EECS 22: Advanced C Programming
Week 6

Tim Schmidt
schmidtt@uci.edu

10/27/2016

Agenda

1. General Information
2. Makefile
3. Advanced DIP operations
4. Submission

Assignment 3

- A menu driven digital image processing program [100 points]
- Bonus: Watermark [10 points]
- Deadline: 2016/11/03, Thursday, 6:00 pm

- Goals
  - Decomposing the PhotoLab in multiple source and header files
  - Adding new DIP operations
    - Add Noise
    - Posterization
    - Shuffle
    - Watermark (bonus)
  - Having support of a debug flag

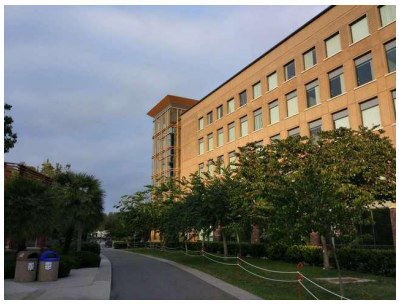
Advanced DIP Operations

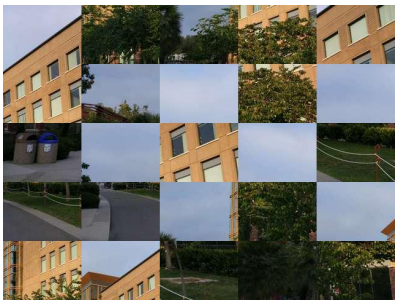
New Operations
- AddNoise
- Shuffle
- Posterize the image
- Watermark (bonus)

The menu looks like:

```
--------------------------------
1:  Load a PPM image
2:  Save an image in PPM and JPEG format
3:  Make a negative of an image
4:  Color filter an image
5:  Sketch the edge of an image
6:  Flip an image horizontally
7:  Mirror an image vertically
8:  Add Border to an image
9:  Zoom an image
10: Add noise to an image
11: Shuffle an image
12: Posterize an image
13: Watermark
14: Test all functions
15: Exit
please make your choice:
```

Shuffle





```
void Shuffle(
          unsigned char R[WIDTH][HEIGHT],
          unsigned char G[WIDTH][HEIGHT],
          unsigned char B[WIDTH][HEIGHT]);
```

- The image will be divided into 25 equally sized blocks

Shuffle

- Generate a sequence of 25 unique numbers in the range from 0 to 24
- Like: 17, 8, 19, 11, 16, 13, 12, 0, 14, 18, 6, 2, 15, 24, 9, 23, 10, 20, 22, 4, 21, 5, 1, 3, 7

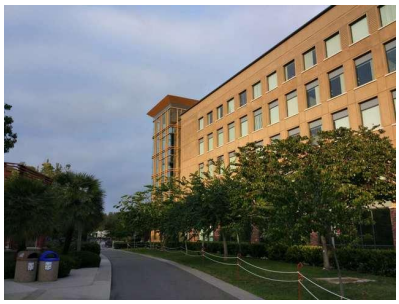
0	1	2	3	4
5	6	7	8	9
10	11	12	13	14
15	16	17	18	19
20	21	22	23	24

Original image with block indexing

17	8	19	11	16
13	12	0	14	18
6	2	15	24	9
23	10	20	22	4
21	5	1	3	7

Shuffled image

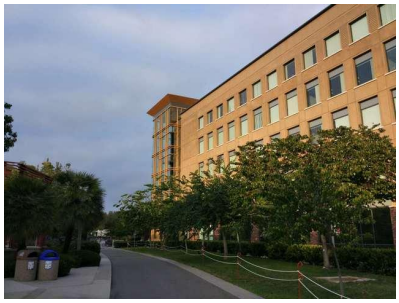
Posterize



```
void Posterize(
          unsigned char R[WIDTH][HEIGHT],
          unsigned char G[WIDTH][HEIGHT],
          unsigned char B[WIDTH][HEIGHT]
          int rbits, int gbits, int bbits);
```

- You need to use the bitwise operators
- Values are 5(rbits), 4(gbits), and 7(bbits)

Posterize



Function call Posterizse(R, G, B, 6, 5, 4);

	8th	7th	6th	5th	4th	3rd	2nd	1st		8th	7th	6th	5th	4th	3rd	2nd	1st
Posterize red with 6 bits	0	0	1	0	1	0	0	1	→	0	0	0	1	1	1	1	1
Posterize green with 5bits	0	1	0	1	0	1	0	0	→	0	1	0	0	1	1	1	1
Posterize blue with 4 bits	1	0	1	0	0	0	1	1	→	1	0	1	0	0	1	1	1

Posterize

Bit Operation

```
unsigned char mask8 = 0xFF;   ⇔ 1111 1111
unsigned char mask1 = 0x1;    ⇔ 0000 0001
unsigned char mask4 = 0xF;    ⇔ 0000 1111
```

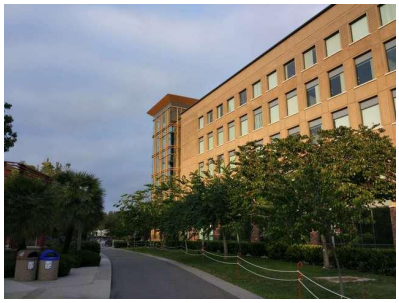
Shifting bits

```
mask4 << 1;   => 0001 1110
mask4 >> 1;   => 0000 0111
```

Combining bits:

```
mask1 & mask4; => 0000 0001
mask1 | mask4; => 0000 1111
```


Watermark



```
void Watermark(
        unsigned char R[WIDTH][HEIGHT],
        unsigned char G[WIDTH][HEIGHT],
        unsigned char B[WIDTH][HEIGHT]);
```

- watermark_template.ppm provides UCI letters
- Black pixels x,y provide coordinates for target pixels
- Multiply pixels with factor 1.45 in the target image

Submission

The submission should include these files

- PhotoLab.c
- PhotoLab.script
- PhotoLab.txt
- FileIO.c
- FileIO.h
- Constants.h
- DIPs.c
- DIPs.h
- Advanced.c
- Advanced.h
- Makefile

Submission

In the Photolab.script, we expect following content

1. Start the *script* by typing the command: *script*
2. Compile and run *PhotoLab* by using your ***Makefile***
3. Choose 'Test all functions' (The file names must be 'negative', 'colorfilter', 'edge', 'hflip', 'vmirror', 'zoom', 'border', 'noise', 'shuffle', 'posterize', and 'watermark' for the corresponding function).
4. Exit the *PhotoLab*.
5. Compile and run *PhotoLabTest*
6. Clean all the object files and executable programs by using your ***Makefile***
7. Stop the *script* by typing the command: exit
8. Rename the *script* file to *PhotoLab.script*