

EECS10 Discussion Week7

TA: Emad Arasteh
emalekza@uci.edu
eeecs10@eeecs.uci.edu

Office Hours: Fri, 8:00-9:00am EH 3404

University of California, Irvine



Functions

- Scope of an identifier
 - Portion of the program where the identifier can be referenced
 - aka. accessibility, visibility
- Scope rules
 - Global variables: file scope
 - Declaration outside any function (at global level)
 - Scope in entire source file after declaration
 - Function parameters: function scope
 - Declaration in function parameter list
 - Scope limited to this function body
 - Local variables: block scope
 - Declaration inside a compound statement (i.e. inside for loop)
 - Scope limited to this compound statement block

Arrays

- A data structure consisting of a collection of elements
- Array of int : `int A[100]`
- Fixed number of elements at compile time (e.g. 100 elements)
- Element access by index (array[index] e.g. `A[42]`)
- Array indexing starts counting from 0
- Multidimensional array: array of an array e.x. `int B[5][7]`

Passing arguments to functions

- Pass by value
- Pass by reference
- In ANSI C, ...
 - ... basic types are passed by value
 - ... arrays are passed by reference

Assignment 7

- A manual driven digital image processing program
- Using function calls for image file handling, image processing, and testing.
 - Function declaration, function definition, function call
 - Function parameters
 - Scope of the variables
- Two-week assignment. Plan and start early!
 - Week1: Setup the working environment. Design the user menu. Try 1~2 operations on the image.
 - Week2: Complete the operations. Test your program.
- Use the web browser to view your image.

Pixels

- How to represent an image in digital computers:
 - An image is composed of picture elements aka pixels

pixel



RGB color components

- Three components (R, G, B) are used to represent one pixel:
 - R: intensity for red color
 - G: intensity for green color
 - B: intensity for blue color
- The range of intensity for each color component in the 'library' image is values between [0 to 255] (8-bit). Therefore, we use **char** type to store these values.
- 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)

Image size

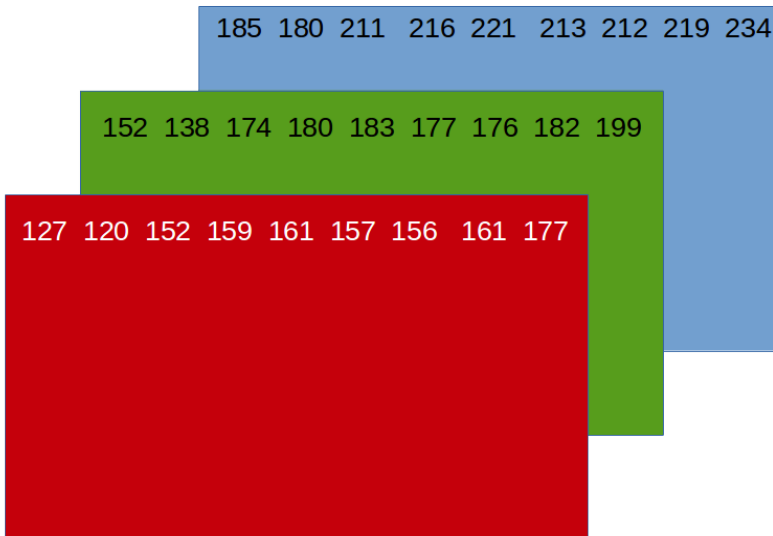
- Example image for Assignment 7 is UCI Science Library
- Size of image is (640 x 480) as (width x height)



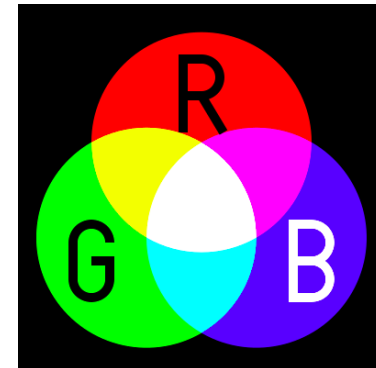
RGB colors

- A 2-dimensional array defines intensity of each color component

```
unsigned char R[WIDTH][HEIGHT];  
unsigned char G[WIDTH][HEIGHT];  
unsigned char B[WIDTH][HEIGHT];
```



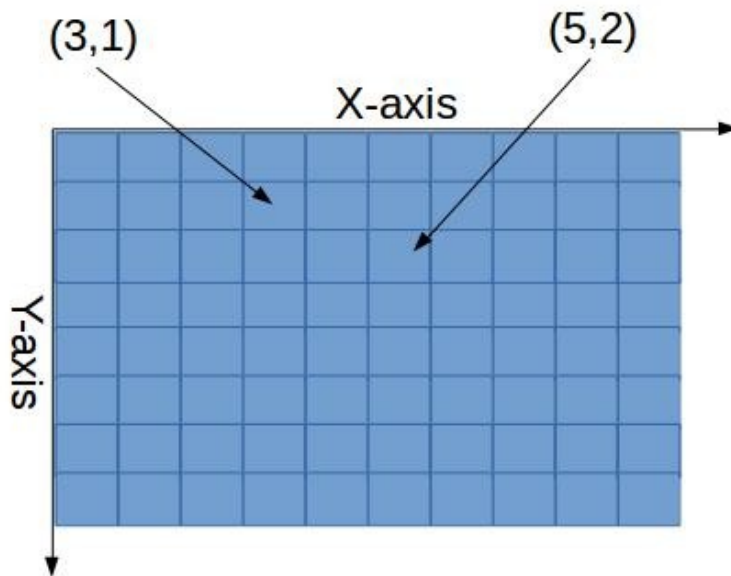
Colors intensities for red, green, blue colors taken from 'library.ppm' image by E. Arasteh



Additive color mixing [Wikipedia]

How to manipulate an image

- First, how to access every pixel in an image?
 - By coordinate of a pixel (x, y) , x is coordinate on the X-axis and y is the coordinate on Y-axis
 - The color tuple of the pixel at coordinate (x, y) is :
 $(R[x][y], B[x][y], G[x][y])$



How to manipulate an image

- You can use nested for loops to manipulate pixels of an image:

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