COMP/ENGN6528 Computer Vision - 2023 S1

Computer Lab 3 (CLab-3)

!!! This is just a template for CLab-3. Please also make sure you follow all the requirements in Clab-3 assignment file if it is not mentioned in this template !!!

!!! Please also note that the sample outputs given in this document are not guaranteed to be correct. They just gives you an idea of how your outputs should look like !!!

**Task-1: 3D-2D Camera Calibration (17 marks)**

(Acknowledgement: Lab material courtesy of Professor. Du Huynh of UWA).

Camera calibration involves finding the geometric relationship between 3D world coordinates and their 2D projected positions in the image.

Four images, [stereo2012a.jpg](http://undergraduate.csse.uwa.edu.au/units/CITS4240/Images/stereo2012a.jpg), [stereo2012b.jpg](http://undergraduate.csse.uwa.edu.au/units/CITS4240/Images/stereo2012b.jpg), [stereo2012c.jpg](http://undergraduate.csse.uwa.edu.au/units/CITS4240/Images/stereo2012c.jpg), and [stereo2012d.jpg](http://undergraduate.csse.uwa.edu.au/units/CITS4240/Images/stereo2012d.jpg), are given for this CLab-3. These images are different views of a calibration target and some objects. For example, the diagram below is **stereo2012a.jpg** with some text superimposed onto it:

A picture containing text

Description automatically generated   
(Do not directly use the above image for your camera calibration work   
as it has been scaled for illustration. Use the original (unlabelled) image files provided.)

On the calibration target there are 3 mutually orthogonal faces. The points marked on each face form a regular grid. They are all 7cm apart.

Write a Matlab function with the following specification

Function to perform camera calibration

|  |  |  |
| --- | --- | --- |
| Function C = calibrate(im, XYZ, uv) | | |
| Input: | im: | is the image of the calibration target. |
|  | XYZ: | is a Nx3 array of XYZ coordinates of the calibration target points. |
|  | uv: | is a N x 2 array of the image coordinates of the calibration target points. |
| Outputs: | C: | is the 3 x 4 camera calibration matrix. |

The variable N should be an integer greater than or equal to 6.

This function should also plot the uv coordinates onto the image of the calibration target. It also projects the XYZ coordinates back into image coordinates using the calibration matrix and plots these points too as a visual check on the accuracy of the calibration process.

The mean squared error between the positions of the uv coordinates and the projected XYZ coordinates is also reported.

Lines from the origin to the vanishing points (namely, world coordinate system) in the X, Y and Z directions are overlaid on the image.

Generally, we ask you to implement a function:

MATLAB user:

function C = calibrate(im, XYZ, uv)

Python user:

def calibrate(im, XYZ, uv)

return C

(Below is just an instruction, you don’t need to show anything here)

Your function should be able to:

1. Output the 3x4 calibration matrix
2. Plot the chosen uv coordinates onto the image (you can annotate them using blue circles)
3. Project the chosen 3D points onto the image and plot them (annotate them using red circles)
4. Calculate the mean squared error between the uv coordinates and the corresponding projected coordinates.
5. For each direction (i.e X, Y, Z direction), draw a line connecting the origin of world coordinate system and a vanishing point in that direction. Show these 3 lines in the image. (annotate lines using green lines)

Sample output of your function:

* Camera calibration matrix:
* The means square error between the chosen uv coordinates and the corresponding projected points is: 1.2345

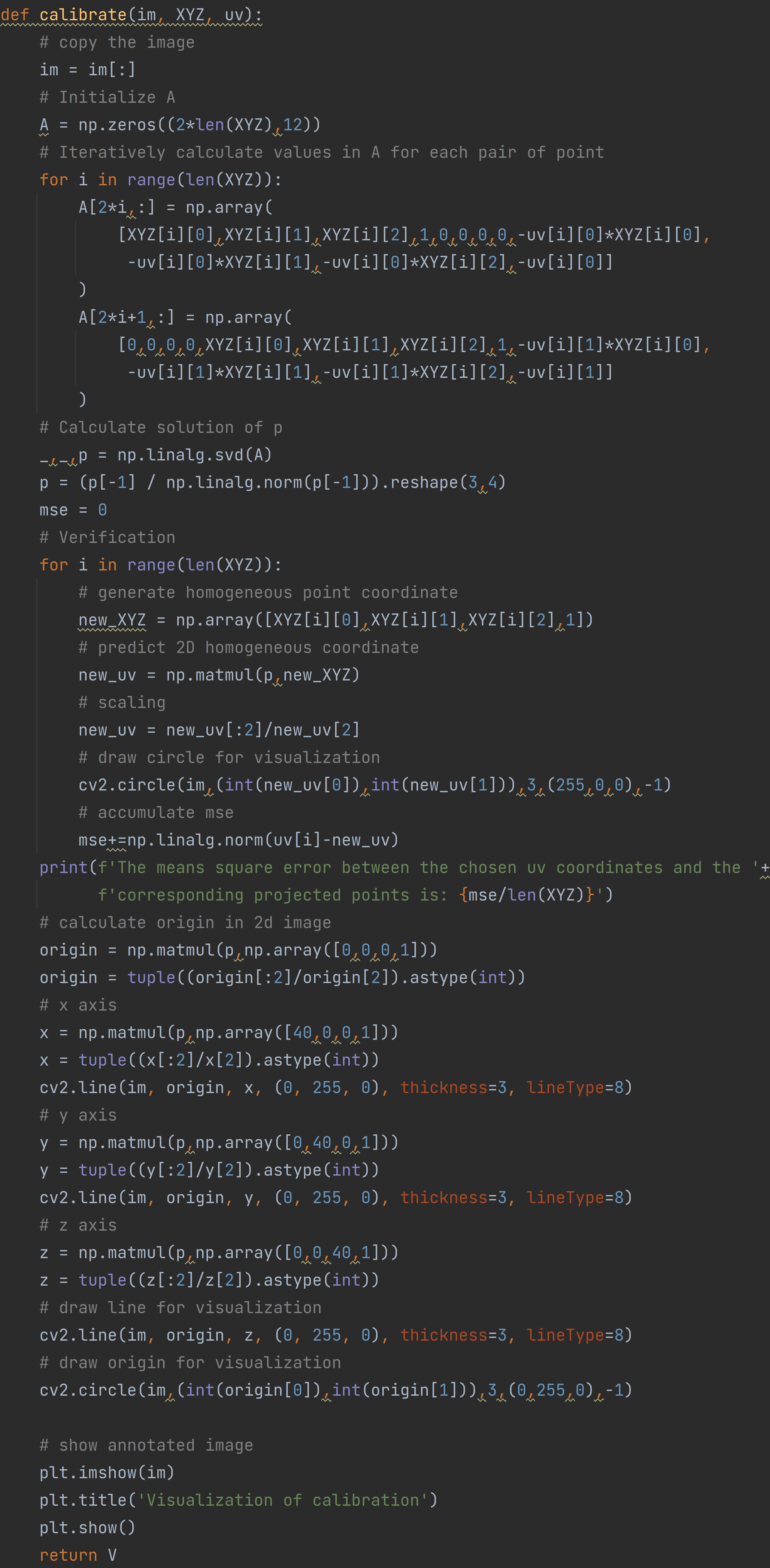
A picture containing text, indoor, wall, floor

Description automatically generated

**For Task-1, you should include the following in your Lab-Report PDF file:**

1. List calibrate function in your PDF file. [3 marks]

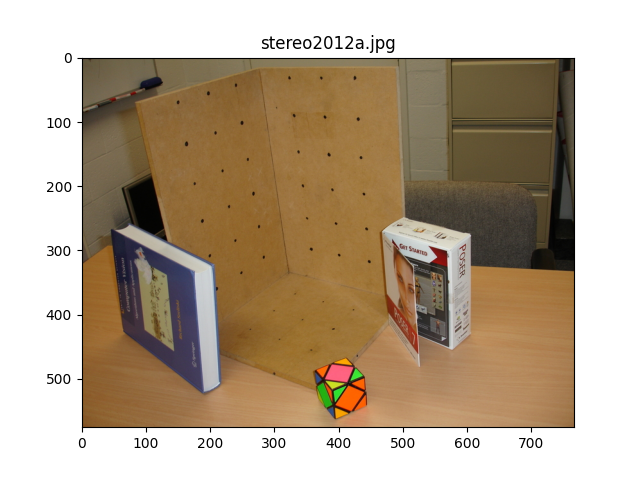
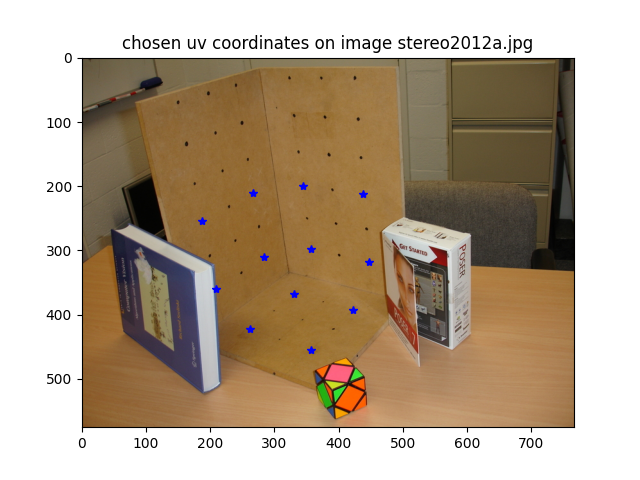
Complete the calibrate function and attach a screenshot of your function here.



|  |  |
| --- | --- |
| **Mistake** | **Marks Deducted** |
| Not return calibration matrix | -1 |
| Not plot uv coordinates | -0.5 |
| Not plot projected 3D coordinates | -0.5 |
| Not plot the require lines | -0.5 |
| Not calculate the mean square error | -0.5 |
| Incorrect implementation | -3 |

1. List the image you have chosen for your experiment, and display the image in your PDF file. [0.5 mark]

Image stereo2012a.jpg is used for the algorithm. The selected uv position is plotted below:

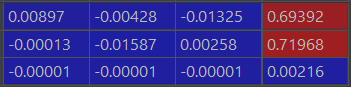
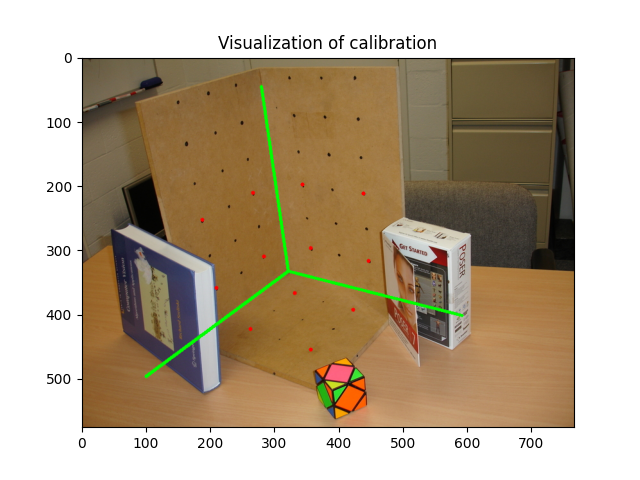
|  |  |
| --- | --- |
| **Mistake** | **Marks Deducted** |
| Not show the chosen image | -0.5 |

1. List the 3x4 camera calibration matrix P that you have calculated for the selected image. Please visualise the projection of the XYZ coordinates back onto image using the calibration matrix P and report the reprojection error (The mean squared error between the positions of the uv coordinates and the projected XYZ coordinates using the estimated projection matrix)[2 marks]

Show three things here (please refer to the sample output above):

* The 3x4 camera calibration matrix produced by your function.
* The annotated image produced by your function.
* The reprojection error produced by your function.

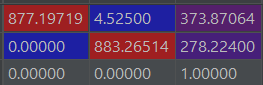
Camera calibration matrix:

* 
* The means square error between the chosen uv coordinates and the corresponding projected points is: 0.43931130052333933
* The plot for the projected XYZ coordinates and axis lines from origin is shown below:
* 

|  |  |
| --- | --- |
| **Mistake** | **Marks Deducted** |
| Incorrect calibration matrix | -1 |
| Incorrect annotated image | -0.5 |
| Incorrect reprojection error | -0.5 |

1. Decompose the P matrix into K, R, t, such that P = K[R|t], by using the following provided code (vgg\_KR\_from\_P.m or vgg\_KR\_from\_P.py). List the results, namely the K, R, t matrices, in your PDF file. [1.5 marks]

Show 3 matrices here: K, R and t







|  |  |
| --- | --- |
| **Mistake** | **Marks Deducted** |
| Incorrect matrix | -0.5 for each matrix |

1. Please answer the following questions:

- what is the focal length (in the unit of pixel) of the camera? [1 mark]

Show the focal length here.

878.8435148177524

|  |  |
| --- | --- |
| **Mistake** | **Marks Deducted** |
| Incorrect focal length | -1 |

- What is the pitch angle of the camera with respect to the X-Z plane in the world coordinate system? (Assuming the X-Z plane is the ground plane, then the pitch angle is the angle between the camera's optical axis and the ground-plane.) Please provide the calculation process [2 marks]

Show the pitch angle here.

|  |  |
| --- | --- |
| **Mistake** | **Marks Deducted** |
| Incorrect angle | -2 |

- What is the camera centre coordinate in the XYZ coordinate system (world coordinate system)? Please provide the calculation process [1 mark]

Show the camera centre coordinate (in world coordinate system) here.

|  |  |
| --- | --- |
| **Mistake** | **Marks Deducted** |
| Incorrect coordinate | -1 |

1. Please resize your selected image using builtin function from matlab or python to (H/2, W/2) where H, and W denote the original size of your selected image. Using the interface function, (ginput in Matlab, and matplotlib.pyplot.ginput in Python) to find the uv coordinates in the resized image. [1 mark]

Display your code for resize the image and picking the uv coordinates of the resized image here.

|  |  |
| --- | --- |
| **Mistake** | **Marks Deducted** |
| Resized image has an incorrect size | -0.5 |

a. Please display your resized image in the report, list your calculated 3x4 camera calibration matrix P’ and the decomposed K’, R’, t’ in your PDF file. [2 marks]

Show 5 things here:

* The resized image
* The calibration matrix P’
* The intrinsic matrix K’
* The rotation matrix R’
* The translation vector t’

|  |  |
| --- | --- |
| **Mistake** | **Marks Deducted** |
| Not show the resize image | -0.4 |
| Incorrect matrix | -0.4 for each matrix |

b. Please analyse the differences between 1) K and K’, 2) R and R’, 3) t and t’. Please provide the reasoning when changes happened or there are no changes. .[2 marks]

Your discussion should include:

- Analyse the differences

- Provide reasoning when changes happened or there are no changes.

|  |  |
| --- | --- |
| **Mistake** | **Marks Deducted** |
| Not analyse the difference Or incorrect discussion. | -1 |
| Not provide reasoning Or incorrect discussion. | -1 |
| Discussion is partially correct | -0.5 |

c. Let us check the focal length (f and f’) (in pixel unit) and the principal points extracted from K and K’, respectively. Please discuss their relationship between (f and f’) and its connection to the image size of the original image and the one after resizing.[2 marks]

Your discussion should include:

- Discuss the relationship between f and f’, and the principal points from K and K’

- Discuss its connection to the image size of the original image.

|  |  |
| --- | --- |
| **Mistake** | **Marks Deducted** |
| Not discuss the relationship Or incorrect discussion | -1 |
| Not discuss the connection to image size Or incorrect discussion | -1 |
| Discussion is partially correct | -0.5 |

**Task-2: Two-View DLT based homography estimation**. (10 marks)

A transformation from the projective space P3 to itself is called homography. A homography is represented by a 3x3 matrix with 8 degree of freedom (scale, as usual, does not matter)

Shape

Description automatically generated with medium confidence

The goal of this task is to the DLT algorithm to estimate a 3x3 homography matrix.

A large white building

Description automatically generated

Pick any **6** corresponding coplanar points in the images left.jpg and right.jpg and get their image coordinates.

In doing this step you may find it useful to check the Matlab function ginput.

Calculate the 3x3 homography matrix between the two images, from the above 6 pairs of corresponding points, using DLT algorithm. You are required to implement your function in the following syntax.

Function to calculate homography matrix

H = homography (u2Trans, v2Trans, uBase, vBase)

|  |  |  |
| --- | --- | --- |
| Usage: | Computes the homography H applying the Direct Linear Transformation | |
| Inputs: | u2Trans, v2Trans: | are vectors with coordinates u and v of the transformed image point (p') |
|  | uBase, vBase: | are vectors with coordinates u and v of the original base image point p |
| Output: | H: | is a 3x3 Homography matrix |

In doing this lab task, you should include the following in your lab report:

1. List your source code for homography estimation function and display the two images and the location of six pairs of selected points (namely, plotted those points on images). Explain what you have done for the homography and what is shown in the images. [5 marks]

Show:

* Screenshot of your homography estimation function.
* Display the two images with the chosen points annotated on the images.

Explain what you have done to estimate the homography matrix and what is shown.

|  |  |
| --- | --- |
| **Mistake** | **Marks Deducted** |
| Incorrect homography estimation | -3 |
| Not display two images | -1 |
| No explanation Or incorrect explanation | -1 |
| Discussion is partially correct | -0.5 |

1. List the 3x3 camera homography matrix H that you have calculated. [2 mark]

Show the homography matrix produced by your function.

|  |  |
| --- | --- |
| **Mistake** | **Marks Deducted** |
| Incorrect homography | -2 |

1. Warp the left image according to the calculated homography. Study the factors that affect the rectified results, e.g., the distance between the corresponding points, e.g the selected points and the warped ones. [3 mark] (Note: you can use builtin image warping functions in matlab and python.)

Show the warped image.

Discuss the factors that affect the rectified results.

|  |  |
| --- | --- |
| **Mistake** | **Marks Deducted** |
| Not show the warped image Or Incorrect warped image | -1.5 |
| No discussion Or Incorrect discussion | -1.5 |
| Discussion is partially correct | -0.5 |

**====================== End of CLab-3 ====================**