# EECS10 DISCUSSION

7/27/12 | Week5 Session2 | Weiwei Chen

---

## Assignment 4 Feedback

2

- Good job!
- Some feedbacks
  - How to use "return"
  - How to use functions
  - Multiple random number generations
- Please check the solution on the course webpage
- Final Exam: 9am~11am July 31$^{st}$, 2012
  - Same format as midterms
  - More questions

## Assignment 5

3

- A manual driven digital image processing (DIP) program.
- Using function calls for image inputing / outputing, image processing, and testing.
  - Function declaration, function definition, function call
  - Function parameters, argument.
  - Scope of the variables.
- One-week assignment. Plan the schedule of your work. Start it early!
  - Lab1: Setup the working environment. Design the user menu. Build up the frame of the operation functions. Try 1~2 operations on the image?
  - Lab2: Complete the operations. Test your program?
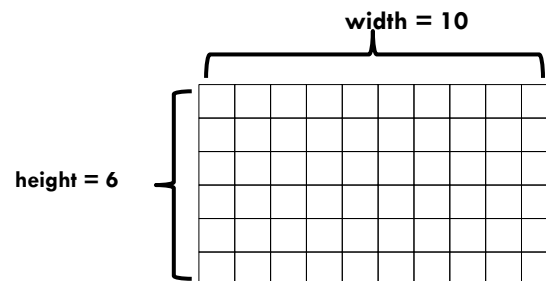- Use the web browser to view your image.

(c) W.Chen EECS UC Irvine                                      7/27/12

## Images (Pictures) in the computer

4

- How to represent a picture in computer:
  - A picture is composed of pixels
  - One color for each pixel



width = 10

height = 6

(c) W.Chen EECS UC Irvine                                      7/27/12

# RGB Color Model

- Three components for one color
- 3-tuple (R, G, B)
  - R: intensity of red
  - G: intensity of green
  - B: intensity of blue
  - Basically, the range of the intensity is [0, 255], use **unsigned char** to for each intensity
- Color Examples
  - Red (255, 0, 0), Green(0,255,0), Blue(0,0,255),
  - Yellow(255,255,0), Cyan(0,255,255), Magenta(255,0,255)
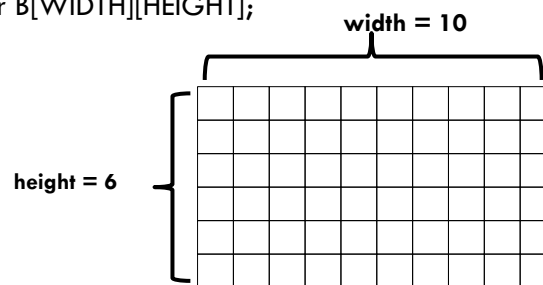  - White(255,255,255), Black(0,0,0)

# How to operate a picture

- The data structure to represent a picture in computer
  - Two-dimensional arrays for the intensities of each pixel
    - unsigned char R[WIDTH][HEIGHT];
    - unsigned char G[WIDTH][HEIGHT];
    - unsigned char B[WIDTH][HEIGHT];

width = 10

height = 6

# Pause and reflection

- ☐ What are the colors of the pixels in the image which is represented as following in the computer
  - ◻ unsigned char R[3][2]  = {{0    , 0    }, {255, 255}, {255, 0}};
  - ◻ unsigned char G[3][2] = {{255, 0    }, {255, 0    }, {255, 0}};
  - ◻ unsigned char B[3][2]  = {{0    , 255}, {0    , 0    }, {255, 0}}
- ☐ Color Examples
  - ◻ **Red (255, 0, 0), Green(0,255,0), Blue(0,0,255),**
  - ◻ **Yellow(255,255,0), Cyan(0,255,255), Magenta(255,0,255)**
  - ◻ **White(255,255,255), Black(0,0,0)**

(c) W.Chen EECS UC Irvine                                            7/27/12

---

# How to operate a picture

- ☐ Use two for-loops to scan all the pixels in a picture
  - ◻ Inner loop: fix the number of the column, iterate the pixels in the same column with different row numbers
  - ◻ Outer loop: iterate all the columns.

```
int x, y;
for (x= 0; x < WIDTH; x ++){
    for(y=0; y<HEIGHT; y++){
        operations for pixel(x,y);
    }
}
```

(c) W.Chen EECS UC Irvine                                            7/27/12

# DIP Operations

**9**

□ Black & White
  - ◘ Get the average value of the three color channels for each pixel (x,y).
  - ◘ Set R[x][y], B[x][y] and G[x][y] to be the average value.
□ Negative
  - ◘ Subtract R[x][y], B[x][y] and G[x][y] from 255 and set the new value back.

(c) W.Chen EECS UC Irvine                                    7/27/12

# DIP Operations

**10**

□ Flip horizontally
  - ◘ Swap pixel (x,y) and pixel (width-1-x, y)
  - ◘ Scan half of the picture

□ Flip vertically?
  - ◘ Swap pixel (x,y) and pixel (x, height - 1 - y)
  - ◘ Scan half of the picture

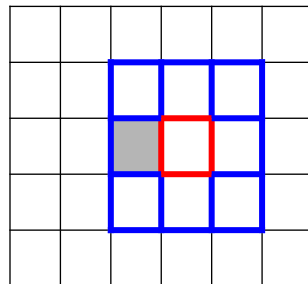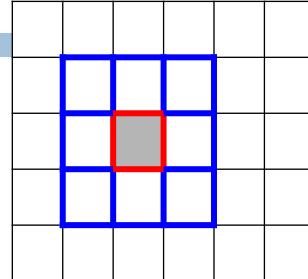(c) W.Chen EECS UC Irvine                                    7/27/12

# DIP Operations

11

□ Blur
- A pixel has 8 neighbors
- Get the average values of the three channels of the current pixel and its 8 neighbors'.
- Set the pixel's color components to the average values respectively.
- In order not to contaminate the original value of the picture, use temporary arrays for computation and copy the result back to the original arrays.
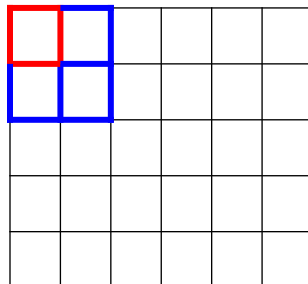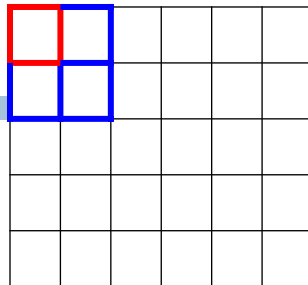
(c) W.Chen EECS UC Irvine

# DIP Operations

12

□ Blur
- Pixels on the corners and the edges.
- Have fewer neighbors
- Handle separately
- Ignore pixels on the edges

(c) W.Chen EECS UC Irvine
7/27/12

## DIP Operations

13

□ Mirror horizontally
- copy pixel (x,y) to pixel (width-1-x, y)
- Scan half of the picture

□ Mirror vertically?
- copy pixel (x,y) to pixel (x, height - 1 - y)
- Scan half of the picture

(c) W.Chen EECS UC Irvine

7/27/12

## Assignment5 Tips

14

□ Test your program
- AutoTest() function
- Call all the other operation functions together in the program.
- Be careful with the arguments for each functions.
- Sample function calls are listed in the assignment.
□ Global constants
□ Scope of the variables
□ Pass by reference when using array parameters.
□ Function prototypes mentioned in the assignment are very helpful hints.

(c) W.Chen EECS UC Irvine

7/27/12

# Farewell

**15**

- □ Thank you very much for session!
- □ It is a great pleasure to work with you all!
- □ Good luck for the finals!

7/27/12