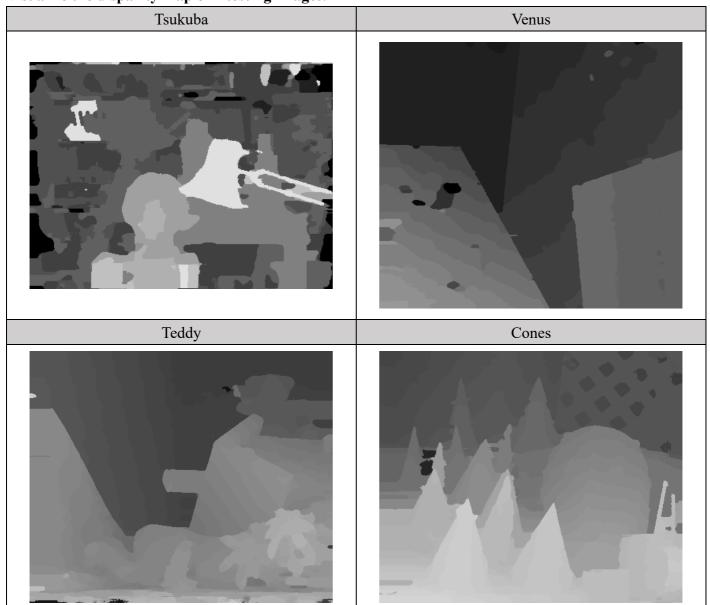
Computer Vision HW4 Report

Student ID: R11921038

Name: 江讀晉

Visualize the disparity map of 4 testing images.



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

	bad pixel ratio
Tsukuba	7.60%
Teddy	9.86%

Describe your algorithm in terms of 4-step pipeline.

1. Cost computation

一般來說,disparity estimation 的流程為給定左右視角的兩張影像,將每個座標點依照不同 disparity 計算出兩張影像指定視窗大小內的 matching cost。如果以迴圈形式遍歷所有像素座標 並計算所有 disparity 對應的 cost,所需要的時間為 $O(h \cdot w \cdot d_{max})$,其中h和w為影像尺寸, d_{max} 為 disparity 數值上限。為了達到較少的計算時間,因此參考作業三,使用 numpy 處理 indexing 讓每個座標像素同 disparity 的 cost 可以平行計算,如此一來計算時間則大約需要 $O(d_{max})$ 即可。

此外,由於 matching cost 採用 census cost,當給定座標點,所使用的資訊不只有該像素點的 intensity,還需要包括鄰近 3x3 區域的資訊,因此,先將尺寸為(h,w,ch)、資訊為 intensity 的原始影像,轉換成尺寸為(h,w,8,ch)、資訊為 local binary pattern 的影像,如此只要給定座標方能得到計算 Hamming distance 所需的 binary pattern。Step 1 的實作流程如 Figure 1 所示。

- 1. padded_Iℓ, padded_Ir ← 將Iℓ, Ir四周各 pad 1 像素
- neighbor_Iℓ, neighbor_Ir ← 透過比較Iℓ, Ir和 8 個不同 indexing 範圍的
 padded_Iℓ, padded_Ir , 得到 local binary pattern 資訊
- 3. Repeat d from 0 to d_{max} :

3.1.
$$cost \binom{(0,d,d)}{\binom{(0,d+1,d)}{\vdots}} \leftarrow 沿著最後兩個軸 ,$$
 計算 $neighbor_{l}\ell \binom{(0,d)}{\binom{(0,d+1)}{\vdots}}$ 和 $neighbor_{l}r \binom{(0,0)}{\binom{(0,1)}{\vdots}}$ 之間的 Hamming distance

3.2. 將cost中第二軸超出d之外的座標以最相鄰座標的數值補齊

Figure 1. Cost computation 流程

2. Cost aggregation

採用 joint bilateral filter: Step 1 計算出的 cost 為一個(h, w, d_{max})的矩陣, 將每一個 disparity 對應的 cost map 作 joint bilateral filtering, 得到新的 cost 矩陣。

joint bilateral filter 的參數: joint 使用經過灰階處理的 left image、diameter of neighborhood 不指定、color sigma 為 4、coordinate space sigma 為 11。

3. Disparity optimization

採用贏者全拿:每個像素座標點的 disparity,為 Step 2 計算出的 cost 矩陣中, cost 值最小的 index (disparity) 值。

4. Disparity refinement

採用 left-right consistency check、hole filling 和 weighted median filter。Weighted median filter的 joint 使用經過灰階處理的 left image、參數 r 設為 3。Step 4 的實作流程如 Figure 1 所示。

- 1. mask ← 將 mesh 座標帶入 $labels_{\ell}(y,x) = labels_{\ell}(y,x) labels_{\ell}(y,x)$)得到有效 disparity
- 2. valid_labels_ℓ, refine_ℓℓ, refine_ℓr ← 透過mask生成,其中valid_labels_ℓ寬度左右各 pad 1
- 3. Repeat (i, j) from (0,0) to (h-1, w-1):

3.1.
$$refine_{\ell\ell(i,j)} \leftarrow$$
 找出 $valid_{labels_{\ell}} \begin{pmatrix} (i,0) \\ \vdots \\ (i,j+1) \end{pmatrix}$ 中最接近的有效 disparity 3.2. $refine_{\ell} r(i,j) \leftarrow$ 找出 $valid_{labels_{\ell}} \ell \begin{pmatrix} (i,j+2) \\ \vdots \\ (i,h) \end{pmatrix}$ 中最接近的有效 disparity

- 4. $labels \leftarrow min(refine_{\ell}, refine_{\ell})$
- 5. $labels \leftarrow weighted_median_filter(I\ell_gray, labels)$

Figure 2. Disparity refinement 流程