Computer Vision Fall 2018

National Taiwan University

Assignment 4:

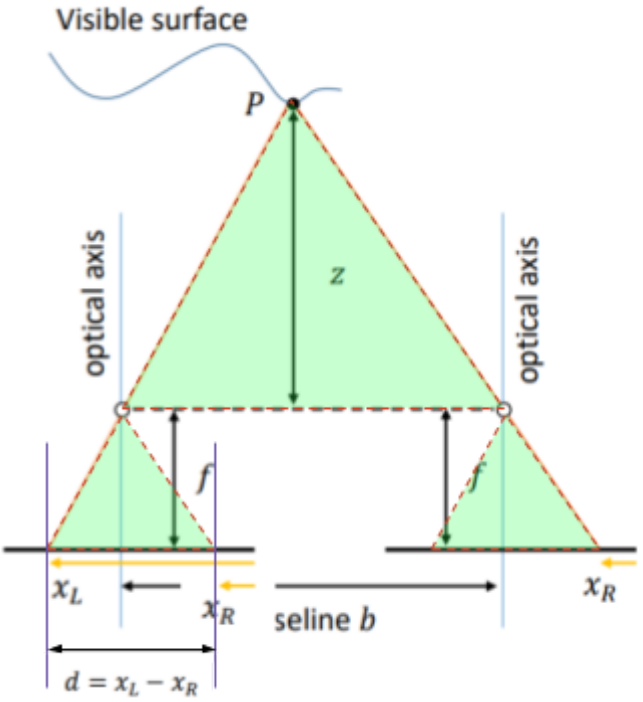
Stereo Matching

姓名：張邵瑀　　學號：r06921081

Part 1: Depth from Disparity

- Let $d = x_L - x_R$
- Prove $d = \frac{f \cdot b}{z}$

(hint: similar triangles)



- 首先，以上3個三角形為相似三角形，而$d = x_L - x_R$則為下面兩個三角形的底邊，兩個小三角形為完全相同的三角形。
- 大三角形的面積為$(b \cdot z) / 2$，小三角形的面積為$(d \cdot f) / 2$
- 大三角形與小三角的面積之間的關係為$((b \cdot (f/z) \cdot z \cdot (f/z)) / 2) = (d \cdot f) / 2$

  $\Rightarrow ((b \cdot z) \cdot (f/z)^2 / 2) = (d \cdot f) / 2 \Rightarrow (b \cdot z) \cdot (f/z)^2 = d \cdot f \Rightarrow b \cdot z \cdot f^2 / z^2 = d \cdot f$

  $\Rightarrow b \cdot f / z = d$ 得證。

Part 2: Disparity Estimation

STEP 1:Cost computation

先將右圖逐個pixel往右再把空的值補上最左邊行的值，因為我覺得與其補固定值不如補上周邊資訊的值因為以訊號的觀點來看，一個訊號有較高的機率與周圍的訊號有相同的分佈與頻率，再用左圖減掉右圖取絕對值，計算出左圖的mathcing cost，再用同樣的算法對左右圖的gradient作計算，算出左圖平行於x軸的gradient的mathcing cost，再由一定比例混合兩者得到深度相同張數的mathcing cost，我使用的比例為(17:83)參考[1]，再用同樣的方法對翻轉後的左右圖，與反轉後的左右圖gradient，算出對右圖計算出深度張數的mathcing cost.

STEP 2:Cost aggregation

接著藉由Guided filter[1]分別對左圖及右圖的compute costs做refine，其主要的原理是以Guide Image中的顏色對各個compute cost做weight filter，與我一開始使用 bilateral filter 時的效果相比，不但比較平滑，而且計算時間減低不少。

STEP 3:Disparity optimization

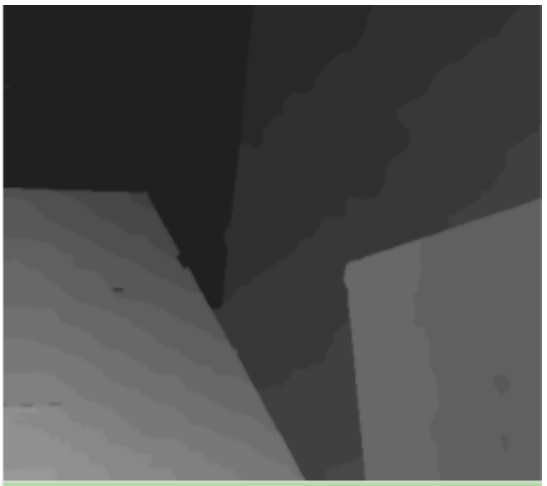
接著分別對左圖及右圖的Refined compute cost把shape為(深度, 高度, 寬度)的np.array用np.argmin做winner takes all最終得到初步的結果，但此時的結果會因為winner takes all對原始深度的資訊敏感度太高產生許多pepper and salt雜訊所以先用一個小的median filter去除，接著為了解決很多空洞(無限遠的深度)會對最終分數產生不小影響的問題，我做一個threshold填值的方式事先做一次refined，首先對此兩張圖做Weighted Median Filtering[2]，得到等會要填值的資訊，接著再把threshold以下值的位置填上做完Weighted Median Filtering後待用的refine reference map的相對位置(左右圖都做相同處理)。

STEP 4:Disparity refinement

接著refine的做Left-right consistency check，若為不相同的值則先標記為-1，產生一個註記過的Disparity map，再對此一map做hole filling，我使用的hole filling最簡單的由左往右掃的方式，而對左圖做，通常是左邊的資訊會消失，所以我把-1碰到最的最右邊的非-1值填過去，也是套用訊號會最相似附近結構值的想法，最後再用Weighted Median Filtering[2]對最後filling好的深度圖做一次以顏色為Guided的weighted filter，再用median filter將一些細微的pepper and salt濾掉雜訊就是我的結果。

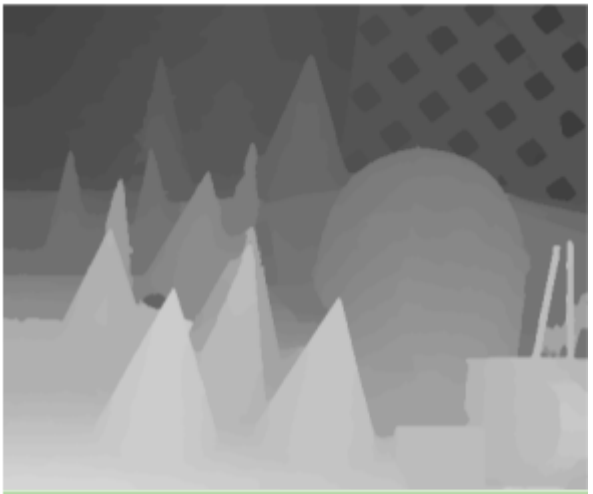
disparity maps

Tsukuba: 1.85%



Venus: 0.26%

Teddy: 8.15%



Cones: 6.75%

bad pixel ratio	
Tsukuba: 1.85%	Venus: 0.26%
Teddy: 8.15%	Cones: 6.75%
Average: 4.25%	

Reference

[1] Christoph Rhemann, ***Matlab demo code accompanying the CVPR11 paper***, https://github.com/coder89/stereo/tree/master/Matlab/testy

[1] Kaiming He, ***Constant Time Weighted Median Filtering for Stereo Matching and Beyond***, 2013, http://kaiminghe.com/