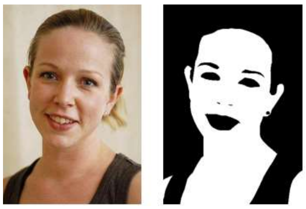
EE 409 – Mini Project 4

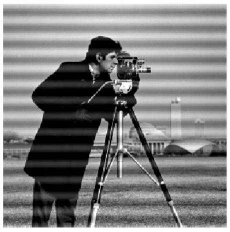
Due: 31.12.2023 (Sunday), 22:00

Notice: A severe penalty will be applied in case of cheating

For this mini project, you will again use a photo of yourself as you did in the previous mini projects. The resolution of the image should be **higher than 0.1 megapixel and lower than 1 megapixel**. For each question you should report the **obtained figures (with titles on top)** together with your **Matlab codes** and the **time** it takes to run your code. Name your report as **NameSurname.docx** and upload to **Aybuzem (Max. file size is 10MB)**



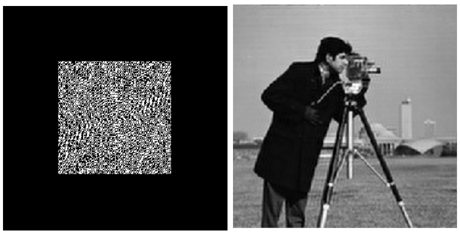
1. (15p) **(Skin color detection in YCbCr color space)** Convert your RGB color image into YCbCr color space. Mark the pixels with skin color as white and others as black using the following YCbCr thresholds:
   Y > 80 , 85 < Cb <135, 135 < Cr < 180



2. (15p) **(Adding a periodic noise to image in frequency domain)** Take the **FFT** of your **original grayscale** image and add a **periodic sinusoidal noise** by changing just the **c**th **vertical** frequency coefficient with a proper value. (**c=d+10**; where **d** is the **last digit** of your student **ID number**.)

3. (20p) **(Distorting image by blurring in frequency domain)** Take the **FFT** of your **original grayscale** image. Then **blur** your photo(**using its FFT**) with an "**n x n**" **averaging filter** in frequency domain. If your image has "**c**" number of **columns** then **n=2*round(c/100)+1**. (Ex: If your image has 437 columns, then n=2*round(437/100) + 1 = 9; So apply 9x9 averaging filter to your image). You should use the **FFT of the averaging mask**. Then take the **inverse FFT** and display the blurred image.

4. (15p) **(Reconstructing blurred image by inverse filtering) Deblur** your blurred image in **frequency** domain. Display the blurred and deblurred images on the same graph by using subplot();.



5. (20p) **(Image compression using FFT)** You will **compress** your grayscale image by **deleting** some of its **highest frequency FFT coefficients**. Preserve only **half the size of rows and columns of coefficients in the center** of the **image FFT**. (Don't forget to use fftshift and ifftshift). Display the **original** grayscale image, **compressed** image and its **fft coefficients** side by side by using subplot();. **Compare** the **quality** of the **compressed** image with the **original** image and comment on the result. How much **space** does your **compressed** image FFT take if your **original** image FFT takes 100?

6. (15p) Assume we have a rotating wheel with “d+1” number of equally spaced spokes where “d” is the last digit of your student ID number. We want to record the video of the wheel while it is rotating clockwise at 30 rpm. What should be the minimum frame rate of our video recorder in order not to see

any temporal aliasing? Justify your answer. If we record with a lower frame rate, what do we see when we watch that video?