Relationships between pixels

A pixel has two types of neighbors:

- \( N_4 \), 4-neighbors (also called edge neighbors)
- \( N_D \), D-neighbors (also called diagonal, or point-neighbors)

Together, 4- and D-neighbors are called
- \( N_8 \), or 8-neighbors

Adjacency and connectivity

In a binary image, two pixels \( p \) and \( q \) are

- 4-adjacent if they have the same value and \( q \) is in the set \( N_4(p) \).
- 8-adjacent if they have the same value and \( q \) is in the set \( N_8(p) \).
- \( m \)-adjacent if they have the same value and \( q \) is in the set \( N_4(p) \) OR \( q \) is in the set \( N_D(p) \) AND the set \( N_4(p) \cap N_4(q) \) is empty.

Two pixels (or objects) are 8-, 4-, or \( m \)-connected if a 8-, 4-, or \( m \)-path can be drawn between them.

Why is this important?

How many objects are there in this image?

The border of an object

8-connected border, requires a 4-connected background

4-connected border, requires an 8-connected background

Spatial filtering

The pixel value in the output image calculated from a local neighborhood of the pixel in the input image.

The local neighborhood is described by a window, mask, kernel, template, or spatial filter (typical sizes 3x3, 5x5, 7x7... pixels)
Spatial filters

Linear filters:
- Smoothing filters
- Mean filters
- Gauss filters
- Edge enhancing filter
- Sobel operator
- Prewitt operator
- Laplace operator

Non-linear filters**
- Median, Min, Max, Percentile filters
  **can not be generalized to frequency domain

Smoothing filters

- Reduce noise
- Blur, or soften the image, remove small details

Original image
- Mean 3x3
- Mean 5x5
- Mean 11x11

Non linear, or order statistics filters

- Median and percentile filters
  - Preserve edges while reducing noise
  - Useful if the character of the noise is known
  - Slow
  - No correspondence in frequency domain

Sharpening and edge enhancing filters

Calculus: Changes are described by derivatives (partial derivatives in 2D)

An edge is described by its gradient magnitude and direction.

In the discrete case, we approximate the derivatives by differences. We use spatial filters, or weighted masks, looking at local neighborhoods and traverse the image by convolution.

Calculating magnitude and direction

One mask is created for each possible direction. In the 3x3 case, we have eight directions, resulting in eight masks with weights. The response of each filter represents the strength, or magnitude, of the edge in that direction.

Ex. $m_x$ gives the strength in x-direction
$m_y$ gives the strength in y-direction
edge magnitude = $(m_x^2 + m_y^2)^{1/2}$
edge direction = $\tan^{-1}(m_x/m_y)$

Alternatively, the mask giving the maximum response approximates both magnitude and direction of the gradient.

Edge enhancing filters

- The Sobel operator:

  - Original image
  - Horizontal down
  - Horizontal up
  - Vertical left
  - Vertical right
  - Sum of results

- So good...

- SPAM
Rotation independent edge detection

- The Laplace operator approximates the second derivative (magnitude only)
- Gives 0 as output in homogenous regions and output ≠ 0 at discontinuities.
- The size of the filter decides the types of edges (discontinuities) that are found.
- Independent of edge direction and very useful when searching for curved edges (faster than 4x Sobel)

Edge enhancing filters

- The Laplace operator: detection of edges independent of direction

Image sharpening, or “crisp filter” by enhancing edges and adding them to the original image

Original image

Mean filter = "smoothing"

Laplace filter = "sharpening"

Smoothing as pre-processing

Broken contours can be re-connected by smoothing.

Problem 3.21

(a) In a character recognition application, text pages are reduced to binary form using a thresholding transformation function of the form shown to the right. This is followed by a procedure that thins the characters until they become strings of binary 1’s on a background of 0’s. Due to noise, the binarization and thinning processes result in broken strings of characters with gaps ranging from 1 to 3 pixels. One way to “repair” the gaps is to run an averaging mask over the binary image to blur it, and thus create bridges of nonzero pixels between gaps. Give the (odd) size of the smallest averaging mask capable of performing the task.

(b) After bridging the gaps, it is desirable to threshold the image in order to convert it back to binary form. For your answer in (a), what is the minimum value of the threshold required to accomplish this, without causing the segments to break again?

Repetition (problem 3.12)

Two images, \(f(x,y)\) and \(g(x,y)\) have histograms \(h_f\) and \(h_g\). Give the conditions under which you can determine the histograms of

(a) \(f(x,y)+g(x,y)\)
(b) \(f(x,y)-g(x,y)\)
(c) \(f(x,y)\cdot g(x,y)\)
(d) \(f(x,y)/g(x,y)\)

In terms of \(h_f\) and \(h_g\). Explain how to obtain the histograms in each case.