**Computer Vision HW4 Report**

Student ID: B09901075

Name: 陳駿瑋

**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 | 4.40% |
| Teddy | 11.86% |

**Describe your algorithm in terms of 4-step pipeline.**

**Step 1: Compute Matching Cost**

一開始先計算matching cost，對於每個pixel，要計算max\_disp＋1個costs。這裡我照著投影片使用Census Cost。為此我先對原圖做padding，讓邊界的pixels在與neighbors比較時不會出錯。而經過測試，padding的方法使用cv2.BORDER\_REPLICATE（複製邊界pixel值）產生出的bad pixel ratio值較低。Padding後，計算出兩個8\*image\_size的binary矩陣，代表每個pixel與8個neighbors比大小後的結果（大於=1）。最後對max\_disp做iteration，每圈對應的disp值決定右圖橫移距離，並在右圖橫移後，每個pixel與左圖對應，用左右圖各8個值計算出hamming distance（同時會把3個channel的hamming distance加起來），如此一個iteration會產生兩個h \* w的cost矩陣（左到右和右到左）。最後能產生兩個shape為(max\_disp + 1, h, w)的cost矩陣。

**Step 2: Cost Aggregation**

這一步中，雖然是step 2，但在我的程式中，融合進上一步的for迴圈。在每個迴圈，可以計算出兩個h \* w的cost矩陣，為了減少雜訊，可以再對這個cost做filter，我選擇的是[Tips]建議的jointBilateralFilter。值得注意的是，jointBilateralFilter的參數是影響結果bad pixel ratio一個很重要的因素。這裡的參數（diameter, sigmaColor, sigmaSpace = 22, 10, 26）是先隨便設定，最後慢慢依照bad pixel ratio的結果調整得出的結果。

**Step 3: Disparity Optimization**

這步很單純的把前面每個pixel算出的max\_disp+1個costs，取最小值的index，並記錄下來，產生disparity map。

**Step 4: Disparity Refinement**

最後是加強第三步產生的disparity map。首先是檢查DL(x, y) = DR(x – DL(x, y), y)，看左到右和右到左的結果是否相同，若是不同，就在左到右圖的map挖一個hole（將index值設為-1）。之後hole filling的部分，利用for迴圈掃過左到右圖的map，如果有hole，就從它左右延伸（while loop），找到pixel距離最近的兩個點（一左一右），並選擇值較小的填入。而如果往左或往右時，沒有找到非hole的pixel就到邊界了，那值就設為max\_disp。最後在填完holes之後，對其做weightedMedianFilter去改善結果。