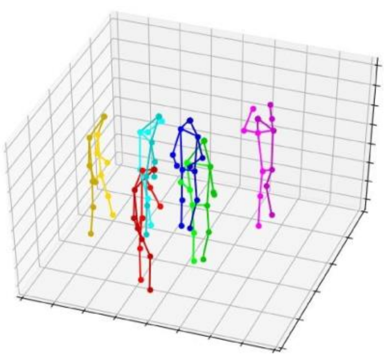
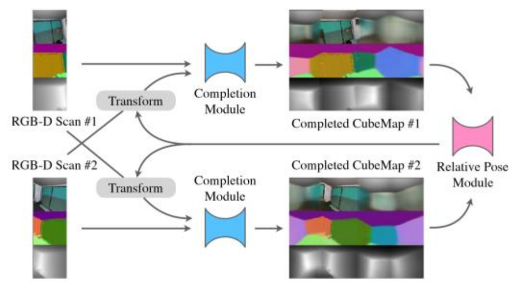
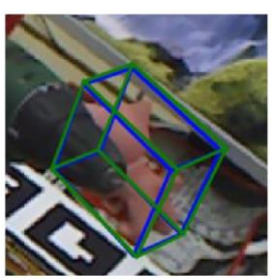
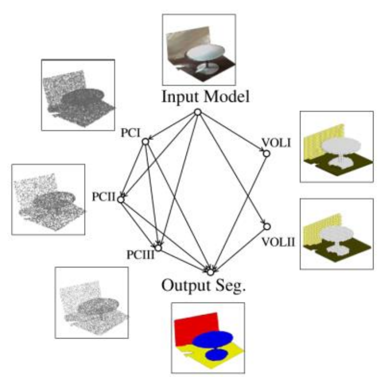
CS376 Computer Vision
Lecture 3: Linear Filters

Qixing Huang
January 30th 2019









Last Lecture

- Image Noise

- Image Filtering
  - Denoising

This Lecture

- Edge Detection
- Template Matching

Edge detection

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

Edge detection is a fundamental tool in image processing, machine vision and computer vision, particularly in the areas of feature detection and feature extraction.

Source:
https://en.wikipedia.org/wiki/Edge_detection#/media/File:%C3%84%C3%A4retuvastuse_n%C3%A4ide.png

Edge detection

Edge detection

From Wikipedia, the free encyclopedia

Edge detection includes a variety of mathematical methods that aim at identifying points in a digital image at which the image brightness changes sharply or, more formally, has discontinuities. The points at which image brightness changes sharply are typically organized into a set of curved line segments termed *edges*. The same problem of finding discontinuities in one-dimensional signals is known as step detection and the problem of finding signal discontinuities over time is known as change detection. Edge detection is a fundamental tool in image processing, machine vision and computer vision, particularly in the areas of feature detection and feature extraction.[1]

Contents [hide]

Feature detection

Edge detection
Canny · Deriche · Differential · Sobel · Prewitt · Roberts cross

Corner detection
Harris operator · Shi and Tomasi · Level curve curvature · Hessian feature strength measures · SUSAN · FAST

Blob detection
Laplacian of Gaussian (LoG) · Difference of Gaussians (DoG) · Determinant of Hessian (DoH) · Maximally stable extremal regions · PCBR

Ridge detection

Hough transform
Hough transform · Generalized Hough transform

Structure tensor
Structure tensor · Generalized structure tensor

Affine invariant feature detection
Affine shape adaptation · Harris affine · Hessian affine

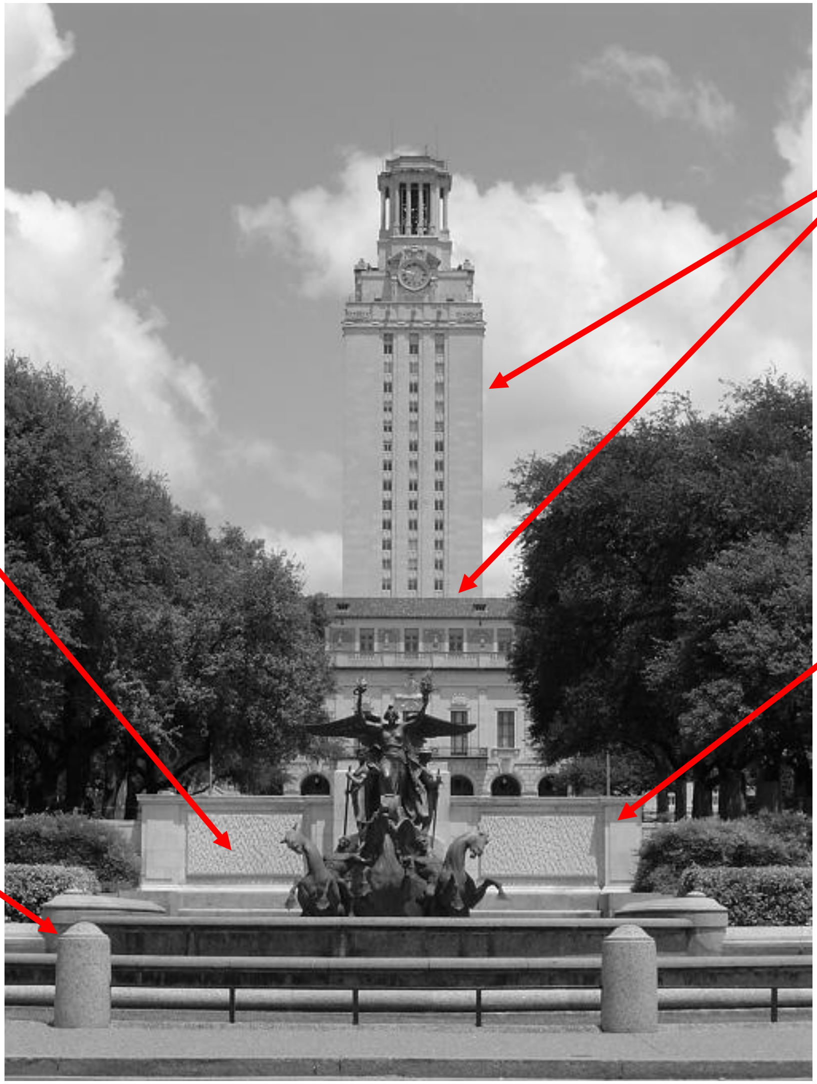
Feature description

Besides the textbook, I also recommend Wikipedia pages
(Up to date updates and fairly accurate)

What causes an edge?

changes in material properties and

discontinuities in surface orientation



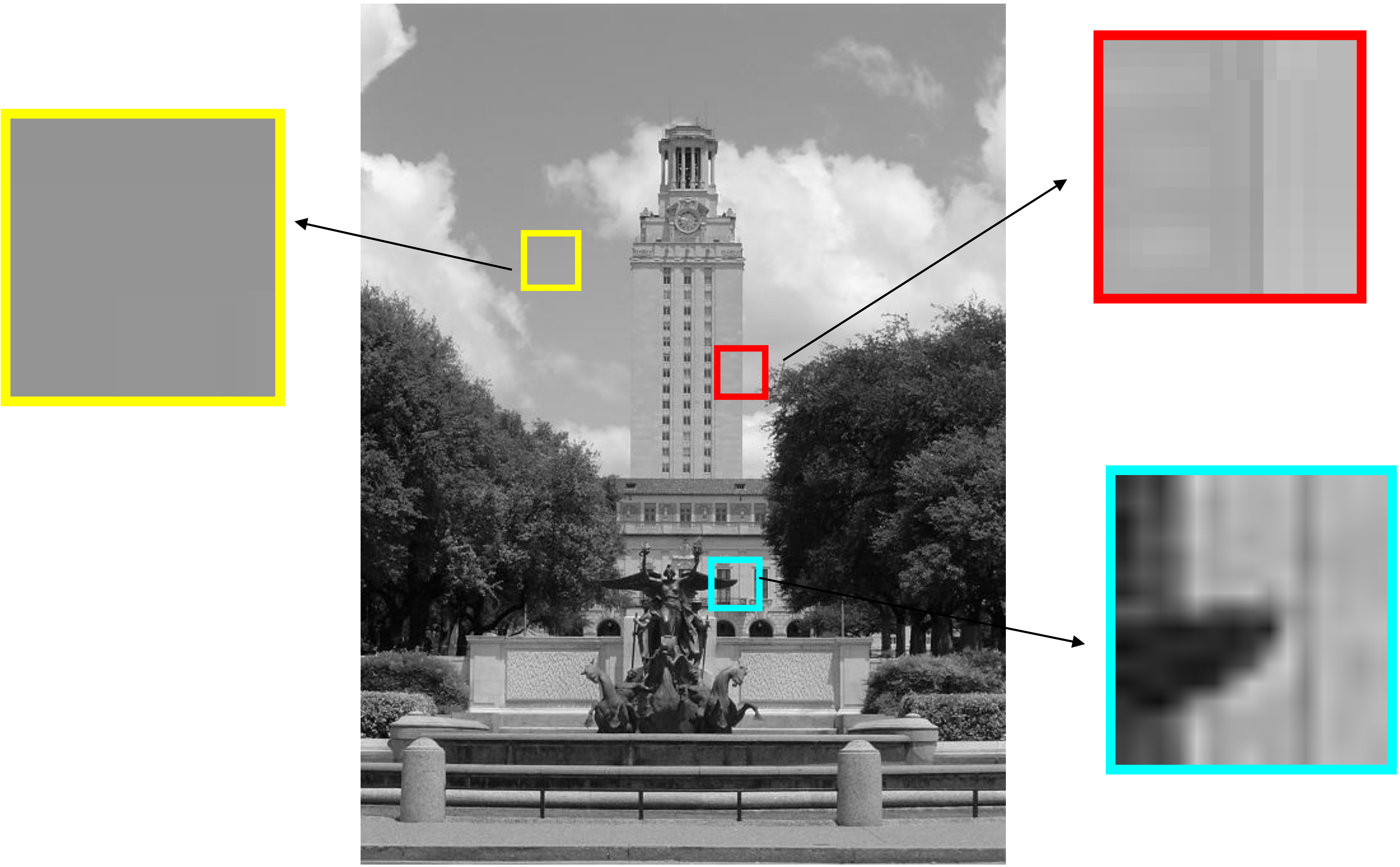
discontinuities in depth

variations in scene illumination/Shadow

More from https://en.wikipedia.org/wiki/Edge_detection

Slide credit: Kristen Grauman

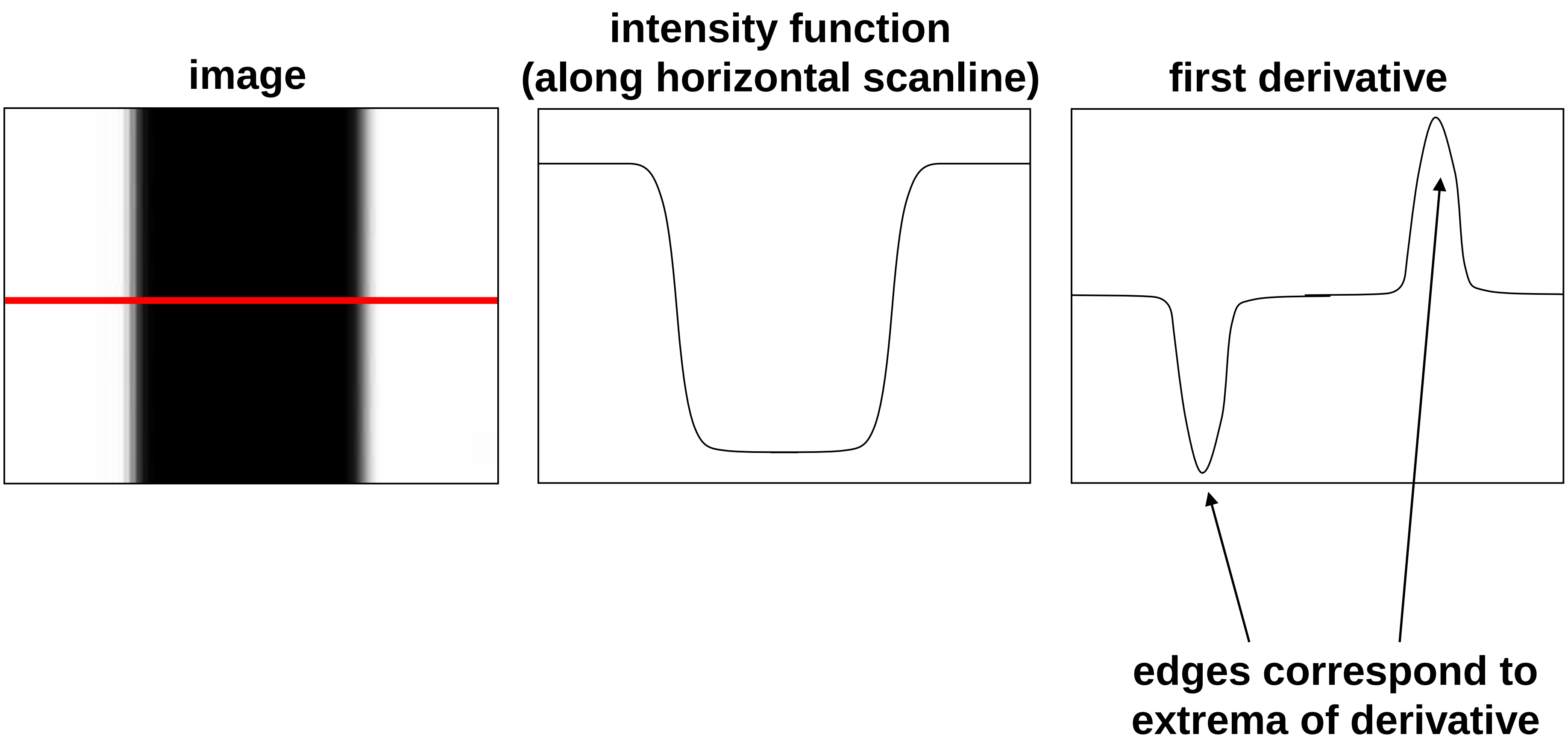
Edges/gradients and invariance



Slide credit:
Kristen Grauman

Derivatives and edges

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

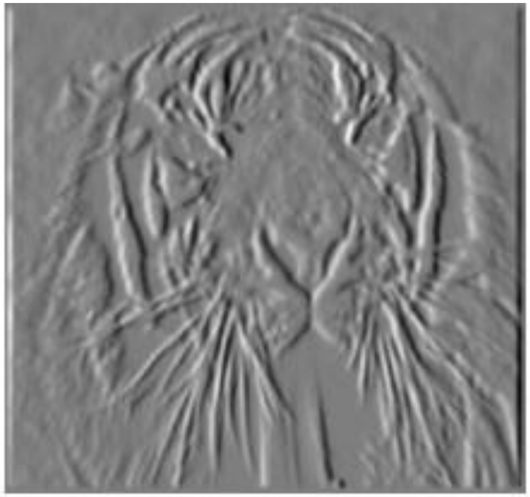


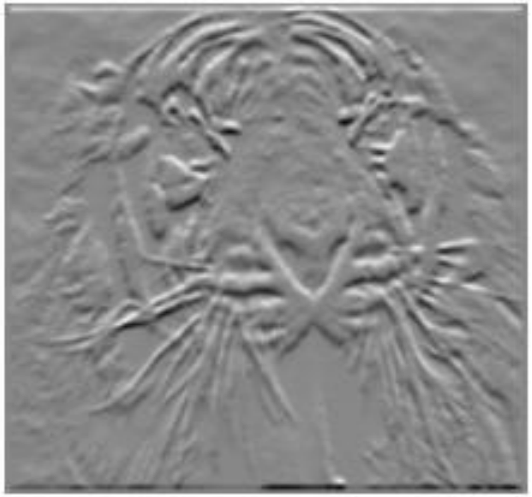
Source: L. Lazebnik

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)

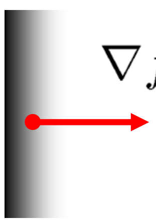
Slide credit:
Kristen Grauman

Image gradient

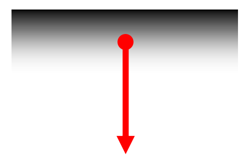
The gradient of an image:

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

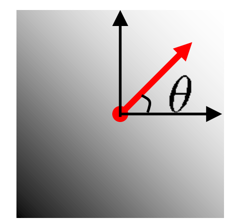
The gradient points in the direction of most rapid change in intensity



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



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



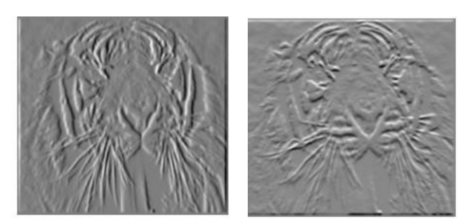
$$\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right]$$

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

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

The **edge strength** is given by the gradient magnitude

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



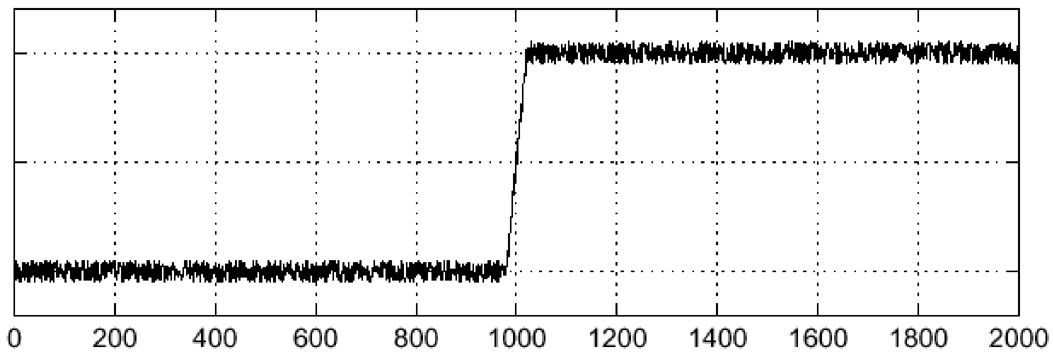
Slide credit Steve Seitz

Effects of noise

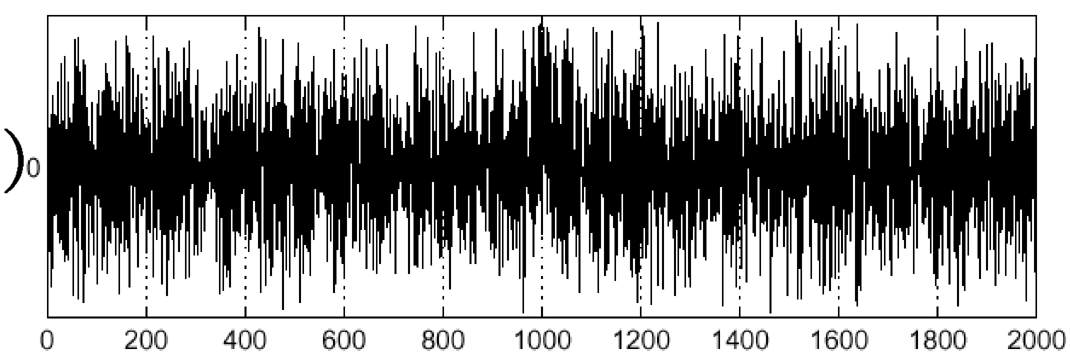
Consider a single row or column of the image

– Plotting intensity as a function of position gives a signal

$$f(x)$$



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



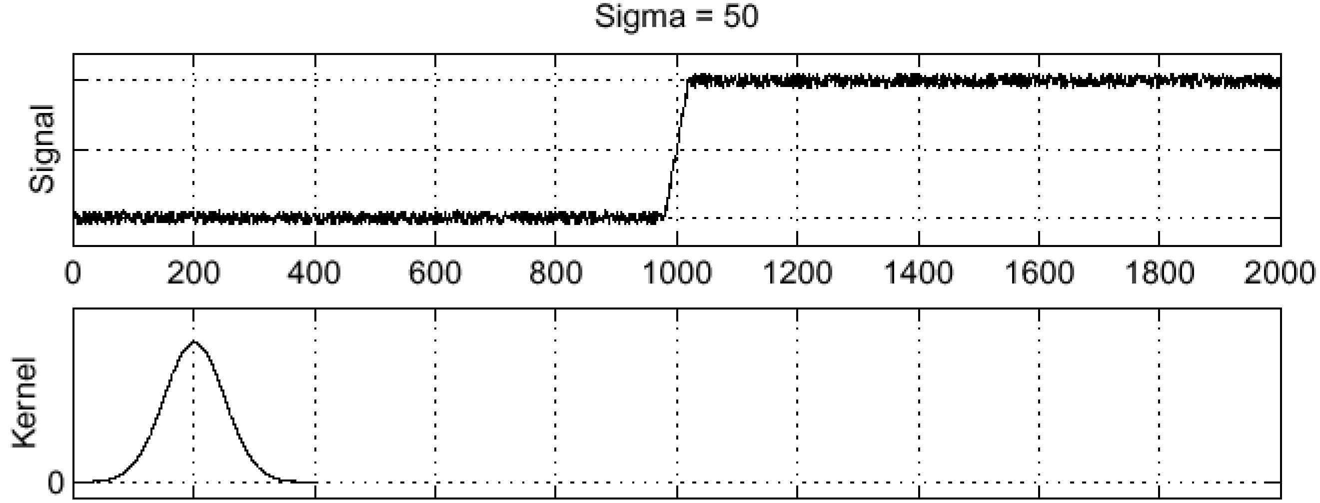
Where is the edge?

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

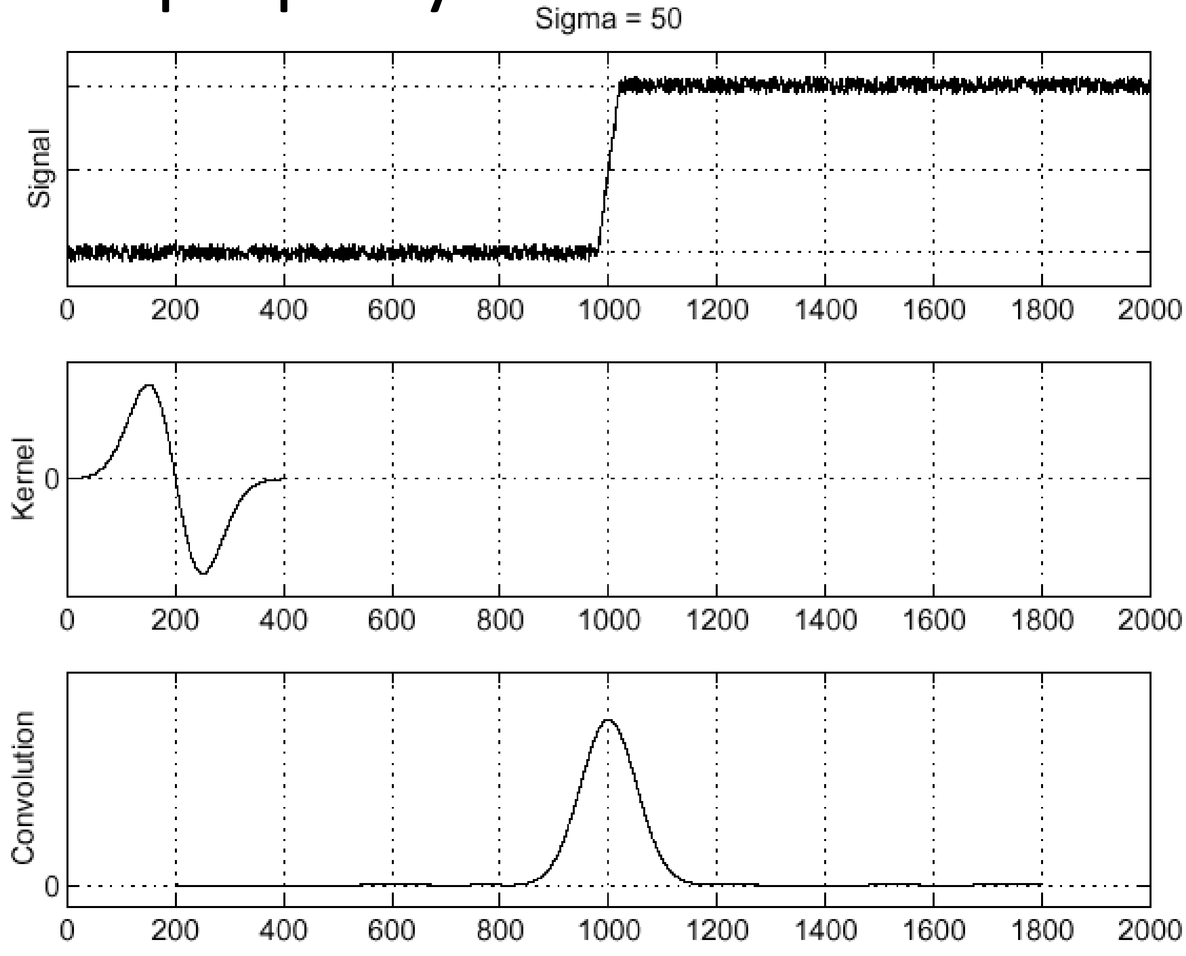
$$\frac{\partial}{\partial x}(h \star f) = (\frac{\partial}{\partial x}h) \star f$$

Differentiation property of convolution.

f

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

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

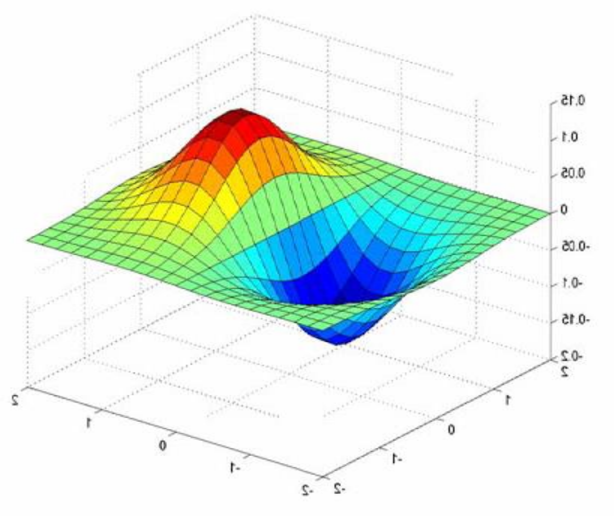


Slide credit Steve Seitz

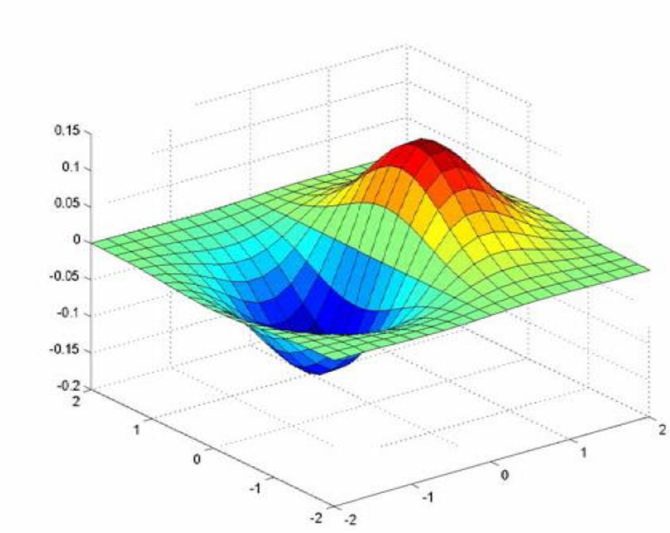
Derivative of Gaussian filters

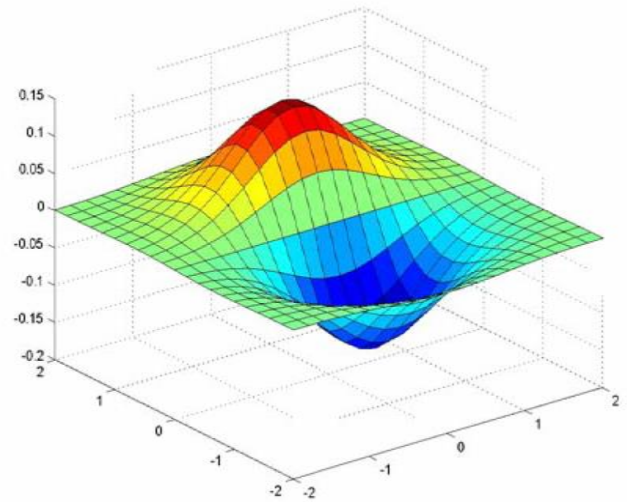
$$(I \otimes g) \otimes h = I \otimes (g \otimes h)$$

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

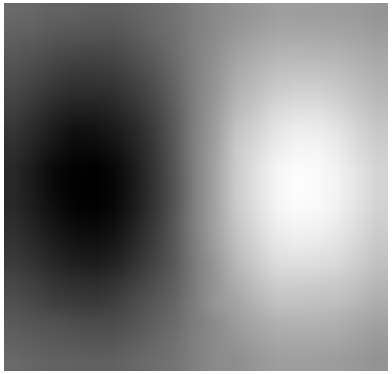


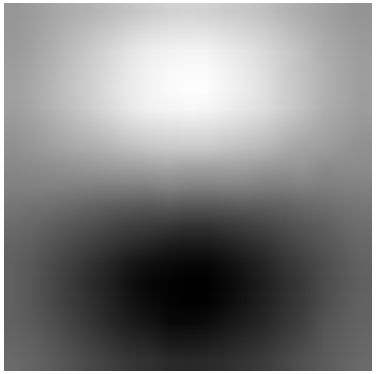
Derivative of Gaussian filters





***x*-direction** ***y*-direction**





Source: L. Lazebnik

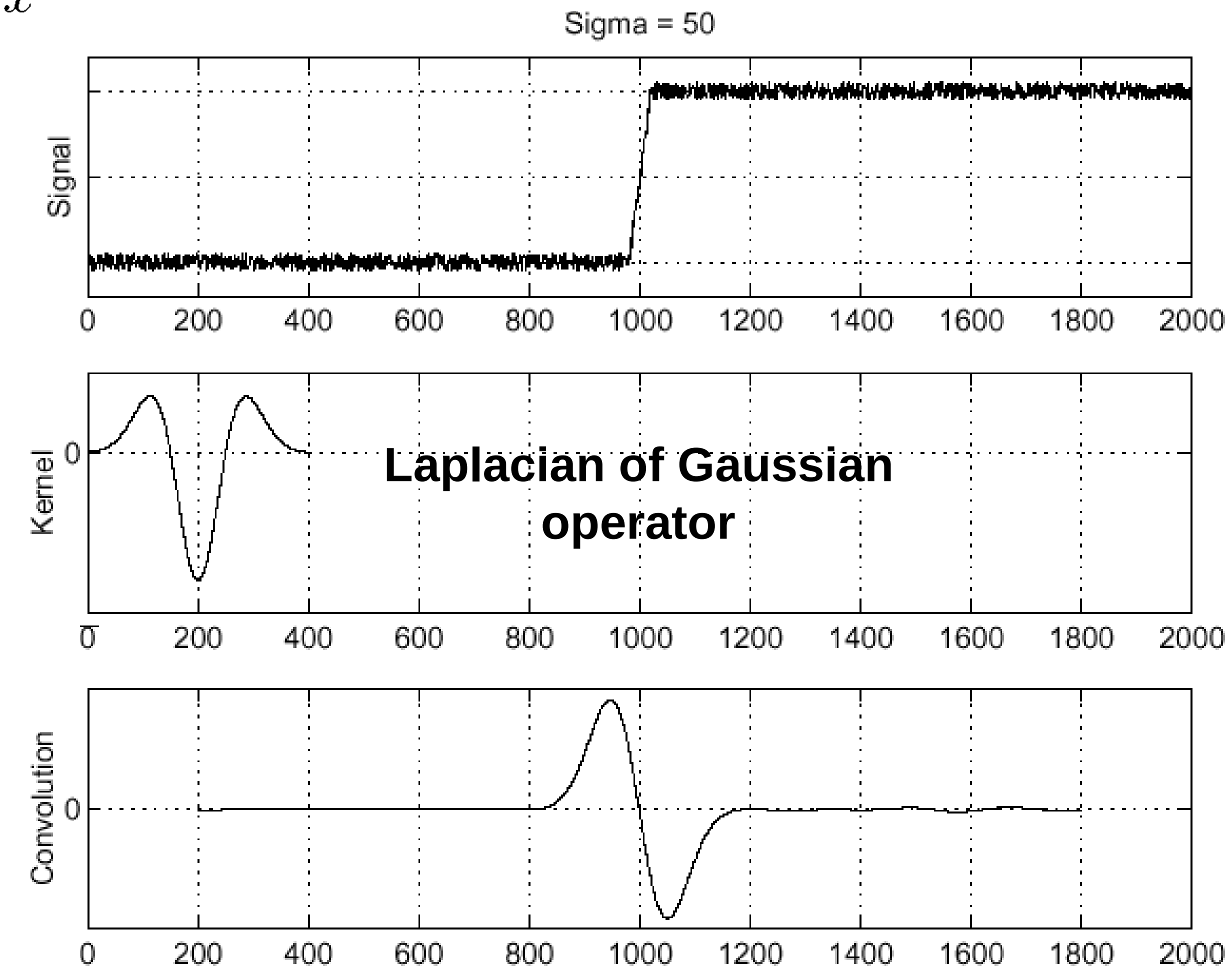
Laplacian of Gaussian

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

f

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

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

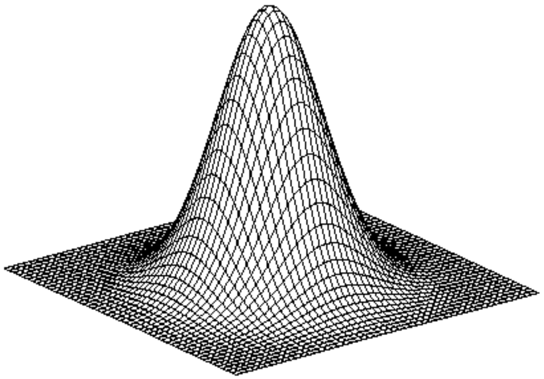


Where is the edge?

Zero-crossings of bottom graph

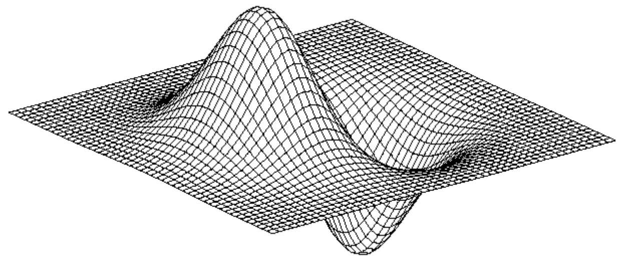
Slide credit: Steve Seitz

2D edge detection filters



Gaussian

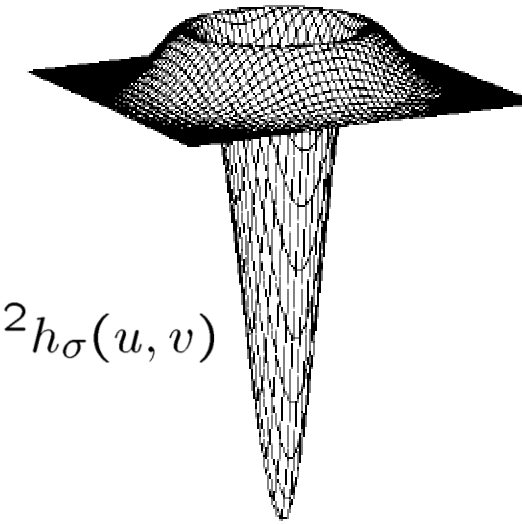
$$h_\sigma(u,v) = \frac{1}{2\pi\sigma^2} e^{-\frac{u^2+v^2}{2\sigma^2}}$$



derivative of Gaussian

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

Laplacian of Gaussian



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

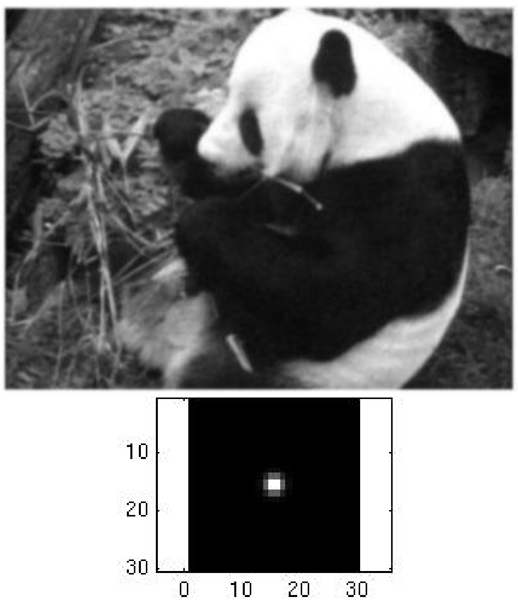
- ∇^2 is the Laplacian operator:

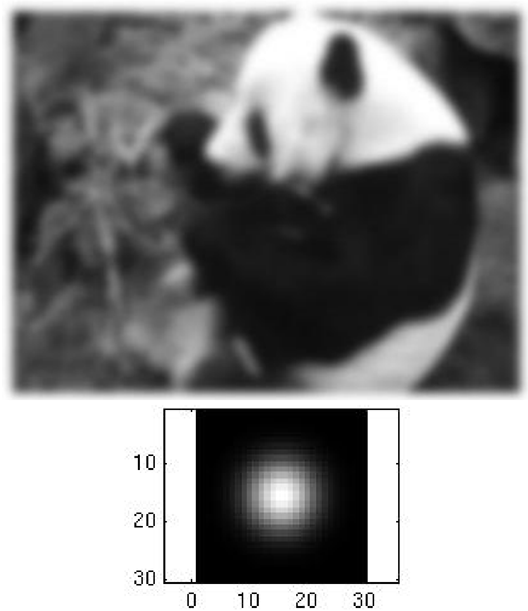
$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

Slide credit: Steve Seitz

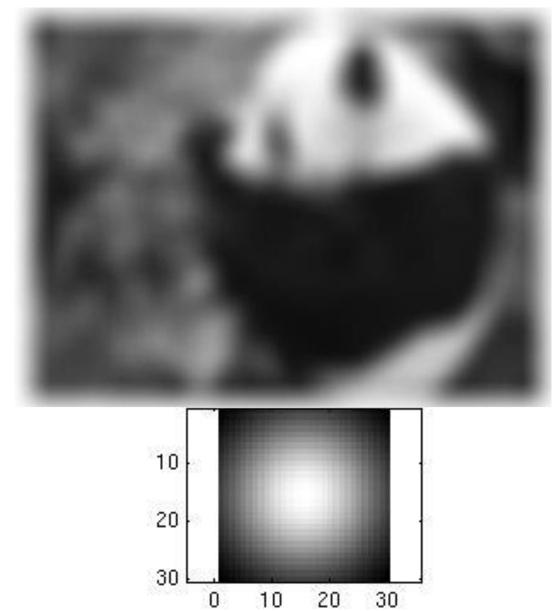
Smoothing with a Gaussian

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





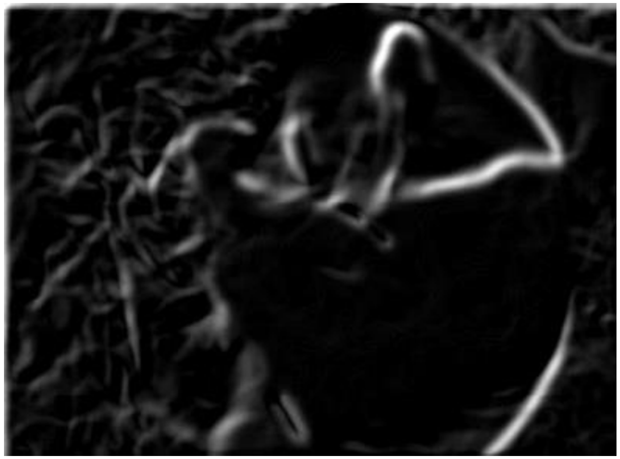
…



Slide credit:
Kristen Grauman

Effect of σ on derivatives

σ = 1 pixel



σ = 3 pixels

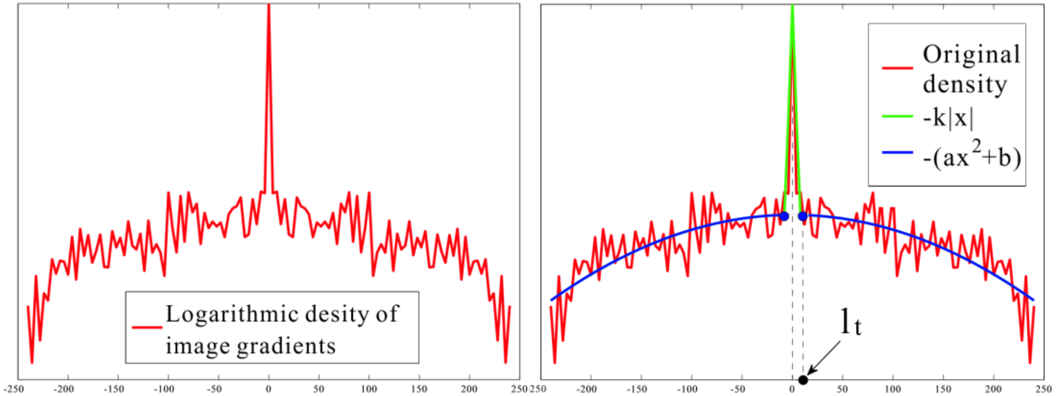
The apparent structures differ depending on Gaussian's scale parameter.

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

Slide credit:
Kristen Grauman

Natural distribution of image gradients

- An effective prior (heavy tailed) in image restoration, e.g., Image deblurring



Blurry input | Ground truth | Without prior | With prior

[Shan et al. 08]

Seam Carving

Seam carving: main idea

[Shai & Avidan, SIGGRAPH 2007]

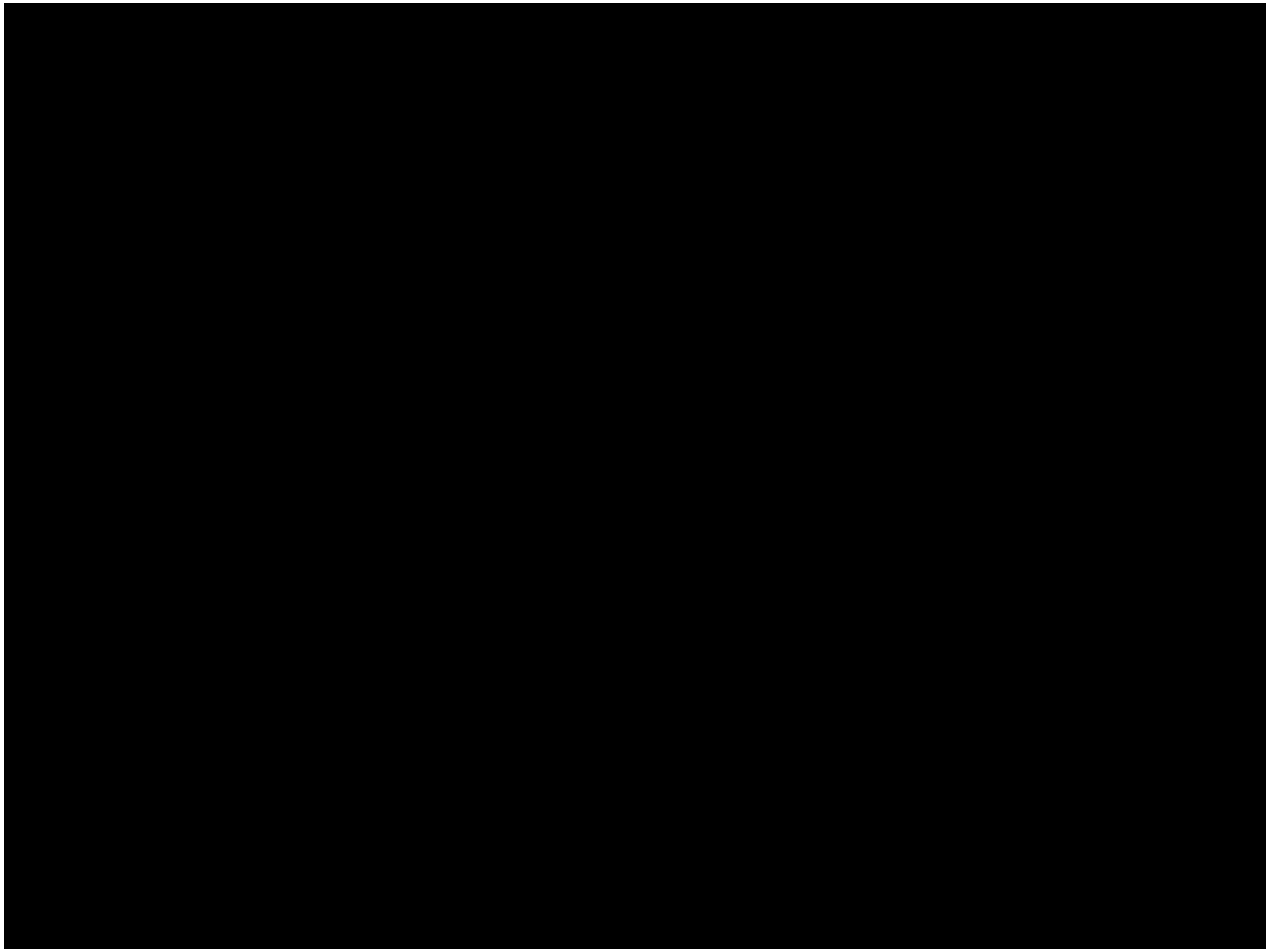
Seam carving: main idea

Content-aware resizing

Traditional resizing

[Shai & Avidan, SIGGRAPH 2007]

Seam carving: main idea



Real image example

Slide credit:
Kristen Grauman

Seam carving: main idea

Content-aware resizing

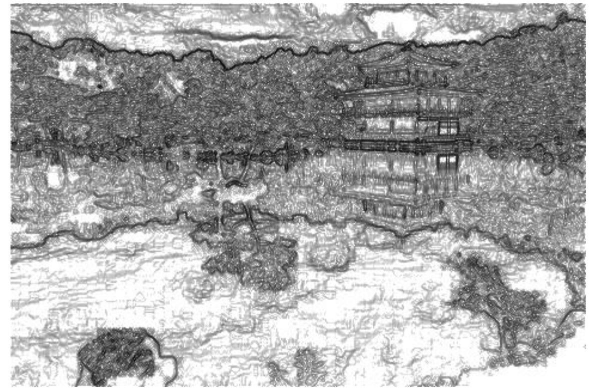
Intuition:

- 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

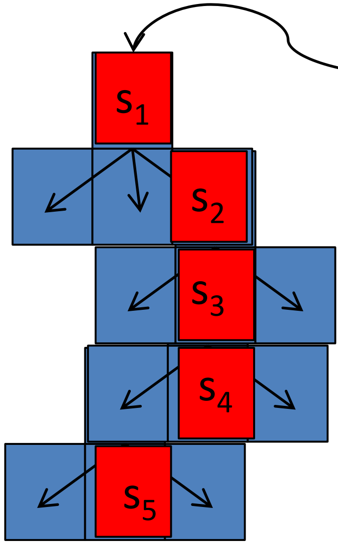
- Want to remove seams w
  - Measure "energ
- Choose seam based on **m**



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

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:

$$\mathbf{s}^* = \min_{\mathbf{s}} Cost(\mathbf{s})$$

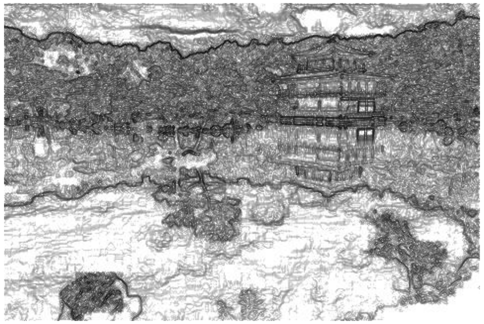
Compute it efficiently with dynamic programming.

Slide credit:
Kristen Grauman

How to identify the minimum cost seam?

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

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



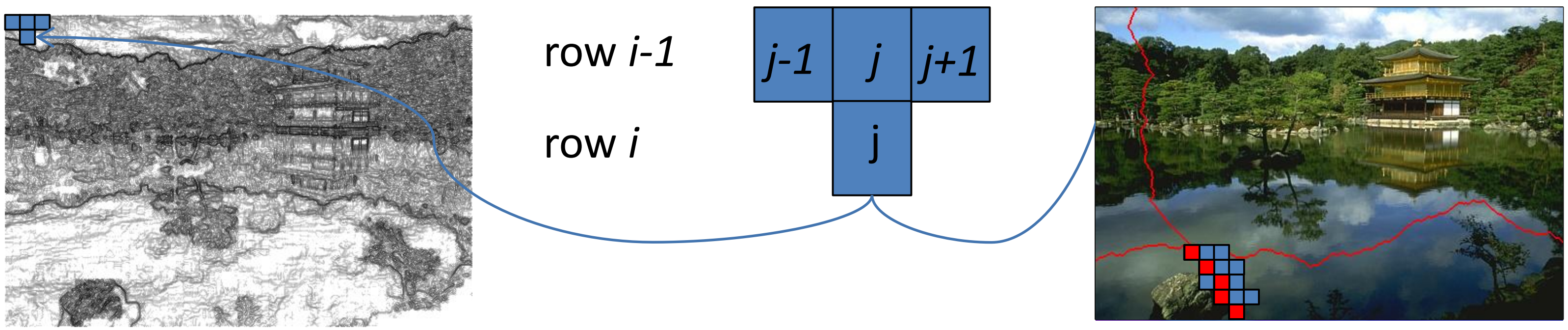
Energy matrix
(gradient magnitude)

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)

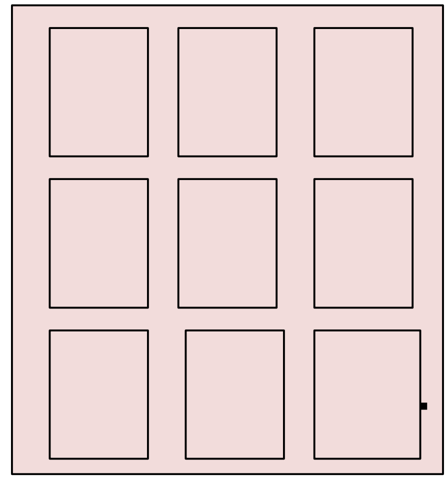
- Then, min value in last row of **M** indicates end of the minimal connected vertical seam.
- Backtrack up from there, selecting min of 3 above in **M**.

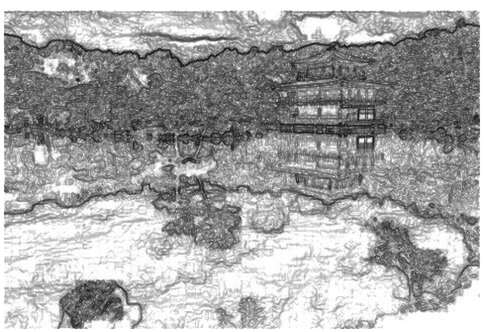
Slide credit: Kristen Grauman

Example

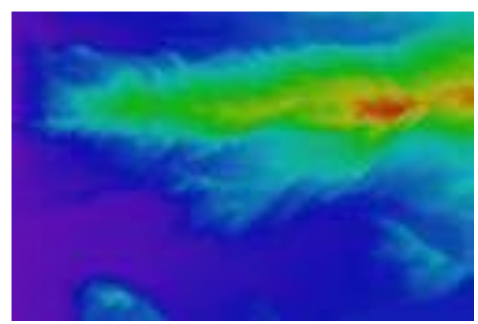
$$\mathbf{M}(i,j) = Energy(i,j) + \min\big(\mathbf{M}(i-1,j-1), \mathbf{M}(i-1,j), \mathbf{M}(i-1,j+1)\big)$$

1	3	0
2	8	9
5	2	6





Energy matrix (gradient magnitude)



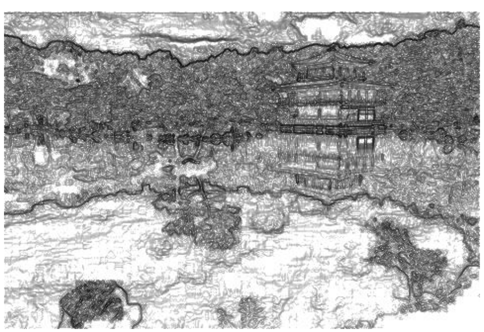
M matrix (for vertical seams)

Slide credit: Kristen Grauman

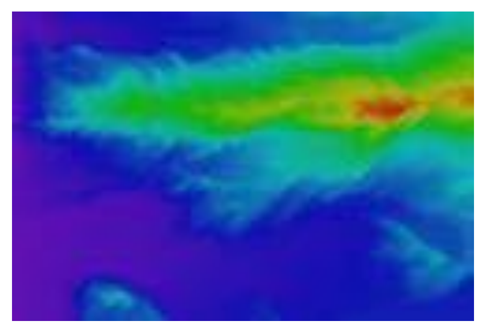
Example

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

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



Energy matrix
(gradient magnitude)



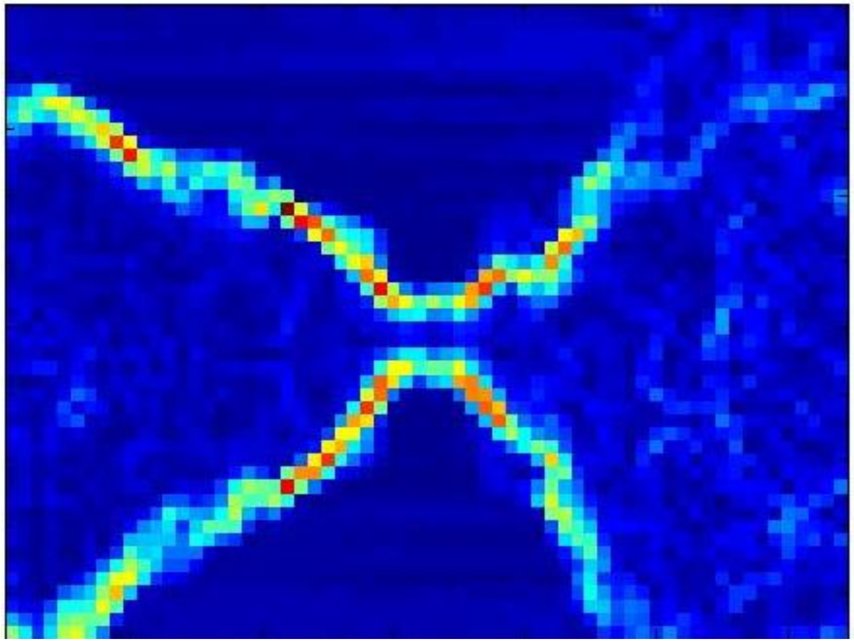
M matrix
(for vertical seams)

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

Seam Carving

Seam carving

From Wikipedia, the free encyclopedia

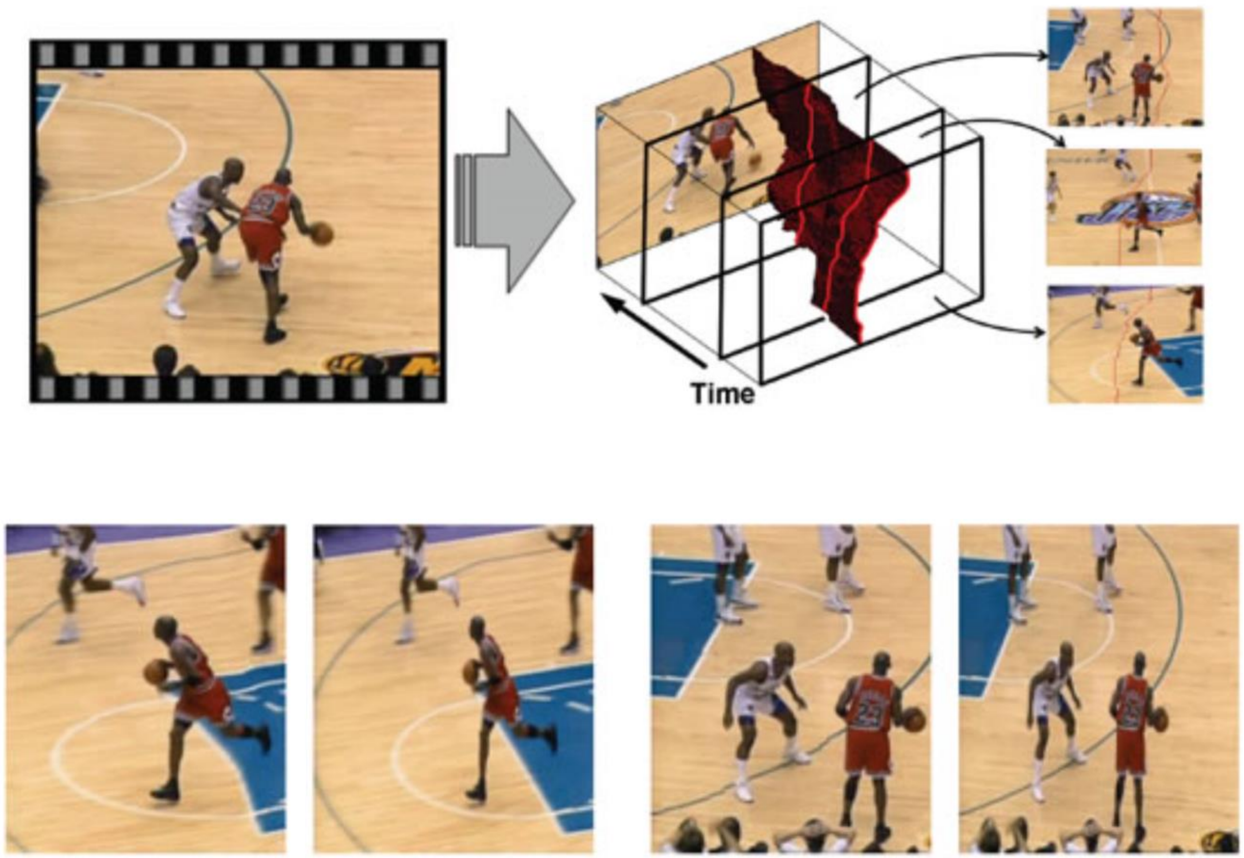
Seam carving (or **liquid rescaling**) is an algorithm for content-aware image resizing, developed by Shai Avidan, of Mitsubishi Electric Research Laboratories (MERL), and Ariel Shamir, of the Interdisciplinary Center and MERL. It functions by establishing a number of *seams* (paths of least importance) in an image and automatically removes seams to reduce image size or inserts seams to extend it. Seam carving also allows manually defining areas in which pixels may not be modified, and features the ability to remove whole objects from photographs.

Original image to be made narrower

The purpose of the algorithm is image retargeting, which is the problem of displaying images without distortion on media of various sizes (cell phones, projection screens) using document standards, like HTML, that already support dynamic changes in page layout and text but not images.[1]

Image Retargeting was invented by Vidya Setlur, Saeko Takage, Ramesh Raskar, Michael Gleicher and Bruce Gooch in 2005.[2] The work by Setlur et al. won the 10-year impact award in 2015.

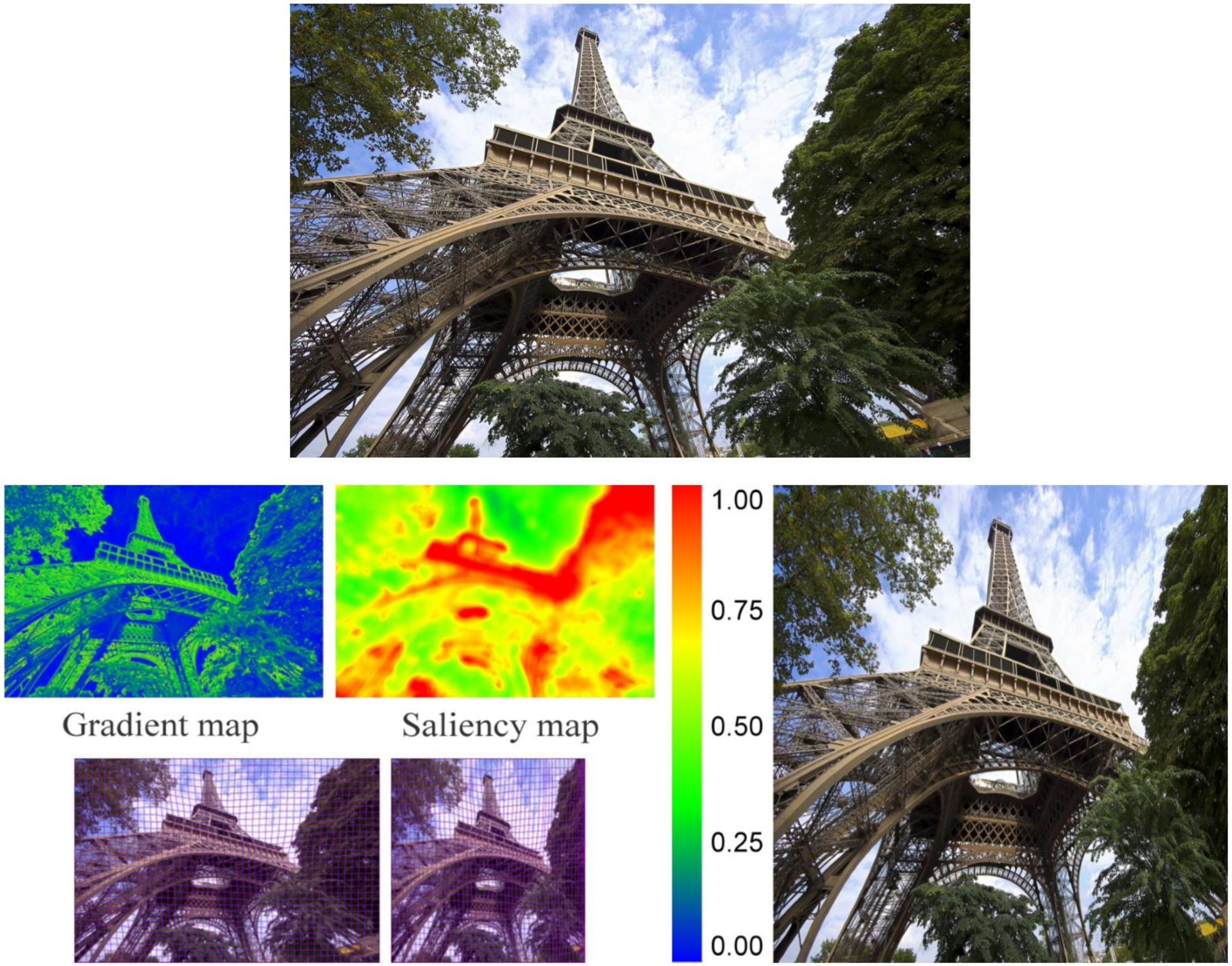
Extensions of Seam Carving



seams scale seams scale

[Rubinstein et al. 08]

Image Deformation



[Wang et al. 08]

Edge Detection

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

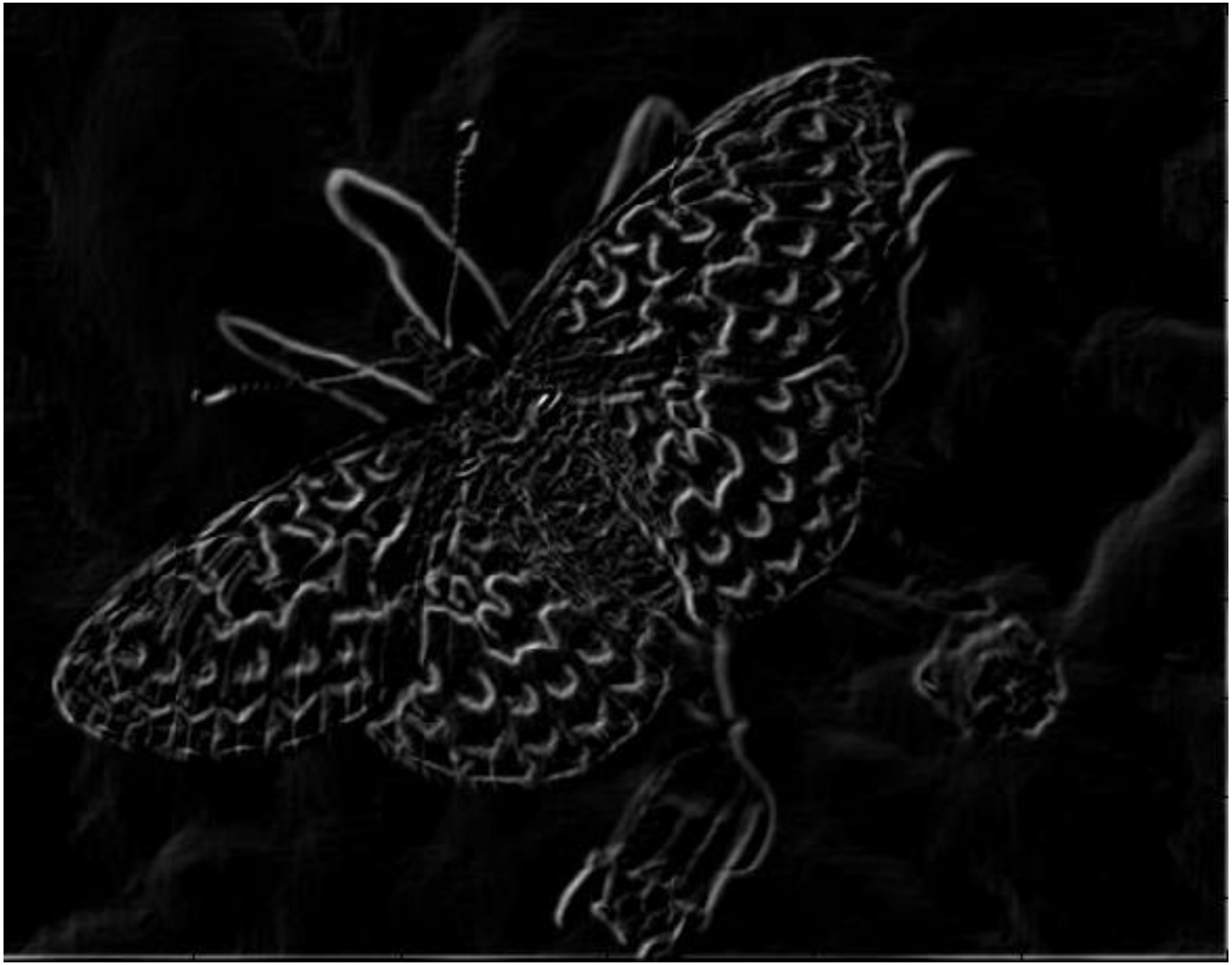
    - Threshold, Thin

Slide Credit: Kristen Grauman

Original image

Slide Credit: Kristen Grauman

Gradient magnitude image



Slide Credit: Kristen Grauman

Thresholding gradient with a lower threshold

Slide Credit: Kristen Grauman

Thresholding gradient with a higher threshold

Slide Credit: Kristen Grauman

Canny Edge Detector

- Filter image with derivative of Gaussian and find magnitude and orientation of gradient
- Non-maximum suppression
  - Thin wide “ridges” down to single pixel with
- 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

Slide Credit: Kristen Grauman

The Canny edge detector

original image (Lena)

Slide credit: Steve Seitz

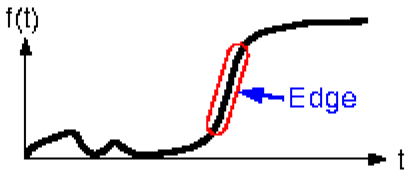
The Canny edge detector

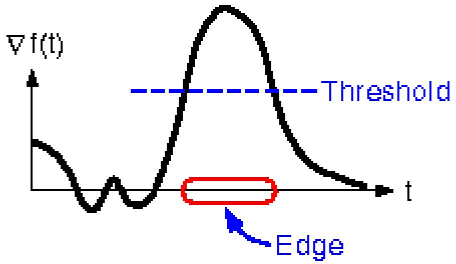
norm of the gradient

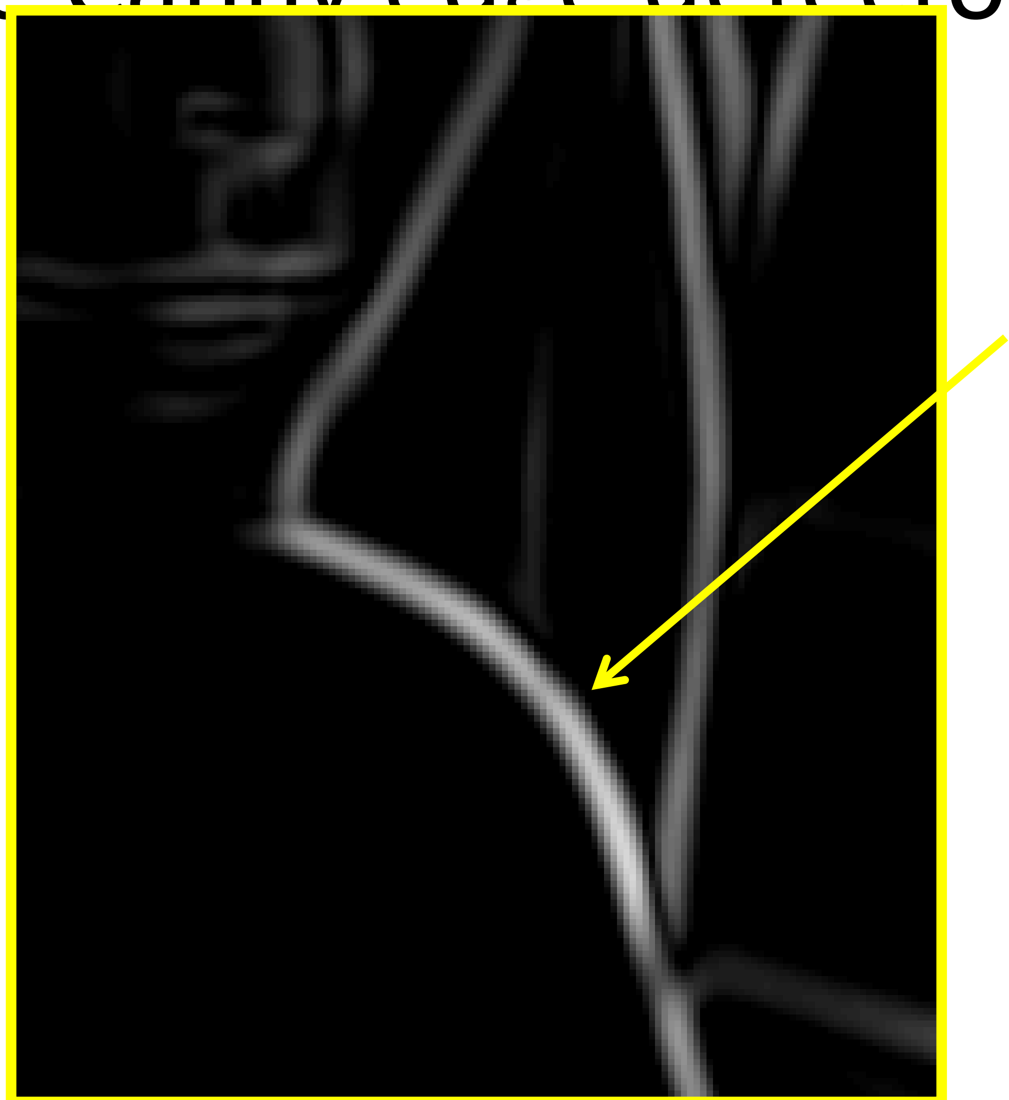
The Canny edge detector

thresholding

The Canny edge detector

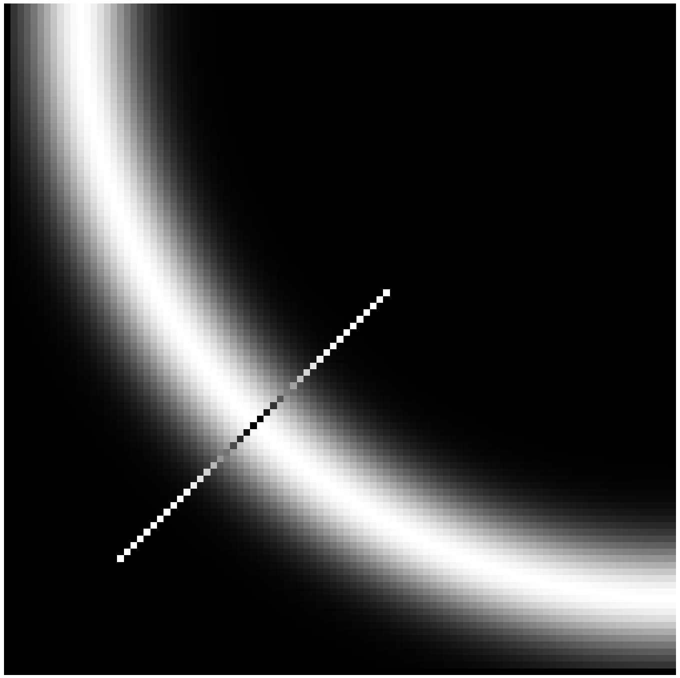


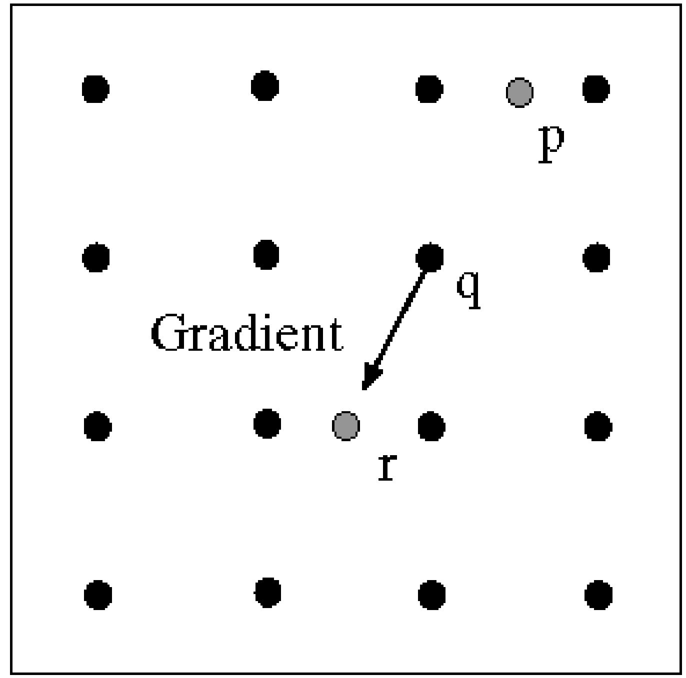




How to turn these thick regions of the gradient into curves?

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)

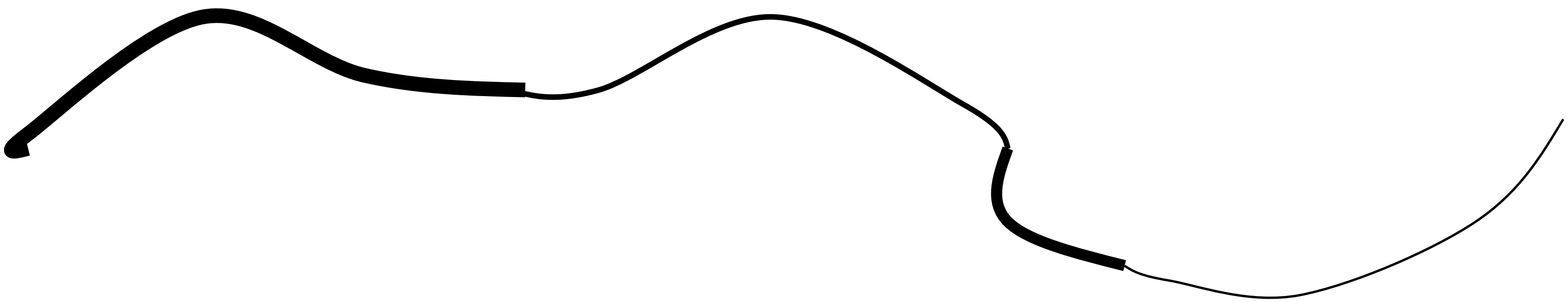
Hysteresis thresholding

- 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

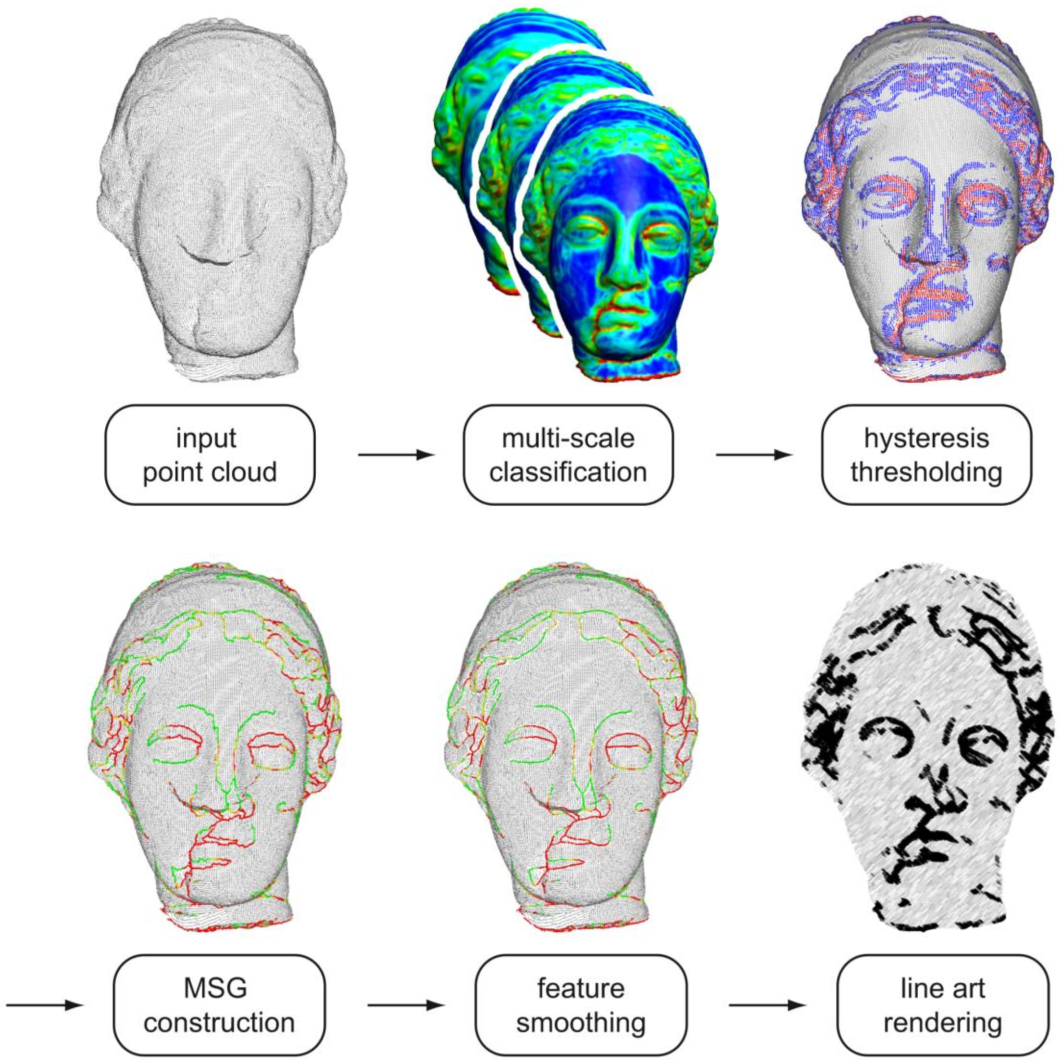
Final Canny Edges

Credit: James Hays

Recap: Canny Edge Detector

- A typical example of successful hand-crafted feature design
  - Derivatives as a starting point
  - Non-maximum suppressing to handle fuzzy edges
  - Linking and thresholding (hysteresis)
  - Each (simple) heuristic addresses one important issue
  - Has been applied in other settings
  - How to improve Canny Edge Detector?

Edge detection on 3D shapes



[Pauly et al. 03]

How to Evaluate is Very Important in Computer Vision Research

Labeled data

Evaluation Protocol

Low-level edges vs. perceived contours

Background

Texture

Shadows

Slide credit: Kristen Graumann

Low-level edges vs. perceived contours

image | **human segmentation** | **gradient magnitude**

- Berkeley segmentation database:
  http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/segbench/

Source: L. Lazebnik

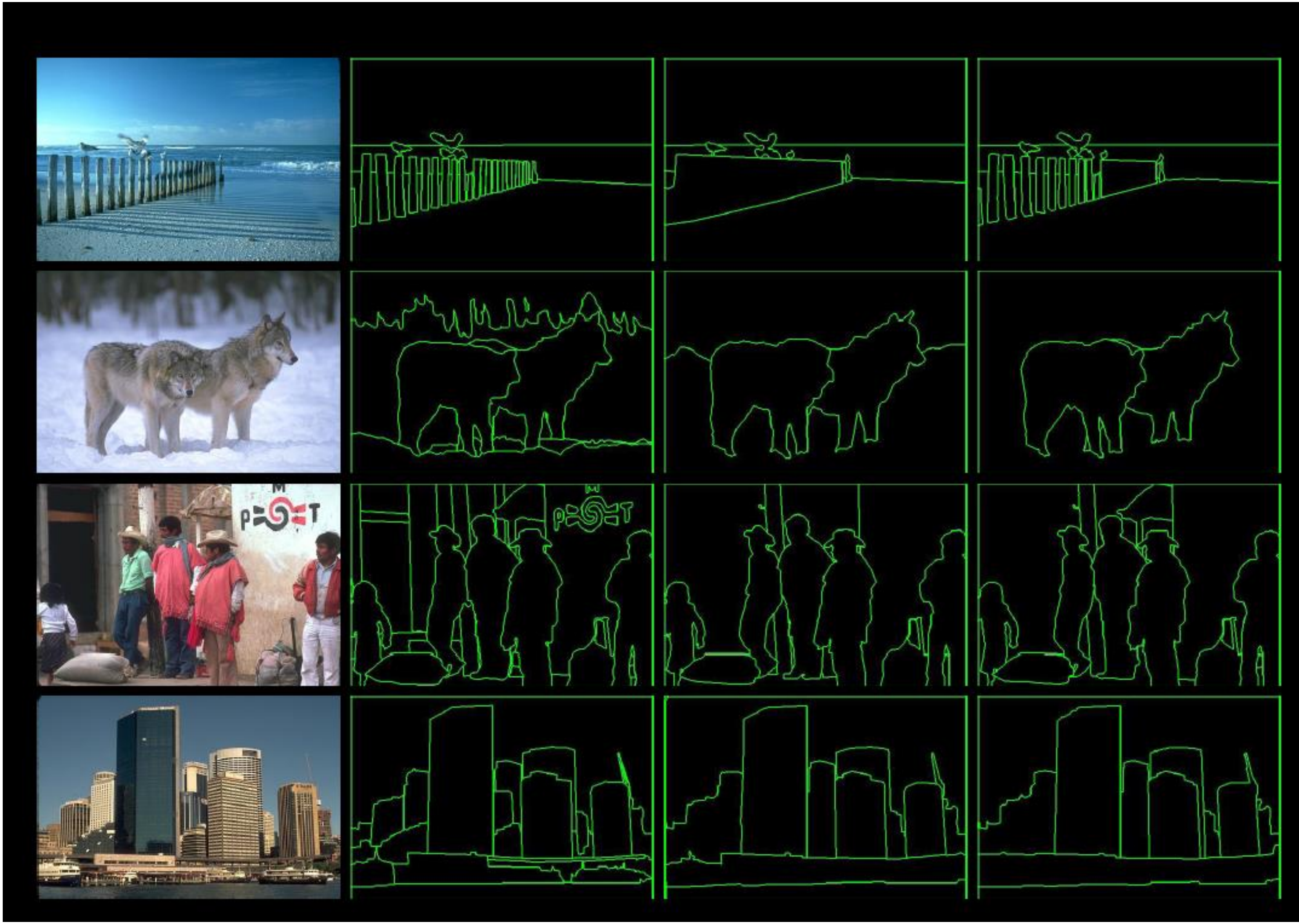
Protocol

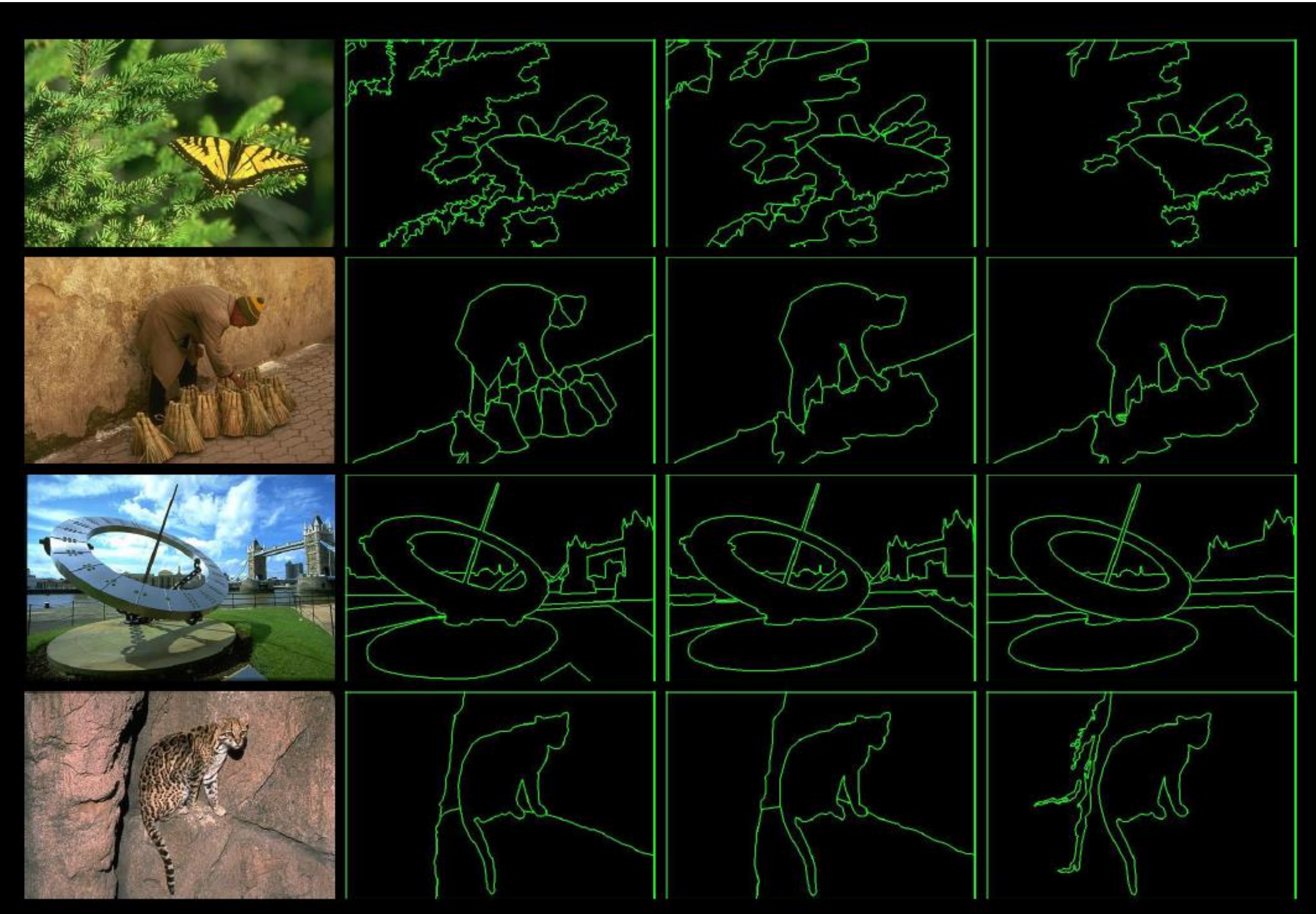
You will be presented a photographic image. Divide the image into some number of segments, where the segments represent "things" or "parts of things" in the scene. The number of segments is up to you, as it depends on the image. Something between 2 and 30 is likely to be appropriate. It is important that all of the segments have approximately equal importance.

- Custom segmentation tool
- Subjects obtained from work-study program (UC Berkeley undergraduates)

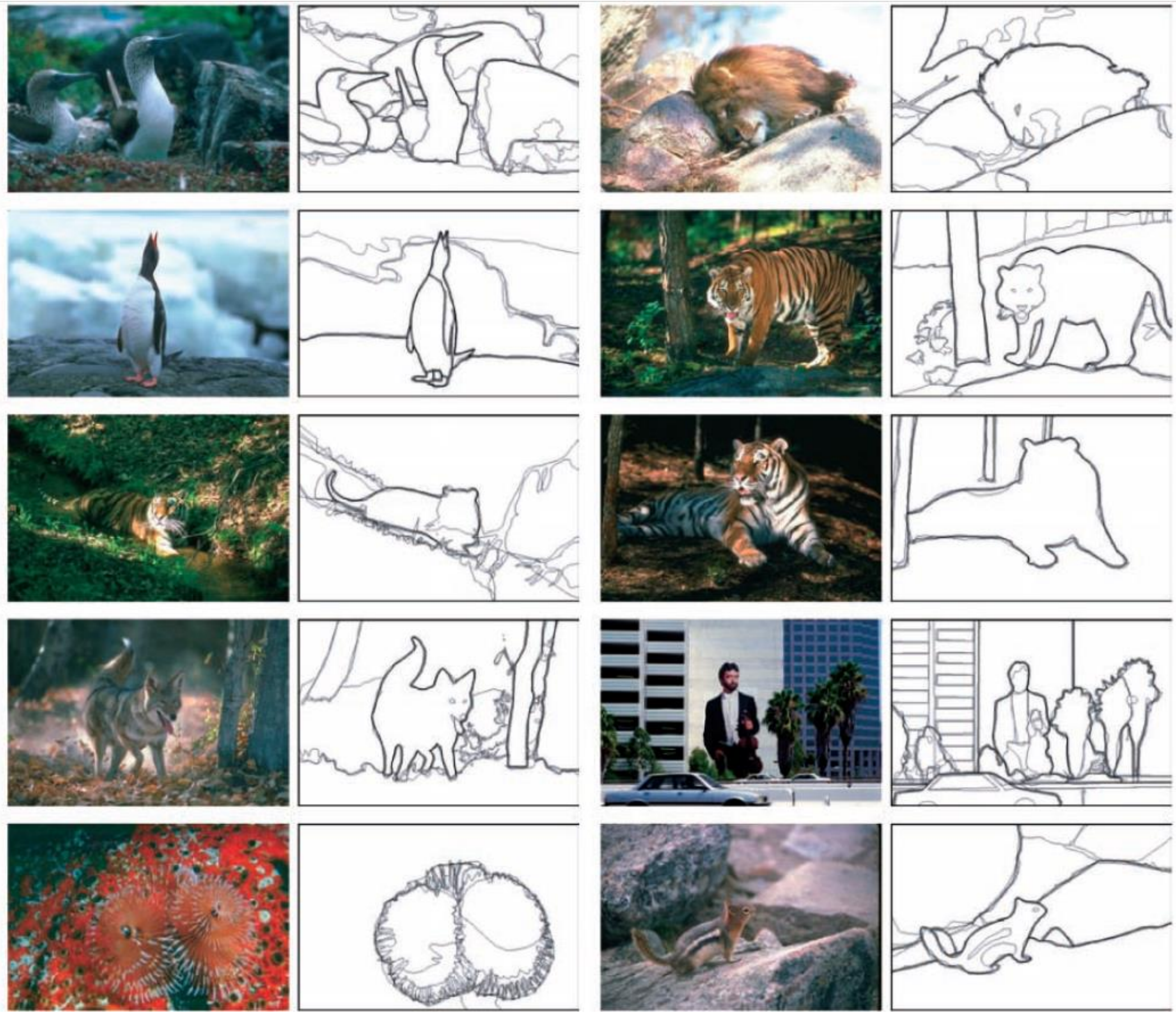
Berkeley Segmentation Data Set
David Martin, Charless Fowlkes, Doron Tal, Jitendra Malik

Credit: David Martin





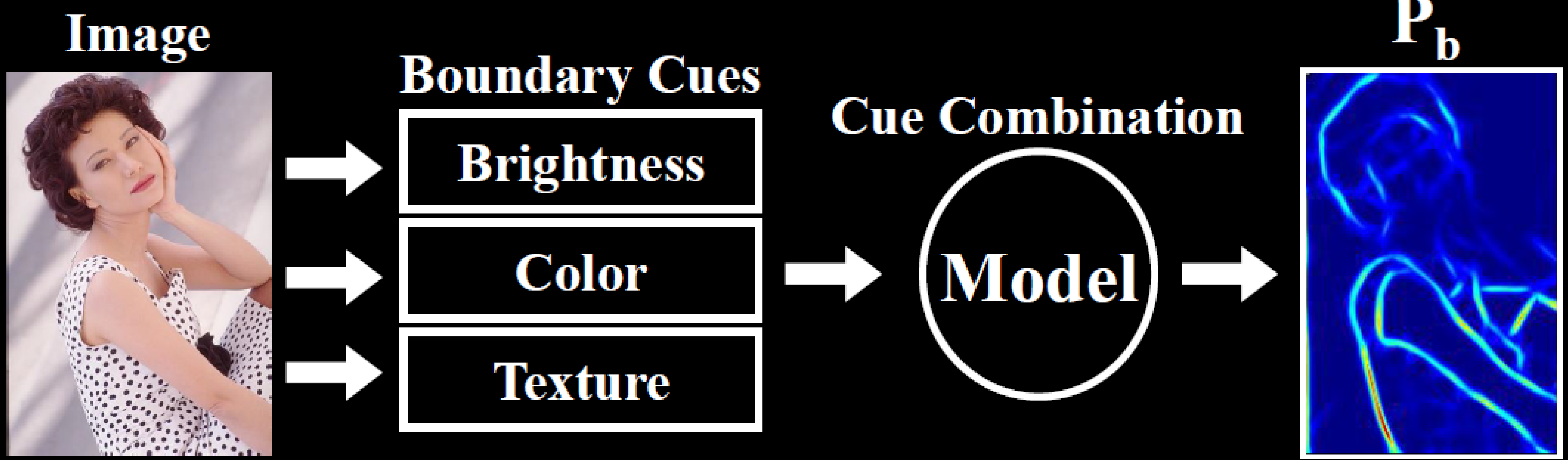
Learn from humans which combination of features is most indicative of a “good” contour?



[D. Martin et al. PAMI 2004]

Human-marked segment boundaries

Dataflow

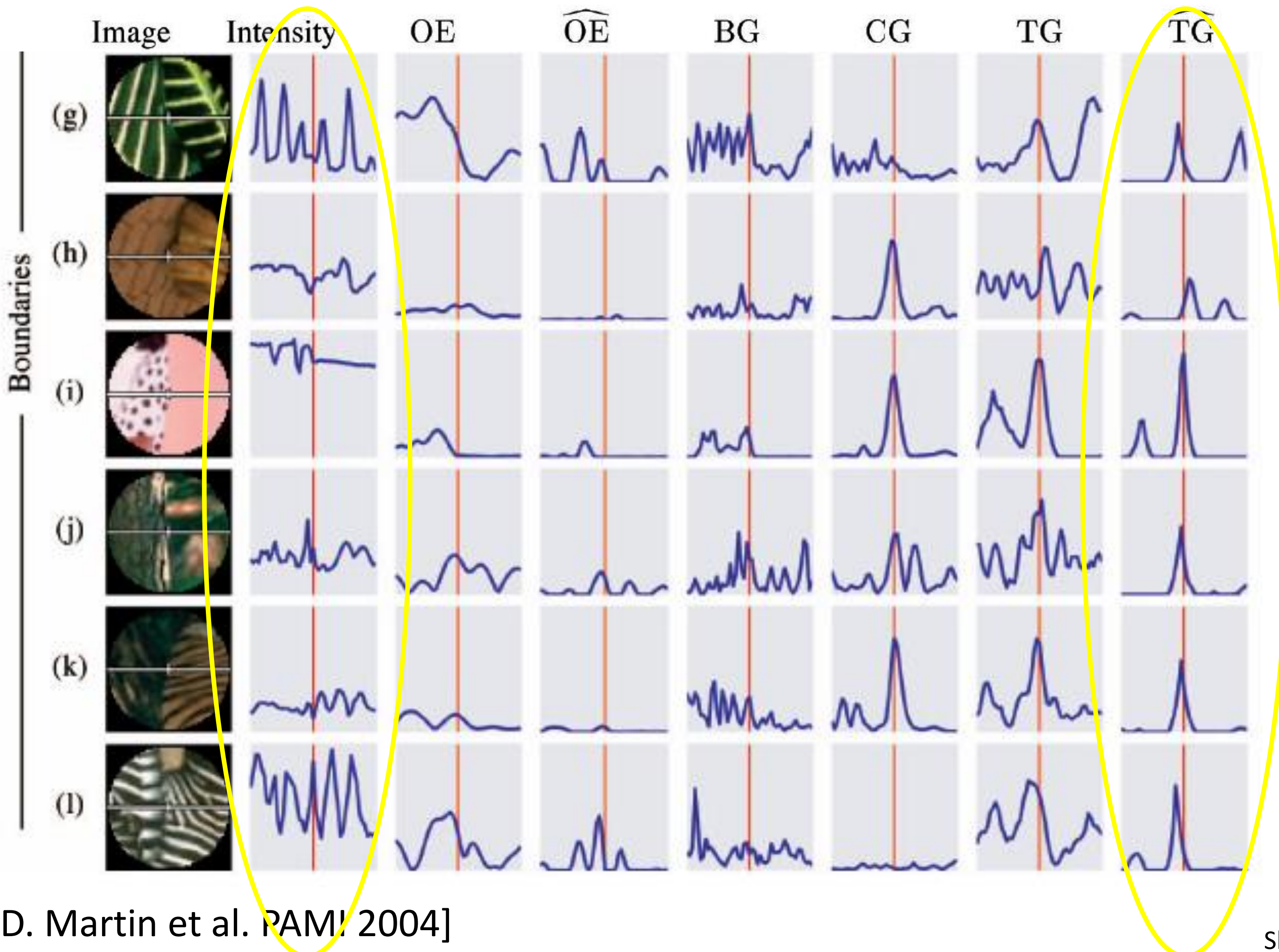


Challenges: texture cue, cue combination

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

What features are responsible for perceived edges?

What features are responsible for perceived edges?

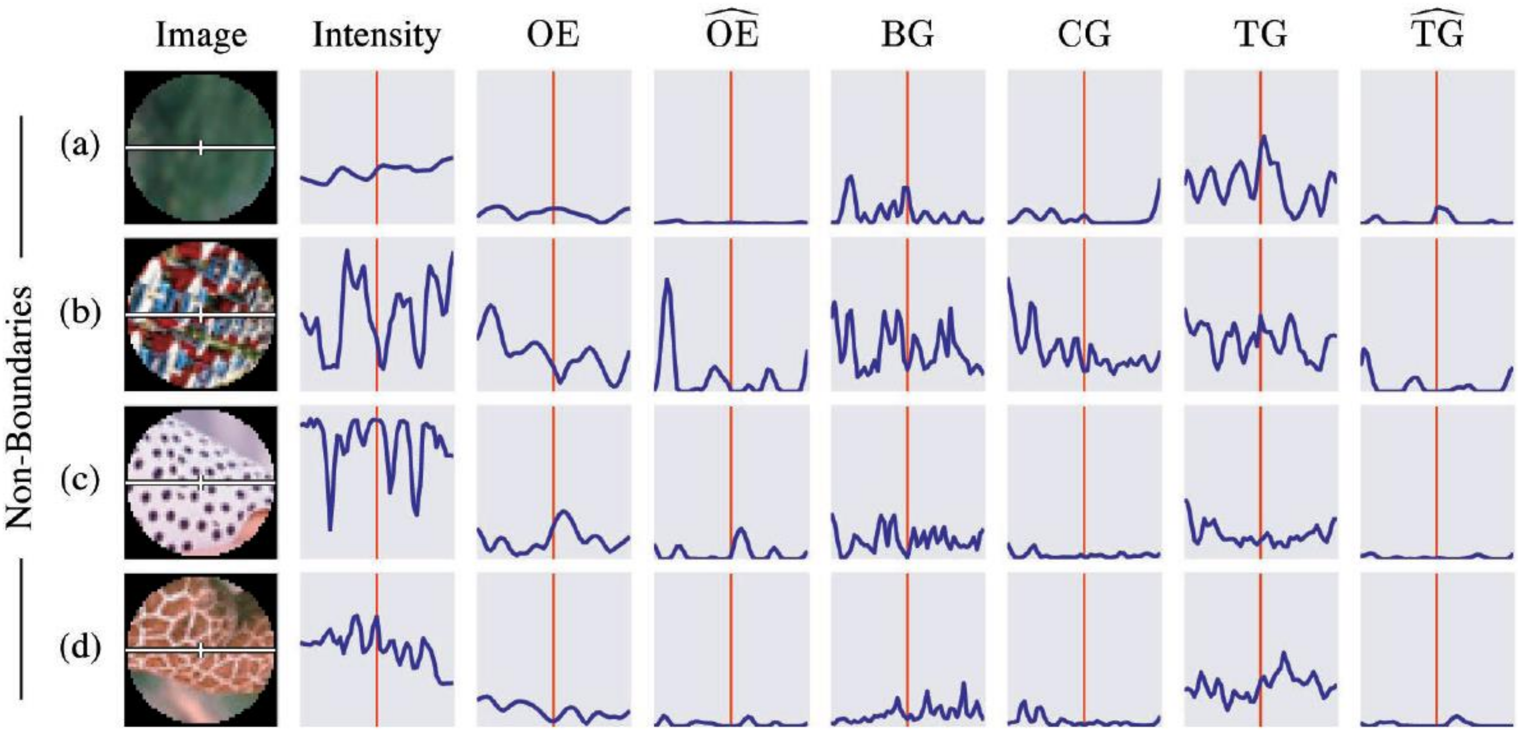


Feature profiles (oriented energy, brightness, color, and texture gradients) along the patch's horizontal diameter

[D. Martin et al. PAMI 2004]

Slide Credit: Kristen Grauman

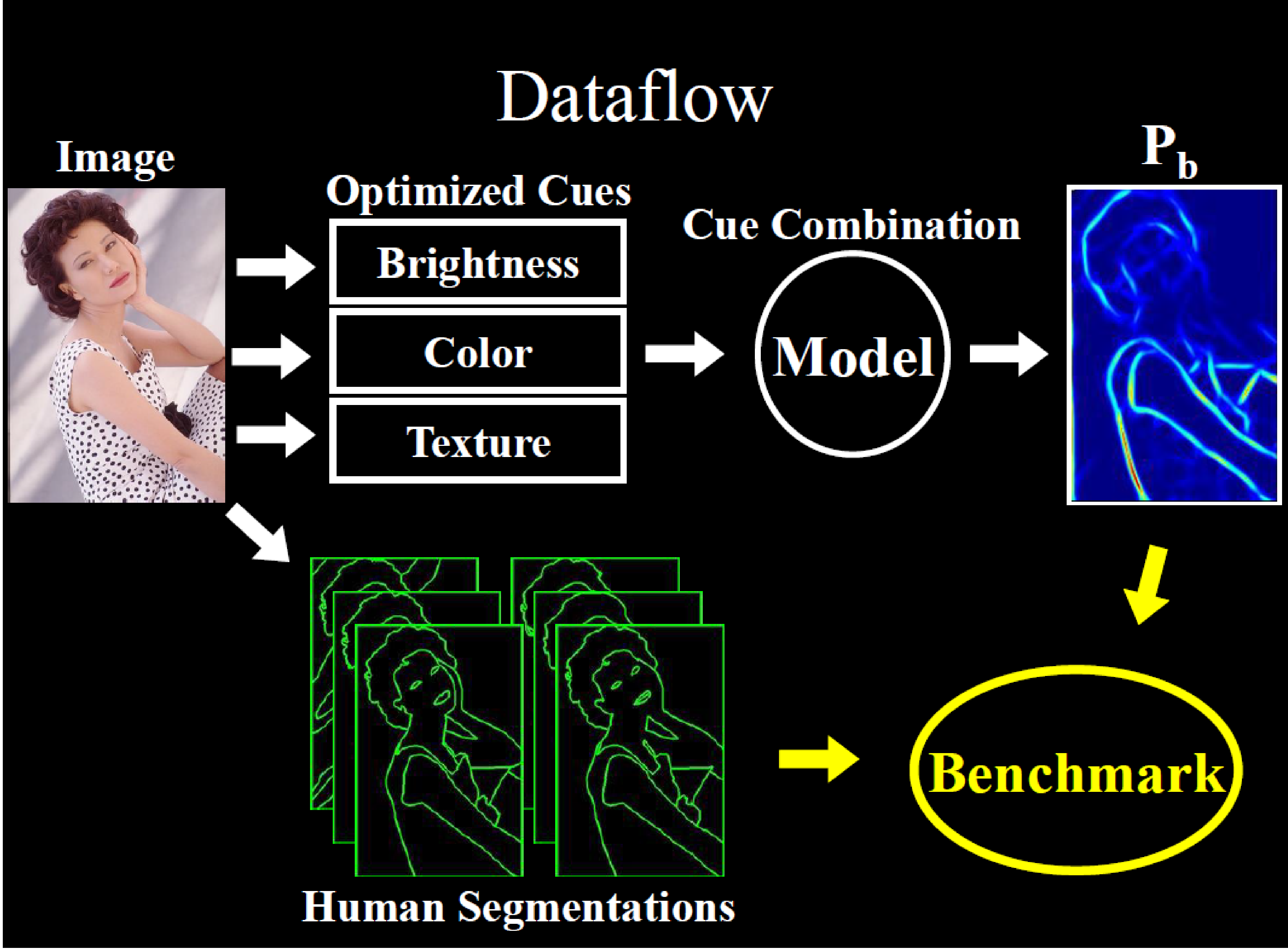
What features are responsible for perceived edges?



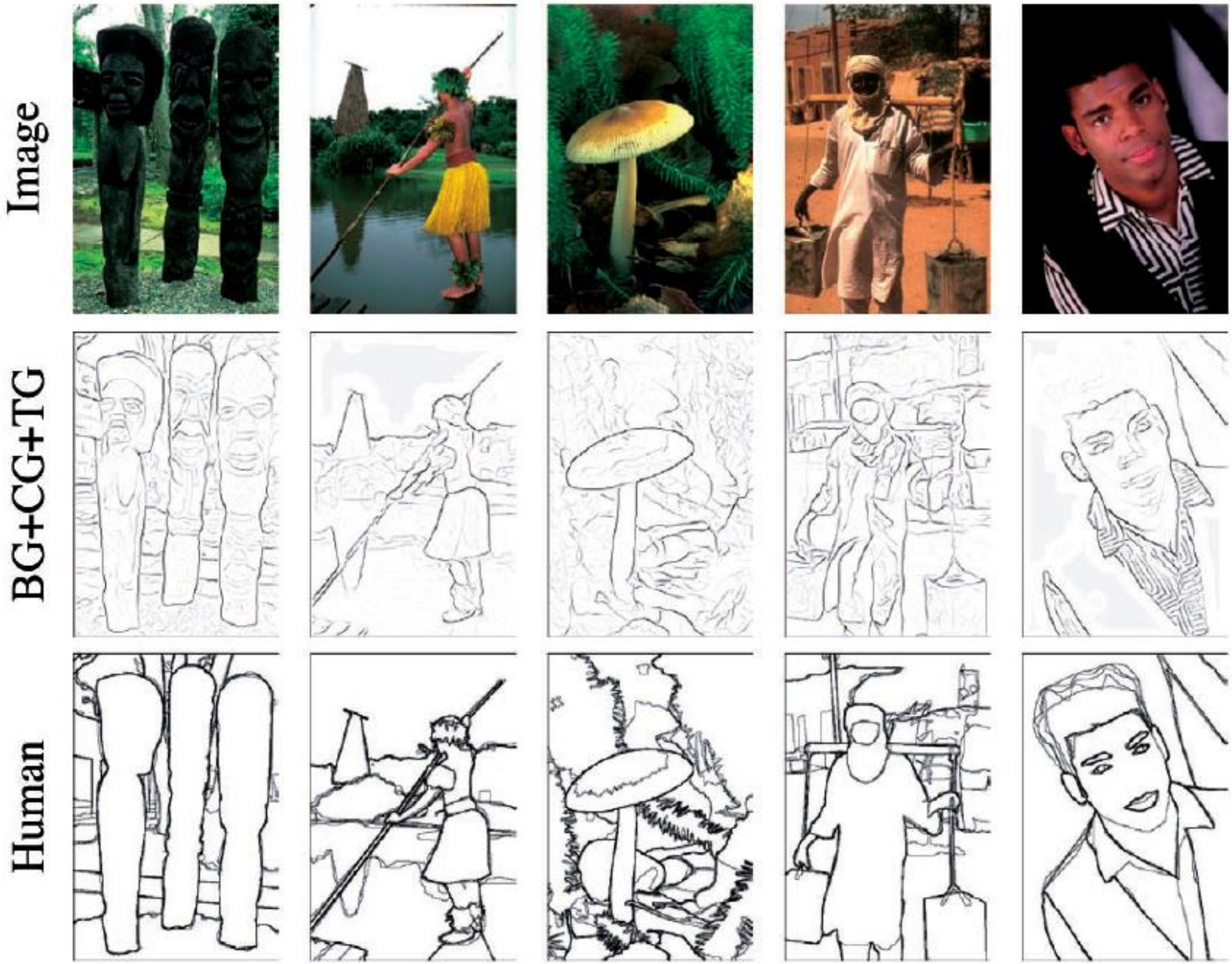
Feature profiles (oriented energy, brightness, color, and texture gradients) along the patch's horizontal diameter

[D. Martin et al. PAMI 2004]

Kristen Grauman, UT-Austin



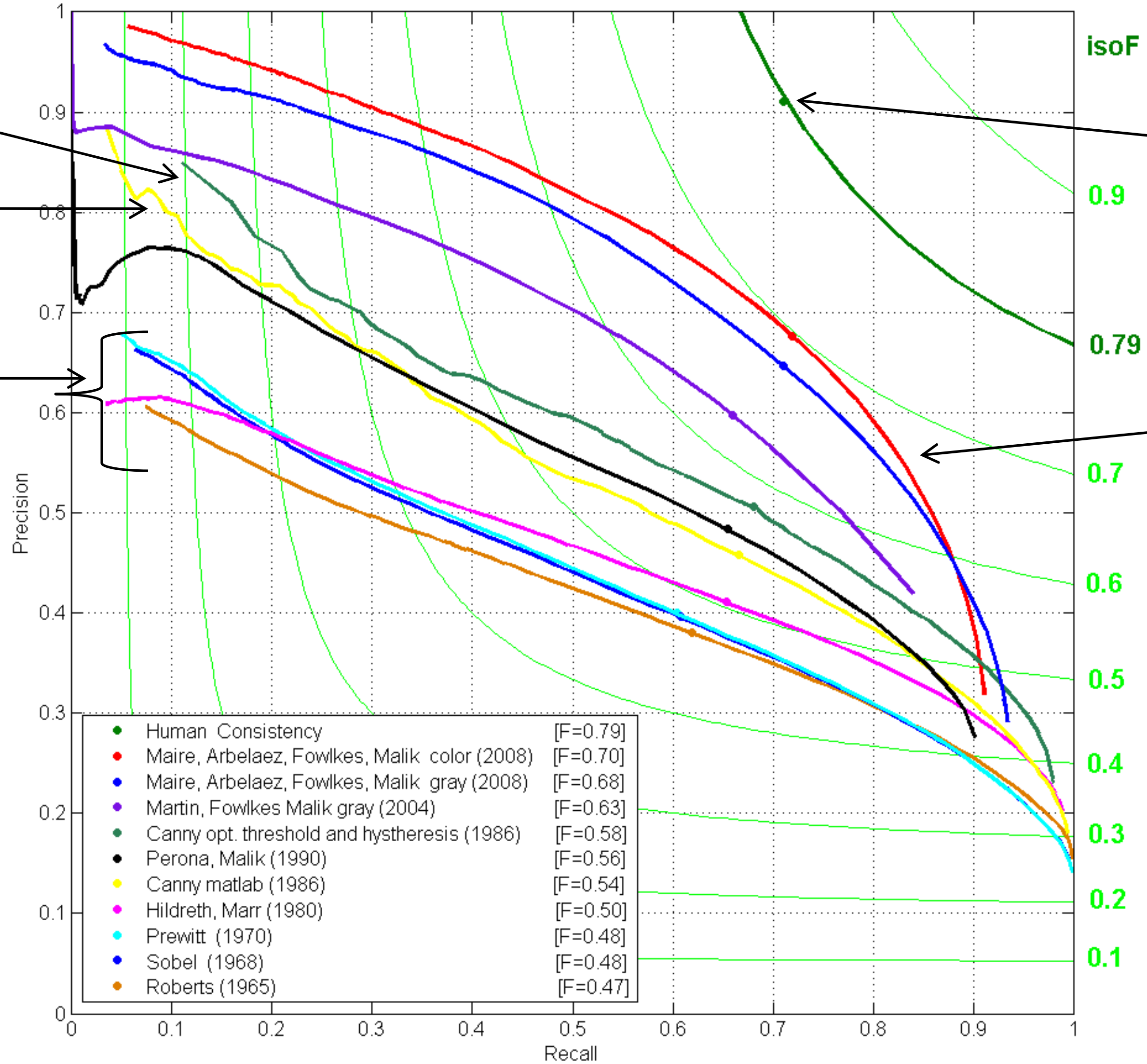
Credit: David Martin



[D. Martin et al. PAMI 2004]

Kristen Grauman, UT-Austin

Contour Detection



Canny+opt thresholds

Canny

Prewitt, Sobel, Roberts

Human agreemen t

Learned with combined features

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

Template Matching

Another Application: Template Matching

- A building block of neural networks is called a filter
  - Map raw pixels to an intermediate (feature) representation
  - Neural networks utilize filters in a hierarchical manner

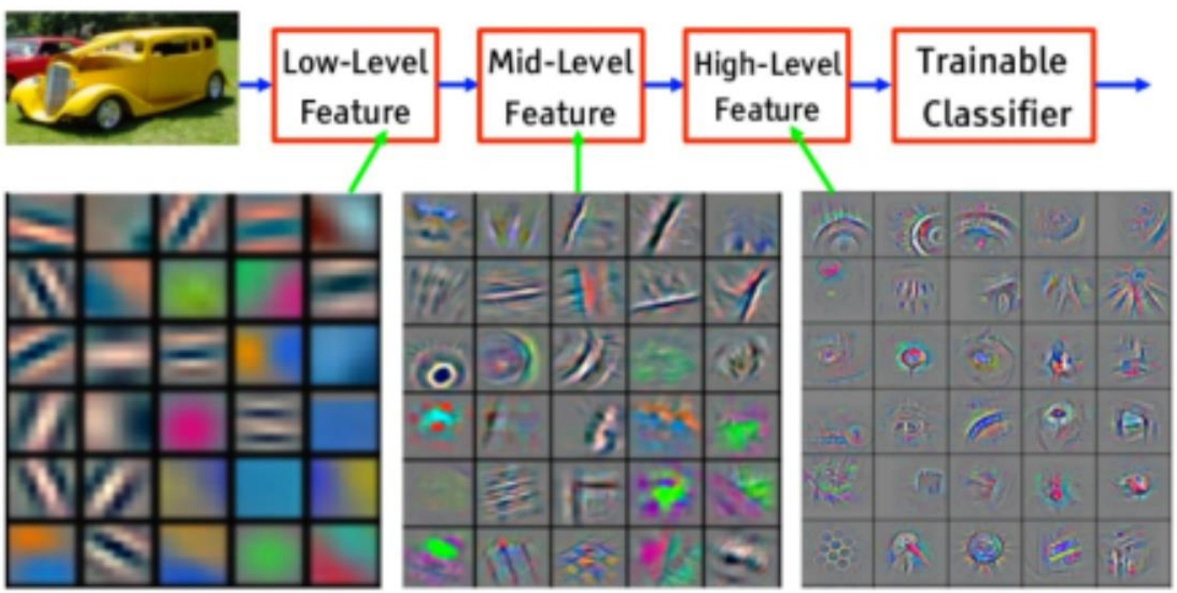
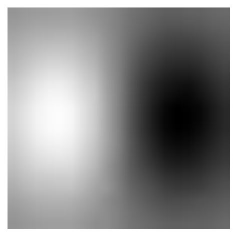
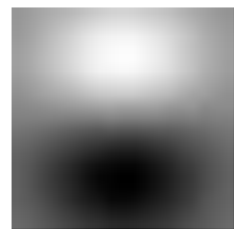


Image credit: Yann Lecun

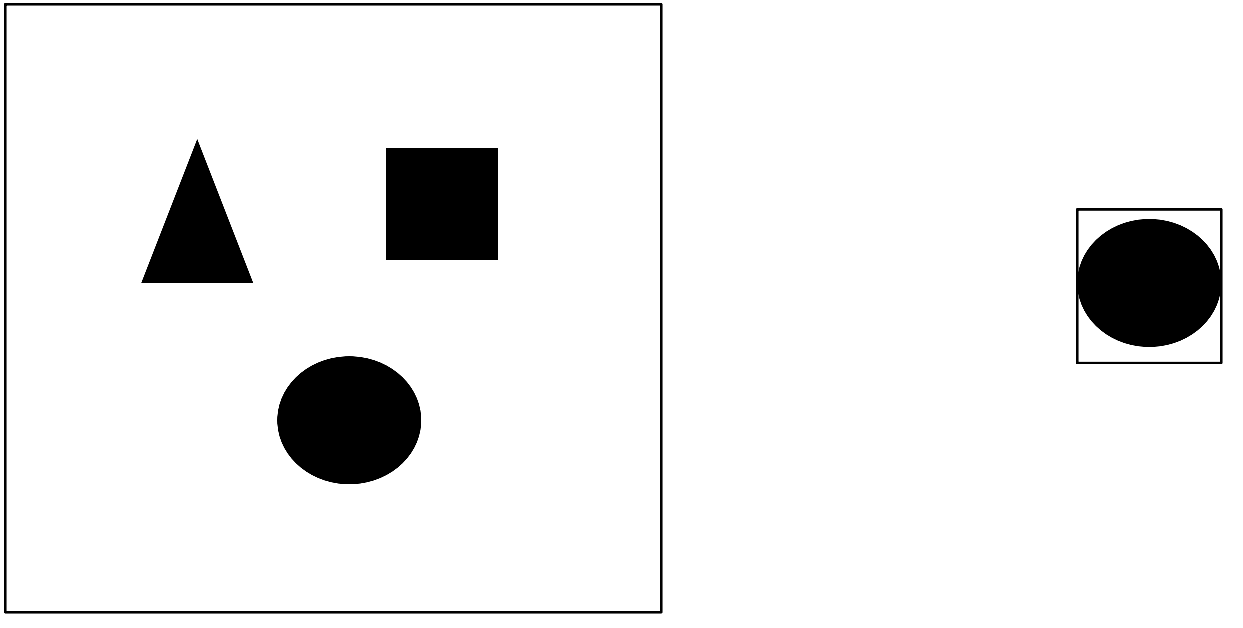
Template matching

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

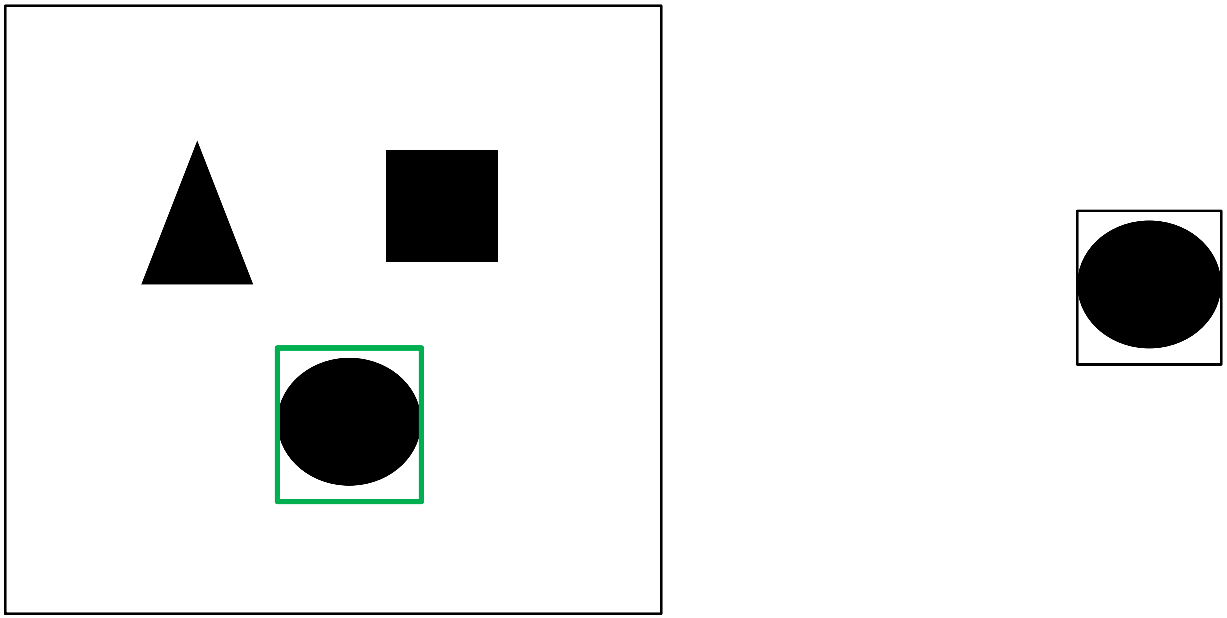
- Use normalized cross-correlation score to find a given pattern (template) in the image
- Normalization needed to control for relative brightnesses

Template matching



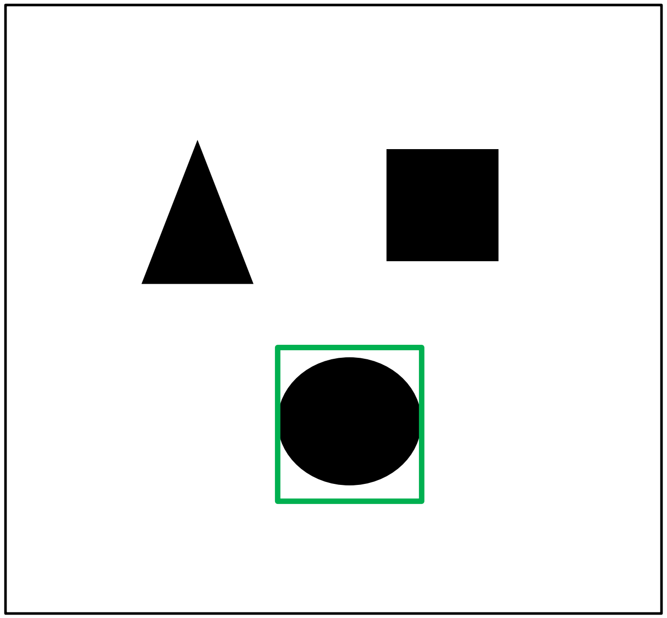
A toy example

Template matching

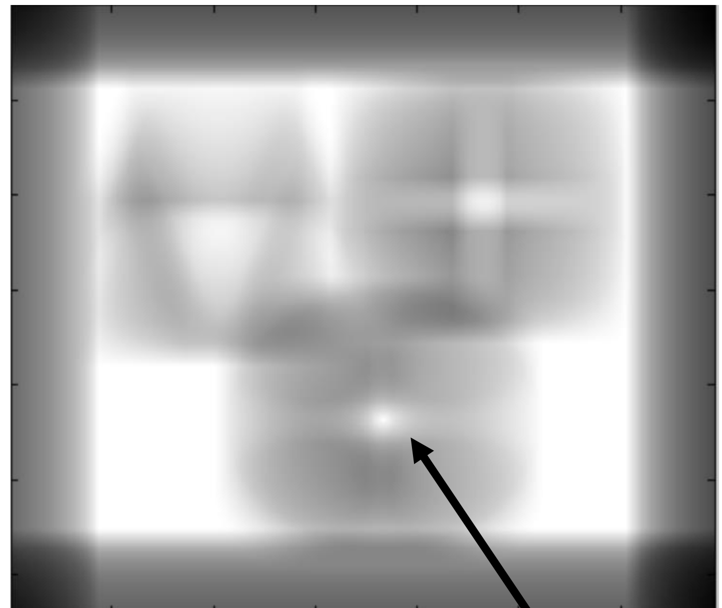


Detected template

Template matching



Detected template



Correlation map

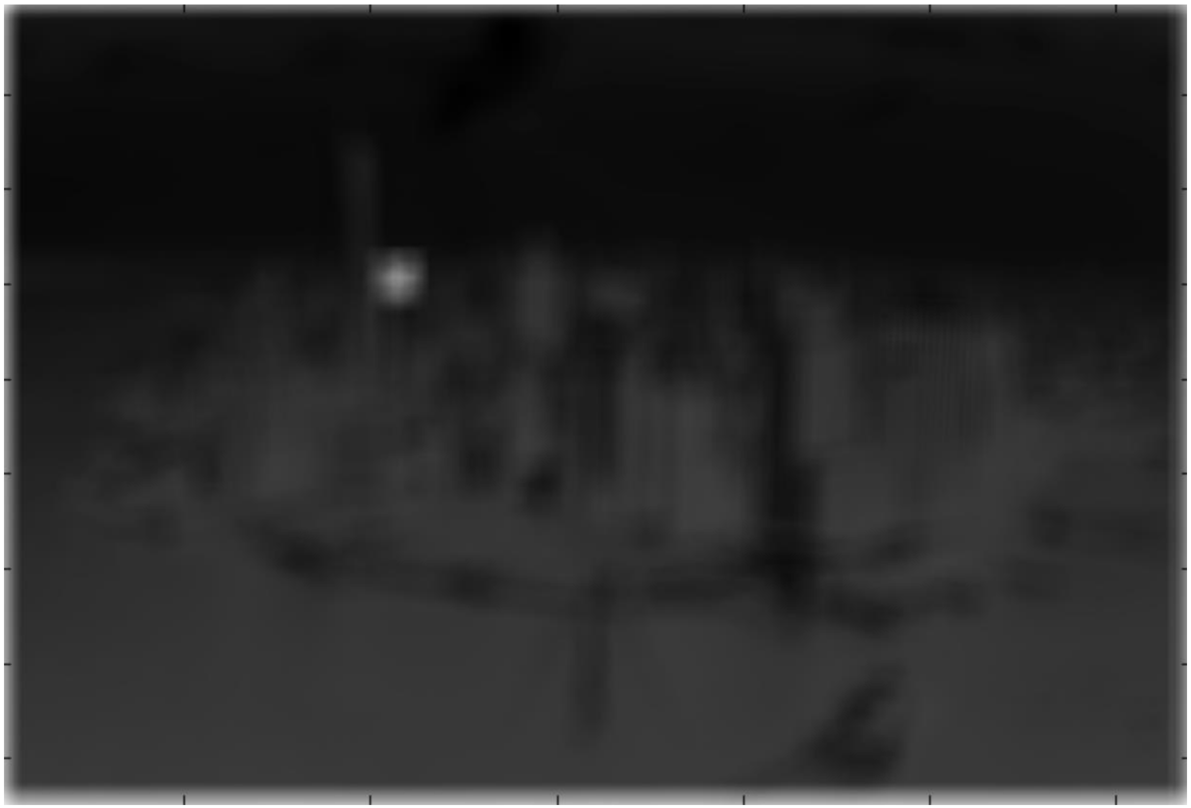
Peak

Where's Lion Messi

Template

Scene

Where’s Lion Messi



Correlation Map

Template

Try multiple scales

Where’s Lion Messi

Template

Scene

Template matching

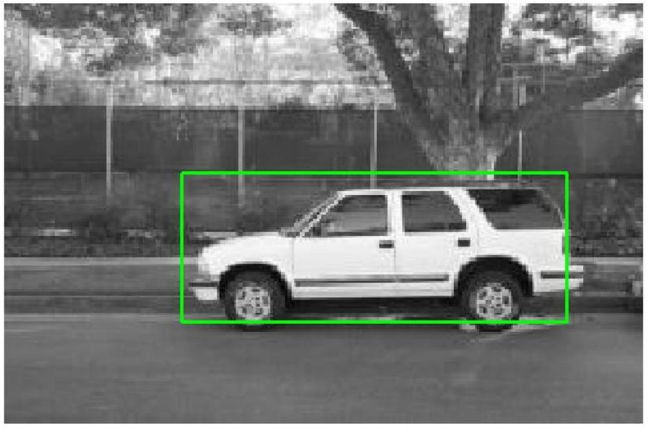
Scene

Template

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

Slide credit:
Kristen Grauman

Template matching



Detected template

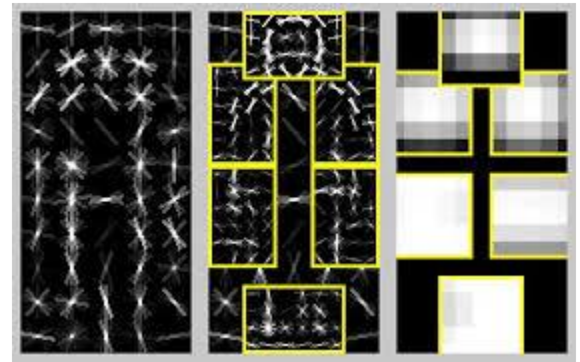
Template

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

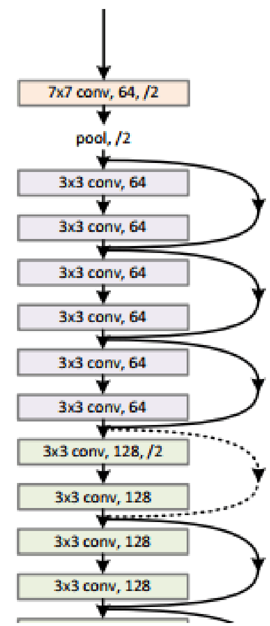
Slide credit:
Kristen Grauman

How about human?

- Deformable part model (deforming template) [Felzenszwalb et al. 10]



- Multilayer-neural network [He et al. 16]
  – The deformation in each layer is close to identity



Recap: Linear Filters

- <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

Summary

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

Coming up

- A1 out tonight (11.59 pm), due in 2 weeks