Introduction to Computer Graphics

4. Viewing in 3D (Example)

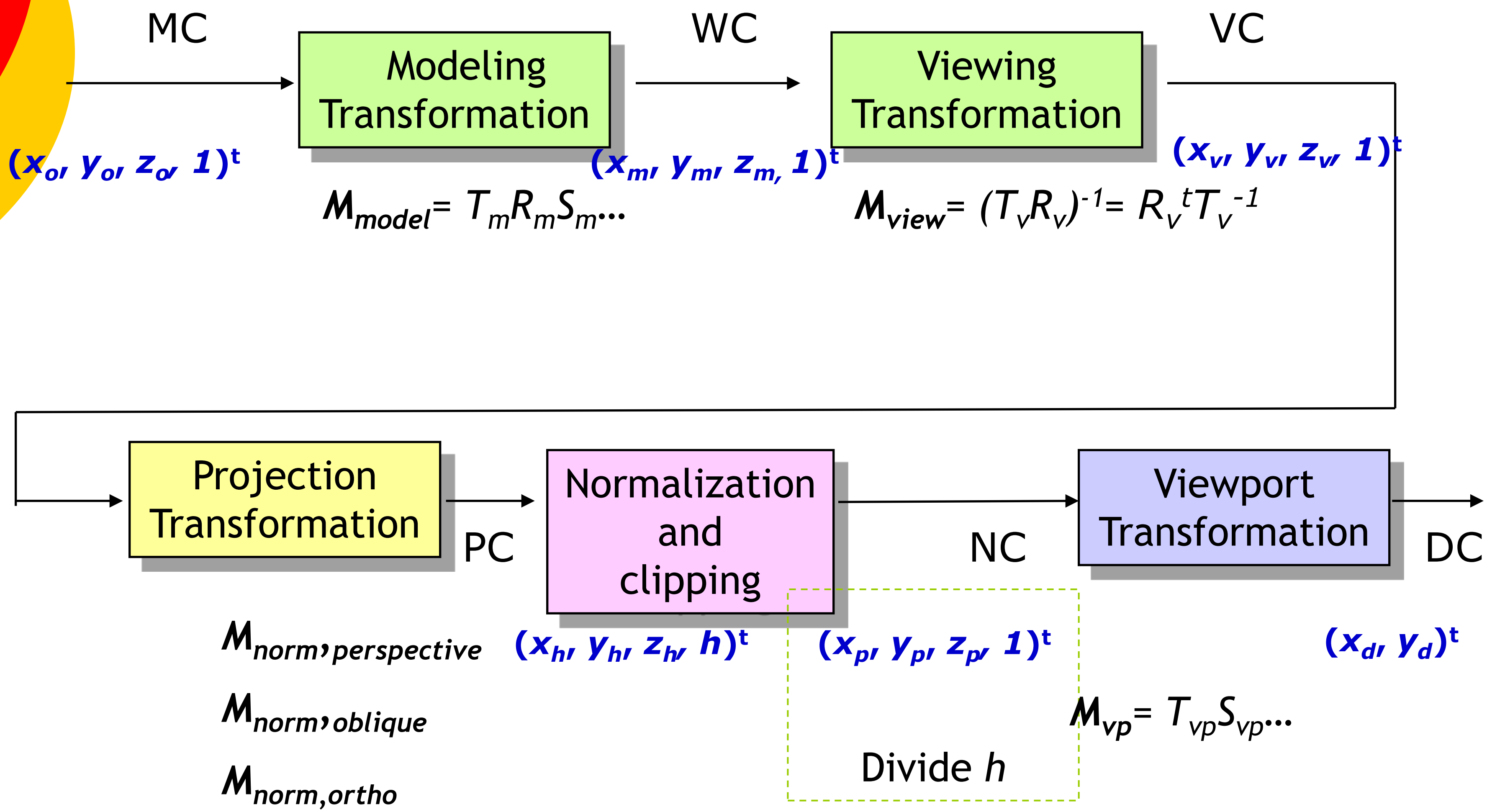
National Chiao Tung Univ, Taiwan

By: I-Chen Lin, Assistant Professor

Textbook: E.Angel, Interactive Computer Graphics, 5th Ed., Addison Wesley

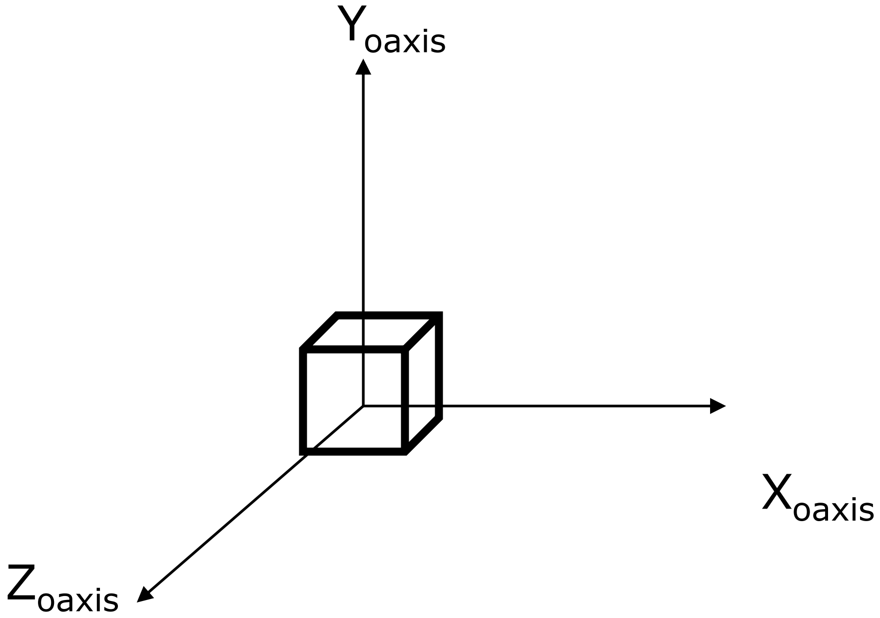
Ref:Hearn and Baker, Computer Graphics, 3rd Ed., Prentice Hall

Pipeline View



Loading an Object

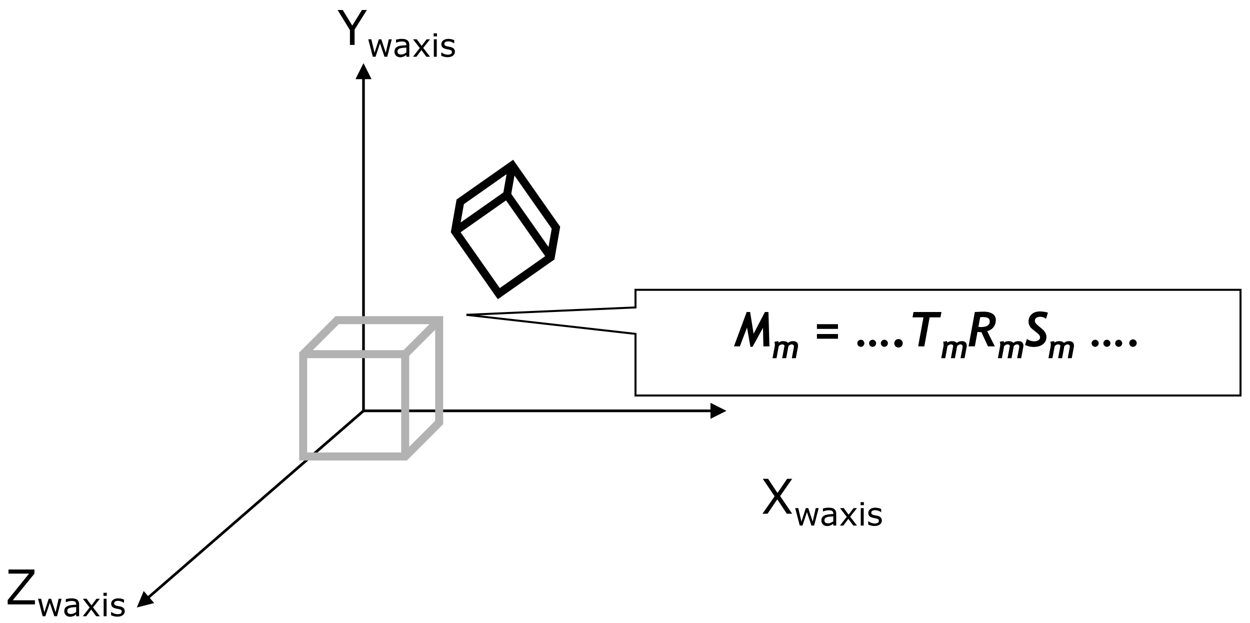
$(x_o, y_o, z_o, 1)^t$



Object coord.

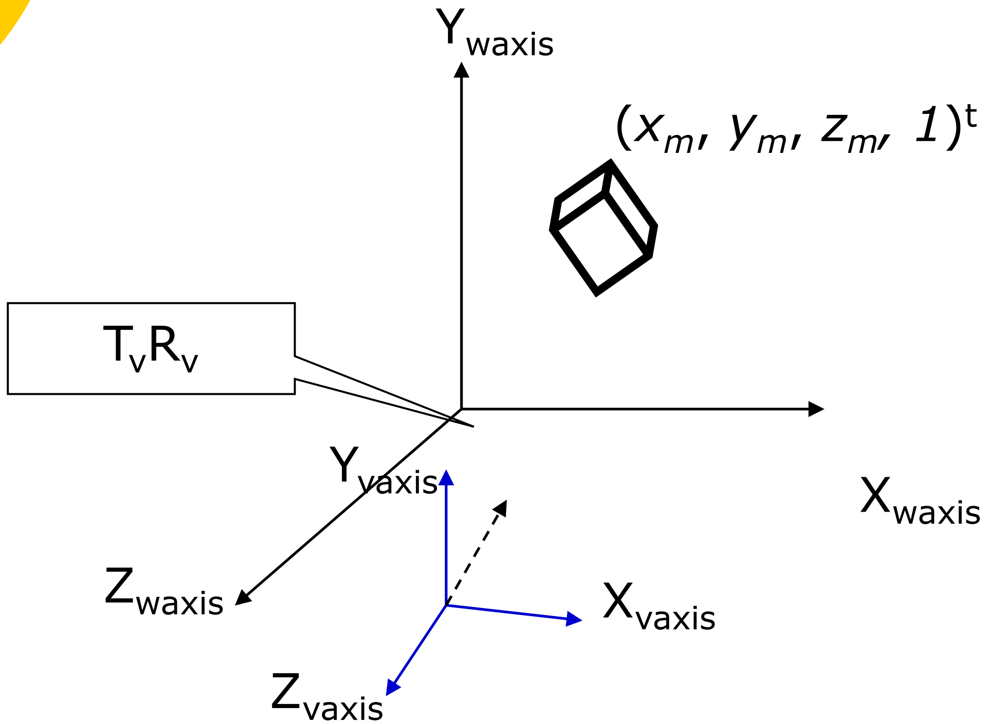
Modeling Transformation

- $(x_m, y_m, z_m, 1)^t = M_m(x_o, y_o, z_o, 1)^t$
  where $M_m = \ldots . T_m R_m S_m \ldots .$



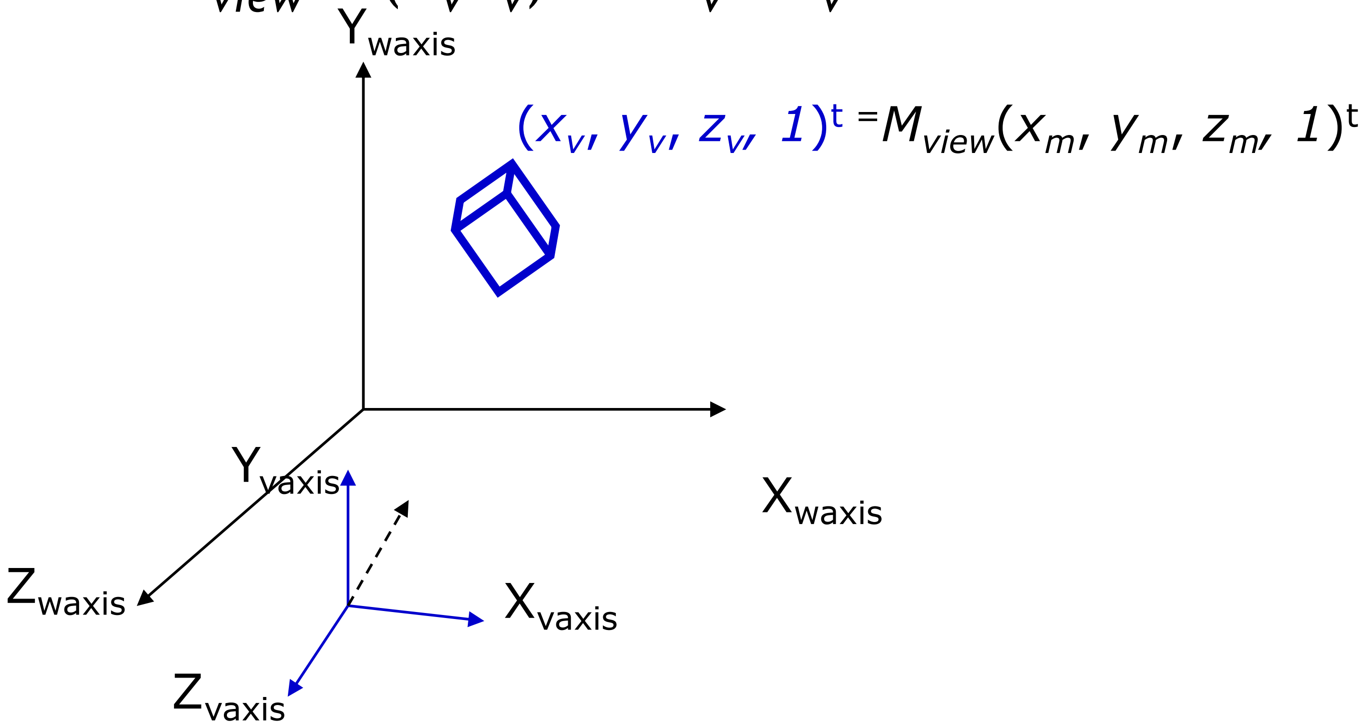
Put a Virtual Camera

- Move a camera from the origin (by T_vR_v)

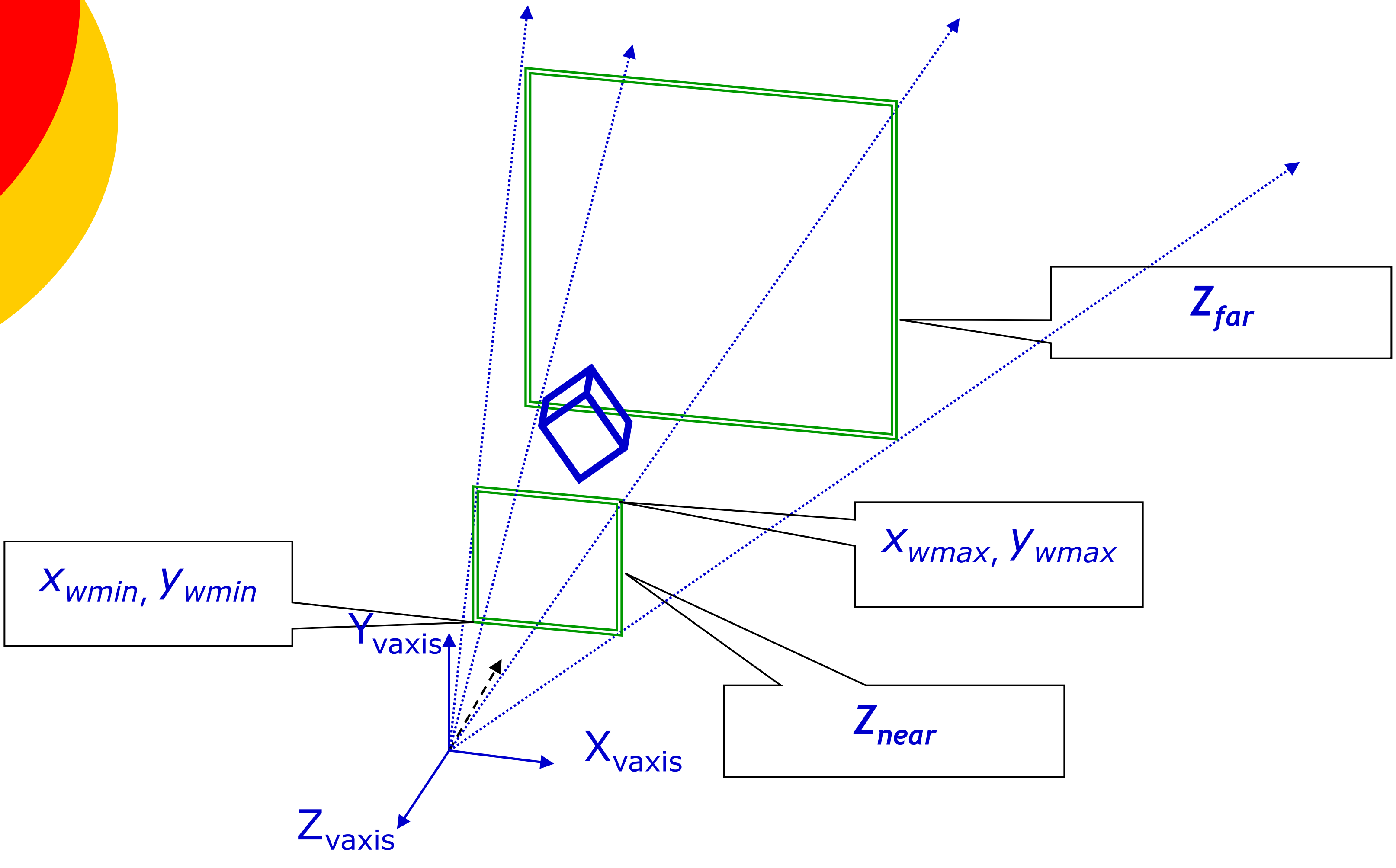


Virtual camera's Coordinate

- Change the object's coordinate
- $(x_v, y_v, z_v, 1)^t = M_{view} (x_m, y_m, z_m, 1)^t$
- $M_{view} = (T_v R_v)^{-1} = R_v^{-1}\ T_v^{-1}$



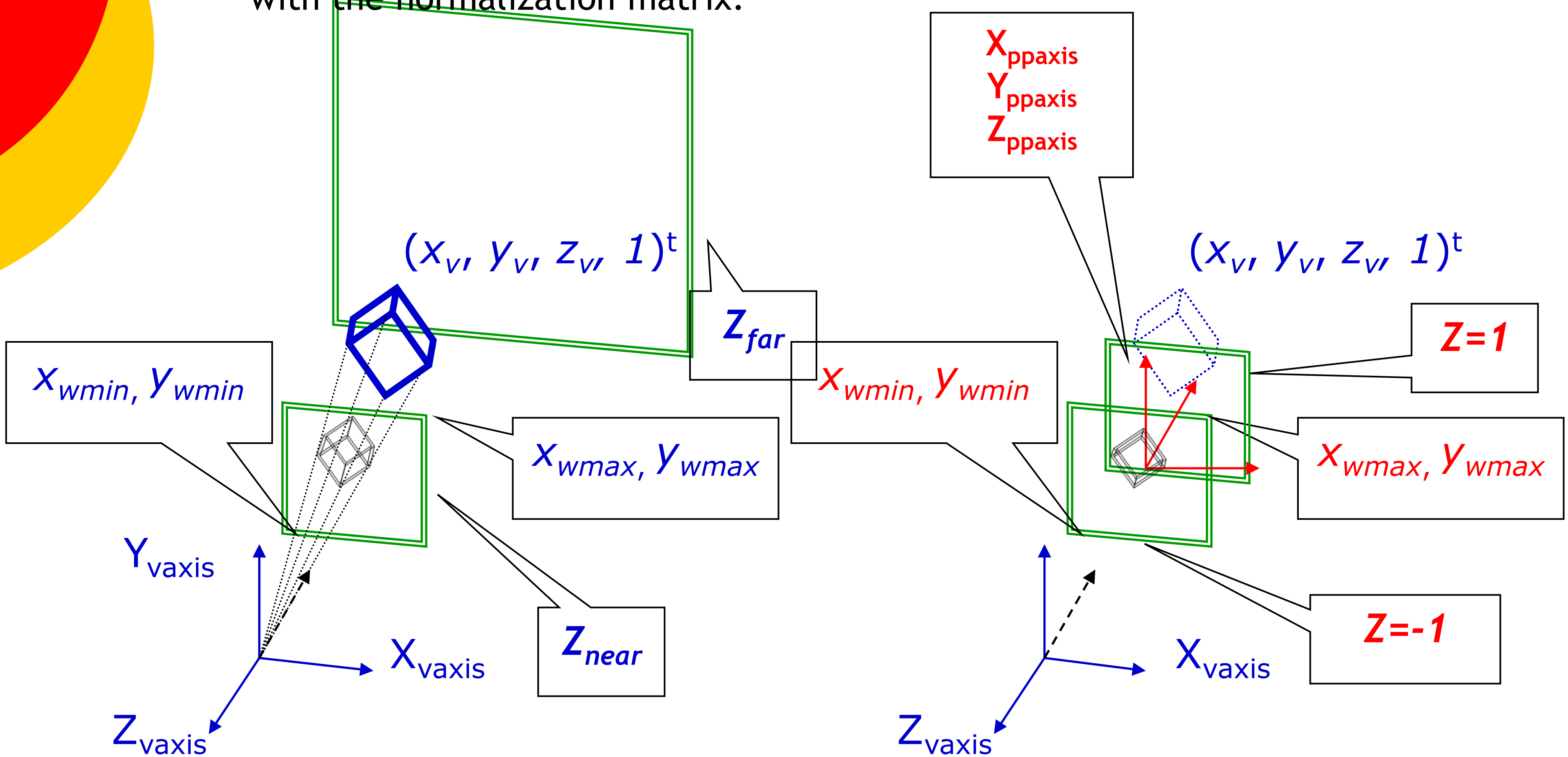
Virtual camera's Coordinate



Perspective Projection

$$M_{pers} = \begin{bmatrix} -z_{near} & 0 & 0 & 0 \\ 0 & -z_{near} & 0 & 0 \\ 0 & 0 & \dfrac{z_{near}+z_{far}}{z_{near}-z_{far}} & \dfrac{-2z_{near}z_{far}}{z_{near}-z_{far}} \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

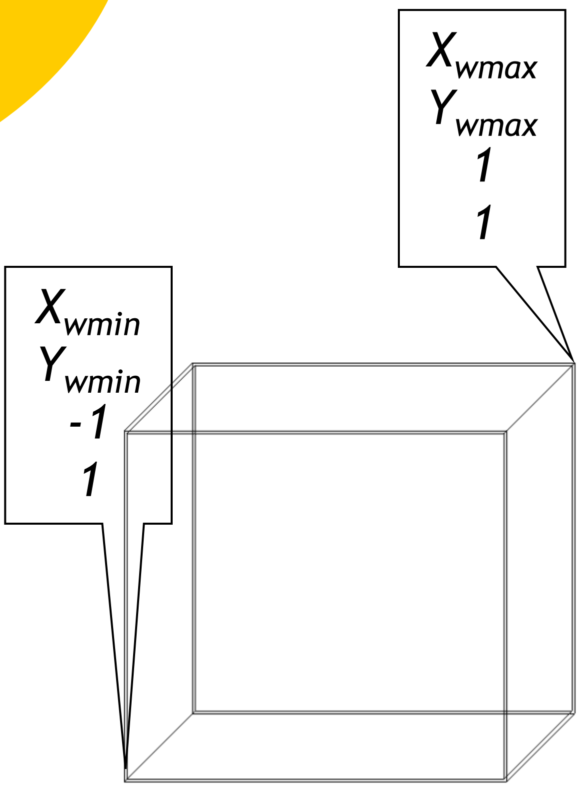
This matrix is usually combined with the normalization matrix.

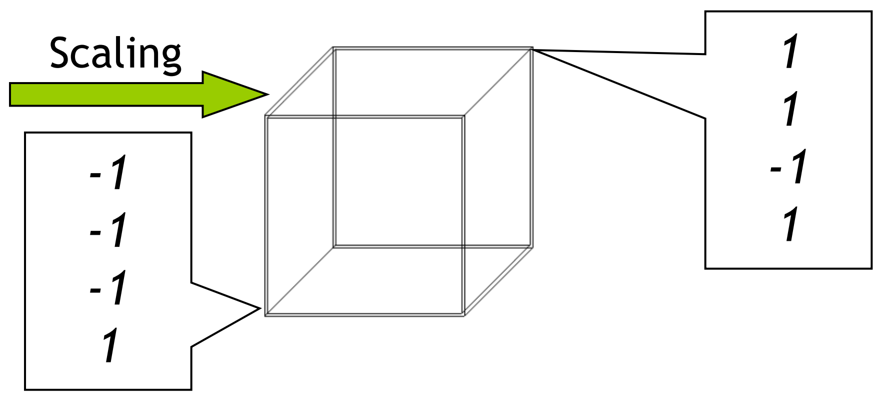


Projection + Normalization

$$M_{normpers} = \begin{bmatrix} -z_{near}\frac{2}{xw_{\max} - xw_{\min}} & 0 & 0 & 0 \\ 0 & -z_{near}\frac{2}{yw_{\max} - yw_{\min}} & 0 & 0 \\ 0 & 0 & \frac{z_{near} + z_{far}}{z_{near} - z_{far}} & \frac{-2z_{near}z_{far}}{z_{near} - z_{far}} \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} \frac{2}{xw_{\max} - xw_{\min}} & 0 & 0 & 0 \\ 0 & \frac{2}{yw_{\max} - yw_{\min}} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} M_{pers}$$

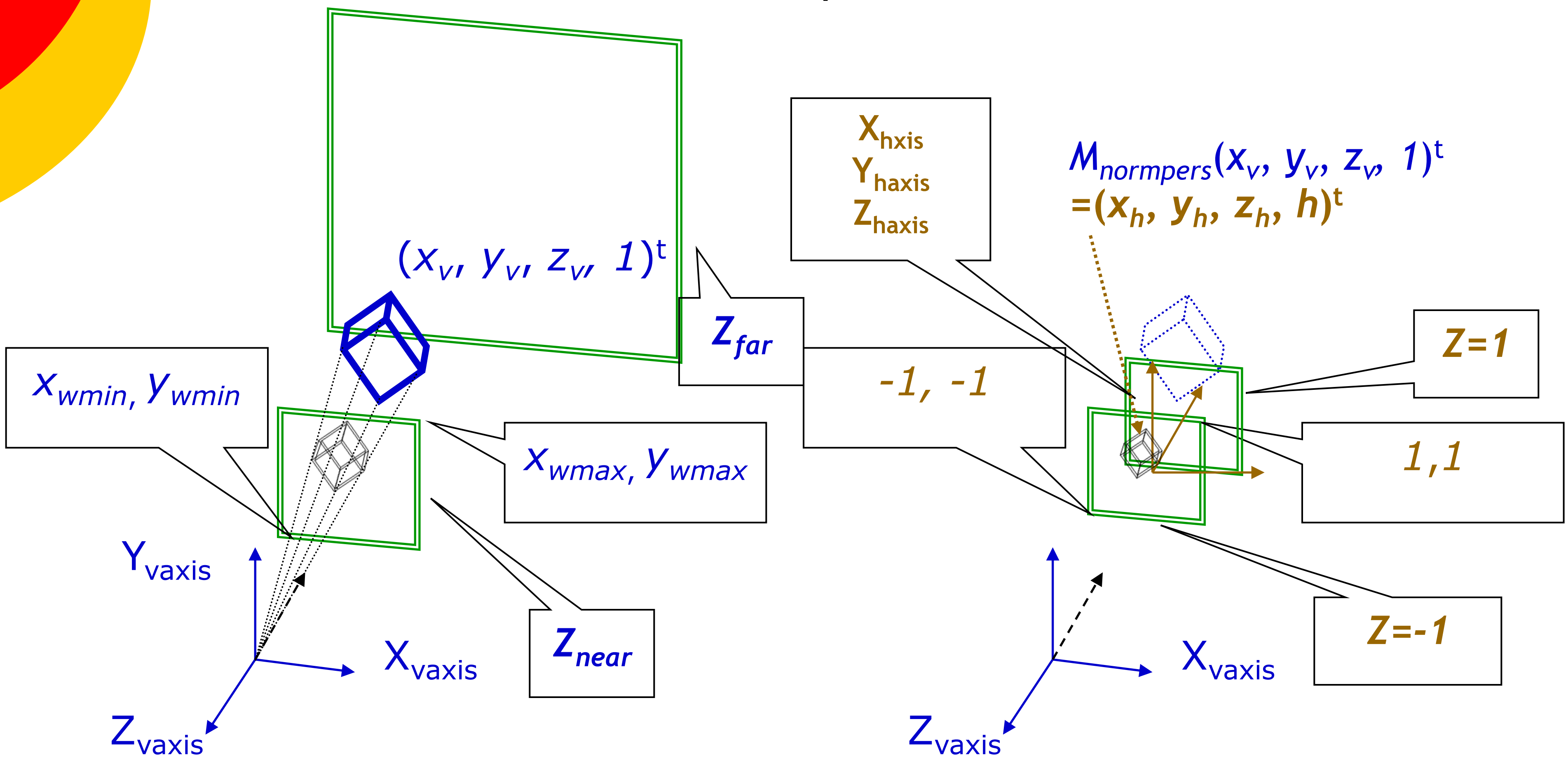




Projection+ Normalization

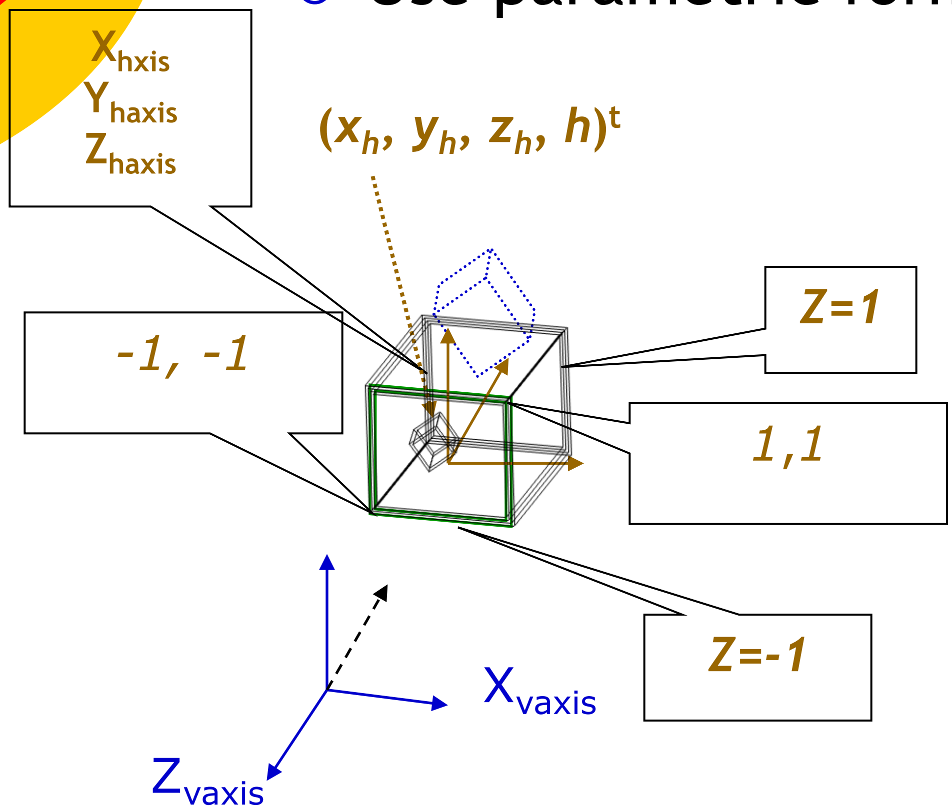
$$M_{normpers} = \begin{bmatrix} -z_{near}\frac{2}{xw_{\max} - xw_{\min}} & 0 & 0 & 0 \\ 0 & -z_{near}\frac{2}{yw_{\max} - yw_{\min}} & 0 & 0 \\ 0 & 0 & \frac{z_{near} + z_{far}}{z_{near} - z_{far}} & \frac{-2z_{near}z_{far}}{z_{near} - z_{far}} \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

- $(x_h, y_h, z_h, h)^t = M_{normpers}(x_v, y_v, z_v, 1)^t$
- Don’t divide h at this step.



Clipping

- Perform clipping with $(x_h, y_h, z_h, h)^t$
- Avoid unnecessary division $-h \le x_h \le h,\ -h \le y_h \le h,\ -h \le z_h \le h$
- Use parametric forms for intersection



$$x_h = x_{ha} + (x_{hb} - x_{ha})u$$

$$y_h = y_{ha} + (y_{hb} - y_{ha})u$$

$$z_h = z_{ha} + (z_{hb} - z_{ha})u$$

$$h = h_a + (h_b - h_a)u$$

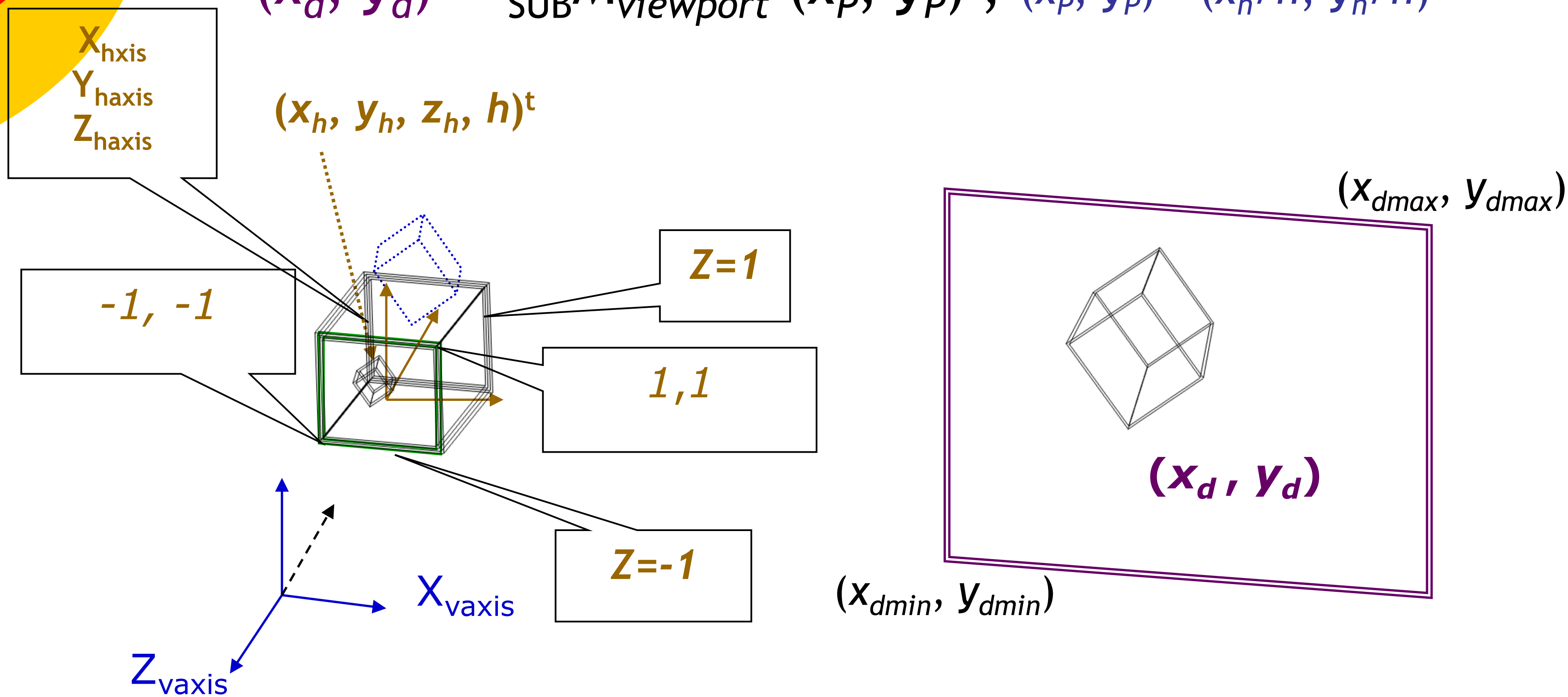
Viewport Transformation

$$M_{viewport} = \begin{bmatrix} \frac{x_{d\max} - x_{d\min}}{2} & 0 & 0 & \frac{x_{d\max} + x_{d\min}}{2} \\ 0 & \frac{y_{d\max} - y_{d\min}}{2} & 0 & \frac{y_{d\max} + y_{d\min}}{2} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$(x_d, y_d, z_d, 1)^t = M_{viewport}\,(x_h, y_h, z_h, h)^t$

OR

$(x_d, y_d)^t = {}_{SUB}M_{viewport}\,(x_P, y_P)^t$, $(x_P, y_P)^t = (x_h/h, y_h/h)^t$



Rasterization

- Line drawing or polygon filling with
  $(x_d, y_d, z_d, 1)^t$ or $(x_d, y_d)^t$ and z_h