

# CV HW1

310552056 楊信一

## 1. Implementation

首先將 6 張圖以灰階圖的形式讀進，並讀進光源的文字檔並 `normalize` 為  $L$ ，再把 6 張圖排成  $I$ ，然後解  $K_d N L = I$ ，做 `pseudo inverse` 取得  $K_d * N$ ，對  $K_d * N$  做 `normalize` 得到  $N$

接著採用第二種矩陣方法解  $Mz = V$ ，矩陣  $M$  用 `scipy.sparse` 下的 `lil_matrix` 矩陣類型來儲存，並使用 `mask` 是否為 0 判斷 `pixel` 是否在物體範圍內，將  $M$  和  $V$  在物體範圍內的值填上 1 或 -1，然後同樣解此方程式得到  $z$ ，並將結果存到  $D$  矩陣中輸出和存檔

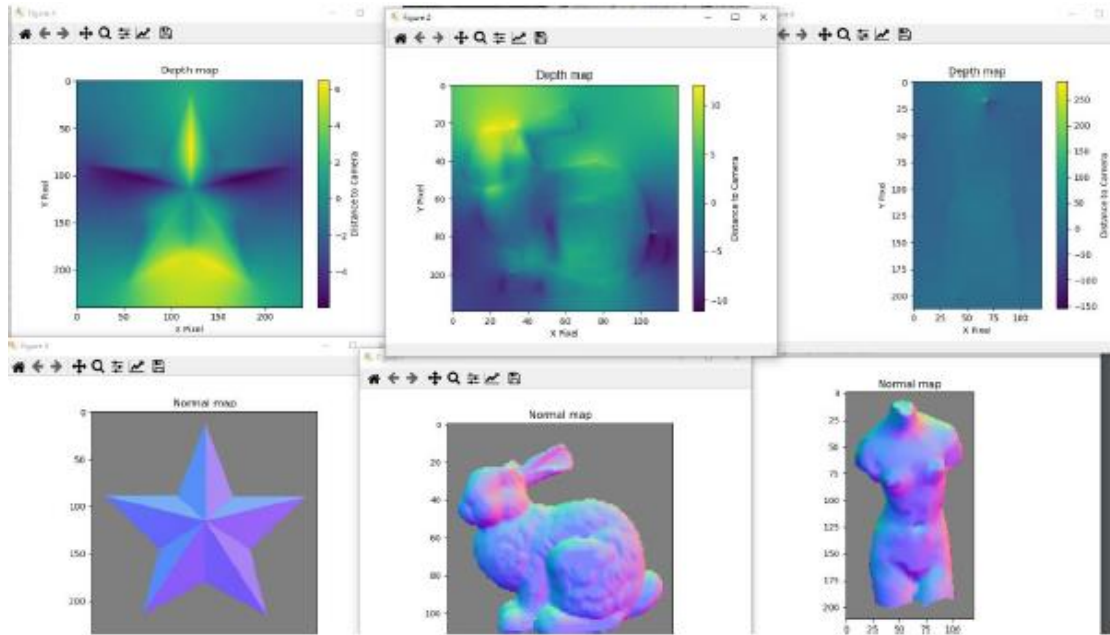
## 2. Method used to enhance result

- 用 `mask` 的方法濾掉已知不在圖案內的 `pixel`，其中以每種圖的 `img1` 當做 `mask` 並判斷 `pixel` 值是否為 0 (為 0 代表不在圖案內)，然後根據是否為 0 來設定矩陣  $M$  值(`img1` 的 `pixel` 值為 0 則  $M$  上對應處就設 0)
- (對 `venus`)設定 `bot` 和 `top` 值做為  $Z$  值的上下限，若超出此限則設為 `bot` 或 `top`

```
171         # fix the outlier
172         if img_namelist[k] == 'venus':
173             bot_Z = np.mean(Z) - 22
174             Z[Z < bot_Z] = bot_Z
175             top_Z = np.mean(Z) + 28
176             Z[Z > top_Z] = top_Z
177
```

## 3. Compare result

- Without enhancement:



● With enhancement:

