Video Signals Exam (29 June 2020)

MATLAB Exercise

You need to develop a video call application in which you can substitute the background behind the subject with a different background picture. In particular you want to develop a simple algorithm able to analyse a video stream and detect moving foreground. In order to do so you want to apply a running avarage approach in which pixels belonging at moving foreground are detected by comparing their difference in respect to a running average, to a threshold that depends on the running standard deviation.

Load the 8-bit grayscale video frames which are stored in a 3-dimensional variable called *frames*, the third dimension corresponds to index frame (hint: use `load('frames.mat')`). Write a MATLAB script able to perform the following steps:

a) Read the 8-bit input color image which will be used as the new background (stored in the file *'beach.jpg'*), convert it to grayscale using a double representation.
b) Resize the grayscale background image obtained in order to match the frames size.
c) Convert to a double representation the first video frame, apply a 3x3 median filter and assign the result to the variable *mu*, which will be used for the running average.
d) Initialize a variable called *stdev* with the same size as *mu* and homogeneous value equal to 0.01. Initialize two scalar variables (*rho* and *k*) respectively equal to 0.0001 and 2.5.
e) Cycle through the video frames, starting from the second one, and for each one of them apply the following operations:
   I. Convert to a double representation the current frame and apply a 3x3 median filter obtaining *img*
   II. Calculate the matrix *d* as $d = |img - mu|$
   III. Update *mu* using the formula $rho * img + (1 - rho) * mu$
   IV. Update *stdev* using the formula $\sqrt{rho * d^2 + (1 - rho) * stdev^2}$
   V. Obtain a binary mask for foreground pixels assigning 1 to pixels in which *d* is greater than *k*stdev*
   VI. Using a square 10x10 structuring element, apply a morphological operation able to remove small isolated black regions (holes) from the mask
   VII. Combine *img* and the new background image using the obtained binary mask in order to substitute the original background with the new one. Visualize the obtained image

Matlab

List of possible functions

```
figure
medfilt2
imread
rgb2gray
imcrop
imfilter
imopen
strel
imclose
imshow
size
imerode
zeros
ones
strel
imresize
imdilate
im2double
```

```
clear all
close all

load('frames.mat')
%a)
bg = imread('beach.jpg');
bg = im2double(rgb2gray(bg));
%b)
bg = imresize(bg,[size(frames,1),size(frames,2)]);
%c)
mu = medfilt2(im2double(frames(:,:,1)),[3, 3]);
%d)
sigma = ones(size(mu))*0.01;
rho = 0.0001;
k = 2.5;
%e)
for i = 2:size(frames,3)
    %e.1)
    img = medfilt2(im2double(frames(:,:,i)),[3, 3]);
    %e.2)
    d = abs(img - mu);
    %e.3)
    mu = rho * img + (1-rho) * mu;
    %e.4)
    sigma = sqrt(rho * d.^2 + (1-rho) * sigma.^2);
    %e.5)
    mask = (d > k*sigma);
    %e.6)
    mask = imclose(mask,strel('square',10));
    %e.7)
    output = mask.*img + (1-mask).*bg;
    imshow(output)
 end
```