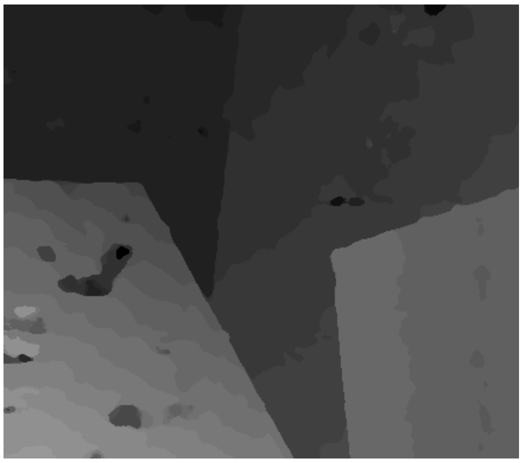
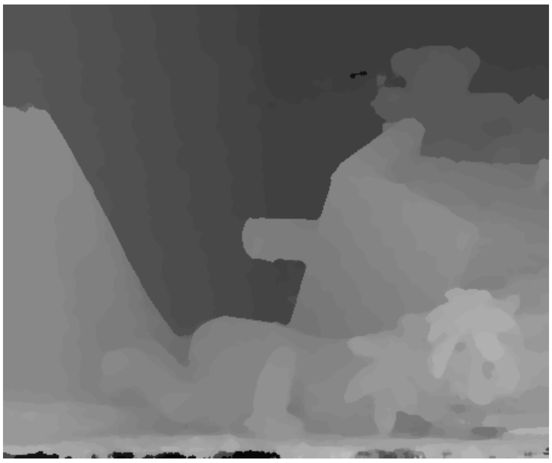
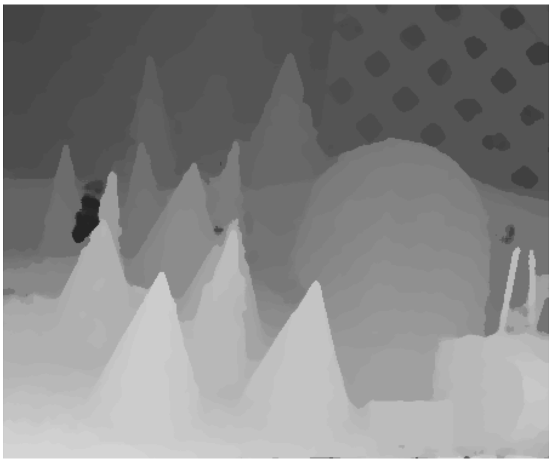
Computer Vision HW4 Report

Student ID: B08902092
Name: 陳翰雯

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	4.53%
Teddy	9.66%

Describe your algorithm in terms of 4-step pipeline.

Step 1: Cost Computation

Census cost is used for computation. The local binary pattern is first produced, followed by the Hamming distance. To prevent for loops, the code is implemented as follows:

Since, we are using a 5x5 patch, the images are first padded with size 2.

```
Il_pad = np.pad(Il, [(2, 2), (2, 2), (0, 0)], mode='edge')
Ir_pad = np.pad(Ir, [(2, 2), (2, 2), (0, 0)], mode='edge')
```

This is first to remember the index of the pixels, not including the center pixel. This is done mainly to prevent having double for loops and reduce it to only using 1 loop in total.

```
window_size = 5
center = window_size // 2
row = []
col = []
for i in range(window_size):
    for j in range(window_size):
        if i != center or j!= center:
            row.append(i)
            col.append(j)
```

The local binary patterns are then calculated, the value 1 signify pixels that has value smaller than the center pixel and 0 for pixels larger than the center pixel.

```
Il_binary =  np.zeros((h,w,ch,len(row)), dtype=np.bool)
Ir_binary =  np.zeros((h,w,ch,len(row)), dtype=np.bool)
for i in range(len(row)):
    Il_binary[:,:,:,i] = Il < Il_pad[row[i]:row[i]+h, col[i]:col[i]+w, :]
    Ir_binary[:,:,:,i] = Ir < Ir_pad[row[i]:row[i]+h, col[i]:col[i]+w, :]
```

The Hamming distance is then calculated, **`np.logical_xor`** is first used to find the ones that does not match. It will output True if both are different, otherwise, False. Next, **`np.sum`** is used to calculate the results, it will count the number of differences found, which is its Hamming Distance. This is done for both left and right images.

```
# Calculate cost for each disparity
Il_cost =  np.zeros((max_disp+1,h,w,ch), dtype=np.float32)
Ir_cost =  np.zeros((max_disp+1,h,w,ch), dtype=np.float32)
for d in range(max_disp+1):
    cost_l = np.logical_xor(Il_binary[:,d:,:,:], Ir_binary[:,:w-d,:,:])
    cost_l = np.sum(cost_l, axis = 3)
    cost_l = np.pad(cost_l, [(0, 0), (d, 0), (0, 0)], mode='edge')

    Il_cost[d] = cost_l

    cost_r = np.logical_xor(Il_binary[:,d:,:,:], Ir_binary[:,:w-d,:,:])
    cost_r = np.sum(cost_r, axis = 3)
    cost_r = np.pad(cost_r, [(0, 0), (0, d), (0, 0)], mode='edge')

    Ir_cost[d] = cost_r
```

Step 2: Cost Aggregation

The filter used is bilateral filter with the following parameters:

```
Il_cost[d] = xip.jointBilateralFilter(Il, Il_cost[d], 9, 0.9, 9)
Ir_cost[d] = xip.jointBilateralFilter(Ir, Ir_cost[d], 9, 0.9, 9)
```

This is done for each disparity with **`np.sum`** to sum it all.

Step 3: Disparity Optimization

Winner-take-all method is used and it is implemented with **`np.argmin`**.

Step 4: Disparity Refinement

Left-right consistency check is first done with the following equation: $D_l(x,y) = D_r(x - D_l(x,y), y)$ and then followed by Hole Filling.

This is then implemented together in the code as follows:

```
if Ir_map[i, j-Il_map[i,j]] == Il_map[i,j]:
    closest = Il_map[i,j]
    Il_enhance[i,j] = Il_map[i,j]
```

In the code above, the variable closest will take note of the closest valid disparity. This is also done for both left and right maps. Next, the final filled disparity map is then implemented using **`np.minimum`** as follows:

```
Il_disparity_map = np.minimum(Il_enhance, Ir_enhance).astype(np.uint8)
```

Lastly, the last step of refinement is weighted median filtering. This is done with the following code implementation:

```
labels = xip.weightedMedianFilter(Il_gray, Il_disparity_map, r=10, sigma=23)
```

To enhance the disparity map, blurring is added before and after the weighted median filtering. Furthermore, different padding mode and window size of census cost also affects the bad pixel ratio, therefore, “edge” padding mode is used and the window size is changed to 5 instead of 3.