

Đồ họa



Tuần 7

Giảng viên: Trần Đức Minh

Nội dung bài giảng



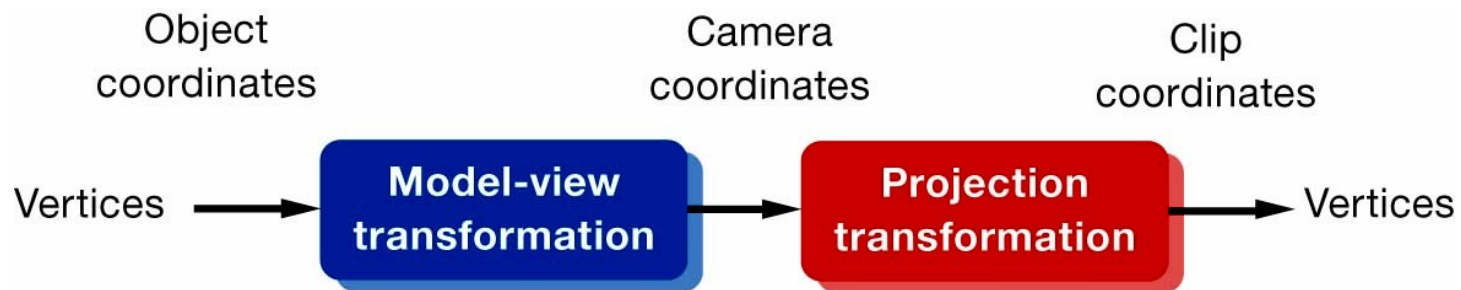
- Hiểu về biến đổi Model-View
- Hiểu về biến đổi phép chiếu
 - Phép chiếu trực giao
 - Phép chiếu phối cảnh
 - Phép chiếu xiên
- Hiểu về phép chia phối cảnh



Hiểu về biến đổi Model-View



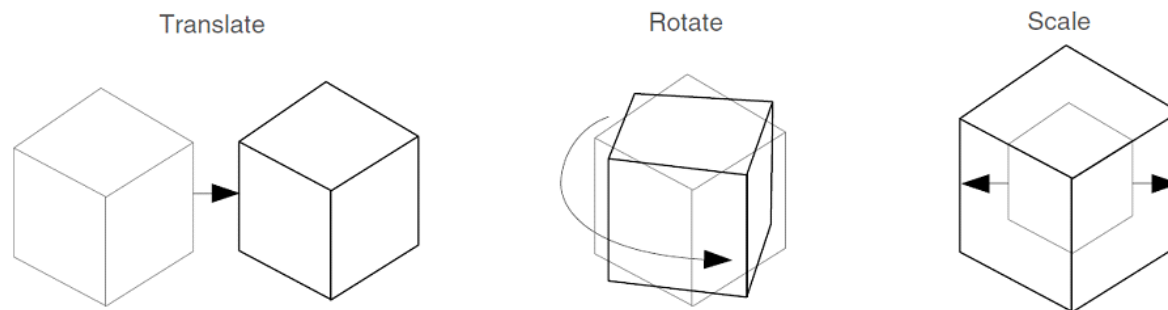
- Biến đổi Model-View là một ánh xạ dùng để chuyển đổi giá trị các đỉnh từ trục tọa độ của người sử dụng sang trục tọa độ camera (mắt).



Hiểu về biến đổi Model-View



- Trục tọa độ người sử dụng
 - Các đối tượng ban đầu được định nghĩa bên trong trục tọa độ của người sử dụng.
 - Định vị các đối tượng rồi biến đổi (xoay, tịnh tiến, tỷ lệ,...) đều được thực hiện ở bên trong trục tọa độ của người sử dụng.
 - Các thao tác biến đổi sử dụng ma trận **biến đổi Model**



Hiểu về biến đổi Model-View



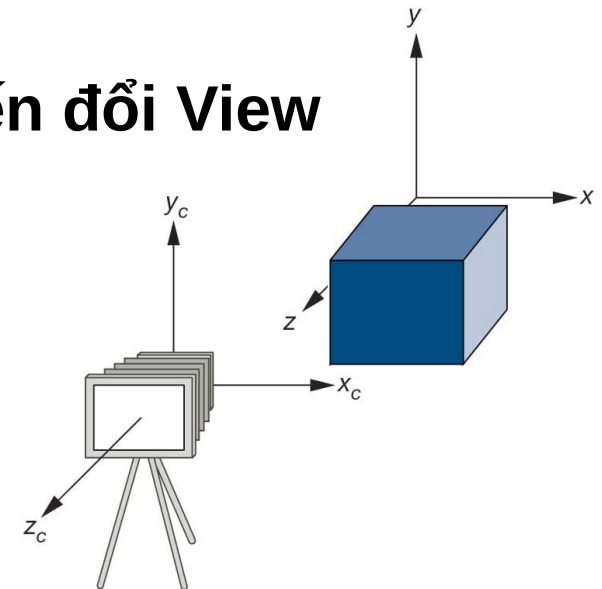
- Trục tọa độ camera
 - Để có góc nhìn trực quan về đối tượng, ta cần mô tả vị trí của đối tượng có liên quan đến camera (điểm nhìn)



Hiểu về biến đổi Model-View



- Trục tọa độ camera
 - Sau khi thiết lập vị trí của camera trên trục tọa độ của người sử dụng, ta cần **chuyển đổi tọa độ của đối tượng trên trục tọa độ người sử dụng sang trục tọa độ của camera** (với gốc của trục tọa độ camera chính là điểm nhìn)
 - Thao tác biến đổi sử dụng ma trận **biến đổi View**



Hiểu về biến đổi Model-View

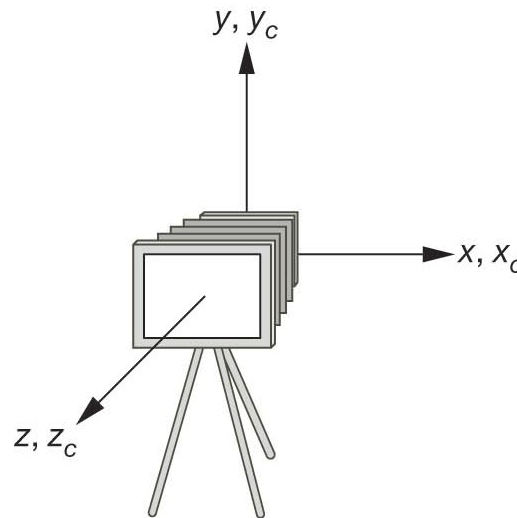


- Giả sử
 - Ma trận M thể hiện cho thao tác biến đổi Model.
 - Ma trận V thể hiện cho thao tác biến đổi View.
- Vậy ta có
 - Ma trận VM (tích của hai ma trận V và M) là ma trận được sử dụng để mỗi khi biến đổi đối tượng ở trục tọa độ người sử dụng, ta sẽ nhận được ngay vị trí mới của đối tượng ở trục tọa độ của camera.
 - Ma trận VM là ma trận **biến đổi Model-View**

Hiểu về biến đổi Model-View



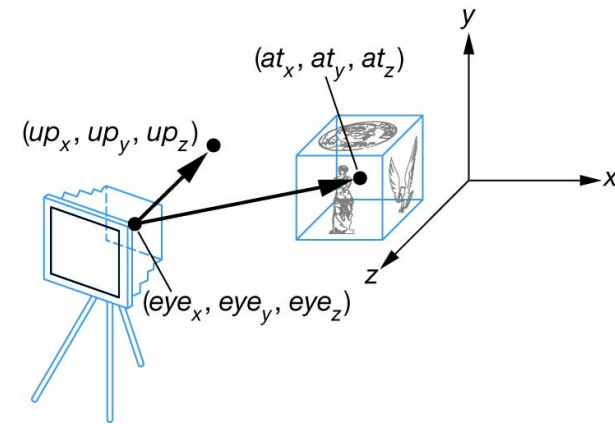
- Chế độ mặc định trong WebGL là trục tọa độ của người sử dụng trùng với trục tọa độ của camera (ma trận V là ma trận đơn vị).
 - Ví dụ: Trong bài biến đổi hình ta có thể nhìn thấy ngay hình ảnh của đối tượng khi biến đổi.



Xây dựng ma trận biến đổi View



- Nhắc lại: Ma trận biến đổi View **chuyển đổi tọa độ của đối tượng trên trục tọa độ người sử dụng sang trục tọa độ của camera**
- Tương tự hàm **lookAt**
- Có 3 tham số cần được thiết lập
 - **eye**: Vị trí đặt điểm nhìn
 - **look at**: tọa độ của một điểm mà véc tơ nối **điểm eye** và **điểm look at** xác định một trục tọa độ của camera.
 - **up**: véc tơ xác định hướng đặt camera



Nhắc lại một số vấn đề trong không gian véc-tơ



- Tích có hướng của hai véc-tơ
 - Ý nghĩa hình học:
 - Tích có hướng của hai véc-tơ là một véc-tơ **vuông góc** với mặt phẳng được tạo bởi hai véc-tơ đầu vào.

- Công thức:

$$[\vec{a} \cdot \vec{b}] = \left(\begin{vmatrix} B_1 & C_1 \\ B_2 & C_2 \end{vmatrix}, \begin{vmatrix} C_1 & A_1 \\ C_2 & A_2 \end{vmatrix}, \begin{vmatrix} A_1 & B_1 \\ A_2 & B_2 \end{vmatrix} \right)$$

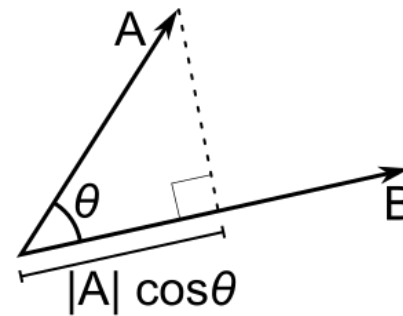
- Với $\vec{a} = (A_1, B_1, C_1)$

$$\vec{b} = (A_2, B_2, C_2)$$

Nhắc lại một số vấn đề trong không gian véc-tơ



- Tích vô hướng của hai véc tơ
 - Ý nghĩa hình học:
 - Tích vô hướng của hai véc tơ là chiếu một véc tơ lên véc tơ còn lại, sau đó lấy độ lớn của hình chiếu rồi nhân với độ lớn của véc-tơ còn lại.



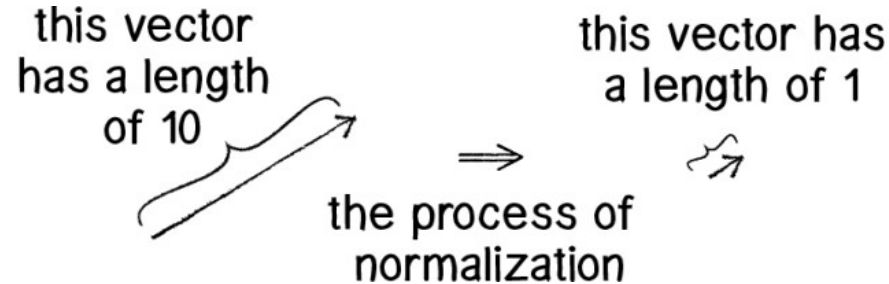
- Công thức:

$$\vec{a} \cdot \vec{b} = A_1 B_1 + A_2 B_2 + A_3 B_3$$

Nhắc lại một số vấn đề trong không gian véc-tơ



- Chuẩn hóa véc-tơ: Thay đổi độ lớn của véc-tơ về 1 (véc-tơ đơn vị)
 - Ta chuẩn hóa véc-tơ khi chỉ quan tâm đến hướng mà không quan tâm đến độ lớn của nó.



- Công thức tính véc tơ đơn vị:

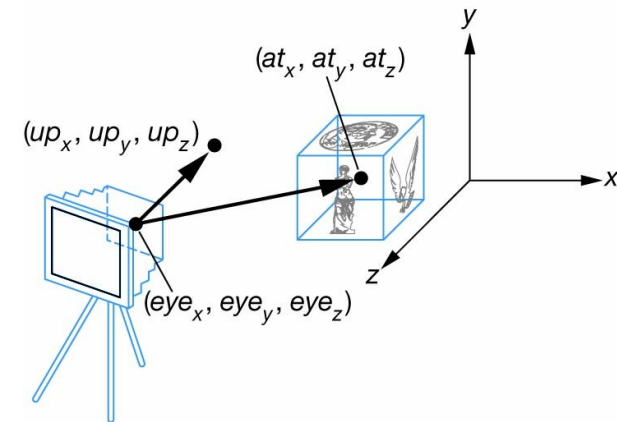
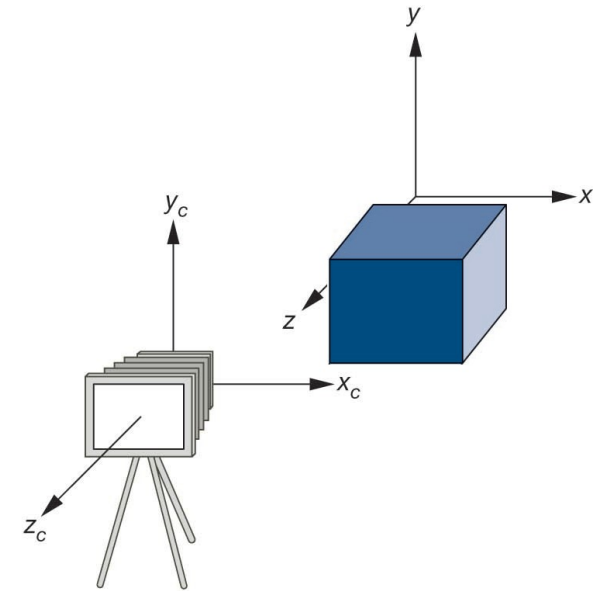
$$\hat{u} = \frac{\vec{u}}{\|\vec{u}\|}$$

← Độ lớn của véc-tơ

Xây dựng ma trận biến đổi View



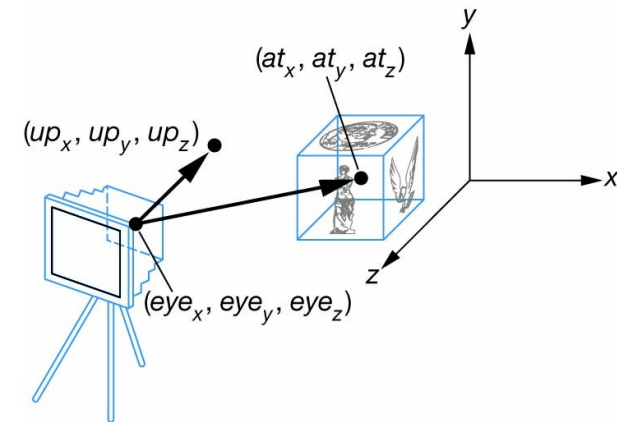
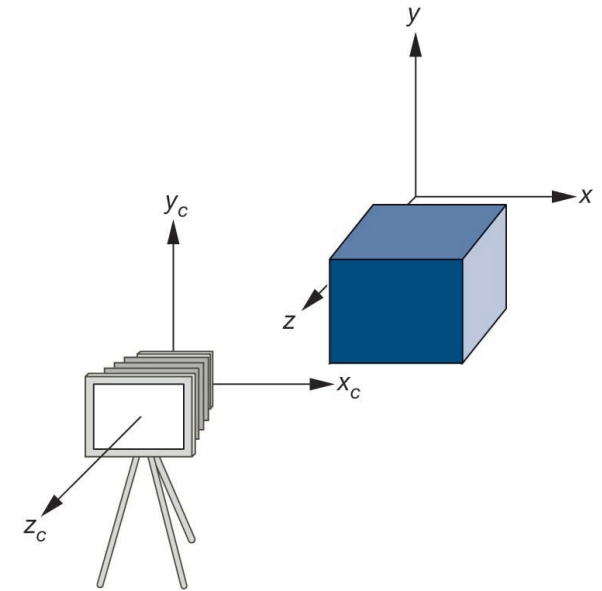
- Bước 1: Xác định các véc tơ đơn vị của trục tọa độ camera.
 - Trục tọa độ z_c có hướng dương ngược hướng với hướng nhìn của camera, do đó
 - $z_c = \text{eye} - \text{look at}$
 - Trục tọa độ x_c có hướng vuông góc với mặt phẳng tạo bởi véc tơ up và trục z_c , do đó
 - $x_c = [up \bullet z_c]$



Xây dựng ma trận biến đổi View



- Bước 1: Xác định các véc tơ đơn vị của trục tọa độ camera.
 - Trục tọa độ y_c có hướng vuông góc với mặt phẳng tạo bởi trục x_c và trục z_c , do đó
 - $y_c = [x_c \bullet z_c]$
 - Vậy x_c, y_c, z_c là 3 véc tơ vuông góc với nhau từng đôi một.
 - Chuẩn hóa 3 véc tơ x_c, y_c, z_c thành véc tơ đơn vị.



Xây dựng ma trận biến đổi View



- Bước 2: Ma trận biến đổi View được xác định như sau:

$$V = \begin{pmatrix} x_c^x & x_c^y & x_c^z & d_x \\ y_c^x & y_c^y & y_c^z & d_y \\ z_c^x & z_c^y & z_c^z & d_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- Trong đó:

$$d_x = -eye \bullet x_c$$

$$d_y = -eye \bullet y_c$$

$$d_z = -eye \bullet z_c$$

- Nhận xét:

- $V \text{ eye} = (0, 0, 0, 1)^T$

- $V x_c = (1, 0, 0, 0)^T$

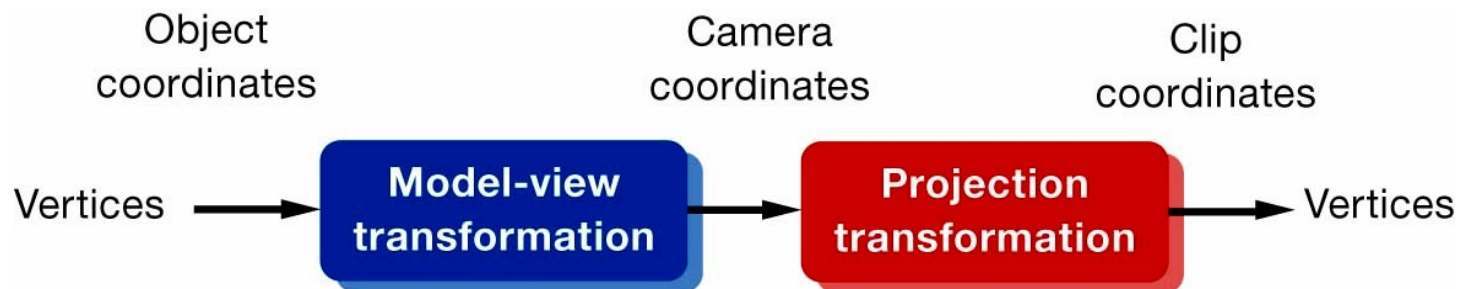
- $V y_c = (0, 1, 0, 0)^T$

- $V z_c = (0, 0, 1, 0)^T$

Hiểu về biến đổi phép chiếu



- **Biến đổi phép chiếu** được sử dụng để **chuẩn hóa** việc chiếu đối tượng lên mặt phẳng chiếu.
- Biến đổi phép chiếu **được thực hiện sau** khi chuyển đổi giá trị các đỉnh của đối tượng từ trục tọa độ của người sử dụng sang trục tọa độ camera (biến đổi Model-View).



Hiểu về biến đổi phép chiếu



- Chiếu trực giao

- Giả sử ta có đối tượng với các đỉnh được **xác định theo trục tọa độ camera** (sau khi nhân với ma trận Model-View)
- Hình chiếu trực giao các đỉnh của đối tượng lên mặt phẳng chiếu được xác định như sau:

$$x_p = x$$

$$y_p = y$$

$$z_p = 0$$

$$w_p = 1$$

- $q_p = M_{orth} p$

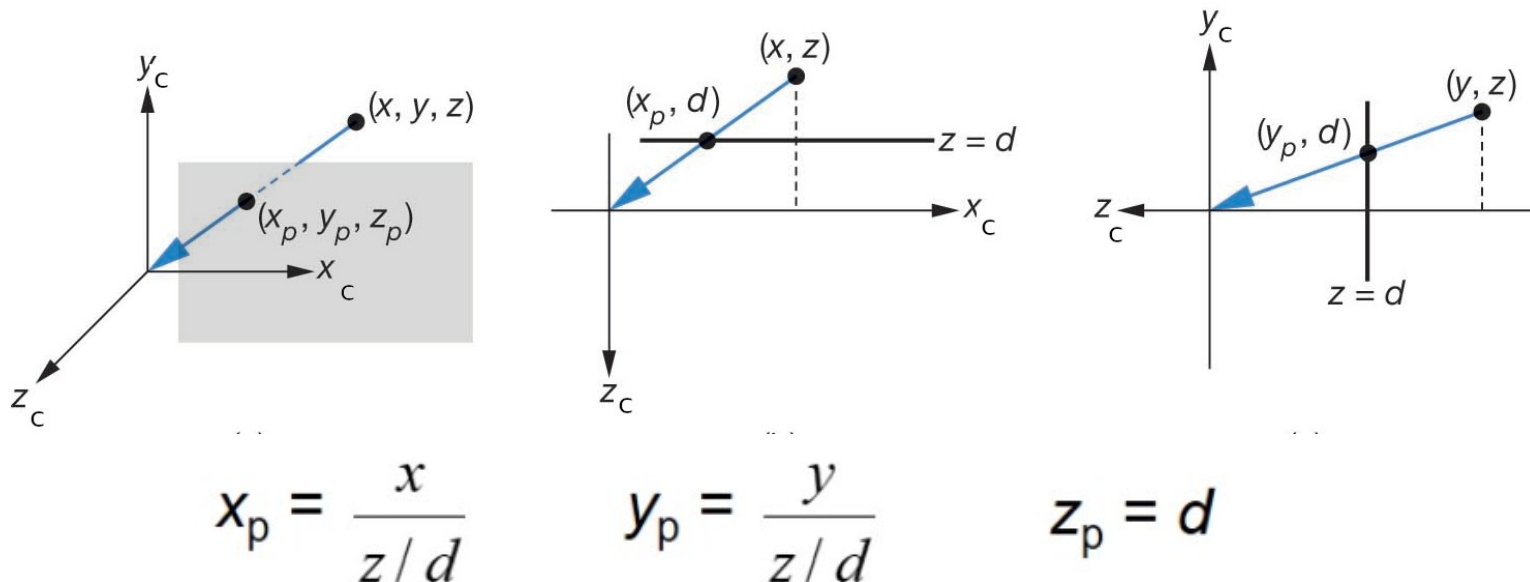
$$M_{orth} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Hiểu về biến đổi phép chiếu



- Chiếu phối cảnh

- Giả sử ta có đối tượng với các đỉnh được xác định theo trục tọa độ camera.
- Hình chiếu phối cảnh các đỉnh của đối tượng lên **mặt phẳng chiếu ($z = d$)** được xác định như sau:



Hiểu về biến đổi phép chiếu



- Chiếu phối cảnh
 - Ta sử dụng ma trận M để tìm hình chiếu phối cảnh
 - $q_p = M p$

$$M = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/d & 0 \end{pmatrix} \quad p = \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \quad \Rightarrow \quad q = \begin{pmatrix} x \\ y \\ z \\ z/d \end{pmatrix}$$

Hiểu về biến đổi phép chiếu



- Chiếu phối cảnh

- Để đưa về hệ trục tọa độ đồng nhất ta cần phải chia các thành phần trong ma trận cho $w \neq 1$.
- Lúc này ta có một **phép chia phối cảnh** (được tự động thực hiện bên trong đường ống đồ họa)

$$q = \begin{pmatrix} \frac{x}{z/d} \\ \frac{y}{z/d} \\ d \\ 1 \end{pmatrix}$$

$$x_p = \frac{x}{z/d}$$

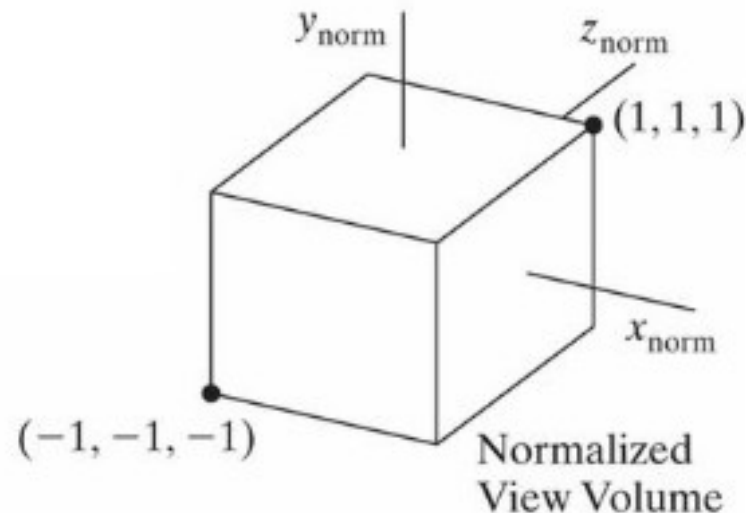
$$y_p = \frac{y}{z/d}$$

$$z_p = d$$

Hiểu về biến đổi phép chiếu



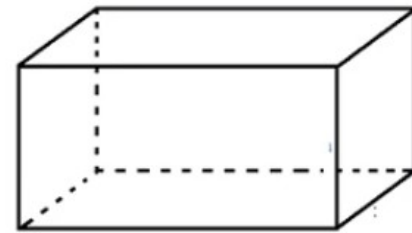
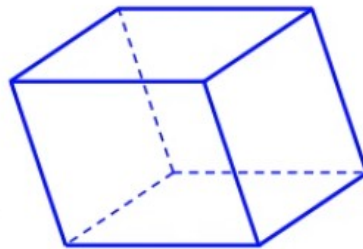
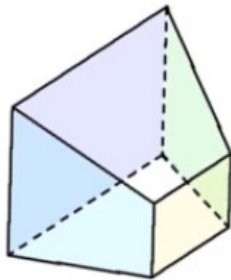
- **Thể tích nhìn được chuẩn hóa** (normalized view volume) là một hình lập phương có **trọng tâm trùng với gốc tọa độ** và **cạnh có chiều dài bằng 2**.



Hiểu về biến đổi phép chiếu



- Với mỗi phép chiếu ta có 3 dạng thể tích nhìn (view volume) khác nhau



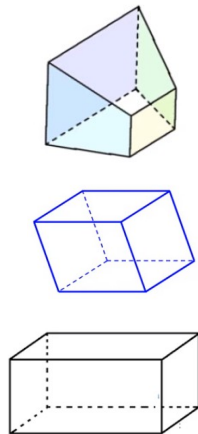
- **Chuẩn hóa thể tích nhìn** là biến đổi tất cả các thể tích nhìn về thể tích nhìn chuẩn.
- **Chuẩn hóa phép chiếu** là biến đổi tất cả các phép chiếu về phép chiếu trực giao.

Hiểu về biến đổi phép chiếu

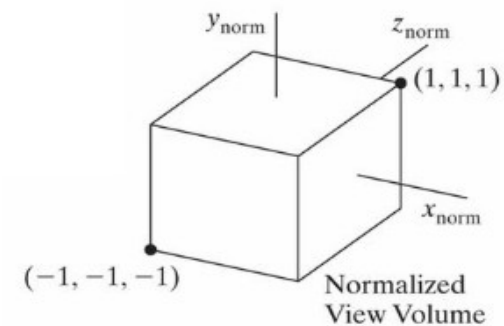


- **Ma trận P**

- Ta cần xây dựng ma trận P để chuẩn hóa cả phép chiếu lẫn thể tích nhìn. Tức là:
 - Với các đối tượng nằm trong một thể tích nhìn nào đó, sau khi nhân với ma trận P thì đối tượng được biến đổi vào trong một thể tích nhìn đã được chuẩn hóa với phép chiếu trực giao.



$$x P =$$

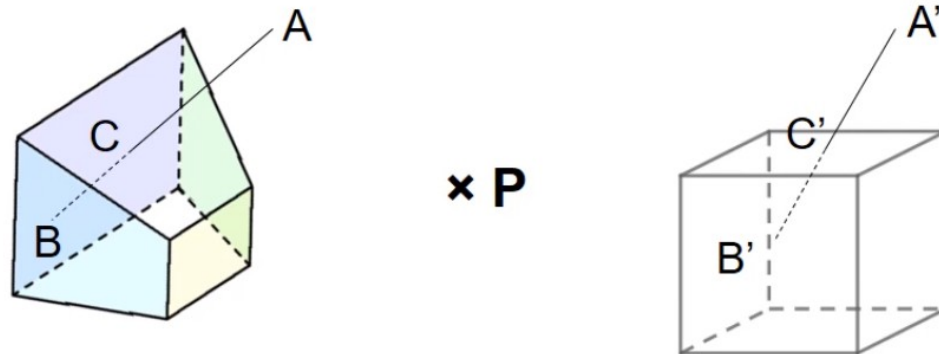


Hiểu về biến đổi phép chiếu



- **Ma trận P**

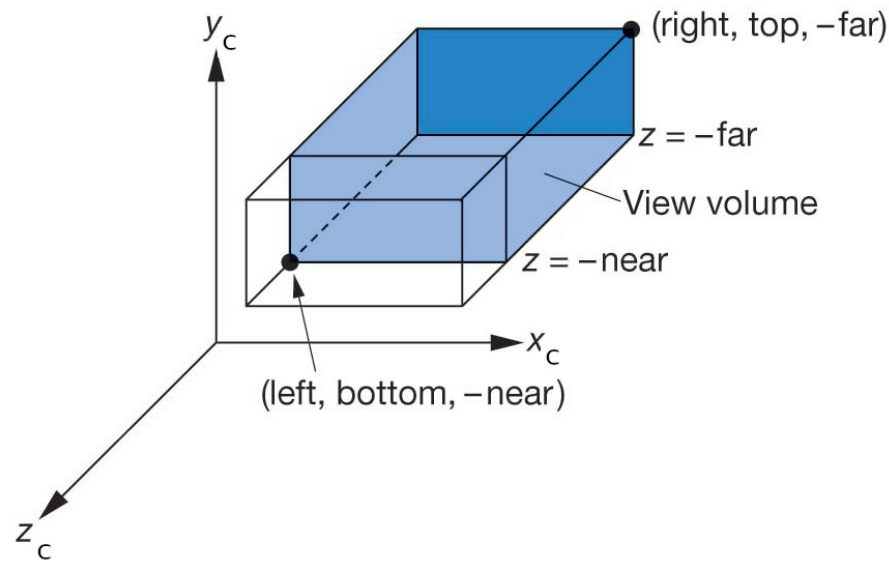
- Ma trận P giúp cho việc cắt xén đối tượng nằm ngoài thể tích nhìn được dễ dàng hơn.
 - Nó đảm bảo một điểm nằm trong thể tích nhìn gốc thì sẽ nhìn thấy được trong thể tích nhìn mới sau khi chuẩn hóa ở vị trí đúng như tỉ lệ.



Phép chiếu trực giao



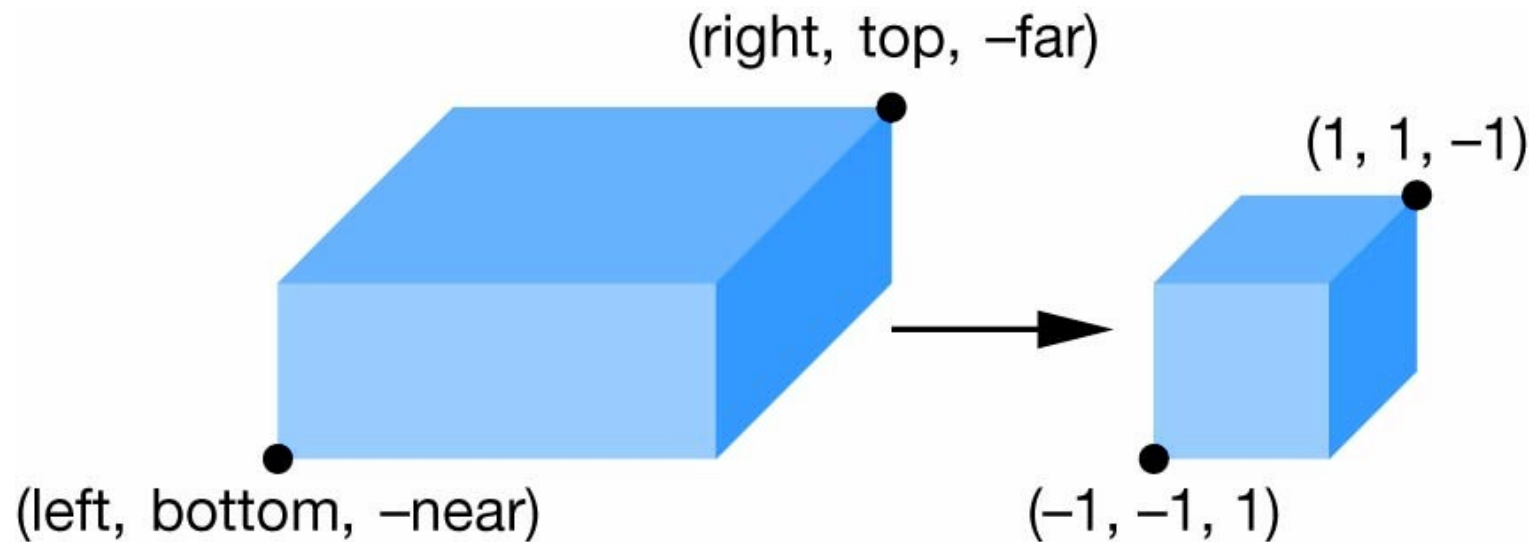
- Tương tự **hàm ortho**
- Các tham số cần thiết lập đều dựa trên trục tọa độ camera.



Phép chiếu trực giao



- Cần tìm ma trận **P** để chuẩn hóa thể tích nhìn.



$$0 < \text{near} < \text{far}$$

Phép chiếu trực giao



- Tìm ma trận P
 - **Bước 1:** Sử dụng ma trận T để tịnh tiến thể tích nhìn sao cho tâm của thể tích nhìn trùng với gốc tọa độ

$$T = \begin{pmatrix} 1 & 0 & 0 & -\frac{\text{right} + \text{left}}{2} \\ 0 & 1 & 0 & -\frac{\text{top} + \text{bottom}}{2} \\ 0 & 0 & 1 & -\frac{\text{far} + \text{near}}{2} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Phép chiếu trục giao



- Tìm ma trận P
 - **Bước 2:** Sử dụng ma trận S để biến đổi tỷ lệ thể tích nhìn sao cho mỗi cạnh của thể tích nhìn có độ dài bằng 2.

$$S = \begin{pmatrix} \frac{2}{\text{right} - \text{left}} & 0 & 0 & 0 \\ 0 & \frac{2}{\text{top} - \text{bottom}} & 0 & 0 \\ 0 & 0 & \frac{2}{\text{near} - \text{far}} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Phép chiếu trục giao



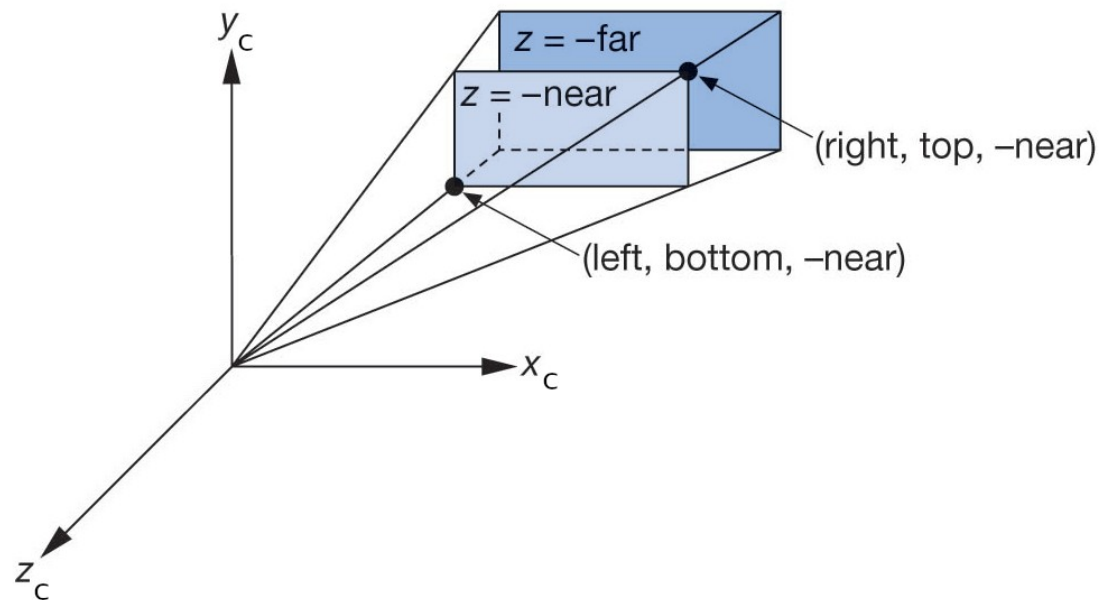
- Tìm ma trận P
 - Ma trận $P = ST$

$$P = ST = \begin{pmatrix} \frac{2}{\text{right} - \text{left}} & 0 & 0 & -\frac{\text{left} + \text{right}}{\text{right} - \text{left}} \\ 0 & \frac{2}{\text{top} - \text{bottom}} & 0 & -\frac{\text{top} + \text{bottom}}{\text{top} - \text{bottom}} \\ 0 & 0 & -\frac{2}{\text{far} - \text{near}} & -\frac{\text{far} + \text{near}}{\text{far} - \text{near}} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Phép chiếu phối cảnh chóp cụt (frustum)



- Các tham số cần thiết lập đều dựa trên trục tọa độ camera.
 - Phương pháp này thường khó lấy được góc nhìn như mong muốn.



Phép chiếu phối cảnh chóp cụt (frustum)



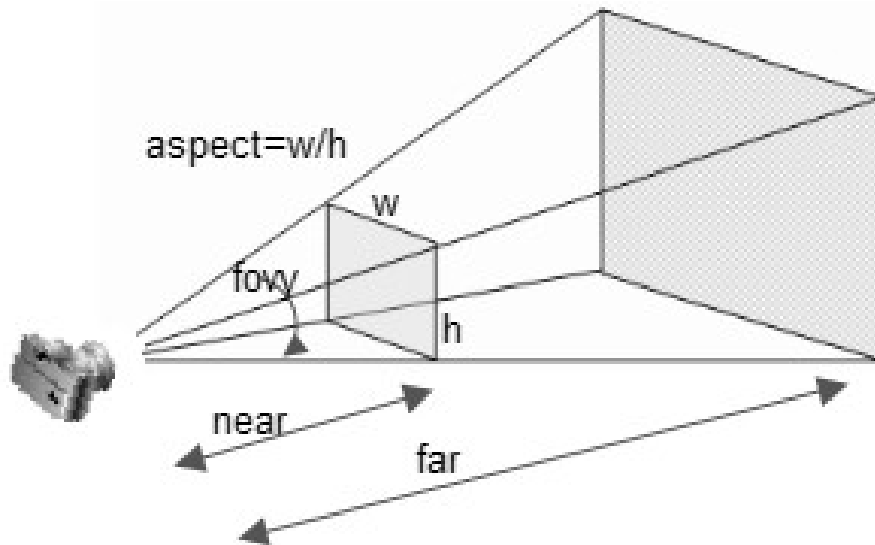
- Ma trận P

$$P = \begin{pmatrix} \frac{2 * \text{near}}{\text{right} - \text{left}} & 0 & \frac{\text{right} + \text{left}}{\text{right} - \text{left}} & 0 \\ 0 & \frac{2 * \text{near}}{\text{top} - \text{bottom}} & \frac{\text{top} + \text{bottom}}{\text{top} - \text{bottom}} & 0 \\ 0 & 0 & -\frac{\text{far} + \text{near}}{\text{far} - \text{near}} & \frac{-2 * \text{far} * \text{near}}{\text{far} - \text{near}} \\ 0 & 0 & -1 & 0 \end{pmatrix}$$

Phép chiếu phối cảnh đối xứng



- Tương tự **hàm perspective**
- Các tham số near, far đều dựa trên trục tọa độ camera.



Phép chiếu phối cảnh đối xứng



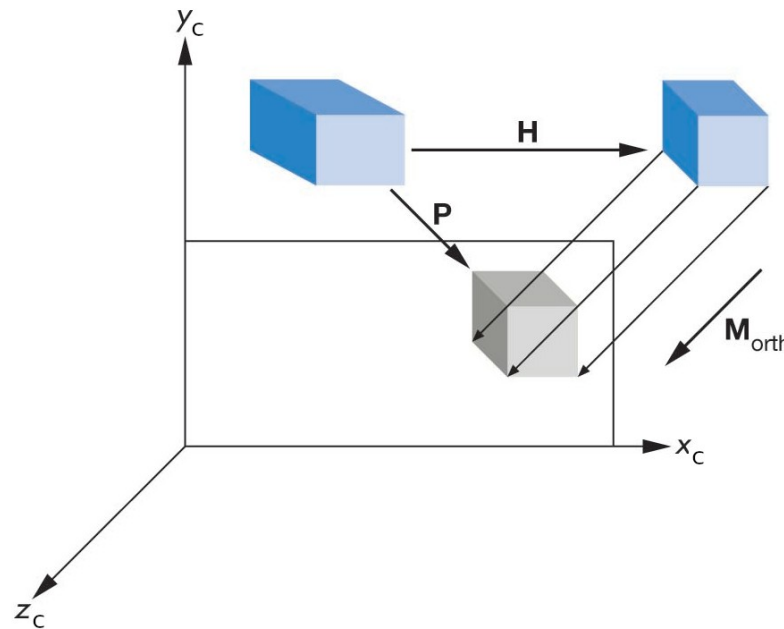
- Ma trận P
 - $\text{top} = \text{near} * \tan(\text{fovy})$
 - $\text{right} = \text{top} * \text{aspect}$

$$P = \begin{pmatrix} \frac{\text{near}}{\text{right}} & 0 & 0 & 0 \\ 0 & \frac{\text{near}}{\text{top}} & 0 & 0 \\ 0 & 0 & \frac{-(\text{far} + \text{near})}{\text{far} - \text{near}} & \frac{-2 * \text{far} * \text{near}}{\text{far} - \text{near}} \\ 0 & 0 & -1 & 0 \end{pmatrix}$$

Phép chiếu xiên



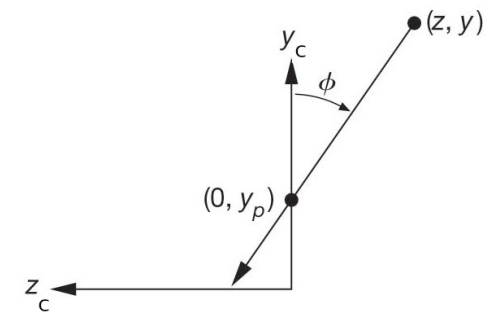
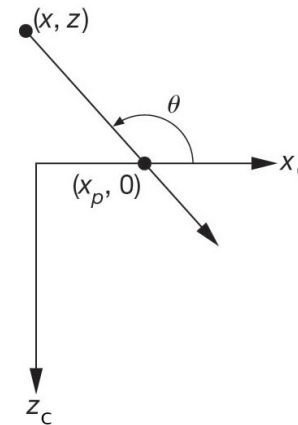
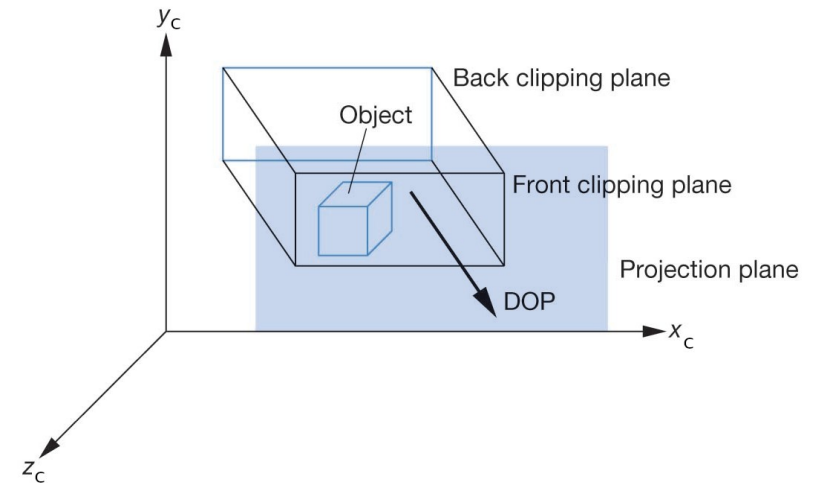
- Phép chiếu xiên = phép trượt + chiếu trực giao
 - Phép trượt dùng để biến đổi đối tượng về hình dạng của thể tích nhìn chuẩn.



Phép chiếu xiên



- Khi thực hiện phép chiếu xiên ta cần biết 2 góc θ (nhìn từ trục y_c xuống) và ϕ (nhìn từ trục x_c vào).
- **DOP**: Hướng chiếu



Phép chiếu xiên



- Tìm ma trận P
 - **Bước 1:** Tính ma trận trượt $H(\theta, \phi)$

$$H(\theta, \phi) = \begin{pmatrix} 1 & 0 & \cot(\theta) & 0 \\ 0 & 0 & \cot(\phi) & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Phép chiếu xiên



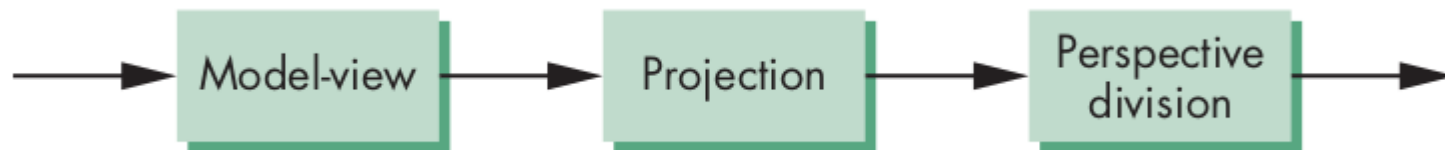
- Tìm ma trận P
 - **Bước 2:** Tính $P = M_{orth} H(\theta, \phi)$
 - Ma trận M_{orth} dùng để chiếu trực giao

$$M_{orth} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Hiểu về phép chia phối cảnh



- **Phép chia phối cảnh** được thực hiện tự động bởi hệ thống sau khi thực hiện **biến đổi Model-View** và **biến đổi phép chiếu**.



$$x_p = \frac{x}{z/d}$$

$$y_p = \frac{y}{z/d}$$

$$z_p = d$$

Hết Tuần 7



Cảm ơn các bạn đã chú ý lắng nghe !!!