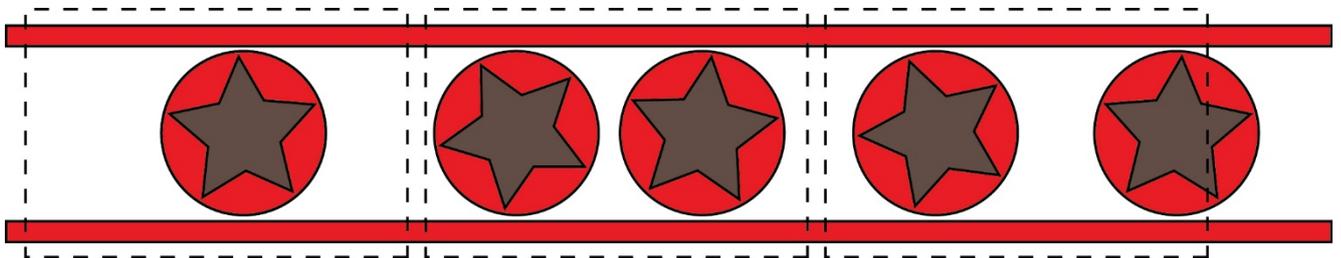


# Video Signals

Date: 17 June 2019

**Ex.1.[8 pt]** A new LCD display has a luminance contrast ratio of 1000:1; assuming a Weber's constant of 0.03 what will be the minimum number of non-uniform quantization levels to avoid any visible intensity transition?  
If the quantization has to be linear what would be the minimum number of levels to satisfy the same constraint?  
Detail every mathematical step.

**Ex.2.[14 pt]** Along a production line we have to monitor bottles moving on a conveyor belt: the bottles are framed by a color camera from the top. Every bottle has a bottle cap with a star shape as seen in the picture. The rectangles with the dashed lines represent possible images acquired by the camera from the top view. The stars of the bottle cap can have different rotations but all the bottles are aligned on the horizontal line and will only move horizontally according to the conveyor belt movement.



Propose and describe in detail a system that is able to localize and count the bottles in an image independently from their horizontal position and the orientation of the star on the bottle cap. Then, in case of the bottle is completely inside the image, the system checks the shape of the bottle cap comparing it with the reference shape of the star independently from its rotation in the image. The system shall throw an alert in case of a wrong shape or a misalignment with the bottle (represented by the circle).

**Es.3. [11 pt MATLAB Exercise]** You are implementing a barcode reader app. You expect the user to acquire an image similar to the one depicted in the figure below. In order to correctly read the barcode, implement a MATLAB script able to rotate the received image so the barcode lines will become vertical.



[Continues overleaf]

- a) Read the input 8-bit color image (*barcode.png*), convert it to a double grayscale image and visualize it.
- b) Define the 3x3 Sobel filters in order to extract horizontal and vertical edges. Apply them to the original image obtaining  $A_{h_r}$  and  $A_{h_c}$ .
- c) Store in two matrixes,  $A_{h\_mag}$  and  $A_{h\_dir}$ , the magnitude and direction of the gradient in each pixel location (hint: consider the canonical convention for angles with zero in the x axes and positive counterclockwise rotation).
- d) Estimate the rotation of the barcode performing the following operations:
  - I. Calculate the threshold value *thr* as the 70% of the grater magnitude value stored in  $A_{h\_mag}$ .
  - II. Select from  $A_{h\_dir}$  the orientation values of the gradients with magnitude greater than *thr*.
  - III. Calculate the histogram of the values obtained in the previous point (hint: in order to obtain the histogram use the default bins and store the bin centers, returned as the second output of the function, in a vector called *angles*).
  - IV. Estimate the most common orientation by selecting the value of *angles* in which the histogram reaches its maximum value.
- e) Convert the estimated angle in degree, obtain the final image applying the correct rotation and visualize the result (hint: remember that in the MATLAB function that perform image rotation by default the rotation is performed counterclockwise).

**Matlab  
List of  
possible  
functions**

figure  
rgb2ind  
im2doubl  
e  
imread  
imrotate  
rgb2gray  
imcrop  
imfilter  
imopen  
imshow  
find  
hist  
min  
atan  
sqrt  
max  
strel  
imnoise  
imdilate  
fspecial

# Solutions

## Ex.1

For the non uniform case:

$$\frac{I_{max}}{I_{min}} = (1 + 0.03)^n = 1000 \rightarrow n \log 1.03 = \log 1000 \rightarrow n \cong 234$$

For the uniform case we would have:  $n = \frac{I_{max} - I_{min}}{0.03} = 33300$

## Ex.2

In order to localize bottles we can adopt a background subtraction method; the background is the image of the conveyor belt when no bottle is over it. Different algorithms for background subtraction have been shown during lectures.

Once we get the foreground mask (pixels set to “true” for foreground objects and “false” for background) we can remove some noise using a morphological closing followed by an opening and then we can find connected foreground objects using a morphological filling algorithm.

The next step is to check that the whole bottle is inside the frame and this can be done simply checking that no pixels of the foreground object touch the image border.

Then for each object, we can go back to the original image and cropping the foreground part. For each single object, we can apply a transformation in polar coordinates with respect to the centre of each circle of the bottle cap. Then, applying the 2D Fourier transform to the transformed images and keeping just its magnitude we got a quite good descriptor of the bottle cap invariant to rotations that can be compare to a reference one. A proper analysis from a set of acquisitions shall be done in order to define the distance threshold between correct and wrong caps.

## Ex.3

```
clc
close all
clear all

%a)
A_rgb = imread('barcode.png');
A = im2double(rgb2gray(A_rgb));
figure()
imshow(A)

%b)
h_r = [1 0 -1;
       2 0 -2;
       1 0 -1];
h_c = [-1 -2 -1;
       0 0 0;
       1 2 1];

A_h_r = imfilter(A, h_r, 'symmetric', 'conv', 'same');
A_h_c = imfilter(A, h_c, 'symmetric', 'conv', 'same');

%c)
A_h_dir = atan(A_h_c./A_h_r);
A_h_mag = sqrt(A_h_r.^2 + A_h_c.^2);

%d)
thr_mag = 0.7*max(A_h_mag(:));
[h,angles] = hist(A_h_dir(A_h_mag > thr_mag));
[vmax,imax] = max(h);

%e)
figure
imshow(imrotate(A_rgb,-180*angles(imax)/pi))
```

