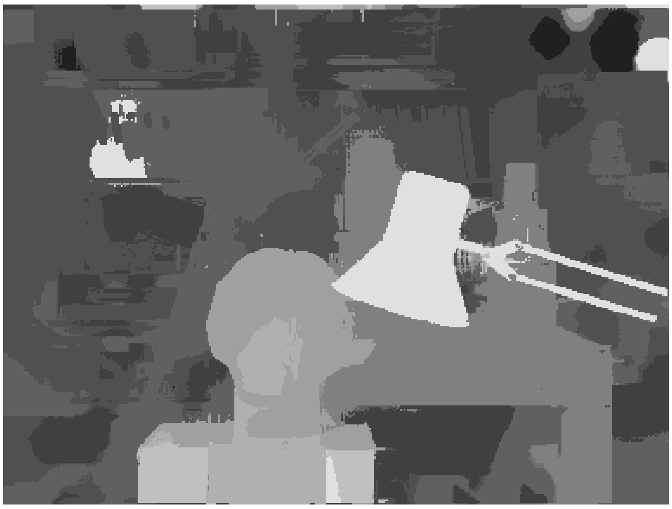
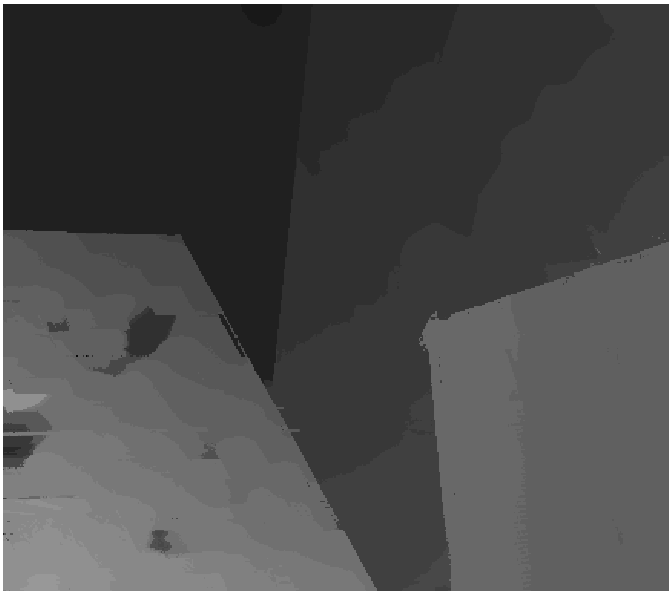
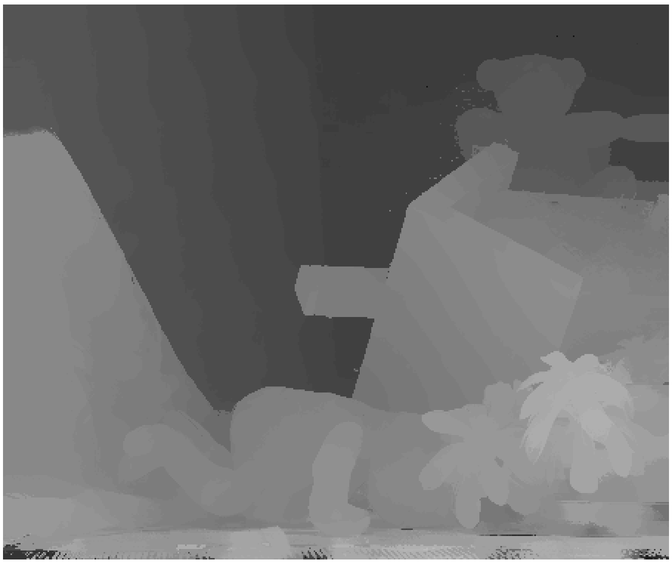
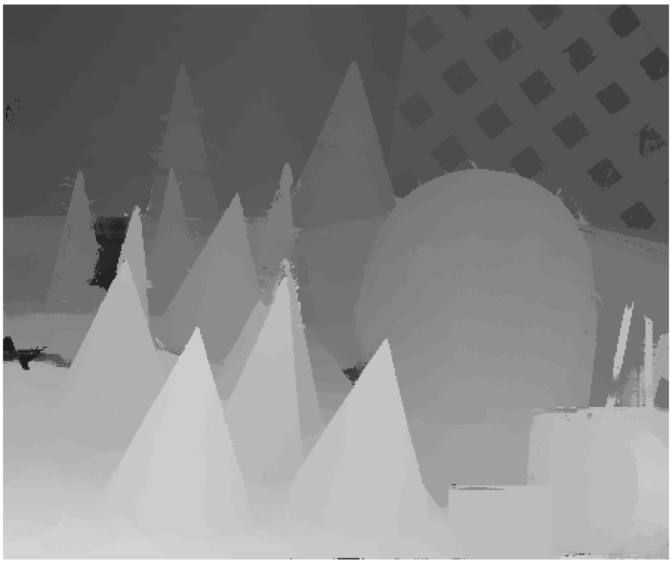
Computer Vision HW4 Report

Student ID: R11921100
Name: 溫威領

Visualize the disparity map of 4 testing images.

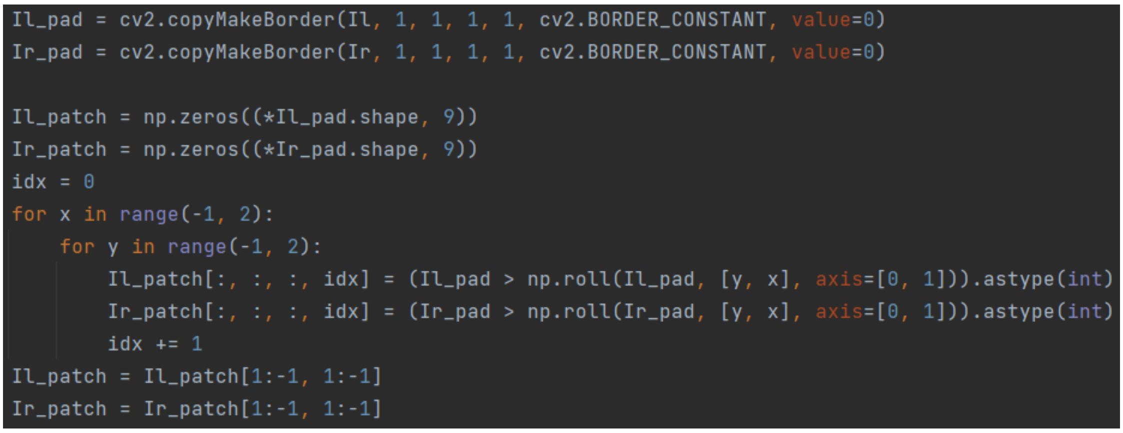
Tsukuba	Venus
	
Teddy	Cones
	

Report the bad pixel ratio of 2 testing images with given ground truth (Tsukuba/Teddy).

	bad pixel ratio
Tsukuba	4.49%
Teddy	9.28%

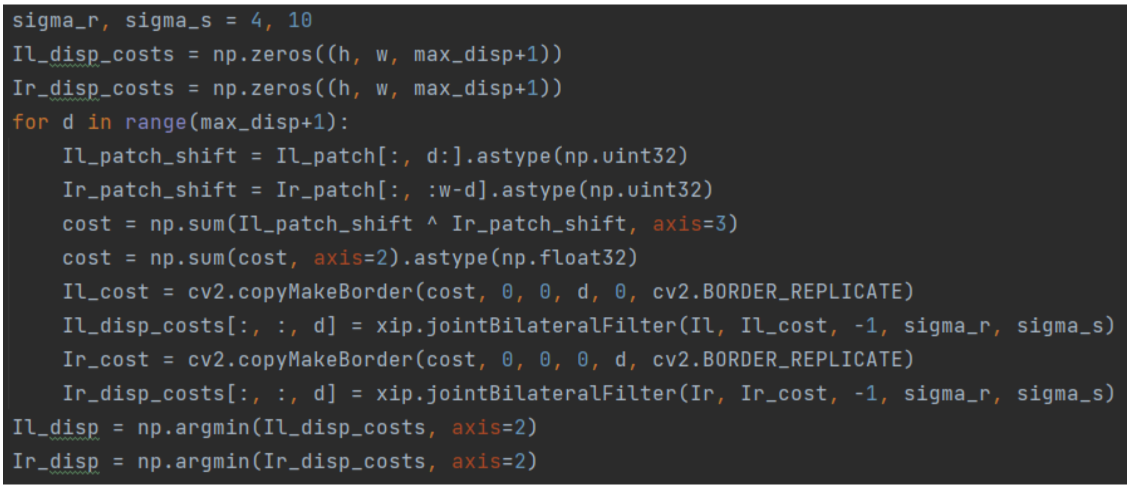
Describe your algorithm in terms of 4-step pipeline.

1. 對 ImgL 和 ImgR padding 後，取每個 pixels 的 3x3 patch。



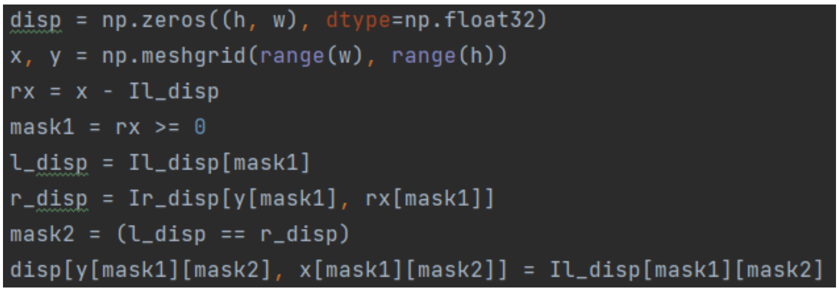
```
Il_pad = cv2.copyMakeBorder(Il, 1, 1, 1, 1, cv2.BORDER_CONSTANT, value=0)
Ir_pad = cv2.copyMakeBorder(Ir, 1, 1, 1, 1, cv2.BORDER_CONSTANT, value=0)

Il_patch = np.zeros((*Il_pad.shape, 9))
Ir_patch = np.zeros((*Ir_pad.shape, 9))
idx = 0
for x in range(-1, 2):
    for y in range(-1, 2):
        Il_patch[:, :, :, idx] = (Il_pad > np.roll(Il_pad, [y, x], axis=[0, 1])).astype(int)
        Ir_patch[:, :, :, idx] = (Ir_pad > np.roll(Ir_pad, [y, x], axis=[0, 1])).astype(int)
        idx += 1
Il_patch = Il_patch[1:-1, 1:-1]
Ir_patch = Ir_patch[1:-1, 1:-1]
```

2. 對 ImgL 和 ImgR 計算 x 軸上不同距離 Local binary pattern 的 Hamming Distance，並對每個距離 cost 進行 JBF filtering，然後 Winner-take-all 找每 pixel 上最小的 cost 是距離多遠作為 disparity。



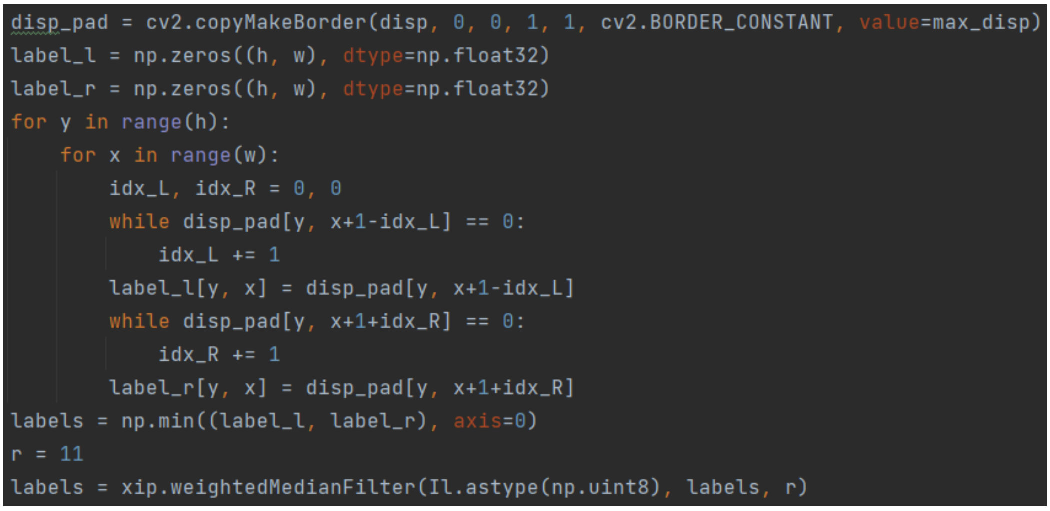
```
sigma_r, sigma_s = 4, 10
Il_disp_costs = np.zeros((h, w, max_disp+1))
Ir_disp_costs = np.zeros((h, w, max_disp+1))
for d in range(max_disp+1):
    Il_patch_shift = Il_patch[:, d:].astype(np.uint32)
    Ir_patch_shift = Ir_patch[:, :w-d].astype(np.uint32)
    cost = np.sum(Il_patch_shift ^ Ir_patch_shift, axis=3)
    cost = np.sum(cost, axis=2).astype(np.float32)
    Il_cost = cv2.copyMakeBorder(cost, 0, 0, d, 0, cv2.BORDER_REPLICATE)
    Il_disp_costs[:, :, d] = xip.jointBilateralFilter(Il, Il_cost, -1, sigma_r, sigma_s)
    Ir_cost = cv2.copyMakeBorder(cost, 0, 0, 0, d, cv2.BORDER_REPLICATE)
    Ir_disp_costs[:, :, d] = xip.jointBilateralFilter(Ir, Ir_cost, -1, sigma_r, sigma_s)
Il_disp = np.argmin(Il_disp_costs, axis=2)
Ir_disp = np.argmin(Ir_disp_costs, axis=2)
```

3. 利用 ImgL 的 disparity 檢查對應距離 ImgR 的 disparity 是否相同，並將相同的 disparity 留存，不相同的 disparity 則作為 hole。



```
disp = np.zeros((h, w), dtype=np.float32)
x, y = np.meshgrid(range(w), range(h))
rx = x - Il_disp
mask1 = rx >= 0
l_disp = Il_disp[mask1]
r_disp = Ir_disp[y[mask1], rx[mask1]]
mask2 = (l_disp == r_disp)
disp[y[mask1][mask2], x[mask1][mask2]] = Il_disp[mask1][mask2]
```

4. 對每個 pixel 找左右最近的有效 disparity 作為 label，並在最後使用 Weighted median filtering。



```
disp_pad = cv2.copyMakeBorder(disp, 0, 0, 1, 1, cv2.BORDER_CONSTANT, value=max_disp)
label_l = np.zeros((h, w), dtype=np.float32)
label_r = np.zeros((h, w), dtype=np.float32)
for y in range(h):
    for x in range(w):
        idx_L, idx_R = 0, 0
        while disp_pad[y, x+1-idx_L] == 0:
            idx_L += 1
        label_l[y, x] = disp_pad[y, x+1-idx_L]
        while disp_pad[y, x+1+idx_R] == 0:
            idx_R += 1
        label_r[y, x] = disp_pad[y, x+1+idx_R]
labels = np.min((label_l, label_r), axis=0)
r = 11
labels = xip.weightedMedianFilter(Il.astype(np.uint8), labels, r)
```