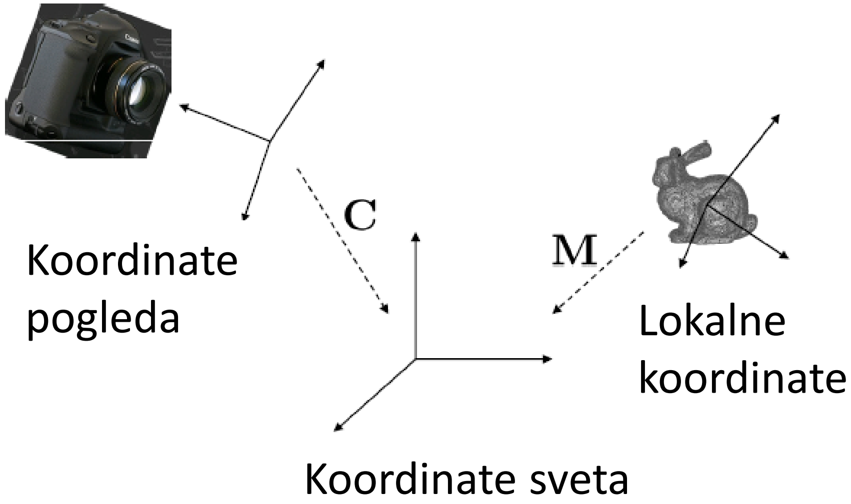
PROJEKCIJE

Projekcije

- Predmet preslikamo
  - lokalne koordinate
  - koordinate sveta
  - koordinate pogleda/kamere

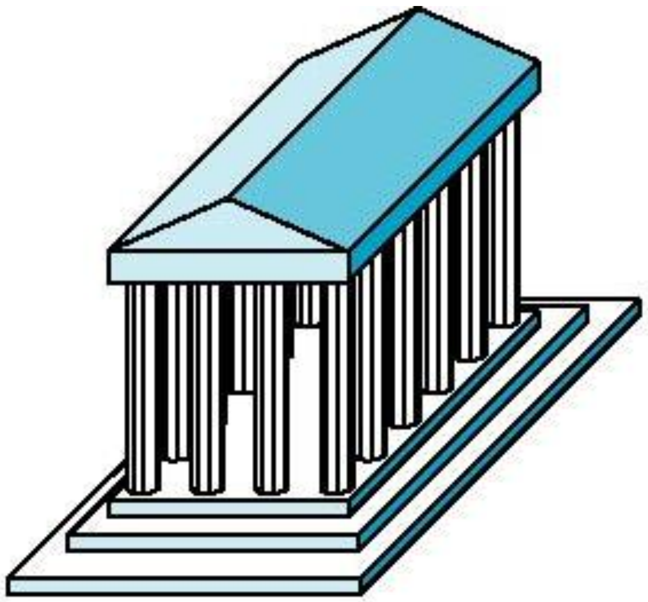
- Še vedno je **3D**

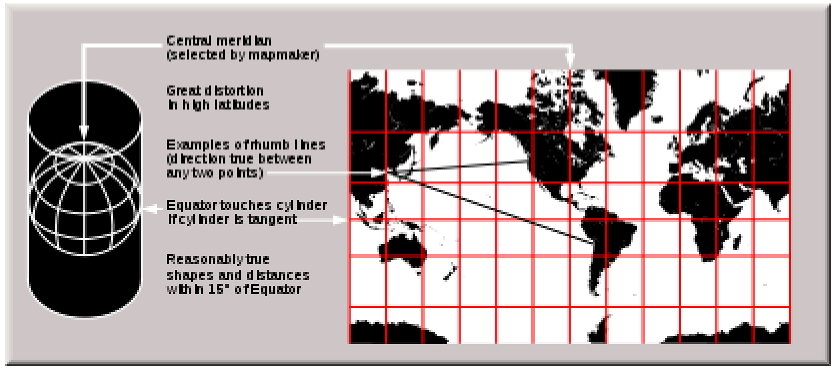
- **Planarna** projekcija
  - preslikava iz 3D predstavitve predmeta na projekcijsko ravnino
  - torej iz 3D v 2D



Planarna projekcija

- Projekcija na **ravnino**
- **Planarne** projekcije **ohranijo črte**
  - ne pa nujno tudi kotov
- **Neplanarne** projekcije se uporabljajo npr. pri zemljevidih (cilindrična itn.)

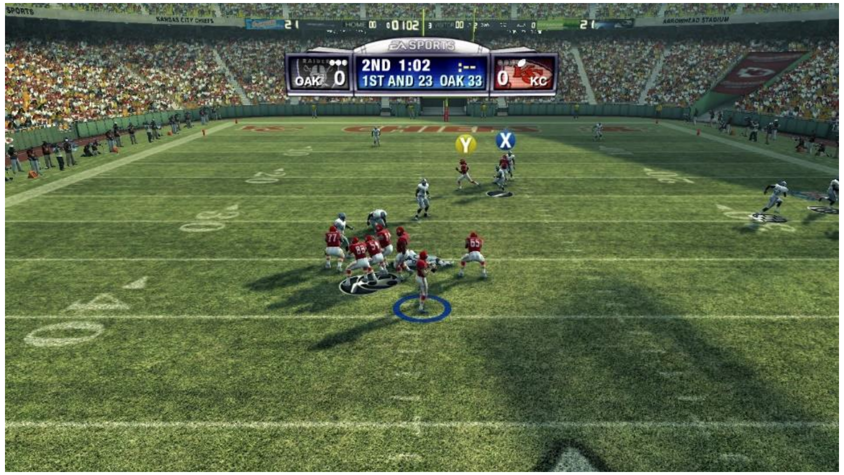




Osnovna delitev planarnih projekcij

- V RG vse planarne projekcije matematično gledano obravnavamo enako
- Delimo jih na:
  - **vzporedne**: projekcijski žarki so vzporedni
  - **perspektivne**: projekcijski žarki konvergirajo v točko

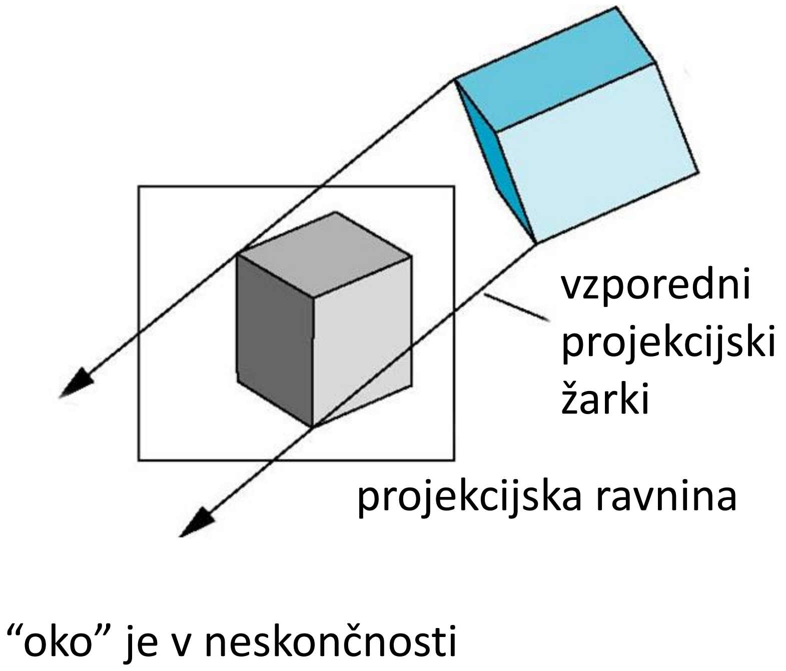
Age of Empires II
©Microsoft Corporation



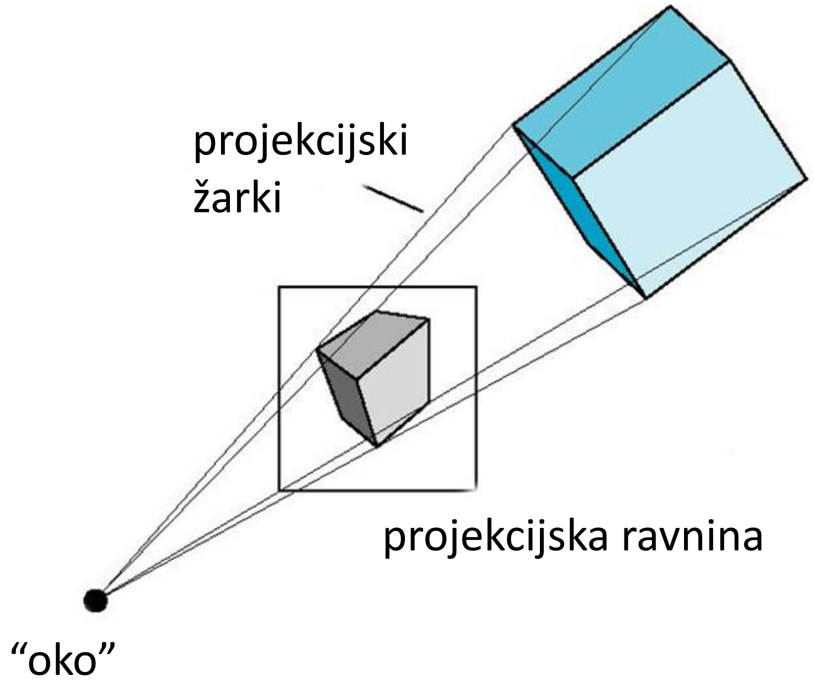
Maden NFL 2009

Primerjava projekcij

Vzporedna projekcija



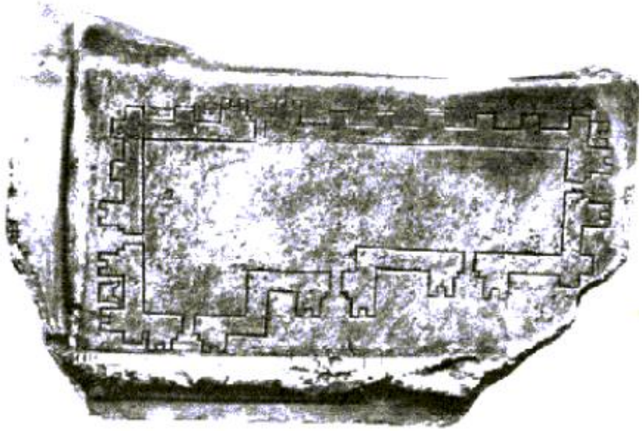
Perspektivna projekcija

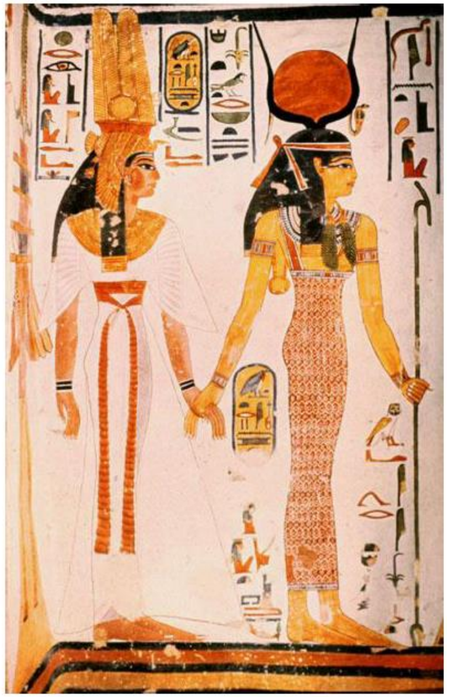


Vzporedna projekcija

Malo zgodovine

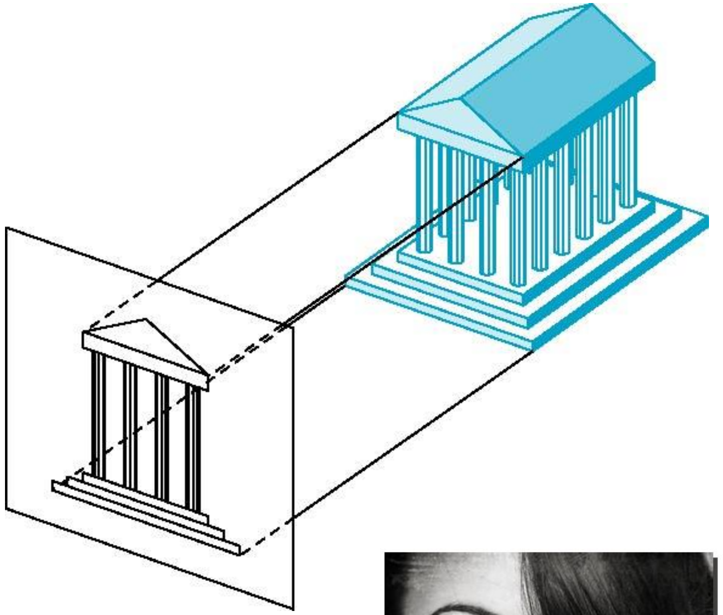
- Vzporedna projekcija iz Mezopotamije (2150 pr. n. št.)
  - najstarejša poznana tehnična risba
- Egipčani (grob Nefertari v Tebah, 1300 pr.n.št.)
  - vzporedna projekcija
  - več gledišč (telo vs. noge in glava)





Pravokotne (ortografske) projekcije

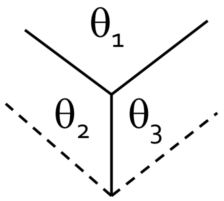
- Projekcijski žarki so **pravokotni** na ravnino projekcije
- Predmeti daleč izgledajo enako veliki kot tisti blizu
  - **ni** efektov **perspektive** – podoben efekt dosežejo telefoto leče



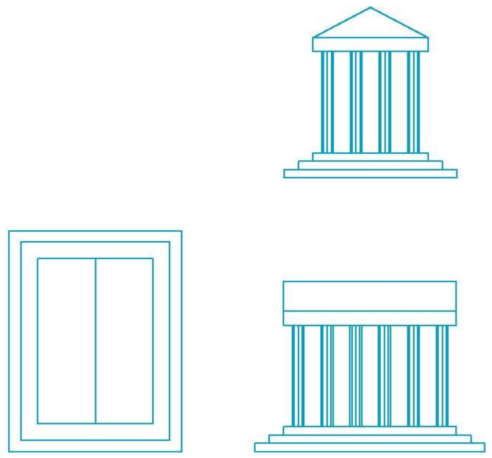
http://farm4.static.flickr.com/3057/2555706112_20a3015ddb.jpg

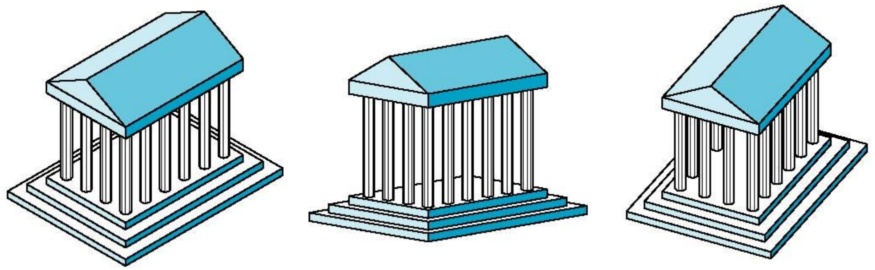
Pravokotne projekcije

- Glede na položaj kamere lahko ločimo različne poglede
  - projekcijska ravnina je **vzporedna** z osnovnimi pogledi na predmet
    - navadno tloris, naris, stranski ris
    - ohranja dolžine stranic in kote
  - projekcijska ravnina je **poševna** glede na predmet
    - glede na število kotov, ki so na projicirani kocki enaki ločimo:
      - **izometrična** (3), dimetrična (2), trimetrična (0)



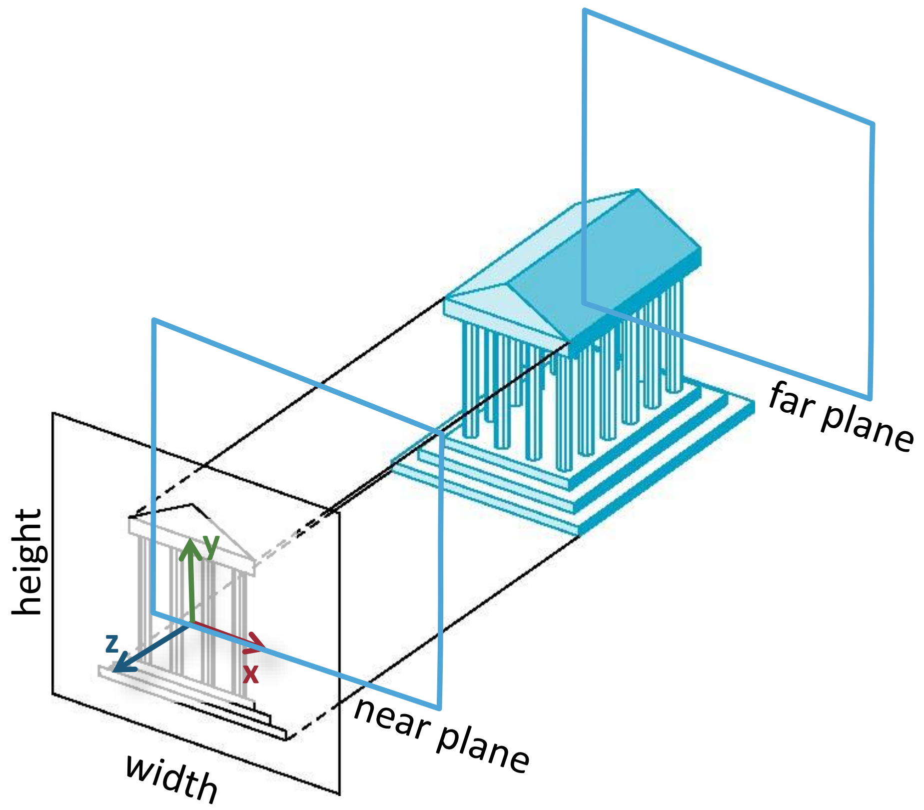
    - razmerja dolžin črt se ohranijo, koti se ne





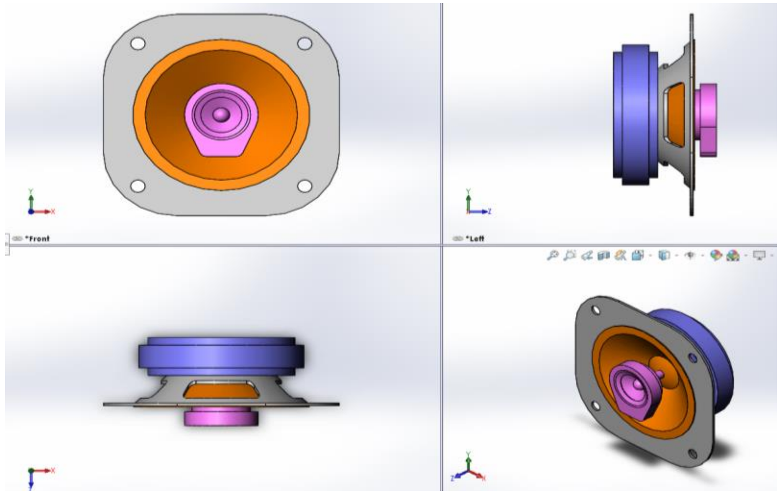
Matematika pravokotnih projekcij

- V koordinatah pogleda sta x in y poravnana z ravnino projekcije, z pa pravokotno kaže v 3D sceno (levosučni) ali stran (desnosučni k.s.)
- Pri pravokotnih projekcijah torej lahko **ohranimo** x in y
  - z obdržimo, saj ga v cevovodu potrebujemo pri določanju kaj je spredaj in kaj je zadaj
- Določimo še **vidno polje** (kaj bo kamera videla)
  - širino in višino slike
  - prednjo in zadnjo ravnino, ki določata kaj vidimo po z koordinati
- Pri transformaciji poskrbimo za
  - x in y v vidnem polju **poskaliramo** na interval [-1,1]
  - z **ohranimo** in poskaliramo na interval [0,1]
  - koordinatni sistem obrnemo iz desnosučnega v **levosučnega** (z množimo z -1), da z raste z globino
  - tem koordinatam rečemo **normalizirane koordinate naprave**
  - $$\mathrm{P}' = \begin{bmatrix} \frac{2}{\mathrm{w}} & 0 & 0 & 0 \\ 0 & \frac{2}{h} & 0 & 0 \\ 0 & 0 & \frac{-1}{f-n} & \frac{-\mathrm{n}}{f-n} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$



Pravokotne projekcije

- CAD, tehnične risbe, ilustracije
  - tloris, naris, stranski ris, izometrična projekcija
  - natančen prikaz predmeta (lahko izračunamo dolžine stranic, kote)
- Igre
  - koristno, ker so oddaljeni predmeti enako veliki kot bližnji - vidimo tudi podrobnosti oddaljenih predmetov



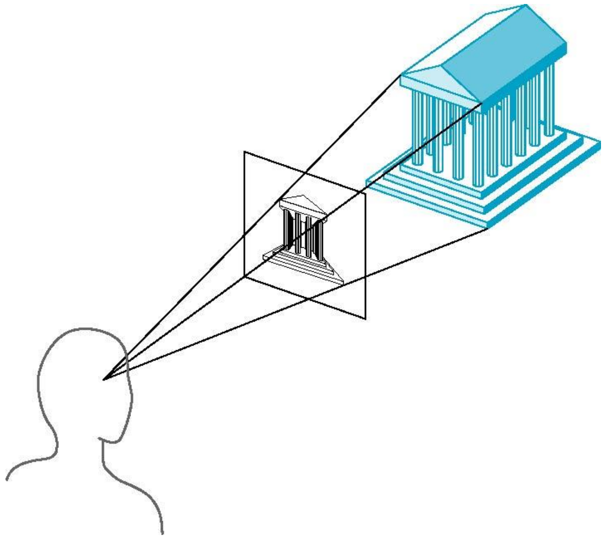
SolidWorks

Maxis: SimCity 4 - trimetrična projekcija

Perspektivna projekcija

Perspektivna projekcija

- Projekcijski žarki niso vzporedni - stikajo se v točki (**oko** - **gledišče**)
- Črte, ki so na predmetu vzporedne, in niso na ravnini, ki je vzporedna projekcijski ravnini, se sekajo v **ponornih točkah**
- Ravnina projekcije je nekje med gledišečem in sceno

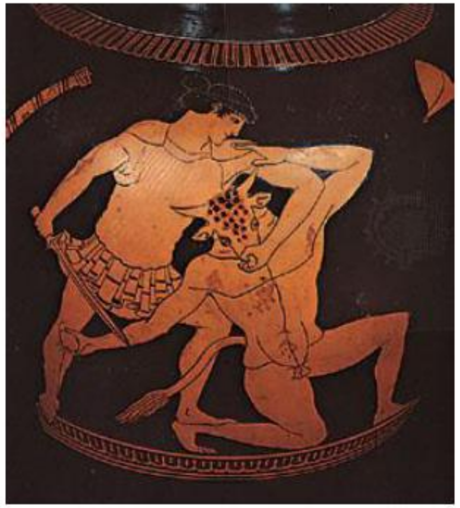


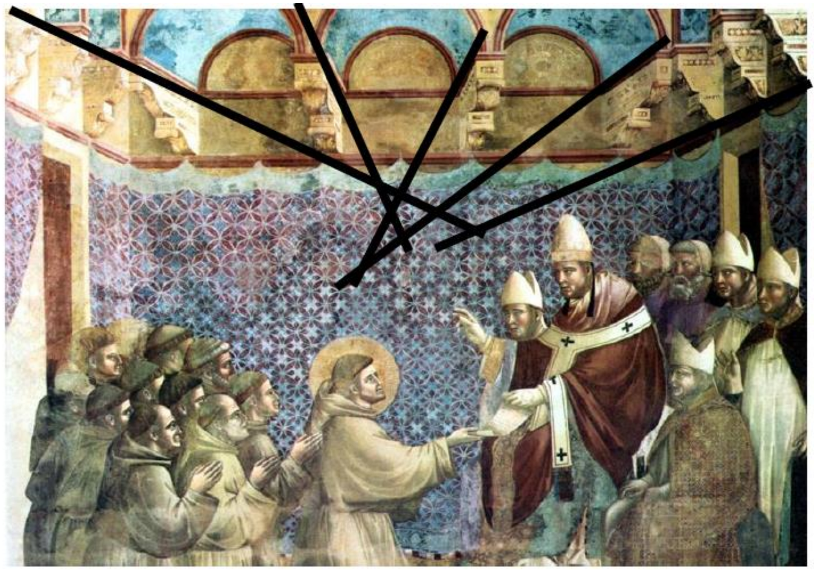
Maden NFL

Malo zgodovine

- Grška vaza iz 6. stoletja pr.n.št.
  - znaki perspektive (npr. noge minotavra)

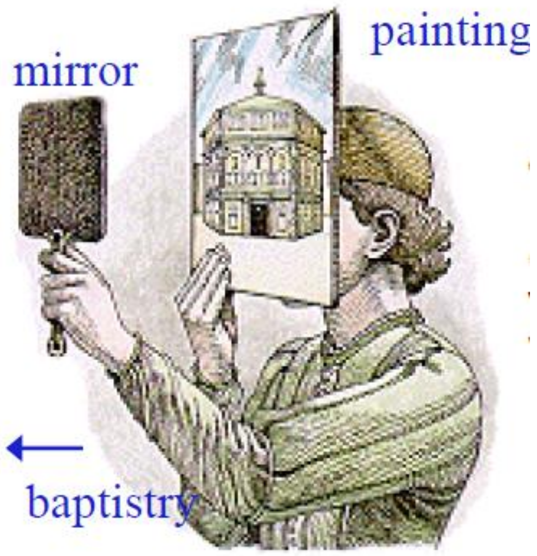
- Renesansa
  - Zgodnji poskusi perspektive – še ne sistematično oz. matematično
  - Npr. Giotto: *“Odobritev Frančiškanskega reda”,* cca. 1300
    - *črte konvergirajo, vendar ne v eno točko*





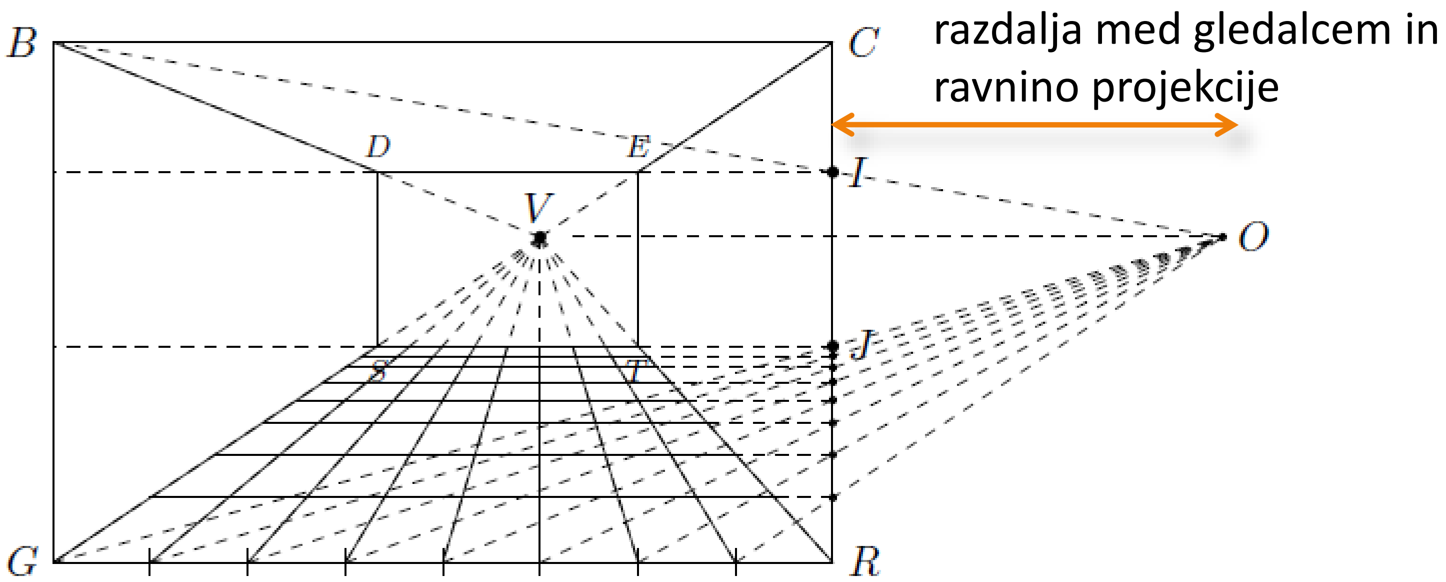
Renesansa

- **Brunelleschi** je izumil sistematično metodo za risanje linearne perspektive (cca. 1400)
  - čeprav neposredni zapisi niso ohranjeni
- Znan poskus s sliko krstilnice v Firencah
  - gledalec skozi luknjo v sliki opazuje pravo krstilnico, hkrati lahko z ogledalom vidi sliko in primerja videno



Renesansa

- **Alberti** (1435 – *Della Pittura*) je postavil prvo matematično pravilno metodo za risanje perspektive
  - primer risanja kvadra (sobe), tla so kockasta
    - prednja stena je GRCB, zadnja stena je STED
    - V je položaj oči gledalca

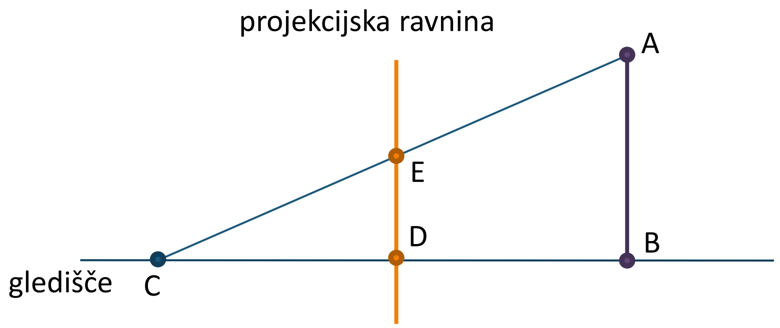


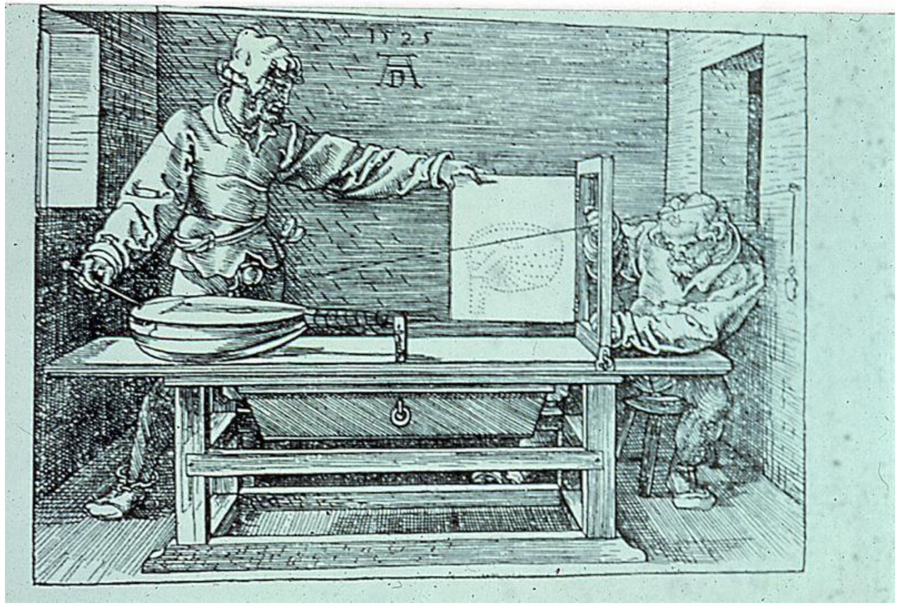
Renesansa

- Princip podobnih trikotnikov je opisal Dürer okoli 1500

AB/CB=ED/CD

  - AB = višina predmeta
  - CB = razdalja od gledalca do predmeta
  - CD = razdalja od gledalca do projekcijske ravnine

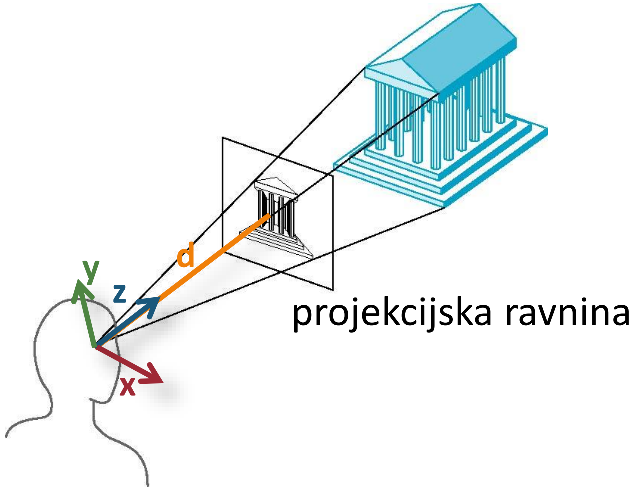


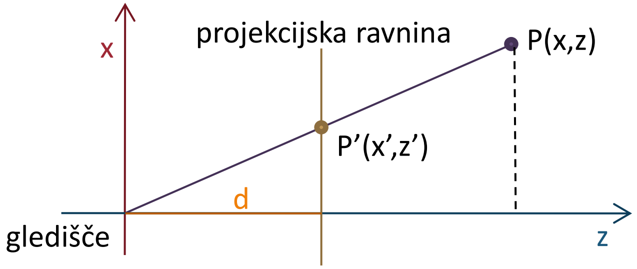


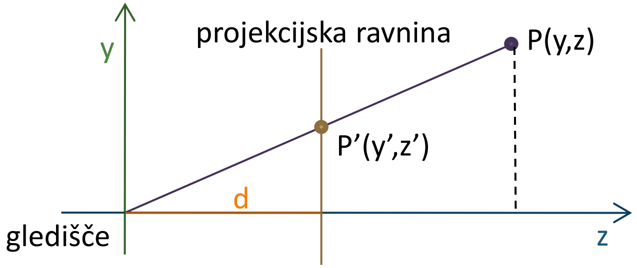
Dürer : Umetnik riše lutnjo (1525)

Matematika perspektive

- Če predpostavimo
  - predmeti so preslikani v k.s. kamere
  - *d* je razdalja od kamere do projekcijske ravnine
- Veljajo enakosti:
  - $\frac{x}{z} = \frac{x'}{d}$, $\frac{y}{z} = \frac{y'}{d}$, kar da $x' = \frac{xd}{z}, y' = \frac{yd}{z}$
- Kar da perspektivno matriko (levo s.):
  - $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \frac{1}{d} & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \\ \frac{z}{d} \end{bmatrix} = \begin{bmatrix} xd/z \\ yd/z \\ d \\ 1 \end{bmatrix}$

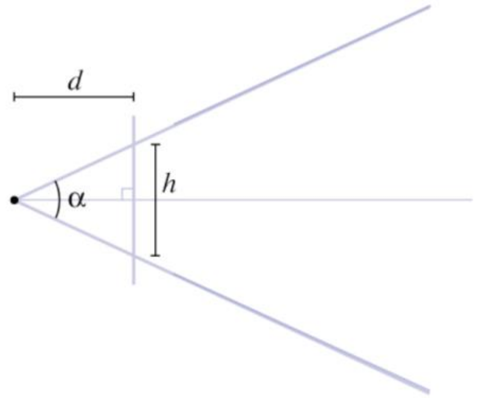






Zorni kot

- Da določimo izsek na projekcijski ravnini, ki ga kamera vidi, definiramo **zorni kot** (*field of view* - α)
  - iz zornega kota dobimo višino izseka *h*
  - širino izseka dobimo iz razmerja med višino in širino (*aspect ratio*)



$$\frac{h}{2d} = \tan\frac{\alpha}{2}$$

- S spreminjanjem zornega kota dosežemo različne učinke, npr. poudarimo efekt perspektive

širok zorni kot

ozek zorni kot

Normalizirane koordinate naprave (NDC)

- Za pretvorbo v **normalizirane koordinate naprave** ustrezno skaliramo vse tri koordinate
  - x in y koordinate oglišč bodo znotraj vidnega polja so na intervalu [-1,1]
  - z ohranimo in skaliramo na interval [0,1] (WebGPU)
  - spremenimo sučnost iz desno v levo sučni k.s. (-z)
- Matrika perspektivne projekcije morata torej ustrezno **skalirati** vse tri koordinate
  - glej npr. implementacijo matrik v glMatrix

Primer matrike perspektivne projekcije, ki preslika točko v NDC. Ta projekcija obrne sučnost koordinat sveta iz desno v levo sučen k.s., zato je d=-1 .

$$M_p = \begin{bmatrix} \frac{2}{w} & 0 & 0 & 0 \\ 0 & \frac{2}{h} & 0 & 0 \\ 0 & 0 & \frac{f}{n-f} & \frac{fn}{n-f} \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

w, h: *širina in višina vidnega polja*
n, f: *bližnja in daljna ravnina*

$$x' = -\frac{1}{z}\frac{2}{w}x$$
$$y' = -\frac{1}{z}\frac{2}{h}y$$
$$z' = \frac{f}{f-n} + \frac{fn}{f-n}\frac{1}{z}$$

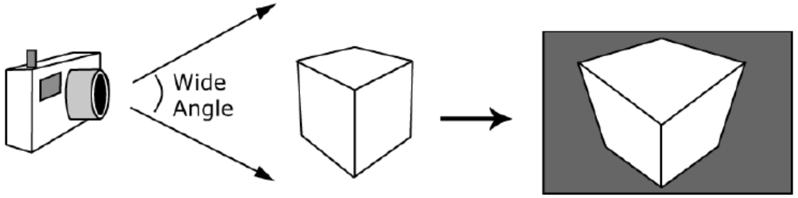
Parametri perspektivne projekcije

- Navadno določimo:

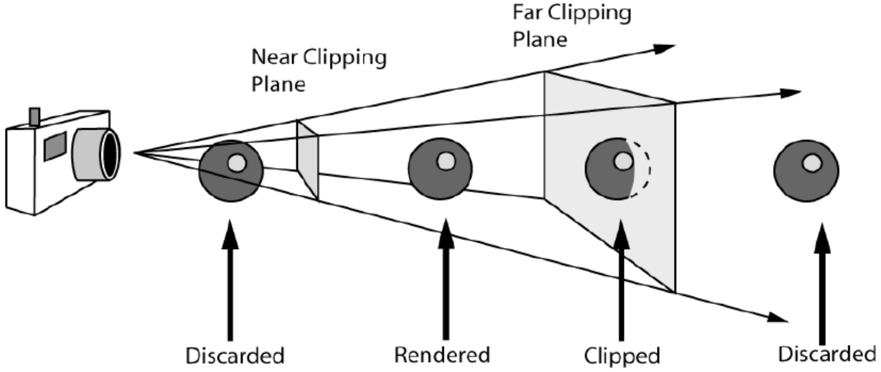
```
// perspective:          fov,    aspect, near, far
mat4.perspectiveZO(P, Math.PI/3, 4/3,  0.1,  100);
```

  - zorni kot (FOV)
    - če vzamemo npr. d=1, lahko izračunamo višino slike
  - razmerje med višino in širino slike (*aspect ratio*), npr. 4/3 ali 16/9
    - izračunamo širino slike
  - bližnjo in daljno ravnino rezanja
    - določata vidno prisekano piramido: **prostor gledanja**
    - le predmeti znotraj te piramide so vidni
      - nočemo gledati predmetov preblizu in izza kamere
      - nočemo prikazovati preveč oddaljenih predmetov

16:9



zorni kot



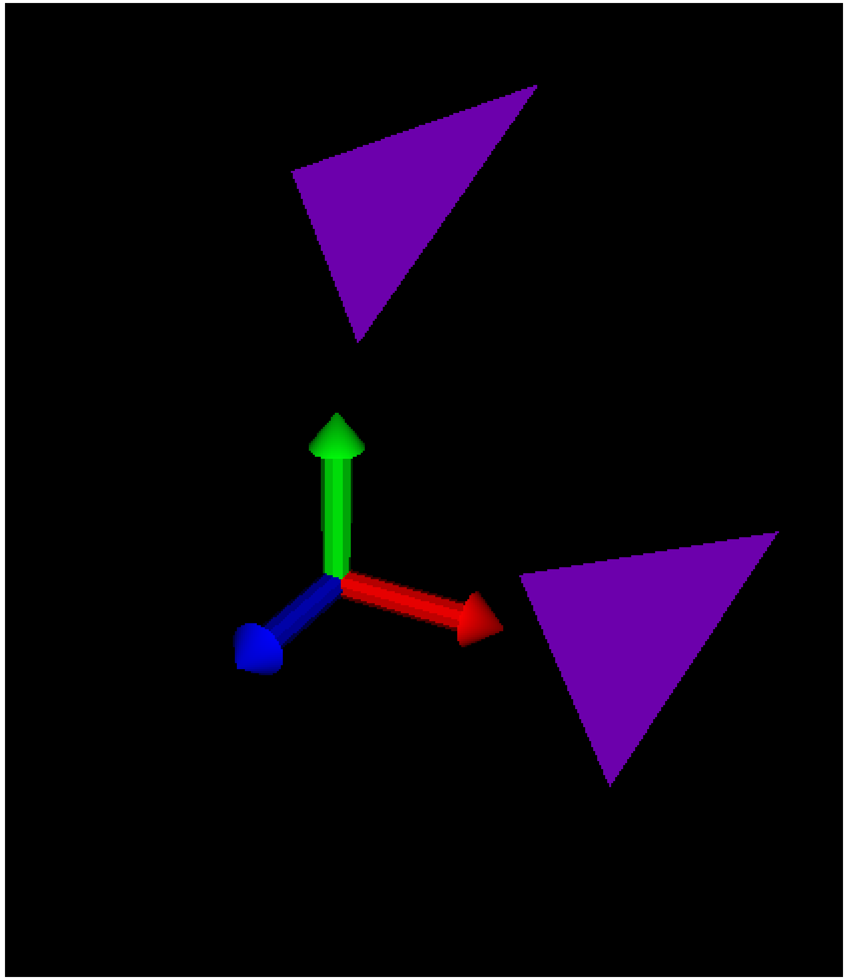
ravnine rezanja

Primer v kodi

- glMatrix za WebGPU
  - vzporedna projekcija z orthoZO
  - perspektiva s perspectiveZO

```
var ar = width/height;
var fld=10;
//                left right bottom top    near far
mat4.orthoZO(P, -fld,  fld,  -fld/ar,   fld/ar, 0.1, 100);

//                           FOV       aspect near far
mat4.perspectiveZO(P, Math.PI / 3,    ar,    0.1,  100);
```



REFERENCE

- R. Hammack: Alberti's method for Perspective Drawing
- N. Guid: Računalniška grafika, FERI Maribor
- J.D. Foley, A. Van Dam et al.: Computer Graphics: Principles and Practice in C, Addison Wesley
- P. Shirley, S. Marschner: Fundamentals of Computer Graphics, A.K. Peters