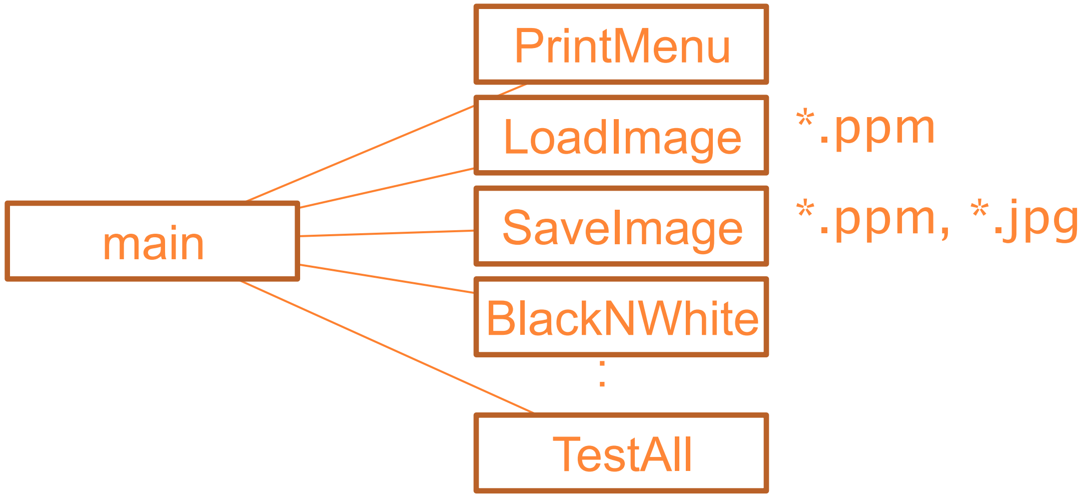
EECS 22: Advanced C Programming
Week 3

Saeed Karimi Bidhendi
skarimib@uci.edu

10/10/2017

Assignment 2

- A menu driven digital image processing program [100 pts]
- Deadline : 2017/10/25, Wednesday, 6:00 pm
- Goal
  - Main function uses function calls to input/output image, process image, and test all of the digital image process functions.



Menu Driven Digital Image Processing

```
eecs22@zuma.eecs.uci.edu: > ./PhotoLab
--------------------------------
 1:  Load a PPM image
 2:  Save an image in PPM and JPEG format
 3:  Change a color image to Black & White
 4:  Make a negative of an image
 5:  Color filter an image
 6:  Sketch the edge of an image
 7:  Shuffle an image
 8:  Flip an image vertically
 9:  Mirror an image vertically
10:  Add Border to an image
11:  Test all functions
12:  Exit
please make your choice:
```

Input File

- Format : ppm
- Option 1: input ppm file
  - `Load a PPM image`
  - example 1:

```
please make your choice: 1
Please input the file name to load: HSSOE
HSSOE.ppm was read successfully!
```

  - File extension is not needed.
  - example 2:

```
please make your choice: 1
Please input the file name to load: HSSOE.ppm
Cannot open file ”HSSOE.ppm.ppm" for reading!
```

  - Function for loading image `LoadImage` is provided !

Output File

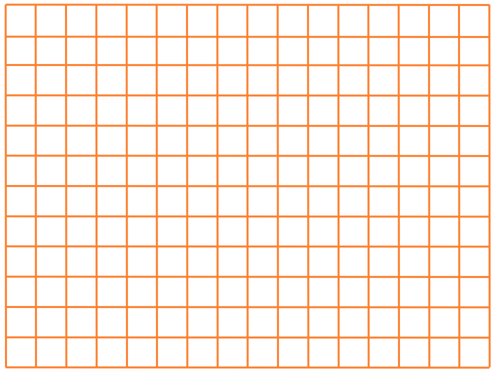
- Format : ppm, jpg
- Option 2: output ppm and jpg files
  - `Save an image in PPM and JPEG format`
  - example:

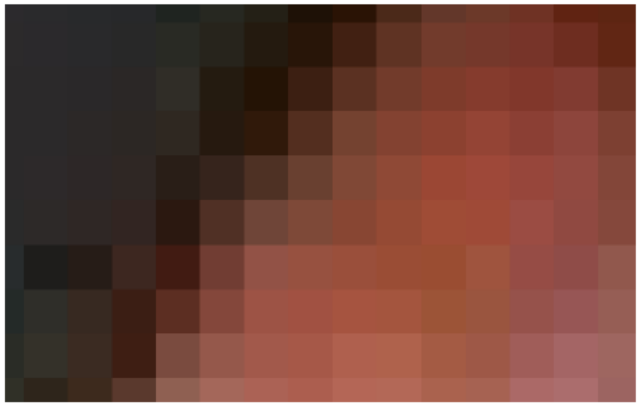
```
Please make your choice: 2
Please input the file name to save: negative
negative.ppm was saved successfully.
negative.jpg was stored for viewing.
```

  - File extension is not needed.
  - Function for saving image `SaveImage` is provided !

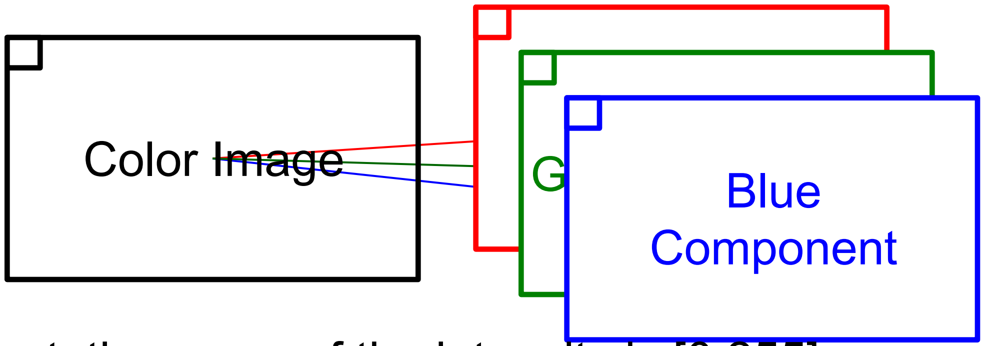
Picture in the program

- How to represent a picture in computer?
  - A picture is composed of pixels
  - One color for each pixel
  - Example: 16x12 = 192 pixels

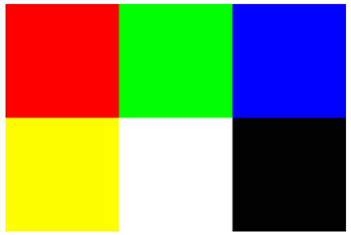




Picture in the program



- 3-tuple (R, G, B)
  - R: intensity of Red
  - G: intensity of Green
  - B: intensity of Blue
  - For image in ppm format, the range of the intensity is [0,255], using unsigned char 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)
- PPM example
- RGBRGBRGBRGB…
-
```
P3    (colors)
3 2   (3 columns, 2 rows)
255   (255 for max color)
255    0  0          0 255    0       0  0 255
255  255  0        255 255  255       0  0   0
```

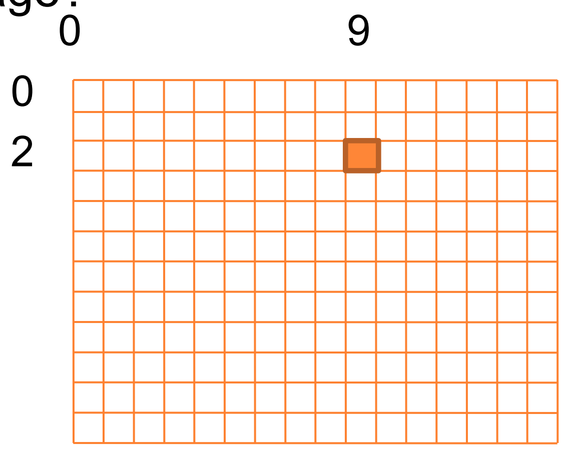


Picture in the program

- The data structure to represent a picture in this assignment
  - Two-dimensional arrays for the intensities of each pixel
    - For an image of size 16x12…

```
unsigned char R[16][12];
unsigned char G[16][12];
unsigned char B[16][12];
```

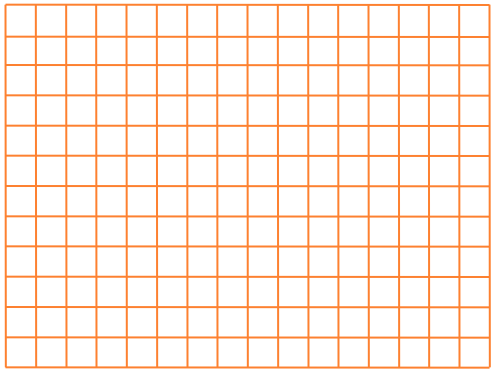
    - How to access a pixel in an image?
      - Coordinate of a pixel (x, y)
      - x = number of the column
      - y = number of the row
      - The color of the pixel (x, y) = (R[x][y], G[x][y], B[x][y])



Picture in the program

- How to access every pixel in the picture?
  - List all possible coordinates of the pixel
  - Two for-loops to scan all the pixels in a 2-D array
  - Inner loop
    - fix the number of the column, iterate the pixel in the same column with different row numbers
  - Outer loop
    - iterate all the columns

```
int x, y ;
for (x=0; x < 16; x++){
 for (y=0; y < 12; y++){
  processing on pixel(x, y);
 }
}
```



Digital Image Processing Function

```
eecs22@zuma.eecs.uci.edu: > ./PhotoLab
--------------------------------
 1:   Load a PPM image
 2:   Save an image in PPM and JPEG format
 3:   Change a color image to Black & White
 4:   Make a negative of an image
 5:   Color filter an image
 6:   Sketch the edge of an image
 7:   Shuffle an image
 8:   Flip an image vertically
 9:   Mirror an image vertically
10:   Add Border to an image
11:   Test all functions
12:   Exit
please make your choice:
```

- Note: Your program should response "Image is not in the program yet" if the user want to choose option 3~9 before using option 1 to read the image.

Initial Setup

```
mkdir hw2
cd hw2
cp ~eecs22/public/PhotoLab.c .
cp ~eecs22/public/HSSOE.ppm .
```

Provided Function

```
#const int WIDTH  600       /* Image width */
#const int HEIGHT 400       /* image height */
#const int SLEN    80       /* maximum length of file names */

int main()
{
 /*
  * Two dimensional arrays to hold the current image data
  * One array for each color component
  */
    unsigned char   R[WIDTH][HEIGHT];
    unsigned char   G[WIDTH][HEIGHT];
    unsigned char   B[WIDTH][HEIGHT];
/*  Please replace the following code with proper menu  */
/*  with function calls for DIP operations              */
    AutoTest(R, G, B);
/*  end of replacing*/

    return 0;
}
```

Provided Function

- Image Input / Output

  -
    ```
    int LoadImage (char fname[SLEN],
                   unsigned char R[WIDTH][HEIGHT],
                   unsigned char G[WIDTH][HEIGHT],
                   unsigned char B[WIDTH][HEIGHT]) ;
    ```

  -
    ```
    int SaveImage (char fname[SLEN],
                   unsigned char R[WIDTH][HEIGHT],
                   unsigned char G[WIDTH][HEIGHT],
                   unsigned char B[WIDTH][HEIGHT]) ;
    ```
  - Arguments are passed to the function by reference.

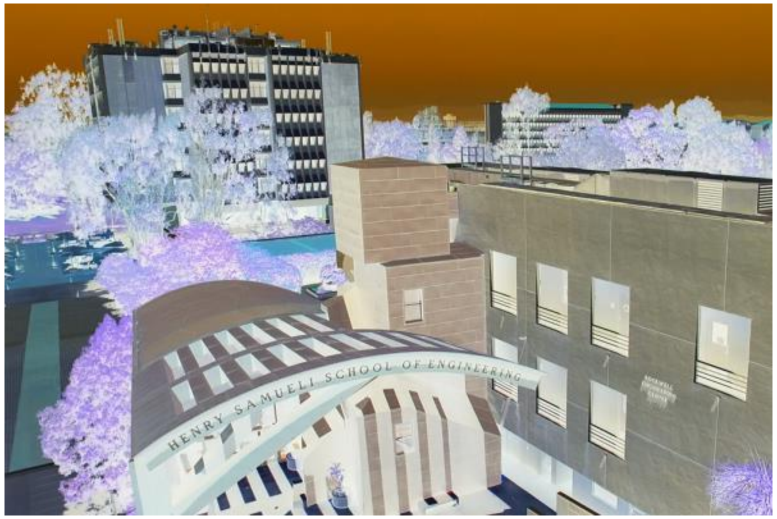
- Use `scanf("%79s", fname)` to input file name

Provided Function

- Aging function – as the sample of DIP function

  -
    ```
    void Aging(unsigned char R[WIDTH][HEIGHT],
               unsigned char G[WIDTH][HEIGHT],
               unsigned char B[WIDTH][HEIGHT])
    {
      int x, y;
      for( y = 0; y < HEIGHT; y++ )
        for( x = 0; x < WIDTH; x++ ) {
          B[x][y] = ( R[x][y]+G[x][y]+B[x][y] )/5;
          R[x][y] = (unsigned char) (B[x][y]*1.6);
          G[x][y] = (unsigned char) (B[x][y]*1.6);
      }
    }
    ```

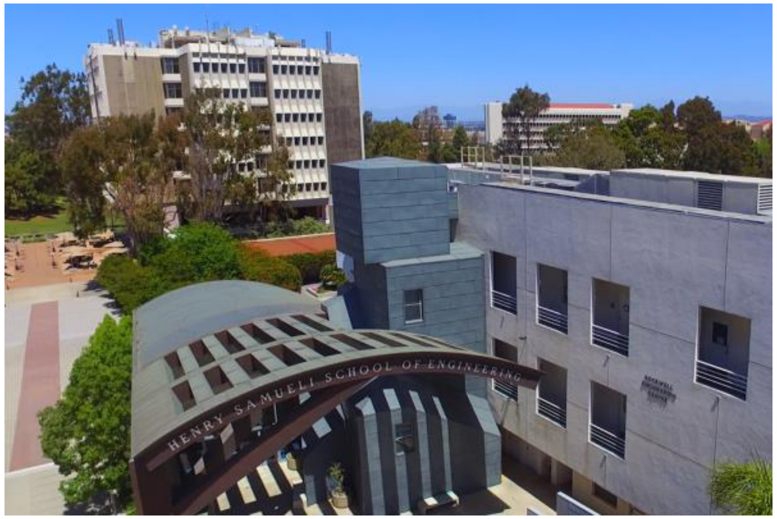
Negative

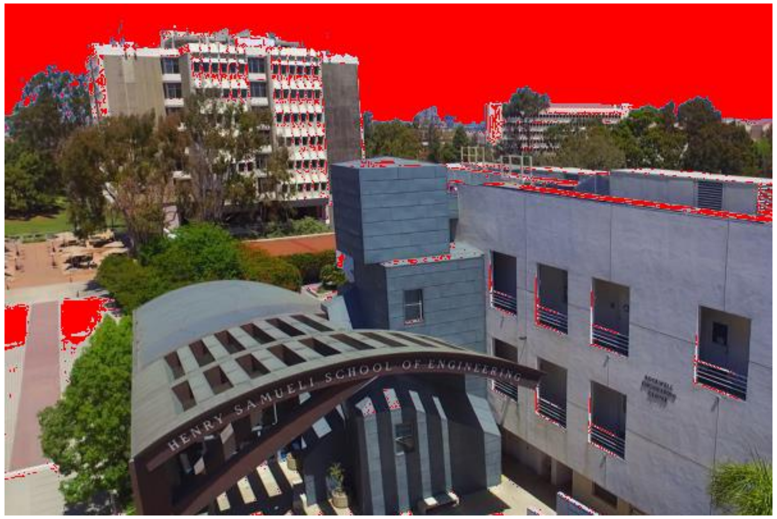


- Pseudo Code:

```
For all pixels in the picture, subtract the
current value from 255 which is the maximum
intensity value
```

Color Filter

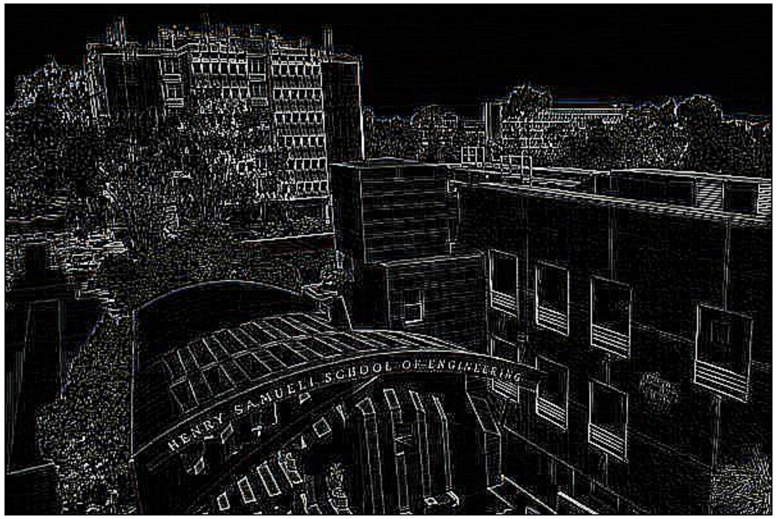




```
• For all pixels in the picture
  if (R in the range of [target_r – threshold, target_r + threshold]) and
     (G in the range of [target_g – threshold, target_g + threshold]) and
     (B in the range of [target_b – threshold, target_b + threshold])
        R = replace_r ;
        G = replace_g ;
        B = replace_b ;
  else
        keep the current color
```

```
target_r = 130   replace_r = 255
target_g = 130   replace_g = 0
target_b = 250   replace_b = 0
Threshold = 70
```

Edge



- Set the pixel's color at E with equation: new_E = 8*E – A – B – C – D – F – G – H – I
- Use temporary array to avoid computing with contaminated color intensities.
- Set border pixels (that have fewer neighbors) to black
- new_E should be in the range [0, 255]