Computer Vision - Lecture 4

Gradients & Edges

05.11.2015

Bastian Leibe
RWTH Aachen
http://www.vision.rwth-aachen.de

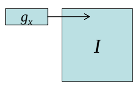
leibe@vision.rwth-aachen.de

Course Outline

- Image Processing Basics
  - Image Formation
  - Binary Image Processing
  - Linear Filters
  - Edge & Structure Extraction
- Segmentation
- Local Features & Matching
- Object Recognition and Categorization
- 3D Reconstruction
- Motion and Tracking

Topics of This Lecture

- Recap: Linear Filters
- Multi-Scale representations
  - How to properly rescale an image?
- Filters as templates
  - Correlation as template matching
- Image gradients
  - Derivatives of Gaussian
- Edge detection
  - Canny edge detector

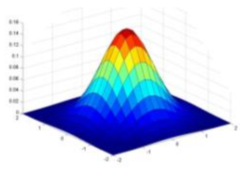


Recap: Gaussian Smoothing

- Gaussian kernel

$$G_\sigma = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$

- Rotationally symmetric
- Weights nearby pixels more than distant ones
  - This makes sense as 'probabilistic' inference about the signal
- A Gaussian gives a good model of a fuzzy blob



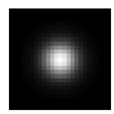
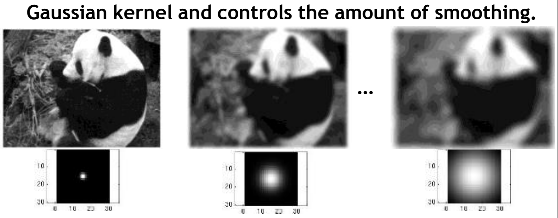


Image Source: Forsyth & Ponce

Recap: Smoothing with a Gaussian

- Parameter σ is the "scale" / "width" / "spread" of the Gaussian kernel and controls the amount of smoothing.

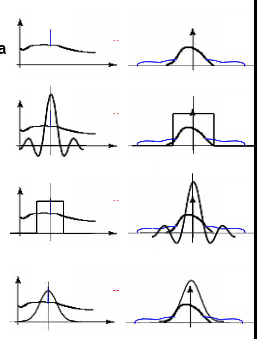


```
for sigma=1:3:10
  h = fspecial('gaussian', fsize, sigma);
  out = imfilter(im, h);
  imshow(out);
  pause;
end
```

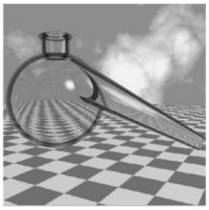
Slide credit: Kristen Grauman

Recap: Effect of Filtering

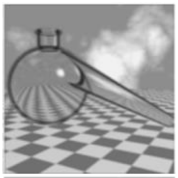
- Noise introduces high frequencies. To remove them, we want to apply a "low-pass" filter.
- The ideal filter shape in the frequency domain would be a box. But this transfers to a spatial sinc, which has infinite spatial support.
- A compact spatial box filter transfers to a frequency sinc, which creates artifacts.
- A Gaussian has compact support in both domains. This makes it a convenient choice for a low-pass filter.



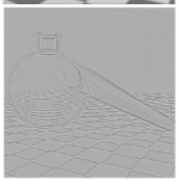
Recap: Low-Pass vs. High-Pass



Original image



Low-pass filtered

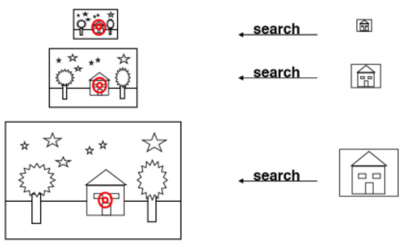


High-pass filtered

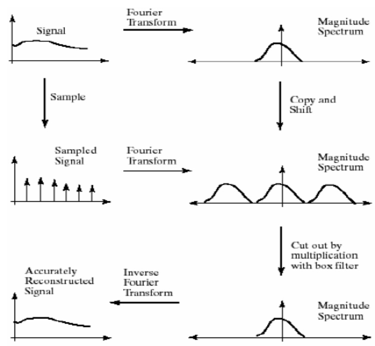
Topics of This Lecture

- Recap: Linear Filters
- Multi-Scale representations
  - How to properly rescale an image?
- Filters as templates
  - Correlation as template matching
- Image gradients
  - Derivatives of Gaussian
- Edge detection
  - Canny edge detector

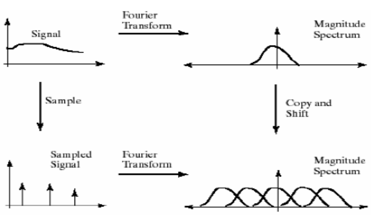
Motivation: Fast Search Across Scales



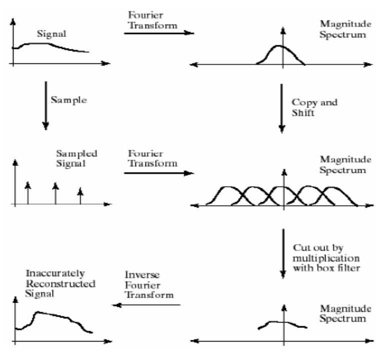
Recap: Sampling and Aliasing



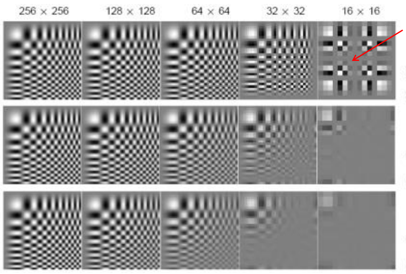
Recap: Sampling and Aliasing



Recap: Sampling and Aliasing



Recap: Resampling with Prior Smoothing



Artifacts!

no smoothing

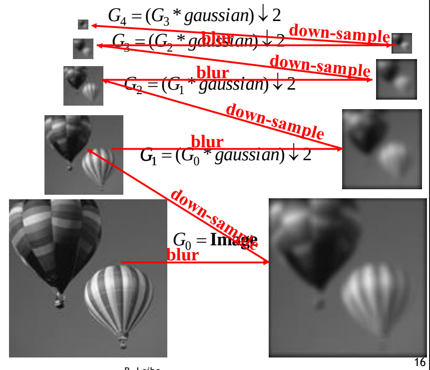
Gaussian $\sigma = 1$

Gaussian $\sigma = 2$

- Note: We cannot recover the high frequencies, but we can avoid artifacts by smoothing before resampling.

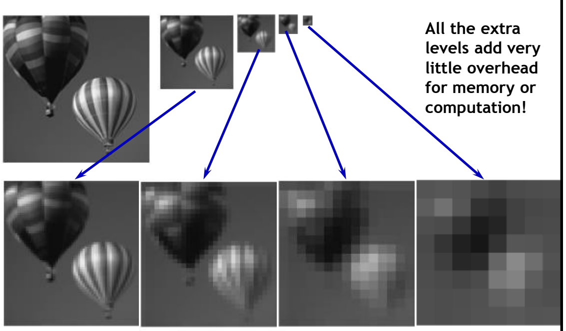
The Gaussian Pyramid

Low resolution



High resolution

Gaussian Pyramid - Stored Information



All the extra levels add very little overhead for memory or computation!

Summary: Gaussian Pyramid

- Construction: create each level from previous one
  - Smooth and sample
- Smooth with Gaussians, in part because
  - a Gaussian*Gaussian = another Gaussian
  - $G(\sigma_1) * G(\sigma_2) = G(\mathrm{sqrt}(\sigma_1^{\,2} + \sigma_2^{\,2}))$
- Gaussians are low-pass filters, so the representation is redundant once smoothing has been performed.
  - ⇒ There is no need to store smoothed images at the full original resolution.

Slide credit: David Lowe

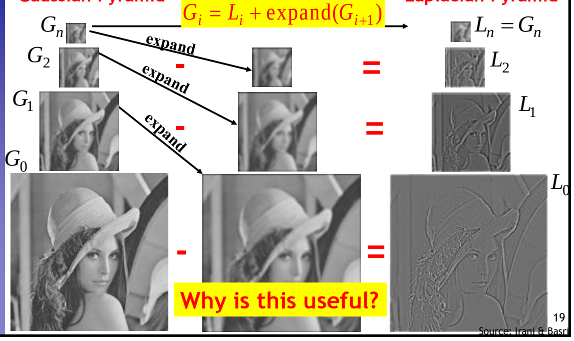
The Laplacian Pyramid

$L_i = G_i - \mathrm{expand}(G_{i+1})$

$G_i = L_i + \mathrm{expand}(G_{i+1})$

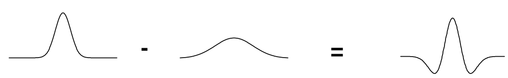
Gaussian Pyramid

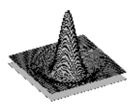
Laplacian Pyramid



Why is this useful?

Laplacian ~ Difference of Gaussian





DoG = Difference of Gaussians

Cheap approximation - no derivatives needed.

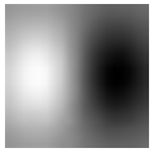
Topics of This Lecture

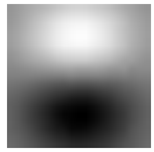
- Recap: Linear Filters
- Multi-Scale representations
  - How to properly rescale an image?
- **Filters as templates**
  - Correlation as template matching
- Image gradients
  - Derivatives of Gaussian
- Edge detection
  - Canny edge detector

Note: Filters are Templates

- Applying a filter at some point can be seen as taking a dot-product between the image and some vector.
- Filtering the image is a set of dot products.

- Insight
  - Filters look like the effects they are intended to find.
  - Filters find effects they look like.





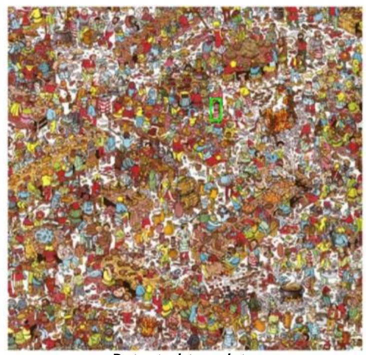
Where's Waldo?

Scene

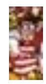
Template

Slide credit: Kristen Grauman

Where's Waldo?



Detected template

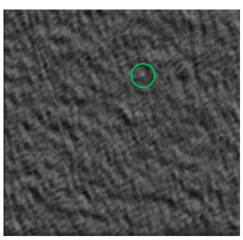


Template

Slide credit: Kristen Grauman

Where's Waldo?

Detected template



Correlation map

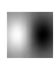
Slide credit: Kristen Grauman

Correlation as Template Matching

- Think of filters as a dot product of the filter vector with the image region
  - Now measure the angle between the vectors

$$a \cdot b = |a||b| \cos\theta \qquad \cos\theta = \frac{a \cdot b}{|a||b|}$$

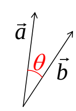
  - Angle (similarity) between vectors can be measured by normalizing the length of each vector to 1 and taking the dot product.



Template

·

Image region

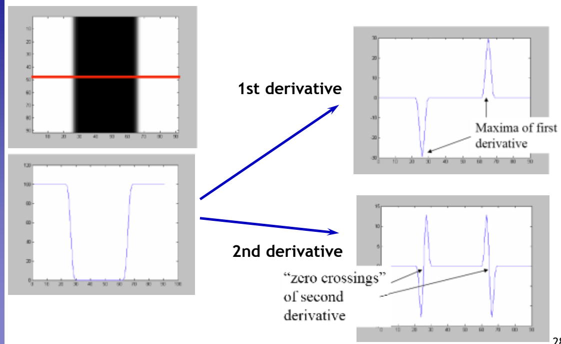


Vector interpretation

Topics of This Lecture

- Recap: Linear Filters
- Multi-Scale representations
  - How to properly rescale an image?
- Filters as templates
  - Correlation as template matching
- **Image gradients**
  - Derivatives of Gaussian
- Edge detection
  - Canny edge detector

Derivatives and Edges…



Differentiation and Convolution

- For the 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 this using finite differences:

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

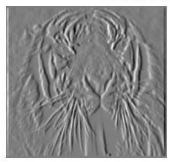
- To implement the above as convolution, what would be the associated filter?

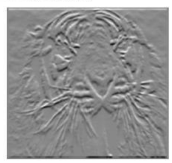
1	-1

Slide credit: Kristen Grauman

Partial Derivatives of an Image

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





-1	1

$\begin{bmatrix} -1 \\ 1 \end{bmatrix}$? or $\begin{bmatrix} 1 \\ -1 \end{bmatrix}$

Which shows changes with respect to x?

Slide credit: Kristen Grauman

Assorted Finite Difference Filters

Prewitt: $M_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$; $M_y = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$

Sobel: $M_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$; $\boxed{M_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}}$

Roberts: $M_x = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$; $M_y = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$

```
>> My = fspecial('sobel');
>> outim = imfilter(double(im), My);
>> imagesc(outim);
>> colormap gray;
```

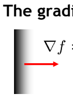
Slide credit: Kristen Grauman

Image Gradient

- The gradient of an image:

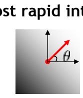
$$\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right]$$

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



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

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



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

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

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

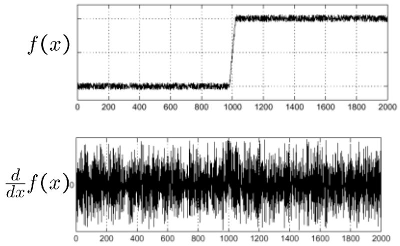
- The edge strength is given by the gradient magnitude

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

Slide credit: Steve Seitz

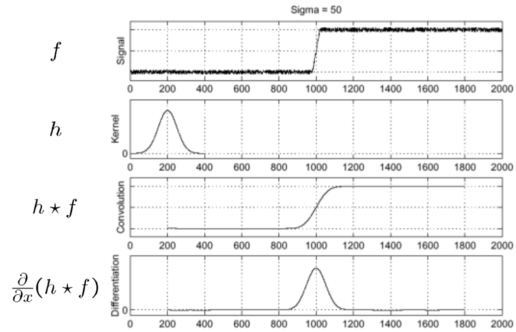
Effect of Noise

- Consider a single row or column of the image
  - Plotting intensity as a function of position gives a signal



Where is the edge?

Solution: Smooth First

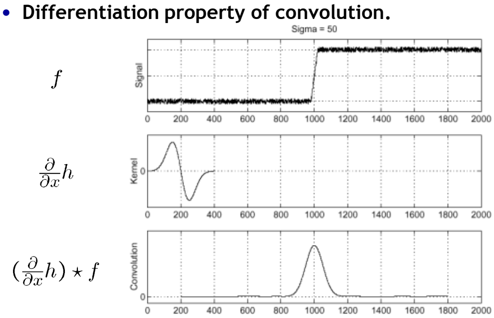


Where is the edge? Look for peaks in $\frac{\partial}{\partial x}(h \star f)$

Derivative Theorem of Convolution

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

- Differentiation property of convolution.

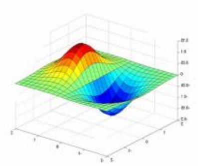


Derivative of Gaussian Filter

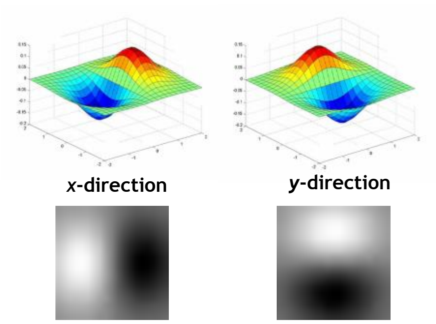
$$g * (h * I) = (g * h) * I$$

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

Why is this preferable?

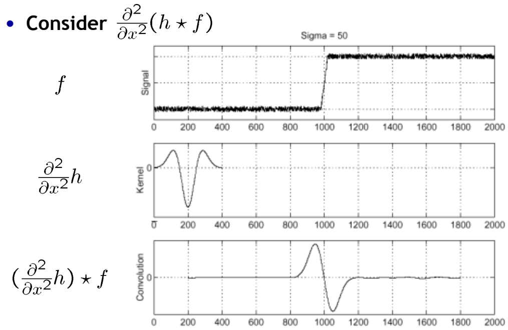


Derivative of Gaussian Filters



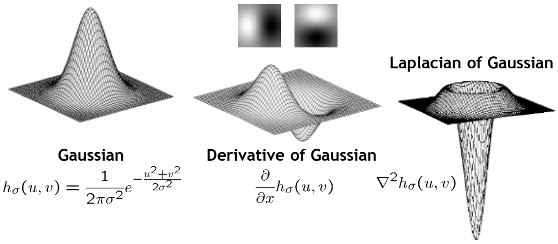
Laplacian of Gaussian (LoG)

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



Where is the edge? Zero-crossings of bottom graph

Summary: 2D Edge Detection Filters



Laplacian of Gaussian

Gaussian

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

Derivative of Gaussian

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

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

- ∇^2 is the Laplacian operator:

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

Slide credit: Kristen Grauman

Topics of This Lecture

- Recap: Linear Filters
- Multi-Scale representations
  - How to properly rescale an image?
- Filters as templates
  - Correlation as template matching
- Image gradients
  - Derivatives of Gaussian
- **Edge detection**
  - Canny edge detector

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

Slide credit: Kristen Grauman, David Lowe

Designing an Edge Detector

- Criteria for an "optimal" edge detector:
  - Good detection: the optimal detector should minimize the probability of false positives (detecting spurious edges caused by noise), as well as that of false negatives (missing real edges).
  - Good localization: the edges detected should be as close as possible to the true edges.
  - Single response: the detector should return one point only for each true edge point; that is, minimize the number of local maxima around the true edge.

Source: Li Fei-Fei

Gradients → Edges

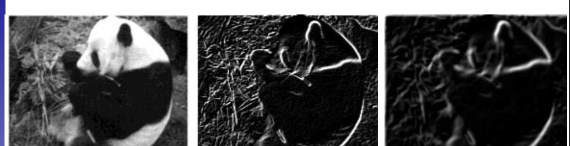
Primary edge detection steps

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
   - Thresholding, thinning

- Two issues
  - At what scale do we want to extract structures?
  - How sensitive should the edge extractor be?

adapted from Kristen Grauman

Scale: Effect of σ on Derivatives



σ = 1 pixel σ = 3 pixels

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

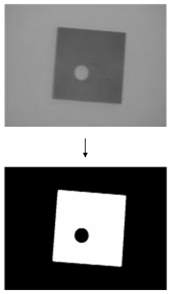
⇒ Larger values: larger-scale edges detected
⇒ Smaller values: finer features detected

Slide credit: Kristen Grauman

Sensitivity: Recall Thresholding

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

$$F_T[i,j] = \begin{cases} 1, & \text{if } F[i,j] \geq t \\ 0, & \text{otherwise} \end{cases}$$



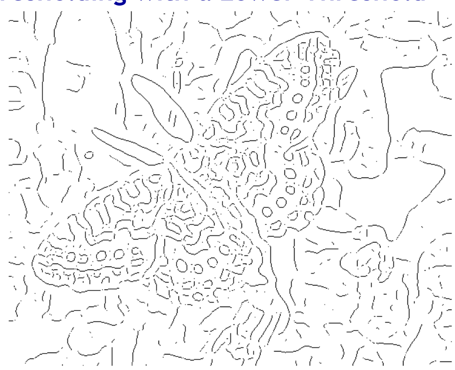
Original Image

Slide credit: Kristen Grauman

Gradient Magnitude Image

Slide credit: Kristen Grauman

Thresholding with a Lower Threshold



Slide credit: Kristen Grauman

Thresholding with a Higher Threshold

Slide credit: Kristen Grauman

Canny Edge Detector

- Probably the most widely used edge detector in computer vision
- Theoretical model: step-edges corrupted by additive Gaussian noise
- Canny has shown that the first derivative of the Gaussian closely approximates the operator that optimizes the product of *signal-to-noise ratio* and localization.

J. Canny, A Computational Approach To Edge Detection, *IEEE Trans. Pattern Analysis and Machine Intelligence*, 8:679-714, 1986.

Source: Li Fei-Fei

Canny Edge Detector

1. Filter image with derivative of Gaussian
2. Find magnitude and orientation of gradient
3. Non-maximum suppression:
   - Thin multi-pixel wide “ridges” down to single pixel width
4. 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
```

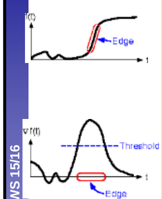

The Canny Edge Detector

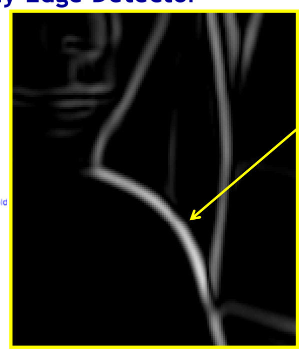
Original image (Lena)

The Canny Edge Detector

Gradient magnitude

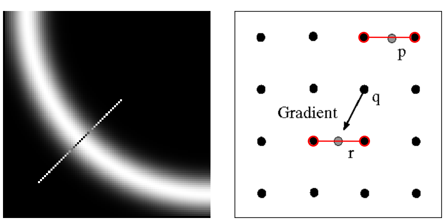
The Canny Edge Detector





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
  - ⇒ Linear interpolation based on gradient direction

The Canny Edge Detector

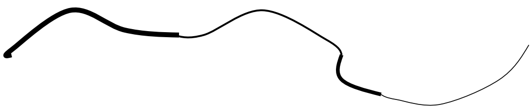
Problem: pixels along this edge didn’t survive the thresholding.

Thinning
(non-maximum suppression)

Solution: Hysteresis Thresholding

- Hysteresis: A lag or momentum factor
- Idea: Maintain two thresholds k_{high} and k_{low}
  - Use k_{high} to find strong edges to start edge chain
  - Use k_{low} to find weak edges which continue edge chain
- Typical ratio of thresholds is roughly

$$k_{high} \,/\, k_{low} = 2$$



Source: D. Lowe, S. Seitz

Hysteresis Thresholding

Original image

High threshold
(strong edges)

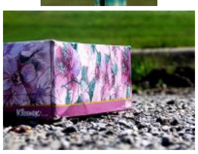
Low threshold
(weak edges)

Hysteresis threshold

Source: L. Fei-Fei

Object Boundaries vs. Edges

Background



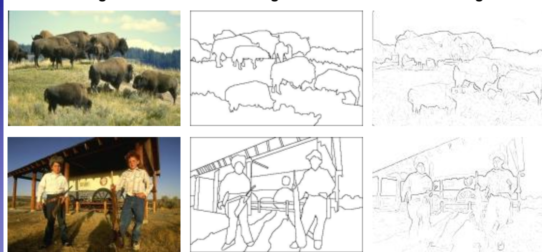
Texture

Shadows

Slide credit: Kristen Grauman

Edge Detection is Just the Beginning…

Image | Human segmentation | Gradient magnitude



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

Source: L. Lazebnik

References and Further Reading

- Background information on linear filters and their connection with the Fourier transform can be found in Chapter 7 of F&P. Additional information on edge detection is available in Chapter 8.
  - D. Forsyth, J. Ponce, *Computer Vision - A Modern Approach.* Prentice Hall, 2003