

RWTH AACHEN  
UNIVERSITY

# Computer Vision - Lecture 4

## Gradients & Edges

28.10.2014

Computer Vision WS 14/15

Bastian Leibe  
RWTH Aachen  
<http://www.vision.rwth-aachen.de>  
leibe@vision.rwth-aachen.de

RWTH AACHEN  
UNIVERSITY

## Announcements

- Exercise sheet 2 is available
  - Thresholding, Morphology
  - Gaussian smoothing
  - Image gradients
  - Edge Detection
 ⇒ *Deadline: Wednesday night, 05.11. (next week).*
- Reminder
  - You're encouraged to form teams of up to 3 people!
  - Make it easy for Aljosa & Dora to correct your solutions:
    - Turn in everything as a single zip archive.
    - Use the provided Matlab framework.
    - For each exercise, you need to implement the corresponding `apply` function. If the screen output matches the expected output, you will get the points for the exercise; else, no points.
    - Matlab helps you to find errors (red lines under your code)!

B. Leibe 2

RWTH AACHEN  
UNIVERSITY

## Announcements (2)

- Exam
  - There will be a written exam.
  - I'm currently organizing the exam date...
  - We'll organize a test exam towards the end of the semester.
- Admission requirements
  - Need to reach at least 50% of the exercise points.
  - Points are given
    - for each exercise sheet.
    - for the test exam.
  - Bonus points will be available on several occasions.
 ⇒ *If you follow the lecture and do the exercises regularly, you won't have to worry about getting admitted.*

B. Leibe 3

RWTH AACHEN  
UNIVERSITY

## Course Outline

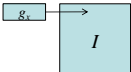
- Image Processing Basics
  - Image Formation
  - Binary Image Processing
  - Linear Filters
  - Edge & Structure Extraction
- Segmentation
- Local Features & Matching
- Object Recognition and Categorization
- 3D Reconstruction
- Motion and Tracking

B. Leibe 4

RWTH AACHEN  
UNIVERSITY

## 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

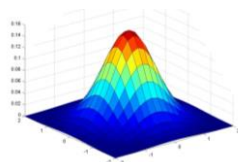
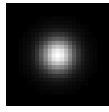


B. Leibe 5

RWTH AACHEN  
UNIVERSITY

## 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

B. Leibe 6  
Image Source: Forsyth & Ponce

RWTH AACHEN UNIVERSITY

## Recap: Smoothing with a Gaussian

- Parameter  $\sigma$  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

7

RWTH AACHEN UNIVERSITY

## 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.

8

RWTH AACHEN UNIVERSITY

## Recap: Low-Pass vs. High-Pass

Original image

Low-pass filtered

High-pass filtered

9

RWTH AACHEN UNIVERSITY

## 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

10

RWTH AACHEN UNIVERSITY

## Motivation: Fast Search Across Scales

11

RWTH AACHEN UNIVERSITY

## Recap: Sampling and Aliasing

12

Computer Vision WS 14/15

### Recap: Sampling and Aliasing

The diagram illustrates the process of sampling a signal. It starts with a continuous signal and its magnitude spectrum. The signal is sampled, resulting in a sampled signal. The magnitude spectrum of the sampled signal shows the original spectrum repeated at intervals, causing aliasing artifacts. The process involves Fourier Transform, Sampling, Copy and Shift, and Inverse Fourier Transform.

B. Leibe  
Image Source: Forsyth & Ponce

Computer Vision WS 14/15

### Recap: Sampling and Aliasing

The diagram illustrates the process of sampling a signal and the resulting aliasing artifacts. It starts with a continuous signal and its magnitude spectrum. The signal is sampled, resulting in a sampled signal. The magnitude spectrum of the sampled signal shows the original spectrum repeated at intervals, causing aliasing artifacts. The process involves Fourier Transform, Sampling, Copy and Shift, and Inverse Fourier Transform. A step is added to cut out the spectrum by multiplication with a box filter, resulting in an inaccurately reconstructed signal.

B. Leibe  
Image Source: Forsyth & Ponce

Computer Vision WS 14/15

### Recap: Resampling with Prior Smoothing

The diagram compares the results of resampling an image at different resolutions (256x256, 128x128, 64x64, 32x32, 16x16) with and without smoothing. The results show that without smoothing, there are significant artifacts (aliasing) in the lower resolution images. Smoothing (Gaussian with  $\sigma = 1$  and  $\sigma = 2$ ) helps to avoid these artifacts.

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.

B. Leibe  
Image Source: Forsyth & Ponce

Computer Vision WS 14/15

### The Gaussian Pyramid

The diagram illustrates the construction of a Gaussian pyramid. It shows the original image  $G_0$  and its successive levels  $G_1, G_2, G_3, G_4$ . Each level is constructed by blurring the previous level and then downsampling by a factor of 2. The equations for the levels are:  $G_1 = (G_0 * \text{gaussian}) \downarrow 2$ ,  $G_2 = (G_1 * \text{gaussian}) \downarrow 2$ ,  $G_3 = (G_2 * \text{gaussian}) \downarrow 2$ , and  $G_4 = (G_3 * \text{gaussian}) \downarrow 2$ .

Low resolution  
High resolution

B. Leibe  
Source: Irani & Bassi

Computer Vision WS 14/15

### Gaussian Pyramid - Stored Information

The diagram illustrates the Gaussian pyramid and the stored information at each level. It shows the original image and its successive levels  $G_0, G_1, G_2, G_3, G_4$ . The diagram highlights that all the extra levels add very little overhead for memory or computation.

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

B. Leibe  
Source: Irani & Bassi

Computer Vision WS 14/15

### Summary: Gaussian Pyramid

- Construction: create each level from previous one
  - Smooth and sample
- Smooth with Gaussians, in part because
  - a Gaussian \* Gaussian = another Gaussian
  - $G(\sigma_1) * G(\sigma_2) = G(\text{sqrt}(\sigma_1^2 + \sigma_2^2))$
- 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  
B. Leibe

Computer Vision WS 14/15

**The Laplacian Pyramid**

**Gaussian Pyramid**  $L_i = G_i - \text{expand}(G_{i+1})$   
**Laplacian Pyramid**  $G_i = L_i + \text{expand}(G_{i+1})$

$G_n$   $L_n = G_n$   
 $G_2$   $L_2$   
 $G_1$   $L_1$   
 $G_0$   $L_0$

Why is this useful?

Source: Irani & Barr

Computer Vision WS 14/15

**Laplacian ~ Difference of Gaussian**

DoG = Difference of Gaussians  
 Cheap approximation - no derivatives needed.

B. Leibe

Computer Vision WS 14/15

**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

B. Leibe

Computer Vision WS 14/15

**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.

B. Leibe

Computer Vision WS 14/15

**Where's Waldo?**

Scene

Template

Slide credit: Kristen Grauman

B. Leibe

Computer Vision WS 14/15

**Where's Waldo?**

Detected template


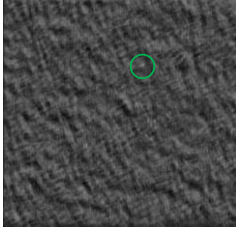
Template

Slide credit: Kristen Grauman

B. Leibe

RWTH AACHEN UNIVERSITY

## Where's Waldo?

Detected template

Correlation map


Computer Vision WS 14/15 25

Slide credit: Kristen Grauman B. Leibe

RWTH AACHEN UNIVERSITY

## 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
 
$$a \cdot b = |a| |b| \cos \theta \quad \cos \theta = \frac{a \cdot b}{|a| |b|}$$
  - Angle (similarity) between vectors can be measured by normalizing the length of each vector to 1 and taking the dot product.



Template


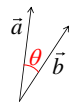


Image region



Vector interpretation


Computer Vision WS 14/15 26

B. Leibe

RWTH AACHEN UNIVERSITY

## 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

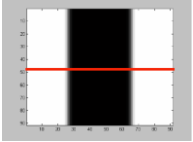


Computer Vision WS 14/15 27

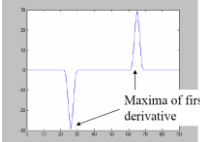
B. Leibe

RWTH AACHEN UNIVERSITY

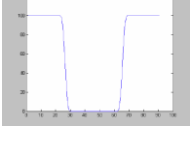
## Derivatives and Edges...



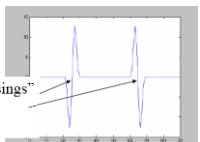
1st derivative



Maxima of first derivative



2nd derivative



"Zero crossings" of second derivative

Computer Vision WS 14/15 28

B. Leibe

RWTH AACHEN UNIVERSITY

## Differentiation and Convolution

- For the 2D function  $f(x,y)$ , the partial derivative is:
 
$$\frac{\partial f(x,y)}{\partial x} = \lim_{\epsilon \rightarrow 0} \frac{f(x+\epsilon, y) - f(x,y)}{\epsilon}$$
- 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

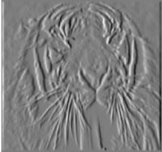
-1

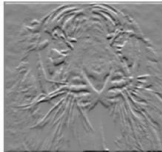
Computer Vision WS 14/15 29

Slide credit: Kristen Grauman B. Leibe

RWTH AACHEN UNIVERSITY

## Partial Derivatives of an Image

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


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


-1

1

-1

?

1

Which shows changes with respect to x?

Computer Vision WS 14/15 30

Slide credit: Kristen Grauman B. Leibe

RWTH AACHEN UNIVERSITY


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

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

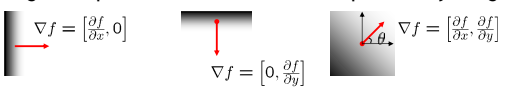

```
>> My = fspecial('sobel');
>> outim = imfilter(double(im), My);
>> imagesc(outim);
>> colormap gray;
```



Slide credit: Kristen Grauman B. Leibe 31

RWTH AACHEN UNIVERSITY

## Image Gradient

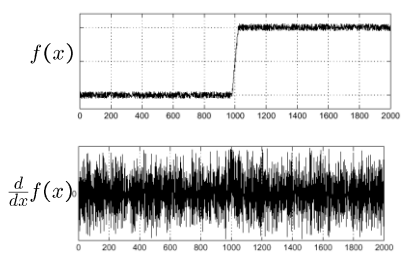
- The gradient of an image:
 
$$\nabla f = \left[ \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right]$$
- The gradient points in the direction of most rapid intensity change
 
- The gradient direction (orientation of edge normal) is given by:
 
$$\theta = \tan^{-1} \left( \frac{\partial f / \partial y}{\partial f / \partial x} \right)$$
- 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 B. Leibe 31

RWTH AACHEN UNIVERSITY

## Effect of Noise

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

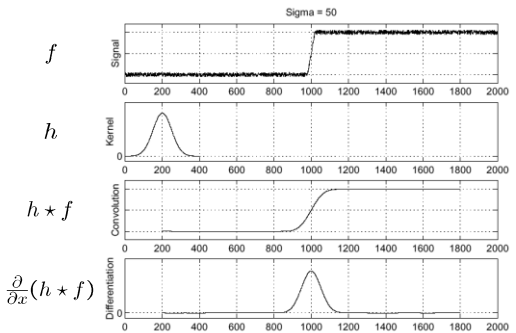


Where is the edge?

Slide credit: Steve Seitz B. Leibe 33

RWTH AACHEN UNIVERSITY

## Solution: Smooth First



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

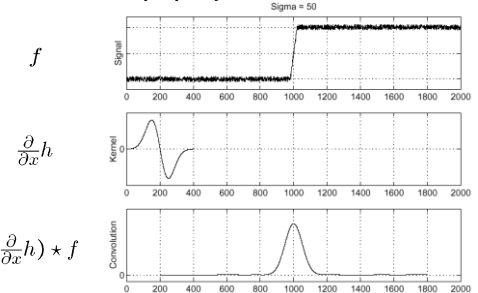
Slide credit: Steve Seitz B. Leibe 34

RWTH AACHEN UNIVERSITY

## Derivative Theorem of Convolution

$$\frac{\partial}{\partial x}(h * f) = \left(\frac{\partial}{\partial x}h\right) * f$$

- Differentiation property of convolution.



Slide credit: Steve Seitz B. Leibe 35

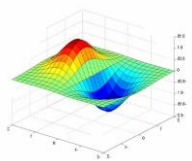
RWTH AACHEN UNIVERSITY

## 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?



Slide adapted from Kristen Grauman 36



Computer Vision WS 14/15

## Derivative of Gaussian Filters

**x-direction**      **y-direction**

B. Leibe      Source: Svetlana Lazebnik

37

Computer Vision WS 14/15

## Laplacian of Gaussian (LoG)

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

Where is the edge?      Zero-crossings of bottom graph

Slide credit: Steve Seitz      B. Leibe

38

Computer Vision WS 14/15

## Summary: 2D Edge Detection Filters

**Gaussian**      **Derivative of Gaussian**      **Laplacian of Gaussian**

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

$$\frac{\partial}{\partial x} h_{\sigma}(u, v) \quad \nabla^2 h_{\sigma}(u, v)$$

- $\nabla^2$  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      B. Leibe

39

Computer Vision WS 14/15

## 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

B. Leibe

40

Computer Vision WS 14/15

## 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      B. Leibe

41

Computer Vision WS 14/15

## 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.

True edge      Poor robustness to noise      Poor localization      Too many responses

Slide credit: B. Leibe      Source: Li Fei-Fei

42

## 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?

Computer Vision WS 14/15 44  
 adapted from Kristen Grauman B. Leibe

## Scale: Effect of $\sigma$ on Derivatives

$\sigma = 1 \text{ pixel}$        $\sigma = 3 \text{ pixels}$

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

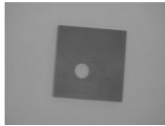
⇒ Larger values: larger-scale edges detected  
 ⇒ Smaller values: finer features detected

Computer Vision WS 14/15 45  
 Slide credit: Kristen Grauman B. Leibe


## 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).

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



↓




Computer Vision WS 14/15 46  
 B. Leibe

## Original Image




Computer Vision WS 14/15 47  
 Slide credit: Kristen Grauman B. Leibe

## Gradient Magnitude Image



Computer Vision WS 14/15 48  
 Slide credit: Kristen Grauman B. Leibe

## Thresholding with a Lower Threshold




Computer Vision WS 14/15 49  
 Slide credit: Kristen Grauman B. Leibe



RWTH AACHEN UNIVERSITY

## Thresholding with a Higher Threshold



Slide credit: Kristen Grauman      B. Leibe

50

RWTH AACHEN UNIVERSITY

## 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.

B. Leibe      Source: Li Fei-Fei

51

RWTH AACHEN UNIVERSITY

## 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
```

B. Leibe      Source: D. Lowe, L. Fei-Fei

52

RWTH AACHEN UNIVERSITY

## The Canny Edge Detector




Original image (Lena)

B. Leibe

53

RWTH AACHEN UNIVERSITY

## The Canny Edge Detector



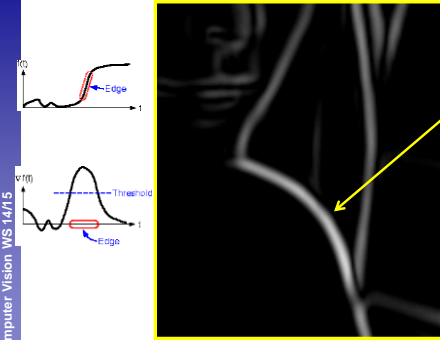
Gradient magnitude

B. Leibe

54

RWTH AACHEN UNIVERSITY

## The Canny Edge Detector



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

B. Leibe

56

Computer Vision WS 14/15

## 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

B. Leibe Source: Forsyth & Ponce 57

Computer Vision WS 14/15

## The Canny Edge Detector

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

Thinning (non-maximum suppression)

Slide credit: Kristen Grauman B. Leibe 58

Computer Vision WS 14/15

## 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  $k_{high} / k_{low} = 2$

B. Leibe Source: D. Lowe, S. Seitz 59

Computer Vision WS 14/15

## Hysteresis Thresholding

Original image

High threshold (strong edges) Low threshold (weak edges) Hysteresis threshold

courtesy of G. Loy

B. Leibe Source: L. Fei-Fei 60

Computer Vision WS 14/15

## Object Boundaries vs. Edges

Background Texture Shadows

Slide credit: Kristen Grauman B. Leibe 61

Computer Vision WS 14/15

## 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/>

B. Leibe Source: L. Lazebnik 62

## 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

