❖**Linear interpolation method**

✓Enlarges an image other than a factor of (2N-1)

✓Finds a line that connects two values in brightness space, and hence sampling it faster to get more samples, thus increasing the resolution

1

✓We do this for every pair of adjacent pixels, first along the rows and then along the columns. This will allow us to enlarge the image to a size of K(N-1)+1, where K is an integer and N x N is the image size. Typically, N is large and K is small, so this is approximately equal to KN

- Additional methods for image enlargement, such as *bilinear interpolation* which uses information in both the row and column directions, are explored in Chapter 9 for geometric restoration

2

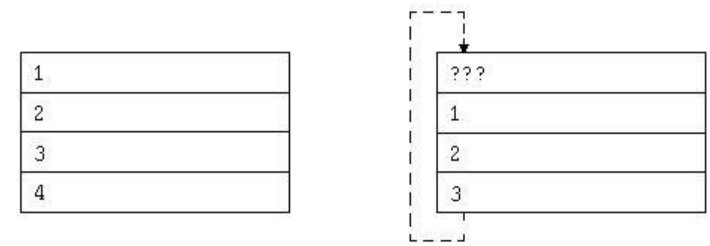
- *Shrink*: Helps to reduce the amount of data that needs to be processed

- *Translation and Rotation*:

✓Performed for application specific reasons like template matching in pattern recognition process

✓Makes certain image details easier to see

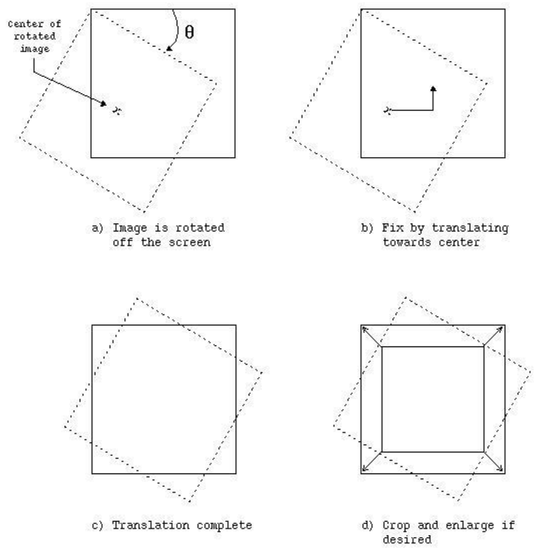
Figure 3.2-4: TRANSLATION



a) BEFORE: A 4-row image translating down by one row, $r_0 = 1$.

b) AFTER: If we wrap-around, row 4 goes into ???. Otherwise the top row is filled with a constant, typically zero

Figure 3.2-5: ROTATION



5

➢Arithmetic and Logic Operations

✓Performed on pixel by pixel basis

✓Arithmetic operations include addition, subtraction, multiplication and division

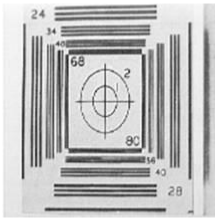
✓Logic operations include AND, OR, and NOT, and are performed in a bit-wise manner

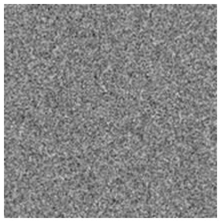
6

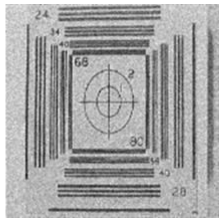

First Original


Second Original


Addition of images


Original image


Gaussian noise,
variance =400, mean =0


Addition of images

7

- **Subtraction**:
- Used for motion detection and background subtraction
- Applications include:
  1. Object tracking
  2. Medical imaging
  3. Law enforcement
  4. Military applications

8

a. Original scene

b. Same scene later

Subtraction of scene a from scene b

Subtracted image with threshold of 100

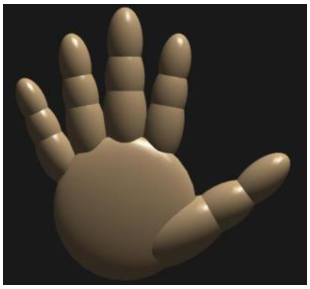
- **Multiplication and Division**:

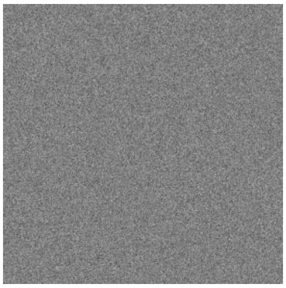
- To adjust the brightness of an image

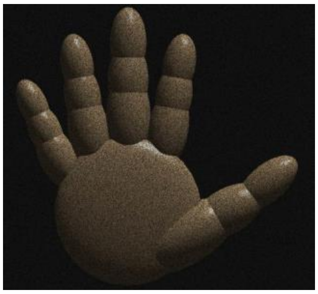
- Applications include:

  1. To combine two images for special effects
  2. For image filtering in spectral domain
  3. To model multiplicative noise process

Image Multiplication


Graphic hand image


Noise image for texture


Multiplication of two images

11

Image Division


Original image


Image divided by value<1


Image divided by value>1

12

➢Logic operations are performed on a pixel by pixel basis, and in a bit-wise manner

Example 3.2.3
We are performing a logic AND on two images. Two corresponding pixel values are 111_{10} in one image and 88_{10} in the second image. The corresponding bit strings are:

$$111_{10} = 01101111_2 \qquad 88 = 01011000_2$$

$$\begin{array}{rr} & 01101111_2 \\ \text{AND} & 01011000_2 \\ \hline & 01001000_2 \end{array}$$

✓**Logic operations**:

- AND, OR and NOT form a complete set of logic operators

- AND and OR are used:
  1. To combine information in two images
  2. For image masking operation
  3. For extracting Region of Interest

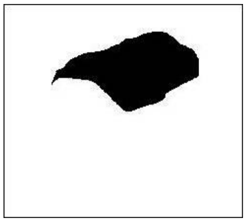
- NOT operation creates a negative of the original image

Image Masking

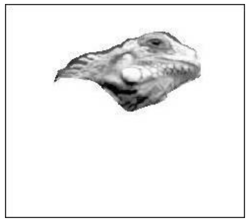
a. Original image

b. Image mask (AND)

c. ANDing a and b



d. Image mask (OR)



e. ORing a and d

NOT operation

Original image

Image after NOT operation

➢Spatial Filters

✓Operate on raw image data in the *(r,c)*
space, by considering small neighborhoods, 3x3, 5x5, 7x7, and moving sequentially across and down the image

✓Returns a result based on a linear or nonlinear operation

✓Consists of three types of filters:
- Mean filters – for noise
- Median filters – for noise
- Enhancement filters – for edges

✓Many spatial filters are linear filters implemented with a convolution mask;
the result is a weighted sum of a pixel and its neighbors

✓*Mask coefficients* tend to effect the image in the following general ways:
- Coefficients are positive: blurs the image
- Coefficients are alternating positive and negative: sharpens the image
- Coefficients sum to 1: brightness retained
- Coefficients sum to 0: dark image

•Mean filters:

- Averaging filters
- Tend to blur the image
- Adds a softer look to the image
- Example 3x3 convolution mask:

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

21

Mean filter

Original image

Mean filtered image

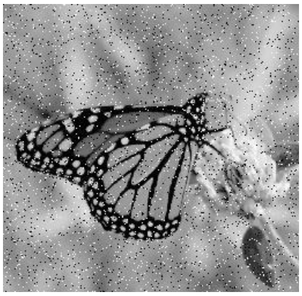
22

•**Median filters**:

- Nonlinear filter
- Sorts the pixel values in a small neighborhood and replaces the center pixel with the middle value in the sorted list
- Output image needs to be written to a separate image (a buffer), so that results are not corrupted
- Neighborhood can be of any size but 3x3, 5x5 and 7x7 are typical

23

Median filter



Original image with salt and pepper noise

Median filtered image (3x3)

24

- **Enhancement filters**:

  ▪ Implemented with convolution masks having alternating positive and negative coefficients

  ▪ Enhance the image by sharpening

  ▪ Two types considered here:
  1. Laplacian-type filters
  2. Difference filters

1. **Laplacian-type filters**:

✓ Are rotationally invariant, that is they enhance the details in all directions equally

✓ Example convolution masks of Laplacian-type filters are:

Filter 1	Filter 2	Filter 3
$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	$\begin{bmatrix} -2 & 1 & -2 \\ 1 & 5 & 1 \\ -2 & 1 & -2 \end{bmatrix}$

Laplacian filter

Original image

Laplacian filtered image

Contrast enhanced version of Laplacian filtered image

27

2. **Difference filters**:

✓Also called as emboss filters

✓Enhances the details in the direction specific to the mask selected

✓Four primary difference filter convolution masks, corresponding to the edges in the vertical, horizontal, and two diagonal directions are:

Vertical	Horizontal	Diagonal 1	Diagonal 2
$\begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & -1 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{bmatrix}$

28

Difference filter

Original image

Difference filtered image

Difference filtered image added to the original image, with contrast enhanced

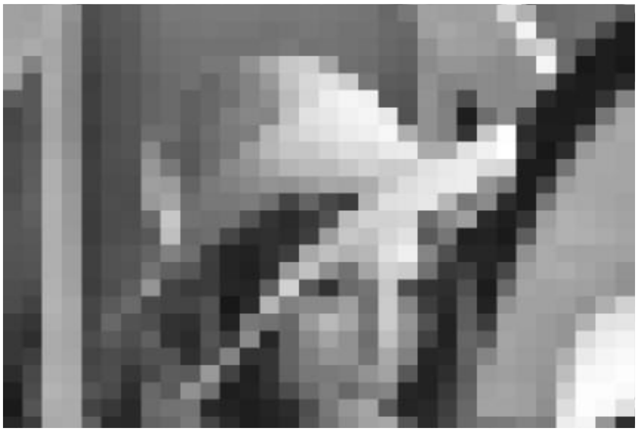
29

➢Image Quantization

✓Process of reducing the image data by removing some of the detail information by mapping groups of data points to a single point

✓Performed on spatial coordinates, *(r,c), for s*patial reduction; or pixel values, *I(r,c,)*
for gray level reduction

30



Spatial Reduction 512x512 to 32x32

Original Image 256 Gray levels

✓**Gray level reduction**:

- Reducing the number of gray levels, typically from 256 levels for 8-bit per pixel data to fewer than 8 bits
- Can be performed by
  - Thresholding
  - AND or OR masks

33

- **Thresholding**:

1. Performed by setting a threshold value and setting all pixels above it to “1” and those below it to “0”
2. Output is a binary image
3. Useful in extracting object features such as shape, area, or perimeter

34

- **Gray level reduction with AND/OR masks**:

  - Perform a logical AND or logical OR operation with a bit string “mask”
  - Number of bits that are masked determine the number of gray levels available
  - AND based method maps quantized gray values to the low end of each range
  - OR based method maps quantized gray values to the high end of each range

35

Example:

To reduce 256 gray levels down to 16 we use an OR mask of 00001111. Now, values in the range of 0-15 are mapped to 15, those ranging from 16 to 31 are mapped to 31, and so on

36

Example:

If we performed the quantization down to 16 levels by an OR with a mask of 00001111, which maps the values to the high end of the range, we could shift the values down to the middle of the range by ANDing with a mask of 11111000

✓**False Contouring**:

- Artificial edges or lines that appear in images with reduced number of gray levels
- Can be improved by using an Improved Gray Scale (IGS) quantization method
  - IGS quantization method improves the results of gray level reduction by adding a random number to each pixel value before the quantization

False Contouring

Original 8-bit image,
256 gray levels

Quantized to 6 bits ,
64 gray levels

Quantized to 3 bits ,
8 gray levels

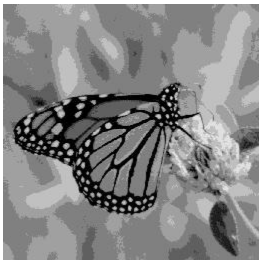
Quantized to 1 bits ,
2 gray levels

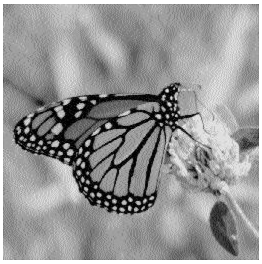
39

IGS quantization

Original
image

Uniform quantization
to 8 levels (3 bits)





IGS quantization
to 8 levels (3 bits)

40