Digital Image Processing

Presented by:

Dr. Moe Moe Myint

Information Technology Department

Technological University (Kyaukse), Myanmar

moemoemyint@moemyanmar.ml

www.slideshare.net/MoeMoeMyint

- Only Original Owner has full rights reserved for copied images.

- This PPT is only for fair academic use.

Dr. Moe Moe Myint
Information Technology Department
Technological University (Kyaukse)

Image Arithmetic (Lab 6)

- the implementation of standard arithmetic operations, such as addition, subtraction, multiplication, and division, on images
- many uses in image processing both as a preliminary step in more complex operations and by itself

Objectives

- To display the absolute difference between a filtered image and the original
- To add two images together, specify an output class and reverse black and white in a binary image
- To divide one image into another
- To multiply two images
- To subtract one image from another

Required Equipment

- Computers with MATLAB software and Projector

Practical Procedures

- Read the image and then calculate the absolute difference of two images, use the *imabsdiff* command
- To add two images, use the *imadd* command
- To complement image, use the *imcomplement* command
- To divide one image into another , use the *imdivide* command
- To multiply two images, use the *immultiply* command
- To subtract one image from another, use the *imsubtract* command

Image Arithmetic

Imabsdiff	Absolute difference of two images
Imadd	Add two images or add constant to image
Imcomplement	Complement image
Imdivide	Divide one image into another or divide image by constant
Imlincomb	Linear combination of images
Immultiply	Multiply tow images or multiply image by constant
Imsubtract	Subtract one image from another or subtract constant from image

imabsdiff

- Absolute difference of two images

Syntax

Z = imabsdiff(X,Y)

Examples

- Calculate the absolute difference between two uint8 arrays. Note that the absolute value prevents negative values from being rounded to zero in the result, as they are with imsubtract.

```
X = uint8([ 255 10 75; 44 225 100]);

Y = uint8([ 50 50 50; 50 50 50 ]);

Z = imabsdiff(X,Y)
```

- Display the absolute difference between a filtered image and the original.

```
I = imread('cameraman.tif');

imshow(I),title('OI');

J = uint8(filter2(fspecial('gaussian'), I));

figure,imshow(J),title('FI');

K = imabsdiff(I,J);

figure,imshow(K,[]),title('AI');
```

imadd

- Add two images or add constant to image

Syntax

```
Z = imadd(X,Y)
```

Examples

- **Add two uint8 arrays. Note the truncation that occurs when the values exceed 255.**

```
X = uint8([ 255 0 75; 44 225 100]);
Y = uint8([ 50 50 50; 50 50 50 ]);
Z = imadd(X,Y)
```

- **Add two images together and specify an output class.**

```
I = imread('rice.png');
J = imread('cameraman.tif');
K = imadd(I,J,'uint16');
imshow(K,[])
```

- **Add a constant to an image.**

```
I = imread('rice.png');
J = imadd(I,50);
subplot(1,2,1), imshow(I)
subplot(1,2,2), imshow(J)
```

imcomplement

- Complement image

Syntax

```
IM2 = imcomplement(IM)
```

In the complement of a binary image, zeros become ones and ones become zeros; black and white are reversed. In the complement of an intensity or RGB image, each pixel value is subtracted from the maximum pixel value supported by the class (or 1.0 for double-precision images) and the difference is used as the pixel value in the output image. In the output image, **dark areas become lighter and light areas become darker.**

Examples

- **Create the complement of a uint8 array.**

```
X = uint8([ 255 10 75; 44 225 100]);
X2 = imcomplement(X)
```

- **Reverse black and white in a binary image.**

```
bw = imread('text.png');
bw2 = imcomplement(bw);
subplot(1,2,1),imshow(bw)
subplot(1,2,2),imshow(bw2)
```

- **Create the complement of an intensity image.**

```
I = imread('glass.png');
J = imcomplement(I);
imshow(I), figure, imshow(J)
```

imdivide

- Divide one image into another or divide image by constant

Syntax

Z = imdivide(X,Y)

Examples

- Divide two uint8 arrays. Note that **fractional values greater than or equal to 0.5 are rounded up to the nearest integer.**

```
X = uint8([ 255 10 75; 44 225 100]);
Y = uint8([ 50 20 50; 50 50 50 ]);
Z = imdivide(X,Y)
```

- Estimate and divide out the background of the rice image.

```
I = imread('rice.png'); imshow(I), title('O');
background = imopen(I,strel('disk',15));
figure,imshow(background);title('Morphological image');
Ip = imdivide(I,background); figure,imshow(Ip,[]),title('Divide I');
```

- Divide an image by a constant factor.

```
I = imread('rice.png');
J = imdivide(I,2);
subplot(1,2,1), imshow(I);subplot(1,2,2), imshow(J)
```

imlincomb

- Linear combination of images

Syntax

Z = imlincomb(K1,A1,K2,A2,...,Kn,An)

Description

Z = imlincomb(K1,A1,K2,A2,...,Kn,An) computes

K1*A1 + K2*A2 + ... + Kn*An

Example 1

- Scale an image by a factor of 2.

```
I = imread('cameraman.tif');
J = imlincomb(2,I);
imshow(J)
```

Example 2

- Form a difference image with the zero value shifted to 128.

```
I = imread('cameraman.tif');
J = uint8(filter2(fspecial('gaussian'), I));
K = imlincomb(1,I,-1,J,128); % K(r,c) = I(r,c) - J(r,c) + 128
figure, imshow(K)
```

immultiply

- Multiply two images or multiply image by constant

Syntax

```
Z = immultiply(X,Y)
```

Examples

- **Multiply an image by itself. Note how the example converts the class of the image from uint8 to uint16 before performing the multiplication to avoid truncating the results.**

```
I = imread('moon.tif');

I16 = uint16(I);

J = immultiply(I16,I16);

imshow(I), figure, imshow(J)
```

- **Scale an image by a constant factor:**

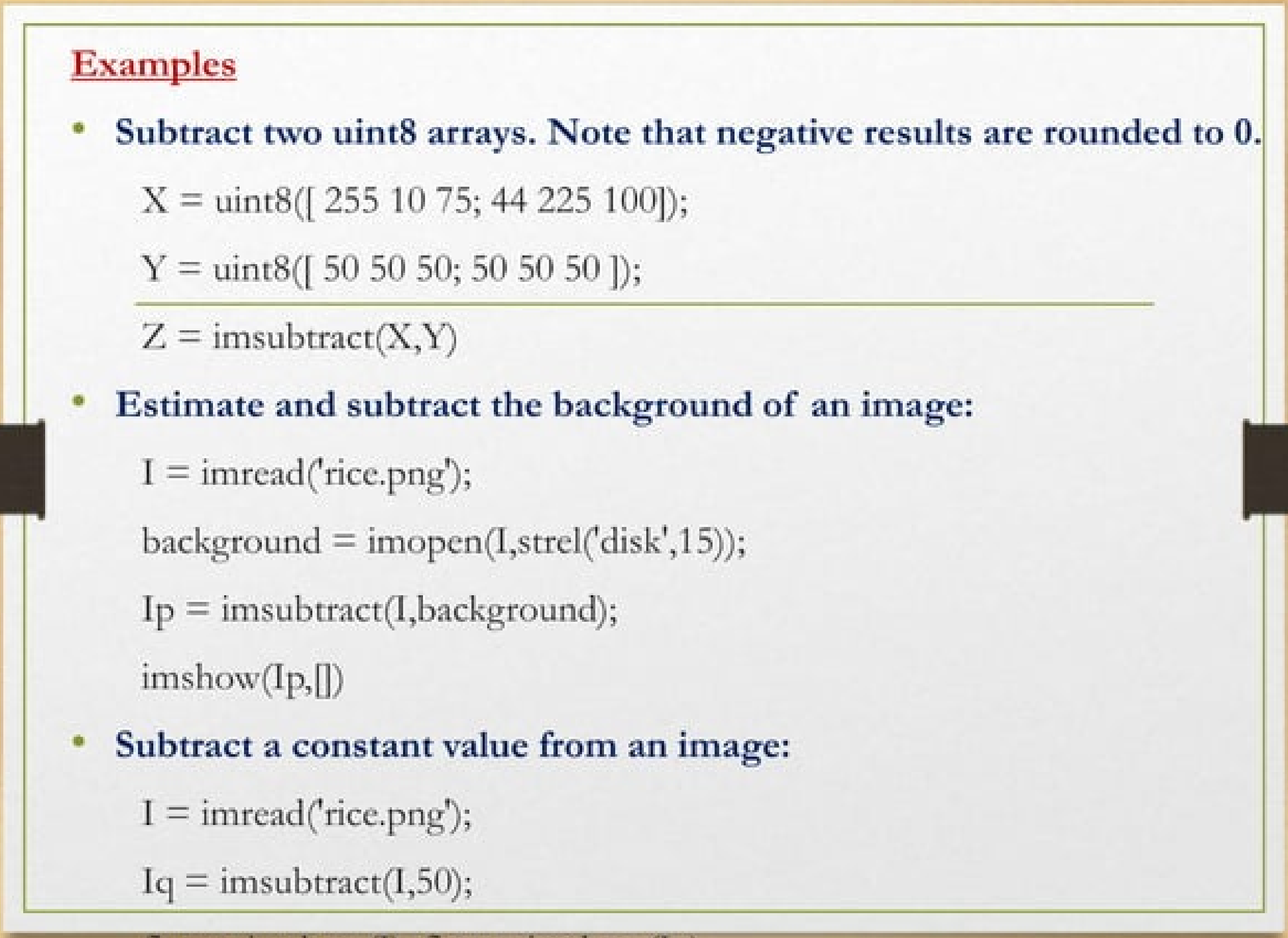
```
I = imread('moon.tif');

J = immultiply(I,0.5);

subplot(1,2,1), imshow(I)

subplot(1,2,2), imshow(J)
```

imsubtract

- Subtract one image from another or subtract constant from image

Syntax

```
Z = imsubtract(X,Y)
```



Examples

- **Subtract two uint8 arrays. Note that negative results are rounded to 0.**

```
X = uint8([ 255 10 75; 44 225 100]);
Y = uint8([ 50 50 50; 50 50 50 ]);
Z = imsubtract(X,Y)
```

- **Estimate and subtract the background of an image:**

```
I = imread('rice.png');
background = imopen(I,strel('disk',15));
Ip = imsubtract(I,background);
imshow(Ip,[])
```

- **Subtract a constant value from an image:**

```
I = imread('rice.png');
Iq = imsubtract(I,50);
figure, imshow(I), figure, imshow(Iq)
```

Questions?