

(c) W.Chen EECS UC Irvine 1

EECS22 LAB WEEK4

10/23/12 Weiwei Chen

Feedback for the Concern Cards

2

- Confusing concepts
 - File I/O: Lecture 17; EECS22L projects
 - Multiple dimensional arrays: Lecture 4, 13; assignment 4
 - Pointers: Lecture 13, assignment 4
 - Function arguments: pass by value vs. pass by reference
 - Use of strings: Lecture 18
 - #include: Lecture 7, 8; assignment 3
 - More variable types: Lecture 11, 18; EECS22L projects
 - Text editors
 - Algorithm: Lecture 14, 15; assignment 5

(c) W.Chen EECS UC Irvine 10/23/12

Code Review Practice

3

- Code review (peer review) is systematic examination of computer source code.
 - Industry standard
 - find and fix mistakes overlooked in the initial development phase
 - Improving the overall quality of software and the developers' skills.
- Please provide 5 constructive critiques for the blackjack program
- The program can be accessed by:
 - `cp ~eecs22/codereview/blackjack.c ./`

(c) W.Chen EECS UC Irvine

10/23/12

Assignment 3

4

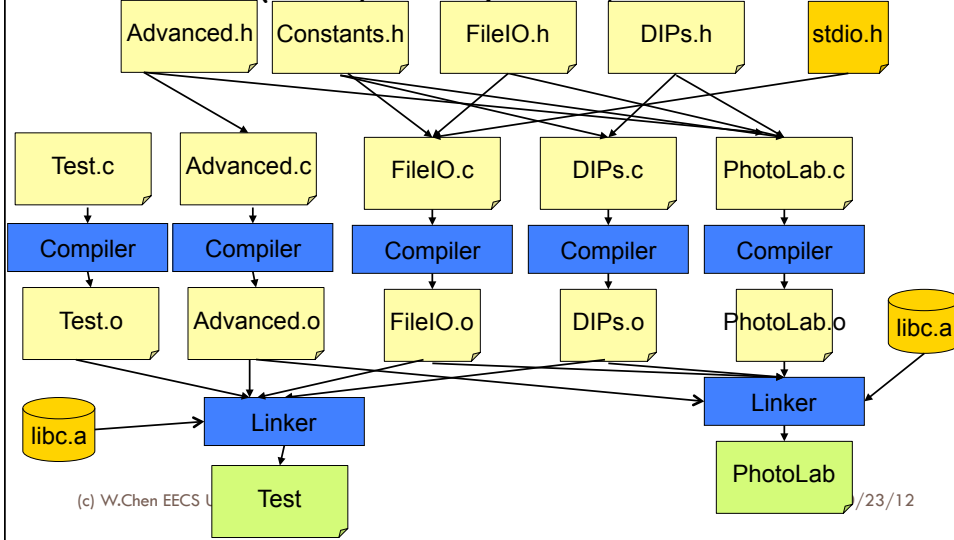
- Decompose a program into multiple modules
- Compile multiple modules into different programs
- Makefile Development
- DEBUG mode support
- Advanced Digital Image Processing Functions
- Bitwise manipulations

(c) W.Chen EECS UC Irvine

10/23/12

PhotoLab Modules

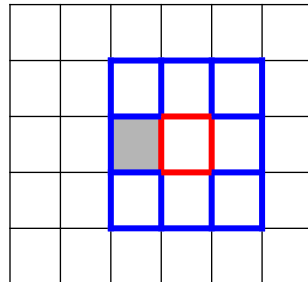
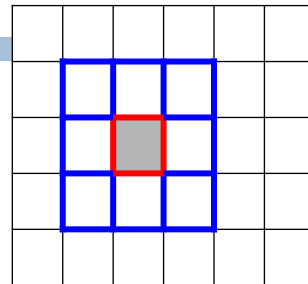
- 5 • PhotoLab (assignment2) decomposed into Modules



DIP Operations

6

- 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.

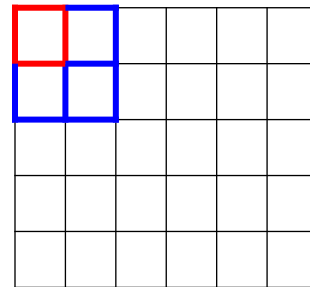
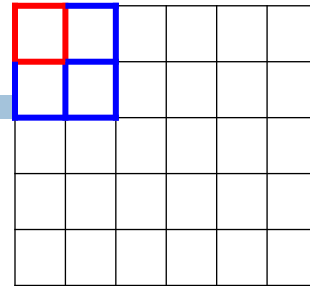


(c) W.Chen EECS UC Irvine

DIP Operations

7

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



(c) W.Chen EECS UC Irvine

10/23/12

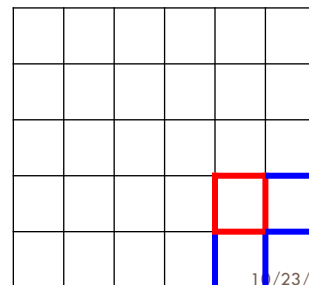
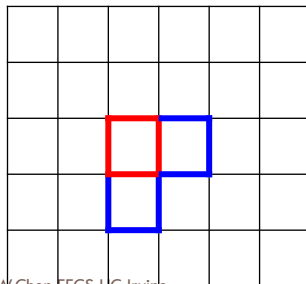
Edge Detection

8

- Color of two pixels $C1=(R1, G1, B1)$, $C2=(R2, G2, B2)$
- Color Difference

$$D(C1, C2) = \sqrt{(R1 - R2)^2 + (B1 - B2)^2 + (G1 - G2)^2}$$

- Compare pixel P and P_{right}, P and P_{bottom}
- If difference exceeds the threshold K, set P to be white (255, 255, 255); otherwise set P to be black(0, 0, 0)



(c) W.Chen EECS UC Irvine

10/23/12