

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.

Image Enhancement (Lab 7)

- To process an image so that the result is more suitable than the original image for a specific application

Objectives

- To apply contrast-limited Adaptive Histogram Equalization (CLAHE) to an image
- To enhance contrast using histogram equalization
- To adjust image intensity values or colormap
- To add noise to image and Display the results

Required Equipment

- Computers with MATLAB software and Projector

Practical Procedures

- Use the *adapthisteq* command
- Use the *histeq* command
- Use the *imadjust* command
- Use the *imnoise* command
- Use the *medfilt2* command

Image Enhancement

- **adapthisteq** Contrast-limited adaptive histogram equalization (CLAHE)
- **histeq** Enhance contrast using histogram equalization
- **imadjust** Adjust image intensity values or colormap
- **imnoise** Convert integer values using lookup table
- **medfilt2** 2-D median filtering

adapthisteq

- Contrast-limited adaptive histogram equalization (CLAHE)

Syntax

J = adapthisteq(I)

J = adapthisteq(I,param1,val1,param2,val2...)

Description

- J = adapthisteq(I) enhances the contrast of the grayscale image I by transforming the values using contrast-limited adaptive histogram equalization (CLAHE).
- J = adapthisteq(I,param1,val1,param2,val2...) specifies any of the additional parameter/value pairs listed in the following table. Parameter names can be abbreviated, and case does not matter.

Examples

- **Apply Contrast-limited Adaptive Histogram Equalization (CLAHE) to an image and display the results.**

```
I = imread('tire.tif');  
A = adapthisteq(I,'clipLimit',0.02,'Distribution','rayleigh');  
figure, imshow(I);  
figure, imshow(A);
```

- **Apply CLAHE to a color image.**

```
[X MAP] = imread('shadow.tif'); % Convert indexed image to true-color (RGB) format  
RGB = ind2rgb(X,MAP); % Convert image to L*a*b* color space  
cform2lab = makecform('srgb2lab'); % Create color transformation structure  
LAB = applycform(RGB, cform2lab); % Scale values to range from 0 to 1  
L = LAB(:, :, 1)/100; % Perform CLAHE  
LAB(:, :, 1) = adapthisteq(L, 'NumTiles', ...  
    [8 8], 'ClipLimit', 0.005)*100; % Convert back to RGB color space  
cform2srgb = makecform('lab2srgb');  
J = applycform(LAB, cform2srgb); % Display the results  
figure, imshow(RGB);  
figure, imshow(J);
```

histeq

- Enhance contrast using histogram equalization

Syntax

`J = histeq(I, hgram)`

`J = histeq(I, n)`

`[J, T] = histeq(I,...)`

`newmap = histeq(X, map, hgram)`

`newmap = histeq(X, map)`

`[newmap, T] = histeq(X,...)`

Description

- `histeq` enhances the contrast of images by transforming the values in an intensity image, or the values in the colormap of an indexed image, so that the histogram of the output image approximately matches a specified histogram.

Examples

- Enhance the contrast of an intensity image using histogram equalization.

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

```
J = histeq(I);
```

```
imshow(I)
```

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

- Display a histogram of the original image.

```
figure; imhist(I,64)
```

- Compare it to a histogram of the processed image.

```
figure; imhist(J,64)
```

imadjust

- Adjust image intensity values or colormap
-

Syntax

`J = imadjust(I)`

`J = imadjust(I,[low_in; high_in],[low_out; high_out])`

`J = imadjust(I,[low_in; high_in],[low_out; high_out],gamma)`

`newmap = imadjust(map,[low_in; high_in],[low_out;
high_out],gamma)`

`RGB2 = imadjust(RGB1,...)`

Examples

- Adjust a low-contrast grayscale image.

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

```
J = imadjust(I);
```

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

- Adjust the grayscale image, specifying the contrast limits.

```
K = imadjust(I,[0.3 0.7],[]);
```

```
figure, imshow(K)
```

- Adjust an RGB image.

```
RGB1 = imread('football.jpg');
```

```
RGB2 = imadjust(RGB1,[.2 .3 0; .6 .7 1],[]);
```

```
imshow(RGB1), figure, imshow(RGB2)
```

imnoise

- Add noise to image

Syntax

`J = imnoise(I,type)`

`J = imnoise(I,type,parameters)`

`J = imnoise(I,'gaussian',m,v)`

`J = imnoise(I,'localvar',V)`

`J = imnoise(I,'localvar',image_intensity,var)`

`J = imnoise(I,'poisson')`

`J = imnoise(I,'salt & pepper',d)`

`J = imnoise(I,'speckle',v)`

Examples

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

```
J = imnoise(I,'salt & pepper',0.02);
```

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

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


medfilt2

- 2-D median filtering

Syntax

`B = medfilt2(A, [m n])`

`B = medfilt2(A)`

`B = medfilt2(A, 'indexed', ...)`

`B = medfilt2(..., padopt)`

Description

Median filtering is a nonlinear operation often used in image processing to reduce "salt and pepper" noise. A median filter is more effective than convolution when the goal is to simultaneously reduce noise and preserve edges.

Examples

- Add salt and pepper noise to an image and then restore the image using medfilt2.

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

```
J = imnoise(I,'salt & pepper',0.02);
```

```
K = medfilt2(J);
```

```
imshow(J), figure, imshow(K)
```

