

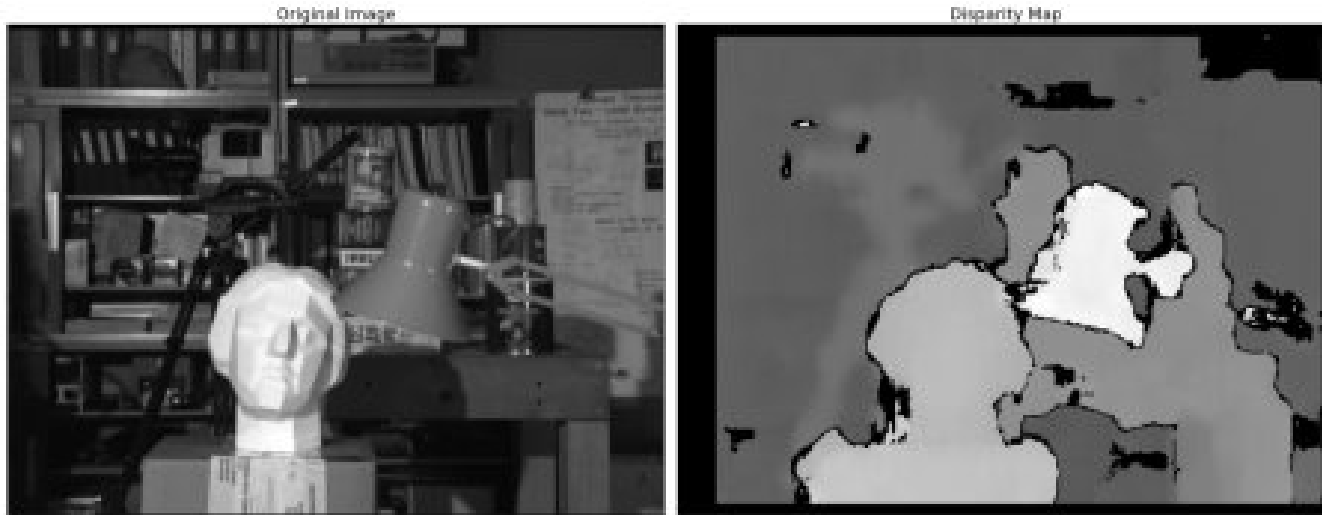
Reporting Assessment 1 – 50pts

Calibrate stereo cameras with chessboard images, and estimate a disparity map of rabbit images. Summarize your used functions and their explanations, and resulting images in your report. Also submit all the code for your implementation.

- (1) Show the undistorted rabbit images. (15pts)
- (2) Show the rectified rabbit images. (15pts)
- (3) Estimate disparity by local and global matching techniques. (20pts)

Stereo matching – simple example

Download and execute tsukuba.py & data/tsukuba_{l|r}.png



```
import numpy as np
import cv2 as cv
from matplotlib import pyplot as plt

imgL = cv.imread('tsukuba_l.png',0)
imgR = cv.imread('tsukuba_r.png',0)

stereo = cv.StereoBM_create(numDisparities=16, blockSize=15)
disparity = stereo.compute(imgL, imgR)
plt.imshow(disparity, 'gray')
plt.show()
```

Rectification

https://docs.opencv.org/3.4.2/d9/db7/tutorial_py_table_of_contents_calib3d.html

1. Estimate intrinsic matrices & distorted parameters + extrinsic parameters of two cameras:

https://docs.opencv.org/3.4.2/dc/dbb/tutorial_py_calibration.html

```
retval, cameraMatrix1, distCoeffs1, cameraMatrix2,  
distCoeffs2, R, T, E, F =  
cv2.stereoCalibrate(obj_points, img_points1,  
img_points2, cameraMatrix1, distCoeffs1,  
cameraMatrix2, distCoeffs2, imageSize)
```

R,T: Extrinsic parameters (1->2)

cameraMatrixN: Intrinsic parameters

distCoeffsN: distortion parameters



Rectification

https://docs.opencv.org/3.4.2/d9/db7/tutorial_py_table_of_contents_calib3d.html

2. Estimate Rectification matrix for two cameras:

```
R1, R2, P1, P2, Q, validPixROI1, validPixROI2 =  
cv2.stereoRectify(cameraMatrix1, distCoeffs1,  
cameraMatrix2, distCoeffs2, imageSize, R, T,  
flags, alpha, newimageSize)
```

https://docs.opencv.org/3.4.2/d9/d0c/group_calib3d.html#ga617b1685d4059c6040827800e72ad2b6

flags=0, alpha=1, newimageSize=(h,w) for example

3. Calculate rectification maps:

```
map1_l, map2_l =  
cv2.initUndistortRectifyMap(cameraMatrix1,  
distCoeffs1, R1, P1, newimageSize, m1type)
```

4. Show remapped images:

```
Re_TgtImg_l = cv2.remap(TgtImg_l, map1_l, map2_l,  
interpolation)
```

Stereo matching

https://docs.opencv.org/3.4.2/d9/db7/tutorial_py_table_of_contents_calib3d.html

```
stereo = cv2.StereoSGBM_create(  
    minDisparity = 32,  
    numDisparities = 112 - min_disp,  
    blockSize = 5,  
    P1 = 8*3*window_size**2,  
    P2 = 32*3*window_size**2,  
    disp12MaxDiff = 1,  
    uniquenessRatio = 10,  
    speckleWindowSize = 100,  
    speckleRange = 32,  
    mode = cv2.STEREO_SGBM_MODE_SGBM_3WAY  
)  
disp = stereo.compute(Re_TgtImg_l,  
Re_TgtImg_r).astype(numpy.float32) / 16.0
```

For instance:

StereoBM(block matching): local

StereoSGBM(semi-global matching): (semi-)global