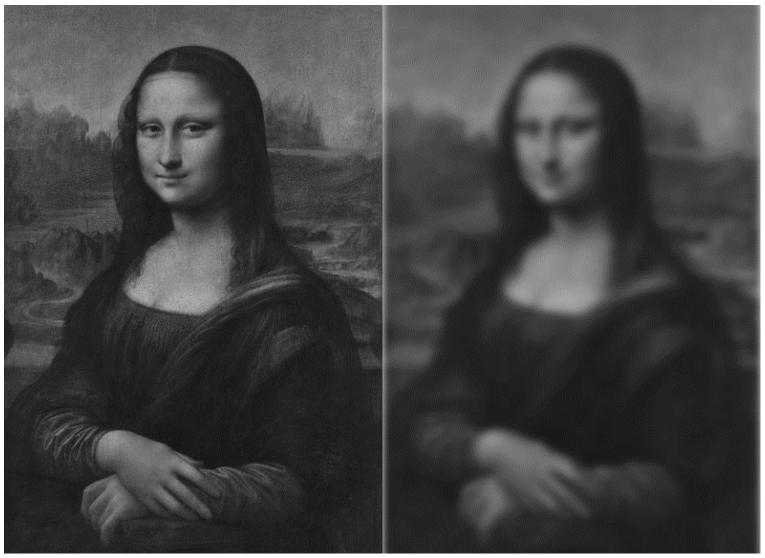
Video Signals

Date: 16/02/2017



Ex.1.[12 Pt] Due to an error in the acquisition settings the Mona Lisa image on the left is acquired as shown on the right.
The estimated PSF of the optics is the following:

$$PSF = \frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix}$$

1. Describe how to the PSF can be estimated once you know the real and the acquired image (working in the frequency domain).
2. Assuming the absence of noise during acquisition, describe all required steps in order to restore the image. What problems could arise?

Es.2. [7pt]
Describe how the Fourier Transform can be adopter in order to retrieve orientation of straight lines inside images. Is the location of the straight line also encoded in the Fourier Transform? If yes, how can we retrieve it?

.**Es.3 is over the page**

Es.3. [13 pt] You are walking in the park when suddenly a wild Pokémon appears! Well, actually the Pokémon needs your help in order to be able to appear in augmented reality. Given a Pokémon stored in the image *pikachu.png*, and the scene *DEIB_park.jpg* captured by your smartphone camera (both are RGB images with 8bit per channel), write the MATLAB code that places the Pokémon inside the captured scene. Proceed as follows:

a) Read and visualize the two images;

b) Prepare the Pokémon image for the mixing:
b1) resize it so that its height is 1/4 of the height of the scene image;
b2) knowing that on the image the background is pure green, find the coordinates of Pokémon pixels, i.e. rows and columns of the image that are not the background (hint: `[row, col] = find(…)`);

c) Elaborate the scene image in order to find a suitable location for our Pokémon (it can't just randomly fly around, right?). To do so, let's suppose that there is a table in the scene and that somehow we know that its color is close to [R_0, G_0, B_0] = [188, 186, 197]. To find the table, divide the image into pieces and find the piece that contains the highest number of pixels that are close to the given color:
c1) we want to divide the image using a 9x6 uniform grid – find the size of the pieces (height and width in pixels) and initialize a matrix that will be used to store the distance measure of each piece with respect to the reference color [R_0, G_0, B_0];
c2) for each piece do the following: knowing the size of the pieces, compute the positions of the current piece pixels and use them to extract the corresponding R, G and B planes. For each color plane analyze its histogram to find the bin with highest number of pixels. Use the corresponding color values R_m, G_m, B_m to compute the color distance of the current piece from the reference color [R_0, G_0, B_0], as

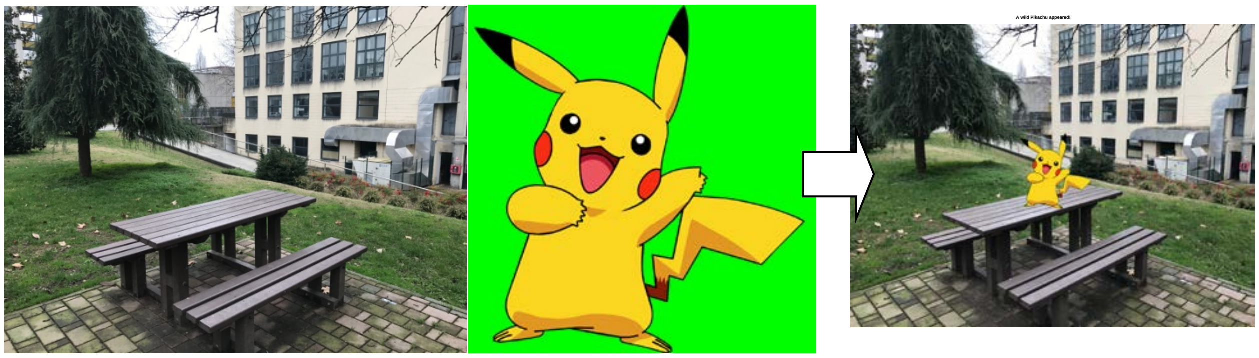
$$dist = \sqrt{(R_m - R_0)^2 + (G_m - G_0)^2 + (B_m - B_0)^2}$$

Store the computed distance in the previously initialized matrix;
c3) given the matrix that contains the distance measure computed for each image piece, find the piece with lowest distance value. We want to use the center of this piece as a Pokémon location – compute its coordinates;

d) Place the Pokémon inside the scene: for each pixel that is not a background, computed in step b2), find a corresponding pixel in the scene image, staring from the coordinates computed in step c3), and replace it with the Pokémon pixel. Visualize the result. Finally, a wild Pokémon appeared!

Matlab List of possible functions

```
figure
im2double
im2bw
rgb2gray
fspecial
imread
imresize
imrotate
imfilter
imnoise
imhist
fft2
ifft2
imshow
imagesc
getimage
size
zeros
find
abs
angle
conj
double
max
min
imerode
imdilate
imopen
imclose
```



Solutions

Ex.1

Ex.2

Ex.3

```
% a)
I1 = imread('pikachu.png'); figure; imshow(I1);
I2 = imread('DEIB_park.jpeg'); figure; imshow(I2);

% b1)
I1 = imresize(I1, 0.25*size(I2,1)/size(I1,1));
% b2)
[pika_r, pika_c] = find(I1(:,:,1) ~= 0 | I1(:,:,2) ~= 255 | I1(:,:,3) ~= 0);

% c1)
Nr = 9; nRows = size(I2,1)/Nr;
Nc = 6; nColumns = size(I2,2)/Nc;
colorDist = zeros(Nr,Nc);
% c2)
for i = 1:Nr
    for j = 1:Nc
        pieceRows = nRows*(i-1)+1:nRows*i;
        pieceColumns = nColumns*(j-1)+1:nColumns*j;
        hR = imhist(I2(pieceRows, pieceColumns,1));
        hG = imhist(I2(pieceRows, pieceColumns,2));
        hB = imhist(I2(pieceRows, pieceColumns,3));
        [~,R] = max(hR); [~,G] = max(hG); [~,B] = max(hB);
        colorDist(i,j) = norm(double([R,G,B]+1) - [188,186,197]);
    end
end
% c3)
[i,j] = find(colorDist == min(colorDist(:)));
start_r = nRows*(i-1) + nRows/2 - size(I1,1);
start_c = nColumns*(j-1) + nColumns/2;

% d)
for n = 1:length(pika_c)
    I2(start_r + pika_r(n), start_c + pika_c(n),:) = I1(pika_r(n),pika_c(n),:);
end
figure; imshow(I2); title('A wild Pikachu appeared!');
```