

Interest Points



Galatea of the Spheres
Salvador Dali

Computational Photography
Derek Hoiem, University of Illinois

Today's class

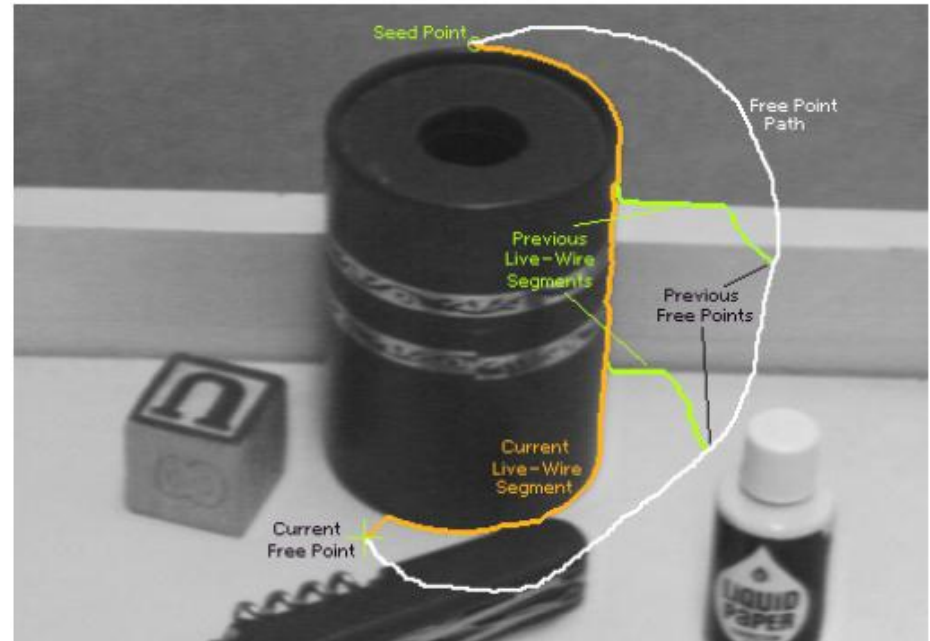
- Proj 2 favorites (thurs – server down)
 - Project: Arun
 - Result: Jiagen (Obama), Jiqin ('star night alma mater'), Arun (face toast)
- Marking up faces (due Thurs)
- Review of recent materials
- Interest points

Review of “The Digital Canvas”

- Cutting
- Growing
- Blending
- Capturing
- Measuring
- Reconstructing

Cutting: Intelligent Scissors and Graph Cuts

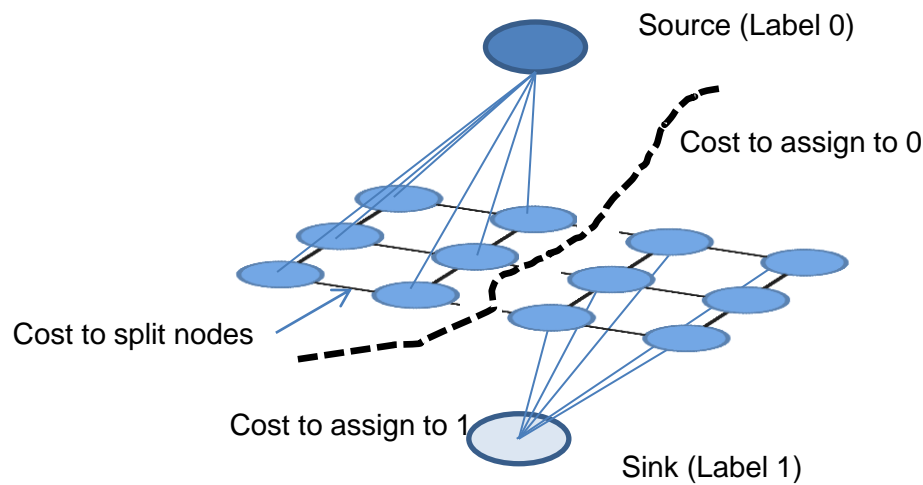
- You can treat the image as a graph
 - Nodes = pixels, edges connect neighboring pixels



Intelligent Scissors: Good boundaries are a short (high gradient) path through the graph

Cutting: Intelligent Scissors and Graph Cuts

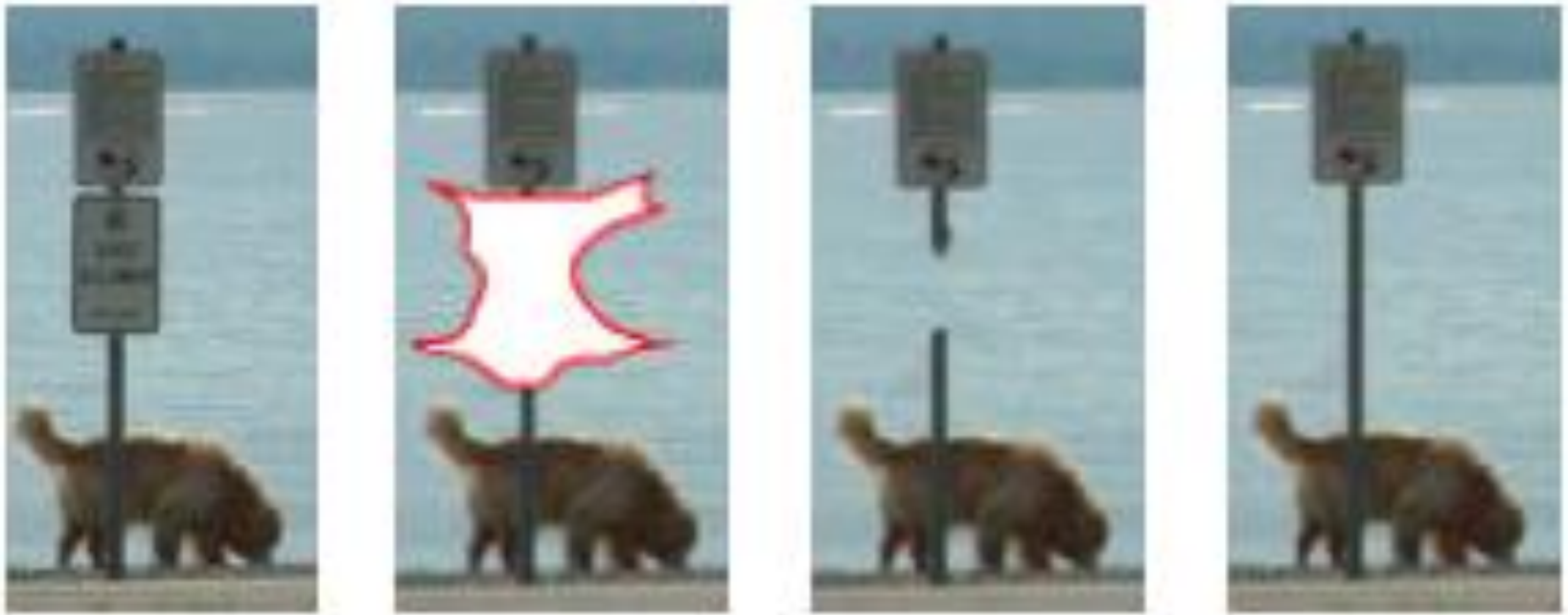
- You can treat the image as a graph
 - Nodes = pixels, edges connect neighboring pixels



Graph cut: Good boundaries are a cheap cut, where some pixels want to be foreground, and some to be background

Growing: Texture synthesis and hole-filling

- Transfer based on simple matching is a powerful idea
 - Hallucinate new pixels from their surroundings

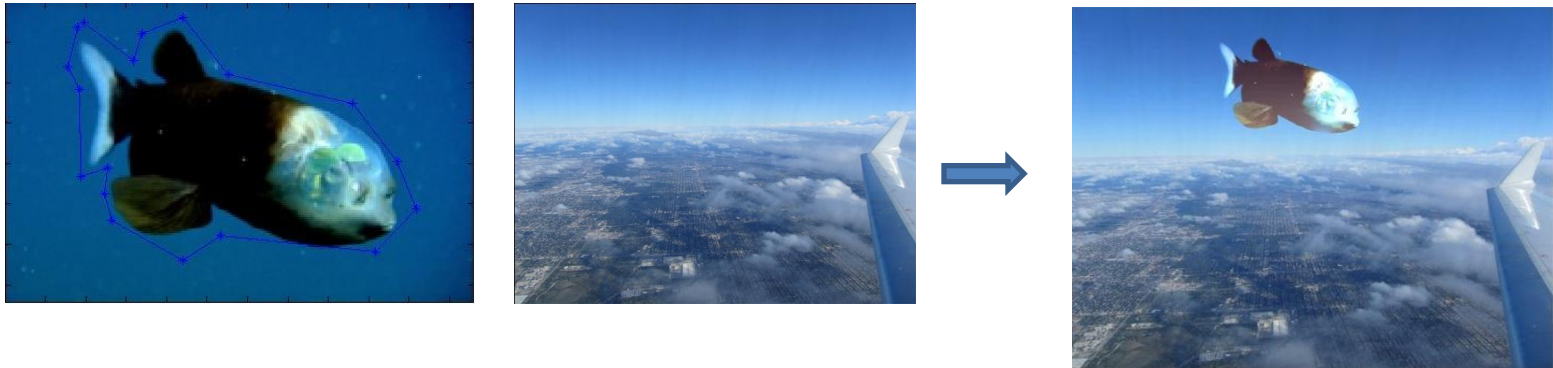


Efros & Leung, Criminisi et al.: To fill in pixels, take the center pixel from a patch that matches known surrounding values. Work from the outside in, prioritizing pixels with strong gradients.

Pasting: Compositing and Blending

- Gradient-domain fusion: Treat the pixels as variables to solve

Adair's Poisson Blending Result



Tim's Poisson Blending Result



Poisson editing: make blended gradient as close as possible to source by solving a system of linear equations.

Pasting: Compositing and Blending

- Laplacian blending: blend low-frequency slowly, high frequency quickly
 - Blend with alpha mask values ranging from 0 to 1

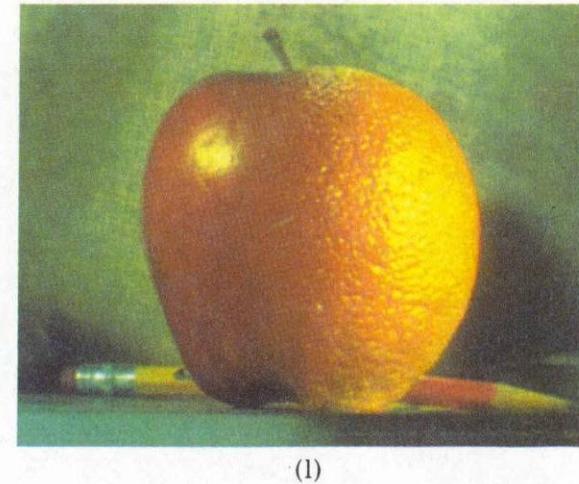
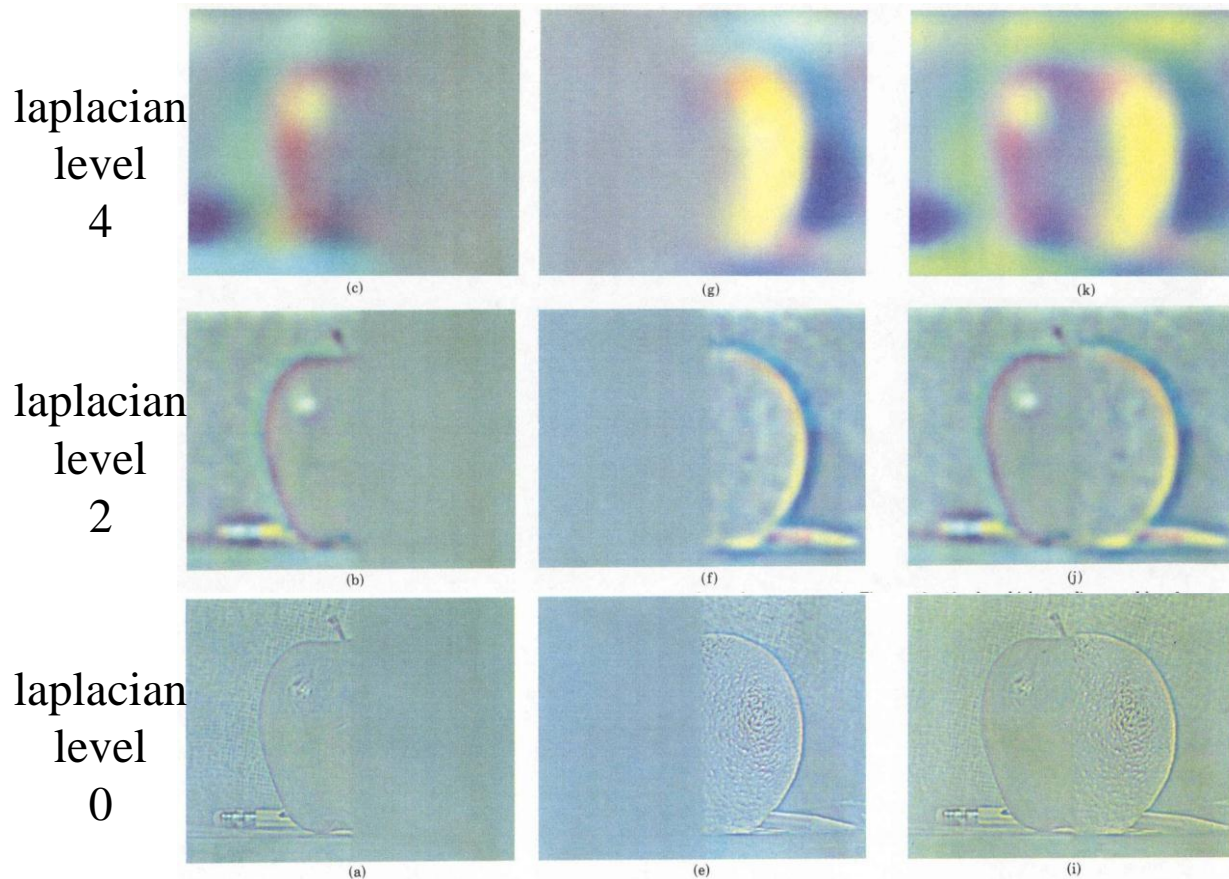







Image warping

- Map between 2D coordinates using linear projection

Name	Matrix	# D.O.F.	Preserves:	Icon
translation	$\begin{bmatrix} I & t \end{bmatrix}_{2 \times 3}$	2	orientation + ...	
rigid (Euclidean)	$\begin{bmatrix} R & t \end{bmatrix}_{2 \times 3}$	3	lengths + ...	
similarity	$\begin{bmatrix} sR & t \end{bmatrix}_{2 \times 3}$	4	angles + ...	
affine	$\begin{bmatrix} A \end{bmatrix}_{2 \times 3}$	6	parallelism + ...	
projective	$\begin{bmatrix} \tilde{H} \end{bmatrix}_{3 \times 3}$	8	straight lines	

$$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \begin{bmatrix} x \\ y \\ w \end{bmatrix}$$

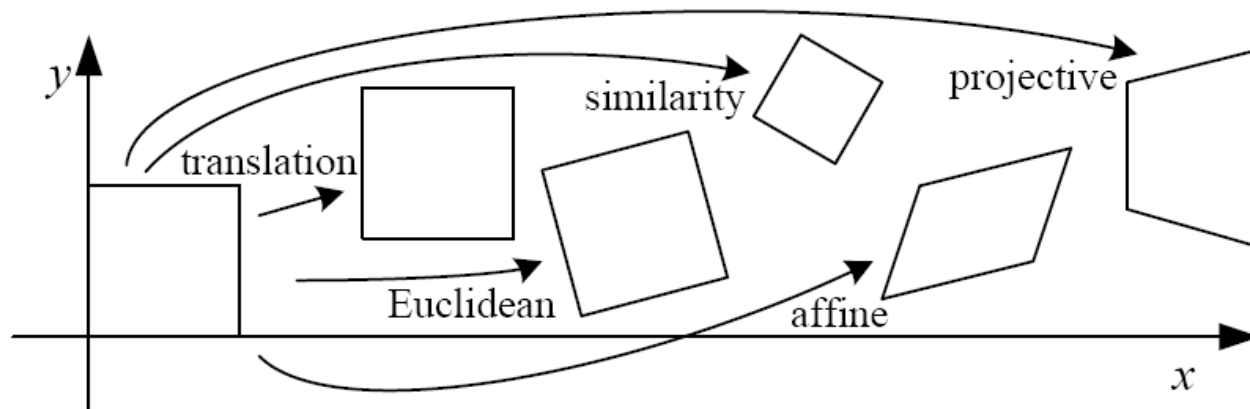
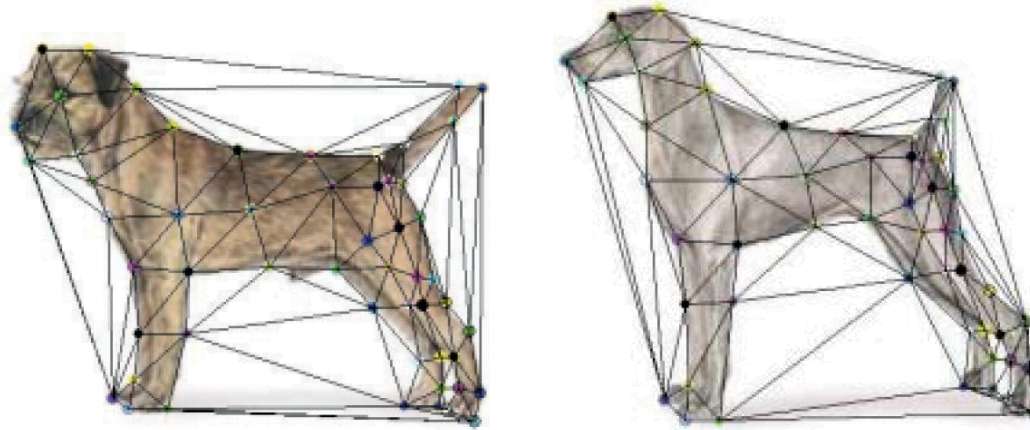
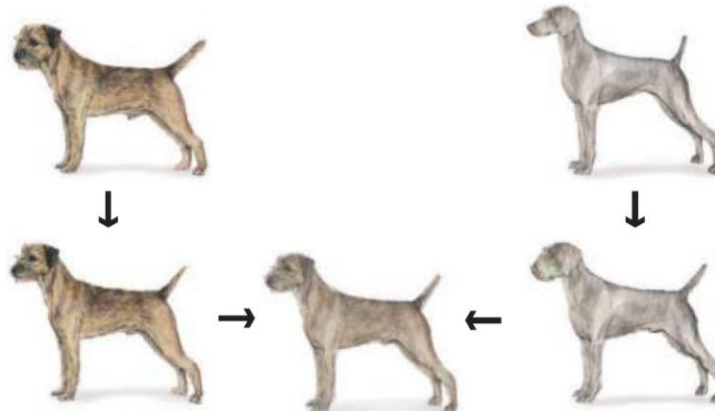


Image morphing = warp + blend

- Warp coordinates with triangular mesh



- Transfer pixels using affine transforms that map between original and intermediate triangles, then blend
 - Use mapping from source to target



Fun with faces

- Use corresponding points to average, manipulate, and extrapolate
- PCA: compactly represent a large number of variables with a few coefficients by projecting onto directions of maximum variation



Average face

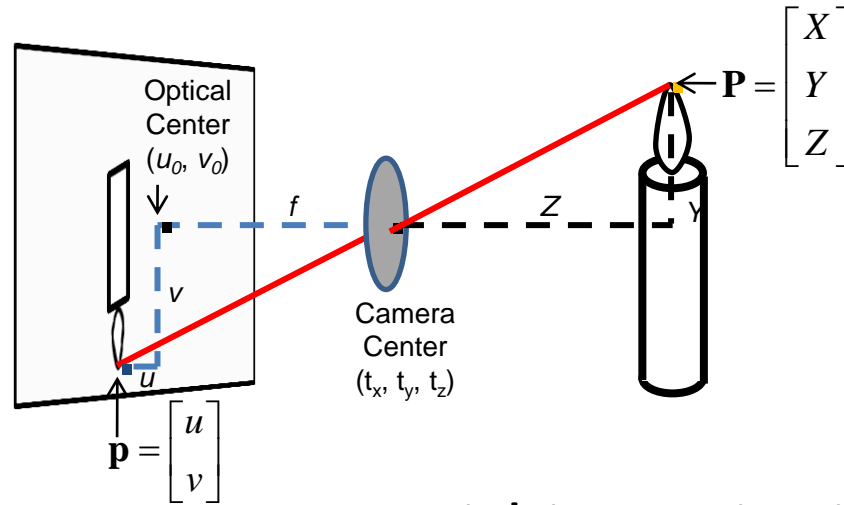


Make more feminine

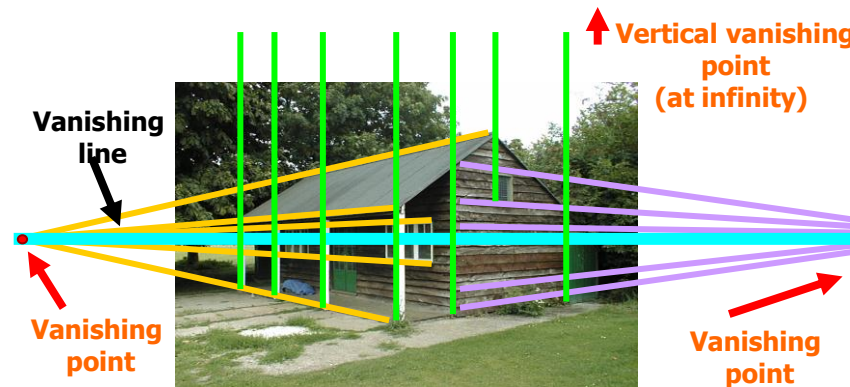
Pinhole camera model

- Linear projection from 3D to 2D
 - Be familiar with projection matrix (focal length, principal point, etc.)

$$\mathbf{x} = \mathbf{K}[\mathbf{R} \quad \mathbf{t}] \mathbf{X}$$

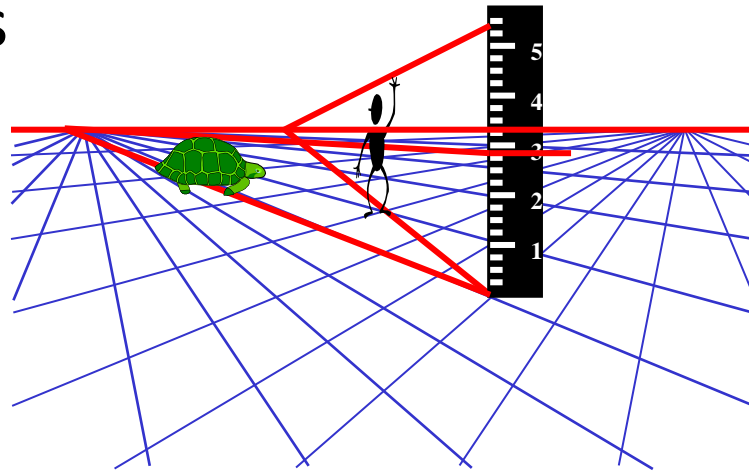


- Parallel lines in 3D intersect at a vanishing point in the image



Single-view metrology and focus

- Can measure relative object heights using vanishing point tricks

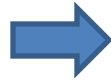


- Aperture size and focal length control amount of exposure needed, depth of field, field of view



Single-view 3D Reconstruction

- Technically impossible to go from 2D to 3D, but we can do it with simplifying models
 - Need some interaction or recognition algorithms
 - Uses basic VP tricks and projective geometry



Next section of topics

- Correspondence
 - How do we find matching patches in two images?
 - How can we automatically align two images of the same scene?
 - How do we find images with similar content?
 - How do we tell if two pictures are of the same person's face?
 - How can we detect objects from a particular category?
- Applications
 - Photo stitching
 - Object recognition
 - 3D Reconstruction

How can we align two pictures?

- Case of global transformation

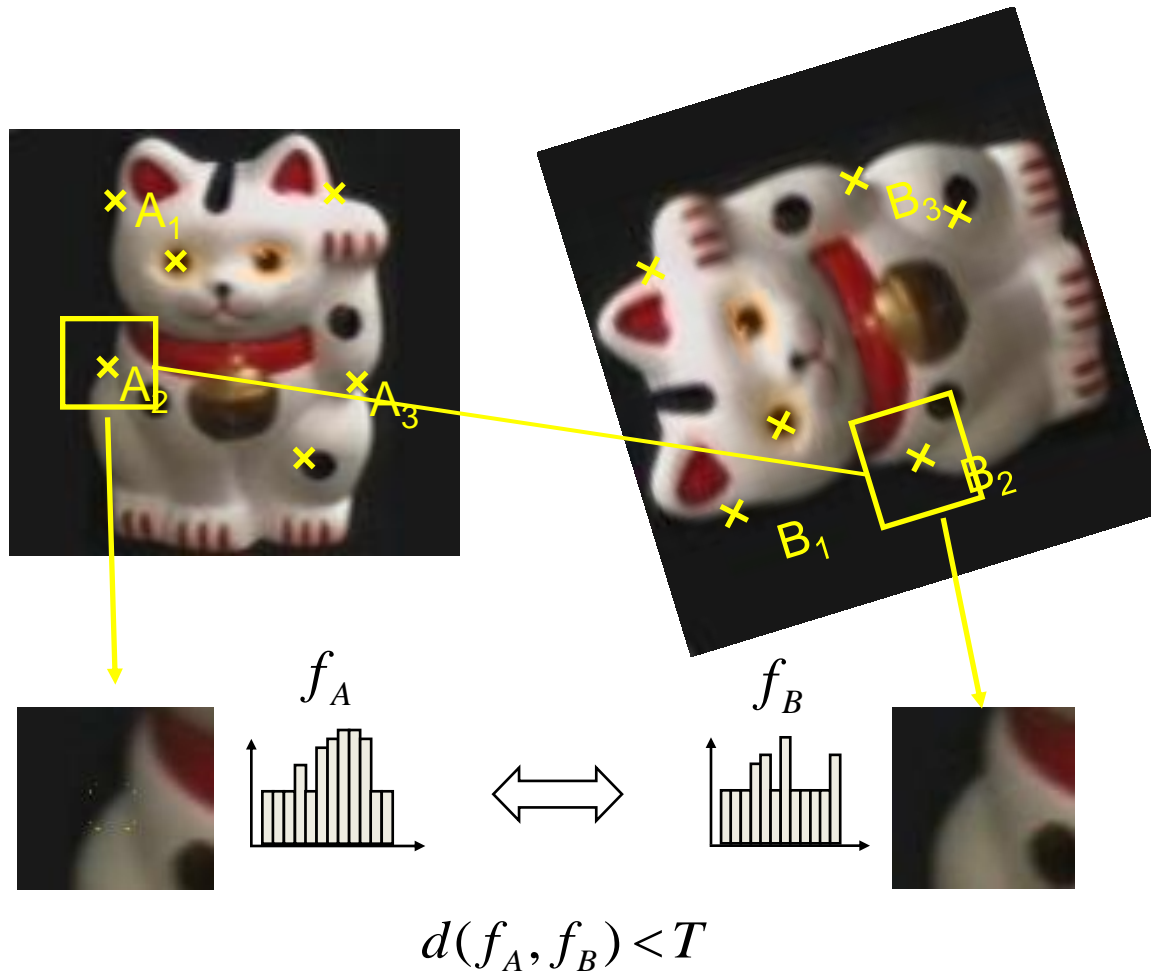


How can we align two pictures?

- Global matching?
 - But what if
 - Not just translation change, but rotation and scale?
 - Only small pieces of the pictures match?



Today: 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

Question

- Why not just take every patch in the original image and find best match in second image?

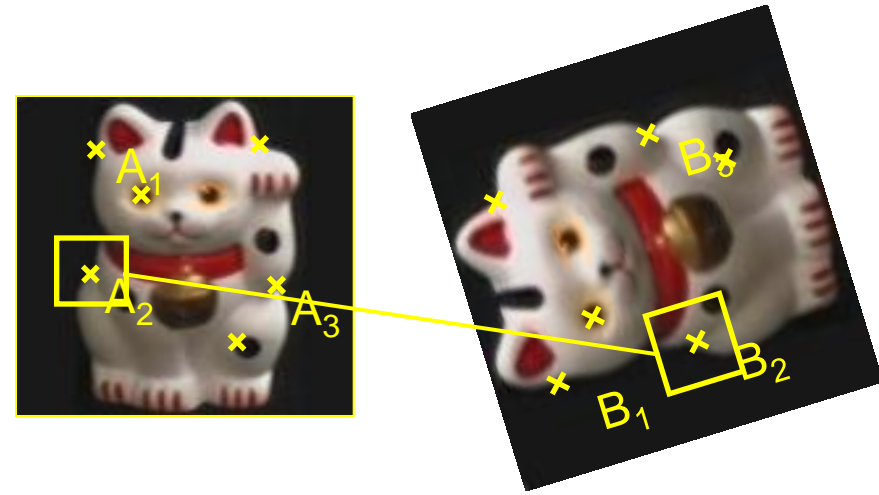


Goals for Keypoints



Detect points that are *repeatable* and *distinctive*

Key trade-offs



Localization



More Points

Robust to occlusion
Works with less texture

More Repeatable

Robust detection
Precise localization

Description



More Robust

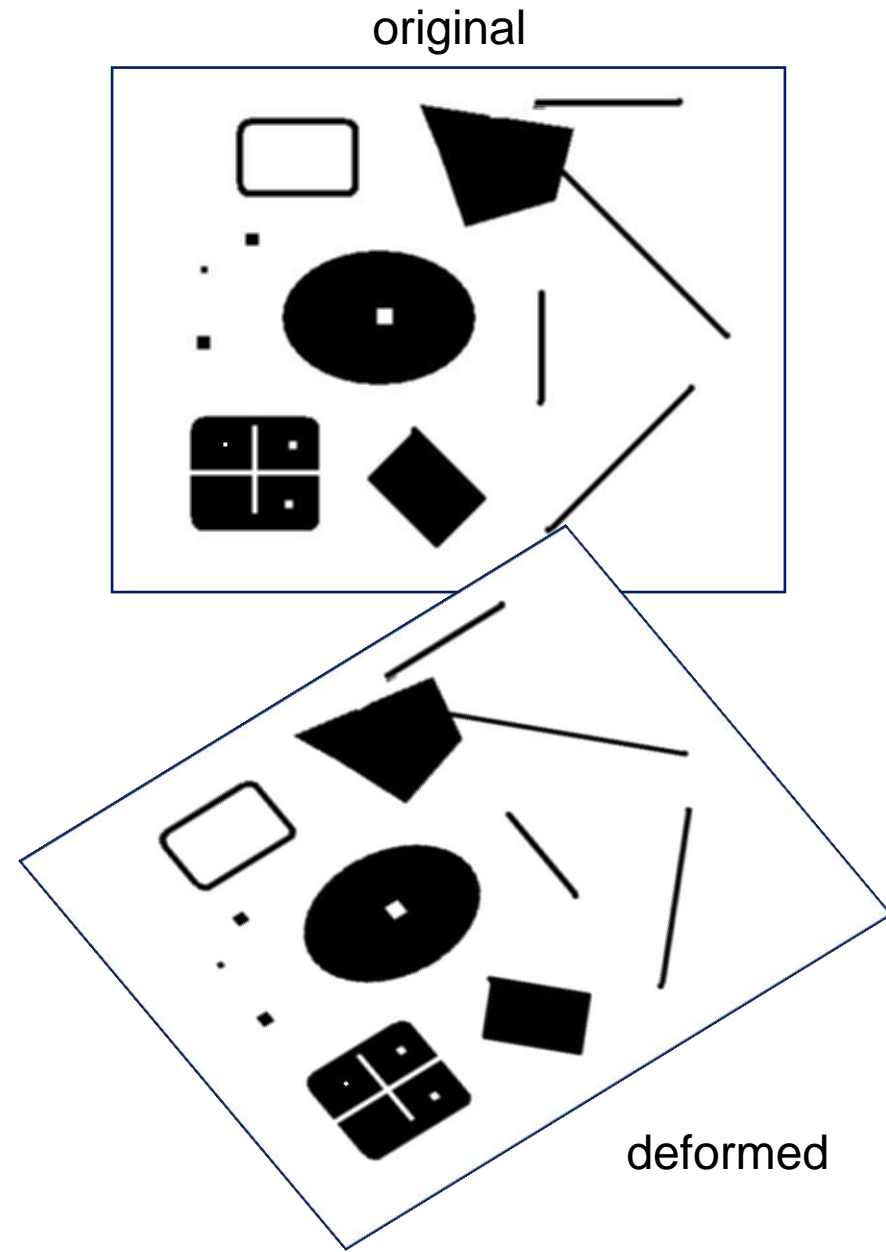
Deal with expected variations
Maximize correct matches

More Selective

Minimize wrong matches

Keypoint localization

- Suppose you have to click on some point, go away and come back after I deform the image, and click on the same points again.
 - Which points would you choose?



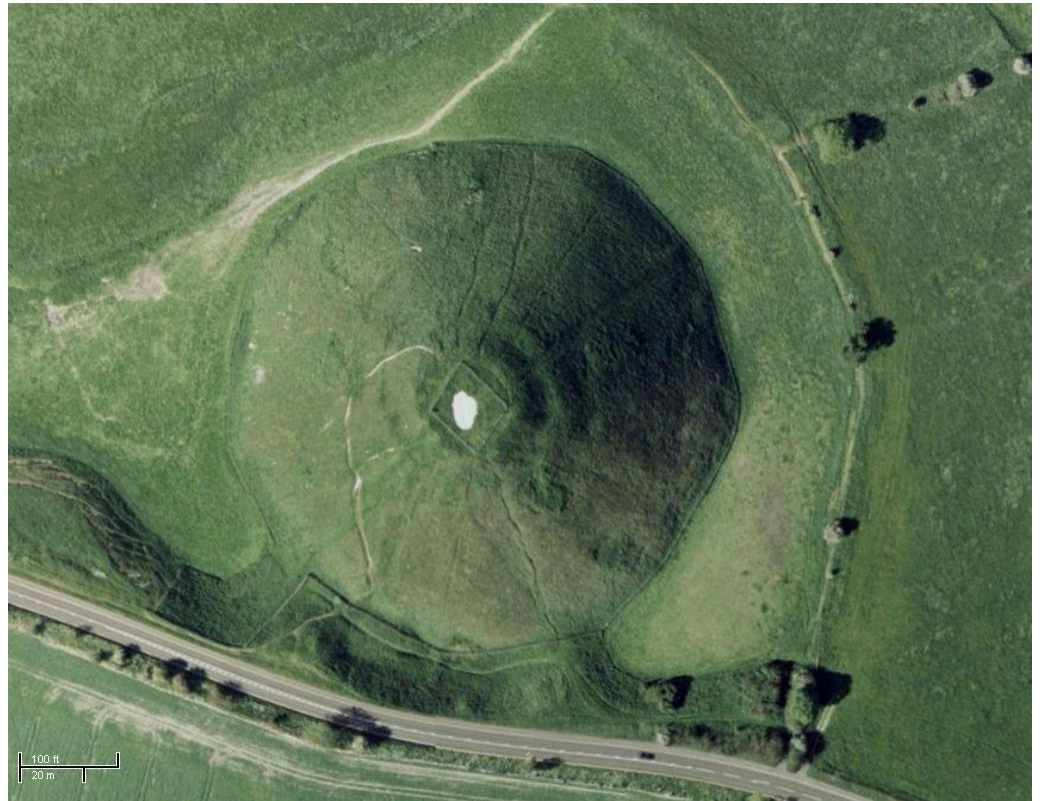
Choosing interest points

Where would you
tell your friend to
meet you?



Choosing interest points

Where would you
tell your friend to
meet you?



Choosing interest points

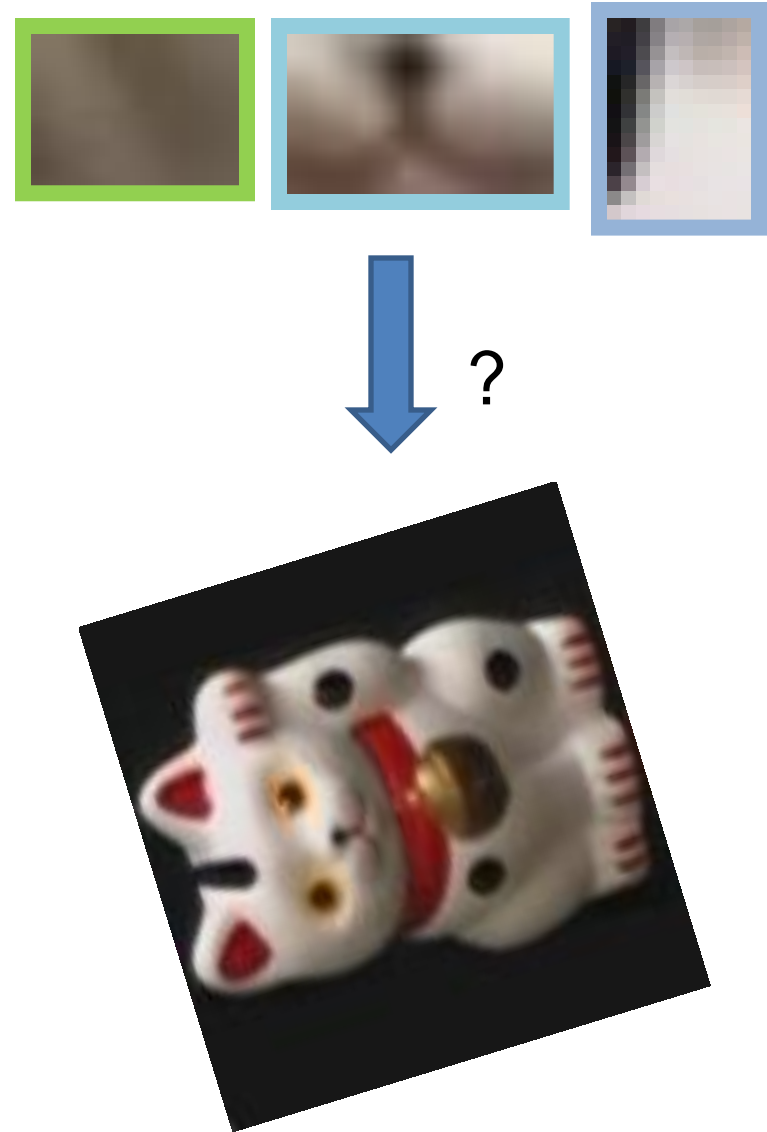
- Corners



- Peaks/Valleys



Which patches are easier to match?



Many Existing Detectors Available

Hessian & Harris

Laplacian, DoG

Harris-/Hessian-Laplace

Harris-/Hessian-Affine

EBR and IBR

MSER

Salient Regions

Others...

[Beaudet '78], [Harris '88]

[Lindeberg '98], [Lowe 1999]

[Mikolajczyk & Schmid '01]

[Mikolajczyk & Schmid '04]

[Tuytelaars & Van Gool '04]

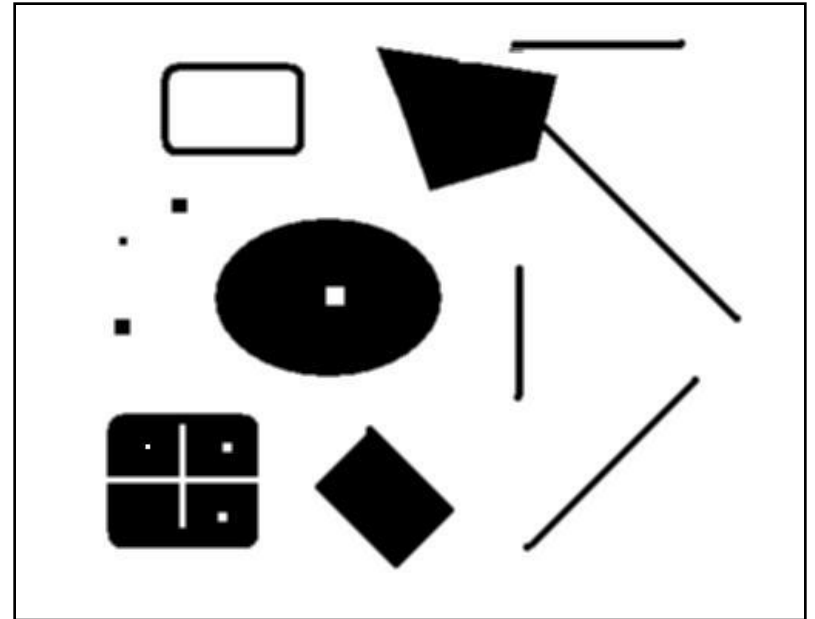
[Matas '02]

[Kadir & Brady '01]

Harris Detector [Harris88]

Second moment matrix

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



Intuition: Search for local neighborhoods where the image content has two main directions (eigenvectors).

Harris Detector [Harris88]

- Second moment matrix

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

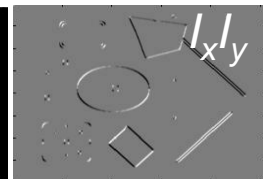
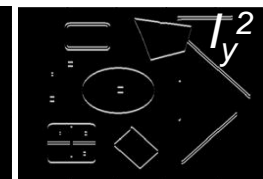
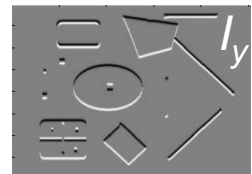
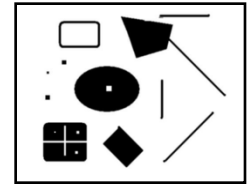
$$\det M = \lambda_1 \lambda_2$$

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

2. Square of derivatives

3. Gaussian filter $g(\sigma_I)$

1. Image derivatives



4. Cornerness function – both eigenvalues are strong

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

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

5. Non-maxima suppression

Matlab code for Harris Detector

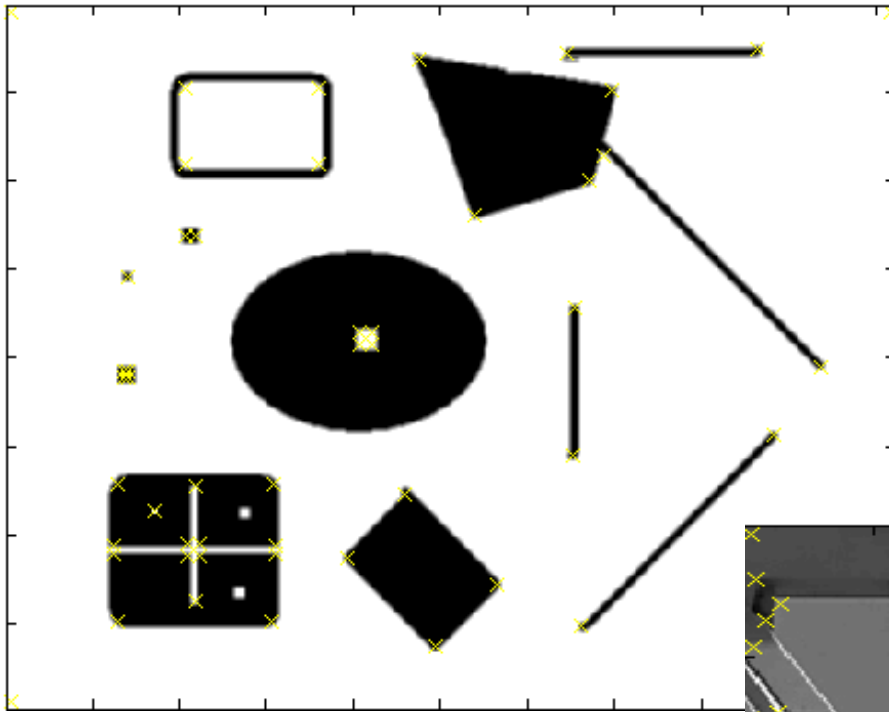
```
function [ptx, pty] = detectKeypoints(im, alpha, N)

% get harris function
gfil = fspecial('gaussian', [7 7], 1); % smoothing filter
imblur = imfilter(im, gfil); % smooth image
[Ix, Iy] = gradient(imblur); % compute gradient
Ixx = imfilter(Ix.*Ix, gfil); % compute smoothed x-gradient sq
Iyy = imfilter(Iy.*Iy, gfil); % compute smoothed y-gradient sq
Ixy = imfilter(Ix.*Iy, gfil);
har = Ixx.*Iyy - Ixy.*Ixy - alpha*(Ixx+Iyy).^2; % cornerness

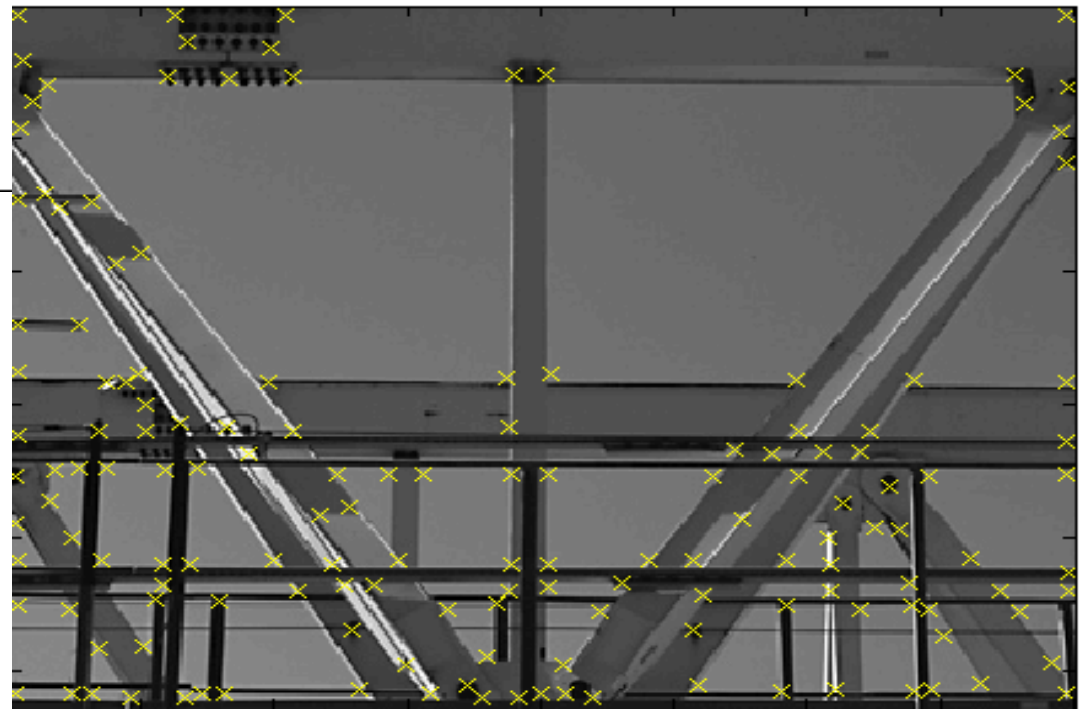
% get local maxima within 7x7 window
maxv = ordfilt2(har, 49, ones(7)); % sorts values in each window
maxv2 = ordfilt2(har, 48, ones(7));
ind = find(maxv==har & maxv~=maxv2);

% get top N points
[sv, sind] = sort(har(ind), 'descend');
sind = ind(sind);
[pty, ptx] = ind2sub(size(im), sind(1:min(N, numel(sind))));
```

Harris Detector – Responses [Harris88]



Effect: A very precise corner detector.



Harris Detector – Responses [Harris88]



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



Automatic Scale Selection

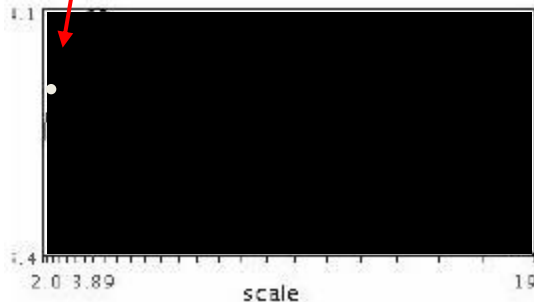


$$f(I_{i_1 \dots i_m}(x, \sigma)) = f(I_{i_1 \dots i_m}(x', \sigma'))$$

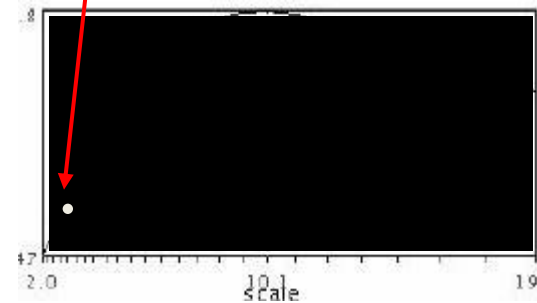
How to find corresponding patch sizes?

Automatic Scale Selection

- Function responses for increasing scale (scale signature)



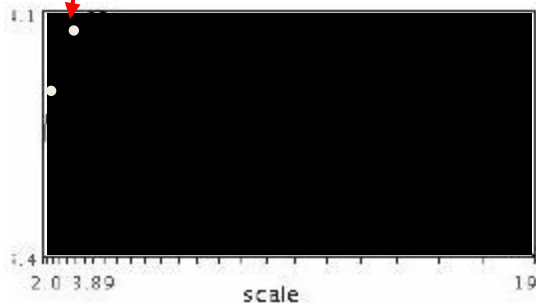
$$f(I_{i_1...i_m}(x, \sigma))$$



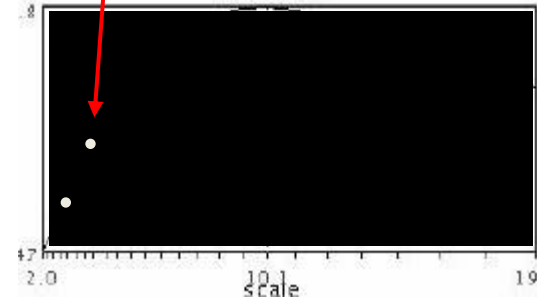
$$f(I_{i_1...i_m}(x', \sigma))$$

Automatic Scale Selection

- Function responses for increasing scale (scale signature)



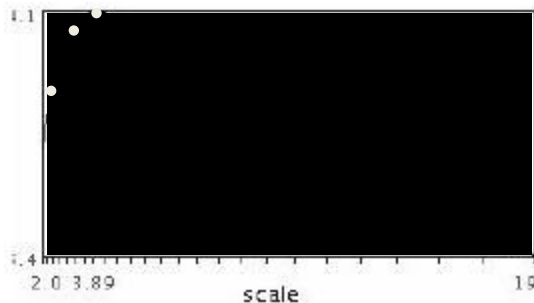
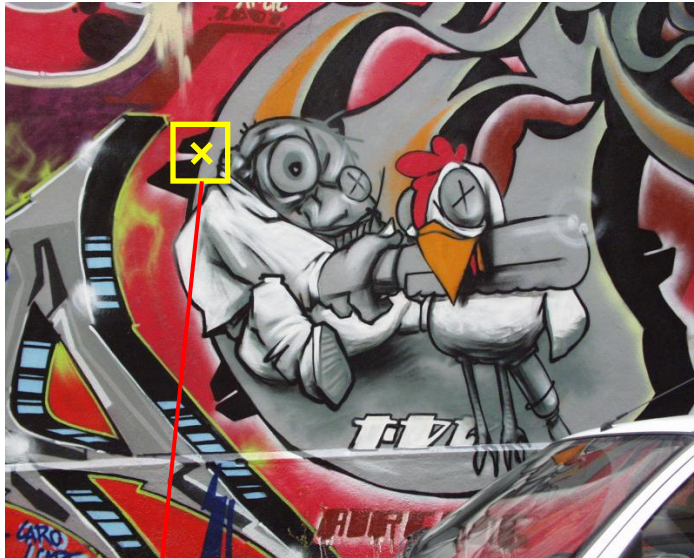
$$f(I_{i_1...i_m}(x, \sigma))$$



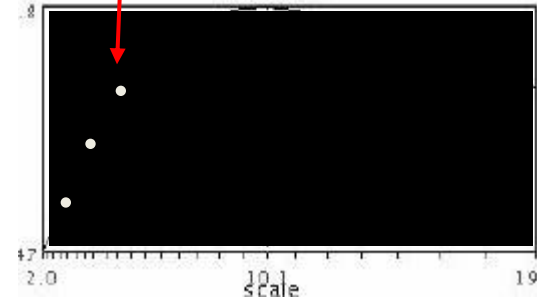
$$f(I_{i_1...i_m}(x', \sigma))$$

Automatic Scale Selection

- Function responses for increasing scale (scale signature)



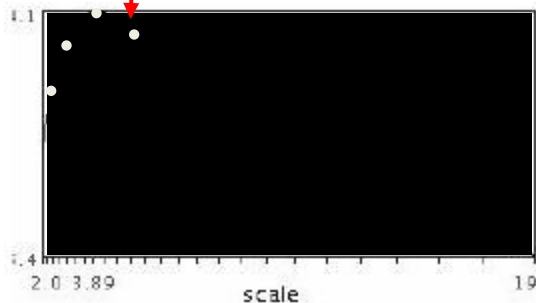
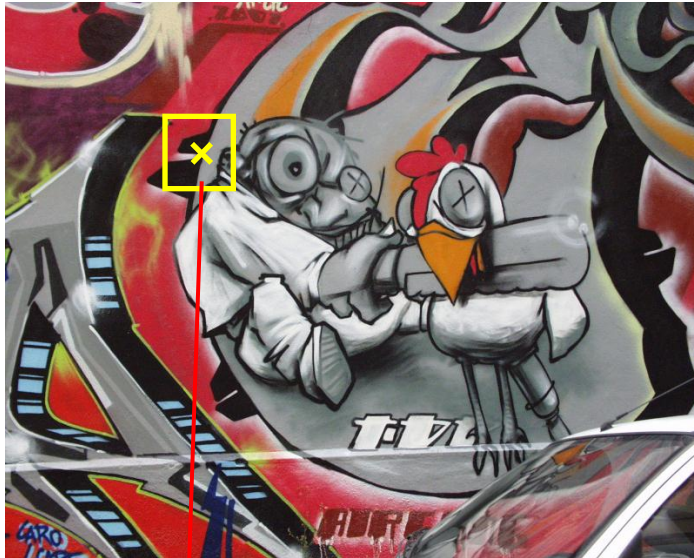
$$f(I_{i_1 \dots i_m}(x, \sigma))$$



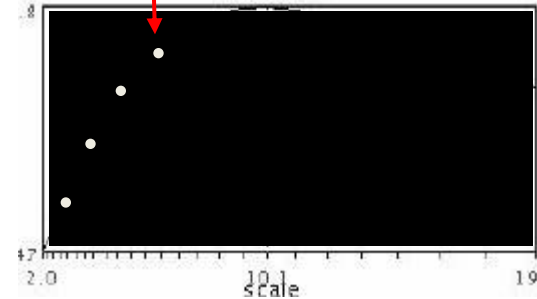
$$f(I_{i_1 \dots i_m}(x', \sigma))$$

Automatic Scale Selection

- Function responses for increasing scale (scale signature)



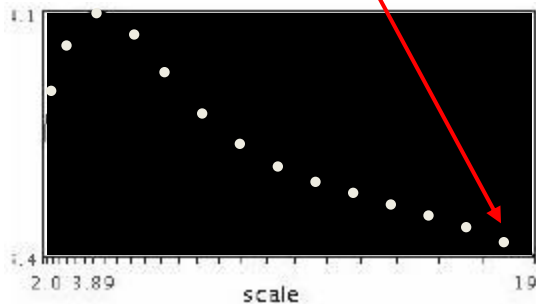
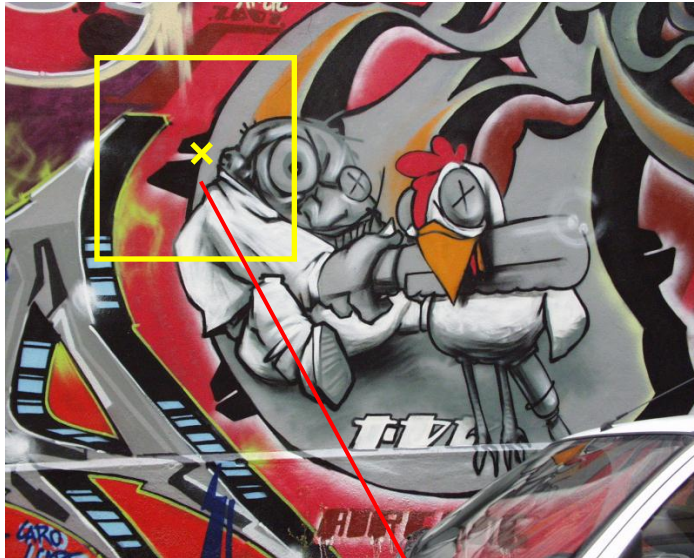
$$f(I_{i_1 \dots i_m}(x, \sigma))$$



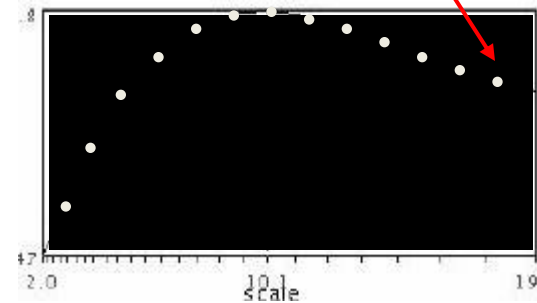
$$f(I_{i_1 \dots i_m}(x', \sigma))$$

Automatic Scale Selection

- Function responses for increasing scale (scale signature)



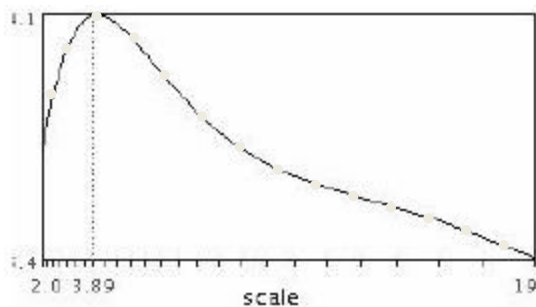
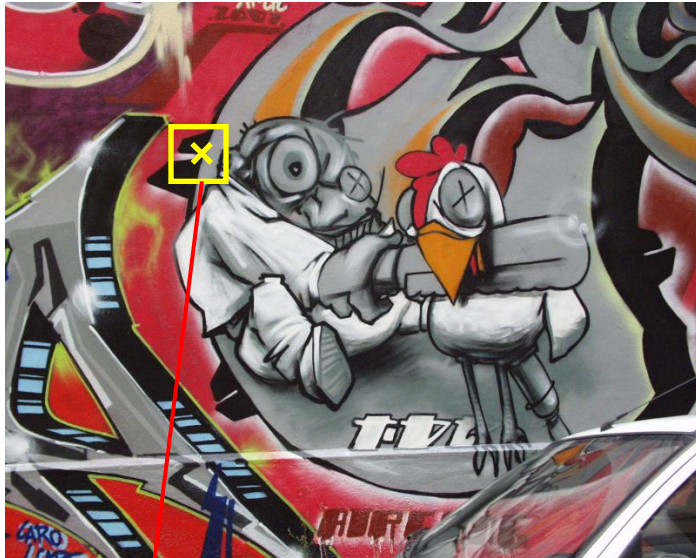
$$f(I_{i_1...i_m}(x, \sigma))$$



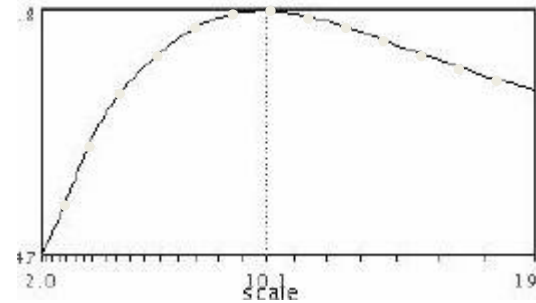
$$f(I_{i_1...i_m}(x', \sigma))$$

Automatic Scale Selection

- Function responses for increasing scale (scale signature)



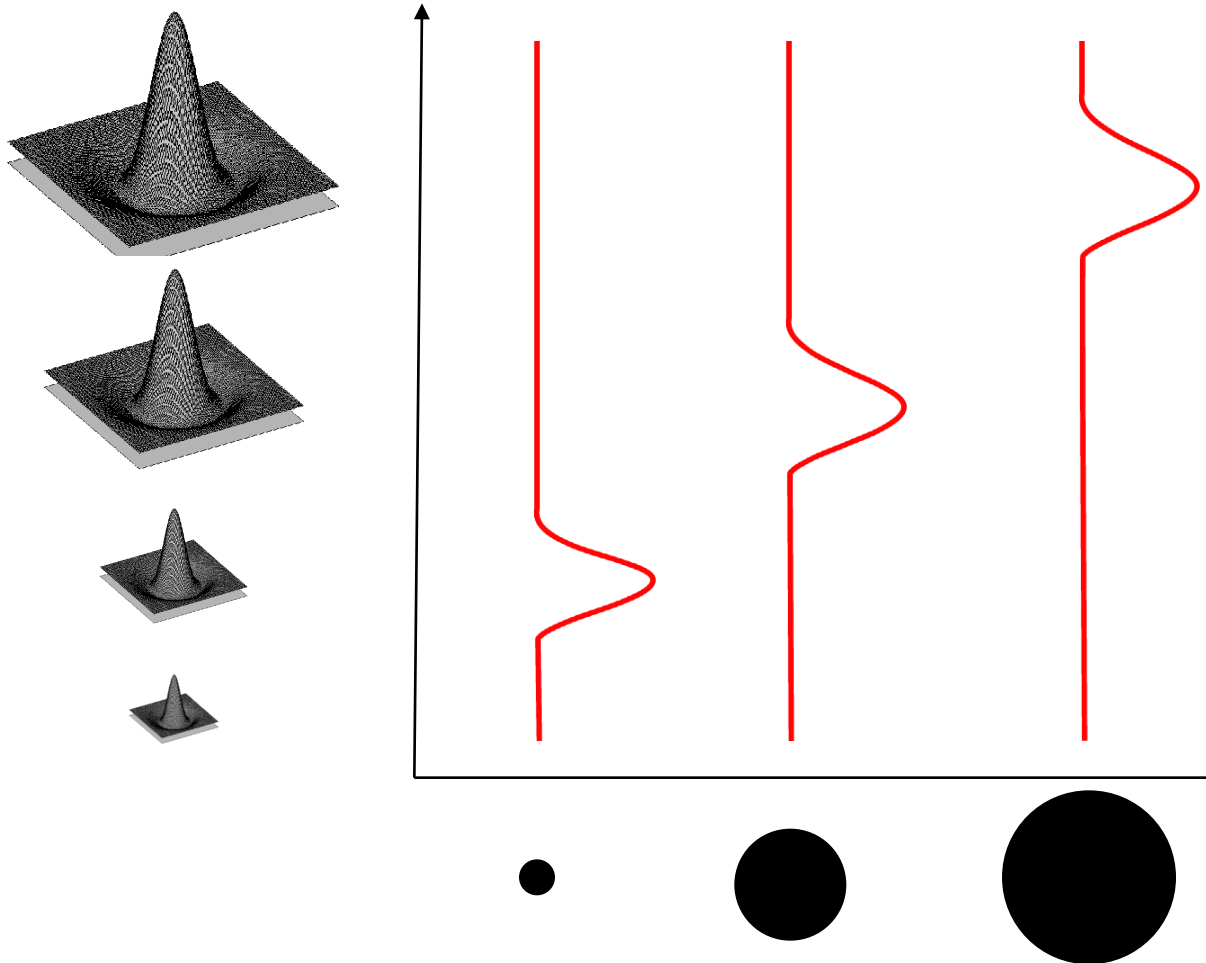
$$f(I_{i_1...i_m}(x, \sigma))$$



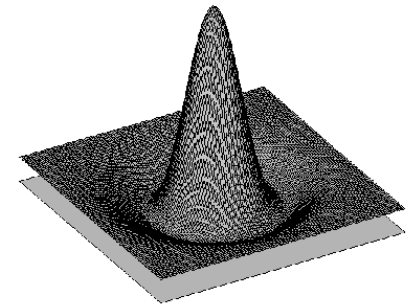
$$f(I_{i_1...i_m}(x', \sigma'))$$

What Is A Useful Signature Function?

- Difference of Gaussian = “blob” detector



Difference-of-Gaussian (DoG)



-

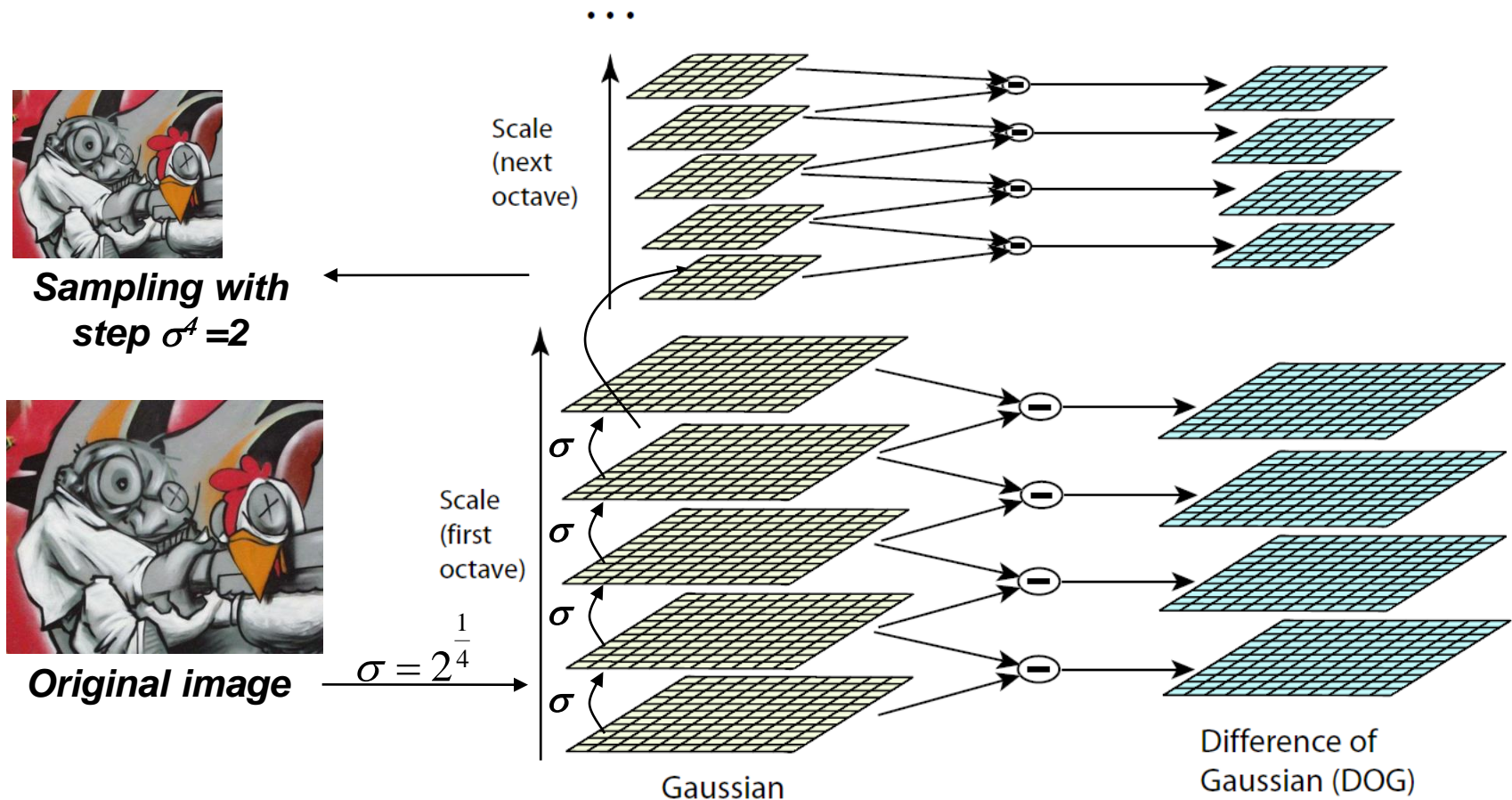


=



DoG – Efficient Computation

- Computation in Gaussian scale pyramid



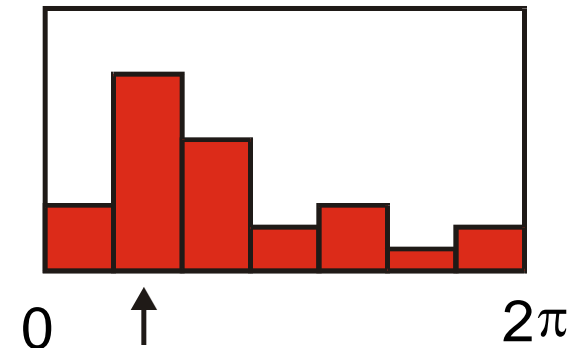
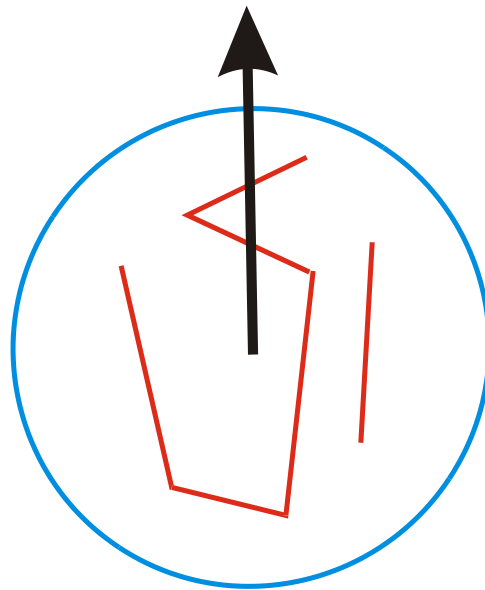
Results: Lowe's DoG



Orientation Normalization

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

[Lowe, SIFT, 1999]



Available at a web site near you...

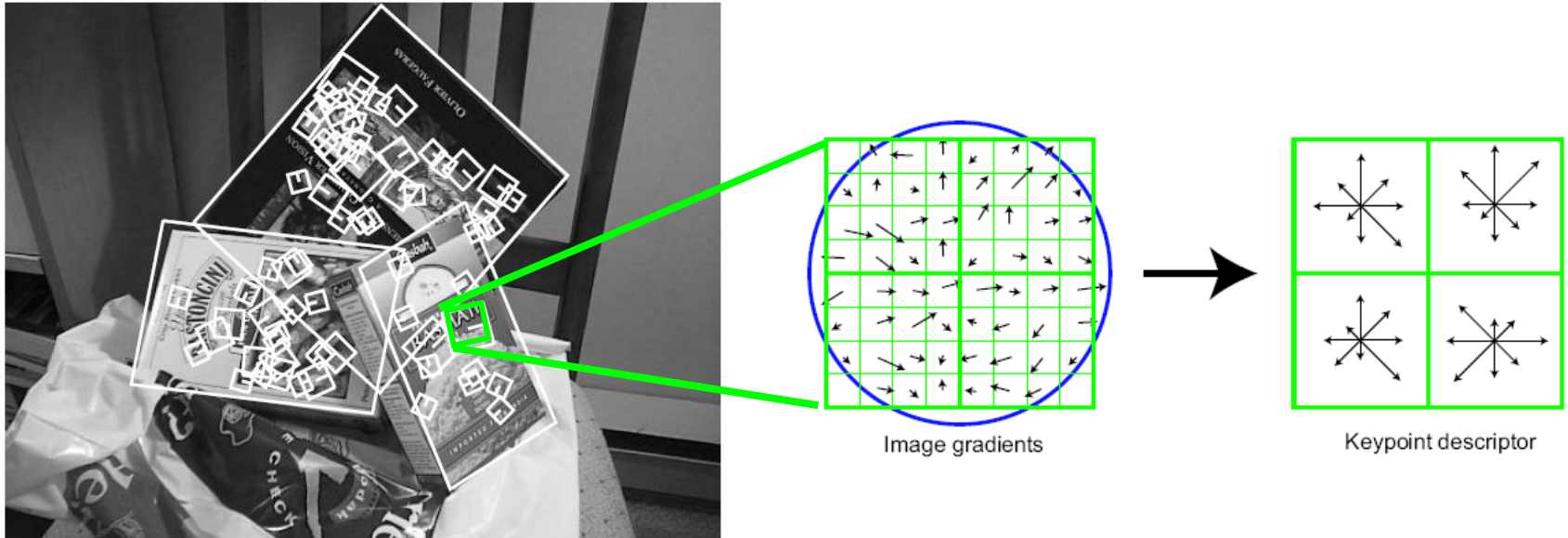
- For most local feature detectors, executables are available online:
 - <http://robots.ox.ac.uk/~vgg/research/affine>
 - <http://www.cs.ubc.ca/~lowe/keypoints/>
 - <http://www.vision.ee.ethz.ch/~surf>

How do we describe the keypoint?

Local Descriptors

- The ideal descriptor should be
 - Robust
 - Distinctive
 - Compact
 - Efficient
- Most available descriptors focus on edge/gradient information
 - Capture texture information
 - Color rarely used

Local Descriptors: SIFT Descriptor



Histogram of oriented gradients

- Captures important texture information
- Robust to small translations / affine deformations

[Lowe, ICCV 1999]

Details of Lowe's SIFT algorithm

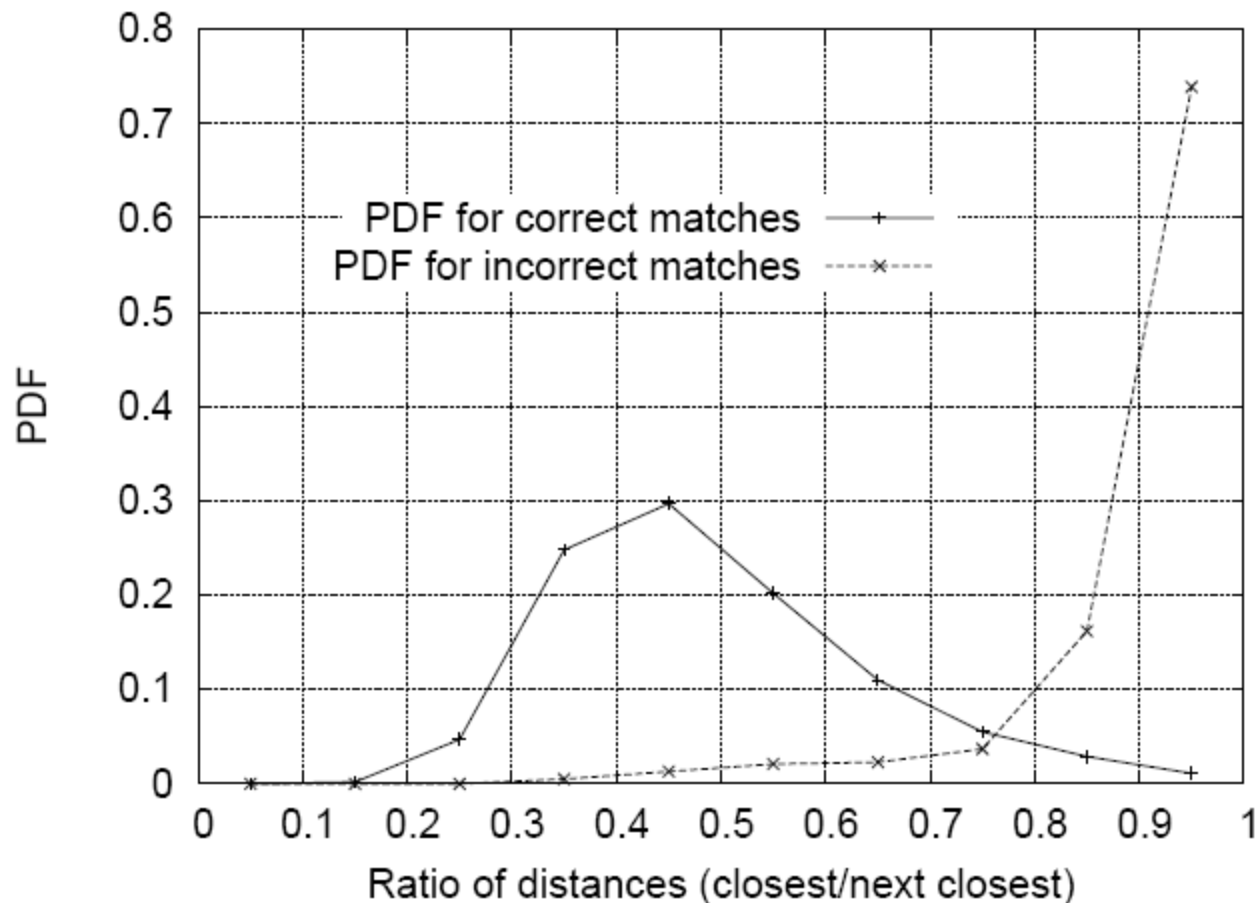
- Run DoG detector
 - Find maxima in location/scale space
 - Remove edge points
- Find all major orientations
 - Bin orientations into 36 bin histogram
 - Weight by gradient magnitude
 - Weight by distance to center (Gaussian-weighted mean)
 - Return orientations within 0.8 of peak
 - Use parabola for better orientation fit
- For each (x,y,scale,orientation), create descriptor:
 - Sample 16x16 gradient mag. and rel. orientation
 - Bin 4x4 samples into 4x4 histograms
 - Threshold values to max of 0.2, divide by L2 norm
 - Final descriptor: 4x4x8 normalized histograms

$$\mathbf{H} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix}$$

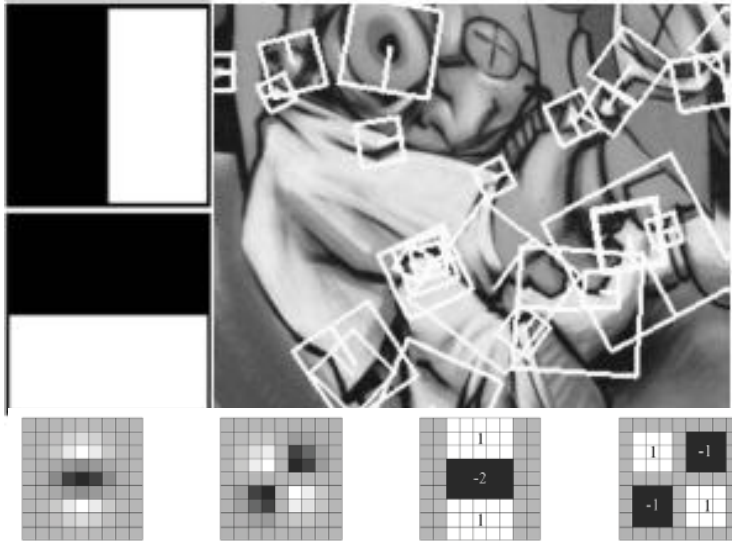
$$\frac{\text{Tr}(\mathbf{H})^2}{\text{Det}(\mathbf{H})} < \frac{(r+1)^2}{r}$$

Matching SIFT Descriptors

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



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>

What to use when?

Detectors

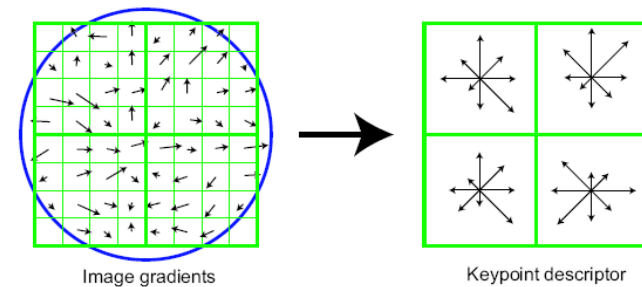
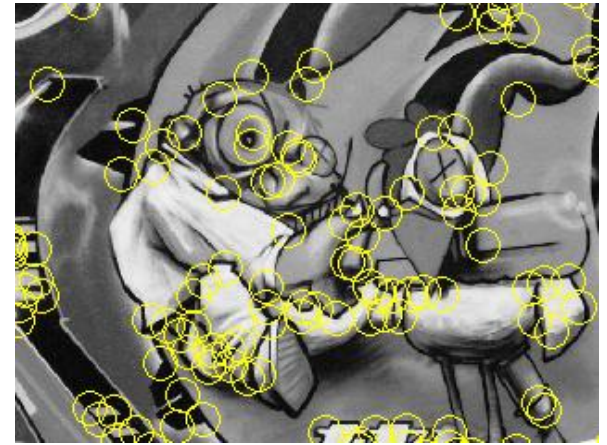
- Harris gives very precise localization but doesn't predict scale
 - Good for some tracking applications
- DOG (difference of Gaussian) provides ok localization and scale
 - Good for multi-scale or long-range matching

Descriptors

- SIFT: good general purpose descriptor

Things to remember

- Keypoint detection: repeatable and distinctive
 - Corners, blobs
 - Harris, DoG
- Descriptors: robust and selective
 - SIFT: spatial histograms of gradient orientation



Next time: Panoramic Stitching

