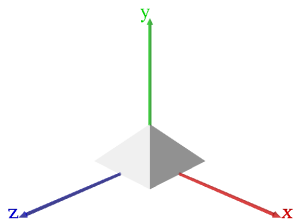


GRK 2

dr Wojciech Palubicki

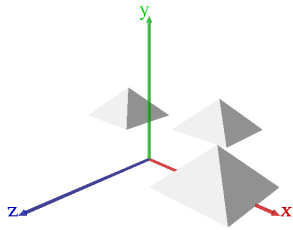
Uproszczony Potok Graficzny (Rendering)

Model Matrix



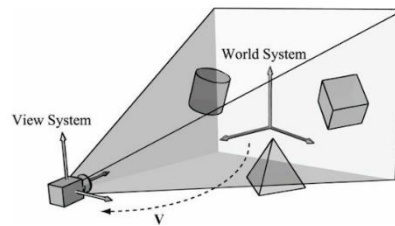
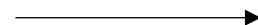
Object Space

View Matrix



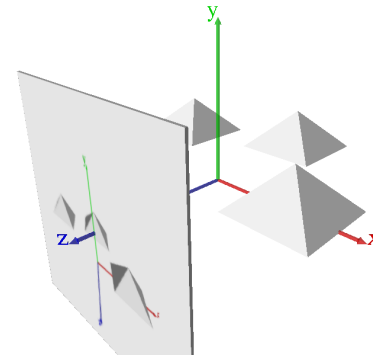
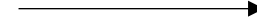
World Space

Projection Matrix

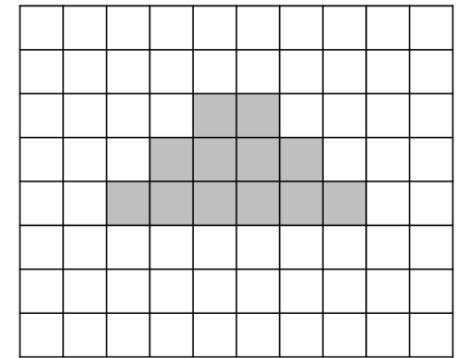


View Space

Viewport Transform

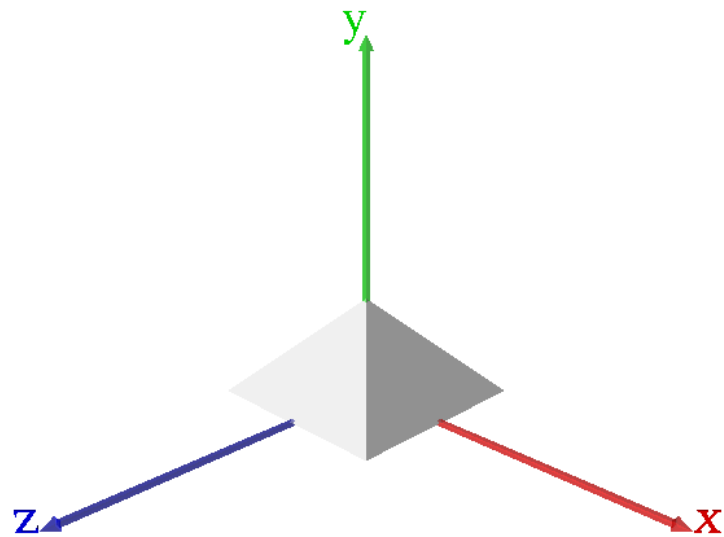


Clip Space

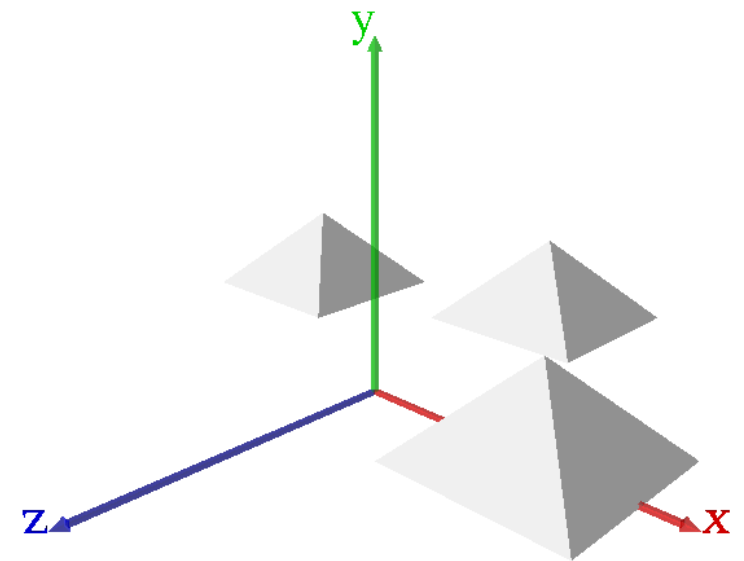


Screen Space

World space transformation

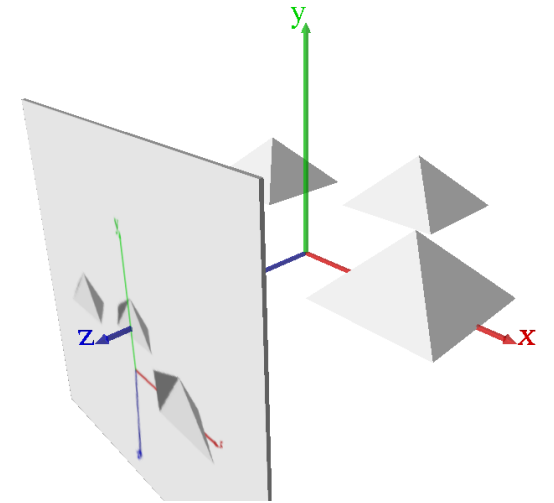
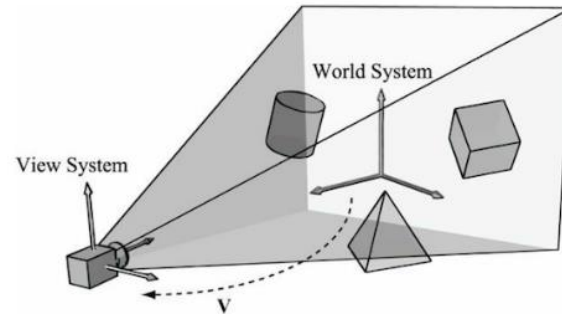
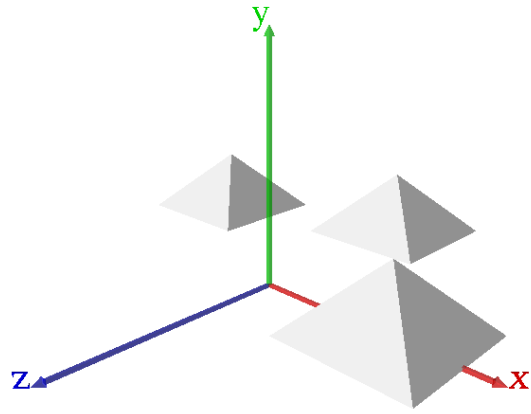


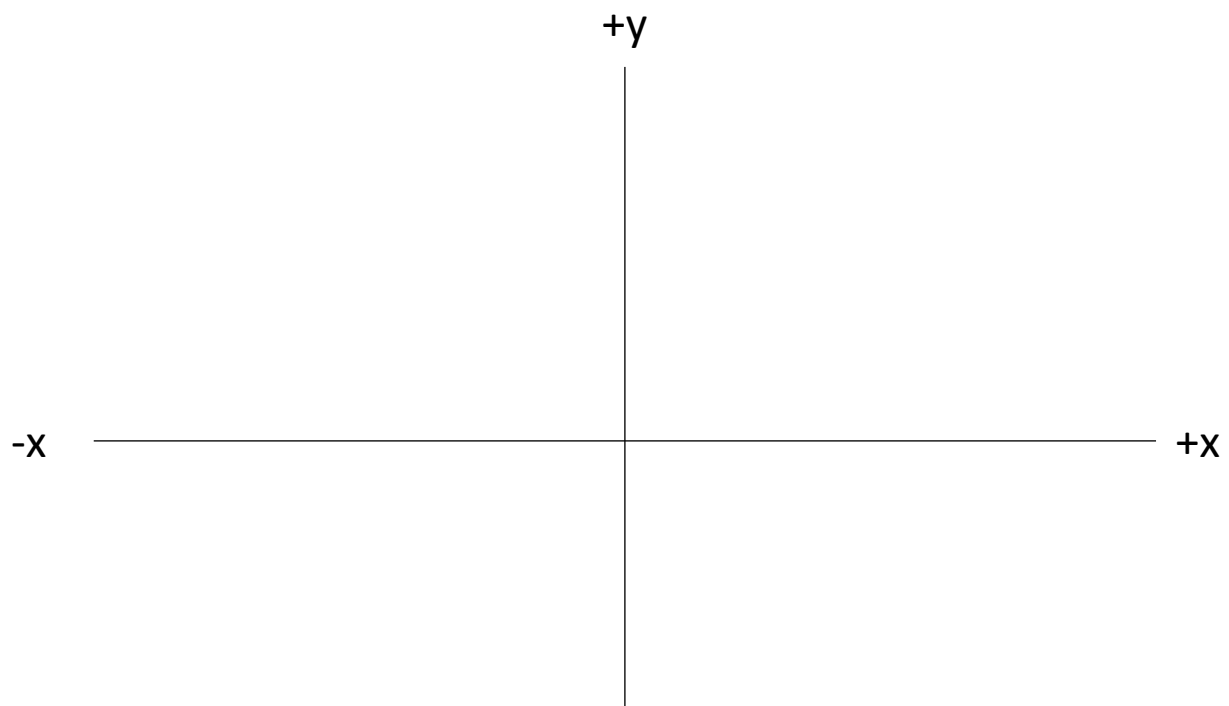
Object Space

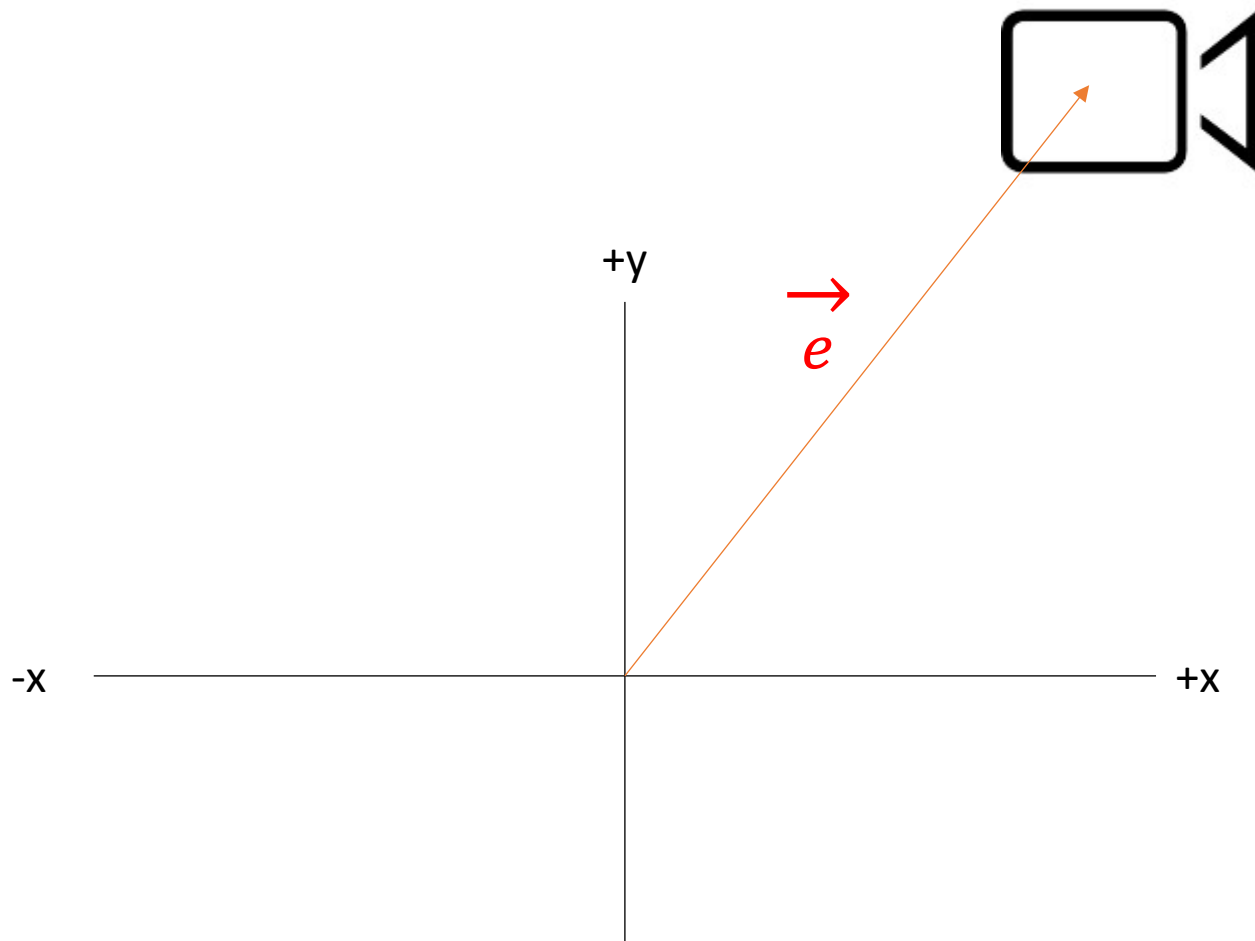


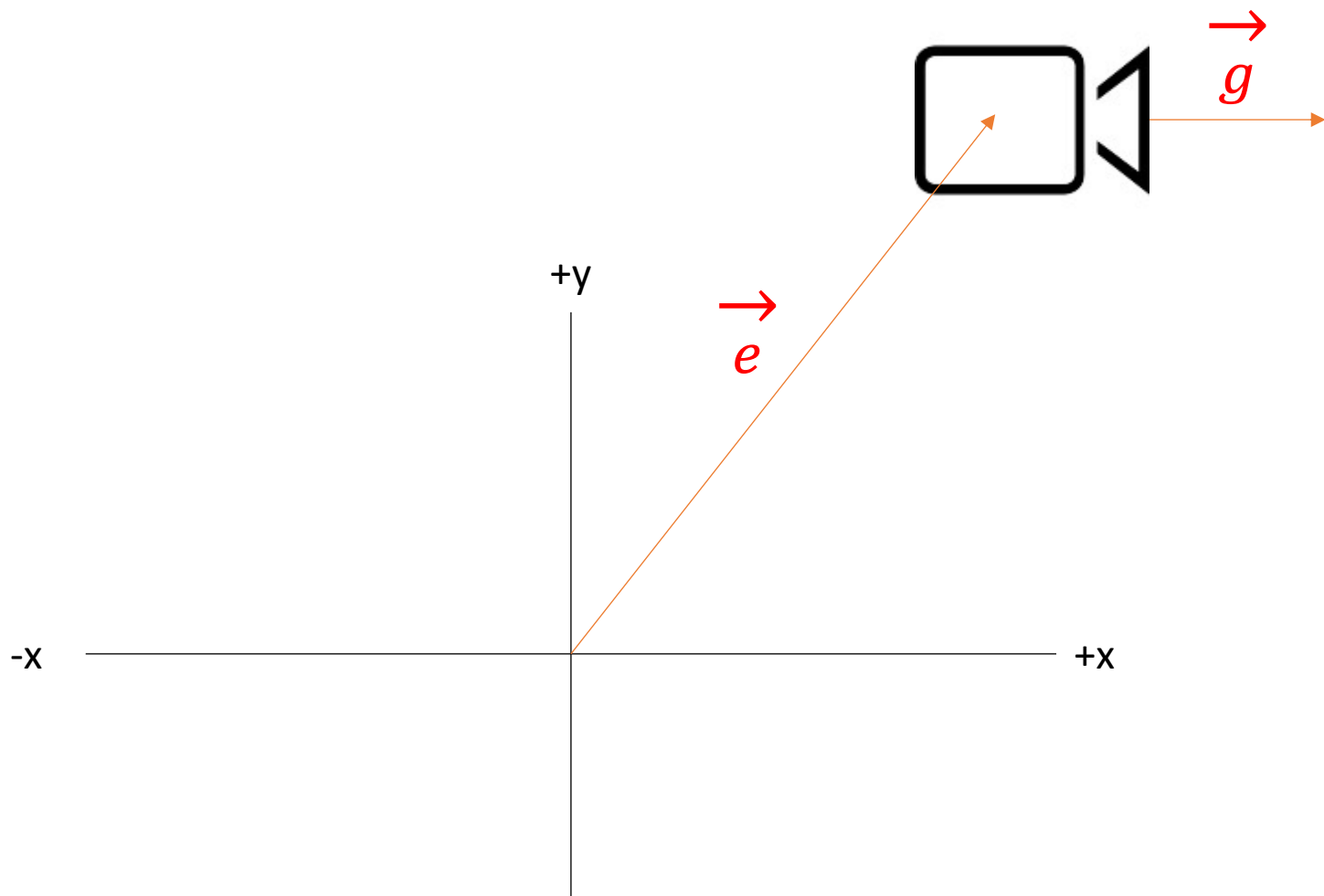
World Space

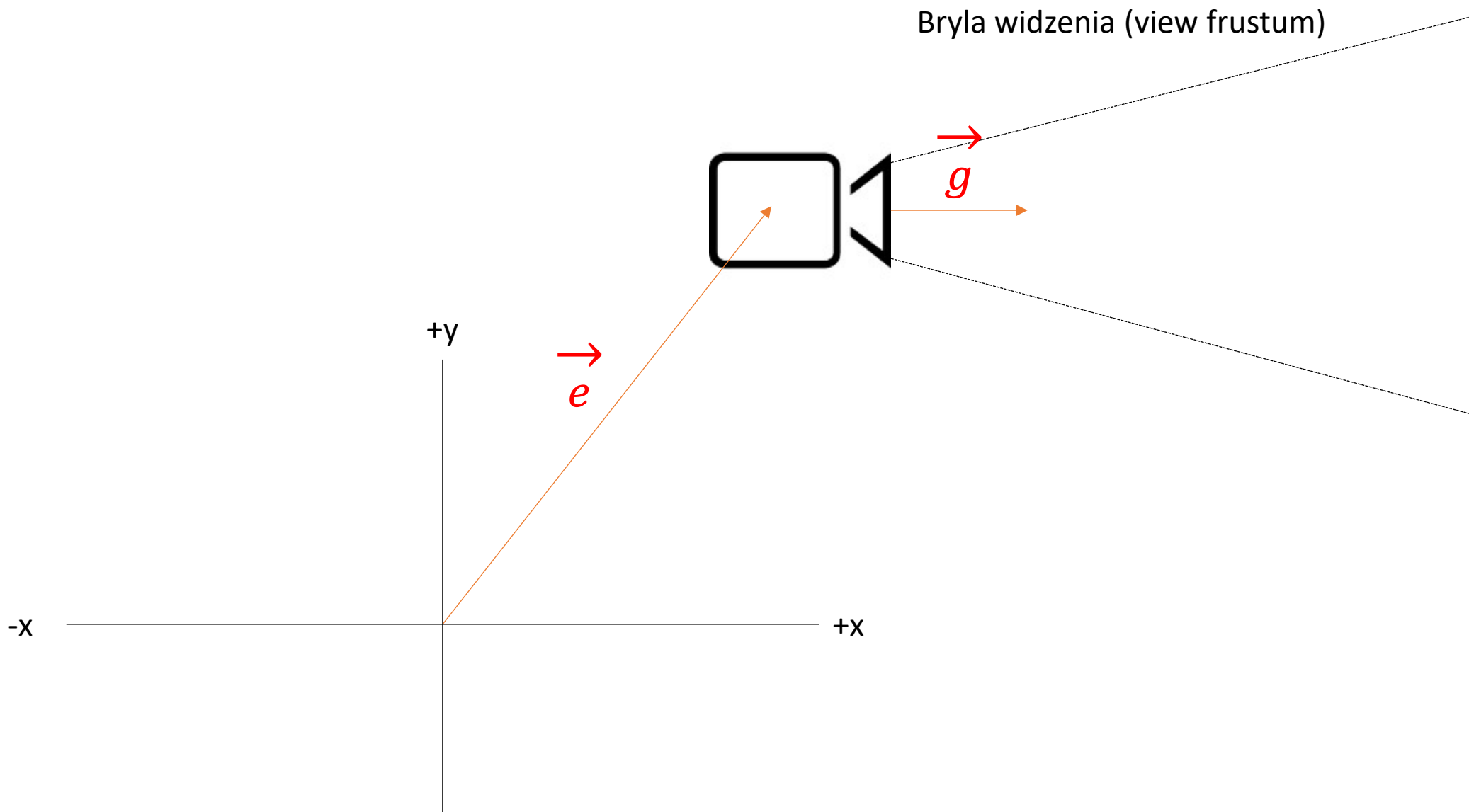
World space $\rightarrow$ 2D

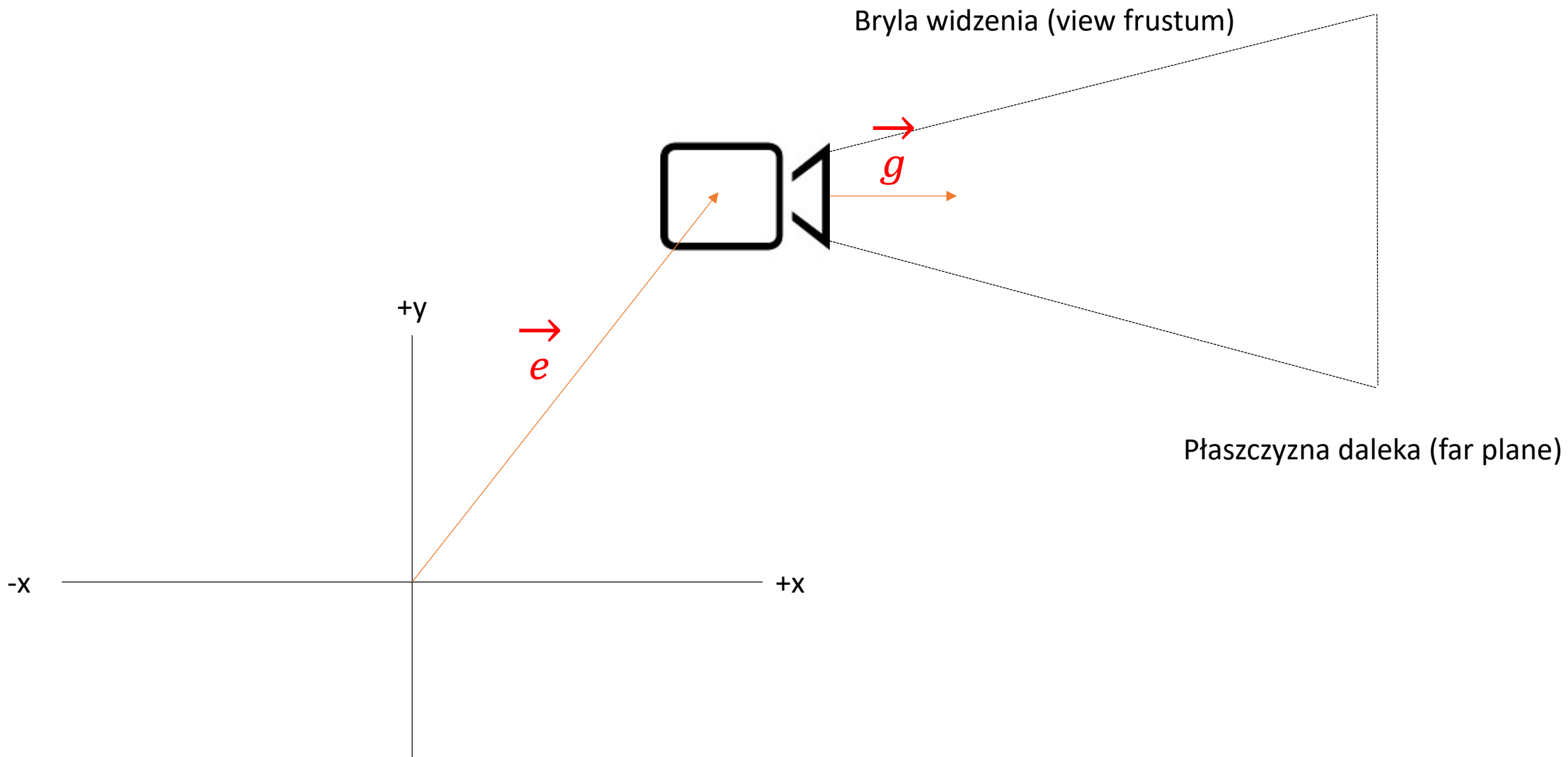


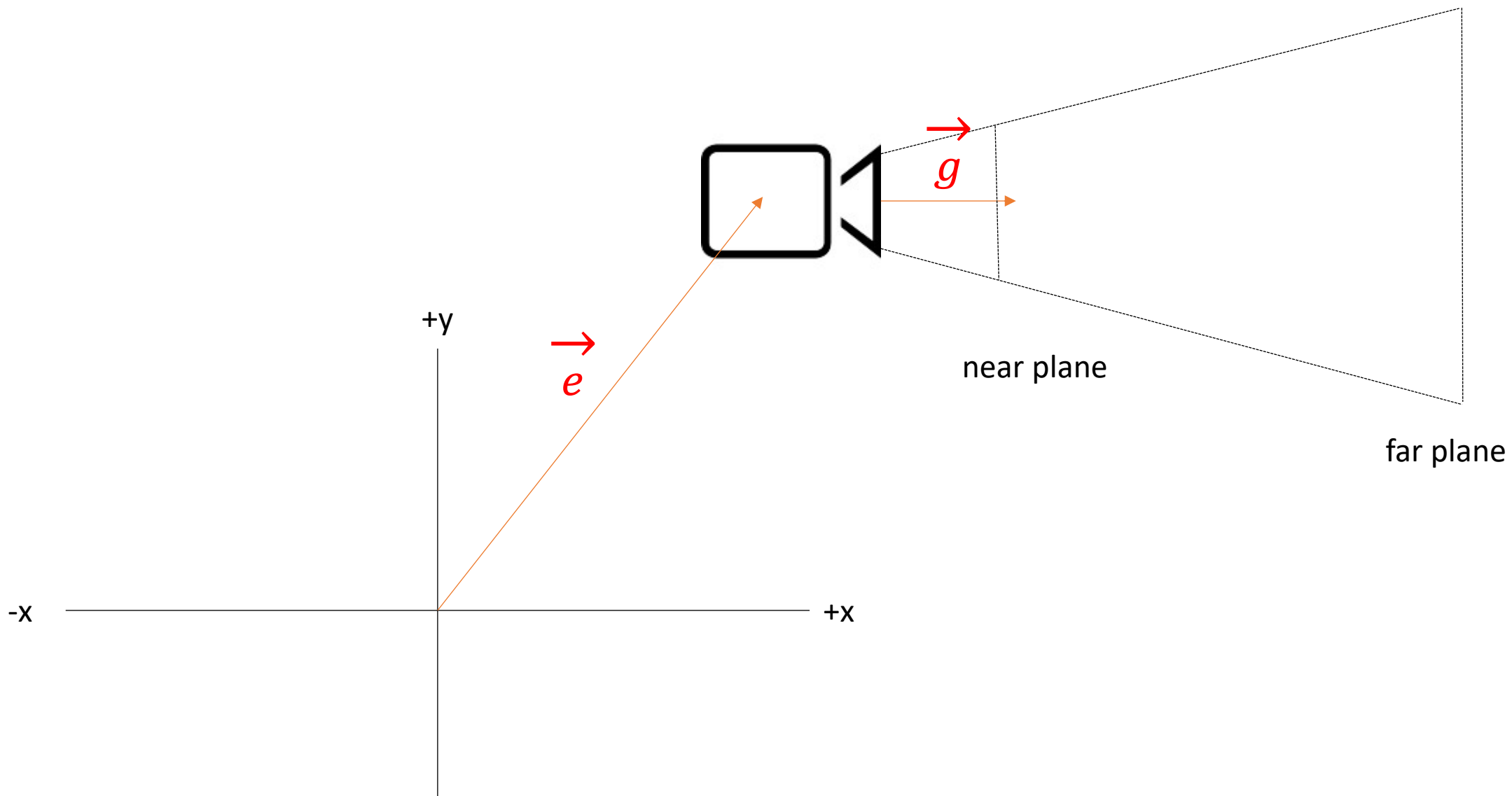


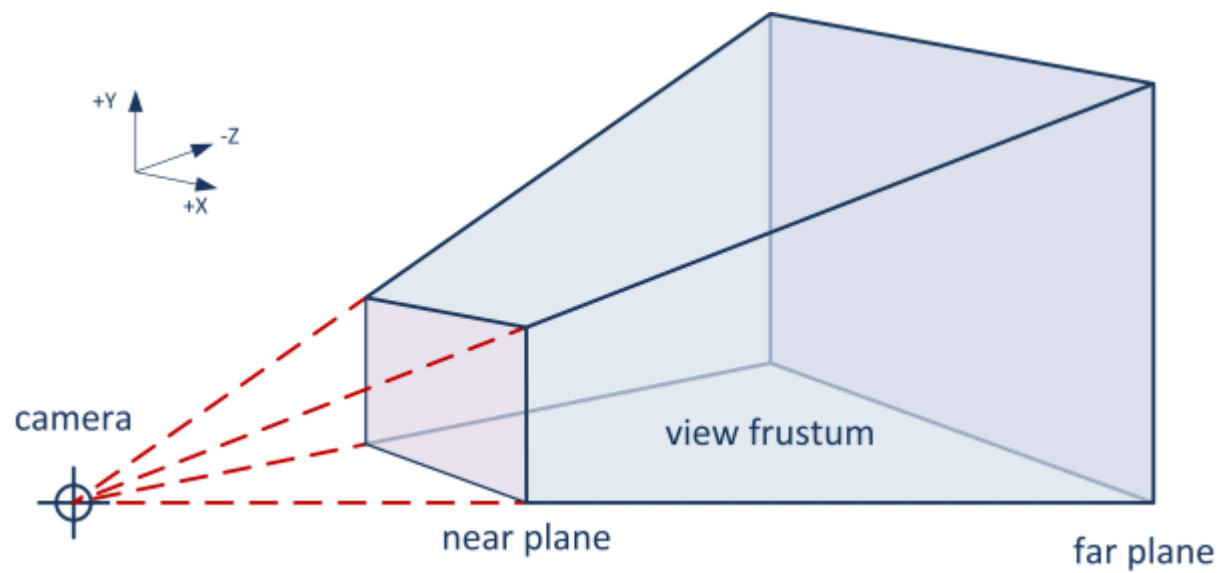


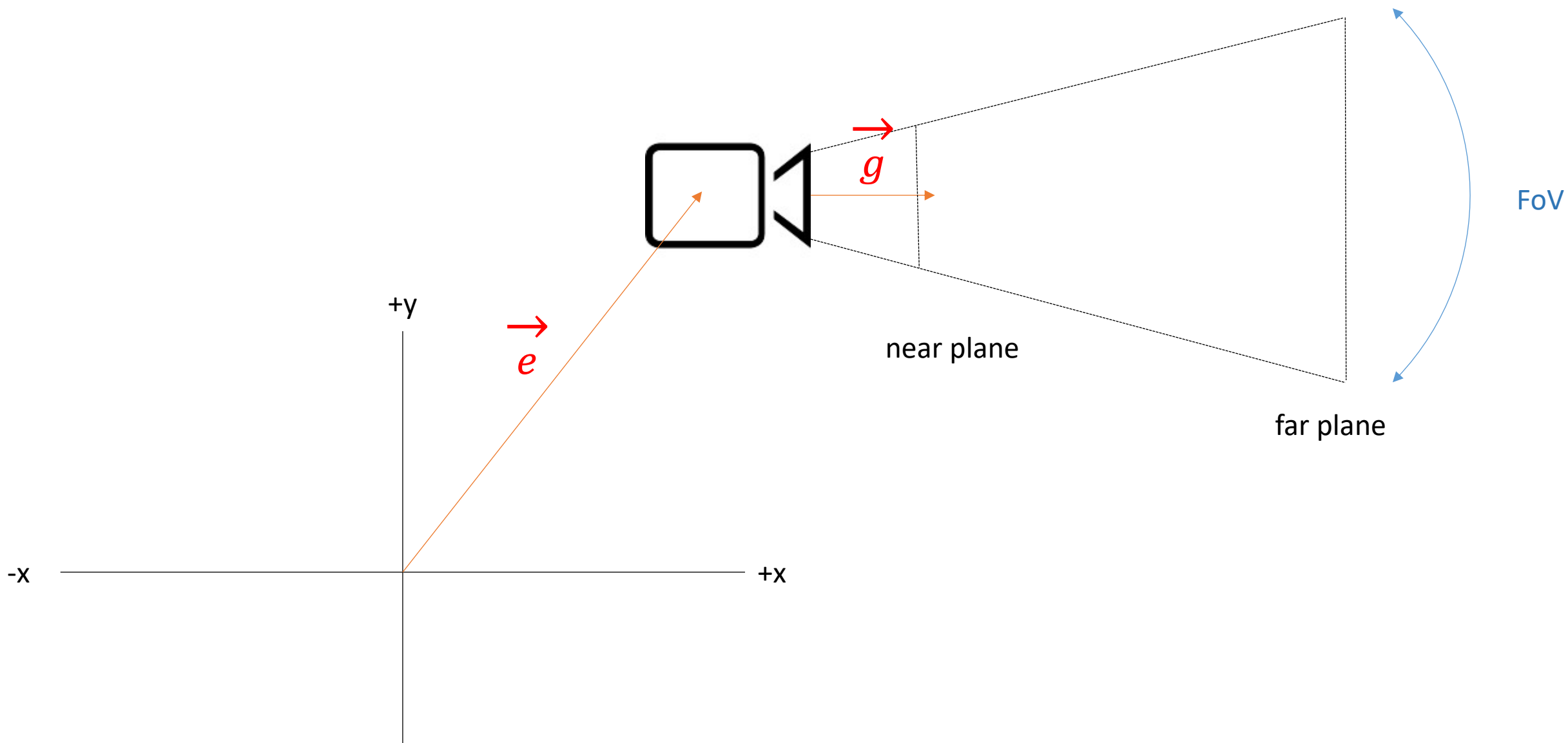




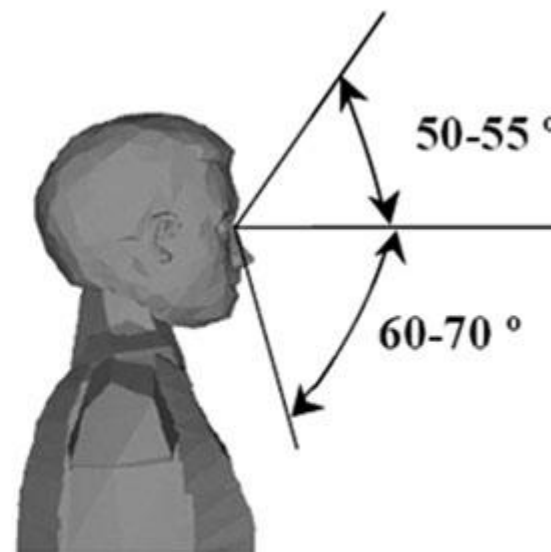
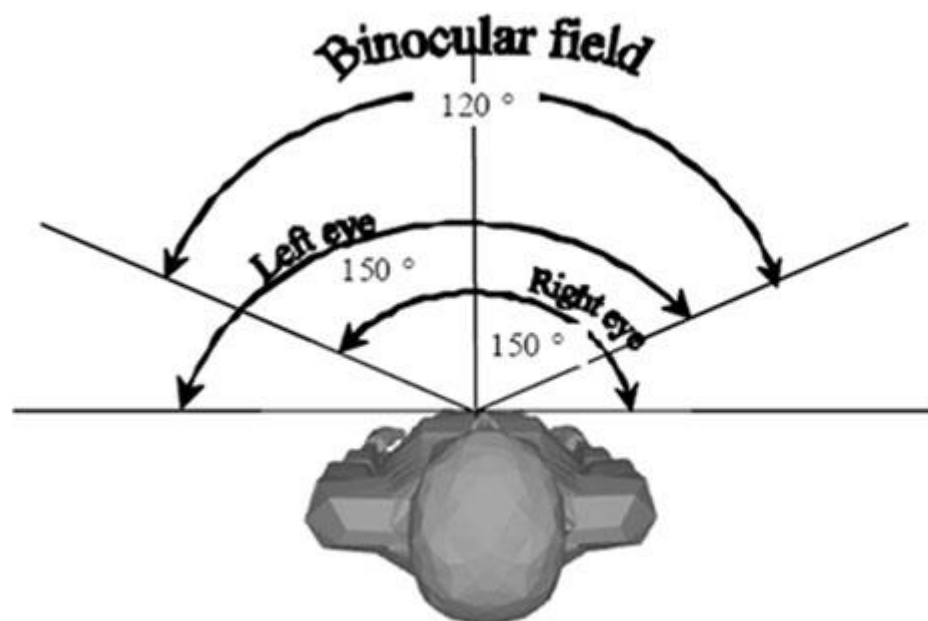




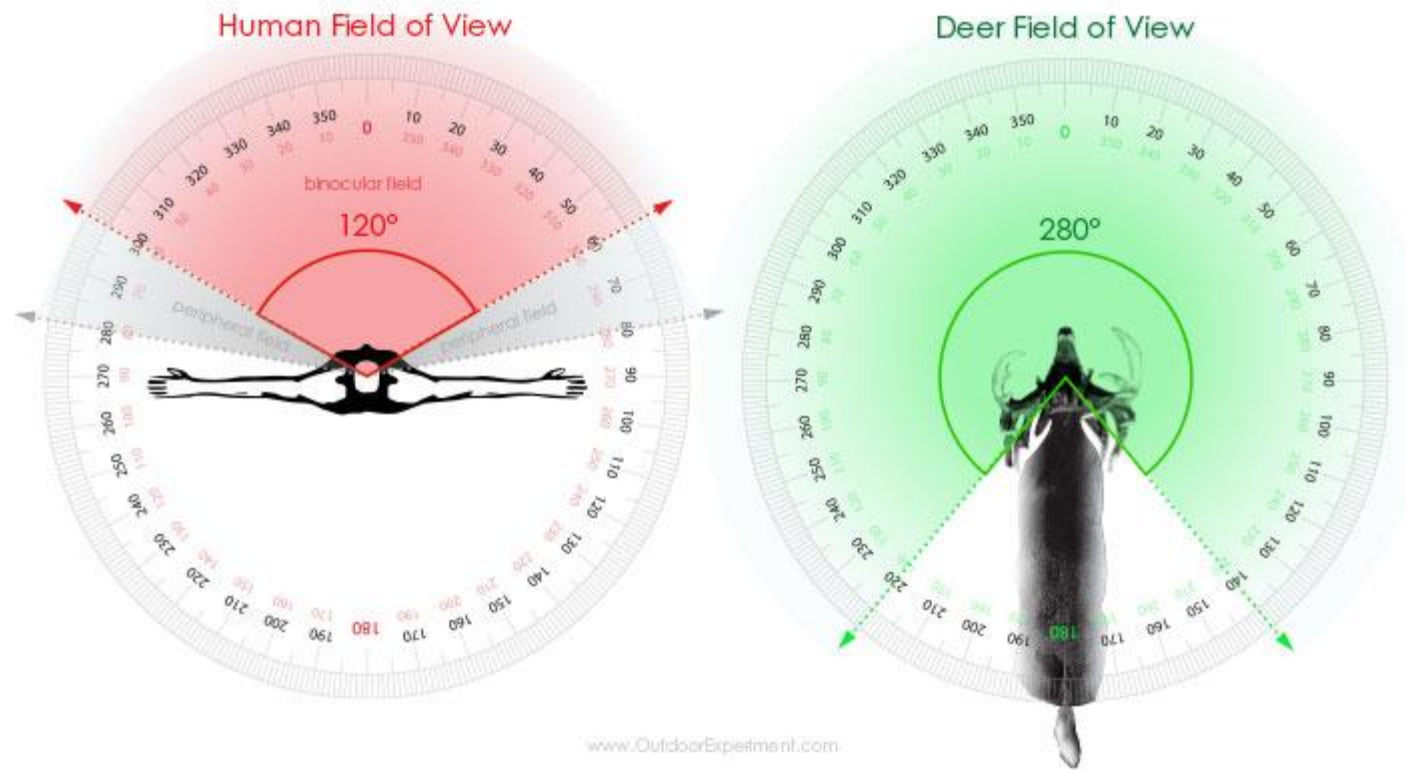




Pole widzenia (Field of View - FoV)

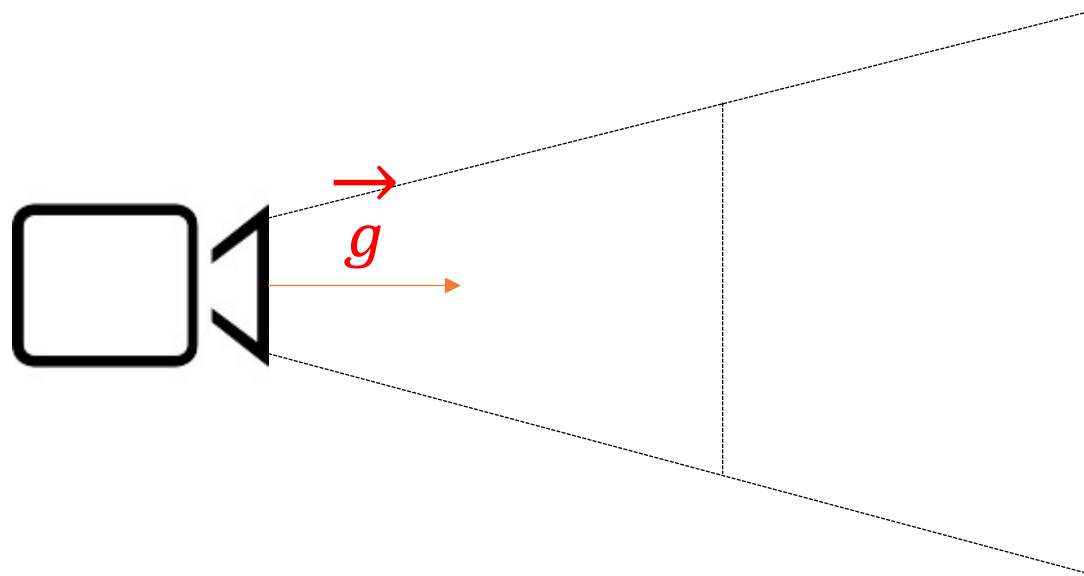


Pole widzenia (Field of View - FoV)



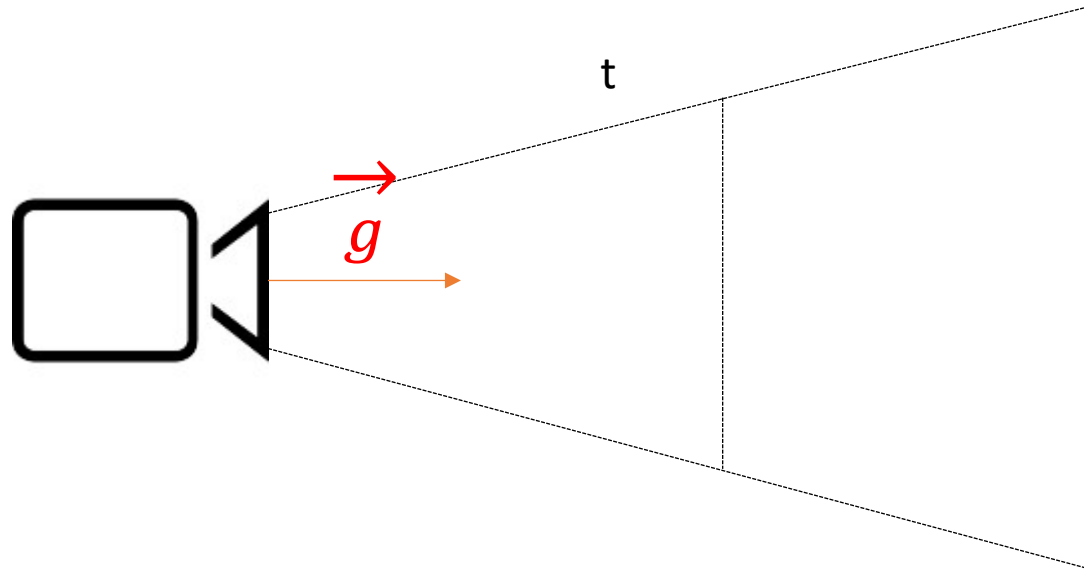
Transformacja kamery

- Bryła widzenia:



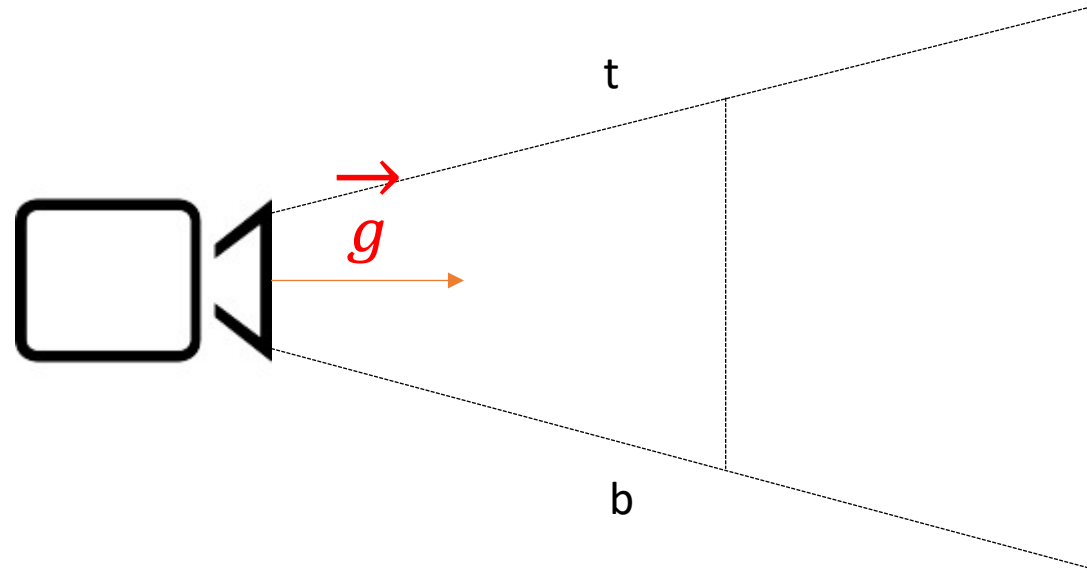
Transformacja kamery

- Bryła widzenia:
 - górna płaszczyzna t



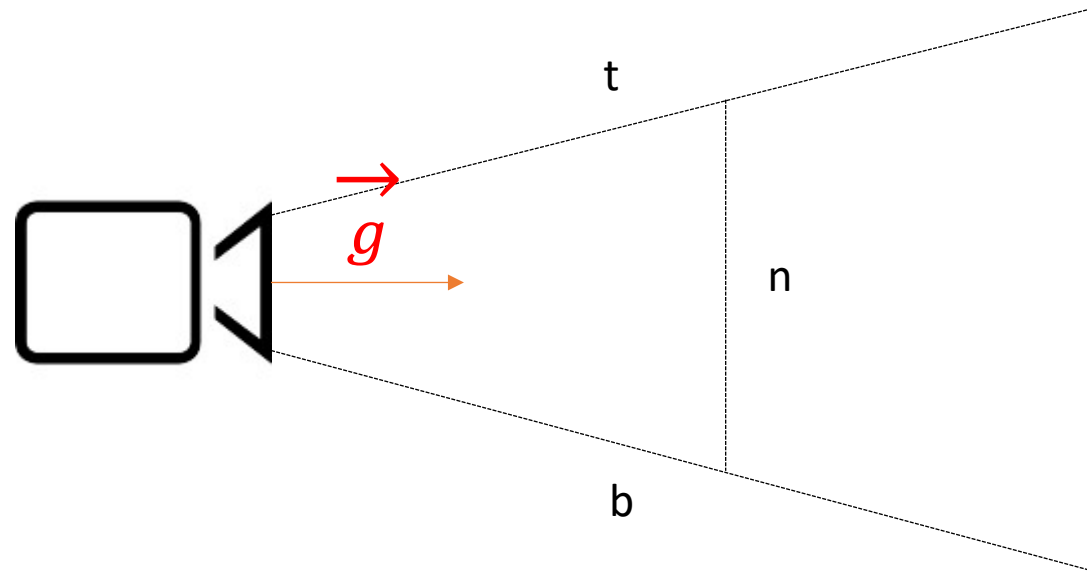
Transformacja kamery

- Bryła widzenia:
 - górna płaszczyzna t
 - dolna płaszczyzna b



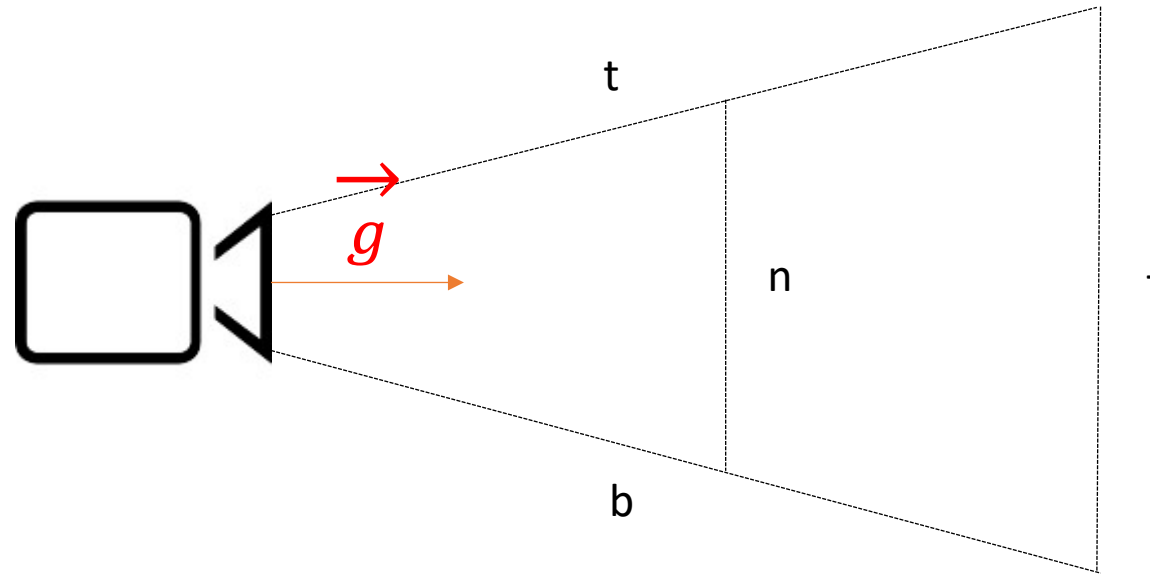
Transformacja kamery

- Bryła widzenia:
 - górna płaszczyzna t
 - dolna płaszczyzna b
 - bliska płaszczyzna n



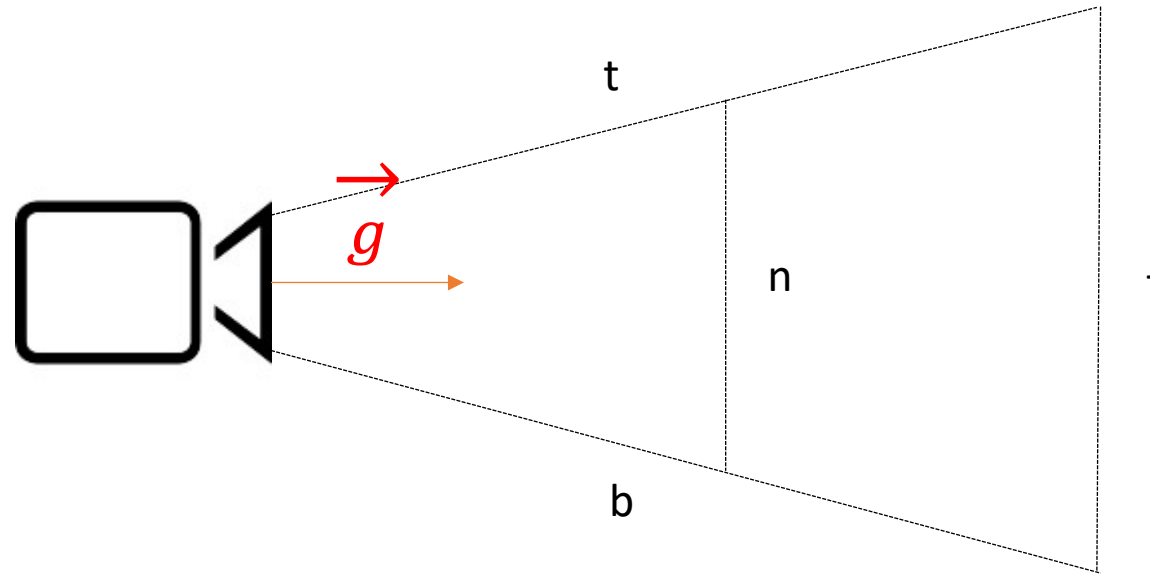
Transformacja kamery

- Bryła widzenia:
 - górna płaszczyzna t
 - dolna płaszczyzna b
 - bliska płaszczyzna n
 - daleka płaszczyzna f

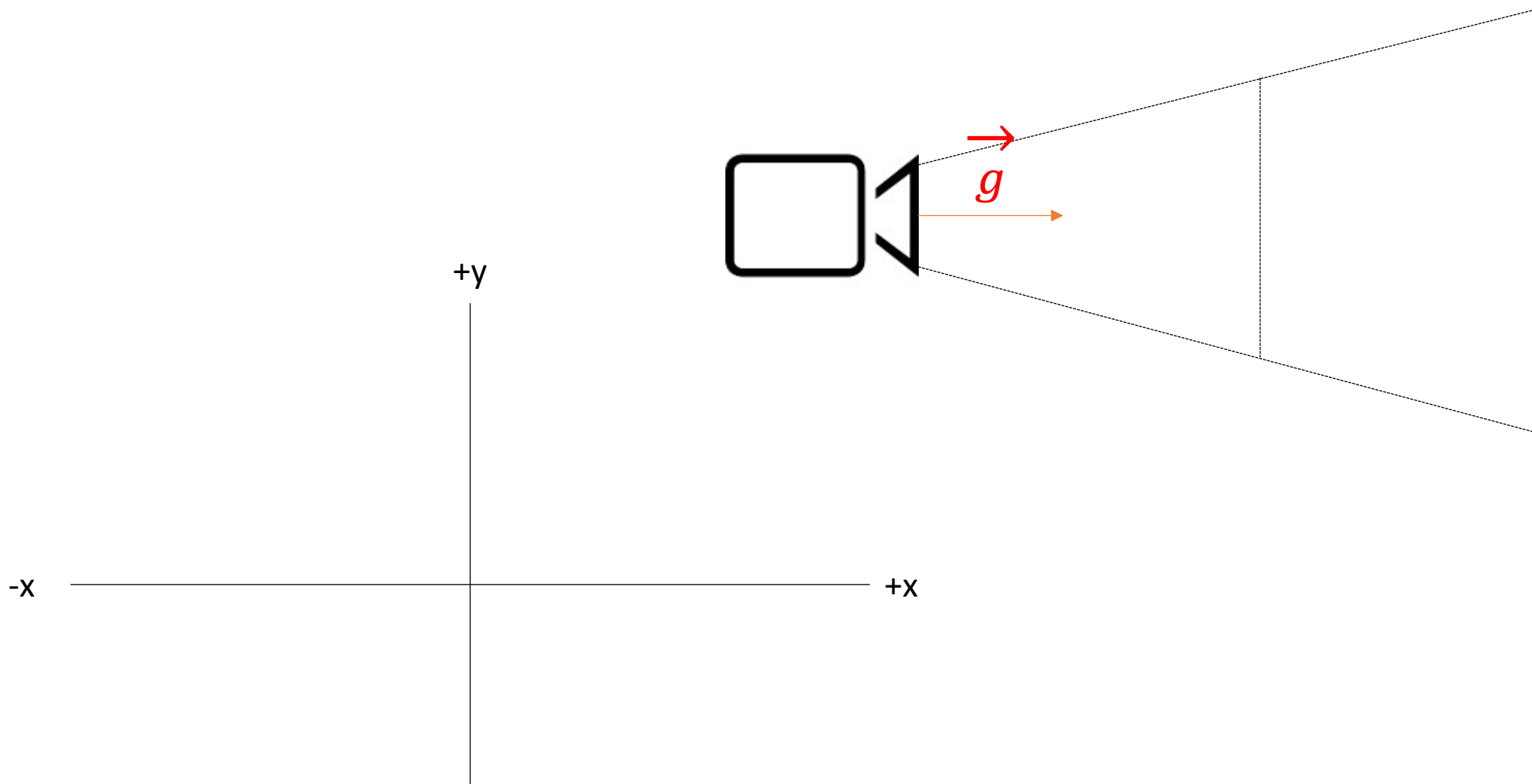


Transformacja kamery

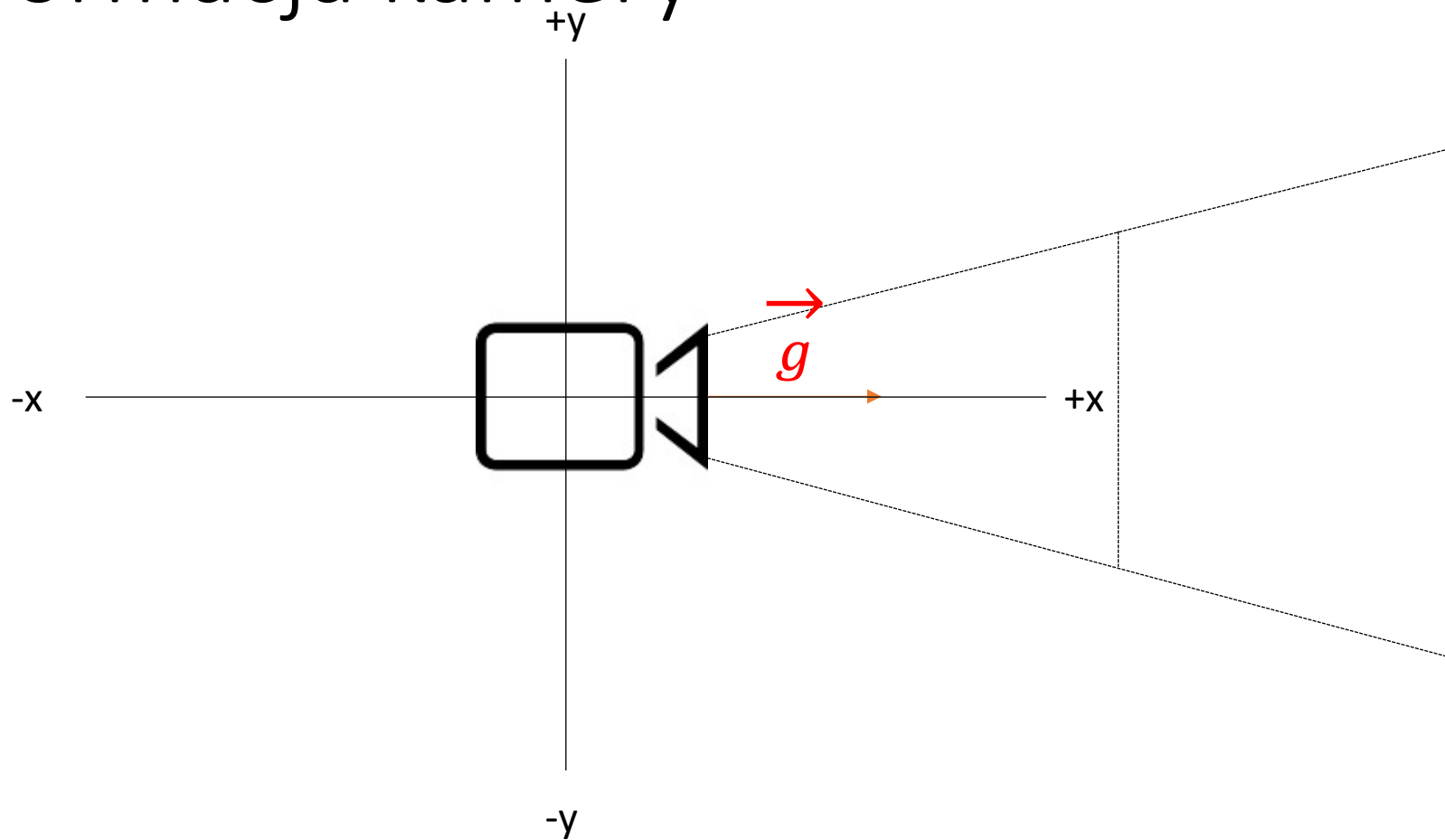
- Bryła widzenia:
 - górna płaszczyzna t
 - dolna płaszczyzna b
 - bliska płaszczyzna n
 - daleka płaszczyzna f
 - lewa płaszczyzna l
 - prawa płaszczyzna r



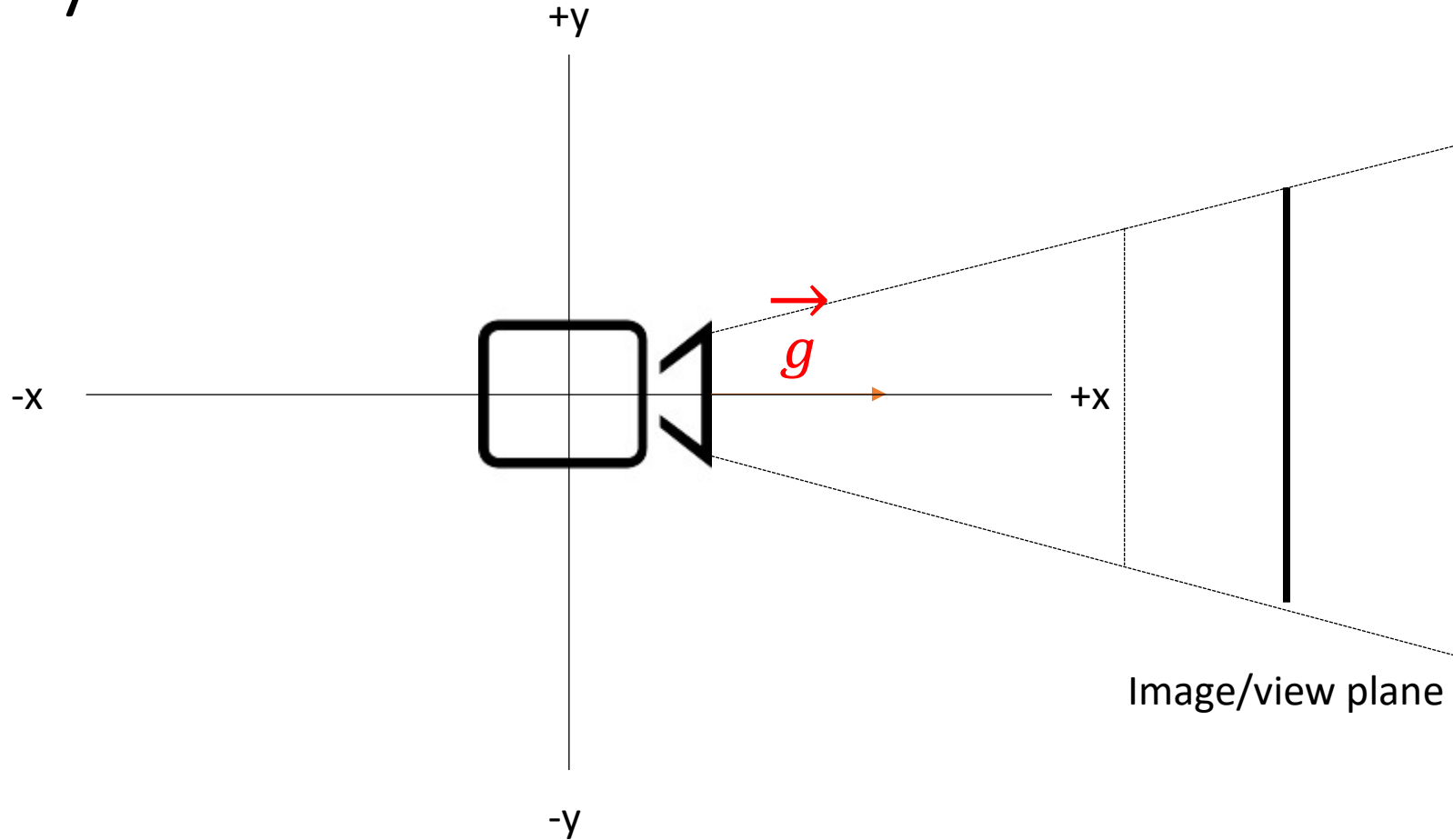
Transformacja kamery



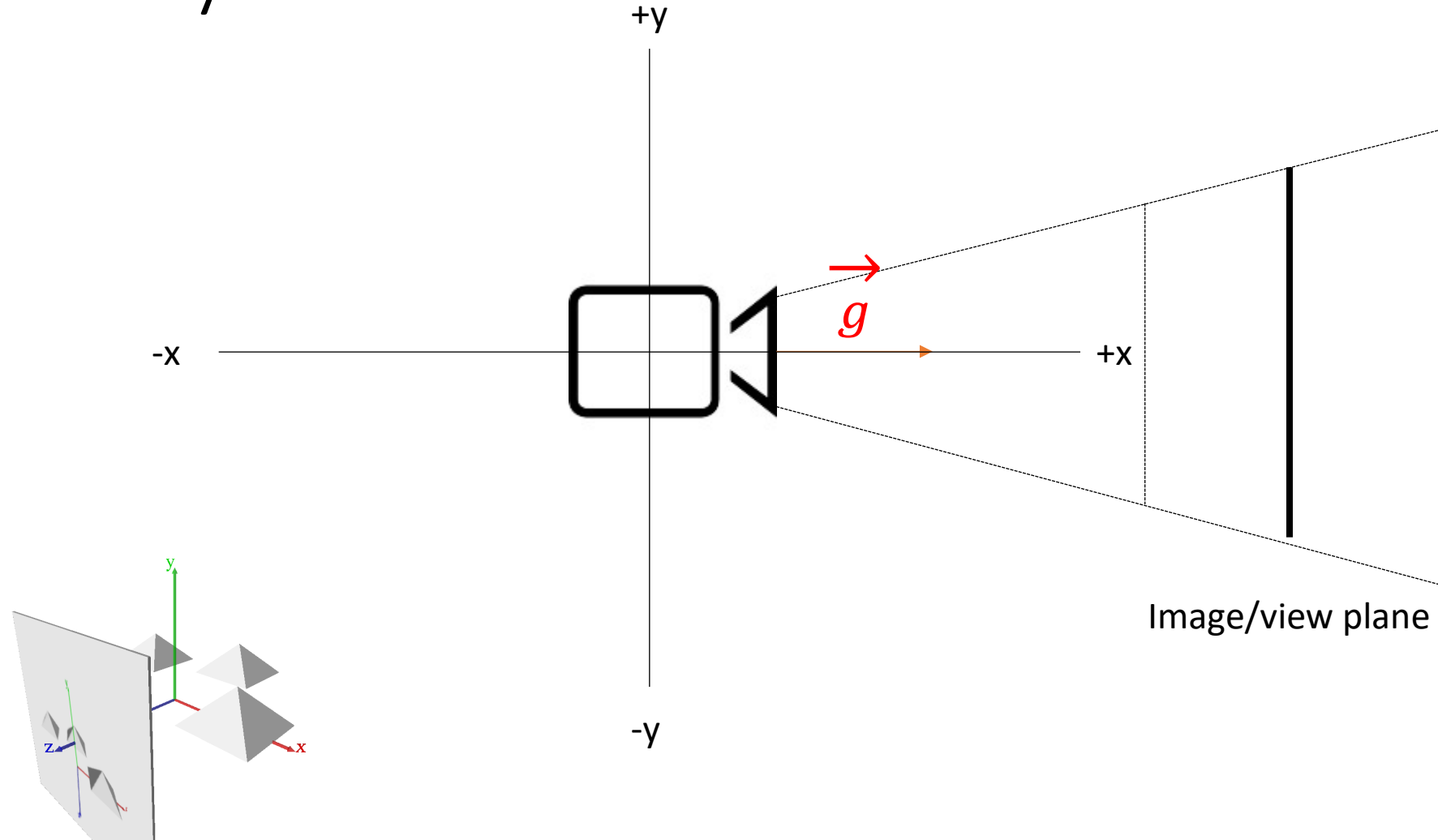
Transformacja kamery



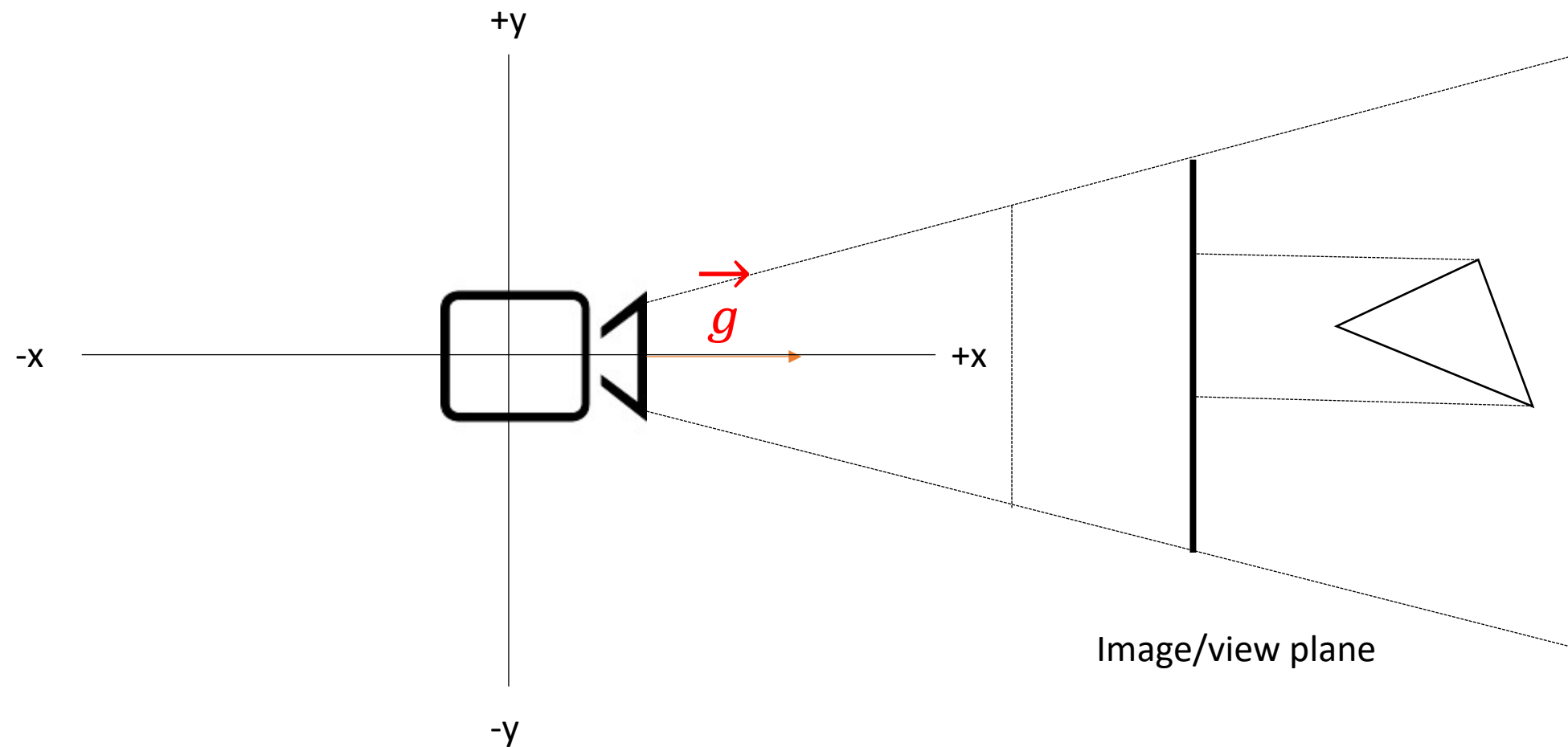
Płaszczyzna rzutowania



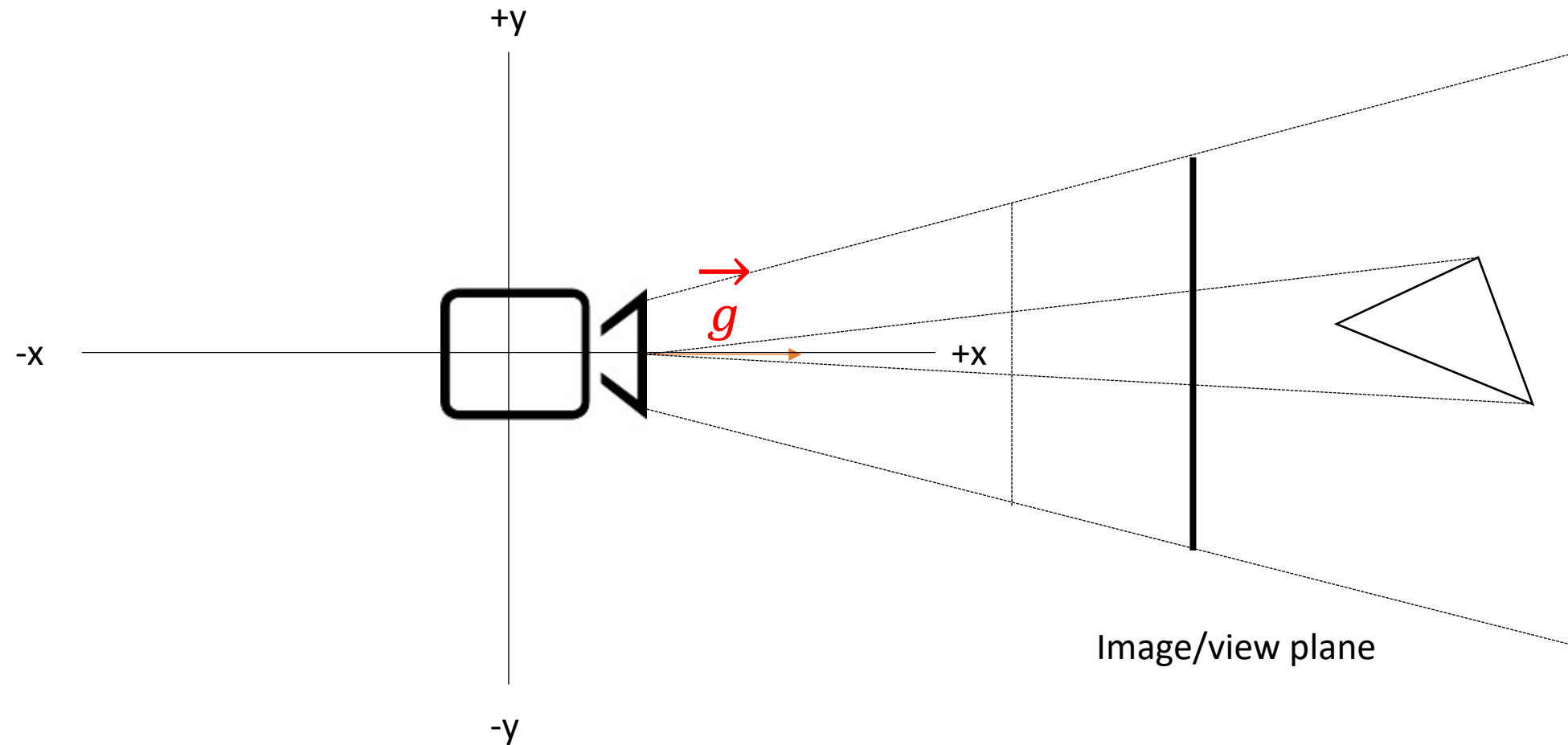
Płaszczyzna rzutowania



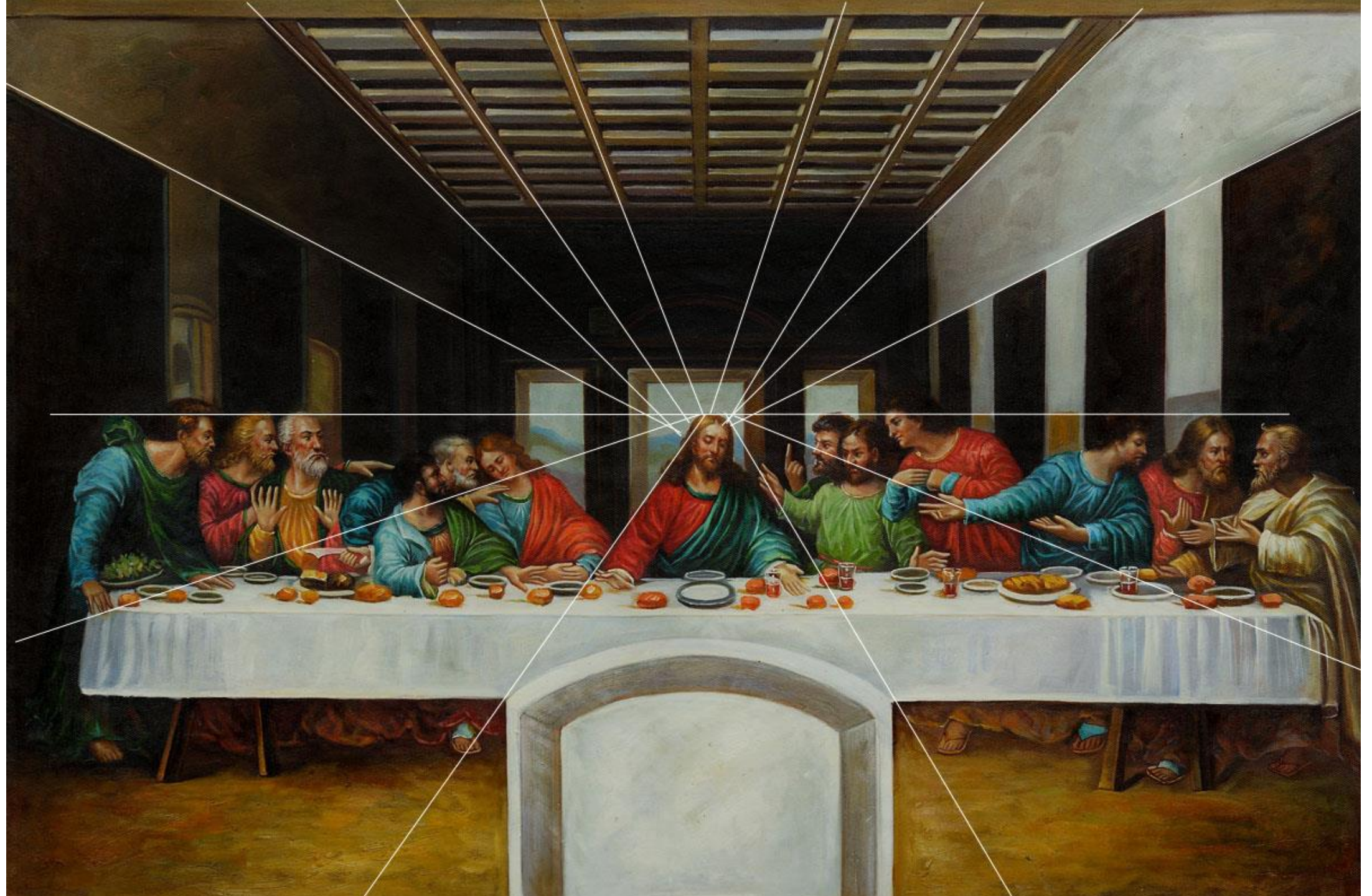
Rzutowanie równoległe



Rzutowanie perspektywiczne



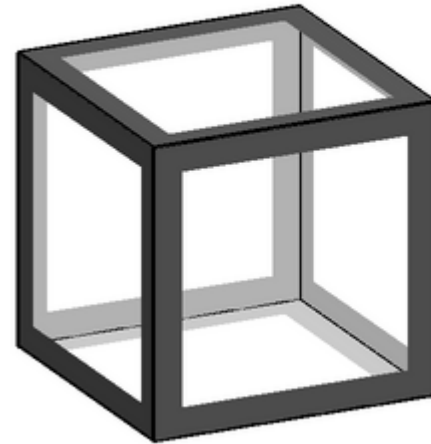




Rzutowania

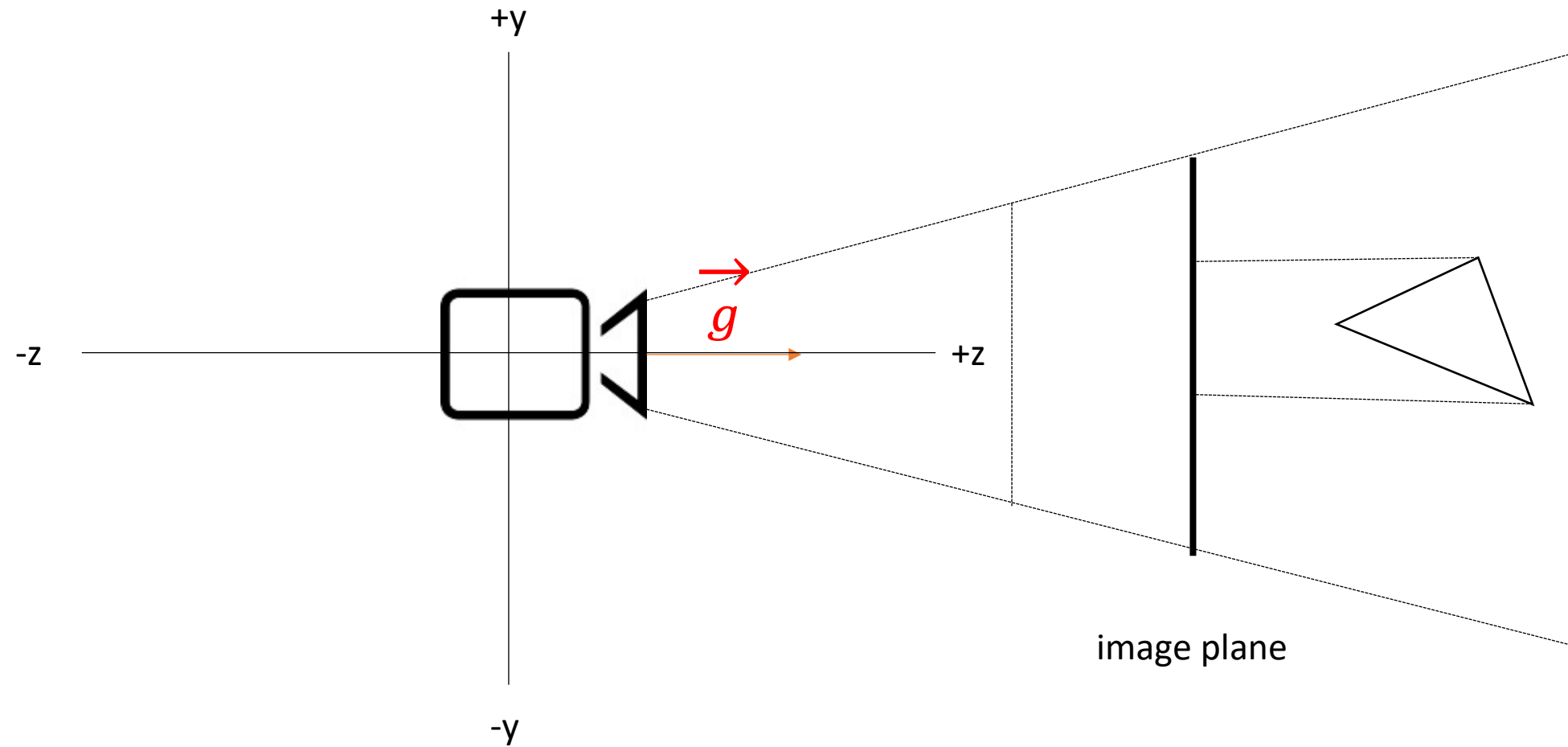


Perspektywicznie rzutowanie

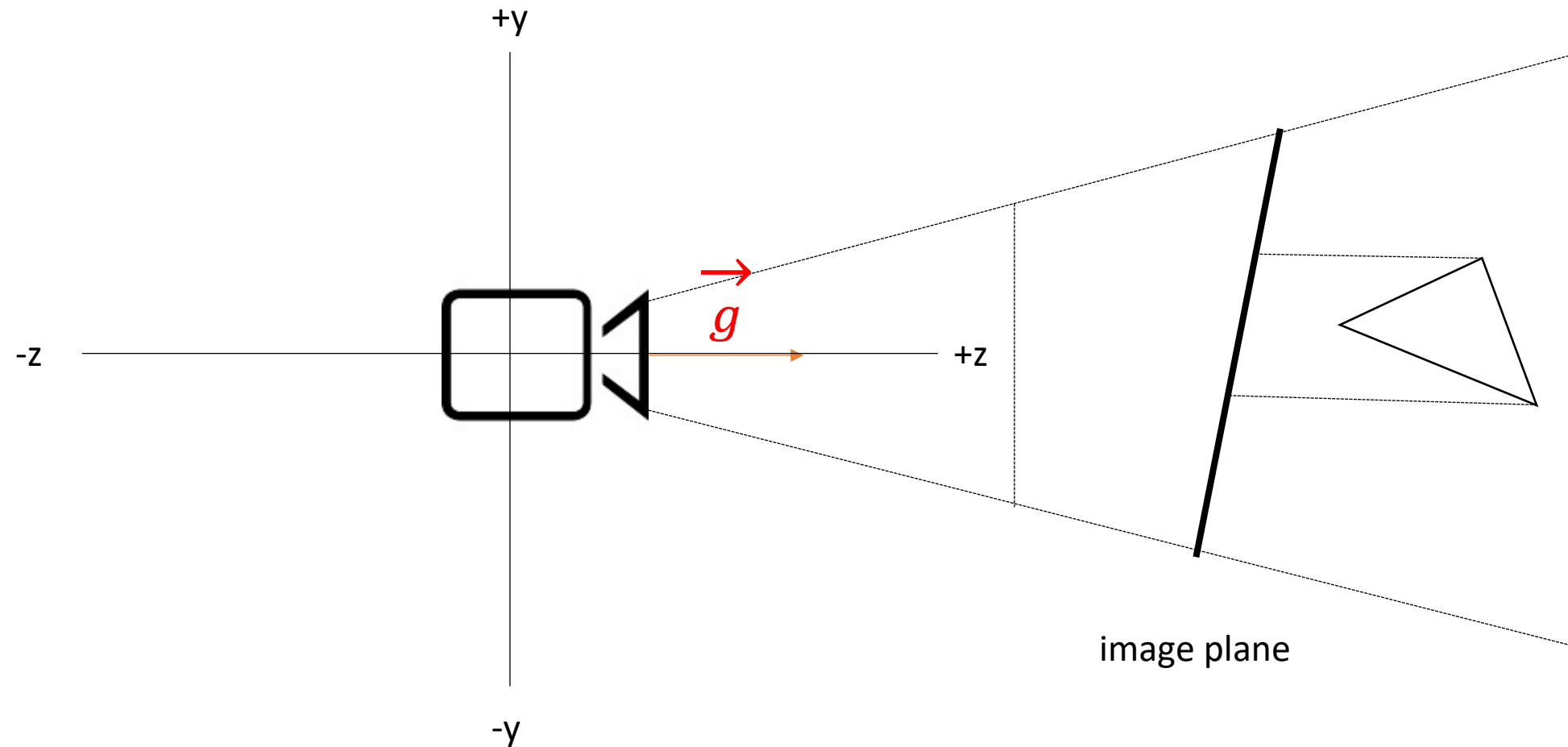


Ortograficznie rzutowanie

Rzutowanie ortograficzne

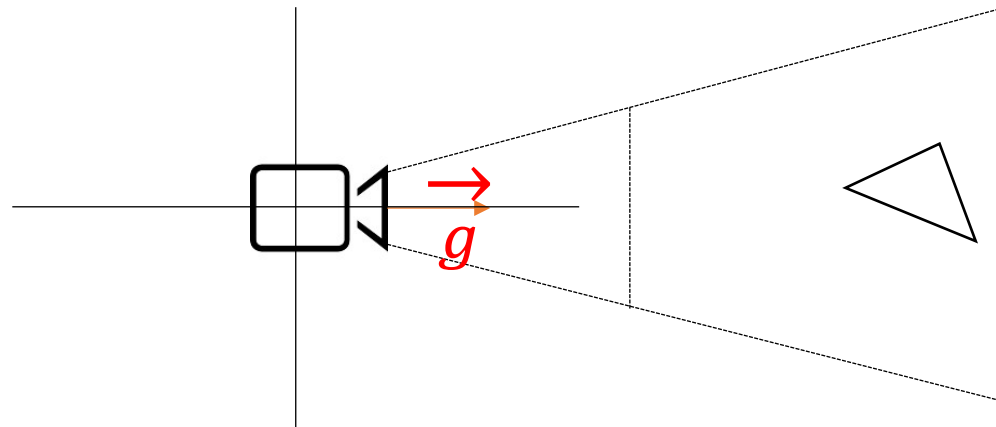


Rzutowanie **nie**-ortograficzne

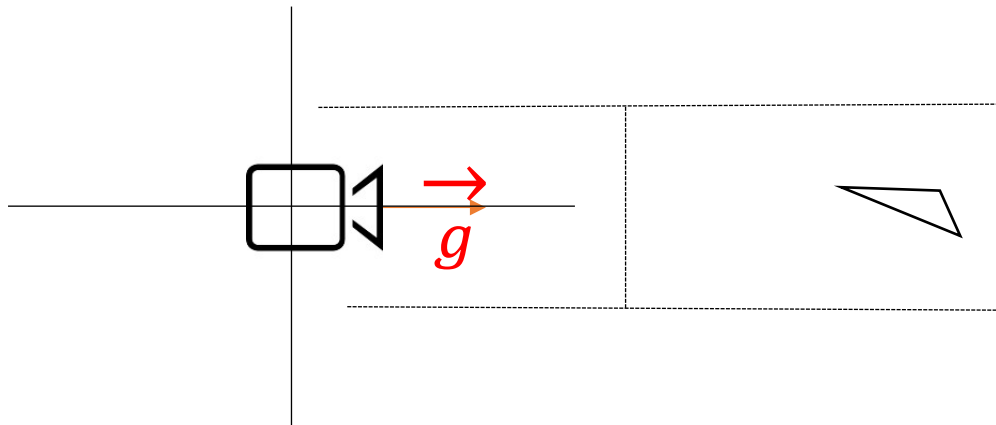


Rzutowanie ortograficzne

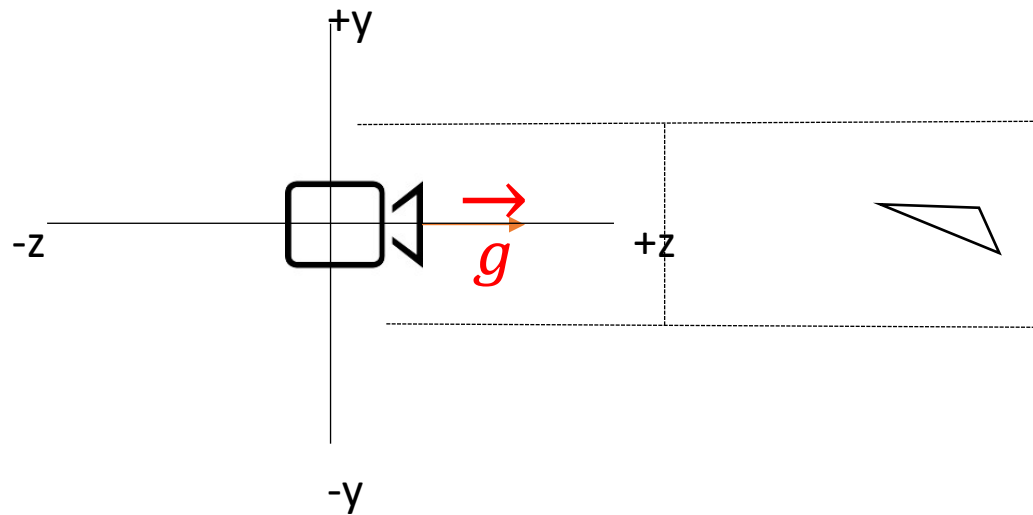
View frustum



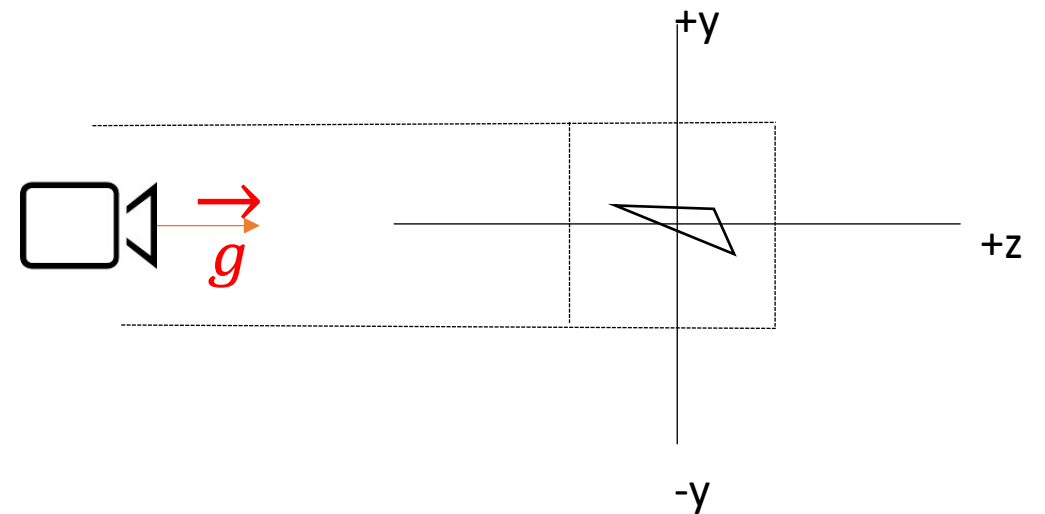
Orthographic view volume



Bryła kanoniczna

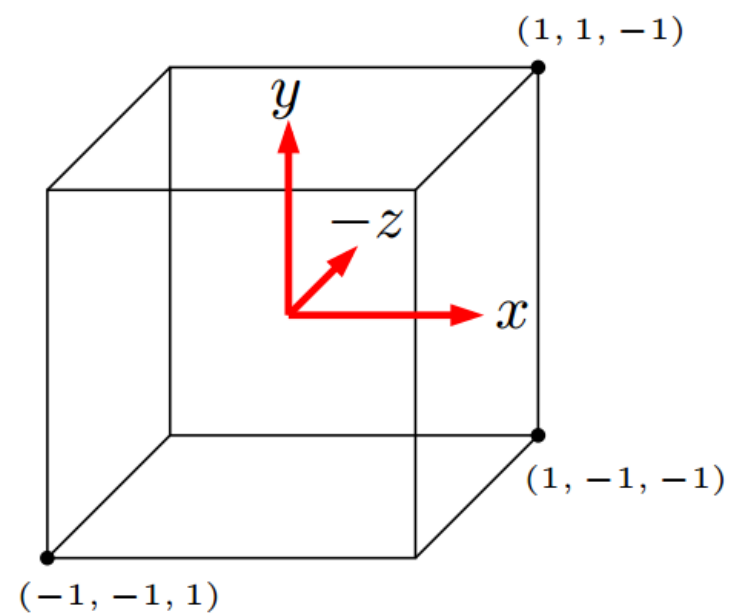


Orthographic view volume

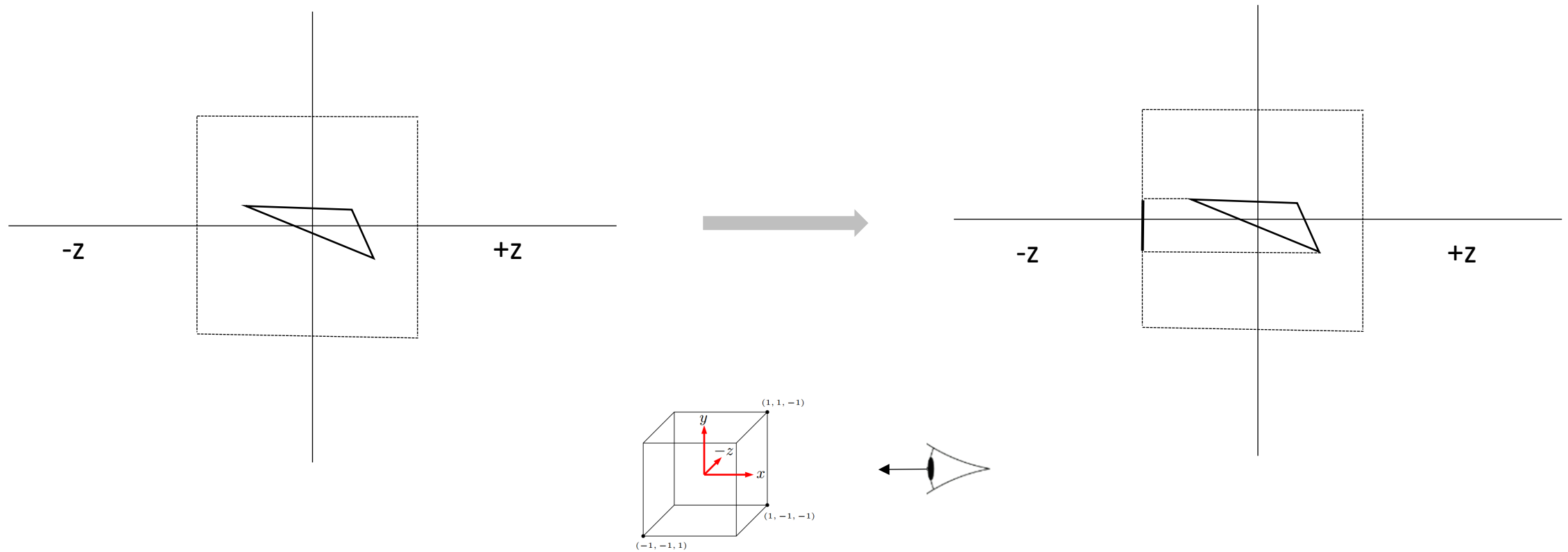


Canonical view volume

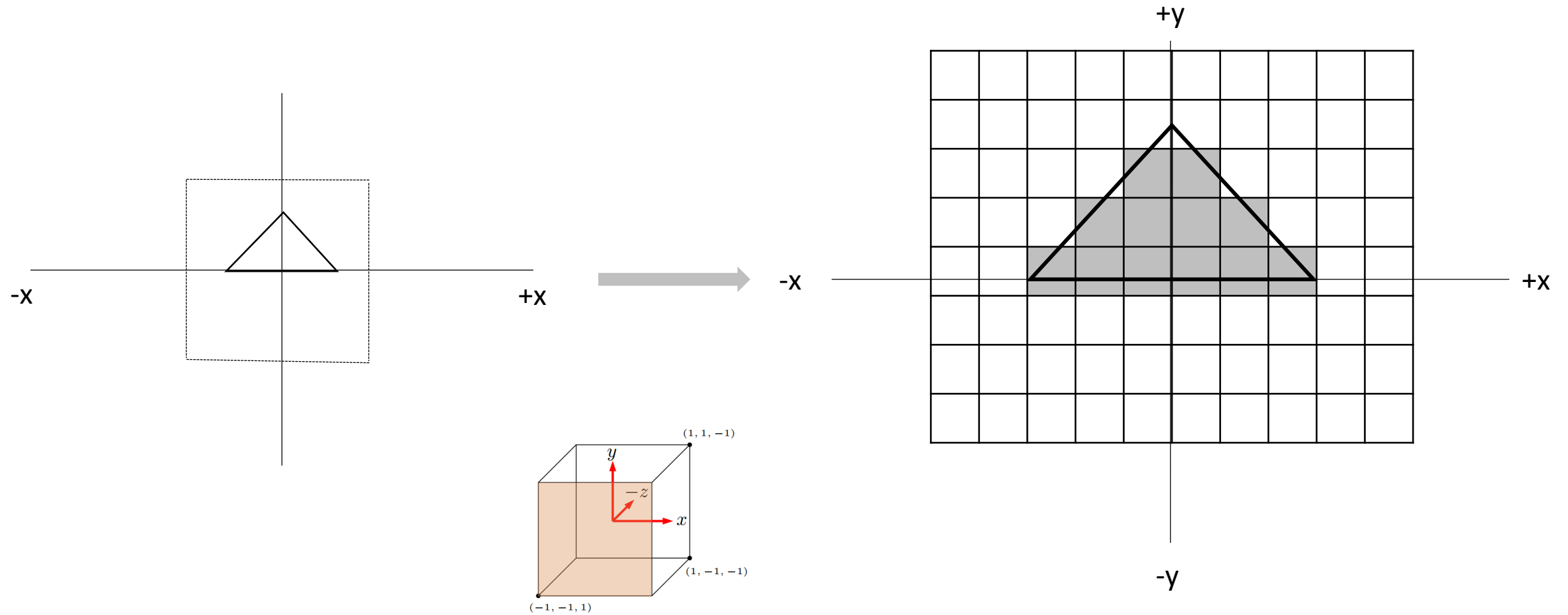
Bryła kanoniczna

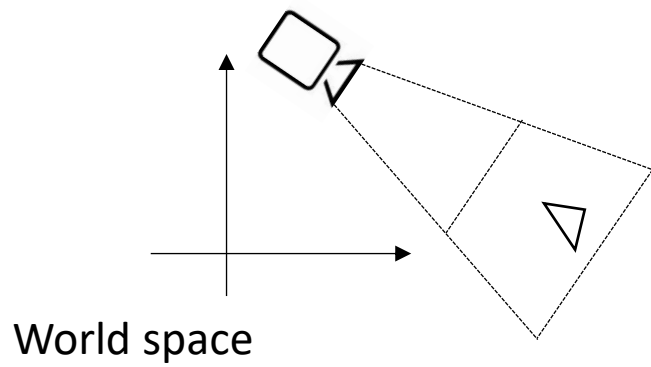


Viewport transformation



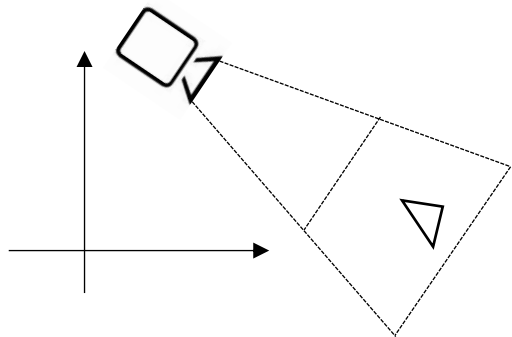
Viewport transformation



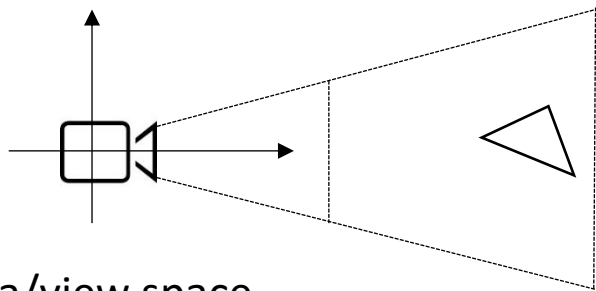


Potok graficzny

- Każdy krok będziemy chcieli wyrazić za pomocy **operacji macierzowej**. Cały potok może być wykonany za pomocy **jednej operacji macierzy**.

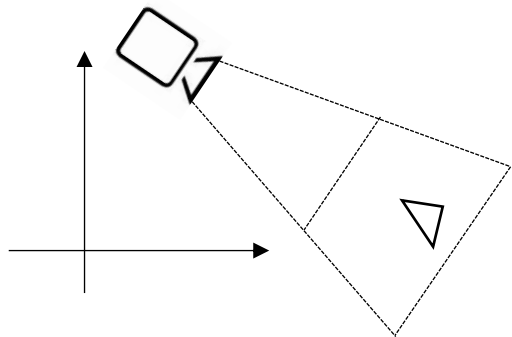


Potok graficzny



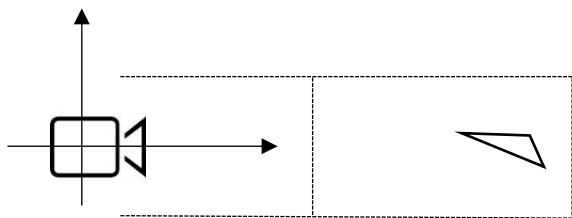
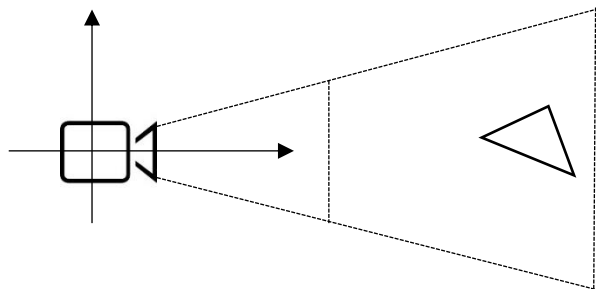
Camera/view space

- Każdy krok będziemy chcieli wyrazić za pomocy **operacji macierzowej**. Cały potok może być wykonany za pomocy **jednej operacji macierzy**.

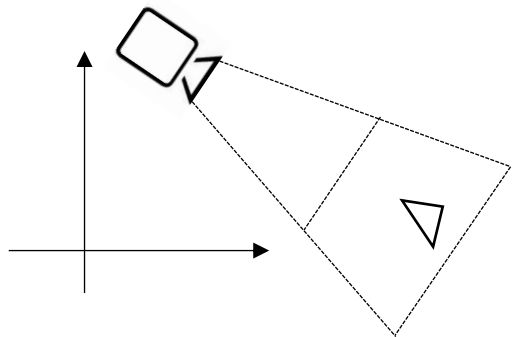


Potok graficzny

- Każdy krok będziemy chcieli wyrazić za pomocy **operacji macierzowej**. Cały potok może być wykonany za pomocy **jednej operacji macierzy**.

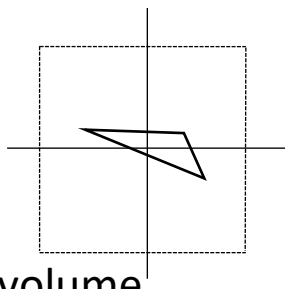
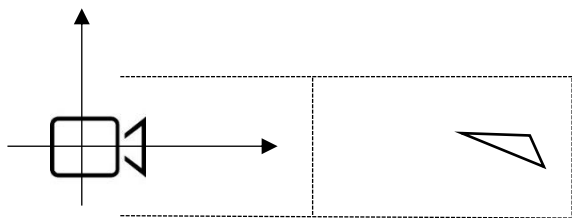
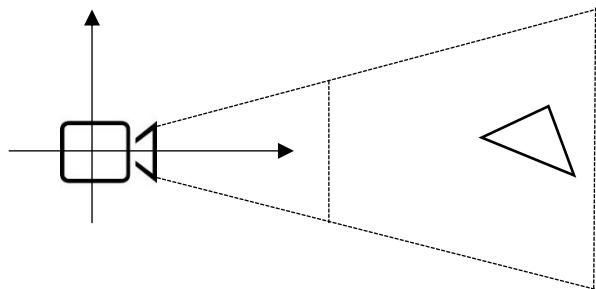


Orthographic view volume

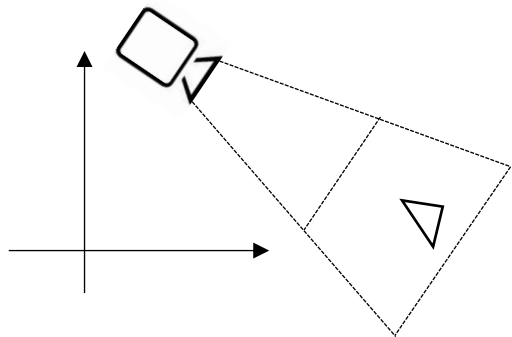


Potok graficzny

- Każdy krok będziemy chcieli wyrazić za pomocy **operacji macierzowej**. Cały potok może być wykonany za pomocy **jednej operacji macierzy**.

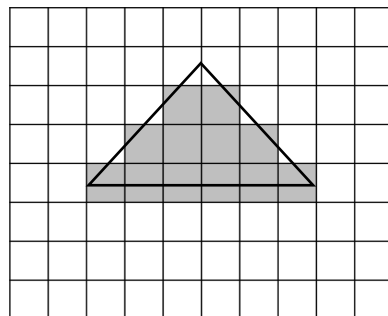
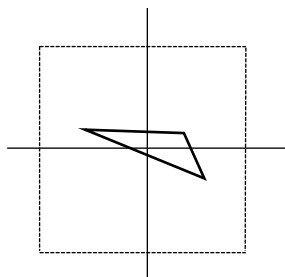
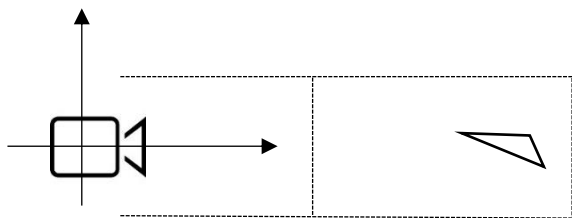
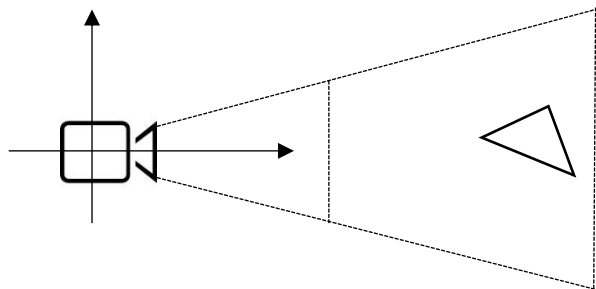


Canonical view volume



Potok graficzny

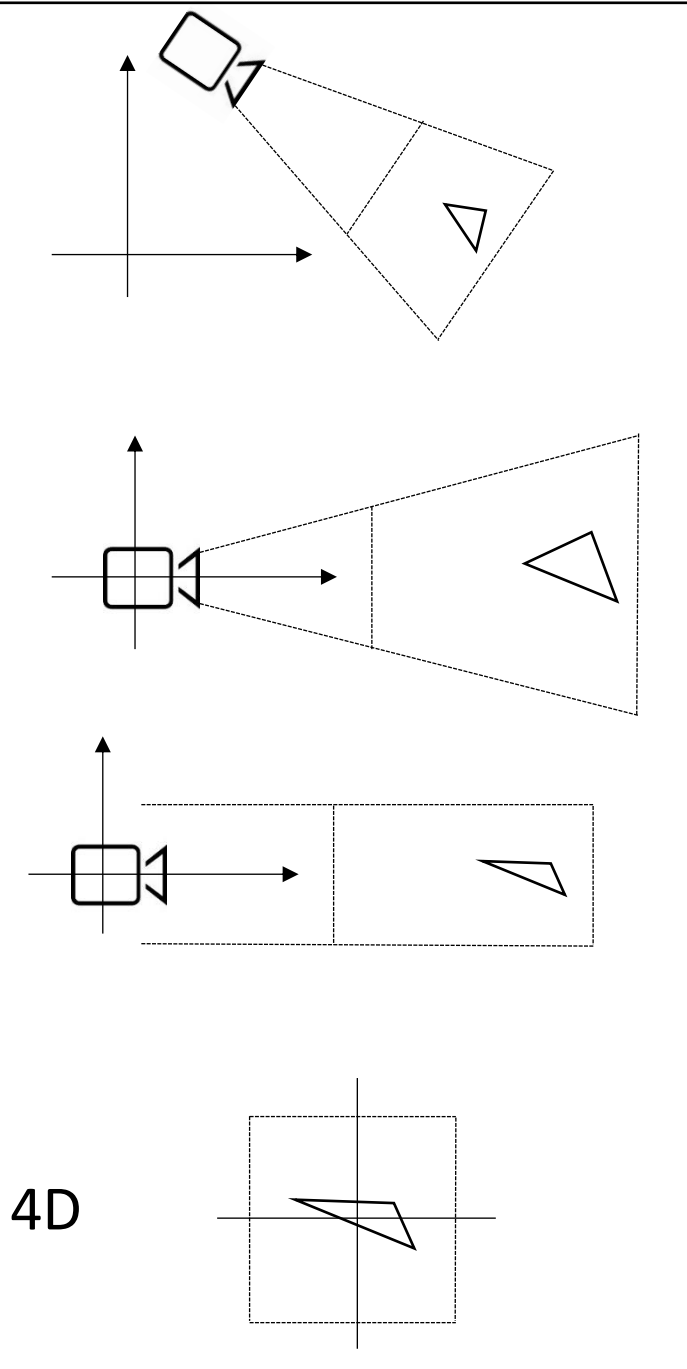
- Każdy krok będziemy chcieli wyrazić za pomocy **operacji macierzowej**. Cały potok może być wykonany za pomocy **jednej operacji macierzy**.



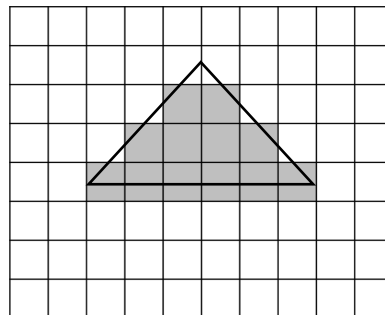
Window space

Potok graficzny

- Każdy krok będziemy chcieli wyrazić za pomocy **operacji macierzowej**. Cały potok może być wykonany za pomocy **jednej operacji macierzy**.

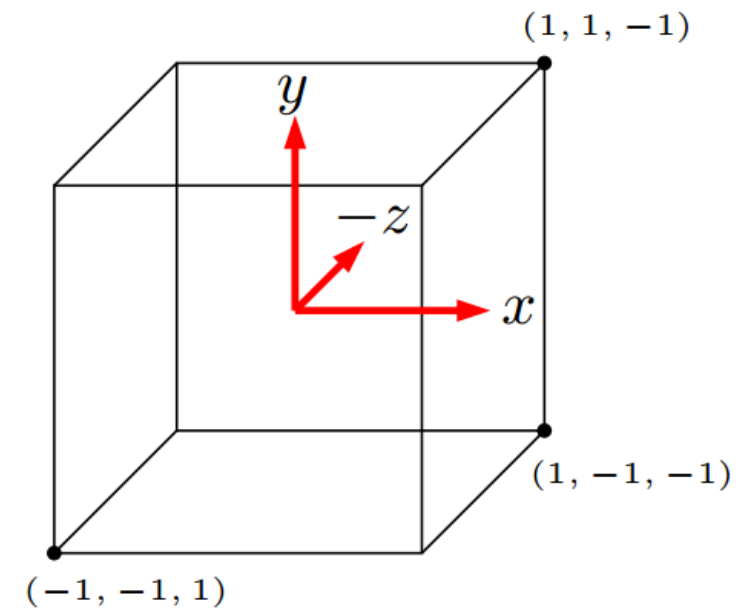


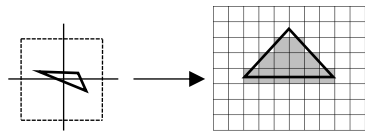
2D



Bryła kanoniczna

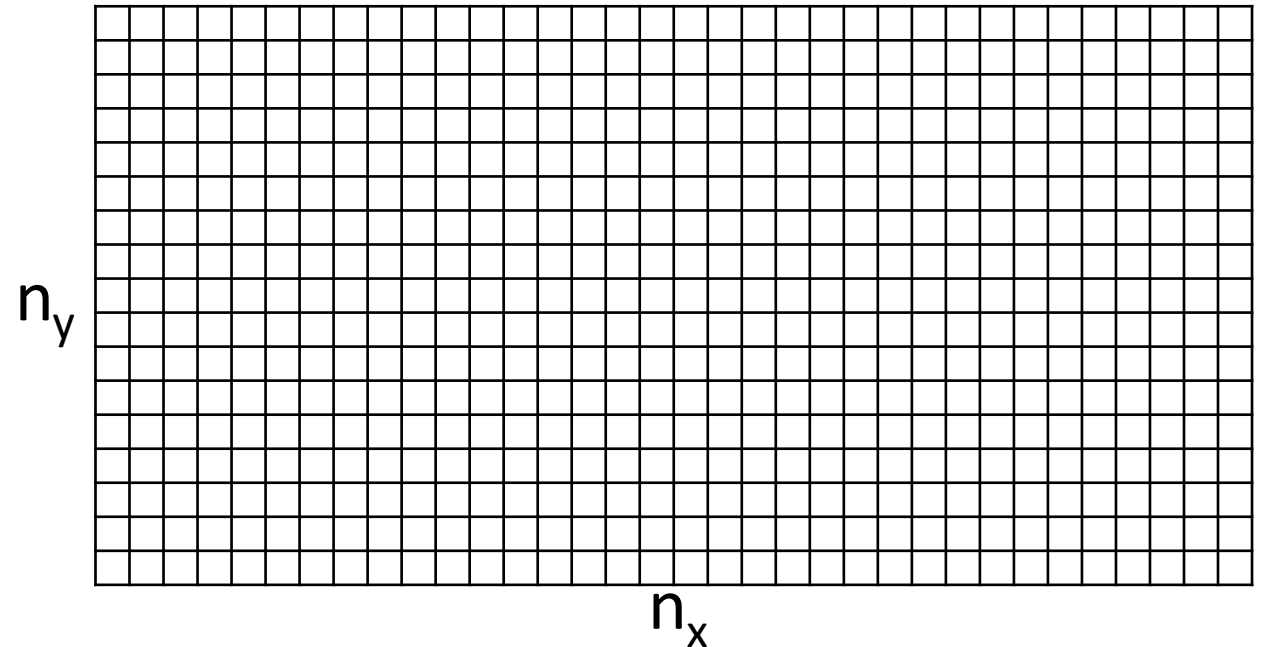
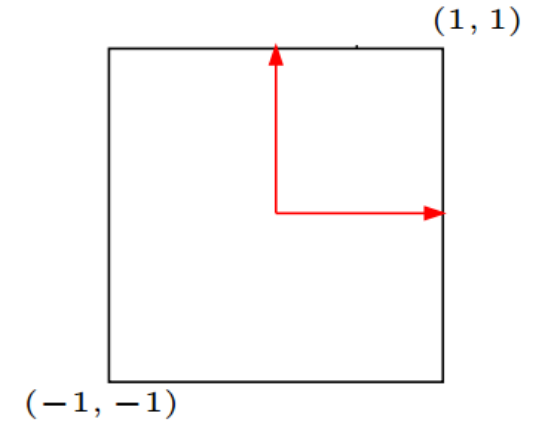
- Bryła kanoniczna to pudełko $2 \times 2 \times 2$ ze środkiem w początku
- Bryła widzenia została przekształcona do postaci takiego pudełka

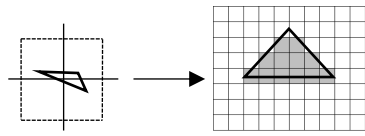




Przekształcenie bryły kanonicznej

- Musimy odwzorować kwadrat $[-1, 1]^2$ na czworokąt $[0, n_x] \times [0, n_y]$





Przekształcenie bryły kanonicznej

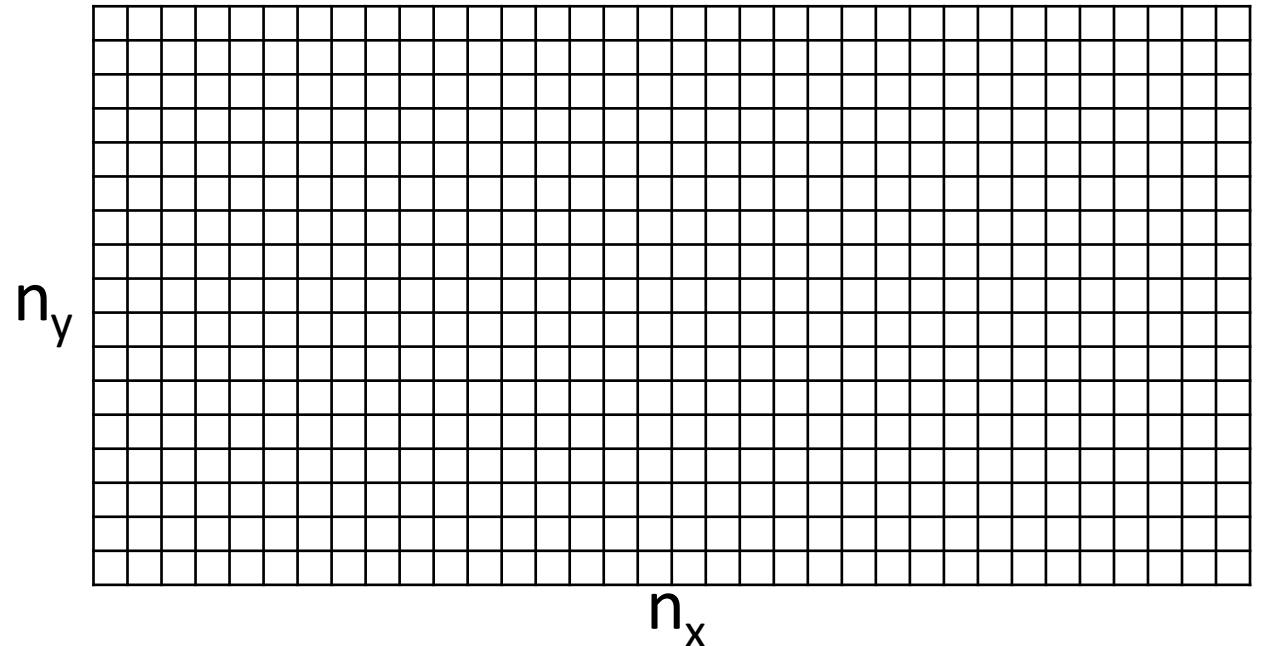
- Musimy odwzorować kwadrat $[-1, 1]^2$ na czworokąt $[0, n_x] \times [0, n_y]$
- Następująca macierz transformacji tego dokonuje:

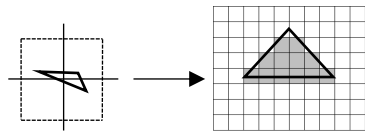
$$\begin{bmatrix} \frac{n_x}{2} & 0 & \frac{n_x}{2} \\ 0 & \frac{n_y}{2} & \frac{n_y}{2} \\ 0 & 0 & 1 \end{bmatrix}$$



- Musimy odwzorować kwadrat $[-1, 1]^2$ na czworokąt $[0, n_x] \times [0, n_y]$
- Następująca macierz transformacji tego dokonuje:

$$\begin{bmatrix} \frac{n_x}{2} & 0 & \frac{n_x}{2} \\ 0 & \frac{n_y}{2} & \frac{n_y}{2} \\ 0 & 0 & 1 \end{bmatrix}$$

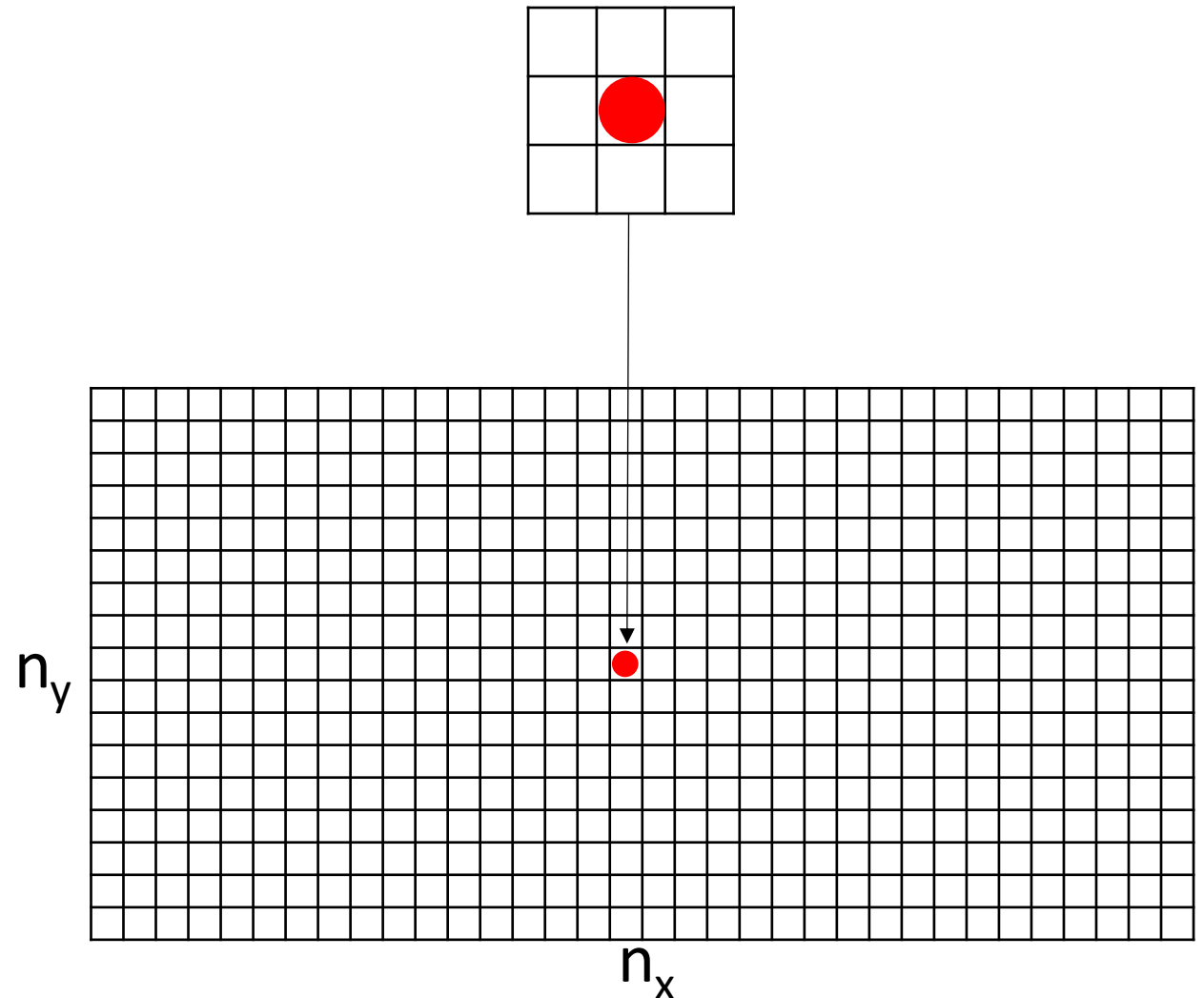


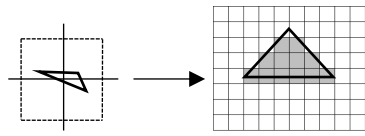


Przekształcenie bryły kanonicznej

- W praktyce piksele reprezentują współrzędne jednostki integer, czyli musimy przekształcać do $[-\frac{1}{2}, n_x - \frac{1}{2}] \times [-\frac{1}{2}, n_y - \frac{1}{2}]$
- Wtedy:

$$\begin{bmatrix} \frac{n_x}{2} & 0 & \frac{n_x}{2} - \frac{1}{2} \\ 0 & \frac{n_y}{2} & \frac{n_y}{2} - \frac{1}{2} \\ 0 & 0 & 1 \end{bmatrix}$$

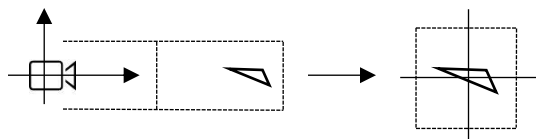




Przekształcenie bryły kanonicznej

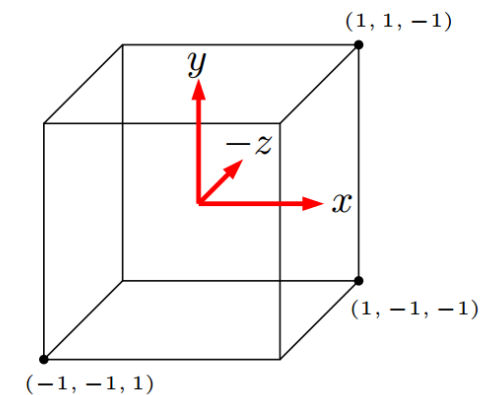
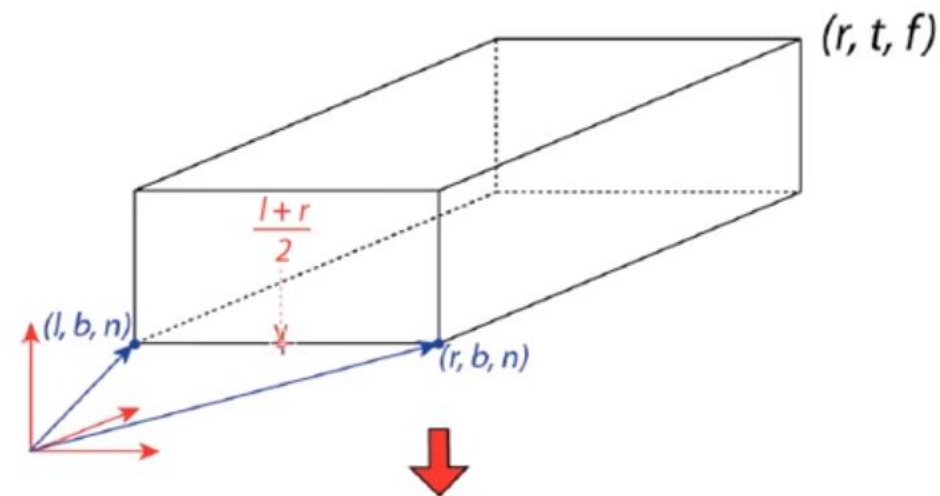
- Rzutowanie dokonaliśmy przy „odrzuconiu” współrzędnej z
- Ale chcemy nadal być w stanie konkatelować naszą macierz z innymi, czyli ostateczna postać naszej macierzy to 4D:

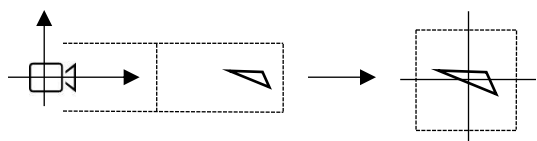
$$M_{vp} = \begin{bmatrix} \frac{n_x}{2} & 0 & 0 & \frac{n_x}{2} - \frac{1}{2} \\ 0 & \frac{n_y}{2} & 0 & \frac{n_y}{2} - \frac{1}{2} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



Ortograficzna bryła widzenia

- Najpierw musimy środek do początku przenieść

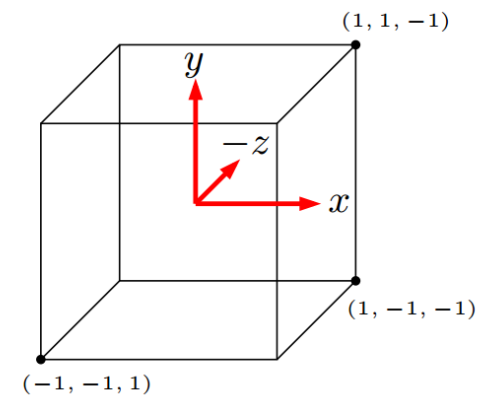
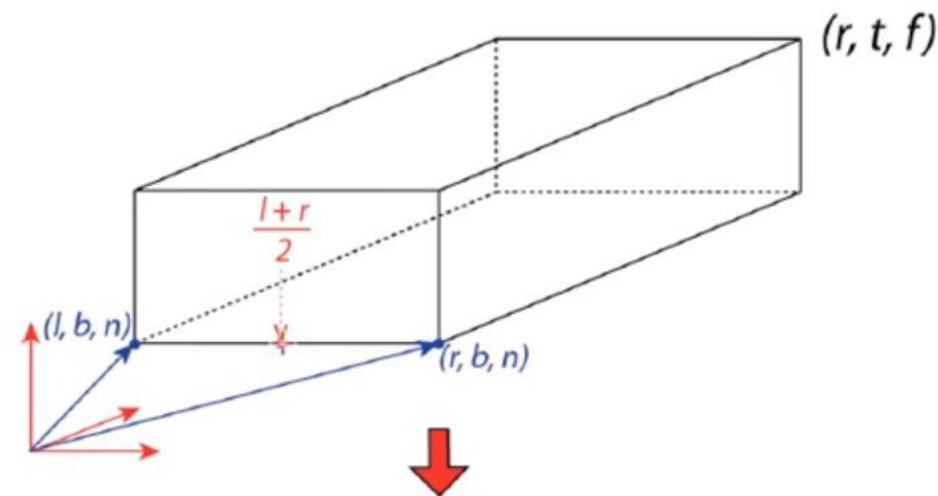


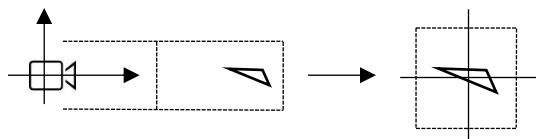


Ortograficzna bryła widzenia

- Najpierw musimy środek do początku przenieść

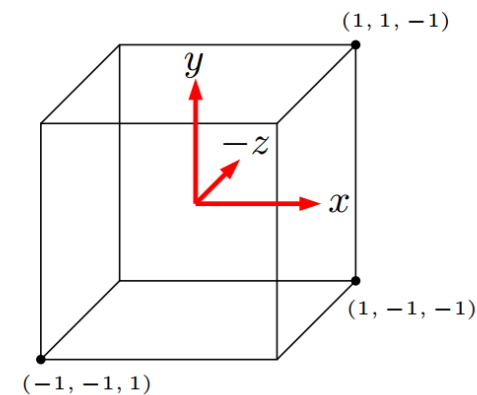
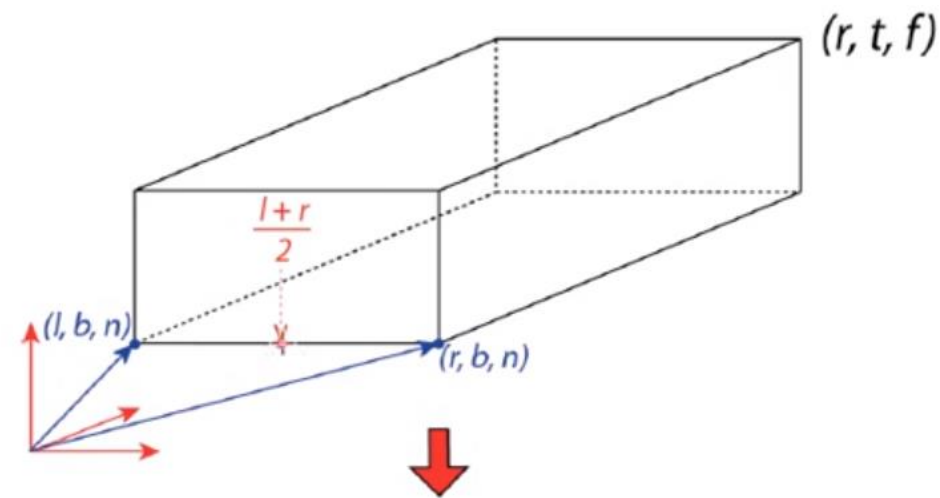
$$\bullet \begin{bmatrix} 1 & 0 & 0 & -\frac{l+r}{2} \\ 0 & 1 & 0 & -\frac{b+t}{2} \\ 0 & 0 & 1 & -\frac{n+f}{2} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

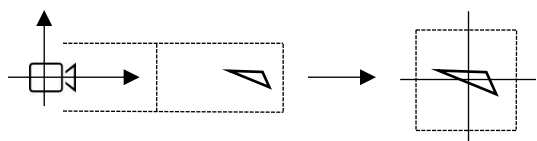




Ortograficzna bryła widzenia

- Potem musimy wszystko skalować do $[-1, 1]$:

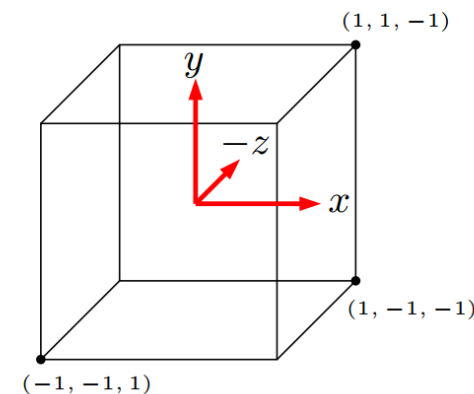
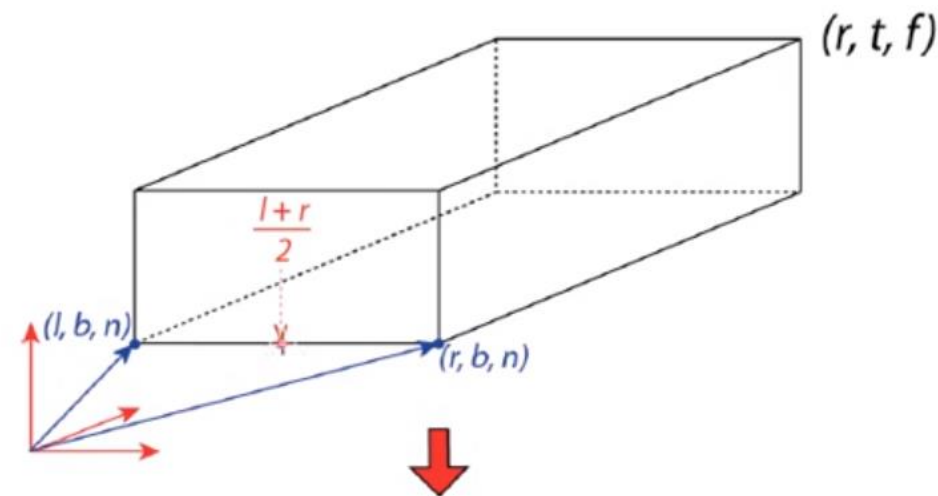


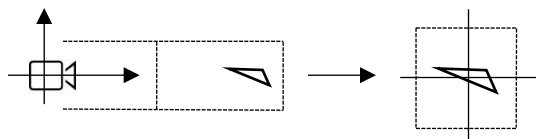


Ortograficzna bryła widzenia

- Potem musimy wszystko skalować do $[-1, 1]$:

$$\bullet \begin{bmatrix} -\frac{2}{r-l} & 0 & 0 & 0 \\ 0 & -\frac{2}{t-b} & 0 & 0 \\ 0 & 0 & -\frac{2}{n-f} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

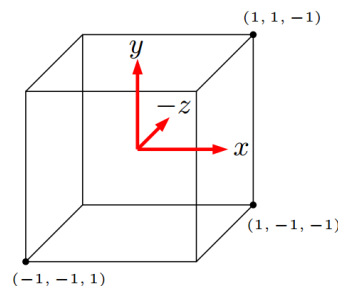


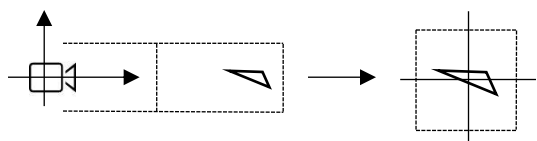


Ortograficzna bryła widzenia

- Konkatenujemy macierze żeby uzyskać jedną macierz transformacji do ortograficznej bryły widzenia:

$$\bullet \quad M_{orth} = \begin{bmatrix} -\frac{2}{r-l} & 0 & 0 & 0 \\ 0 & -\frac{2}{t-b} & 0 & 0 \\ 0 & 0 & -\frac{2}{n-f} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -\frac{l+r}{2} \\ 0 & 1 & 0 & -\frac{b+t}{2} \\ 0 & 0 & 1 & -\frac{n+f}{2} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

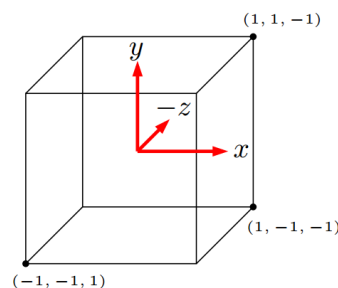




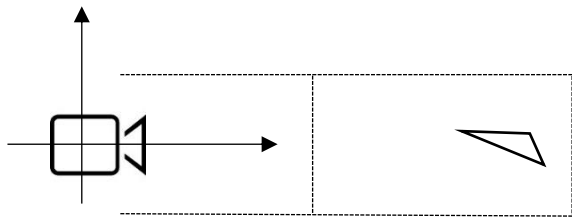
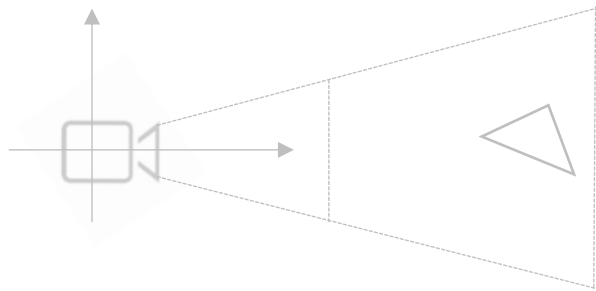
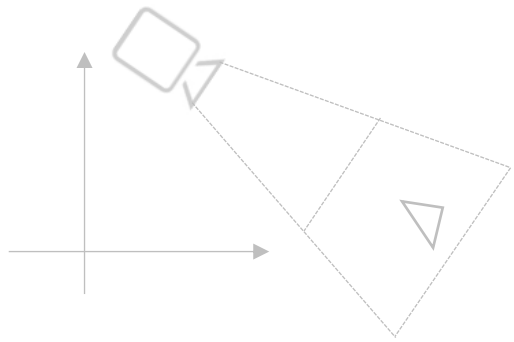
Ortograficzna bryła widzenia

- Konkatenujemy macierze żeby uzyskać jedną macierz transformacji do ortograficznej bryły widzenia:

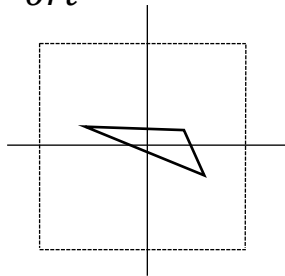
$$\bullet \quad M_{orth} = \begin{bmatrix} -\frac{2}{r-l} & 0 & 0 & 0 \\ 0 & -\frac{2}{t-b} & 0 & 0 \\ 0 & 0 & -\frac{2}{n-f} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -\frac{l+r}{2} \\ 0 & 1 & 0 & -\frac{b+t}{2} \\ 0 & 0 & 1 & -\frac{n+f}{2} \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \frac{2}{r-l} & 0 & 0 & -\frac{l+r}{r-l} \\ 0 & \frac{2}{t-b} & 0 & -\frac{b+t}{t-b} \\ 0 & 0 & \frac{2}{n-f} & -\frac{n+f}{n-f} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



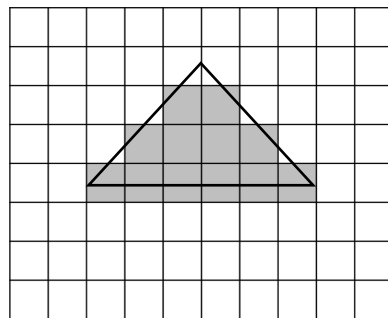
Potok graficzny



M_{ort}



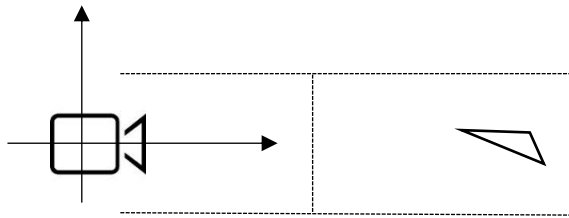
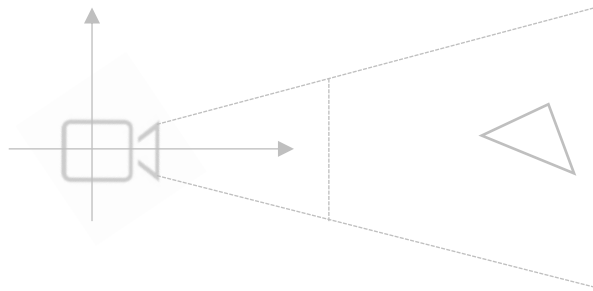
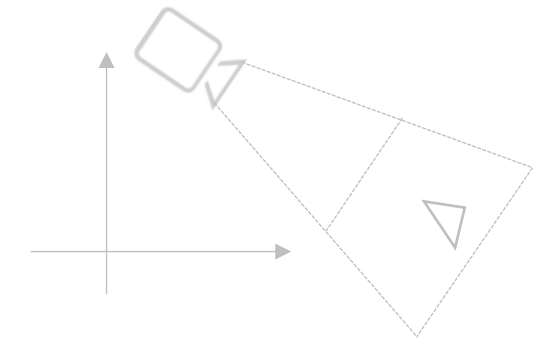
M_{vp}



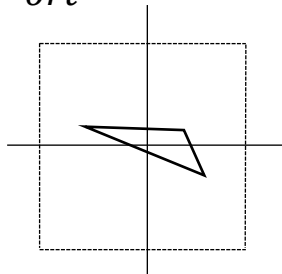
Potok graficzny

- Transformacja:

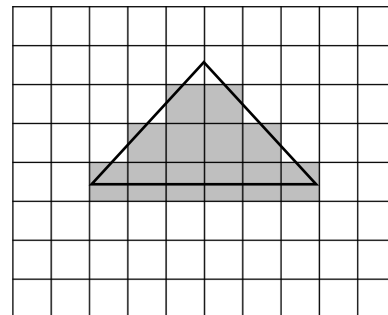
$$\bullet \begin{bmatrix} x_{piksel} \\ y_{piksel} \\ z_{kanoniczny} \\ 1 \end{bmatrix} = M_{vp} M_{ort} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

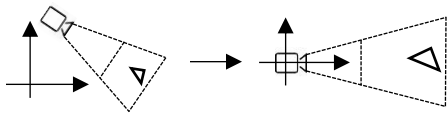


M_{ort}



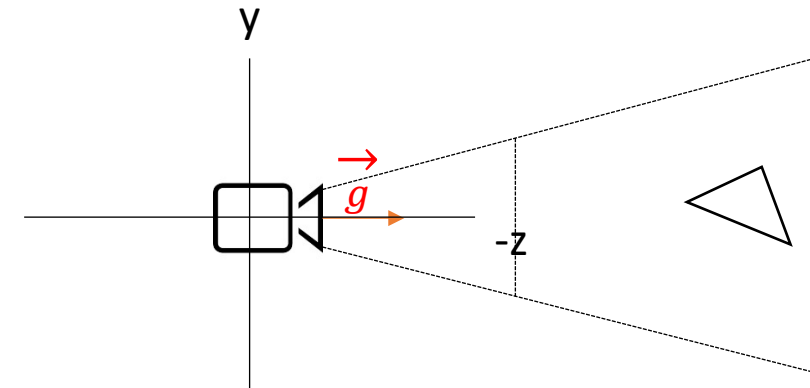
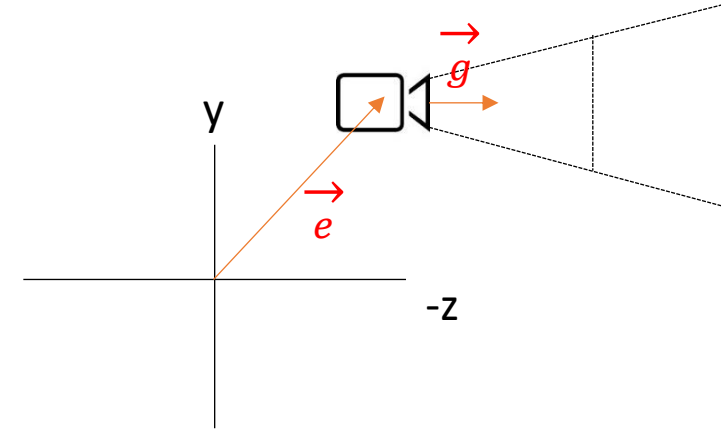
M_{vp}

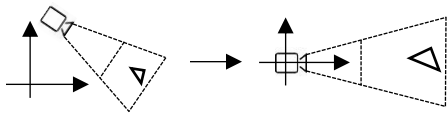




Wyrównanie systemów współrzędnych

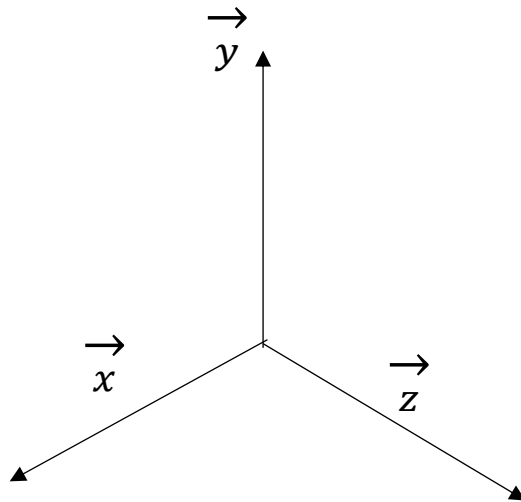
- World space:
 - Wektory, $\vec{x}$, $\vec{y}$, i $\vec{z}$
- Camera space:
 - Wektory $\vec{e}$, i $\vec{g}$



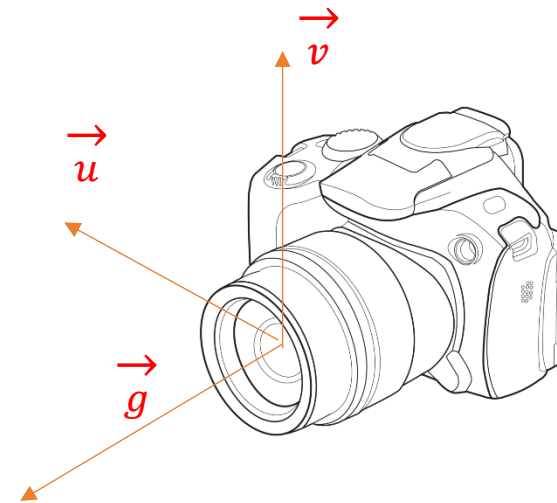


Wyrównanie systemów współrzędnych

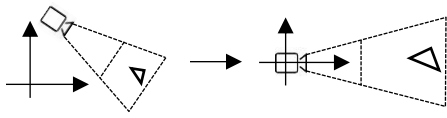
- Żeby jedną przestrzeń na drugą przestrzeń przekształcić potrzebujemy jednego systemu współrzędnych dla obydwóch przestrzeni



Układ świata



Układ kamery



Wyrównanie systemów współrzędnych

- Pierwszy wektor bazowy:

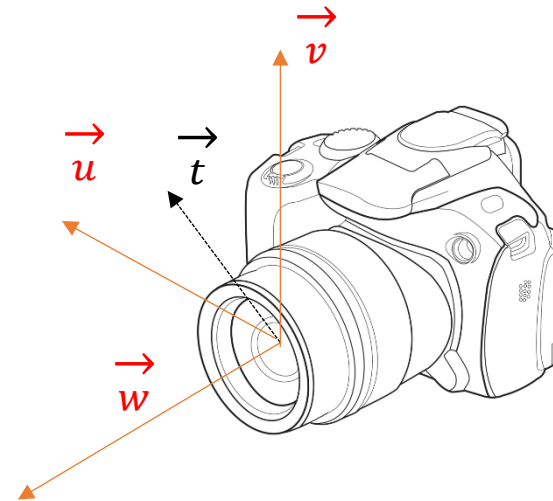
- $\vec{t} \times \vec{g} = \vec{u}$

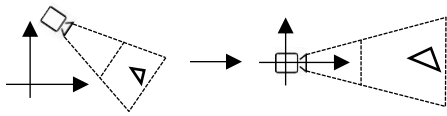
- Drugi wektor bazowy:

- $\vec{g} \times \vec{u} = \vec{v}$

- Trzeci wektor bazowy:

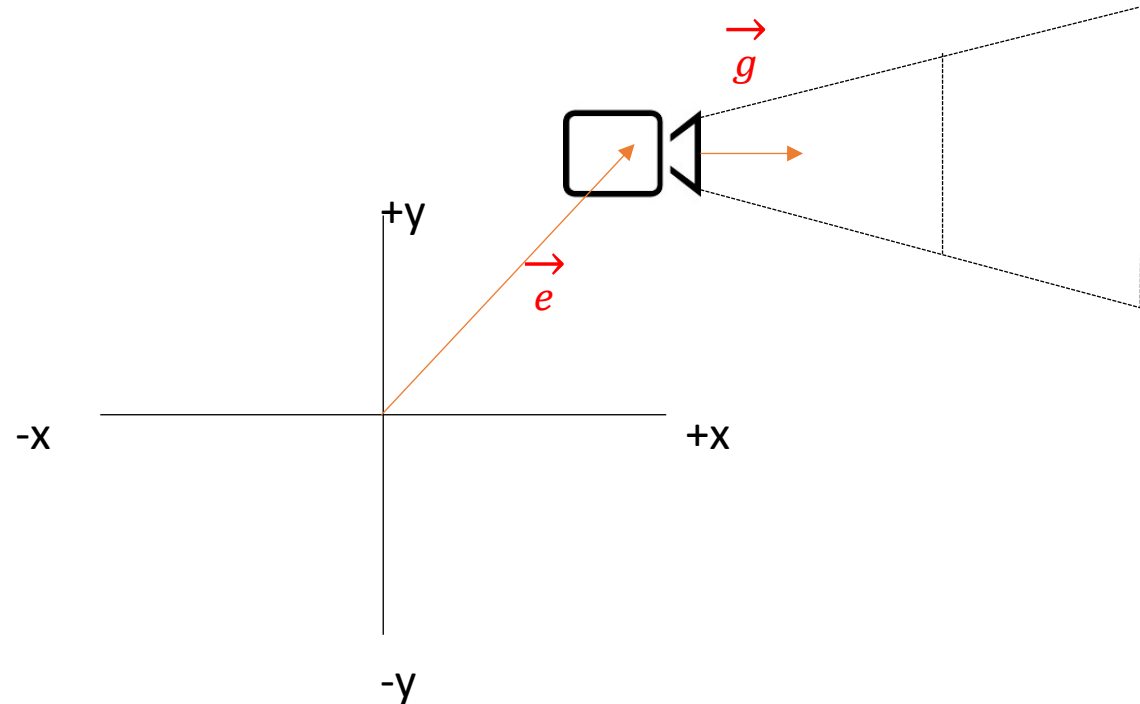
- $-\vec{g} := \vec{w}$

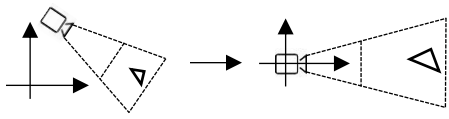




Wyrównanie systemów współrzędnych

- Wyrównać początki naszych systemów współrzędnych
- Wyrównać wektory bazowe

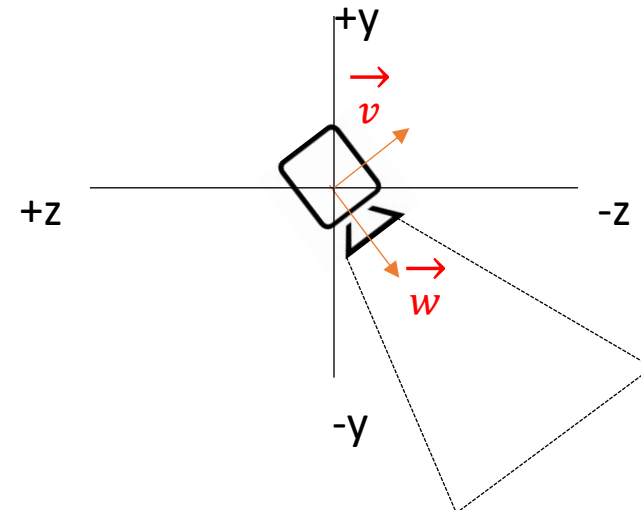
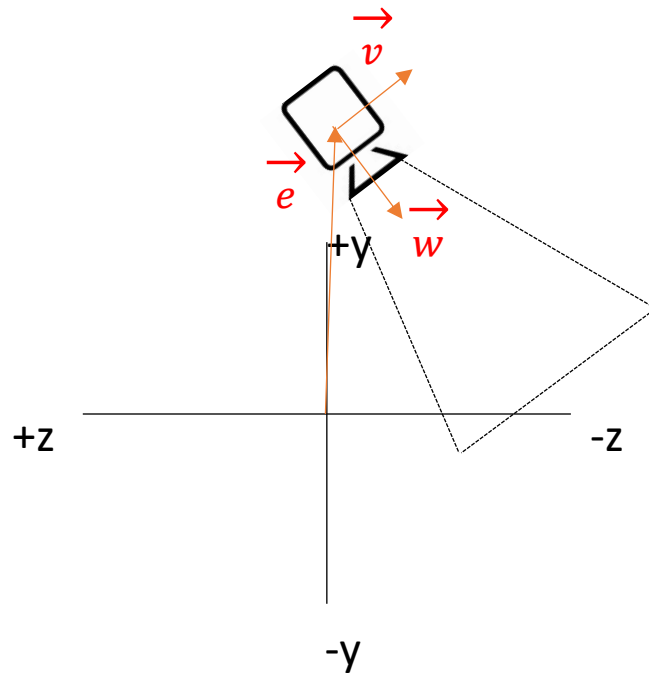


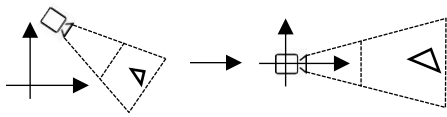


Wyrównanie systemów współrzędnych

- Wyrównanie początków przez translacje:

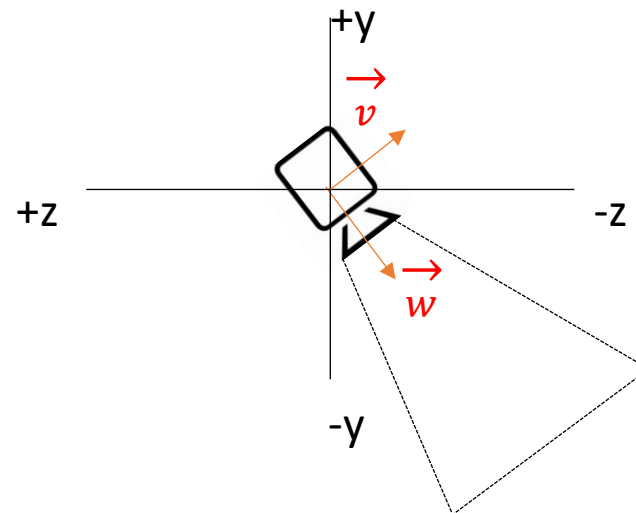
$$\begin{bmatrix} 1 & 0 & 0 & -e_x \\ 0 & 1 & 0 & -e_y \\ 0 & 0 & 1 & -e_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

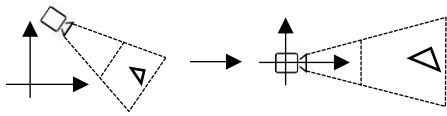




Wyrównanie systemów współrzędnych

- Rotacja w przeciwnym kierunku jest trywialna:
$$\begin{bmatrix} u_x & v_x & w_x & 0 \\ u_y & v_y & w_y & 0 \\ u_z & v_z & w_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

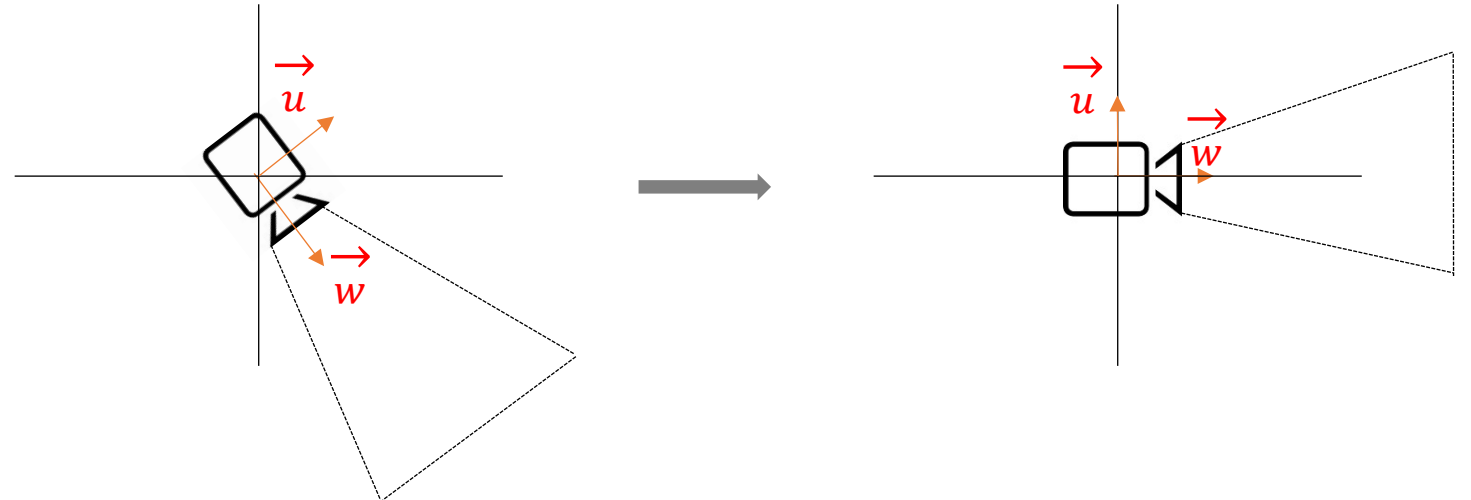


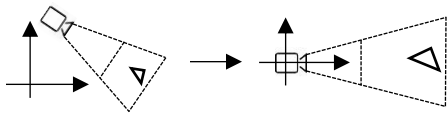


Wyrównanie systemów współrzędnych

- Macierz odwrotna w przypadku ortonormalnej bazy to transpozycja tej macierzy (wektory u, v, w muszą być znormalizowane):

$$\bullet \begin{bmatrix} u_x & u_y & u_z & 0 \\ v_x & v_y & v_z & 0 \\ w_x & w_y & w_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



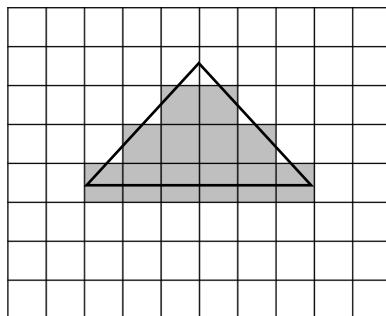
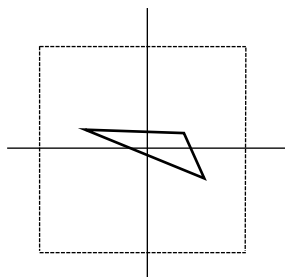
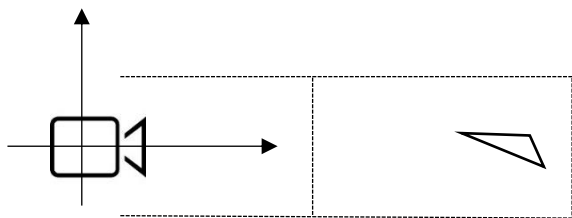
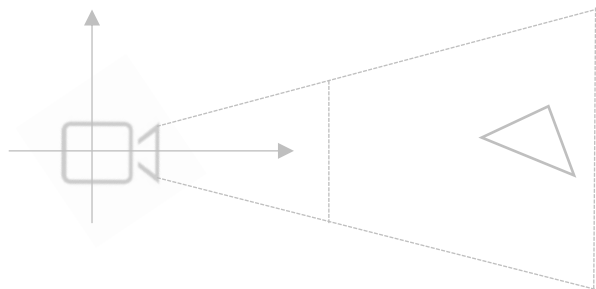
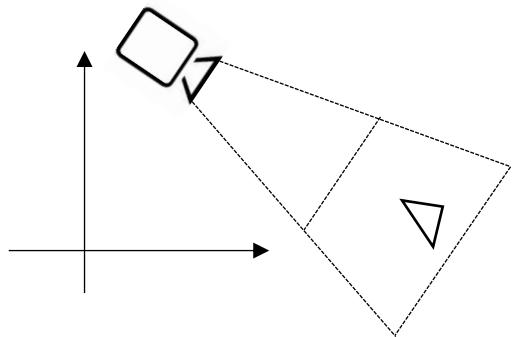


Wyrównanie systemów współrzędnych

- Macierz transformacji z world space na view space jest wtedy:

$$M_{cam} = \begin{bmatrix} u_x & u_y & u_z & 0 \\ v_x & v_y & v_z & 0 \\ w_x & w_y & w_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -e_x \\ 0 & 1 & 0 & -e_y \\ 0 & 0 & 1 & -e_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rzutowanie ortograficzne



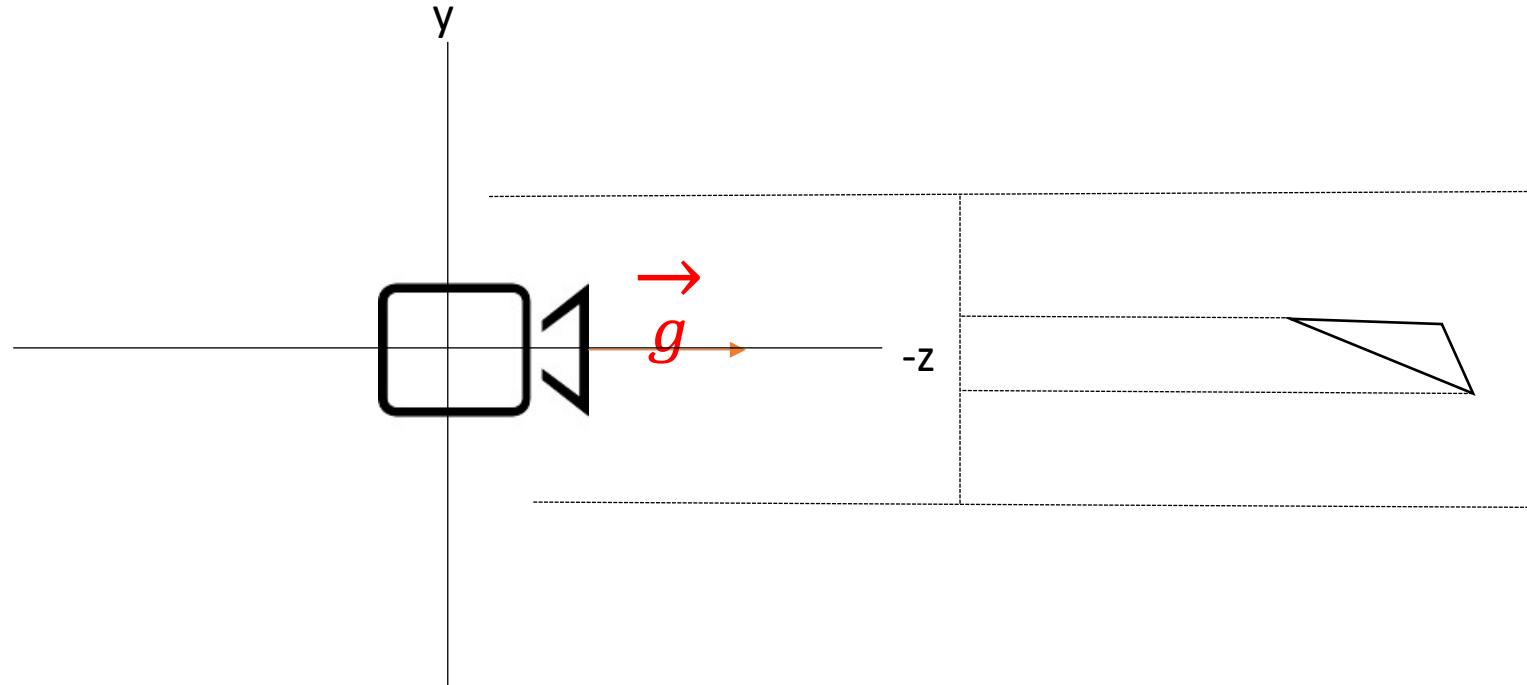
$$\begin{bmatrix} x_{piksel} \\ y_{piksel} \\ z_{kanoniczny} \\ 1 \end{bmatrix} = M_{vp} M_{orth} M_{cam} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Rzutowanie równoległe



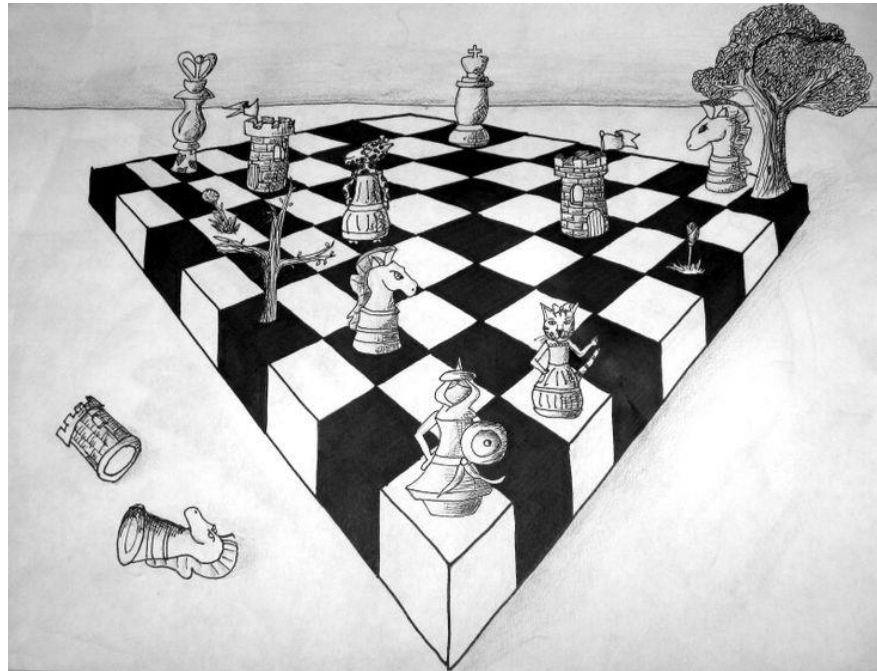
The Simpsons Tapped Out

Rzutowanie równoległe



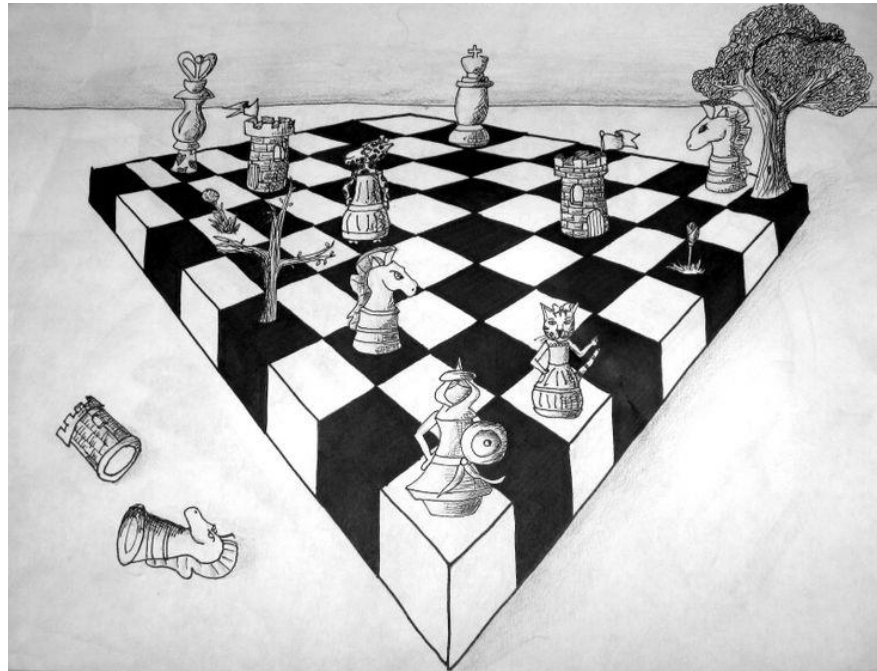
Rzutowanie perspektywiczne

- Co jest perspektywa?

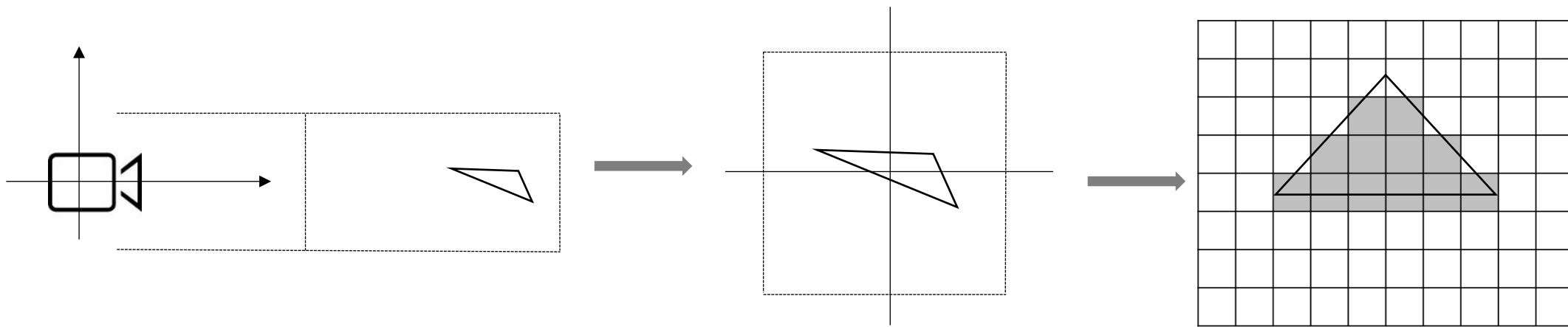


Rzutowanie perspektywiczne

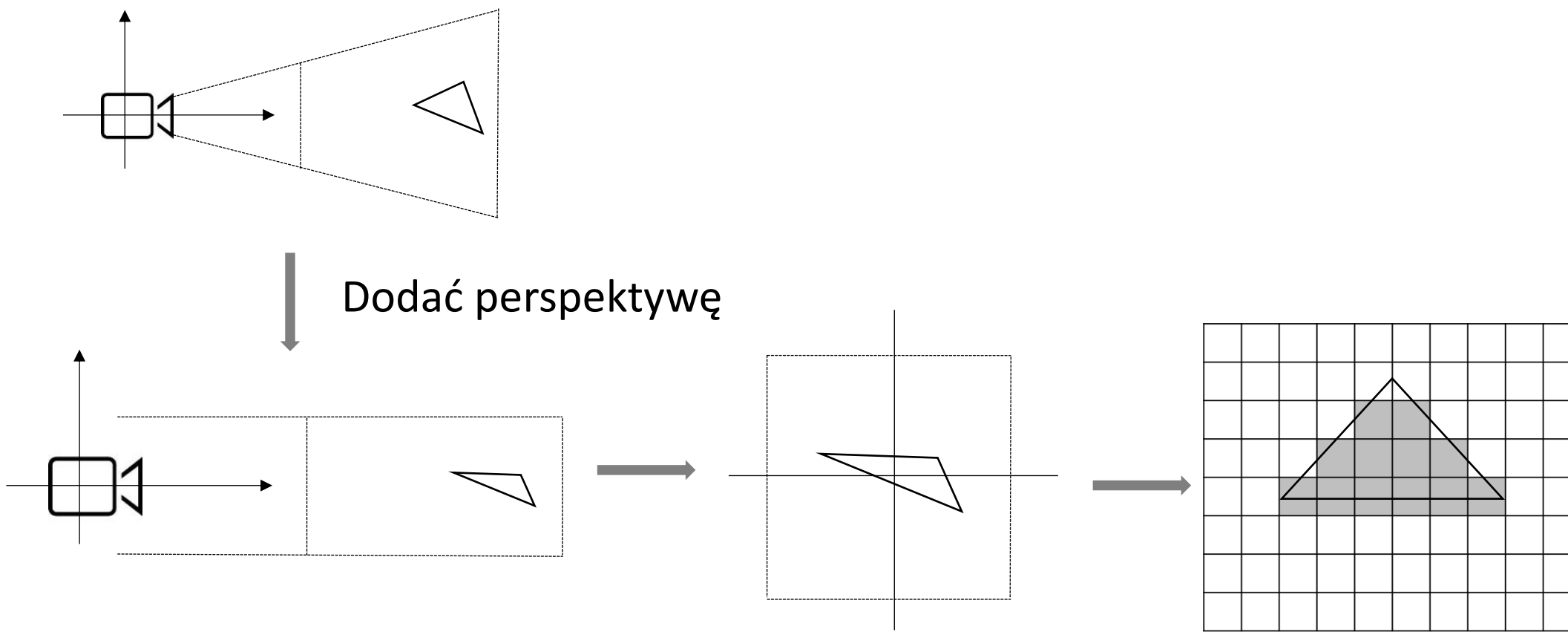
- Co jest perspektywa?
- Wielkość obiektu jest proporcjonalna do dystansu.



Potok graficzny

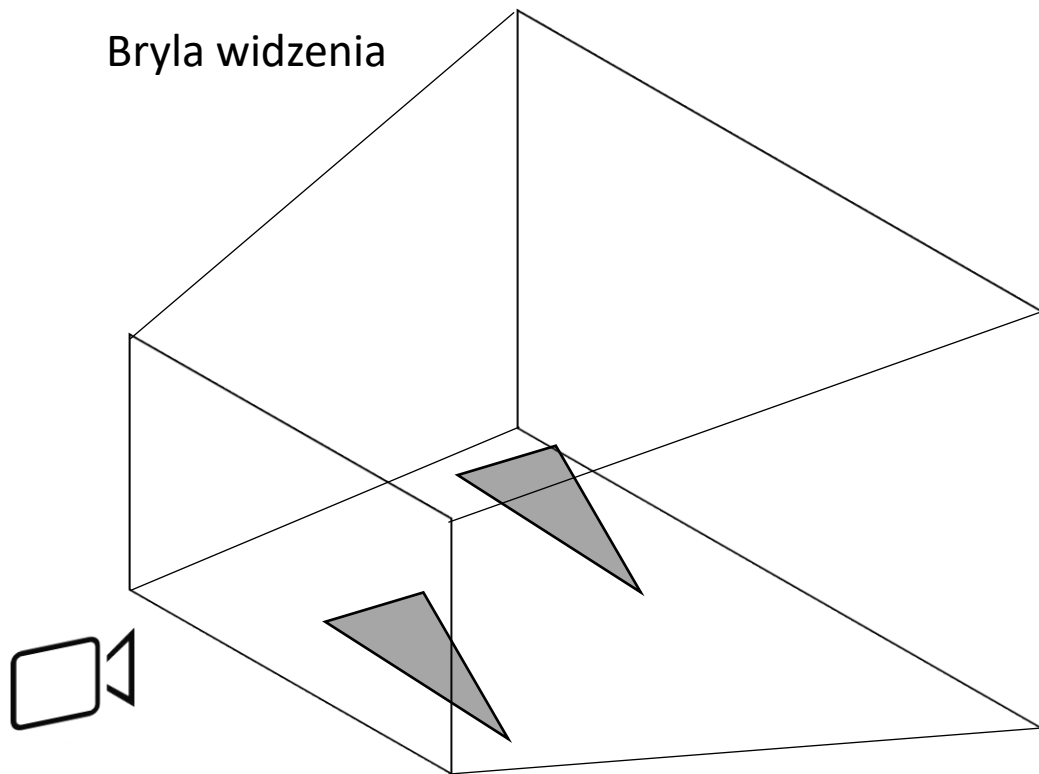


Potok graficzny

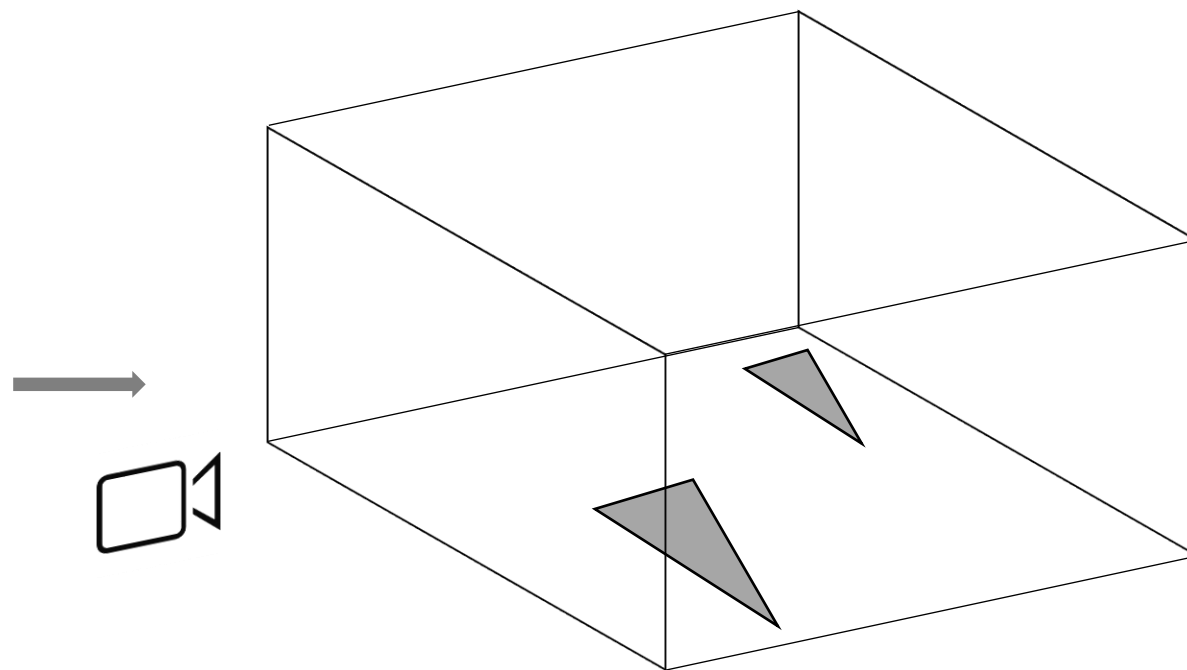


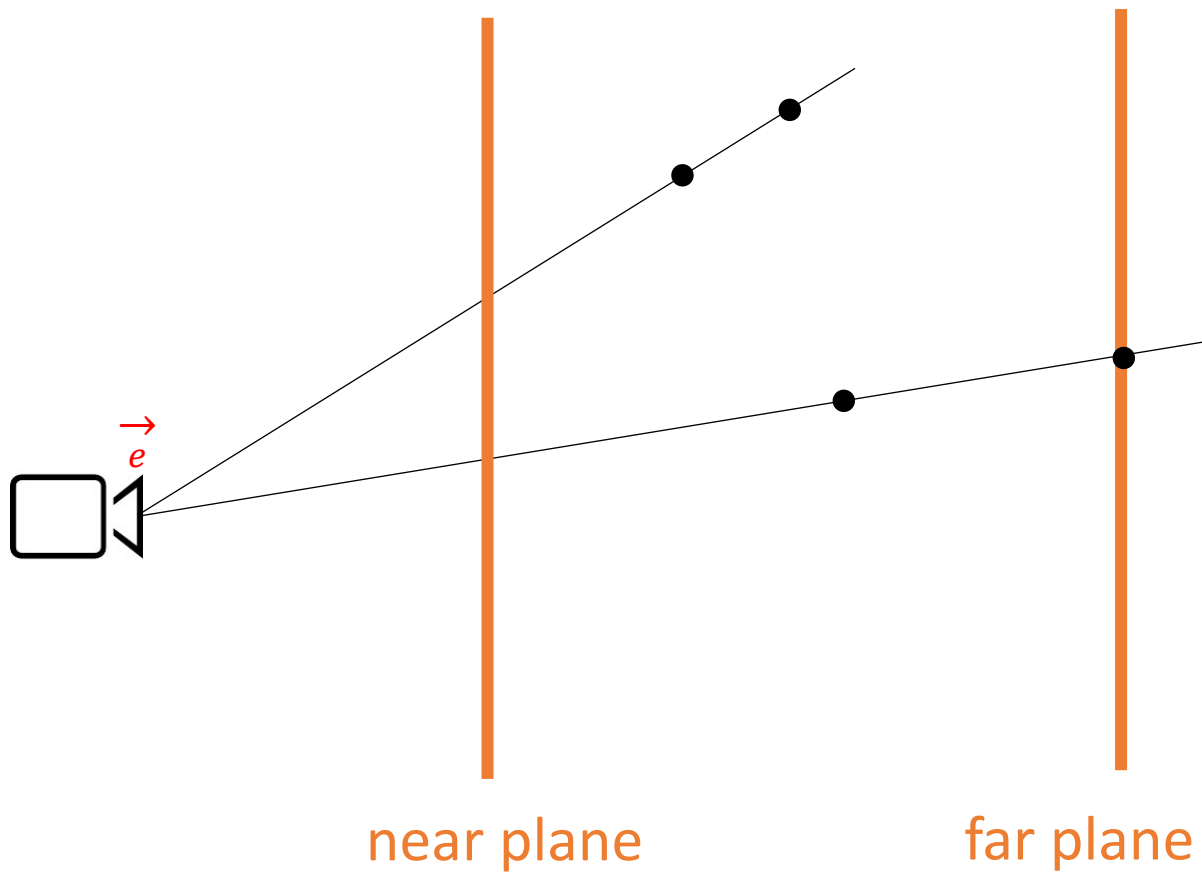
Transformacja bryły widzenia

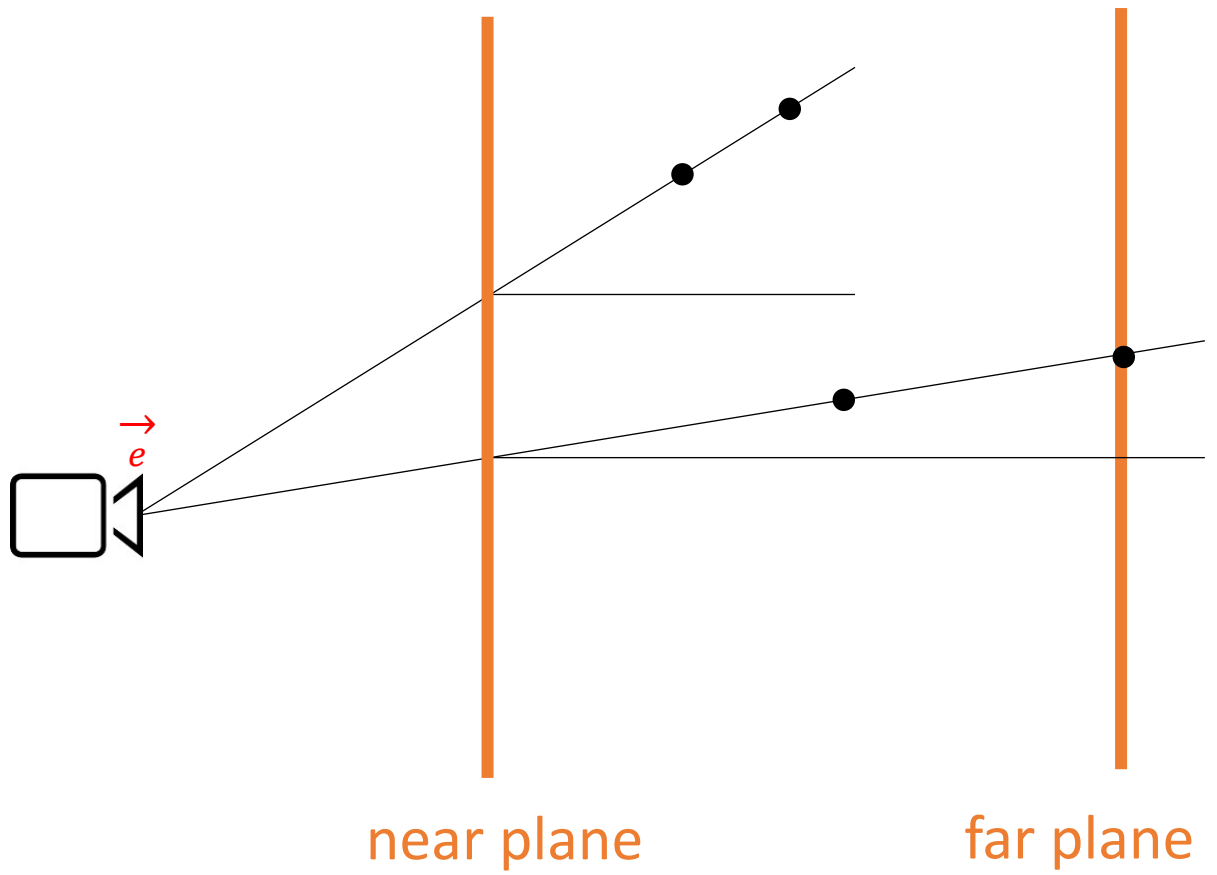
Bryła widzenia

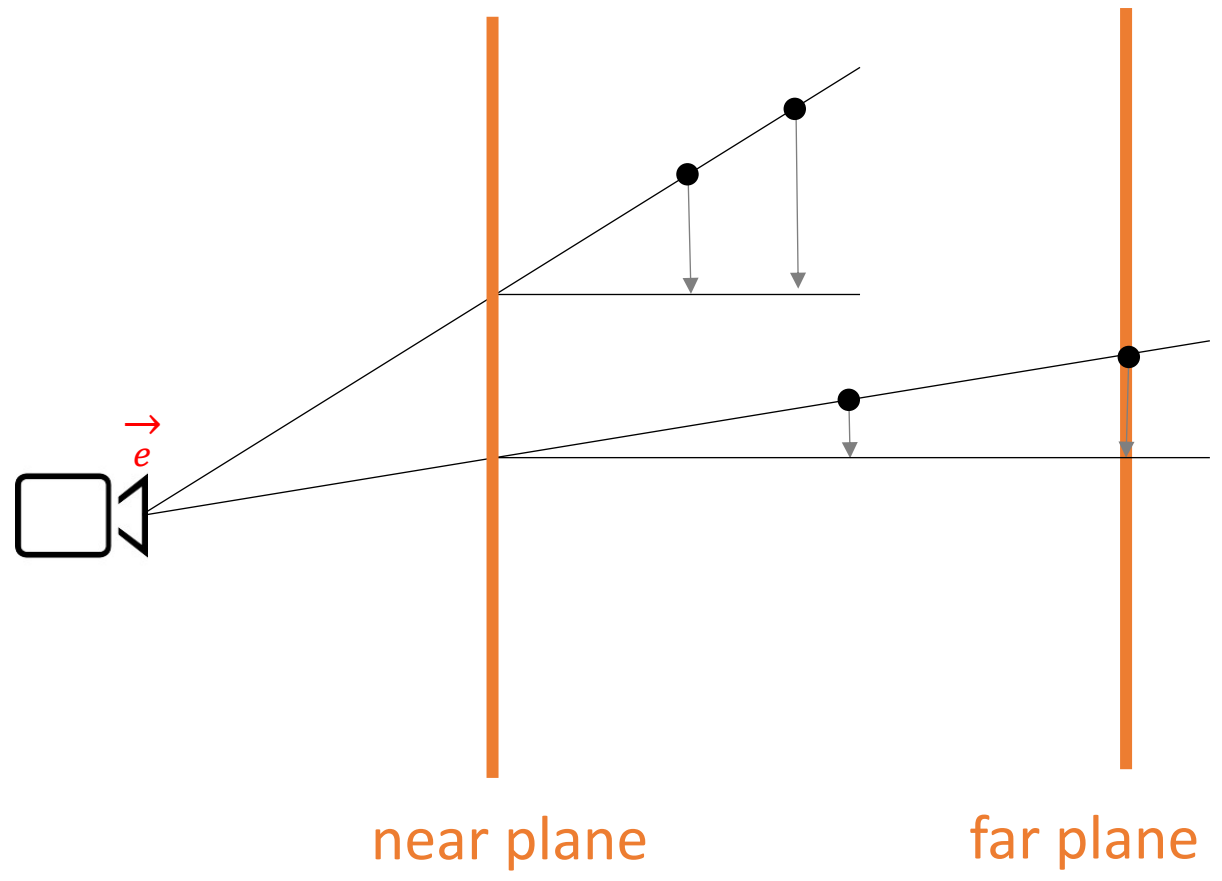


Ortograficzna bryła widzenia

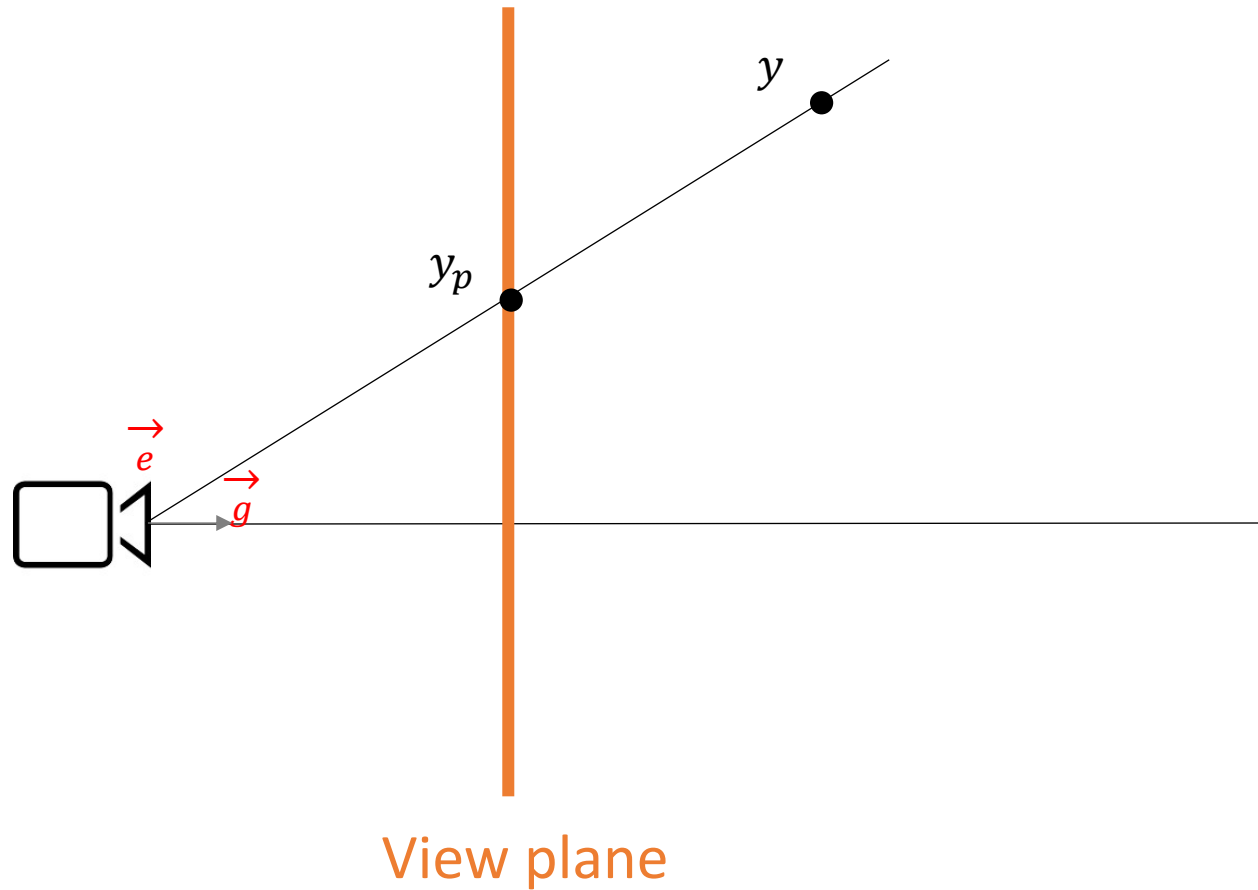




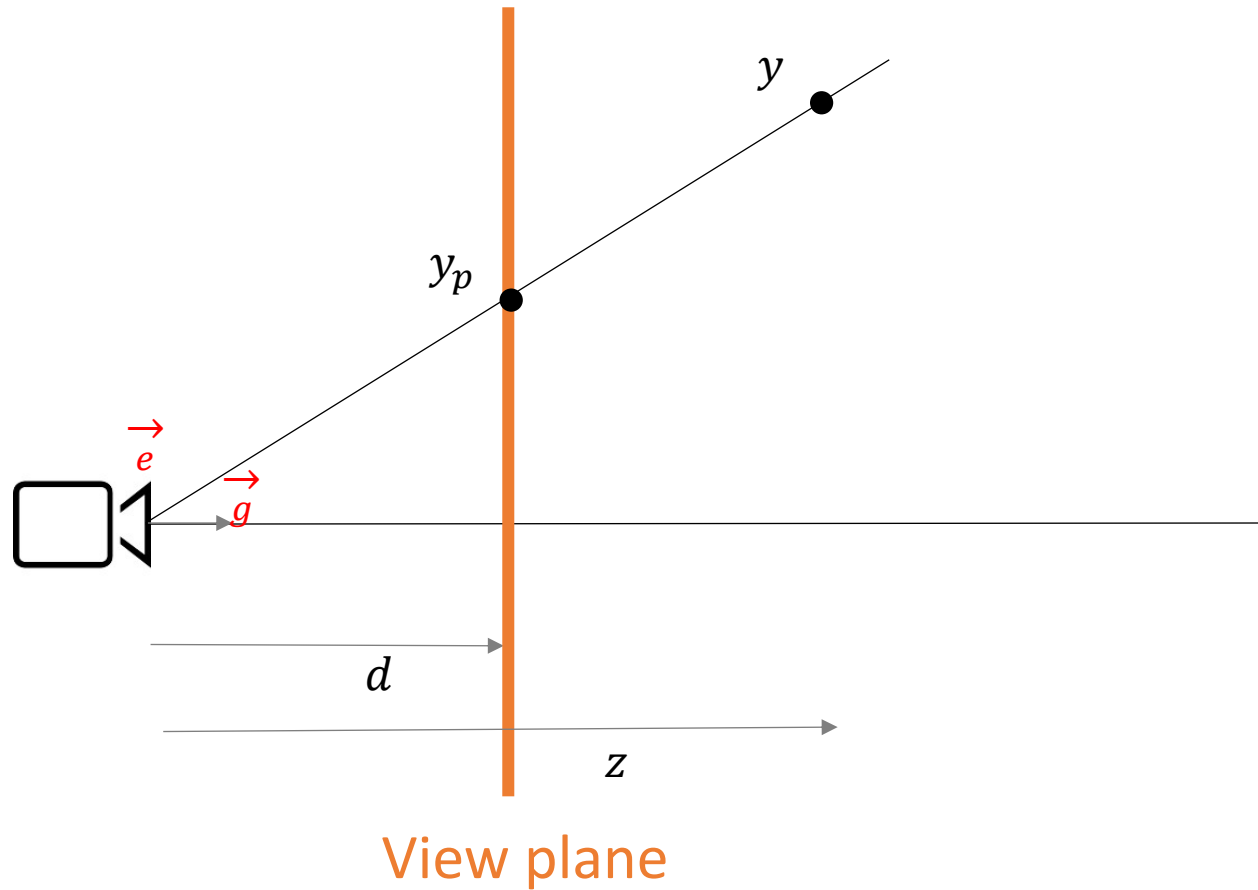




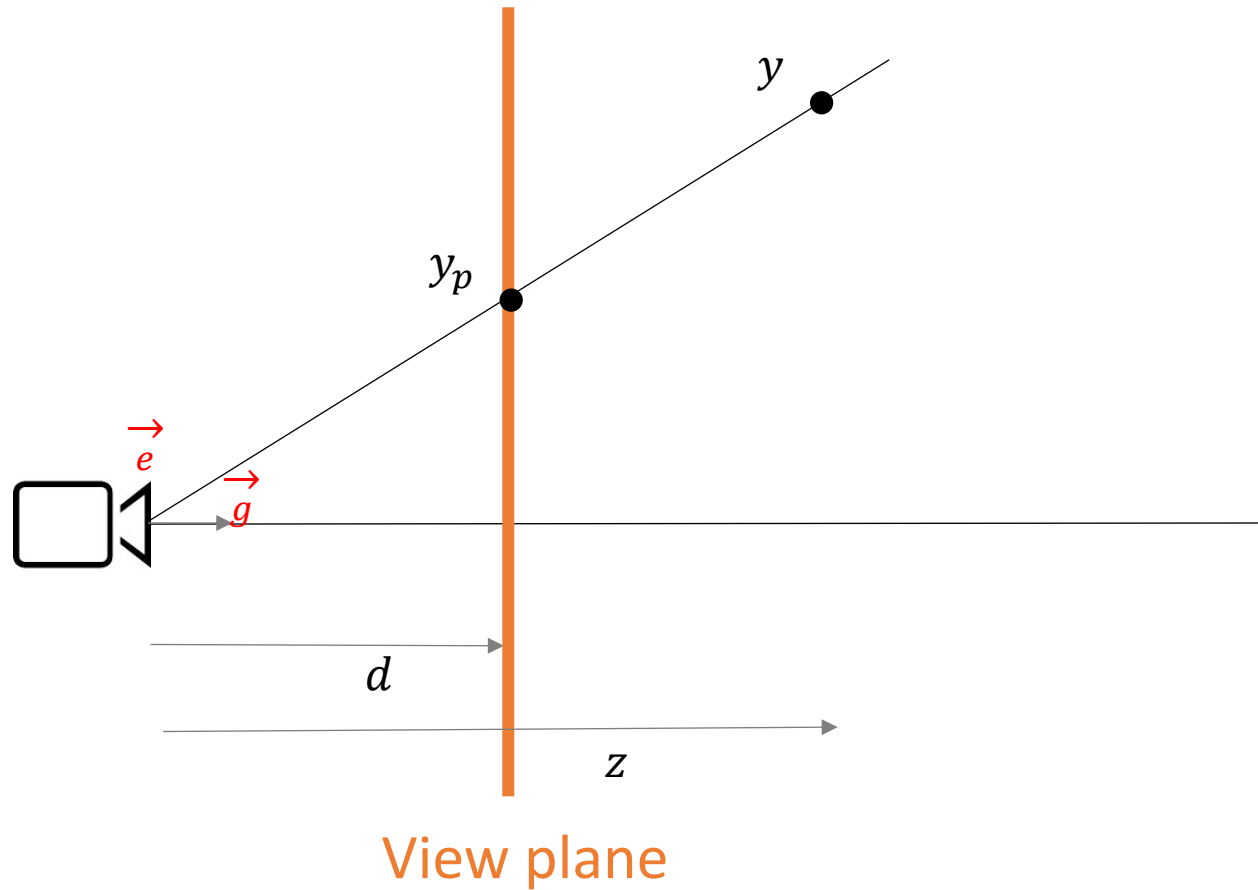
Rzutowanie na płaszczyznę



Rzutowanie na płaszczyznę

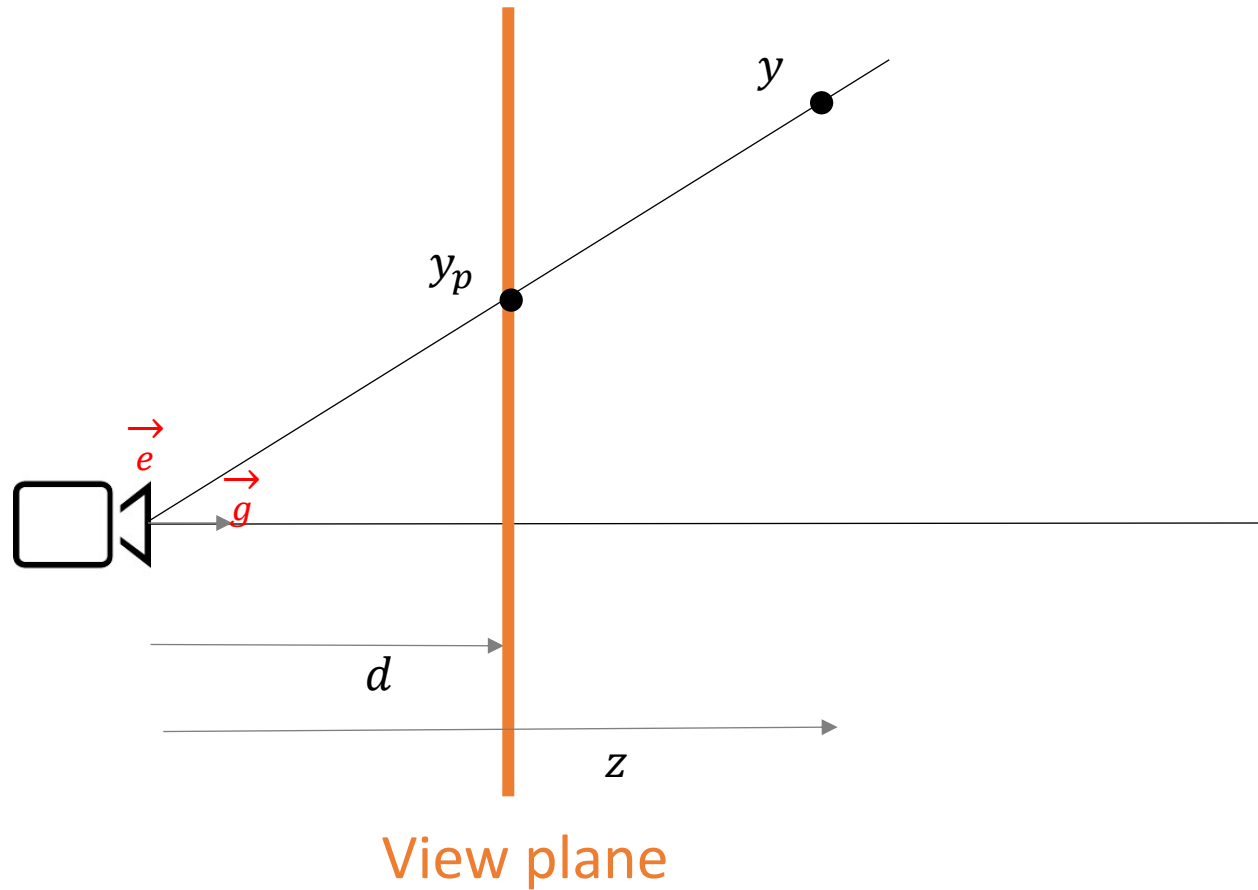


Rzutowanie na płaszczyznę



$$\frac{y_p}{y} = \frac{d}{z} \text{ czyli}$$

Rzutowanie na płaszczyznę



$$\frac{y_p}{y} = \frac{d}{z} \text{ czyli}$$

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

Założenia

- Przyjmujemy ze:
 - Patrzymy w kierunku negatywnej osi z
 - Rzutujemy na powierzchnie bliską (near plane)
- Czyli dystans $d = -n$, co znaczy:
 - $x_p = \frac{d}{-z} x$
 - $y_p = \frac{d}{-z} y$

Założenia

- Przyjmujemy ze:
 - Patrzymy w kierunku negatywnej osi z
 - Rzutujemy na powierzchnie bliską (near plane)
- Czyli dystans $d = -n$, co znaczy:

- $x_p = \frac{d}{-z} x = \frac{n}{z} x$

- $y_p = \frac{d}{-z} y = \frac{n}{z} y$

Założenia

- Przyjmujemy ze:
 - Patrzymy w kierunku negatywnej osi z
 - Rzutujemy na powierzchnie bliską (near plane)
- Czyli dystans $d = -n$, co znaczy:

$$\begin{aligned} \bullet \quad x_p &= \frac{d}{-z} x = \frac{n}{z} x \\ \bullet \quad y_p &= \frac{d}{-z} y = \frac{n}{z} y \end{aligned}$$

$$M \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{nx}{z} \\ \frac{ny}{z} \\ \frac{z}{z} \\ 1 \end{bmatrix}$$

System współrzędnych jednorodnych

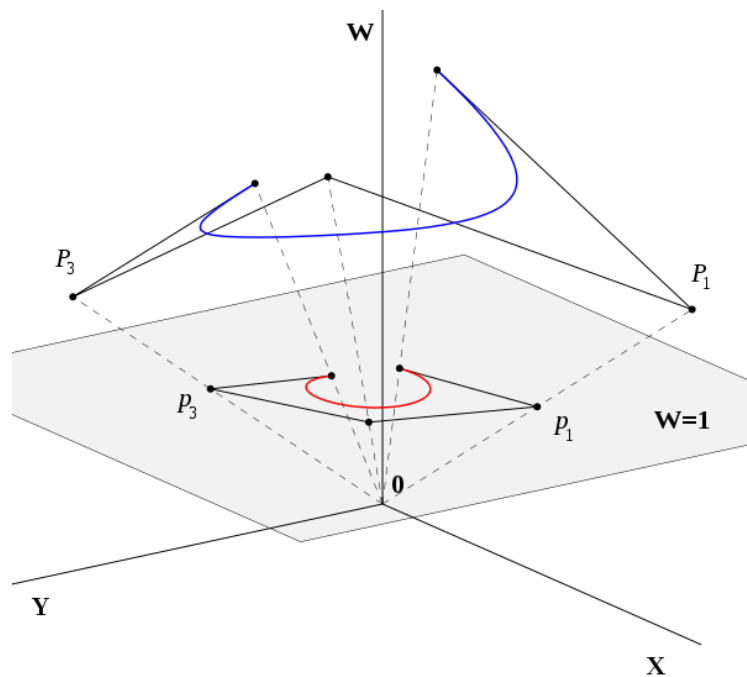
- We współrzędnych jednorodnych wektor:

$(x, y, z, 1)$ reprezentuje punkt **(x, y, z)**

Teraz wprowadzamy możliwość reprezentacji punktów za pomocą:

(x, y, z, w)	co daje nam punkt	$(\frac{x}{w}, \frac{y}{w}, \frac{z}{w})$
jednorodny		kartezjański

August Möbius



https://en.wikipedia.org/wiki/Homogeneous_coordinates

System współrzędnych jednorodnych

- Wtedy transformacja macierzowa staje się:

$$\bullet \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} a_1 & b_1 & c_1 & d_1 \\ a_2 & b_2 & c_2 & d_2 \\ a_3 & b_3 & c_3 & d_3 \\ a_4 & b_4 & c_4 & d_4 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} a_1x + b_1y + c_1z + d_1 \\ a_2x + b_2y + c_2z + d_2 \\ a_3x + b_3y + c_3z + d_3 \\ a_4x + b_4y + c_4z + d_4 \end{bmatrix}$$

System współrzędnych jednorodnych

- Wtedy transformacja macierzowa staje się:

$$\bullet \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} a_1 & b_1 & c_1 & d_1 \\ a_2 & b_2 & c_2 & d_2 \\ a_3 & b_3 & c_3 & d_3 \\ a_4 & b_4 & c_4 & d_4 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} a_1x + b_1y + c_1z + d_1 \\ a_2x + b_2y + c_2z + d_2 \\ a_3x + b_3y + c_3z + d_3 \\ a_4x + b_4y + c_4z + d_4 \end{bmatrix}$$

$$\bullet \begin{bmatrix} a_1x + b_1y + c_1z + d_1 \\ a_2x + b_2y + c_2z + d_2 \\ a_3x + b_3y + c_3z + d_3 \\ a_4x + b_4y + c_4z + d_4 \end{bmatrix} \xrightarrow{\text{homogenizacja}}$$

System współrzędnych jednorodnych

- Wtedy transformacja macierzowa staje się:

$$\bullet \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} a_1 & b_1 & c_1 & d_1 \\ a_2 & b_2 & c_2 & d_2 \\ a_3 & b_3 & c_3 & d_3 \\ a_4 & b_4 & c_4 & d_4 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} a_1x + b_1y + c_1z + d_1 \\ a_2x + b_2y + c_2z + d_2 \\ a_3x + b_3y + c_3z + d_3 \\ a_4x + b_4y + c_4z + d_4 \end{bmatrix}$$

$$\bullet \begin{bmatrix} a_1x + b_1y + c_1z + d_1 \\ a_2x + b_2y + c_2z + d_2 \\ a_3x + b_3y + c_3z + d_3 \\ a_4x + b_4y + c_4z + d_4 \end{bmatrix} \xrightarrow{\text{homogenizacja}} \begin{bmatrix} \frac{a_1x + b_1y + c_1z + d_1}{a_4x + b_4y + c_4z + d_4} \\ \frac{a_2x + b_2y + c_2z + d_2}{a_4x + b_4y + c_4z + d_4} \\ \frac{a_3x + b_3y + c_3z + d_3}{a_4x + b_4y + c_4z + d_4} \\ 1 \end{bmatrix}$$

Transformacja bryły widzenia

Kalkulacja:

$$\begin{bmatrix} a_1x + b_1y + c_1z + d_1 \\ \hline a_4x + b_4y + c_4z + d_4 \\ a_2x + b_2y + c_2z + d_2 \\ \hline a_4x + b_4y + c_4z + d_4 \\ a_3x + b_3y + c_3z + d_3 \\ \hline a_4x + b_4y + c_4z + d_4 \\ 1 \end{bmatrix}$$

Cel:

$$\begin{bmatrix} nx \\ \hline z \\ ny \\ \hline z \\ z \\ 1 \end{bmatrix}$$

Transformacja bryły widzenia

Kalkulacja:

$$\begin{bmatrix} a_1x + b_1y + c_1z + d_1 \\ \hline a_4x + b_4y + c_4z + d_4 \\ a_2x + b_2y + c_2z + d_2 \\ \hline a_4x + b_4y + c_4z + d_4 \\ a_3x + b_3y + c_3z + d_3 \\ \hline a_4x + b_4y + c_4z + d_4 \\ 1 \end{bmatrix}$$

Cel:

$$\begin{bmatrix} nx \\ \hline z \\ ny \\ \hline z \\ z \\ 1 \end{bmatrix}$$

Transformacja bryły widzenia

Kalkulacja:

$$\begin{bmatrix} a_1x + b_1y + c_1z + d_1 \\ \frac{a_4x + b_4y + c_4z + d_4}{a_2x + b_2y + c_2z + d_2} \\ a_3x + b_3y + c_3z + d_3 \\ \frac{a_4x + b_4y + c_4z + d_4}{1} \end{bmatrix}$$

Cel:

$$\begin{bmatrix} nx \\ \frac{z}{ny} \\ \frac{z}{z} \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & \mathbf{1} & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ \mathbf{z} \\ 1 \end{bmatrix}$$

Czwarty wiersz

Transformacja bryły widzenia

Kalkulacja:

$$\begin{bmatrix} a_1x + b_1y + c_1z + d_1 \\ \frac{a_4x + b_4y + c_4z + d_4}{a_2x + b_2y + c_2z + d_2} \\ \frac{a_4x + b_4y + c_4z + d_4}{a_3x + b_3y + c_3z + d_3} \\ \frac{a_4x + b_4y + c_4z + d_4}{1} \end{bmatrix}$$

Cel:

$$\begin{bmatrix} nx \\ \frac{z}{ny} \\ \frac{z}{z} \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Czwarty wiersz

Transformacja bryły widzenia

Kalkulacja:

$$\begin{bmatrix} a_1x + b_1y + c_1z + d_1 \\ \frac{a_4x + b_4y + c_4z + d_4}{a_2x + b_2y + c_2z + d_2} \\ \frac{a_4x + b_4y + c_4z + d_4}{a_3x + b_3y + c_3z + d_3} \\ \frac{a_4x + b_4y + c_4z + d_4}{1} \end{bmatrix}$$

Cel:

$$\begin{bmatrix} nx \\ \frac{z}{ny} \\ \frac{z}{z} \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Czwarty wiersz

$$\begin{bmatrix} n & 0 & 0 & 0 \\ 0 & n & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

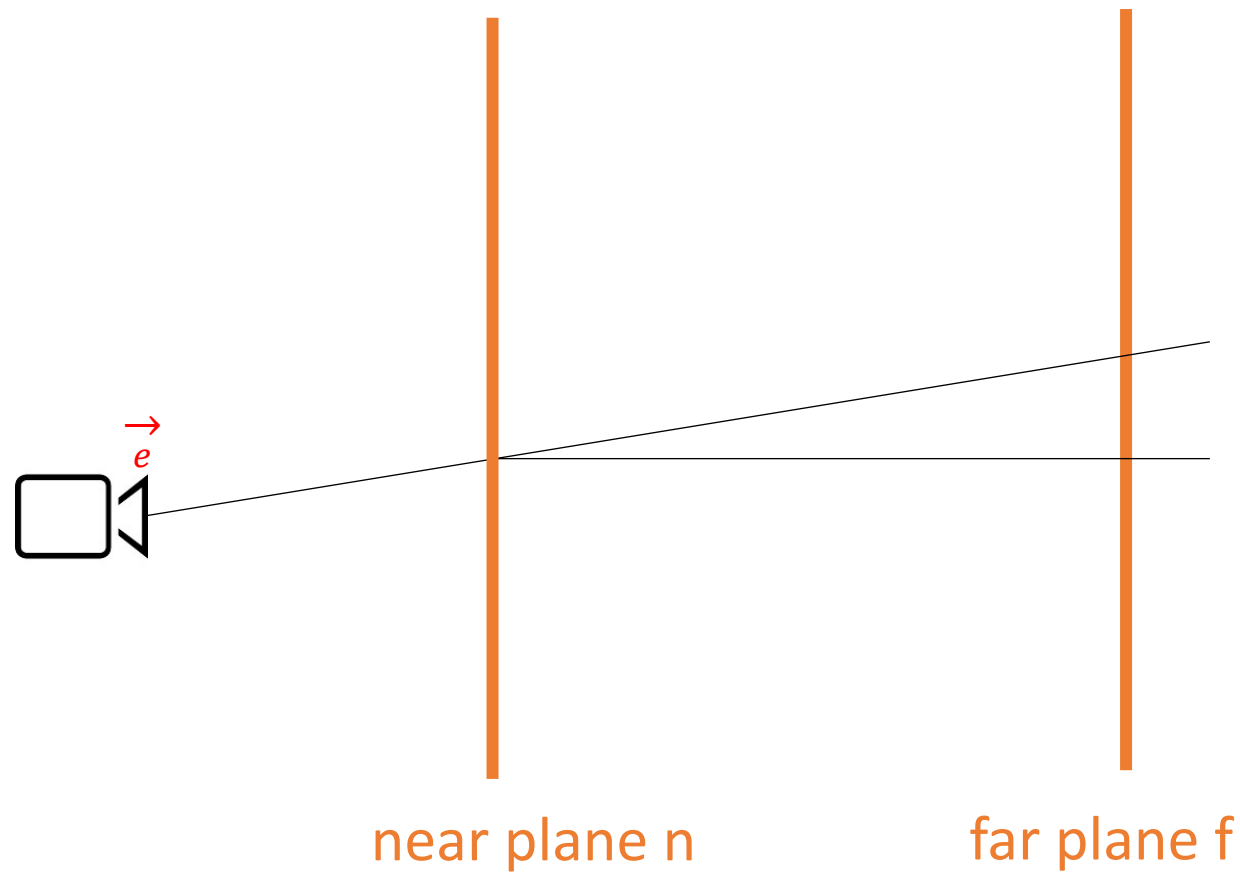
Pierwszy i drugi wiersz

Co z trzecim wierszem?

$$\begin{bmatrix} n & 0 & 0 & 0 \\ 0 & n & 0 & 0 \\ ? & ? & ? & ? \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

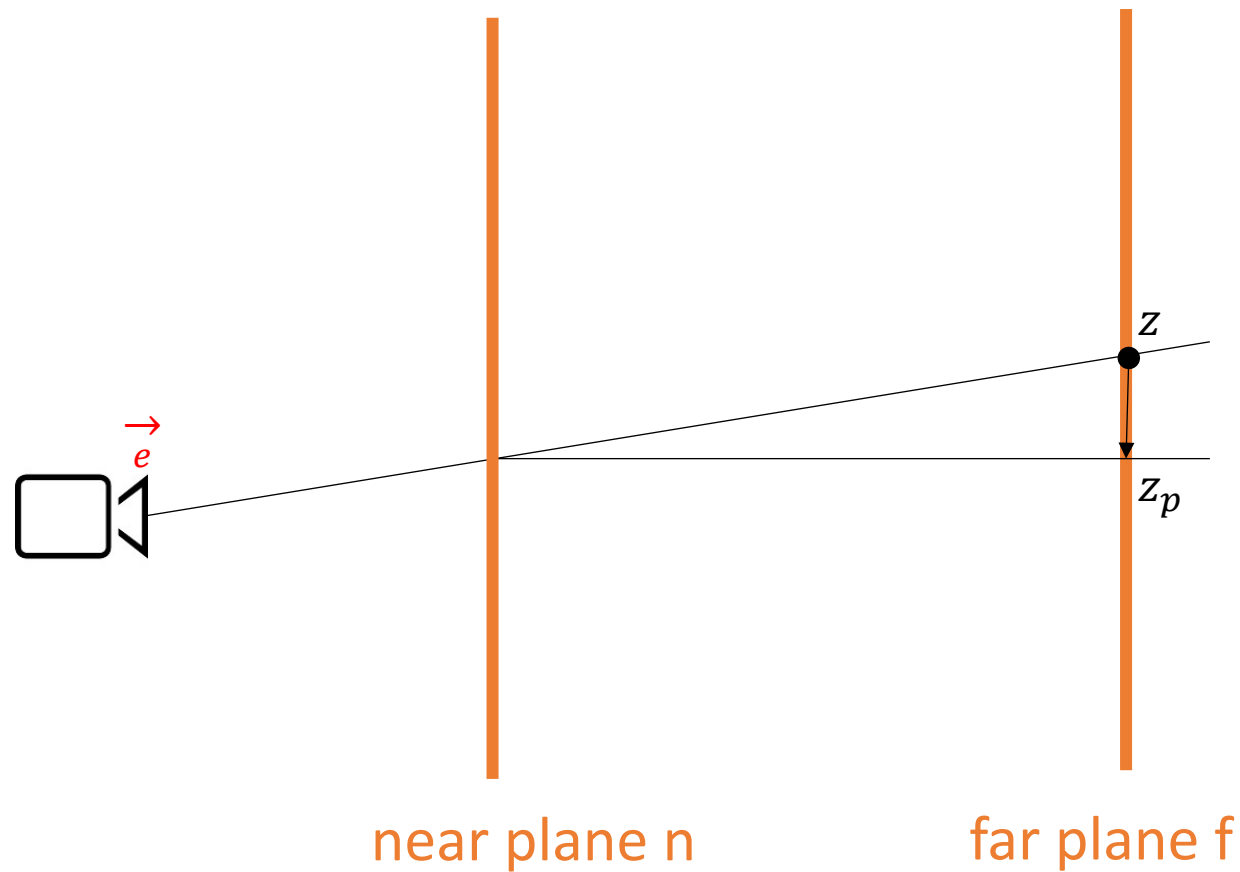
$$\begin{bmatrix} nx \\ \hline z \\ ny \\ \hline z \\ \textcolor{red}{z} \\ 1 \end{bmatrix}$$

Transformacja bryły widzenia



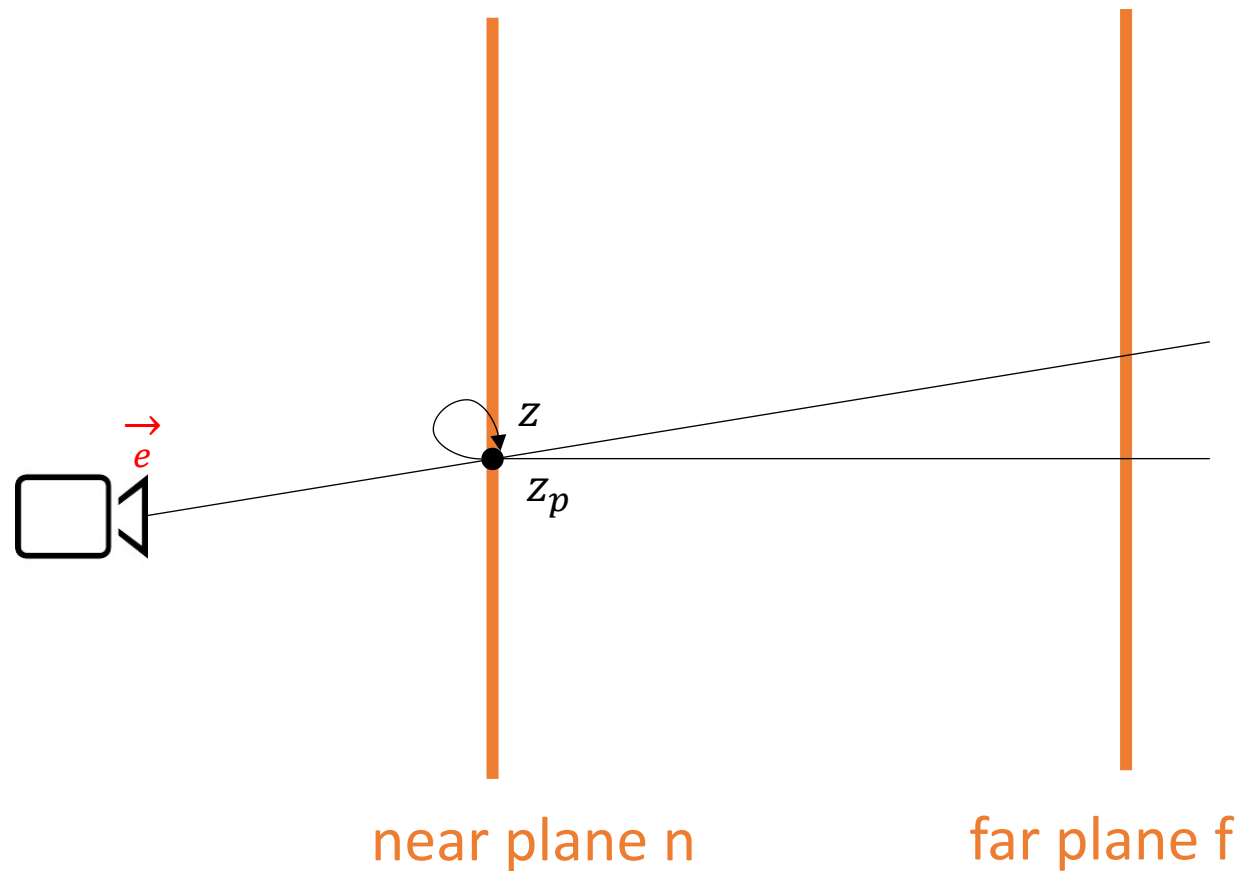
Transformacja bryły widzenia

$$z_p = f$$



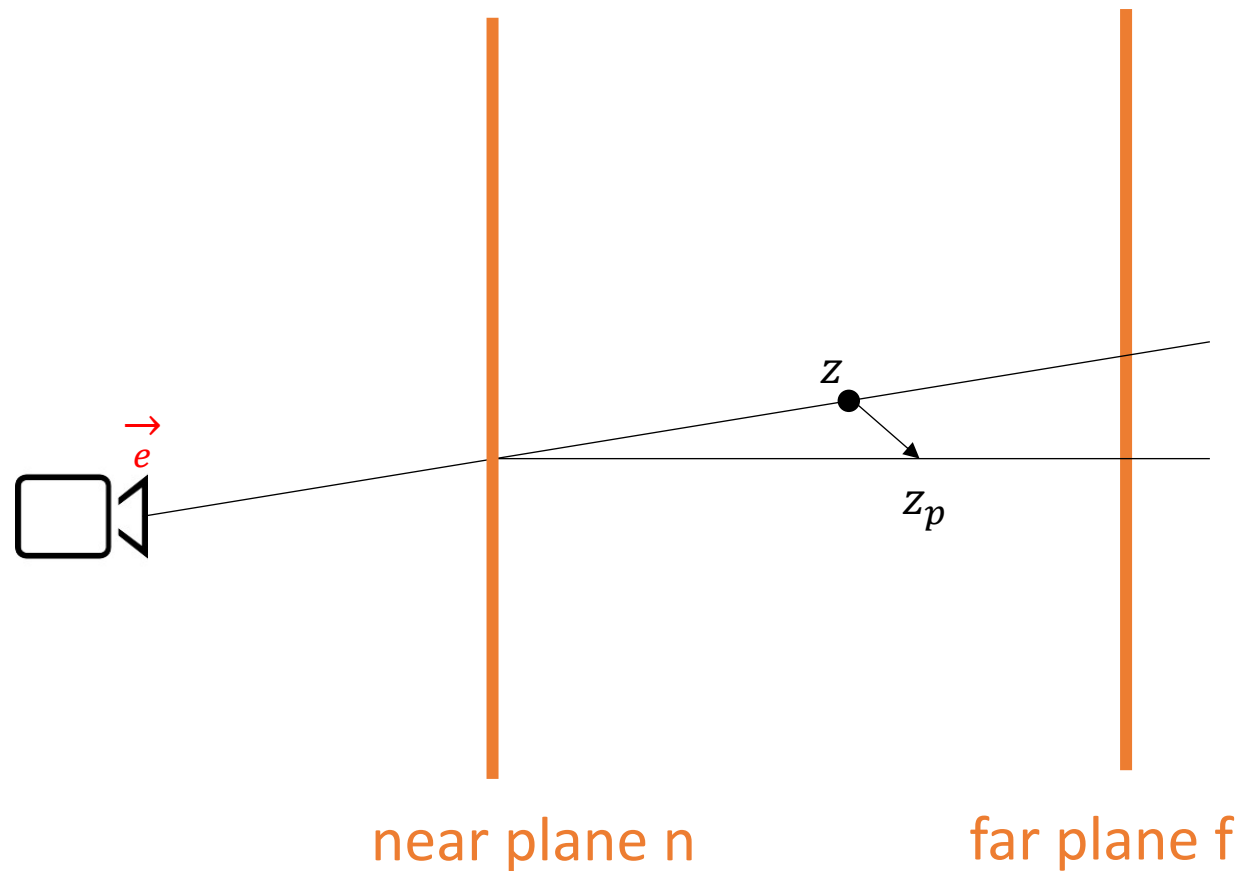
Transformacja bryły widzenia

$$z_p = n$$



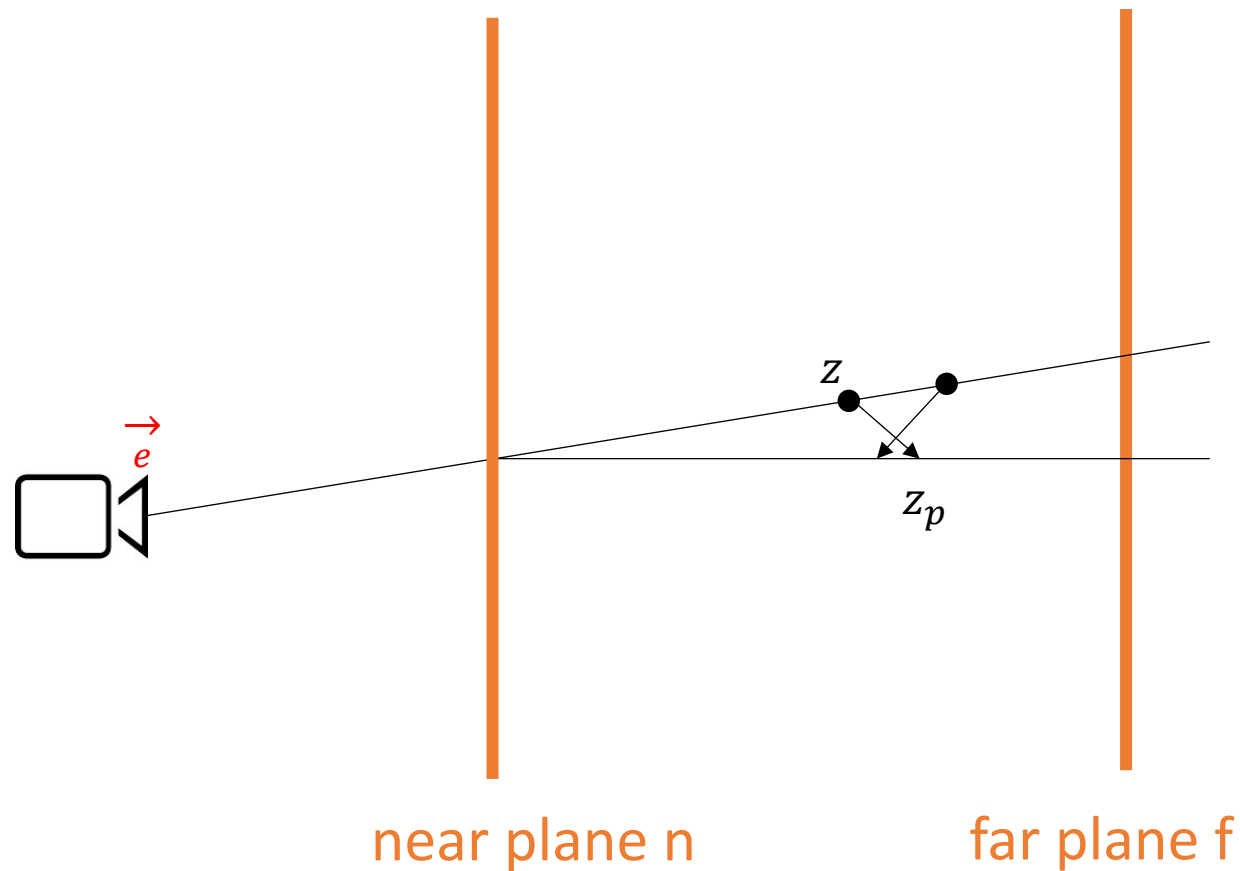
Transformacja bryły widzenia

$$z_p \in [n, f]$$



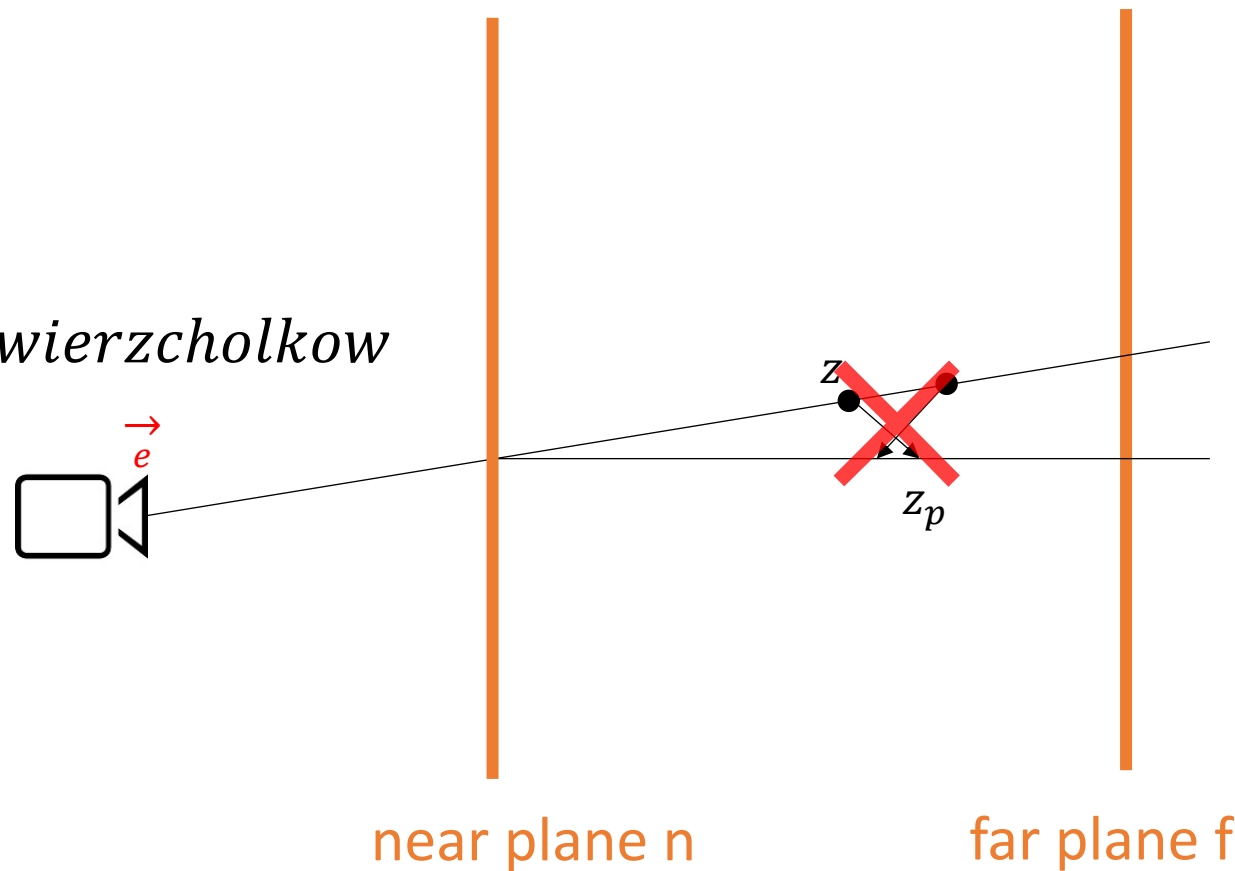
Transformacja bryły widzenia

$$z_p \in [n, f]$$



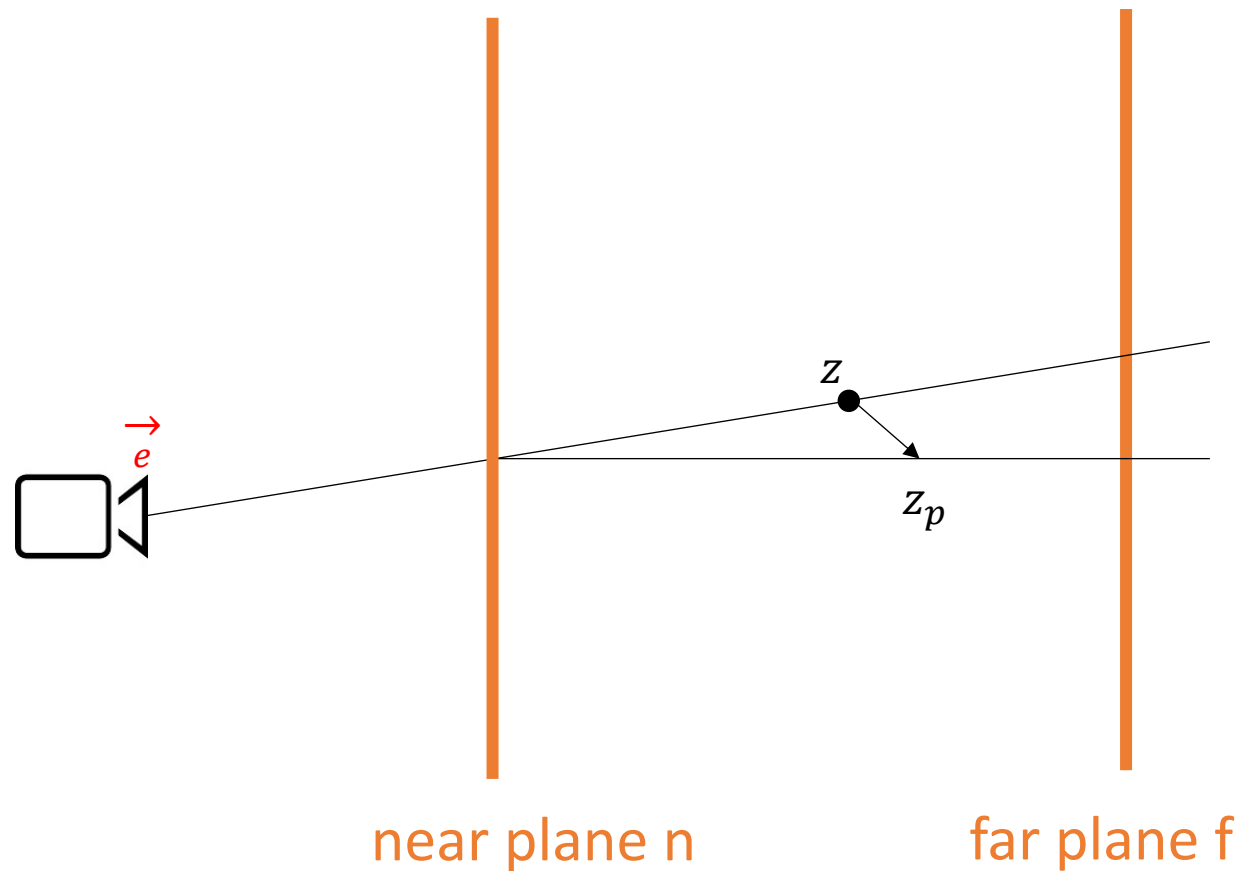
Transformacja bryły widzenia

z_p ma zachować porządek wierzchołków



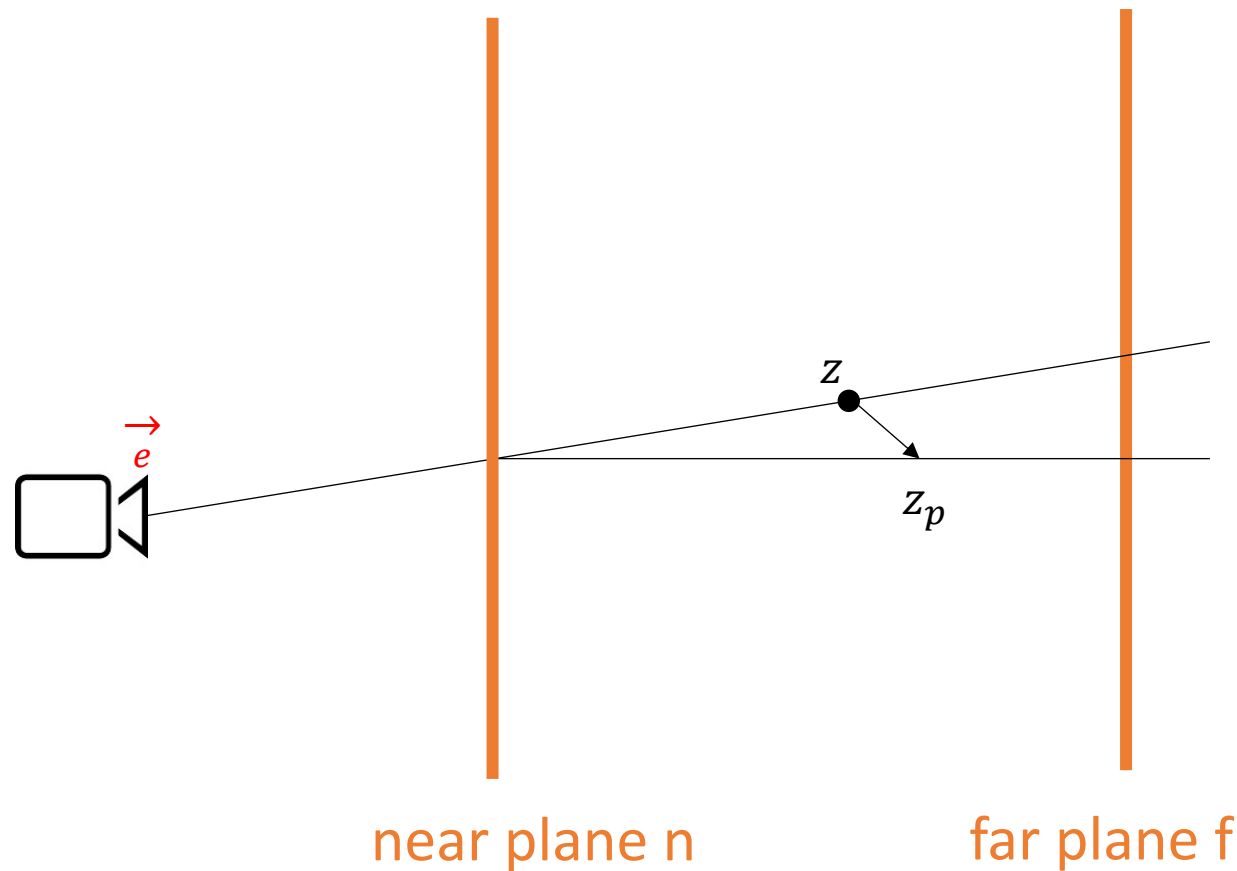
Transformacja bryły widzenia

$$z_p = \frac{nf}{z}, z \in [n, f]$$



Transformacja bryły widzenia

$$z_p = n + f - \frac{nf}{z}$$



Dowód

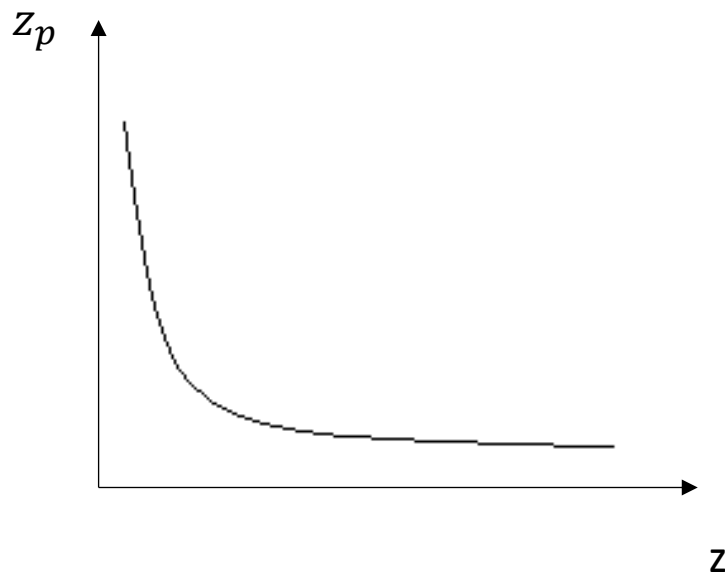
- $z_p = n + f - \frac{fn}{z}$
- Porządek wzdłuż osi z jest zachowany gdy $0 > n \geq z_1 > z_2 \geq f$ to $z_{1p} > z_{2p}$
- $z_{1p} - z_{2p} = \frac{fn}{z_2} - \frac{fn}{z_1} = \frac{(z_1 - z_2)fn}{z_1 z_2}$
- Bo $f, z_1, z_2, n < 0$ mamy $\frac{fn}{z_1 z_2} > 0$ i bo $z_1 > z_2$ mamy $z_1 - z_2 > 0$
- Czyli $z_{1p} > z_{2p}$

Transformacja bryły widzenia

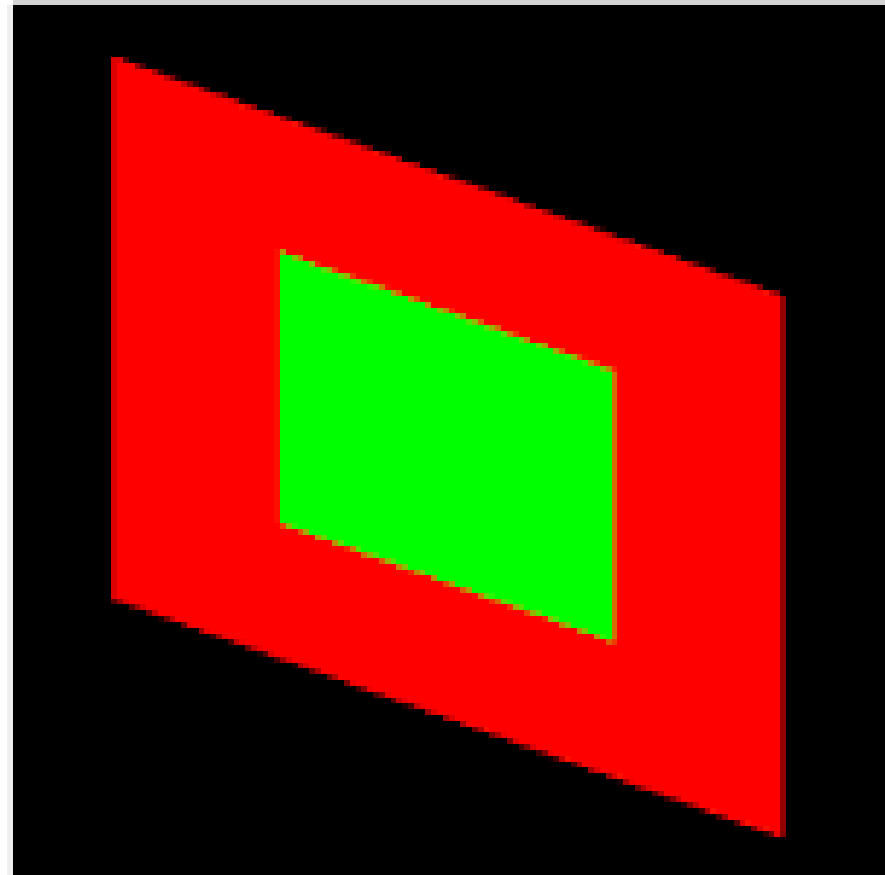
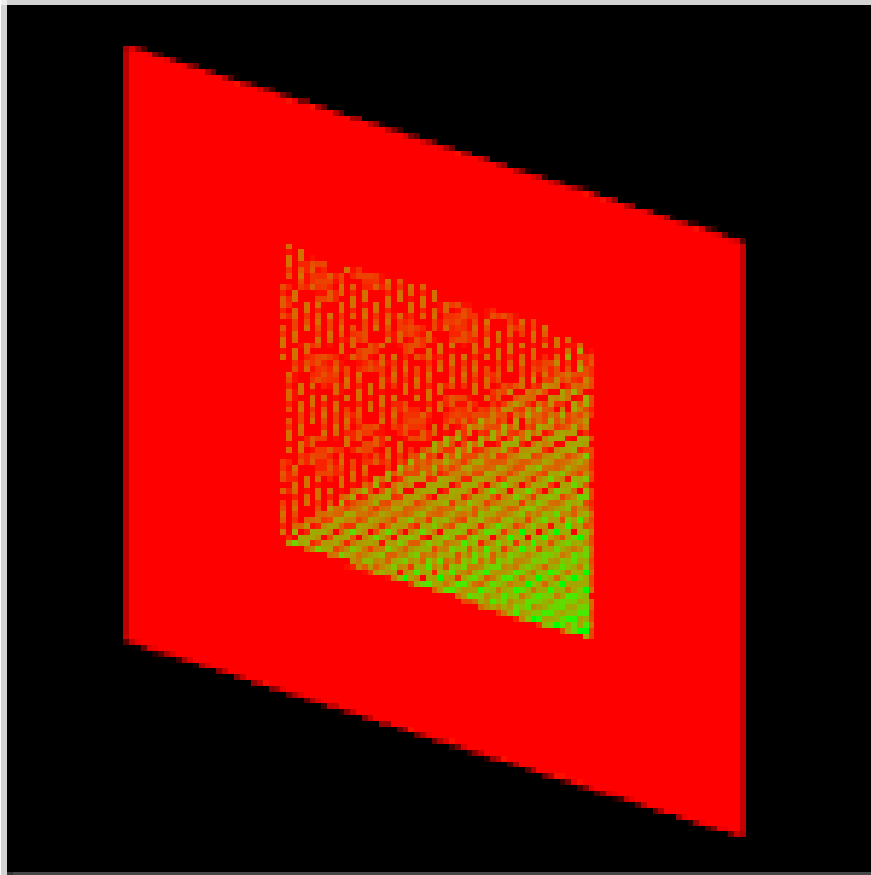
$$z_p = n + f - \frac{fn}{z} \rightarrow z_p \sim \frac{1}{z}$$

Transformacja bryły widzenia

$$z_p = n + f - \frac{fn}{z} \rightarrow z_p \sim \frac{1}{z}$$



Z-fighting



Macierz P

$$\begin{bmatrix} n & 0 & 0 & 0 \\ 0 & n & 0 & 0 \\ 0 & 0 & n+f & -nf \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Macierz P

$$\begin{bmatrix} n & 0 & 0 & 0 \\ 0 & n & 0 & 0 \\ 0 & 0 & n + f & -nf \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{nx}{z} \\ \frac{ny}{z} \\ n + f - \frac{nf}{z} \\ 1 \end{bmatrix}$$

Macierz transformacji perspektywicznej

- Czyli nasza końcowa macierz transformacji perspektywicznej M_{pers} jest wtedy:
- $M_{per} = M_{orth}P$

Macierz transformacji perspektywicznej

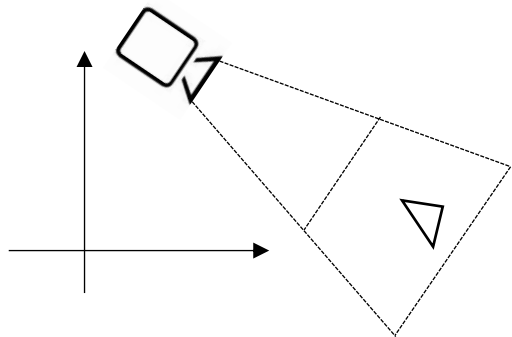
- Czyli nasza końcowa macierz transformacji perspektywicznej M_{pers} jest wtedy:

$$\bullet M_{per} = M_{orth}P = M_{orth} \begin{bmatrix} n & 0 & 0 & 0 \\ 0 & n & 0 & 0 \\ 0 & 0 & n+f & -nf \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

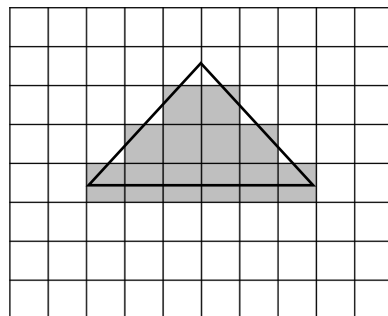
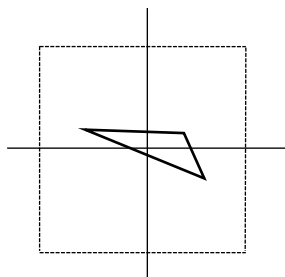
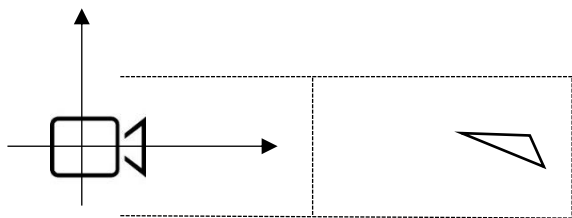
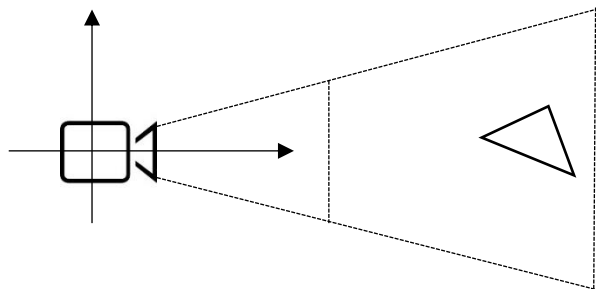
Macierz transformacji perspektywicznej

- Czyli nasza końcowa macierz transformacji perspektywicznej M_{pers} jest wtedy:

$$\bullet M_{per} = M_{orth}P = \begin{bmatrix} \frac{2n}{r-l} & 0 & \frac{l+r}{l-r} & 0 \\ \frac{2n}{t-b} & 0 & \frac{b+t}{b-t} & 0 \\ 0 & 0 & \frac{f+n}{n-f} & \frac{2fn}{f-n} \\ 0 & 0 & 1 & 0 \end{bmatrix}$$



Potok graficzny



- Potok graficzny z **ortograficznym** rzutem:

- $M_{vp}M_{ort}M_{cam} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$

- Potok graficzny z **perspektywnym** rzutem:

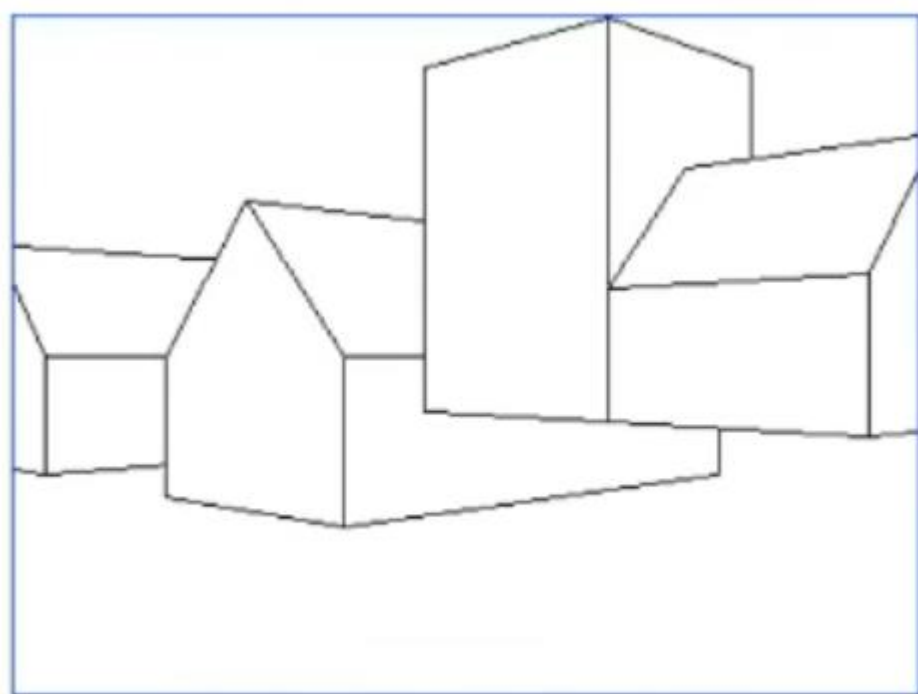
- $M_{vp}M_{per}M_{cam} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$

Rysowanie na ekranie

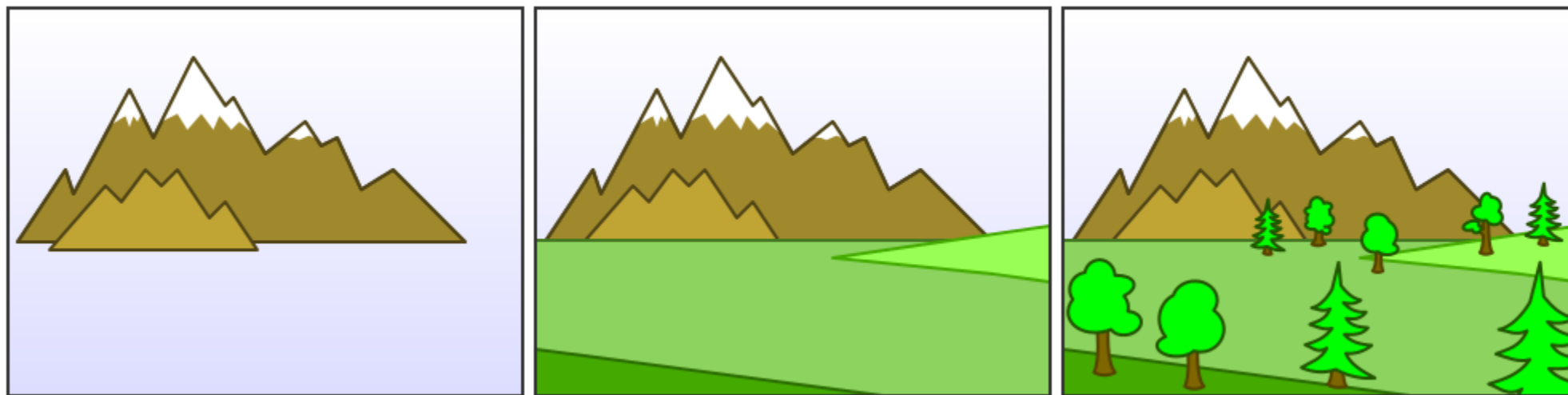
- Żeby narysować krawędz na ekranie możemy się wtedy posługiwać następującym pseudokodem:

```
compute  $M_{vp}$ 
compute  $M_{per}$ 
compute  $M_{cam}$ 
 $M = M_{vp} M_{per} M_{cam}$ 

for each line segment (a, b) do
     $p = M a$ 
     $q = M b$ 
    drawline( $p_x/p_w$ ,  $p_y/p_w$ ,  $q_x/q_w$ ,  $q_y/q_w$ )
```



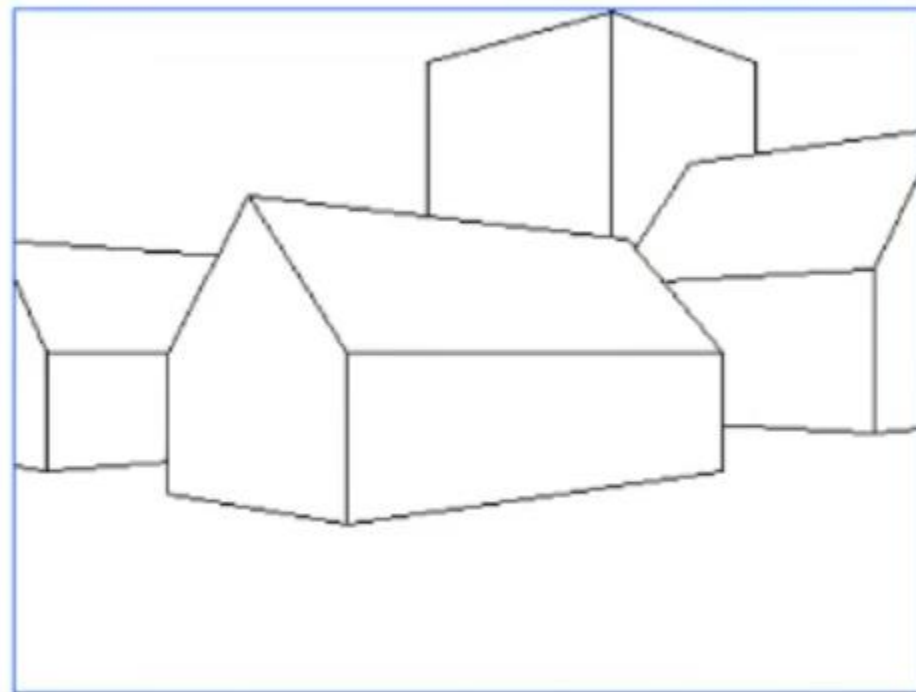
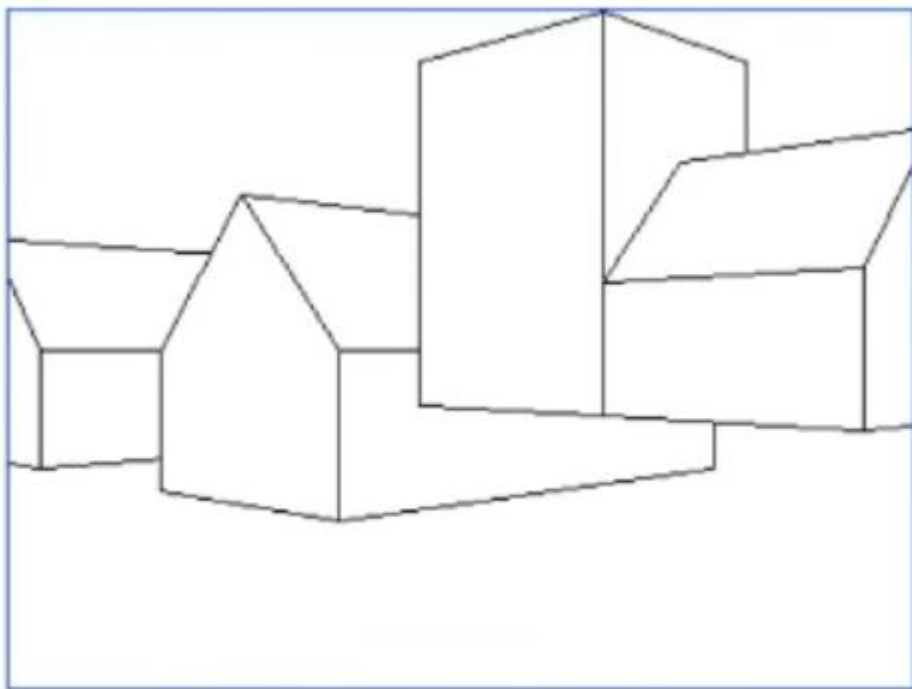
Algorytm malarza (painter's algorithm)



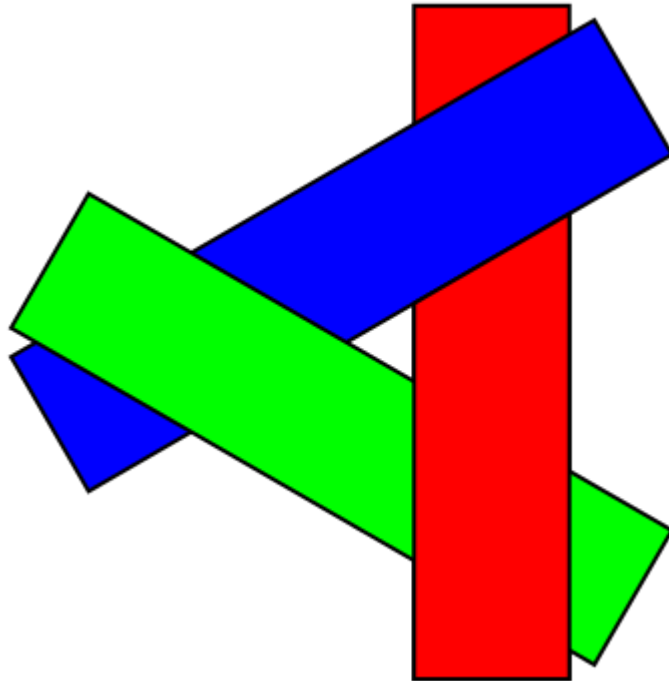
Algorytm malarza (painter's algorithm)

- Algorytm rysuje wielokąty w kolejności odległości od kamery
- Najmniejsza współrzędna z_s od wszystkich wielokątów jest używana żeby odległość obliczyć
- Blizsze wielokąty zasłaniają te które są dalej

Przykład



Częściowe pokrywanie wielokątów



Z-bufor

- Algorytm z-buforu jest podobny w koncepcji do algorytmu malarza ale na poziomie pikseli
- Dwie tablice są stworzone gdzie każdy piksel koresponduje jednemu elementu tablicy
- **Bufor głębokości** (depth buffer) zatrzymuje dystans **z** od najbliższej powierzchni w ułożonej przestrzeni świata dla każdego pikselu
- **Bufor klatek** (frame buffer) zatrzymuje indeksy wielokątów (żeby później wybrać właściwe kolory)

Z-buffer

Algorithm 1 Z buffer

Require: a set of polygons P , a depth buffer array Z and a frame buffer array F
initialise Z to $z_{\max}$
for all polygons in P **do**
 for all pixels in the current polygon **do**
 calculate the z co-ordinate of the point corresponding to the current pixel
 if $z < Z(x, y)$ **then**
 Replace $Z(x, y)$ with z
 Replace $F(x, y)$ with the colour of the current polygon
 end if
 end for
end for
Display F on screen

∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
∞	∞	∞	∞	∞	∞	∞	∞	∞	∞

Z buffer

Frame buffer

∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
∞	1	2	3	4	5	6	7	∞	∞
∞	1	2	3	4	5	6	∞	∞	∞
∞	1	2	3	4	5	∞	∞	∞	∞
∞	1	2	3	4	∞	∞	∞	∞	∞
∞	1	2	3	∞	∞	∞	∞	∞	∞
∞	1	2	∞	∞	∞	∞	∞	∞	∞
∞	1	∞	∞	∞	∞	∞	∞	∞	∞
∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
∞	∞	∞	∞	∞	∞	∞	∞	∞	∞

Z buffer

Frame buffer

∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
∞	1	2	3	4	5	6	7	∞	∞
∞	1	2	3	4	4	3	2	1	∞
∞	1	2	3	4	4	3	2	1	∞
∞	1	2	3	4	4	3	2	1	∞
∞	1	2	3	5	4	3	2	1	∞
∞	1	2	6	5	4	3	2	1	∞
∞	1	∞	∞	∞	∞	∞	∞	∞	∞
∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
∞	∞	∞	∞	∞	∞	∞	∞	∞	∞

Z buffer

Frame buffer

Wymogi pamięciowe

- Z bufor jest obliczeniowo efektywny ale zużywa znaczaca ilosc pamięci
- Dla wyświetlacza 1920 x 1080 pikseli używając 16b RGB model kolorowy i 24b buforu głębokości zużycie pamięci jest:
- $pamiec = bufor\ glebokosci + bufor\ klatek$
 - $= (1920 \times 1080 \times 24) + (1920 \times 1080 \times 16)$
 - $= 49766400 + 33,177,600$
 - $= 82,944,000b$
 - $= 10,368,600B \sim 10MB$