5. Transformacija pogleda i perspektivna projekcija

Kako bi našu scenu promatrali iz bilo koje pozicije kamere (točka očišta) u bilo kojem smjeru (točka gledišta) potrebno je ostvariti transformaciju pogleda. Sustav scene je trodimenzijski sustav, slika 5.1. Sustav oka je trodimenzijski sustav čija je z-os upravljena u smjeru pogleda (gledišta), tako da z-os predstavlja dubinu slike. Sustav prikaza je dvodimenzijski sustav i smješten je u ravnini projekcije **R**.

Preslikavanje točaka iz sustava scene u sustav oka naziva se transformacija pogleda. Za transformaciju pogleda potrebno je odrediti matričnu vezu između koordinatnog sustava oka i sustava scene. Nakon transformacije pogleda obavlja se projekcija. Projekcija točaka iz sustava oka u sustav prikaza može se načiniti kao paralelna ili perspektivna projekcija.

Korištene oznake:

$x_s \quad y_s \quad z_s$ - sustav scene,

$x_0 \quad y_0 \quad z_0$ - sustav oka,

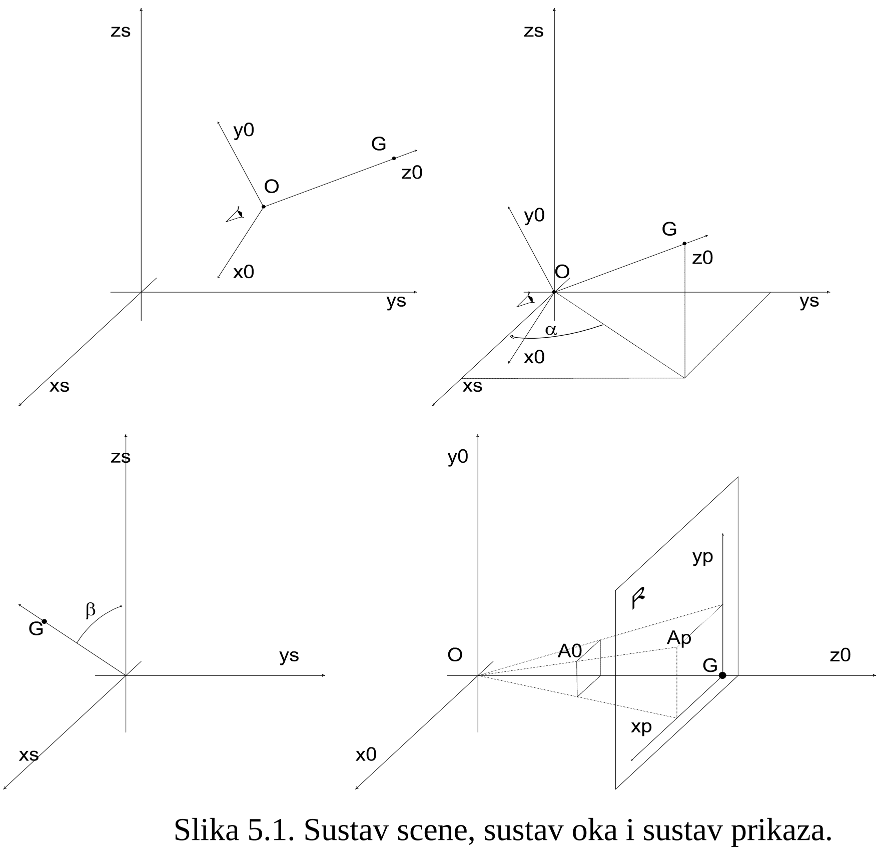
$x_p \quad y_p$ - sustav prikaza,

O - očište, položaj promatrača u sustavu scene,

G - gledište, točka u sustavu scene u koju je usmjeren pogled,

H - udaljenost ravnine projekcije od očišta, **H**=d(**O, G**),

R - ravnina projekcije, točka **G** leži u ravnini projekcije,



Slika 5.1. Sustav scene, sustav oka i sustav prikaza.

5.1. Transformacija pogleda

Za transformaciju pogleda i projekciju treba odrediti matricu **T** koja će točku scene $\mathbf{A}_o$ preslikati u točku $\mathbf{A}_s$ i nakon toga matricu **P** kojom točku $\mathbf{A}_s$ projiciramo u točku $\mathbf{A}_p$ na projekcijskoj ravnini **R**:

$$\mathbf{A}_s = \mathbf{A}_o\mathbf{T}, \quad \mathbf{A}_p = \mathbf{A}_s\mathbf{P} \tag{5.1}$$

Matrica **T** je sastavljena od matrica elementarnih transformacija koje prvo obavljaju translaciju odnosno pomak koordinatnog sustava oka u ishodište a zatim rotaciju i promjenu predznaka. Cilj transformacije pogleda je podudaranje koordinatnih osi sustava oka s koordinatnim osima sustava scene. Dakle, prvi korak je pomak koordinatnog sustava oka u ishodište, pa prva matrica je matrica translacije:

$\mathbf{T_1}$ - pomak ishodišta koordinatnog sustava oka u ishodište scene (Slika 5.1 prvi red),

Točke $\mathbf{O}(x_0 \quad y_0 \quad z_0)$ i $\mathbf{G}(x_g \quad y_g \quad z_g)$ mjere se u sustavu scene. Koordinatama točke **O** određena je matrica $\mathbf{T_1}$

$$\mathbf{T_1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -x_0 & -y_0 & -z_0 & 1 \end{bmatrix}. \tag{5.2}$$

Djelovanje $\mathbf{T_1}$ na **G** daje

$$\mathbf{G_1} = \mathbf{G}\,\mathbf{T_1} \quad \text{ili po koordinatama} \quad \begin{aligned} x_{g1} &= x_g - x_0 \\ y_{g1} &= y_g - y_0 \\ z_{g1} &= z_g - z_0 \end{aligned}$$

Nakon što su oba sustava smještena u isto ishodište, potrebno je još poklopiti osi oba koordinatna sustava. Podudaranje koordinatnih osi sustava oka s koordinatnim osima sustava scene možemo napraviti na dva načina. Prvi način koji ćemo izložiti temelji se na određivanju vektora koji čine koordinatni sustav oka a drugi način je određen nizom elementarnih matrica rotacije oko pojedinih koordinatnih osi sustava scene. U laboratorijskoj vježbi potrebno je odabrati i implementirati jedan od ova dva načina.

Prvi način određivanja matrice rotacije

Matrica rotacije zapravo je sačinjena od vektora koji predstavljaju osi rotiranog koordinatnog sustava. Kako bi nam ova činjenica intuitivno bila bliža prisjetimo se da je cilj matrice rotacije jednostavno zarotirati čitav koordinatni sustav, pa tako i sve točke u našem originalnom koordinatnom sustavu. To bi značilo da su se nakon rotacije koordinate svih točaka, gledano iz originalnog koordinatnog sustava, promijenile. Upravo nam matrica rotacije služi kako bismo odredili koordinate rotiranih točaka gledano iz originalnog koordinatnog sustava. No postavlja se pitanje kako jednostavno možemo odrediti matricu rotacije. Prisjetimo se da se svaka točka u prostoru može zapisati kao linearna kombinacija koordinatnih osi sustava, pa se tako točka $P_r = (p_{1r}, p_{2r}, p_{3r})$ može prikazati linearnom kombinacijom vektora osi rotiranog sustava $P_r = p_{1r}\overrightarrow{x_r} + p_{2r}\overrightarrow{y_r} + p_{3r}\overrightarrow{z_r}$, kao što je prikazano i na slici 5.2. Upravo ovo svojstvo možemo iskoristiti i za određivanje koordinata neke točke u koordinatnom sustavu nakon rotacije. Neka nam p_{1r}, p_{2r} i p_{3r} predstavljaju koordinate točke u rotiranom koordinatnom sustavu, a $\overrightarrow{x_r}, \overrightarrow{y_r}$ i $\overrightarrow{z_r}$ (označeni crvenom, zelenom i plavom bojom na slici 5.2) vektore koji predstavljaju osi rotiranog koordinatnog sustava gledano iz originalnog koordinatnog sustava (pri čemu su primjerice koordinate vektora $\overrightarrow{x_r}$ označene kao x_{1r}, x_{2r} i x_{3r}). Na temelju tih informacija možemo koordinate neke točke u originalnom koordinatnom sustavu odrediti na način da se za svaki od tih vektora pomaknemo za odgovarajuće iznose koji su određeni vrijednostima koordinata točke gledano iz rotiranog koordinatnog sustava. Dakle, zapravo se radi linearna kombinacija osi rotiranog koordinatnog sustava kako bismo odredili koordinate točke u originalnom koordinatnom sustavu. Čitav izraz onda možemo zapisati kao:

$$[p_{1o} \quad p_{2o} \quad p_{3o}] = [p_{1r} \quad p_{2r} \quad p_{3r}] \begin{bmatrix} x_{1r} & x_{2r} & x_{3r} \\ y_{1r} & y_{2r} & y_{3r} \\ z_{1r} & z_{2r} & z_{3r} \end{bmatrix}, \tag{5.3}$$

odnosno kraće kao

$$\mathbf{P}_o = \mathbf{P}_r \cdot \mathbf{R}_{\mathbf{rot}} \tag{5.4}$$

pri čemu $\mathbf{P_r}$ označava koordinate točke u našem rotiranom koordinatnom sustavu, $\mathbf{R_{rot}}$ matricu rotacije, a $\mathbf{P_o}$ koordinate rotirane točke gledano iz originalnog koordinatnog sustava. No mi zapravo želimo napraviti obrnutu transformaciju, odnosno želimo iz koordinata točke u originalnom sustavu doći do koordinata točke u rotiranom sustavu. Ako imamo poznate koordinate točaka gledano iz originalnog koordinatnog sustava $\mathbf{P_o}$ a potrebno je odrediti točke u rotiranom koordinatnom sustavu $\mathbf{P_r}$ matricu $\mathbf{R_{rot}}$ je potrebno invertirati:

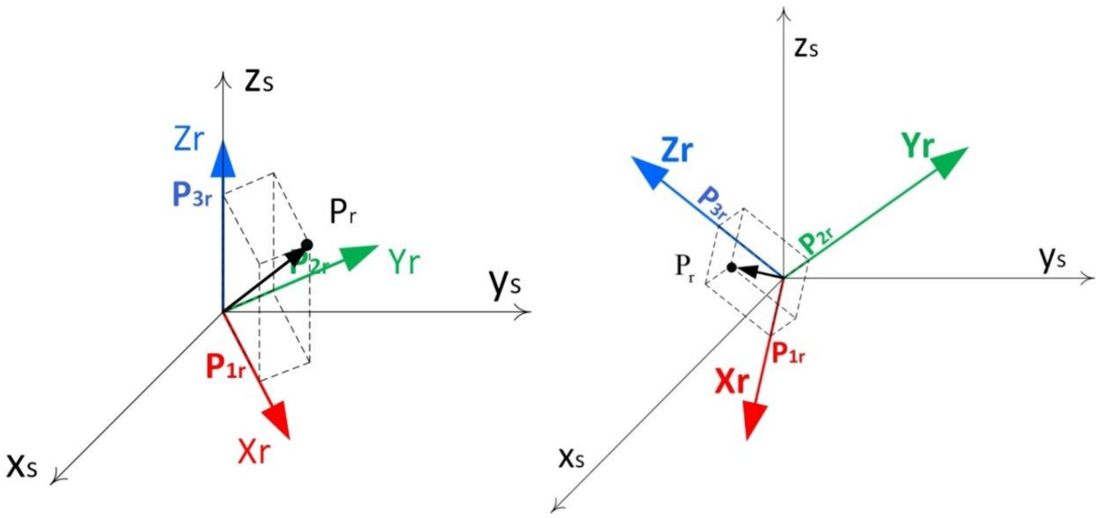
$$\mathbf{P}_r = \mathbf{P}_o \cdot \mathbf{R}_{\mathbf{rot}}^{-1} \tag{5.5}$$

Kako su stupci matrice R ortonormalni vektori, onda vrijedi $\mathbf{R_{rot}}^{-1} = \mathbf{R_{rot}}^{\mathbf{T}}$:

$$\mathbf{P}_r = \mathbf{P}_o \cdot \mathbf{R}_{\mathbf{rot}}^{\mathbf{T}} \tag{5.6}$$

odnosno

$$[p_{1r} \quad p_{2r} \quad p_{3r}] = [p_{1o} \quad p_{2o} \quad p_{3o}] \begin{bmatrix} x_{1r} & y_{1r} & z_{1r} \\ x_{2r} & y_{2r} & z_{2r} \\ x_{3r} & y_{3r} & z_{3r} \end{bmatrix}. \tag{5.7}$$



Slika 5.2. Prikaz vektora koji čine osi rotiranog koordinatnog sustava gledano iz originalnog koordinatnog sustava: (lijevo) jednostavniji primjer rotacije sustava oko z_s-osi, npr. $P_r = (2,\ 1,\ 3)$ rotacija za 45 oko z_s-osi, (desno) proizvoljna rotacija u prostoru.

Vidimo da je matrica rotacije zapravo sačinjena od vektora koji predstavljaju osi rotiranog koordinatnog sustava. Dakle, ako znamo vektore koji predstavljaju osi rotiranog koordinatnog sustava, onda lagano možemo odrediti matricu rotacije. No potrebno je prvo odrediti osi rotiranog koordinatnog sustava.

Određivanje osi rotiranog koordinatnog sustava možemo napraviti korištenjem točaka očišta i gledišta. Prethodno smo definirali da se ishodište koordinatnog sustava oka nalazi u očištu. Os *z* definirat ćemo na način da je ta os usmjerena od očišta prema gledištu. Normirani vektor koji će predstavljati *z*-os u novom koordinatnom sustavu možemo onda izračunati kao

$$z = \mathbf{G} - \mathbf{O} \tag{5.8}$$

$$\hat{z} = \frac{z}{\|z\|} \tag{5.9}$$

Sada je potrebno odrediti i preostale dvije osi. Nažalost, korištenjem samo očišta i gledišta to nije moguće napraviti, jer nam te dvije točke ne daju nikakvu informaciju o tome kako su orijentirane preostale dvije osi. Zbog toga nam je potreban dodatan vektor, pod nazivom *ViewUp* (v_{UP}), koji određuje *y*-os rotiranog koordinatnog sustava. Sada kada imamo dvije osi, treću možemo izračunati preko vektorskog produkta preostalih dviju osi, jer znamo da ta os mora biti okomita na prethodne dvije. Dakle, os *x* ćemo izračunati na način da vektor koji predstavlja *z*-os vektorski pomnožimo s *ViewUp* vektorom:

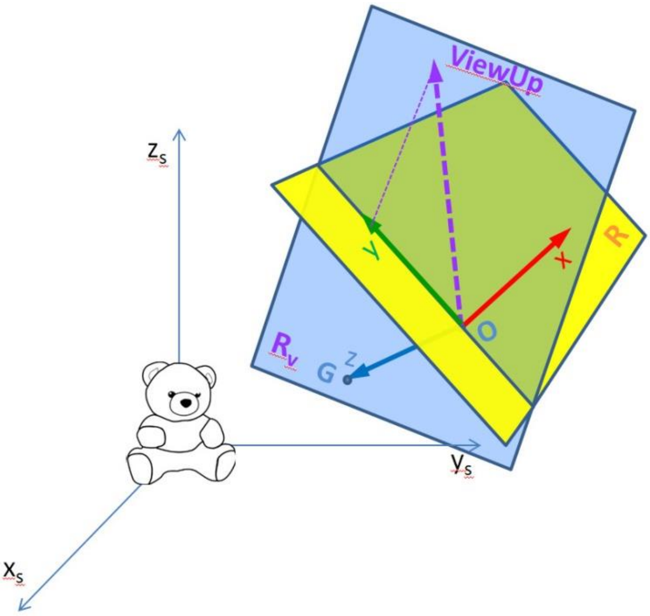
$$\hat{v}_{UP} = \frac{v_{UP}}{\|v_{UP}\|} \tag{5.10}$$

$$\hat{x} = \hat{z} \times \hat{v}_{UP} \tag{5.11}$$

No, teško je za očekivati da će korisnik zadati vektor koji će uistinu biti okomit na vektor koji predstavlja *z*-os (odnosno leži u ravnini **R**). Zbog toga ćemo pretpostaviti da *ViewUp* vektor kojeg korisnik zadaje ne mora nužno biti okomit na vektor koji predstavlja *z*-os, već da on samo definira smjer u kojem gleda *y*-os. To znači da se *ViewUp* vektor ne mora nužno nalaziti u ravnini projekcije. Takva situacija je prikazana i na slici 5.3, na kojoj se vidi kako se vektori *z*, *ViewUp* i *y* nalaze u ravnini $\mathbf{R_V}$, dok se vektori *x* i *y* nalaze u ravnini projekcije *R*. Kako želimo da su sve osi našeg koordinatnog sustava međusobno okomite te da se i *y*-os nalazi u ravnini projekcije zajedno s *x*-osi, potrebno je na temelju zadanog *ViewUp* vektora odrediti *y*-os. Jedan način kako to možemo ostvariti bi bio taj da jednostavno odredimo projekciju *ViewUp* vektora na ravninu projekcije **R**. Računanje projiciranog vektora *ViewUp* na *R* možemo izbjeći korištenjem jednog trika. Naime, kako smo već odredili dvije osi koordinatnog sustava, treću možemo opet izračunati preko njihovog vektorskog produkta i na taj način dobiti os koja je okomita na njih dvije, ali zadržava smjer sličan *ViewUp* vektoru. Dakle, *y*-os se onda izračuna kao:

$$\hat{y} = \hat{x} \times \hat{z} \tag{5.12}$$

Jedino na što moramo pripaziti jest da *ViewUp* vektor kojeg korisnik zada nije kolinearan s vektorom koji predstavlja *z*-os, jer u tom slučaju ne bi mogli odrediti *x* i *y*-osi.



Slika 5.3. Primjer *ViewUp* vektora koji se ne nalazi u ravnini projekcije **R**.
Vektori *x* i *y* su u ravnini **R**, a vektori *ViewUp*, *y* i *z* su u ravnini $\mathbf{R_v}$.

Na taj način izračunali smo sve vektore koji određuju osi novog koordinatnog sustava. Matricu rotacije onda možemo konstruirati pomoću komponenti vektora koji određuju novi koordinatni sustav (u homogenom prostoru):

$$\mathbf{R_{uku}} = \begin{bmatrix} & \mathbf{R_{ot}}^{\mathbf{T}} & & 0 \\ & & & 0 \\ & & & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \hat{x}_{1r} & \hat{y}_{1r} & \hat{z}_{1r} & 0 \\ \hat{x}_{2r} & \hat{y}_{2r} & \hat{z}_{2r} & 0 \\ \hat{x}_{3r} & \hat{y}_{3r} & \hat{z}_{3r} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{5.13}$$

No, ne smijemo zaboraviti pretpostavku s kojom smo krenuli, a to je da je globalni sustav bio zadan kao desni koordinatni sustav, dok je sustav scene zadan kao lijevi koordinatni sustav (slika 5.1). Dakle, nakon što je obavljena rotacija, postignuta su poklapanja svih osi globalnog koordinatnog sustava i sustava scene, osim osi *z* koje iako leže na istom pravcu pokazuju u suprotnim smjerovima, što je posljedica činjenice da su koordinatni sustavi bili različite orijentacije. Zbog toga je nakon rotacije potrebno dodatno napraviti zamjenu predznaka *z*-osi korištenjem matrice transformacije T_Z, koja će jednostavno zrcaliti *z*-os:

$$T_Z = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.14}$$

Ukupnu matricu transformacije onda možemo dobiti umnoškom pojedinih matrica transformacija. Dakle, prvo je potrebno napraviti podudaranje ishodišta oba koordinatna sustava korištenjem matrice T_1, nakon toga je potrebno rotirati koordinatni sustav korištenjem matrice $\mathbf{R_{uku}}$ kako bi se sve osi međusobno poklopile, i konačno korištenjem matrice T_Z potrebno je okrenuti predznak *z*-osi zato što su koordinatni sustavi bili zadani u suprotnim orijentacijama. Znači, ukupna matrica transformacije je onda jednaka:

$$T = T_1 \mathbf{R_{uku}} T_Z. \tag{5.15}$$

Potrebno je napomenuti da OpenGL u pozadini radi veoma slično prethodno opisanom postupku. Jedina razlika je ta što OpenGL pretpostavlja da je i sustav oka desni koordinatni sustav, pa zapravo kod transformacije pogleda ne koristi matricu T_Z za okretanje *z*-osi. No onda tijekom projekcije OpenGL okrene *z*-os, što je zapravo ekvivalentno našoj transformaciji T_Z.

Drugi način određivanja matrice rotacije

Matricu rotacije možemo izračunati i na drugačiji način. Ovaj način temelji se na određivanju niza jednostavnih transformacijskih matrica, odnosno rotacija oko pojedinih osi koordinatnog sustava, te promjenom predznaka na *x*-osi:

T_2 - rotacija za kut α oko *z*-osi, (Slika 5.1 drugi red),

T_3 - rotacija za kut β oko *y*-osi,

T_4 - rotacija za kut 90° oko *z*-osi,

T_5 - promjena predznaka na *x*-osi.

Matrica **T_2** glasi

$$\mathbf{T_2} = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 & 0 \\ \sin\alpha & \cos\alpha & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.16}$$

pri čemu je

$$\sin\alpha = \frac{y_{g1}}{\sqrt{x_{g1}^2 + y_{g1}^2}}, \qquad \cos\alpha = \frac{x_{g1}}{\sqrt{x_{g1}^2 + y_{g1}^2}},$$ gdje su x_{g1}, y_{g1} dobiveni izrazom (5.2)

Djelovanje matrice T_2 na G_1 daje

$$G_2 = G_1 T_2 \ \text{ili} \quad \begin{aligned} x_{g2} &= \sqrt{x_{g1}^2 + y_{g1}^2} \\ y_{g2} &= 0 \\ z_{g2} &= z_{g1} \end{aligned} \quad .$$

Matrica T_3 glasi

$$T_3 = \begin{bmatrix} \cos\beta & 0 & \sin\beta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\beta & 0 & \cos\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{5.17}$$

pri čemu je

$$\sin\beta = \frac{x_{g2}}{\sqrt{x_{g2}^2 + z_{g2}^2}}, \qquad \cos\beta = \frac{z_{g2}}{\sqrt{x_{g2}^2 + z_{g2}^2}} .$$

Djelovanje matrice T_3 na G_2 daje

$$G_3 = G_2 T_3 \ \text{ili} \quad \begin{aligned} x_{g3} &= 0 \\ y_{g3} &= 0 \\ z_{g3} &= \sqrt{x_{g2}^2 + z_{g2}^2} \end{aligned}$$

Nakon ovih transformacija ostvareno je podudaranje *z*-osi koordinatnog sustava oka z_o sa *z*-osi koordinatnog sustava scene z_s. Potrebno je još ostvariti podudaranja *x* i *y*-osi sustava oko sa sustavom scene. Koordinatni sustav na zaslonu obično je postavljen tako da je ishodište u gornjem lijevom uglu zaslona, *x*-os postavljena je horizontalno prema desno, *y*-os vertikalno prema dolje, a *z*-os usmjerena je prema promatraču. Koordinatni sustav oka je prema tome lijevi koordinatni sustav, a sustav scene desni. Potrebne matrice T_4 i T_5 za ostvarivanje podudaranja *x* i *y*-osi sustava oka sa sustavom scene glase

$$T_4 = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad T_5 = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.18}$$

Matrica T_{2-5} je umnožak

$$T_{2-5} = T_2 T_3 T_4 T_5 \tag{5.19}$$

Odnosno, ukupna matrica transformacije T_{1-5} je jednaka:

$$T_{2-5} = T_1 T_{2-5} \tag{5.20}$$

5.2. Perspektivna projekcija

Sada kad smo napravili transformaciju pogleda, potrebno je još napraviti perspektivnu projekciju jer točke iz trodimenzijskog sustava oka moramo prebaciti u dvodimenzijski sustav scene. Zadaća je odrediti matricu P koja će po zakonu perspektive projicirati točke iz sustava oka u ravninu projekcije, slika 5.1, odnosno u sustavu prikaza,

$$A_P = A_0 P\,. \tag{5.21}$$

Udaljenost ravnine projekcije od očišta je

$$H = \sqrt{\left(x_o - x_g\right)^2 + \left(y_o - y_g\right)^2 + \left(z_o - z_g\right)^2} = z_{g3}\,. \tag{5.22}$$

Iz sličnosti trokuta slijedi

$$x_p = \frac{x_0}{z_0} H, \quad y_p = \frac{y_0}{z_0} H\;. \tag{5.23}$$

Izraz 5.19 napisan u matričnom obliku glasi

$$A_p{}' = A_0 P$$

ili po koordinatama

$$\begin{bmatrix} x_p{}' & y_p{}' & z_p{}' & h_p{}' \end{bmatrix} = \begin{bmatrix} x_0 & y_0 & z_0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{H} \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\begin{aligned} x_p' &= x_0 \\ y_p' &= y_0 \\ z_p' &= 0 \\ h_p' &= \frac{z_0}{H} \end{aligned} \tag{5.24}$$

Iz 5.23 slijedi 5.22, što je povratak u nehomogeni prostor tj.

$$x_p = \frac{x_p'}{h_p'} = \frac{x_0}{z_0}H, \qquad y_p = \frac{y_p'}{h_p'} = \frac{y_0}{z_0}H.$$

5.3. Radni zadatak

Zadati poligon te načiniti transformaciju pogleda i perspektivnu projekciju.

1. Iz datoteke učitati koordinate očišta, gledišta i vrhova poligona u sustavu scene.

Gledište se obično zadaje u ishodištu scene G = (0 0 0) ili u središtu tijela (poligona). Očište je točka iz koje gledamo i ovisit će o veličini objekta. Ako su koordinate objekta u rasponu (-1, 1) očište može biti npr. O = (1 1 3). Moramo paziti da očište ne zadamo u unutrašnjosti objekta. Ako se rotacijska matrica računa korištenjem prvog opisanog postupka, onda je potrebno dodatno zadati i *ViewUp* vektor.

2. Odrediti matricu transformacije pogleda T po formuli 5.7 ili 5.20.
3. Odrediti matricu perspektivne projekcije P.
4. Načiniti transformaciju i projekciju zadanih vrhova poligona.
5. Iscrtati poligon.
6. Ponoviti korake 1-5. za tijelo iz prethodne vježbe. Potrebno je omogućiti promjenu pozicija očišta i gledišta korištenjem tipkovnice ili miša. Primjerice, na pritisak određene tipke uvećati odnosno umanjiti željenu koordinatu.