

ECEn 631 Camera Calibration & Distortion Correction

Objectives:

- Learn camera geometry.
- Learn to calibrate camera using OpenCV functions.
- Learn to undistort image from camera.
- Learn to perform pose estimation of a known object.
- You can use either C++ or Python version. All functions shown in this document are C++ functions.

Instructions:

- Use OpenCV and a webcam to complete this assignment.
- Generate a PDF file that includes (with proper headings) one image (Task 1), intrinsic and distortion parameters (Task 2), three difference images (Task 3), rotation and translation matrices (Task 4), intrinsic and distortion parameters (Task 5), one absolute difference image (Task 6), and your explanations.
- Submit your PDF file and source code file(s) in one zip file without the folder or directory.
- Use your first name and last name (e.g., justinsmith.zip) as the file name.
- Login to myBYU and submit your zip file through Learning Suite online submission.

Task 1: Corner Detection 10 points

- ~~Download Camera Calibration Images from Learning Suite.~~
- ~~Write your code to read in one of those calibration images.~~
- ~~The input image must first be converted to grayscale using `cvtColor()` function (CV_RGB2GRAY).~~
- ~~Use OpenCV function `findChessboardCorners()` to find chessboard inner corners.~~
- ~~Use OpenCV function `cornerSubPix()` to refine corner locations.~~
- ~~Use OpenCV function `drawChessboardCorners()` to draw corners (convert grayscale back to color before drawing).~~
- ~~Include **one** output image with corners circled in color in your PDF file.~~
- ~~This task is only an intermediate stage of calibration procedure. You don't have to submit your code.~~

Task 2: Intrinsic Parameters 20 points

- ~~Write a program to read in all 40 of the calibration images one at a time in a loop.~~
- ~~In the loop, find chessboard corners for each input image.~~
- ~~Arrange corner points in the format for `calibrateCamera()` function. You need to learn how to use vector and vector of vectors if you use C++. A PDF file (Using Vectors and Mat.pdf) can be found in Course Material/Reference on Learning Suite.~~
- ~~Use OpenCV function `calibrateCamera()` to calculate the intrinsic and distortion parameters.~~
- ~~The camera spec sheet says that the pixel size is $7.4\mu\text{m} \times 7.4\mu\text{m}$ and the sensor size is $4.8\text{mm} \times 3.6\text{mm}$ (1/3" format). **Calculate and report the actual focal length in mm.**~~
- ~~Include the intrinsic (3×3) and distortion (5×1) parameters in your PDF file. (5-point deduction if not in these formats)~~
- ~~Submit your code for this task.~~

Task 3: Distortion Correction 20 points

- ~~Read in your saved intrinsic and distortion parameters from file(s).~~
- ~~Download the three test images (Far, Close, Turned).~~
- ~~Use OpenCV function `undistort()` to correct the distortion of these three images.~~
- ~~Use OpenCV function `absdiff()` to compute the absolute difference between the original and undistorted images.~~
- ~~Include all three absolute difference images in your PDF file.~~
- ~~Submit your code for this task.~~

Task 4: Object Pose Estimation 20 points

- ~~Download the "Object with Corners" image to see the known object. You don't have to process this image. Data are provided.~~
- ~~Download the data file DataPoints.txt that has 20 image points (for x and y in pixels) and 20 object points (for x, y, z in inches).~~
- ~~Write a program to read in the image and object points and use your calibration parameters from Task 2.~~
- ~~Use the C++ version `solvePnP()` function or C version `cvFindExtrinsicCameraParams2()` (not preferable) to estimate the object pose (measured by the camera). **Convert R and T to 3×3 and 3×1 matrix (5-point deduction if not in these formats).**~~
- ~~Include the output rotation in both (3×3) and (3×1) and translation in (3×1) in your PDF file.~~
- ~~Submit your code for this task.~~

Task 5: Intrinsic Parameters 15 points

- ~~• Repeat Task 2 (including saving calibration parameters in a file) using your own camera.~~
- ~~• You can use your real-time acquisition code for Assignment 1 to capture images.~~
- ~~• Use the chessboard for Assignment 1 (blocks must be square) and your code for Task 2 above to calibrate your camera.~~
- ~~• Make sure to change the number of corners entered to the calibration function in your code for Task 2.~~
- ~~• Make sure the chessboard paper is on a planar surface.~~
- ~~• Include the intrinsic (3×3) and distortion (5×1) parameters in your PDF file. (5-point deduction if not in these formats)~~
- ~~• Submit your code for this task.~~

Task 6: Distortion Correction 15 points

- ~~• Repeat Task 3 (including reading calibration parameters from a file) using your own camera.~~
- ~~• Include an image of the absolute difference between the original (captured from your camera) and the undistorted images.~~
- ~~• Submit your code for this task.~~