

Computer Vision Homework 1 Report

311554057 陳宇呈

1. Normal Estimation :

在 `read_all_bmp()` 中我依序將 6 張 bmp 圖片讀入，並且將每張圖片從二維攤平成一維向量然後依序放入陣列中，之後將陣列轉換為矩陣方便後面的矩陣計算，這樣就能得到以下公式中的 I 。

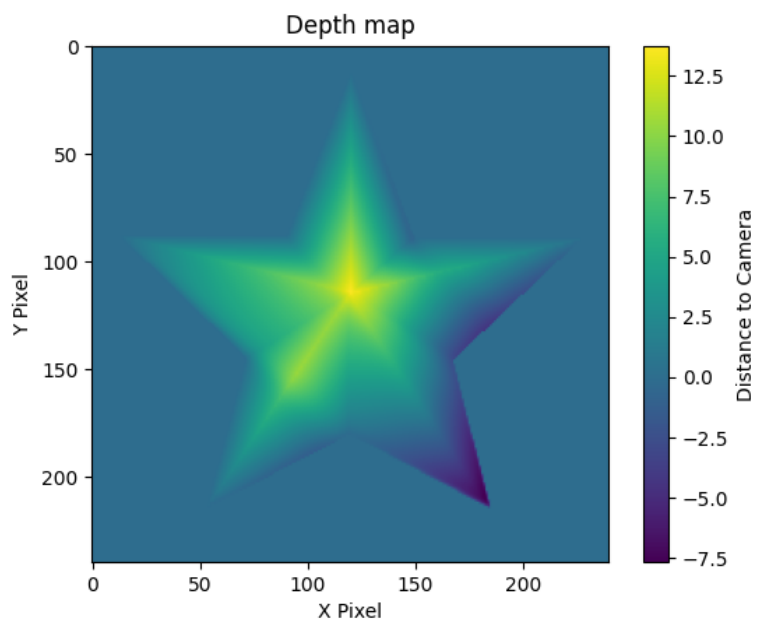
接下來是要讀入光線單位向量，一樣先依序讀入光線向量然後對每一行個別做 `Normalize` 我們就會得到 6 個光線單位向量在陣列中，並且剛前面一樣將陣列轉換為矩陣，得到以下公式中的 L 。

$$K_d N = (L^T L)^{-1} L^T I$$

有了 I 和 L 後我們就可以去計算 $K_d N$ ，只要按照公式去計算我們就可以得到 $K_d N$ ，得到 $K_d N$ 後對其做 `Normalize`，之後用提供的 `normal_visualization()` 就能夠得到 Normal Map。

2. Surface Reconstruction :

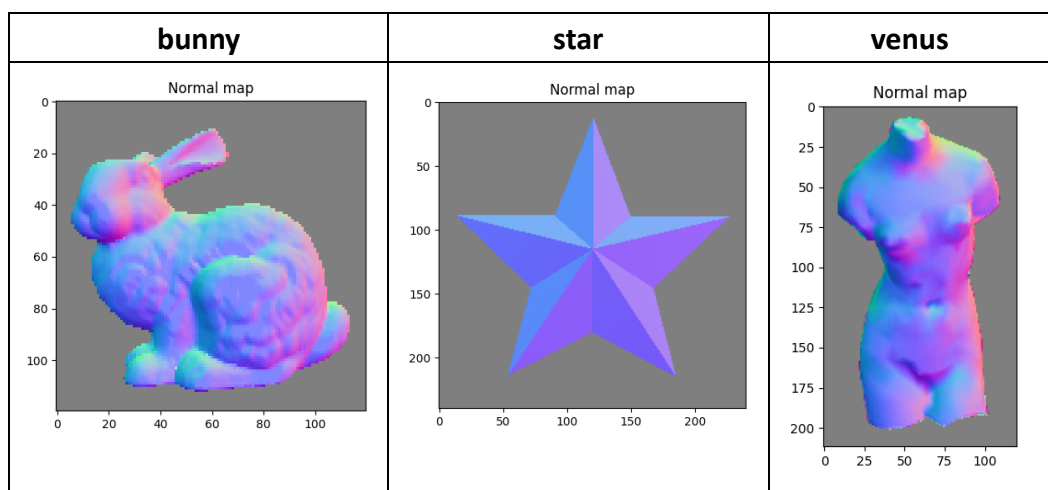
這邊我採用積分法來完成 Surface Reconstruction 的，首先我用上面最後得到的 Normal Map 來計算每個 pixel 的 x 軸梯度 $\left(-\frac{n_1}{n_3}\right)$ 以及 y 軸梯度 $\left(\frac{n_2}{n_3}\right)$ ，因為 `tangent plane` 建立在原點在左下角的情況，所以導出來的原點也是在左下，簡單來說就是 3D 座標原點與影像不同，因為我一開始是從左上角當作原點出發，而 y 方向不同所以取負號，因此求 y 軸梯度不按照 ppt 上的公式而是改為 $\left(\frac{n_2}{n_3}\right)$ 。得到 x 和 y 軸梯度後就可以開始構建 `depth map`，我是將起點跟若 x 和 y 軸梯度同時為 0 時就將深度設為 0，其他情況就是將上方及左方 pixel 和目前 x 和 y 軸梯度去積分並取平均，因為 pixel 是離散所以積分就是梯度累加，而上方及左方 pixel 其實就是累加後的結果因此這邊會加入上方及左方 pixel，就可以得到下方結果。



以“star”來看可以看到說右下角離原點越遠誤差越大，因此我選擇從四個頂點出發分別計算它們的 x 和 y 軸梯度然後積分，最後將得到的 4 張 depth map 取平均就是最後的結果。

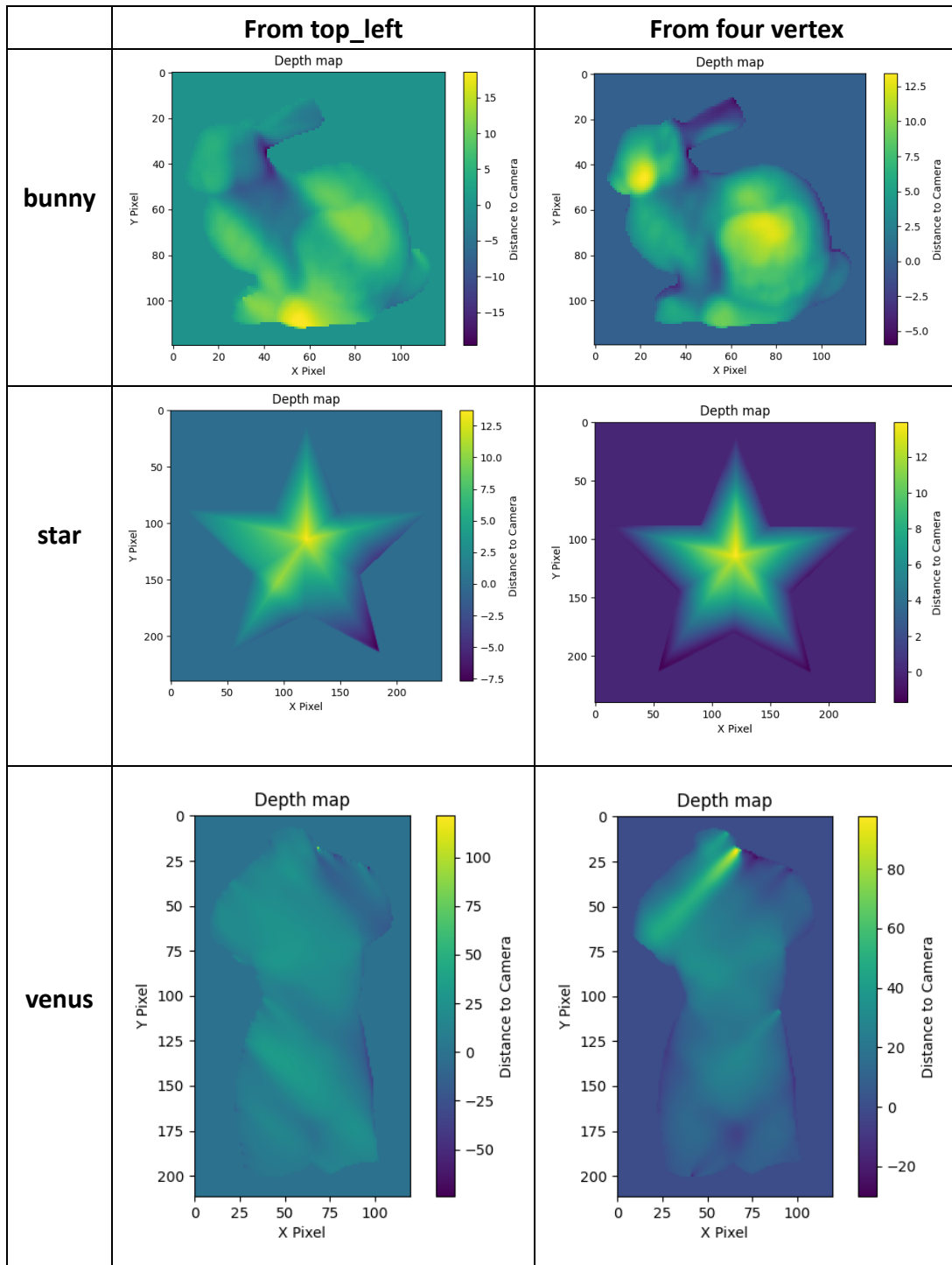
3. Result :

Normal Map :



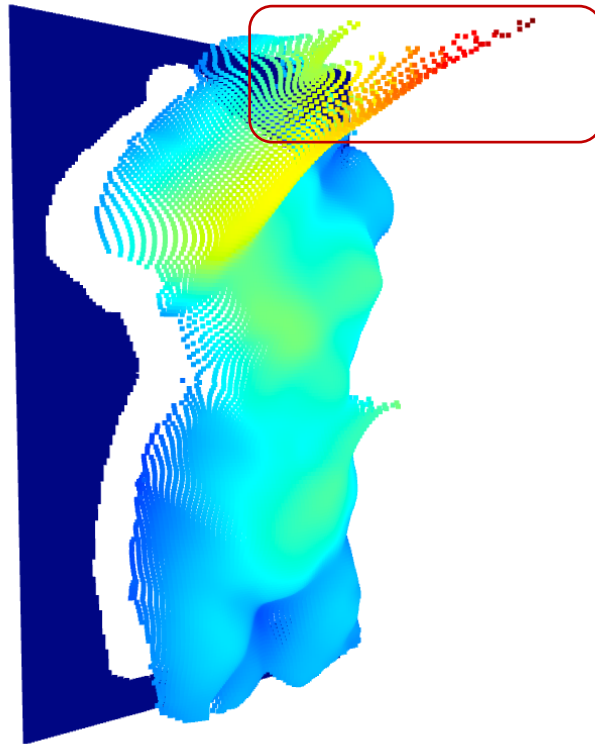
Depth Map :

左邊為從左上角當作原點，右邊為從四個頂點出發取平均。

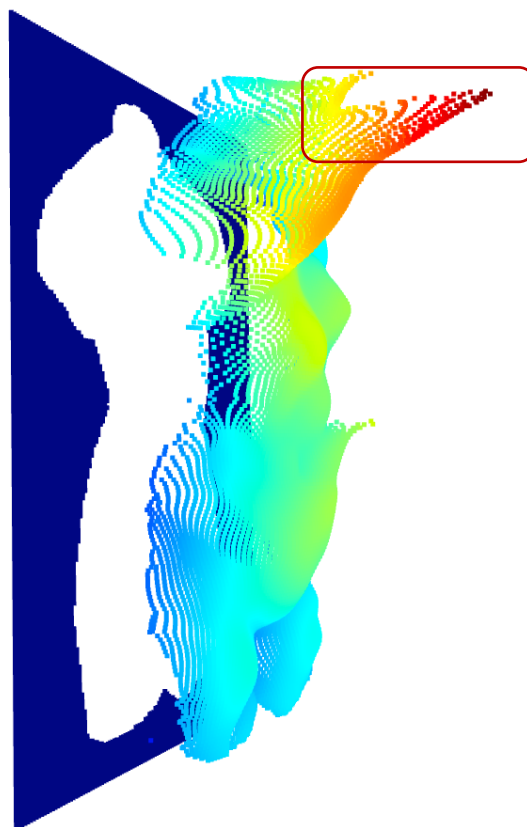


4. Reconstruct Surface of “venus” :

在這部分我一直找不到很好的方法去處理 extreme normal result，可以看到下圖為未處理時的“venus”，會有一大塊凸出來的部分。



最後我選擇設定 threshold 為 50，作為若超過這個高度就將目前 pixel 的值設為其左上、右上、右下和左下相合的平均雖然說不能完全解決突出問題但能夠看到有減緩的趨勢且更密集。



其 depth map 如下：

