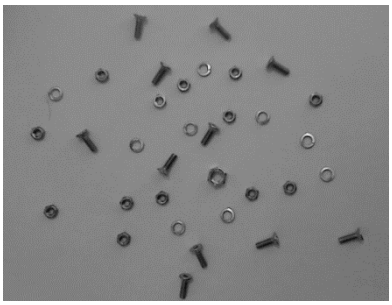
Video Signals

3 Feb 2021



You want to develop an application able to monitor the number of bolts on a conveyor belt. You assume to have an overhead shot, as in the figure reported below. Moreover, you assume that the conveyor belt has a homogenous color and that bolts are the only objects extending in a particular direction while the other objects (washers and nuts) are circular ones.

Write a MATLAB script able to perform the following steps:

a) Read the 8-bit input grayscale image (the filename is *'input_image.png'*), convert it into a double representation and visualize it.
b) Apply the horizontal and vertical Sobel filters, apply them to the input image and combine the results in order to obtain the pixelwise magnitude and direction (in degrees) of the edges.
c) Calculate a 255 bins histogram of the input image. Obtain the gray value that corresponds to the highest peak and assume that as the background color. Using the estimated background color and a ±30/255 margin, define a binary image with false values in the background and true values in correspondence to the foreground objects.
d) Using a 5 pixels radius disk structuring element apply a morphological closing in order to be sure that each object is represented by no more than one connected region.
e) Create a new image called *labelled* in which a different integer value is assign to each connected component present in the mask image (*hint*: in order to label each connected component present in the mask image use a function already implemented in the MATLAB image processing toolbox).
f) Initialize the number of bolts as zero and for each positive integer values in *labelled* (0 corresponds to the background), perform the following operations:
    I. Obtain a new mask specific for the considered object (*hint*: use a logical condition on *labelled*)
    II. Obtain a rectangle region centered around each object using a 10-pixel margin (*hint*: be careful not to exceed the image limits). Using the obtain region crop the magnitude and directions matrices obtained in b), obtaining two submatrices related to the considered object.
    III. Obtain a vector of all the edge angular directions relative to pixels having an edge magnitude greater than 20% of the maximum edge magnitude value inside the object's region.
    IV. Calculate the histogram of the vector obtain in f).III using 7 equally spaced bins between -90° and 90°. Since +90 and -90 are related to the same direction, sum the values in the last bin with the one of the first one and remove it from the histogram vector.
    V. If the greatest value inside the histogram is greater than 2 times the median one, classify the object as a bolt and increase the bolt count by one.

Solution

```
close all
clear all
%a)
I = im2double(imread('input_image.png'));
figure
imshow(I)
%b)
h_r = fspecial('sobel');
h_c = h_r';
I_x = imfilter(I,h_c);
I_y = imfilter(I,h_r);
I_mag = sqrt(I_x.^2+I_y.^2);
I_dir = rad2deg(atan(I_x./I_y));
%c)
im_hist = imhist(I);
[bg_count bg_c] = max(im_hist);
thr = 30/255;
bg_c = bg_c/255;
mask = I > bg_c+thr | I < bg_c-thr;
%d)
B = strel('disk',5);
mask = imclose(mask,B);
%e)
labelled = bwlabel(mask);
%f)
bolts = 0;
for i=1:max(labelled(:))
    %f.I)
    mask_obj_i = (labelled == i);
    %f.II)
    [r c] = find(mask_obj_i);
    b = 10;
    r_max = min(max(r)+b,size(I,1));
    c_max = min(max(c)+b,size(I,2));
    r_min = max(min(r)-b,1);
    c_min = max(min(c)-b,1);
    crop_mag = I_mag(r_min:r_max,c_min:c_max);
    crop_dir = I_dir(r_min:r_max,c_min:c_max);
    %f.III)
    angle_v = crop_dir(crop_mag>0.2*max(crop_mag(:)));
    %f.IV)
    [h_angle,x] = hist(angle_v,[-90 -60 -30 0 30 60 90]);
    h_angle(1) = h_angle(1) + h_angle(end);
    h_angle = h_angle(1:end-1);
    h_angle = sort(h_angle,'d');
    if(h_angle(1) > 2*median(h_angle))
       bolts = bolts+1;
    end
end
```