Image gradients and edges

April 11th, 2019

Yong Jae Lee
UC Davis

Last time

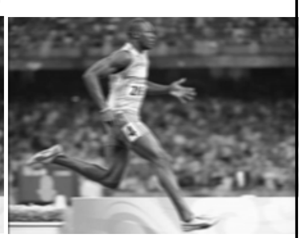
- Image formation
- Linear filters and convolution useful for
  - Image smoothing, removing noise
    - Box filter
    - Gaussian filter
    - Impact of scale / width of smoothing filter
- Separable filters more efficient
- Median filter: a non-linear filter, edge-preserving

2

Review

Filter f = 1/9 x [1 1 1 1 1 1 1 1 1]

original image g



filtered

3

Slide credit: Kristen Grauman

Review

Filter $f = 1/9 \times [1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1]^T$

original image g   filtered

4

Slide credit: Kristen Grauman

Review

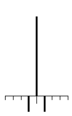
How do you sharpen an image?

5

Practice with linear filters

Original

$$\begin{bmatrix}0&0&0\\0&2&0\\0&0&0\end{bmatrix} - \frac{1}{9}\begin{bmatrix}1&1&1\\1&1&1\\1&1&1\end{bmatrix}$$

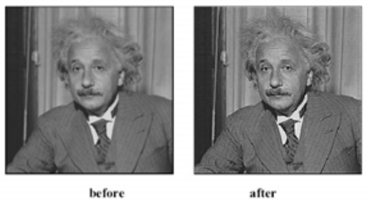


Sharpening filter:
accentuates differences
with local average

6

Slide credit: David Lowe

Filtering examples: sharpening



7

Slide credit: Kristen Grauman

Sharpening revisited

What does blurring take away?

Let's add it back:

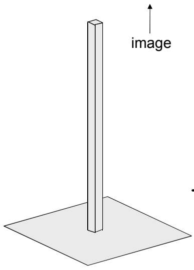
8

Slide credit: Svetlana Lazebnik

Unsharp mask filter

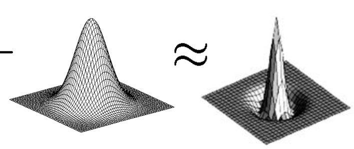
$$f + \alpha(f - f * g) = (1+\alpha) f - \alpha\, f * g = f * ((1+\alpha) e - g)$$

image; blurred image; unit impulse (identity)



unit impulse

—



Gaussian

≈

Laplacian of Gaussian

9

Slide credit: Svetlana Lazebnik

Review

Median filter f:

Is f(a+b) = f(a)+f(b)?

Example:
a = [10 20 30 40 50]
b = [55 20 30 40 50]

Is f linear?

10

Slide credit: Devi Parikh

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, increase contrast, etc)
  - Extract information (texture, edges, interest points, etc)
  - Detect patterns (template matching)

11

Slide credit: Kristen Grauman, Adapted from Derek Hoiem

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, increase contrast, etc)
  - Extract information (texture, edges, interest points, etc)
  - Detect patterns (template matching)

12

Slide credit: Kristen Grauman, 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?**

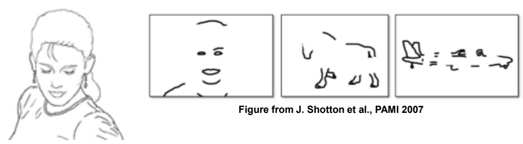


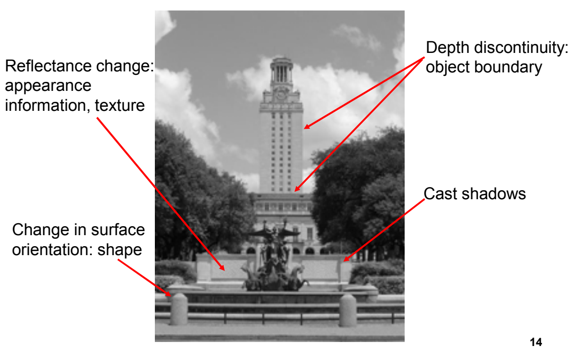
Figure from J. Shotton et al., PAMI 2007

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

13

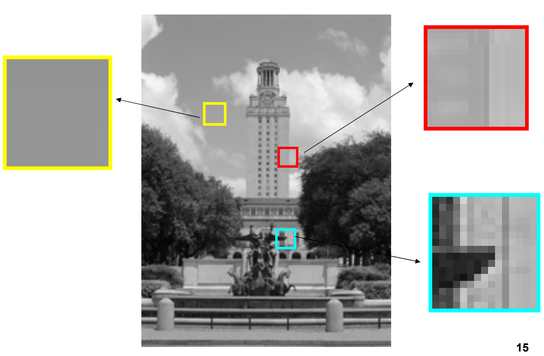
Slide credit: Kristen Grauman

What causes an edge?



14

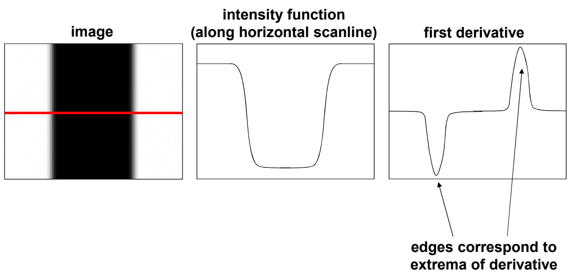
Edges/gradients and invariance



15

Derivatives and edges

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



16

Slide credit: Svetlana 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?

17

Slide credit: Kristen Grauman

Partial derivatives of an image

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

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

-1	1

-1	? or	1
1		-1

Which shows changes with respect to x?

(showing filters for correlation)

18

Slide credit: Kristen Grauman

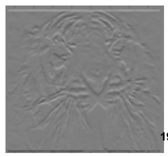
Assorted finite difference filters

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

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;
```



19

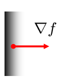
Slide credit: Kristen Grauman

Image gradient

The gradient of an image:

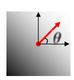
$$\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right]$$

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



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

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



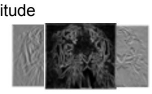
$\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right]$

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

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



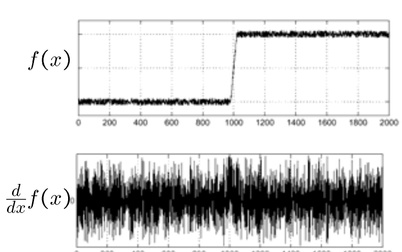
Slide credit: Steve Seitz

20

Effects 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?

21

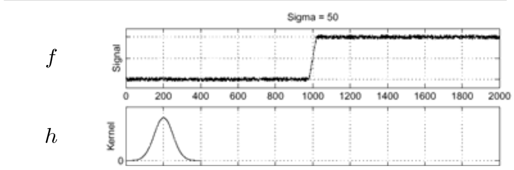
Slide credit: Steve Seitz

Effects of noise

- Difference filters respond strongly to noise
  - Image noise results in pixels that look very different from their neighbors
  - Generally, the larger the noise the stronger the response
- What can we do about it?

22
Source: D. Forsyth

Solution: smooth first



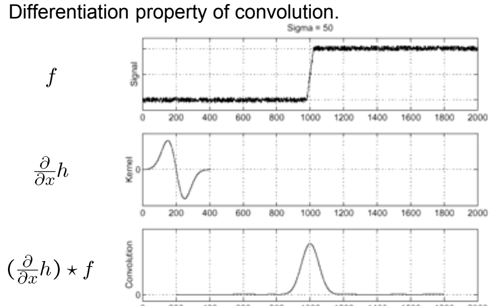
Where is the edge? Look for peaks in $\frac{\partial}{\partial x}(h \star f)$

23
Slide credit: Kristen Grauman

Derivative theorem of convolution

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

Differentiation property of convolution.



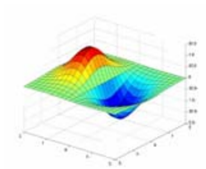
24
Slide credit: Steve Seitz

Derivative of Gaussian filters

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

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

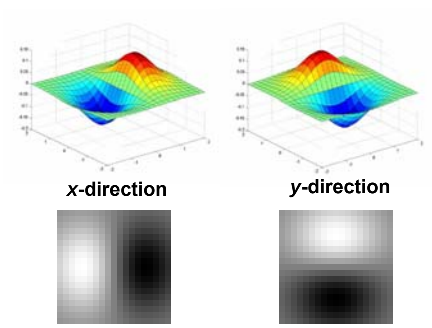
$$\otimes \begin{bmatrix} 1 & -1 \end{bmatrix}$$



25

Slide credit: Kristen Grauman

Derivative of Gaussian filters



26

Slide credit: Svetlana Lazebnik

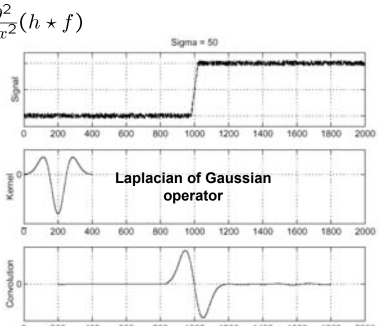
Laplacian of Gaussian

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

f

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

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

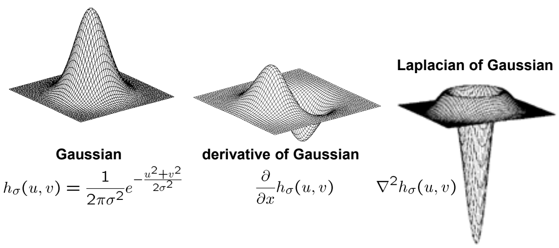


Where is the edge? Zero-crossings of bottom graph

27

Slide credit: Steve Seitz

2D edge detection filters



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

- ∇^2 is the Laplacian operator:

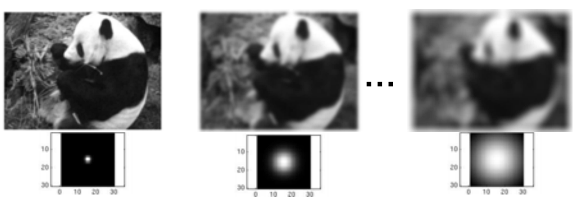
$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

28

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.



29

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

30

So, what scale to choose?

It depends what we're looking for.

31

Slide credit: Kristen Grauman

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>
  - ___________ signs used to get high response in regions of high contrast
  - Sum to ___ → no response in constant regions
  - High absolute value at points of high contrast

32

Slide credit: Kristen Grauman

Seam carving: main idea

[Shai & Avidan, SIGGRAPH 2007]

33

Slide credit: Kristen Grauman

Seam carving: main idea

Content-aware resizing

Traditional resizing

[Shai & Avidan, SIGGRAPH 2007]

34

Slide credit: Kristen Grauman

Seam carving: main idea

video

35

Seam carving: main idea

Content-aware resizing

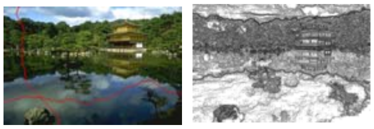
Intuition:

- 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.

36

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.

37

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

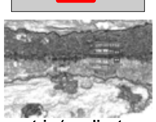
Compute it efficiently with dynamic programming. 38

Slide credit: Kristen Grauman

How to identify the minimum cost seam?

- How many possible seams are there?
  - height *h*, width *w*
- First, consider a **greedy** approach:

1	3	0
2	8	9
5	2	6



Energy matrix (gradient magnitude)

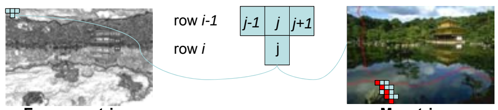
39

Slide credit: Adapted from Kristen Grauman

Seam carving: algorithm

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

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



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**.

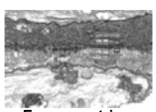
40

Slide credit: Kristen Grauman

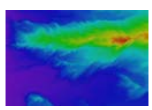
Example

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

1	3	0
2	8	9
5	2	6



Energy matrix (gradient magnitude)



M matrix (for vertical seams)

41

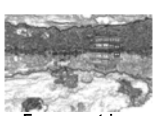
Slide credit: Kristen Grauman

Example

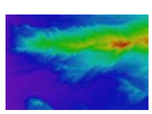
$$\mathbf{M}(i, j) = Energy(i, j) + \min(\mathbf{M}(i-1, j-1), \mathbf{M}(i-1, j), \mathbf{M}(i-1, j+1))$$

1	3	0
2	8	9
5	2	6

1	3	0
3	8	9
8	5	14



Energy matrix (gradient magnitude)



M matrix (for vertical seams)

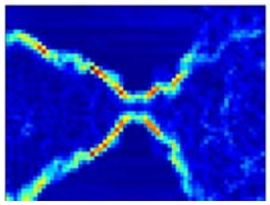
42

Slide credit: Kristen Grauman

Real image example

Original Image

Energy Map

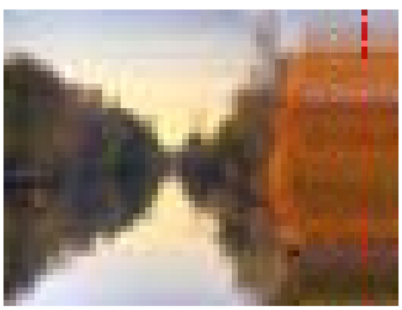


Blue = low energy
Red = high energy

43

Slide credit: Kristen Grauman

Real image example



44

Slide credit: Kristen Grauman

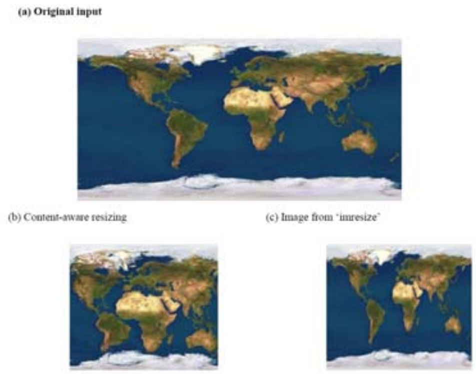
Other notes on seam carving

- 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

45

Slide credit: Kristen Grauman

Example seam carving results



46

47

48

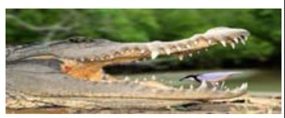
Original image

Conventional resize

Seam carving result

49

Original image



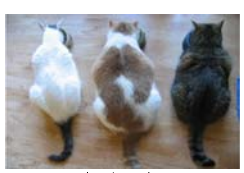
Conventional resize

Seam carving result

50

Original image (599 by 799)

Conventional resize (399 by 599)



Seam carving (399 by 599)

51

Removal of a marked object

(a) Selected an area.

(b) Object is removed.

(c) Selected an area.

(d) Object is removed.

52

Removal of a marked object

53

Removal of a marked object

Figure A

Figure C

54

“Failure cases” with seam carving

55

“Failure cases” with seam carving

56

Questions?

See you Tuesday!

57