Video Signals

Date: 17 June 2024

Ex.1.[12 pts] In the following image there is a periodic noise:

[4 pts] Describe the noise and provide its Discrete Fourier Transform.

[4 pts] Describe a possible approach to remove the noise.

[4 pts] Describe how such a noise can be interpreted in the Wiener Filter and how it can be applied to restore the image.

Ex.2.[11 pts] Given two views of the same flat wall we need to describe the transformation between them.

[3 pts] In order to do this provide a possible criterion to define candidate pixels to be matched between the two images.

[4 pts] Define a possible local descriptor to be associated to each candidate point detailing its structure and construction.

[4 pts] Three iterations of the Ransac algorithm provide the following matches:

1st iteration	A→A, B→C, C→D, D→B, E→E, F→F, G→H, H→G
2nd iteration	A→C, B→A, C→D, D→A, E→H, F→G, G→F, H→E
3rd iteration	A→B, B→A, C→C, D→D, E→E, F→G, G→F, H→H

Describe the evaluations performed by the Ransac in each case and its decisions.

Ex. 3 is overleaf

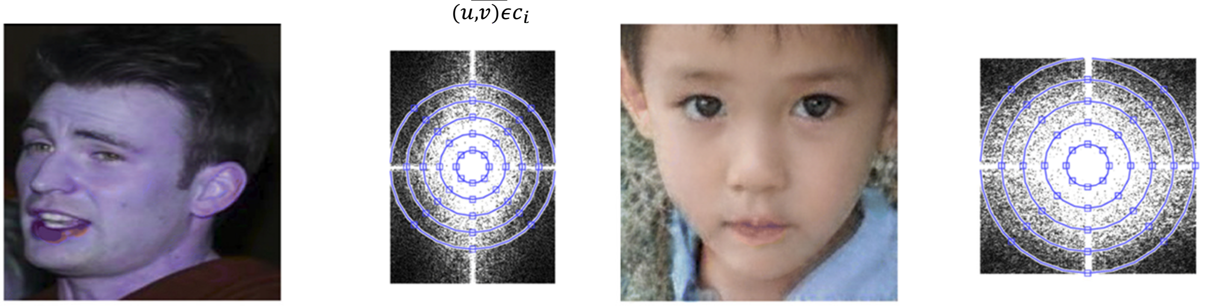
Es.3. [11 pts to be solved writing on the paper a suitable MATLAB code]

You want to develop an application that is able to detect fake deep learning generated face images. In particular, from the figure below it can be observed that fake face images have a wide range of high frequency components whereas real face images have strong low frequency components and less high frequency components. Your goal is to extract the following 5 features:

$$f = [\beta_1, \beta_2 - \beta_1, \beta_3 - \beta_2, \beta_4 - \beta_3, \beta_5 - \beta_4]$$

Where β_i are the sum of magnitudes of Fourier transform within the i^th concentric circle c_i.

$$\beta_i = \sum_{(u,v)\epsilon c_i} |F(u,v)| \ , i = 1,2,3,4,5$$



(a) Real Face Image (b) Magnitude of FFT of Real Face (c) Fake Face Image (d) Magnitude of FFT of Fake Face

Write a MATLAB script able to perform the following steps:

a) Read the *'fake.jpeg'* image, convert it to a double representation grayscale image and visualize it.
b) Obtain M and N the vertical and horizontal size of the input image.
c) Compute the Fourier transform of the input image and shift it so the DC component will be in the middle.
d) Obtain the magnitude information from the Fourier transform output.
e) Define the r_max, largest circle radius, as half of the minimum between M and N and r_step as one fifth of r_max.
f) Initialize b as a 5 elements vectors full of zeros and using a cycle perform the following operation for each circles (5 times):
    I. Initialize a mask as a binary matrix having the same size of the input image and populate it with false values.
    II. Set to true the pixels of mask inside the circle you are considering (hint: to find the radius you can multiply r_step by the cycle variable)
    III. Sum all the magnitude values which corresponds to true values inside of mask. Store the result inside vector b.
g) From b obtain the feature vector f using the definition given above.

List of possible Matlab functions

```
figure
im2double
imread
fftshift
rgb2gray
imcrop
imfilter
imhist
imopen
im2bw
histeq
ifft2
imshow
ifftshift
imerase
```

Solutions

Es.1

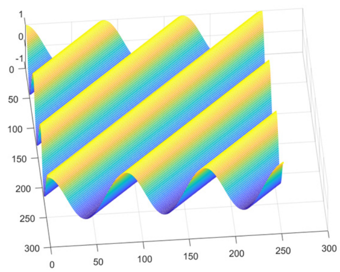
The noise is additive deterministic with an equation like:

$$w(m,n) = A\cos\left(2\pi\frac{3m+3n}{L}\right)$$

Assuming that L is the size in pixels of the image and the term 3 represents the horizontal and vertical spatial frequency.

As a MATLAB example (not requested) you can obtain the noise description, assuming $L = 256$, as:

```
L=256;
w=zeros(L);
for m=1:L
    for n=1:L
        w(m,n)=cos(2*pi*(3*m+3*n)/L);
    end
end
mesh(w);
```



Its 2D Discrete Fourier Transform will present two peaks in positions (4,4) and (L-3,L-3) corresponding to the normalized frequencies $f_x = f_y = \frac{3}{256}$ and $f_x = f_y = \frac{256-3}{256} \leftrightarrow -\frac{3}{256}$ due to the periodicity in the frequency domain.

The noise can be removed in the frequency domain, for example, removing the two peaks at the aforementioned frequencies and substituting them with the average value of neighbor frequencies.

Concerning the Wiener filter, in this case we are dealing with a deterministic noise and not a stochastic one, the noise power spectrum can be directly obtained from the two peaks in the DFT obtained above.

Es.2

Candidate points can be obtained, for example, localizing corners using Harris corner detector: corners allow to identify points candidates without ambiguities as in uniform regions or edges.

Many local descriptors can be used, like SIFT, SURF, ORB… see course material for more information.

Concerning Ransac, in the first iteration there are 5 wrong matches, for the second one all 8 matches are wrong and so it will be discarded. The last iteration has just 4 wrong matches from the resulting homography so it will become the new best choice.

Es.3

```
close all
clear all
clc

%a)
I = im2double(imread('fake.jpeg'));
I = rgb2gray(I);
figure; imshow(I)
```

```
%b)
[M, N] = size(I);

%c)
I_fft = fftshift(fft2(I));

%d)
I_mag = abs(I_fft);

%e)
r_max = min([M N])/2;
r_step = r_max/5;

%f)
b = zeros(1,5);
for k=1:5

    %f1)
    mask = zeros(size(I));
    mask = mask>0;

    %f2)
    for i = 1:size(I,1)
        for j = 1:size(I,2)
            r = sqrt((i-size(I,1)/2)^2 + (j-size(I,2)/2)^2);
            if(r<=r_step*k)
                mask(i,j) = 1;
            end
        end
    end

    %f3)
    b(k) = sum(I_mag(mask));

end

%g)
f = [b(1) diff(b)];
```