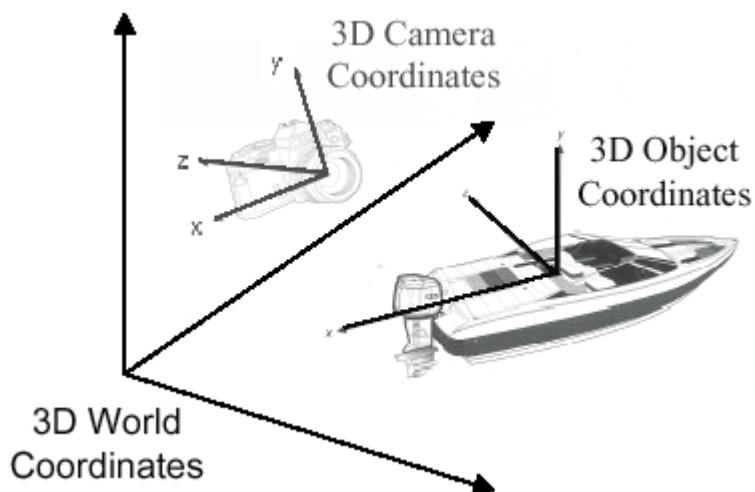


VIEWING TRANSFORMATIONS

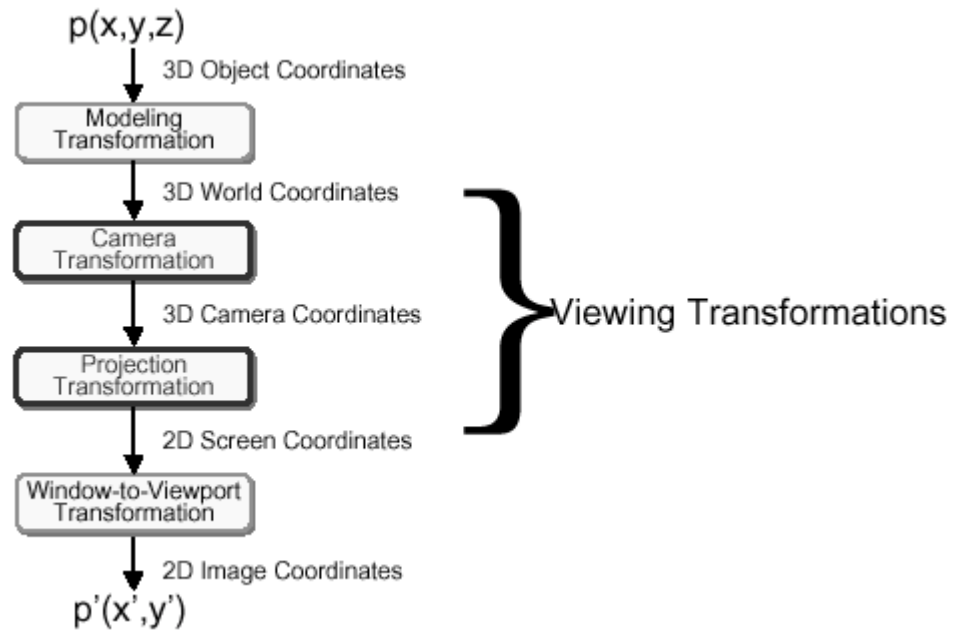


Đầu nhập

- Sau công đoạn modeling transformation, tất cả các đối tượng đều nằm trong cùng một hệ tọa độ chung (world coordinates).
- Sau công đoạn trivial rejection và illumination, chúng ta sẽ xem xét công đoạn biến đổi vào không gian quan sát (view transformation). Mục đích của công đoạn này là chuyển đổi các đối tượng từ hệ tọa độ chung vào hệ tọa độ quan sát (eye coordinates hay 3D camera coordinates)

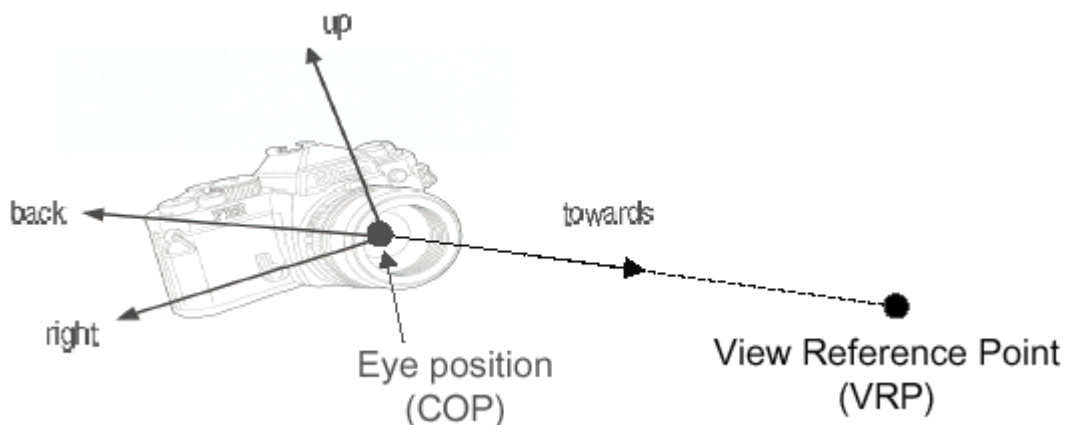


Quy trình hiển thị



Camera

- Các tham số của Camera
 - ◆ Vị trí mắt nhìn (x, y, z)
 - ◆ Hướng nhìn (towards vector, up vector)
 - ◆ Vùng quan sát



Camera Transformation



- Trong cảnh trên, góc toạ ñĩa của world space ñặt ngay döõ ñĩa gheá trục z höông lên ñĩ qua tâm của bình traø. Ñĩa thuận tieãn, trục x vaø y ñöôc chọn song song với các böc töông (chui yù các vieãn gaich trên nền nhaø). Với hệ toạ ñĩa này, gheá vaø bình traø ñể ñể dạng biểu ñiển.
- Böôùc tieáp theo, ta cần mô tả ảnh của mô hình ta ñang mong muốn ñiển tại Công việc này sẽ ñể dạng hơn nhiều nếu góc toạ ñĩa trung với vị trí quan sát (vị trí của mắt hay camera). (Xem hình bên döõ ñĩa)



- Ta có thể nghĩ tới những phép biến đổi tịnh tiến và quay (*rigid body transformations*). Trước tiên, ta cần thực hiện phép quay để cho 2 trục tọa độ (world và camera) cùng phương.



- Sau đó ta thực hiện phép tịnh tiến để đưa gốc tọa độ của world space về trùng với gốc tọa độ của eye space.



- Tại sao ta lại quay trước rồi mới tịnh tiến? Ta có thể thực hiện theo một cách khác không?
- Cách tiếp cận với trình bày không những trực quan và sẽ gây không ít khó khăn khi ta muốn giao tiếp với người dùng trong một hệ xử lý đồ họa 3 chiều. Ta thường tiếp cận theo một cách khác.

- Thay cho việc xác định một hệ tọa độ quan sát mong muốn bằng 1 phép quay và 1 phép tịnh tiến hệ tọa độ gốc ta có thể sử dụng phương pháp sau:

New Camera Transformation

- Trước tiên, ta xác định vị trí mắt camera (hoặc vị trí quan sát) trong không gian gốc. Ta gọi nó là vị trí mắt (*eye point*). Sau đó ta xác định một vị trí trong cảnh (scene) mà ta muốn nó sẽ xuất hiện ở trung tâm của cửa sổ nhìn. Ta gọi điểm này là điểm nhìn (*look-at point*). Tiếp theo ta xác định 1 vector dùng để chỉ hướng đi lên của ảnh tính từ **look-at point**. Ta gọi nó là vector hướng lên (*up-vector*).



- Cách biểu diễn trên rất tối giản. Ta có thể sử dụng cách biểu diễn này để mô tả một quá trình của camera bằng cách thay đổi vị trí eye-point, look-at point và up-vector không đổi. Hoặc ta có thể quét camera từ một vị trí này đến một vị trí khác trên ảnh bằng cách thay đổi look-at point.

- Bây giờ chúng ta sẽ xem xét, với mô tả trên, ta sẽ xây dựng một phép biến đổi từ hệ tọa độ thế sang hệ tọa độ quan sát nhờ thế này.
- Trước tiên, ta sẽ xác định phần quay của camera transformation (V).
- Ta có thể xác định vector I có phương trùng với tia nhìn theo công thức:

$$\begin{bmatrix} I_x \\ I_y \\ I_z \end{bmatrix} = \begin{bmatrix} \text{lookat}_x \\ \text{lookat}_y \\ \text{lookat}_z \end{bmatrix} - \begin{bmatrix} \text{eye}_x \\ \text{eye}_y \\ \text{eye}_z \end{bmatrix}$$

- Chuẩn hóa vector I ta được vector I_0 :

$$\vec{I}_0 = \frac{I}{\sqrt{I_x^2 + I_y^2 + I_z^2}}$$

- Ta có thể dễ dàng thấy rằng, phép biến đổi V mà ta đang xây dựng sẽ chuyển I_0 thành vector $[0, 0, -1]$ (Tại sao?).

$$\begin{bmatrix} 0 & 0 & -1 \end{bmatrix} = I_0 V$$

- Ta còn có thể xác định một vector khác. Đó là vector r là tích hữu hướng của vector I và up-vector:

$$\vec{r} = \vec{I} \times \vec{up}$$

- Sau phép biến đổi V , r_0 (vector r đã được chuẩn hóa) sẽ biến thành vector $[1, 0, 0]$.

$$\begin{bmatrix} 1 & 0 & 0 \end{bmatrix} = \vec{r}_0 V$$

trong đó

$$\vec{r}_0 = \frac{\vec{r}}{\sqrt{r_x^2 + r_y^2 + r_z^2}}$$

- Cuối cùng, ta có thể xác định vector u thứ 3, vector u vuông góc với 2 vector r và l :

$$\vec{u} = \vec{r} \times \vec{l}$$

- Vector này, sau khi được chuẩn hóa (thành vector u_0), sẽ biến thành vector $[0, 1, 0]$ bởi V .

$$\begin{bmatrix} 0 & 1 & 0 \end{bmatrix} = \vec{u}_0 V$$

$$\begin{bmatrix} 0 & 1 & 0 \end{bmatrix} = \frac{\vec{u}}{\sqrt{u_x^2 + u_y^2 + u_z^2}} V$$

- Tổng hợp các kết quả trên ta được:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \vec{r}_0 \\ \vec{u}_0 \\ -\vec{l}_0 \end{bmatrix} V$$

- Phương trình trên có thể viết lại như sau:

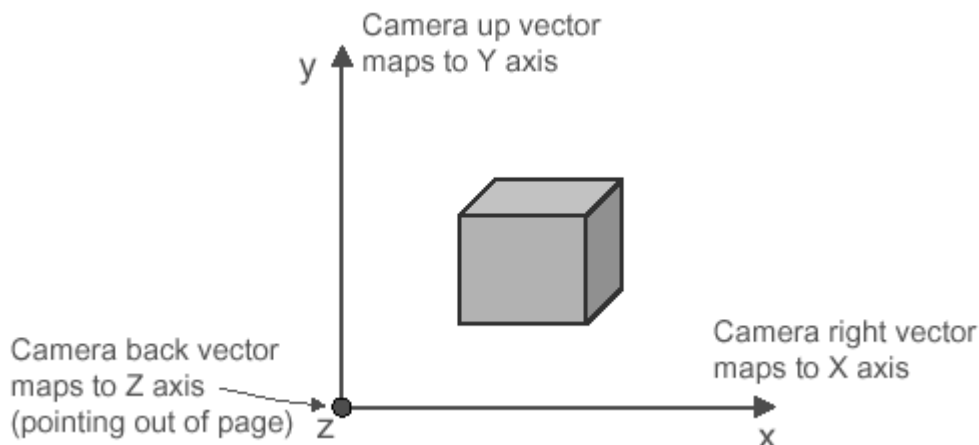
$$[x' \ y' \ z'] = [x \ y \ z] \begin{bmatrix} r_x & u_x & -l_x \\ r_y & u_y & -l_y \\ r_z & u_z & -l_z \end{bmatrix} -$$

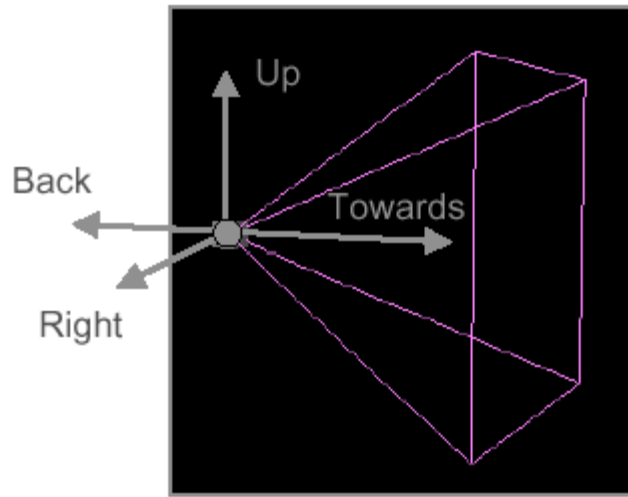
$$\begin{bmatrix} eye_x & eye_y & eye_z \end{bmatrix} \begin{bmatrix} r_x & u_x & -l_x \\ r_y & u_y & -l_y \\ r_z & u_z & -l_z \end{bmatrix}$$

- Cuối cùng, ta có thể chuyển phép biến đổi sang dạng biểu diễn trong hệ tọa độ thuận nhất. Nội chính là công thức cuối cùng của V:

$$[x' \ y' \ z' \ 1] = [x \ y \ z \ 1] \begin{bmatrix} r_x & u_x & -l_x & 0 \\ r_y & u_y & -l_y & 0 \\ r_z & u_z & -l_z & 0 \\ -r_0 \cdot eye & -u_0 \cdot eye & l_0 \cdot eye & 1 \end{bmatrix}$$

- Nhờ vậy, ta có mối quan hệ giữa hệ tọa độ quan sát và hệ tọa độ thế giới như sau:





View Frustum