

**Računarska grafika**  
(20ER7002)

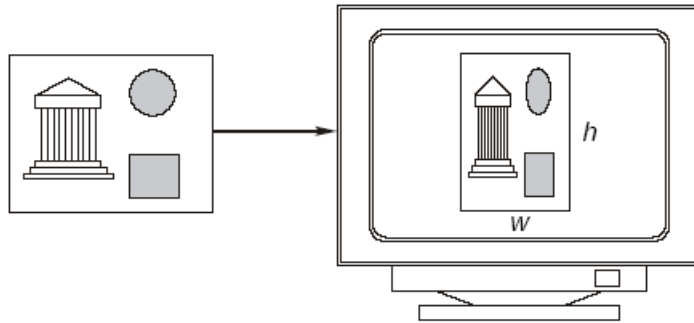
# **2D i 3D Pogled (*viewing*)**

**Predavanja**



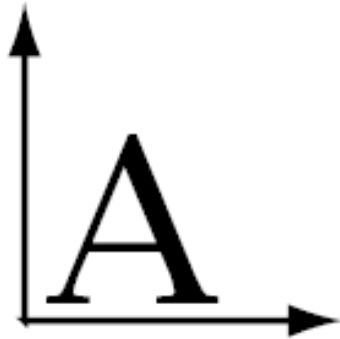
## 2D pogled

- 2D pogled predstavlja proces računarskog predstavljanja objekata iz 2D sveta na nekom od izlaznih grafičkih uređaja (šampač, ekran, ploter,...)



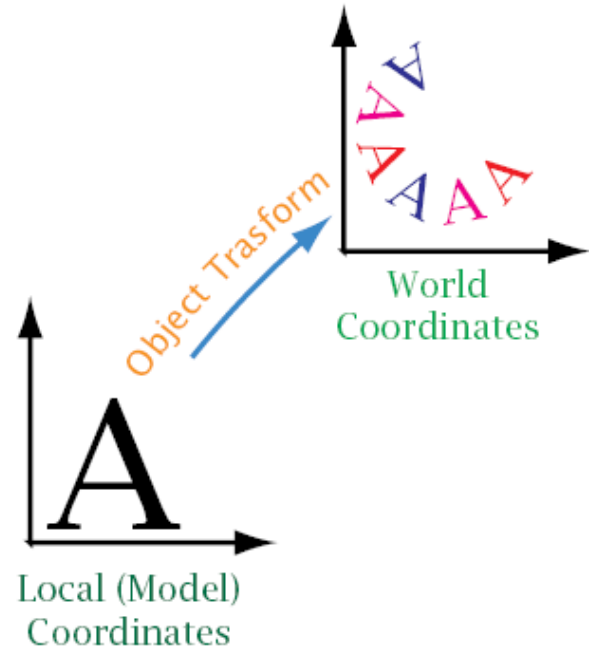
## Lokalne koordinate

- Lokalne koordinate su koordinate objekta (modela).



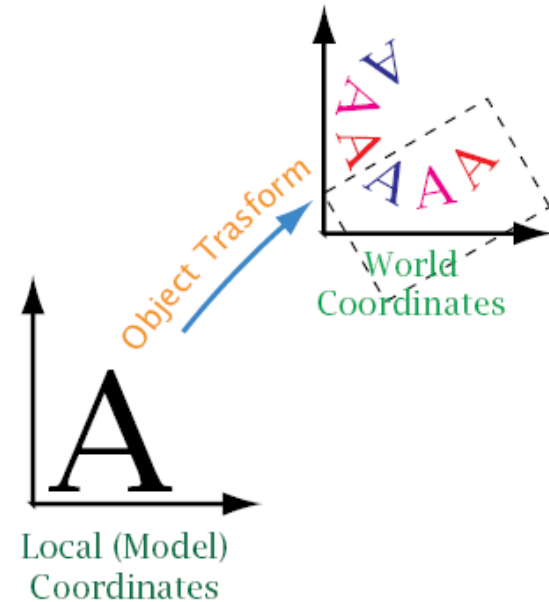
## Svetski koordinatni sistem

- Lokalne koordinate treba transformisati u **svetske** (prirodne) koordinate. Svetski (prirodni) koordinatni sistem opisuje relativnu poziciju i orijentaciju svih objekata na sceni.
- Za transformaciju iz lokalnog u svetski koordinatni sistem se koriste **elementarne 2D transformacije**.



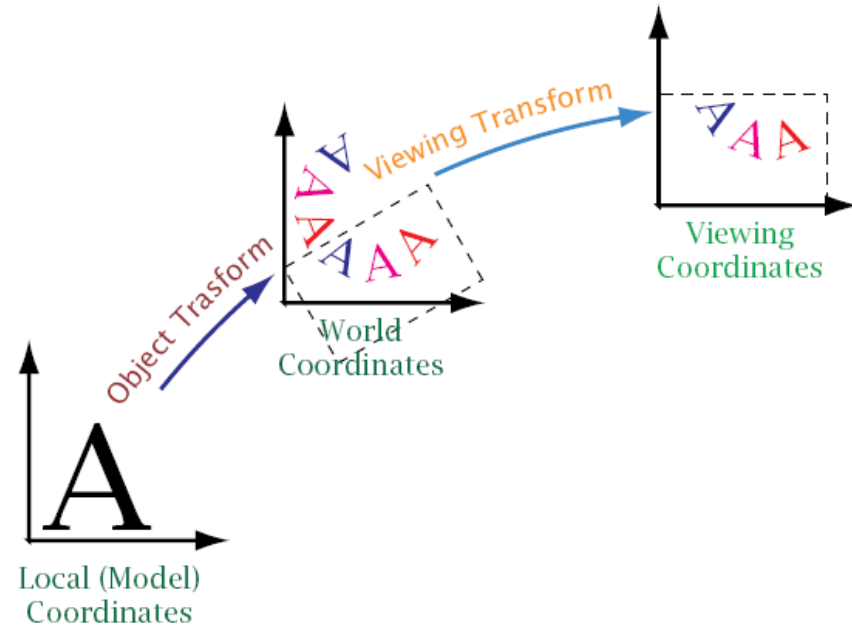
## Prozor u svetskom koordinatnom sistemu

- **Prozor** (*window*) u svetskom koordinatnom sistemu određuje pravougaonu oblast iz svetskog koordinatnog sistema koja će se prikazati na izlaznom uređaju.
- **Zaslona** (*viewport*) oblast u koordinatama uređaja (*device*) gde će se prikazati objekti.



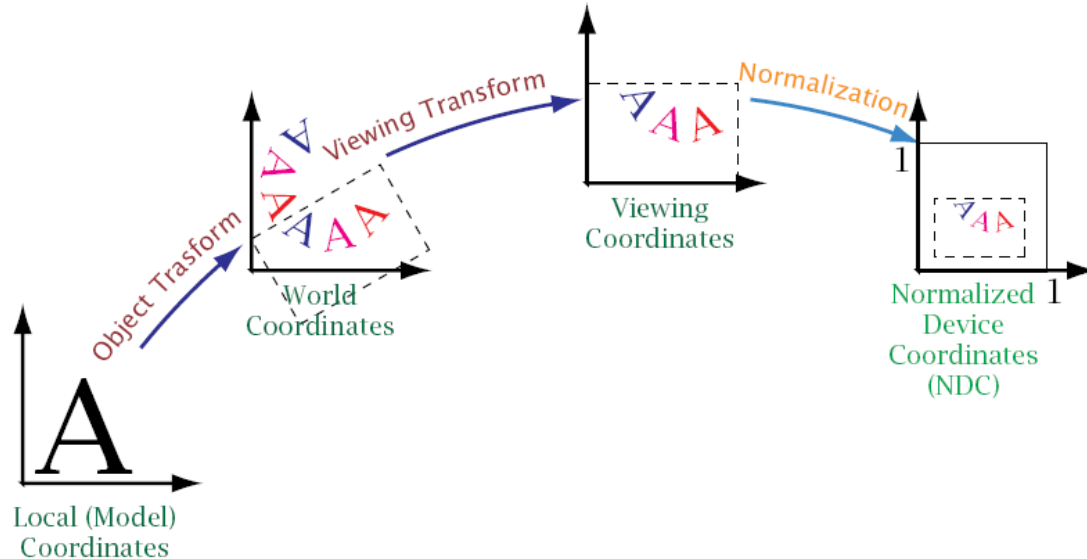
# Koordinate pogleda

- Koordinate **pogleda** opisuju poziciju i orijentaciju svakog objekta u odnosu na zadati **prozor**. Vrší se isecanje (*clipping*) objekata koji su van prozora.



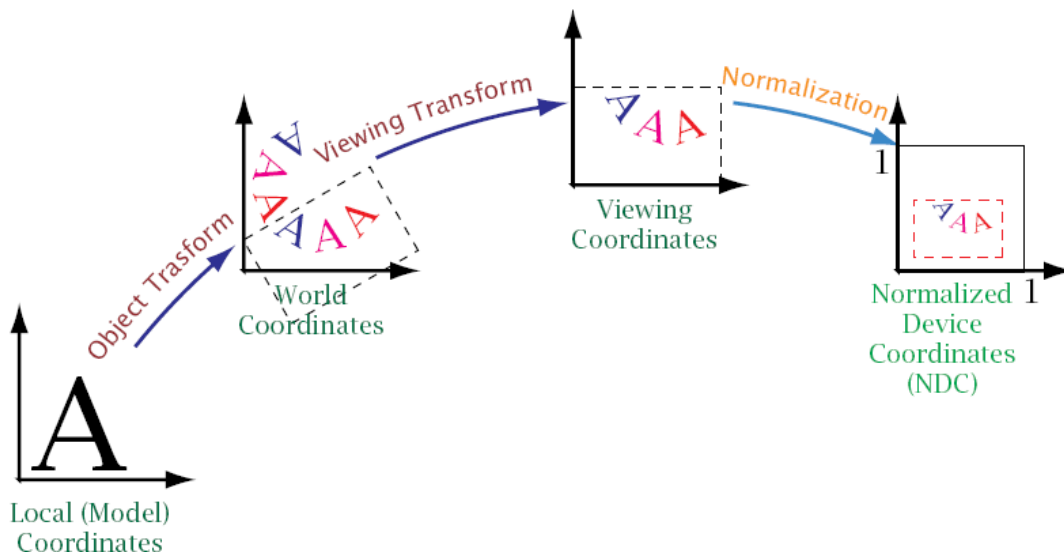
# Normalizovane koordinate uređaja

- Normalizovane koordinate opisuju virtuelni uređaj.
- Ovo se radi zbog kompatibilnosti sa različitim izlaznim uređajima.



## Definisanje zaslona

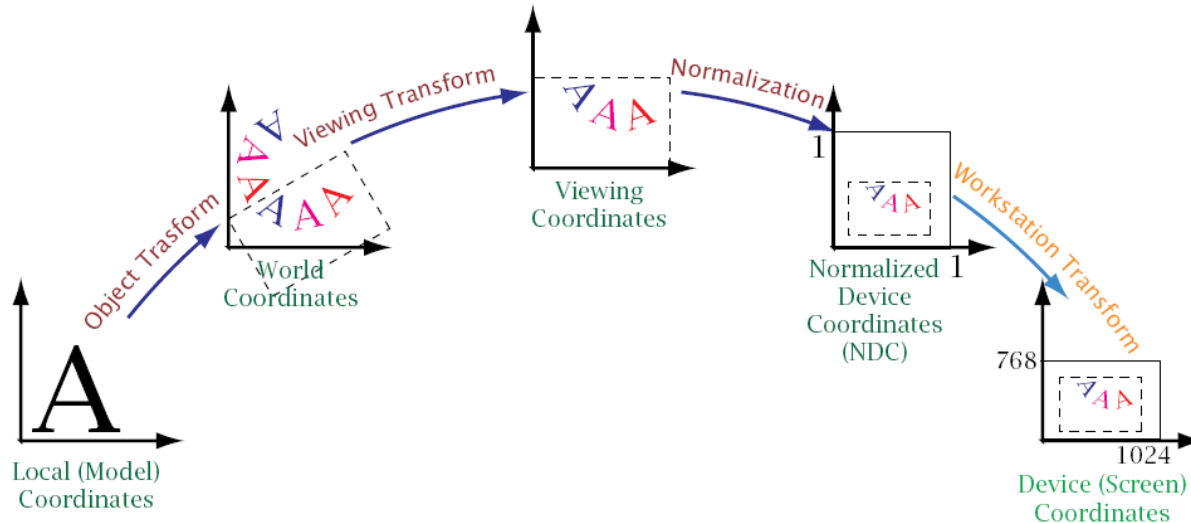
- Zaslon** (viewport) predstavlja pravougaonu oblast u normalizovanim koordinatama u koju se preslikava **prozor** iz svetskog koordinatnog sistema.



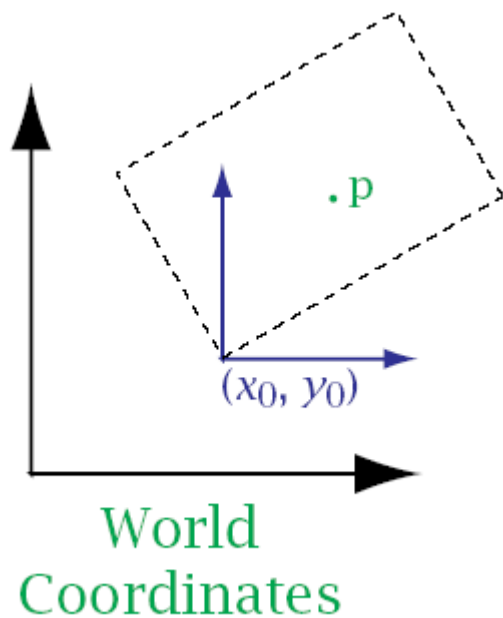


# Koordinate uređaja

- Koordinate uređaja** (DC – *device coordinates*) predstavljaju koordinate konkretnog izlaznog uređaja i zavise od **aspekta** i **rezolucije** uređaja.



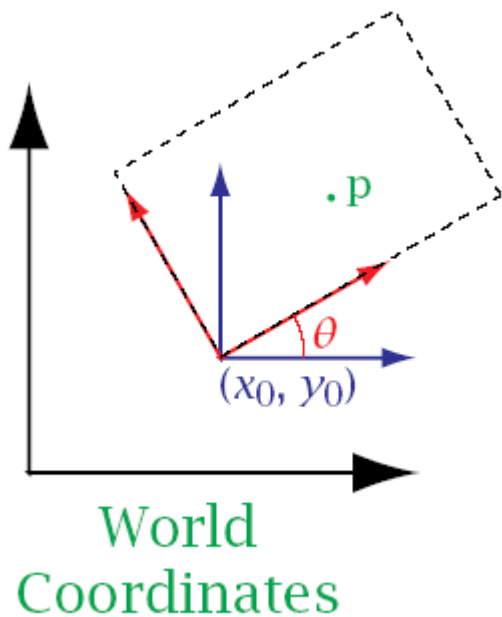
## Svetske koordinate → pogled



$(x_0, y_0)$  – pomeraj prozora  
u svetskim koordinatama

$$P' = P \cdot T(-x_0, -y_0)$$

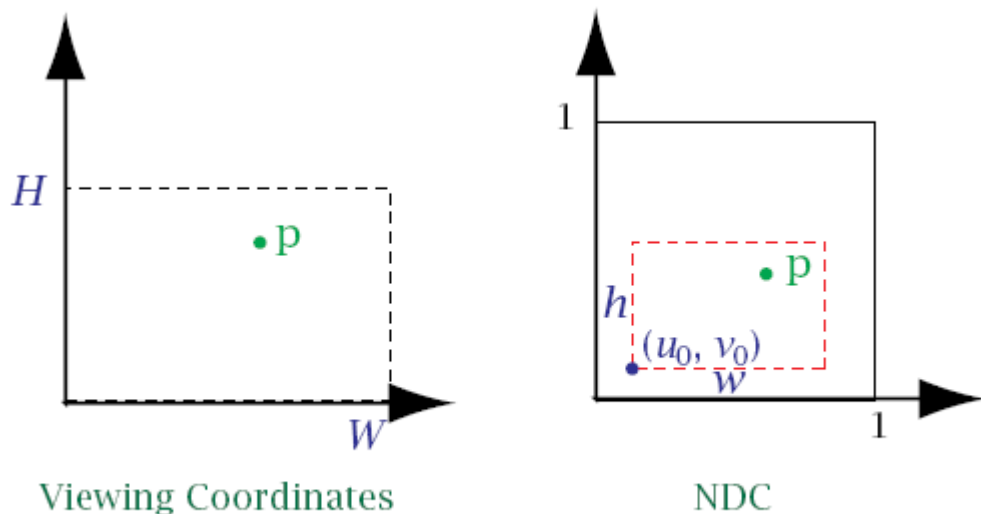
## Svetske koordinate → pogled



$\theta$  – ugao rotacije pogleda

$$P' = P \cdot T(-x_0, -y_0) \cdot R(-\theta)$$

## Koordinate pogleda → normalizovane koordinate



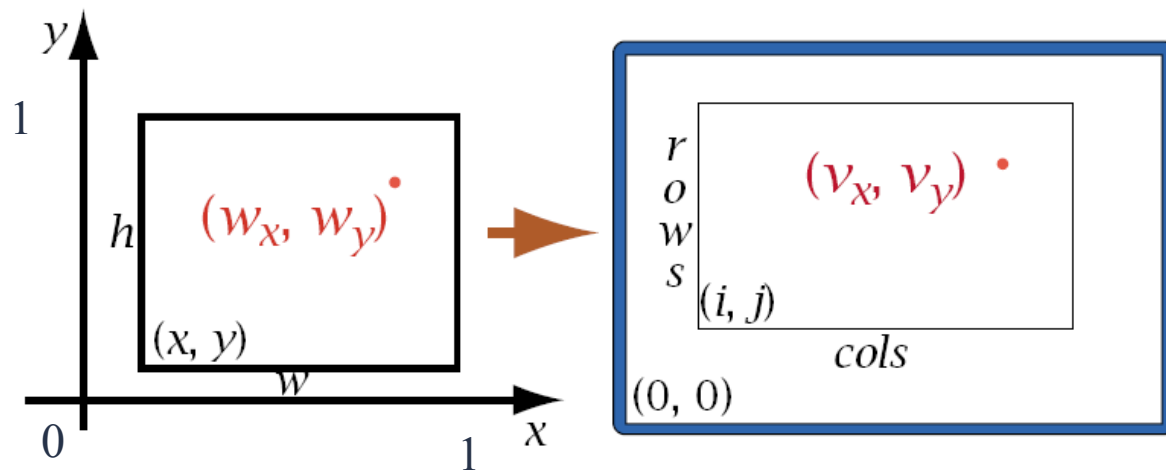
$W \times H$  – dimenzije prozora  
u svetskim koord.

$w \times h$  – dimenzije zaslona  
u NDC

$(u_0, v_0)$  – donji levi ugao  
zaslona u NDC

$$P' = P \cdot S(w/W, h/H) \cdot T(u_0, v_0)$$

## Normalizovane koordinate → koordinate uređaja



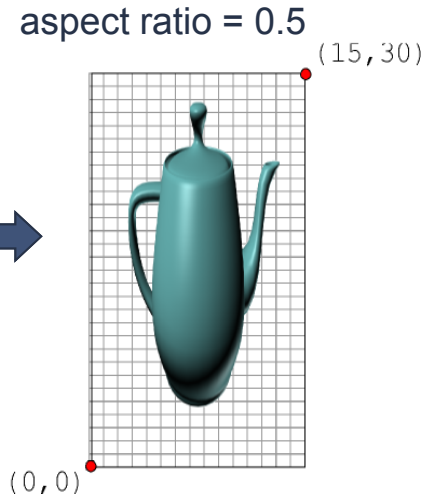
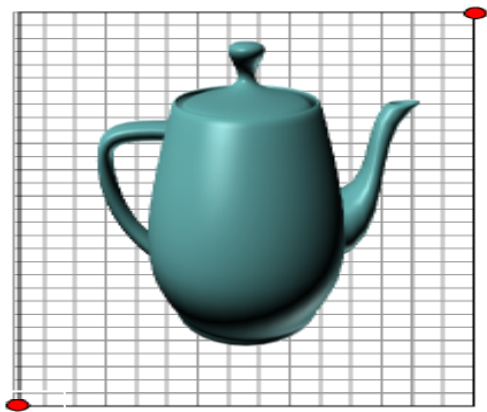
$$P' = P \cdot S(c, r)$$

$c$  – broj kolona (*cols*)  
 $r$  – broj vrsta (*rows*)

# Aspekt

- **Aspekt** definiše odnos širine i visine slike.

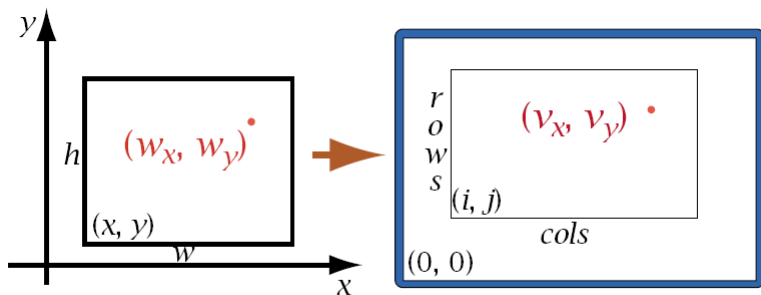
$$\text{aspect} = w/h$$



## Transformacija prozora u zaslon

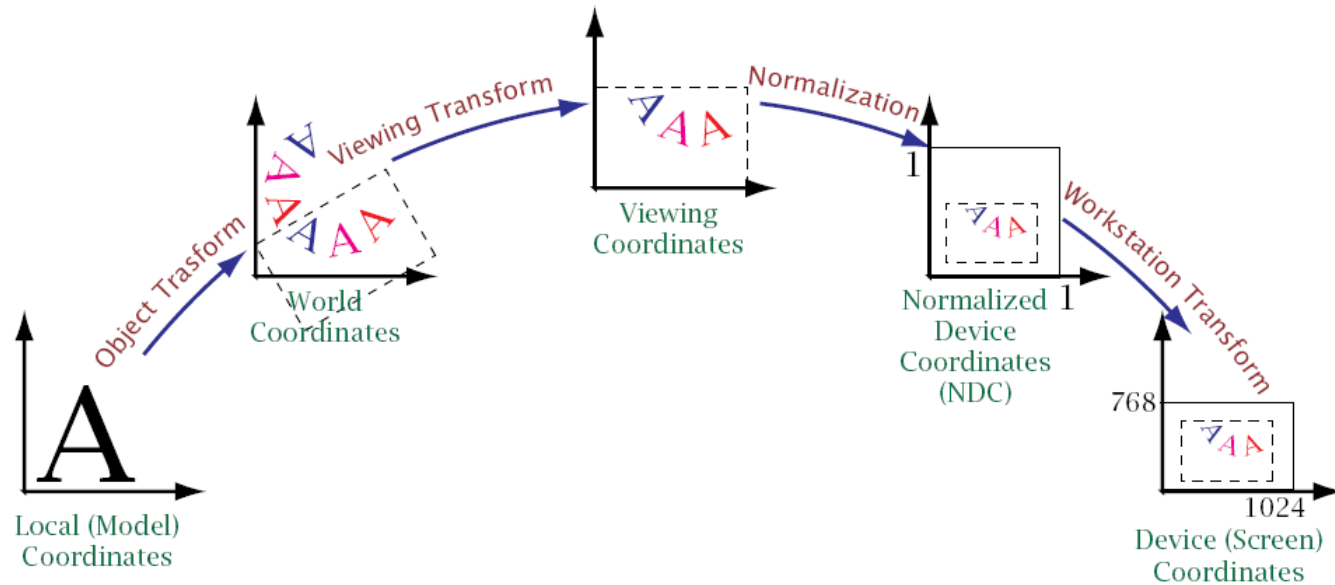
- Obično GAPI obezbeđuju funkciju za transformaciju prozora u zaslon (*Window-to-Viewport* transformation).

```
glViewport(i, j, rows, cols);  
gluOrtho2D(x, y, x + w, y + h);
```



$$P' = P \cdot T(-x_0, -y_0) \cdot S(w/W, h/H) \cdot T(u_0, v_0)$$

# Implementacija 2D pogleda

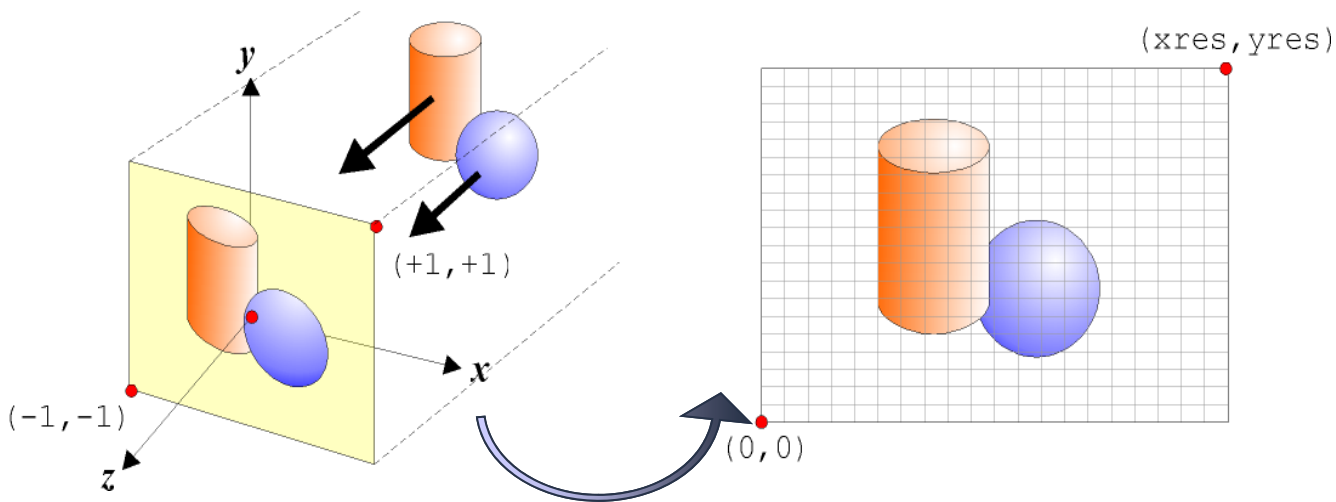


$$P' = P \cdot T(-x_0, -y_0) \cdot R(-\theta) \cdot S(w/W, h/H) \cdot T(u_0, v_0) \cdot S(c, r)$$



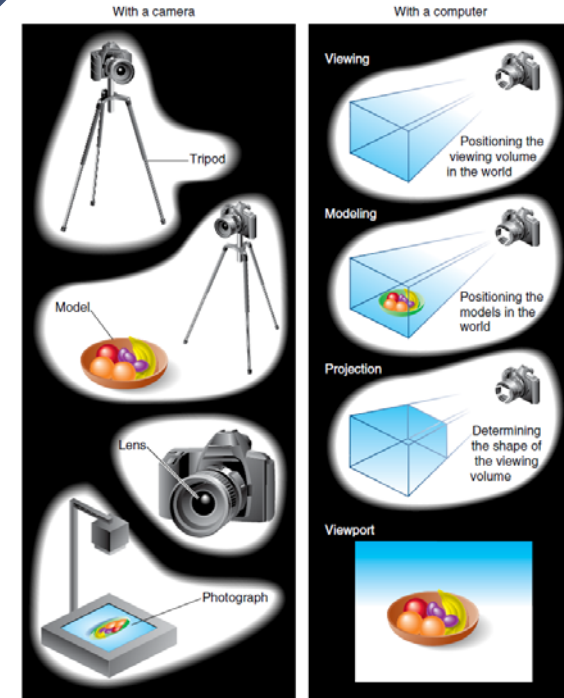
## 3D pogled

- **3D pogled** predstavlja proces računarskog predstavljanja objekata iz 3D sveta na nekom od izlaznih grafičkih uređaja (štampač, ekran, ploter,...)

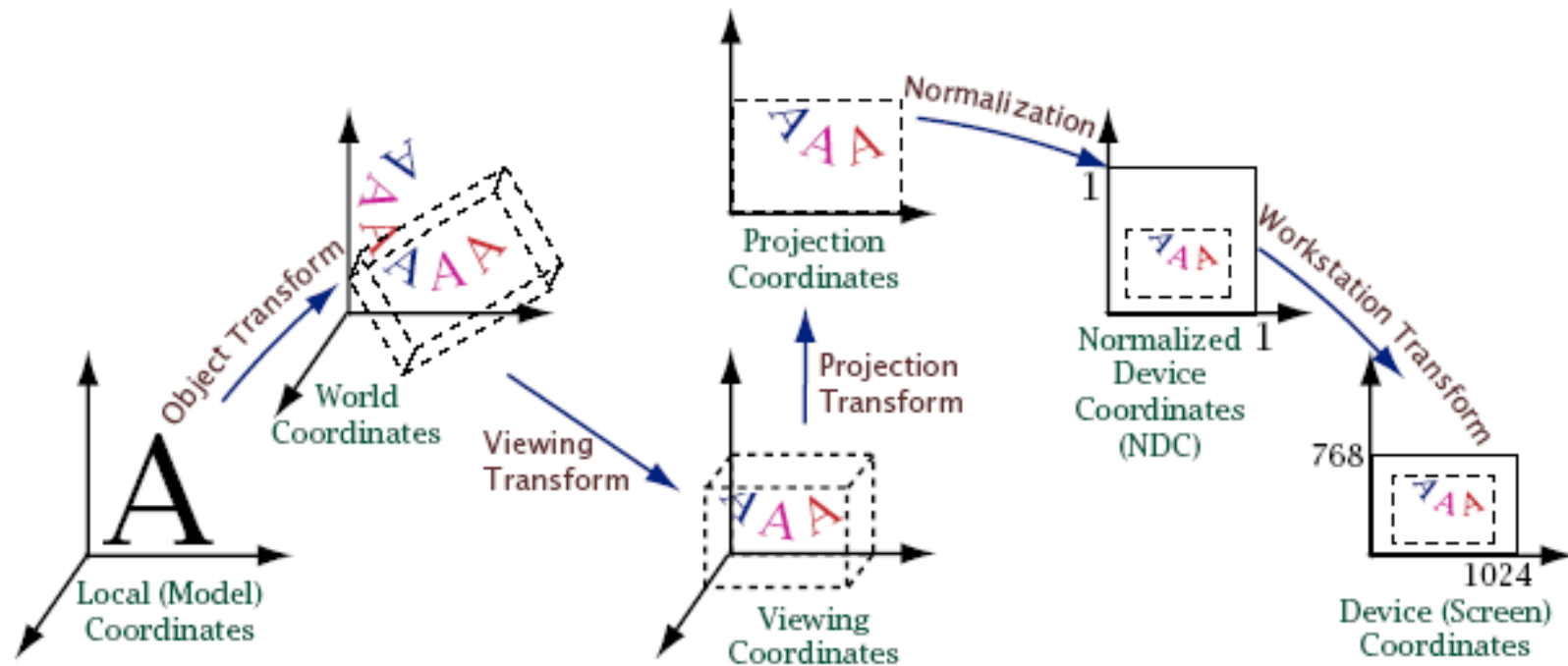


## 3D pogled – analogija sa fotografisanjem

- Osnovni elementi i kod klasičnog i kod kompjuterskog pristupa posmatranju objekata u principu su isti. Tako postoje objekti, posmatrač, projektori i ravan projekcije.
- Postoji analogija između računarskog posmatranja objekata i fotografisanja.



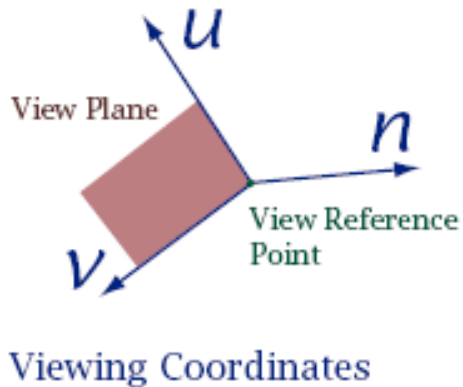
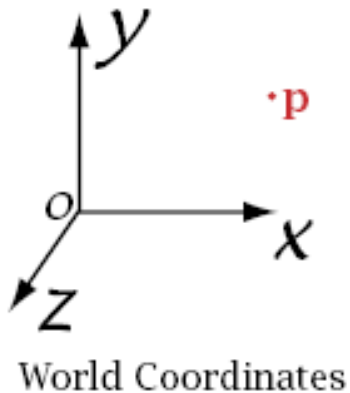
# Implementacija 3D pogleda



## Svetske koordinate → projektovane koordinate

1. Definisanje 3-D prozora, ili **volumena pogleda**.
2. Transformisanje u koordinate pogleda.
3. Transformisanje volumena pogleda u **kanonički volumen**.
3. Isecanje u odnosu na kanonički volumen.
4. Primena projekcije i dobijanje projektovanih koordinata.

## Svetske koordinate → koordinate pogleda



$$u = x$$

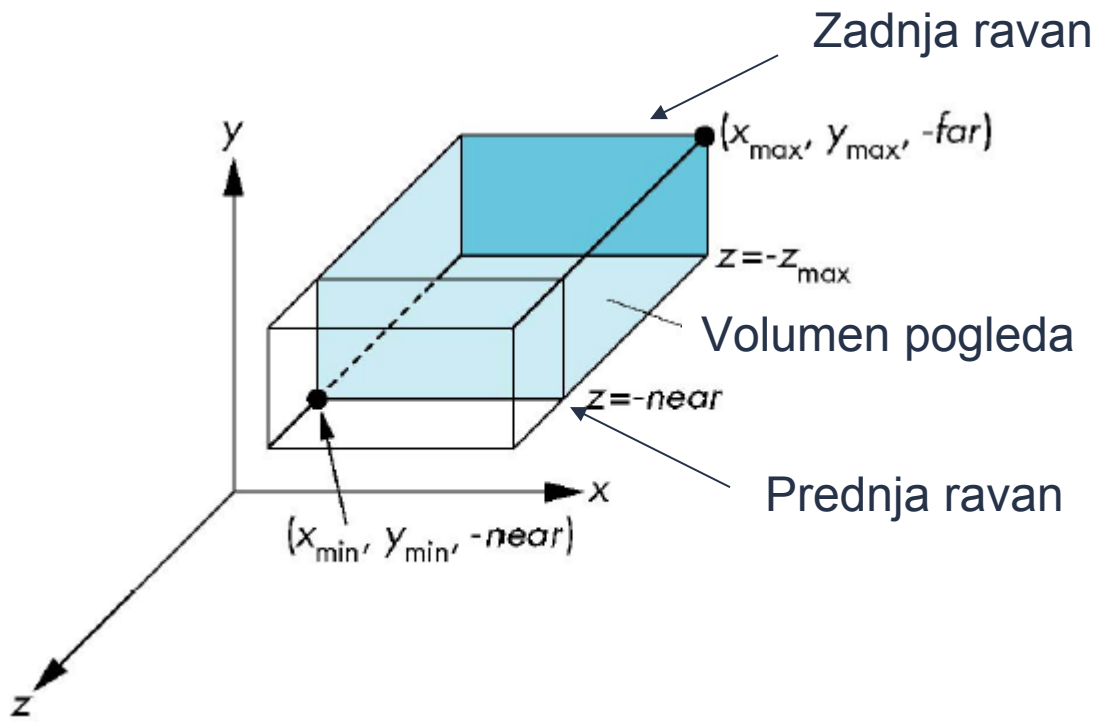
$$v = y$$

$$n = z$$

$$P' = P \cdot T(-x_0, -y_0, -z_0) \cdot R_y(\theta) \cdot R_x(\varphi) \cdot R_z(\alpha)$$

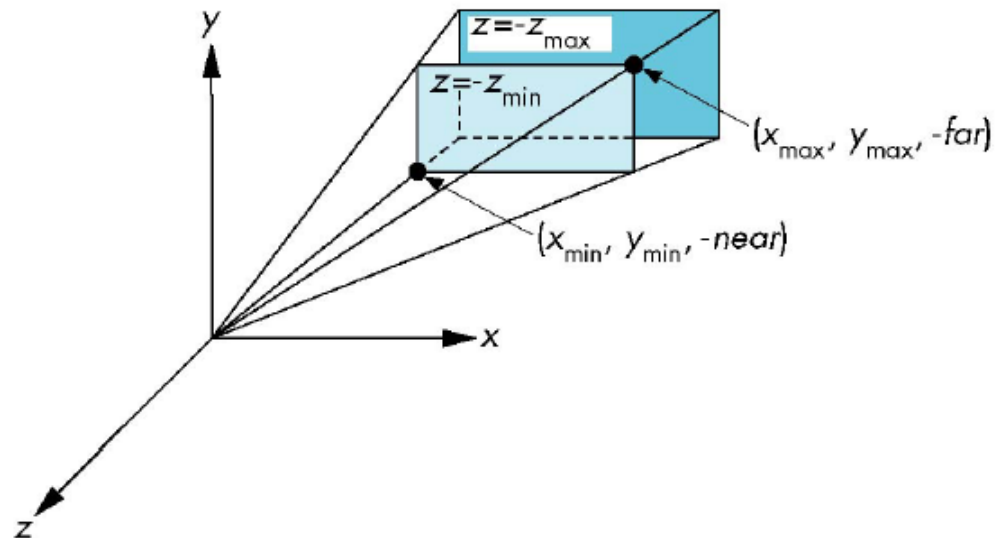
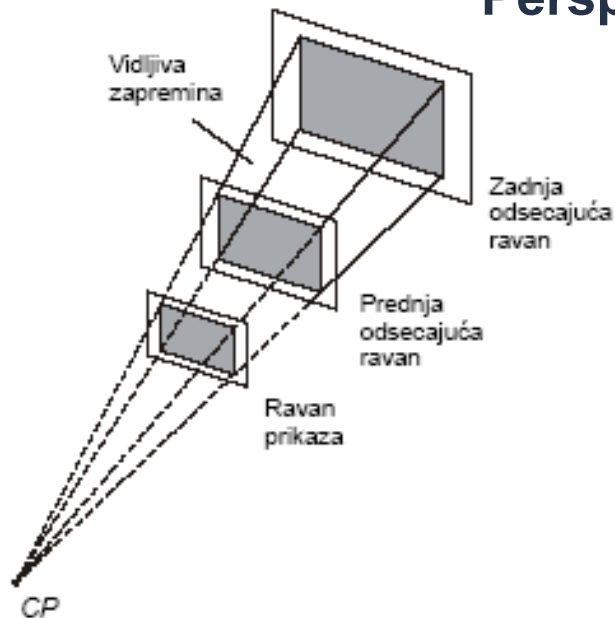
# Određivanje volumena pogleda

## Paralelna projekcija

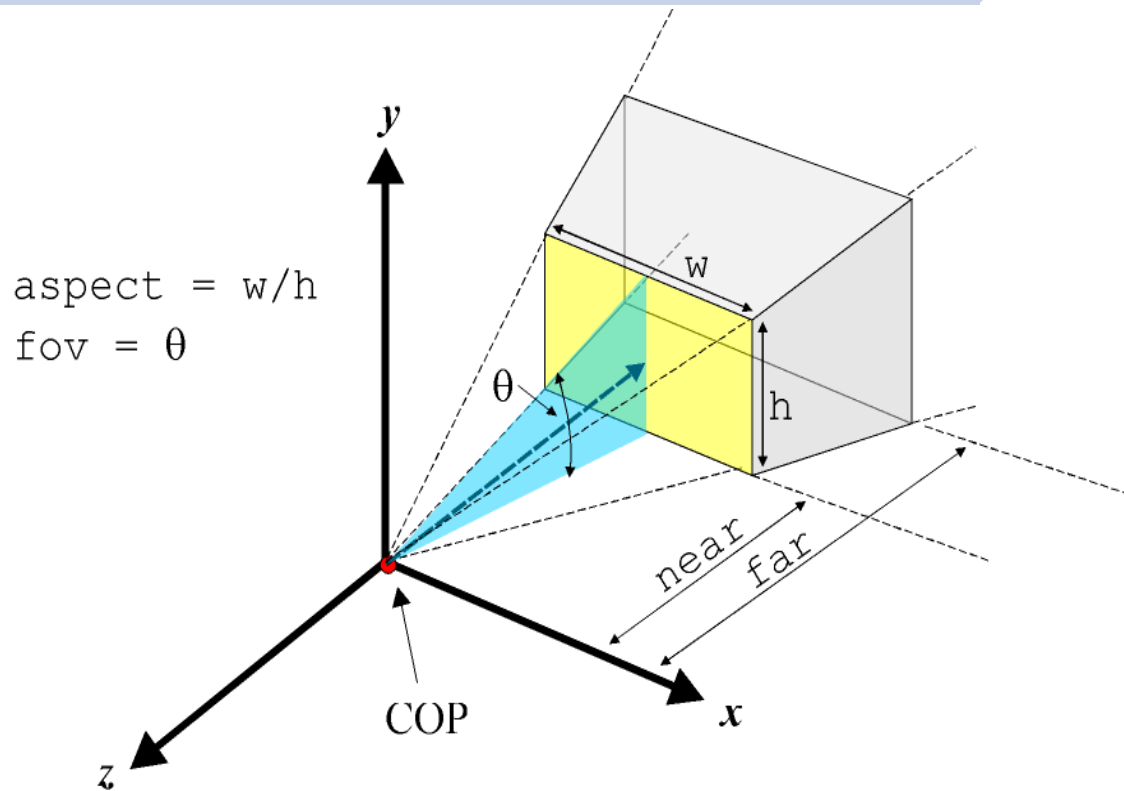


# Određivanje volumena pogleda

## Perspektivna projekcija



## Ugao pogleda (FOV - field of view)





# Standardna sočiva

- **48.24 mm** – ekvivalent ljudskog vida
- **širokougona sočiva** (obuhvataju veću scenu – umanjuju prikaz) <50mm
  - ▷ 35 mm i 28 mm – standardna širokougona sočiva
  - ▷ 10-15 mm – “riblje oko” (sferno sočivo), velika distorzija
- **telefoto sočiva** (uvećavaju prikaz, približavaju udaljene objekte) >50mm

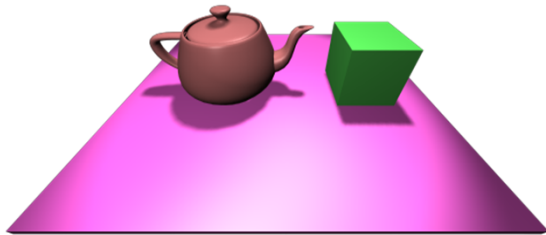
Koristiti FOV u opsegu:

**30-60°**,

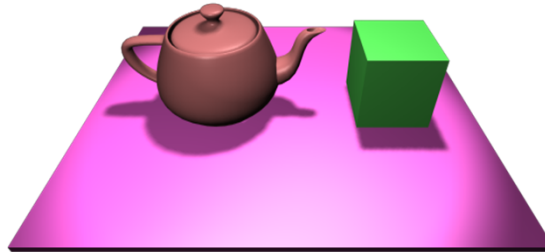
za realne scene, da ne bi došlo do distorzije.

Sočivo	FOV
15 mm	100.389
20 mm	83.974
24 mm	73.74
28 mm	65.47
35 mm	54.432
50 mm	39.598
85 mm	23.913
135 mm	15.189
200 mm	10.286

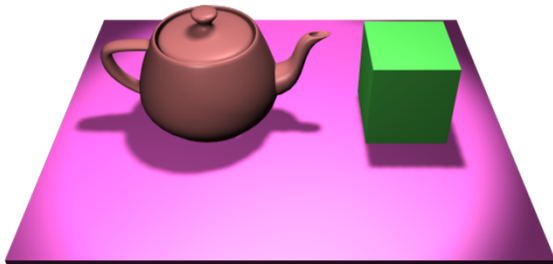
## Primer korišćenja različitih sočiva



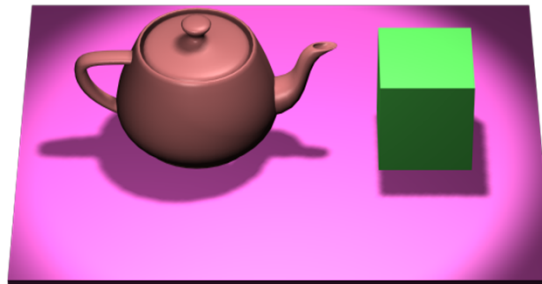
10mm Lens (fov = 122°)



20mm Lens (fov = 84°)



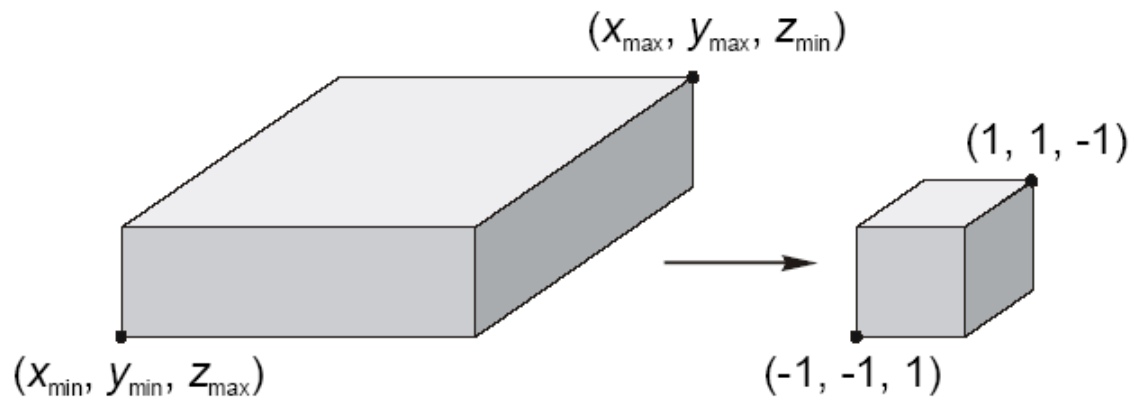
35mm Lens (fov = 54°)



200mm Lens (fov = 10°)

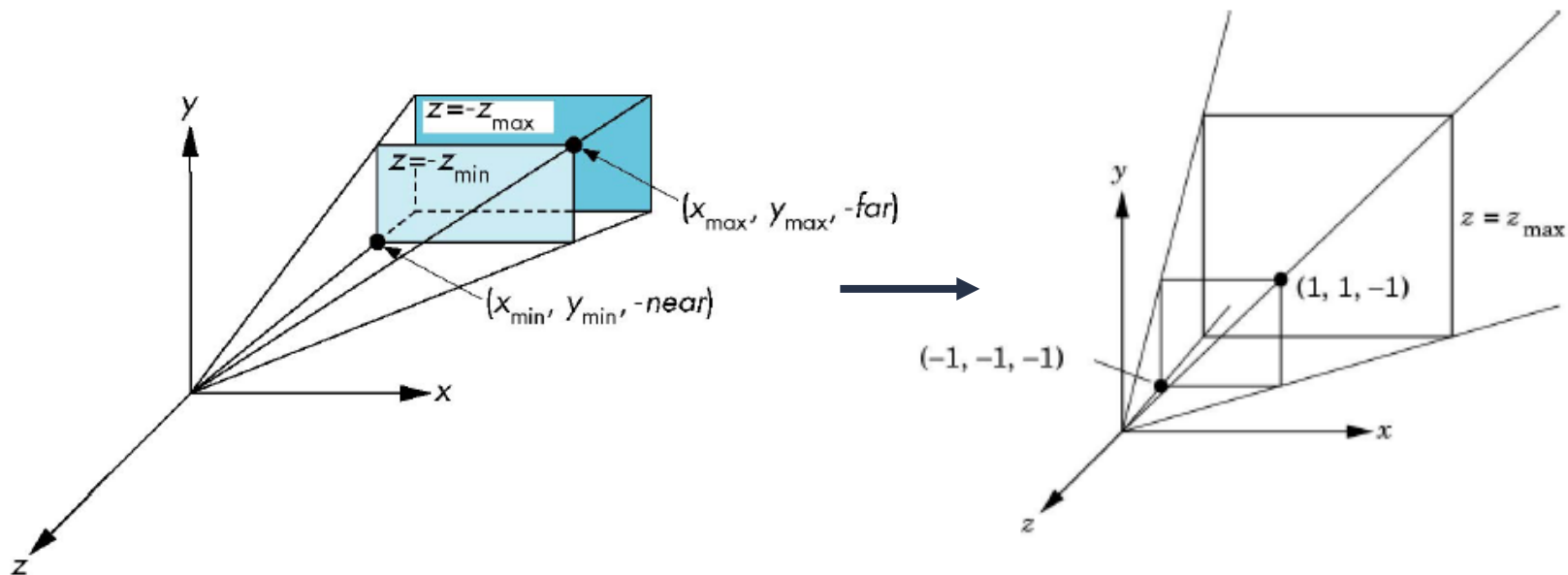
# Kanonički volumen

## Paralelna projekcija

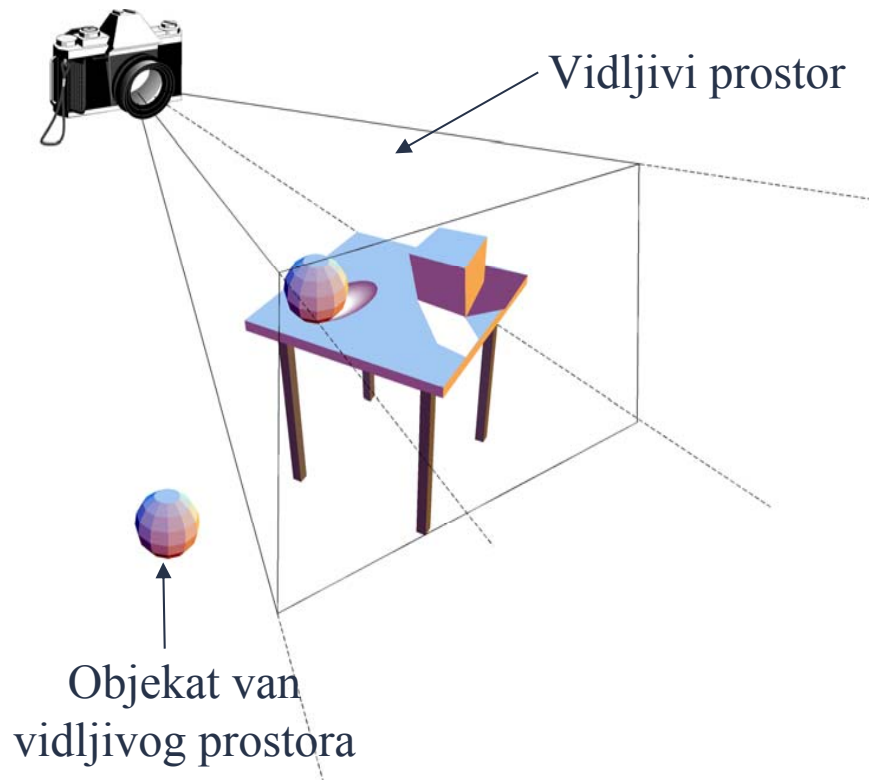


# Kanonički volumen

## Perspektivna projekcija



# Isecanje (clipping)



## Proširenje Cohen-Sutherland-ovog algoritma za isecanje u kanoničkom volumenu

- Svakoj oblasti se pridružuje 6-bitni položajni kod (*outcode*):  $b_5b_4b_3b_2b_1b_0$ , gde svaki bit označava jednu oblast:

$b_5$  – iznad volumena,  $y > 1$

$b_4$  – ispod volumena,  $y < -1$

$b_3$  – desno od volumena,  $x > 1$

$b_2$  – levo od volumena,  $x < -1$

$b_1$  – ispred volumena,  $z > 1$

$b_0$  – iza volumena,  $z < -1$

**Paralelna projekcija**

## Proširenje Cohen-Sutherland-ovog algoritma za isecanje u kanoničkom volumenu

- Svakoj oblasti se pridružuje 6-bitni položajni kod (*outcode*):  $b_5b_4b_3b_2b_1b_0$ , gde svaki bit označava jednu oblast:

$b_5$  – iznad volumena,  $y > z$

$b_4$  – ispod volumena,  $y < -z$

$b_3$  – desno od volumena,  $x > z$

$b_2$  – levo od volumena,  $x < -z$

$b_1$  – ispred volumena,  $z > -1$

$b_0$  – iza volumena,  $z < -z_{\max}$

**Perspektivna projekcija**

## Proširenje Cohen-Sutherland-ovog algoritma za isecanje u kanoničkom volumenu

- Linija se trivijalno prihvata ako obe krajnje tačke imaju kodove 0.
- Linija se trivijalno odbacuje ako obe krajnje tačke posle AND operacije imaju kod  $\neq 0$ .
- U ostalim slučajevima se traži presek linije i volumena.



# PITANJA

