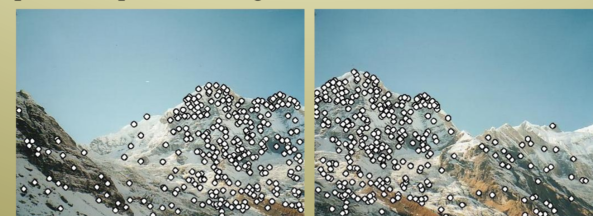
# Feature Point Detection and Matching

## Wide Baseline Matching

• Images taken by cameras that are far apart make the ***correspondence problem*** very difficult

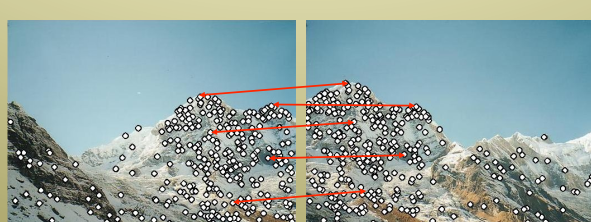
• Feature-based approach: Detect and match feature points in pairs of images



## Matching with Features

• Detect feature points

• Find corresponding pairs



## Matching with Features

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

**no chance to match!**

We need a **repeatable detector**

## Matching with Features

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


We need a reliable and distinctive **descriptor**

## Properties of an Ideal Feature

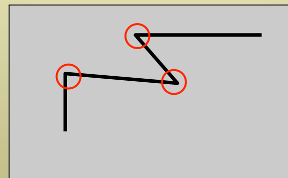
- **Local:** features are local, so robust to occlusion and clutter (no prior segmentation)
- **Invariant** (or covariant) to many kinds of geometric and photometric transformations
- **Robust:** noise, blur, discretization, compression, etc. do not have a big impact on the feature
- **Distinctive:** individual features can be matched to a large database of objects
- **Quantity:** many features can be generated for even small objects
- **Accurate:** precise localization
- **Efficient:** close to real-time performance

## Problem 1: Detecting Good Feature Points

## Feature Detectors

- Hessian
- Harris
- Lowe: SIFT (DoG)
- Mikolajczyk & Schmid: Hessian/Harris-Laplacian/Affine
- Tuytelaars & Van Gool: EBR and IBR
- Matas: MSER
- Kadir & Brady: Salient Regions
- and many others

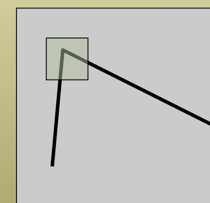
## Harris Corner Point Detector



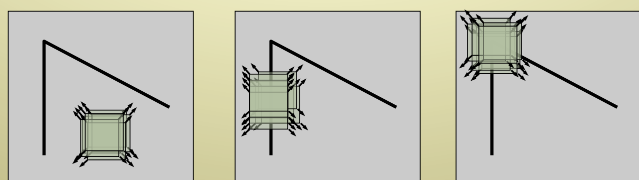
C. Harris, M. Stephens, "A Combined Corner and Edge Detector," 1988

## Harris Detector: Basic Idea

- We should recognize the point by looking through a small window
- Shifting a window in ***any** direction* should give *a **large** change* in response



## Harris Detector: Basic Idea



"flat" region:
no change in
all directions

"edge":
no change along
the edge direction

"corner":
significant change
in *all* directions

## Harris Detector: Derivation

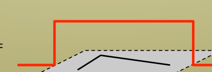
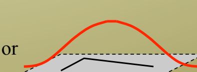
Change of intensity for a (small) shift by $[u,v]$ in image $I$:

$$E_{x_0,y_0}(u,v) = \sum_{(x,y)\in N(x_0,y_0)} w(x,y)[I(x+u,y+v)-I(x,y)]^2$$

Weighting function

Shifted intensity

Intensity

Weighting function $w(x,y) =$  or 

1 in window, 0 outside

Gaussian

## Harris Detector

Approximate using 1st order Taylor series expansion of *I*:

$$I(x+u, y+v) \approx I(x,y) + I_x u + I_y v$$

$$\approx I(x,y) + \begin{bmatrix} I_x & I_y \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}$$

Plugging this into previous formula, we get:

$$E(u,v) = Au^2 + 2Cuv + Bv^2$$

$$A = \sum_{x,y} w(x,y) I_x^2(x,y)$$

$$B = \sum_{x,y} w(x,y) I_y^2(x,y)$$

$$C = \sum_{x,y} w(x,y) I_x(x,y) I_y(x,y)$$

$$E(u,v) = \begin{bmatrix} u & v \end{bmatrix} \begin{bmatrix} A & C \\ C & B \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}$$

where $I_x = \partial I(x,y)/\partial x$

$I_y = \partial I(x,y)/\partial y$

## Harris Corner Detector

In summary, expanding $E(u,v)$ in a Taylor series, we have, for small shifts, $[u,v]$, a *bilinear* approximation:

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

where **M** is a 2 x 2 matrix computed from image derivatives:

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

$I_x = \partial I(x,y)/\partial x$

$I_y = \partial I(x,y)/\partial y$

Note: Sum computed over small neighborhood around given pixel

## Harris Corner Detector

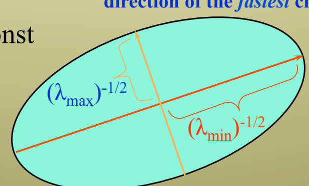
Intensity change in shifting window: eigenvalue analysis

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

$\lambda_{max}$, $\lambda_{min}$ – **eigenvalues** of $\mathbf{M}$

**Eigenvector associated with $\lambda_{max}$ = direction of the *fastest* change in $E$**

Ellipse $E(u,v)$ = const




**Eigenvector = direction of the *slowest* change in $E$**

## Selecting Good Features

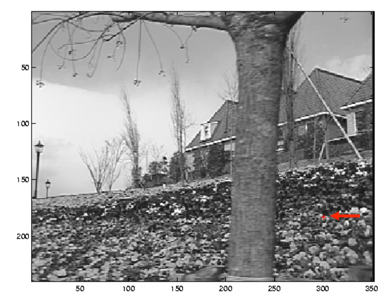
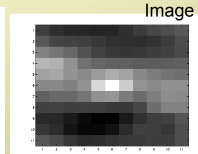
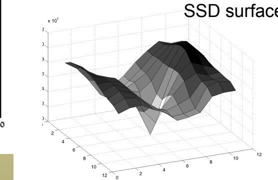



Image patch

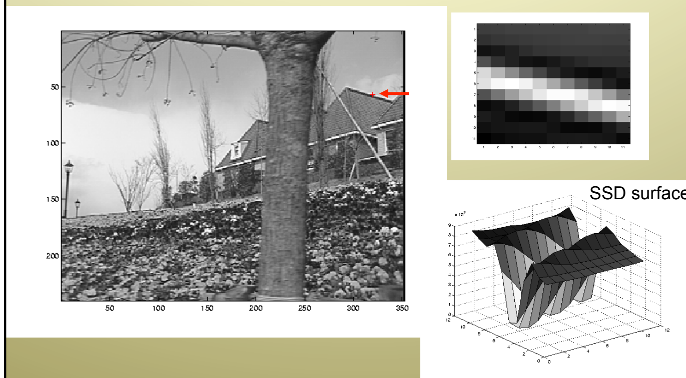


SSD surface



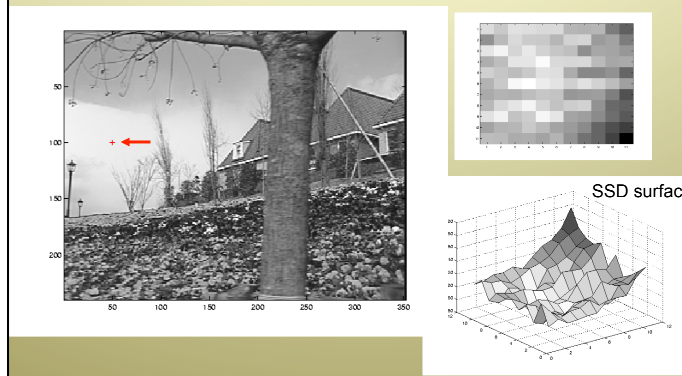
$\lambda_1$ and $\lambda_2$ both large

## Selecting Good Features




large $\lambda_1$, small $\lambda_2$

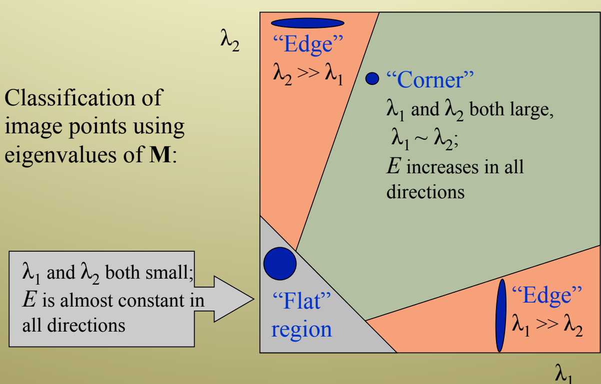
## Selecting Good Features




small $\lambda_1$, small $\lambda_2$

## Harris Corner Detector

Classification of image points using eigenvalues of **M**:




## Harris Corner Detector

Measure of corner response:

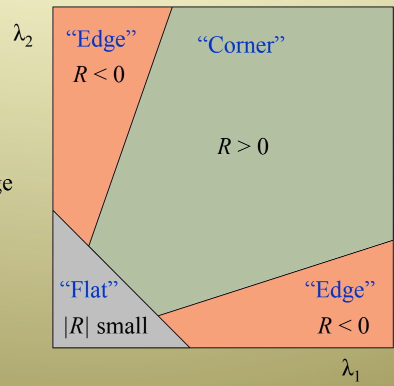
$$R = \det \mathbf{M} - k(\operatorname{trace} \mathbf{M})^2$$

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

$k$ is an empirically-determined constant; e.g., $k = 0.05$

## Harris Corner Detector

- $R$ depends only on eigenvalues of $\mathbf{M}$
- $R$ is large for a corner
- $R$ is negative with large magnitude for an edge
- $|R|$ is small for a flat region




## Harris Corner Detector: Algorithm

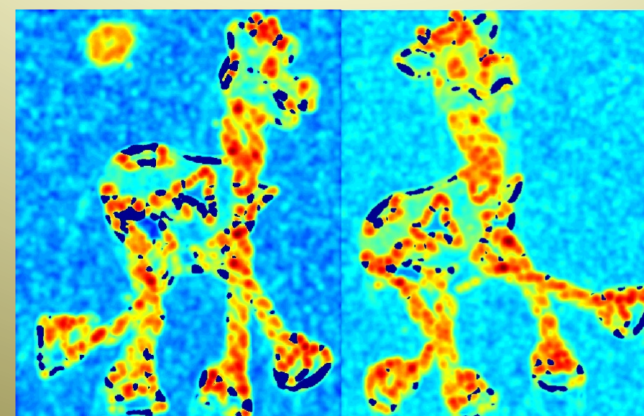
1. Find points with large corner response function $R$
   (i.e., $R$ > threshold)

2. Keep the points that are local maxima of $R$
   (i.e., value of $R$ at a pixel is greater than the values of $R$ at *all* neighboring pixels)

## Harris Detector: Example

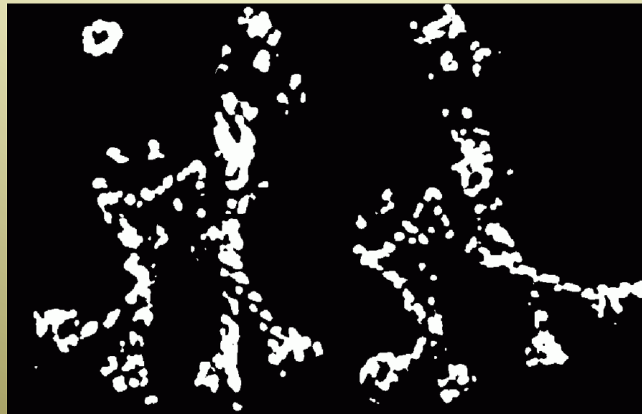
## Harris Detector: Example

Corner response $R = \lambda_1\lambda_2 - k(\lambda_1 + \lambda_2)^2$



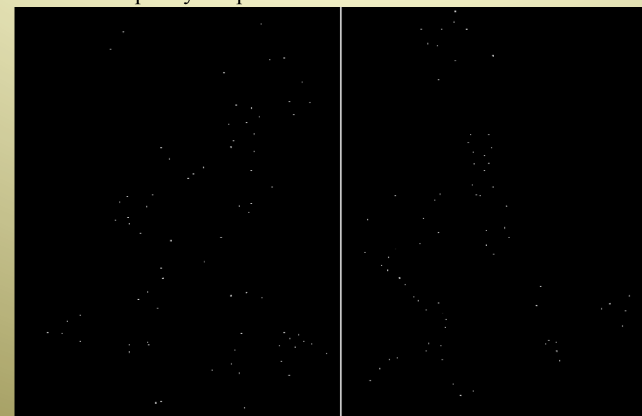
## Harris Detector: Example

Points with large corner response: $R >$ threshold



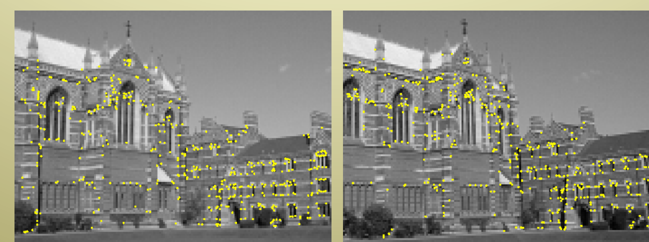
## Harris Detector: Example

Keep only the points that are local maxima of $R$



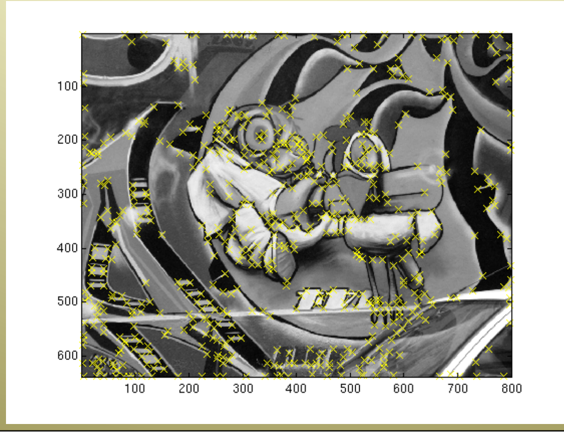
## Harris Detector: Example

## Harris Detector: Example



Interest points extracted with Harris (~ 500 points)

## Harris Detector: Example




## Harris Detector: Summary

- Average intensity change in direction [*u*,*v*] can be expressed (approximately) in bilinear form:

$$E(u,v) \cong [u \;\; v]\,\mathbf{M}\begin{bmatrix} u \\ v \end{bmatrix}$$

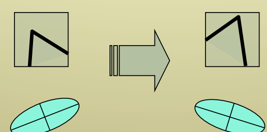
- Describe a point in terms of the eigenvalues of **M**: *measure of "cornerness":*

$$R = \lambda_1\lambda_2 - k(\lambda_1 + \lambda_2)^2$$

- A good (corner) point should have a *large intensity change* in *most directions*, i.e., *R* should be a large positive value

## Harris Detector Properties

**Rotation invariance**

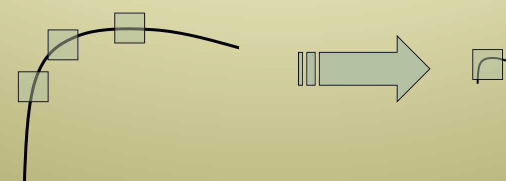


Ellipse rotates but its shape (i.e., eigenvalues) remains the same

*Corner response* $R$ is invariant to image rotation

## Harris Detector Properties

But ***not*** invariant to ***image scale***



Fine scale: All points will be classified as **edges**

Coarse scale: **Corner**

## Harris Detector: Scale

$R_{min}$= 0

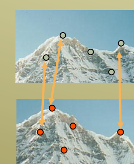
$R_{min}$= 1500

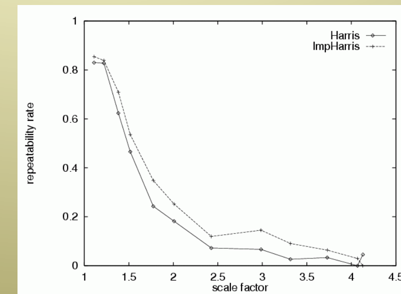
## Harris Detector Properties

Quality of Harris detector for different scale changes

Repeatability rate:

$$\frac{\text{\# correct correspondences}}{\text{\# possible correspondences}}$$

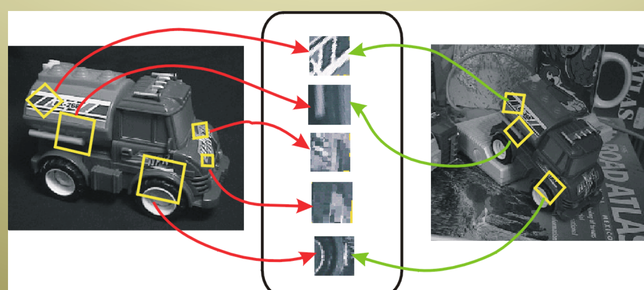




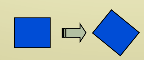
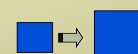
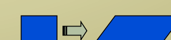
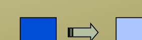

C. Schmid et al., "Evaluation of Interest Point Detectors," IJCV 2000

## Invariant Local Features

Goal: Detect *the same* interest points regardless of *image changes* due to **translation**, **rotation**, **scale**, and **viewpoint**

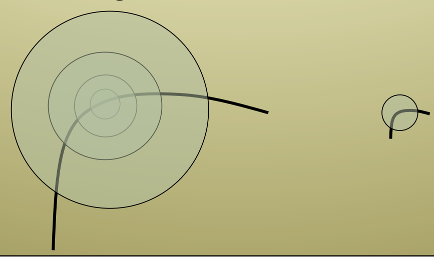


## Models of Image Change

- Geometry
  - Rotation 
  - Similarity (rotation + uniform scale) 
  - Affine (scale dependent on direction) 
    valid for: orthographic camera, locally planar object
- Photometry
  - Affine intensity change ($I \rightarrow a\ I + b$) 

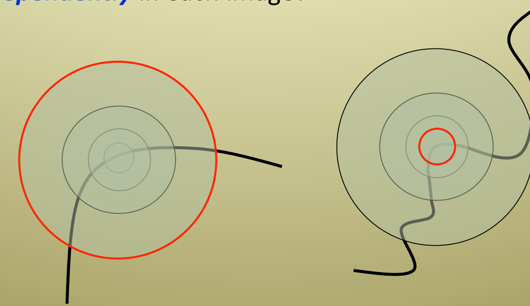
## Scale Invariant Detection

- Consider regions (e.g., circles) of different sizes around a point
- Regions of *corresponding sizes* will look the same in both images



## Scale Invariant Detection

Problem: How do we choose corresponding circles ***independently*** in each image?



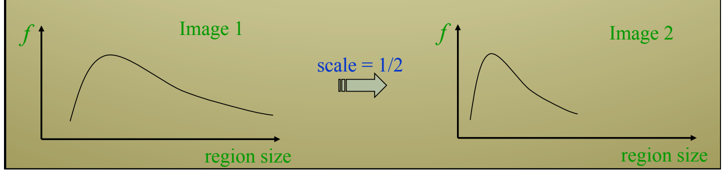
## Scale Invariant Detection

Solution:

- Design a function on the region (circle) that is "scale invariant," i.e., the same for corresponding regions, even if they are at different scales

**Example: Average intensity. For corresponding regions (even of different sizes) it will be the same**

- For a point in one image, we can consider it as a function of region size (i.e., circle radius)

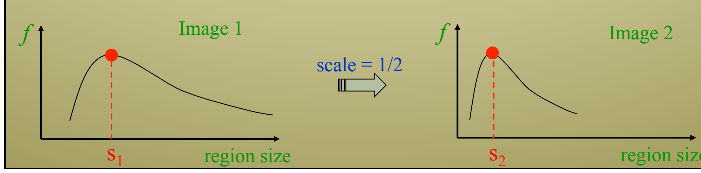



## Scale Invariant Detection

Key idea: Use the **local maximum** of this function

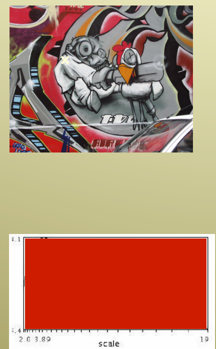
Observation: The region size, for which the maximum is achieved, should be *invariant* to image scale

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



# Automatic Scale Selection

Lindeberg *et al.*, 1996

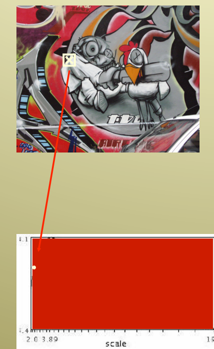



$$f(I_{i_1 \ldots i_m}(x,\sigma))$$

One possible definition of the function *f*: Absolute value of the Laplacian of Gaussian ($\nabla^2$G) (or DoG)

# Automatic Scale Selection

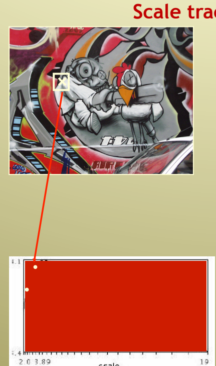
Function responses for increasing scale
Scale trace (signature)




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

# Automatic Scale Selection

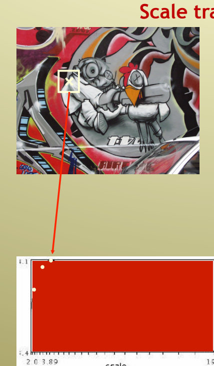
Function responses for increasing scale
Scale trace (signature)




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

# Automatic Scale Selection

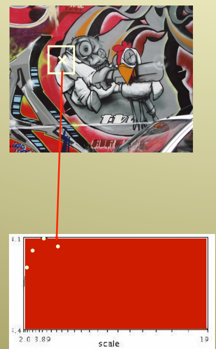
Function responses for increasing scale
Scale trace (signature)




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

## Automatic Scale Selection

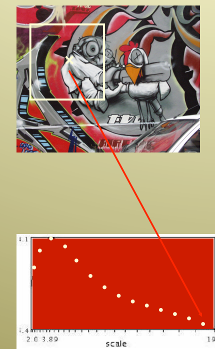
Function responses for increasing scale
Scale trace (signature)




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

## Automatic Scale Selection

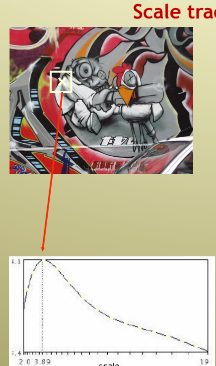
Function responses for increasing scale
Scale trace (signature)




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

## Automatic Scale Selection

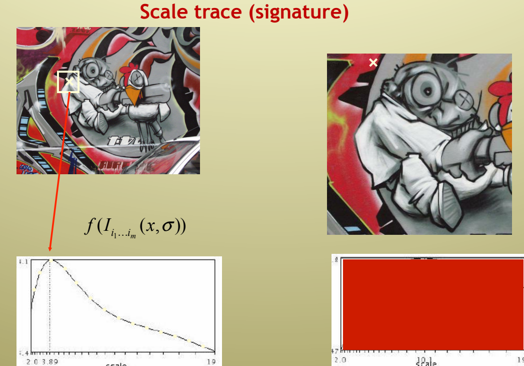
Function responses for increasing scale
Scale trace (signature)



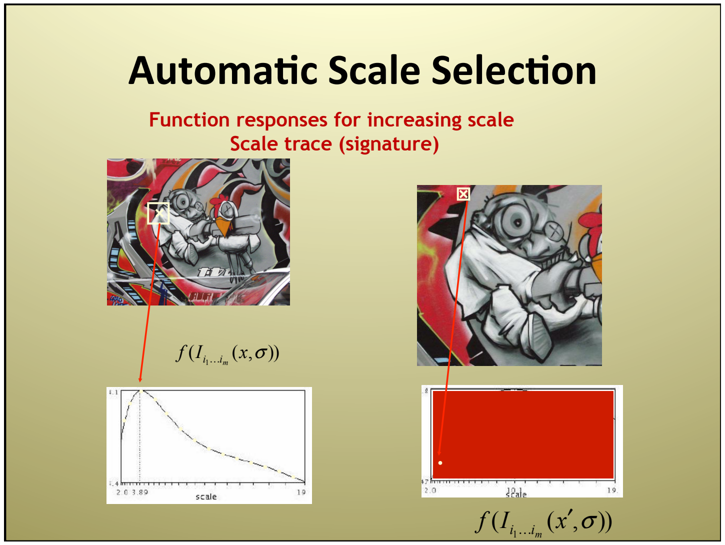

$$f(I_{i_1\ldots i_m}(x,\sigma))$$

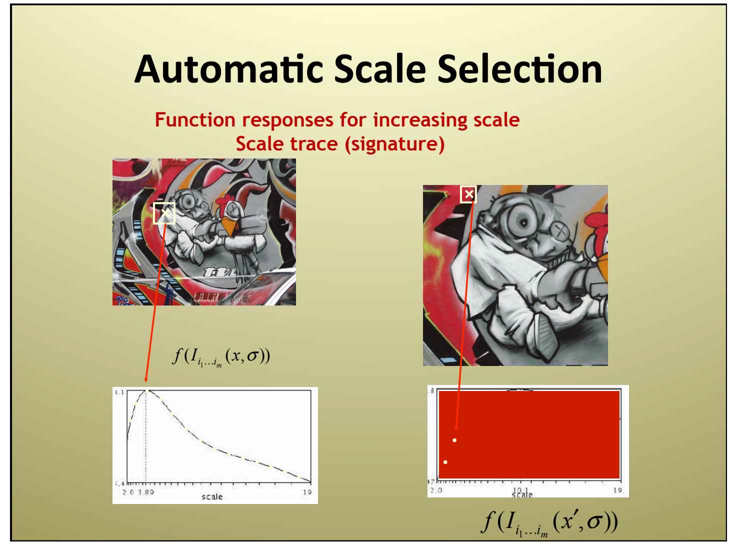
## Automatic Scale Selection

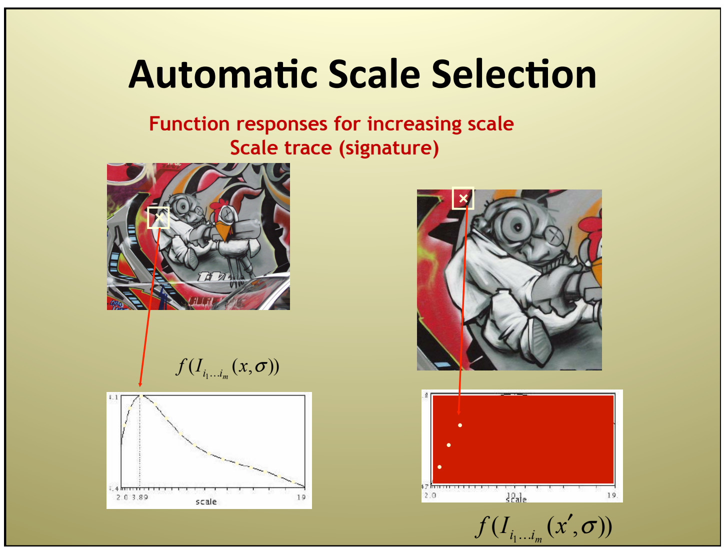
Function responses for increasing scale
Scale trace (signature)

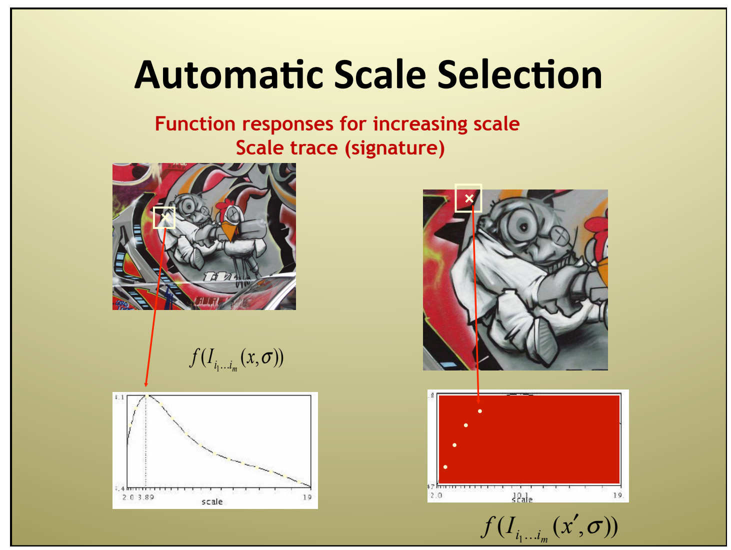



$$f(I_{i_1\ldots i_m}(x',\sigma))$$


Automatic Scale Selection
Function responses for increasing scale
Scale trace (signature)
$f(I_{i_1 \ldots i_m}(x,\sigma))$
scale
$f(I_{i_1 \ldots i_m}(x',\sigma))$

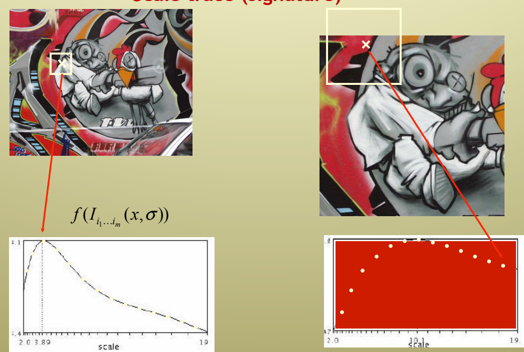

Automatic Scale Selection
Function responses for increasing scale
Scale trace (signature)
$f(I_{i_1 \ldots i_m}(x,\sigma))$
scale
$f(I_{i_1 \ldots i_m}(x',\sigma))$


Automatic Scale Selection
Function responses for increasing scale
Scale trace (signature)
$f(I_{i_1 \ldots i_m}(x,\sigma))$
scale
$f(I_{i_1 \ldots i_m}(x',\sigma))$


Automatic Scale Selection
Function responses for increasing scale
Scale trace (signature)
$f(I_{i_1 \ldots i_m}(x,\sigma))$
scale
$f(I_{i_1 \ldots i_m}(x',\sigma))$

## Automatic Scale Selection

Function responses for increasing scale
Scale trace (signature)

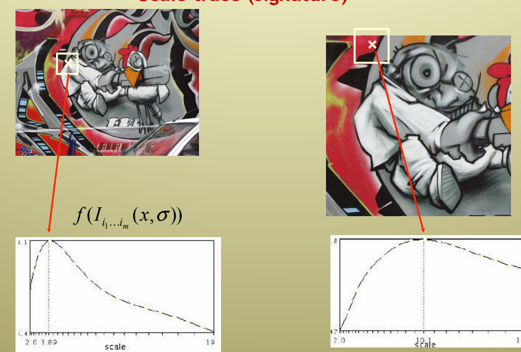



$f(I_{i_1 \ldots i_m}(x,\sigma))$

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

## Automatic Scale Selection

Function responses for increasing scale
Scale trace (signature)




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

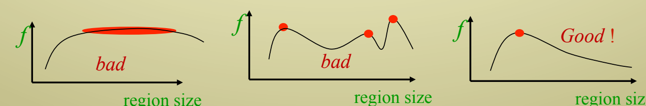
$f(I_{i_1 \ldots i_m}(x',\sigma'))$

## Automatic Scale Selection

- Normalize: rescale to fixed size

## Scale Invariant Detection

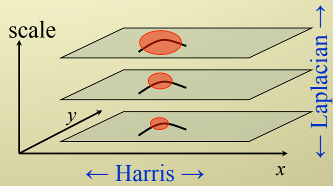
- A "good" function for scale detection has **one stable, sharp peak**



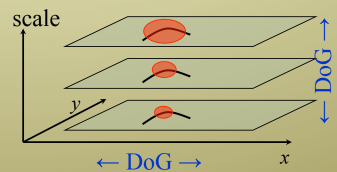

- For many images, a good function is one that responds to contrast (i.e., sharp local intensity change)

# Scale Invariant Detectors

- Harris-Laplacian[1]
  *Find local maxima of:*
  - Harris corner detector in space (image coordinates)
  - Laplacian in scale



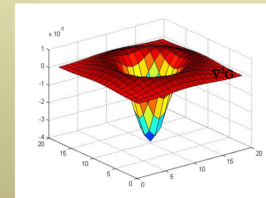

- SIFT keypoints[2]
  *Find local extrema of:*
  - Difference of Gaussians in space and scale

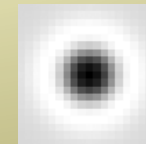



[1] K.Mikolajczyk, C.Schmid. "Indexing Based on Scale Invariant Interest Points," ICCV 2001
[2] D.Lowe. "Distinctive Image Features from Scale-Invariant Keypoints," *IJCV*, 2004

# Blob Detection

Laplacian of Gaussian ($\nabla^2$G): Circularly symmetric operator for "blob" detection

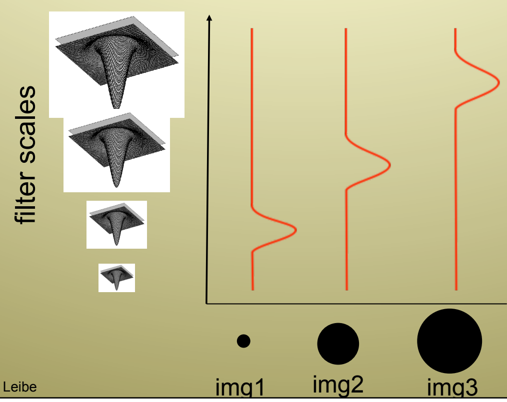




Laplacian definition: $$\nabla^2 I = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}$$

# Blob Detection: Scale Selection

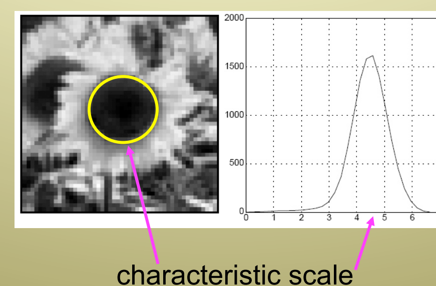
Laplacian-of-Gaussian = "blob" detector $\nabla^2 I = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}$



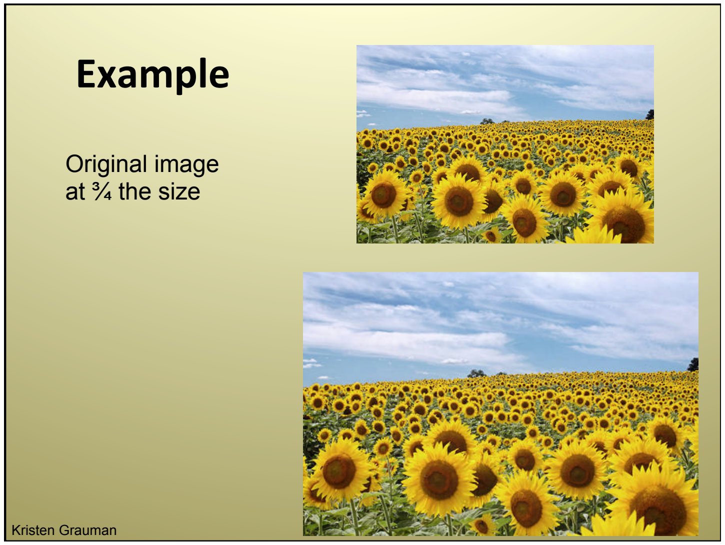

Bastian Leibe

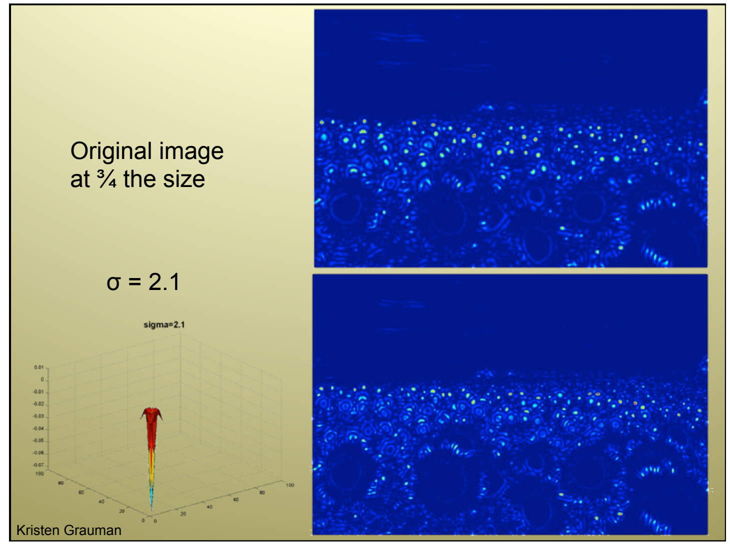
# Blob Detection in 2D

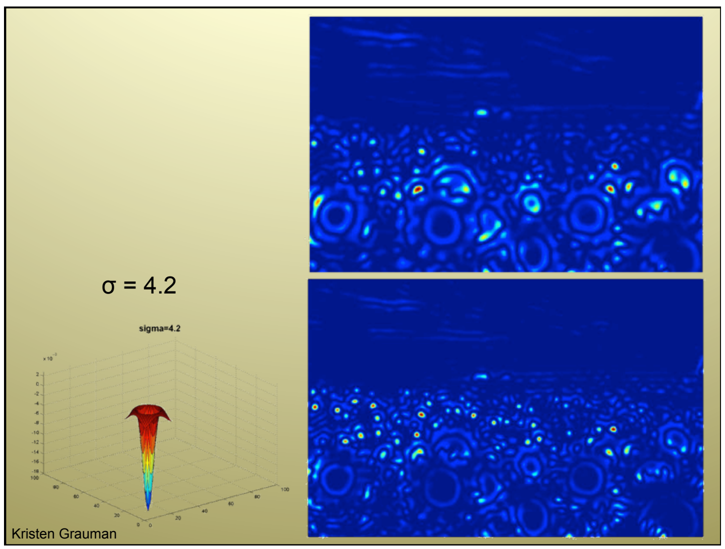
Define the *characteristic scale* as the scale that produces the *maximum* Laplacian absolute value response

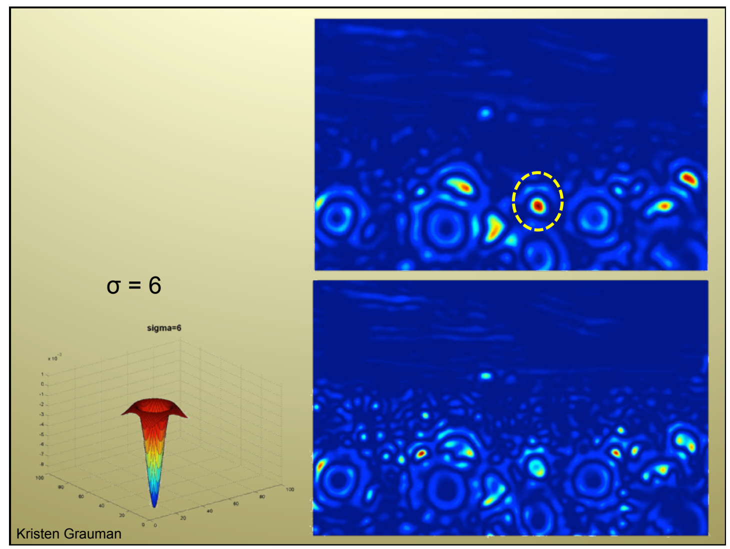


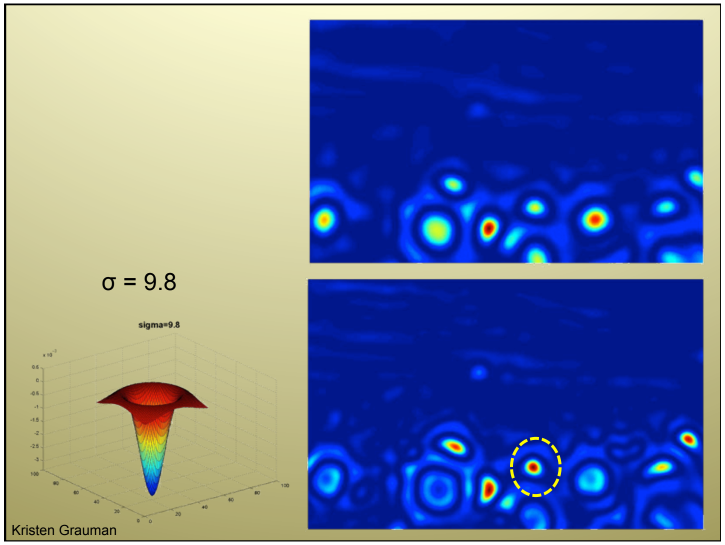

Slide credit: Lana Lazebnik

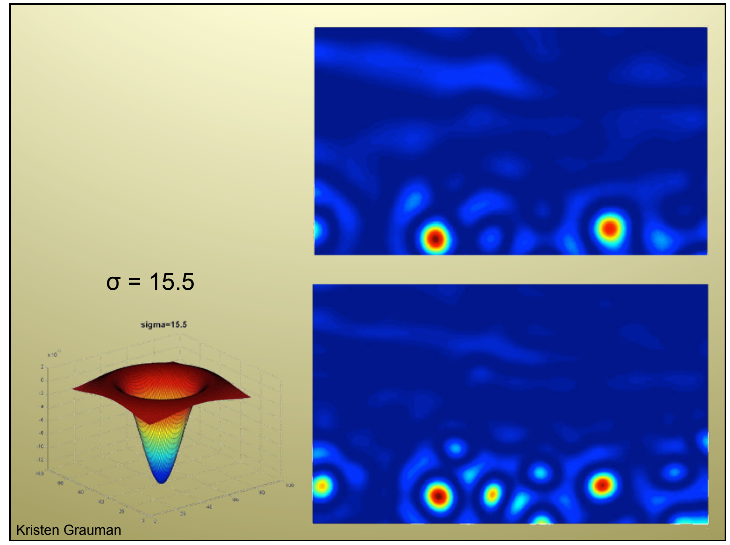

Example
Original image
at ¾ the size
Kristen Grauman

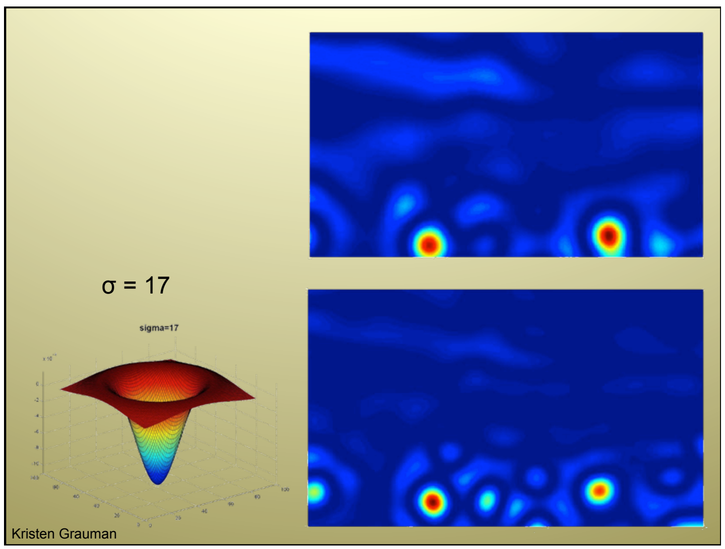

Original image
at ¾ the size
σ = 2.1
sigma=2.1
Kristen Grauman

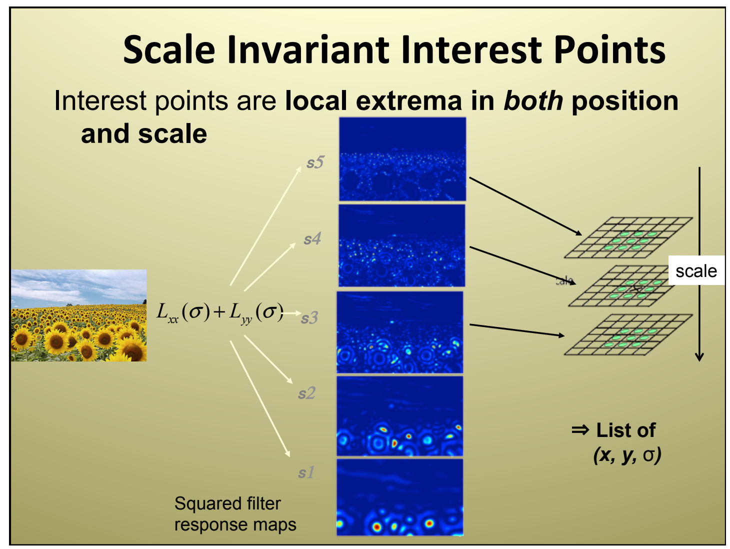

σ = 4.2
sigma=4.2
Kristen Grauman


σ = 6
sigma=6
Kristen Grauman


σ = 9.8
sigma=9.8
Kristen Grauman


σ = 15.5
sigma=15.5
Kristen Grauman

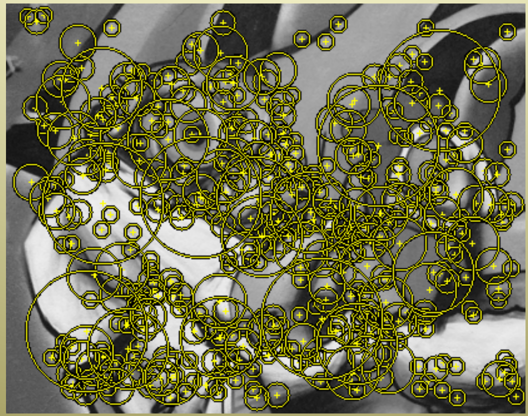

σ = 17
sigma=17
Kristen Grauman


Scale Invariant Interest Points
Interest points are local extrema in both position and scale
s5
s4
s3
s2
s1
$L_{xx}(\sigma)+L_{yy}(\sigma)$
scale
⇒ List of (x, y, σ)
Squared filter response maps

## SIFT Detector Example Result

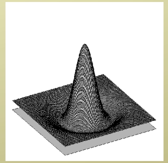
Image credit: Lana Lazebnik

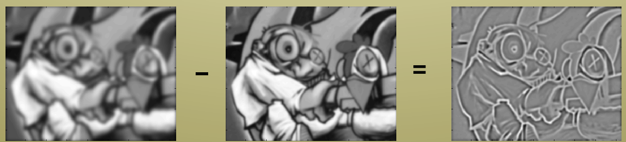
## SIFT Detector Example Result



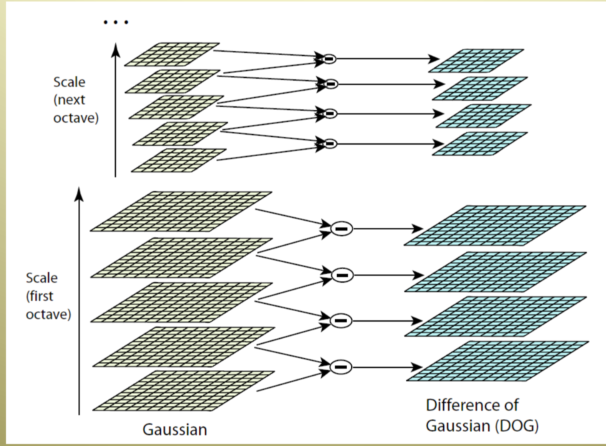
## SIFT Detector Details

- Difference-of-Gaussian (DoG) is an approximation of the Laplacian-of-Gaussian (LoG)





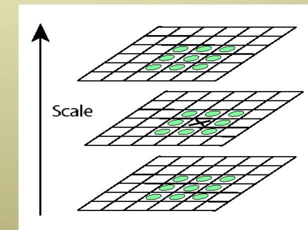
## SIFT Detector



## SIFT Detector

## SIFT Detector Algorithm Summary

- Detect local maxima in position and scale of squared values of Difference-of-Gaussian
- Fit a quadratic to surrounding values for sub-pixel and sub-scale interpolation
- Output = list of ($x$, $y$, $\sigma$) points

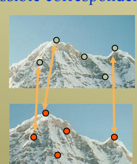


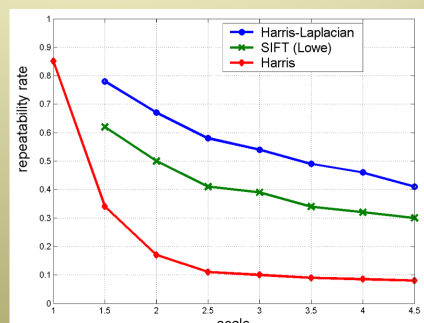

## Scale Invariant Detectors

- Experimental evaluation of detectors w.r.t. scale change

Repeatability rate:

$$\frac{\text{\# correspondences}}{\text{\# possible correspondences}}$$

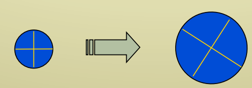




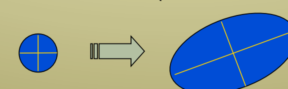

K.Mikolajczyk, C.Schmid. "Indexing Based on Scale Invariant Interest Points," ICCV 2001

## Affine Invariant Detection

- Previously we considered:
  **similarity transform** (rotation + uniform scale)

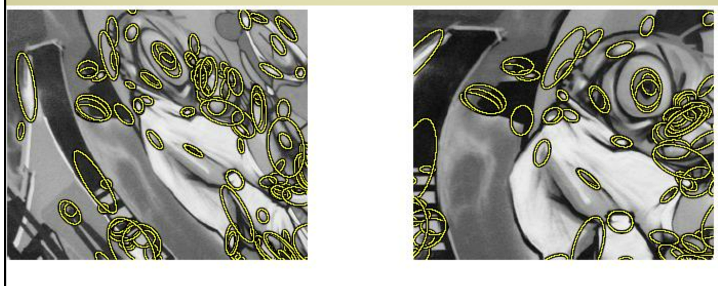


- Now we go on to invariance under an
  **affine transformation** (rotation + scale + skew)

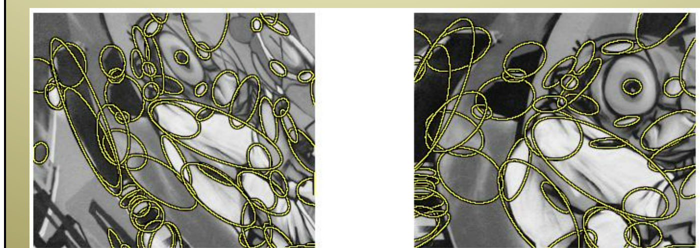


Circle projects to an ellipse in general

## Harris-Affine Detector

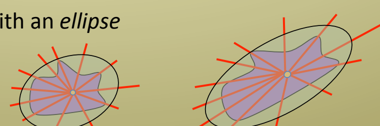


## Tuytelaars's Affine-Invariant Detector

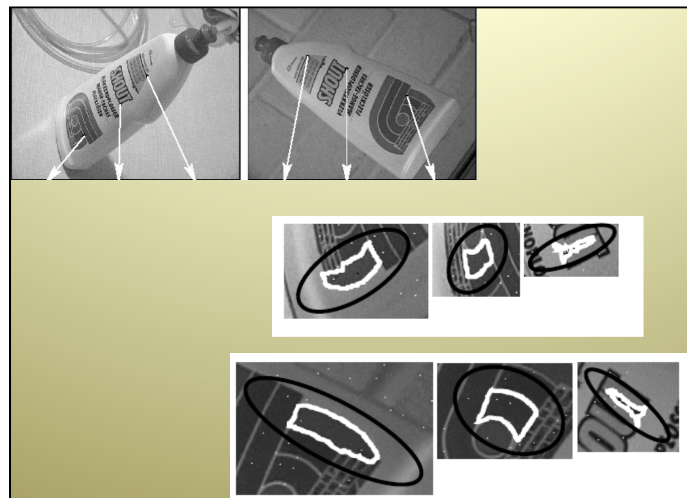


## Affine Invariant Region Detection

- Algorithm
  - Start from a *local intensity extremum* point
  - Go in *every direction* until the point of extremum of some function $f$
  - Curve connecting the points is the region boundary
  - Compute *geometric moments* of orders up to 2 for this region
  - Replace the region with an *ellipse*

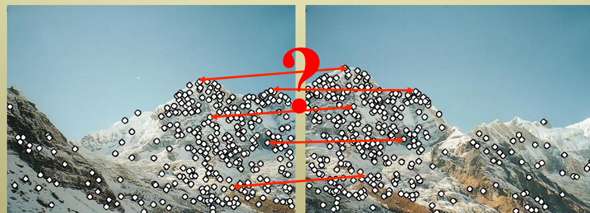


T.Tuytelaars, L.V.Gool. "Wide Baseline Stereo Matching Based on Local, Affinely Invariant Regions," BMVC 2000



## Feature Point Descriptors

- After detecting points (and patches) in each image,
- Next question: **How to match them?**



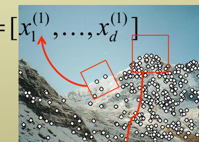

Point descriptor should be:
1. Invariant
2. Distinctive

## Local Features: Description

1) Detection: Identify the interest points

2) Description: Extract feature vector for each interest point

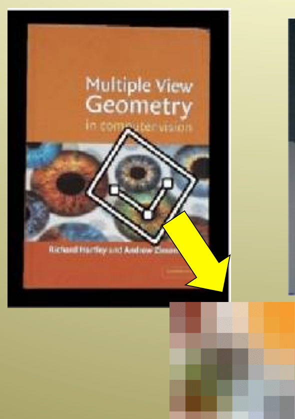
$\mathbf{x}_1 = [x_1^{(1)}, \ldots, x_d^{(1)}]$

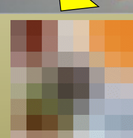


$\mathbf{x}_2 = [x_1^{(2)}, \ldots, x_d^{(2)}]$

3) Matching: Determine correspondence between descriptors in two views

## Geometric Transformations



e.g. scale, translation, rotation

## Photometric Transformations

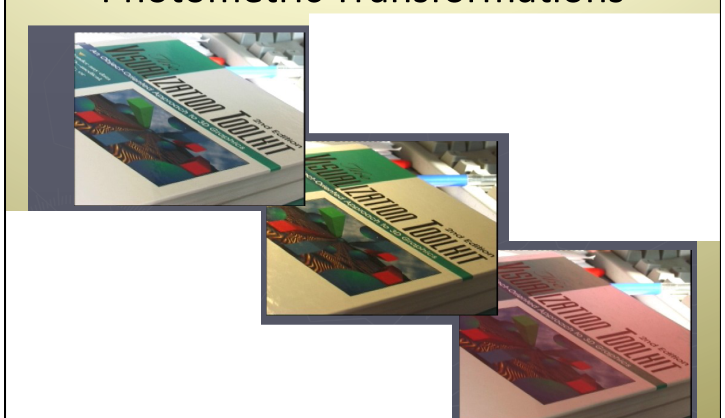
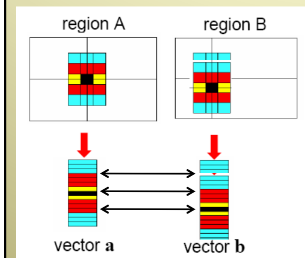


Figure from T. Tuytelaars ECCV 2006 tutorial

## Raw Patches as Local Descriptors




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

But this is *very sensitive to even small shifts or rotations*

## Making Descriptors Invariant to Rotation

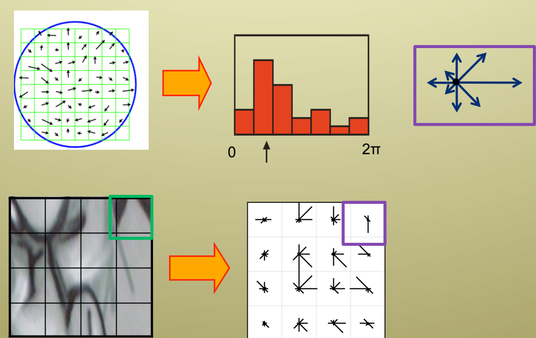
1. Find local dominant orientation

Direction of gradient:

2. Compute description ***relative*** to this orientation

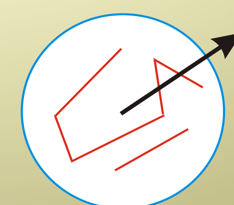
[1] K.Mikolajczyk, C.Schmid. “Indexing Based on Scale Invariant Interest Points”. ICCV 2001
[2] D.Lowe. “Distinctive Image Features from Scale-Invariant Keypoints”. Accepted to IJCV 2004

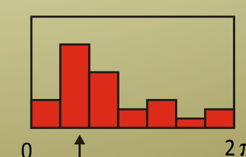
## SIFT Descriptor




## SIFT Descriptor

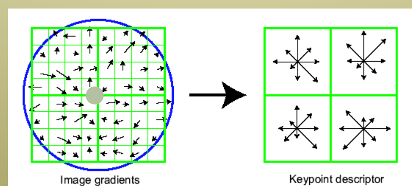
- Compute **histogram of local gradient directions** computed at selected scale in the neighborhood of a feature point *to obtain its dominant local orientation*
- Compute gradients within sub-patches, and compute *histogram of orientations* using discrete “bins” relative to dominant orientation direction
- Descriptor is rotation and scale invariant, and also has some illumination invariance





# SIFT Descriptor

- Compute gradient orientation histograms on 4 x 4 neighborhoods over 16 x 16 array of locations in scale space around each feature point position, relative to the feature point orientation using thresholded image gradients from Gaussian pyramid level at feature point's scale
- Quantize orientations to 8 values
- 4 x 4 array of histograms
- SIFT feature vector of length 4 x 4 x 8 = 128 values for each feature point
- Normalize the descriptor to make it invariant to intensity change




D.Lowe. "Distinctive Image Features from Scale-Invariant Keypoints," IJCV 2004

# Sensitivity to Number of Histogram Orientations

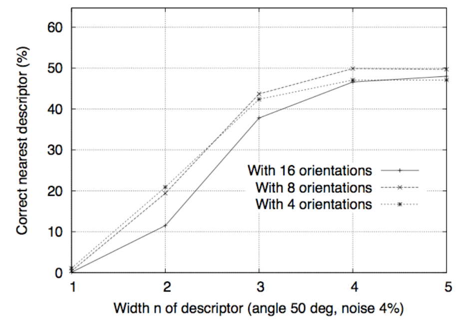
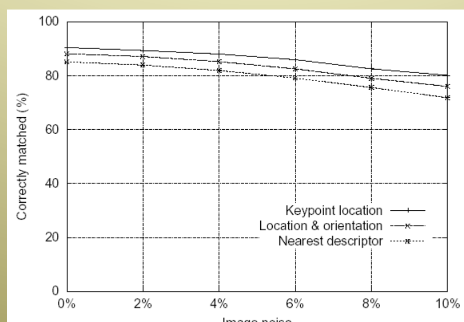



Figure 8: This graph shows the percent of keypoints giving the correct match to a database of 40,000 keypoints as a function of width of the $n \times n$ keypoint descriptor and the number of orientations in each histogram. The graph is computed for images with affine viewpoint change of 50 degrees and addition of 4% noise.

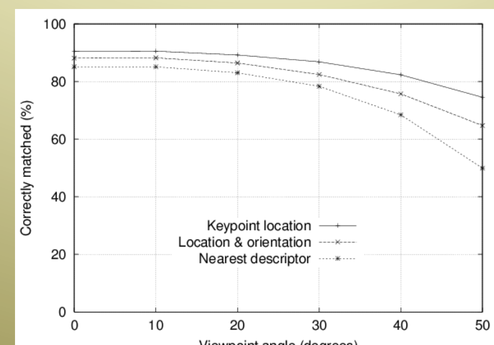
# Feature Stability to Noise

- Match features after random change in image scale & orientation, with differing levels of image noise
- Find nearest neighbor in database of 30,000 features



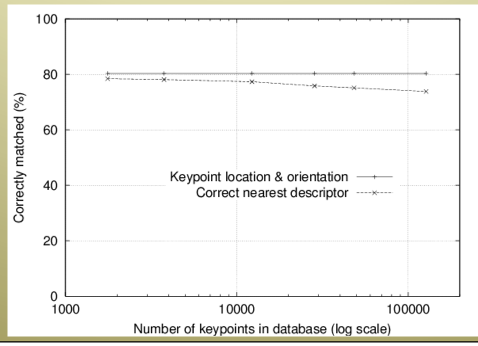

# Feature Stability to Affine Change

- Match features after random change in image scale & orientation, with 2% image noise, and affine distortion
- Find nearest neighbor in database of 30,000 features



## Distinctiveness of Features

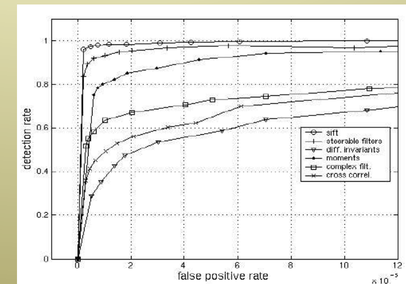
- Vary size of database of features, with 30 degree affine change, 2% image noise
- Measure % correct for single nearest neighbor match




## SIFT Descriptor Performance

Empirically found to have good performance in terms of invariance to *image rotation*, *scale*, *intensity change*, and to moderate *affine* transformations, illumination changes, viewpoint, occlusion

Scale = 2.5
Rotation = $45^0$




[1] D.Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," IJCV 2004
[2] K.Mikolajczyk, C.Schmid, "A Performance Evaluation of Local Descriptors," CVPR 2003

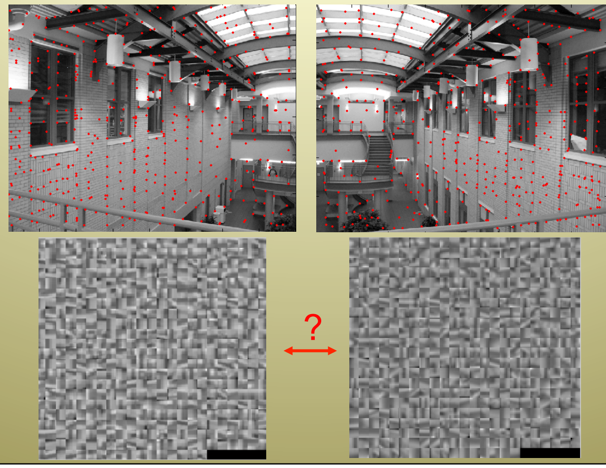
## Feature Detection and Description Summary

- Stable (repeatable) feature points can currently be detected that are invariant to
  - Rotation, scale, and affine transformations, but ***not*** to more general perspective and projective transformations
- Feature point descriptors can be computed, ***but***
  - are *noisy* due to use of differential operators
  - are *not* invariant to projective transformations

## How to Improve Feature Detectors and Descriptors?

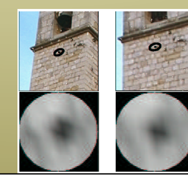
- Invariant to large changes in camera viewpoint, i.e., projective invariance
- Robust to image noise by using descriptor based on integral invariants instead of differential features
- Incorporate color and texture into descriptor

## Feature Matching




## Feature Matching

- Standard approach for pairwise matching:
  - For each feature point in image A
  - Find the feature point with the closest descriptor in image B
    - Euclidean distance between descriptors
    - Angle between unit-length normalized vectors



From Schaffalitzky and Zisserman '02

## Feature Matching

- Compare the distance, *d1*, to the closest feature, to the distance, *d2*, to the second closest feature
- Accept if *d1*/*d2* < 0.6
  - If the ratio of distances is less than a threshold, keep the feature
- Why the ratio test?
  - Eliminates hard-to-match repeated features
  - Distances in SIFT descriptor space seem to be non-uniform

## Feature Matching

- Exhaustive search
  - for each feature in one image, look at *all* the other features in the other image(s)
- Hashing
  - compute a short descriptor from each feature vector, or hash longer descriptors (randomly)
- Nearest neighbor techniques
  - *kd*-trees and variants

## Wide-Baseline Feature Matching

- Because of the high dimensionality of features, ***approximate nearest neighbors*** are often used for computational efficiency
- See ANN package, Mount and Arya
  http://www.cs.umd.edu/~mount/ANN/

## Summary

- Interest point detection
  - Harris corner detector
  - SIFT (Laplacian of Gaussian, automatic scale selection)
- Compute descriptors
  - Rotation according to dominant gradient direction
  - Histograms for robustness to small shifts and translations (SIFT descriptor)
- Match descriptors
  - Approximate nearest-neighbor in feature space