ID 251
Digital Image Processing Techniques

07 / 28 / 2015

Instructor: Michael Eckmann
Today’s Topics

• A few quick Programming in Java concepts
• Walk through some of the code of the programs from last time
• Cross-correlation and filters
• Edge masks
• Smoothing masks (and median smoothing)
Java Programming

• A variable name can store a single value of a certain type (kind)
  – Our code uses several types including:
    • int (numeric integer values, e.g. -83)
    • String (words or names, e.g. “Mike”)
    • RGBPixel (3 integer values, a Red, Green and Blue)
    • RGBPixel[][] (a 2 dimensional grid of RGBPixels which holds an entire image)
Java Programming

• Our code uses assignment statements to assign (give) a value on the right to the variable on the left. The right side could be an expression that needs to be computed.

  – e.g.

```java
int size;       // size is an integer variable
int width = 16; // width is an integer variable whose value is 16
size = 5 * width; // 80 is stored in the variable size
```
Java Programming

• Our code uses loops for code that is to be repeated

• e.g.

    for (int i=0; i<10; i++)
    {
        System.out.println(i);
    }
Our code uses if /else statements to decide whether to do one set of code or another, depending on whether something is true or not. e.g.

```java
int width = 16; // width is an integer variable whose value is 16
String sentence; // sentence will hold one of two sentences
if (width < 10)
{
    sentence = “It is not very wide”; 
}
else
{
    sentence = “It is wide”; 
}
System.out.println(sentence);
```
Java Programming

• Let's look at a simple program that just has two loops, one nested inside the other that shows how these loops operate.
Java Programming

• Now let's look at the code and I'll walk us through how it works. We'll reexamine the output images and then we'll look closely at the code that created them.
Filters / Cross-correlation

- A mask (aka filter) is usually a small square grid of values
- Cross-correlation is a procedure that applies the mask to an image in a certain way.
- Let's consider grayscale images only here
- Given a mask of values, multiply each value in the mask by the corresponding pixel value in the image and add them up
  - Divide by total of the mask values (if they don't add up to 0 or 1)
Derivative and Smoothing masks

• Derivative masks
  – values have opposite signs in order to obtain a high response (high absolute values) in signal regions of high contrast
  – sum of values is 0, so a 0 results from constant regions

• Smoothing masks
  – values are positive and sum to 1, so resulting value is same as input in constant regions
Let's look at an example

• Let's consider this mask
  
  \[ \begin{array}{ccc}
  1 & 0 & -1
  \end{array} \]

• Applied to this 1 dimensional “image”:
  
  \[ \begin{array}{cccccccccccc}
  50 & 50 & 50 & 50 & 50 & 75 & 75 & 75 & 75 & 75 & 75
  \end{array} \]
Let's look at an example

• Let's consider this mask
  
  1  0  -1

• Applied to this 1 dimensional “image”:
  
  50  50  50  50  50  125  125  125  125  125  125

• Resulting “image”:
  
  0  0  0  -75  -75  0  0  0  0
Let's look at an example

• Let's consider this mask
  
  1   0   -1

• Applied to this 1 dimensional “image”:
  
  50   50   50   50   50   125  125  125  125  125

• Resulting “image”:
  
  0    0    0    -75   -75    0    0    0    0

• Notice: in areas of constant values, the result is 0, but in areas where there are differences/high contrast, the result is non-zero (high absolute value).