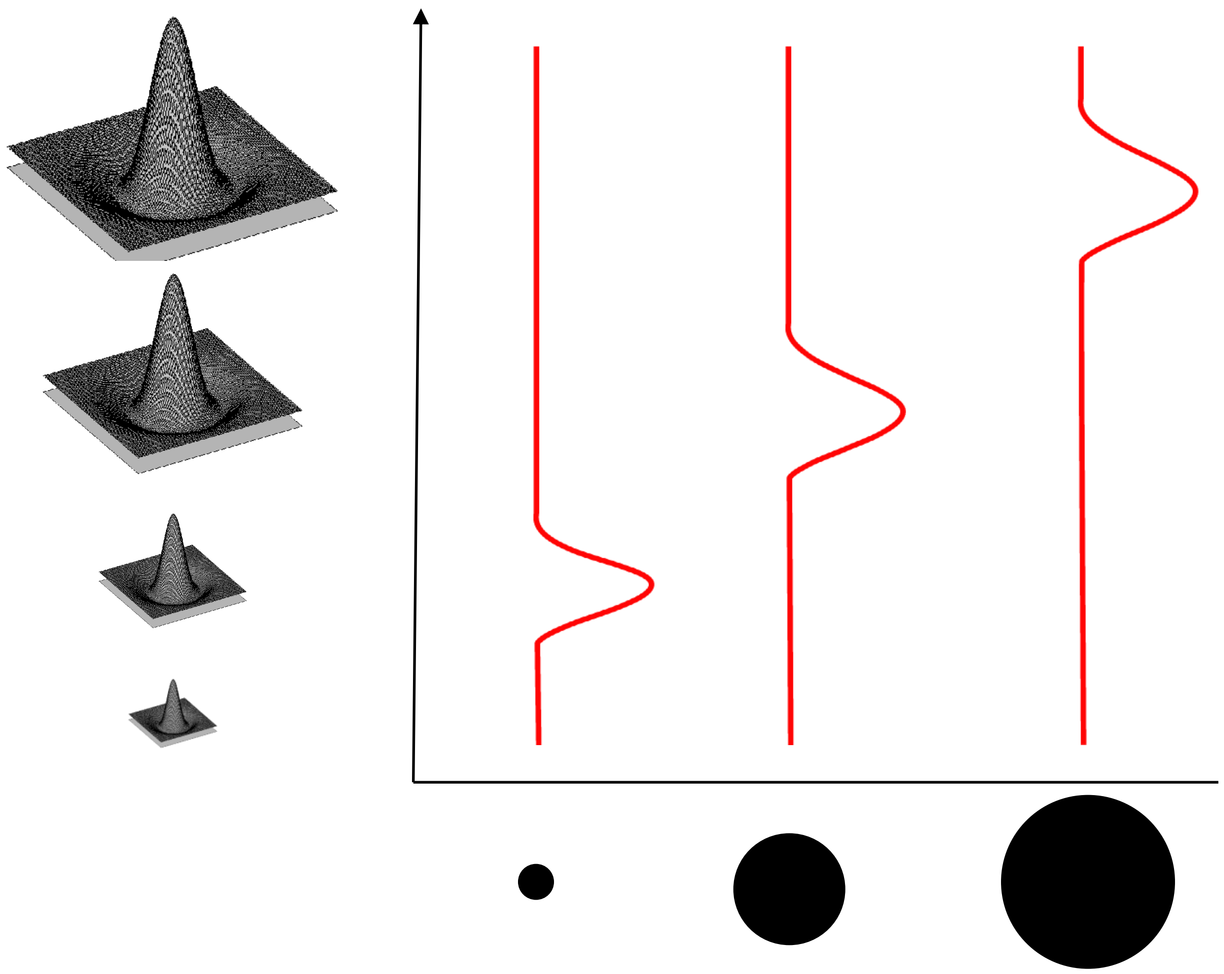


$f(I_{i_1 \ldots i_m}(x', \sigma'))$

K. Grauman, B. Leibe

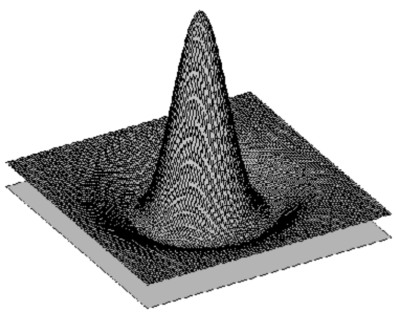
What Is A Useful Signature Function?

- Difference-of-Gaussian = “blob” detector



K. Grauman, B. Leibe

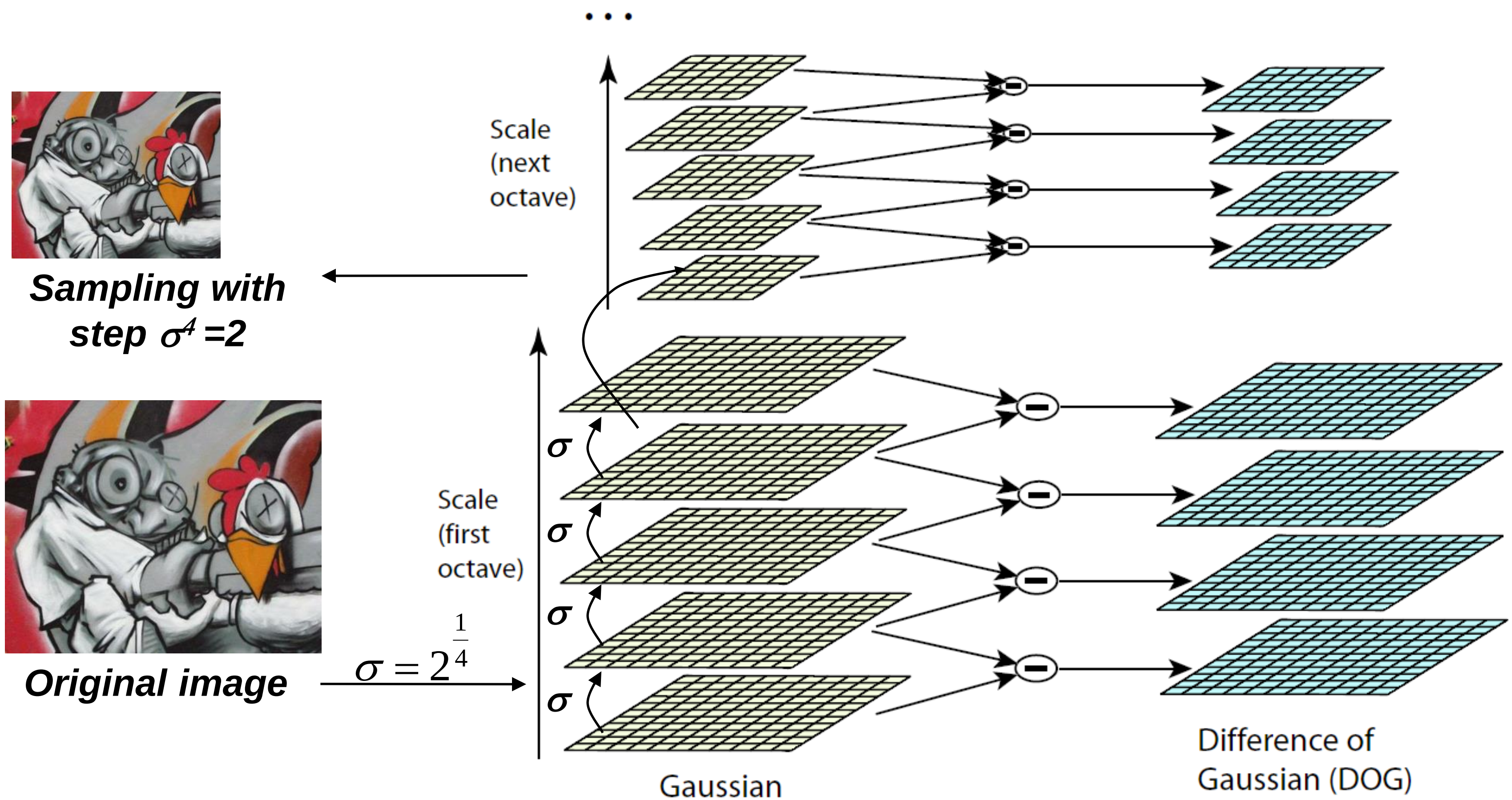
Difference-of-Gaussian (DoG)



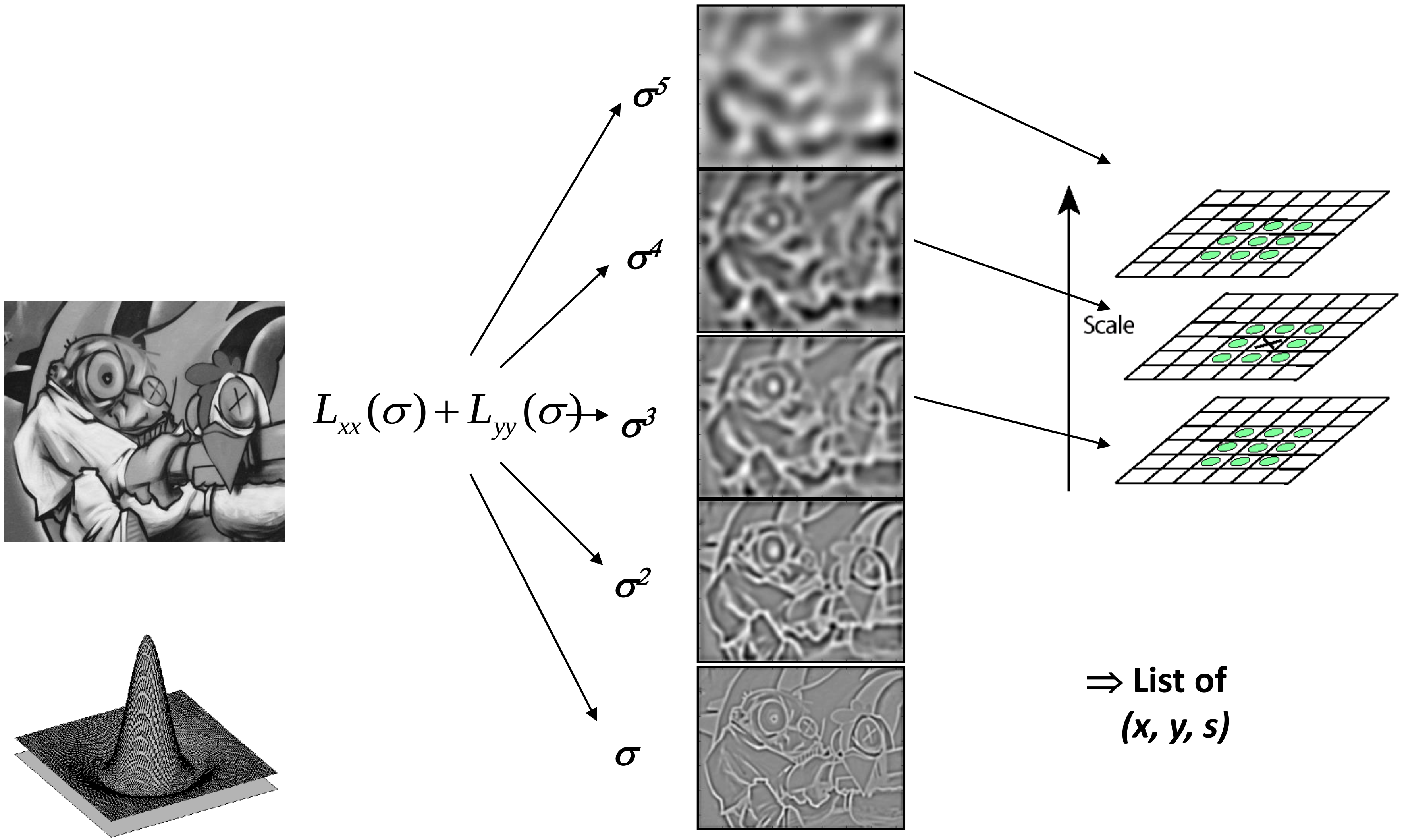
K. Grauman, B. Leibe

DoG – Efficient Computation

- Computation in Gaussian scale pyramid



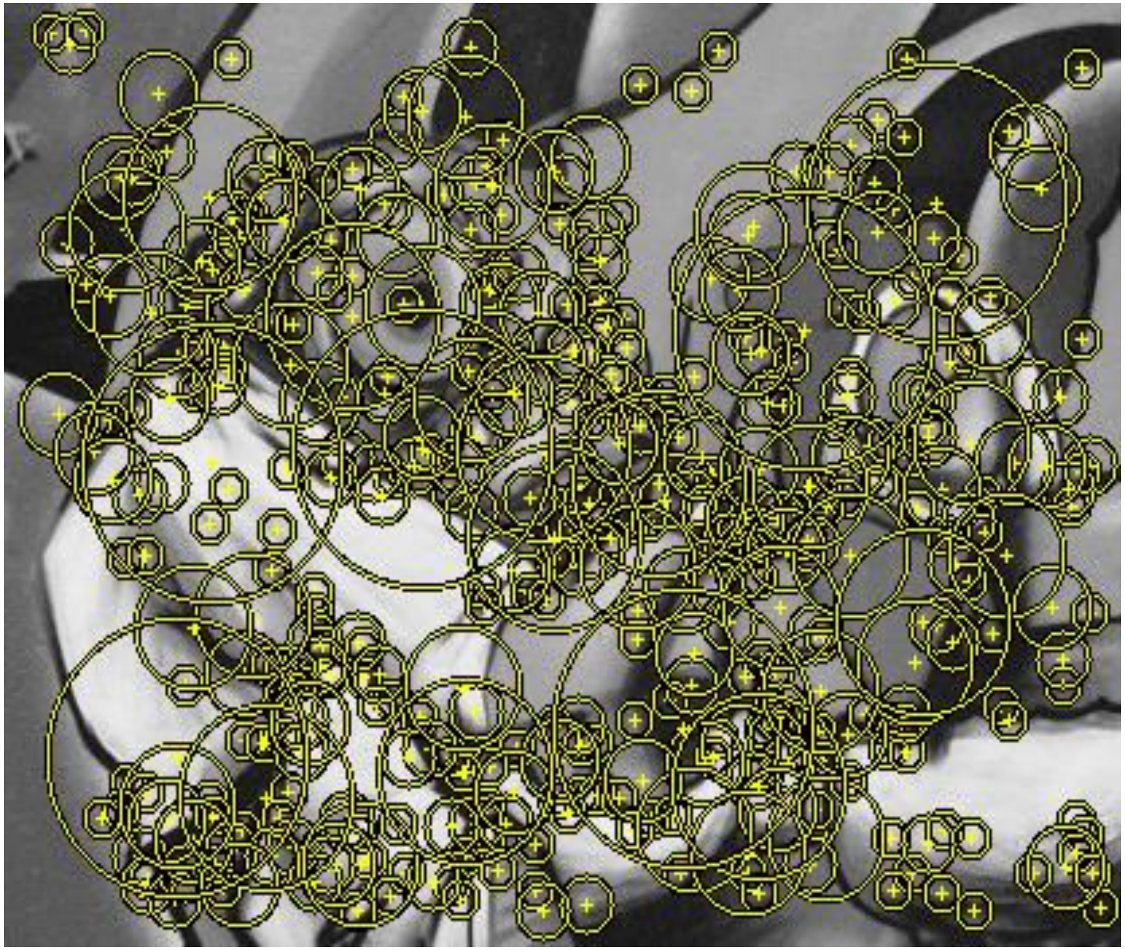
Find local maxima in position-scale space of Difference-of-Gaussian



⇒ **List of *(x, y, s)***

K. Grauman, B. Leibe

Results: Difference-of-Gaussian

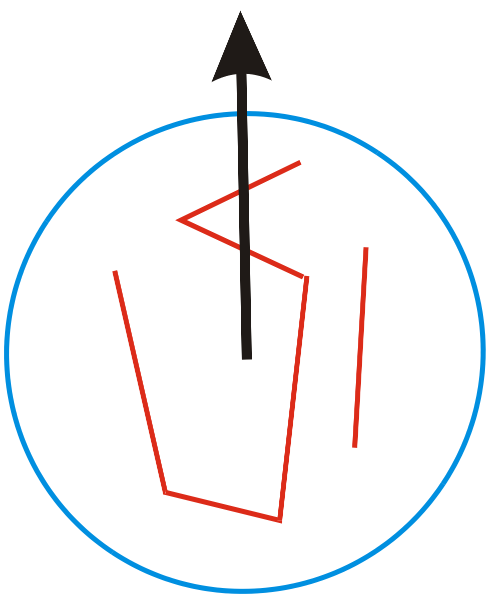


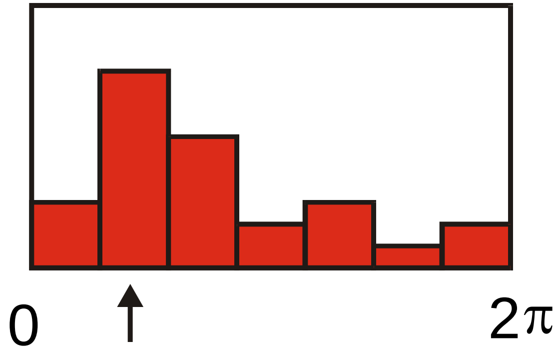
K. Grauman, B. Leibe

Orientation Normalization

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

[Lowe, SIFT, 1999]





Maximally Stable Extremal Regions [Matas '02]

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

K. Grauman, B. Leibe

Example Results: MSER

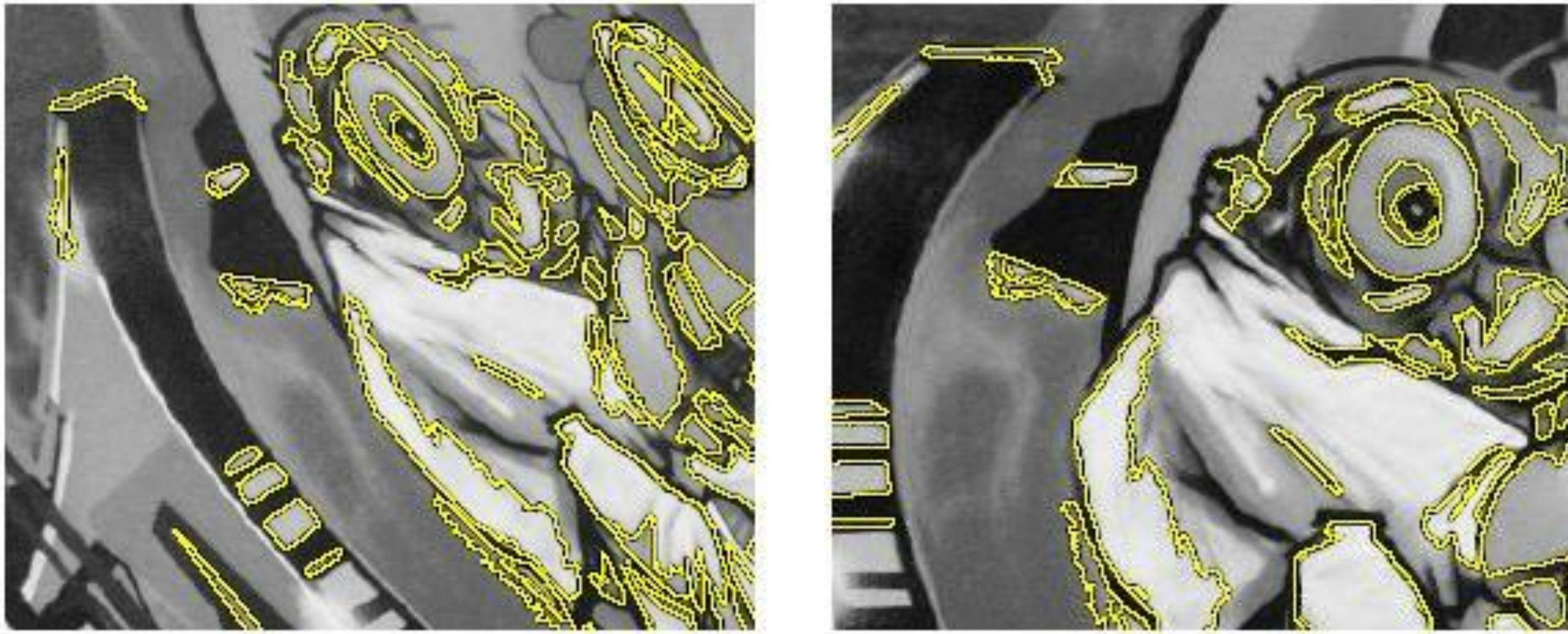
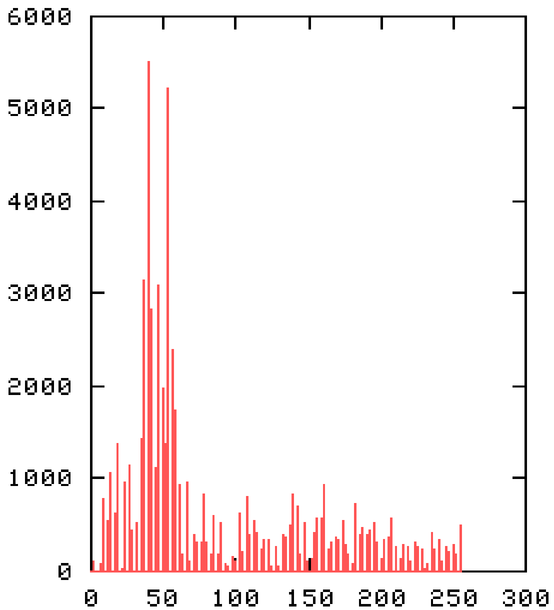


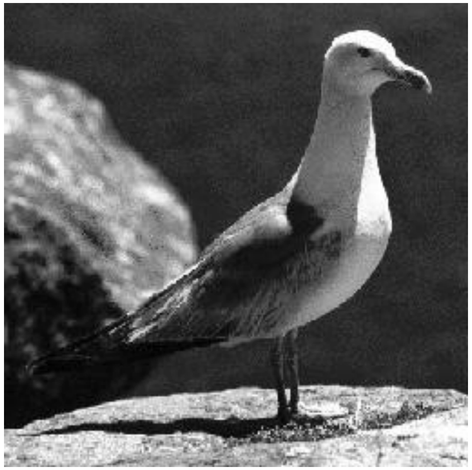
Image representations

- Templates
  - Intensity, gradients, etc.

- Histograms
  - Color, texture, SIFT descriptors, etc.

Image Representations: Histograms





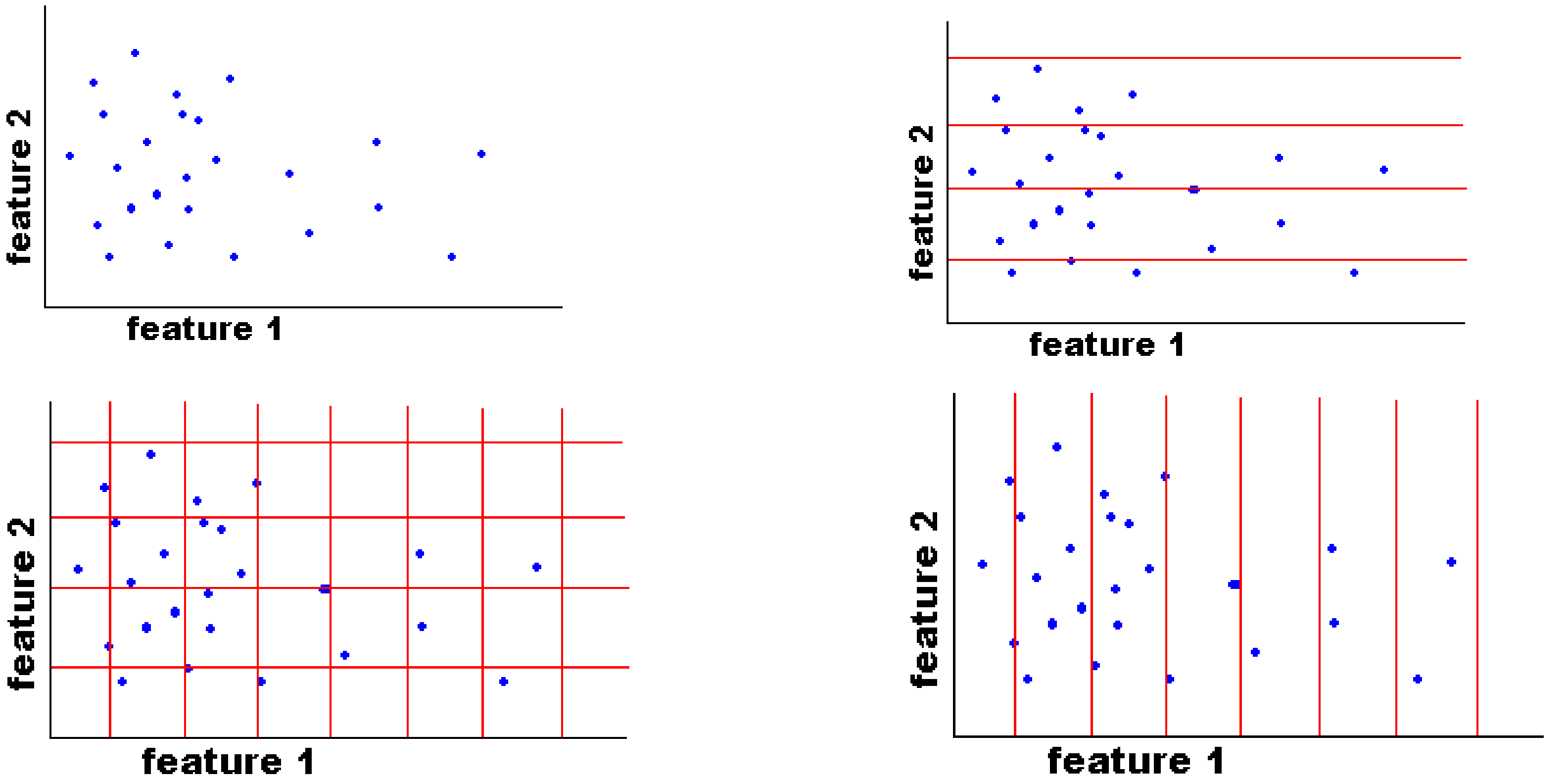
Global histogram

- 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

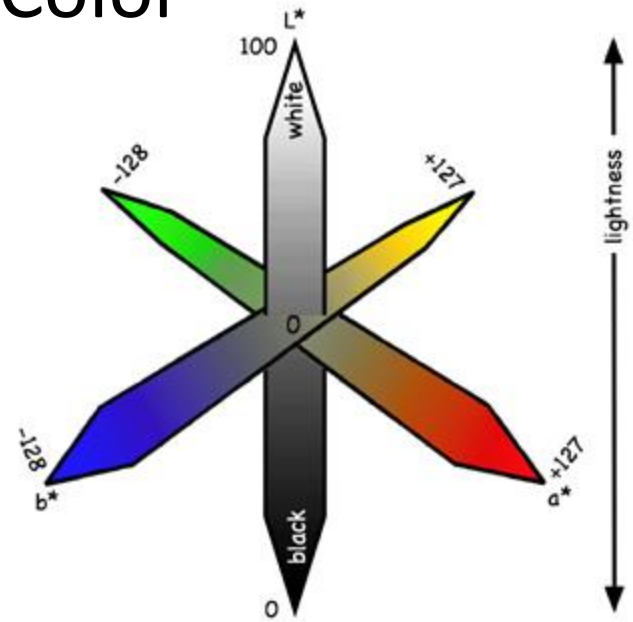
Marginal histogram

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

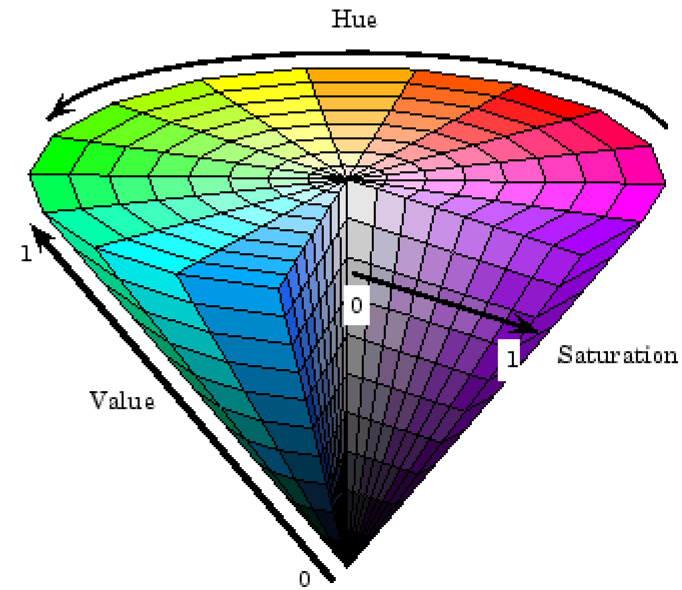
Images from Dave Kauchak

What kind of things do we compute histograms of?

- Color



L*a*b* color space

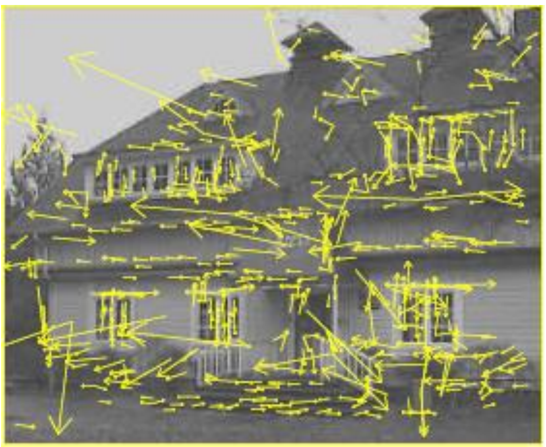


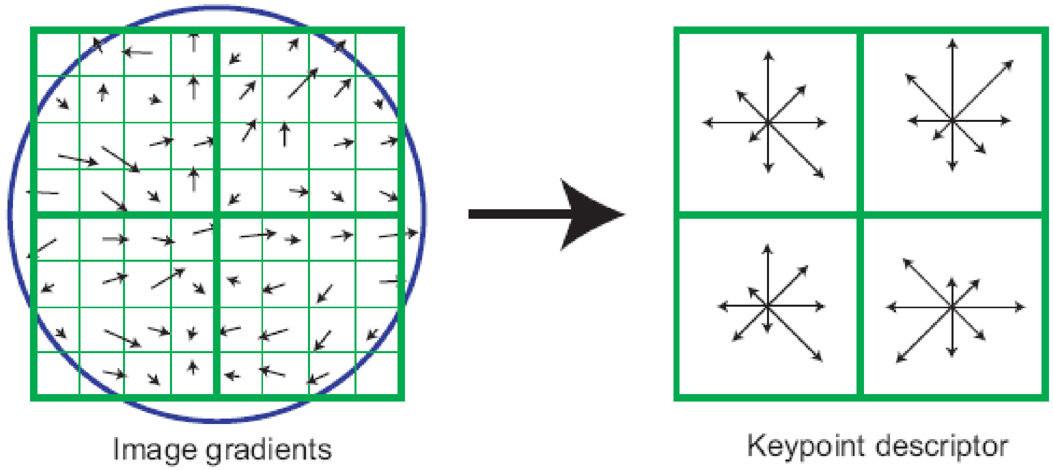
HSV color space

- Texture (filter banks or HOG over regions)

What kind of things do we compute histograms of?

- Histograms of oriented gradients





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)

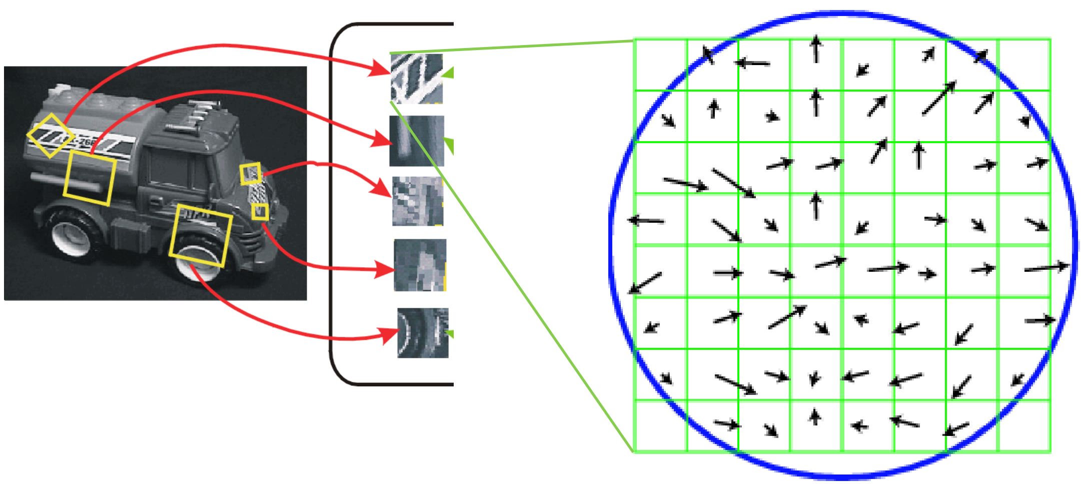
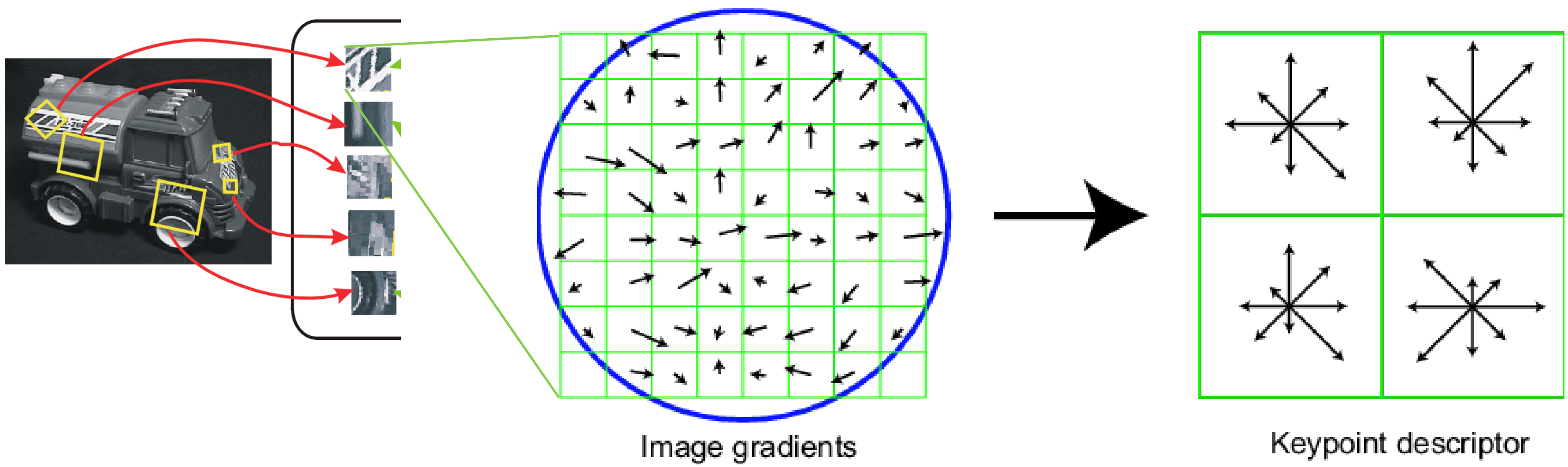


Image gradients

SIFT vector formation

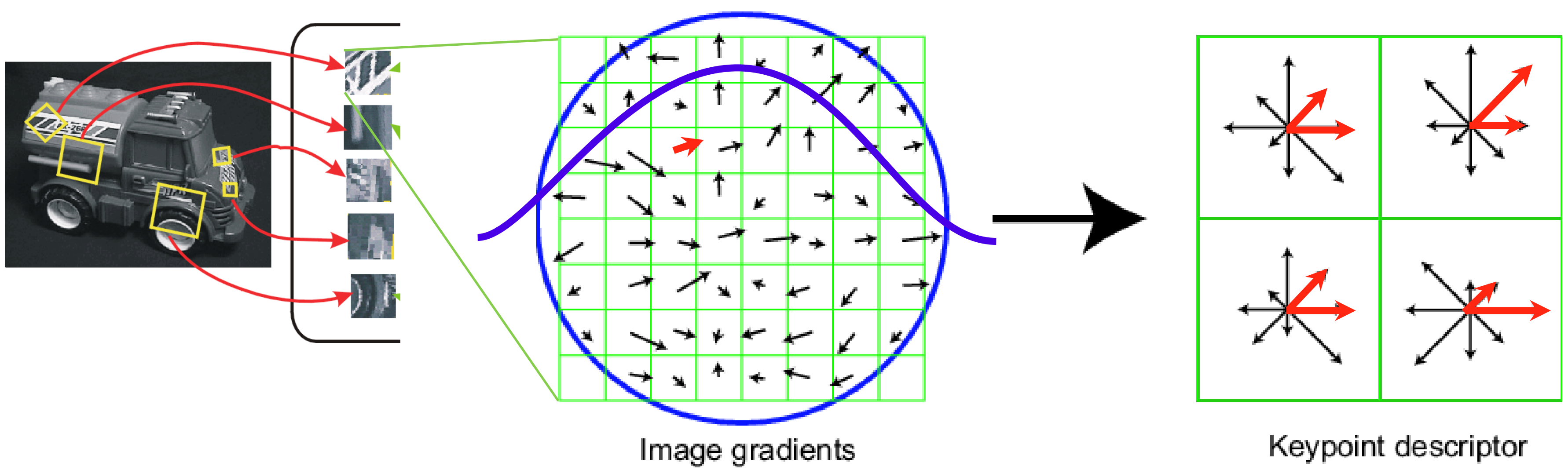
- 4x4 array of gradient orientation histogram weighted by magnitude
- 8 orientations x 4x4 array = 128 dimensions
- Motivation: some sensitivity to spatial layout, but not too much.



showing only 2x2 here but is 4x4

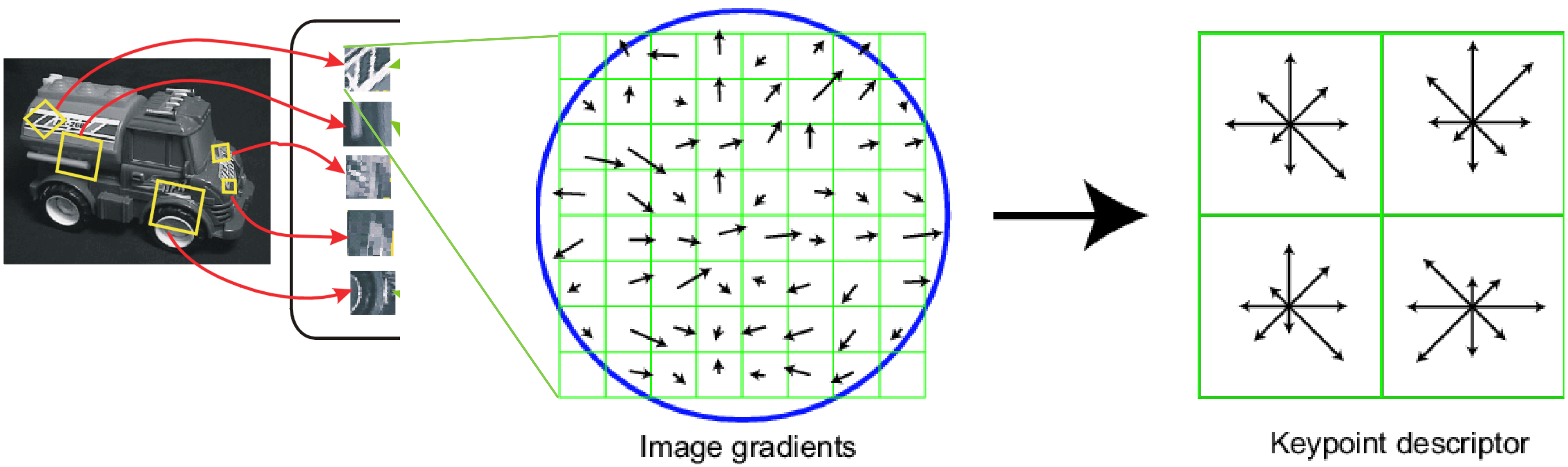
Ensure smoothness

- Gaussian weight
- Trilinear interpolation
  - a given gradient contributes to 8 bins: 4 in space times 2 in orientation

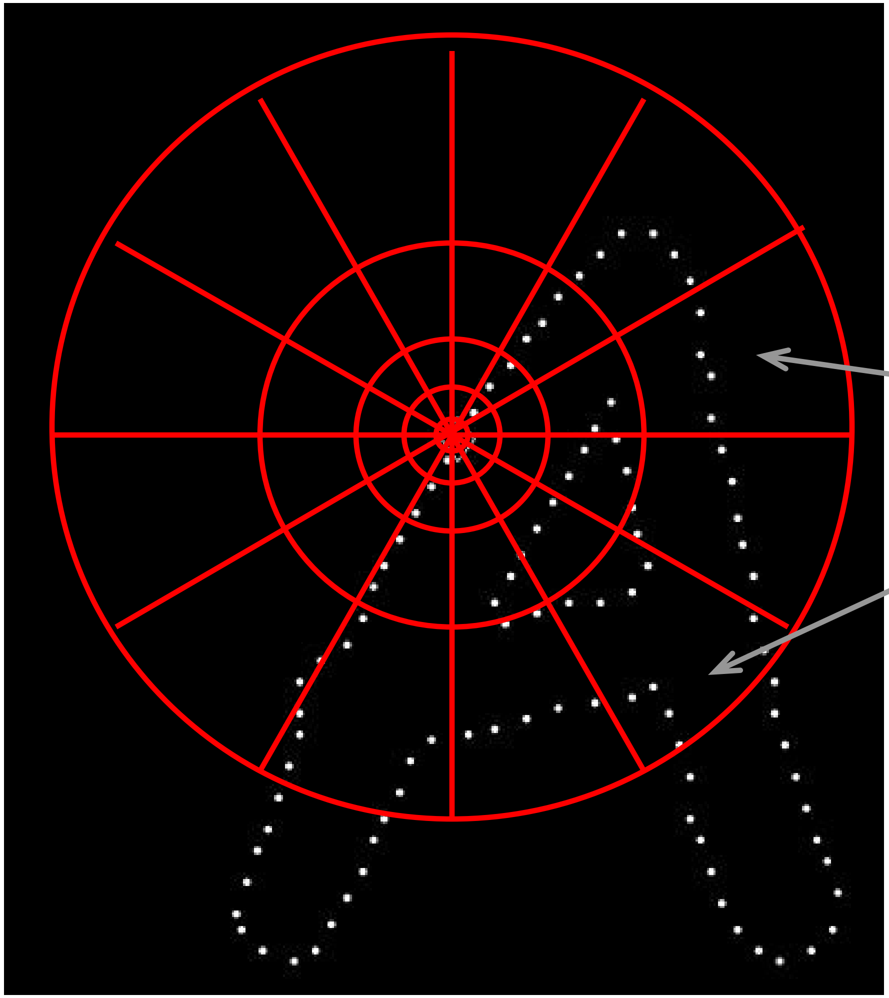


Reduce effect of illumination

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



Local Descriptors: Shape Context



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

Count = 4

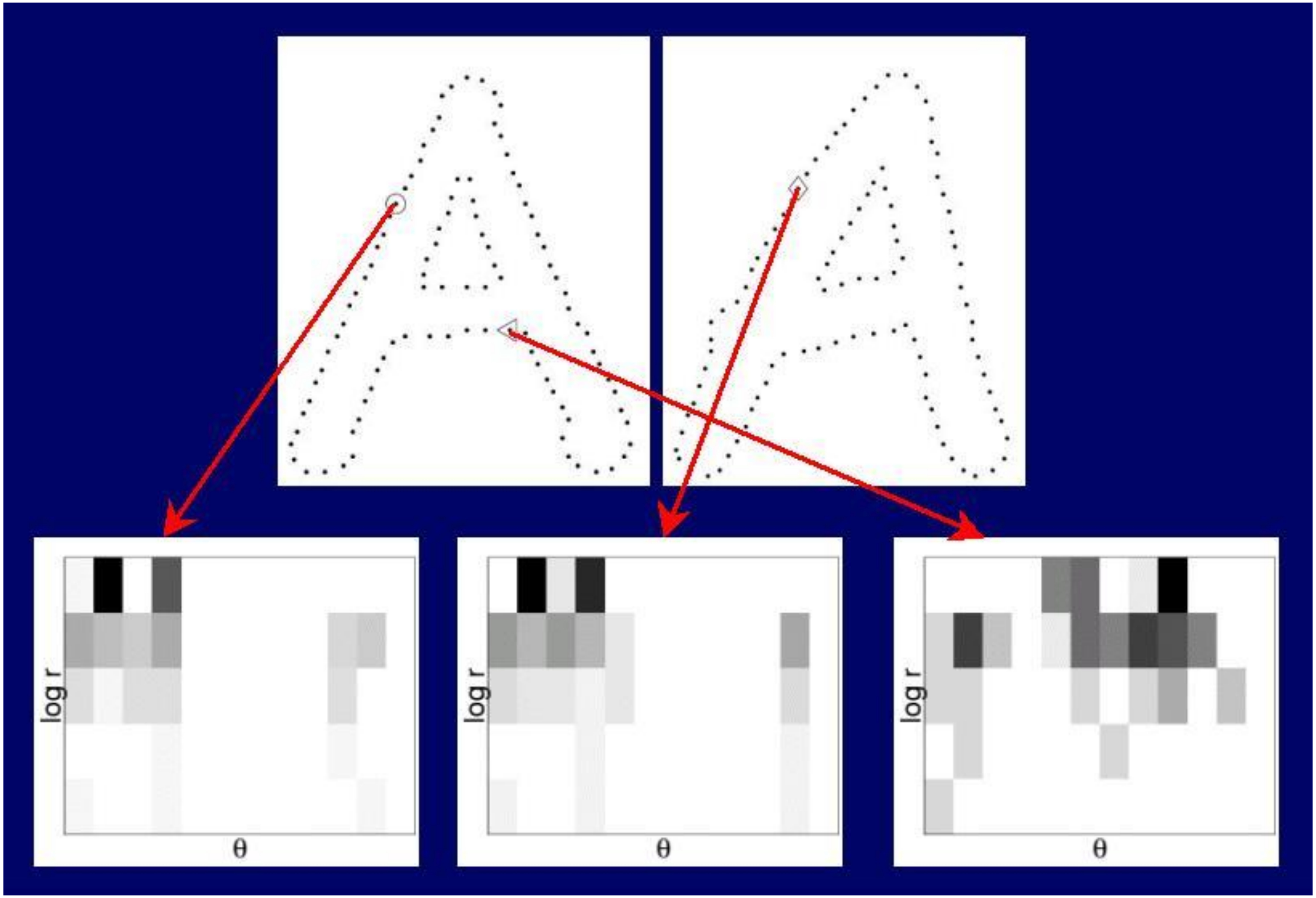
⋮

Count = 10

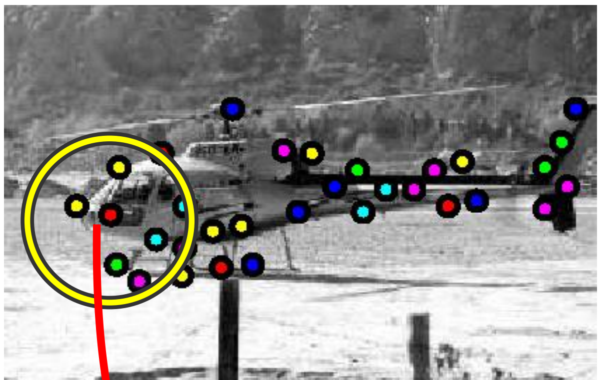
Log-polar binning: more precision for nearby points, more flexibility for farther points.

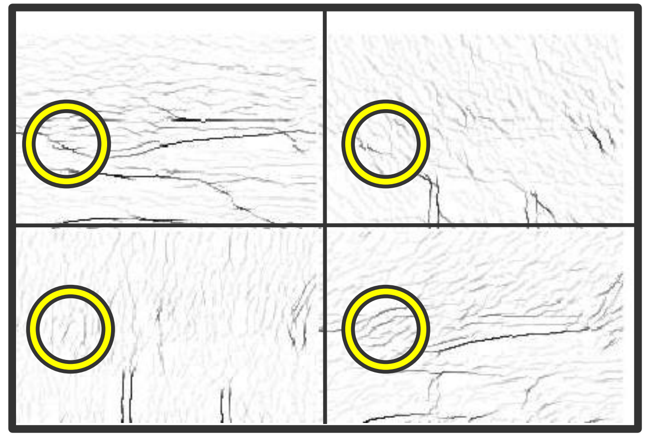
Belongie & Malik, ICCV 2001

Shape Context Descriptor



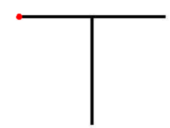
Local Descriptors: Geometric Blur





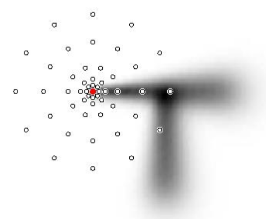
Compute edges at four orientations

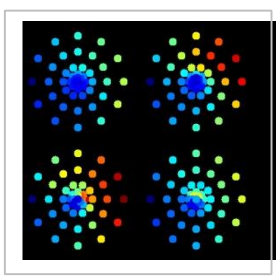
Extract a patch in each channel



Apply spatially varying blur and sub-sample

(Idealized signal)





Example descriptor

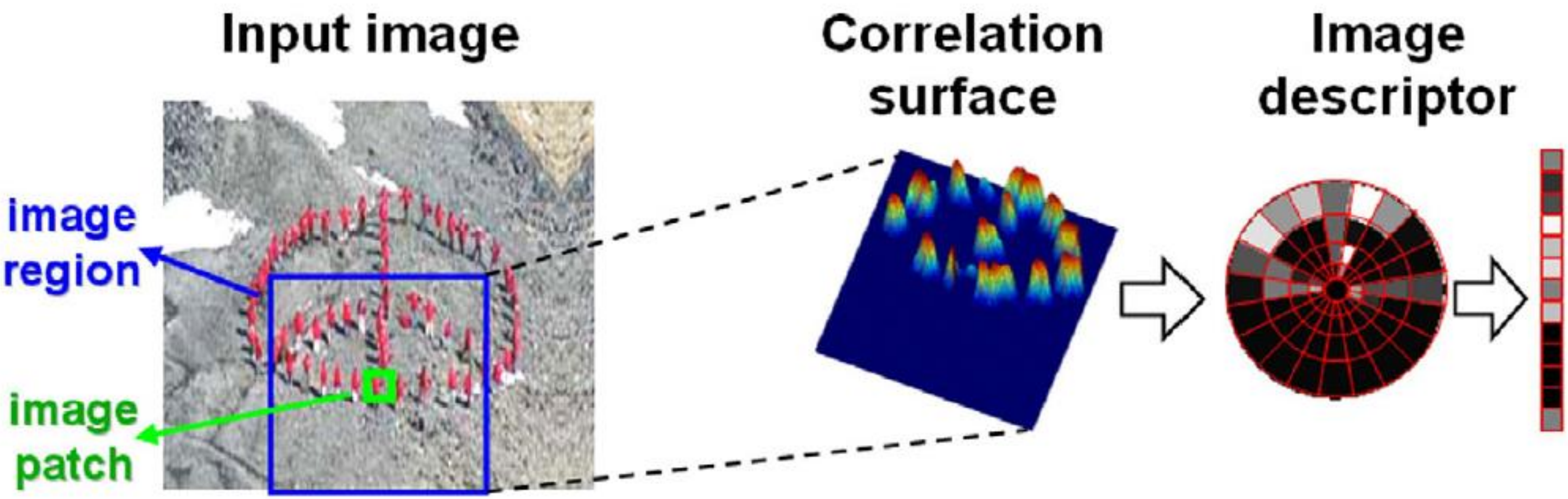
Berg & Malik, CVPR 2001

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.*

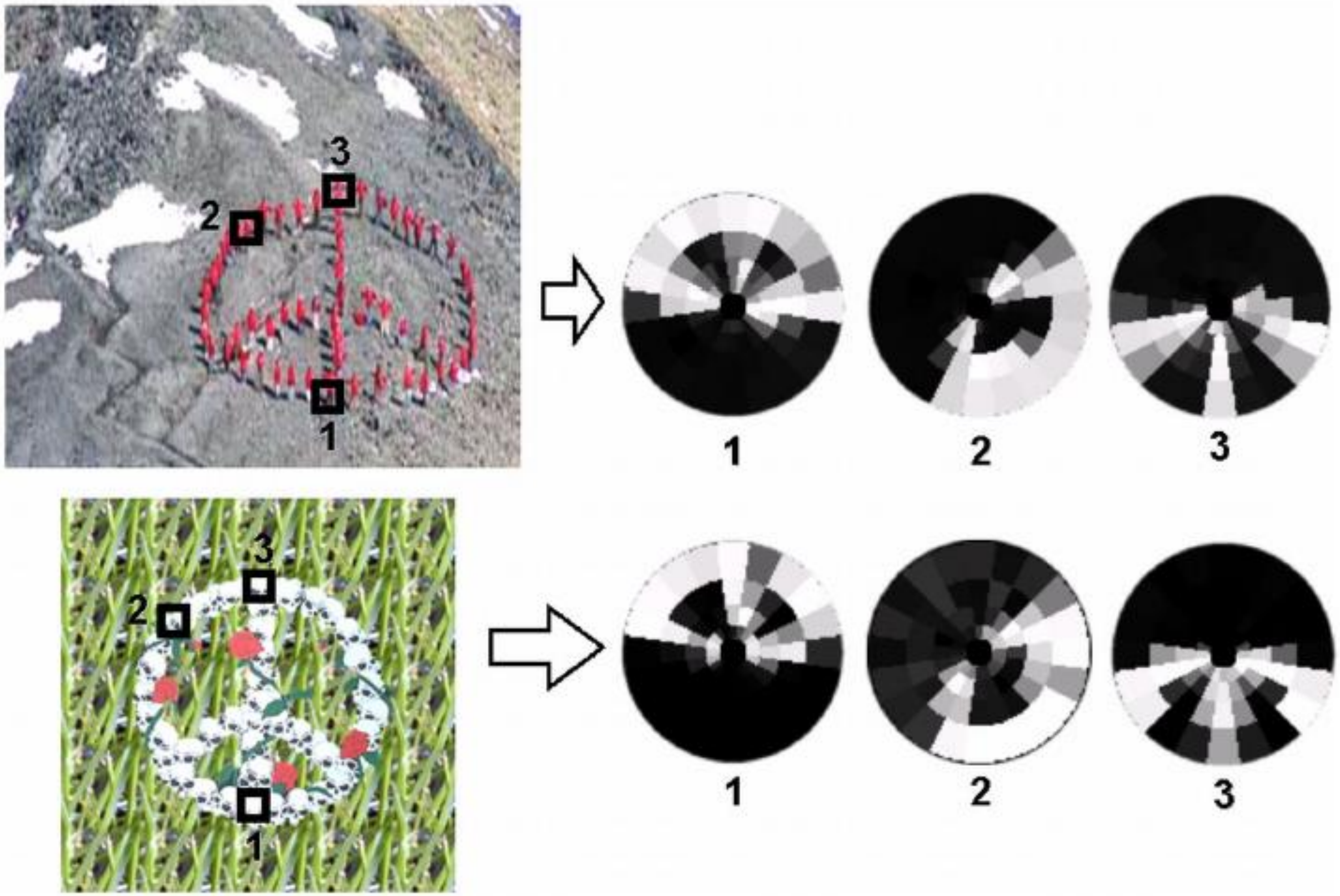
Matching Local Self-Similarities across Images and Videos, Shechtman and Irani, 2007

Self-similarity Descriptor



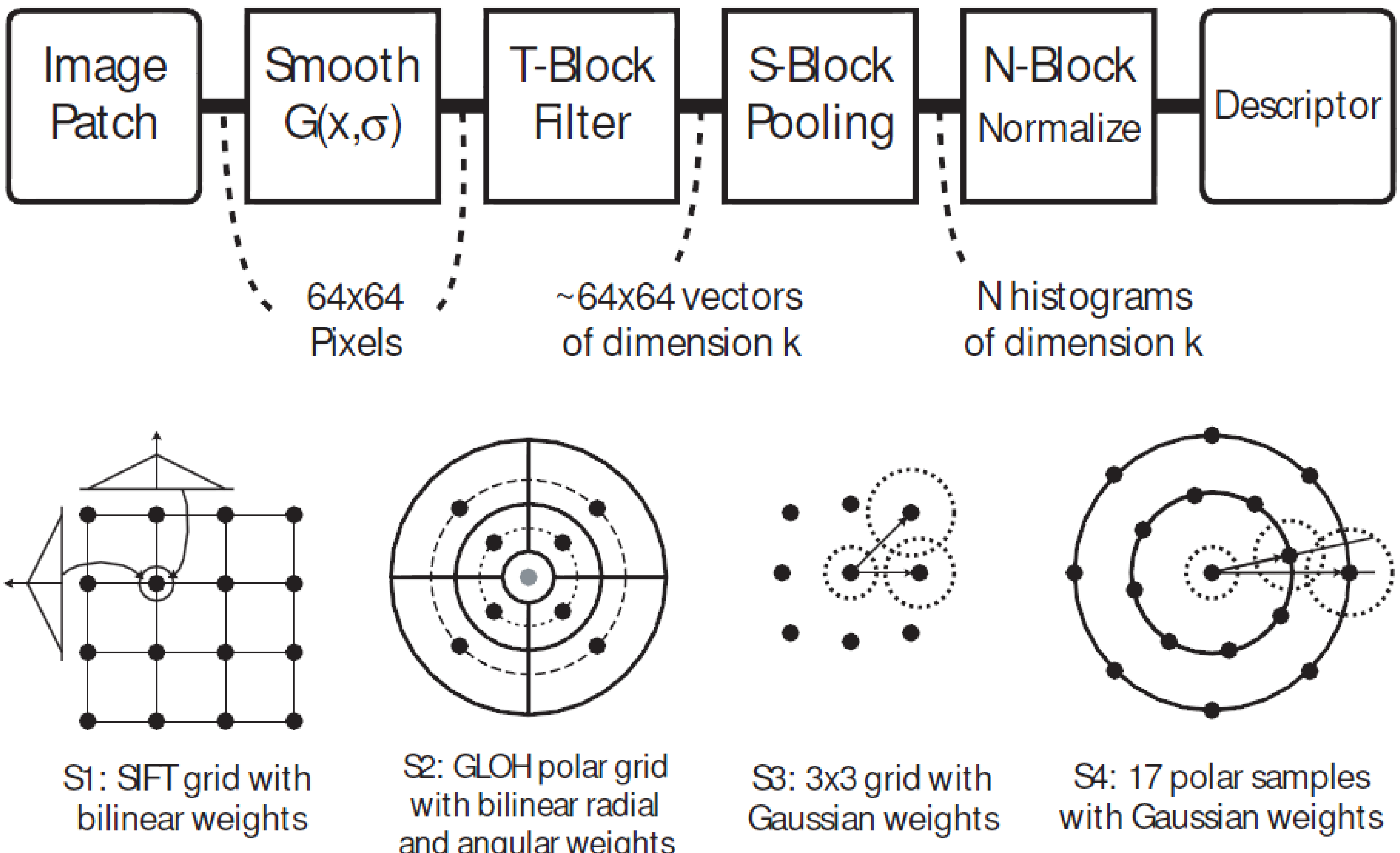
Matching Local Self-Similarities across Images and Videos, Shechtman and Irani, 2007

Self-similarity Descriptor



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

Learning Local Image Descriptors, Winder and Brown, 2007

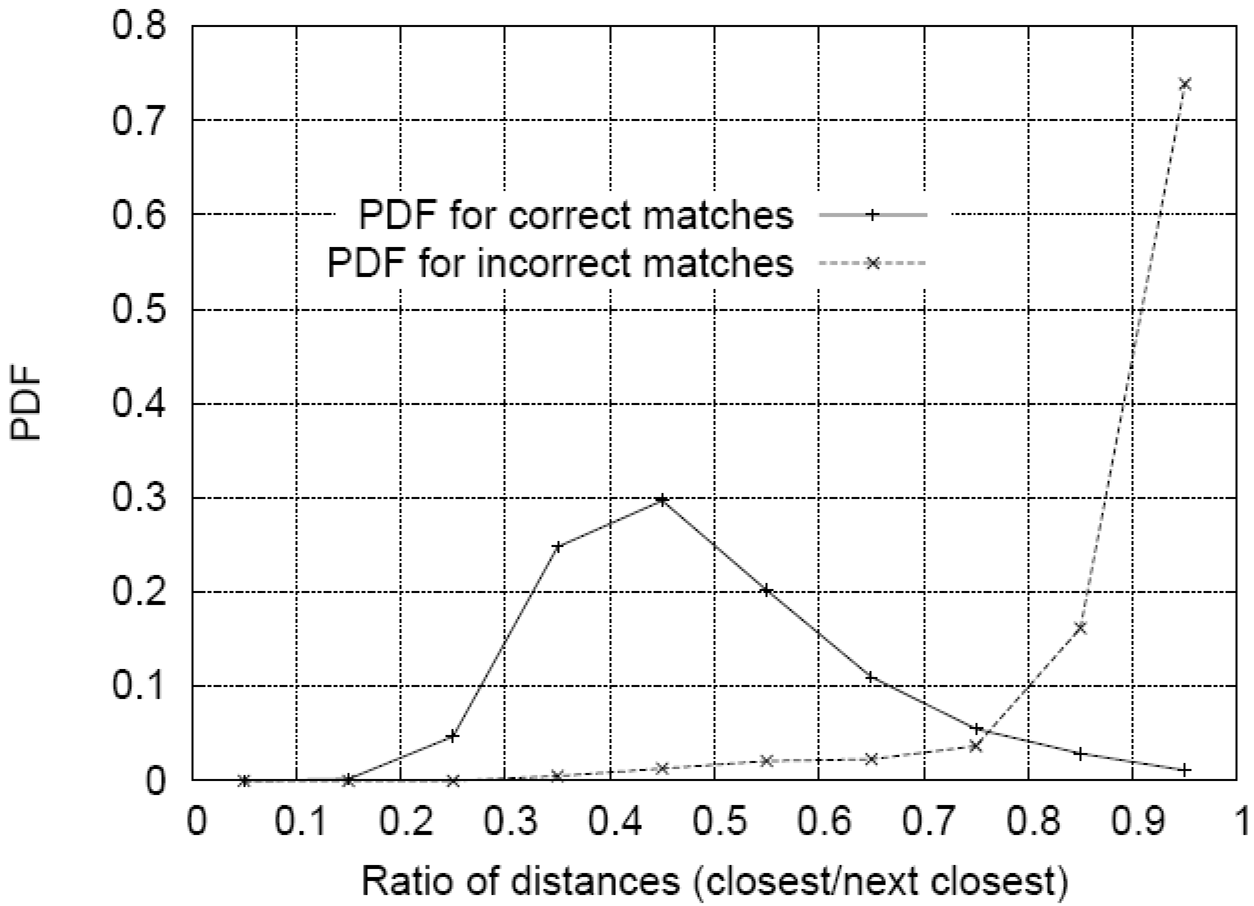


Local Descriptors

- 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

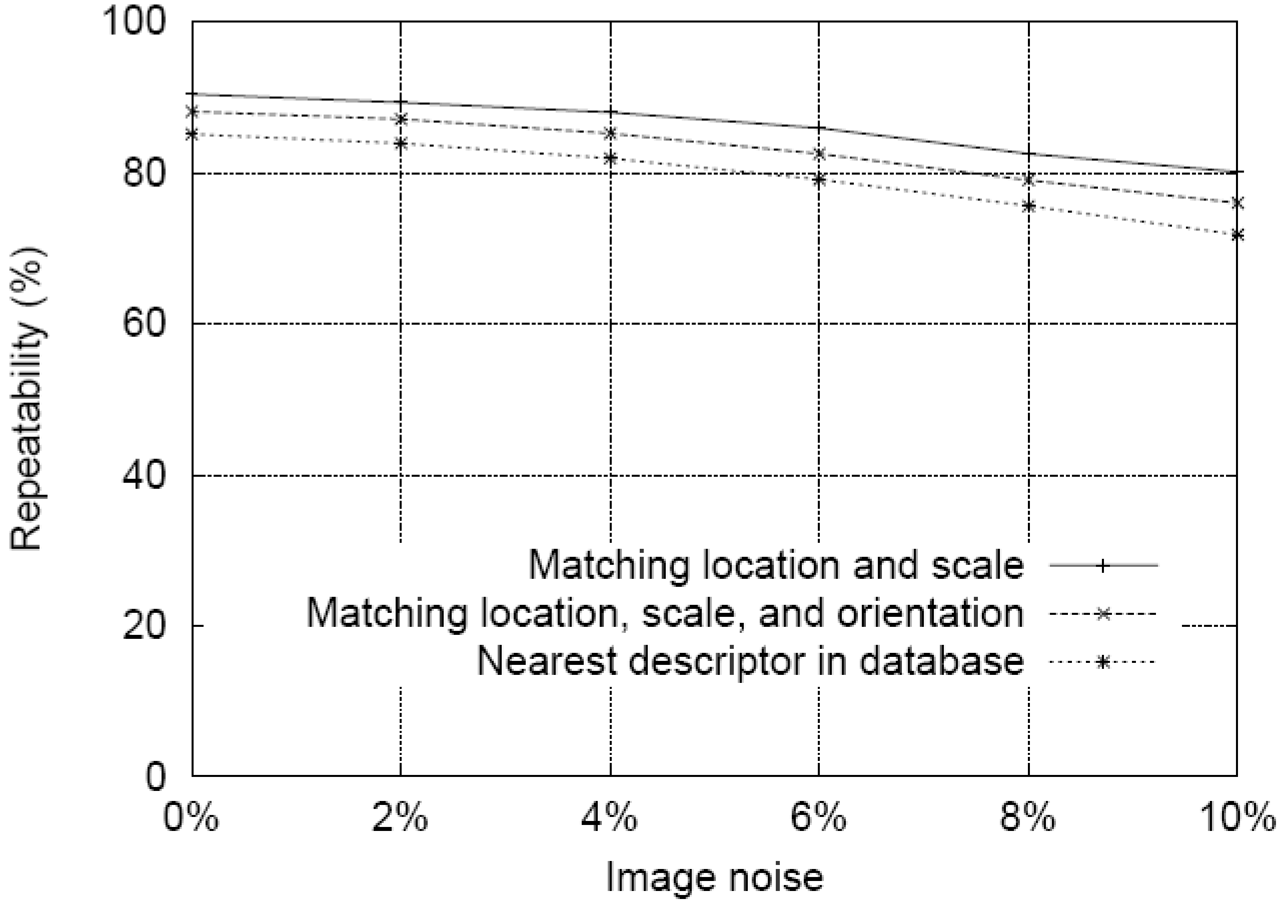
Matching Local Features

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



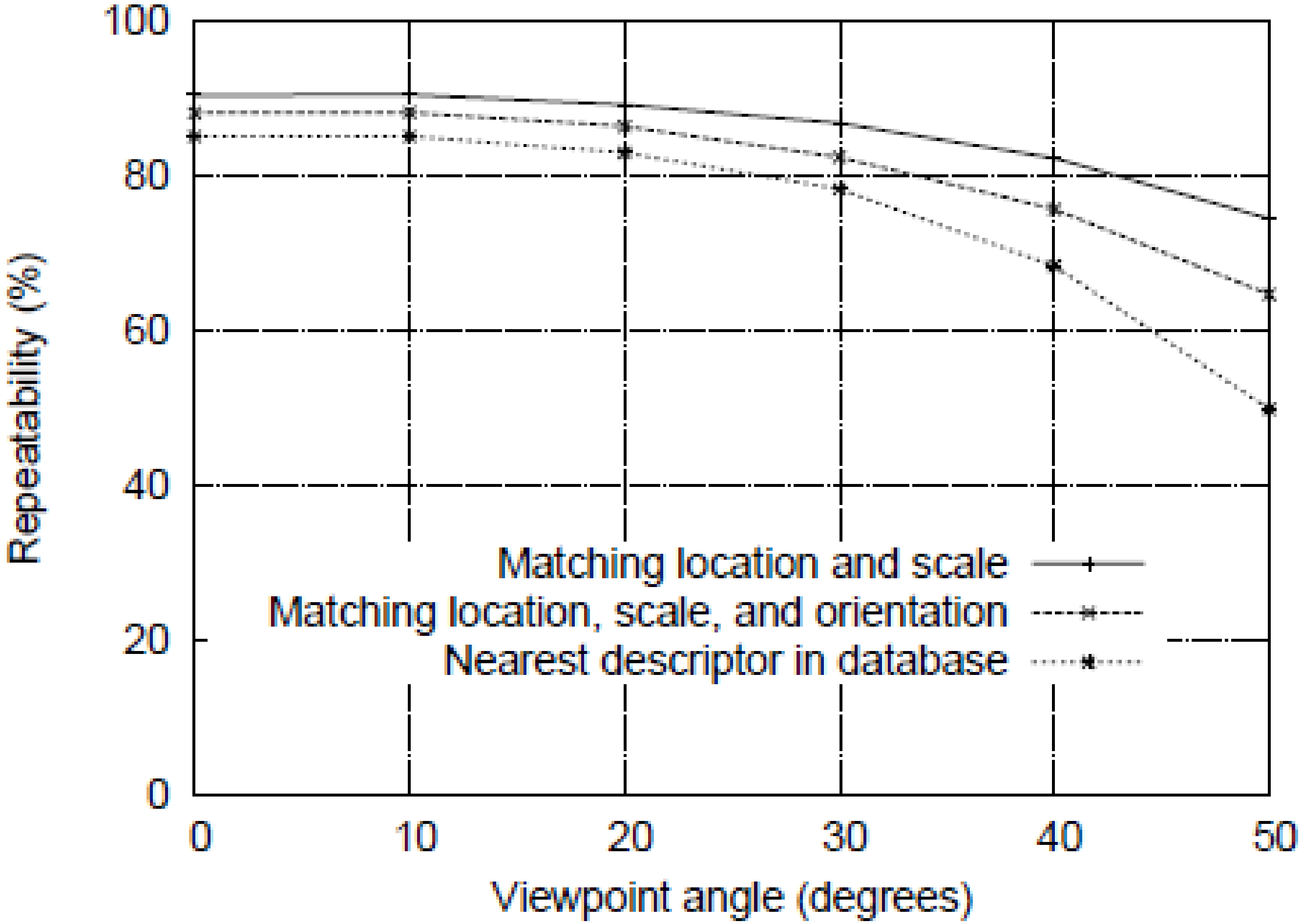
Lowe IJCV 2004

SIFT Repeatability

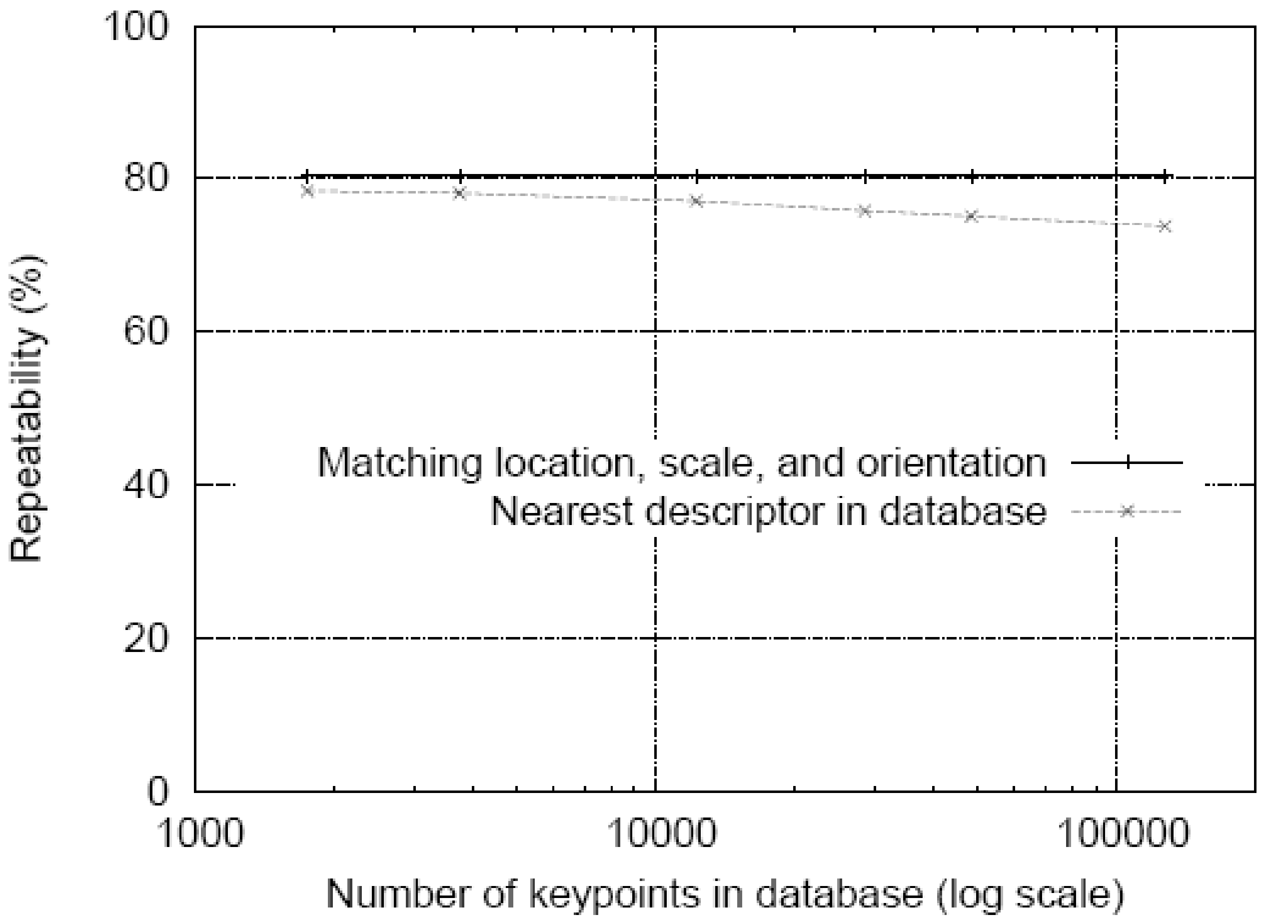


Lowe IJCV 2004

SIFT Repeatability

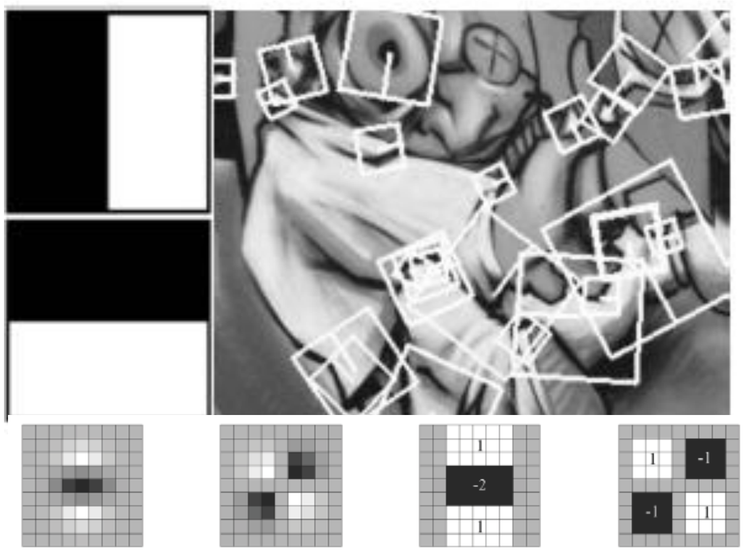


SIFT Repeatability



Lowe IJCV 2004

Local Descriptors: SURF



Fast approximation of SIFT idea

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

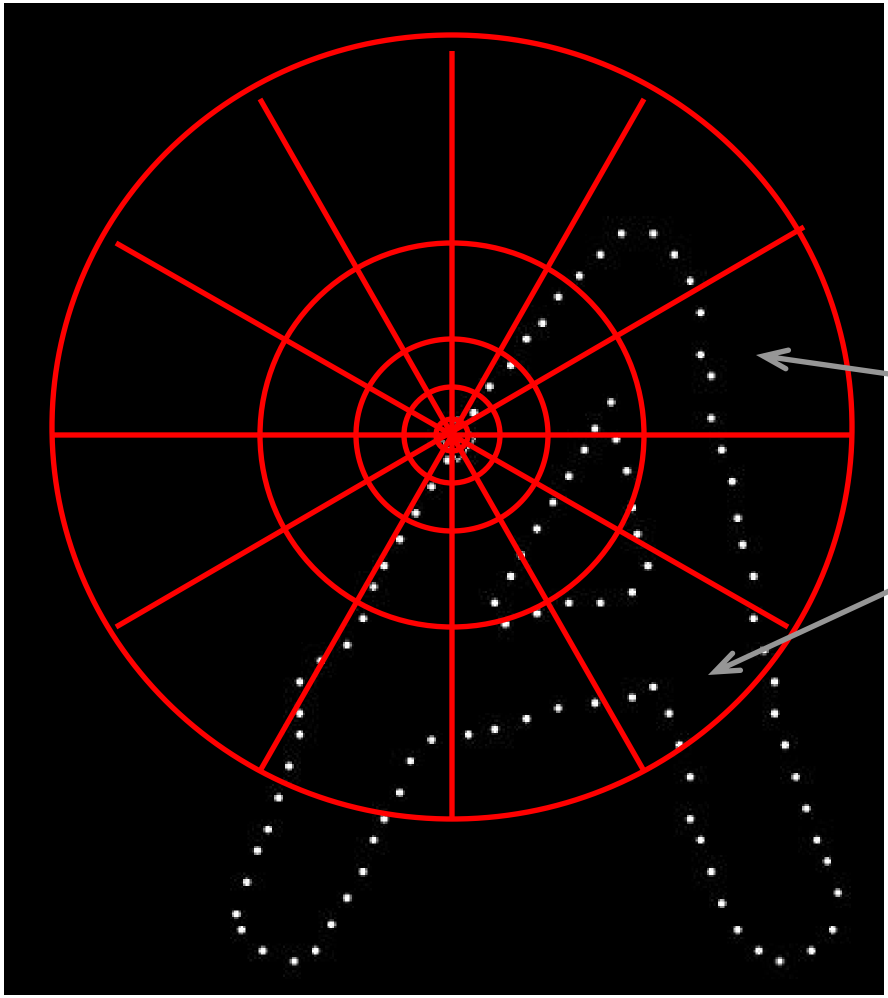
GPU implementation available

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

[Bay, ECCV’06], [Cornelis, CVGPU’08]

K. Grauman, B. Leibe

Local Descriptors: Shape Context



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

Count = 4

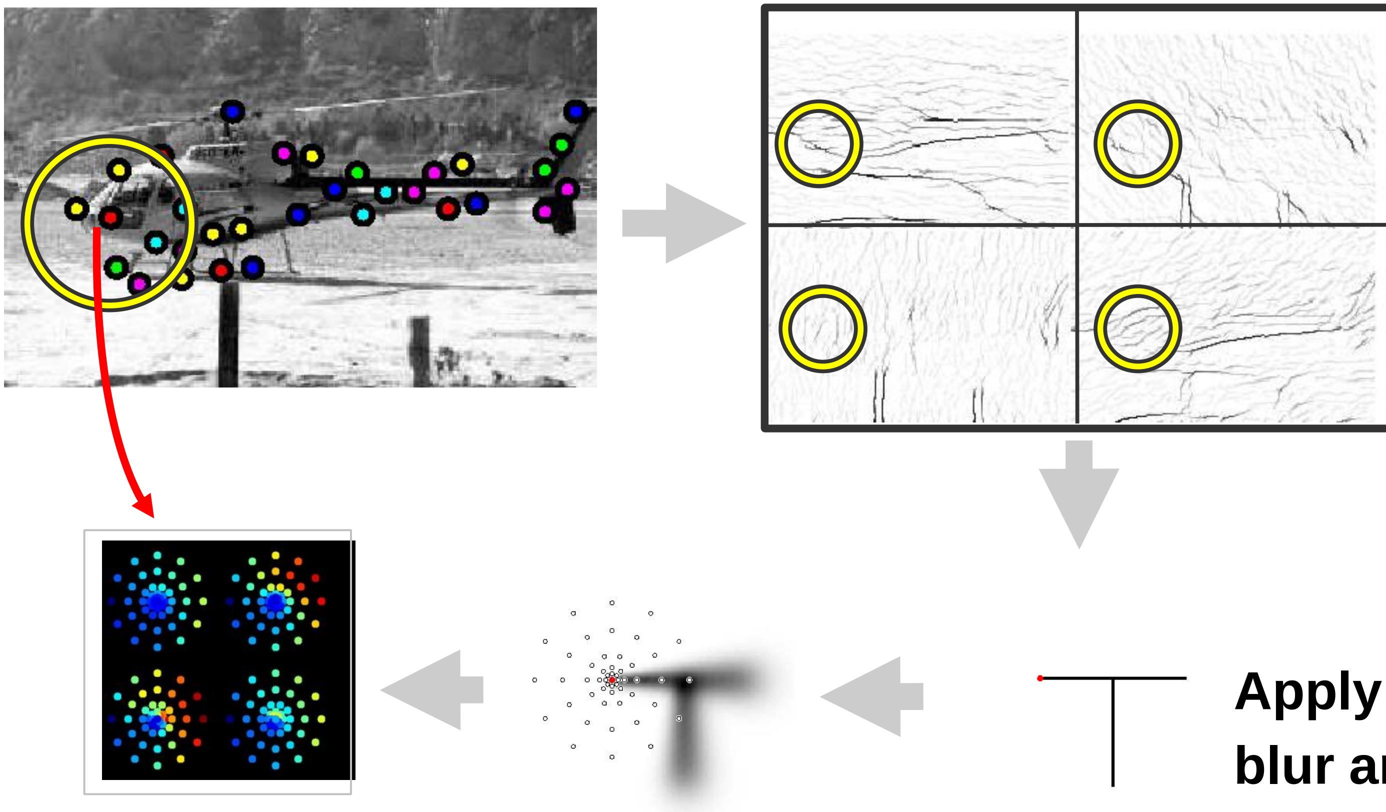
⋮

Count = 10

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

Belongie & Malik, ICCV 2001

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

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

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			√	√	(√)	(√)	+	+	+	+

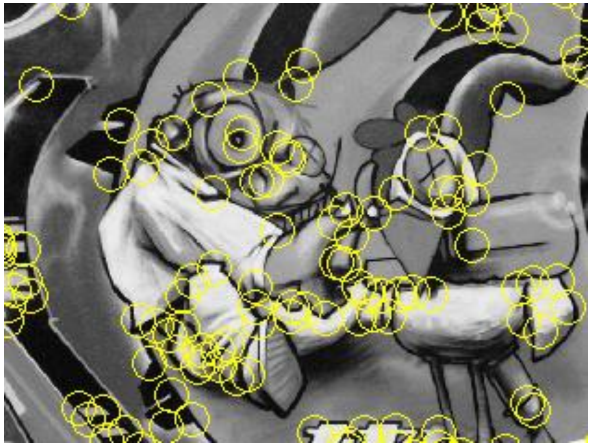
Tuytelaars Mikolajczyk 2008

Choosing a descriptor

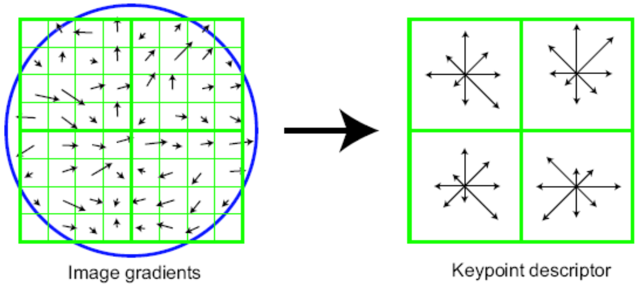
- Again, need not stick to one
- For object instance recognition or stitching, SIFT or variant is a good choice

Things to remember

- Keypoint detection: repeatable and distinctive
  - Corners, blobs, stable regions
  - Harris, DoG



- Descriptors: robust and selective
  - spatial histograms of orientation
  - SIFT



Next time

- Feature tracking