Grafika komputerowa

Wykład

Przekształcenia geometryczne

Co decyduje o realizmie tworzonych obrazów …

Realizm obrazu – odwzorowanie w syntetycznym obrazie efektów wizualnych obserwowanych w naturze.

Czynniki decydujące o realizmie obrazu:

- **modelowanie geometryczne** - opis kształtu i położenia w przestrzeni wizualizowanych obiektów
- **modelowanie koloru** - opis cech powierzchni obiektów związanych z jej kolorem (*barwa, nasycenie i jasność*)
- **modelowanie tekstury –** opis cech obiektów związanych z charakterystycznymi własnościami makropowierzchni
- **modelowanie oświetlenia -** opis rozkładu oświetlenia na powierzchni obiektów (*rozkład cieni, odblasku, rozjaśnień*) określony na podstawie definicji źródeł światła oraz własności optycznych powierzchni związanych z rozproszeniem energii świetlnej (*pochłanianie, załamanie i odbicie światła*)

Modelowanie geometryczne

Modelowanie krzywych i powierzchni
(ang. *Curves & surface modeling*)

1. Metody interpolacyjne
   - interpolacja liniowa i wielomianowa
   - interpolacja paraboliczna
   - interpolacja Akima
   - powierzchnie Coonsa
2. Metody aproksymacyjne
   - krzywe i powierzchnie Beziera
   - krzywe i powierzchnie B-sklejane
   - krzywe i powierzchnie β i β2-sklejane

Modelowanie brył
(ang. *Solid modeling)*

1. Prymitywy przestrzenne
2. Lokalizacja przestrzenna
3. Drzewa ósemkowe (ang. *octrees*)
4. Zakreślanie przestrzeni (ang. *sweeping*)
5. Konstrukcyjna geometria brył – CSG (ang. *Constructive Solid Geometry*)
6. Siatki wielokątów
7. Opis brzegowy – BR (Brep) (ang. *boundary representation*)

Przekształcenia geometryczne

Modelowanie sceny
- przekształcenia modelujące (*translacja, skalowanie, obroty*)

Odwzorowanie sceny na płaszczyźnie (wireframe)
- przekształcenia rzutujące (*rzut perspektywiczny, ortogonalny*)

Definiowanie obserwatora
Definiowanie okna zobrazowania
- przekształcenia geometryczne układu współrzędnych

Treść wykładu

Przekształcenia na płaszczyźnie

- przekształcenia 2D
- współrzędne jednorodne i macierzowa reprezentacja przekształceń 2D
- składanie przekształceń 2D

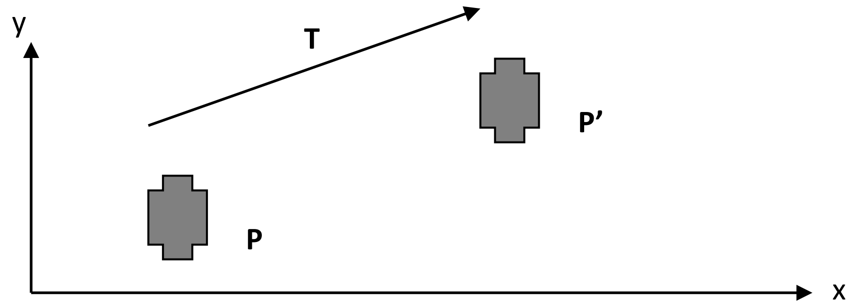
Przekształcenia w przestrzeni 3D

- układy współrzędnych 3D
- przekształcenia 3D
- składanie przekształceń 3D

Model procesu rzutowania 3D

- rzut perspektywiczny
- rzut ortogonalny
- normalizacja do układu współrzędnych urządzenia
- definiowanie okna zobrazowania (viewport)
- definiowanie obserwatora

Translacja (przesunięcie) – zmiana lokalizacji



$P = \{P_x, P_y\}$ - punkty obiektu przed przekształceniem
$P' = \{P'_x, P'_y\}$ - punkty obiektu po przekształceniu
$T = [d_x, d_y]$ - wektor translacji

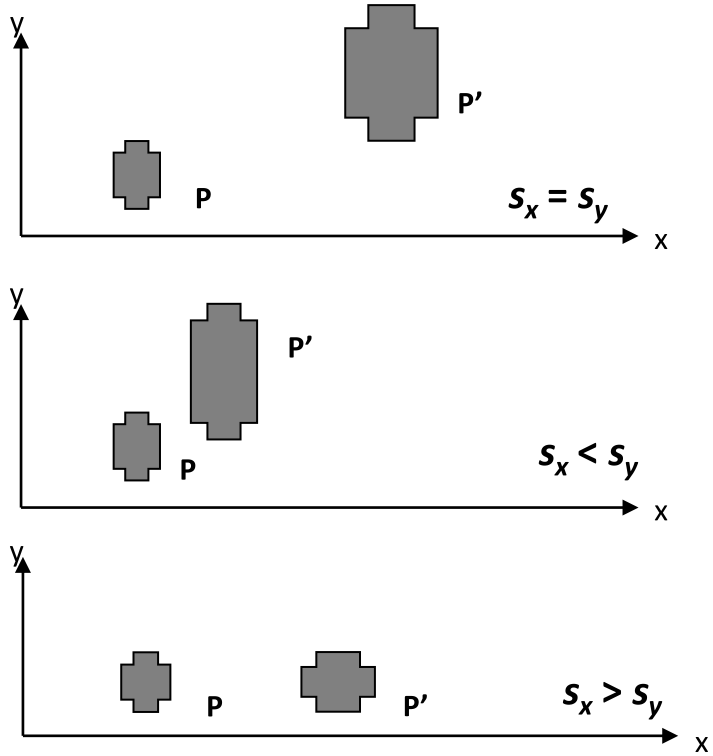
Zapis macierzowy:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} d_x \\ d_y \end{bmatrix}$$

albo $\boldsymbol{P' = P + T}$

gdzie: $\boldsymbol{T}$ – macierz przekształcenia

Skalowanie – zmiana rozmiaru



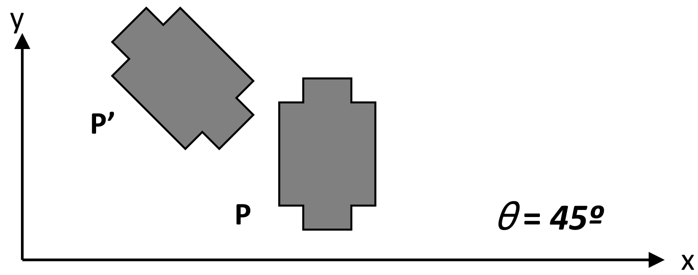
Zapis macierzowy:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \qquad \boldsymbol{P' = SP}$$

gdzie: $\boldsymbol{S}$ = [s_x, s_y] – macierz przekształcenia, s_x i s_y - współczynniki skalowania

Obrót – zmiana orientacji

Obrót o kąt θ wokół początku układu współrzędnych



Zapis macierzowy:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

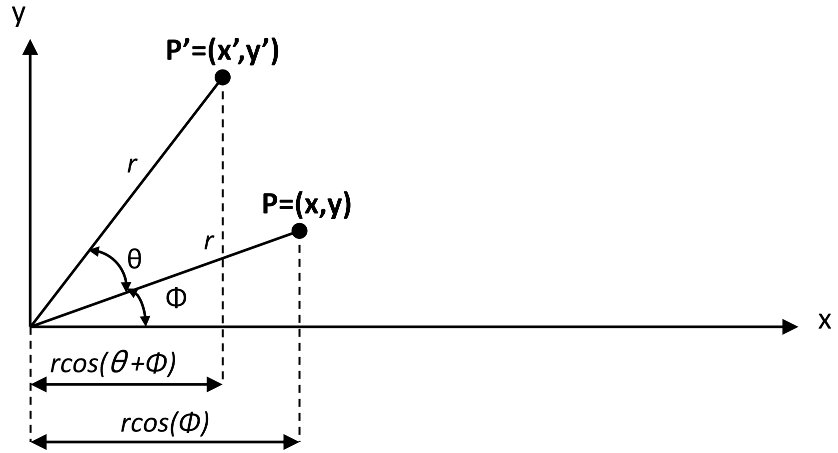
albo

$$\boldsymbol{P' = RP}$$

gdzie: $\boldsymbol{R}$ macierz przekształcenia

Obrót – zmiana orientacji

Obrót o kąt θ wokół początku układu współrzędnych



$$x = r\cos\Phi, \qquad y = r\sin\Phi \qquad (1)$$

$$x' = r\cos(\theta+\Phi) = r\cos\Phi\cos\theta - r\sin\Phi\sin\theta \qquad (2)$$

$$y' = r\sin(\theta+\Phi) = r\cos\Phi\sin\theta + r\sin\Phi\cos\theta \qquad (3)$$

Po podstawieniu *(1)* do *(2)* i *(3)* otrzymamy:

$$x' = x\cos\theta - y\sin\theta \;,\; y' = x\sin\theta + y\cos\theta \qquad (4)$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Współrzędne jednorodne

Współrzędne punktów:

$$P = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \qquad P' = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}$$

trzecia współrzędna *w=1*

Macierze przekształceń:

1. Translacja

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & d_x \\ 0 & 1 & d_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

P′ = TP

2. Skalowanie

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

P′ = SP

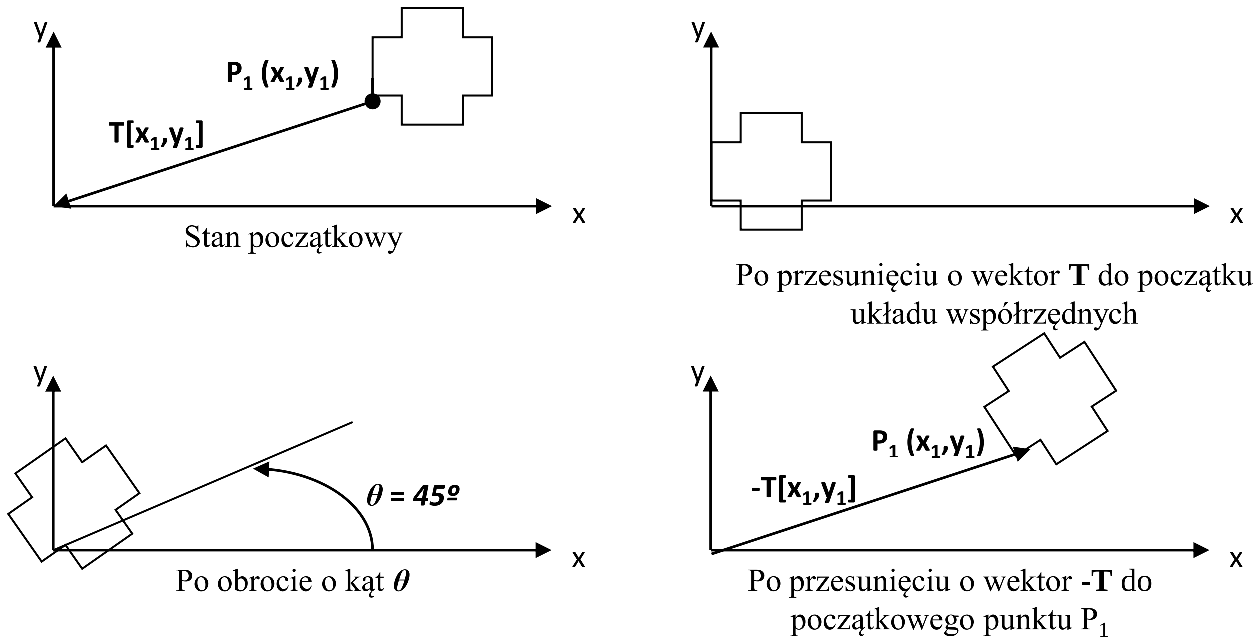
3. Obrót

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

P′ = RP

Składanie przekształceń 2D

<u>Obrót obiektu wokół dowolnego punktu $\mathbf{P_1}$ o kąt θ</u>



Macierz przekształcenia:

$$T(x_1, y_1)R(\theta)T(-x_1,-y_1) = \begin{bmatrix} 1 & 0 & x_1 \\ 0 & 1 & y_1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -x_1 \\ 0 & 1 & -y_1 \\ 0 & 0 & 1 \end{bmatrix}$$

Składanie przekształceń 2D

<u>Skalowanie obiektu względem dowolnego punktu **P₁** [*x₁*,*y₁*]</u>

Kolejne kroki:

1. Przesunięcie o wektor **T**[x_1,y_1] tak aby punkt $\mathbf{P_1}$ znalazł się w początku układu współrzędnych
2. Skalowanie ze współczynnikami s_x, s_y
3. Przesunięcie o wektor **–T**

Macierz przekształcenia:

$$T(x_1, y_1)S(s_x, s_y)T(-x_1, -y_1) = \begin{bmatrix} 1 & 0 & x_1 \\ 0 & 1 & y_1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -x_1 \\ 0 & 1 & -y_1 \\ 0 & 0 & 1 \end{bmatrix}$$

Składanie przekształceń 2D

Skalowanie i obrót względem punktu **P₁** [x_1,y_1] będącym środkiem obrotu i skalowania i przesunięcie obiektu do punktu **P₂** [x_2,y_2]

Kolejne kroki:

1. Przesunięcie o wektor **T**[x_1,y_1] tak aby punkt $\mathbf{P_1}$ znalazł się w początku układu współrzędnych
2. Skalowanie ze współczynnikami s_x, s_y
3. Obrót o kąt $\boldsymbol{\theta}$ wokół początku układu współrzędnych
4. Przesunięcie o wektor **T**[x_2,y_2] do punktu $\mathbf{P_2}$

Macierz przekształcenia:

$$T(x_1,y_1)S(s_x,s_y)R(\theta)T(x_2,y_2)=\begin{bmatrix}1&0&x_1\\0&1&y_1\\0&0&1\end{bmatrix}\begin{bmatrix}s_x&0&0\\0&s_y&0\\0&0&1\end{bmatrix}\begin{bmatrix}\cos\theta&-\sin\theta&0\\\sin\theta&\cos\theta&0\\0&0&1\end{bmatrix}\begin{bmatrix}1&0&x_2\\0&1&y_2\\0&0&1\end{bmatrix}$$

Składanie przekształceń 2D

Przypadki przemienności mnożenia macierzy

$$M_1 M_2 = M_2 M_1$$

M_1	M_2
Translacja	*Translacja*
Skalowanie	*Skalowanie*
Obrót	*Obrót*
Skalowanie z $s_x = s_y$	*Obrót*

W tych przypadkach nie musimy dbać o kolejność składania macierzy

Własności przekształceń odwrotnych

$\boldsymbol{T}^{-1}(x,y) = \boldsymbol{T}(-x,-y)$

$\boldsymbol{R}^{-1}(\theta) = \boldsymbol{R}(-\theta)$

$\boldsymbol{S}^{-1}(s_x,s_y) = \boldsymbol{S}(1/s_x, 1/s_y)$

Składanie przekształceń 2D

Przypadki przemienności mnożenia macierzy

$$\boldsymbol{M_1 M_2 = M_2 M_1}$$

$\boldsymbol{M_1}$	$\boldsymbol{M_2}$
Translacja	*Translacja*
Skalowanie	*Skalowanie*
Obrót	*Obrót*
Skalowanie z $s_x = s_y$	*Obrót*

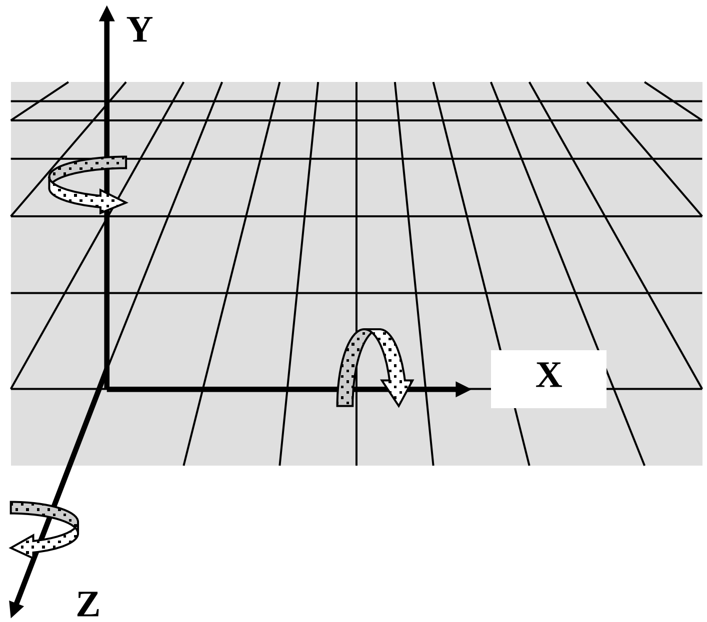
W tych przypadkach nie musimy dbać o kolejność składania macierzy

Własności przekształceń odwrotnych

$$\boldsymbol{T}^{-1}(x,y) = \boldsymbol{T}(-x,-y)$$
$$\boldsymbol{R}^{-1}(\theta) = \boldsymbol{R}(-\theta)$$
$$\boldsymbol{S}^{-1}(s_x,s_y) = \boldsymbol{S}(1/s_x,1/s_y)$$

Układy współrzędnych 3D

Prawoskrętny układ współrzędnych (wykorzystywany w OpenGL)

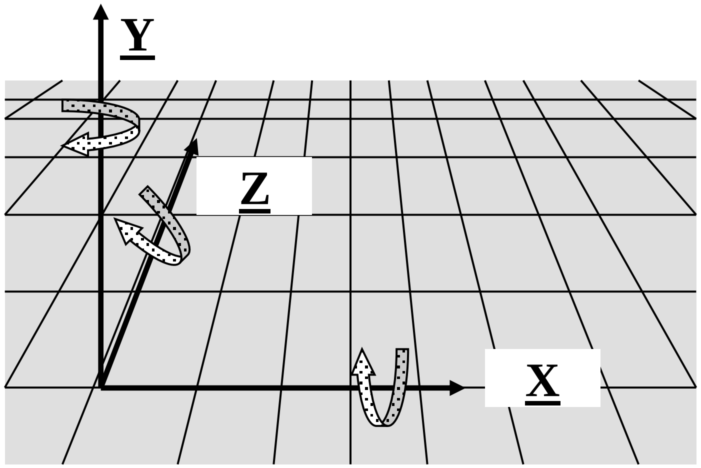


Dodatnie kąty obrotów - w kierunku przeciwnym do ruchu wskazówek zegara, gdy patrzymy od strony dodatniej osi w kierunku początku układu

CCW – *Counter Clock Wise*

Układy współrzędnych 3D

Lewoskrętny układ współrzędnych



Dodatnie kąty obrotów - w kierunku zgodnym do ruchu wskazówek zegara, gdy patrzymy od strony dodatniej osi w kierunku początku układu

CW – *Clock Wise*

Macierz konwersji układów

$$M_{R \to L} = M_{L \to R} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Przekształcenia geometryczne 3D

Przekształcenia geometryczne można przedstawić za pomocą macierzy, przez którą należy pomnożyć współrzędne wierzchołka którego chcemy zmodyfikować. Ogólna postać takiej macierzy to:

W zapisie współrzędnych punktu za pomocą wektora wierszowego:

$$[\mathrm{x}' \quad \mathrm{y}' \quad \mathrm{z}' \quad 1] = [\mathrm{x} \quad \mathrm{y} \quad \mathrm{z} \quad 1] \cdot \begin{bmatrix} M_{11} & M_{12} & M_{13} & M_{14} \\ M_{21} & M_{22} & M_{23} & M_{24} \\ M_{31} & M_{32} & M_{33} & M_{34} \\ M_{41} & M_{42} & M_{43} & M_{44} \end{bmatrix}$$

W zapisie współrzędnych punktu za pomocą wektora kolumnowego:

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} M_{11} & M_{21} & M_{31} & M_{41} \\ M_{12} & M_{22} & M_{32} & M_{42} \\ M_{13} & M_{23} & M_{33} & M_{43} \\ M_{14} & M_{24} & M_{34} & M_{44} \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Przy przejściu z jednej konwencji zapisu do drugiej należy dokonać transponowania macierzy

$$(P \cdot M)^T = M^T \cdot P^T$$

Przekształcenia geometryczne 3D

Przekształcenia geometryczne dzielimy na:

- przekształcenia modelujące
  - translacja
  - skalowanie
  - obroty
- przekształcenia rzutujące
  - rzutowanie perspektywiczne
  - rzutowanie ortogonalne (równoległe)

Translacja

Macierz translacji:

$$[x' \quad y' \quad z' \quad 1] = [x \quad y \quad z \quad 1] \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ T_x & T_y & T_z & 1 \end{bmatrix}$$

operację translacji można przedstawić za pomocą wzorów:

$$x' = x * 1 + y * 0 + z * 0 + 1 * T_x = x + T_x$$
$$y' = x * 0 + y * 1 + z * 0 + 1 * T_y = y + T_y$$
$$z' = x * 0 + y * 0 + z * 1 + 1 * T_z = z + T_z$$

gdzie:

- x',y',z' - współrzędne wierzchołka po translacji
- x,y,z - współrzędne wierzchołka przed translacją
- T_x, T_y, T_z - wartości wektora translacji dla kolejnych osi X,Y,Z

[OpenGL: glTranslate* (T_x, T_y, T_z);]

Skalowanie

Macierz skalowania:

$$[x' \quad y' \quad z' \quad 1] = [x \quad y \quad z \quad 1] \cdot \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

za pomocą wzorów:

$x' = x * s_x$
$y' = y * s_y$
$z' = z * s_z$

gdzie:

- x, y, z – współrzędne przed skalowaniem
- x', y', z' – współrzędne po skalowaniu
- s_x, s_y, s_z – współczynniki operacji skalowania

[OpenGL: glScale* (s_x, s_y, s_z);]

Obroty

Macierz obrotu wokół osi X:

$$[x' \quad y' \quad z' \quad 1] = [x \quad y \quad z \quad 1] \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta) & \sin(\theta) & 0 \\ 0 & -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

za pomocą wzorów:

$y' = y_o + (y - y_o)*\mathit{cos}(\theta)+(z-z_o)*\mathit{sin}(\theta);$
$z' = z_o + (z - z_o)*\mathit{cos}(\theta)-(y-y_o)*\mathit{sin}(\theta);$
$x' = x_o$

gdzie:

- x, y, z – współrzędne punktu przed obrotem
- x', y', z' – współrzędne punktu po obrocie
- x_o, y_o, z_o - punkt wokół którego nastąpi obrót
- θ - kąt o jaki nastąpi obrót

Obroty

Macierz obrotu wokół osi Y:

$$\begin{bmatrix} x' & y' & z' & 1 \end{bmatrix} = \begin{bmatrix} x & y & z & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) & 0 \\ 0 & 1 & 0 & 0 \\ \sin(\theta) & 0 & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

za pomocą wzorów:

$x' = x_o + (x - x_o)*\mathit{cos}(\theta)-(z-z_o)*\mathit{sin}(\theta);$
$z' = z_o + (z - z_o)*\mathit{cos}(\theta)+(x-x_o)*\mathit{sin}(\theta);$
$y' = y_o$

gdzie:

- x, y, z – współrzędne punktu przed obrotem
- x', y', z' – współrzędne punktu po obrocie
- x_o, y_o, z_o - punkt wokół którego nastąpi obrót
- θ - kąt o jaki nastąpi obrót

Obroty

Macierz obrotu wokół osi Z:

$$[x' \quad y' \quad z' \quad 1] = [x \quad y \quad z \quad 1] \cdot \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 & 0 \\ -\sin(\theta) & \cos(\theta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

za pomocą wzorów:

$x' = x_o + (x - x_o)*\mathit{cos}(\theta)+(y-y_o)*\mathit{sin}(\theta);$
$y' = y_o + (y - y_o)*\mathit{cos}(\theta)-(x-x_o)*\mathit{sin}(\theta);$
$z' = z_o$

gdzie:

- x, y, z – współrzędne punktu przed obrotem
- x', y', z' – współrzędne punktu po obrocie
- x_o, y_o, z_o - punkt wokół którego nastąpi obrót
- θ - kąt o jaki nastąpi obrót

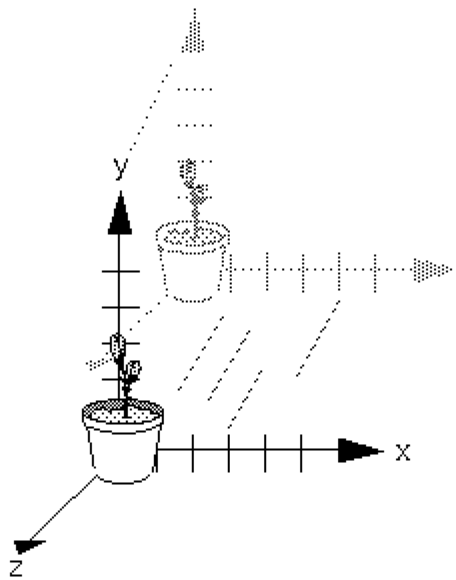
Przekształcenia modelujące w OpenGL

Translacja

glTranslate* (T_x, T_y, T_z);

Przykład:

glTranslatef (*0.0, 0,0, -50.0)*;



Przekształcenia modelujące w OpenGL

Obrót

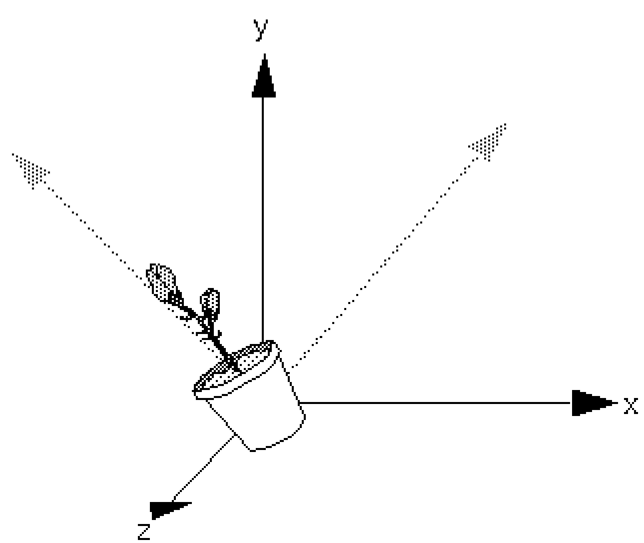
glRotate* (*θ, x, y, z*);

gdzie:

- *θ* – kąt obrotu w stopniach;
- (*x, y, z*) – oś obrotu

Przykład:

glRotatef (*45.0, 0.0, 0.0, 1.0*);



(*x, y, z*) = (*1, 0, 0*) - obrót wokół os X;
(*x, y, z*) = (*0, 1, 0*) - obrót wokół os Y;
(*x, y, z*) = (*0, 0, 1*) - obrót wokół os Z;

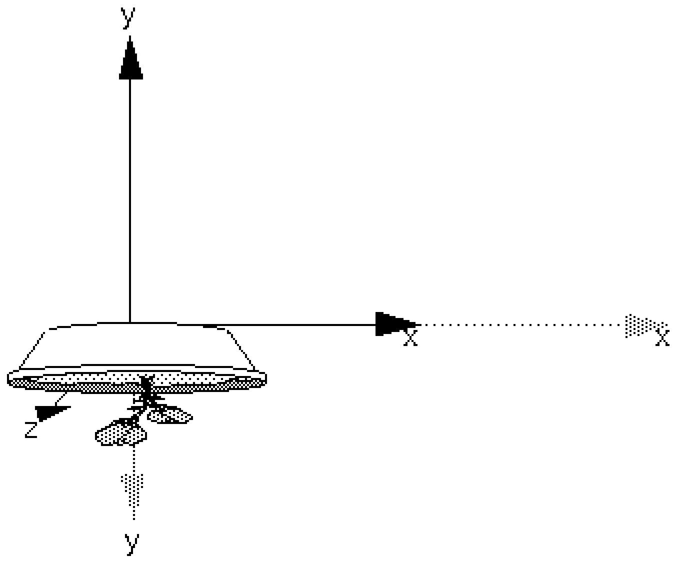
Przekształcenia modelujące w OpenGL

Skalowanie

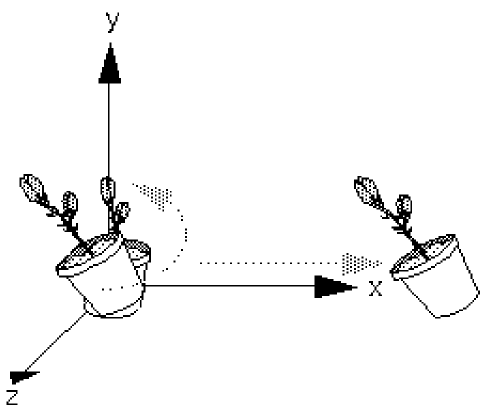
glScale* (s_x , s_y , s_z);

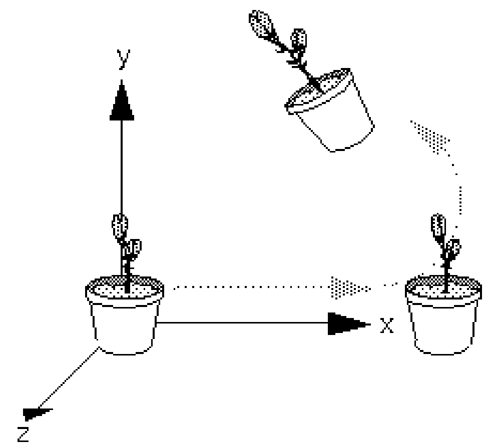
Przykład:

glScalef (*2.0, -0,5, 1.0*);



Składanie przekształceń





a) obrót + translacja

```
{
glTranslatef (50.0, 0.0, 0.0);
glRotatef (45.0, 0.0, 0.0, 1.0);
draw_object();
}
```

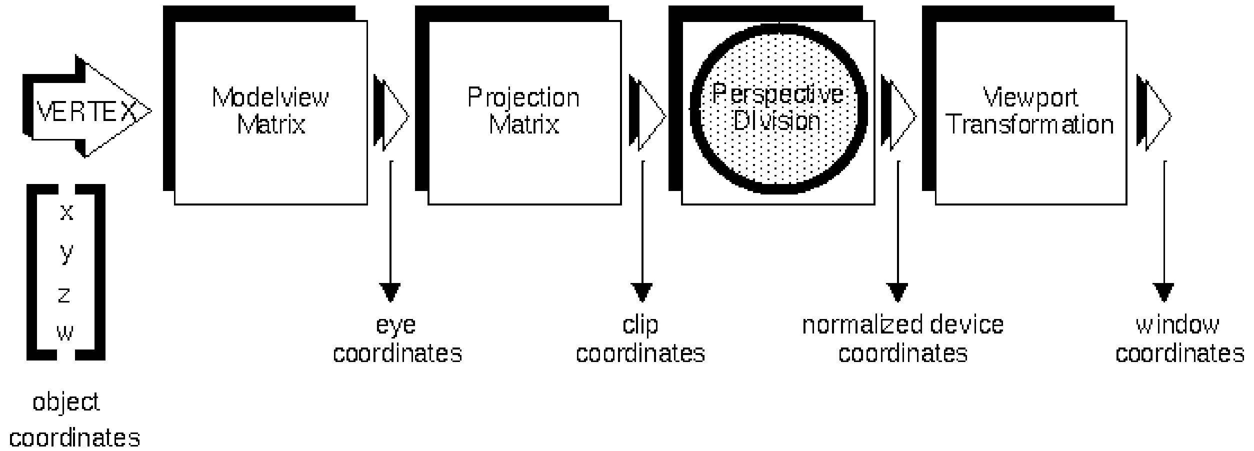
b) translacja + obrót

```
{
glRotatef (45.0, 0.0, 0.0,1.0);
glTranslatef (50.0, 0.0, 0.0);
draw_object();
}
```

<u>Uwaga:</u>

Przekształcenia macierzowe w OpenGL wykonywane są w odwrotnej kolejności niż wynikałoby to z kodu programu. Przekształcenia wykonywane są na macierzy opisującej aktywny (aktualny) układ współrzędnych.

Model procesu rzutowania 3D

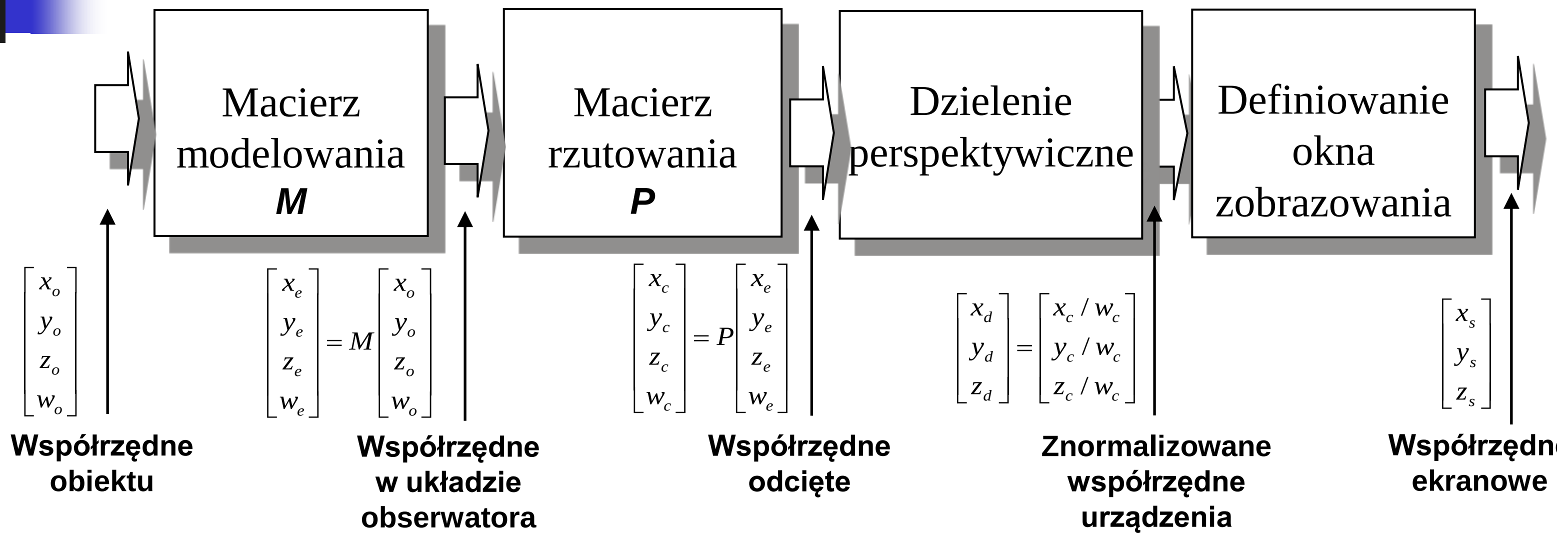


WEJŚCIE: Modele obiektów 3D *(układ współrzędnych obiektu)*

1. Macierz modelowania sceny: *(układ współrzędnych obserwatora)*
   - translacja
   - skalowanie
   - obrót
2. Macierz rzutowania: *(układ współrzędnych odciętych - płaszczyzny odcięcia)*
   - rzut perspektywiczny [OpenGL: glFrustum(); glPerspective();]
   - rzut ortogonalny [OpenGL: glOrtho();]
3. Dzielenie perspektywiczne - przekształcenie normalizujące:
   (układ współrzędnych urządzenia zobrazowania – współrzędne znormalizowane)
4. Definiowanie okna zobrazowania [OpenGL: glVievport();]
   (układ współrzędnych okna zobrazowania - współrzędne ekranowe)

WYJŚCIE: obraz sceny w oknie zobrazowania

Model procesu rzutowania 3D

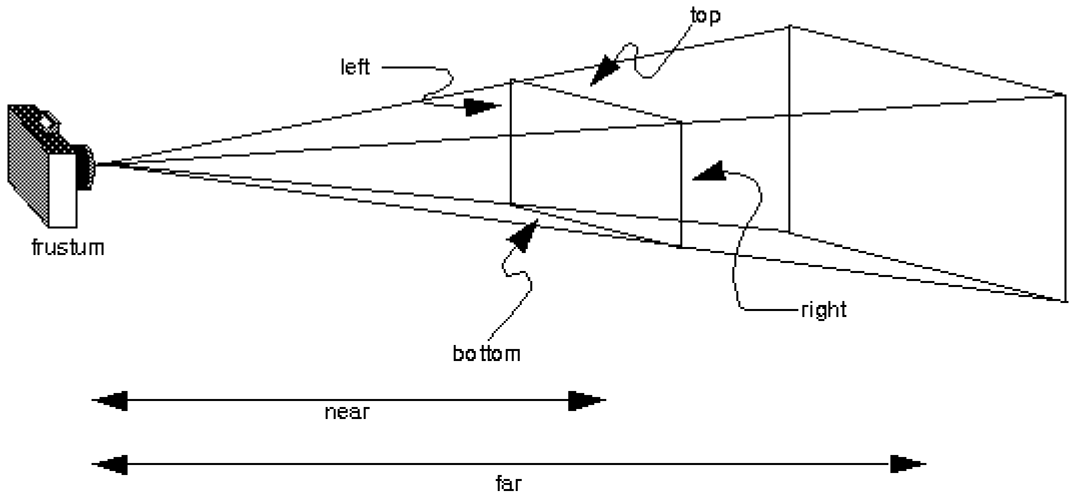


WEJŚCIE: Modele obiektów 3D *(układ współrzędnych obiektu)*

1. Macierz modelowania sceny: *(układ współrzędnych obserwatora)*
   - translacja
   - skalowanie
   - obrót
2. Macierz rzutowania: *(układ współrzędnych odciętych - płaszczyzny odcięcia)*
   - rzut perspektywiczny [OpenGL: glFrustum(); glPerspective();]
   - rzut ortogonalny [OpenGL: glOrtho();]
3. Dzielenie perspektywiczne - przekształcenie normalizujące:
   (układ współrzędnych urządzenia zobrazowania – współrzędne znormalizowane)
4. Definiowanie okna zobrazowania [OpenGL: glVievport();]
   (układ współrzędnych okna zobrazowania - współrzędne ekranowe)

WYJŚCIE: obraz sceny w oknie zobrazowania

Rzutowanie perspektywiczne



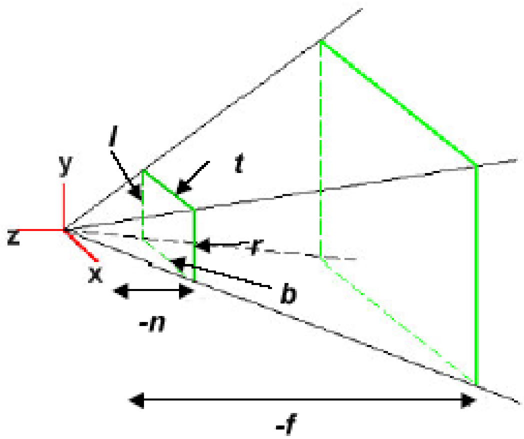
OpenGL: glFrustum*(*left, right, bottom, top, near, far*);

gdzie:

- *left, bottom, near* - współrzędne (*x, y, -z*) lewego dolnego wierzchołka
- *right, top, near* - współrzędne (*x, y, -z*) prawego górnego wierzchołka
- *near, far* – odległość płaszczyzn odcięcia

Funkcja generuje macierz przekształcenia rzutu perspektywicznego ***P***:

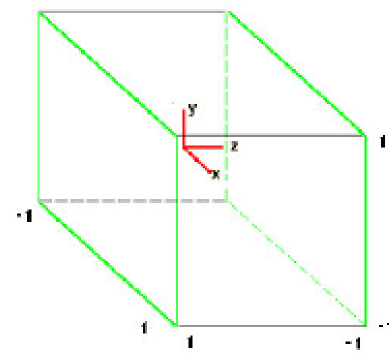
Rzutowanie perspektywiczne



Układ współrzędnych obserwatora

$$\begin{bmatrix} x_c \\ y_c \\ z_c \\ w_c \end{bmatrix} = P \begin{bmatrix} x_e \\ y_e \\ z_e \\ w_e \end{bmatrix}$$

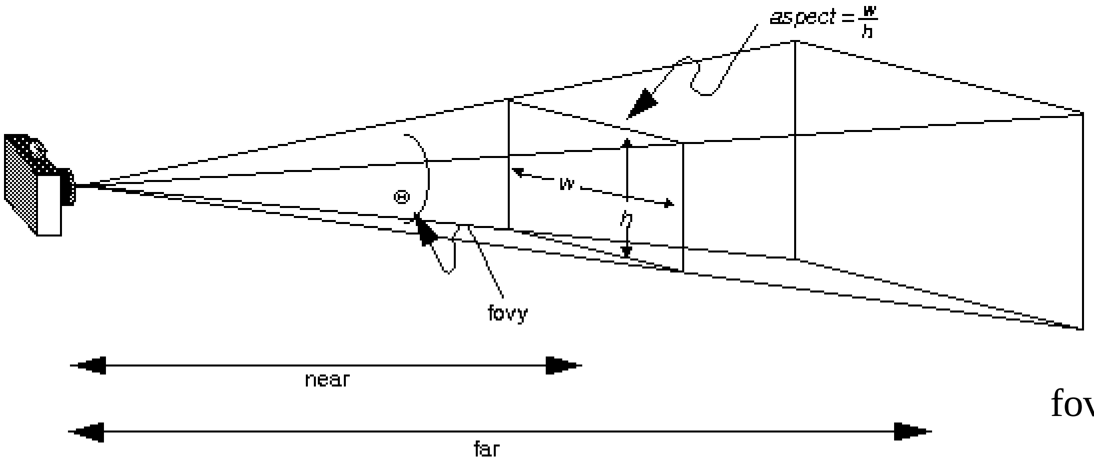
$$\begin{bmatrix} x_d \\ y_d \\ z_d \end{bmatrix} = \begin{bmatrix} x_c / w_c \\ y_c / w_c \\ z_c / w_c \end{bmatrix}$$



Układ znormalizowanych współrzędnych urządzenia

$$\mathrm{glFrustum}(l, r, b, t, n, f) \qquad \mathbf{P} = \begin{bmatrix} \frac{2n}{r-l} & 0 & \frac{r+l}{r-l} & 0 \\ 0 & \frac{2n}{t-b} & \frac{t+b}{t-b} & 0 \\ 0 & 0 & \frac{-(f+n)}{f-n} & \frac{-2fn}{f-n} \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

Rzutowanie perspektywiczne



$$\text{fovy} = 2 \cdot arctg\left(\frac{h}{2 \cdot near}\right)$$

OpenGL: gluPerspective*(*fovy, aspect, near, far***);**

gdzie:

- *fovy* - kąt pola widzenia w płaszczyźnie pionowej YZ [0,180°]
- *aspect* - stosunek szerokości obszaru rzutowania do jego wysokości *w/h*
- *near, far* – odległość płaszczyzn odcięcia (*-z*)

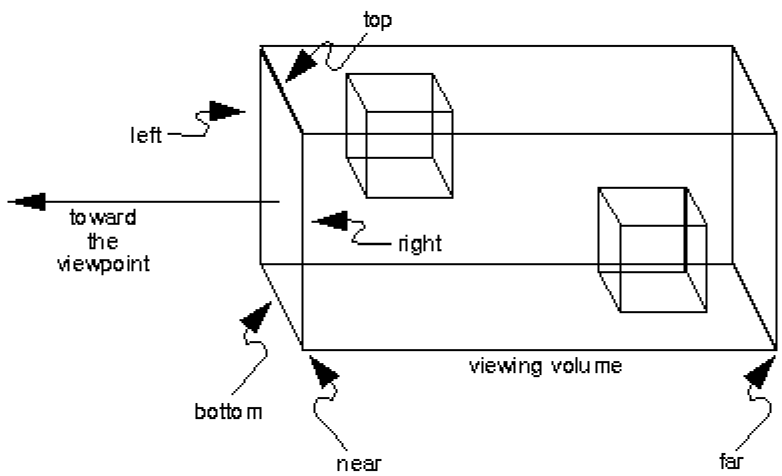
<u>Przykład:</u>

Zdefiniować obszar rzutowania obejmujący cały ekran monitora 15″ (28 cm x 20 cm). Obserwator znajduje się w odległości 80 cm od monitora, obiekty znajdujące się w odległości większej niż 5 m nie będą rysowane.

1. glFrustumf(-28.0/2, 28.0/2, -20.0/2, 20.0/2, 80.0, 500.0);

2. gluPerspectivef(14.25, 1.4, 80.0, 500.0);

Rzutowanie równoległe



OpenGL: glOrtho*(*left, right, bottom, top, near, far***);**

gdzie:

- *left, bottom, near* - współrzędne (*x, y, -z*) lewego dolnego wierzchołka
- *right, top, near* - współrzędne (*x, y, -z*) prawego górnego wierzchołka
- *near, far* – odległość płaszczyzn odcięcia

Funkcja **glOrtho*();** generuje macierz przekształcenia rzutu równoległego ***R***:

Rzutowanie równoległe

Funkcja **glOrtho***(*left, right, bottom, top, near, far*) generuje macierz przekształcenia rzutu równoległego:

$$R = \begin{bmatrix} \frac{2}{r-l} & 0 & 0 & -\frac{r+l}{r-l} \\ 0 & \frac{2}{t-b} & 0 & -\frac{t+b}{t-b} \\ 0 & 0 & \frac{-2}{f-n} & -\frac{f+n}{f-n} \\ 0 & 0 & 0 & 1 \end{bmatrix} \text{ and } R^{-1} = \begin{bmatrix} \frac{r-l}{2} & 0 & 0 & \frac{r+l}{2} \\ 0 & \frac{t-b}{2} & 0 & \frac{t+b}{2} \\ 0 & 0 & \frac{f-n}{-2} & \frac{n+f}{2} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

gdzie:

- *l: left, r : right, b: bottom, t: top, n: near, f: far*
- *$l \neq r$, $t \neq b$, $n \neq f$*

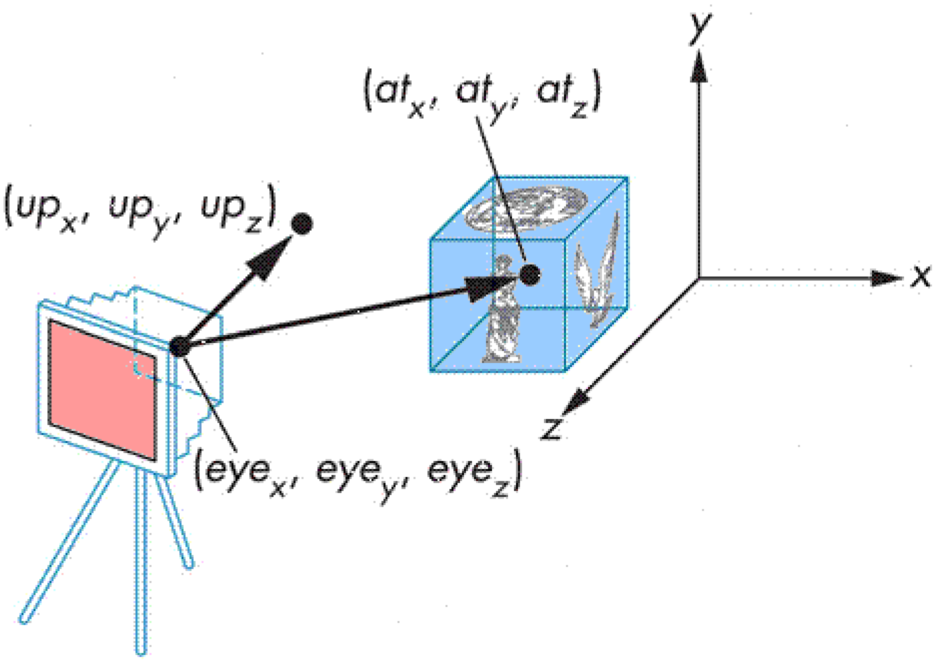
Definiowanie obserwatora

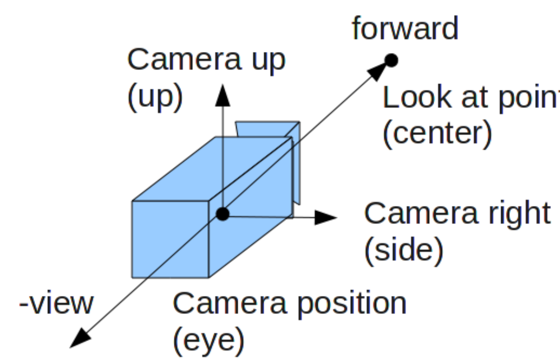
OpenGL: gluLookAt*(eyex, eyey, eyez, centerx, centery, centerz, upx, upy, upz);*

Funkcja umiejscawia obserwatora w punkcie o współrzędnych (*eyex, eyey, eyez*).

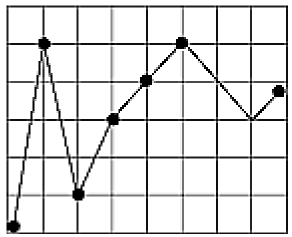
Punkt (*centerx, centery, centerz*) definiuje dowolny punkt leżący na osi widzenia, typowo jest to centralny punkt sceny (*punkt obserwacji*).

Współrzędne (*upx, upy, upz*) wyznaczają wektor obserwacji sceny (*up vector)* w oknie zobrazowania – typowa wartość (*0,1,0*). *Up vector* wykorzystywany jest do definiowania obrotu kamery.

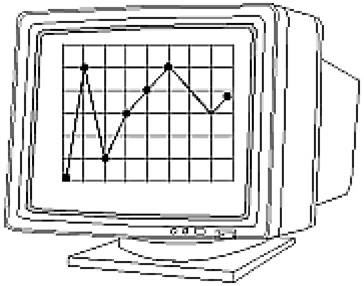




Definiowanie okna zobrazowania



Obszar zobrazowania
(Viewing Volume)



Okno graficzne
(Viewport)

OpenGL: glVievport(*x, y, width, hight***);**

Funkcja służy do zdefiniowania prostokątnego obszaru okna graficznego, w którym odwzorowywany będzie obraz

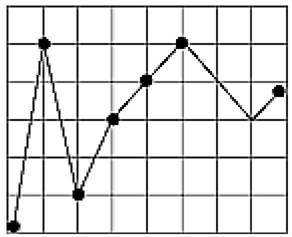
Parametry funkcji:

- *x, y* - współrzędne (*x, y*) lewego dolnego wierzchołka okna względem lewego, dolnego wierzchołka okna zobrazowania ekranu – wartość początkowa (*0, 0*)
- *width, hight* - szerokość i wysokość okna

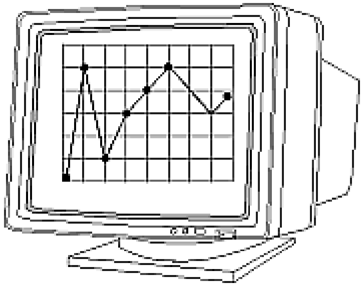
Uwaga:

Wszystkie wartości wyrażone w układzie współrzędnych ekranu (układ współrzędnych pikselowych)

Definiowanie okna zobrazowania



Obszar zobrazowania
(Viewing Volume)



Okno graficzne
(Viewport)

OpenGL: glVievport(*x, y, width, hight***);**

Funkcja służy do zdefiniowania prostokątnego obszaru okna graficznego, w którym odwzorowywany będzie obraz

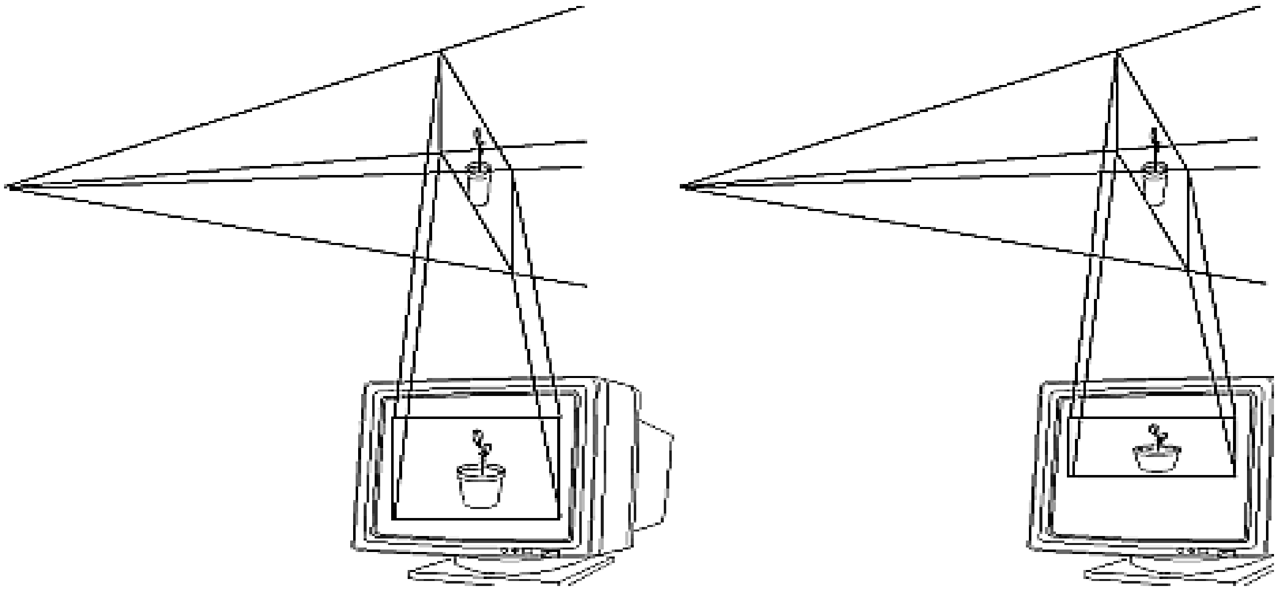
Parametry funkcji:

- *x, y* - współrzędne (*x, y*) lewego dolnego wierzchołka okna względem lewego, dolnego wierzchołka okna zobrazowania ekranu – wartość początkowa (*0, 0*)
- *width, hight* - szerokość i wysokość okna

Uwaga:

Wszystkie wartości wyrażone w układzie współrzędnych ekranu (układ współrzędnych pikselowych)

Odwzorowanie obszaru zobrazowania w okno graficzne



Okno obszaru zobrazowania
(Viewing Volume)

Okno graficzne
(Viewport)

zachowana proporcja
(równe aspekty okien)

brak proporcji