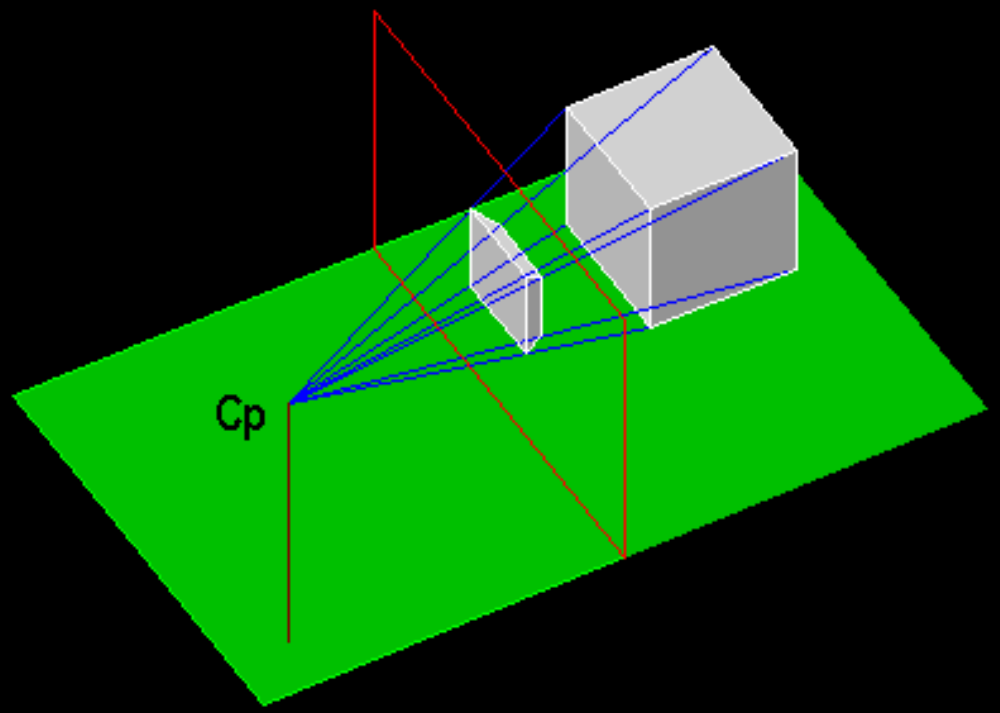
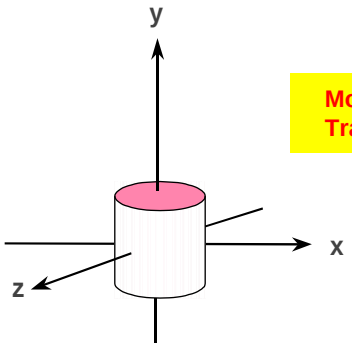
Gledanje - kamera



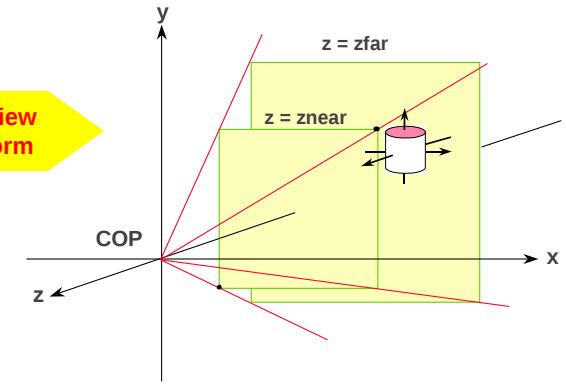
Cevovod gledanja (Graphics Pipeline)

Model Coordinates



Modelview Transform

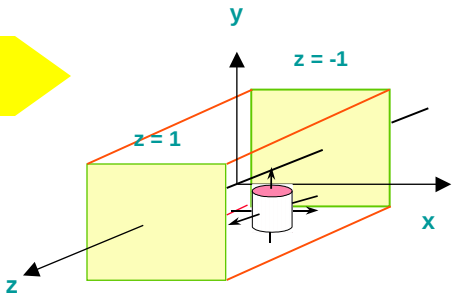
Eye Coordinates




```
glScalef()
glRotatef()
glTranslatef()
```

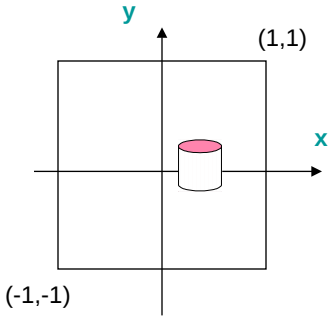
Projection Transform

Normalized Coordinates




```
glFrustrum()
gluPerspective()
```

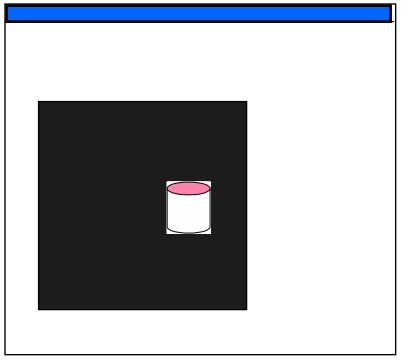
Projected Normalized Coordinates



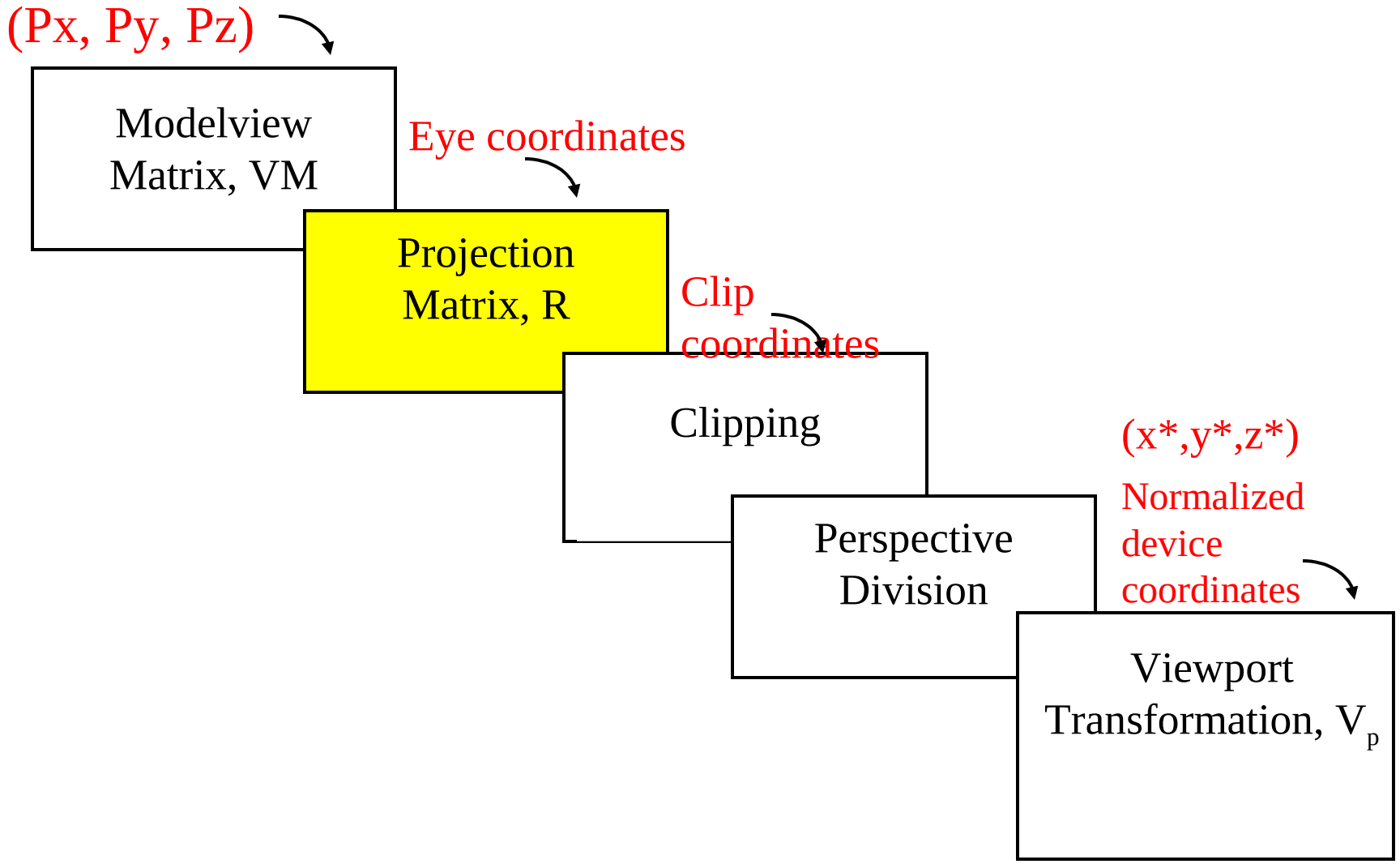
Viewport Transform

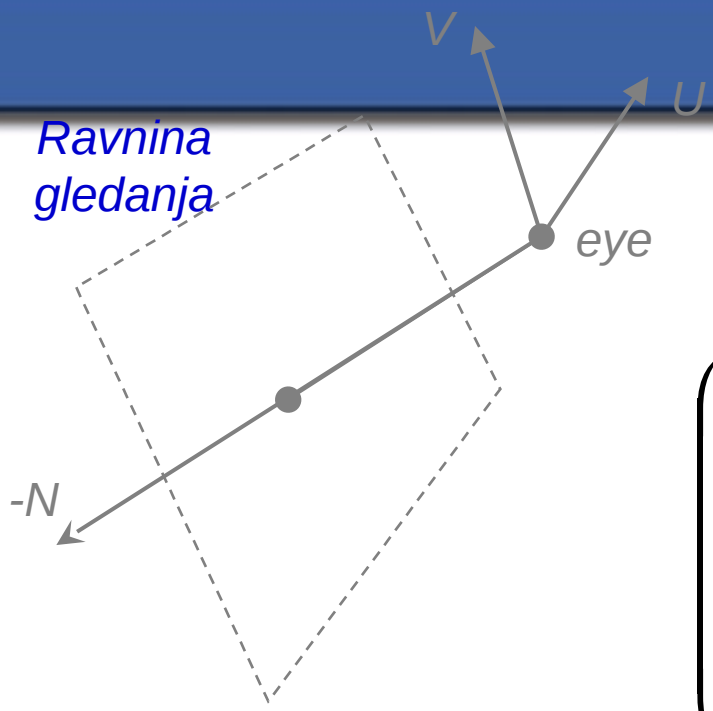
```
glViewport()
```

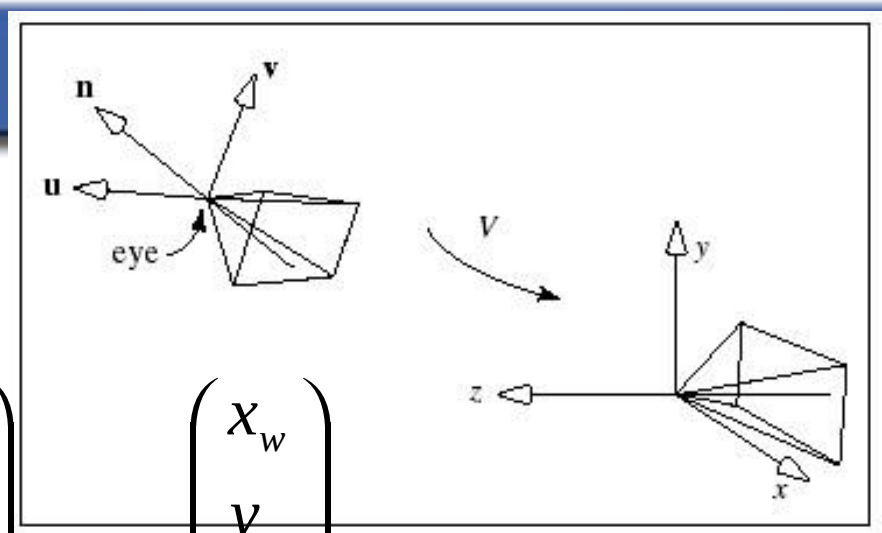
Window Coordinates



Cevovod gledanja (viewing pipeline)







$$\begin{pmatrix} x_v \\ y_v \\ z_v \\ 1 \end{pmatrix} = T_{view} \begin{pmatrix} x_w \\ y_w \\ z_w \\ 1 \end{pmatrix}$$

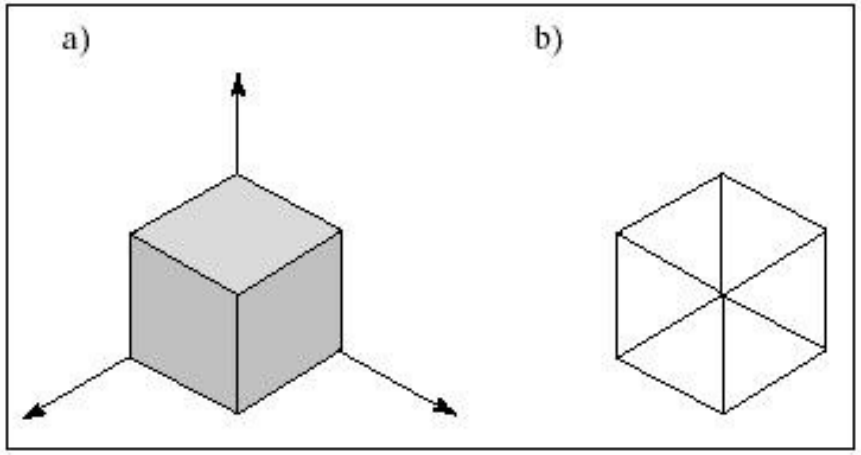
$$T_{view} = RT$$

$$T = \begin{pmatrix} 1 & 0 & 0 & -eVec \cdot U \\ 0 & 1 & 0 & -eVec \cdot V \\ 0 & 0 & 1 & -eVec \cdot N \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

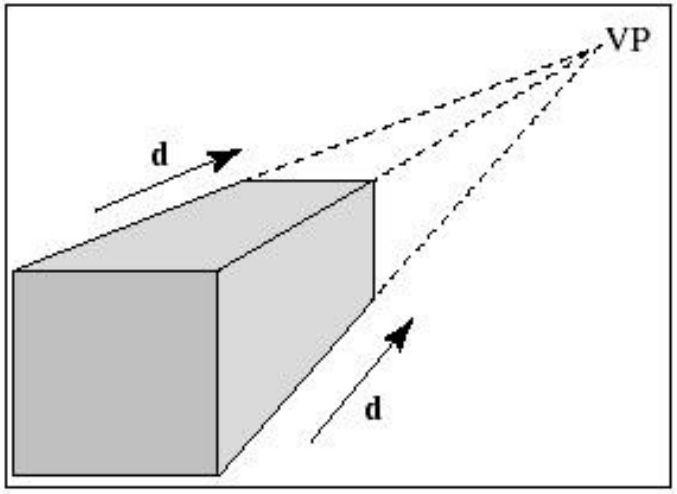
$$R = \begin{pmatrix} U_x & U_y & U_z & 0 \\ V_x & V_y & V_z & 0 \\ N_x & N_y & N_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$eVec = \left(eye - \begin{pmatrix} 0 & 0 & 0 \end{pmatrix} \right)$$

Groba klasifikacija projekcij

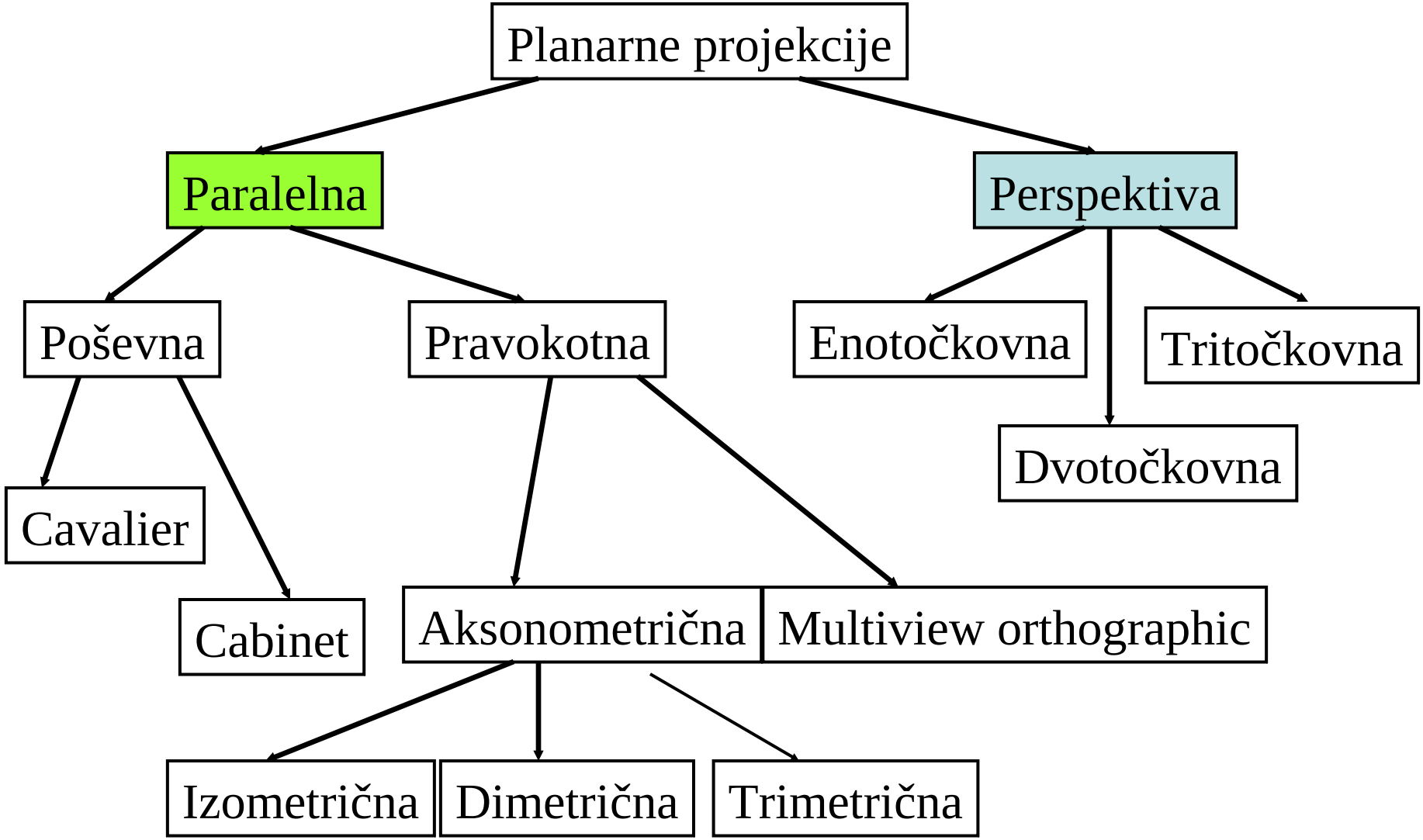


Paralelna



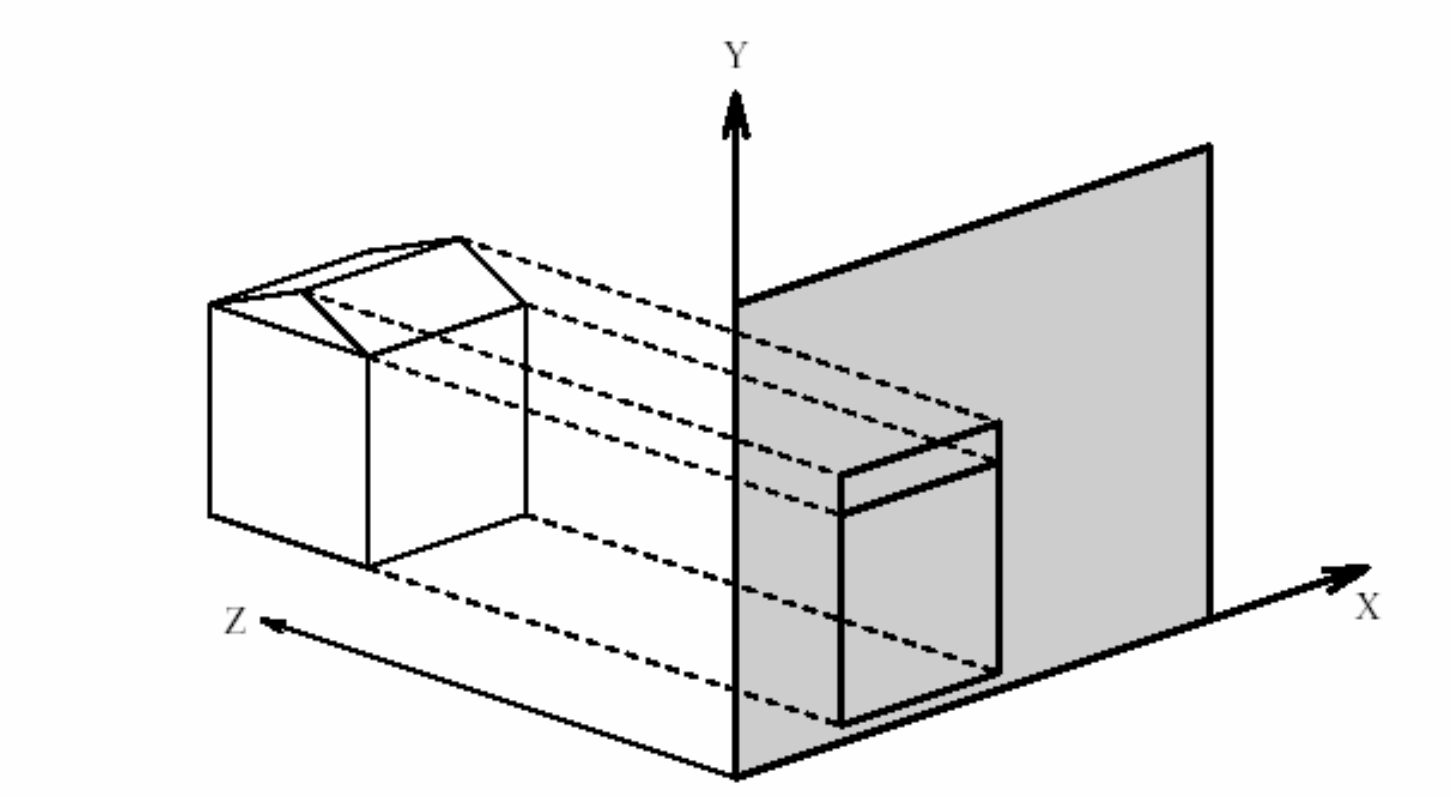
Perspektiva

Klasifikacija planarnih projekcij



Pravokotna (ortografska) projekcija

Je najbolj enostavna projekcija



Ortografska projekcija

Orthographic Projection

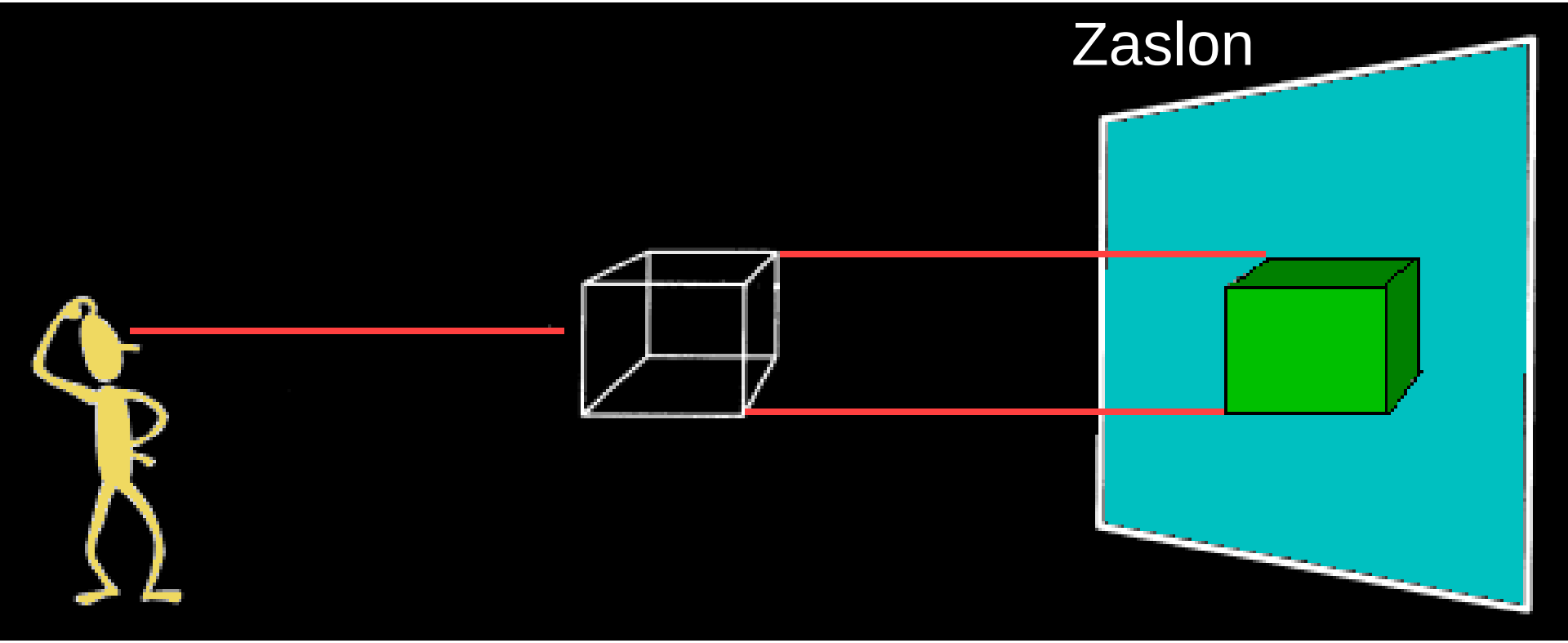
We define a vector (x_p, y_p, z_p) along which we will project the image. Each scene point (x, y, z) is projected in parallel along this vector until it strikes the X-Y viewing plane at point (x_{pl}, y_{pl}).

$$\begin{aligned} x_{pl} &= x - \frac{z}{z_p} x_p \\ y_{pl} &= y - \frac{z}{z_p} y_p \\ z_{pl} &= 0 \end{aligned}$$

When the projection vector is $(0, 0, -1)$ (i.e., projection along the $-Z$ axis), this reduces to

$$\begin{aligned} x_{pl} &= x \\ y_{pl} &= y \\ z_{pl} &= 0 \end{aligned}$$

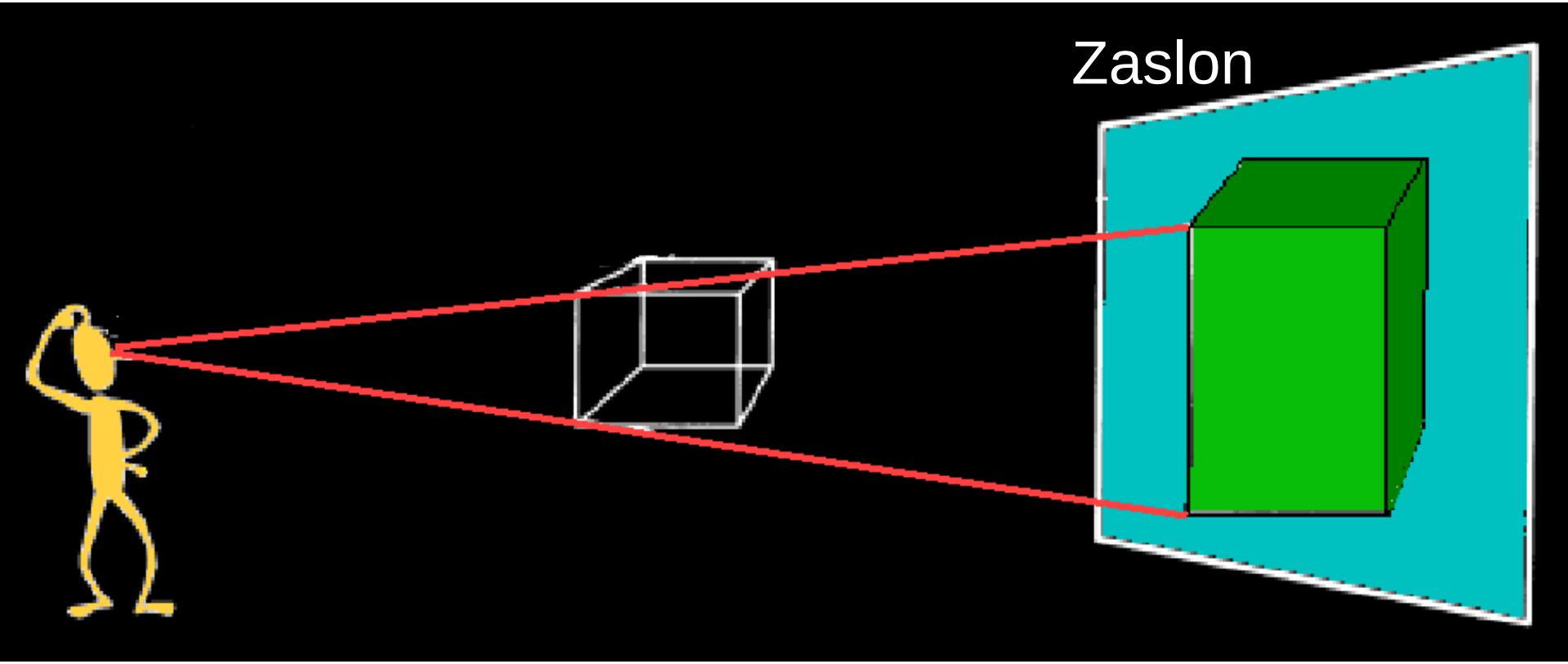
Paralelna projekcija



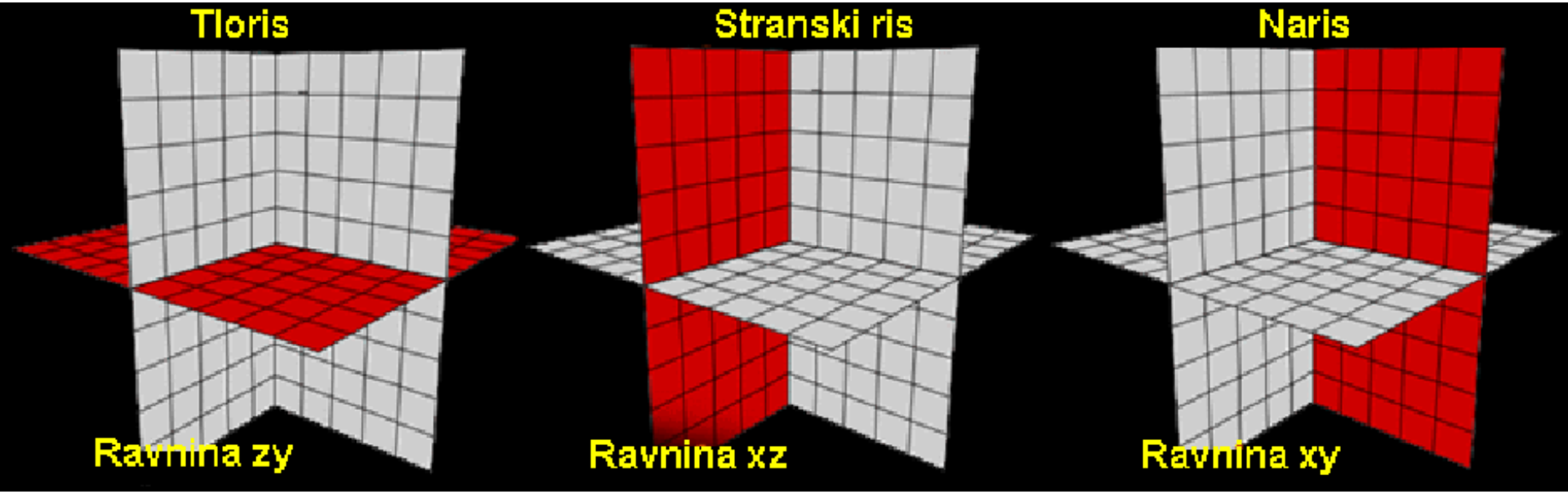
Dodajanje perspektive pogledu je le vprasanje skaliranja višine in širine predmeta, odvisno od tega, koliko je predmet oddaljen od zaslona.

Video

Perspektiva

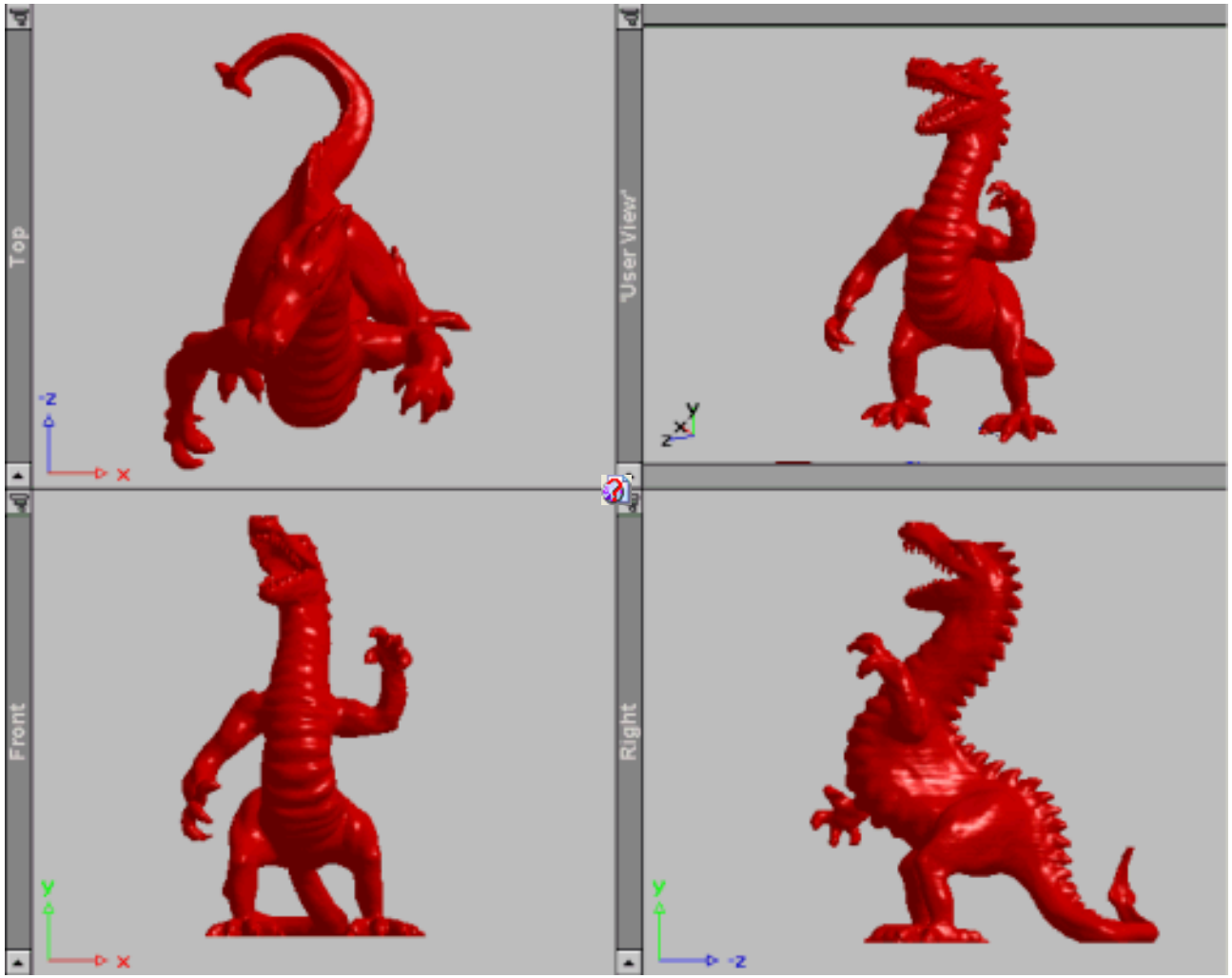


Pogledi - projekcije na ravnine



Vzporedne poglede dobimo običajno s pravokotno, vzporedno projekcijo karakterističnih točk predmetov na eno od ravnin xz, yz ali xz

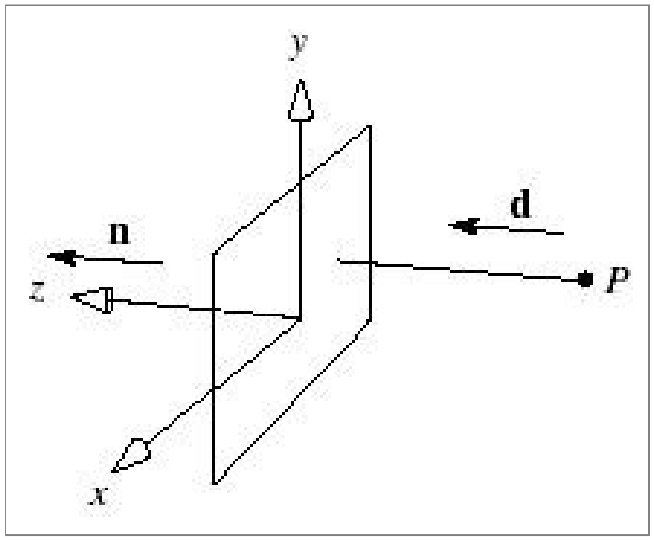
Pogledi v tipičnem modelirniku

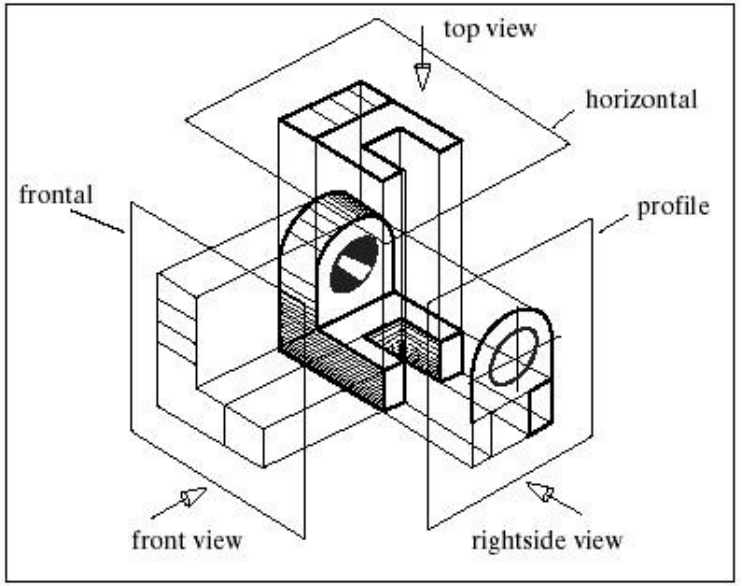


Sprednji, stranski-desni, tloris-zgornji pogled so vzporedni projekcijski pogledi.

Črte v teh pogledih ne konvergirajo.Razdalja med objektom in kamero ne vpliva na velikost objekta, oddaljeni in bliznji predmeti so enako veliki.

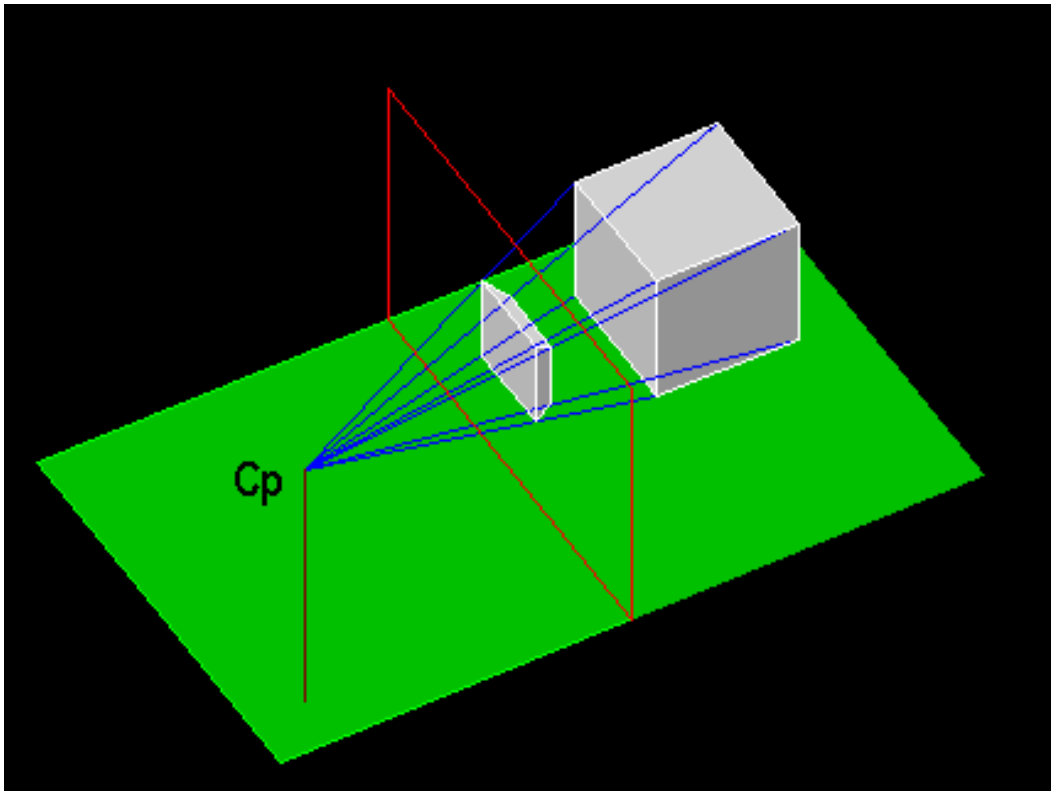
Paralelne pravokotne projekcije (ortografske)





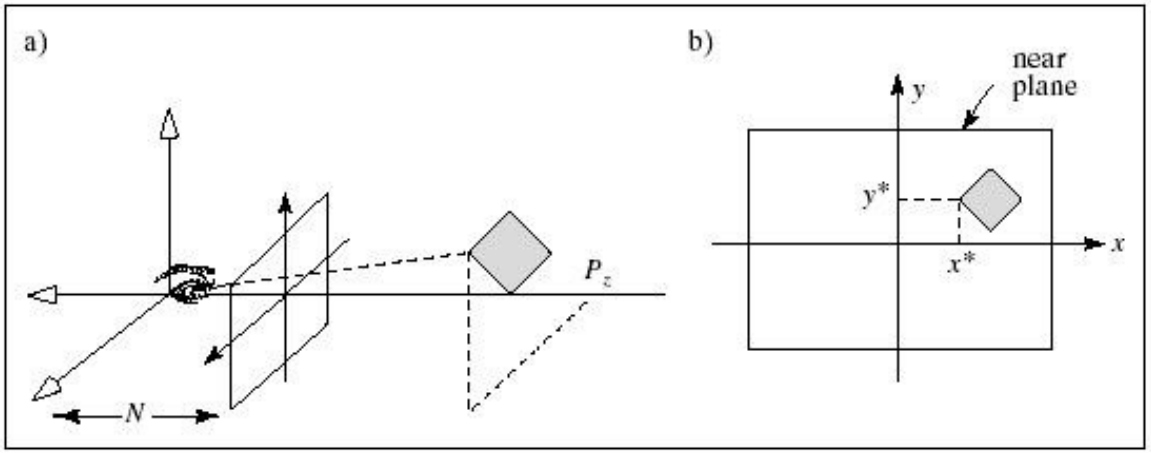
$$T_{proj} = T_{ort} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Perspektivna projekcija



Perspektivna projekcija vsebuje določen Cp in oponaša naš vid. Pri taki projekciji je velikost predmeta obratnosorazmerna od tega, kako daleč je objekt od CP. Taka projekcija zgleda realistično, vendar ne ohranja kotov in vzporednosti črt.

Perspektivne projekcije



$$X = x_v$$
$$Y = y_v$$
$$Z = z_v$$
$$w = z_v / d$$

$$T_{proj} = T_{pers} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/d & 1 \end{pmatrix}$$

opomba: ***d*** *je na sliki* ***N***

Perspektiva

$$x_s = X/w$$

$$y_s = Y/w$$

$$z_s = Z/w$$

kjer je

$$X = x_v$$

$$Y = y_v$$

$$Z = z_v$$

$$w = z_v/d$$

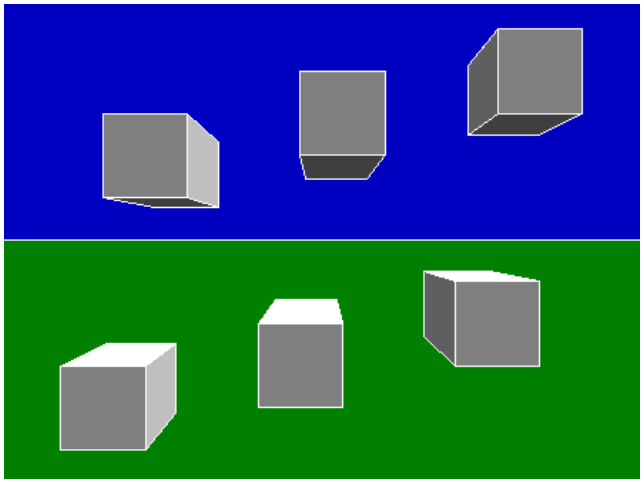
Paralelna (ortografska)

$$x_s = x_v$$

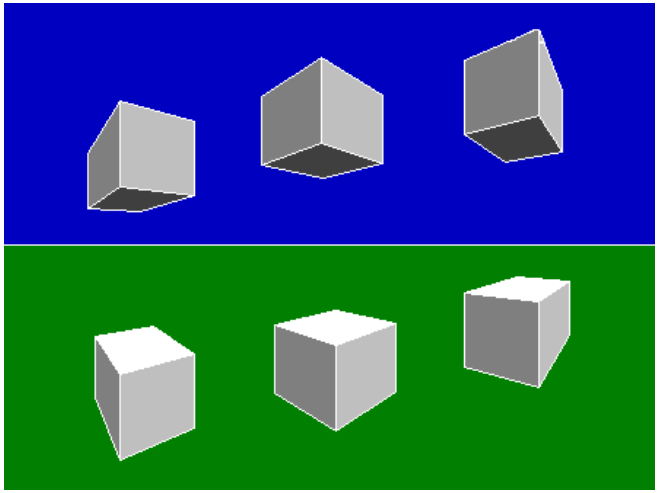
$$y_s = y_v$$

$$z_s = 0$$

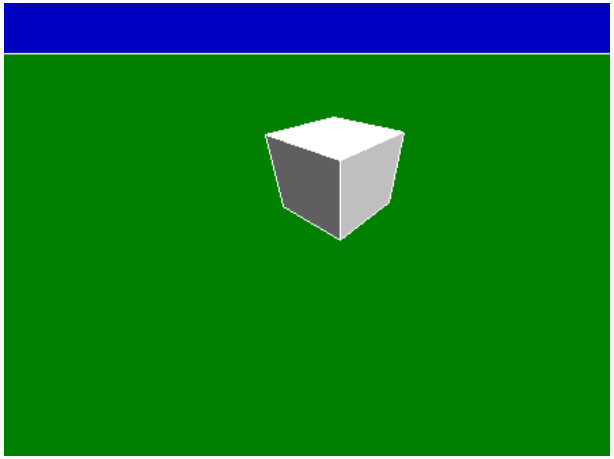
Tipi perspektiv



Enotočkovna perspektiva

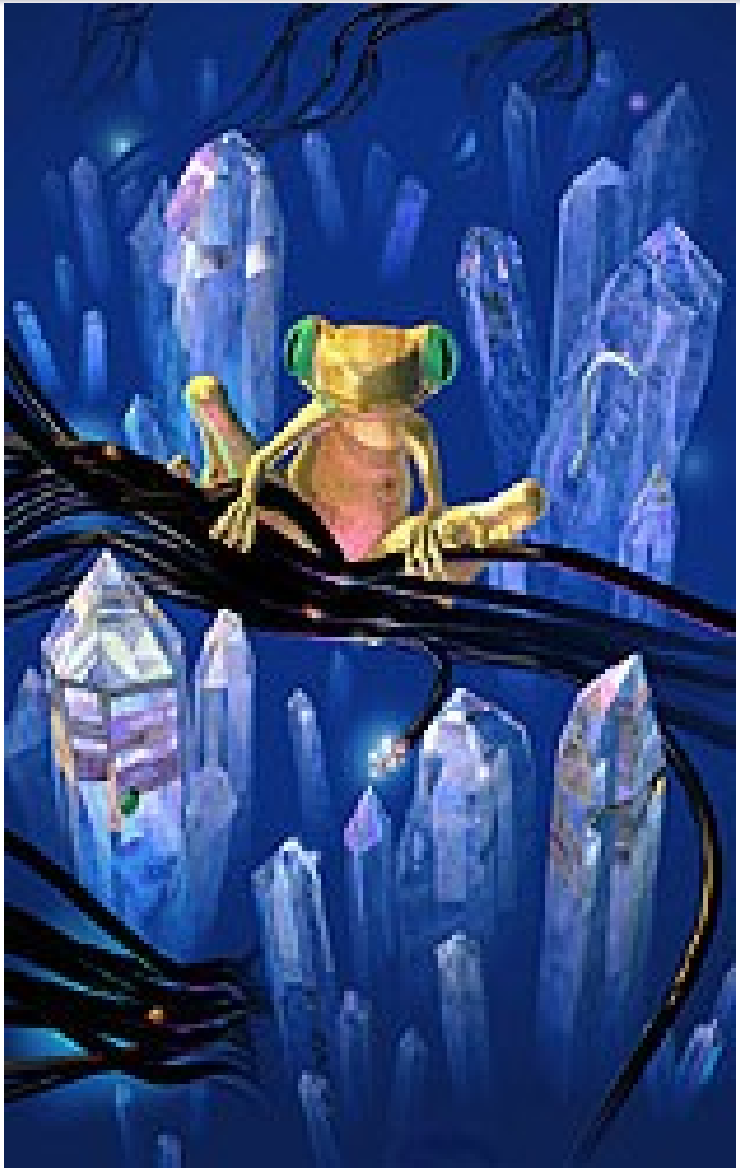


Dvotočkovna perspektiva



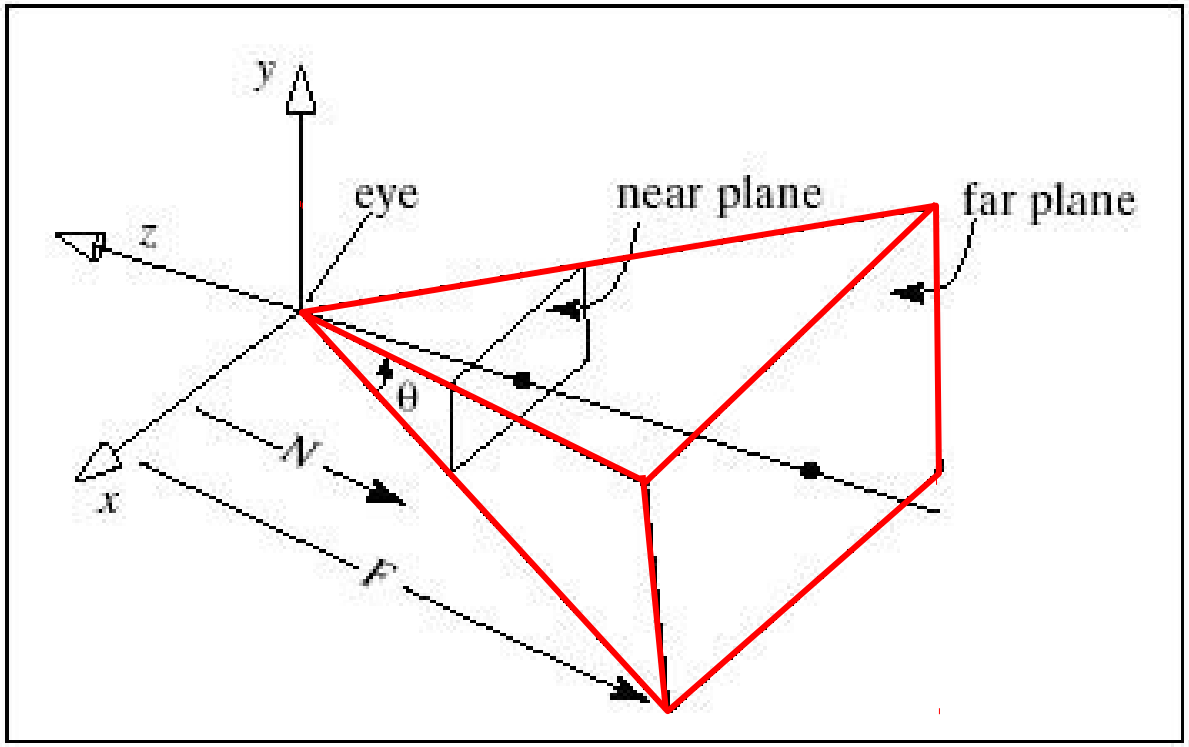
Tritočkovna perspektiva

Lažna perspektiva

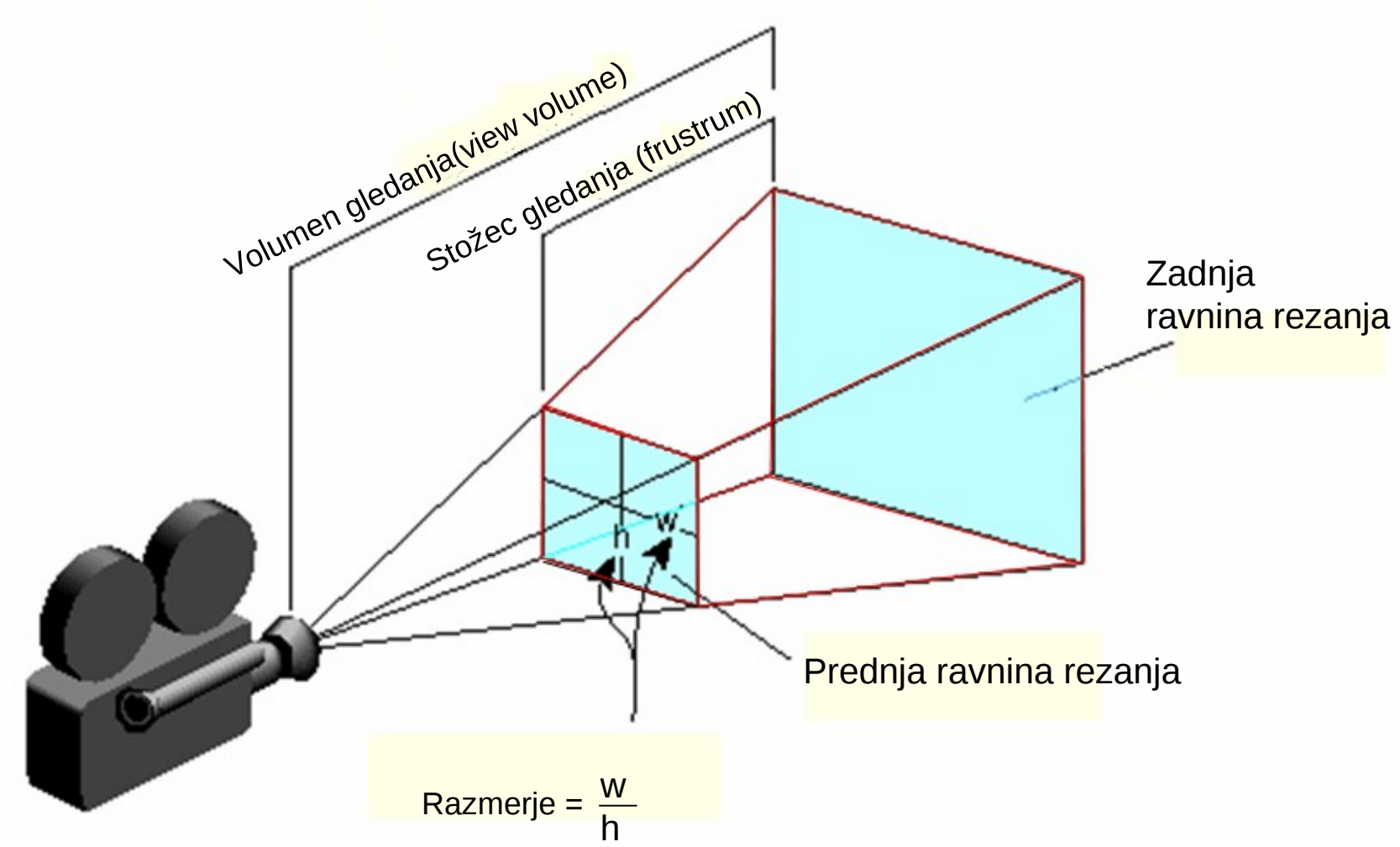


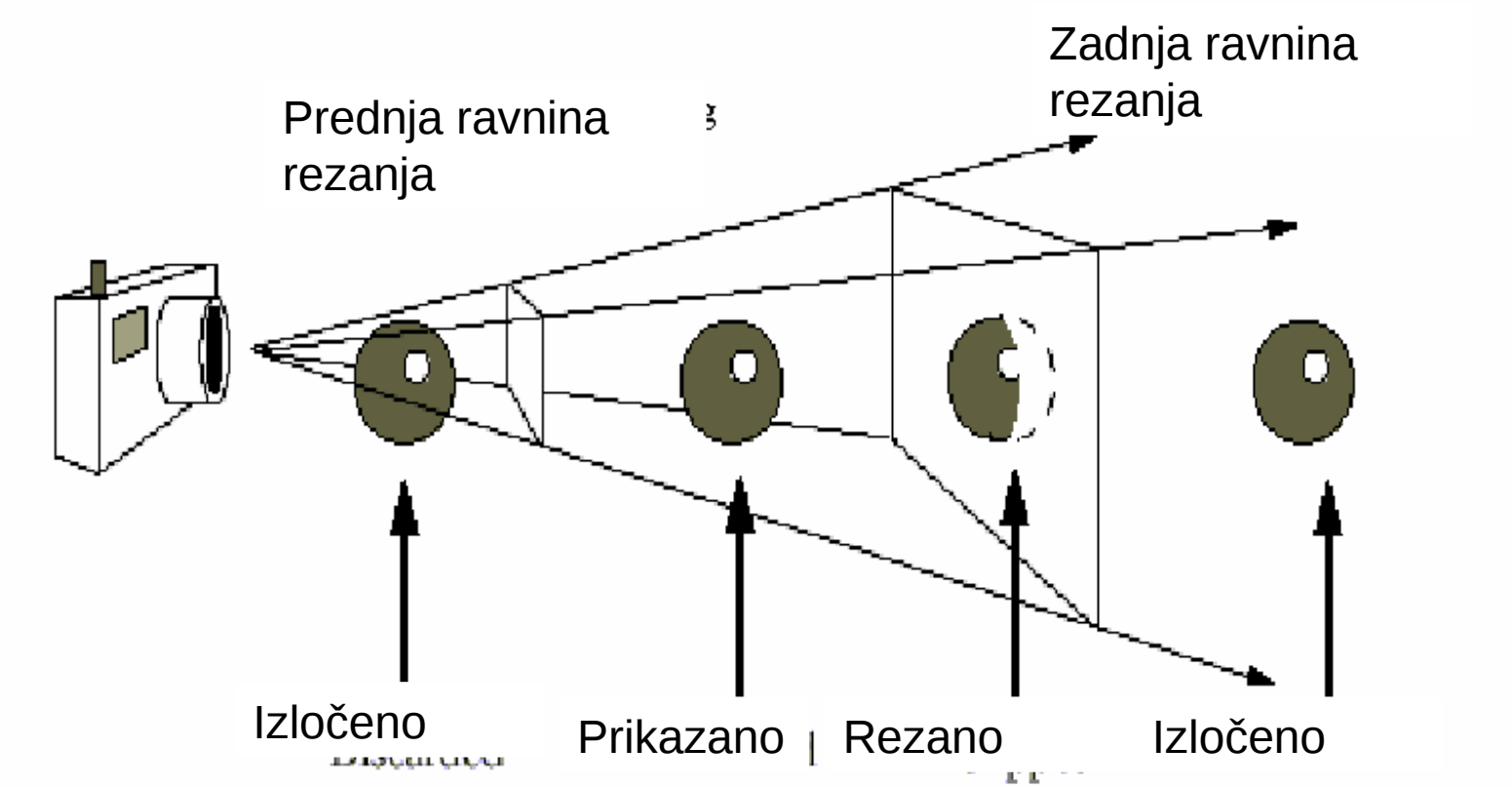
Volumen gledanja

V prejšne preproste transformacije moramo vključiti volumen gledanja (view volume):

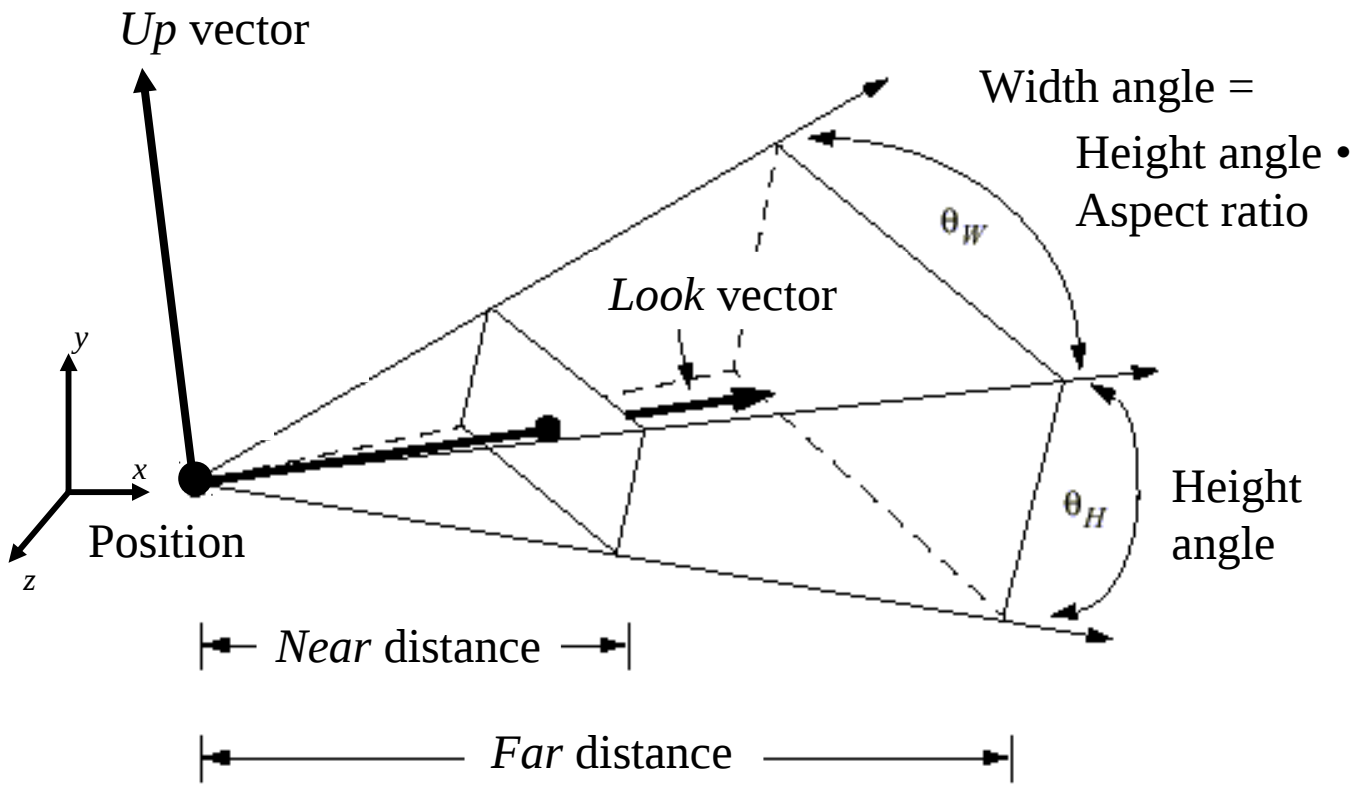


Volumen gledanja, stožec gledanja



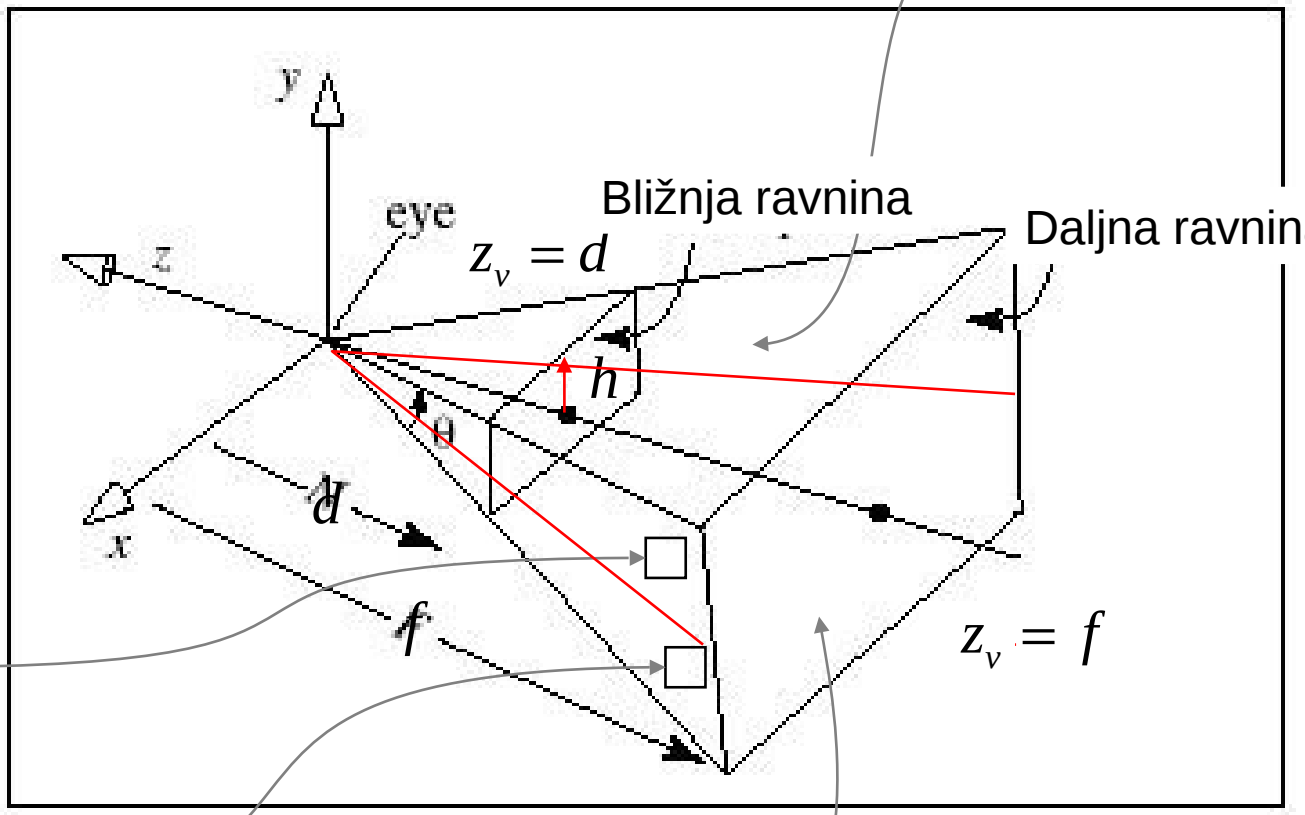

Zadnja ravnina rezanja
Prednja ravnina rezanja
Izločeno
Prikazano
Rezano
Izločeno

Nekaj pojmov



Nekaj matematike

$$y_{vol} = \frac{hz_v}{d}$$



$$x_v = -\frac{hz_v}{d}$$

$$x_v = \frac{hz_v}{d}$$

$$y_v = -\frac{hz_v}{d}$$

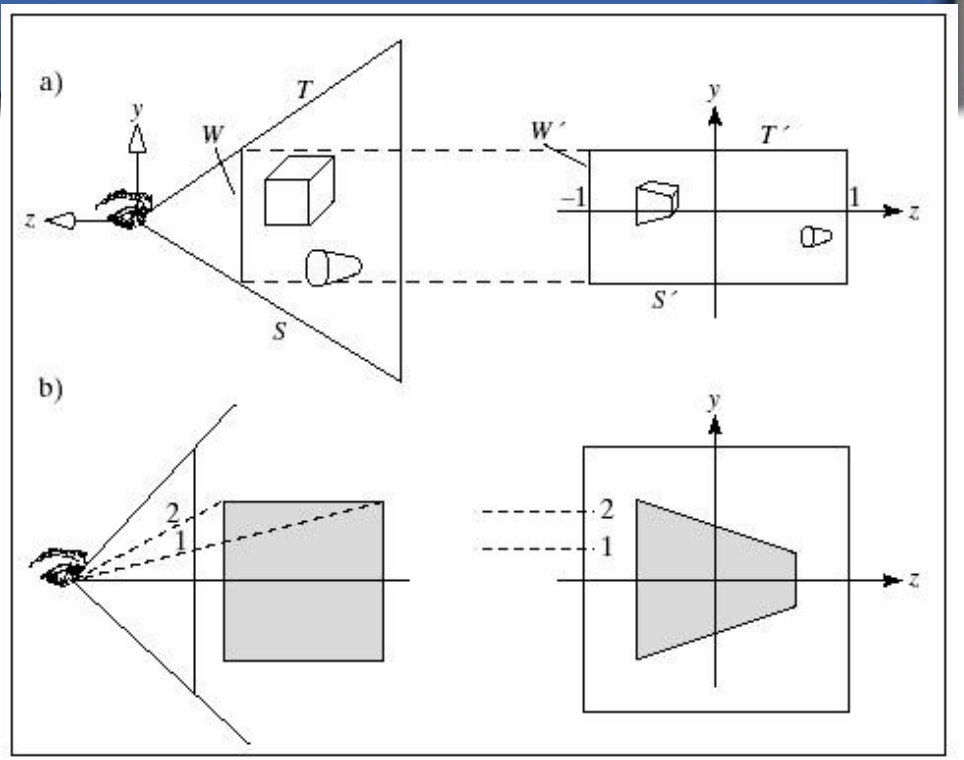
Transformacijske matrike

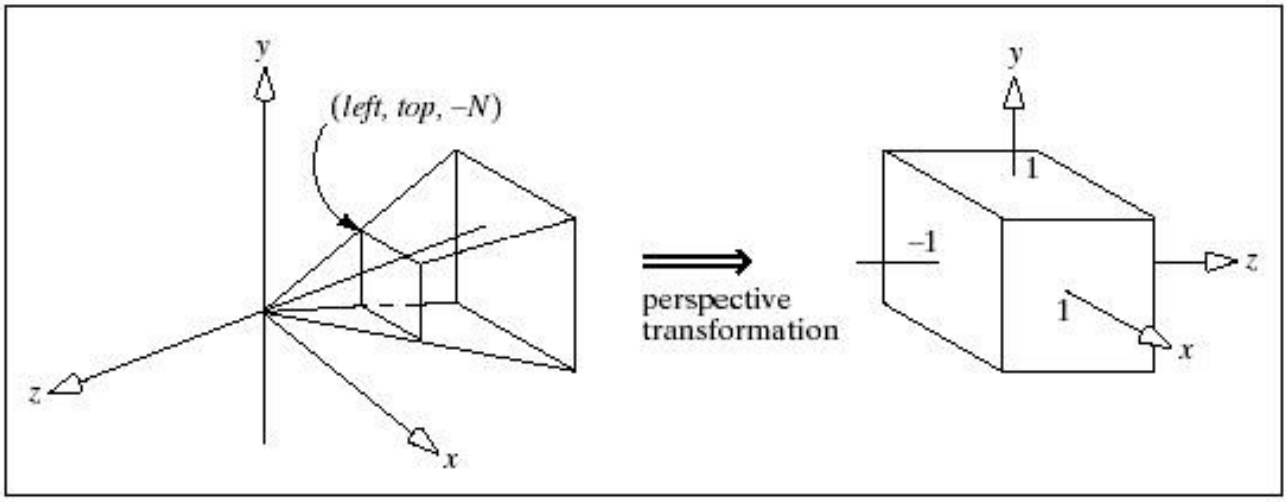
$$T_{pers} = \begin{pmatrix} d/h & 0 & 0 & 0 \\ 0 & d/h & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & f/(f-d) & 1 \\ 0 & 0 & -fd/(f-d) & 0 \end{pmatrix}$$

Skaliranje (d/h v x in y)
Prirezana piramida v navadno piramido

Navadna piramida v kvader

$$T_{pers} = \begin{pmatrix} d/h & 0 & 0 & 0 \\ 0 & d/h & 0 & 1 \\ 0 & 0 & f/(f-d) & 1 \\ 0 & 0 & -fd/(f-d) & 0 \end{pmatrix}$$





Cevovod 3D upodabljanja

Lokalni koordinatni sistem

Definicija predmeta

Transformacije modela

Koordinatni prostor sveta

Kompozicija scene
Definicija pogleda
Definicija osvetlitve

Transformacija gledanja

Prostor gledanja

Projekcija

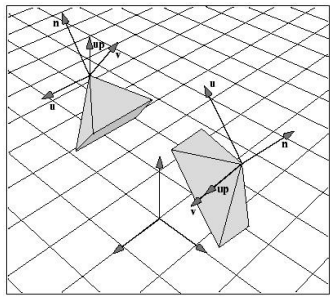
Odstranjevanje
Rezanje v 3D prostor gledanja

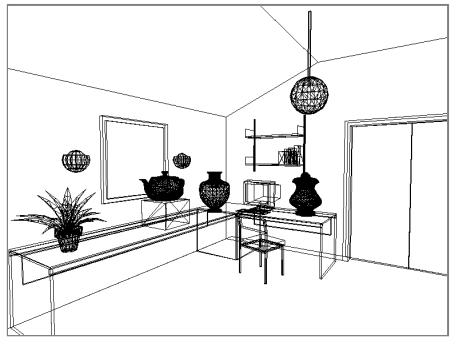
3D prostor zaslona

Deljenje projekcije

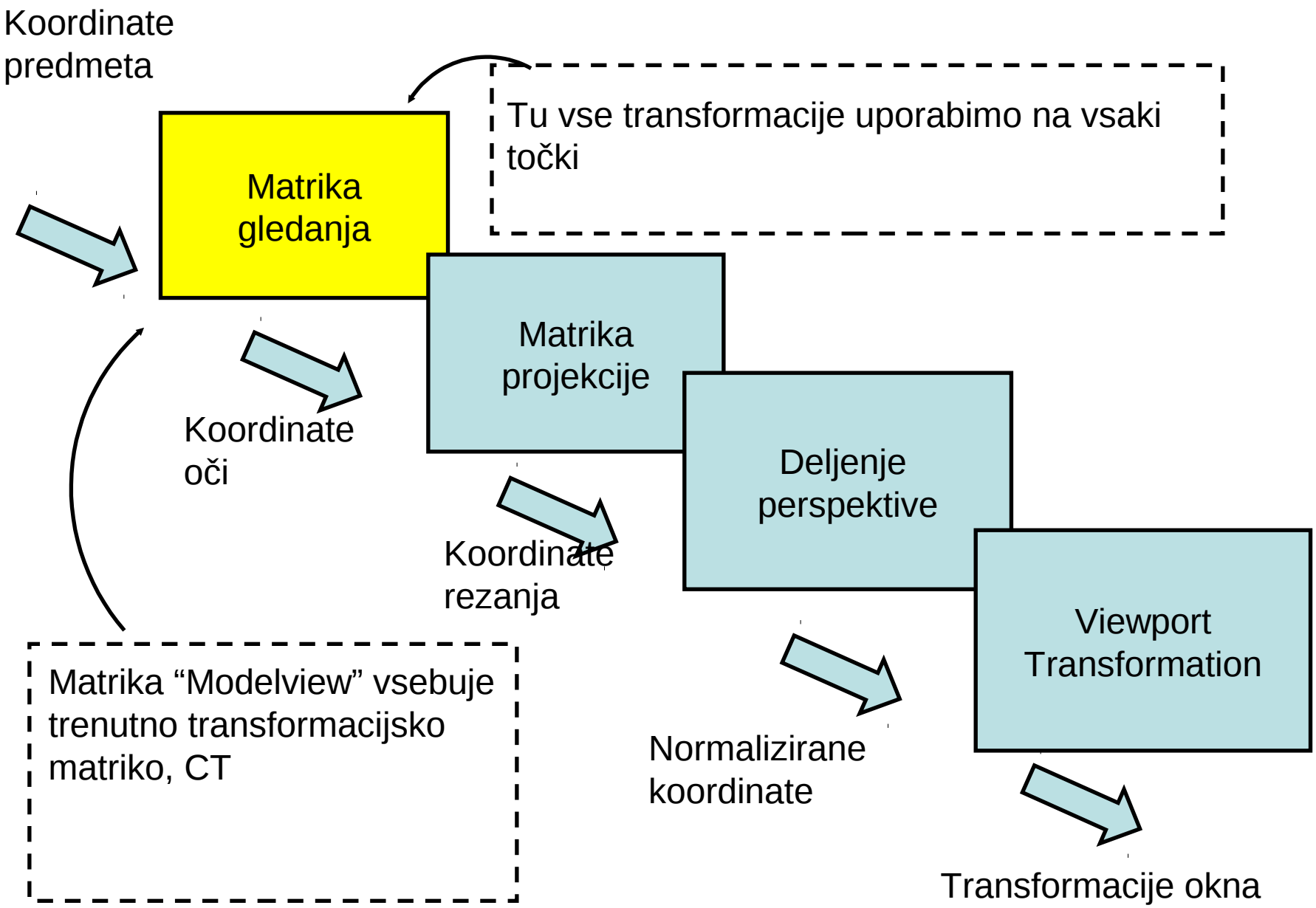
Odstranjevanje zakritih ploskev
Rasterizacija
Senčenje

Prostor zaslona

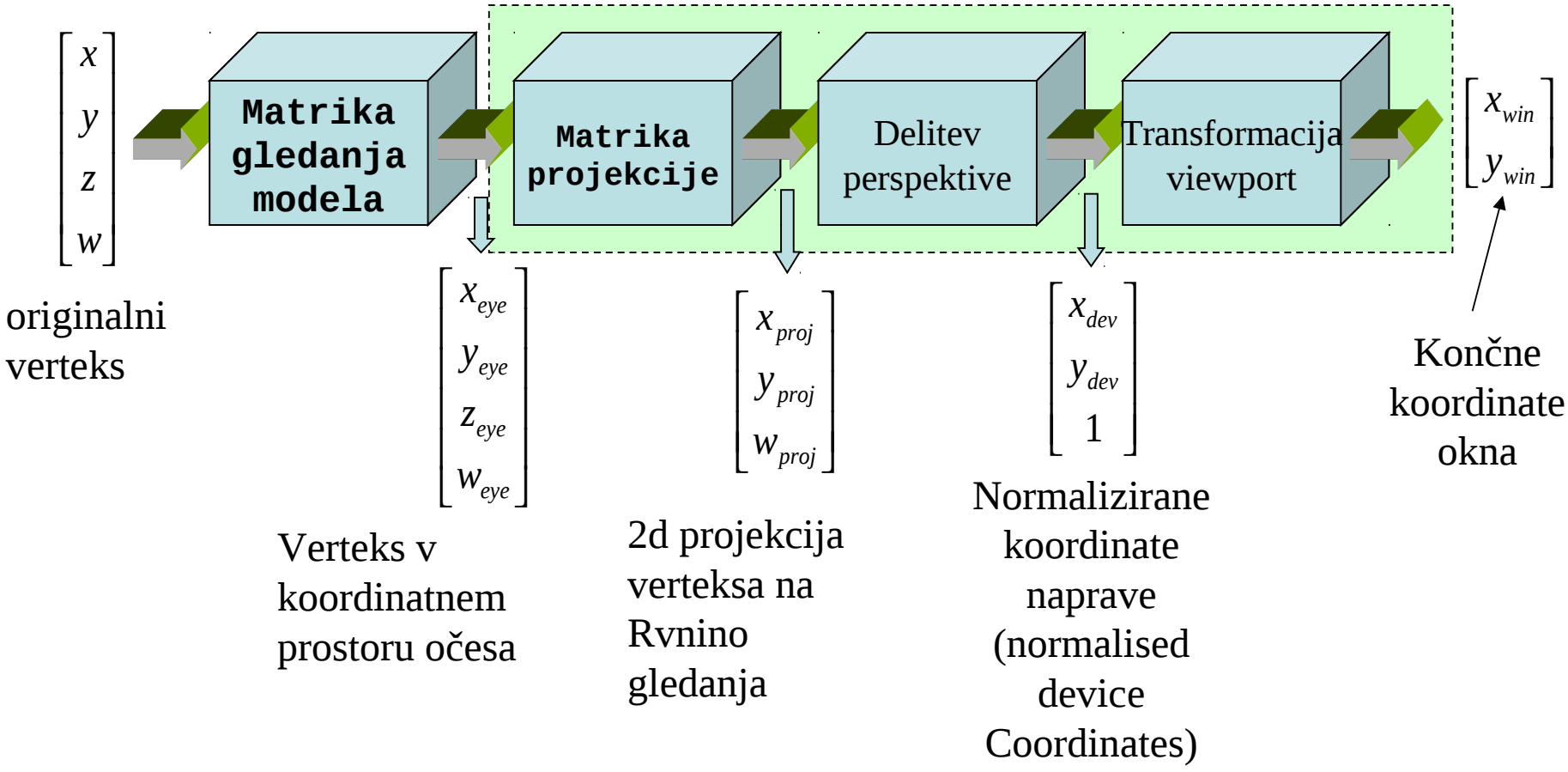




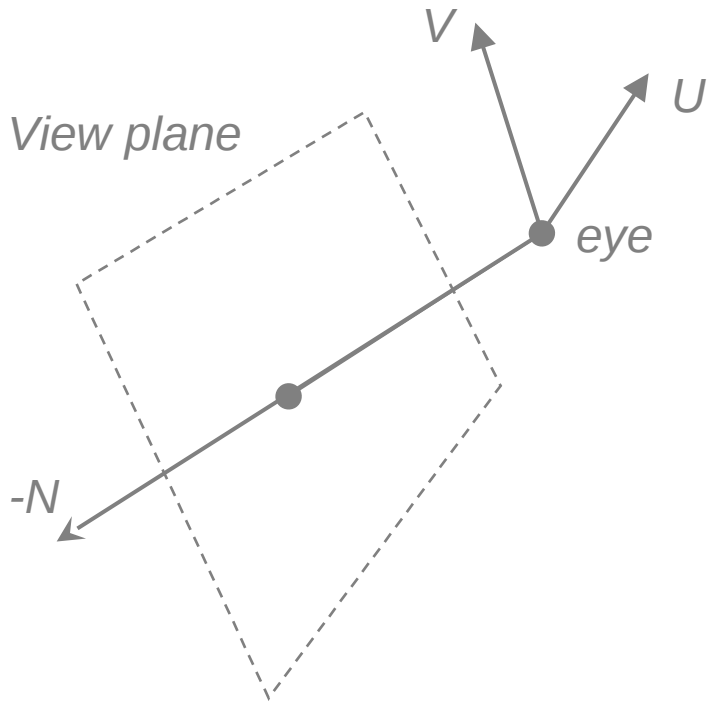
Stopnje transformacije verteksov

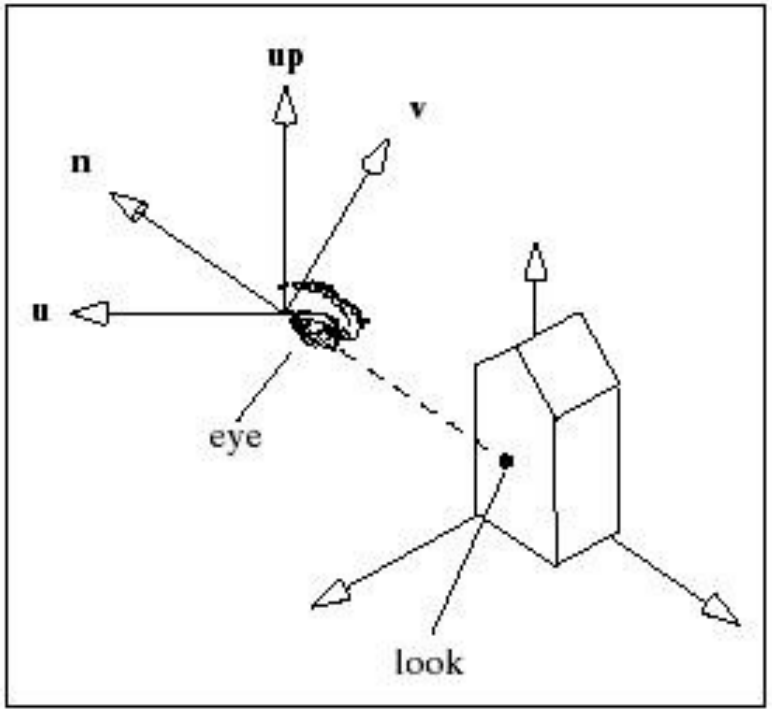


OpenGL® Cevovod preslikav



OpenGL: Položaj in usmeritev kamere

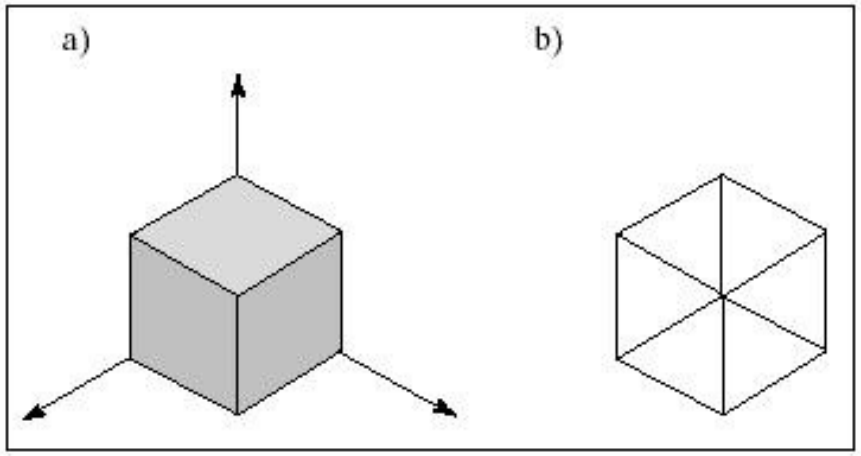





```
glMatrixMode (GL_MODELVIEW)
glLoadIdentity();

glLookAt (eye.x, eye.y, eye.z,  look.x, look.y, look.z,  up.x, up.y, up.z);
```

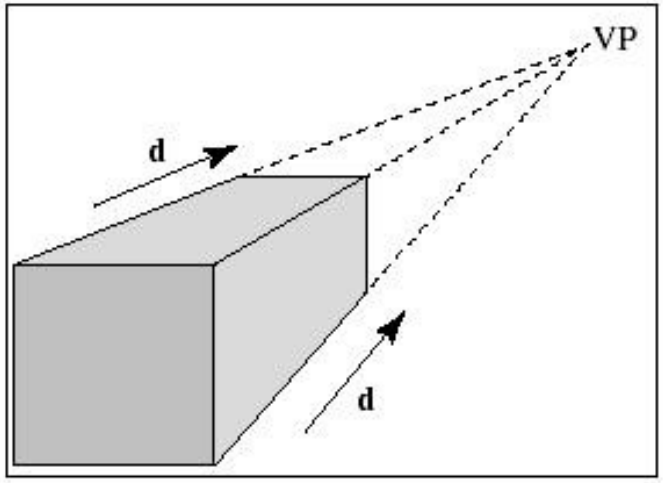
OpenGL: Postavitev kamere (Paralelna projekcija)

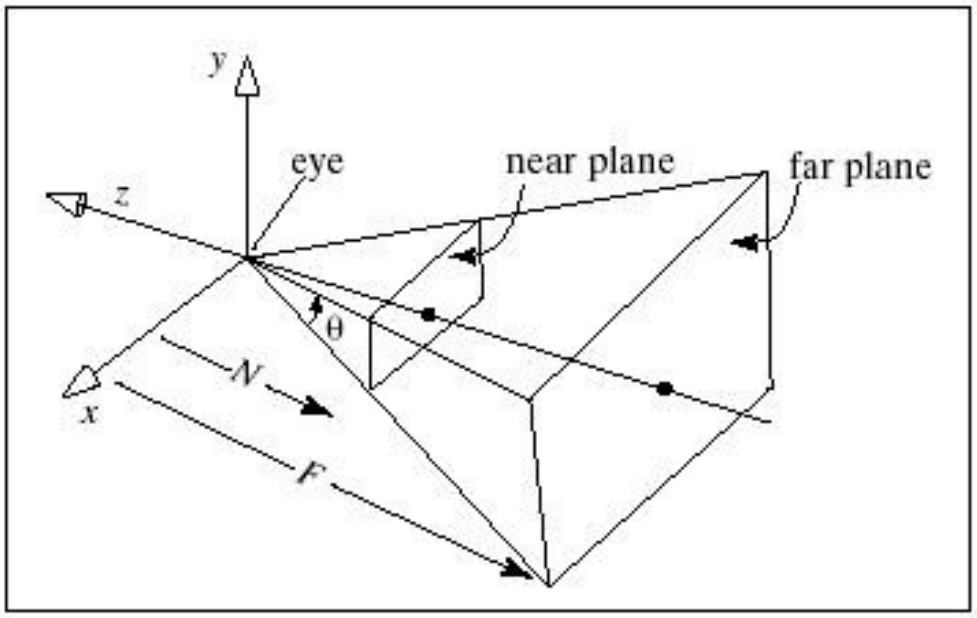



```
glMatrixMode (GL_PROJECTION)
glLoadIdentity();

glOrtho (left, right, bottom, top, near, far);
```

OpenGL: Postavitev kamere (Perspektivna projekcija)






```
glMatrixMode (GL_PROJECTION)
glLoadIdentity();
```

glFrustum (left, right, bottom, top, near, far);

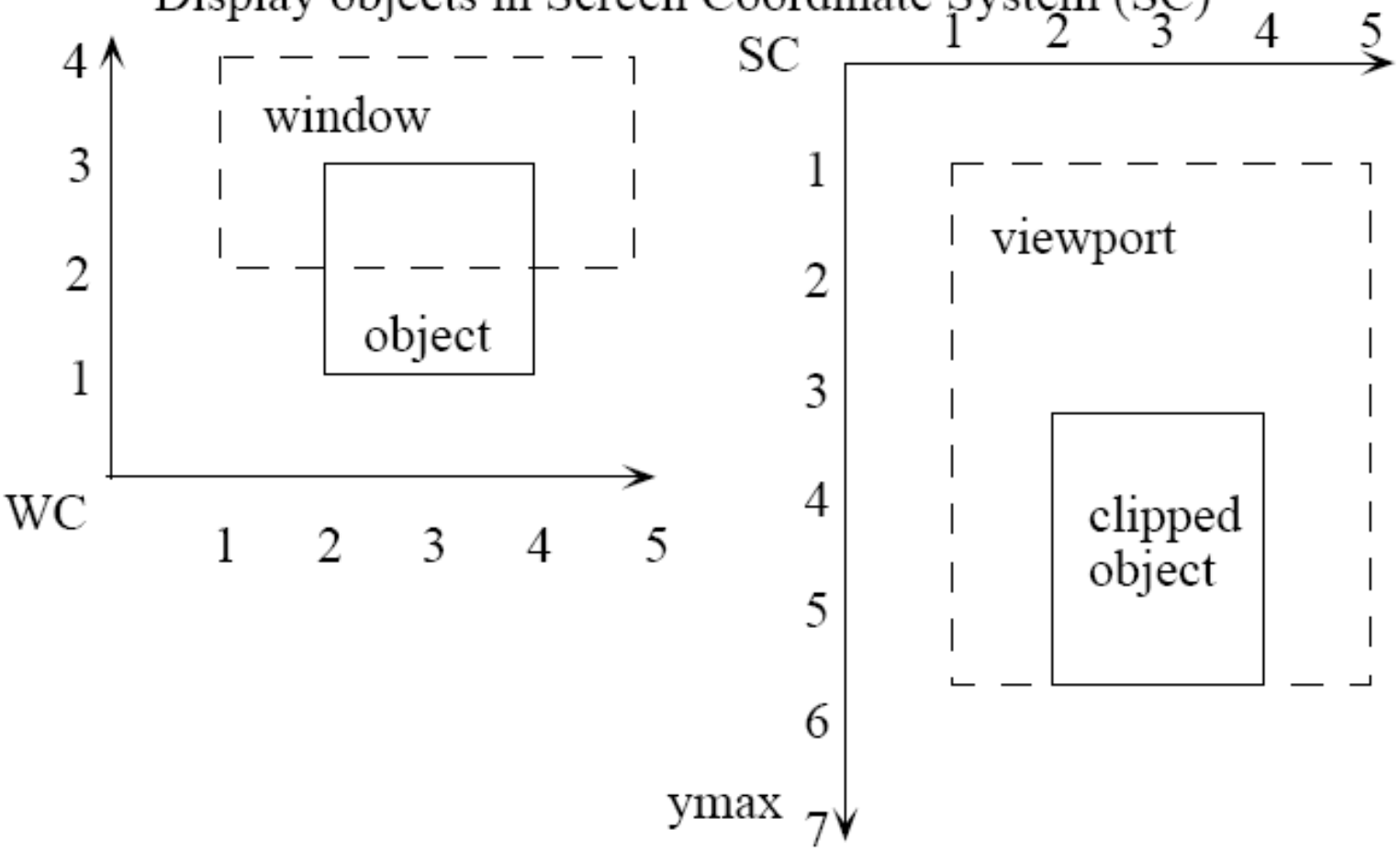
or

more intuitive → **gluPerspective (viewAngle, aspect, near, far);**

Transformacija window - viewport

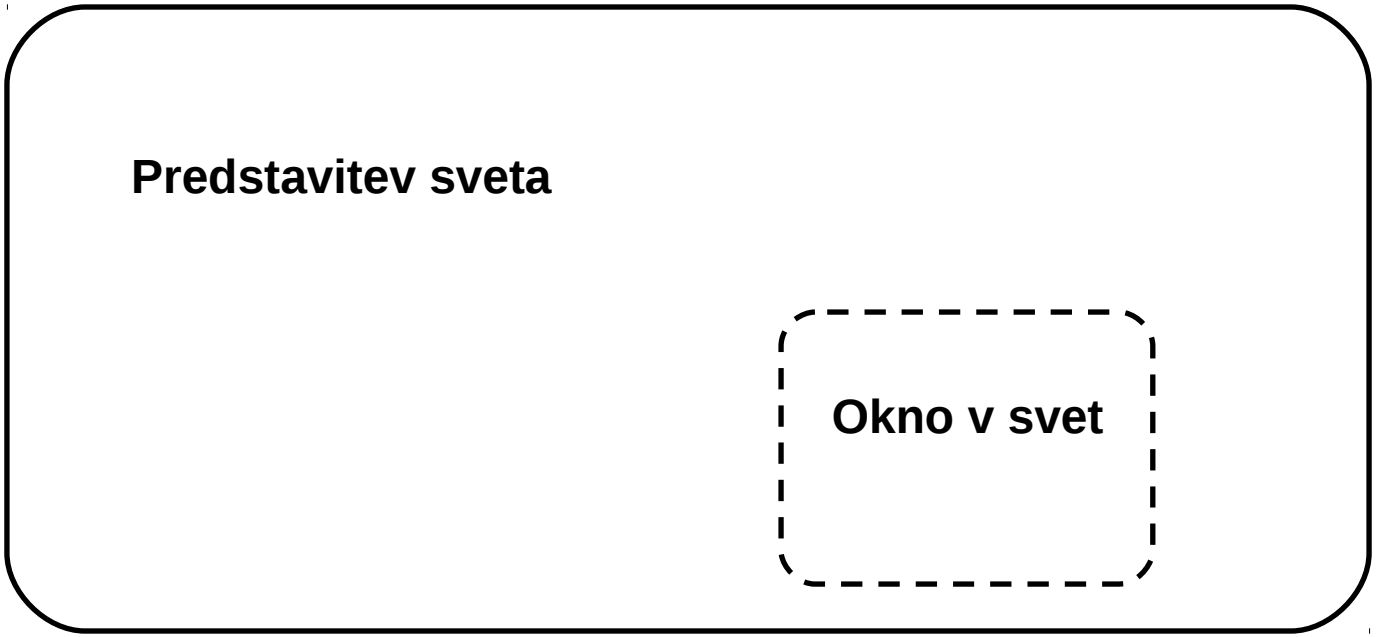
Define objects in World Coordinate System (WC)

Display objects in Screen Coordinate System (SC)



Svet in okno v svet

***Osnovna ideja*: Svet in okno v ta svet**



- **Predstavitev *sveta ostaja ista*.**
- **Okno v svet *se spreminja*, ko se premikamo v svetu, tj. okno v svet se premika.**

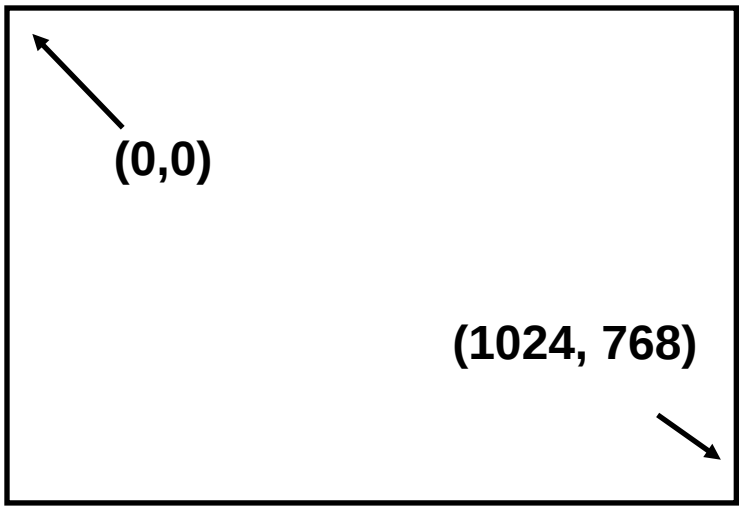
Koordinate

- **Tako “okno v svet” kot “svet” imata oba koordinatni sistem.**

- **Zaslon je množica *pikslov* (pixel = picture element). Kot tudi okno na namizju zaslona. Kot tudi območje znotraj okna.**

- **Piksli tvorijo dvodimenzionalno mrežo s koordinatami (0,0) v zgornjem levem in maksimalnim številom pikslov za dimenzijo v spodnjem desnem kotu, npr.: (1024,768).**

- **Tako so koordinate zaslona naslednje:**



Koordinate sveta

- **Črte so podane v koordinatah z izhodiščem, podanim v središču zaslona, npr.:**

  resolucija 1024 x 768 pomeni središčno točko (512, 384)

- **Vse transformacije:**

  - **izvedemo na koordinatah *sveta,***

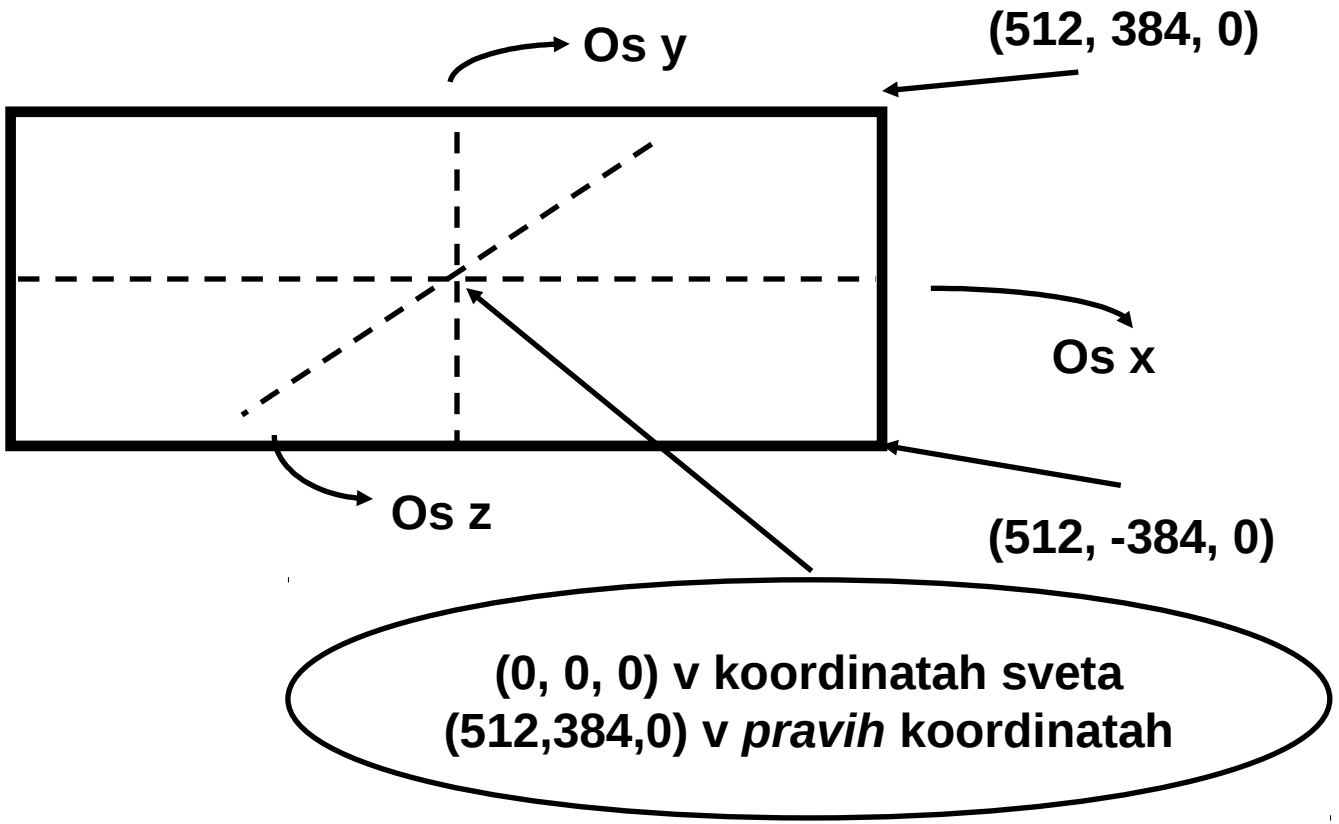
  - **ki jih preslikamo na koordinate zaslona.**

- **Tako je izračun logičnih transformacij ločen od podrobnosti vidnega področja.**

- **Model nočemo imeti vezan na resolucijo. Preslikava koordinat sveta na koordinate zaslona nam to omogoča.**

Koordinate sveta

Prikaz:



- os Z je pravokotna na osi X in Y.

Koordinate sveta

***Primer*:**

- **točka (0,0,0) v svetu je (512, 384) na zaslonu.**
- **točka (100, -20,0) v svetu je (612, 404) na zaslonu.**

- **zaslonska koordinata X postane večja s *pozitivno* koordinato sveta X.**
- **zaslonska koordinata Y postane večja z *negativno* koordinato sveta Y.**