Graphics en Game Technologie

5. Viewing en Bézier krommen

Robert Belleman

Computational Science Lab
Universiteit van Amsterdam
R.G.Belleman@uva.nl
(voeg a.u.b. "[GGT]" toe aan subject)



Utah teapot in the Computer History Museum, Mountainview, CA, USA

Verzin een tentamenvraag!

- Verzin een tentamenvraag n.a.v. dit college
- Dien in op <u>dit formulier</u>
- Als je vraag wordt gebruikt tijdens een (deel)tentamen ontvang jij een half punt extra op je eindcijfer voor dat (deel)tentamen (bv.: had je een 7, dan krijg je een 7,5)!*

^{*:} Vragen die voorgaande jaren al eens op een (deel)tentamen voor dit vak zijn gesteld zijn van deze regeling uitgesloten.

Overzicht

- 1. Viewing: creëren van 2D beelden van 3D scenes
 - Orthografische (of parallelle) projectie
 - Perspectief projectie
- 2. Bézier krommen

Viewing

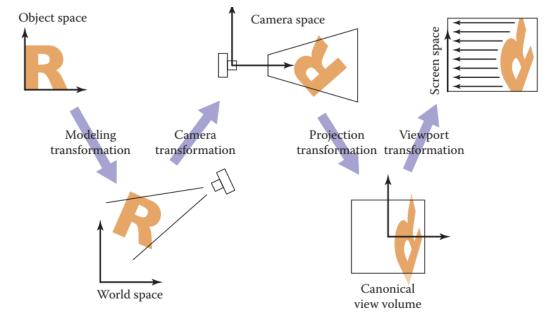
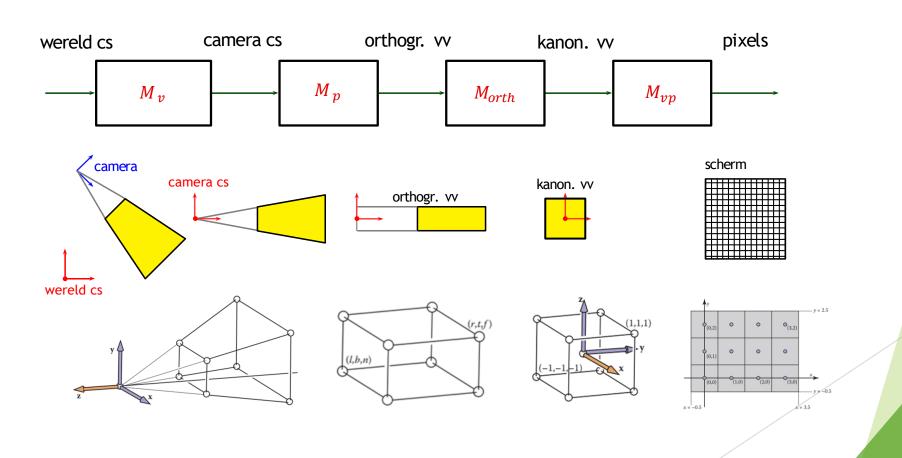


Figure 7.2. The sequence of spaces and transformations that gets objects from their original coordinates into screen space.

Grafische pijplijn



Projecties

- Methoden voor genereren van projecties van lijnsegmenten
- orthografische projectie
- perspectief projectie

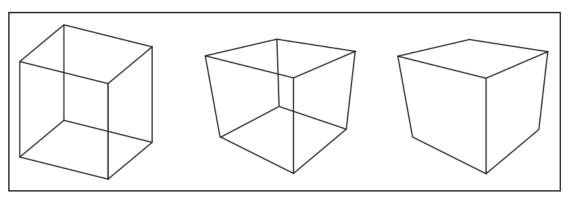
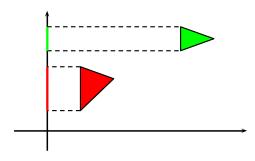


Figure 7.1. Left: wireframe cube in orthographic projection. Middle: wireframe cube in perspective projection. Right: perspective projection with hidden lines removed.

Parallelle projectie

Transformeert 3D punt naar 2D langs projectierichting op beeldvlak

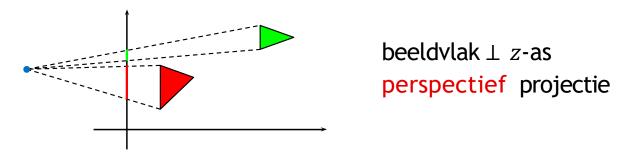


beeldvlak \(\perp \) projectie richting: orthografische projectie

- parallelle lijnen blijven parallel
- grootte en vorm van objecten blijven behouden

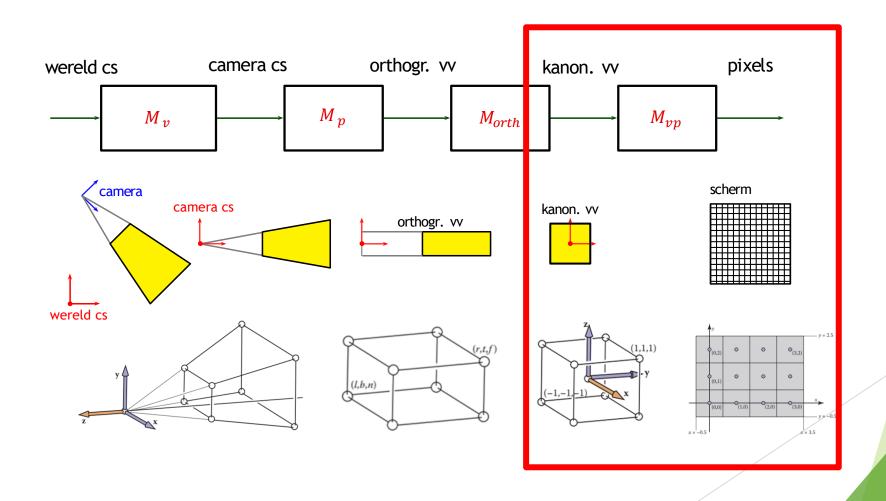
Perspectief projectie

Transformeert 3D punt naar 2D langs lijn door viewpunt tot snijpunt met beeldvlak



objecten verder van viewpunt vandaan worden kleiner

M_{vp} : kanonieke view volume \rightarrow scherm



M_{vv} : kanonieke view volume \rightarrow scherm

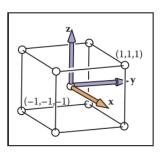


Figure 7.3. The canonical view volume is a cube with side of length two centered at the origin.

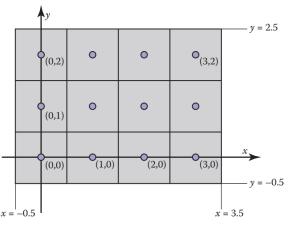


Figure 3.10. Coordinates of a four pixel \times three pixel screen. Note that in some APIs the *y*-axis will point downward.

Zoek afbeelding view volume naar scherm langs z-as

Punten in kanonieke view volume worden orthografisch geprojecteerd op $n_x \times n_y$ beeld

Schermcoördinaten

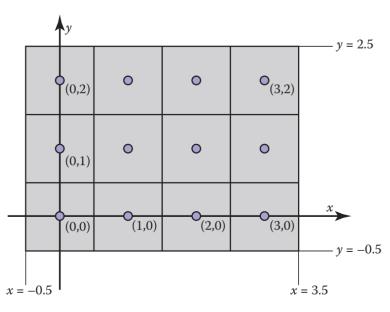


Figure 3.10. Coordinates of a four pixel \times three pixel screen. Note that in some APIs the y-axis will point downward.

Window van $[-0.5, n_x - 0.5]$ tot $[-0.5, n_y - 0.5]$ $(n_x = 4, n_y = 3)$

Transformatie van viewcoördinaten naar schermcoördinaten

```
rechthoek (a,b)(A,B) \rightarrow (c,d)(C,D)

transformatie =

transleer (c,d) schaal (\frac{C-c}{A-a},\frac{D-d}{B-b}) transleer (-a,-b)

rechthoek (-1,-1)(1,1) \rightarrow (-0.5,-0.5)(n_x-0.5,n_y-0.5)

transformatie =

transleer (-0.5,-0.5) schaal (\frac{n_x}{2},\frac{n_y}{2}) transleer (1,1)
```

Transformatie van view volume naar scherm

transformatie =

transleer (-0.5, -0.5) schaal $(\frac{n_x}{2}, \frac{n_y}{2})$ transleer (1, 1)

$$\begin{pmatrix} 1 & 0 & -0.5 \\ 0 & 1 & -0.5 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \frac{n_x}{2} & 0 & 0 \\ 0 & \frac{n_y}{2} & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} =$$

$$\begin{pmatrix} \frac{n_x}{2} & 0 & -0.5 \\ 0 & \frac{n_y}{2} & -0.5 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} \frac{n_x}{2} & 0 & \frac{n_x - 1}{2} \\ 0 & \frac{n_y}{2} & \frac{n_y - 1}{2} \\ 0 & 0 & 1 \end{pmatrix}$$

Transformatie M_{vp}

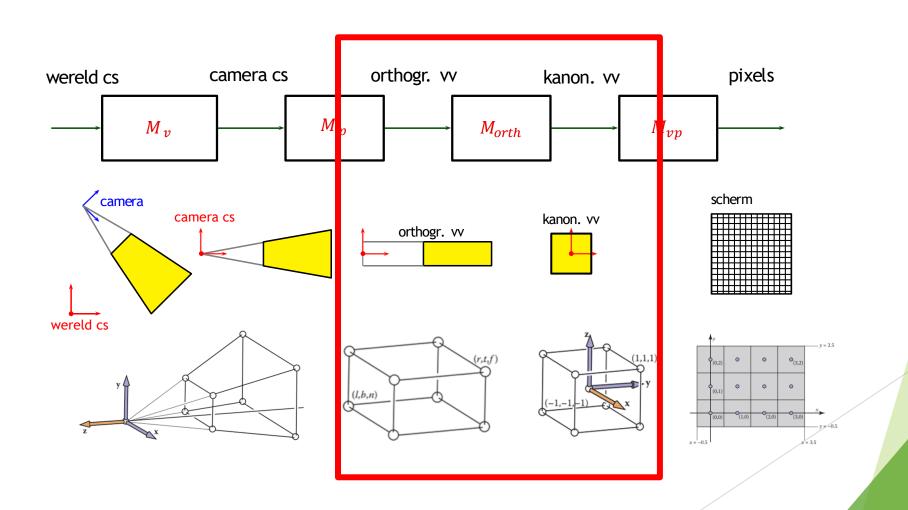
$$\begin{pmatrix} x_{pixel} \\ y_{pixel} \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{n_x}{2} & 0 & \frac{n_x - 1}{2} \\ 0 & \frac{n_y}{2} & \frac{n_y - 1}{2} \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_{canonical} \\ y_{canonical} \\ 1 \end{pmatrix}$$

Orthografische projectie heeft z-waarden "weggegooid"

Vaak z-waarden gebruikt voor ordenen van oppervlakken

$$\begin{pmatrix} x_{pixel} \\ y_{pixel} \\ z_{canonical} \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{n_x}{2} & 0 & 0 & \frac{n_x - 1}{2} \\ 0 & \frac{n_y}{2} & 0 & \frac{n_y - 1}{2} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_{canonical} \\ y_{canonical} \\ z_{canonical} \\ 1 \end{pmatrix}$$

M_{orth} : orthografische vv \rightarrow kanonieke vv



M_{orth} : orthografische vv \rightarrow kanonieke vv

Zoek afbeelding van orthografische view volume naar kanonieke view volume

box $[l,r] \times [b,t] \times [n,f]$

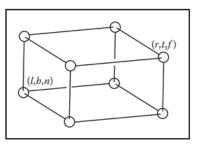


Figure 7.4. The orthographic view volume.

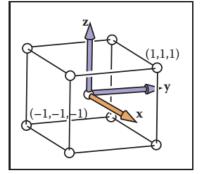


Figure 7.3. The canonical view volume is a cube with side of length two centered at the origin.

Orthografische view volume

- ▶ Viewer kijkt in richting van negatieve z-as, dus n > f
- ► Orthografische view volume \rightarrow standaard view volume is 3D versie van windowtransformatie: S_cT_O

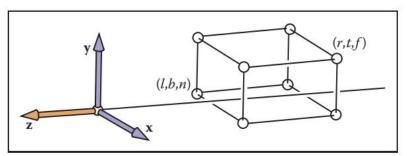


Figure 7.5. The orthographic view volume is along the negative z-axis, so f is a more negative number than n, thus n > f.

M_{orth} : orthografische vv \rightarrow kanonieke vv

$$x = l \rightarrow x = -1$$
 $x = r \rightarrow x = 1$
 $y = b \rightarrow y = -1$ $y = t \rightarrow y = 1$
 $z = n \rightarrow z = -1$ $z = f \rightarrow z = 1$

schaal

transleer naar oorsprong

$$\begin{pmatrix} x_{can} \\ y_{can} \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{2}{r-l} & 0 & 0 & 0 \\ 0 & \frac{2}{t-b} & 0 & 0 \\ 0 & 0 & \frac{2}{n-f} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & -\frac{l+r}{2} \\ 0 & 1 & 0 & -\frac{b+t}{2} \\ 0 & 0 & 1 & -\frac{n+f}{2} \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

$M_o = M_{vp} M_{orth}$: orthografische vv \rightarrow scherm

3D lijnsegmenten projecteren op scherm. Wat gebeurt er met de z-waarden?

$$\boldsymbol{M_0} = \begin{pmatrix} \frac{n_x}{2} & 0 & 0 & \frac{n_x - 1}{2} \\ 0 & \frac{n_y}{2} & 0 & \frac{n_y - 1}{2} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \frac{2}{r - l} & 0 & 0 & 0 \\ 0 & \frac{2}{t - b} & 0 & 0 \\ 0 & 0 & \frac{2}{n - f} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & -\frac{l + r}{2} \\ 0 & 1 & 0 & -\frac{b + t}{2} \\ 0 & 0 & 1 & -\frac{n + f}{2} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\begin{pmatrix} x_{pixel} \\ y_{pixel} \\ z_{canonical} \\ 1 \end{pmatrix} = \mathbf{M_0} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

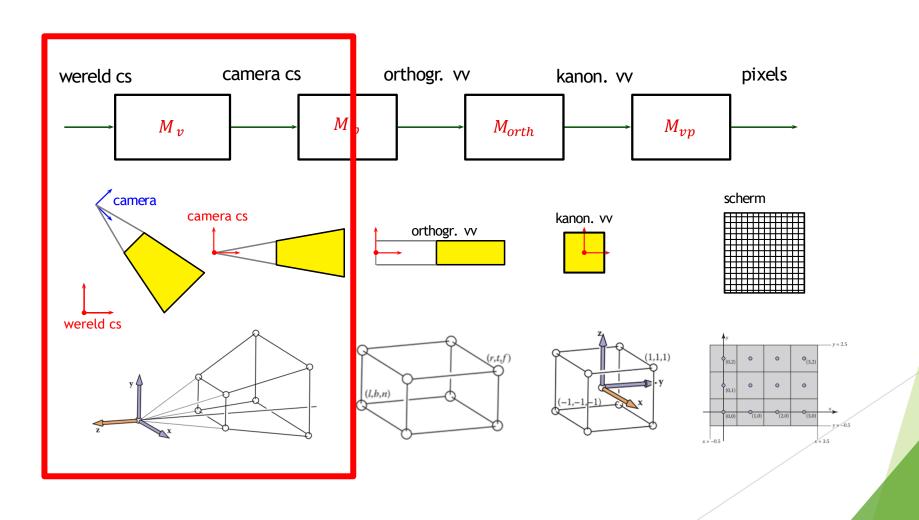
M_0 : orthografische vv \rightarrow scherm

$$\begin{pmatrix} x_{pixel} \\ y_{pixel} \\ z_{canonical} \\ 1 \end{pmatrix} = \mathbf{M_0} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

 $z_{canonical}$ in [-1,1] (gebruikt in z-buffer algoritme)

```
bereken MO
for elk lijnsegment (a,b)
do
    p = MO a
    q = MO b
    draw(p, q)
```

M_v : wereldcoördinaten naar cameracoördinaten



M_v : wereldcoördinaten naar cameracoördinaten

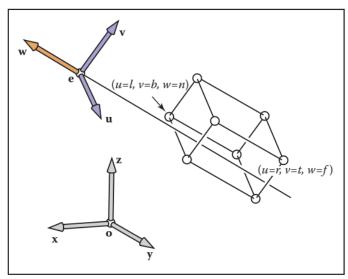


Figure 7.7. For arbitrary viewing, we need to change the points to be stored in the "appropriate" coordinate system. In this case it has origin **e** and offset coordinates in terms of **uvw**.

- ightharpoonup wereldcoördinaten in standaardbasis x, y en z
- ightharpoonup cameracoördinaten met camerapositie e en basis u, v en w

Orthonormale basis van cameracoördinaten

- Gebruiker specificeert:
 - ▶ oogpositie e
 - ▶ kijkrichting *g*
 - ▶ view-up *t*
- Construeer orthonormale basis uvw met w tegengesteld aan kijkrichting en v in vlak van g en t

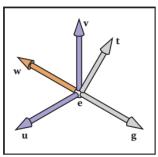


Figure 7.6. The user specifies viewing as an eye position \mathbf{e} , a gaze direction \mathbf{g} , and an up vector \mathbf{t} . We construct a right-handed basis with \mathbf{w} pointing opposite to the gaze and \mathbf{v} being in the same plane as \mathbf{g} and \mathbf{t} .

Coördinatensysteem met oorsprong e en basis uvw

$$u = \frac{t \times w}{\|t \times w\|}$$

$$v = w \times u$$

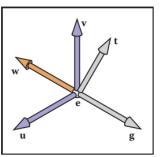


Figure 7.6. The user specifies viewing as an eye position **e**, a gaze direction **g**, and an up vector **t**. We construct a right-handed basis with **w** pointing opposite to the gaze and **v** being in the same plane as **g** and **t**.

Orthonormale basis *uvw*

1. Alignen van uvw naar xyz

$$u \xrightarrow{R} x$$
 $v \xrightarrow{R} y$
 $w \xrightarrow{R} z$

2. Veranderen coördinatensysteem

$$\begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \end{pmatrix}_{\mathbf{x}, \mathbf{y}, \mathbf{z}} \xrightarrow{R} \begin{pmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \end{pmatrix}_{\mathbf{u}, \mathbf{v}, \mathbf{w}}$$

$$R = R_{uvw}$$

$$\boldsymbol{R}_{\boldsymbol{u}\boldsymbol{v}\boldsymbol{w}} = \begin{pmatrix} x_u & y_u & z_u \\ x_v & y_v & z_v \\ x_w & y_w & z_w \end{pmatrix}$$

 R_{uvw} voert punt in wereldcoördinaten over naar punt in cameracoördinaten op offset na: namelijk de positie van camera

Transformatie:

- 1. translatie van e naar oorsprong
- 2. uvw roteren naar xyz

Wereldcoördinaten → cameracoördinaten

(View volume t.o.v. *uvw*)

$$\mathbf{M}_{v} = \begin{pmatrix} x_{u} & y_{u} & z_{u} & 0 \\ x_{v} & y_{v} & z_{v} & 0 \\ x_{w} & y_{w} & z_{w} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & -x_{e} \\ 0 & 1 & 0 & -y_{e} \\ 0 & 0 & 1 & -z_{e} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

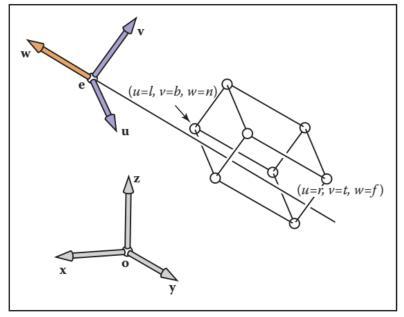


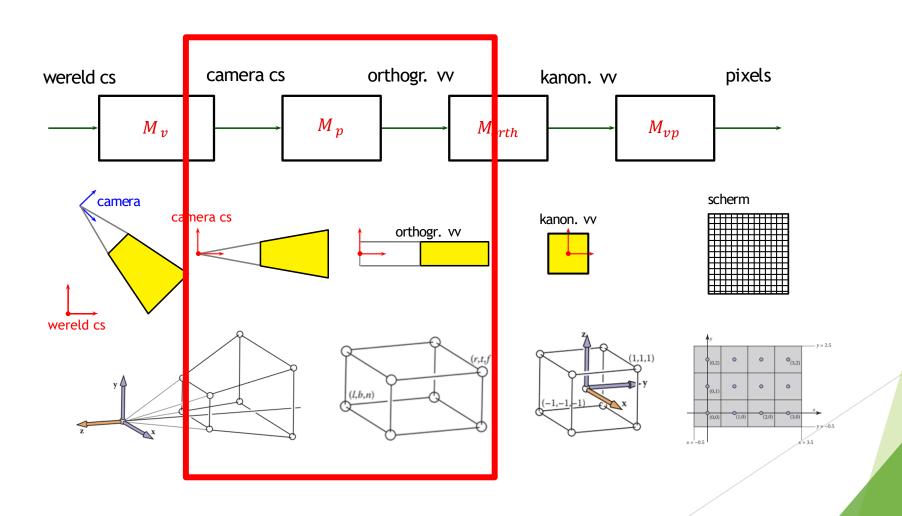
Figure 7.7. For arbitrary viewing, we need to change the points to be stored in the "appropriate" coordinate system. In this case it has origin **e** and offset coordinates in terms of **uvw**.

Wereld

$\xrightarrow{orthografische\ projectie} \xrightarrow{scherm}$

```
bereken MV
M = MO MV
for elk lijnsegment (a,b)
do
p = Ma
q = Mb
draw(p,q)
\begin{pmatrix} x_{pixel} \\ y_{pixel} \\ z_{canonical} \end{pmatrix} = M_{vp} M_0 M_v \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}
```

M_p : transformeren van view frustum



M_p : transformeren van view frustum

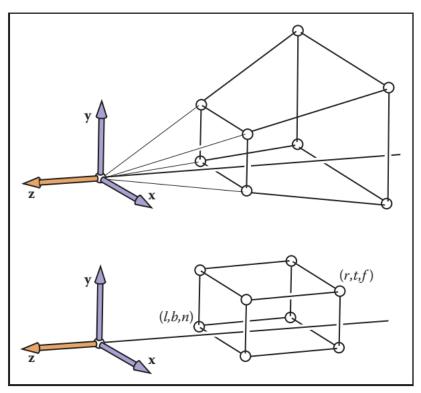
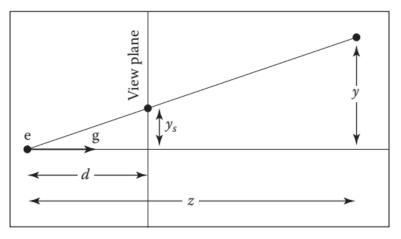


Figure 7.12. The perspective projection leaves points on the z = n plane unchanged and maps the large z = f rectangle at the back of the perspective volume to the small z = f rectangle at the back of the orthographic volume.

Perspectief projectie

positie oog op e
kijkrichting g



viewplane op afstand d

$$y_{s} = \frac{d}{z}y$$

Figure 7.8. The geometry for Equation (7.5). The viewer's eye is at \mathbf{e} and the gaze direction is \mathbf{g} (the minus z-axis). The view plane is a distance d from the eye. A point is projected toward \mathbf{e} and where it intersects the view plane is where it is drawn.

Perspectief transformatie

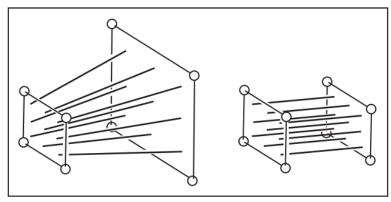


Figure 7.13. The perspective projection maps any line through the origin/eye to a line parallel to the z-axis and without moving the point on the line at z = n.

We hebben matrix nodig die het volgende doet:

1.
$$x_S = \frac{d}{z}x$$
 en $y_S = \frac{d}{z}y$

2. z-waarde blijft hetzelfde voor alle punten in near (n) en far(f) vlak

Perspectief transformatie M_p

$$\mathbf{M}_{p} = \begin{pmatrix} n & 0 & 0 & 0 \\ 0 & n & 0 & 0 \\ 0 & 0 & n+f & -fn \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Opmerkingen:

- We kijken in negatieve z-richting
- \triangleright n en f geven voor- en achtervlak aan van view frustum
- n dient als projectievlak

Perspectief transformatie M_p

$$\begin{pmatrix} n & 0 & 0 & 0 \\ 0 & n & 0 & 0 \\ 0 & 0 & n+f & -fn \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} nx \\ ny \\ z(n+f)-fn \\ z \end{pmatrix} \xrightarrow{homogeen} \begin{pmatrix} \frac{nx}{z} \\ \frac{ny}{z} \\ n+f-\frac{fn}{z} \\ 1 \end{pmatrix}$$

$$z = n \xrightarrow{M_p} ?$$

$$z = f \xrightarrow{M_p} ?$$

Perspectief transformatie M_p

$$\mathbf{M}_{p}(\mathbf{z} = \mathbf{n}) = \begin{pmatrix} n & 0 & 0 & 0 \\ 0 & n & 0 & 0 \\ 0 & 0 & n+f & -fn \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ n \\ 1 \end{pmatrix} = \begin{pmatrix} nx \\ ny \\ n(n+f) - fn \\ n \end{pmatrix} = \frac{1}{n} \begin{pmatrix} x \\ y \\ n \\ 1 \end{pmatrix}$$

 $\Rightarrow x$ en y in voorvlak

$$M_{p}(z = f) = \begin{pmatrix} n & 0 & 0 & 0 \\ 0 & n & 0 & 0 \\ 0 & 0 & n+f & -fn \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ f \\ 1 \end{pmatrix} = \begin{pmatrix} nx \\ ny \\ f(n+f) - fn \\ f \end{pmatrix} = \frac{1}{f} \begin{pmatrix} \overline{f} \\ \underline{ny} \\ f \\ f \end{pmatrix}$$

 $\Rightarrow x$ en y in achtervlak geschaald met $\frac{n}{f}$

Homogene perspectief transformatie M_p

$$\mathbf{M}_{p} = \begin{pmatrix} n & 0 & 0 & 0 \\ 0 & n & 0 & 0 \\ 0 & 0 & n+f & -fn \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

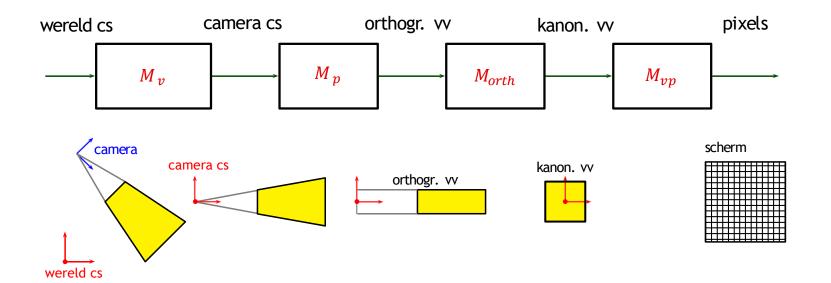
Homogene perspectief transformatie ${\it M}_p$ zet perspectief view volume om in orthografisch view volume

Wereldcoördinaten → schermcoördinaten

```
bereken MO
bereken Mv
bereken Mp
M = MO Mp Mv
for elk lijnsegment (a,b)
do
    p = Ma
    q = Mb
    deel homogene coördinaten van p en q door w
    draw(p,q)
```

wereld $\stackrel{M_v}{\rightarrow}$ camera $\stackrel{M_p}{\rightarrow}$ orthogr.view v $\stackrel{M_o}{\rightarrow}$ scherm

Grafische pijplijn

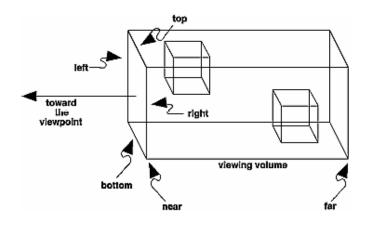


Andere aspecten van de grafische pijplijn

We kunnen nu punten en lijnen tekenen

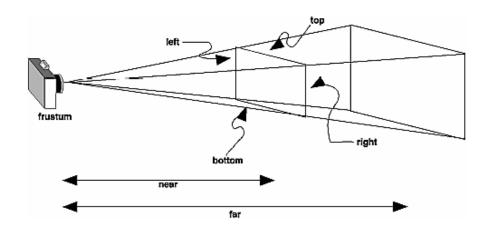
- Driehoeken die gedeeltelijk buiten view frustum liggen moeten geclipped worden
- Overblijvende driehoeken moeten geprojecteerd worden als ze front facing zijn
- Geprojecteerde driehoeken moeten ingekleurd en/of textured worden

Orthografische projectie in OpenGL (M_{orth})



```
glMatrixMode(GL_PROJECTION);
glLoadIdentity();
gluOrtho(left, right, bottom, top, near, far);
```

Perspectiefprojectie in OpenGL $(M_{orth}M_p)$

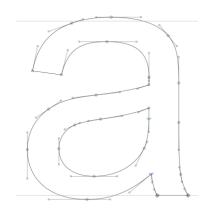


```
glMatrixMode(GL PROJECTION);
glLoadIdentity();
glFrustum(left, right, bottom, top, near, far);
// of
gluPerspective(fovy, aspect, near, far);
```

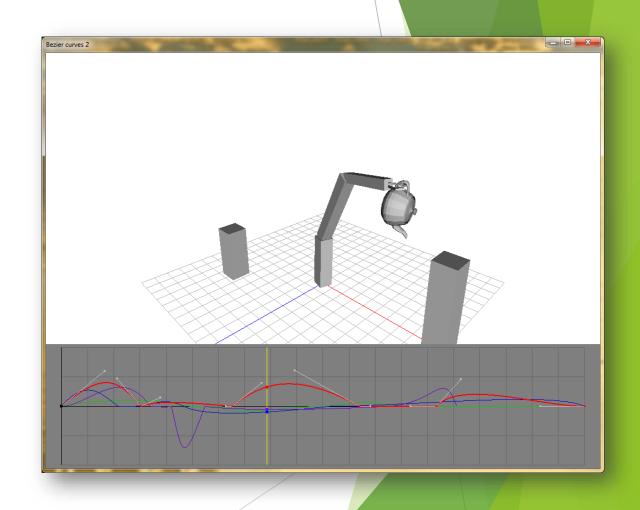
VR Experiment: grafische pijplijn in VR

- Testpersonen gezocht
- A.s. vrijdag 23 september in Visualisation Lab tussen 09:00 en 17:00
- ▶ 15 minuten max
- Schrijf in via dit formulier!



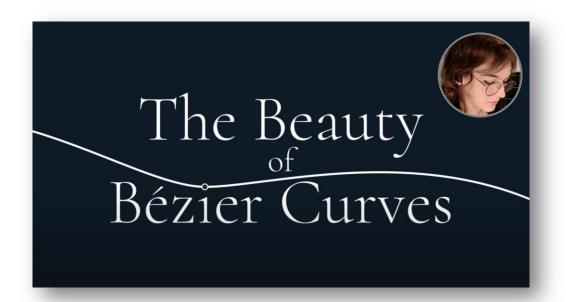


Bézier krommen



Tip

YouTube video by Freya Holmér (25min):



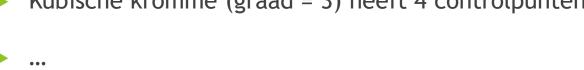
Bézier krommen

Bézier kromme wordt bepaald door

- orde
- aantal controlpunten

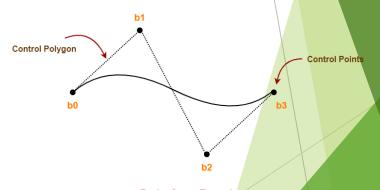
Een kromme van graad n heeft n+1 controlpunten P_0, P_1, \dots, P_n

- Lineaire kromme (graad = 1) heeft 2 controlpunten
- Kwadratische kromme (graad = 2) heeft 3 controlpunten
- Kubische kromme (graad = 3) heeft 4 controlpunten



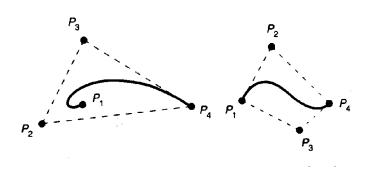


Pierre Bézier



Source

Convexe omhulsel eigenschap



Kromme binnen convexe omhulsel (stippellijnen)

Lineaire Bézier kromme

- Gegeven punten P_0 en P_1
- Lineaire Bézier kromme is rechte lijn tussen P_0 en P_1
- $B(t) = P_0 + t(P_1 P_0) = (1 t)P_0 + tP_1, \ t \in [0, 1]$
- Opmerking: som van gewichten is 1

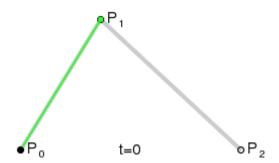
Kwadratische Bézier kromme

Gegeven punten P_0 , P_1 en P_2

Kwadratische Bézier kromme is parabool door P_0 en P_2 (begin- en eindpunt)

$$B(t) = (1-t)^2 P_0 + 2(1-t)t P_1 + t^2 P_2, \qquad t \in [0,1]$$

Opmerking: som van gewichten is 1



Kwadratische Bézier kromme

Wat doet P_1 ?

 P_1 bepaalt de raaklijn in begin- en eindpunt

$$B(t) = (1-t)^2 P_0 + 2(1-t)t P_1 + t^2 P_2, t \in [0,1]$$

$$B'(t) = -2(1-t)P_0 + 2(1-2t)P_1 + 2t P_2$$

$$B'(0) = -2P_0 + 2P_1 = 2(P_1 - P_0)$$
 is de raaklijn in P_0

$$B'(1) = -2P_1 + 2P_2 = 2(P_2 - P_1)$$
 is de raaklijn in P_2

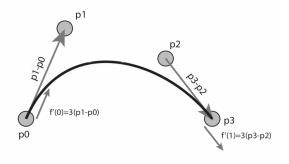
Derdegraads Bézier kromme

Gegeven punten P_0 , P_1 , P_2 en P_3

Derdegraads Bézier kromme is 3 degraads kromme door P_0 en P_3

$$B(t) = (1-t)^3 P_0 + 3t(1-t)^2 P_1 + 3t^2(1-t) P_2 + t^3 P_3, \qquad t \in [0,1]$$

Opmerking: som van gewichten is 1



ne graads Bézier kromme

$$P(u) = \sum_{i=0}^{n} B_i^n(u) P_i$$

Met $B_i^n(u)$ Bernstein basis polynomen

Bernstein basis polynomen

Vier weegfuncties (blending functions) van derdegraads Bézier kromme heten Bernstein basis polynomen

$$b_{0,3} = (1-t)^3 = 1 - 3t + 3t^2 - t^3$$

$$b_{1,3} = 3t(1-t)^2 = 3t - 6t^2 + 3t^3$$

$$b_{2,3} = 3t^2(1-t) = 3t^2 - 3t^3$$

$$b_{3,3} = t^3 = t^3$$

Som weegfuncties 1 en niet-negatief ⇒ convex-hull eigenschap (denk aan barycentrische coördinaten: elk punt op kromme ligt binnen vierhoek)

Bézier kromme door meerdere punten?

Driehoek van Pascal: binomiaalcöefficiënten $\binom{n}{k} = \frac{n!}{k!(n-k)!}$

$$\binom{n}{0}$$
 $\binom{n}{1}$ $\binom{n}{2}$ $\binom{n}{3}$ \cdots $\binom{n}{n}$ rij n

Binomiaalcöefficiënten

Binomiaalformule:

$$(x+y)^n = \sum_{k=0}^n \binom{n}{k} x^{n-k} y^k$$

$$1 = 1^{n} = ((1-t) + t)^{n} = \sum_{k=0}^{n} {n \choose k} (1-t)^{n-k} t^{k}$$

Voorbeeld n = 3

$$(x+y)^3 = {3 \choose 0}x^3 + {3 \choose 1}x^2y + {3 \choose 2}xy^2 + {3 \choose 3}y^3 = x^3 + 3x^2y + 3xy^2 + y^3$$

$$1 = (1 - t)^3 + 3t(1 - t)^2 + 3t^2(1 - t) + t^3$$
 (som weegfuncties 1)

$$B(t) = (1-t)^3 P_0 + 3t(1-t)^2 P_1 + 3t^2(1-t)P_2 + t^3 P_3$$

Dus coëfficiënten Bézier kromme: $\binom{n}{k}(1-t)^{n-k}t^k$

negraads Bézier kromme

 n^e graads Bézier kromme door n+1 punten P_0 t/m P_n :

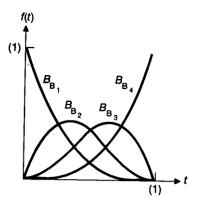
$$B(t) = \sum_{k=0}^{n} {n \choose k} (1-t)^{n-k} t^{k} P_{k}$$

n=3 en Bernstein basis polynomen

$$B(t) = \sum_{k=0}^{n} {n \choose k} (1-t)^{n-k} t^{k} P_{k}$$

$$B(t) = (1-t)^3 P_0 + 3t(1-t)^2 P_1 + 3t^2(1-t)P_2 + t^3 P_3$$

Coëfficiënten: $\binom{3}{0}\binom{3}{1}\binom{3}{2}\binom{3}{3}$



n=4

$$B(t) = \sum_{k=0}^{n} {n \choose k} (1-t)^{n-k} t^{k} P_{k}$$

4egraads Bézier kromme door 5 punten P_0 t/m P_4 :

$$B(t) = (1-t)^4 P_0 + 4t(1-t)^3 P_1 + 6t^2(1-t)^2 P_2 + 4t^3(1-t)P_3 + t^4 P_4$$

 $B(0) = P_0$, dus gaat door P_0

 $B(1) = P_4$, dus gaat door P_4

