# ELIC 629

# Digital Image Processing

Winter 2006

## Image Enhancement in the Spatial Domain:

Sharpening Spatial Filters, Introduction to edges, First order derivatives, Second order derivatives

Bill Kapralos

Tuesday, March 7 2006



## Overview (1):

- **Before We Begin**
  - Administrative details
  - Review → some questions to consider
- **Image Edges**
  - Introduction
  - Importance of edge detection
  - Modeling an edge

## Overview (2):

- **Introduction to Sharpening Filters**
  - First order derivatives – the gradient
  - Second order derivative
- **First Order Derivatives - The Gradient**
  - Introduction
  - Properties
- **Second Order Derivatives - The Laplacian**
  - Introduction
  - Defining the Laplacian operator

# Before We Begin

## Administrative Details (1):

- **Lab 6 Today**
  - This lab may be spread across two days
  - There is a report required for this lab but no lab assignment
    - Due Mar. 21 2006 → if one week to complete
    - Due Mar. 28 2006 → if two weeks to complete
- **Mid-Term Exams**
  - Will be returned during the lab period
  - We will discuss the solutions at a latter time

## Some Questions to Consider (1):

- What is spatial filtering ?
- What is a smoothing spatial filter ?
- What is an averaging filter ?
- What is a weighted averaging filter ?
- What is a sharpening filter ?

# Image Edges

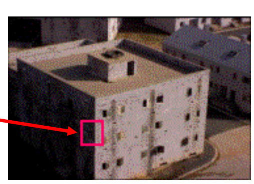
## Introduction (1):

- **What is an Edge ?**
  - Intuitively → a border between two regions, where each region has (approximately) uniform brightness (gray level)
  - In an image edges typically arise from
    1. Occluding contours in an image
       - Two image regions correspond to two different surfaces
    2. Abrupt changes in surface orientation
    3. Discontinuities in surface reflectance

## Introduction (2):

- **What is an Edge ? (cont…)**

Edge due to an occluding contour



Edge due to an abrupt change in surface orientation

## Introduction (3):

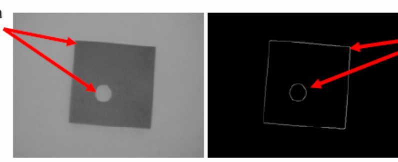
- **What is an Edge ? (cont…)**

Edge due to a change in surface reflectance

## Importance of Edge Detection (1):

- **In Typical Images, Edges Characterize Object Boundaries/Borders**
  - Allows us to locate/identify objects in a scene e.g., to segment an image → many, many applications!

Two objects in image



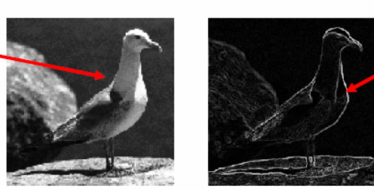
Boundary of objects in image

Original Image | "Edge" Image

## Importance of Edge Detection (2):

- **A More realistic Example**

Object of interest

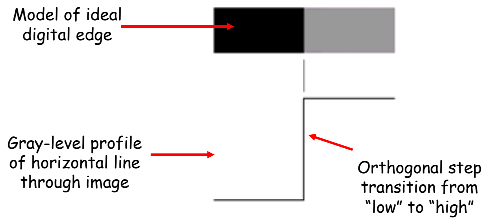


Boundary of object of interest

Original Image | "Edge" Image

## Modeling an Edge (1):

- **Ideal Edge Model**
  - A set of connected pixels, each of which is located at an orthogonal step transition in gray level

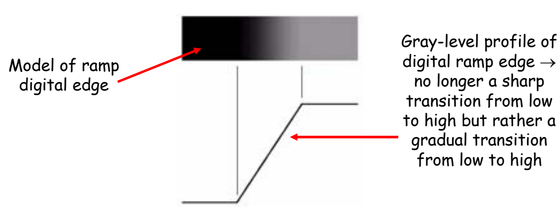



## Modeling an Edge (2):

- **In Practice, Ideal Edges Don't Exist!**
  - Sampling and the fact that sampling acquisition equipment etc. is far from perfect leads to edges that are blurred
  - Changing illumination (lighting conditions) will cause changes to edges & all parts of an image in general
    - Changing lighting conditions are actually a HUGE problem for vision/image processing tasks → many algorithms will not generalize across different lighting conditions
    - Color constancy → a big field in computer vision but still an un-solved problem!

## Modeling an Edge (3):

- **Reality → Edges Have a "Ramp-Like" Profile**
  - The slope of the ramp is inversely proportional to the degree of blurring in the edge
  - Updated definition → region of image in which the gray-level changes significantly over short distance



## Modeling an Edge (4):

- **In Practice, Ideal Edges Dont Exist! (cont.)**
  - Edge is no longer a one-pixel thick path
    - An edge point is now any point contained in the ramp and an edge would be a set of such points which are connected
    - Thickness of edge depends on length of ramp which is determined by its slope which itself is determined by the amount of blurring
    - Blurred edges are typically thicker e.g., the greater the blurring → the thicker the edge

# Sharpening Filters (Review)

## Foundation (5):

- **First Order Derivative in Greater Detail**
  - Basic definition of a first order 1D function f(x) is the difference

$$\frac{\partial f}{\partial x} = f(x+1) - f(x) \qquad \frac{\partial f}{\partial y} = f(y+1) - f(y)$$

  - Remember → above definition is of one variable (x) only since images are a function of two variables x,y e.g., f(x,y) we will be dealing with derivatives along both spatial axis "separately" hence the use of "partial derivative"

## Foundation (6):

- **Second Order Derivative in Greater Detail**
  - Basic definition of a first order 1D function f(x) is the difference

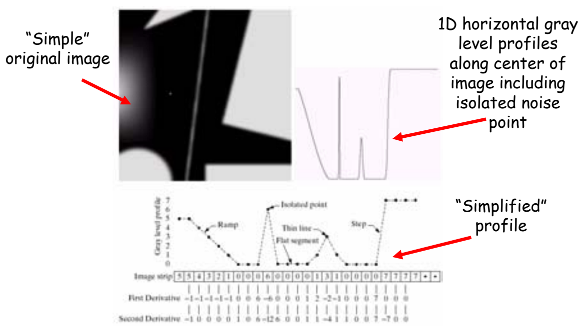
$$\frac{\partial^2 f}{\partial x^2} = f(x+1) + f(x-1) - 2f(x)$$

$$\frac{\partial^2 f}{\partial y^2} = f(y+1) + f(y-1) - 2f(y)$$

  - Once again, remember, above definition is for one variable only whereas in digital images we are dealing with two variables, x,y

## Foundation (7):

- **Graphical Illustration of Digital Derivatives**




## Foundation (8):

- **Graphical Illustration Explained**
  - Traversing profile from left to right
    - First order derivative is non-zero along entire ramp but second order derivative is non-zero only at onset and end of ramp
    - Since edges in image have similar profile, we can conclude first order derivative produces "thick" edges while second order derivatives produces "finer" edges
    - A second order derivative enhances much more finer detail than first order derivative (but also enhances noise as well!)

## Foundation (9):

- **Graphical Illustration Explained (cont…)**
  - To summarize
    1. First order derivatives generate thicker edges in an image
    2. Second order derivatives have stronger response to fine detail e.g., thin lines and isolated points (noise as well)
    3. First order derivatives have stronger response to gray level step
    4. Second order derivatives produce double response at step changes in gray level

## Foundation (10):

- **Graphical Illustration Explained (cont…)**
  - To summarize (cont…)
    - Generally, second order derivatives are better for image enhancement as opposed to first order derivatives since they are able to enhance such fine detail

# First Order Derivatives The Gradient

## Introduction (1):

- **Gradient Defined**
  - Gradient is a measure of change in a function
  - An image can be considered to be an "array" of samples of some continuous function of intensity
    - Significant changes in gray levels in image can thus be detected using discrete approximation of gradient
    - Edge detection → detecting significant local changes in an image
  - Two-dimensional equivalent of the first derivative

## Introduction (2):

- **Gradient Defined (cont…)**
  - For function f(x,y) gradient of f at coordinates (x,y) is defined as a two-dimensional column vector

$$G[f(x,y)] = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

## Gradient Properties (1):

- **Two Important Properties of Gradient**
  1. Vector G[f(x,y)] points in direction of maximum increase of function f(x,y)
  2. Magnitude of gradient equals maximum rate of increase of f(x,y) per unit distance in direction G. Magnitude given as

$$mag(G[f(x,y)]) = \sqrt{G_x^2 + G_y^2} = \sqrt{\left[\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2\right]}$$

$$\theta = \tan^{-1}(y/x)$$

## Gradient Properties (2):

- **Properties of the Gradient**
  - Components of gradient vector are linear
  - Magnitude of gradient vector is not linear given squaring and square root operations
  - Partial derivates of gradient vector are not isotropic (e.g., not rotation invariant)
  - Magnitude of gradient is isotropic
  - Often, although incorrect, we refer to the magnitude of the gradient as the gradient itself

## Gradient Properties (3):

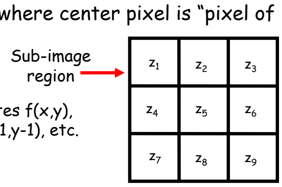
- **Properties of the Gradient (cont…)**
  - Implementing the gradient magnitude equation for an entire image is very computationally expensive and certainly not a trivial matter!
    - Approximate gradient mag. using absolute values

$$mag(G[f(x,y)]) = |G_x| + |G_y|$$

    - Above equation is easier to compute and preserves relative changes in gray levels
    - Isotropic property generally lost → as with Laplacian preserved for limited number of rotational increments, depending on mask

## Approximating the Gradient (1):

- **Digital Approximation to Gradient**
  - Recall → derivatives in images are approximated by differences between pixel intensity (gray levels)
    - Gradient approximated by differences
  - For simplification, will use previous definition of 3x3 image region, where center pixel is "pixel of interest"




Recall, $z_5$ denotes f(x,y), $z_1$ denotes f(x-1,y-1), etc.

## Approximating the Gradient (2):

- **Digital Approximation to Gradient (cont…)**
  - Simplest approximation to first order derivative satisfying previously stated conditions is

    $G_x = (z_8 - z_5)$ and $G_y = (z_6 - z_5)$
  - Other definitions available including one proposed by Roberts in 1965, uses "cross differences" and known as the Roberts cross gradient operators

    $G_x = (z_9 - z_5)$ and $G_y = (z_8 - z_6)$

## Gradient Approximations (1):

- **Roberts Cross Gradient Operator**
  - Implemented with the following masks

$G_x$

| | |
|---|---|
| −1 | 0 |
| 0 | 1 |

$G_y$

| | |
|---|---|
| 0 | −1 |
| 1 | 0 |

  - Difficult to implement given its "awkward" size
    - Minimum mask we are interested in is 3x3!
    - Approximation using a 3x3 mask can be given

$$G[f(x,y)] = |(z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)| + |(z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)|$$

## Gradient Approximations (2):

- **Roberts Cross Gradient Operator (cont…)**
  - Implemented with the following masks

$G_x$

| | | |
|---|---|---|
| −1 | −2 | −1 |
| 0 | 0 | 0 |
| 1 | 2 | 1 |

$G_y$

| | | |
|---|---|---|
| −1 | 0 | 1 |
| −2 | 0 | 2 |
| −1 | 0 | 1 |

These two masks are known as Sobel operators

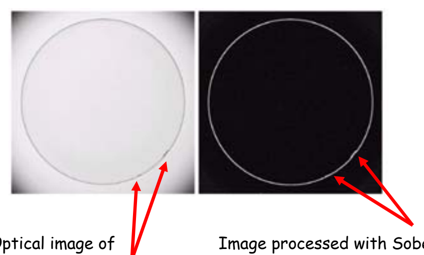
  - Difference between third and first rows approximates derivative in x direction
  - Difference between third and first columns approximates derivative in y direction

## Gradient Applications (1):

- **Many Applications and Uses**
  - Industrial applications
    - Aid humans in detecting defects → enhances defects and eliminates slowly changing background features
    - Pre-processing step in automated inspection
  - Edge detection
  - Highlight small specs not visible in gray scale image
  - Enhance small discontinuities in flat gray field

## Gradient Applications (2):

- **Example of the Gradient Operator**



Optical image of contact lens with defects at the boundary

Image processed with Sobel operator → edges revealed, background eliminated and defects are more visible now

# Second Order Derivatives The Laplacian Operator

## Introduction (1):

- **2D, Second Order Derivative Operator**
  - Basic approach
    - Define some discrete formulation for the second derivative
    - Using this formulation, define a filter mask (template etc.)
  - Isotropic filters → rotation invariant filters
    - Filter response independent of the direction of discontinuity
    - Rotating image and applying filter yields same results!

## Introduction (2):

- **Simplest Isotropic Derivative Operator is the Laplacian, Defined for Image f(x,y) as**

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

  - Laplacian is a linear operator
    - Derivatives of any order are linear operators
  - Above expression is of course formulated in a continuous form
    - Must "convert" to discrete form if it is to be of any use for image processing

## Defining the Discrete Laplacian (1):

- **Several Ways to Define a Discrete Laplacian Using Neighborhoods**
  - Must however satisfy the second order derivative properties previously described
  - Recall second order derivative previously given

$$\frac{\partial^2 f}{\partial x^2} = f(x+1) + f(x-1) - 2f(x)$$

  - We will basically "expand" on this formulation to account for both spatial variables x,y

## Defining the Discrete Laplacian (2):

- **Defining a Discrete Laplacian (cont…)**
  - Partial second order (discrete) derivative in the "x" direction defined as

$$\frac{\partial^2 f}{\partial x^2} = f(x+1,y)+f(x-1,y)-2f(x,y)$$

  - Partial second order (discrete) derivative in the "y" direction defined as

$$\frac{\partial^2 f}{\partial y^2} = f(x,y+1)+f(x,y-1)-2f(x,y)$$

## Defining the Discrete Laplacian (3):

- **Defining a Discrete Laplacian (cont…)**
  - By summing the x,y components, we obtain the digital implementation of the 2D Laplacian

$$\nabla^2 f = [f(x+1,y)+f(x-1,y)+f(x,y+1)+f(x,y-1)]-4f(x,y)$$

  - Can be implemented using the following mask (kernel)
  - Isotropic results for rotations in multiples of 90° only!
    - In other words, diagonal directions ignored!

| | | |
|---|---|---|
| 0 | 1 | 0 |
| 1 | -4 | 1 |
| 0 | 1 | 0 |

## Defining the Discrete Laplacian (4):

- **Defining a Discrete Laplacian (cont…)**
  - Diagonal directions can however be incorporated by adding two more terms, one for each of the two diagonal directions
  - Can be implemented using the following mask (kernel)
  - Isotropic results for rotations in multiples of 45° only!

| | | |
|---|---|---|
| 1 | 1 | 1 |
| 1 | -8 | 1 |
| 1 | 1 | 1 |

## Defining the Discrete Laplacian (5):

- **Defining a Discrete Laplacian (cont…)**
  - A "negative version" of the Laplacian definition is also available in which the coefficients of the mask are negative of the ones given
    - Yields the same results

| | | |
|---|---|---|
| 0 | −1 | 0 |
| −1 | 4 | −1 |
| 0 | −1 | 0 |

| | | |
|---|---|---|
| −1 | −1 | −1 |
| −1 | 8 | −1 |
| −1 | −1 | −1 |

## Defining the Discrete Laplacian (6):

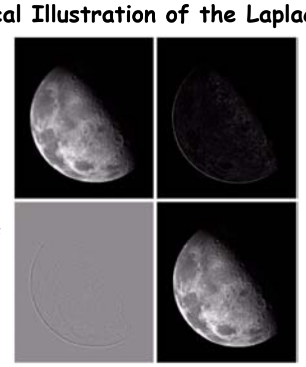
- **Laplacian in Practice**
  - Since it is a derivative operator, it highlights gray level discontinuities and deemphasizes regions with slow varying gray levels
    - Produces images that have grayish edge lines and other discontinuities on featureless background
  - Typically, add (or subtract if negative version of mask used) the Laplacian output image to the original input image
    - Recover the background
    - Preserve sharpening effect of the Laplacian

## Defining the Discrete Laplacian (7):

- **Graphical Illustration of the Laplacian**

Original image – north pole of moon

Applying the Laplacian filter – contains both positive and negative values!



Scaled by taking absolute value of previous image to eliminate negative values – not really "correct"!

Laplacian and original image added together

## Defining the Discrete Laplacian (8):

- **Laplacian in Practice – Simplifications**
  - Can incorporate the two steps of performing the Laplacian and adding results to the original image using a single mask

| | | |
|---|---|---|
| 0 | −1 | 0 |
| −1 | 5 | −1 |
| 0 | −1 | 0 |

Diagonal directions ignored

| | | |
|---|---|---|
| −1 | −1 | −1 |
| −1 | 9 | −1 |
| −1 | −1 | −1 |

Diagonal directions emphasized