EECS10 DISCUSSION

7/25/12 | Week5 Session1 | Weiwei Chen

Muddiest Card Feedback

- Data structures
  - Differences between **struct** and **union**
- How to use *for* loops
- Recursion
- Arrays
- How to use functions
  - Declaration, definition, function calls
  - How to use functions in large program
  - Pass by reference vs. pass by value
  - Function parameters vs. arguments, local variables
  - Function return types
- How to study for the exams
- More on Homework

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.

Images (Pictures) in the computer

4

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

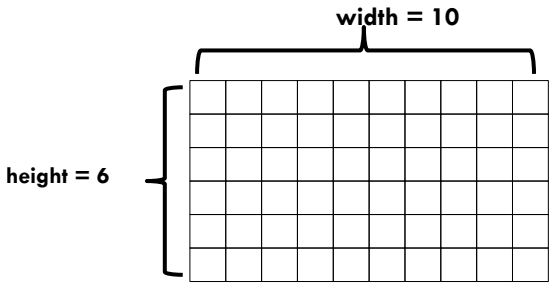


Image with pixels

5

RGB Color Model

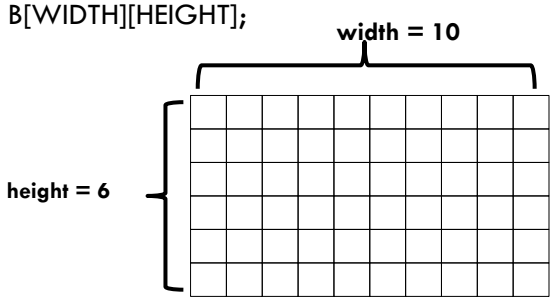
6

- 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

7

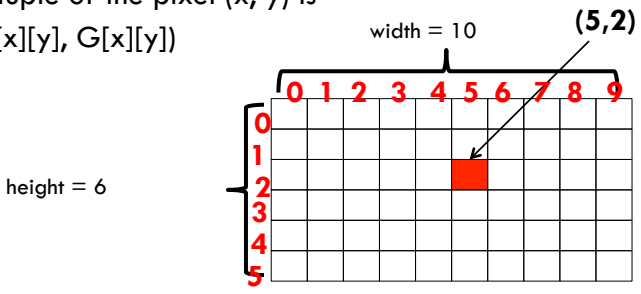
- The data structure to represent a picture in computer
  - Two-dimensional arrays for the intensities of each pixel
    - unsigned char R[WIDTH][HEIGHT];
    - unsigned char G[WIDTH][HEIGHT];
    - unsigned char B[WIDTH][HEIGHT];



How to operate a picture

8

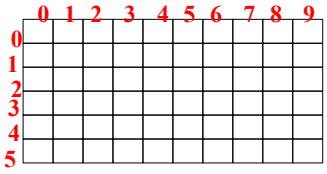
- How to access every pixels in a picture
  - Coordinate of a pixel (x, y),
    - **x** is the number of the column
    - **y** is the number of the row
  - The color tuple of the pixel (x, y) is (R[x][y], B[x][y], G[x][y])



How to operate a picture

9

- How to access every pixels in a picture in C
  - List all the possible coordinates of a pixel
    - (0,0), (1,0), (2,0), …, (9,0)
    - (0,1), (1,1), (2,1), …, (9,1)
    - …
    - (0,5), (1,5), (2,5), …, (9,5)



How to operate a picture

10

- 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);
      }
}
```

DIP Operations

11

- 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
  - Substract R[x][y], B[x][y] and G[x][y] from 255 and set the new value back.
- Flip horizontally
  - Swap pixel (x,y) and pixel (width-1-x, y)
  - Scan half of the picture

DIP Operations

12

- Mirror horizontally
  - Copy pixel (x, y) to pixel (width-1-x,y)
  - Scan half of the picture.
- Flip Vertically, Mirror Vertically

Assignment5 Tips

- 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.

EECS10 LABORATORY

7/25/12 Week5 Session1 Weiwei Chen

It is a time for programing!

15

- Raise your hand if you need help

Pointers

16

- Pointers are variables whose values are addresses
  - The "address-of" operator (&) returns a pointer!
- Pointer Definition
  - The unary * operator indicates a pointer type in a definition
- Pointer initialization or assignment
  - A pointer may be set to the "address-of" another variable
  - A pointer may be set to 0 (points to no object)
  - A pointer may be set to NULL (points to "NULL" object)

Pointers

- Pointer Dereferencing
  - The unary * operator dereferences a pointer to the value it points to ("content-of" operator)
  - The -> operator dereferences a pointer to a structure to the content of a structure member