Geometrie computațională

Laborator 3

1. Transformări 2D

Dacă x şi y sunt coordonatele unui punct M, ecuaţiile:

$x' = f(x, y)$, $y' = g(x, y)$

permit trecerea de la punctul $M(x, y)$ la punctul $M'(x', y')$. Aceste ecuaţii definesc o transformare plană. Punctul M' este transformatul lui M. Putem considera coordonatele unui punct ca o matrice 2×1 (un vector) ^{T}**(x,y)** (^{T} semnifică "transpus"). Fie produsul matriceal următor

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} Ax + By \\ Cx + Dy \end{bmatrix}$$

Toate punctele planului xOy înmulţite cu o matrice 2×2 vor da noi puncte (x', y') conform acestor relaţii.

Există însă tipuri de transformări care nu pot fi reprezentate printr-o matrice 2×2. Pentru rezolvarea situaţiei se va introduce o a treia componentă a vectorilor (x, y) şi $(x'\ y')$, obţinând $(x\ y\ 1)$, $(x'\ y'\ 1)$. Matricea transformărilor va fi obligatoriu de dimensiune 3x3.

Spunem că reprezentarea poziţiei unui punct în plan printr-un vector cu trei componente este o reprezentare în ***coordonate omogene***.

Matricile de dimensiuni 3×3 dau posibilitatea efectuării transformărilor inverse dând familiilor de transformări caracteristica de grup algebric.

Translaţia

Translaţia reprezintă deplasarea unei figuri geometrice paralel cu un vector dat **v=(a,b)**. Ea este reprezentată de ecuaţiile:

$\boldsymbol{x' = x + a,\ y' = y + b.}$

Astfel, matricea translaţiei este

$$\begin{bmatrix} 1 & 0 & a \\ 0 & 1 & b \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x+a \\ y+b \\ 1 \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}.$$

Scalarea

Scalarea cu factorii S_x, S_y pe axele OX, OY se defineşte ca transformarea care modifică o imagine prin redimensionarea ei cu factorul de multiplicitate S_x pe orizontală şi S_y pe verticală. Ecuaţiile carteziene ale scalării sunt

$$\begin{cases} X^* = S_x \cdot X \\ Y^* = S_y \cdot Y \end{cases}$$

Astfel, matricea scalării este

$$S = \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Simetria

Simetriile sunt cazuri particulare de scalări cu factori unitari negativi. Există două tipuri de simetrii: faţă de o dreaptă sau faţă de un punct. Cele mai simple simetrii sunt cele faţă de axe sau origine.

- Simetria în raport cu axa OX:

$$S^X = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

- Simetria în raport cu axa OY:

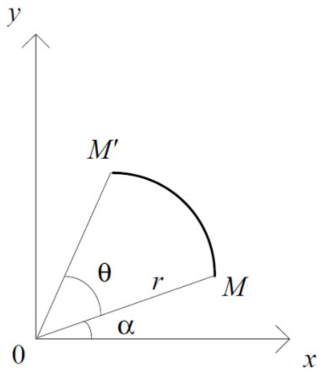
$$S^Y = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

- Simetria în raport cu originea O:

$$S^O = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Rotaţia

Este considerată transformare elementară rotaţia în jurul originii, în sens trigonometric, cu un un unghi oarecare, θ.



Transformarea rotaţiei este

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

Aplicaţie:

Realizaţi o rotaţie (în sens trigonometric) a unui triunghi.

```
#include "graphics.h"
#include <math.h>
#include <stdlib.h>
#include <dos.h>
#include <windows.h>
#include <iostream>
using namespace std;
#define pi 3.14159265359
int gd, gm;
int n, i, j;
double r, x, y, xp, yp, fi;

float x_1, x_2, y_1, y_2, alfa = 3.1415 / 8; int x_max, y_max;//coord ecran
float theta = pi / 36.f; //unghiul de rotatie, in radiani
float xa, ya, xb, yb, xc, yc;//coord. Triunghiului

int xe(float x)// normalizarea coordonatei x
{
	return((int)floor((x - x_1) / (x_2 - x_1)*x_max));
}
int ye(float y)// normalizarea coordonatei y
{
	return((int)floor((y_2 - y) / (y_2 - y_1)*y_max));
}

void axe2D()
{
	setcolor(0);
	outtextxy(xe(x_2) - 20, ye(0) - 20, "x");
	outtextxy(xe(x_2) - 18, ye(0) - 7, ">");
	outtextxy(xe(0) - 15, ye(y_2) + 15, "y");
	outtextxy(xe(0) - 15, ye(0) - 15, "O");
	outtextxy(xe(0) - 1, ye(y_2) , "^");
	line(xe(x_1), ye(0), xe(x_2), ye(0));
	line(xe(0), ye(y_1), xe(0), ye(y_2));
}

void linie2D(float xa, float ya, float xb, float yb)
{
	line(xe(xa), ye(ya), xe(xb), ye(yb));
}

void deseneazatriunghi()   //void grafic()
{
	outtextxy(xe(xa) + 10, ye(ya) - 10, "A");
	circle(xe(xa), ye(ya), 2);
	outtextxy(xe(xb) + 10, ye(yb) - 10, "B");
	circle(xe(xb), ye(yb), 2);
	outtextxy(xe(xc) - 10, ye(yc) + 10, "C");
	circle(xe(xc), ye(yc), 2);
	linie2D(xa, ya, xb, yb);
	linie2D(xb, yb, xc, yc);
	linie2D(xa, ya, xc, yc);
}

void rotatie2D()
{
	xa = xa * cos(theta) - ya * sin(theta); ya = xa * sin(theta) + ya * cos(theta);
	xb = xb * cos(theta) - yb * sin(theta); yb = xb * sin(theta) + yb * cos(theta);
	xc = xc * cos(theta) - yc * sin(theta); yc = xc * sin(theta) + yc * cos(theta);
```

```
}

int main()
{
	printf("Limitele domeniului orizontal:\n");
	printf("x_1="); scanf("%f", &x_1); //x_1<0<x_2
	printf("x_2="); scanf("%f", &x_2);
	printf("Limitele domeniului vertical:\n");
	printf("y_1="); scanf("%f", &y_1); //y_1<0<y_2
	printf("y_2="); scanf("%f", &y_2);

	initwindow(800, 600, "AXE", 200, 200);
	x_max = getmaxx(); //nr. maxim de pixeli pe coord. x
	y_max = getmaxy(); //nr. maxim de pixeli pe coord. y
	float sum_t = 0.f;
	xa = 7; ya = 10;//initializare coord triunghi
	xb = 10; yb = 4;
	xc = 1; yb = 1;
	do
	{
		setbkcolor(15);
		cleardevice();
		axe2D();
		deseneazatriunghi();
		delay(100);
		rotatie2D();
		sum_t = sum_t + theta;
	} while (sum_t <= 2 * pi);
	getchar(); getchar();
	closegraph();
	return 0;
}
```

2. Transformări 3D

În cadrul acestui capitol, vom considera 5 tipuri de transformări ale obiectelor 3D. Primele două, translaţia şi scalarea sunt simple extensii ale transformărilor 2D în trei dimensiuni. Celelalte 3 transformări sunt rotaţii în spaţiu, câte una pentru fiecare axă de coordonate.

Un obiect care este format din puncte conectate prin segmente de dreaptă (având o suprafaţă poligonală) poate fi transformat prin transformarea succesivă a capetelor segmentelor. Aceste puncte transformate vor fi apoi conectate între ele prin segmente de dreaptă. Proprietatea care permite această abordare este liniaritatea transformărilor 3D considerate.

Translaţia

Translaţia transformă punctul P(x,y,z) în punctul P'(x',y',z') după formulele

$x' = x + tx$

$y' = y + ty$

$z' = z + tz$

unde *tx* , *ty* , *tz* – deplasarea în direcţiile *x, y* şi *z*.

Matricea transformării este:

$$M = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ t_x & t_y & t_z & 1 \end{bmatrix}$$

Scalarea

Scalarea unui punct în spaţiu are ca efect relocarea sa în relaţie cu un punct de referinţă. Aplicată asupra unui obiect în spaţiu, ea produce mărirea sau micşorarea obiectului în direcţiile *x, y, z*. Factorii de scalare pe cele 3 axe sunt notaţi *sx* , *sy* , *sz* . Dacă *sx* = *sy* = *sz* se obţine o schimbare a dimensiunilor obiectului. Factorii 0<*s*<1 produc micşorarea dimensiunilor obiectului iar factorii *s*>1 produc creşterea dimensiunilor obiectului.

Modelul transformării este :

$x' = x*sx$

$y' = y*sy$

$z' = z*sz$

Matricea care realizează transformarea va fi de forma :

$$M = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Cea mai frecventă utilizare a scalării este în scopul modificării dimensiunii unui obiect. Scalarea este întotdeauna relativă faţă de un punct de referinţă. Formulele de mai sus sunt aplicabile atunci când punctul de referinţă este originea. Pentru modificarea dimensiunii unui obiect faţă de un punct al său numit centru, este necesară compunerea a trei transformări: o translaţie care aduce centrul obiectului în originea sistemului de axe *T-v*, o scalare *S* şi apoi o translaţie care readuce centrul obiectului în poziţia iniţială *T-v*. Centrul corpului se va specifica printr-un vector de poziţie *V (vx , vy , vz)*. Matricea transformării este :

$M = T\text{-}v * S * Tv$

Rotaţia

Rotaţia în spaţiu prezintă 3 cazuri :

- rotaţia în jurul axei Ox
- rotaţia în jurul axei Oy
- rotaţia în jurul axei Oz

Matricea care permite rotaţia în jurul axei OX, în sensul acelor de ceasornic cu un unghi *Q* (privind în lungul semiaxei Ox pozitive înspre origine) este:

$$R_x = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(Q) & \sin(Q) & 0 \\ 0 & -\sin(Q) & \cos(Q) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotaţia în jurul axei OY, în sensul opus acelor de ceasornic cu un unghi *Q* (privind în lungul semiaxei Oy pozitive înspre origine) este descrisă de matricea:

$$R_y = \begin{bmatrix} \cos(Q) & 0 & \sin(Q) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(Q) & 0 & \cos(Q) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotaţia în jurul axei OZ, în sensul acelor de ceasornic cu un unghi *Q* (privind în lungul semiaxei Oz pozitive înspre origine) este descrisă de matricea:

$$R_z = \begin{bmatrix} \cos(Q) & \sin(Q) & 0 & 0 \\ -\sin(Q) & \cos(Q) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotaţiile în sensul opus celui indicat se obţin schimbând semnul celor doi factori *sin(Q)* care apar în matricile de rotaţie.

Aplicaţie:

Realizaţi câte o rotaţie in jurul axei OX, axei OY si axei OZ a unei piramide.