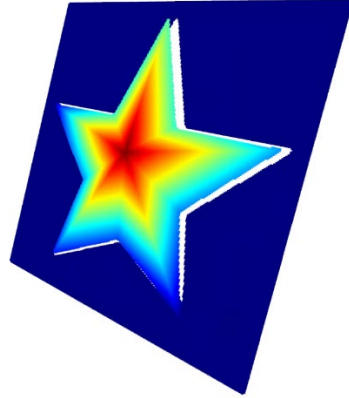
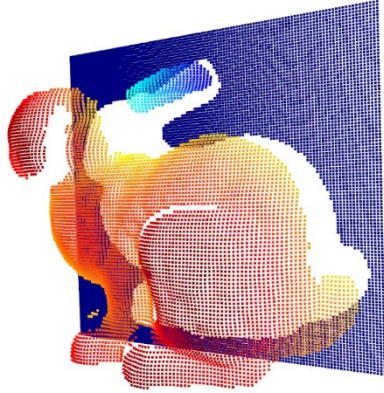


# Report

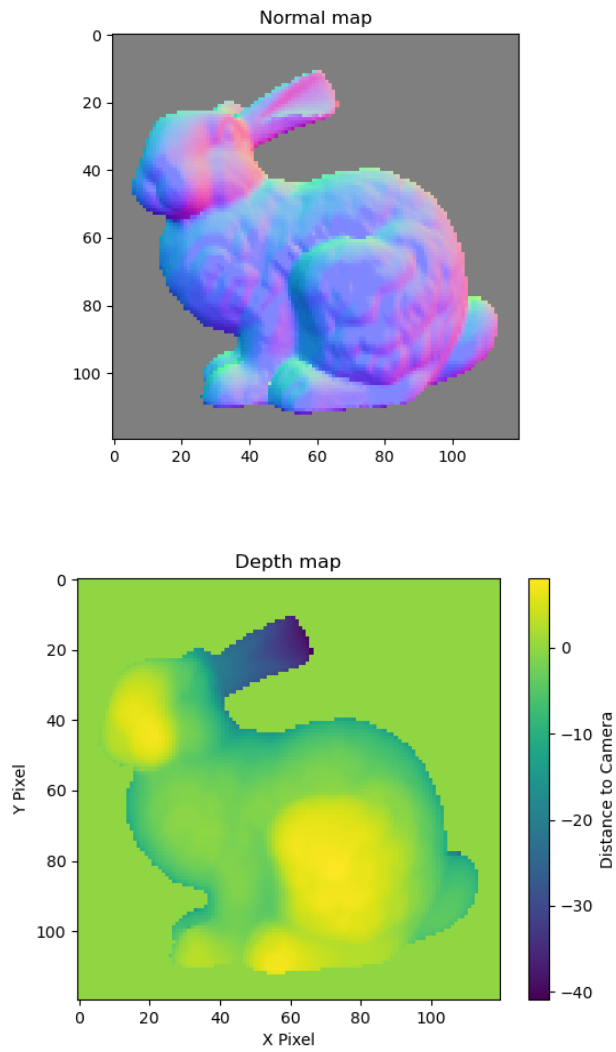
## 1. Reconstruct surfaces of “bunny” & “star” data

- 兩張圖皆順利重建出表面並以 ply 存檔，下圖為完成截圖

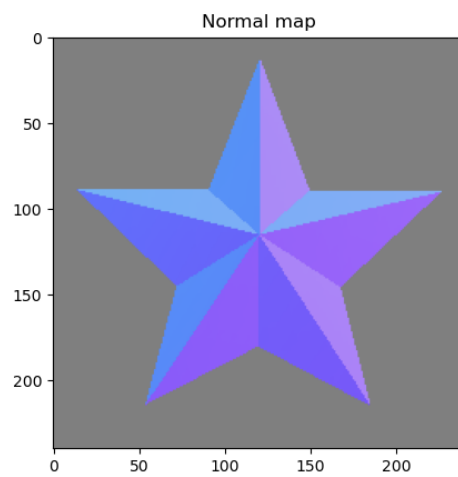


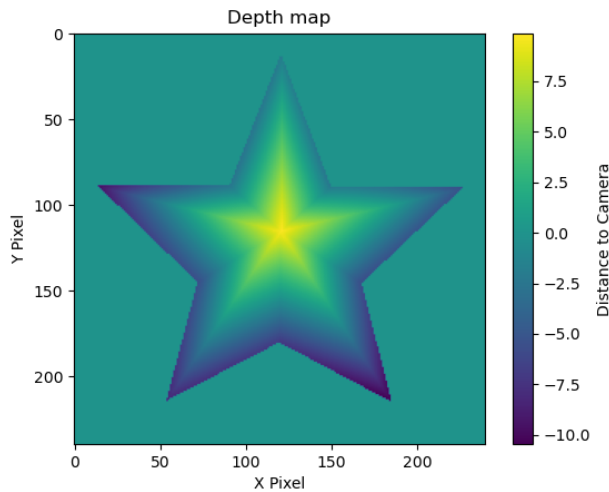
2. show the normal map and depth map of “bunny” & “star”

- bunny



- star





3. Simply explain your implementation and what kind of “method” you use to enhance the result and compare the result

- 首先將 3D 重建欲使用到的圖片以及光源文件檔讀入，並且將光源資訊轉換成單位向量再以 array 方式儲存。
- 第一步使用 Lambertian Reflection 公式尋找欲重建物件之表面法向量值，公式如下：

$$I = LK_d \mathbf{N}$$

透過從每張照片得知的表面光源強度  $I$  及光源向量  $L$ ，對每個 pixel 求出  $K_d \mathbf{N}$ ，計算方式則使用 SVD 奇異值分解求處該值。接著由於  $\mathbf{N}$  值必須為單位向量，透過計算  $K_d \mathbf{N}$  的單位長度再對其相除取得，同時其單位長就是  $K_d$  的資訊，該法向量資訊即可拿來計算後續深度重建部分。

$$\mathbf{N} = \frac{K_d \mathbf{N}}{\|K_d \mathbf{N}\|}$$

- 第二部首先使用前項步驟求出的 albedo 當作 mask，將無深度資訊的 pixel 濾除，並記錄含深度值的 pixel 行列值。接著透過切平面特性(切線向量與法向量內積為 0)，得出以下方程式：

$$z_{x+1,y} - z_{x,y} = -\frac{n_x}{n_z}$$

$$z_{x,y+1} - z_{x,y} = -\frac{n_y}{n_z}$$

每個 pixel 值皆可求出此兩項方程式，因此可以將所有 pixel 對應的方程式以矩陣形式儲存等號左邊之  $z$  值係數(M)以及等號右邊之法向量相除數值(V)：

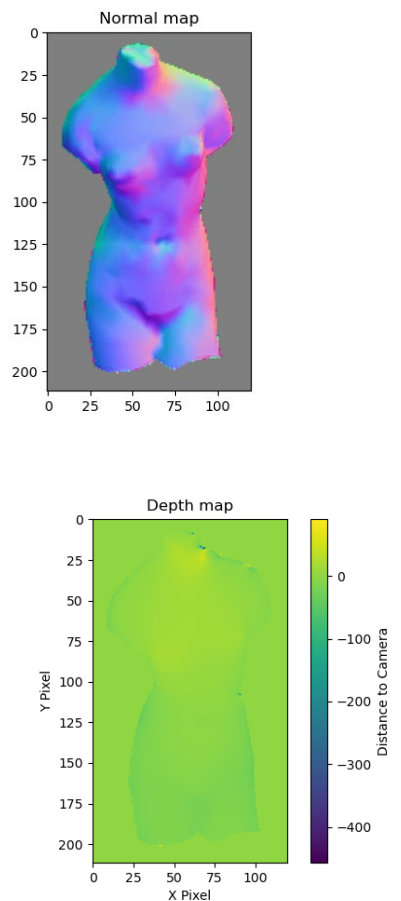
$$\begin{matrix} & \textcolor{blue}{M} & & \textcolor{blue}{z} & & \textcolor{blue}{V} \\ \textcolor{red}{2S \times S} \left[ \begin{array}{cccccc} \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \cdots & -1 & 1 & \cdots & \cdots & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \cdots & -1 & \cdots & \cdots & 1 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{array} \right] & \textcolor{red}{S \times 1} \left[ \begin{array}{c} \vdots \\ z_{50,50} \\ z_{51,50} \\ \vdots \\ z_{50,51} \\ \vdots \end{array} \right] & = & \textcolor{red}{2S \times 1} \left[ \begin{array}{c} \vdots \\ -\frac{n_x^{50,50}}{n_z^{50,50}} \\ \vdots \\ \vdots \\ -\frac{n_y^{50,50}}{n_z^{50,50}} \\ \vdots \end{array} \right] \end{matrix}$$

透過迴圈取得每個 pixel 的係數後，最後解出  $Mz=V$  之  $z$  值矩陣，並將其映射成深度圖，重建出 3D 表面模型。由於該矩陣為一稀疏矩陣，因此我先將其轉換成 lil sparse matrix 再進行計算，以提升計算效率。

- 前面的 M 值設計部分，我另外考慮到邊界問題：當該 pixel 周圍包含無深度之 pixel，則必須改由挑選其他合適的向量來修正前述方程式；若皆無包含深度之 pixel，則該 row 係數皆為 0(不納入考量)，此方法能夠對物件與背景之邊界區域做優化，重建出的模型也會更加 smooth。

4. reconstruct surfaces of “venus”

- 首先為 venus 的法向量及深度圖：



- 為了優化 venus 之重建效果，我們使用 open3d 所提供的 outlier removal 函式將雜訊或距離太遠的點雲移除，同時考慮到邊界問題的 M 值設計，也能夠改善重建效果。最後結果如下(也有 venus.ply 紀錄)：

