IMAGE PROCESSING: POINT PROCESSES

N. C. State University

CSC557 ♦ Multimedia Computing and Networking

Fall 2001

Lecture # 11

IMAGE PROCESSING: POINT PROCESSES

N. C. State University

CSC557 ♦ Multimedia Computing and Networking

Fall 2001

Lecture # 11

Announcements, Questions, …

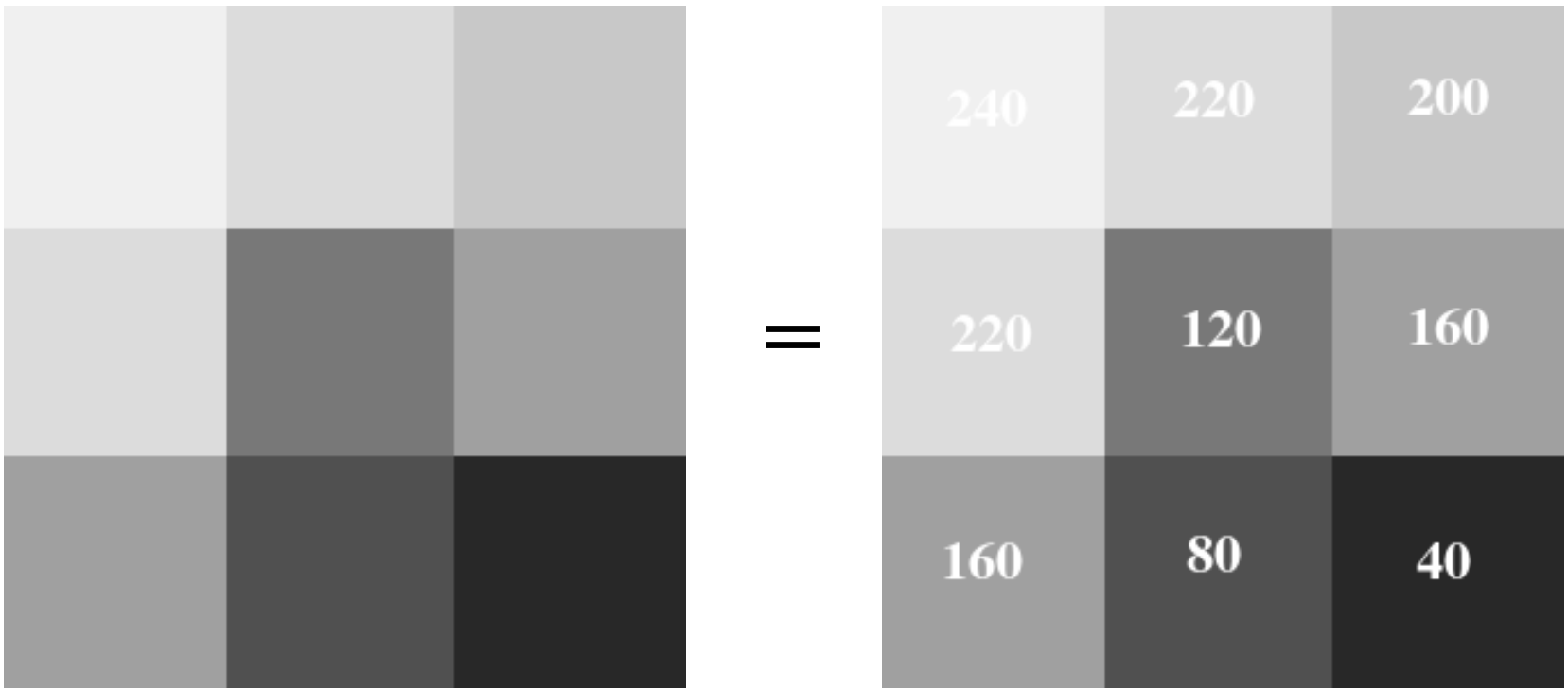
- ???

"Point" Operations

- Pixels are processed independently, one by one
- Examples:
  1. Change brightness and/or contrast of all pixels
  2. Convert color to grayscale (monochrome)

Example of Pixel Values

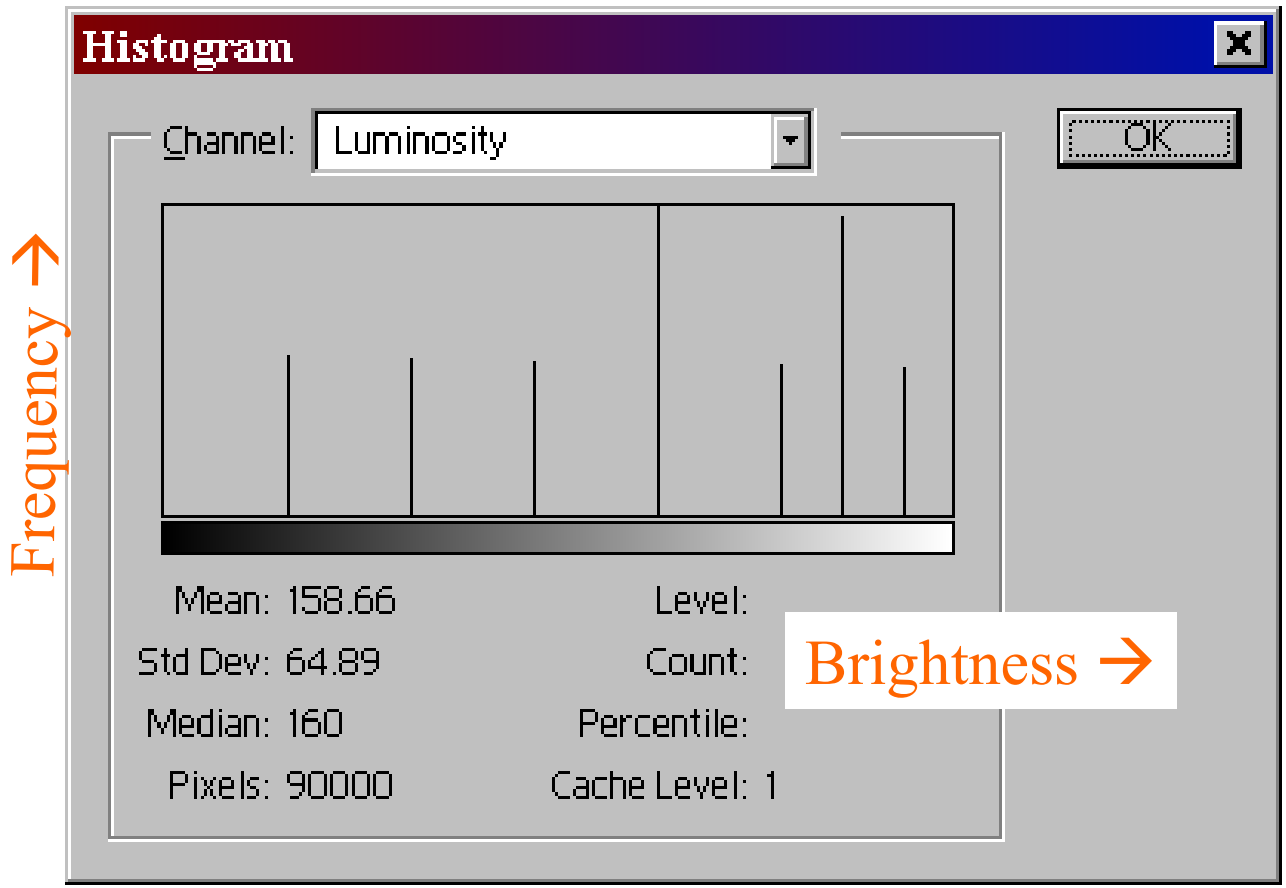
- Typically, pixel values range from 0…2^n-1, where n is the number of bits allocated
  - Most common: n=8, total number of colors = 256 (0…255)



Another Example

Histograms

- A *histogram* = a graph of brightness vs. the frequency of pixels with that brightness



Brightness Change

- Adding or subtracting a constant to all pixels
  - ”Shifts" the histogram to left or right
- Must “saturate” (or “clip”) at the maximum or minimum allowed values
  - Cannot exceed 255 or be less than 0

Old pixel value = 160

Add 40 to 160 to “brighten” pixel

result = 200 = new pixel value

Old pixel value = 220

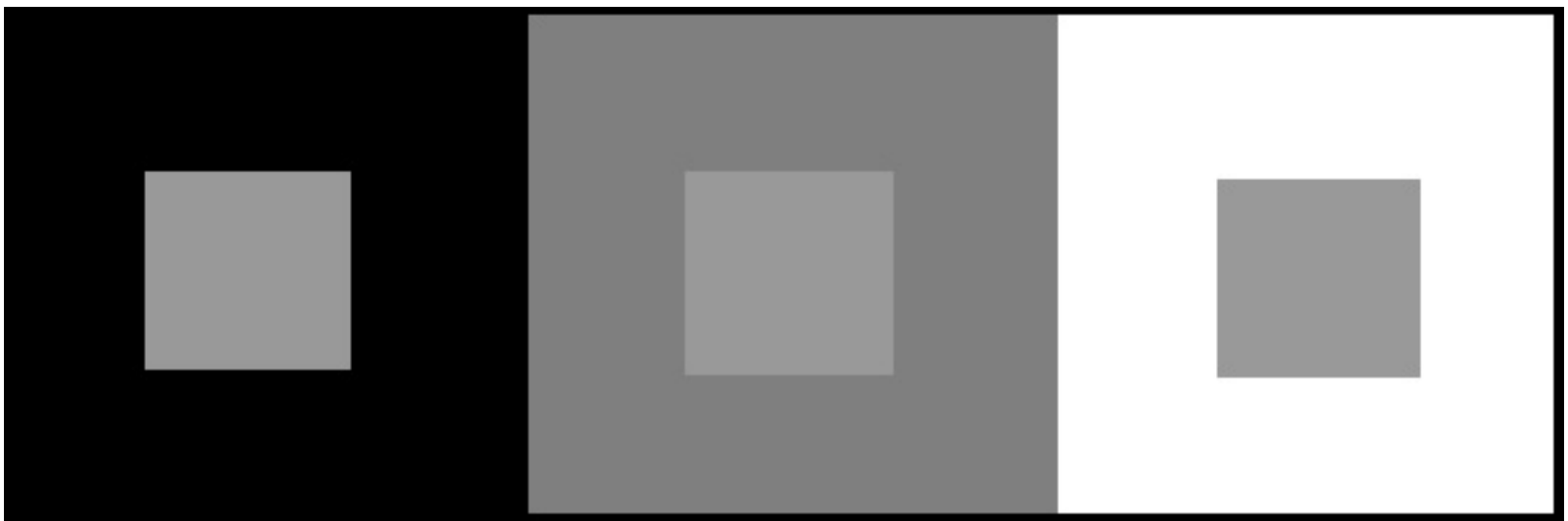
220 + 40 = 260, but maximum possible value is 255

result = 255 = new pixel value

Contrast Sensitivity

- Our eyes are sensitive to intensity differences
- Higher contrast seems to improve image detail
- *Contrast* = difference between pixel value and *average* pixel value

Which middle square below is the brightest?



Changing Contrast

- Equivalent to multiplying the difference to the median by a constant value
- → "Compression" or "expansion" of the histogram

Example (increasing, or doubling the contrast):

Median = 60

Old pixel value = 40

Difference to median = old pixel value – median = 40 – 60 = -20

New pixel value

= median + 2 * difference to median

= 60 + **2***(-20)

= 20

Doubling the contrast!

Intensity: Other

- Posterize = reduce # of possible pixel values
  - → coarser quantization scheme, fewer number of bits

Example (4 posterize levels):

new pixel values must be one of 0, 85 (=255/3), 170 (=2*255/3), 255

old pixel value = 60, convert to nearest of (0,85,170,255)

new pixel value = 85

- Threshold = convert to black or white

Example (threshold = 110):

old pixel value = 80 (less than threshold), new value = 0

old pixel value = 160 (greater than threshold), new value = 255

Other (cont.)

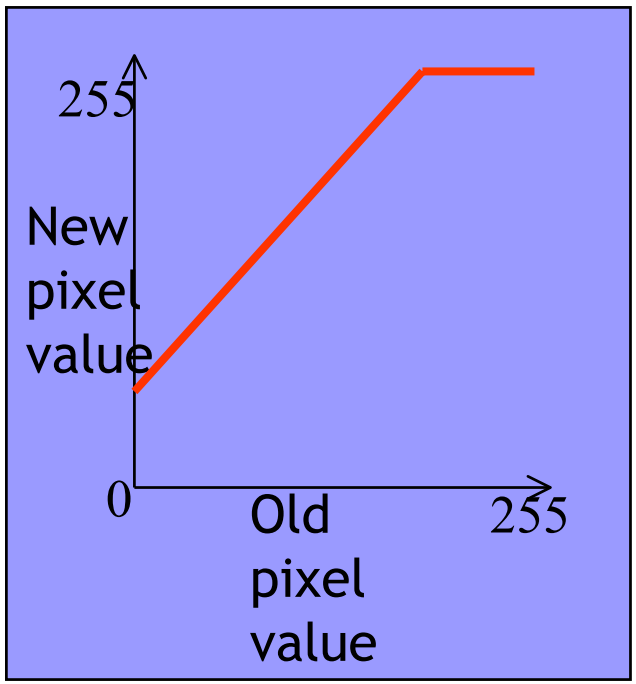
- Invert
  - Grayscale: Swap black for white, dark gray for light gray, etc.
  - Color: swap color for *opposite* color
- Looks like photographic negative

Example:

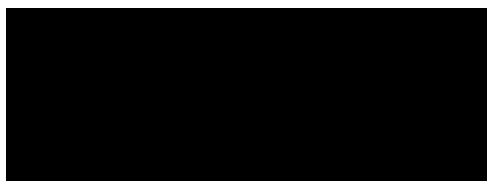
old pixel value = 60, new value = 255 – 60 = 195

"Dynamics"

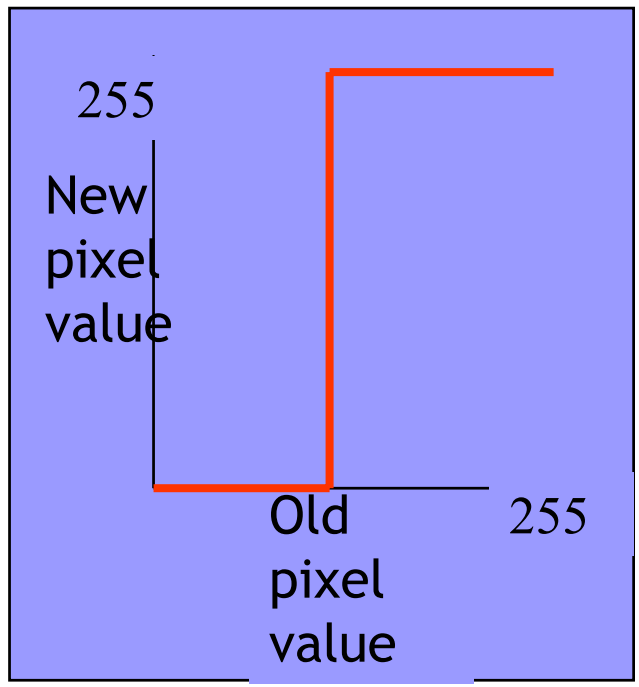
- Graphical way to express intensity transformations

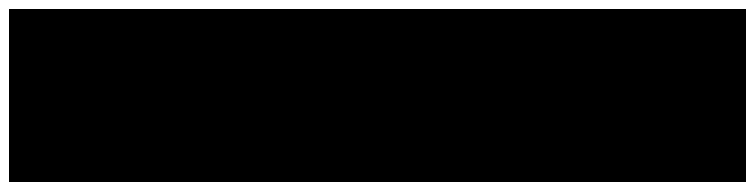


a. What is it?



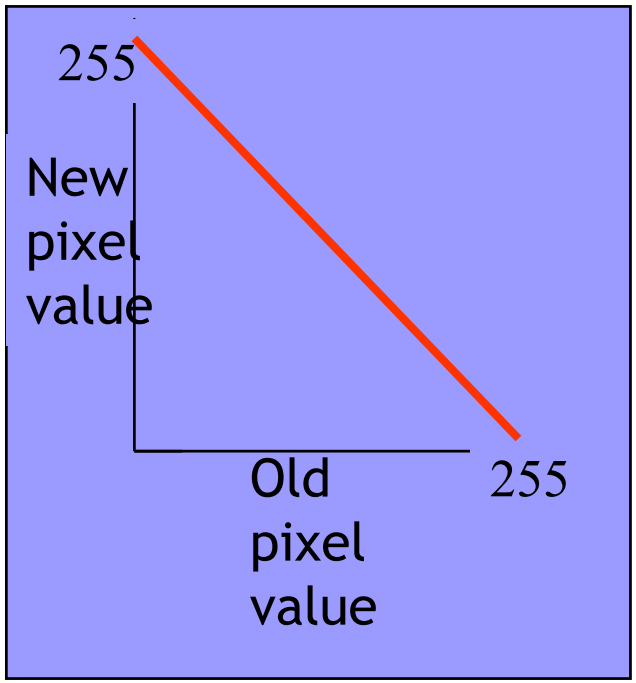
b. What is it?





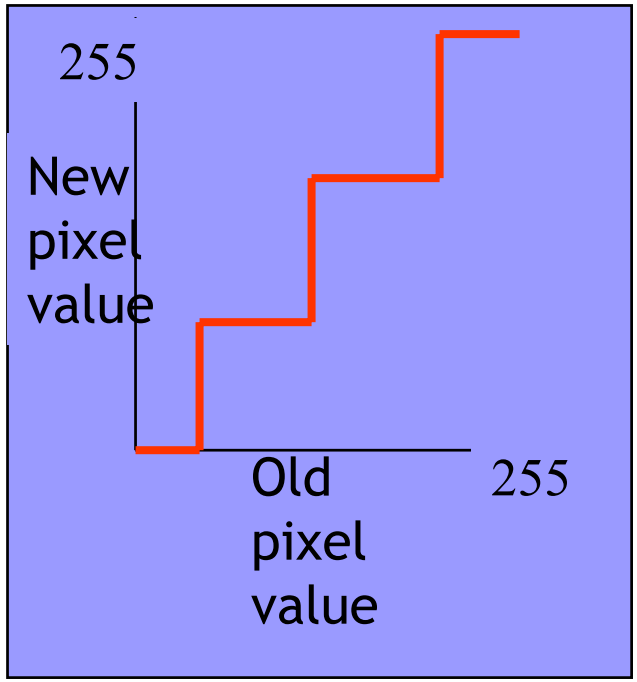
"Dynamics" (cont.)

c. What is it?



[redacted]

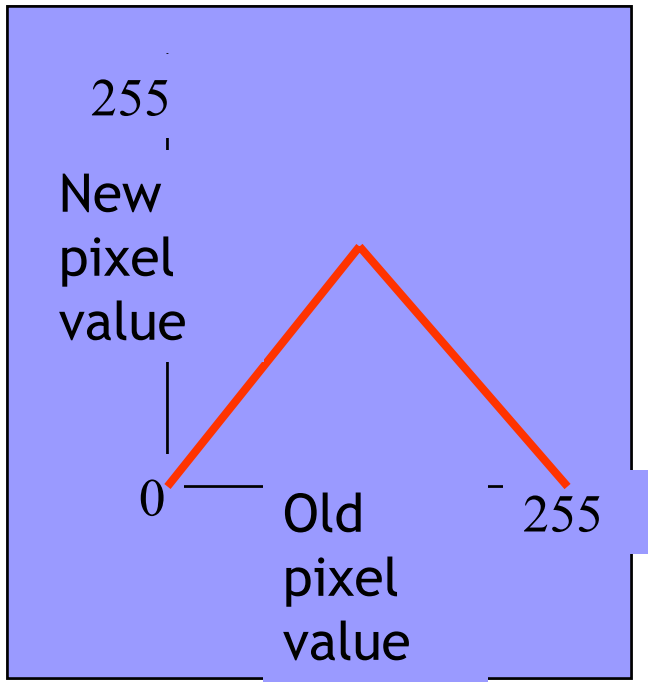
d. What is it?



[redacted]

Another Example of Dynamics

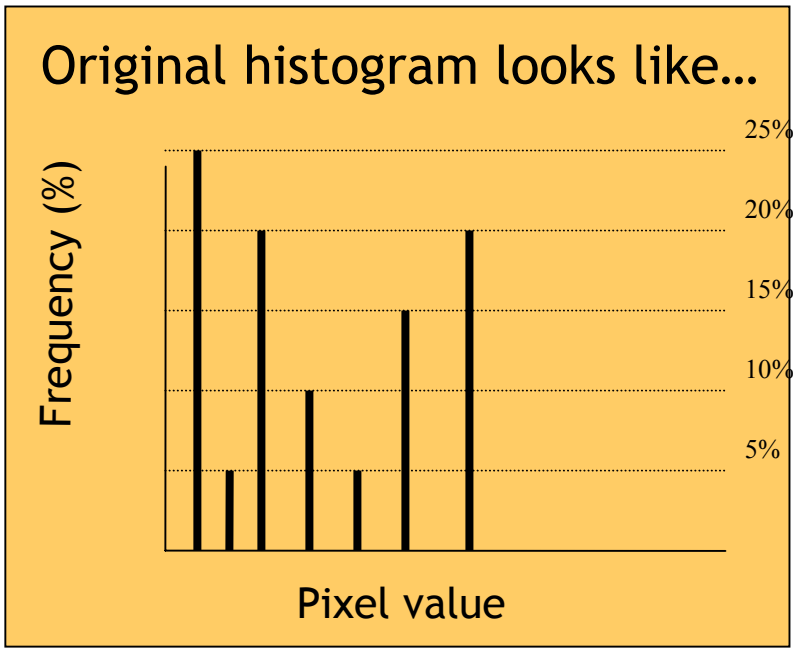
- Solarize

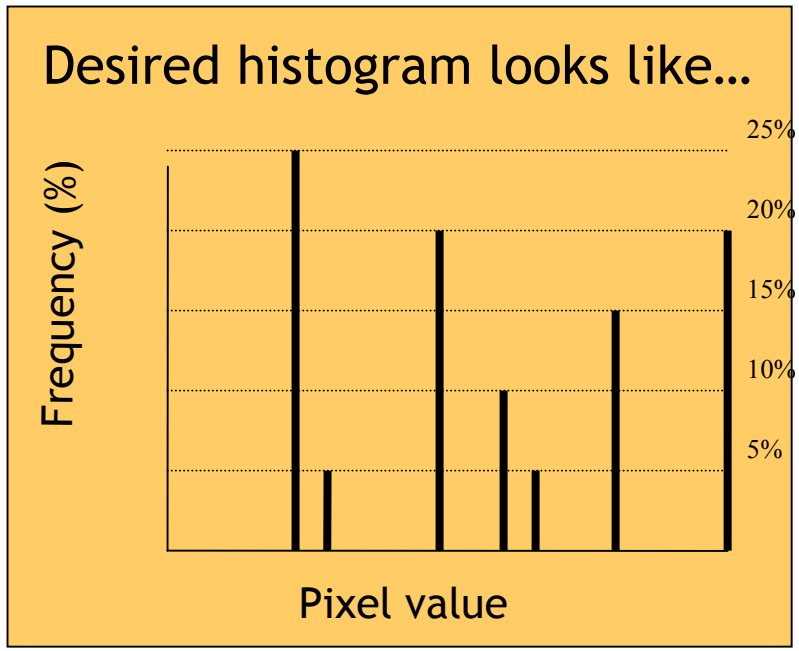


- What does it look like?

Histogram Specification

- *Specification* = "spreading" the original histogram to approximate some desired histogram
- *Equalization* = specification, where the desired histogram is the uniform distribution





SPECIFICATION (Cont.)

```
For j = 0 to 255
    OrigFrac[j] = Fraction of pixels in input image with value ≤ j
    DesiredFrac[j] = Fraction of pixels in desired histogram With value ≤ j
EndFor

NewValue = 0

For j = 0 to 255
    While ((DesFrac[NewValue] < OrigFrac[j]) && (NewValue < 255))
        NewValue = NewValue + 1
     OutValue[j] = NewValue
Endfor
```

SPECIFICATION (cont'd)

Example:

3-bit grayscale, range of possible values = 0..7

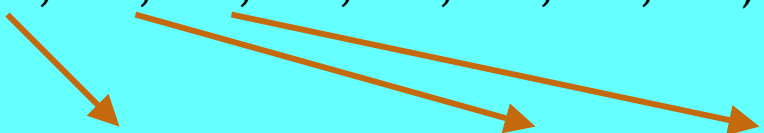
4 pixels; old values = (0, 1, 2, 1)

Desired: uniform distribution

(spread these values as uniformly as possible)

old values = *0 1 2 3 4 5 6 7*

Origfrac = (.25, .75, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0)



DesFrac = (.125, .25, .375, .5, .625, .75, .875, 1.0)

new values = *0 1 2 3 4 5 6 7*

New values = (1, 5, 7, 5)

Arithmetic Combination Of Images

- To combine two images, they must be the same size (same width and height)
  - Combine pixel in position [i,j] in image1
    with pixel in position [i,j] of image2
    to produce pixel in position [i,j] of image3 (the output)
- Addition (→ "mixing")

Example:

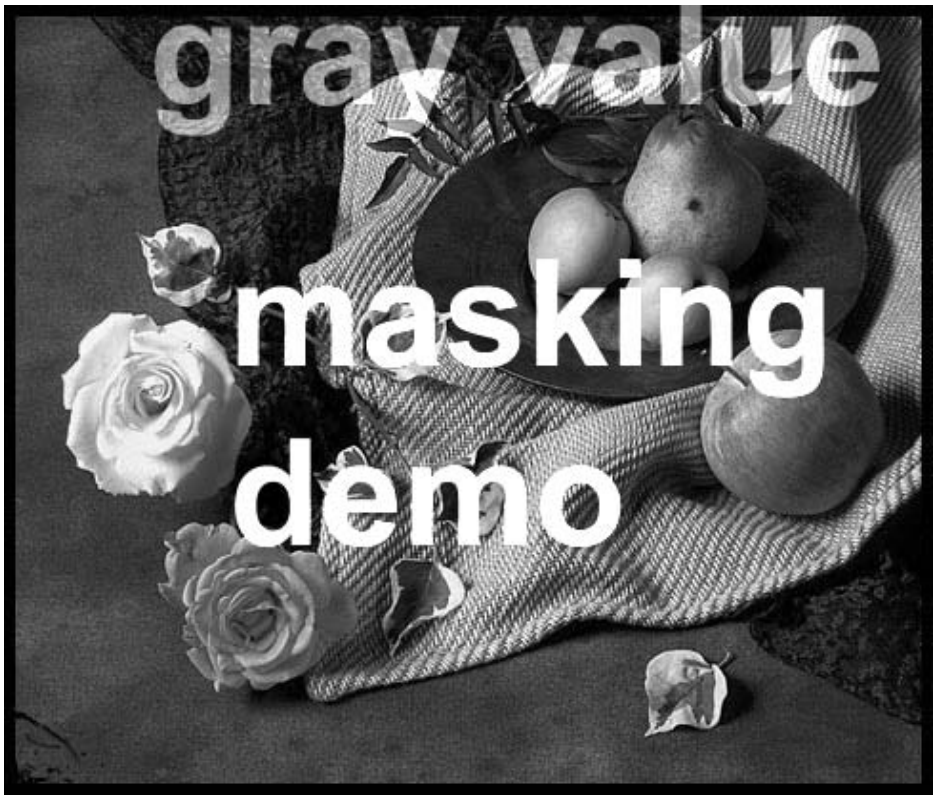
Image1 pixel = 80, image2 pixel = 40, image3 pixel = 80 + 40 = 120

Image1 pixel = 220, image2 pixel = 160, image3 = min(220+160,255) = 255

Arithmetic on Images: Examples

- Combine fruit image with mask image, using several operations

Example (cont'd)



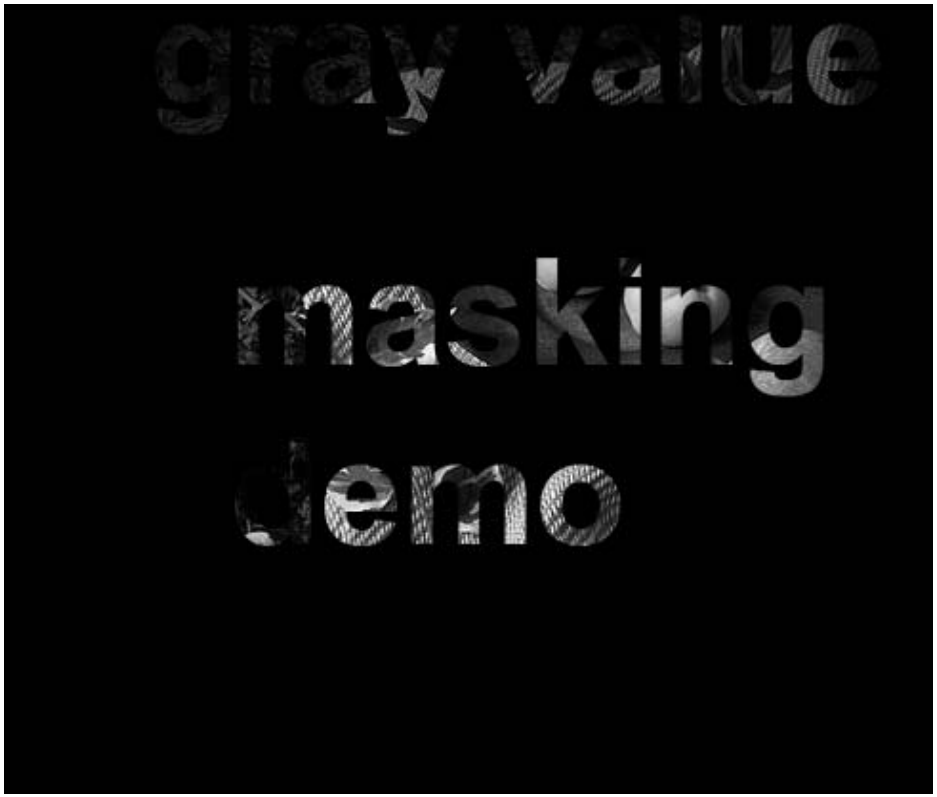
Operation = ????

Operation = ????

Boolean Combination Of Images

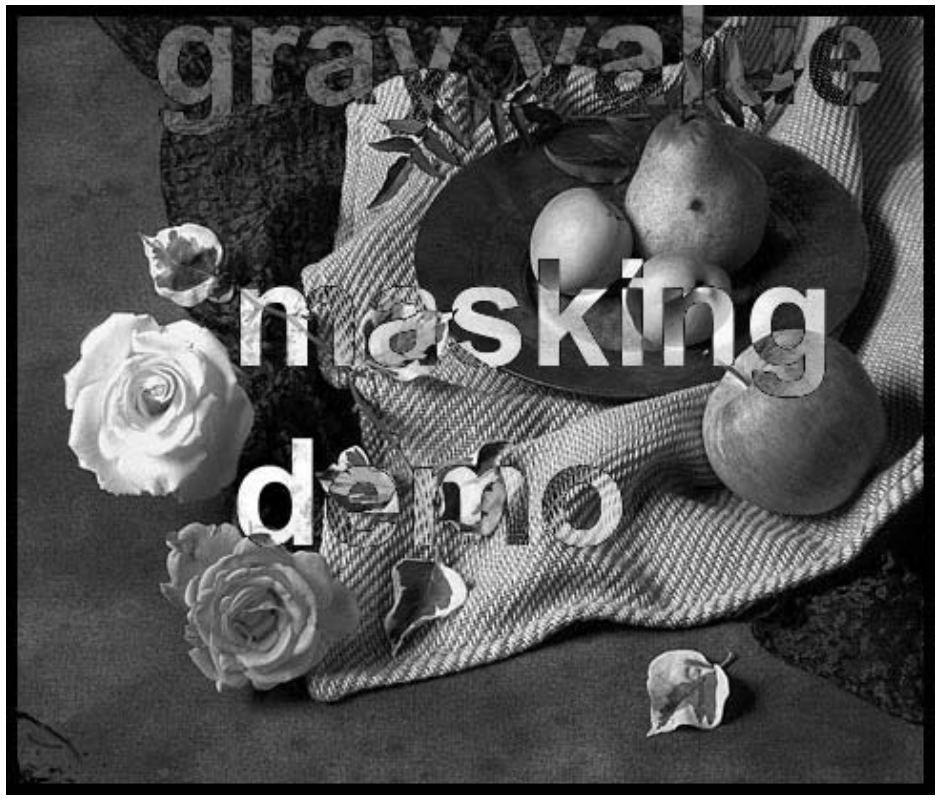
- Minimum possible pixel value is black
  - = 0 (decimal)
  - = 00000000 (8-bit binary)
- Maximum possible pixel value is white
  - = 255 (decimal)
  - = 11111111 (8-bit binary)
- OR image1 with image2
  - Result = white where image2 = white ("x OR 1 = 1")
  - Result = no change where image2 = black ("x OR 0 = x")
- AND image1 with image2
  - Result = black where image2 = black ("x AND 0 = 0")
  - Result = no change where image2 = white ("x AND 1 = x")

Example (cont'd)



Operation = ????

Operation = ????



Sources Of Info

- [Crane97] *A Simplified Approach to Image Processing*
  - Chapter 2