

Multimedia Systems

Lecture 3

LECTURER

MOUHSEN IBRAHIM

Contents

- ▶ Extended Contrast modification
- ▶ Filters
- ▶ Noise
- ▶ Average Filters
- ▶ Median Filters
- ▶ GIF
- ▶ GIF – example horizontal movement
- ▶ Exercise

Extended Contrast modification

- ▶ We can use the stretchlim function to find the best range of colors to stretch using it.
- ▶ `I = imread('gray.jpg');`
- ▶ `Lim = stretchlim(I);`
- ▶ `J = imadjust(I,Lim,[]);`
- ▶ `subplot(2,2,1),imshow(I),title('Original Image');`
- ▶ `subplot(2,2,2),plot(imhist(I)),title('Original Image Histogram');`
- ▶ `subplot(2,2,3),imshow(J),title('Contrast modified Image');`
- ▶ `subplot(2,2,4),plot(imhist(J)),title('Contrast Modified Histogram');`
- ▶ We can reverse the order of numbers in the last parameter of `imadjust`.

Filters

- ▶ Image filters are used to apply various effects on images.
- ▶ The filter is a matrix of odd dimensions, here we will use only square filters.
- ▶ We apply a filter to an image using this formula
 - ▶ $I(x, y) = \sum_{u=-a}^a \sum_{v=-b}^b f(u, v) * I(x + u, y + v)$.
- ▶ For every pixel we take the sum of products of the current pixel or a neighbor of it with the corresponding value in the filter matrix.
- ▶ This method of applying filters is called **Correlation**.
- ▶ If the sum of elements in the filter matrix equals 1 then the resulting image will have the same brightness value, if it is smaller than 1 then the result will be darker and bigger than 1 the result will be brighter.

Noise

- ▶ We can add noise to an image using the `imnoise` function.
- ▶ We will use four types of noise:
 - ▶ Gaussian noise: `G = imnoise(I,'gaussian',0,0.05);`
 - ▶ Salt and Pepper noise: `SP = imnoise(I,'salt & pepper',0.02);`
 - ▶ Poisson noise: `P = imnoise(I,'poisson');`
 - ▶ Speckle noise: `S = imnoise(I,'speckle');`
- ▶ In the next slides we will try to use filters for removing this noise.

Average Filters

- ▶ To apply a filter in matlab we use two functions
 - ▶ `fspecial('type',parameters)` we pass the filter type to this function and any parameters specified for this type.
 - ▶ `filter2(filter,image)` this function uses the output from the first one to apply the filter on the image.
- ▶ **Average Filters**
- ▶ We will apply the average filter on the noisy images to see how it works:
 - ▶ `H = fspecial('average',5);`
 - ▶ `im = uint8(filter2(h,G))`
 - ▶ `imshow(im)`

Average Filters

- ▶ Now we try to do the same but with the salt & pepper noisy image
 - ▶ `H = fspecial('average',5);`
 - ▶ `im = uint8(filter2(h,SP))`
 - ▶ `imshow(im)`
- ▶ We can do the same for poisson and speckle noises.
- ▶ **Median Filters**
- ▶ In this kind of the filters the value of each pixel is modified to equal the median of the pixel values in the specified window.

Median Filters

- ▶ `Im2 = medfilt2(G,[3 3]);`
- ▶ `Imshow(Im2)`
- ▶ `Im2 = medfilt2(SP,[3 3]);`
- ▶ `Imshow(Im2)`
- ▶ `Im2 = medfilt2(P,[3 3]);`
- ▶ `Imshow(Im2)`
- ▶ `Im2 = medfilt2(S,[3 3]);`
- ▶ `Imshow(Im2)`
- ▶ Which noise type is this filter the best to remove?

Adaptive Filter

- ▶ These filters use special statistics to calculate the value of each pixel based on surrounding pixels to decrease signal to noise ratio.
- ▶ `im4=wiener2(G,[5 5]);`
- ▶ `imshow(im4);`
- ▶ `im4=wiener2(SP,[5 5]);`
- ▶ `imshow(im4);`
- ▶ `im4=wiener2(P,[5 5]);`
- ▶ `imshow(im4);`
- ▶ `im4=wiener2(S,[5 5]);`
- ▶ `imshow(im4);`

Filters results

- ▶ These results are left for the students to find out by them selves and to encourage them to work at home instead of wasting time on facebook and social media.....
- ▶ Compare the use of filters on different noise types and share the results with your teacher.

GIF

- ▶ A GIF image is a set of images stored in a single file displayed one by one creating the illusion of animation.
- ▶ These images are stored using indexed color model to reduce size.
- ▶ We cannot use `imshow` to display GIF images.
- ▶ GIF images are created using the `imwrite` function, the first and second parameters are the image's table and map.
- ▶ The third parameter is the file's name.
- ▶ Extra parameters include:
 - ▶ `DelayTime`: Transition time between images.
 - ▶ `loopcount`: specified when writing the first image, determines how many times to loop the images.
 - ▶ `writemod`: is used to append more images to the file.

GIF – example horizontal movement

- ▶ `I = imread('gray.jpg');`
- ▶ `I1 = imresize(I,[256 256]);`
- ▶ `I2 = I1;`
- ▶ `for i=1:16:256`
 - ▶ `I2(:,1:i+1) = I1(:,256-i:256);`
 - ▶ `[tbl map] = gray2ind(I2,256);`
 - ▶ `if (i == 1)`
 - ▶ `imwrite(tbl,map,'horizontal.gif','DelayTime',0.24,'loopcount',2);`
 - ▶ `else`
 - ▶ `imwrite(tbl,map,'horizontal.gif','DelayTime',0.24,'writemode','append');`
 - ▶ `end`
- ▶ `end`

Exercise

- ▶ Write a matlab script to create a vertical movement of an image.
- ▶ Write a matlab script to create a resizing movement of an image.

THE END
GOOD LUCK