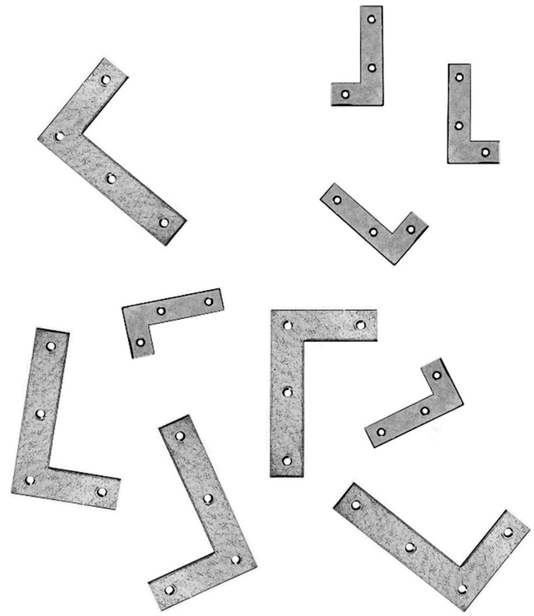
Video Signals

Date: 30 August 2024

Ex.1.[12 pts]

The image on the right shows a series of flat angle brackets placed on a conveyor belt and captured vertically from above using industrial cameras. We are required to develop an algorithm capable of distinguishing brackets of different sizes for automatic counting. The algorithm must be able to classify brackets of the same size into the same category, regardless of their **position**, **rotation**, **or** possible **flipping**. However, it must be able to distinguish between brackets of different **sizes**.



Describe the steps in order to:

- Localize each bracket in the image.
- Extract a suitable descriptor.
- Define a classifier.

For each of the required steps, provide a detailed description of a possible implementation, highlighting any potential challenges or aspects that should be considered during the implementation phase.

Ex.2.[6 pts]

Provide detailed answers to the following questions:

- Why are three independent components sufficient to represent the colors of the visible spectrum?
- Why can negative components of a specific color be necessary in the CIE XYZ color system? What are the practical implications of this?

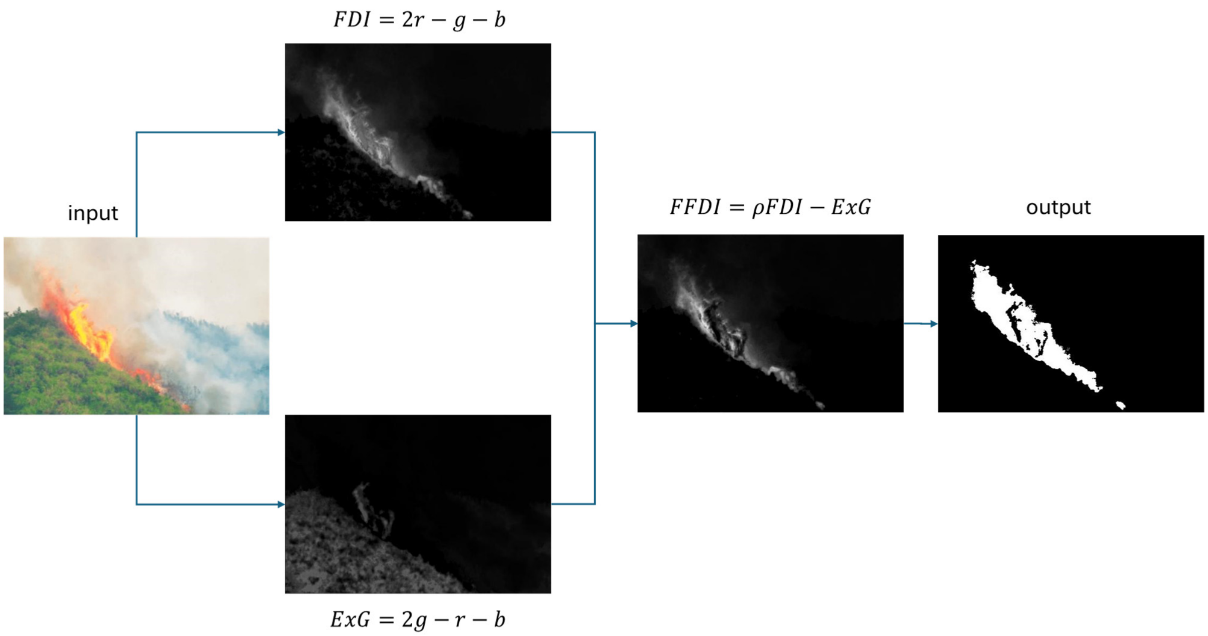
Ex.3.[6 pts]

A display with HDR10 features indicates that it uses 10 bits to represent the intensity of each color. Assuming that the visibility threshold is 2% and coincides with the quantization step of such a display, and the minimum intensity is 0.001 cd/m², what will be the maximum intensity emitted by the display? Justify the answer by detailing the steps.

Ex. 4 is overleaf

Es.4. [11 pts to be solved writing on the paper a suitable MATLAB code]

Current environmental conditions have recently been producing more frequent and severe wildfires, causing the destruction of sizeable forested areas every year. You want to develop an algorithm that is able to locate fire in arial images of forest regions. The algorithm you need to implement is summurized in the image below.



Write a MATLAB script able to perform the following steps:

a) Read the color input image (stored in *'forest.png'* file) and convert it to a double representation.
b) Extract the R,G,B component
c) Normalize R,G,B with the following equations to obtain r,g,b: $r = \frac{R}{R+G+B}$ $g = \frac{G}{R+G+B}$ and $b = \frac{B}{R+G+B}$
d) Calculate the Fire Detection Index (FDI) and the Excess Green colors index (ExG), $FDI = 2r - g - b$ and $ExG = 2g - r - b$
e) Calculate the Forest Fire Detection Index (FFDI) using the following formula: $FFDI = \rho FDI - ExG$ where $\rho = 0.5$
f) Obtain σ_{FDI} and σ_{ExG} as the standard deviations of FDI and ExG respectively and obtain the threshold value T_{FFD} by computing the average between σ_{FDI} and σ_{ExG}
g) Obtain the output binary map by setting to 1 all the locations in which the value of FFDI is greater or equal to T_{FFD}
h) Show on the same plot on the left the original image and on the right the obtained binary map.

List of possible Matlab functions

```
figure
im2double
imread
imcrop
imfilter
imhist
conj
rgb2ind
imagesc
rgb2hsv
imshow
fspecial
fft2
ifft2
subplot
```

Solutions

Es.1

Concerning the localization of brackets in the image, an approach can be applied where different blobs of pixels with intensity different from the background color are localized. After thresholding with the background intensity, the different objects can then be identified by extracting the connected components, for example using morphological connected component extraction starting from a foreground pixel. Since there are background pixels corresponding to the background within the brackets, morphological region filling can then be used within each region previously identified through connected component extraction.

Once these regions are extracted, geometric moments can be computed. Since intensity is not a discriminating parameter, one can operate directly with a pixel value of 1 for the pixels belonging to each bracket and a value of 0 for the external pixels.

Once the centroid of each bracket is evaluated with reference to the image coordinates, the central moments can then be extracted:

The moment is: $m_{pq} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x^p y^q I(x,y)\,dxdy$

The central moment is: $\mu_{pq} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} (x-\overline{x})^p (y-\overline{y})^q I(x,y)\,dxdy$ dove $\overline{x} = \frac{m_{10}}{m_{00}}$ e $\overline{y} = \frac{m_{01}}{m_{00}}$

Normalized central moments should not be used because the objects we want to distinguish have very similar shapes and differ mainly in their size.
The Hu's Moments can then be extracted for each object and the seventh moment should be considered just in magnitude because we can find flipped objects.
Each object will then be converted in a point in a seven-dimensions space and the classification could be based on the closeness (based on Euclidean distance) to samples of each class.

Es.2

Three independent components are sufficient to represent the colors of the visible spectrum due to the way human vision works and the properties of light.

- Human Vision and Trichromacy:

  The human eye has three types of cone cells, each sensitive to different ranges of wavelengths in the visible spectrum. These cones are commonly referred to as:
  S-cones (short wavelengths, peaking in the blue region),
  M-cones (medium wavelengths, peaking in the green region),
  L-cones (long wavelengths, peaking in the red region).
  The brain interprets the signals from these three types of cones to create the perception of color. Because our visual system relies on these three types of receptors, we can represent any color we perceive using a combination of three components corresponding to the stimulation of these cones.
- Color Space Representation:

  Colors can be described in a three-dimensional space, often referred to as a color space. Common examples include the RGB (Red, Green, Blue) color model, where each color is represented by a combination of three values corresponding to the intensity of red, green, and blue light.
  These three components are sufficient to represent the vast range of colors perceivable by humans, as they align with the trichromatic nature of human vision.

Negative components of a specific color can be necessary in the CIE XYZ color system due to the mathematical properties of the color space and the way it was designed to encompass all perceivable colors, even those that cannot be physically produced by a single light source.

- Understanding the CIE XYZ Color System:

  The CIE XYZ color system is a linear color space developed by the International Commission on Illumination (CIE) in 1931. It is designed to represent all colors perceivable by the human eye using three components: X, Y, and Z.

  The XYZ color space is based on a set of color matching functions that describe how a color can be matched by combining different amounts of three primary colors. These primary colors, however, are not real physical colors but rather hypothetical ones designed to simplify color measurement and encompass all visible colors.
- 2. Reason for Negative Components:

  The XYZ color system is designed to cover all colors visible to the human eye, even those outside the gamut of real-world display systems (like monitors or printers) or physical primaries.

  Because of this, some colors in the real world can only be represented in this system by using a combination of these hypothetical primaries, which can result in negative values for one or more components.

  In practical terms, this occurs because the XYZ primaries are linear combinations of actual color matching functions, which means that some real colors, when transformed into the XYZ space, may fall outside the positive range of the X, Y, or Z axes.
- 3. Practical Implications:

  Color Gamut Limitations: If you're working within a device-dependent color space like RGB, which has a limited gamut (range of colors it can reproduce), a color that requires a negative XYZ component cannot be reproduced exactly by that device. This might lead to approximations or adjustments when converting colors to fit within the RGB gamut.

  Color Conversion: When converting from XYZ to another color space (like RGB), if one of the XYZ components is negative, the conversion algorithm might need to clip or map these values into the RGB space, which can lead to a loss of color fidelity or differences in appearance.

  Color Calibration: In some color reproduction tasks, like printing or display calibration, the presence of negative components indicates that the color cannot be reproduced using the standard colorants or lights available. Special care must be taken to approximate such colors or to manage expectations about color accuracy.

  Metamerism: In practice, negative components highlight the fact that certain colors can only be matched under specific conditions or by using specific combinations of lights, which is crucial in industries like printing and manufacturing, where color consistency is key.

  Conclusion:

  Negative components in the CIE XYZ color system arise due to the mathematical design of the system to cover all visible colors, including those outside the gamut of real-world physical primaries. While necessary for theoretical completeness, they present practical challenges in color reproduction, conversion, and calibration, especially when working with device-specific color spaces like RGB.

Es.3

The intensity quantization in displays, [particularly for HDR (High Dynamic Range) like HDR10], is typically done in a logarithmic manner rather than a linear one. This is because human perception of brightness is more sensitive to relative changes in intensity (logarithmic scale) rather than absolute changes (linear scale) (see Weber's law in the lectures).

- **HDR10 display:** Uses 10 bits per color channel, providing 1024 levels.
- **Visibility threshold:** 2% of intensity.
- **Minimum intensity:** $I_{min} = 0.001\, cd/m^2$.
- The intensity scale is logarithmic.

Since the intensity is quantized logarithmically, each step corresponds to a multiplicative factor (rather than an additive one as in linear quantization).

$$I_n = I_{min} \cdot \left(\frac{I_{max}}{I_{min}} \right)^{\frac{n}{1023}}$$

The maximum intensity is then: $I_{max} = I_{min} \cdot (1.02)^{1023} = 1.0857\, cd/m^2$

Es.4

```
clc
close all
clear all

%a)
I = im2double(imread('forest.png'));

%b)
R = I(:,:,1);
G = I(:,:,2);
B = I(:,:,3);

%c)
r = R./(R+G+B);
g = G./(R+G+B);
b = B./(R+G+B);

%d)
FDI = 2*r - g - b;
ExG = 2*g - r - b;

%e)
rho = 0.5;
FFDI = rho*FDI - ExG;

%f)
s_FDI = std(FFDI(:));
s_ExG = std(ExG(:));
T_FFD = (s_FDI + s_ExG)/2;

%g)
out = FFDI >= T_FFD;

%h)
figure
subplot(1,2,1)
imshow(I)
title('Original image')
subplot(1,2,2)
imshow(out)
title('Fire regions')
```