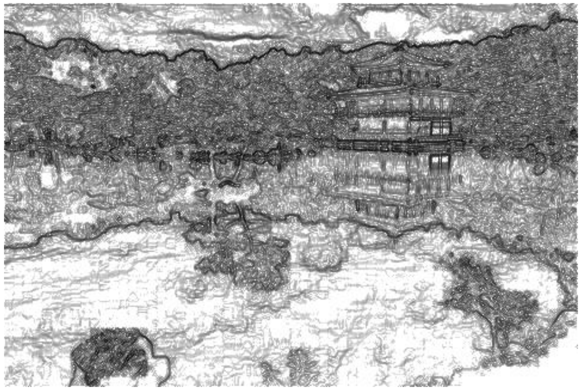
Image gradients and edges

Thursday, Sept 10, 2020

Richang Hong, Hefei University of Technology



Last time

- Various models for image “noise”
- Linear filters and convolution useful for
  - Image smoothing, removing noise
    - Box filter
    - Gaussian filter
    - Impact of scale / width of smoothing filter
- Separable filters more efficient
- Median filter: a non-linear filter, edge-preserving

Image filtering

- Compute a function of the local neighborhood at each pixel in the image
  - Function specified by a “filter” or mask saying how to combine values from neighbors.

- Uses of filtering:
  - Enhance an image (denoise, resize, etc)
  - Extract information (texture, edges, etc)
  - Detect patterns (template matching)

Today

Adapted from Derek Hoiem

Edge detection

- **Goal**: map image from 2d array of pixels to a set of curves or line segments or contours.
- **Why?**

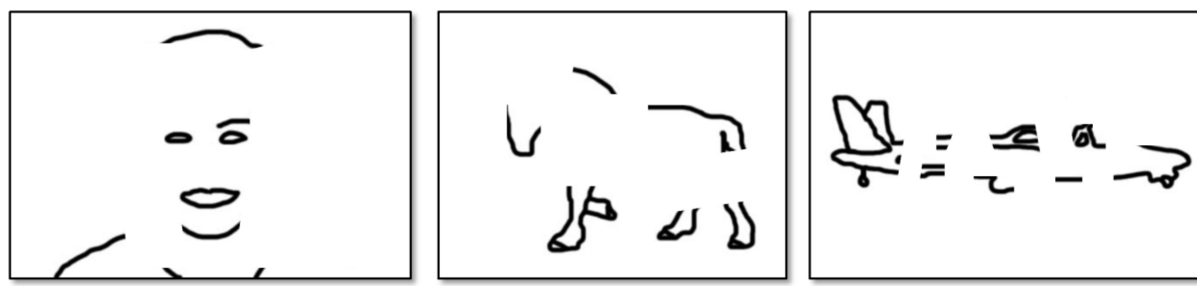


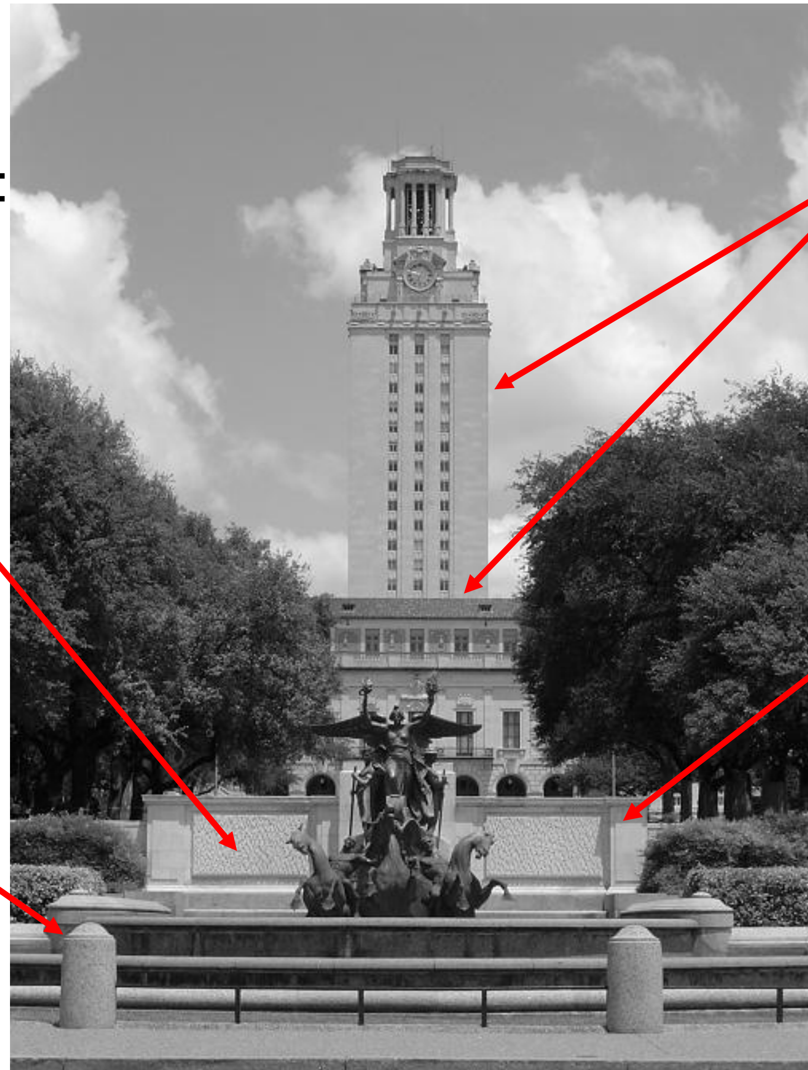
Figure from J. Shotton et al., PAMI 2007

- **Main idea**: look for strong gradients, post-process

What causes an edge?

Reflectance change: appearance information, texture

Change in surface orientation: shape

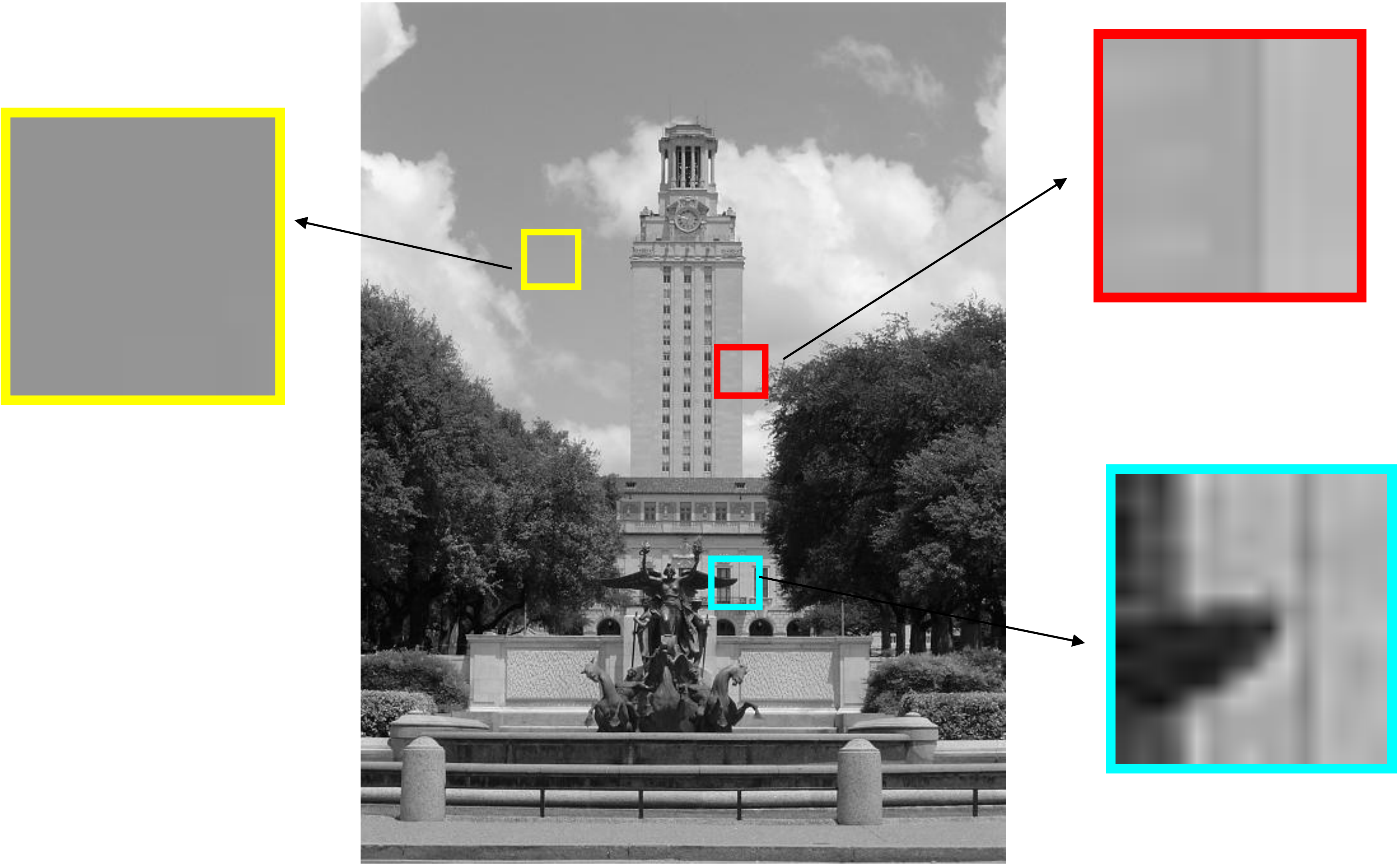


Depth discontinuity: object boundary

Cast shadows

Slide credit: Kristen Grauman

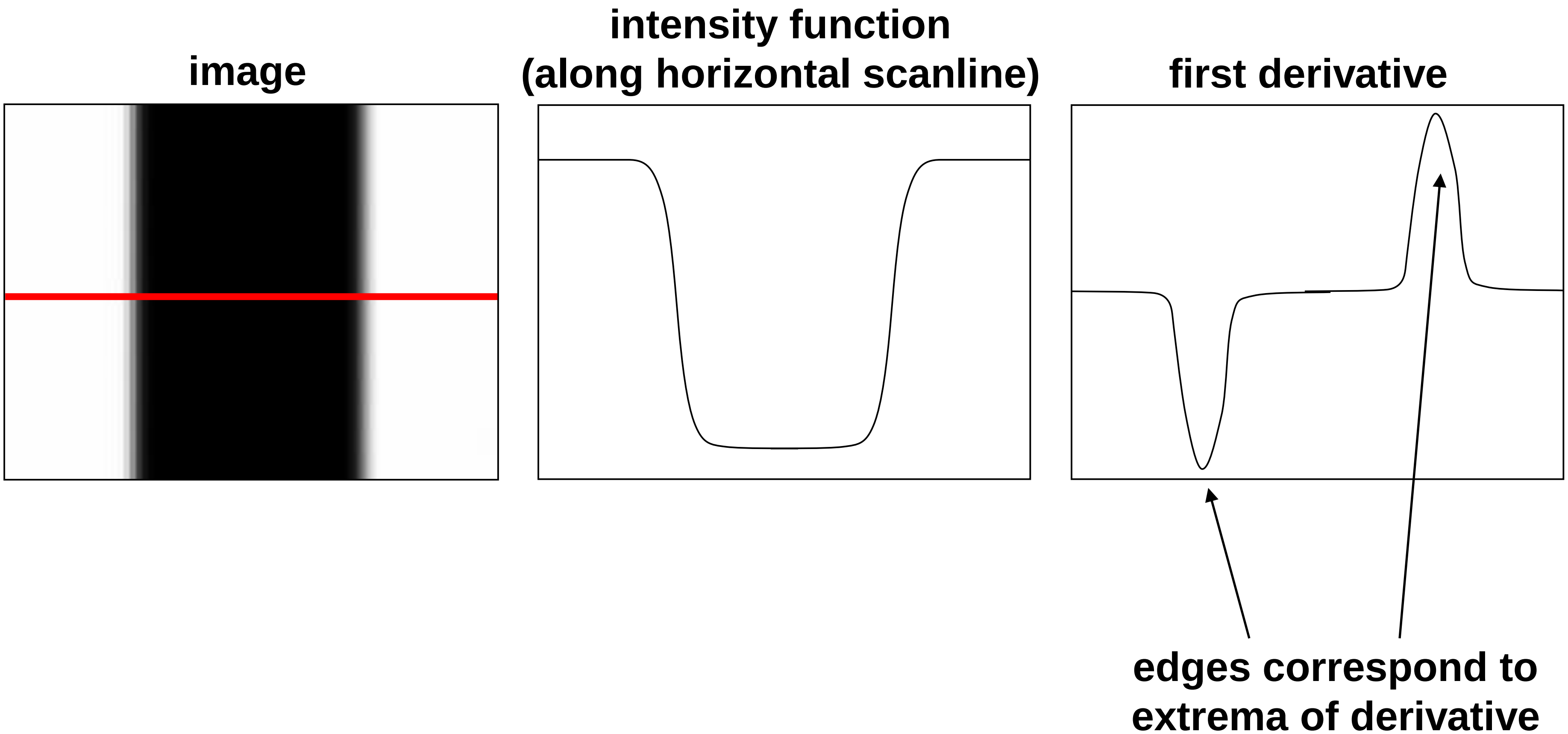
Edges/gradients and invariance



Slide credit:
Kristen Grauman

Derivatives and edges

An edge is a place of rapid change in the image intensity function.



Source: L. Lazebnik

Derivatives with convolution

For 2D function, f(x,y), the partial derivative is:

$$\frac{\partial f(x,y)}{\partial x} = \lim_{\varepsilon \to 0} \frac{f(x+\varepsilon, y) - f(x,y)}{\varepsilon}$$

For discrete data, we can approximate using finite differences:

$$\frac{\partial f(x,y)}{\partial x} \approx \frac{f(x+1, y) - f(x,y)}{1}$$

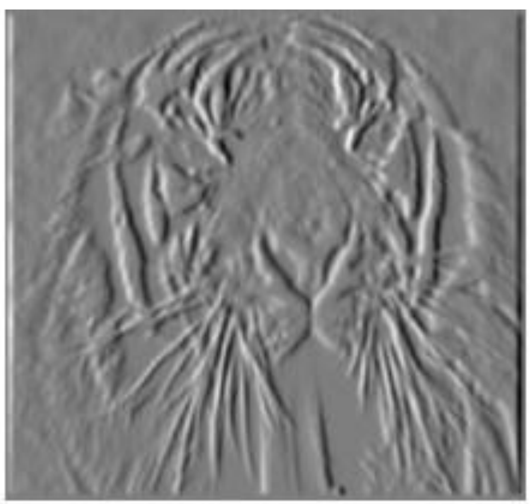
To implement above as convolution, what would be the associated filter?

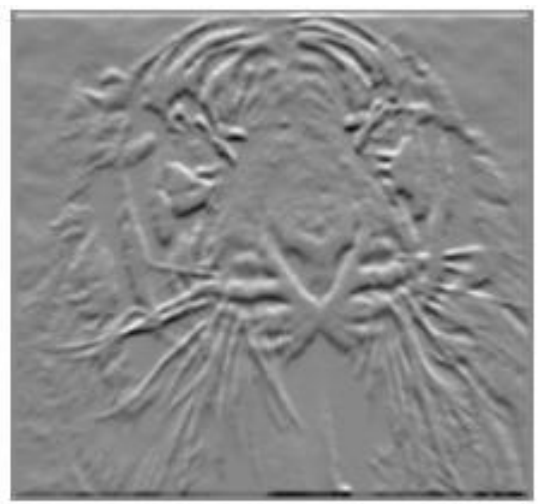
Slide credit:
Kristen Grauman

Partial derivatives of an image

$$\frac{\partial f(x,y)}{\partial x}$$

-1	1





$$\frac{\partial f(x,y)}{\partial y}$$

-1
1

?

or

1
-1

Which shows changes with respect to x?

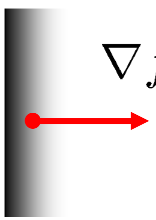
(showing filters for correlation)

Image gradient

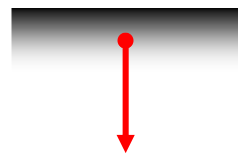
The gradient of an image:

$$\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right]$$

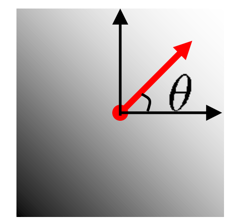
The gradient points in the direction of most rapid change in intensity



$$\nabla f = \left[\frac{\partial f}{\partial x}, 0\right]$$



$$\nabla f = \left[0, \frac{\partial f}{\partial y}\right]$$



$$\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right]$$

The **gradient direction** (orientation of edge normal) is given by:

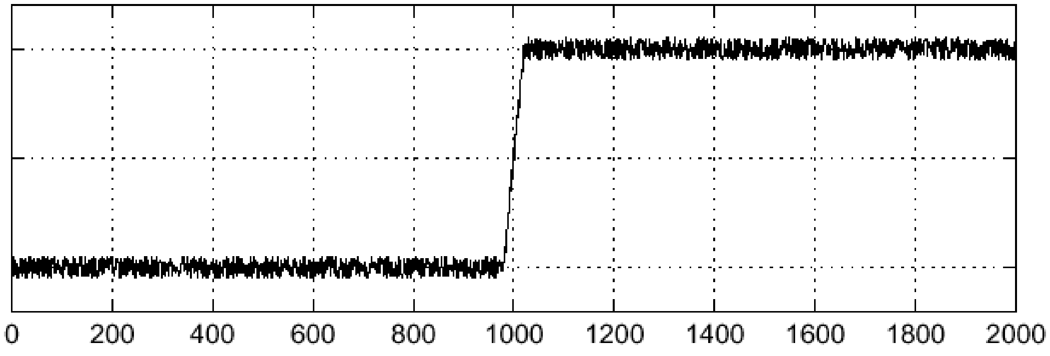
$$\theta = \tan^{-1}\left(\frac{\partial f}{\partial y} \Big/ \frac{\partial f}{\partial x}\right)$$

Effects of noise

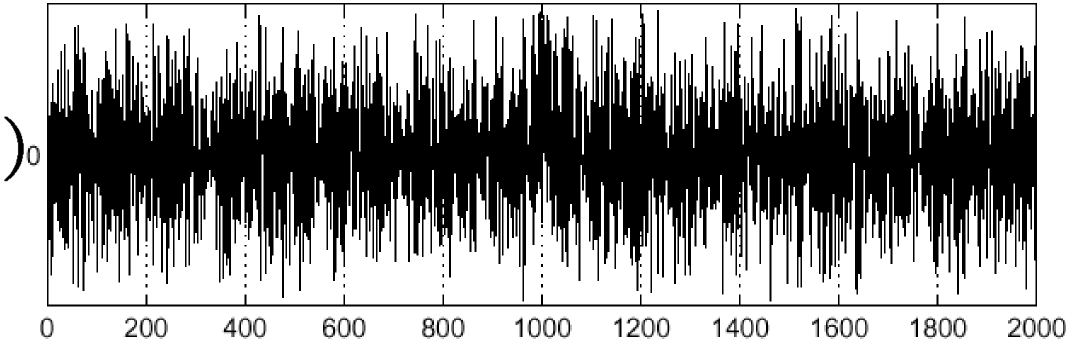
Consider a single row or column of the image

- Plotting intensity as a function of position gives a signal

$f(x)$



$\frac{d}{dx}f(x)$



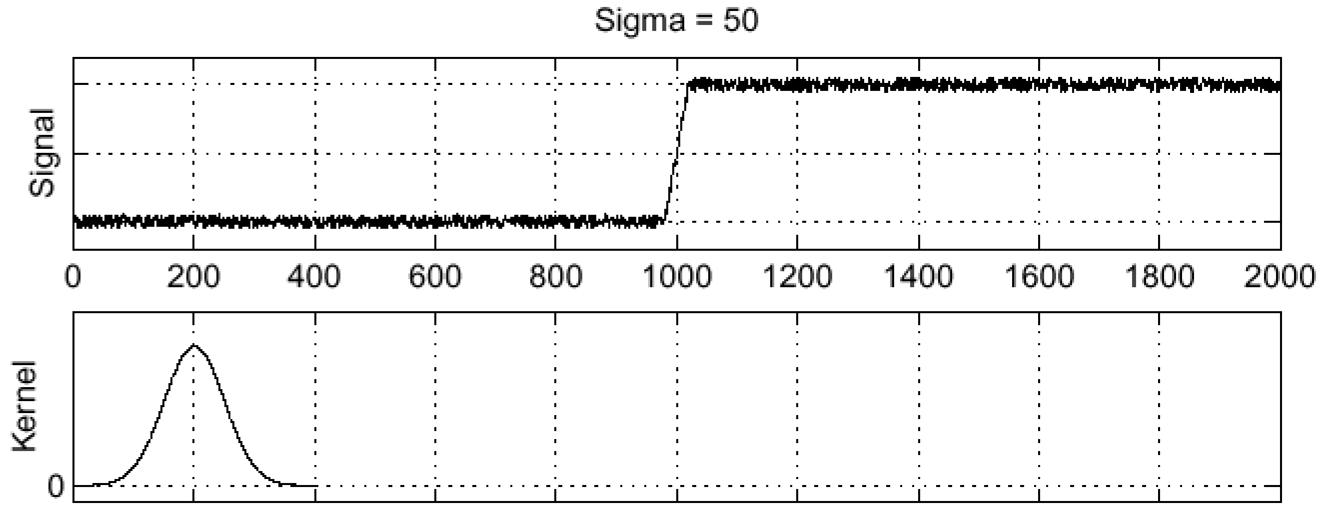
Where is the edge?

Slide credit Steve Seitz

Solution: smooth first

f

h



Where is the edge?

Look for peaks in $\frac{\partial}{\partial x}(h \star f)$

Derivative theorem of convolution

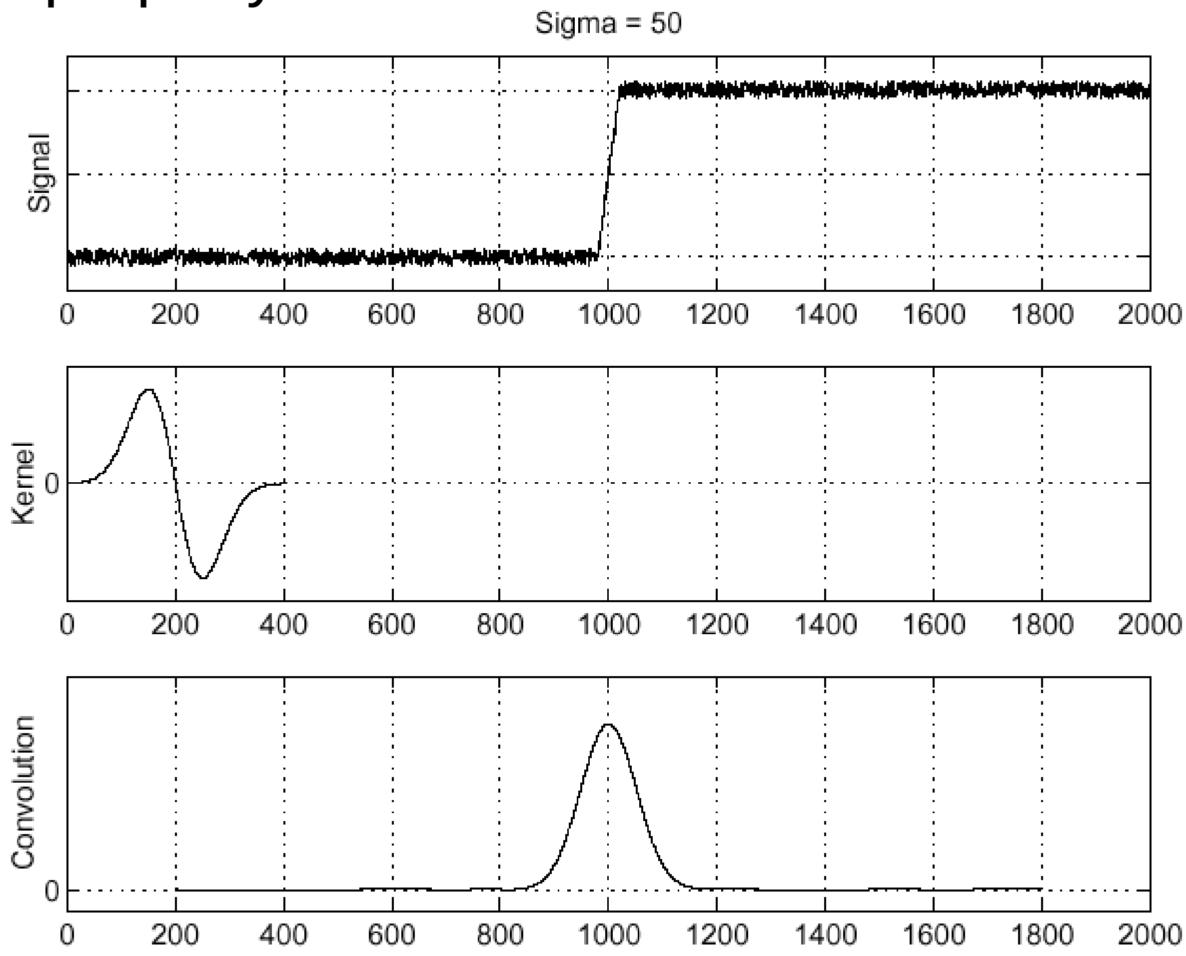
$$\frac{\partial}{\partial x}(h \star f) = (\frac{\partial}{\partial x}h) \star f$$

Differentiation property of convolution.

f

$\frac{\partial}{\partial x}h$

$(\frac{\partial}{\partial x}h) \star f$

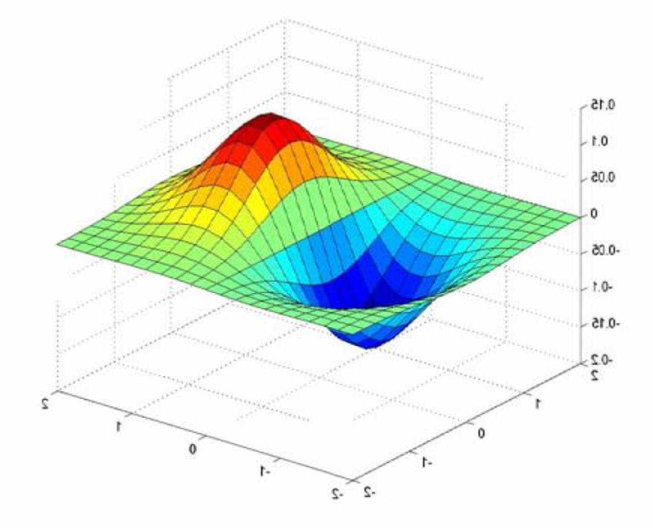


Slide credit Steve Seitz

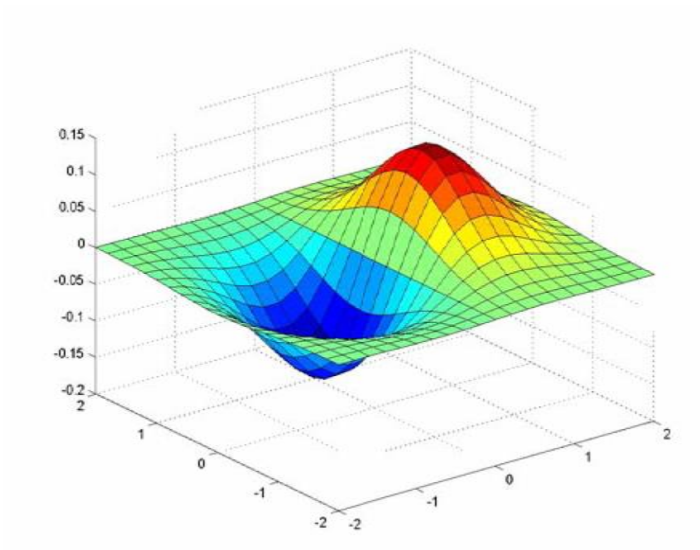
Derivative of Gaussian filters

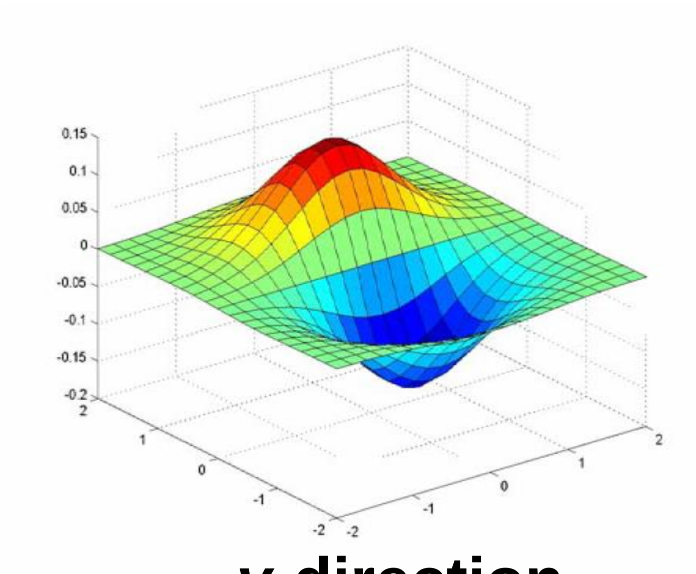
$$(I \otimes g) \otimes h = I \otimes (g \otimes h)$$

$$\begin{bmatrix} 0.0030 & 0.0133 & 0.0219 & 0.0133 & 0.0030 \\ 0.0133 & 0.0596 & 0.0983 & 0.0596 & 0.0133 \\ 0.0219 & 0.0983 & 0.1621 & 0.0983 & 0.0219 \\ 0.0133 & 0.0596 & 0.0983 & 0.0596 & 0.0133 \\ 0.0030 & 0.0133 & 0.0219 & 0.0133 & 0.0030 \end{bmatrix} \otimes \begin{bmatrix} 1 & -1 \end{bmatrix}$$



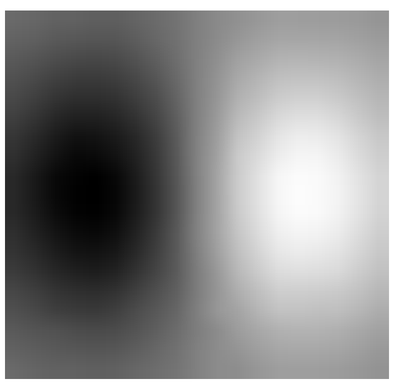
Derivative of Gaussian filters

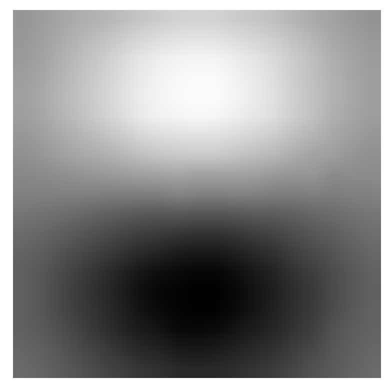




***x*-direction**

***y*-direction**





Source: L. Lazebnik

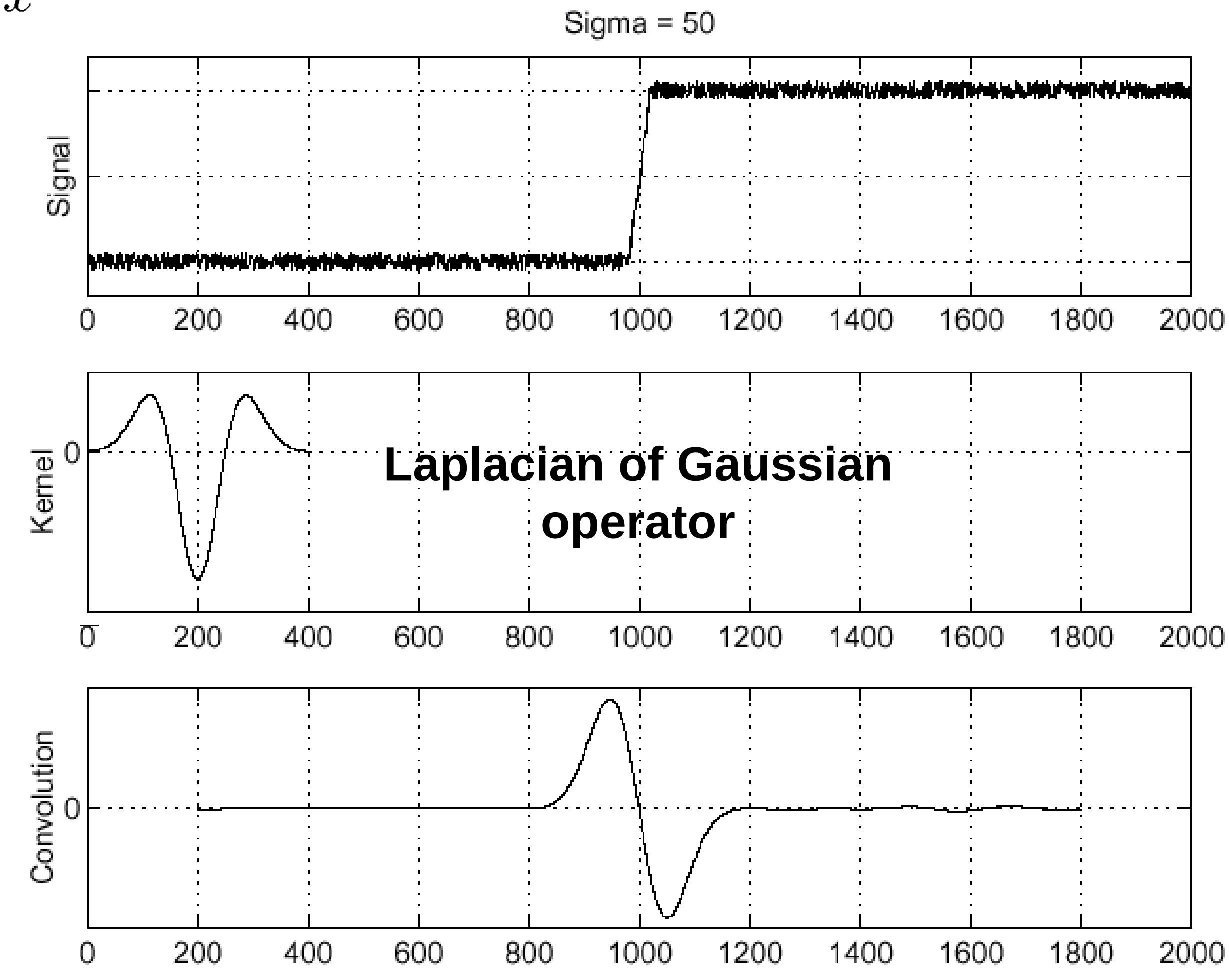
Laplacian of Gaussian

Consider $\frac{\partial^2}{\partial x^2}(h \star f)$

f

$\frac{\partial^2}{\partial x^2}h$

$(\frac{\partial^2}{\partial x^2}h) \star f$

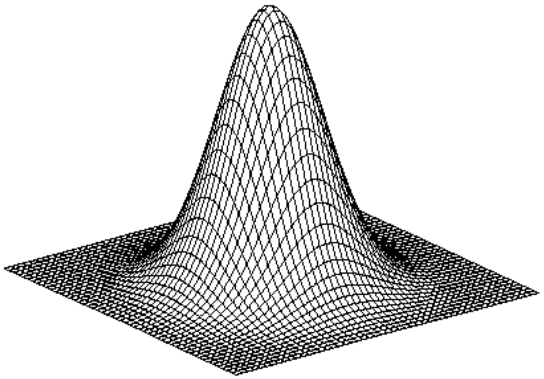


Where is the edge?

Zero-crossings of bottom graph

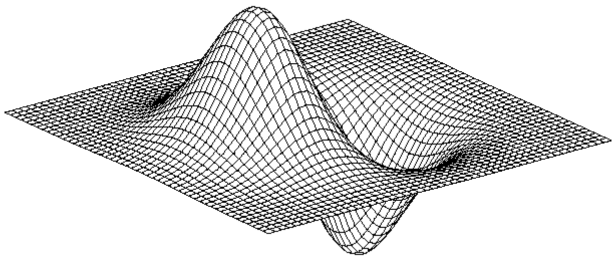
Slide credit: Steve Seitz

2D edge detection filters



Gaussian

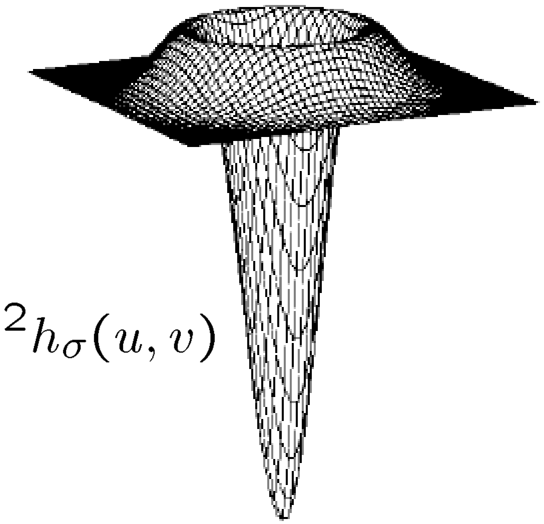
$$h_\sigma(u,v) = \frac{1}{2\pi\sigma^2} e^{-\frac{u^2+v^2}{2\sigma^2}}$$



derivative of Gaussian

$$\frac{\partial}{\partial x} h_\sigma(u,v)$$

Laplacian of Gaussian



$$\nabla^2 h_\sigma(u,v)$$

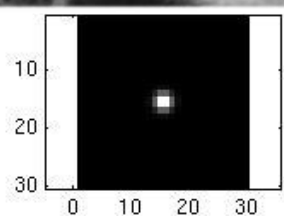
- ∇^2 is the Laplacian operator:

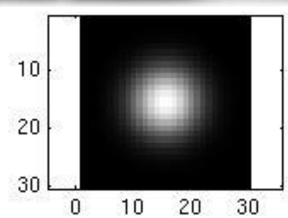
$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

Slide credit: Steve Seitz

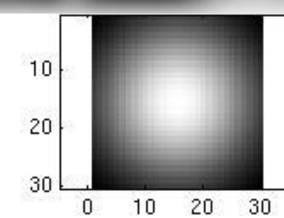
Smoothing with a Gaussian

Recall: parameter σ is the “scale” / “width” / “spread” of the Gaussian kernel, and controls the amount of smoothing.





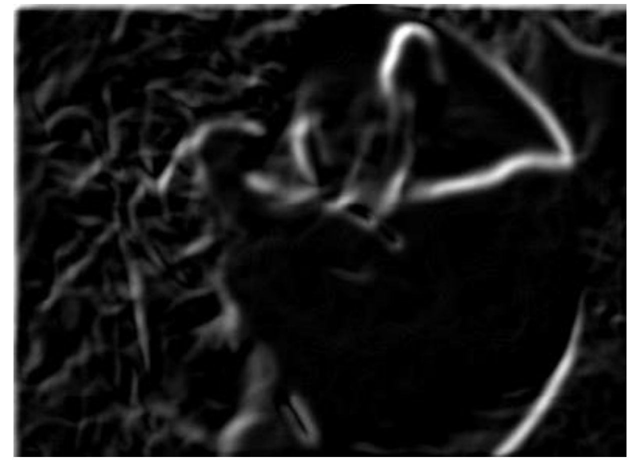
…



Slide credit:
Kristen Grauman

Effect of σ on derivatives

σ = 1 pixel



σ = 3 pixels

The apparent structures differ depending on Gaussian's scale parameter.

Larger values: larger scale edges detected
Smaller values: finer features detected

Slide credit:
Kristen Grauman

So, what scale to choose?

It depends what we're looking for.

Slide credit:
Kristen Grauman

Mask properties

- Smoothing
  - Values positive
  - Sum to 1 → constant regions same as input
  - Amount of smoothing proportional to mask size
  - Remove “high-frequency” components; “low-pass” filter
- Derivatives
  - __Opposite__ signs used to get high response in regions of high contrast
  - Sum to _0_ → no response in constant regions
  - High absolute value at points of high contrast

Slide credit: Kristen Grauman

Seam carving: main idea

[Seam Carving for Content-Aware Image Resizing ,
Shai & Avidan, ACM SIGGRAPH 2007]

Seam carving: main idea

Content-aware resizing

Traditional resizing

[Shai & Avidan, SIGGRAPH 2007]

Real image example

Slide credit:
Kristen Grauman

Seam carving: main idea

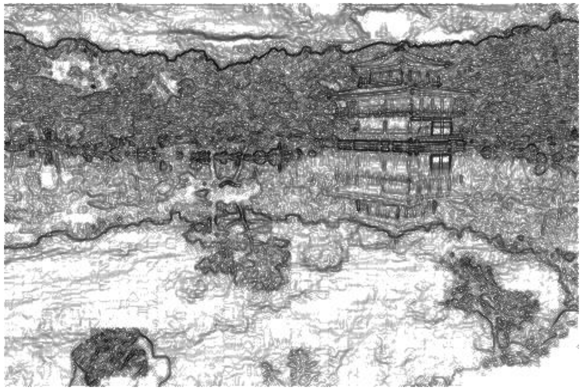
Content-aware resizing

Intuition:

- Preserve the most “interesting” content
  → Prefer to remove pixels with low gradient energy
- To reduce or increase size in one dimension, remove irregularly shaped “seams”
  → Optimal solution via dynamic programming.

Slide credit:
Kristen Grauman

Seam carving: main idea

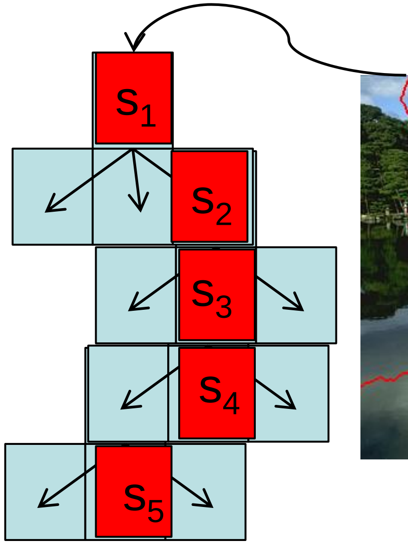


$$Energy(f) = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$

- Want to remove seams where they won’t be very noticeable:
  - Measure “energy” as gradient magnitude
- Choose seam based on **minimum total energy path** across image, subject to 8-connectedness.

Slide credit:
Kristen Grauman

Seam carving: algorithm



$$Energy(f) = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$

Let a vertical seam **s** consist of *h* positions that form an 8-connected path.

Let the cost of a seam be: $Cost(\mathbf{s}) = \sum_{i=1}^{h} Energy(f(s_i))$

Optimal seam minimizes this cost: $\mathbf{s}^* = \min_{\mathbf{s}} Cost(\mathbf{s})$

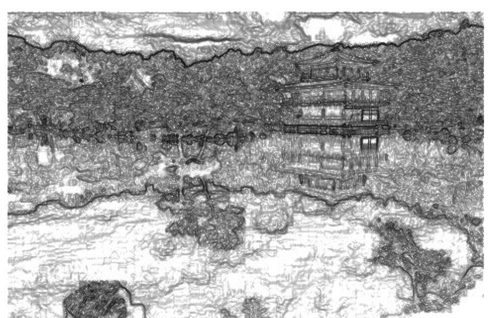
Compute it efficiently with dynamic programming.

Slide credit: Kristen Grauman

How to identify the minimum cost seam?

- First, consider a **greedy** approach:

$$\begin{matrix} 1 & 3 & \boxed{0} \\ 2 & \boxed{8} & 9 \\ 5 & \boxed{2} & 6 \end{matrix}$$



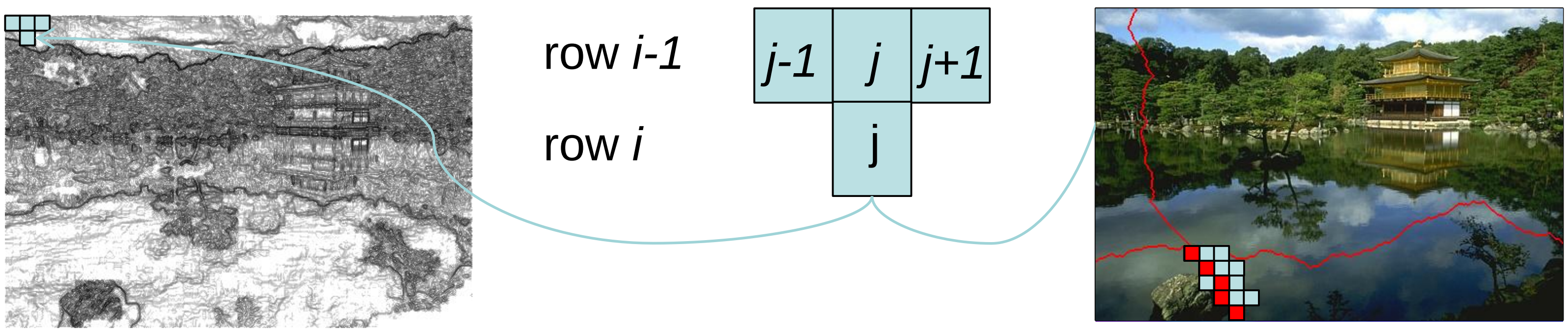
Energy matrix (gradient magnitude)

Slide credit: Kristen Grauman

Seam carving: algorithm

- Compute the cumulative minimum energy for all possible connected seams at each entry *(i,j)*:

$$\mathbf{M}(i,j) = Energy(i,j) + \min\big(\mathbf{M}(i-1,j-1), \mathbf{M}(i-1,j), \mathbf{M}(i-1,j+1)\big)$$



Energy matrix (gradient magnitude)

M matrix: cumulative min energy (for vertical seams)

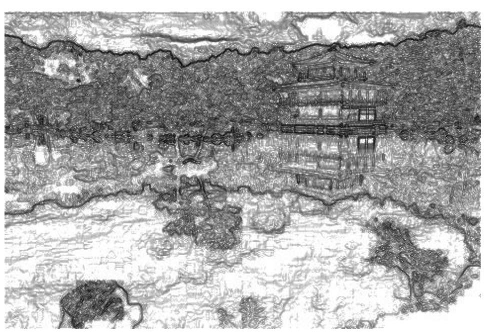
- Then, min value in last row of **M** indicates end of the minimal connected vertical seam.
- Backtrack up from there, selecting min of 3 above in **M**.

Slide credit: Kristen Grauman

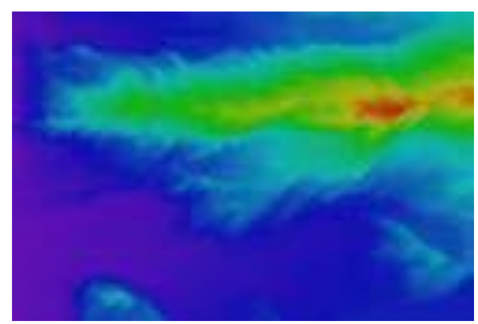
Example

$$\mathbf{M}(i,j) = Energy(i,j) + \min\left(\mathbf{M}(i-1, j-1), \mathbf{M}(i-1, j), \mathbf{M}(i-1, j+1)\right)$$

1	3	0
2	8	9
5	2	6



Energy matrix (gradient magnitude)



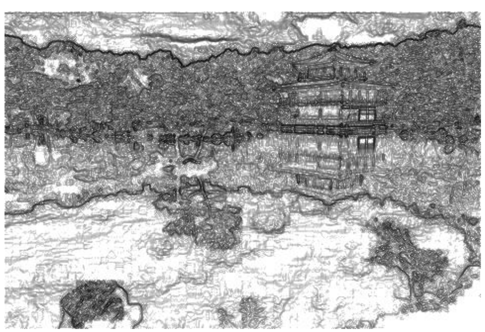
M matrix (for vertical seams)

Slide credit: Kristen Grauman

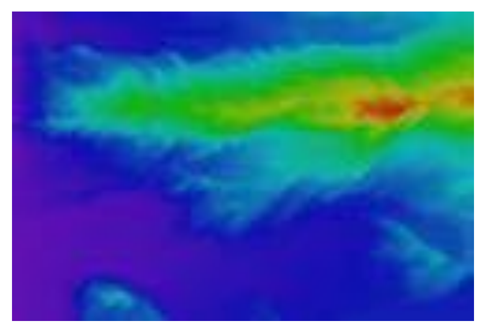
Example

$$\mathbf{M}(i,j) = Energy(i,j) + \min\left(\mathbf{M}(i-1,j-1), \mathbf{M}(i-1,j), \mathbf{M}(i-1,j+1)\right)$$

$$\begin{matrix} \boxed{1} & 3 & 0 \\ \boxed{2} & 8 & 9 \\ 5 & \boxed{2} & 6 \end{matrix} \qquad \begin{matrix} \boxed{1} & 3 & 0 \\ \boxed{3} & 8 & 9 \\ 8 & \boxed{5} & 14 \end{matrix}$$



Energy matrix (gradient magnitude)



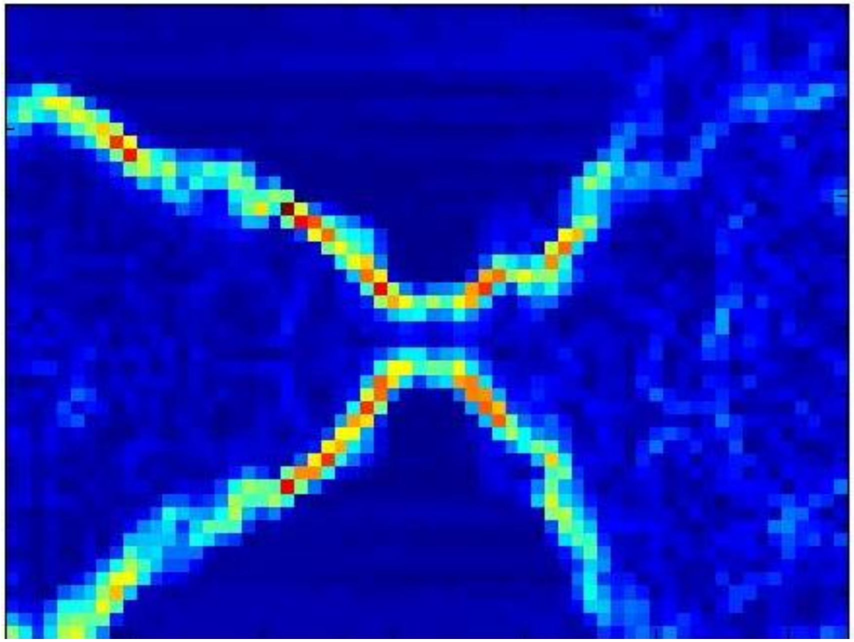
M matrix (for vertical seams)

Slide credit: Kristen Grauman

Real image example

Original Image

Energy Map



Blue = low energy
Red = high energy

Slide credit:
Kristen Grauman

Real image example

Slide credit:
Kristen Grauman

Other notes on seam carving

- Analogous procedure for horizontal seams
- Can also insert seams to *increase* size of image in either dimension
  - Duplicate optimal seam, averaged with neighbors
- Other energy functions may be plugged in
  - E.g., color-based, interactive,…
- Can use combination of vertical and horizontal seams

Gradients -> edges

Primary edge detection steps:

1. Smoothing: suppress noise
2. Edge enhancement: filter for contrast
3. Edge localization

   Determine which local maxima from filter output are actually edges vs. noise

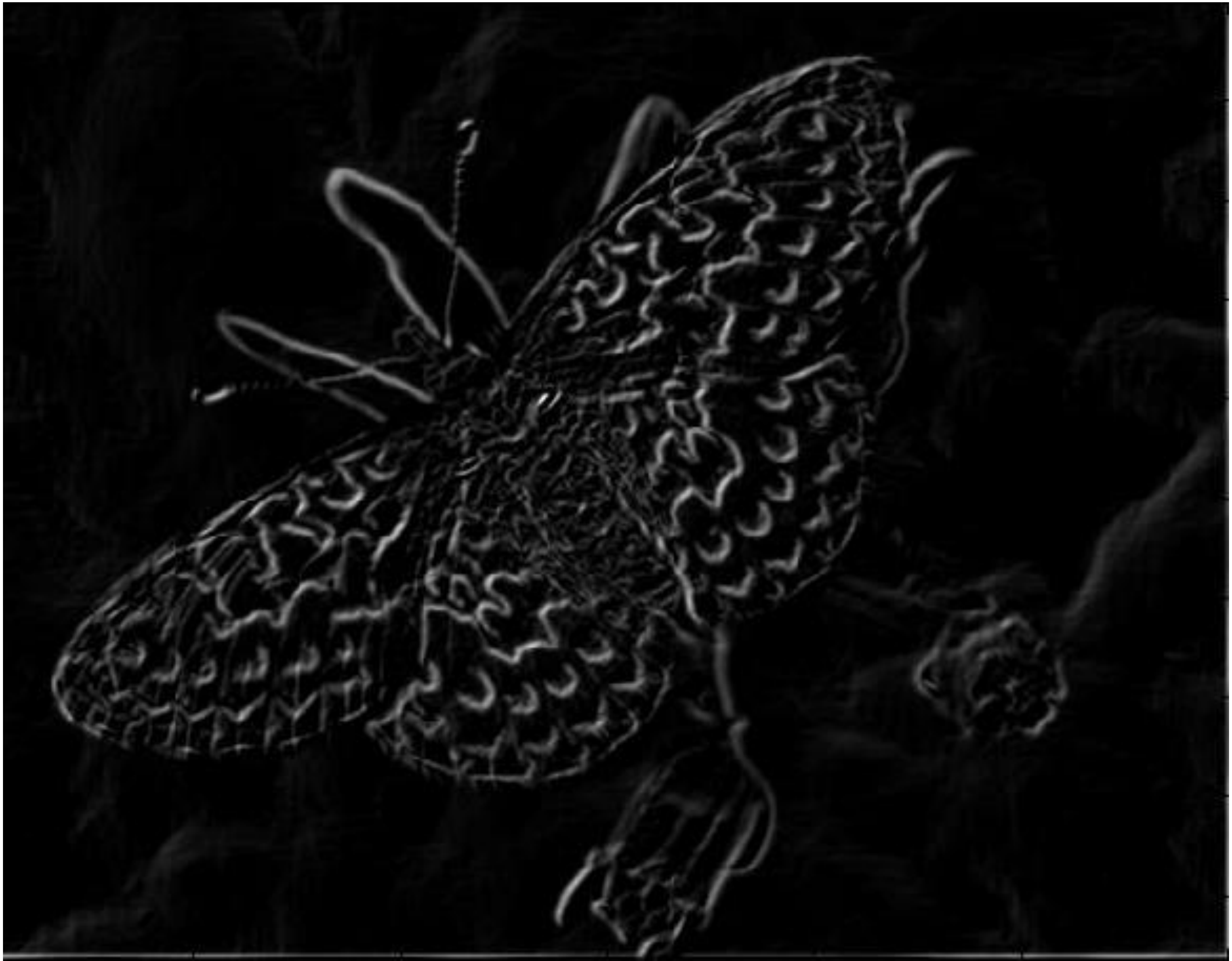
   - Threshold, Thin

Thresholding

- Choose a threshold value t
- Set any pixels less than t to zero (off)
- Set any pixels greater than or equal to t to one (on)

Original image

Gradient magnitude image



Thresholding gradient with a lower threshold

Thresholding gradient with a higher threshold

Canny edge detector

- Filter image with derivative of Gaussian
- Find magnitude and orientation of gradient
- **Non-maximum suppression**:
  - Thin wide “ridges” down to single pixel width
- **Linking and thresholding** (**hysteresis**):
  - Define two thresholds: low and high
  - Use the high threshold to start edge curves and the low threshold to continue them

- MATLAB: **`edge(image, ‘canny’);`**
- **`>>help edge`**

Source: D. Lowe, L. Fei-Fei

The Canny edge detector

original image (Lena)

Slide credit: Steve Seitz

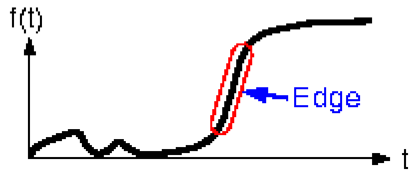
The Canny edge detector

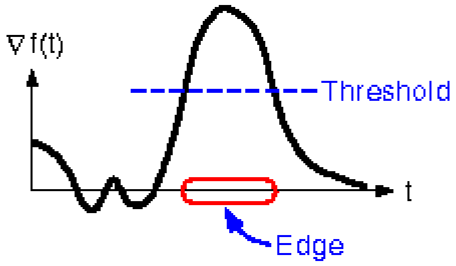
norm of the gradient

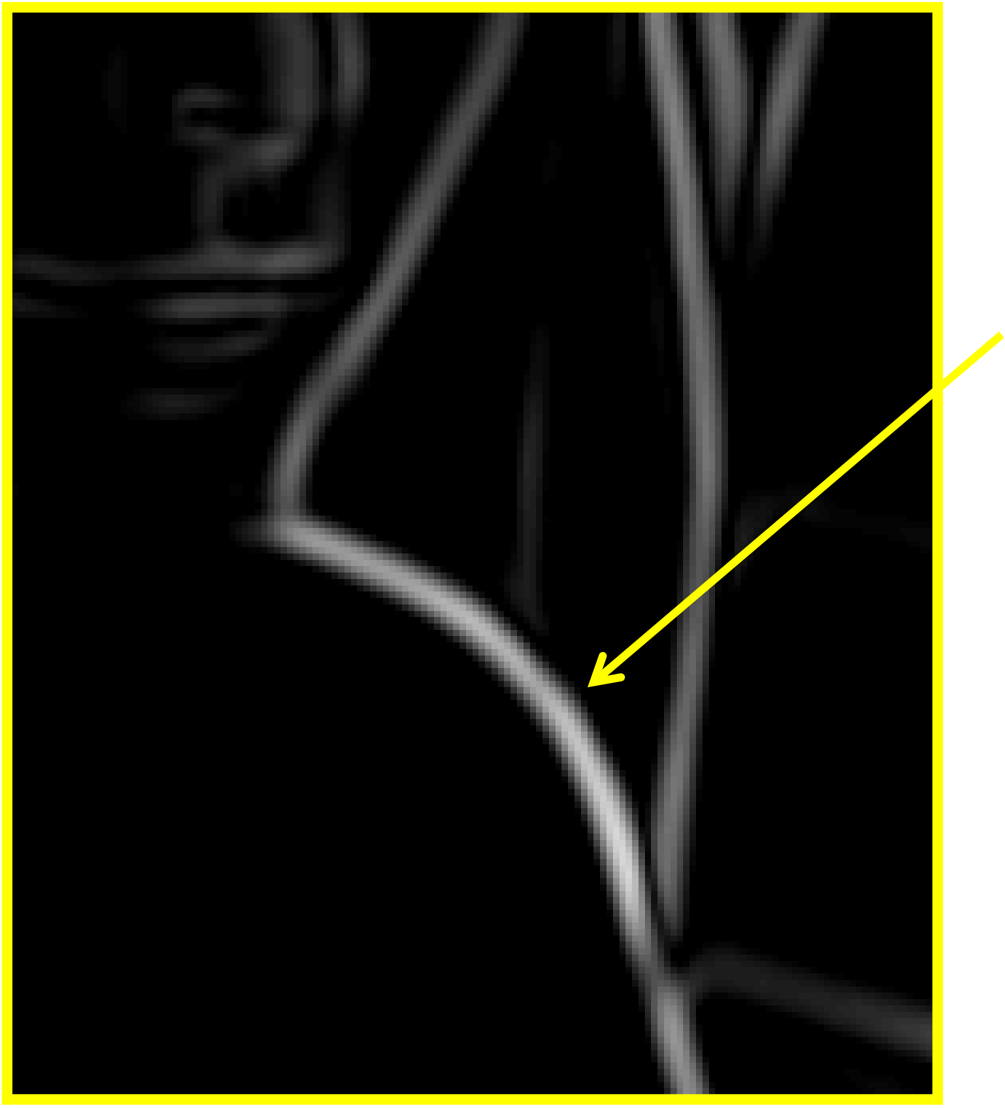
The Canny edge detector

thresholding

The Canny edge detector

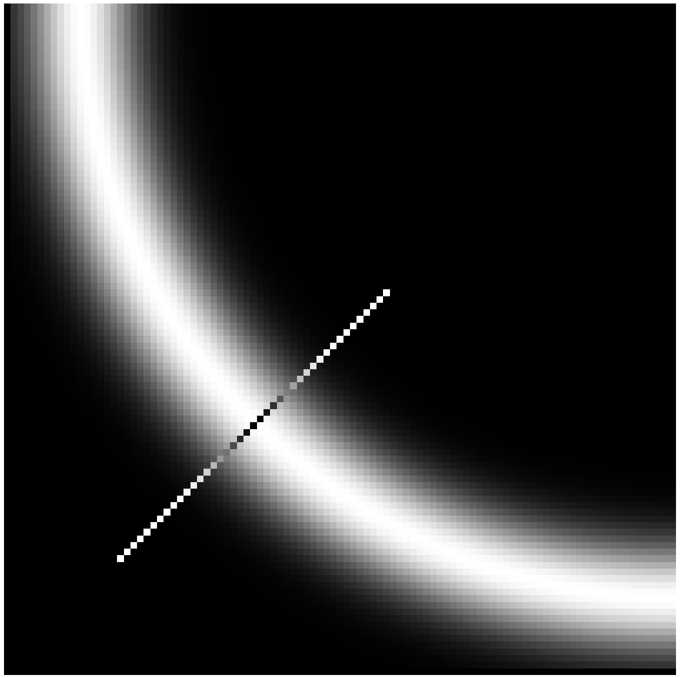


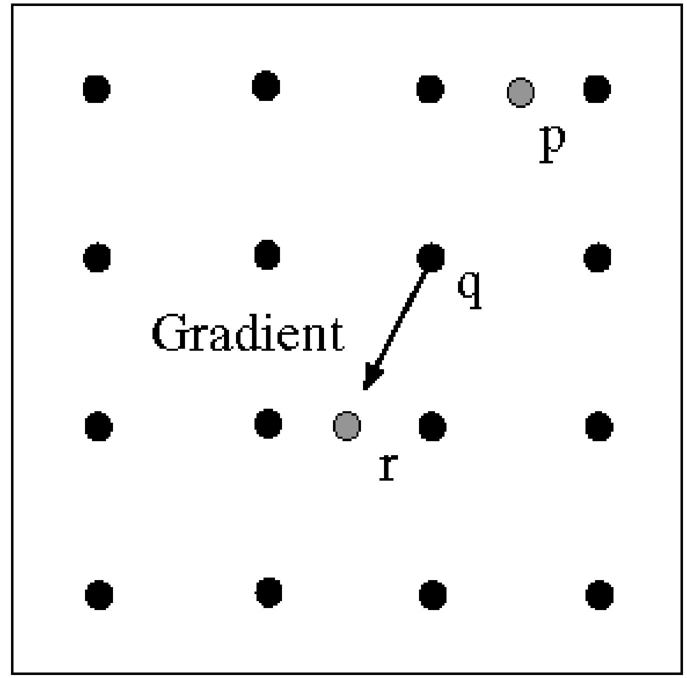




How to turn these thick regions of the gradient into curves?

Non-maximum suppression





Check if pixel is local maximum along gradient direction, select single max across width of the edge

- requires checking interpolated pixels p and r

The Canny edge detector

Problem: pixels along this edge didn't survive the thresholding

thinning
(non-maximum suppression)

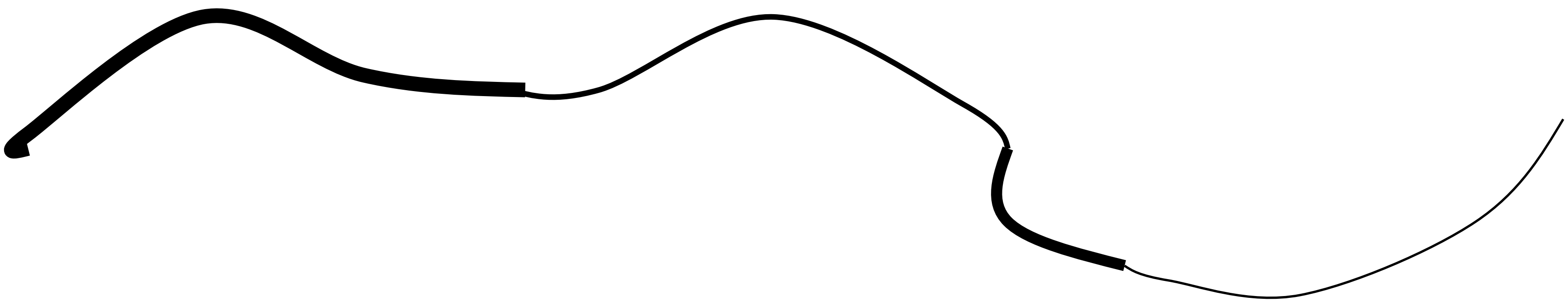
Hysteresis thresholding

- Threshold at low/high levels to get weak/strong edge pixels
- Do connected components, starting from strong edge pixels

Credit: James Hays

Hysteresis thresholding

- Use a high threshold to start edge curves, and a low threshold to continue them.



Source: Steve Seitz

Final Canny Edges

Credit: James Hays

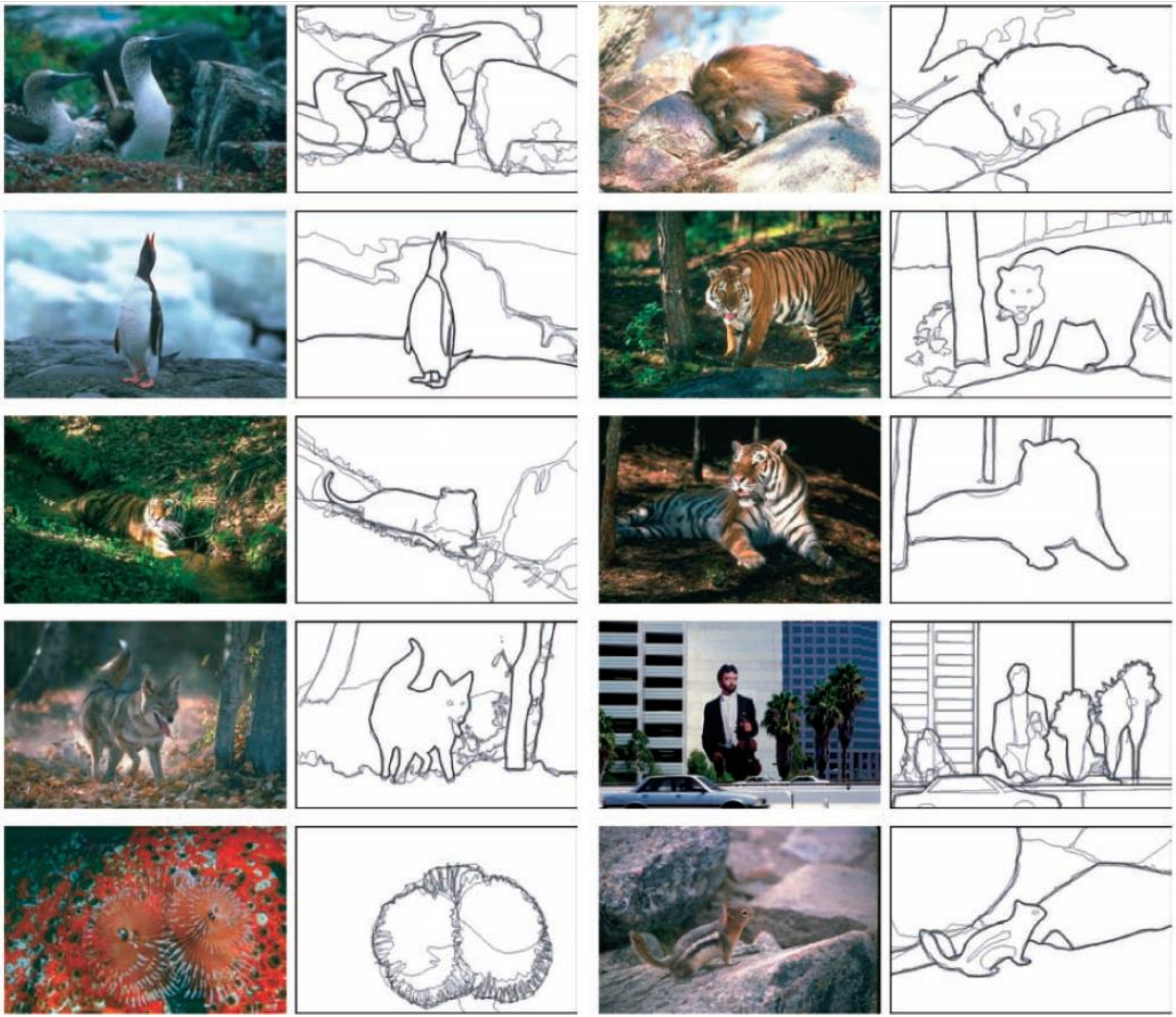
Recap: Canny edge detector

- Filter image with derivative of Gaussian
- Find magnitude and orientation of gradient
- **Non-maximum suppression**:
  - Thin wide “ridges” down to single pixel width
- **Linking and thresholding** (**hysteresis**):
  - Define two thresholds: low and high
  - Use the high threshold to start edge curves and the low threshold to continue them

- MATLAB: **`edge(image, ‘canny’);`**
- **`>>help edge`**

Source: D. Lowe, L. Fei-Fei

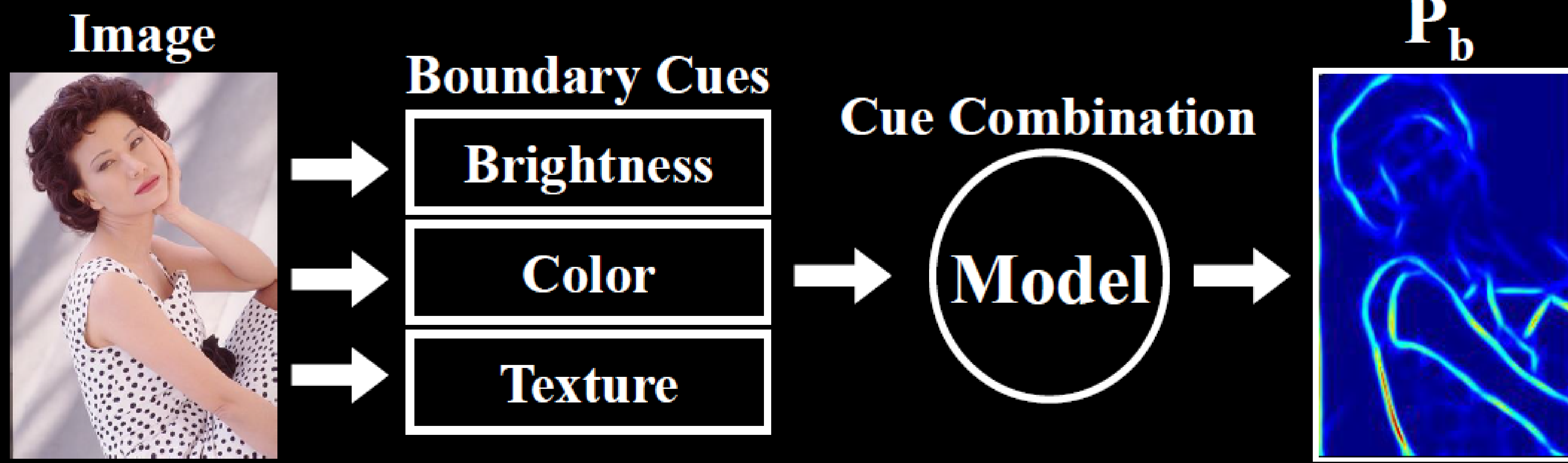
Learn from humans which combination of features is most indicative of a "good" contour?



[D. Martin et al. PAMI 2004]

Human-marked segment boundaries

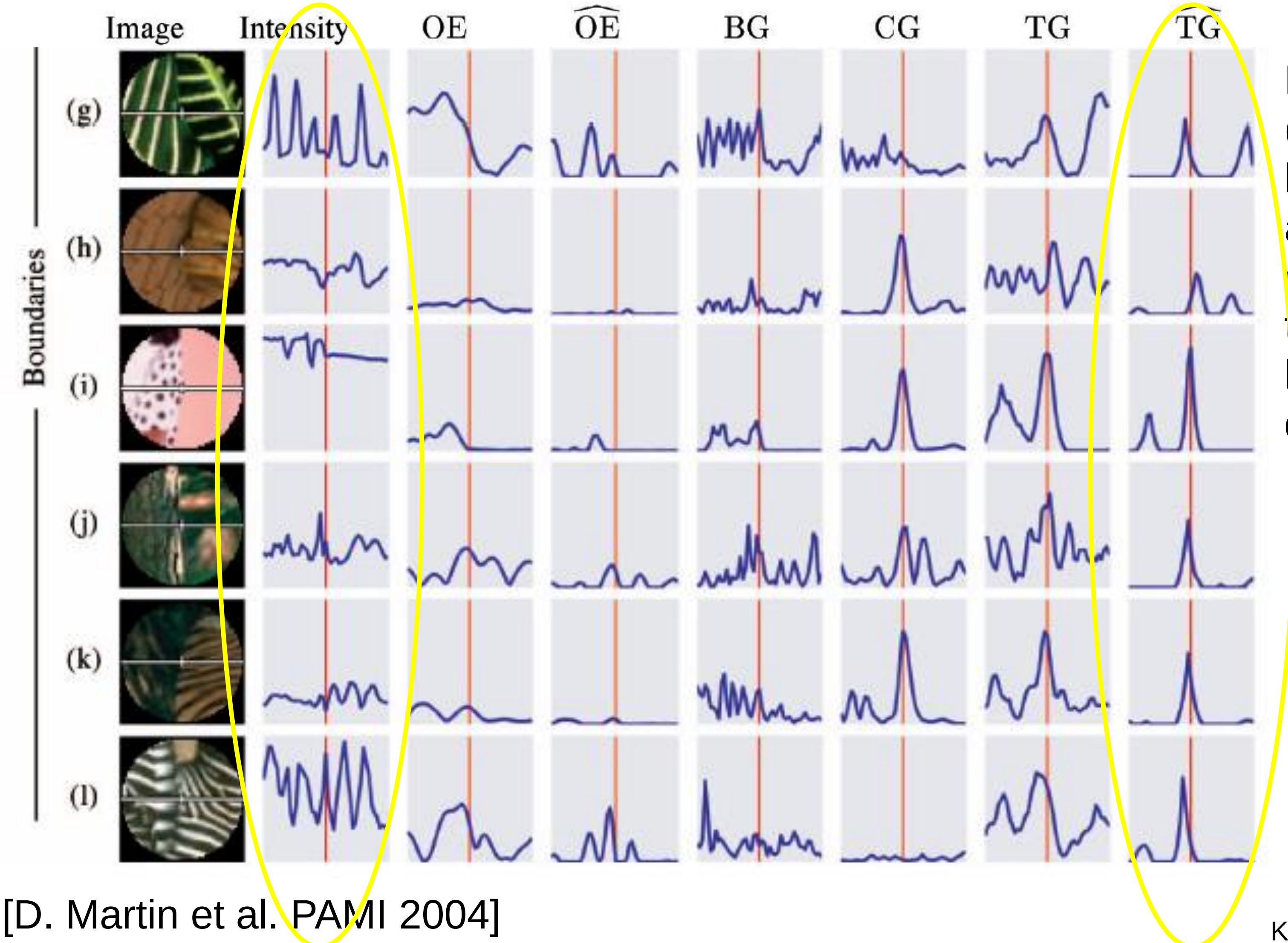
Dataflow



Challenges: texture cue, cue combination

Goal: learn the posterior probability of a boundary $P_b(x,y,\theta)$ from local information only

What features are responsible for perceived edges?

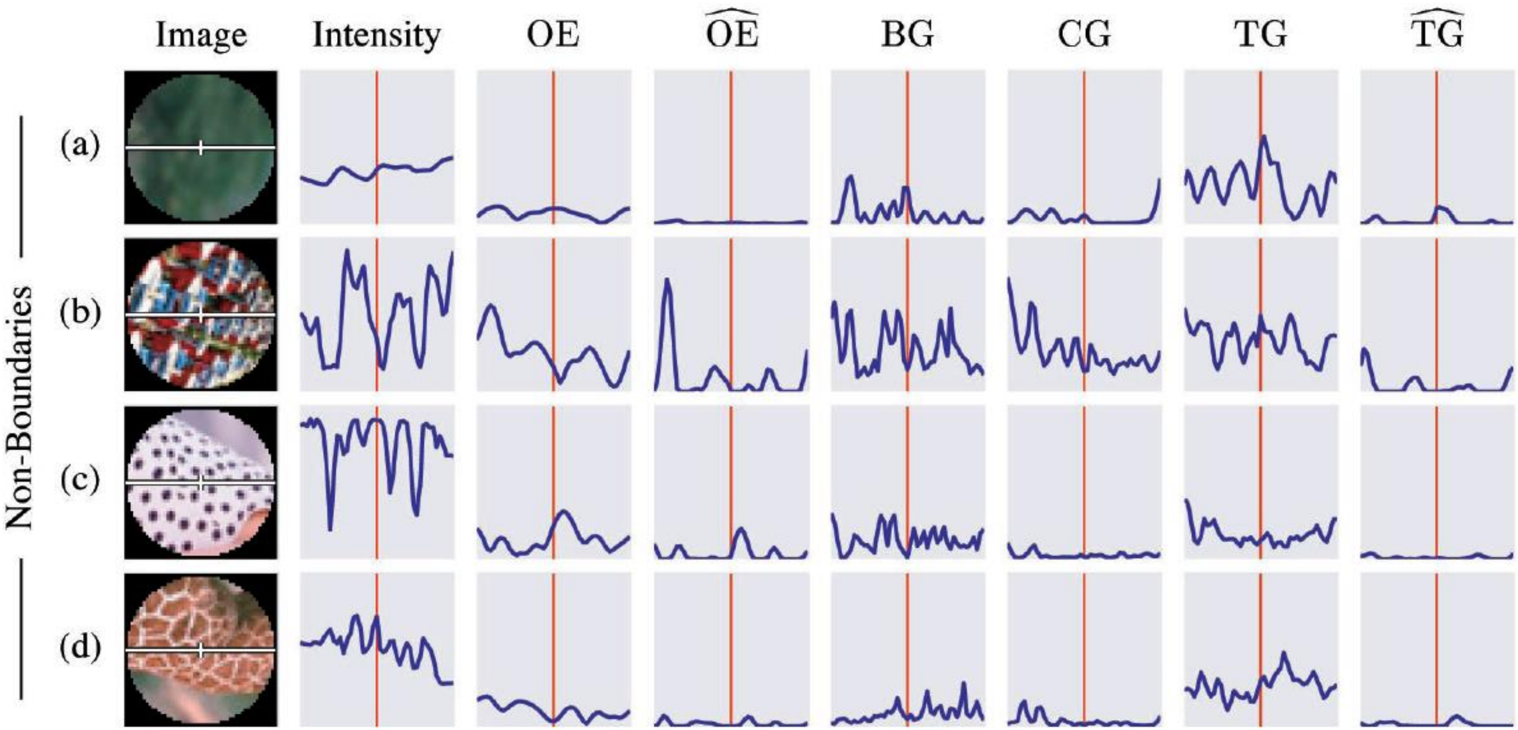


Feature profiles (oriented energy, brightness, color, and texture gradients) along the patch's horizontal diameter

[D. Martin et al. PAMI 2004]

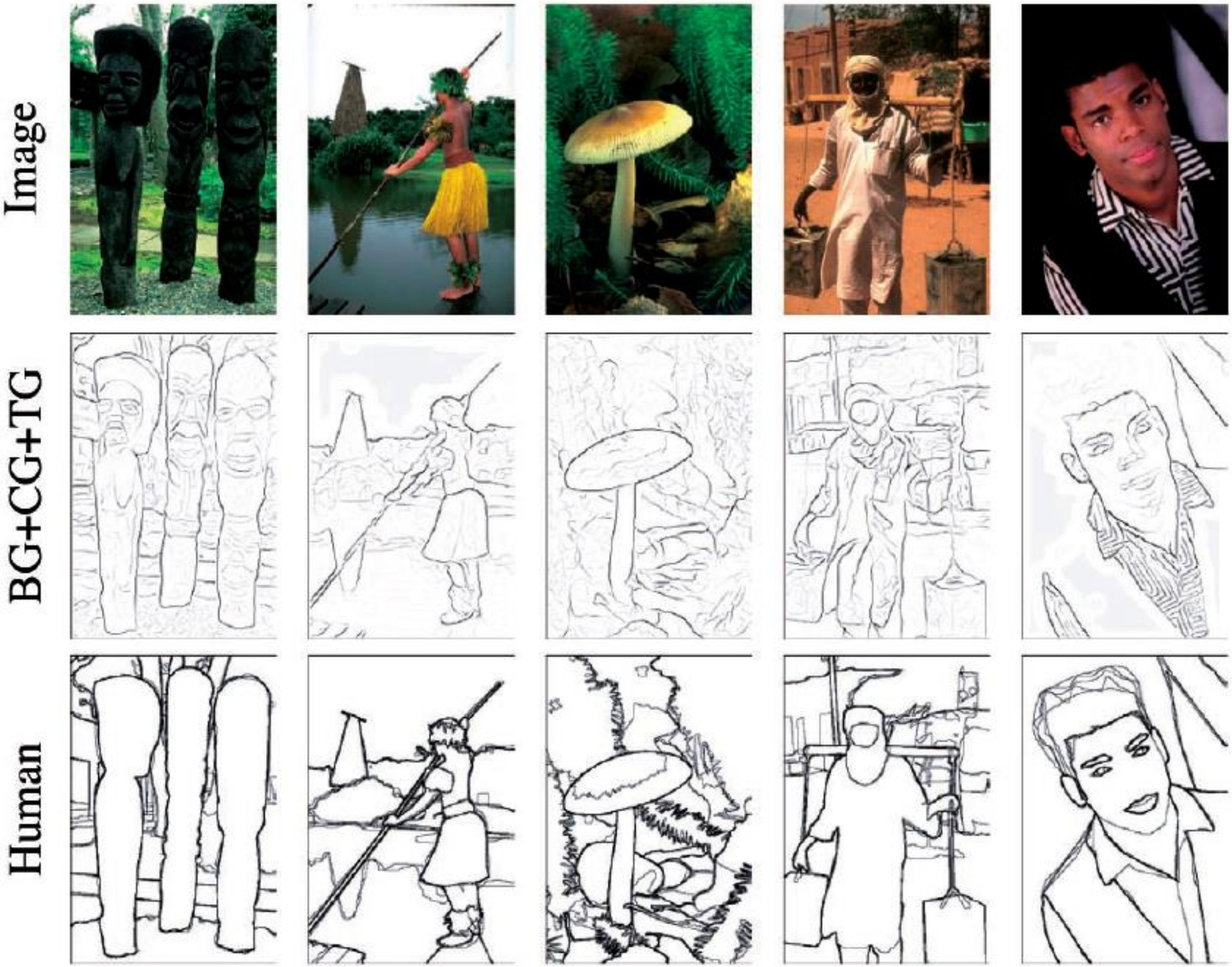
Kristen Grauman, UT-Austin

What features are responsible for perceived edges?



Feature profiles (oriented energy, brightness, color, and texture gradients) along the patch's horizontal diameter

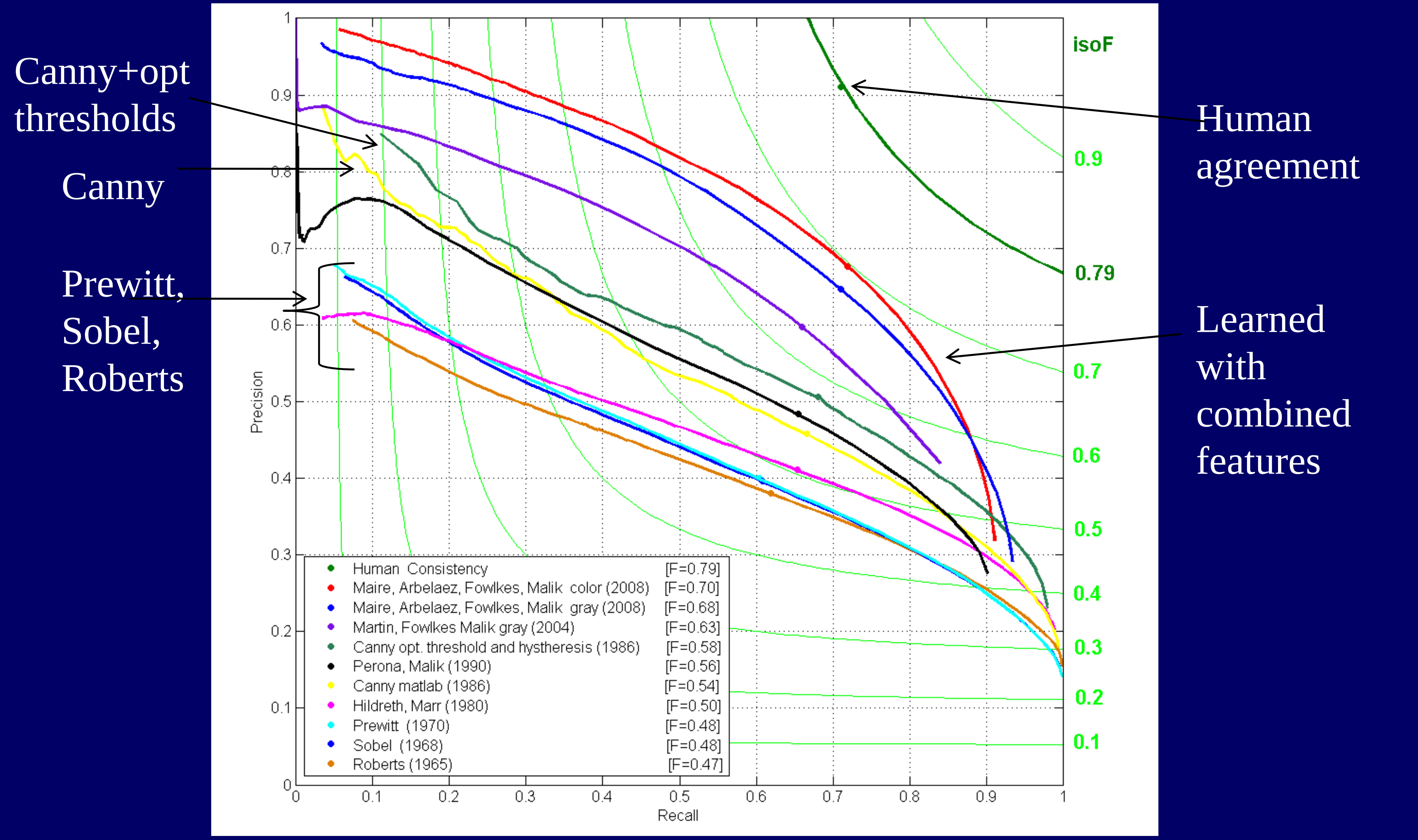
[D. Martin et al. PAMI 2004]



[D. Martin et al. PAMI 2004]

Kristen Grauman, UT-Austin

Contour Detection



Source: Jitendra Malik:
http://www.cs.berkeley.edu/~malik/malik-talks-ptrs.html

Recall: image filtering

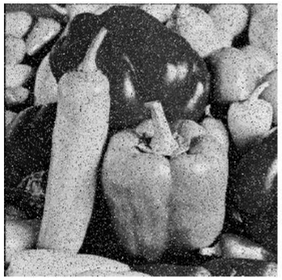
- Compute a function of the local neighborhood at each pixel in the image
  - Function specified by a “filter” or mask saying how to combine values from neighbors.

- Uses of filtering:
  - Enhance an image (denoise, resize, etc)
  - Extract information (texture, edges, etc)
  - Detect patterns (template matching)

Adapted from Derek Hoiem

Filters for features

- Map raw pixels to an intermediate representation that will be used for subsequent processing

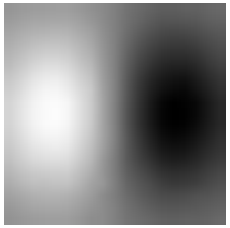


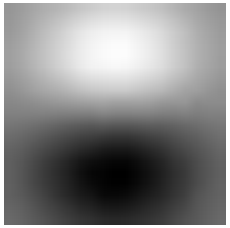
- Goal: reduce amount of data, discard redundancy, preserve what's useful

Template matching

- Filters as **templates**:

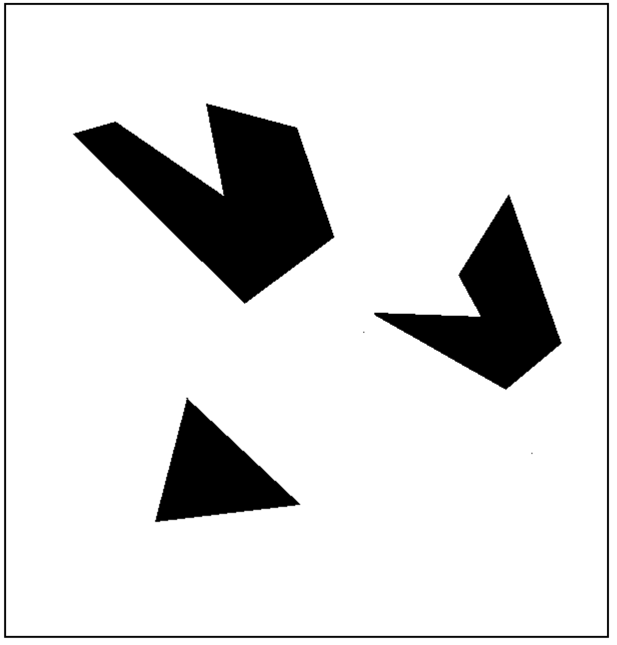
  Note that filters look like the effects they are intended to find --- “matched filters”



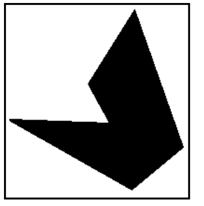


- Use normalized cross-correlation score to find a given pattern (template) in the image.
- Normalization needed to control for relative brightnesses.

Template matching



Scene

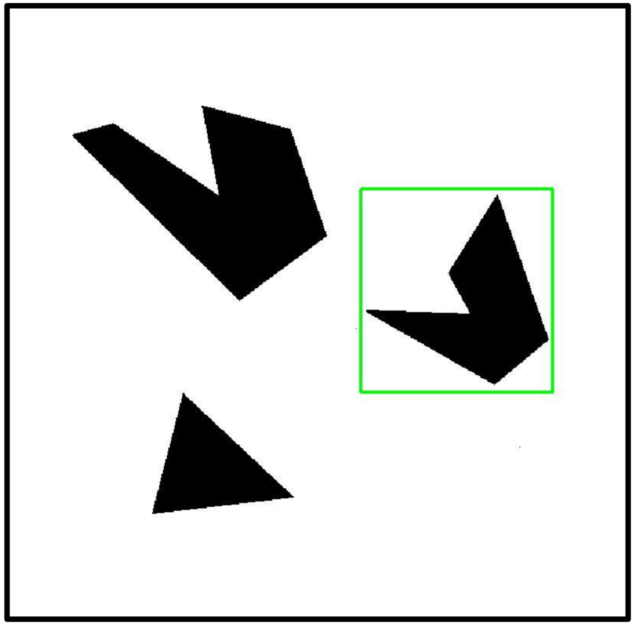


Template (mask)

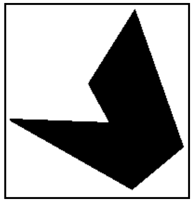
A toy example

Slide credit:
Kristen Grauman

Template matching



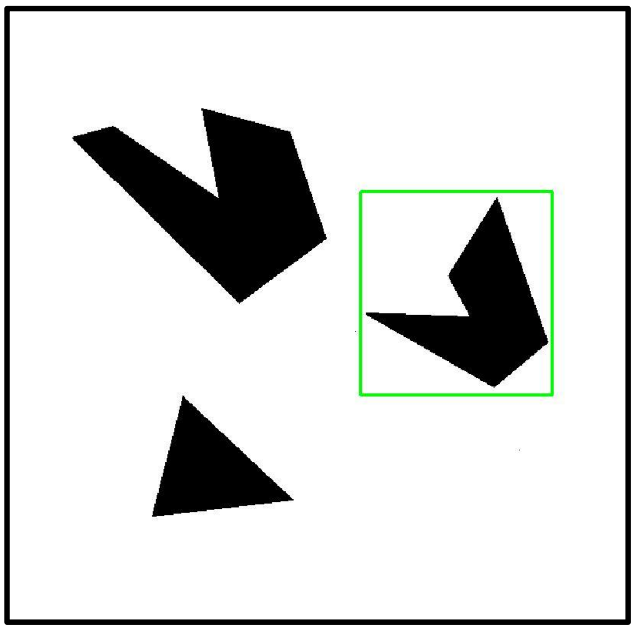
Detected template



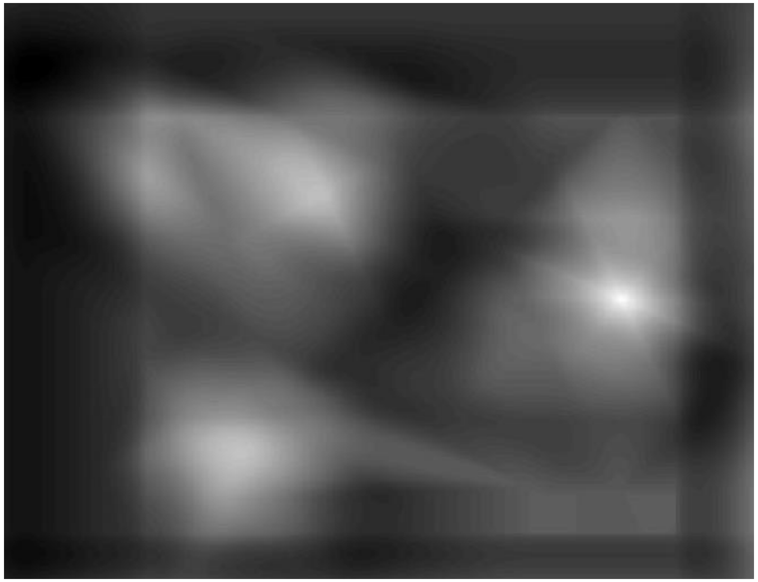
Template

Slide credit:
Kristen Grauman

Template matching



Detected template

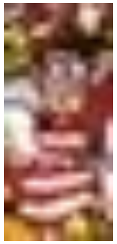


Correlation map

Slide credit:
Kristen Grauman

Where’s Waldo?

Scene

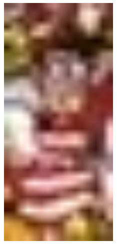


Template

Slide credit:
Kristen Grauman

Where's Waldo?

Detected template

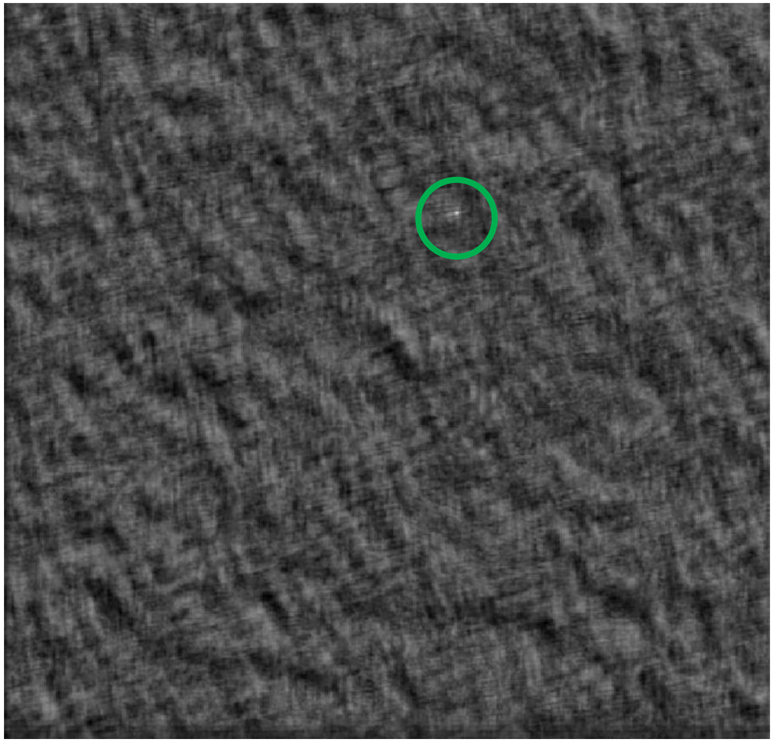


Template

Slide credit:
Kristen Grauman

Where’s Waldo?

Detected template



Correlation map

Slide credit:
Kristen Grauman

Template matching

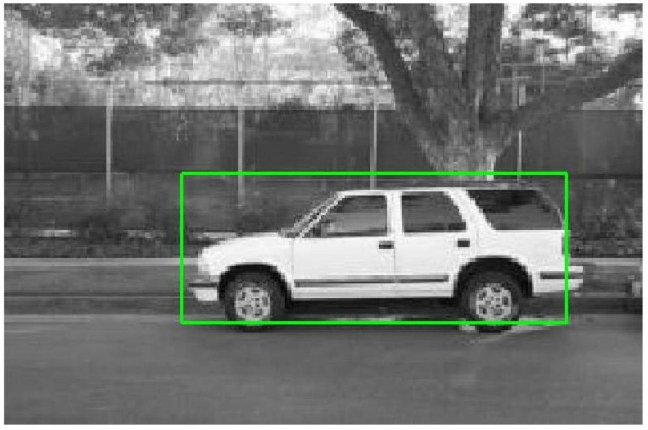
Scene

Template

What if the template is not identical to some subimage in the scene?

Slide credit:
Kristen Grauman

Template matching



Detected template

Template

Match can be meaningful, if scale, orientation, and general appearance is right.

How to find at any scale?

Slide credit:
Kristen Grauman

Recap: Mask properties

- <u>Smoothing</u>
  - Values positive
  - Sum to 1 → constant regions same as input
  - Amount of smoothing proportional to mask size
  - Remove “high-frequency” components; “low-pass” filter
- <u>Derivatives</u>
  - Opposite signs used to get high response in regions of high contrast
  - Sum to 0 → no response in constant regions
  - High absolute value at points of high contrast
- <u>Filters act as templates</u>
  - Highest response for regions that “look the most like the filter”
  - Dot product as correlation

Slide credit:
Kristen Grauman

Summary

- Image gradients
- Seam carving – gradients as “energy”
- Gradients → edges and contours
- Template matching
  – Image patch as a filter

Assignments

- 自己构建一个边缘检测器（以matlab、octave、C++均可）。注释每一步的含义。（可以参考书中习题4.7和4.8，观察与Canny等边缘检测算子实现效果的差异）

- 阅读论文：Seam Carving for Content-Aware Image Resizing，Shai & Avidan, ACM SIGGRAPH 2007。