POLITÉCNICO DE LEIRIA | ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO

MEE / MEEE

Visão Computacional / Computer Vision

Histograms and Camera Calibration

1. Histograms

Histogram is as a graph or plot, that provides the general idea about the intensity distribution of an image. It is a plot with pixel values (ranging from 0 to 255 for 8-bits depth) in X-axis and corresponding number of pixels in the image on Y-axis. From the histogram observation of an image, you easily get intuition about contrast, brightness, intensity distribution etc. of that image. We are exploring this subject for RGB color space, but keep in mind that can be applied to all the other color spaces introduced in the previous classes.

1.1 Consider the grayscale image present in the tutorial folder (python_opencv_gray.png):

To be able to plot the image and histogram side-by-side, consider the usage of matplotlib's **subplot()** function for multiple plots.

a) Apply the matplotlibs's **plt.hist()** function to obtain the histogram (this is probably the fastest method to display it).
b) Similarly, you can use the numpy's function **np.histogram()** and display it. This function requires some reshape of the image data.
c) Now, explore the in-built function in OpenCV to find the histogram, **cv2.calcHist()**.
d) Let's perform histogram equalization (that improves the contrast in an image, to stretch out the intensity range), applying **cv2.equalizeHist()** function.
e) (**Optional**) Explore the previous scripts with your own images (or with the low contrast in the tutorial folder "landscape_gray.jpg").

1.2 Consider the colored image present in the tutorial folder (python_opencv.png):

a) Apply the previous explored **cv2.calcHist()** to the color image present in the tutorial folder to obtain the color histogram.

b) Perform histogram equalization in the color image, applying **cv2.equalizeHist()** function.

Note: The most naïve approach is to process each RGB channel individually, although this process changes the relative distributions of the color and may consequently yield to dramatic changes in the image's color balance. As alternative, ones could convert the image to the HSV color space and then apply the histogram equalization only on the lightness or value channel by leaving the hue and the saturation of the image unchanged.

c) (**Optional**) Explore the previous scripts with your own images (or with the low contrast in the tutorial folder “landscape.jpg”).

2. Camera calibration

Camera calibration is process that sometimes is required due to some cameras introduce significant distortion to images. Two major kinds of distortion are radial distortion and tangential distortion. The radial distortion causes straight lines to appear curved, becomes larger the farther points are from the center of the image. Radial distortion is represented as follows:

$$x_{distorted} = x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6)$$
$$y_{distorted} = y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6)$$

Tangential distortion occurs due to the imperfect alignment between the image-taking lens and the imaging plane (not perfectly parallel), so some areas in the image may look nearer than expected. The amount of tangential distortion can be represented as below:

$$x_{distorted} = x + [2p_1 xy + p_2(r^2 + 2x^2)]$$
$$y_{distorted} = y + [p_1(r^2 + 2y^2) + 2p_2 xy]$$

This result in 5 parameters, named as distortion coefficients, given by:

$$Distortion\ coefficients = (k_1 \quad k_2 \quad p_1 \quad p_2 \quad k_3)$$

In addition to these parameters, it is necessary to must obtain other information, such as the intrinsic and extrinsic parameters of the camera. Intrinsic parameters are

specific to a camera (depends on focal length, optical centers, skew), and are used to obtain the camera matrix.

$$camera\ matrix = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

Extrinsic parameters correspond to rotation and translation vectors which translates a coordinate of a 3D point to a coordinate system. You can also check this IPLeiria MEE students tutorial here 😊.

2.1 Explore the camera calibration example in the tutorial folder

a) In this part of the tutorial is presented an example of an entire camera calibration process. Your task is to evaluate such code, and go to the process of explore the functions used to perform such task, namely cv2.findChessboardCorners(), cv2.cornerSubPix(), cv2.calibrateCamera(), and so on.

b) (**Optional**) Explore the previous script with your own distorted images, gathered from your own cameras, with a chessboard.