

## Video Processing

Date: 24/02/2016

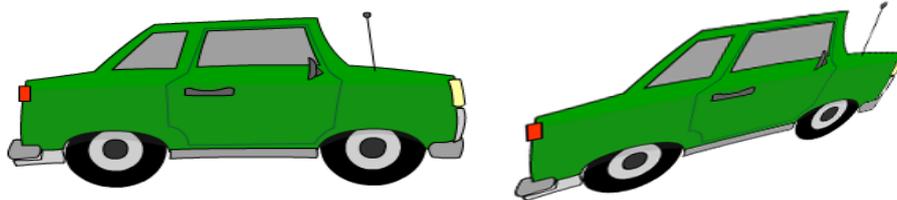
---

**Ex.1.[10 Pt]** In the image (100x100 pixels) on the right there is a disturbing deterministic wave pattern, which should be removed. We want to remove this noise working in the frequency domain.

1. Describe the noise in the frequency domain: what will be its exact shape? At which exact frequencies it will be placed?
2. Describe a criterion to remove it in the frequency domain minimizing the impact on the restored image.
3. What are the assumptions of the 2D Discrete Fourier Transform? How could be defined the SNR of the original Image?



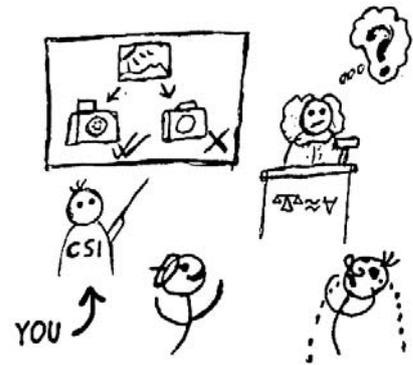
**Es.2. [10 pt]** We want to match point-to-point the object on the left with the object on the right.



Describe a possible approach to find corresponding points and then to match them between the two views.

**Es.3. [13 pt] MATLAB exercise ON THE BACK!**

**[MATLAB Exercise: Image Forensics Example]** There is a dispute regarding the authorship of the award winning *veryCoolPicture.png*. Two world-famous photographers claim the ownership of the picture. In a high profile case the court summons you as an expert in image processing\*.



Knowing that each imaging sensor is characterized by a fixed pattern noise, using the two contending devices you acquired two reference images, *cam\_1.png* and *cam\_2.png*. Write the MATLAB code that finds which device is most likely to have captured the disputed photograph. Proceed as follows:

- Knowing that all the photos are saved as truecolor images with 8 bit per channel; read, load and visualize the three images;
- Prior to processing convert all images in grayscales and rescale the values to [0,1] interval using double precision;
- In order to estimate the PRNU (*Photo Response Non-Uniformity*) of the two devices, for each reference image compute

$$I_N(x, y) = I(x, y) - I_{denoised}(x, y),$$

where  $I$  is the original image and  $I_{denoised}$  its denoised version. Suppose that it is enough to low-pass the image in order to remove the noise (use the Gaussian low-pass filter with size 5 and standard deviation 1).

- Repeat the step c) for the image under test.
- Compute the phase correlation between the test noise  $I_N^{(t)}$  and each reference noise  $I_N^{(i)}$  ( $i = 1, 2$ ). Proceed in frequency domain by computing the cross-power spectrum

$$R_i(\omega_x, \omega_y) = \frac{N_i(\omega_x, \omega_y)N_t^*(\omega_x, \omega_y)}{|N_i(\omega_x, \omega_y)N_t^*(\omega_x, \omega_y)|},$$

where  $N_t$  and  $N_i$  are, respectively, the 2D Fourier transforms of  $I_N^{(t)}$  and  $I_N^{(i)}$ . Finally, the normalized cross-correlation is given by the inverse Fourier transform of  $R_i$ .

- Visualize the correlation results. Device that captured the contested image is the one that shows the highest peak in the correlation function – find which one is it. **CASE SOLVED!**

### Matlab List of functions

```
figure
im2double
im2bw
rgb2gray
fspecial
imread
imresize
imrotate
imfilter
imnoise
fft2
ifft2
imshow
imagesc
getimage
size
zeros
find
abs
angle
conj
max
```

\* you should be an expert after finishing the Video Signals course, right?

# Solutions

## Ex.1

The noise is a vertical sinusoid with  $\lambda = \frac{100}{4}$  pixels that implies a  $\omega_y = \frac{2\pi}{\lambda} = 0.251 \frac{\text{rad}}{\text{pixel}}$ .

Applying the FFT2D will give two vertical values (the horizontal frequency  $k_x$  will be zero),  $k_y = 4$  and  $k_y = 100 - 4 = 96$

To remove the noise we can remove the values at the two aforementioned frequencies and substituting their values with the mean value or the nearest neighbourhood.

The assumption of the 2D DFT is that the signal is periodic in both dimensions.

To evaluate the SNR we could simply evaluate the average difference in the overall image power spectrum of the values of the two noise peaks with respect to nearest neighbours (that will be the noise power). The remaining power will be the signal power.

## Ex.2

A typical pipeline to extract corresponding points consists in:

Extracting corners using, e.g. the Harris procedure.

Define a local descriptor in the neighbourhood of each point robust, in this case, to rotation, scaling and translation. [ Even if the deformation, like in this case, is, at least projective, since parallel lines are not preserved, locally, in a close neighbourhood of a point, it can be approximated as a similarity transform].

The matching between features in the two images can be based on Sum of Squared Differences (SSD) or other distances.

The RanSaC algorithm can then be applied to find the best Projective transform between corresponding points and to remove wrong matches.

## Ex.3

```
% a) Load images
I_ref1 = imread('cam_1.png');
I_ref2 = imread('cam_2.png');
I_test = imread('veryCoolPicture.png'); % (same
as cam 1)
figure, imshow(I_ref1); figure, imshow(I_ref2); figure, imshow(I_test);

% b) Conversion
I_ref1 = double(rgb2gray(I_ref1));
I_ref2 = double(rgb2gray(I_ref2));
I_test = double(rgb2gray(I_test));

% c) Extract reference noise
H = fspecial('gaussian', 5, 1);
N_ref1 = I_ref1 - imfilter(I_ref1, H);
N_ref2 = I_ref2 - imfilter(I_ref1, H);

% d) Extracte test noise
N_test = I_test - imfilter(I_test, H);

% e) Check camera in frequency domain
NN_ref1 = fft2(N_ref1); NN_ref2 = fft2(N_ref2); NN_test = fft2(N_test);
CC_1 = ifft2( (NN_ref1.*conj(NN_test)) ./ abs(NN_ref1.*conj(NN_test)) );
CC_2 = ifft2( (NN_ref2.*conj(NN_test)) ./ abs(NN_ref2.*conj(NN_test)) );
```

```
% f) Find camera
figure
subplot(121), imagesc(fftshift(CC_1)), axis image
subplot(122), imagesc(fftshift(CC_2)), axis image
[~, winner] = max([max(CC_1(:)), max(CC_2(:))]);
disp(winner);
```