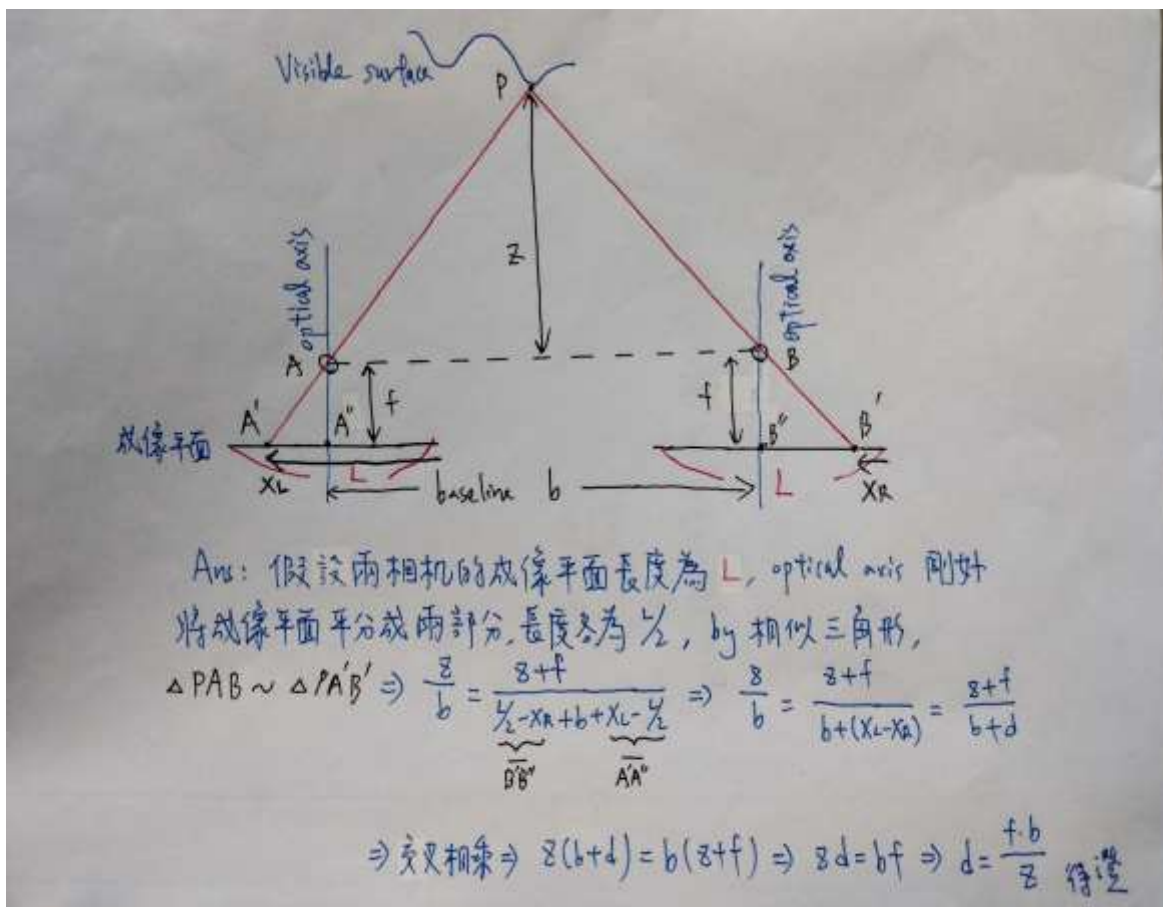


Computer Vision HW4 Report

B05901182 電機三 潘彥銘

Part 1.



Part 2.

1. Algorithm(作法參考[1]、[2] 兩篇 paper)

註: 由於我的演算法需要用到灰階圖，因此我的 computeDisp 額外輸入了兩張左右圖兩張灰階。

(a.) Cost assumption

引進 BRIEF descriptor 的概念，如下圖(一)，在 $S \times S$ 的 window 中，sample n 組 $\langle p_i, q_i \rangle$ ，其中 $\langle p_i, q_i \rangle$ 是由一平均值為 0.0，標準差為 4.0 的高斯分布 sample 出來的 pixel，paper 裡設 $S = 26$ (我設 27)， $n = 4096$ ，因此 $\langle p_i, q_i \rangle$ 共有 4096 組。

$\tau(p_i, q_i)$ 表示法如下圖(二)，比較 p_i 與 q_i 的 intensity，結果輸出 1 或 0。

因此下圖(一)的 $B(x)$ 可視為一串 binary，這裡我用 numpy array 儲存。

$$B(x) = \sum_{1 \leq i \leq n} 2^{i-1} \tau(p_i, q_i) \quad \tau(p_i, q_i) = \begin{cases} 1 & : I(p_i) > I(q_i) \\ 0 & : I(p_i) \leq I(q_i) \end{cases}$$

圖(一)

圖(二)

有了 $B(x)$ 的概念後，就可以計算 cost volume，如圖(三)，而 x_d 是在右圖中，使得 $B(x_d)$ 與 $B(x)$ 的 hamming distance 最小的 disparity。

$$C(x, d) = \| B(x) \text{ XOR } B(x_d) \|_1$$

圖(三)

(b.) Cost aggregation

由於只採用(a)部分的做法的話，會發生 **edge flattening** 的問題，這部分採用被作者稱作 **binary mask** 的方法，如下圖(四)，其中 x 是位於 window center 的 pixel， $\delta(x, p_i, q_i)$ 表示法如下圖(五)。

$$\Phi(x) = \sum_{1 \leq i \leq n} 2^{i-1} \delta(x, p_i, q_i) \quad \delta(x, p_i, q_i) = \begin{cases} 1 & : w(x, p_i, q_i) \leq T \\ 0 & : w(x, p_i, q_i) > T \end{cases}$$

圖(四)

圖(五)

$w(x, p_i, q_i)$ 是 x 與 p_i 的 SAD、 x 與 q_i 的 SAD 兩者取大者，如下圖(六)。

$$w(x, p_i, q_i) = \max(SAD(x, p_i), SAD(x, q_i))$$

圖(六)

SAD 則是先將 RGB 轉成 LAB 的色彩空間，再比較兩者的 intensity，如下圖(七)。

$$SAD(x, y) = \sum_{c \in [L, A, B]} |I_c(x) - I_c(y)|$$

圖(七)

如此新的 cost volume 表示法如下圖(八)。

$$C(x, d) = \|B(x) \mathbf{XOR} B(x_d) \mathbf{AND} \Phi(x)\|_1$$

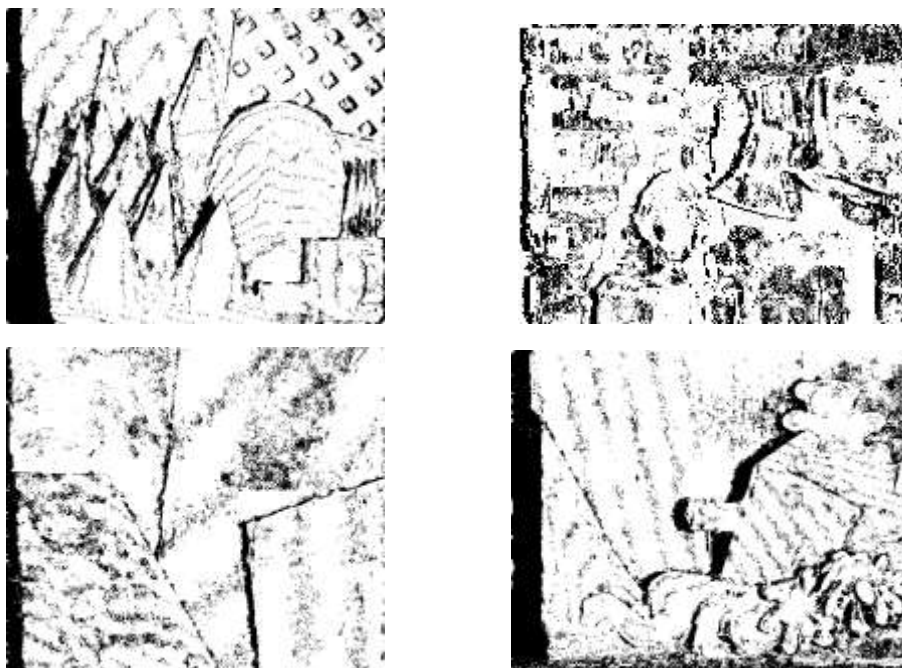
圖(八)

(c.) Disparity optimization

採用 Winner-take-all 的作法。

(d.) Disparity refinement

先做 left-right consistency check，得到 occlusion map，如下圖(九-十二)，透過 occlusion map 我可以知道哪些 pixels 是被擋住或 mismatch，這些點稱作 invalid，反之，valid。



圖(九-十二)

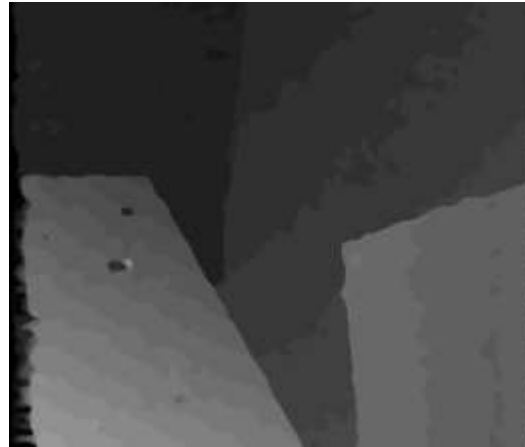
找出這些 invalid 點，分別從其左右邊分別找一個 valid 點，比較這兩個 valid 點的 disparity，取小的當作 invalid 點新的 disparity。

最後輸入 opencv 內建的 medianBlur，得到最後的 disparity map。

2. Disparity map



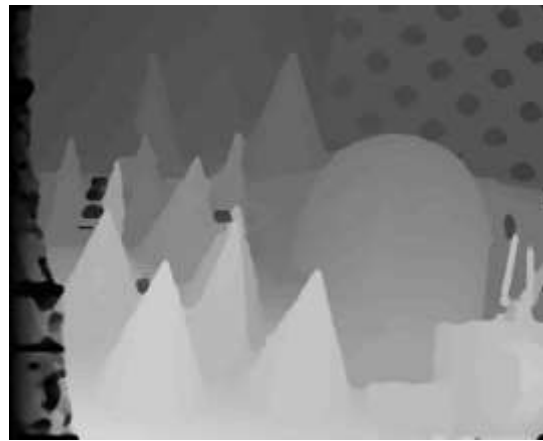
Tsukuba



Venus



Teddy



Cones

3. Bad pixel ratio

```
[Bad Pixel Ratio]
Tsukuba: 4.83%
Venus: 4.37%
Teddy: 13.85%
Cones: 11.83%
Average: 8.72%
```

4. Reference paper:

[1] **Binary Stereo Matching**, Kang Zhang et.al. Feb 2014

[2] **Patchmatch stereo - stereo matching with slanted support windows**, C. R. Michael Bleyer et.al. In Proceedings of the British Machine Vision Conference, pages 14.1–14.11. BMVA Press, 2011