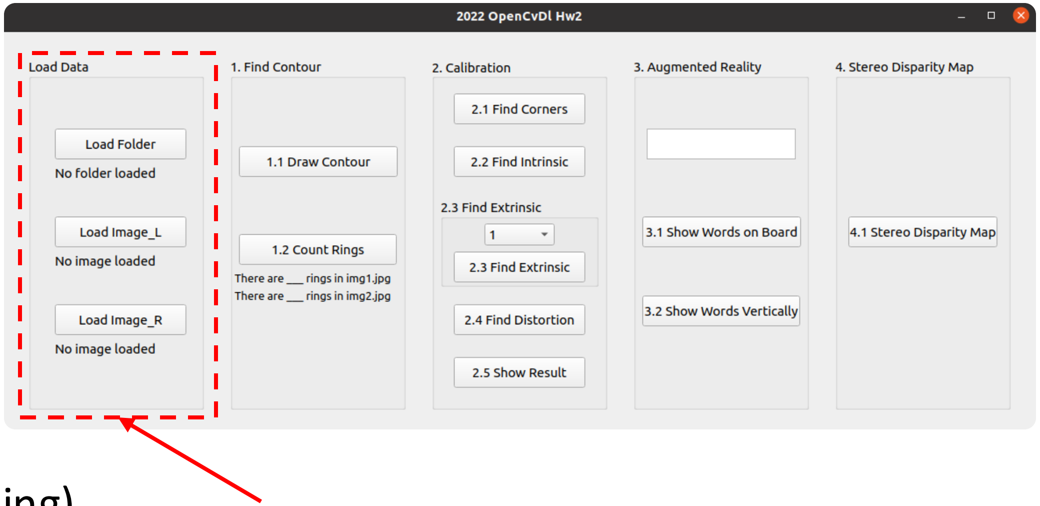
Notice (2/2)

- Python
  - ➢ Python 3.7 (https://www.python.org/downloads/)
  - ➢ opencv-contrib-python (3.4.2.17)
  - ➢ Matplotlib 3.1.1
  - ➢ UI framework: pyqt5 (5.15.1)

Assignment scoring (Total: 100%)

1. (20%) Image Processing (出題：Mei)
   - 1.1 (15%) Draw Contour
   - 1.2 (5%) Count Rings
2. (20%) Camera Calibration (出題：Jessica)
   - 2.1 (4%) Corner detection
   - 2.2 (4%) Find the intrinsic matrix
   - 2.3 (4%) Find the extrinsic matrix
   - 2.4 (4%) Find the distortion matrix
   - 2.5 (4%) Show the undistorted result
3. (20%) Augmented Reality (出題：Ming)
   - 3.1 (10%) Show words on board
   - 3.2 (10%) Show words vertically
4. (20%) Stereo Disparity Map (出題：Maton)
   - 4.1 (10%) Stereo Disparity Map
   - 4.2 (10%) Checking the Disparity Value
5. (20%) Train a Cat-Dog Classifier Using ResNet50 (出題：Benjamin)
   - 5.1 (3%) Load the dataset and resize images
   - 5.2 (3%) Plot class distribution of training dataset
   - 5.3 (3%) Show the structure of ResNet50 model
   - 5.4 (3%) Set up 2 kinds of loss functions to train 2 ResNet50 models
   - 5.5 (3%) Compare the accuracies of 2 ResNet50 models on validation dataset
   - 5.6 (4%) Use the better-trained model to run inference and show the predicted class label



Don’t fix your data path
(There is another dataset for demonstration)

Question 5 needs to upload separately.

1. (20%) Find Contour

(出題：Mei)

1.1 (15%) Draw Contour

1.2 (5%) Count Rings

1.1 Find Contour – Draw Contour

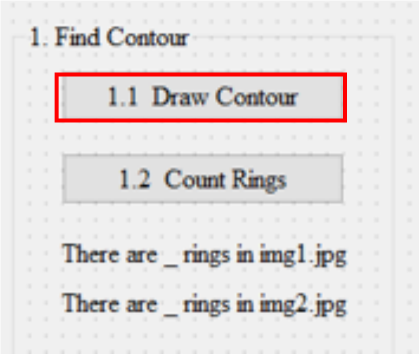
(出題：Mei)

□ Given: two color images, “img1.jpg” and “img2.jpg”

□ Q: 1) **Draw Contour**: Using OpenCV functions to find the contours of rings in two images.

□ Hint: Textbook Chapter 8, p.234 ~ p.241

1. RGB ➔ Resize(1/2) ➔ Grayscale ➔ Binary
2. Remember to remove the noise. (use **Gaussian Blur & other function**)
3. Using some **edge detection functions** to get better results. (Ex: cv2.Canny)



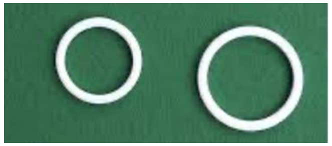
1. Find Contour

1.1 Draw Contour

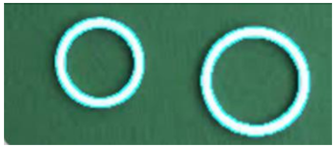
1.2 Count Rings

There are _ rings in img1.jpg

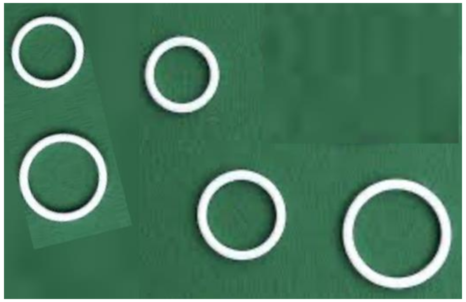
There are _ rings in img2.jpg



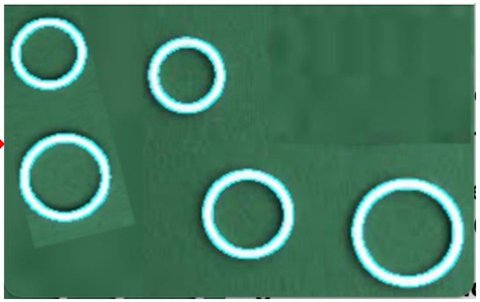
img1.jpg



Draw Contours



img2.jpg



Draw Contours

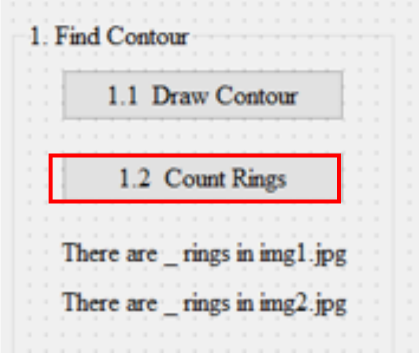
1.2 Find Contour – Count Rings

(出題：Mei)

❑ Given: two color images, “img1.jpg” and “img2.jpg”

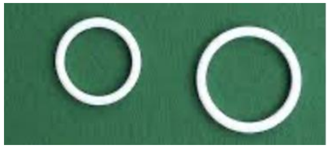
❑ Q: 2) **Count Rings**: Using OpenCV functions to find how many rings in two images.

❑ Hint: Textbook Chapter 8, p.234 ~ p.241
Calculate how many rings (contour/2)

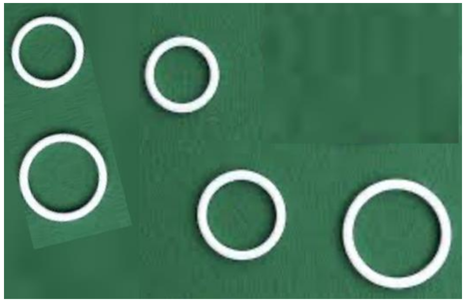


There are _ rings in img1.jpg

There are _ rings in img2.jpg



img1.jpg



img2.jpg

2. (20%) Camera Calibration

(出題：Jessica)

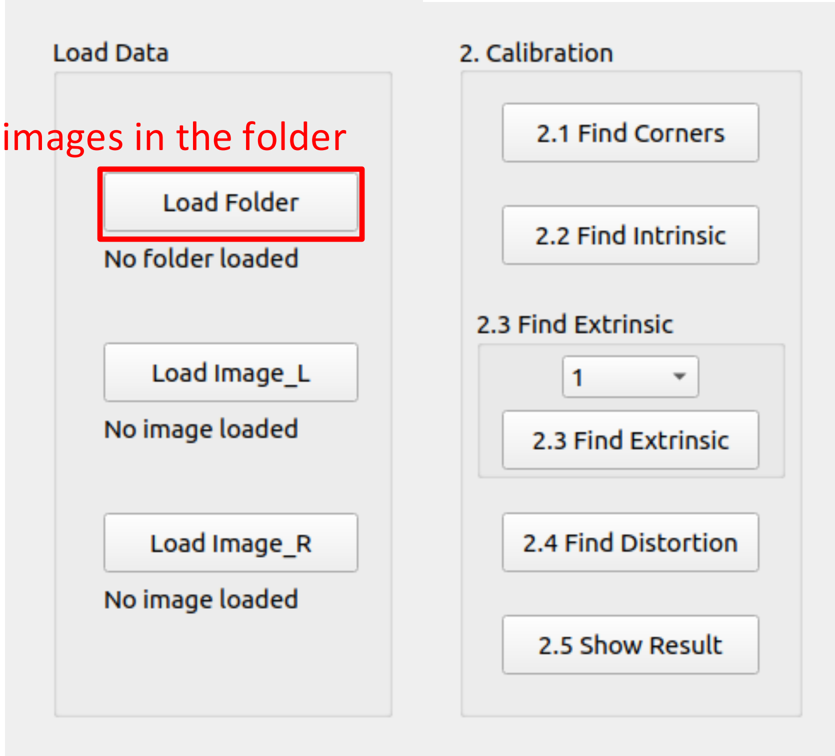
2.1 (4%) Corner detection

2.2 (4%) Find the intrinsic matrix

2.3 (4%) Find the extrinsic matrix

2.4 (4%) Find the distortion matrix

2.5 (4%) Show the undistorted result



Load all images in the folder

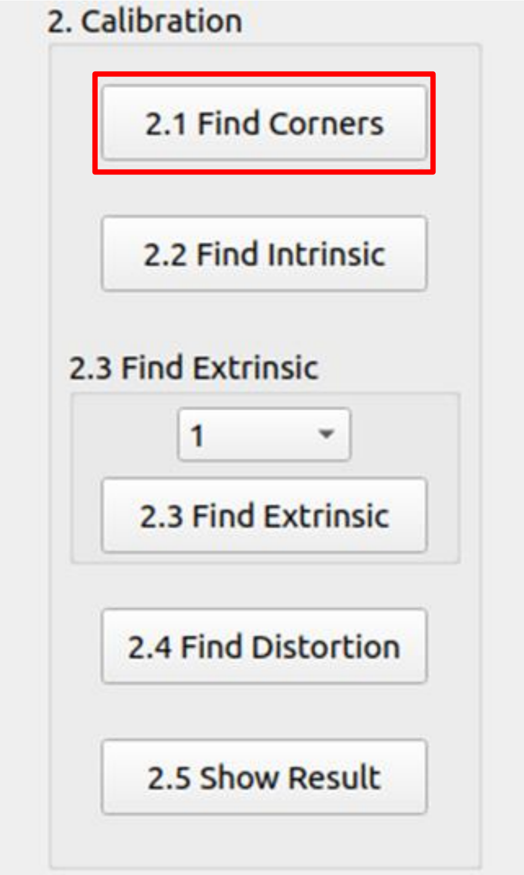
2.1 Corner Detection (4%)

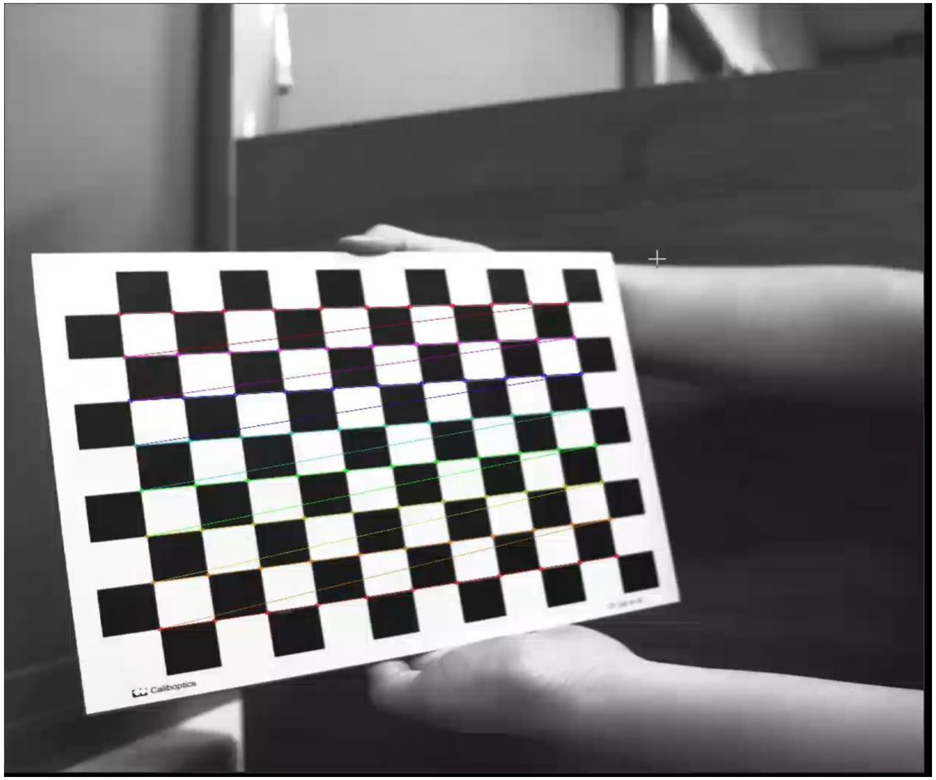
(出題：Jessica)

- ❑ Given: 15 images, 1.bmp ~ 15.bmp
- ❑ Q: 1) Find and draw the corners on the chessboard for each image.
  2) Click button “2.1” to show each picture 0.5 seconds.
- ❑ Hint :

  OpenCV Textbook Chapter 11 (p. 398 ~ p. 399)

  cv.findChessboardCorners(…)
- ❑ Ex:





2.2 Find the Intrinsic Matrix (4%)

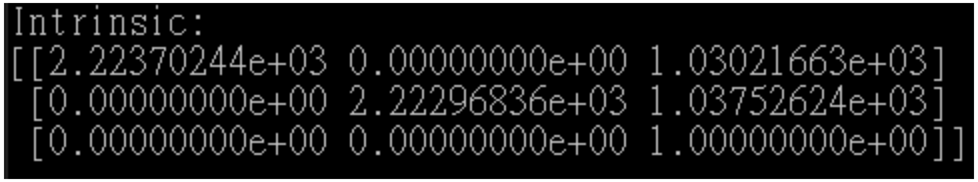
(出題：Jessica)

- ❑ Given: 15 images, 1.bmp ~ 15.bmp
- ❑ Q: 1) Find the intrinsic matrix ():

$$\begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

2) Click button “2.2” and then show the result on the console window.

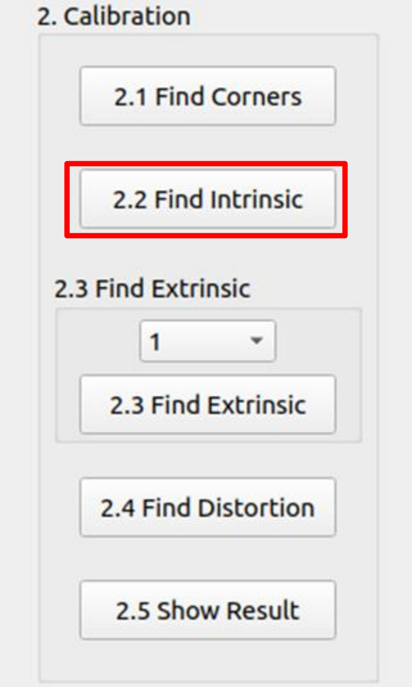
- ❑ Output format:



```
Intrinsic:
[[2.22370244e+03 0.00000000e+00 1.03021663e+03]
 [0.00000000e+00 2.22296836e+03 1.03752624e+03]
 [0.00000000e+00 0.00000000e+00 1.00000000e+00]]
```

(Just an example)

- ❑ Hint: OpenCV Textbook Chapter 11 (P.398 ~ p.400)



2.3 Find the Extrinsic Matrix (4%)

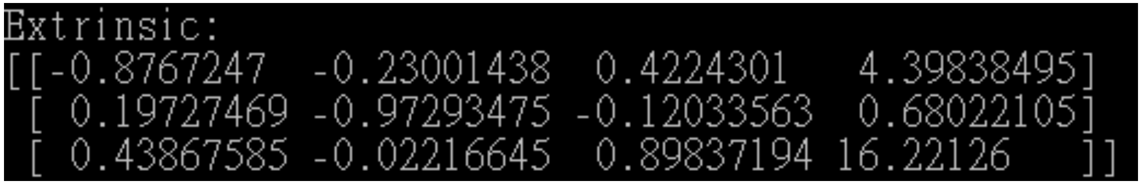
(出題：Jessica)

- ❑ Given: Intrinsic parameters, distortion coefficients, and the list of 15 images
- ❑ Q: 1) Find the extrinsic matrix of the chessboard for each of the 15 images, respectively:

$$\begin{bmatrix} R_{11} & R_{12} & R_{13} & T_1 \\ R_{21} & R_{22} & R_{23} & T_2 \\ R_{31} & R_{32} & R_{33} & T_3 \end{bmatrix}$$

2) Click button “2.3” and then show the result on the console window.

- ❑ Output format:



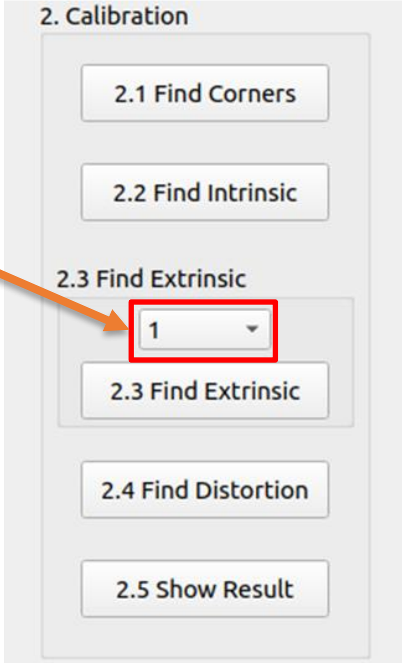
```
Extrinsic:
[[-0.8767247   -0.23001438   0.4224301    4.39838495]
 [ 0.19727469 -0.97293475 -0.12033563   0.68022105]
 [ 0.43867585 -0.02216645   0.89837194 16.22126    ]]
```

(Just an example)

- ❑ Hint: OpenCV Textbook Chapter 11, p.370~402

(1) List of numbers: 1~15

(2) Select 1, then 1.bmp will be applied, and so on



2.4 Find the Distortion Matrix (4%)

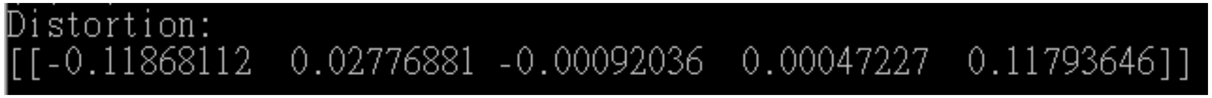
(出題：Jessica)

❑ Given: 15 images

❑ Q: 1) Find the distortion matrix: $[k_1, k_2, p_1, p_2, k_3]$

2) Click button “2.4” to show the result on the console window.

❑ Output format:

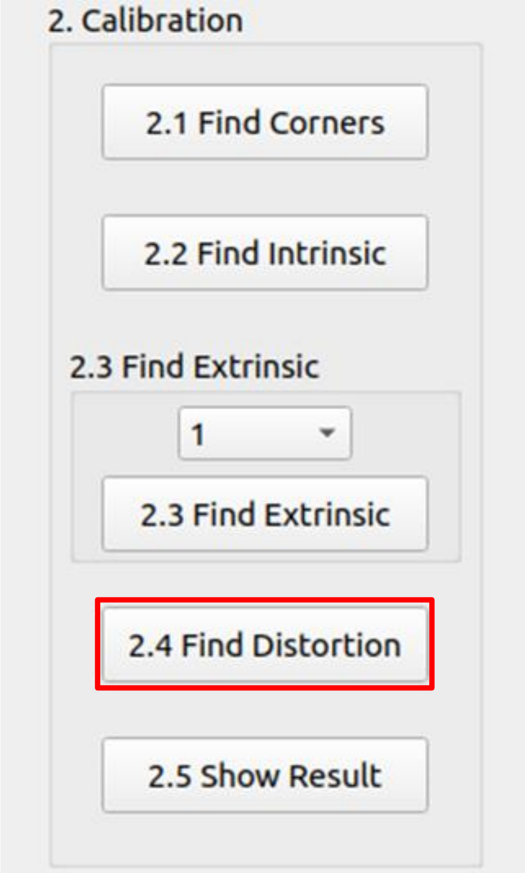


```
Distortion:
[[-0.11868112  0.02776881 -0.00092036  0.00047227  0.11793646]]
```

(Just an example)

❑ Hint:

- Distortion coefficients can be obtained simultaneously with intrinsic parameters
- OpenCV Textbook Chapter 11 (P.398 ~ p.400)



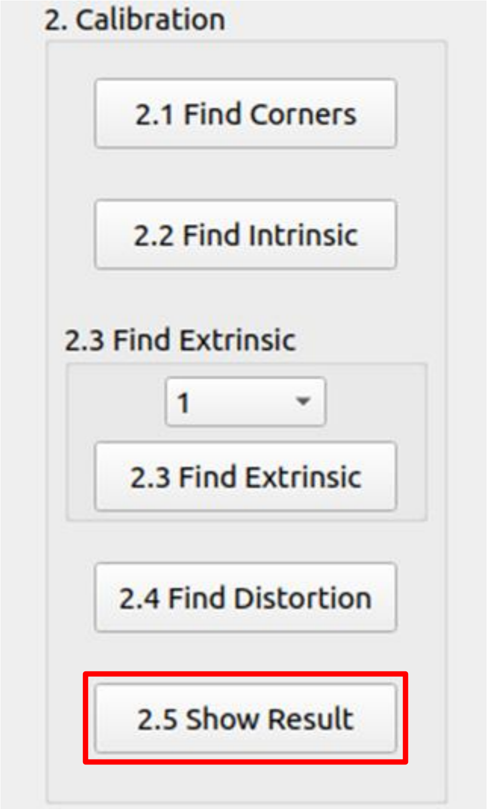
2.5 Show the undistorted result (4%)

(出題：Jessica)

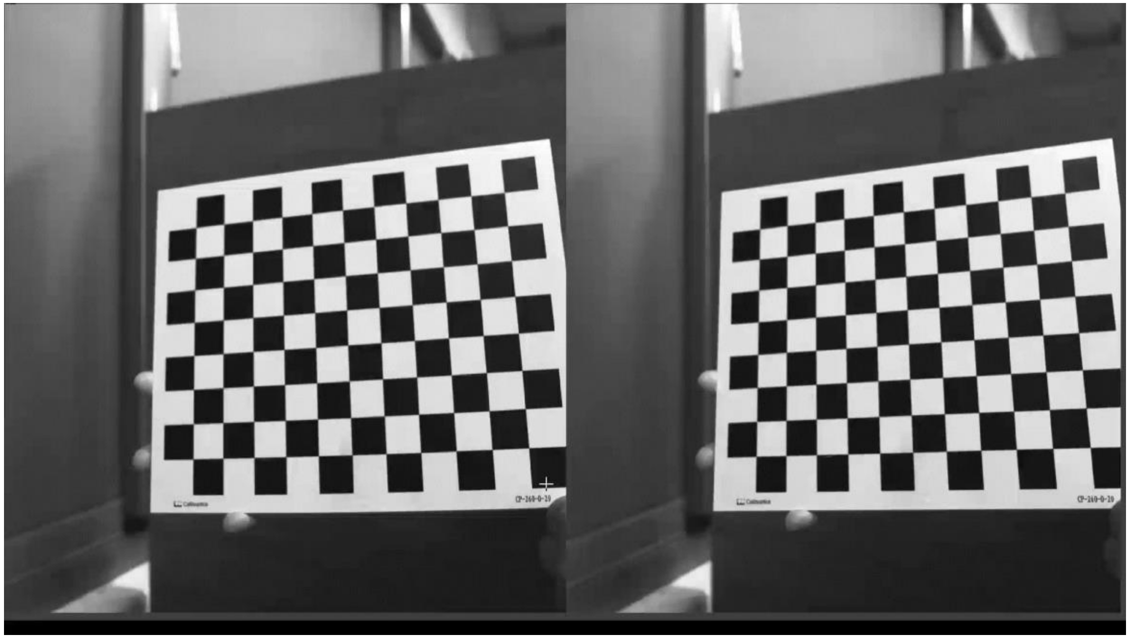
- ❑ Given: 15 images
- ❑ Q: 1) Undistort the chessboard images.

  2) Show each distorted and undistorted images 0.5 seconds.

- ❑ Hint:
  - ▪ cv::undistort(...) or cv::initUndistortRectifyMap(...)
  - ▪ OpenCV Textbook Chapter 11 (P.398 ~ p.400)



Distorted image Undistorted image

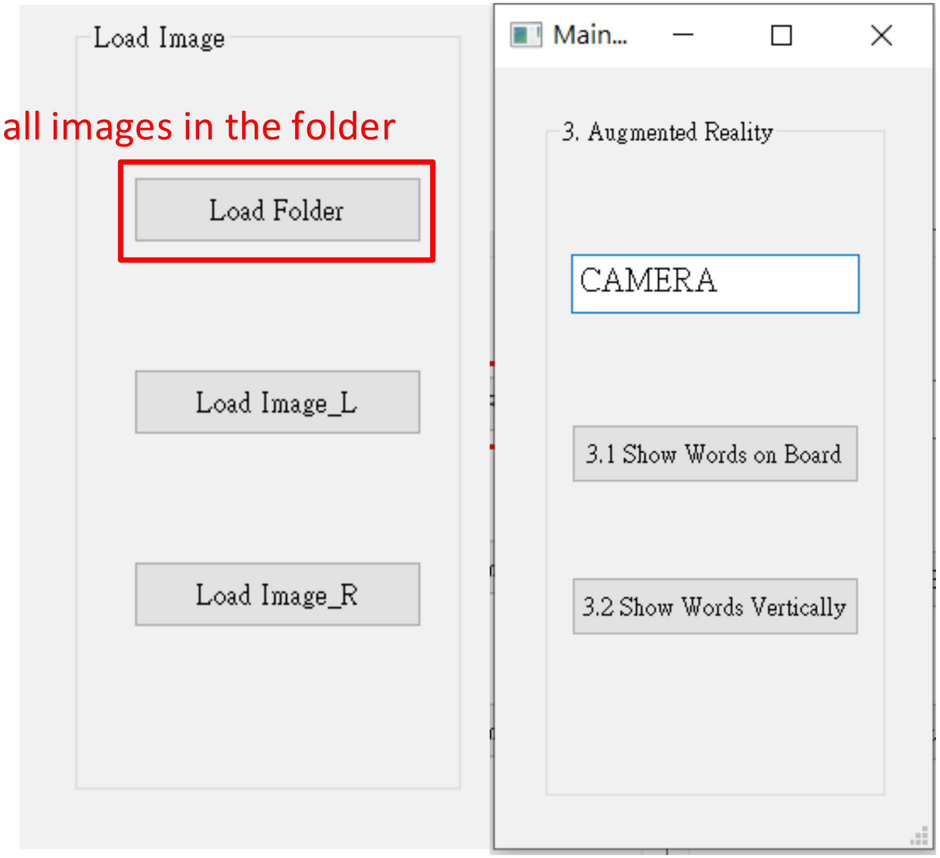


3. (20%) Augmented Reality

(出題：Ming)

3.1 (10%) Show words on board

3.2 (10%) Show words vertically


Load all images in the folder

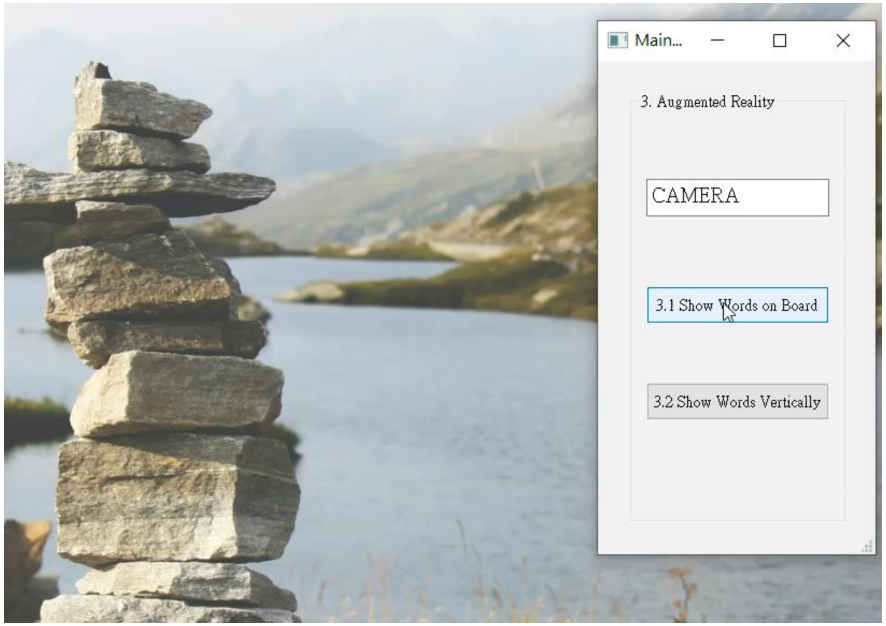
3. (20%) Augmented Reality

(出題：Ming)

- ❑ Given: 5 images: 1~5.bmp
- ❑ Q:

  1) Calibrate 5 images to get intrinsic, distortion and extrinsic parameters
  2) Input a “Word” less than 6 char in English in the textEdit box
  3) Derive the shape of the “Word” by using the provided library
  4) Show the “Word” on the chessboards images(1.bmp to 5.bmp)
  5) Show the “Word” vertically on the chessboards images(1.bmp to 5.bmp)
  6) Click the button to show the “Word” on the picture. Show each picture for 1 second (total 5 images)

Demo:



CAMERA

Hint : Textbook Chapter 11,
p.387~395 Calibration
p.405~412 Projection

cv2.calibrateCamera()
cv2.projectPoints()

3. (20%) Augmented Reality

(出題：Ming)

❑ Guides and Requirements:

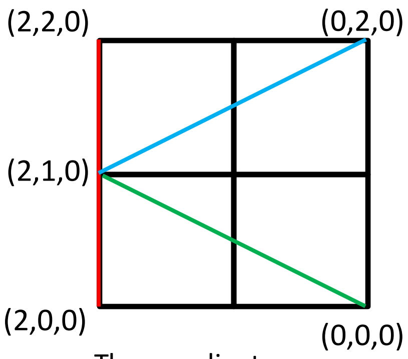
1) How to use the library: (alphabet_lib_onboard.txt, alphabet_lib_vertical.txt)

   – Use OpenCV function to read and derive the array or matrix of the char
   Here take ‘K’ in 'alphabet_lib_onboard.txt' for example

   Ex (Python):

```
fs = cv2.FileStorage('alphabet_lib_onboard.txt', cv2.FILE_STORAGE_READ)
ch = fs.getNode(‘K’).mat() ➔ get the lines of ‘K’

    ch = [[[2, 2, 0], [2, 0, 0]],
          [[0, 2, 0], [2, 1, 0]],
          [[2, 1, 0], [0, 0, 0]]]
```

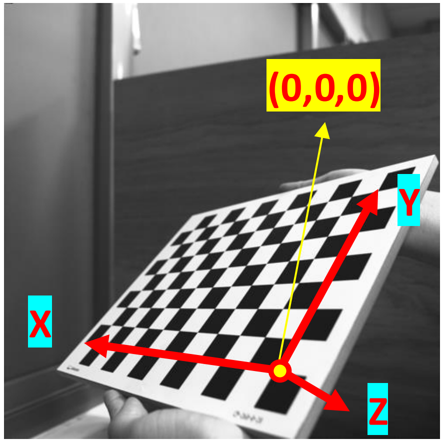


The coordinates are in Word Frame

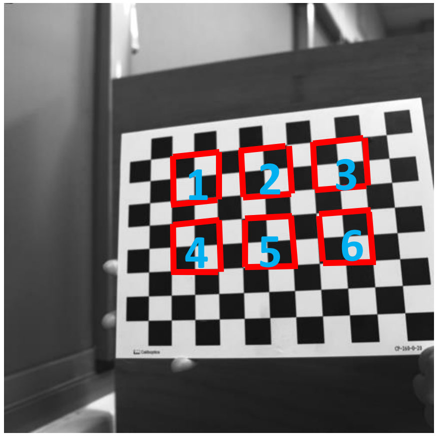
   – ‘K’ consist of 3 lines, so the ‘ch array’ consists 3 pairs of 3D coordinates in Word Frame representing two ends of the line shown in the upper right image.

2) Chessboard Coordinates

   – The chessboard x, y, z axis and (0,0,0) coordinate are shown in the bottom left image
   – Each Char should be place in the order and position shown in the bottom right image



Chessboard Frame



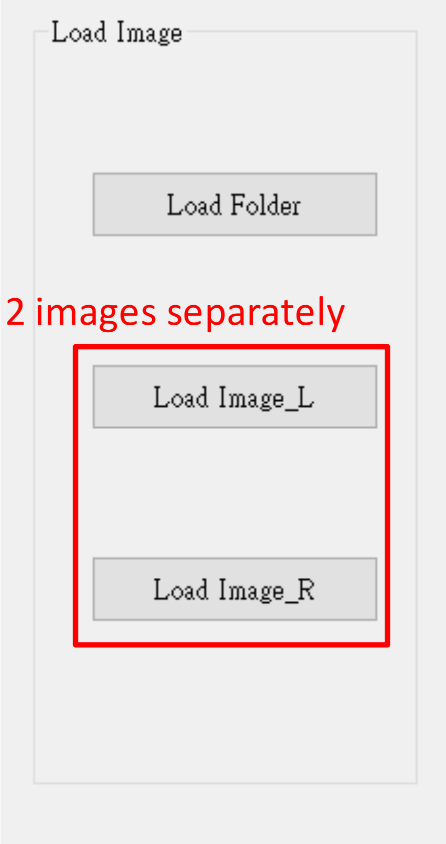
Position and Order

4. (20%) Stereo Disparity Map

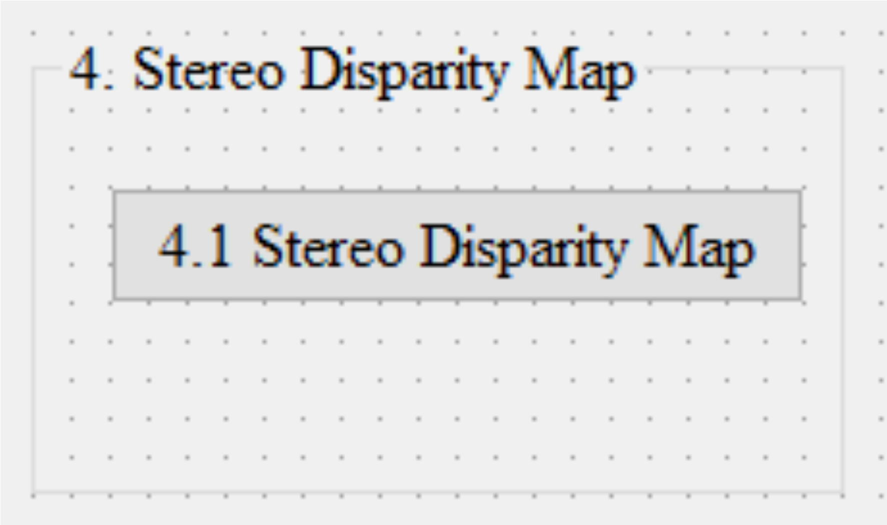
(出題：Maton)

4.1 (10%) Stereo Disparity Map

4.2 (10%) Checking the Disparity Value



Load 2 images separately



4.1 (10%) Stereo Disparity Map

(出題：Maton)

❑ Given: a pair of images, imL.png and imR.png (have been rectified)

❑ Q:

- Find the disparity map/image based on Left and Right stereo images

❑ Guides:

(1) Window Size: Must be odd and within the range [5, 255]

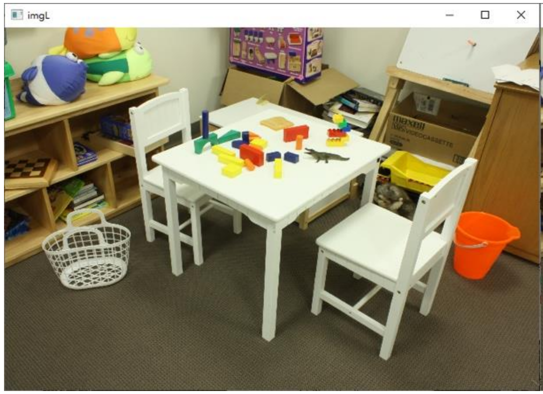
(2) Search range and direction:

- Disparity range:
  - Must be positive and divisible by 16.
  - Map disparity range to gray value range 0~255 for the purpose of visualization.
- If the left image is the reference image (the one used to cal. depth info for each pixel of that Img), then the search direction at right image will go from the right to left direction.

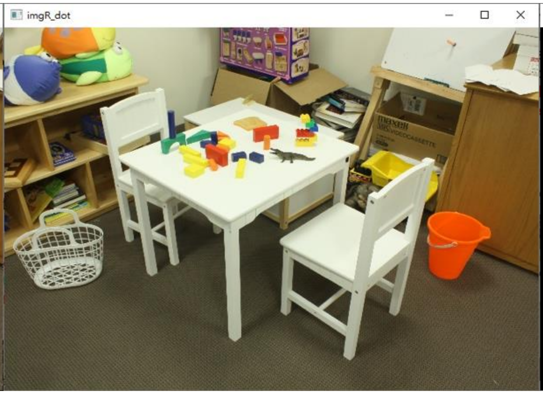
Camera information: 1) baseline=342.789mm,
2) focal length=4019.284 pixel,
3) $c_x^{right} - c_x^{left}$=279.184 pixel

OpenCV Textbook Chapter 11 (P.372-373) & OpenCV Textbook Chapter 12 (P.436)

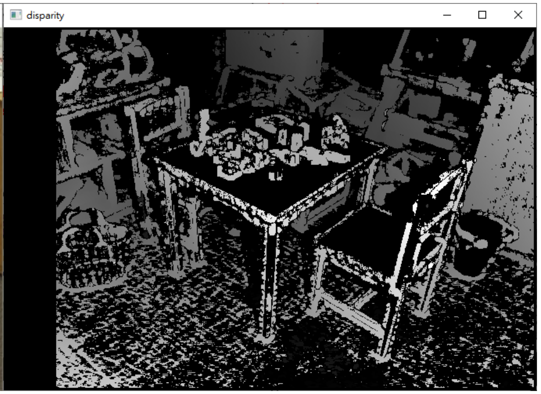
➢ Hint: OpenCV Textbook Chapter 12 (P.451)
StereoBM::create(256, 25)



imL.png
Left Image (Reference Image)



imR.png
Right Image



Result

4.2 (10%) Checking the Disparity Value

(出題：Maton)

❑ Given: a pair of images, imL.png and imR.png and disparity map from Q4.1.

❑ Q:

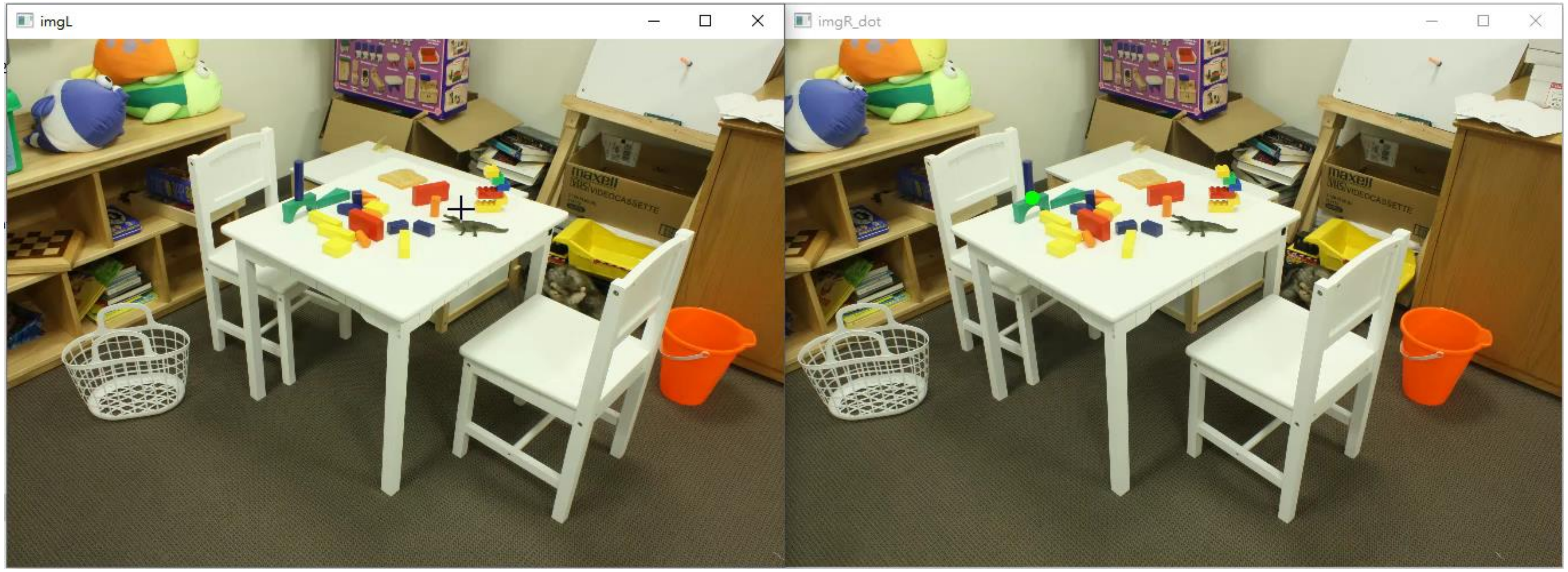
- Click at left image and draw the corresponding dot at right image.

❑ 1) Click at left image and draw the dot on the right image at accurate position.

❑ 2) User should allow to repeat 1).

➢Note: Click at gray position at disparity map result from Q.4.1, ignore the position with 0 disparity(e.g. Failure case).

➢Result Video:


imgL
imgR_dot