

Video Signals

Date: 8 July 2024

Ex.1.[12 pts]

The first two (non-scale invariant) “Hu moments invariants” are described as:

$$\Phi_1 = \mu_{20} + \mu_{02}$$

$$\Phi_2 = (\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2$$

Where μ_{pq} is the central moment that, for a digital image $I(x, y)$, is described as:

$$\mu_{pq} = \sum_x \sum_y (x - \bar{x})^p (y - \bar{y})^q I(x, y), \text{ where } \bar{x} \text{ and } \bar{y} \text{ localize the centroid.}$$

Given the three following objects A, B and C:

0	0	0	0	0	0	0	0
0	0	8	0	0	0	0	0
0	0	0	0	0	0	0	0
4	0	0	0	0	4	0	0
0	0	0	0	0	0	0	0
0	0	8	0	0	0	0	0
0	0	0	0	0	0	0	0

0	0	0	0	0	0	0	0
0	0	0	4	0	0	0	0
0	0	0	0	0	0	0	0
0	8	0	0	0	8	0	0
0	0	0	0	0	0	0	0
0	0	0	4	0	0	0	0
0	0	0	0	0	0	0	0

0	0	0	0	0	0	0	0
0	0	0	0	8	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	4	0	0	0	4	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	8	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

[7 pts.] Find, for each of them, the values of Φ_1 and Φ_2 detailing the steps.

[5 pts.] Describe why the Hu approach is able to recognize the object B as a simple rotation of the object A while the object C is a completely different object.

Ex.2.[11 pts]

Given the following image on the right we want to apply a 3x3 Prewitt vertical edge extractor.

[6 pts] Provide the Prewitt filter and the filtered image.

[5 pts] We need to replicate the same result using the 2D Discrete Fourier Transform: Describe (without calculations) all the steps to obtain the same result working in the frequency domain, in particular detailing the size of the chosen image and filter.

0	0	0	0	0	0
0	0	0	0	0	0
0	0	2	0	0	0
0	0	3	0	0	0
0	4	4	4	0	0
0	0	3	0	0	0
0	0	2	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0

Ex. 3 is overleaf

Es.3. [11 pts to be solved writing on the paper a suitable MATLAB code]

You want to decompose image x as the sum between x_L , an image containing the smooth component of x , and x_R , containing high frequency information. You want to select x_L as the argument that minimizes the following cost function:

$$\min_z \frac{1}{2} \|x - f_L * z\|_2^2 + \lambda (\|g_1 * z\|_2^2 + \|g_2 * z\|_2^2) \quad (1)$$

Where $*$ is the convolution operation, f_L is a low pass filter and g_1 and g_2 are two high pass filter highlighting horizontal and vertical edges respectively. The idea behind this is that x_L must be as similar as possible to x while having as low high-frequency components as possible. The parameter λ controls the trade off between this two requirements. It could be proven that solution x_L could be obtained from the following formula:

$$x_L = FFT^{-1}\{X_L\}, \quad X_L = H \cdot X \quad (2)$$

Being the filter H obtained from the following formula:

$$H = \frac{\overline{F_L} \cdot F_L}{\overline{F_L} \cdot F_L + \lambda_d (\overline{G_1} \cdot G_1 + \overline{G_2} \cdot G_2)} \quad (3)$$

Where $\bar{\cdot}$ is the complex conjugate operator and all the multiplications and divisions are performed element wise; F_L , G_1 and G_2 are the Fourier transformations of f_L , g_1 and g_2 respectively.

Write a MATLAB script able to perform the following steps:

- Read the grayscale input image (stored in 'test.png' file), convert it to a double representation and visualize it.
- Define the filter kernel f_L as a gaussian filter with size 15 and standard deviation 2.
- Define the filter kernels g_1 and g_2 as the 3x3 Sobel operators for extracting horizontal and vertical edges respectively.
- Obtain the Fourier transform of the input image and of the kernels defined in the previous points. They all must have the same size as the input image.
- After defining $\lambda = 0.1$ obtain H using the Eq.3.
- Obtain x_L using Eq.2 and visualize it
- Obtain x_R by subtracting x_L from the input image and visualize it. (*hint: x_R can contain negative values*).
- What would happen if $\lambda = 0$?

List of possible Matlab functions

```
figure
im2double
imread
rgb2gray
imcrop
imfilter
imhist
conj
rgb2ind
imagesc
rgb2hsv
imshow
fspecial
fft2
ifft2
imdilate
fftshift
```

Solutions

Es.1

Since the centroid is the central pixel of each image in this case, we can write for A:

$$\mu_{20} = 2^2 \cdot 4 + 0^2 \cdot 8 + 0^2 \cdot 8 + (-2)^2 \cdot 4 = 32$$

$$\mu_{02} = 2^2 \cdot 8 + 0^2 \cdot 4 + 0^2 \cdot 4 + (-2)^2 \cdot 8 = 64$$

$$\mu_{11} = 2 \cdot 0 \cdot 4 + 0 \cdot 2 \cdot 8 + 0 \cdot (-2) \cdot 8 + (-2) \cdot 0 \cdot 4 = 0$$

$$\Phi_1 = 96$$

$$\Phi_2 = 32^2$$

For B:

$$\mu_{20} = 2^2 \cdot 8 + 0^2 \cdot 4 + 0^2 \cdot 4 + (-2)^2 \cdot 8 = 64$$

$$\mu_{02} = 2^2 \cdot 4 + 0^2 \cdot 8 + 0^2 \cdot 8 + (-2)^2 \cdot 4 = 32$$

$$\mu_{11} = 2 \cdot 0 \cdot 8 + 0 \cdot 2 \cdot 4 + 0 \cdot (-2) \cdot 4 + (-2) \cdot 0 \cdot 8 = 0$$

$$\Phi_1 = 96$$

$$\Phi_2 = 32^2$$

For C:

$$\mu_{20} = 2^2 \cdot 4 + 0^2 \cdot 8 + 0^2 \cdot 8 + (-2)^2 \cdot 4 = 32$$

$$\mu_{02} = 3^2 \cdot 8 + 0^2 \cdot 4 + 0^2 \cdot 4 + (-3)^2 \cdot 8 = 144$$

$$\mu_{11} = 2 \cdot 0 \cdot 4 + 0 \cdot 3 \cdot 8 + 0 \cdot (-3) \cdot 8 + (-2) \cdot 0 \cdot 4 = 0$$

$$\Phi_1 = 176$$

$$\Phi_2 = 112^2$$

Then, the first two moments are the same for A and B (since they are just rotated) while they are completely different for the C object.

Es.2

Using the Prewitt filter: $\begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$ and convolving it with the image we get:

0	0	0	0	0	0	0
0	0	2	0	-2	0	0
0	0	5	0	-5	0	0
0	4	9	0	-9	-4	0
0	4	10	0	-10	-4	0
0	4	9	0	-9	-4	0
0	0	5	0	-5	0	0
0	0	2	0	-2	0	0
0	0	0	0	0	0	0

In the Frequency domain we have to take two images of the same size, since the original image is 5x3 and the filter is 3x3 the output of the linear convolution is (5+3-1)x(3+3-1)=7x5. In order to avoid circular convolution effects we can then take a 7x7 square image and square filter (zero-

padding the starting ones). Take the 2D DFT of both of them. Multiply element-wise the two transforms and then evaluate the inverse 2D DFT to obtain the result.

Es.3

```
clc
close all
clear all

%a)
x = im2double(imread('test.png'));
[M,N] = size(x);
figure
imshow(x)

%b)
f_L = fspecial('gaussian',15,2);

%c)
g_1 = fspecial('sobel');
g_2 = fspecial('sobel')';

%d)
X = fft2(x);
F_L = fft2(f_L,M,N);
G_1 = fft2(g_1,M,N);
G_2 = fft2(g_2,M,N);

%e)
lambda = 0.1;
H =
F_L.*conj(F_L)./(conj(F_L).*F_L+lambda*(conj(G_1).*G_1+conj(G_2).*
G_2));

%f)
x_L = ifft2(X.*H);
figure
imshow(x_L)

%g)
x_R = x-x_L;
figure
imagesc(x_R)
axis equal
```

h) If $\lambda = 0$ the filter H in the frequency domain would just be a matrix full of ones so it would have no impact on the input image x and x_L would be equal to x .