Assignment 2

AI 6121 Computer Vision

Sun Guangbiao

sung0012@e.ntu.edu.sg

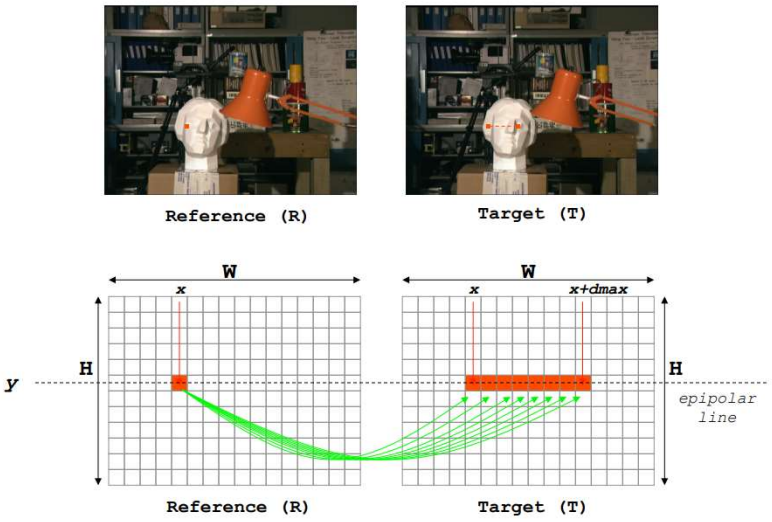
Matriculation No. G2003483F

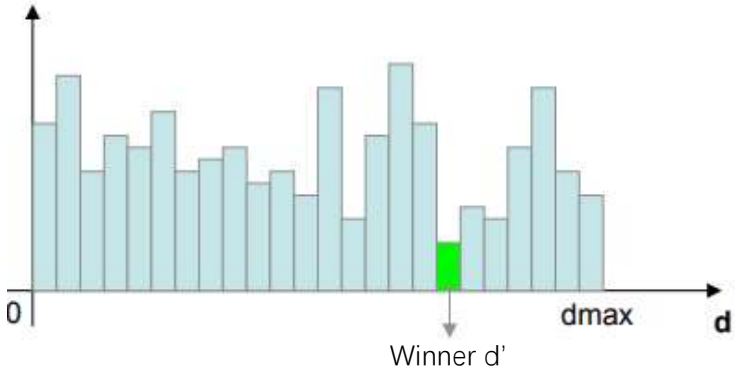
1 The procedure of computing disparity

- Matching cost : Squared differences between pixel intensities

  Squared differences: $e(x,y,d) = \left(I_R(x,y) - I_T(x+d,y)\right)^2$

- Disparity computation : The one（d'） with the lowest differences





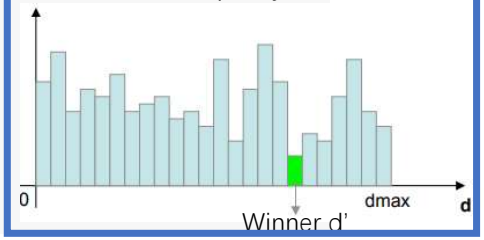
2 Algorithm and source code

```
import cv2
import numpy as np
import matplotlib.pyplot as plt

maxDisparity=15
window_size=7

limg = cv2.imread('corridorl.jpg',0)
rimg= cv2.imread('corridorr.jpg',0)
img_size=np.shape(limg)[0:2]

imgDiff=np.zeros((img_size[0],img_size[1],maxDisparity))
e = np.zeros(img_size)
for i in range(0,maxDisparity):
    e=np.square(rimg[:,0:(img_size[1]-i)]- limg[:,i:img_size[1]])
    e2=np.zeros(img_size)
    for x in range((window_size),(img_size[0]-window_size)):
        for y in range((window_size),(img_size[1]-window_size)):
            e2[x,y]=np.sum(e[(x-window_size):(x+window_size),(y-window_size):(y+window_size)])

        imgDiff[:,:,i]=e2

dispMap=np.zeros(img_size)

for x in range(0,img_size[0]):
    for y in range(0,img_size[1]):
        val=np.sort(imgDiff[x,y,:])
        val_id=np.argsort(imgDiff[x,y,:])
        dispMap[x,y]=val_id[0]/maxDisparity*255

plt.imshow(dispMap)
plt.axis('off')
plt.show()
```

Squared difference

$$e(x,y,d) = \left(I_R(x,y) - I_T(x+d,y)\right)^2$$

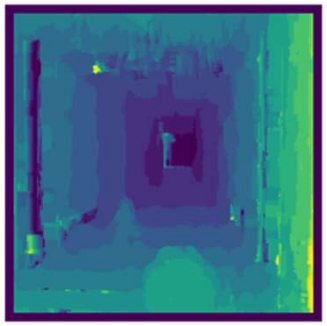
Get the corresponding point and the value of disparity



3 Results

left

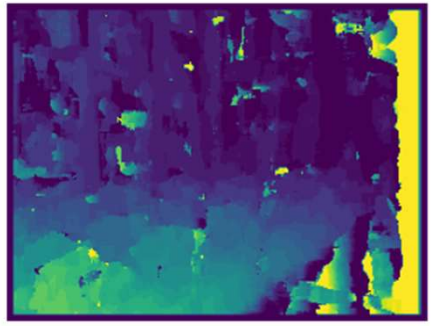
right



Disparity map

left

right

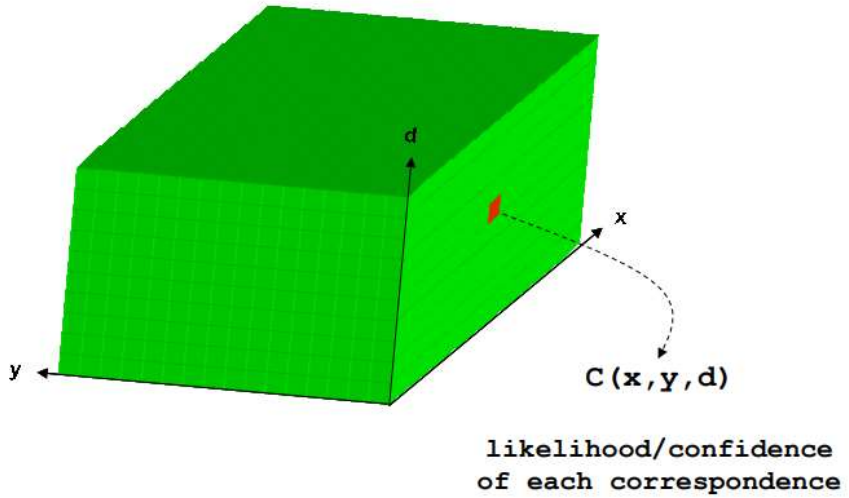


Disparity map

Problems : noises , corresponding points are not correctly detected and making disparity computing error

4 improvement

algorithms search for disparity assignments that minimize an energy function over the **whole** stereo pair using a pixel-based matching cost



$$E(D) = \sum_p \left(C\left(p, D_p\right) + \sum_{q \in N_p} P_1 I\left[\left|D_p - D_q\right| = 1\right] + \sum_{q \in N_p} P_2 I\left[\left|D_p - D_q\right| > 1\right] \right)$$ (SGBM)

4 improvement

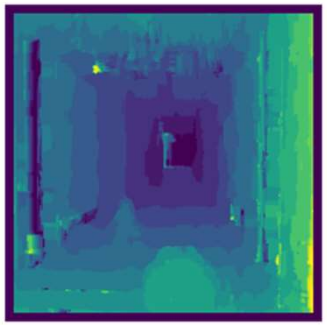
source code (SGBM Algorithm with OpenCV API):

```
imgL = cv2.imread('triclopsi2l.jpg',0)
imgR = cv2.imread('triclopsi2r.jpg',0)
num = 1
blockSize = 1
stereo_sgbm = cv2.StereoSGBM_create(
    minDisparity=0,
    numDisparities=16 * num,
    blockSize=blockSize,
    P1=8 * 3 * blockSize * blockSize,
    P2=32 * 3 * blockSize * blockSize,
    disp12MaxDiff=50,
    preFilterCap=15,
    uniquenessRatio=0,
    speckleWindowSize=50,
    speckleRange=2,
    mode=cv2.STEREO_SGBM_MODE_SGBM_3WAY
)


disparity_left = stereo_sgbm.compute(imgL, imgR)
disp_left = cv2.normalize(src=disparity_left, dst=disparity_left, beta=0, alpha=255,
                          norm_type=cv2.NORM_MINMAX, dtype=cv2.CV_8U)
plt.imshow(disp_left)
plt.axis('off')
plt.show()
```

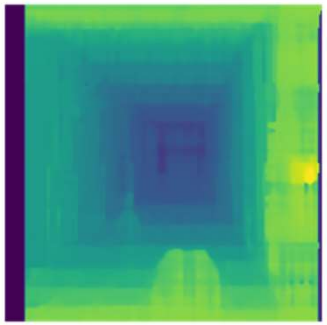
4 improvement

left

right



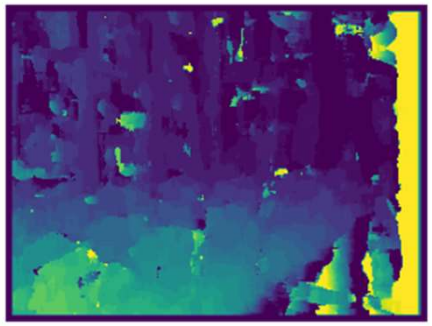
Disparity map (SSD)



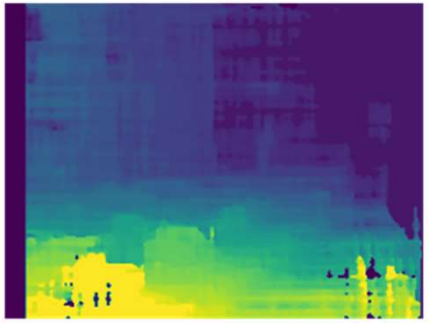
Disparity map(SGBM)

left

right



Disparity map (SSD)



Disparity map(SGBM)

The END

Thank you!