Advanced Image Processing - Camera calibration and histograms

Ing. Viktor Kocur
viktor.kocur@fmph.uniba.sk

DAI FMFI UK

16.10.2019

Calibration process

Calibration pattern

For Matlab calibration a checkerboard calibration pattern is used. It is possible to use a different pattern, but its detection has to be implemented. The whole calibration pattern has to be visible in the photos.

Taking photos

When taking photos it is important to not change the camera configuration. It is therefore necessary to turn off features such as autofocus and to not use zoom. Such changes can change some of the intrinsic parameters of the camera.

Calibration App

cameraParams

We will use the Matlab Calibration Map to obtain the camera parameters via the struct cameraParams.

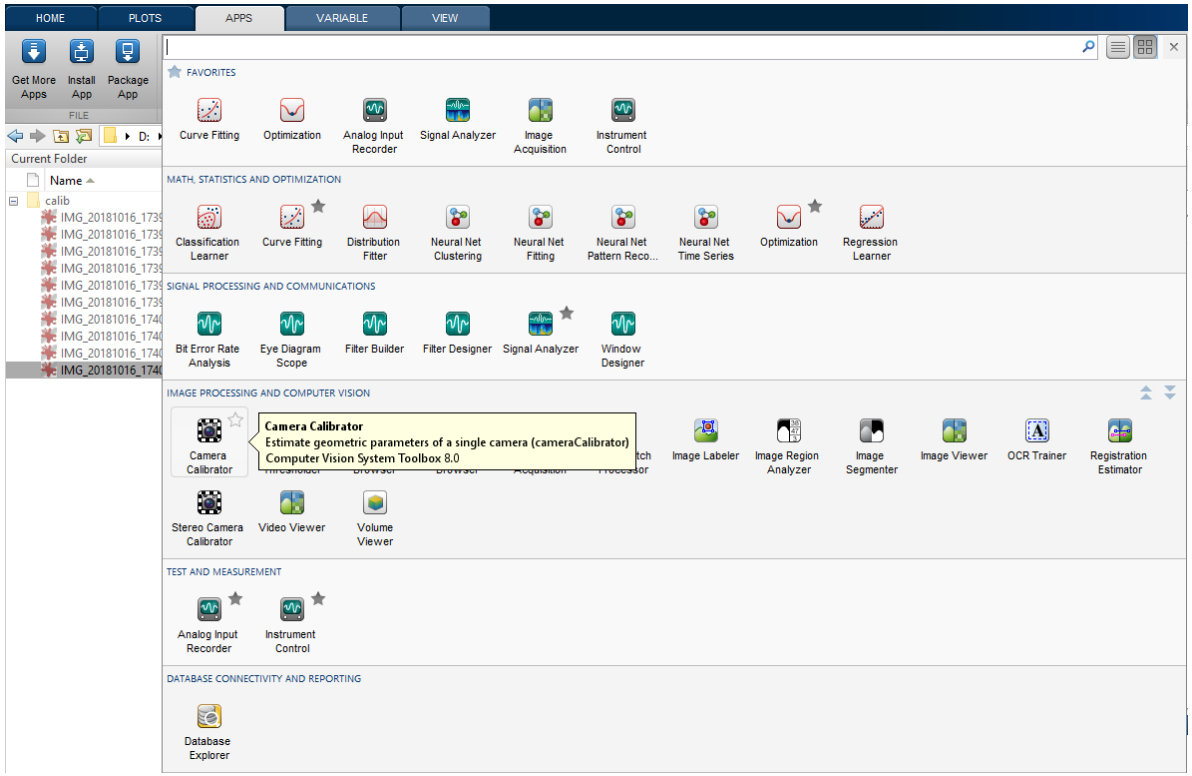

Image processing

undistort

[im, newOrigin] = undistortImage(I, cameraParams) - returns image without distortion. newOrigin indicates the shift of origin between the coordinate system in the image and the undistorted image. In case of this value being [0, 0] it can be ignored.

Exercise

Display the undistorted image for one of the captured images. Try to use undistortImage(I, cameraParams, 'OutputView', 'full').

Pattern detection

detectCheckerboardPoints

[imagePoints, boardSize] = detectCheckerboardPoints(im) - returns positions of the corners in the calibration pattern

detectCheckerboardPoints

If there was a shift of the origin in the undistortion step it is necessary to shift all of these points with the new origin, e.g. imagePoints = imagePoints + newOrigin

generateCheckerboardPoints

worldPoints = generateCheckerboardPoints(boardSize, squareSize) - returns the positions of the points in real world coordinates. The parameter squareSize indicates the size of the square in millimeters.

Extrinsic parameters

extrinsics

[R, t] = extrinsics(imagePoints, worldPoints, cameraParams) - returns the rotation matrix R and the translation vector t for the given pairs of points and intrinsic camera parameters.

Note

If we try to apply this to an image we used in the app it is possible to obtain these values from the cameraParams struct.

Point transformation

pointsToWorld

pointsToWorld(cameraParams, R, t, imPoints) - returns the coordinates of the imPoints in the coordinate system of the plane of the calibration pattern. The arguments are the camera parameter struct, the rotation matrix, the translation vector and the points in the coordinate system of the image.

Pozor!

If we use imPoints from the ginput() function it is important to correct for the origin shift by adding the newOrigin value,

Exercise

Use ginput and measure dimensions of an object in the image and check if this measurement corresponds to real world dimensions of the object.

Histogram

imhist

imhist(I) - displays the histogram, in case the output is directed to a variable the histogram is not shown and the values are stored in the variable as a vector.

Exercise

Convert the zatisie.jpg to grayscale (rgb2gray function). In its histogram we can see three peaks. Change the image so that the values belonging to just one of the three peaks are fully white (maximum intensity).

Intensity processing

Gamma correction

Contrast in the image can be changed with the gamma correction: $i_{out} = A \cdot i^{\gamma}$, where i are the intensities of the pixels of the image. The intensity values are expected to be between 0 and 1!

Linear expansion

To linearnly expand the intensity we can use the following:

$$i_{out} = \frac{i - max(I)}{max(I) - min(I)},$$

where i are intensity values of the pixels and I is the set of all the intensities in the image. We expect the values to be between 0 and 1.

Histogram equalization

Equalization

Histogram equalization is a method to change the intensity values in the image in a way that results in the most balanced histogram (closest to the uniform distribution).

histeq

histeq(I) - returns the image after histogram equalization.

Exercise

For the image krajinka.jpg use all of the 3 methods to change the intensity values. Display both the image and the corresponding histograms.

Thresholding

imbinarize

imbinarize(I) - returns the binarized image using the Otsu method.

imbinarize

imbinarize(I, t) - returns the binarized image with the threshold t.

Exercise

Try to use thresholding on images coins.png, qr.jpg a zatisie.jpg.

Adaptive thresholding

imbinarize

imbinarize(I, 'adaptive') - returns binarized image using adaptive thresholding.

Exercise

Try adaptive thresholding for the images coins.png a qr.jpg.