

# Advanced Graphics Techniques

Computer Graphics

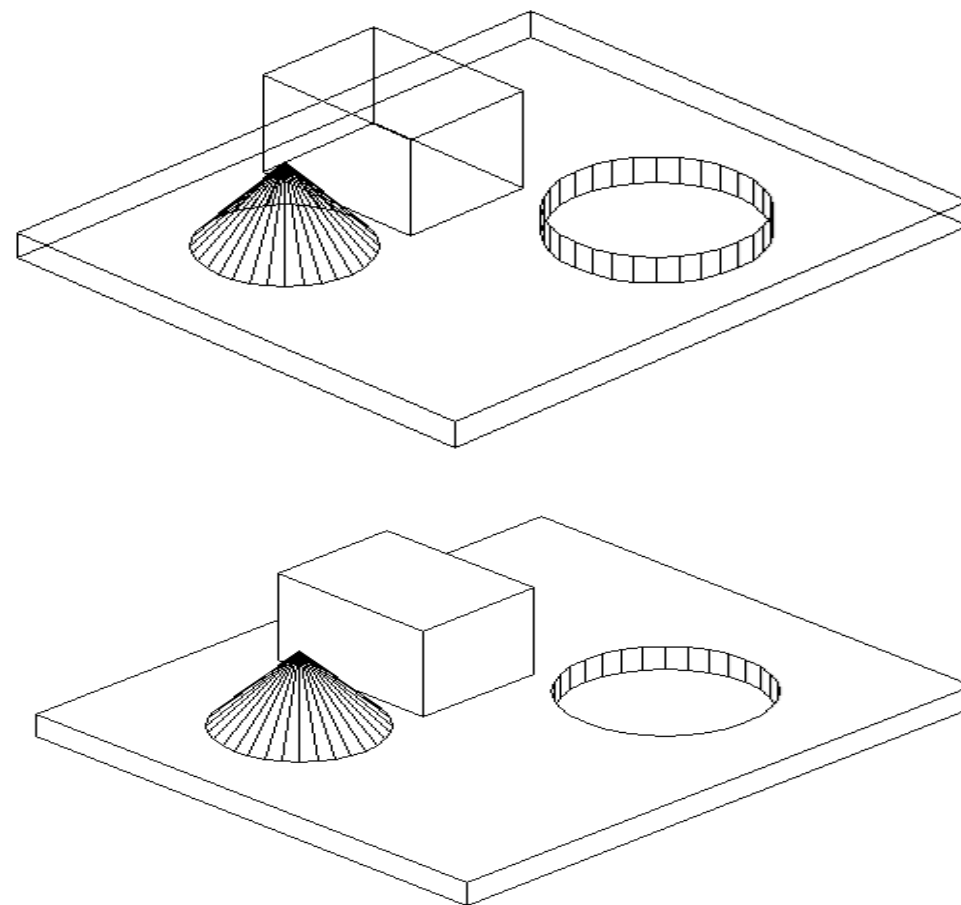
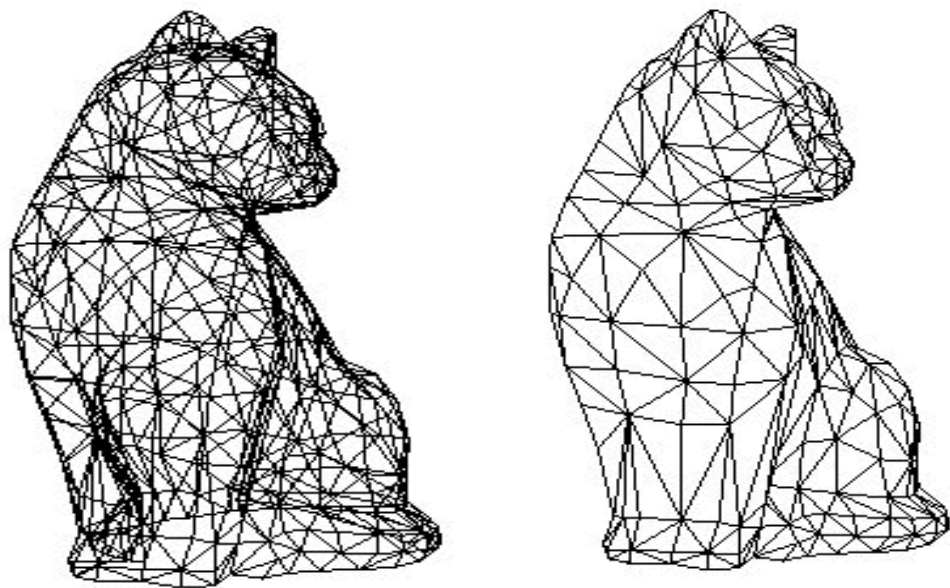


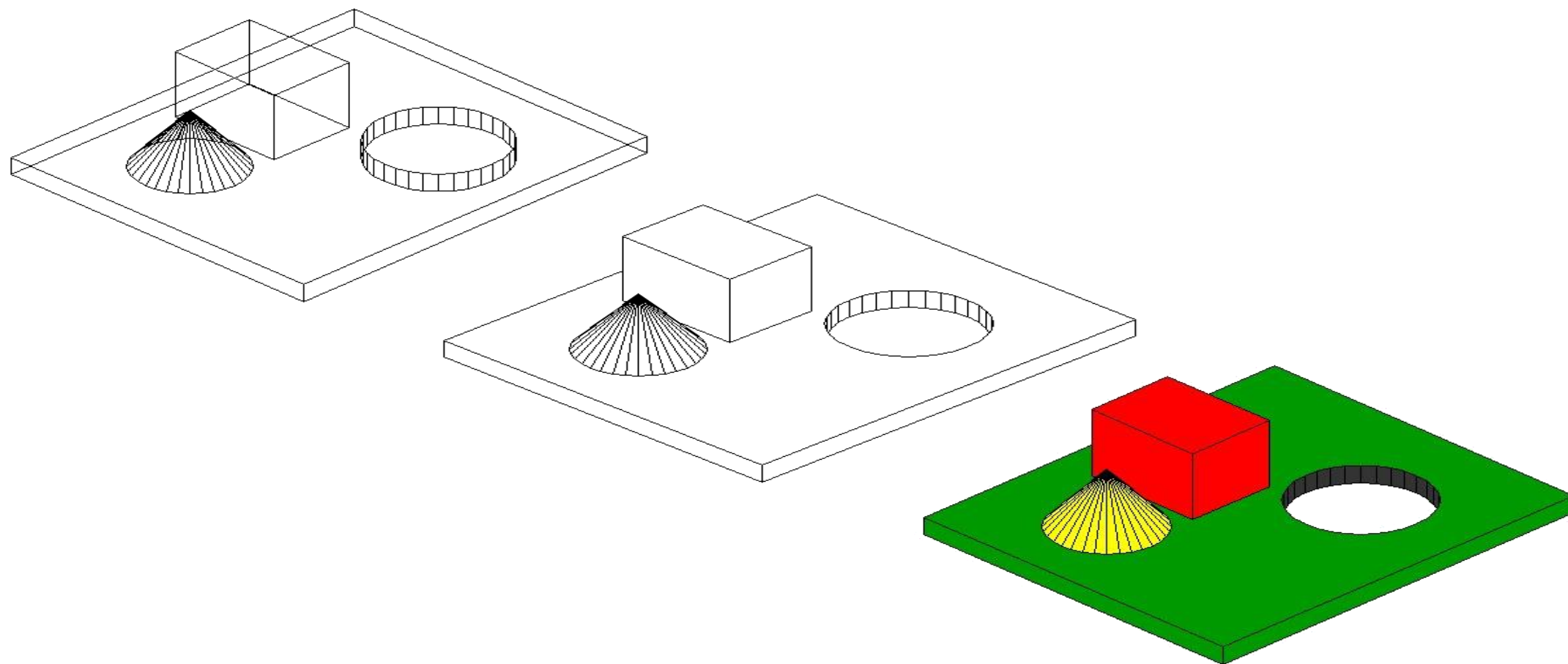
- **Khử đường, mặt khuất**
  - Thuật toán Back-face
  - Thuật toán Deep buffer
  - Thuật toán Deep sorting
- **Chiếu sáng và tô bóng**
  - Tô bóng hằng Lambert
  - Tô bóng nội suy Gouraud
  - Tô bóng nội suy Phong

# CONTENTS

- **Khử đường, mặt khuất**
  - Thuật toán Back-face
  - Thuật toán Deep buffer
  - Thuật toán Deep sorting
- **Chiếu sáng và tô bóng**
  - Tô bóng hằng Lambert
  - Tô bóng nội suy Gouraud
  - Tô bóng nội suy Phong

# Khử đường, mặt khuất





- **Các thuật toán khử khuất**
  - Thuật toán Back-face
  - Thuật toán Deep sorting
  - Thuật toán Deep buffer
  - ...

## Đại số vectơ

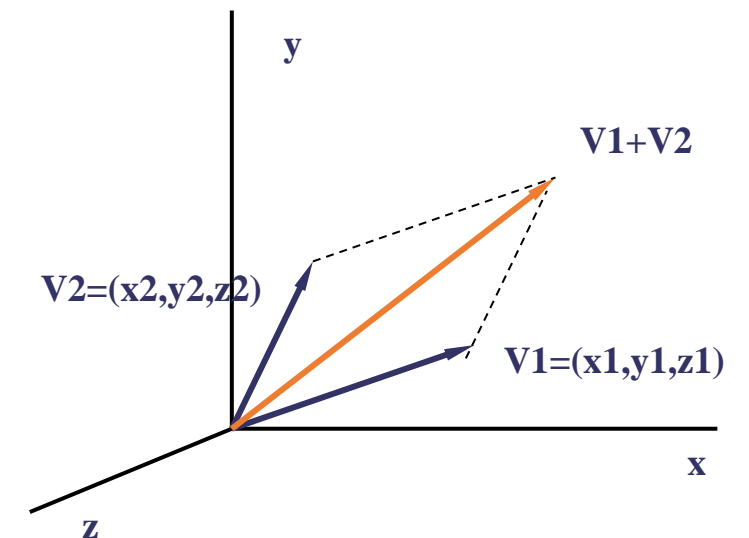
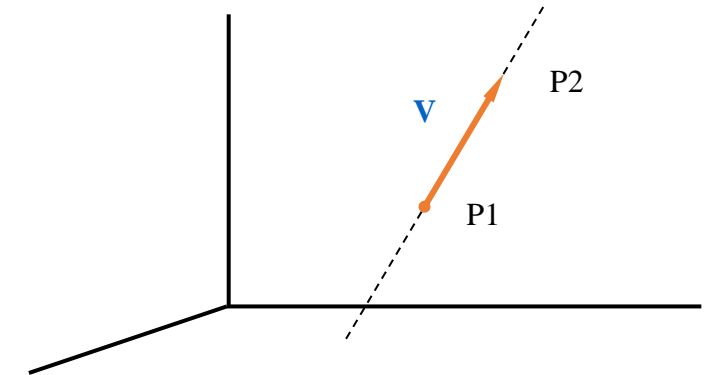
👉 Biểu diễn vectơ: đoạn thẳng có hướng

👉 Độ dài vectơ:

$$|\vec{V}| = \sqrt{\vec{V} \cdot \vec{V}} = \sqrt{(x^2 + y^2 + z^2)}$$

👉 Cộng hai vectơ:

$$\vec{V}_1 + \vec{V}_2 = (x_1 + x_2, y_1 + y_2, z_1 + z_2)$$



## Đại số véctơ

☞ Nhân hai vectơ:

-Tích vô hướng:  $\vec{n} \cdot \vec{v} = |\vec{n}| \cdot |\vec{v}| \cdot \cos(\vec{n}, \vec{v})$

$$= n_x \cdot v_x + n_y \cdot v_y + n_z \cdot v_z$$

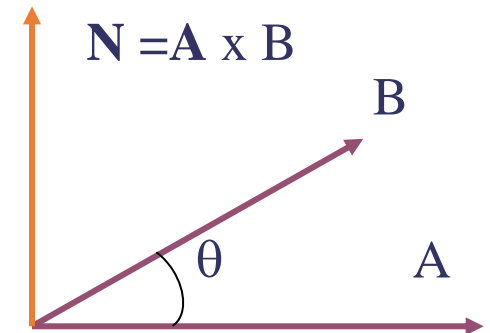
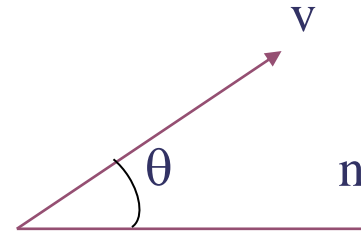
-Tích có hướng:

+ Hệ toạ độ theo quy tắc bàn tay phải

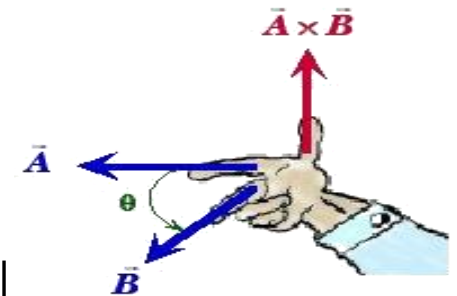
+ Là vector vuông góc với mặt phẳng tạo bởi 2 vector

+ Được xác định như sau:

$$n_x = \begin{vmatrix} a_y & a_z \\ b_y & b_z \end{vmatrix} \quad n_y = -\begin{vmatrix} a_x & a_z \\ b_x & b_z \end{vmatrix} \quad n_z = \begin{vmatrix} a_x & a_y \\ b_x & b_y \end{vmatrix}$$



**Right Hand Rule**

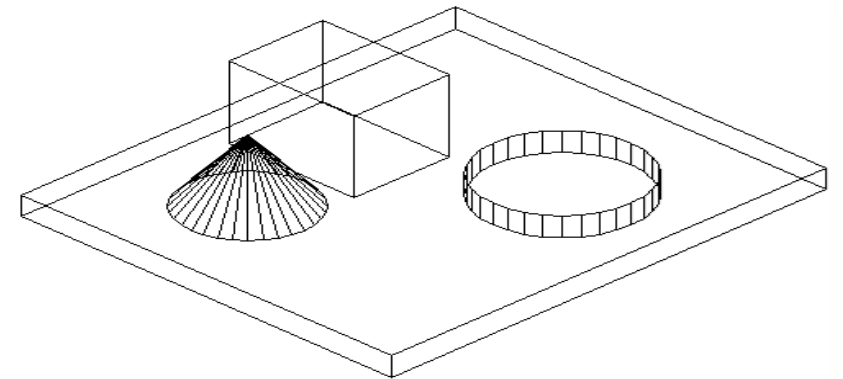




- **Giả sử ta có tập đa giác cần hiển thị trên màn hình**

- Chúng có khoảng cách khác nhau tới điểm quan sát
- Chúng che khuất nhau
- Nếu hiển thị chúng một cách ngẫu nhiên thì hình dáng đối tượng không đúng như mong muốn

⇒ Nhu cầu loại bỏ mặt hay phần mặt bị khuất

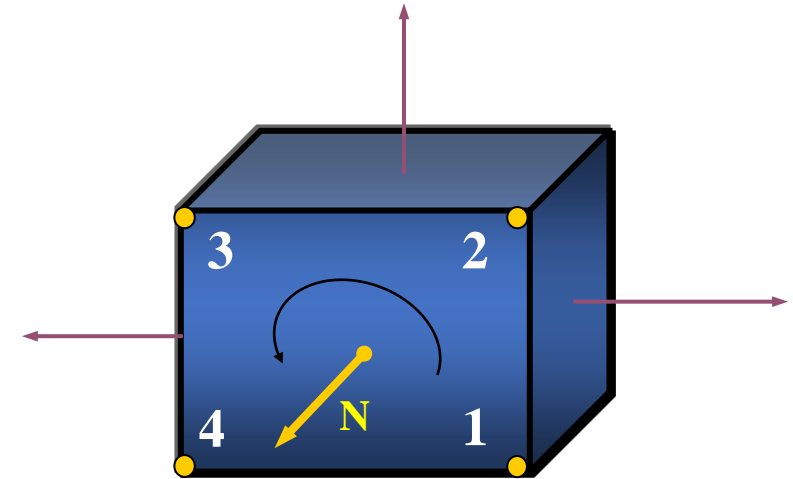


- Hai tiếp cận chính để loại bỏ mặt khuất
  - Tiếp cận không gian đối tượng (*object-space*)
    - Làm việc trực tiếp với đối tượng.
    - So sánh đối tượng hay một phần của đối tượng với đối tượng khác để quyết định đường thẳng hay mặt phẳng nào bị loại bỏ.
  - Tiếp cận không gian ảnh (*image-space*)
    - Làm việc với hình chiếu của vật thể. Xác định khả năng nhìn thấy của từng pixel trên mặt phẳng chiếu.
    - Được áp dụng rộng rãi

## Thuật toán khử mặt sau (Back-face)

### ○ Véc tơ pháp tuyến **N**

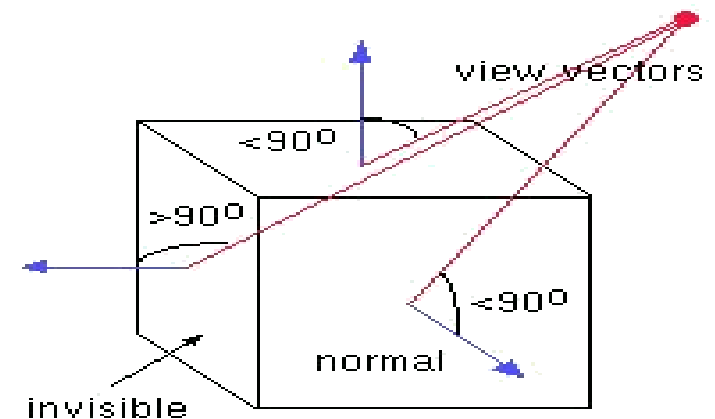
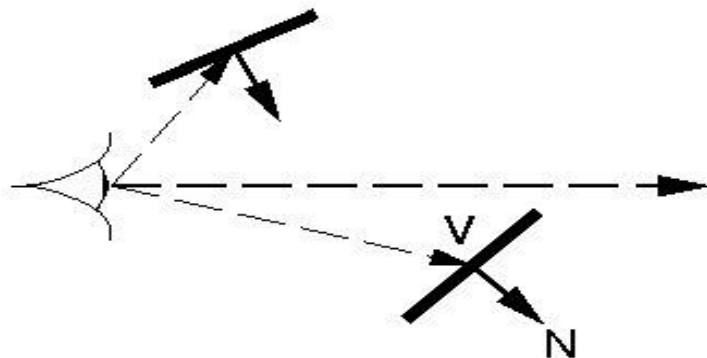
- Véc tơ vuông góc với phía mặt phẳng nhìn thấy là véc tơ pháp tuyến
- Xác định véc tơ pháp tuyến
  - Sắp xếp các điểm xác định mặt phẳng theo quy tắc bàn tay phải
  - Từng cặp điểm liên tiếp hình thành véc tơ trên mặt phẳng (thí dụ **12**, **13**)
  - Véc tơ pháp tuyến **N** là tích có hướng của véc tơ bất kỳ trên mặt phẳng với véc tơ tiếp theo nó (**12** và **13**)



## Thuật toán khử mặt sau (Back-face)

$$\vec{V} \cdot \vec{N} = |\vec{V}| |\vec{N}| \cos \theta$$

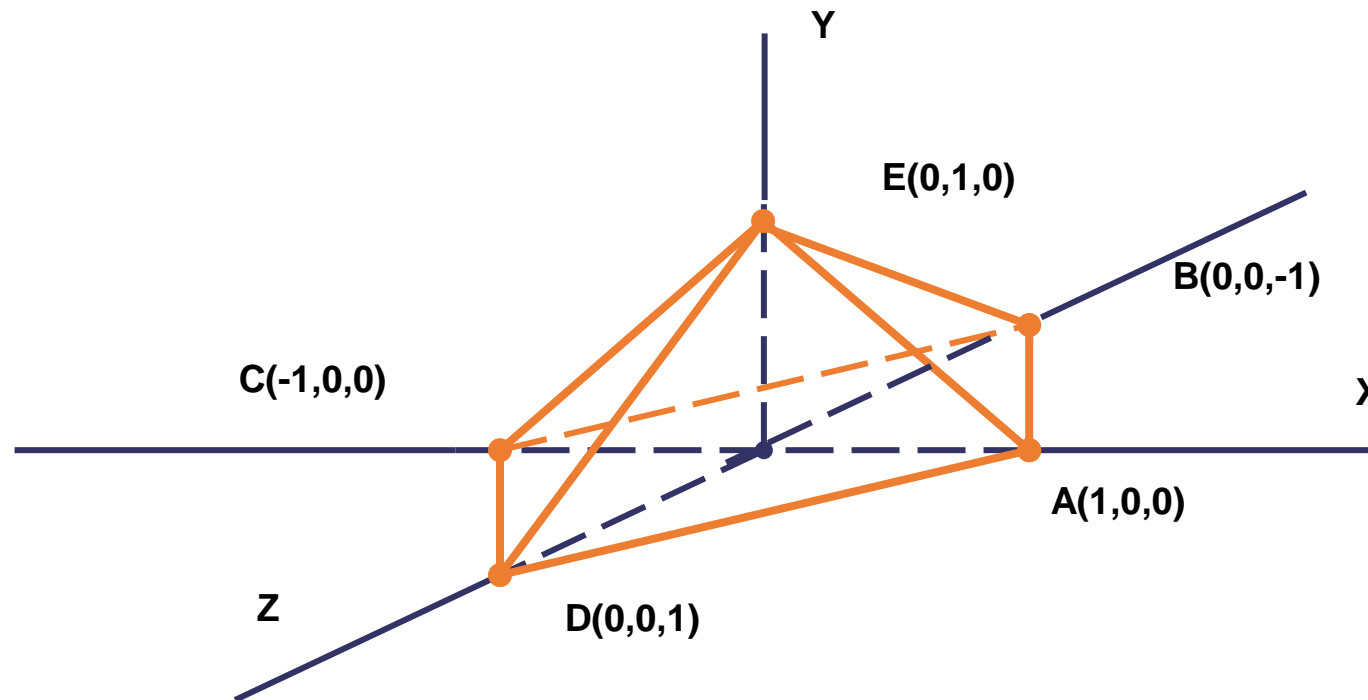
- Véc tơ hướng nhìn  $\mathbf{V}$
- Nếu góc  $\theta$  giữa  $\mathbf{V}$  và  $\mathbf{N}$  trong khoảng  $[-90^\circ, 90^\circ]$  hay  $\cos\theta \geq 0$  thì mặt phẳng nhìn thấy
- Xét dấu  $\cos\theta$  bằng cách kiểm tra dấu véc tơ là kết quả tích vô hướng:  
Nếu  $\mathbf{V} \cdot \mathbf{N} \geq 0$  thì mặt của đối tượng nhìn thấy



## Thuật toán khử mặt sau (Back-face)

Ví dụ:

Tìm các mặt phẳng nhìn thấy được từ điểm  $P(5, 5, 5)$  đến hình chóp



## Thuật toán Deep sorting

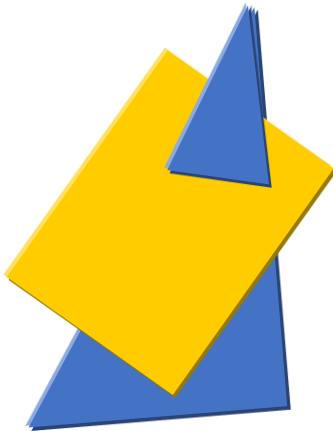
- Phương pháp sắp xếp theo chiều sâu (Depth-Sorting) hay Painter.
- Tác giả: Newell, Sancha (1972). Sử dụng cả thao tác không gian đối tượng và thao tác không gian ảnh.
- Để vẽ tranh dầu
  - Họa sỹ vẽ nền trước
  - Vẽ các đối tượng từ xa trước, cận cảnh vẽ sau
  - Vẽ các đối tượng theo độ sâu

## Thuật toán Deep sorting

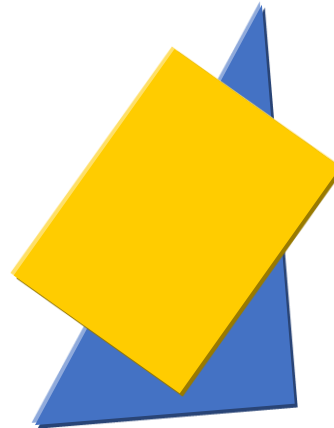
- ☞ Sắp xếp các đa giác theo chiều sâu  $z$  giảm dần
- ☞ Trình tự hiển thị (chuyển đổi đường quét) các đa giác là từ  $z$  lớn nhất đến nhỏ nhất
- ⇒ Vì đa giác gần nhất được hiển thị sau cho nên sẽ ở trên đỉnh (do vậy có thể nhìn thấy)
- ☞ Giải quyết vấn đề nhập nhằng khi  $z$  của các đa giác gối lên nhau (cần bẻ gãy các đa giác)

## Thuật toán Deep sorting

- Thực hiện theo giá trị Z trung bình
- Hạn chế: Không phân biệt chính xác các vùng của các mặt bị phủ lấp lên nhau



Hình ảnh thật



Khi vẽ bằng giải thuật





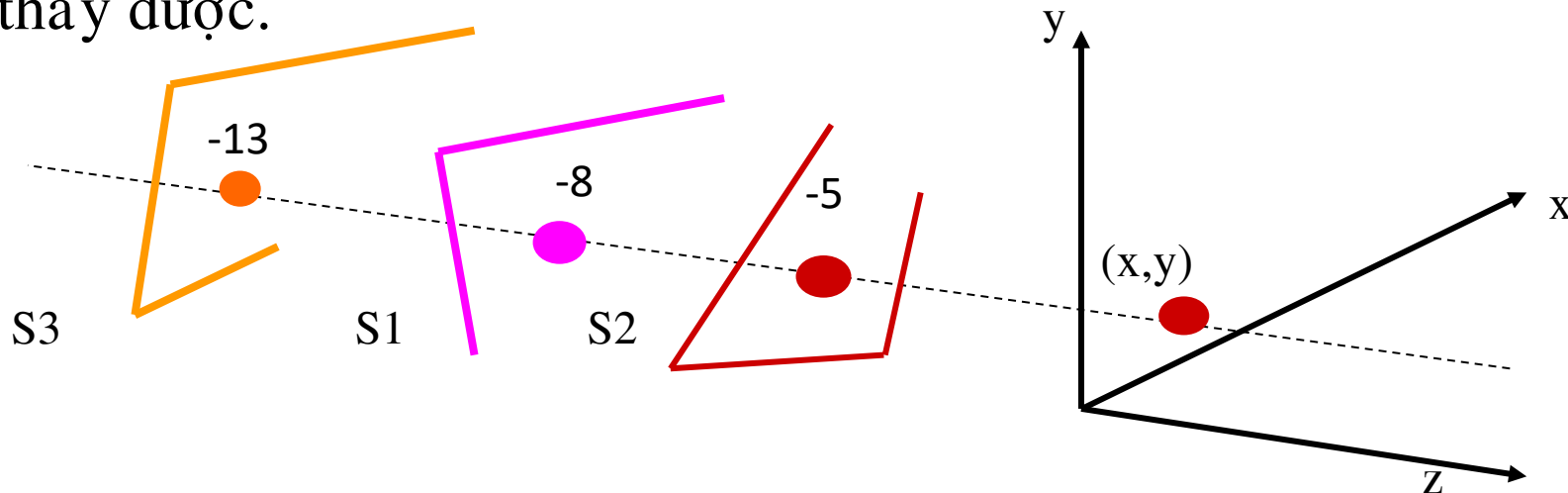
## Thuật toán Deep buffer

- Vùng đệm chiều sâu (**Depth-buffer hay Z-buffer**)
- Tác giả: Ed. Catmull (1974)
- Dễ cài đặt, được sử dụng khá rộng rãi
- ☞ Mỗi điểm  $(x, y, z)$  trên vật thể tương ứng với tọa độ  $(x, y)$  của nó trên mặt quan sát. Giá trị  $z$  tăng theo khoảng cách tới camera.
- ☞ Sử dụng Z-buffer để lưu trữ giá trị  $z$  (hay độ sâu) của mỗi điểm sẽ được vẽ trên màn hình.

## Thuật toán Deep buffer

Dựa trên hướng tiếp cận không gian ảnh

- ☞ So sánh độ sâu của các bề mặt tạo nên đối tượng tại vị trí mỗi pixel trên mặt phẳng chiếu.
- ☞ Với mỗi điểm  $(x,y,z)$  trên bề mặt đa giác tương ứng với điểm  $(x,y)$  nằm trên mặt phẳng chiếu, mặt nào có tọa độ  $Z$  nhỏ nhất xét tại điểm đó sẽ là mặt thấy được.



## Thuật toán Deep buffer

☞ Khởi tạo vùng đệm sâu, vùng đệm khung với mọi vị trí (x,y):

+  $\text{Depth}(x,y) = \text{max}$

+  $\text{Fresh}(x,y) = \text{Background}(x,y)$

☞ Với mỗi mặt của đối tượng, tại mỗi vị trí (x,y) trên mặt đó, so sánh độ sâu Z với giá trị đã lưu trước đó để xác định mặt thấy:

For (mỗi mặt của đối tượng)

For (mỗi dòng của mặt)

For (mỗi cột của mặt ) {

+ Tính độ sâu Z tại mỗi vị trí (x,y) của mặt.

+ If ( $Z < \text{Depth}(x,y)$ ) {

$\text{Depth}(x,y) = Z;$

$\text{Fresh}(x,y) = \text{Surface}(x,y);$

}

}

## Thuật toán Deep buffer

Tính độ sâu Z:

Mặt phẳng được biểu diễn bởi phương trình:  $Ax+By+Cz+D=0$

$\Rightarrow$  độ sâu Z được tính theo công thức:  $z=-(Ax+By+D)/C$

Giá trị x,y biết trước vì quá trình quét sẽ sinh ra chúng. Nếu  $z_i$  là độ sâu tại vị trí (x,y) thì độ sâu  $z_{i+1}$  của điểm tiếp theo (x+1,y) như sau:

$$z_{i+1} = -(A(x+1)+By+D)/C = -(Ax+By+D)/C - A/C = z_i - A/C$$

Tương tự khi y tăng thì giá trị z mới cũng được tính:  $Z_{i+1} = z_i - B/C$

☞ Vậy sau khi các bề mặt được xử lý thì vùng đệm sâu chứa giá trị độ sâu các điểm của các mặt thấy được, còn vùng đệm khung chứa các giá trị màu của bề mặt này.

# CONTENTS

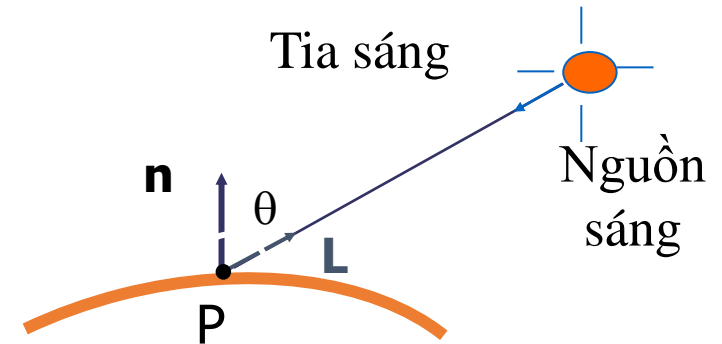
- Khử đường, mặt khuất
  - Thuật toán Back-face
  - Thuật toán Deep buffer
  - Thuật toán Deep sorting
- **Chiếu sáng và tô bóng**
  - Tô bóng hằng Lambert
  - Tô bóng nội suy Gouraud
  - Tô bóng nội suy Phong

- **Các vấn đề liên quan tô bóng (shading)**
  - Mô tả nguồn sáng:
    - vị trí, cường độ sáng
  - Đặc điểm bề mặt tô
  - Khoảng cách giữa mặt tô và nguồn sáng

- Hai loại nguồn sáng
  - Nguồn sáng điểm hay nguồn sáng định hướng
    - ánh sáng từ một điểm chiếu lên vật thể, theo hướng nhất định
  - Nguồn sáng môi trường hay nguồn sáng xung quanh
    - ánh sáng đi đến từ mọi hướng, không quan tâm đến vị trí nguồn sáng
- Nhiệm vụ
  - Tính cường độ ánh sáng trên mỗi điểm của ảnh đối tượng.

## Kỹ thuật tô bóng Lambert

- Gọi:
  - Tia sáng chiếu vào đối tượng tại P
  - Góc giữa tia sáng và vectơ pháp  $\mathbf{n}$  là  $\theta$
  - Vectơ đơn vị của tia phản xạ  $\mathbf{L}$



Áp dụng định luật phản xạ ánh sáng của Lambert:

Bức xạ lý tưởng của tia sáng

$$\mathbf{I} = \mathbf{I}_s k_d \cos \theta$$

$I_s$  - cường độ điểm nguồn

$k_d$  - hệ số phản xạ có giá trị trong khoảng  $[0..1]$



## Kỹ thuật tô bóng Lambert

Bức xạ lý tưởng của tia sáng:  $I = I_s k_d \cos \theta$

$I_s$  - cường độ điểm nguồn

$k_d$  - hệ số phản xạ có giá trị trong khoảng [0..1]

- Khi quan tâm đến

- nguồn sáng môi trường

$$I = I_a k_a + I_s k_d \cos \theta$$

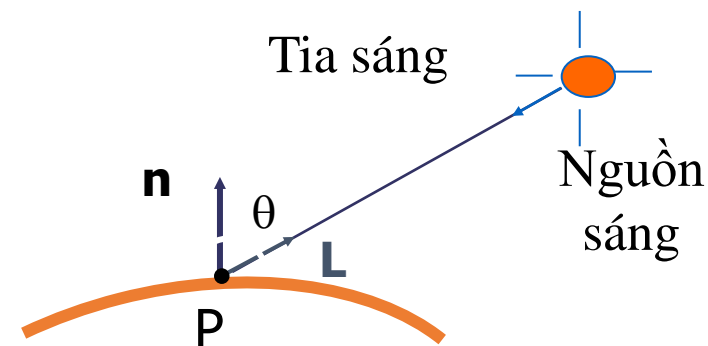
- khoảng cách giữa đối tượng và nguồn sáng

$$I = I_a k_a + \frac{I_s k_d \cos \theta}{d^2}$$

$I_a$  - cường độ

$k_a$  - tham số của ánh sáng môi trường

$D$  - khoảng cách từ nguồn sáng tới vật thể



## Kỹ thuật tô bóng Lambert

- Với nguồn sáng môi trường và nhiều nguồn sáng:

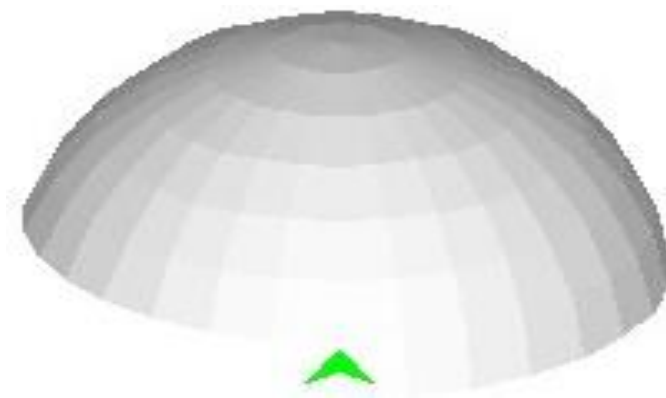
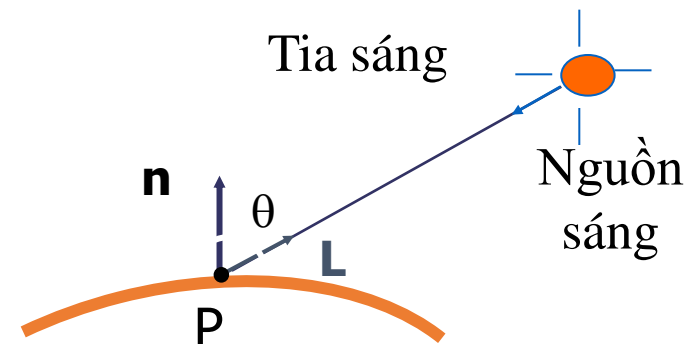
$$I = I_a k_a + \sum_{j=1}^m \frac{I_j k_d \cos \theta_j}{D_j^2} = I_a k_a + \sum_{j=1}^m \frac{I_j k_d (i_j L_j)}{D_j^2}$$

$I_a$  = cường độ của ánh sáng môi trường

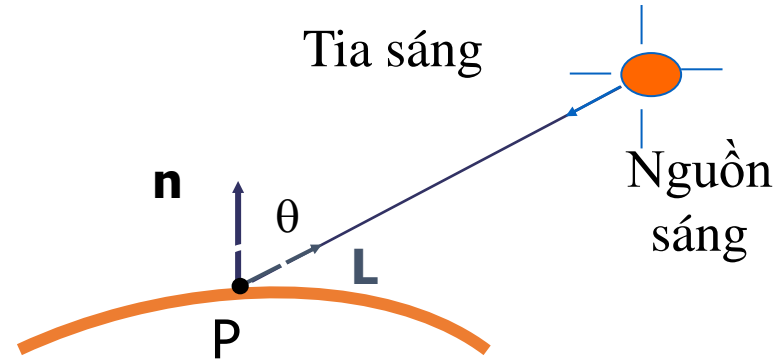
$I_j$  = cường độ của ánh sáng nguồn

- Nhận xét

- Nếu đối tượng cấu tạo bằng các mặt đa giác thì phương pháp này tạo ra các cường độ sáng như nhau cho các điểm trên cùng mặt, do vậy ảnh gồm nhiều ô sáng.
- Giải pháp này có tốc độ nhanh



## Kỹ thuật tô bóng Lambert



Ví dụ:

Cho ba điểm  $A(0,0,1)$ ,  $B(1,0,0)$  và  $C(0,1,0)$  và nguồn sáng có cường độ là 9 đặt tại khoảng cách xa theo hướng:  $(\sqrt{2}, 3, 4)$

Xác định cường độ bức xạ lý tưởng tô bóng với hệ số phản chiếu là 0.25.

## Kỹ thuật tô bóng Gouraud

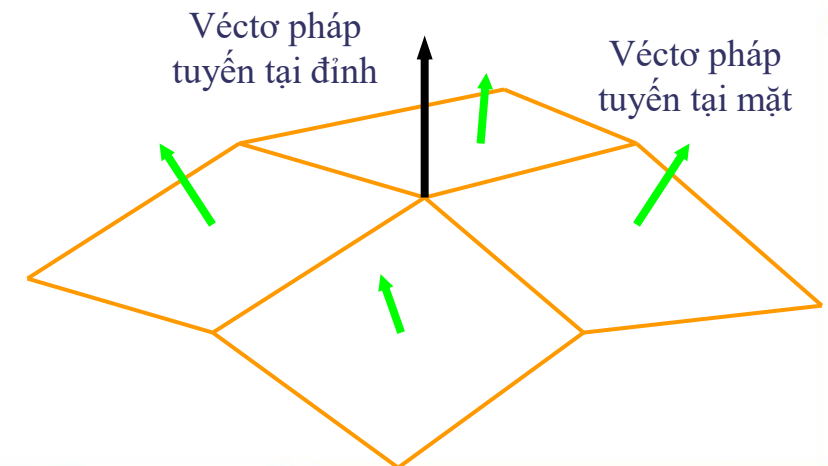
Phương pháp này do Henry Gouraud đưa ra năm 1971.

- Khắc phục nhược điểm của phương pháp Lambert
- Kỹ thuật Gouraud: cường độ sáng được tính toán tại mỗi điểm
  - Cường độ tại các đỉnh được pha trộn và làm mịn trên toàn bề mặt
- Tính cường độ tại các đỉnh chung của nhiều đa giác:
  - lấy trung bình các pháp tuyến của các đa giác có chung đỉnh

$n_i$  là pháp tuyến đơn vị của các mặt chung đỉnh

- Nội suy tuyến tính cường độ cho điểm tô bóng trong đa giác từ cường độ tại đỉnh

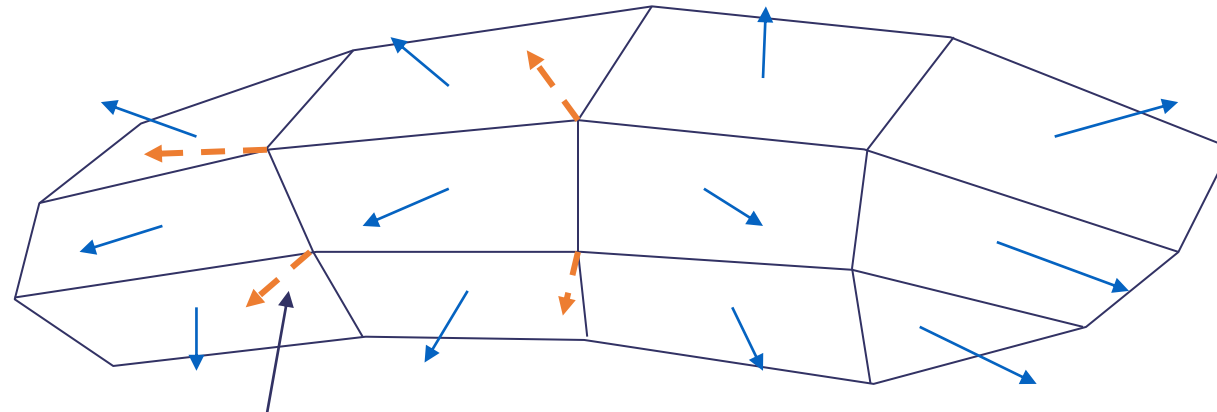
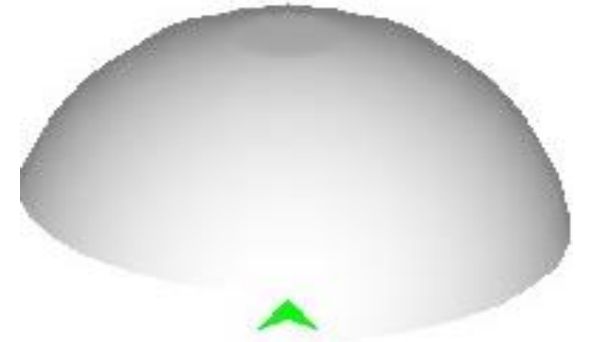
Nhận xét: Hiệu quả, ảnh mịn



## Kỹ thuật tô bóng Gauraud

Gọi P là đỉnh chung của k mặt phẳng.  
Pháp tuyến P được tính như sau:

$$\vec{n}_p = (\vec{n}_1 + \vec{n}_2 + \dots + \vec{n}_k) / k$$



Vector trung bình cộng bằng trung bình cộng của  
các vector pháp tuyến lân cận

## Kỹ thuật tô bóng Gauraud

Phương pháp này nội suy cường độ sáng của các điểm trên biên và sau đó là các điểm trên mặt đang xét.

Gọi  $1(x_1, y_1)$ ,  $2(x_2, y_2)$ ,  $3(x_3, y_3)$  lần lượt là hình chiếu của 3 đỉnh A, B, C tạo nên một mặt của đối tượng. Giả sử cường độ sáng tại các đỉnh A, B, C lần lượt là  $I_1, I_2, I_3$

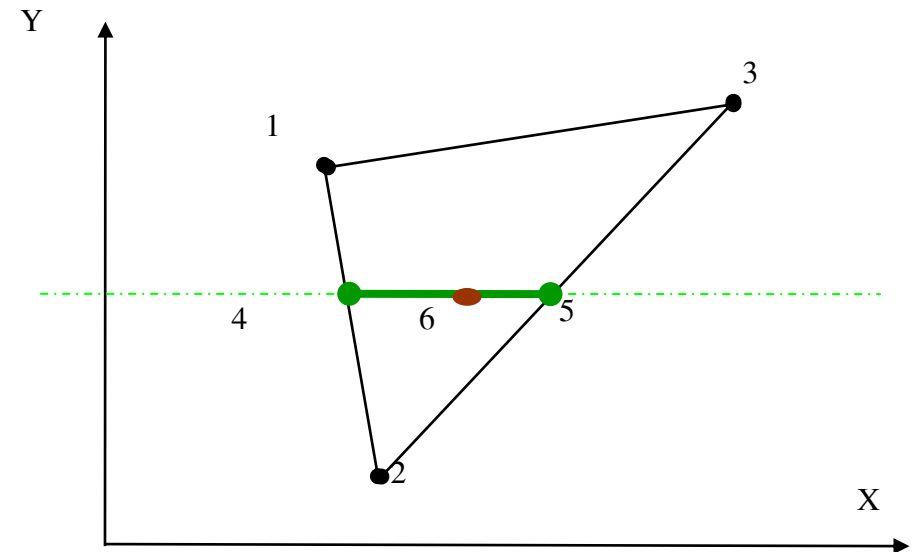
Ta có:

$$I_4 = I_1 \frac{y_4 - y_2}{y_1 - y_2} + I_2 \frac{y_1 - y_4}{y_1 - y_2}$$

Tương tự:

$$I_5 = I_3 \frac{y_5 - y_2}{y_3 - y_2} + I_2 \frac{y_3 - y_5}{y_3 - y_2}$$

$$I_6 = I_4 \frac{x_5 - x_6}{x_5 - x_4} + I_5 \frac{x_6 - x_4}{x_5 - x_4}$$



## Kỹ thuật tô bóng Gauraud

Ví dụ:

Mặt phẳng chữ nhật tạo bởi  $A(0,0)$ ,  $B(1,0)$ ,  $C(1,1)$  và  $D(0,1)$ .

Tính cường độ phản chiếu tại điểm  $P(0.5, 0.5)$  bằng kỹ thuật tô bóng Gauraud. Cường độ trung bình của ánh sáng phản chiếu tại bốn đỉnh là:

$$I_A=8, \quad I_B=9, \quad I_C=2, \quad I_D=4$$

## Kỹ thuật tô bóng Phong

- Tính cường độ ánh sáng tô cho các pháp tuyến vừa nội suy
- Thực hiện nội suy pháp tuyến thay cho nội suy cường độ tô bóng của Gouraud
  - Tính vectơ pháp tuyến tại các đỉnh của lưới đa giác mô phỏng bề mặt
  - Nội suy tuyến tính để tính vectơ pháp tại các điểm mới
  - Tính cường độ ánh sáng tô cho các pháp tuyến vừa nội suy



## Kỹ thuật tô bóng Phong

- Ví dụ: Tính cường độ ánh sáng tô cho điểm M trong tứ giác ABCD

- Giải

- Tính vectơ pháp tuyến tại các đỉnh A, B, C, D
- Tính cường độ ánh sáng tại các đỉnh này
- Nội suy vectơ pháp tuyến tại P và Q

- Nếu P chia AD theo tỷ lệ m ( $AP/PD=m$ ) và Q chia BC theo tỷ lệ n ( $BQ/QC=n$ ) thì vectơ pháp tuyến tại P và Q sẽ là

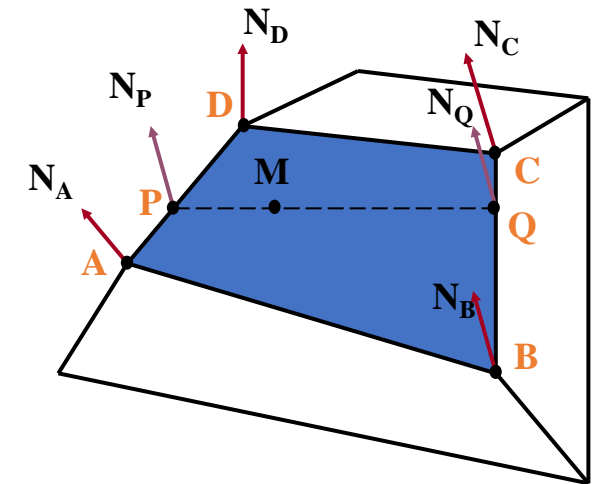
$$\mathbf{N}_P = m \mathbf{N}_A + (1-m) \mathbf{N}_D$$

$$\mathbf{N}_Q = n \mathbf{N}_B + (1-n) \mathbf{N}_C$$

- Nếu M chia PQ với tỷ số x ( $MP/MQ=x$ ) thì:  $\mathbf{N}_M = x \mathbf{N}_P + (1-x) \mathbf{N}_Q$

$$\begin{aligned} \mathbf{N}_M &= x (m \mathbf{N}_A + (1-m) \mathbf{N}_D) + (1-x)(n \mathbf{N}_B + (1-n) \mathbf{N}_C) \\ &= x n \mathbf{N}_A + x(1-m) \mathbf{N}_D + n(1-x) \mathbf{N}_B + (1-x)(1-n) \mathbf{N}_C \end{aligned}$$

- Tính độ sáng tô theo vectơ pháp tuyến  $\mathbf{N}_M$



# Advanced Graphics Techniques



# Computer Graphics



**Thank You...!**