

## Primjena matrica u praksi 2: Geometrijske transformacije

TIHANA STRMEČKI<sup>1</sup>, GORAN SIROVATKA<sup>2</sup>, RENATO RAK<sup>3</sup>, LUKA PAJKOVIĆ<sup>4</sup>

U prvom smo dijelu članka detaljnije proučili praktičnu primjenu matrica pri digitalnoj obradi fotografije. U nastavku ćemo razmotriti njihovu korisnost u području računalne grafike koja se bavi transformacijom geometrijskih objekata. Transformacije su operacije nad objektima u grafičkom prostoru. Spomenut ćemo četiri osnovne vrste: translaciju, skaliranje, rotaciju i refleksiju. Transformacija smicanja (*engl. shearing*) rijetko se koristi, stoga je u ovom članku nećemo detaljnije pojašnjavati. Primjena transformacija najznačajnija je u računalnoj animaciji, posebno računalnim igrama.

Transformacije u grafičkom prostoru digitalno se zapisuju matricama na način da geometrijske podatke pretvaraju u digitalne i lako izračunjive informacije. Za to će nam biti potrebno osnovno znanje o zbrajanju i množenju matrica. Zbrajanje matrica obradili smo u prvom dijelu članka. Za razliku od zbrajanja, množenje matrica složenija je operacija kojoj prethodi uvjet *ulančanosti*. Matrice su ulančane ako je broj stupaca prve matrice u množenju jednak broju redaka druge matrice kojom se množi. Rezultat množenja bit će matrica koja ima broj redaka koliko i prva matrica, a broj stupaca koliko i druga. Matrica  $C = [c_{ij}]$ , umnožak matrice  $A = [a_{ij}]_{m \times n}$  i  $B = [b_{ij}]_{n \times q}$  ima redaka koliko i matrica  $A$  te stupaca koliko i matrica  $B$ , tj. vrijedi  $C \in M_{m \times q}$ . Elementi  $c_{ij}$  računaju se tako da za sve parove indeksa  $i = 1, \dots, m$  i  $j = 1, \dots, q$  vrijedi:

$$c_{ij} = \sum_{k=1}^n a_{ik} b_{kj}.$$

Iz gornje se formule lako vidi da se svaki element  $c_{ij}$  matrice  $C = AB$  dobije kao zbroj umnožaka elemenata  $i$ -tog reda prve matrice,  $A$ , s odgovarajućim elementima  $j$ -tog stupca druge matrice,  $B$ . Pritom zbog uvjeta ulančanosti redak i stupac imaju jednak broj elemenata, što nam i omogućava ovakav račun. Na donjem primjeru vidi se algoritam izračuna množenja dviju matrica:

<sup>1</sup>Tihana Strmečki, Tehničko veleučilište u Zagrebu

<sup>2</sup>Goran Sirovatka, Tehničko veleučilište u Zagrebu

<sup>3</sup>Renato Rak, student Tehničkog veleučilišta u Zagrebu

<sup>4</sup>Luka Pajković, student Tehničkog veleučilišta u Zagrebu

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \cdot \begin{bmatrix} 9 & 8 \\ 7 & 6 \\ 5 & 4 \end{bmatrix} = \begin{bmatrix} 1 \cdot 9 + 2 \cdot 7 + 3 \cdot 5 & 1 \cdot 8 + 2 \cdot 6 + 3 \cdot 4 \\ 4 \cdot 9 + 5 \cdot 7 + 6 \cdot 5 & 4 \cdot 8 + 5 \cdot 6 + 6 \cdot 4 \end{bmatrix} = \begin{bmatrix} 38 & 32 \\ 101 & 86 \end{bmatrix}.$$

Prva transformacija koju ćemo pobliže upoznati jest *translacija*, postupak kojim se objekt pomiče, odnosno translira u ravnini ili prostoru. Ako želimo translirati neki dvodimenzionalni ili trodimenzionalni objekt, moramo ga prikazati koristeći matrice. Najčešće se prvo identificiraju vrhovi i bridovi objekta te se potom translacija radi nad tim točkama i pravcima. Krenimo od najjednostavnijeg slučaja: translacije točke u ravnini ili prostoru. Točku možemo prikazati pomoću matrice koja sadrži redaka koliko je i koordinata točke i samo jedan stupac. Takvu matricu nazivamo *jednostupčanom matricom* te u nju upisujemo koordinate zadane točke. U dvije dimenzije koristimo oznake koordinata apscise i ordinate, dok se u tri dimenzije dodaje i koordinata aplikate:

$$t_{2D} = (x, y) = \begin{bmatrix} x \\ y \end{bmatrix}, \quad t_{3D} = (x, y, z) = \begin{bmatrix} x \\ y \\ z \end{bmatrix}.$$

Pretpostavit ćemo da je čitatelj upoznat s pojmom vektora, bez daljnjeg navođenja detalja o vektorima i njihovim svojstvima. Translacijom točke dobivamo drugu točku, koje spojene čine vektor u smjeru translirane točke. Tim vektorom  $\vec{v}$  određujemo duljinu translacije, prikazujući ga u odgovarajućem koordinatnom sustavu preko rastava na pomak u smjeru apscise  $v_x$ , ordinate  $v_y$  te aplikate  $v_z$ :

$$\vec{v}_{transl.2D} = (v_x, v_y) = \begin{bmatrix} v_x \\ v_y \end{bmatrix}, \quad \vec{v}_{transl.3D} = (v_x, v_y, v_z) = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix}.$$

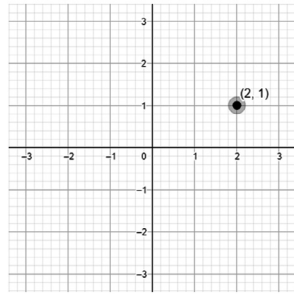
Koordinate translirane točke  $t'_{2D} = (x', y')$  ili  $t'_{3D} = (x', y', z')$  dobivamo zbrajanjem dviju matrica istog tipa:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} v_x \\ v_y \end{bmatrix} = \begin{bmatrix} x + v_x \\ y + v_y \end{bmatrix}, \quad \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} = \begin{bmatrix} x + v_x \\ y + v_y \\ z + v_z \end{bmatrix}.$$

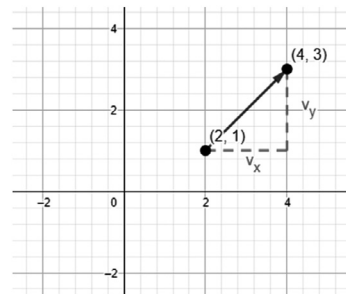
Očito je da translirana točka ima pomak u smjeru osi  $x$ , koliki je i pomak vektora translacije u smjeru te osi, te analogno za ostale osi. Pogledajmo primjer translacije točke (2,1) u ravnini za vektor  $\vec{v}_{transl.2D} = (2,2)$  (Slika 1. i Slika 2.).

Pogledajmo i primjer translacije točke  $t_{3D} = (2.75, 0.88, 1.19)$  za vektor pomaka

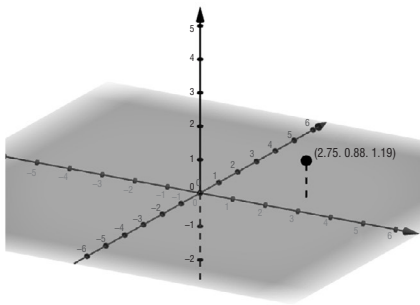
$$\vec{v}_{transl.3D} = (1.07, -5.26, 3.17) \text{ (Slika 3. i Slika 4.)}.$$



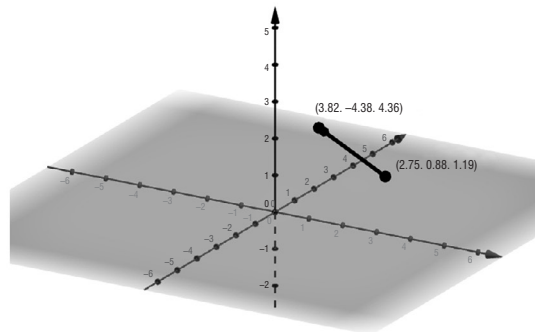
Slika 1. Točka u ravnini



Slika 2. Translacija točke u ravnini



Slika 3. Točka u prostoru



Slika 4. Translacija točke u prostoru

Kako bi se svi dijelovi nekog objekta brzo i jednostavno translatali, zanima nas kako izgleda matrica translacije koja se može kombinirati s drugim transformacijama. U praksi se najčešće primjenjuje nekoliko transformacija istovremeno pa ćemo ubrzo vidjeti da se primjena ostalih transformacija svodi na množenje matrica. S obzirom na to da se translacija objekta svodi na zbrajanje matrica, kako bi uzastopna primjena transformacija bila moguća i efektivna u digitalnom smislu, pokazalo se korisnim uvesti *homogene koordinate* točaka. Pomoću njih se translacija primjenjuje množenjem umjesto zbrajanjem. Homogene koordinate često se koriste u računalnoj grafici na način da svaku točku u ravnini  $(x,y)$  proširujemo realnom varijablom  $w$  do njezinih homogenih koordinata  $(x,y,w)$ . Iako sada imamo tri koordinate, i dalje se radi o točki u ravnini. Bez da ulazimo u puno detalja o homogenim koordinatama, koristit ćemo njihov standardni oblik, onaj za koji vrijedi  $w = 1$ . Homogene koordinate točke u ravnini tada su dane s  $(x,y,1)$  te u prostoru s  $(x,y,z,1)$ .

Sada imamo sve potrebne alate da bismo naveli matricu translacije koja je u ravnini tipa  $M_{3 \times 3}$ , a u prostoru  $M_{4 \times 4}$ :

$$M_{trans.2D} = \begin{bmatrix} 1 & 0 & v_x \\ 0 & 1 & v_y \\ 0 & 0 & 1 \end{bmatrix} \quad M_{trans.3D} = \begin{bmatrix} 1 & 0 & 0 & v_x \\ 0 & 1 & 0 & v_y \\ 0 & 0 & 1 & v_z \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Točku u ravnini zapisujemo kao jednostupčanu matricu preko homogenih koor-

dinata  $\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$ , a translativanu točku kao  $\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}$ .

Sada se translacija točke standardizirano može prikazati operacijom množenja matrice translacije s matricom homogenih koordinata točke za  $w=1$ :

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & v_x \\ 0 & 1 & v_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x+v_x \\ y+v_y \\ 1 \end{bmatrix} \quad \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & v_x \\ 0 & 1 & 0 & v_y \\ 0 & 0 & 1 & v_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x+v_x \\ y+v_y \\ z+v_z \\ 1 \end{bmatrix}.$$

Dakle, umjesto korištenja vektora translacije, matricu translacije množimo matricom koja sadrži homogene koordinate točke koju translativamo, čime direktno dobivamo homogene koordinate translativane točke u matrici stupcu s desne strane. Uočavamo da su elementi rezultatne matrice jednaki kao kod zbrajanja matrica.

Sljedeća transformacija koju ćemo proučiti jest *skaliranje*. Upotrebom ove tehnike manipuliramo objektom na način da on postaje veći ili manji, ovisno o proizvoljnoj vrijednosti faktora  $k$ . Pokazat ćemo skaliranje na primjeru mnogokuta u ravnini i geometrijskog tijela u prostoru. Kako bismo skalirali, vrijednosti koordinata točaka koje predstavljaju objekt množimo racionalnim faktorom  $k$ . Ako je faktor jednak za svaku koordinatu ( $k_x = k_y = k_z$ ), rezultat skaliranja bit će proporcionalan. Za točku u prostoru  $(x,y,z)$  dobivamo skaliranu točku  $(x',y',z')$  na sljedeći način:

$$x' = k_x \cdot x \quad y' = k_y \cdot y \quad z' = k_z \cdot z.$$

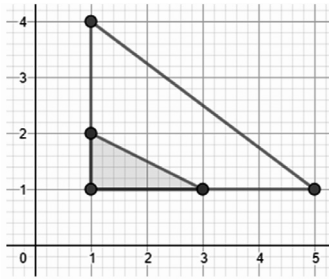
Matrice skaliranja u kojima također koristimo homogene koordinate u ravnini i prostoru su sljedeće:

$$M_{skal.2D} = \begin{bmatrix} k_x & 0 & 0 \\ 0 & k_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad M_{skal.3D} = \begin{bmatrix} k_x & 0 & 0 & 0 \\ 0 & k_y & 0 & 0 \\ 0 & 0 & k_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

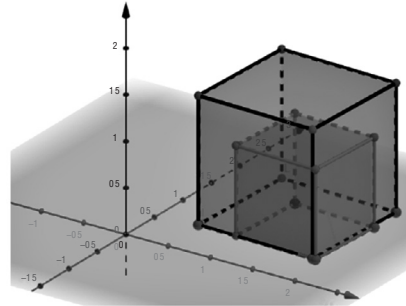
Kako bismo skalirali neki objekt, svaku točku  $(x,y,1)$  ili  $(x,y,z,1)$  koja predstavlja neki njegov dio potrebno je prikazati matricno te potom matricu skaliranja pomnožiti njome:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} k_x & 0 & 0 \\ 0 & k_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} k_x \cdot x \\ k_y \cdot y \\ 1 \end{bmatrix} \quad \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} k_x & 0 & 0 & 0 \\ 0 & k_y & 0 & 0 \\ 0 & 0 & k_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} k_x \cdot x \\ k_y \cdot y \\ k_z \cdot z \\ 1 \end{bmatrix}.$$

Pogledajmo na primjeru skaliranje trokuta i kocke. Originalni veći trokut je proziran, a skalirani manji lagano osjenčan, dok je originalna manja kocka tamnije osjenčana, a skalirana veća svjetlije osjenčana<sup>5</sup>.



Slika 5. Skaliranje trokuta



Slika 6. Skaliranje kocke

*Rotacija* je još jedna transformacija koja se često koristi. Najjednostavnije su rotacije oko točke ili oko brida. Njima se postiže rotacija cjelokupnog objekta s obzirom na određeni kut u odnosu na početnu poziciju. Za rotaciju je korisno koristiti *polarne koordinate* kojima se navodi udaljenost točke od ishodišta i otklon polupravca kroz ishodište i točku od osi apscise. Polarne koordinate točke u ravnini  $(x,y)$  zapisujemo kao  $x = r \cos \alpha$  i  $y = r \sin \alpha$ . Konačnu poziciju točke  $(x',y')$  dobivenu rotacijom za kut  $\beta$  računamo na sljedeći način:

$$\begin{aligned}x' &= r \cos(\alpha + \beta) = r \cos \alpha \cos \beta - r \sin \alpha \sin \beta = x \cos \beta - y \sin \beta \\y' &= r \sin(\alpha + \beta) = r \cos \alpha \sin \beta + r \sin \alpha \cos \beta = x \sin \beta + y \cos \beta.\end{aligned}$$

Rotacija u ravnini zapisana preko matrice transformacije s homogenim koordinatama izgleda ovako:

$$M_{rotac.3D} = \begin{bmatrix} \cos \beta & -\sin \beta & 0 \\ \sin \beta & \cos \beta & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Za objekte u prostoru navodimo matrice rotacije s obzirom na os oko koje rotiramo:

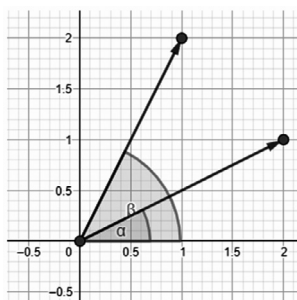
$$M_{x,rotac.3D} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \beta & -\sin \beta & 0 \\ 0 & \sin \beta & \cos \beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad M_{y,rotac.3D} = \begin{bmatrix} \cos \beta & 0 & -\sin \beta & 0 \\ 0 & 1 & 0 & 0 \\ \sin \beta & 0 & \cos \beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

<sup>5</sup>U časopisu su slike objavljene u crno – bijeloj varijanti. Da bi se dobilo na zornosti, poželjno je pogledati i slike u boji. Sve slike iz članka u boji dostupne su na poveznici:

<https://drive.google.com/drive/folders/18Z6C1M0SYZ8XjuOnGLBV6AFb8WFFhsqOY>

$$M_{z,rotac.3D} = \begin{bmatrix} \cos \beta & -\sin \beta & 0 & 0 \\ \sin \beta & \cos \beta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Slijedi primjer rotacije vektora  $\vec{v}_{rotac.2D} = (2,1)$  za  $\beta \approx 37^\circ$ :



Slika 7. Rotacija vektora u ravnini

*Refleksija* je postupak zrcaljenja točke, geometrijskog lika ili objekta u odnosu na određenu točku, pravac, koordinatnu os ili proizvoljnu ravninu. Stoga je matrice refleksije najjednostavnije zapisati u odnosu na točke, pravce ili ravnine oko kojih objekt reflektiramo.

Objasnit ćemo refleksiju na primjeru mnogokuta iz prvog kvadranta s donje slike. Prva refleksija preslikat će mnogokut u odnosu na os, što nam daje rezultat

u drugom kvadrantu. Matrica te refleksije jest:  $M_{reflek\_1} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ . Druga re-

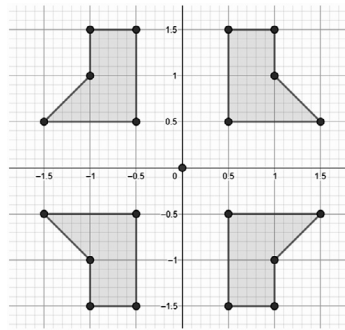
fleksija reflektira mnogokut u odnosu na os  $x$  i rezultat je u četvrtom kvadrantu:

$M_{reflek\_2} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ . Treća transformacija je refleksija u odnosu i na os  $x$  i na os

$y$ , odnosno, s obzirom na ishodište koordinatnog sustava, što daje rezultat u trećem

kvadrantu:  $M_{reflek\_3} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ . Na slici su istovremeno prikazane sve tri tran-

sformacije.



Slika 8. Refleksije mnogokuta

Konačni rezultat transformiranja nekog lika ili tijela može se prikazati uzastopnim množenjem transformacijskih matrica. Treba imati na umu da množenje matrica općenito nije komutativno (za neke umnoške matrica vrijedi  $A \cdot B \neq B \cdot A$ ). Lako je zamisliti da primjenom skaliranja te potom rotacije ne dobivamo isti objekt kao obrnutim redoslijedom – primjenom rotacije i zatim skaliranja.

Za točku u ravnini ili prostoru nad kojom je potrebno izvršiti uzastopne proizvoljne transformacije  $M_1, M_2, M_3, \dots, M_n$ , homogene koordinate transformirane točke dobivamo na sljedeći način:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = M_1 \cdot M_2 \cdot M_3 \cdots M_n \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = M_1 \cdot M_2 \cdot M_3 \cdots M_n \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}.$$

Ako želimo objekt vratiti u stanje prije transformacije, dvije su nam opcije na raspolaganju. Prva je opcija izračunati inverznu matricu određene transformacije. Iako je ta opcija matematički intuitivnija, programski troši više memorije (koristi se dodatna inverzna funkcija), stoga je spretnije koristiti drugu opciju – transformiranje putem suprotnih vrijednosti. Kao primjer uzmimo matricu rotacije čija je varijabla rotiranja kut  $\beta$ . Nakon primjene te rotacije na neki objekt, primjenom iste matrice, ali upotrebom suprotne vrijednosti  $-\beta$ , objekt se vraća natrag u početni položaj.

Pogledajmo primjer rotacije točke  $(x, y)$  u ravnini za kut  $\beta$ . Matricu rotacije pomnožimo matricom homogenih koordinata točke, čime dobivamo homogene koordinate rotirane točke:

$$\begin{bmatrix} \cos \beta & -\sin \beta & 0 \\ \sin \beta & \cos \beta & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}.$$

Matricu rotacije s vrijednosti  $-\beta$  pomnožimo rotiranom točkom, što je vizualno rotacija u suprotnom smjeru. Primijenit ćemo trigonometrijska svojstva funkcija sinus i kosinus te time dobivamo sljedeću matricu:

$$\begin{bmatrix} \cos(-\beta) & -\sin(-\beta) & 0 \\ \sin(-\beta) & \cos(-\beta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \beta & \sin \beta & 0 \\ -\sin \beta & \cos \beta & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} x' \cos \beta + y' \sin \beta \\ -x' \sin \beta + y' \cos \beta \\ 1 \end{bmatrix}.$$

Kod rotacije se koriste polarne koordinate i prethodno smo pokazali da vrijedi  $x' = x \cos \beta - y \sin \beta$  i  $y' = x \sin \beta + y \cos \beta$ . Uvrštavamo te jednadžbe u dobivenu matricu, primijenimo osnovni trigonometrijski identitet  $\cos^2 \beta + \sin^2 \beta = 1$  i time zaključno očekivano dobivamo koordinate polazne točke:

$$\begin{aligned} \begin{bmatrix} x' \cos \beta + y' \sin \beta \\ -x' \sin \beta + y' \cos \beta \\ 1 \end{bmatrix} &= \begin{bmatrix} (x \cos \beta - y \sin \beta) \cos \beta + (x \sin \beta + y \cos \beta) \sin \beta \\ -(x \cos \beta - y \sin \beta) \sin \beta + (x \sin \beta + y \cos \beta) \cos \beta \\ 1 \end{bmatrix} \\ &= \begin{bmatrix} x \cos^2 \beta - y \sin \beta \cos \beta + x \sin^2 \beta + y \cos \beta \sin \beta \\ -x \cos \beta \sin \beta + y \sin^2 \beta + x \sin \beta \cos \beta + y \cos^2 \beta \\ 1 \end{bmatrix} \\ &= \begin{bmatrix} x (\cos^2 \beta + \sin^2 \beta) \\ y (\cos^2 \beta + \sin^2 \beta) \\ 1 \end{bmatrix}. \end{aligned}$$

Ova nam spoznaja omogućava i sljedeći zapis:

$$\left[ M_{rotac.2D}(\beta) \right]^{-1} = M_{rotac.2D}(-\beta).$$

Ostale inverzije transformacija također slijede pravilo primjene suprotnih vrijednosti.

Za prikaz matričnih transformacija u MATLAB-u dovoljna je osnovna verzija MATLAB-a. On je izrazito praktičan jer je dizajniran na način da sve varijable sprema kao matrice, tako da nije potrebno matrice pisati ručno, već ih program sam nadopunjava. Za transformaciju grafičkih objekata koriste se ugrađene MATLAB funkcije *hgtransform* i *makehgtform*.

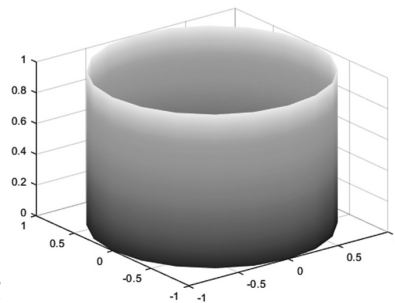
Definirajmo valjak koji će nam poslužiti kao tijelo nad kojim vršimo transformacije.



```
[x,y,z]=cylinder(1); %valjak s radijusom=1

%naredba surf iscrtava 3D prostor,
%naredba hgtransform daje cilindru svojstvo objekta
%nad kojim će se vršiti transformacije
surf(x,y,z,'Parent',hgtransform);

%naredbe za ljepši grafički prikaz
grid on
view(3)
shading interp
```

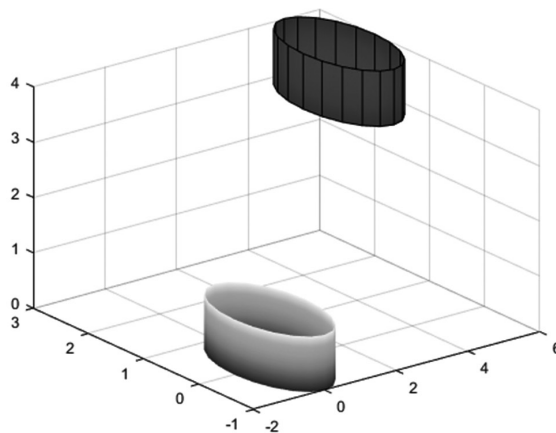


Slika 9. Valjak radijusa 1

Slijedi primjer translacije valjka za vektor  $\vec{v}_{transl.3D} = (5, 2, 3)$ .

```
hold on %zadržavamo postojeći cilindar u prikazu

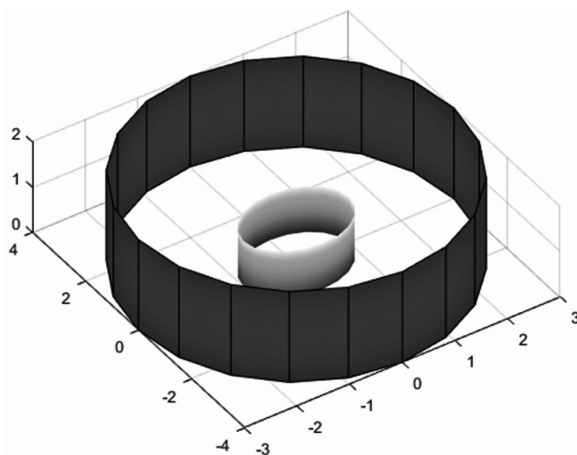
%definicija varijable u kojoj spremamo transformacijsku funkciju,
%vršimo translaciju za 5 jedinica na x-osi, 2 na y-osi i 3 na z-osi
ttrans=hgtransform('Parent',hgtransform,'Matrix',makehgtform('translate',[5 2 3]));
s2=surf(x,y,z,'Parent',ttrans);
```



Slika 10. Bazni i translirani valjak

Pogledajmo skaliranje valjka s faktorima (3,4,2) za svaku koordinatu.

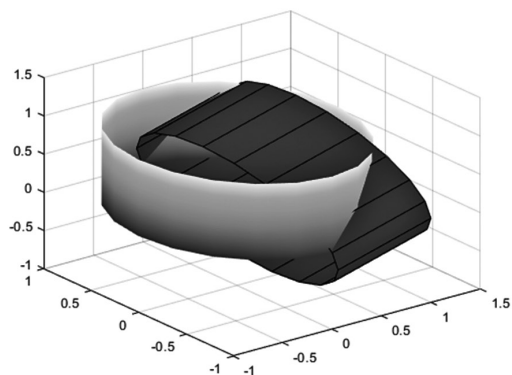
```
hold on
t_skal=hgtransform('Parent',hgtransform,'Matrix',makehgtform('scale',[3 4 2]));
s3=surf(x,y,z,'Parent',t_skal);
```



Slika 11. Bazni i skalirani valjak

Za rotaciju valjka potrebno je u MATLAB-u zadati raspon vrijednosti u radijanima koje će funkcija transformacije koristiti pri računanju. Slijedi primjer rotacije oko  $y$ -osi (korišten je skalirani valjak iz prošlog primjera radi jasnijeg prikaza).

```
for raspon = 1:0.5:2.4*pi
    t_roty=makehgtform('yrotate',raspon);
    Ry=set(t_skal,'Matrix',t_roty);
    drawnow
end
```



Slika 12. Rotacija skaliranog valjka oko osi  $y$

Matrica refleksije nije ugrađena u MATLAB-u, stoga kod prelazi opseg ovog članka.

U ovom smo se dvodijelnom članku detaljnije upoznali s primjenom matrica u računalnoj grafici, preciznije u digitalnoj obradi fotografije i primjeni geometrijskih transformacija na grafičkim objektima. Matrice na intuitivan način zapisuju objekte kojima se digitalno manipulira i s obzirom na njihova svojstva i na operacije nad njima predstavljaju važan dio matematičkih alata u primjeni. Program MATLAB omogućava laganu primjenu znanja o matricama i nudi širok spektar funkcija koje korisniku olakšavaju rad i vizualizaciju.

### Literatura:

1. Strmečki T.: *Matematika 1*, Tehničko veleučilište u Zagrebu, Zagreb, 2021.
2. Kovačić B., Marohnić L., Strmečki T.: *Repetitorij matematike za studente elektrotehnike, informatike i računarstva*, Tehničko veleučilište u Zagrebu, Zagreb, 2016.
3. Bakić D.: *Linearna algebra*, Školska knjiga, Zagreb, 2008.
4. Gržina H.: *Utvrđivanje izvornika analognih i digitalnih fotografija (doktorski rad)*, Filozofski fakultet u Zagrebu, Zagreb, 2017.
5. Bruno J., Jan V., Slaby A.: *Computer Graphics in Computer Science Education. Problems of Education in the 21st Century*, 11, 60-68, 2009.
6. Vince J.: *Mathematics for Computer Graphics, 3rd Edition*, Springer-Verlag London Limited, 2010.
7. Džaja B.: *Digitalna obrada i analiza slike (nastavni materijal)*, Sveučilište u Splitu, Split, 2017.
8. Novak Stanko E.: *Digitalna obrada slike: usporedba programa za obradu slike (diplomski rad)*, Filozofski fakultet u Zagrebu, Zagreb, 2018.
9. Attaway S.: *MATLAB: A Practical Introduction to Programming and Problem Solving (Fifth Edition)*, Elsevier Inc., Boston, 2019.
10. Farid H.: *The Fundamentals of Image Processing*, Berkley University, Berkely, 2010.
11. Lončarić N., Keček D., Kraljić M.: *Matrices in computer Graphics*, Tehnički glasnik Vol 12, 2018.