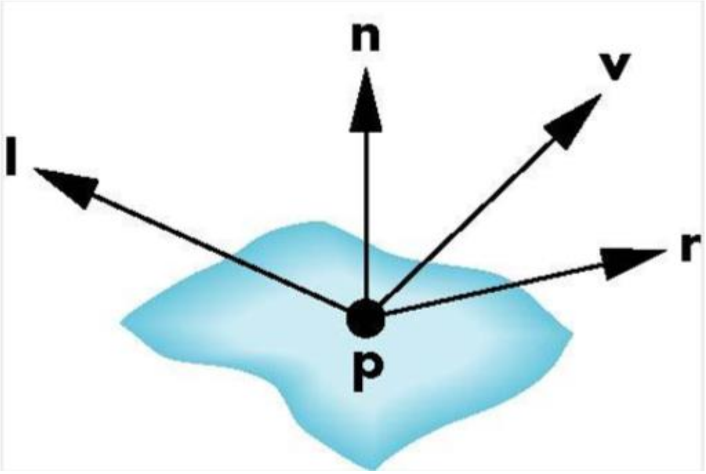
HW 1 (Due :04/03)

Photometric Stereo

Phong reflection model

- A simple model that can be computed or analyzed rapidly.
- Has three components
  - Ambient
  - Diffuse
  - Specular



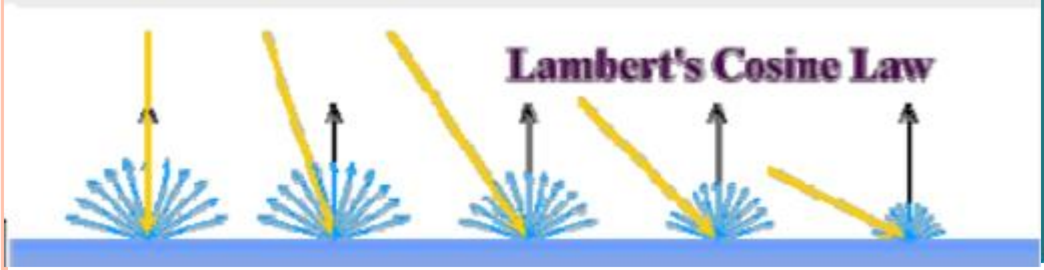
- Uses four vectors
  - To source *l*
  - To viewer *v*
  - Normal *n*
  - Perfect reflector *r*

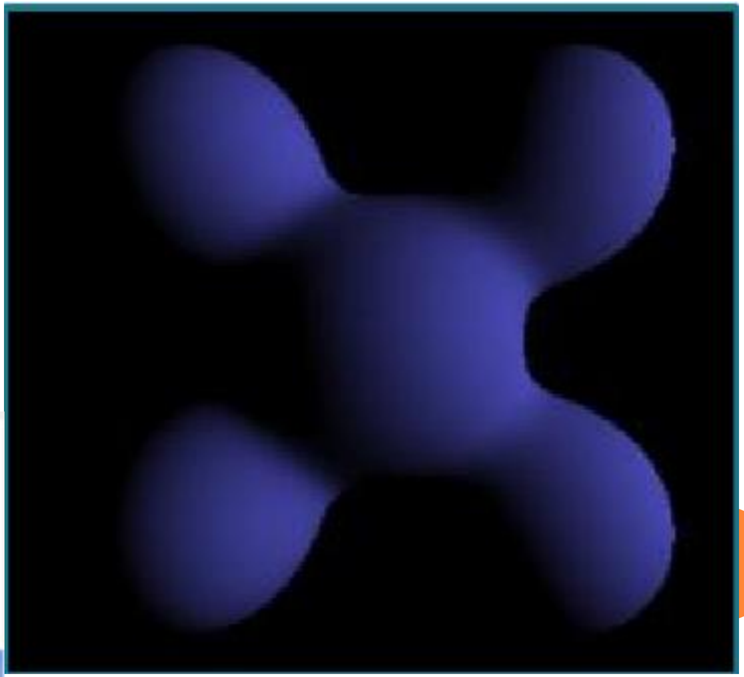
AMBIENT LIGHT

- The result of multiple interactions between (large) light sources and the objects in the environment.
- $I_{Ambient} = K_a * I_a$

DIFFUSE LIGHT

- Light scattered equally in all directions
- Reflected intensities vary with the direction of the light
- Lambertian Surface
  - Perfect diffuse reflector
- $I_{Diffuse} = K_d * I_d(n\boldsymbol{l})$





MODELING SPECULAR REFLECTIONS

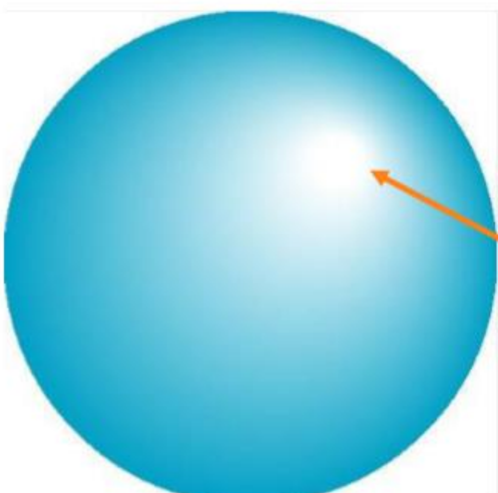
- Phong proposed
- $I_r \sim K_s I cos^a \Phi$

Shininess coef.

Reflected intensity

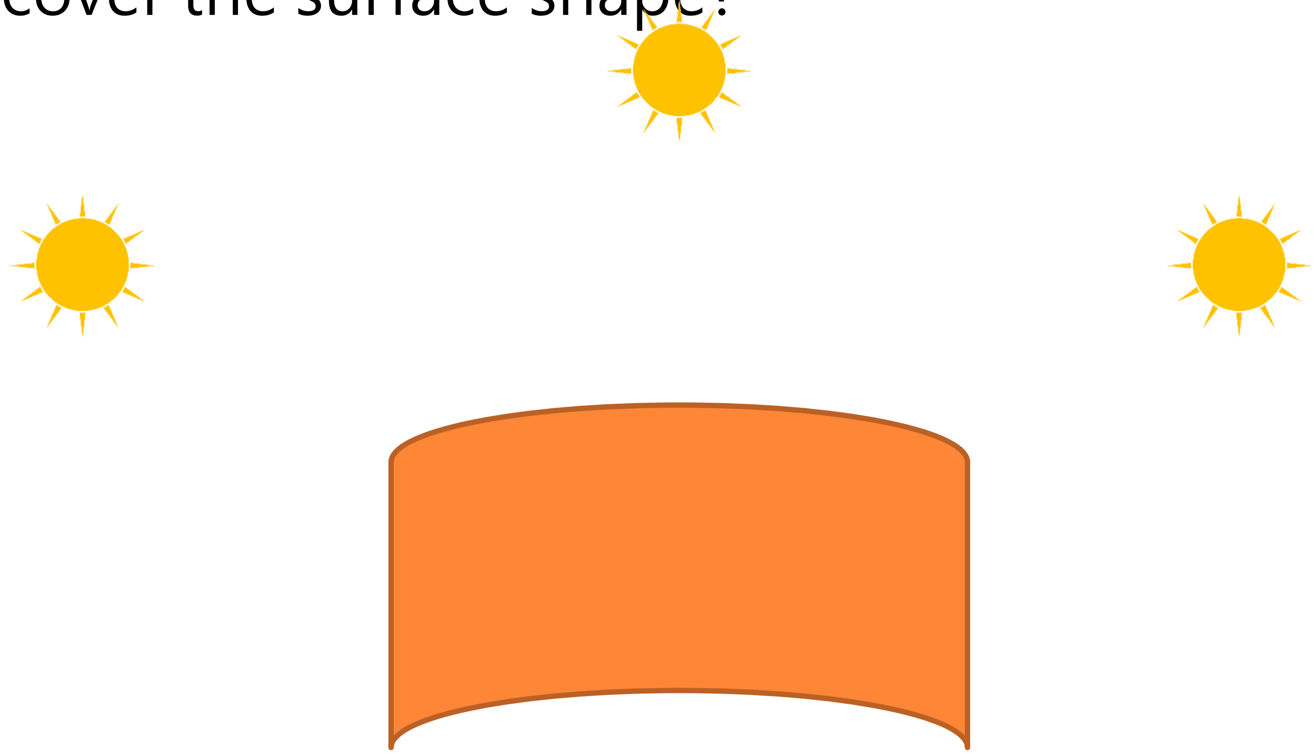
Incoming intensity

Absorption coef.



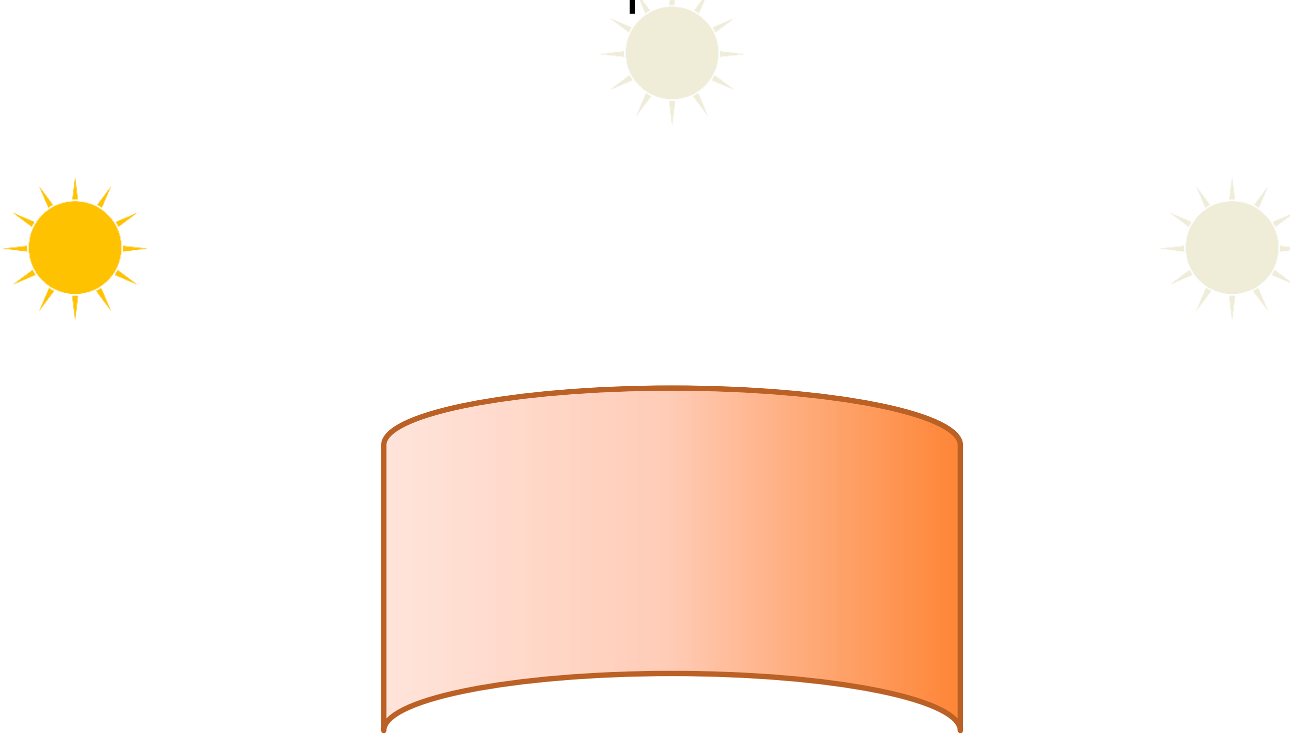
PHOTOMETRIC STEREO

- Given multiple images of the same surface under different known lighting conditions, can we recover the surface shape?



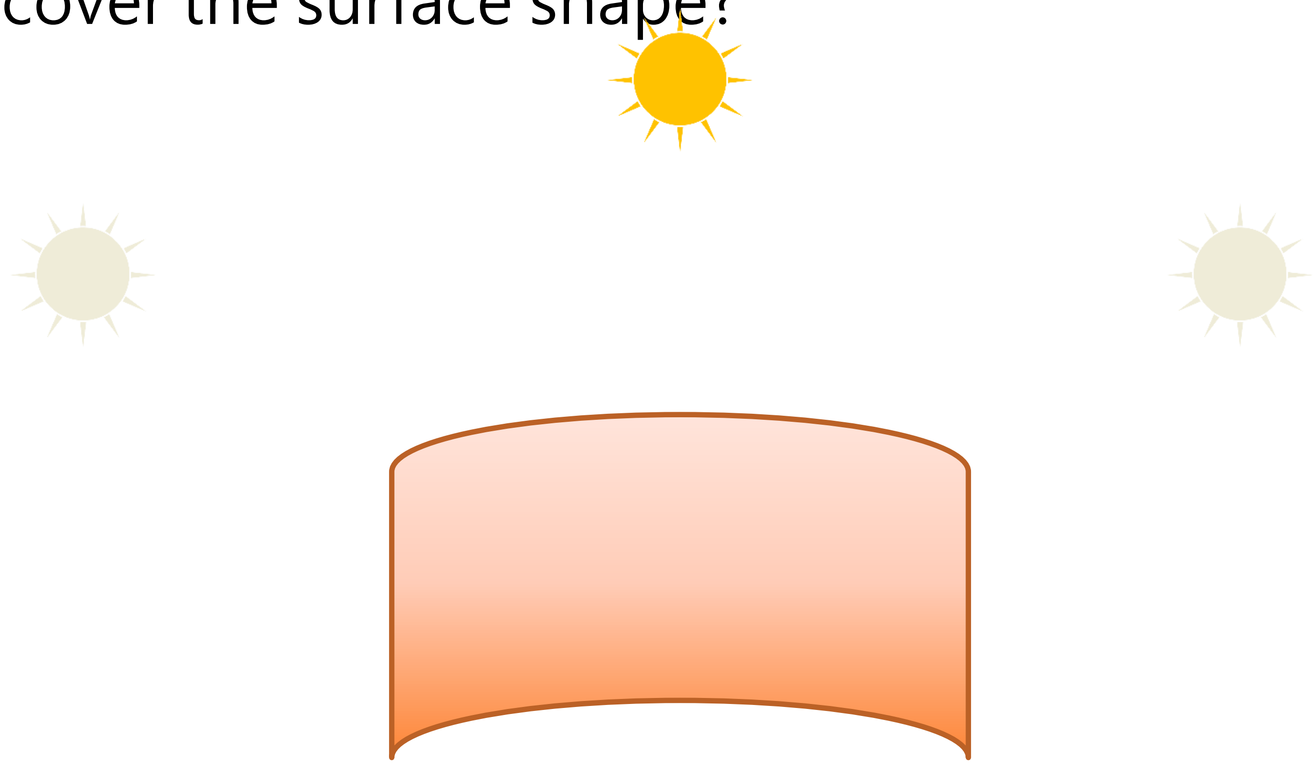
Photometric stereo

- Given multiple images of the same surface under different known lighting conditions, can we recover the surface shape?



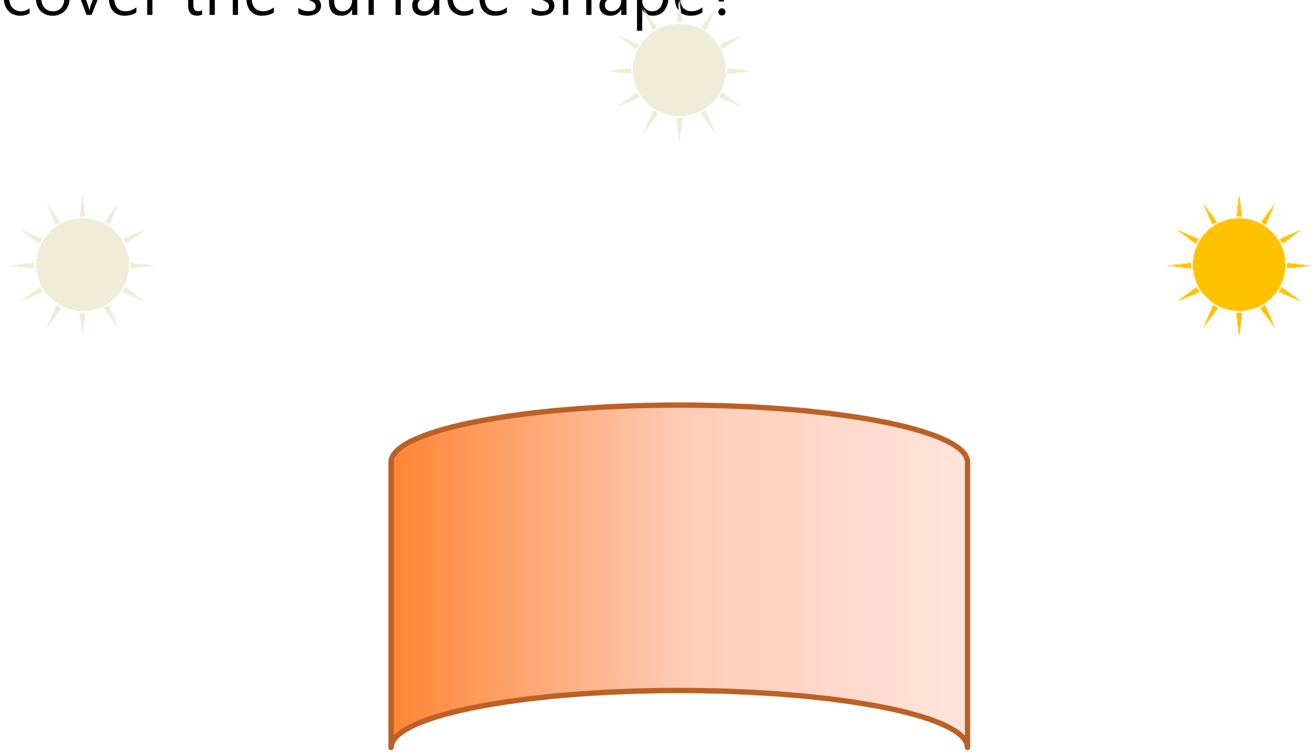
Photometric stereo

- Given multiple images of the same surface under different known lighting conditions, can we recover the surface shape?



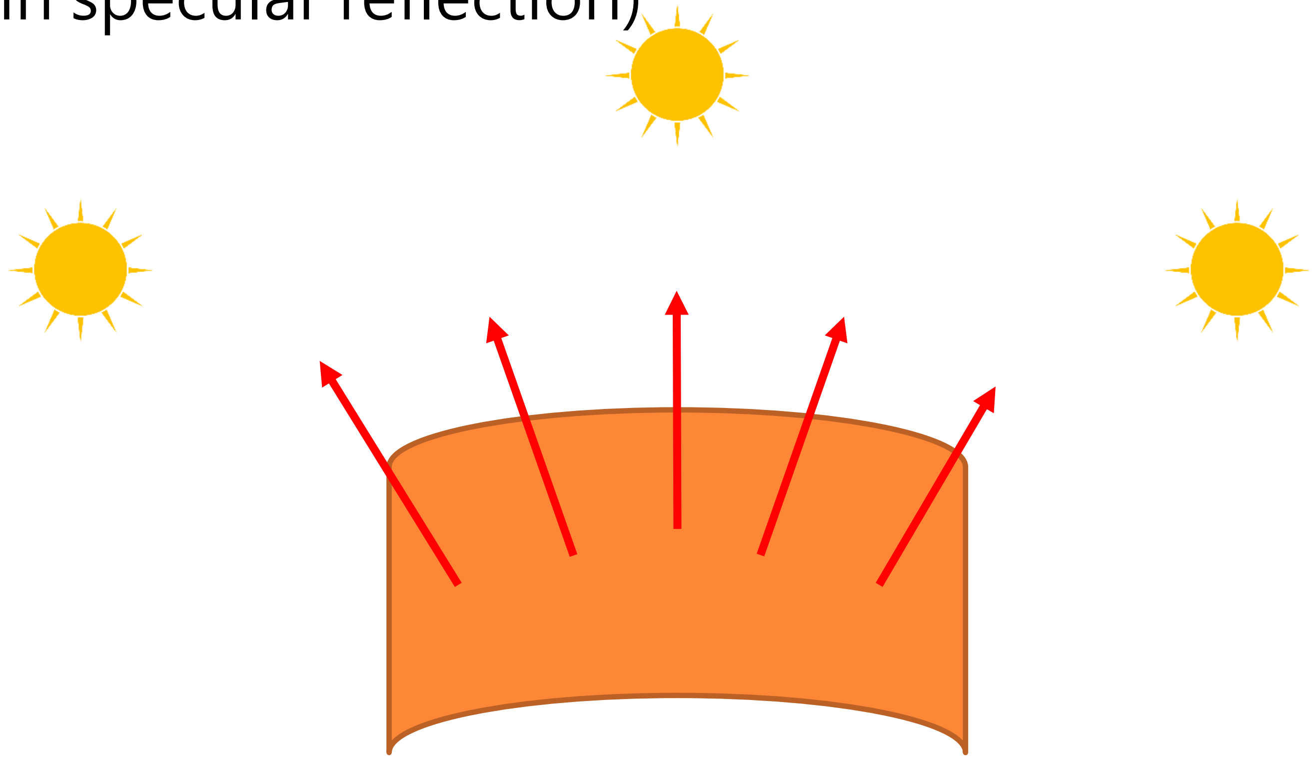
PHOTOMETRIC STEREO

- Given multiple images of the same surface under different known lighting conditions, can we recover the surface shape?



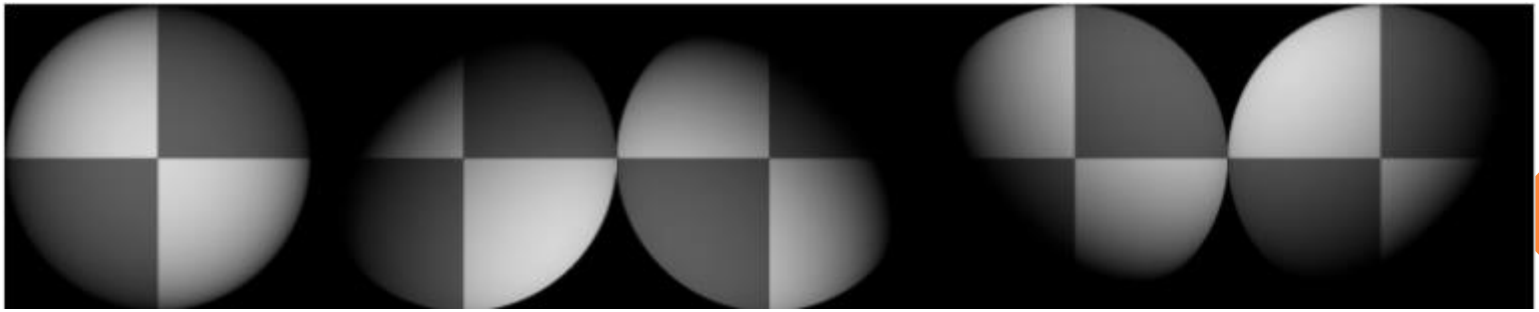
Photometric stereo

- Reminder: surface reflection is related to surface normal *N* and light source *L* (and view direction *V* in specular reflection)

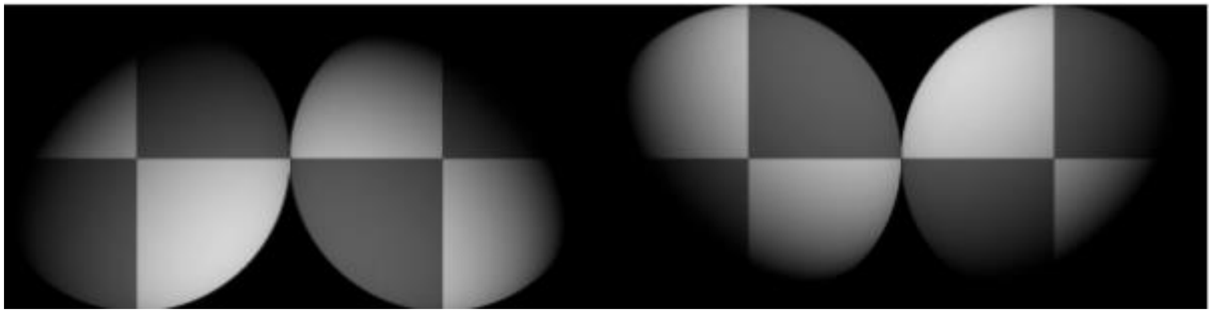


PHOTOMETRIC STEREO

- Using a local shading model
- A set of point sources that are infinitely distant
- A set of pictures of an object, obtained in exactly the same camera/object configuration but using different light sources
- A Lambertian (diffuse) object for simplification
  - (or the specular component has been identified and removed)



LAMBERTIAN SURFACES



對畫面中任一個點顏色值為I_i, $i = 1\ to\ 4$：

該點的法向量(unknown)： $n = (n_x, n_y, n_z)$

光線方向(known)： $s_i = (s_{x_i}, s_{y_i}, s_{z_i})$

$$
\begin{aligned}
I_1 &= \rho(n_1 \cdot n) \\
I_2 &= \rho(n_2 \cdot n) \\
I_3 &= \rho(n_3 \cdot n) \\
I_4 &= \rho(n_4 \cdot n)
\end{aligned}
$$

$$
\begin{bmatrix} I_1 \\ I_2 \\ I_3 \\ I_4 \end{bmatrix} = \rho \underbrace{\begin{bmatrix} s_1^T \\ s_2^T \\ s_3^T \\ s_4^T \end{bmatrix}}_{= S} n
$$

$$
\rho n = S^{-1} \begin{bmatrix} I_1 \\ I_2 \\ I_3 \\ I_4 \end{bmatrix} = N
$$

$$
n = N/|N|
$$

$$
\rho = |N|
$$

Recovering the surface

The surface can be represented as $(x, y, f(x,y))$.

From the surface gradient vectors, we can evaluate the surface normal as:

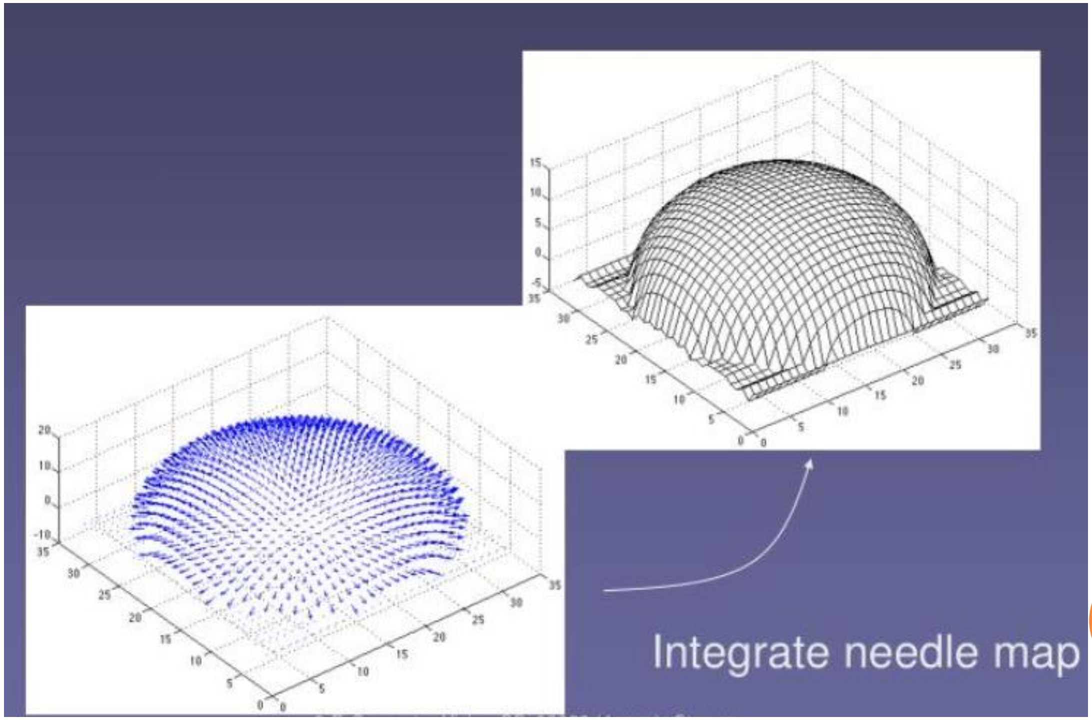
$$\boxed{N(x,y) = \frac{\left(-\frac{\partial f}{\partial x},\ -\frac{\partial f}{\partial y},\ 1\right)^T}{\sqrt{1 + \frac{\partial f}{\partial x}^2 + \frac{\partial f}{\partial y}^2}}}$$

Therefore, if estimated $N(x,y)$ is $(\, N_a(x,y), N_b(x,y), N_c(x,y)\,)^T$, we get the partial derivative:

$$\boxed{\frac{\partial f}{\partial x} = \frac{-N_a(x,y)}{N_c(x,y)}} \qquad \boxed{\frac{\partial f}{\partial y} = \frac{-N_b(x,y)}{N_c(x,y)}}$$

還原深度

- Normal 可以視為兩點間的梯度，所以…
  - 從任意位置開始(四角 or 中間)，根據梯度累加出平面
  - 缺點：平面會歪、只考慮單一方向而非考慮整體的梯度關係
  - How to solve?



ALGORITHM$_{-1}$

- Obtain many images in a fixed view under different illuminants

- Determine the matrix ***S*** from light source

- Create arrays for albedo, normal (3 components),

  p (measured value of $\partial f/\partial x$) and

  q (measured value of $\partial f/\partial y$)

ALGORITHM$_{-2}$

For each point in the image array

  Stack image values into a vector $\boldsymbol{i}$

  Construct the weight matrix $\boldsymbol{w}$

  Solve $\boldsymbol{wi} = \boldsymbol{wSb}$ to obtain b for this point

  Albedo $\boldsymbol{\rho}$ at this point is $|\boldsymbol{b}|$

  Normal $\boldsymbol{N}$ at this point is $\boldsymbol{b}/|\boldsymbol{b}|$

  $\boldsymbol{p}$ at this point is $-\boldsymbol{N_a}/\boldsymbol{N_c}$

  $\boldsymbol{q}$ at this point is $-\boldsymbol{N_b}/\boldsymbol{N_c}$

End

Check: is $(\partial p/\partial y - \partial q/\partial x)^2$ small everywhere?

Algorithm$_{-3}$

top left corner of height map is zero

for each pixel in the left column of height map

height value=previous height value + corresponding q value

End

for each row

for each element of the row except for leftmost

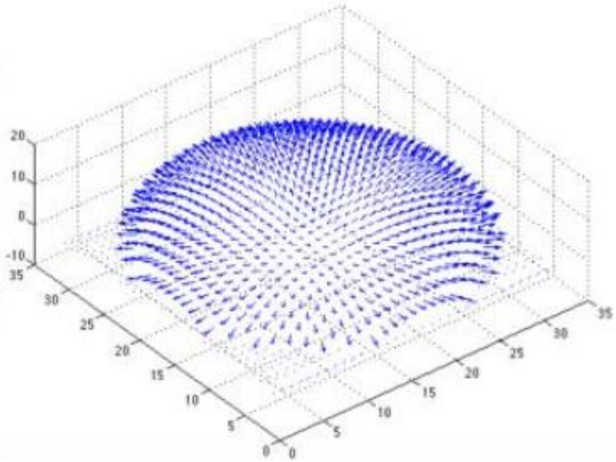
height value = previous height value + corresponding p value

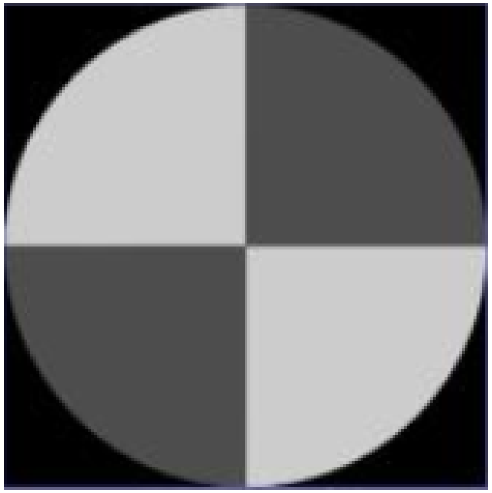
end

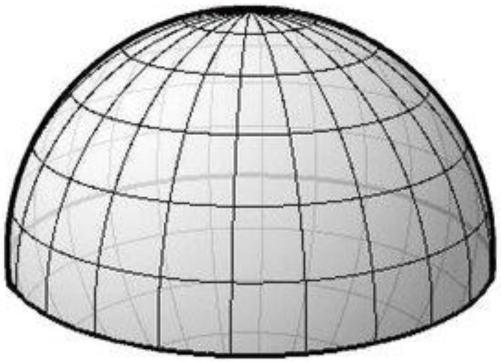
end

作業要求

- 根據範例測資計算出albedo、orientation… 70%
- 根據範例測資計算出三維表面(累加)… 20%
- 自己挑選一個模型拍攝並計算出其三維表面 10%
- Bonus: 改良計算出來的三維表面Ex: 最佳化 20%
- 嚴禁抄襲







輸出

- For each case (範例測資1、2 & 自己拍的測資):
- Output orientation as a color map
- Output albedo as a color map
- Output depth file as a text file
  - 依照row/colomn, 一個一個輸出每個點的深度值，中間用空白隔開，一行過後換行
  - Ex: 255 255 240 240 10 10

    255 255 240 240 10 10

    240 240 240 240 10 10

    10 10 10 10 10 10

    0 0 0 0 0 0

    0 0 0 0 0 0

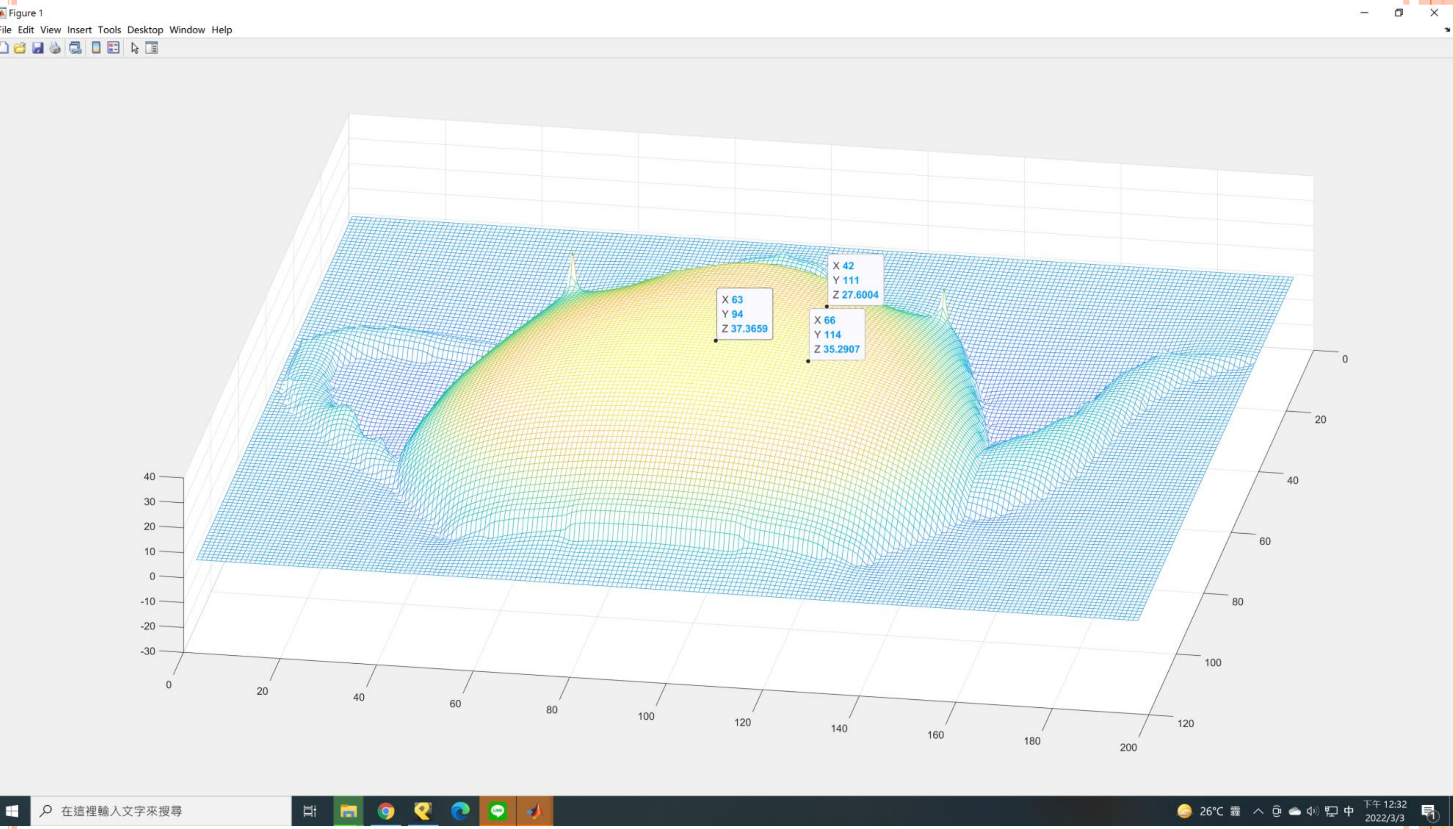
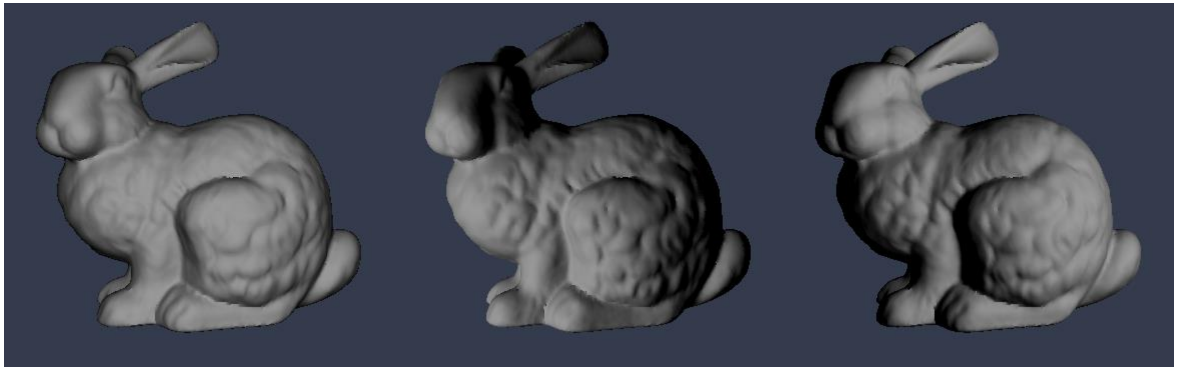


Figure 1
File Edit View Insert Tools Desktop Window Help
X 42
Y 111
Z 27.6004
X 63
Y 94
Z 37.3659
X 66
Y 114
Z 35.2907
40
30
20
10
0
-10
-20
-30
0
20
40
60
80
100
120
140
160
180
200
0
20
40
60
80
100
120
在這裡輸入文字來搜尋
26°C
下午 12:32
2022/3/3

繳交方式

- Deadline : 04/03(SUN) 23:59:59
- 請將
  - 報告(包含code, 註解,心得 兩組測資分別的結果圖包含albedo, Nx, Ny, Nz, Normal的彩色圖)，
  - 還原出來的深度值矩陣存成純文字檔(.txt)
- 壓縮後上傳至ilearn
- 要Demo
  - 我來想想怎麼Demo…



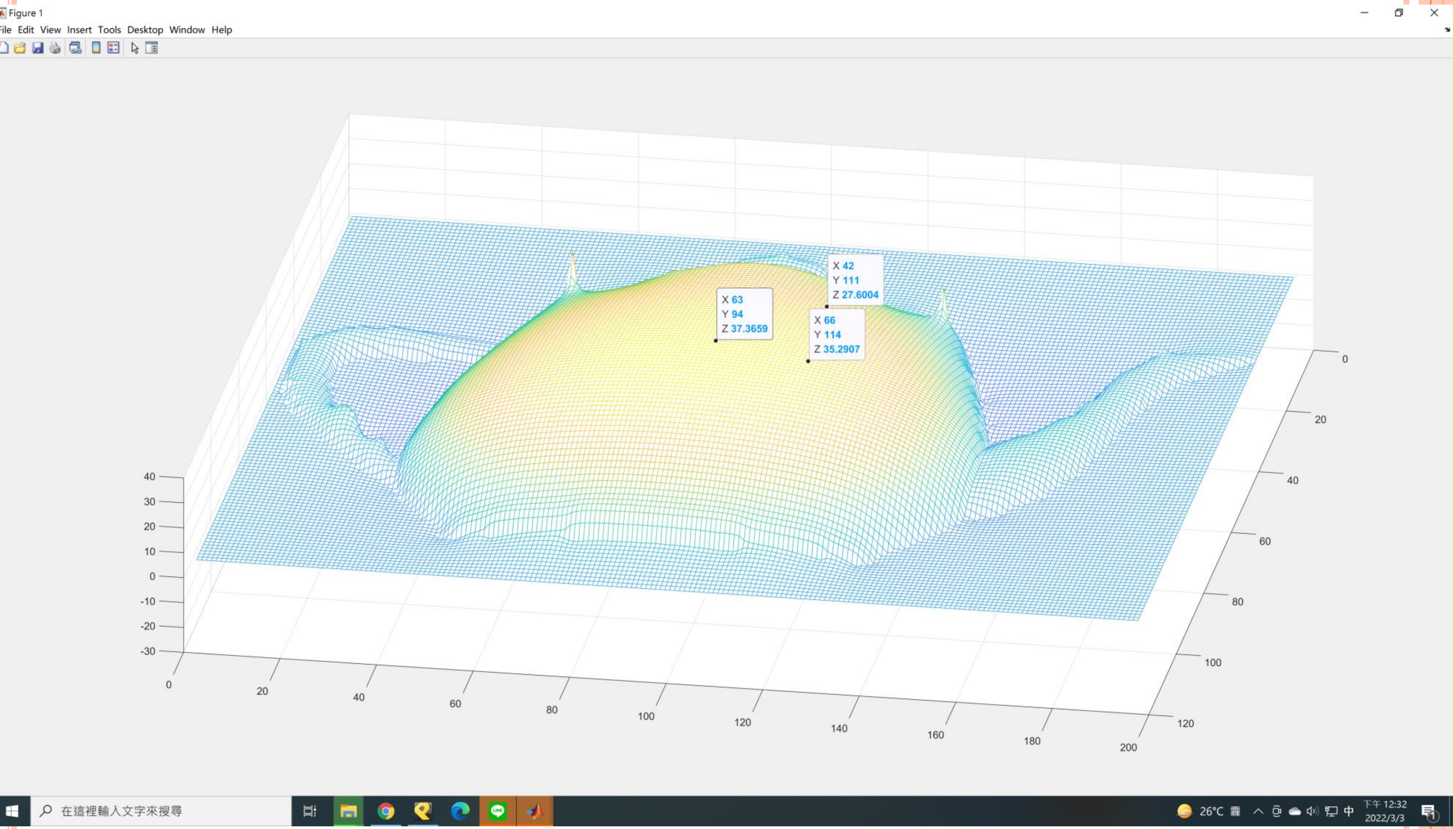


Figure 1
File Edit View Insert Tools Desktop Window Help
X 42
Y 111
Z 27.6004
X 63
Y 94
Z 37.3659
X 66
Y 114
Z 35.2907
40
30
20
10
0
-10
-20
-30
0
20
40
60
80
100
120
140
160
180
200
0
20
40
60
80
100
120
在這裡輸入文字來搜尋
26°C
下午 12:32
2022/3/3