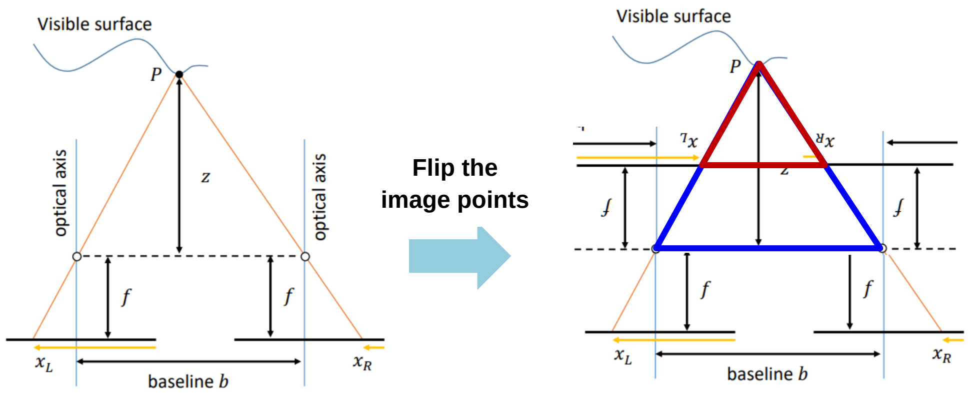
Stereo Matching HW4
Fall 2019

Name: 楊仲萱 **Department:** 電子所 ICS組 博一 **Student ID:** F07943023

Part 1: Depth from Disparity (1%)

Let $d = x_L - x_R$, according to the figure below, prove that $d = \frac{f \cdot b}{z}$. (hint: similar triangles)



After flipping the image points, we can now use the characteristic of the similar triangles to know the relationship between disparity d and depth z. Based on the above figure, the red triangle and blue triangle are similar triangles, therefore we can know that the ratios of bottom edge to height of two triangles are the same.

For the blue triangle, the ratio is b : z. For the red one, the ratio is $(b - x_L + x_R) : (z - f)$. Since they are the same, it can be written as $\frac{b}{z} = \frac{b - x_L + x_R}{z - f}$.

Therefore,

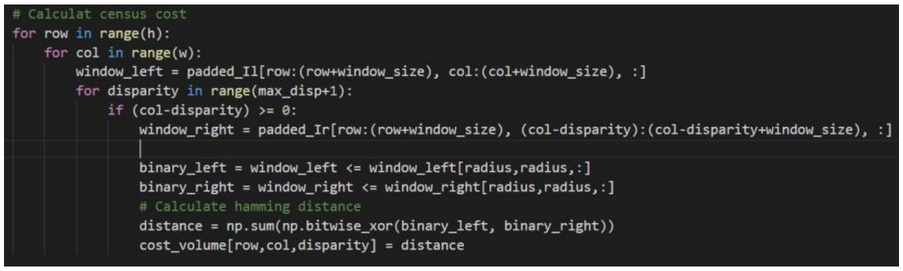
$$
\begin{aligned}
& b(z-f) = z(b - x_L + x_R) \\
\rightarrow\ & zb - fb = zb - zx_L + zx_R \\
\rightarrow\ & -fb = -zx_L + zx_R \\
\rightarrow\ & z(x_L - x_R) = fb \\
\rightarrow\ & (x_L - x_R) = \frac{f \cdot b}{z} \\
\therefore\ & d = \frac{f \cdot b}{z}
\end{aligned}
$$

Part 2: Disparity Estimation (4%)

1. **Explain the algorithm of the standard 4-step pipeline**

   <u>Step.1 cost computation</u>

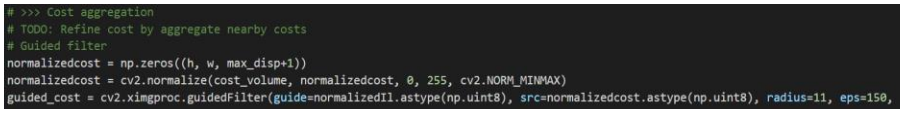
   In this step, I decided to use the census cost. First, I changed the window matrixes from left image and right image into the binary patterns. Second, the hamming distance is calculated by exclusive-oring the two binary patterns. Last, the cost volume was built from these hamming distance.



```
# Calculat census cost
for row in range(h):
    for col in range(w):
        window_left = padded_Il[row:(row+window_size), col:(col+window_size), :]
        for disparity in range(max_disp+1):
            if (col-disparity) >= 0:
                window_right = padded_Ir[row:(row+window_size), (col-disparity):(col-disparity+window_size), :]
                
                binary_left = window_left <= window_left[radius,radius,:]
                binary_right = window_right <= window_right[radius,radius,:]
                # Calculate hamming distance
                distance = np.sum(np.bitwise_xor(binary_left, binary_right))
                cost_volume[row,col,disparity] = distance
```

   <u>Step.2 cost aggregation</u>

   The guided filter is chosen to refine the cost volume. The code is shown below.



```
# >>> Cost aggregation
# TODO: Refine cost by aggregate nearby costs
# Guided filter
normalizedcost = np.zeros((h, w, max_disp+1))
normalizedcost = cv2.normalize(cost_volume, normalizedcost, 0, 255, cv2.NORM_MINMAX)
guided_cost = cv2.ximgproc.guidedFilter(guide=normalizedIl.astype(np.uint8), src=normalizedcost.astype(np.uint8), radius=11, eps=150,
```

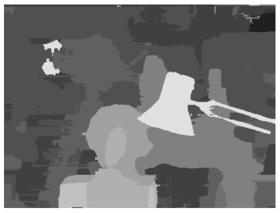
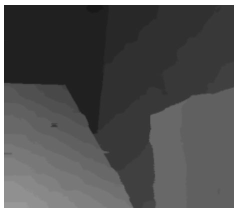
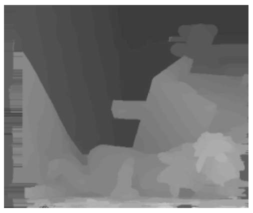
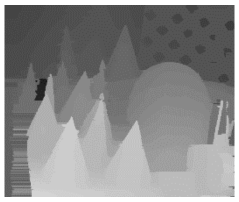
   <u>Step.3 disparity optimization</u>

   I simply used winner-take-all policy to construct the disparity map.

   <u>Step.4 disparity refinement</u>

   Left-right consistency check, hole filling and weighted median filtering which are introduced in the lecture are implemented in this step to get better disparity map.

2. **The output disparity maps**

Tsukuba	Venus
	
Teddy	**Cones**
	

3. The bad pixel ratio of each disparity maps

Tsukuba	Venus	Teddy	Cones
4.73%	1.38%	14.02%	10.34%

4. Reference papers or websites

[1] https://www.jianshu.com/p/9a9000d226b6

[2] https://blog.csdn.net/rs_lys/article/details/83754473

[3] https://medium.com/@omar.ps16/stereo-3d-reconstruction-with-opencv-using-an-iphone-camera-part-iii-95460d3eddf0

[4] https://www.mathworks.com/matlabcentral/fileexchange/29386-weighted-median-filter

[5] http://media.ee.ntu.edu.tw/courses/cv/19F/hw/hw4/cv2019_stereo.pdf

[6] http://media.ee.ntu.edu.tw/courses/cv/19F/hw/hw4/hw4.pdf