

Grafica 3D+

Cresterea Realismului Imaginilor

Grafica 3D+

Vasile Prejmerean
Cluj - Napoca
2010

Editura Galaxia Gutenberg

Cuprins

Prefață.....	3
1 Introducere.....	4
2 Elemente de grafică 2D	6
2.1 Desenarea pe ecran a figurilor din planul real	7
2.2 Interpretarea imaginilor de pe ecran.....	8
3 Elemente de grafică 3D	9
3.1 Transformări geometrice uzuale.....	9
3.2 Reprezentarea curbelor, suprafețelor și corpurilor	12
3.3 Observarea unui sistem 3_D de puncte	17
3.4 Modelarea corpurilor	19
3.5 Spații de culoare (RGB, CIE XYZ, CIE Lab, CIE Luv)	21
3.5.1 Sistemul RGB	21
3.5.2 Interpretări colorimetrice computerizate	23
3.5.3 Caracterizarea și măsurarea culorilor în sisteme tricromatice.....	23
3.5.4 Conversia RGB → CIE-Lab.....	27
3.5.5 Conversia CIE-Lab → RGB.....	27
3.5.6 Conversia RGB → CIE-Luv.....	28
3.5.7 Conversia CIE-Luv → RGB.....	28
3.6 Creșterea realismului imaginilor tridimensionale.....	28
3.6.1 Eliminarea suprafețelor acoperite.....	29
3.6.2 Texturi.....	35
3.6.3 Lumină și umbră	40
3.6.4 Stereografia.....	42
3.6.4.1 Stereograme	43
3.6.4.2 Vizualizarea unei stereograme.....	44
3.6.4.3 Construcția unei stereograme	47

Cresterea Realismului Imaginilor

3.6.4.4	Realizarea animației	53
3.6.4.5	Stereograme realizate prin puncte aleatoare	54
3.6.4.6	Anaglife	60
4	Reprezentarea și prelucrarea imaginilor digitale	63
4.1	Reprezentarea imaginilor prin funcții și cuvinte picturale	63
4.2	Reprezentarea imaginilor prin arbori	76
4.3	Îmbunătățirea imaginilor	79
4.3.1	Operațiuni punctuale	80
4.3.2	Operațiuni spațiale	87
4.4	Transformări ale imaginilor	95
4.4.1	Determinarea conturului	95
4.4.2	Scheletizare	97
4.4.3	Subțiere	98
4.5	Transformări morfologice (Morphological Processing)	99
4.5.1	Transformări morfologice pentru imagini <i>Alb-Negru</i>	100
4.5.2	Transformări morfologice pentru imagini cu nuanțe de <i>Gri</i>	107
4.5.3	Transformări morfologice pentru imagini <i>Color</i>	112
4.5.3.1	Vectori în spațiul HSV	112
4.5.3.2	Transformări morfologice <i>Soft (Soft Morphological Color)</i> ...	113
4.6	Lipirea (concatenarea) Imaginilor (The Stitching of Images)	115
4.6.1	Problema concatenării	115
4.6.2	Determinarea tăieturii	116
4.6.3	Algoritmul genetic utilizat	118
5	Concluzii – Final	119
6	Bibliografie	120

Prefață

Încă înainte de apariția calculatorului, oamenii de știință, artiștii, ... au fost preocupați de creșterea realismului imaginilor în operele de artă (James Tissot - *Portrait du marquis et de la marquise de Miramon et de leurs enfants*), în diverse dispozitive de redare vizuală a realității (*cutia ... cu oglinzi*).

Este absolut natural, ca și pe calculator, imaginile realizate să fie cât mai apropiate de realitate, atât prin pârgii soft cât și hard. Atât în Grafica pe Calculator cât și în domenii precum Realitate Virtuală, Modelare-Simulare (Vizualizarea Simulării), Sisteme support pentru luarea deciziilor, ... și multe altele, imaginea ocupă poate cel mai important loc, urmată de sunet.

Ceea ce ne propunem, este ca un cititor interesat în realizarea unor imagini cât mai apropiate de realitate, să poată obține informații teoretice și tehnice care să îl ajute în acest scop. Sunt prezentate câteva elemente de bază din grafica bidimensională (2-D) și tridimensională (3-D), precum și de prelucrare a imaginilor și spații de culori. Un mare avantaj față de o imagine de tip fotografie (listabilă – pe hârtie) față de una plană căreia i se aplică doar metodele cunoscute din desen, arte, ... (proiecție perspectivă, lumini, umbre, etc.) îl constituie imaginea dinamică realizabilă pe display prin rotație, animație,... . Considerăm mai interesant în această lucrare, dintre tehnicile de creștere a realismului imaginilor, cele legate de stereografie (stereograme și anaglife geometrice) pentru redarea adâncimii. Aceste tehnici sunt cunoscute de mult timp, ceea ce face ca odată cu creșterea performanțelor hard să fie tot mai utilizate și pe calculator. Este evident că o astfel de imagine spațială (stereo) este de o calitate superioară mult mai apropiată de realitate.

1 Introducere

În ultima perioadă, în tot mai multe aplicații din cele mai diverse domenii apare necesitatea utilizării imaginilor, prezentarea acestora utilizatorului, prelucrarea și interpretarea acestora. Practic, acum este de neimaginat o aplicație fără elemente vizuale care să fie modificate dinamic prezentând utilizatorului o situație reală, o sinteză în scopul luării unor decizii, dinamica unei simulări, etc.

Grafica pe calculator este un domeniu modern, interdisciplinar, cu multiple aplicații practice în tot mai multe domenii de activitate (medicină, arhitectură, artă, și multe altele), aplicații care pot fi realizate datorită calculatoarelor moderne dar și dezvoltării disciplinelor matematice specializate în această direcție.

Grafica pe calculator este o ramură a informaticii care interacționează cu celelalte ramuri, utilizând pe lângă metode, tehnici și algoritmi specifici ei, și unele noțiuni și instrumente de lucru aparținând altor discipline (structuri de date, teoria grafelor, etc.).

În această lucrare sunt abordate aspecte doar din cele două ramuri ale graficii pe calculator, definite de Pavlidis în [[27]] (vezi schema din Figura 1) și anume:

- a) *Grafică propriu-zisă* (prin care sunt construite imagini) și
- b) *Prelucrarea imaginilor* (care studiază transformarea imaginilor).

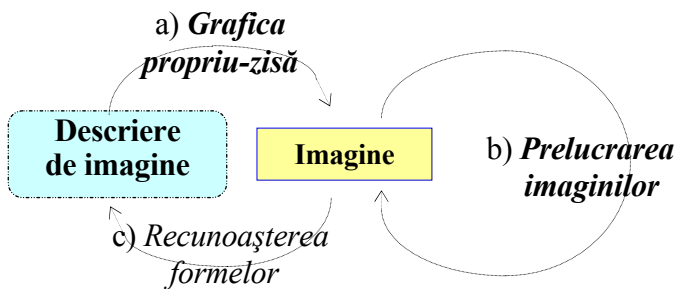


Figura 1 – Componentele Graficii pe Calculator

Grafica 3D+

În capitolul trei al lucrării sunt prezentate câteva elemente ale graficii bidimensionale, scopul acestei părți fiind reprezentarea unei imagini definite în planul real și apoi extragerea de informații din această imagine (interpretarea elementelor grafice).

Capitolul patru conține elemente de grafică tridimensională având ca scop creșterea realismului imaginilor, precum și algoritmi de trecere a unei imagini dintr-o clasă în alta (vezi Figura 2 descrisă în [[27]]). Sunt prezentate modele de reprezentare și de vizualizare a corpurilor, spații de culori, precum și diverse metode de creșterea a realismului imaginilor reprezentate (*eliminarea suprafețelor acoperite, texturi, realizarea de lumini și umbre* precum și tehnici de reprezentare în relief a obiectelor grafice tridimensionale prin *stereograme și anagliffe geometrice*).

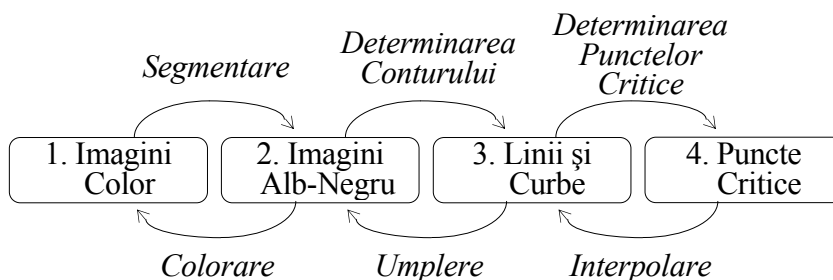


Figura 2 - Clase de imagini

În capitolul cinci sunt prezentate diverse metode, tehnici și algoritmi utilizați în reprezentarea și prelucrarea digitală a imaginilor, atât în vederea creșterii calității imaginii destinate ochiului uman (paragraful doi intitulat *Îmbunătățirea imaginilor, prin operațiuni punctuale și spațiale*) cât și în scopul recunoașterii formelor (paragraful trei *Transformări ale imaginilor* cuprinzând algoritmi de *determinare a conturului, scheletizare și subțiere*). Sunt prezentați algoritmi clasici de prelucrare a imaginilor apoi transformările morfologice aplicate pe imagini alb-negru, cu nuanțe de gri și color. De asemenea mai este prezentată o modalitate de concatenare a imaginilor.

2 Elemente de grafică 2D

Dacă dorim să trasăm pe ecran primitive grafice referite prin coordonate aflate într-un domeniu real, va trebui să aplicăm o transforme a acestor coordonate, pentru că instrucțiunile grafice referă puncte de coordonate întregi din domeniul $[0, Dmx] \times [0, Dmy]$.

Există posibilitatea de a realiza un desen doar pe o porțiune a ecranului, pe un subdomeniu $[u_1, u_2] \times [v_1, v_2]$ al domeniului maxim $[0, Dmx] \times [0, Dmy]$.

Acest subdomeniu îl vom numi *fereastră fizică (ViewPort)* și va fi precizat prin coordonatele ecran a două puncte diagonal opuse $P_1(u_1, v_1)$ și $P_2(u_2, v_2)$.

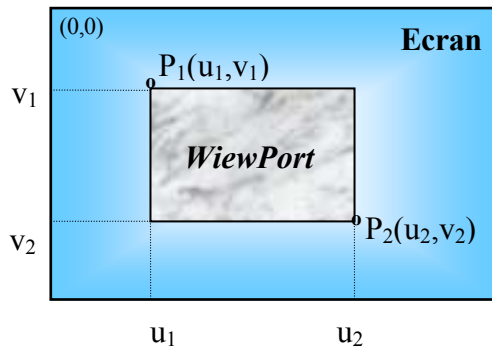


Figura 3 - Viewport

Prin u_1, v_1, u_2, v_2 vom preciza fereastra fizică în care vom efectua operații grafice. Acești parametri sunt de tip întreg și îndeplinesc următoarele restricții:

$$0 \leq u_1 < u_2 < Dmx = \text{GetMaxX}$$

$$0 \leq v_1 < v_2 < Dmy = \text{GetMaxY}$$

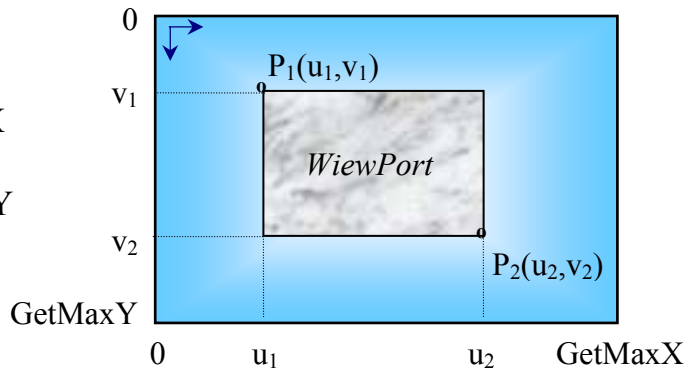


Figura 4 – Defnire Viewport

2.1 Desenarea pe ecran a figurilor din planul real

În fereastra ecran $[u_1, u_2] \times [v_1, v_2]$ (*ViewPort*) putem reprezenta puncte $P(x,y)$ dintr-un domeniu $[a,b] \times [c,d] \subset \mathbf{R}^2$ numit *fereastră reală* (*Window*) (vezi Figura 5). Această fereastră se poate defini prin două puncte diagonal opuse de exemplu stânga-sus (a,d) și dreapta-jos (b,c) , care corespund la reprezentarea pe ecran punctelor $P_1(u_1,v_1)$ și $P_2(u_2,v_2)$. Pentru a putea referi în instrucțiunile grafice un astfel de punct, va trebui să aplicăm o transformare de coordonate $P(x,y) \rightarrow M(u,v)$, care verifică egalitățile:

$$\left\{ \begin{array}{l} \frac{x-a}{b-a} = \frac{u-u_1}{u_2-u_1} \\ \frac{y-d}{c-d} = \frac{v-v_1}{v_2-v_1} \end{array} \right. \text{ de unde rezultă } \left\{ \begin{array}{l} u := \text{Round}((x-a) * (u_2-u_1) / (b-a)) + u_1 \\ v := \text{Round}((y-d) * (v_2-v_1) / (c-d)) + v_1 \end{array} \right.$$

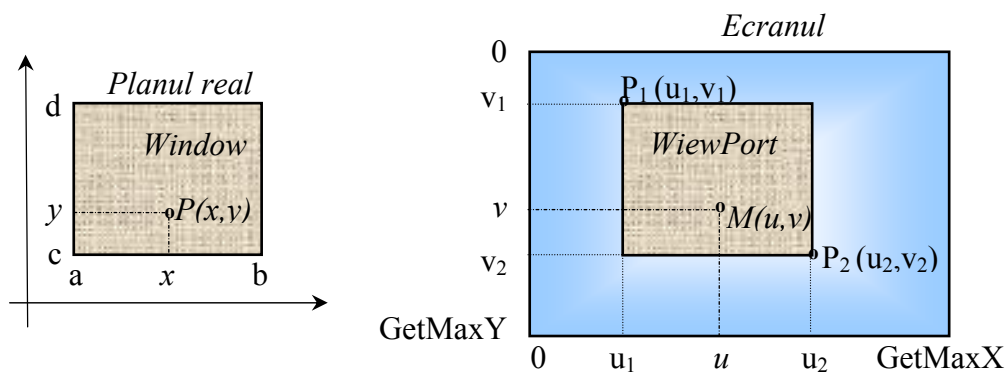


Figura 5 – Fereastră Reală - Window

Pentru reprezentarea unei funcții, a și b pot fi capetele domeniului de definiție, iar c și d minimul respectiv maximul funcției pe intervalul $[a,b]$.

Dacă dorim să referim un punct de coordonate reale din fereastra logică (*Window*) după declararea ferestrei ecran (*ViewPort*), deci în coordonate relative față de colțul ferestrei ecran, atunci transformarea

$$P(x,y) \rightarrow M(u',v'), \quad (x,y) \in [a,b] \times [c,d],$$

Cresterea Realismului Imaginilor

necesară pentru apelul unei rutine grafice este următoarea:

$$\begin{cases} u' := \text{Round}((x-a)*(u_2-u_1)/(b-a)) \\ v' := \text{Round}((y-d)*(v_2-v_1)/(c-d)) \end{cases}$$

2.2 Interpretarea imaginilor de pe ecran

În acest paragraf ne interesează cum putem afla coordonatele reale ale unui punct precizat printr-un pixel de pe ecran, deci problema inversă și anume transformarea din fereastra ecran în cea reală ($M(u,v) \rightarrow P(x,y)$, vezi Figura 6). Acest lucru este necesar în situația în care avem reprezentat pe ecran un desen și ne interesează, de exemplu, coordonatele aproximative ale unui punct pe care îl precizăm cu ajutorul dispozitivului de mouse. Aceste transformări care se pot deduce ușor tot din proporțiile prezentate în primul paragraf sunt următoarele (vezi Figura 6):

$$\begin{cases} x := (u-u_1)/(u_2-u_1)*(b-a)+a & (= u^{-1}) \\ y := (v-v_1)/(v_2-v_1)*(c-d)+d & (= v^{-1}) \end{cases}$$

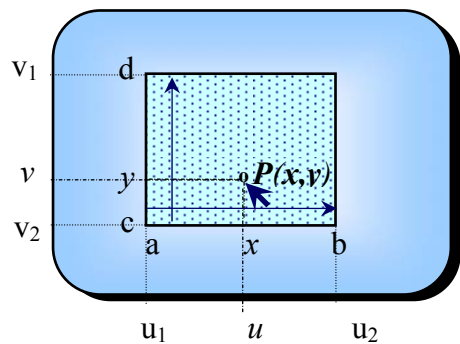


Figura 6 – Interpretarea imaginilor

Prin interpretarea imaginilor putem să aproximăm maximum sau minimum unei funcții după desenarea graficului acesteia (vezi Figura 7), putem să rezolvăm o ecuație $f(x)=0$, prin aproximarea rădăcinilor aflate la intersecția graficului cu axa Ox , să determinăm intersecția graficului cu axa Oy , să descoperim asimptotele, și multe altele din diverse domenii (geometrie,

Grafica 3D+

arhitectură, construcții, etc.). Acest studiu este avantajos în situația în care este dificil să determinăm aceste valori prin metodele matematice clasice.

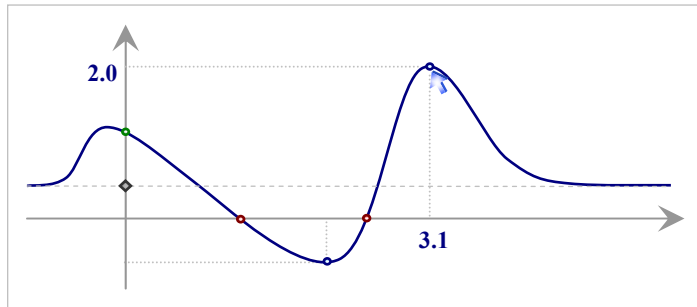


Figura 7 – Interpretarea graficelor

3 Elemente de grafică 3D

În acest capitol sunt prezentate câteva noțiuni elementare de grafică tridimensională necesare reprezentării obiectelor din \mathbf{R}^3 (curbe, suprafețe și corpuri) apoi câteva tehnici de creștere a realismului imaginilor.

3.1 Transformări geometrice uzuale

Printr-o transformare a unui punct $P(x,y,z) \in \mathbf{R}^3$ se va obține un alt punct $P'(x',y',z') \in \mathbf{R}^3$.

a) Translație

Prin *translația* cu $(\Delta x, \Delta y, \Delta z)$ a punctului $P(x,y,z)$ se obține punctul $P'(x',y',z')$ după formula:

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta z \end{pmatrix}$$

Cresterea Realismului Imaginilor

b) Rotație

Rotația în jurul axei Oz cu unghiul α (vezi Figura 8) se obține astfel :

$$x' = OP' \cdot \cos(\alpha + \beta) = OP \cos \alpha \cos \beta - OP \sin \alpha \sin \beta = x \cos \alpha - y \sin \alpha ,$$

$$y' = OP' \cdot \sin(\alpha + \beta) = OP \sin \alpha \cos \beta + OP \cos \alpha \sin \beta = x \sin \alpha + y \cos \alpha ,$$

$$z' = z .$$

De aici rezultă:

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{pmatrix} * \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

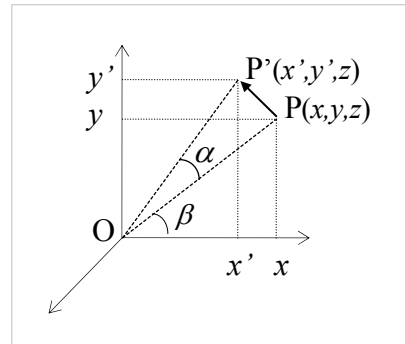


Figura 8 - Rotație

Analog se deduc și formulele de rotație în jurul axelor Ox respectiv Oy .

c) Scalare

În funcție de factorul de scalare, imaginea se *dilată* (factor supraunitar) sau se *contractă* (factor subunitar) astfel:

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = f_g * \begin{pmatrix} f_x & 0 & 0 \\ 0 & f_y & 0 \\ 0 & 0 & f_z \end{pmatrix} * \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

d) Simetrie

Dând valori coeficienților S_x , S_y și S_z din mulțimea $\{-1,0,1\}$ se vor obține simetriile față de origine axele de coordonate și planele determinate de acestea.

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & S_z \end{pmatrix} * \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

Grafica 3D+

e) Proiecție

Pentru a putea reprezenta pe ecran obiecte tridimensionale vom proiecta mai întâi punctele din \mathbf{R}^3 în planul real, apoi prin transformările de fereastră (u și v prezentate în primul capitol) vom face trecerea din \mathbf{R}^2 în fereastra ecran. În cele ce urmează vom prezenta formulele de calcul pentru proiecția *paralelă* (sau *cilindrică*) și proiecția *perspectivă* (sau *conică*).

Pentru proiecția *paralelă* a un punct oarecare $P(x,y,z)$ va trebui să calculăm coordonatele x' și y' ale proiecției acestuia după direcția precizată (prin r și α). Se poate observa că triunghiul AOB este asemenea cu triunghiul PDP' , de unde rezultă că $1/r = z/DP'$, deci $DP' = r \cdot z$. Pentru că $x' = x + DP' \cdot \cos\alpha$ și

$$y' = y + DP' \cdot \sin\alpha$$

rezultă că:

$$Prx(x,z) = x' = x + r \cdot z \cdot \cos\alpha \quad \text{iar}$$

$$Pry(y,z) = y' = y + r \cdot z \cdot \sin\alpha.$$

Pentru proiecția *conică* (definită prin distanța la care se află observatorul $\Omega(0,0,d)$ notată cu $d=O\Omega$ și distanța la care se află planul de proiecție notată cu $q=OO'$, vezi Figura 10) observăm că $\Delta\Omega AD \sim \Delta\Omega O'C \sim \Delta\Omega PD \sim \Delta\Omega P'C$, de unde rezultă că:

$$x/x' = y/y' = (d-z)/(d-q).$$

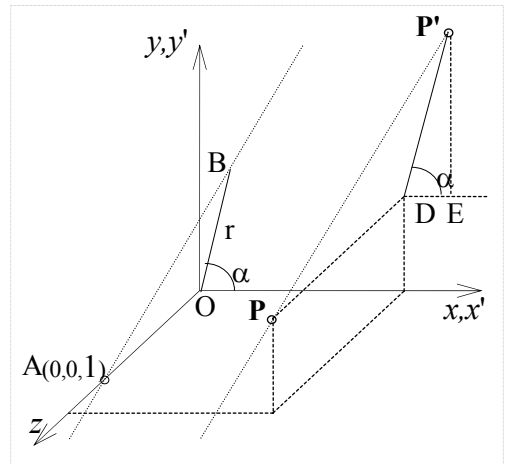


Figura 9 – Proiecție paralelă

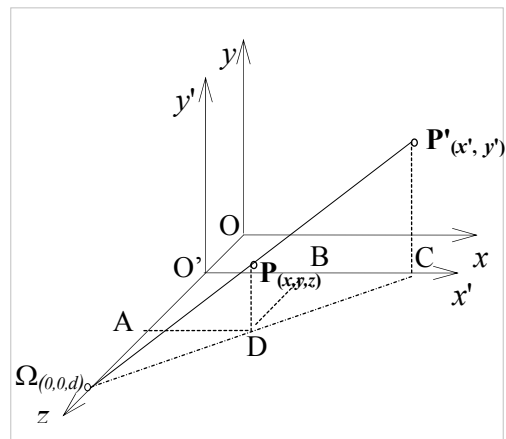


Figura 10 – Proiecție conică

Cresterea Realismului Imaginilor

Deducem că:

$$Prx(x,z) = x' = x \cdot (d-q) / (d-z), \quad \text{iar}$$

$$Pry(y,z) = y' = y \cdot (d-q) / (d-z).$$

3.2 Reprezentarea curbelor, suprafețelor și corpurilor

Pentru a reprezenta pe ecran un sistem de puncte 3D, acestea le vom aduce în planul de proiecție ($\mathbf{R}^3 \rightarrow \mathbf{R}^2$) utilizând transformările de proiecție descrise anterior, apoi din planul real de proiecție, mai exact din fereastra reală (definită sau determinată), aceste proiecții se vor aduce în fereastra ecran utilizând transformările de fereastră ($Window \rightarrow Viewport$).

$$\begin{array}{ccc} Prx, Pry & u(x), v(y) & \\ P(x,y,z) \rightarrow P'(x',y') \in & \rightarrow M(u,v) & \\ \in \mathbf{R}^3 & \in \mathbf{R}^2 & \in Ecran \end{array}$$

Etapele realizării unei reprezentări pe ecran a unui sistem de puncte 3D $P_i(x_i, y_i, z_i)$, $i=1,2,\dots,n$ sunt următoarele:

a) Definirea proiecției (tipul proiecției – paralelă sau perspectivă, parametrii – r, α respectiv d, q);

b) Precizarea sau determinarea ferestrei reale (*Window*):

$$\begin{aligned} a &= \text{Min}(Prx(P_i)), \quad b = \text{Max}(Prx(P_i)), \\ c &= \text{Min}(Pry(P_i)), \quad d = \text{Max}(Pry(P_i)), \quad i=1,2,\dots,n. \end{aligned}$$

c) Precizarea ferestrei ecran: $Viewport = ((u_1, v_1), (u_2, v_2))$;

d) Utilizarea coordonatelor de pixeli $u(Prx(P_k)), v(Pry(P_k))$.

Reprezentarea unei **curbe** se poate realiza prin unirea proiecțiilor unui șir de puncte de pe aceasta. Practic, curba se va desena printr-un șir de segmente (vezi Figura 11 a). Ca exemplu (vezi Figura 11 b), am ales reprezentarea unei curbe definite parametric $x=x(\alpha)$, $y=y(\alpha)$, $z=z(\alpha)$. La

Grafica 3D+

reprezentarea din Figura 11 b acționează iluzia optică, de aceea reprezentarea printr-o anaglifă (ca cea din Figura 11 c, prezentată ca element de creștere a realismului imaginilor în cele ce urmează) redă spațialitatea (adâncimea) și elimină iluzia optică.

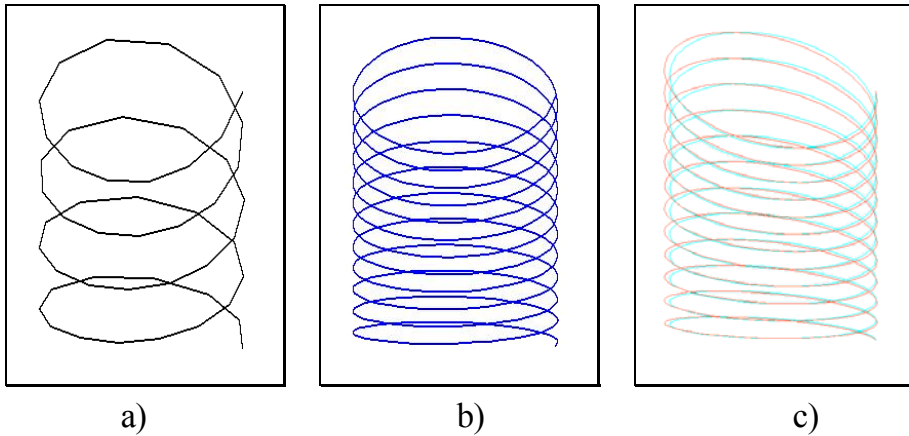


Figura 11 – Spirala

Un algoritm simplu de trasare a unei curbe poate fi următorul:

```

n = 500; // n = Numarul de puncte/segmente
Alfa = 24 * Pi; // Alfa = Lungimea spiralei ~ 12
rotații
Pentru i = 0,n Execută
    α = Alfa * i / n;
    Vi = ( x(α), y(α), z(α) )
Sf_Pentru;
ViewPort(400, 100, 600, 400);
DefPr (500, 100); // Pr. Persp.(d,q);
a = b = Prx(V0); // Determinare Window
c = d = Pry(V0);
Pentru i = 1,n Execută
    px = Prx(Vi); Dacă (px < a) a = px; Altfel Dacă (px > b) b = px;
    py = Pry(Vi); Dacă (py < c) c = py; Altfel Dacă (py > d) d = py;
Sf_Pentru;
Window(a, d, b, c); // Declarare Window
Pentru i = 1,n Execută // Desenare segmente
    Linie (u(Prx(Vi-1)), v(Pry(Vi-1)), u(Prx(Vi)), v(Pry(Vi)));
Sf_Pentru;
    
```

Cresterea Realismului Imaginilor

Pentru a reprezenta o **suprafață** definită de o funcție $z:[x_1,x_2] \times [y_1,y_2] \rightarrow \mathbb{R}$, va fi necesară o proiecție a punctelor din \mathbb{R}^3 în planul XOY . Vom considera o proiecție paralelă după o direcție precizată prin cele două elemente: r și α , așa cum am văzut în Figura 9.

Trasarea suprafeței se poate reduce la desenarea a două sisteme de curbe. Pentru un punct x fixat, iar y variabil din intervalul $[y_1,y_2]$ se trasează graficul proiecției corespunzătoare punctelor $P(x,y,z(x,y))$. Acest lucru se realizează pentru mai multe puncte din intervalul $[x_1,x_2]$ ($n+1$ puncte echidistante). Același lucru se realizează pentru y fix și x variabil, realizând astfel o imagine sub forma unei plase (ca în Figura 12 a).

Practic se reprezintă două șiruri de curbe $z(x_i,y)$ și $z(x,y_j)$, unde :

$$x_i = x_1 + i * (x_2 - x_1) / n \quad (i=0,1,\dots,n),$$

$$y_j = y_1 + j * (y_2 - y_1) / m \quad (j=0,1,\dots,m), \text{ iar } n \text{ și } m \text{ reprezintă finețea}$$

rețelei.

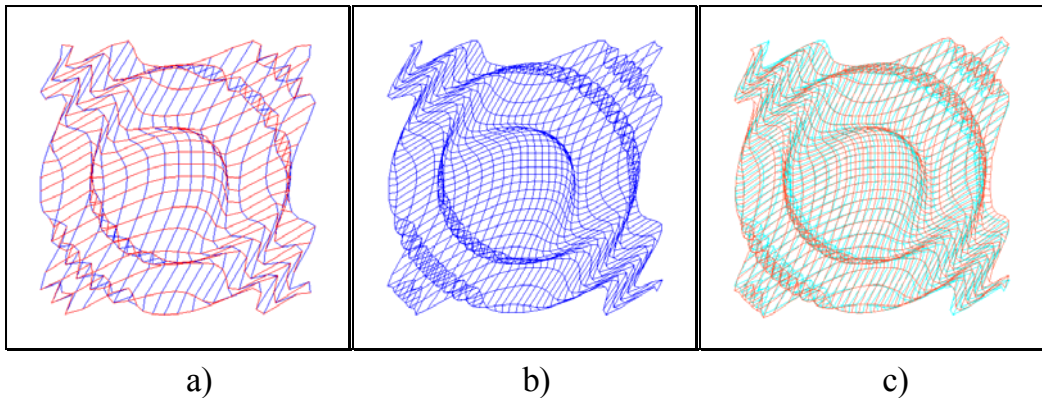


Figura 12 – Suprafață

Față de imaginea din Figura 12-b unde doar ne închipuim această suprafață ca fiind în spațiu, în imaginea din Figura 12-c această suprafață chiar se vede în spațiu deoarece această suprafață este reprezentată din două unghiuri diferite cu două culori pentru a fi văzute fiecare cu câte un ochi, realizând în acest fel efectul stereografic. Percepția diferită se va realiza cu ajutorul ochelarilor special având cele două culori utilizate (*Aqua* și *Tomato*).

Grafica 3D+

Asupra acestor efecte, precum și asupra modului de realizare a anaglifelor geometrice vor reveni în capitolul următor.

În exemplul următor, se desenează (vezi Figura 12 b) o suprafață descrisă de funcția:

$$z: [-\pi, \pi] \times [-\pi, \pi] \rightarrow \mathbf{R}, \quad z(x, y) = \cos(x^2 + y^2),$$

aplicând o proiecție paralelă de direcție $(r, \alpha) = (1, \pi/4)$.

```
n = 35; m = 35; // Numărul de curbe pe
x/y
x1 = -pi, y1 = pi; x2 = pi, y2 = pi; // Dom. Def.
Pentru i = 0, n Execută // Matricea de puncte
    x = x1 + (x2 - x1) * i / n;
    Pentru j = 0, m Execută
        y = y1 + (y2 - y1) * j / m;
        Vij = (x, y, z(x, y));

ViewPort(400, 100, 800, 500); // Fereastra Ecran
DefPr (1, Pi / 4); // Pr. Par.(r,α);

a = b = Prx(V00); c = d = Pry(V00); // Det. Fereastra Reală
Pentru i = 0, n Execută
    Pentru j = 0, m Execută
        px = Prx(Vij); Dacă (px < a) a = px; Altfel Dacă (px > b) b = px;
        py = Pry(Vij); Dacă (py < c) c = py; Altfel Dacă (py > d) d = py;
    Sf_Pentru;
Sf_Pentru;

Window(a, d, b, c); // Def. Fereastra Reală

Pentru i = 0, n Execută
    Pentru j = 1, m Execută
        Linie(u(Prx(Vi,j-1)), v(Pry(Vi,j-1)), u(Prx(Vi,j)), v(Pry(Vi,j)));
    Sf_Pentru;
Sf_Pentru;

Pentru j = 0, m Execută
    Pentru i = 1, n Execută
        Linie(u(Prx(Vi-1,j)), v(Pry(Vi-1,j)), u(Prx(Vi,j)), v(Pry(Vi,j)));
    Sf_Pentru;
Sf_Pentru;
```


Cresterea Realismului Imaginilor

O altă problemă pe care o propunem cititorului ca temă, constă în construirea și reprezentarea grafică a unei suprafețe de rotație. Suprafața din Figura 13 a fost generată plecând de la funcția $z(y) = \sin y^2$, al cărui grafic se rotește în jurul axei Oy , apoi se reprezintă segmentele ce unesc punctele obținute prin rotire. În felul acesta sunt reprezentate două sisteme de curbe de tip *meridiane* și *paralele* așa cum se poate vedea în Figura 13.

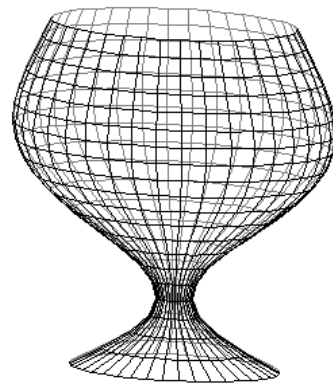


Figura 13 – Suprafață de rotație

Reprezentarea pe ecran a unui **corp** modelat prin muchiile sale (date într-un fișier text) este realizată prin desenarea proiecțiilor acestora. Fișierul va conține următoarele date :

n *Lista de vârfuri dată explicit:*

x_1	y_1	z_1	
x_2	y_2	z_2	$P_i(x_i, y_i, z_i), i = \overline{1, n}$
\dots			
x_i	y_i	z_i	
\dots			
x_n	y_n	z_n	

m *Lista de muchii dată*

s_1	d_1	c_1	<i>implicite prin indici de vârfuri</i>
s_2	d_2	c_2	<i>(s_j, d_j) și caracteristici de</i>
\dots			<i>muchii (c_j) :</i>
s_j	d_j	c_j	
\dots			
s_m	d_m	c_m	$S_j(s_j, d_j, c_j), j = \overline{1, m}$

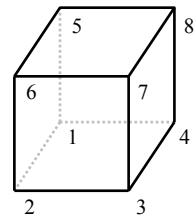
Tip r/d α/q *Definirea proiecției:*

Tip, (r, α) sau (d, q)

De exemplu pentru cubul de mai jos, datele sunt următoarele :

```

8
0 0 0
1 0 0
1 1 0
0 1 0
0 0 1
1 0 1
1 1 1
0 1 1
12
1 2 1
2 3 2
3 4 2
4 1 1
5 6 2
6 7 2
7 8 2
8 5 2
1 5 1
2 6 2
3 7 2
4 8 2
1 1.0 0.4
    
```



// 1=Pr.Par (1,0.4)

Grafica 3D+

Următorul algoritm reprezintă un corp prin muchiile sale:

```

Citește (f, n); // n= Nr. de vârfuri
Citește (f, Vi), pentru i = 1,n; // xi yi zi
Citește (f, m); // m= Nr. de muchii
Citește (f, Mj), pentru j = 1,m; // Stj Drj Carj
ViewPort(400, 100, 1000, 800); // Def. Fer. Ecran
Citește (f, TipPr, rd,alq); // Def. Pr.: Tip, (r, α) sau (d,q)
DefPr(TipPr, rd,alq); // Calcul Fer. Reală
a = b = Prx(V1); c = d = Pry(V1);
Pentru i = 2,n Execută
    px = Prx(Vi); Dacă (px < a) a = px; Altfel Dacă (px > b) b = px;
    py = Pry(Vi); Dacă (py < c) c = py; Altfel Dacă (py > d) d = py;
Window(a, d, b, c); // Def. Fer. Reală
Pentru j = 1,m Execută // Desenare Muchii
    Linie (u(Prx(VMj.St)), v(Pry(VMj.St)), u(Prx(VMj.Dr)),v(Pry(VMj.Dr)),Carj);
    
```

3.3 Observarea unui sistem 3_D de puncte

Un corp solid, tridimensional, îl putem modela printr-un sistem de puncte tridimensionale $S = \{ P_i(x_i, y_i, z_i) \in \mathbf{R}^3, i=1,2,\dots,n \}$.

Considerăm că observatorul se află în punctul $\Omega(x_0, y_0, z_0)$ și privește spre *centrul* obiectului care poate fi considerat ca fiind centrul de greutate $C(x_c, y_c, z_c)$, (x_c, y_c și z_c le putem calcula ca medie aritmetică a coordonatelor x_i, y_i respectiv z_i ale punctelor $P_i, i=1,2,\dots,n$).

Pe direcția de observare ΩC (și perpendicular pe aceasta), la o distanță d precizată (în punctul $O_1(x_1, y_1, z_1)$) se așează planul de proiecție după o verticală de asemenea dată (vezi Figura 14).

Coordonatele punctului O_1 se determină astfel:

$$O_1 = (1-t) \cdot \Omega + t \cdot C, \quad \text{unde } t = \Omega O_1 / \Omega C$$

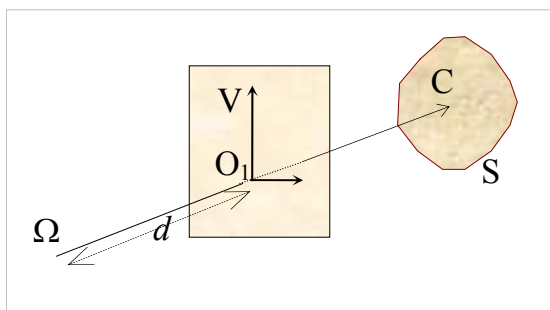


Figura 14 – Planul de proiecție

Cresterea Realismului Imaginilor

Proiecția unui punct $P \in S$ se determină prin intersecția segmentului ΩP cu planul $x'O_1y'$ determinat anterior. Deoarece aceste calcule sunt relativ complexe, vom prezenta în continuare o modalitate mai simplă de rezolvare a acestei probleme. În paragraful precedent, au fost date formulele de proiecție în ipoteza că observatorul se află pe axa Oz (vezi Figura 10). Pentru a putea utiliza aceste formule vom proceda după cum urmează:

a) Se translatează toate punctele din sistem, cu $(-x_1, -y_1, -z_1)$, astfel încât $O_1=O$, iar $\Omega(x_0, y_0, z_0)$ se va translata în punctul $\Omega_1(x_0-x_1, y_0-y_1, z_0-z_1)$.

b) Se aplică o rotație (tuturor punctelor) în jurul axei Oz cu unghiul α (vezi Figura 15) astfel încât observatorul să ajungă în planul xOz .

c) Se aplică o a doua rotație în jurul axei Oy cu unghiul β (care se poate determina în funcție de coordonatele punctului Ω_2) pentru ca observatorul să ajungă pe axa Oz (vezi Figura 16). Prin această rotație, observatorul ajunge în punctul Ω_3 pe axa Oz .

d) În final pentru ca imaginea să nu fie răsturnată, vom aduce verticala V pe axa Oy , printr-o rotație cu unghiul γ (calculat în funcție de coordonatele verticalei V) în jurul axei Oz (vezi Figura 17).

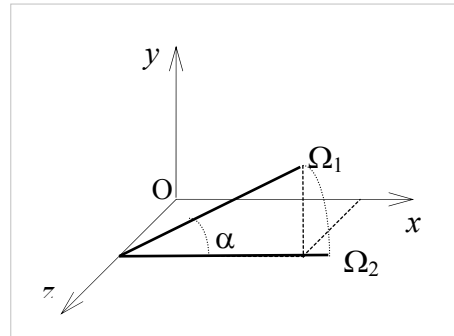


Figura 15 – Prima rotație

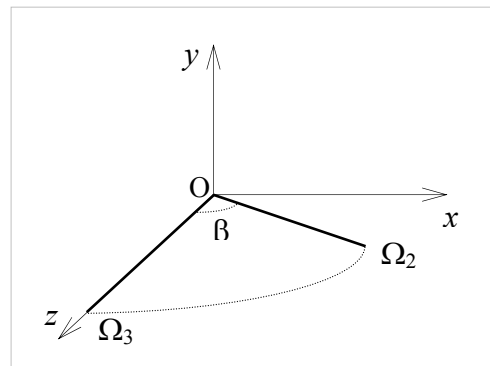


Figura 16 – A doua rotație

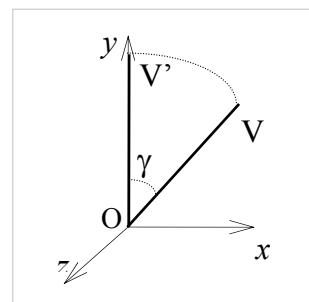


Figura 17 – Verticala pe Oy

3.4 Modelarea corpurilor

Există mai multe modele de reprezentare a obiectelor grafice tridimensionale (descrise în [[7]) din care vom prezenta câteva în cele ce urmează.

a) Instanțierea primitivelor pure

Această metodă se utilizează în industrie, fiind foarte specializată pe un anumit tip de obiect (șurub, elice, etc), care este puternic tipizat și parametrizat. Aceste prototipuri generice prin instanțiere vor conduce la reprezentarea grafică, deci la întocmirea unui proiect corespunzător unui obiect dorit de proiectant.

Lipsa operatorilor, face ca acest model să fie greu de utilizat în realizarea de corpuri compuse, complexe sau altele decât acelea pentru care este conceput.

b) Enumerarea ocupării spațiale

Acest model utilizează o rețea tridimensională pe care se pot defini *cubele* (celule spațiale numite *voxeli*) de dimensiuni egale și și cu ajutorul cărora se *construiesc* (se descriu) corpurile.

Deoarece la descriere se utilizează multe componente, aceasta conduce la o memorie mare și timp de realizare costisitor.

c) Descompunerea în celule elementare disjuncte

Modelul acesta vine să completeze, să generalizeze modelul anterior, prin faptul că de această dată este permisă utilizarea de celule de dimensiuni diferite, cu intersecții disjuncte și fără *goluri*. În acest mod, atât timpul cât și memoria necesară este redusă. Elementele mici de construcție vor fi utilizate doar la frontiera obiectului, pentru finisarea acestuia, în rest pot fi folosite elemente de dimensiuni cât mai mari.

Cresterea Realismului Imaginilor

d) Interpolare

Obiectul este considerat (definit) ca fiind o reuniune de segmente ale căror extremități se află (*se plimbă*) pe o mulțime de puncte din spațiu (\mathbf{R}^3). De exemplu suprafața laterală a unui con este reuniunea segmentelor având o extremitate într-un punct (vârful conului) și cealaltă pe un cerc (baza conului).

Lipsa operatorilor conduce la imposibilitatea compunerii corpurilor descrise prin *interpolare*.

e) Geometria solidă constructivă

Această metodă pune la dispoziția utilizatorului o mulțime de primitive grafice solide tridimensionale (cum ar fi cub, paralelipiped, cilindru, con, etc.), precum și o mulțime de operatori (reuniune, intersecție, diferență, etc.) care permit descrierea de corpuri complexe prin expresii asemănătoare cu cel aritmetice și care la rândul lor pot fi reprezentate prin arbori binari. În felul acesta se pot defini adevărate biblioteci de corpuri grupate pe diverse domenii. Primitivele pot avea și anumite caracteristici (diverse atribute cum ar fi textură culoare, etc.), ceea ce face ca această metodă să fie o metodă generoasă prin care se pot construi scene complexe plecând de la anumite corpuri deja construite sau construite direct de către utilizator.

f) Reprezentarea prin măturare

Metoda este folosită în diverse procese tehnologice unde anumite mașini de prelucrare a materialelor sunt concepute (sau pot fi programate) să realizeze o suprafață a materialului definind conturul printr-o curbă care glisează (alunecă) pe o altă curbă (*sweep de translație* sau *de rotație*).

g) Reprezentarea prin frontiere

Reprezentarea corpurilor prin definirea unor elemente de frontieră (vârfuri, muchii, fețe) poate fi realizată prin mai multe modele și anume: reprezentarea printr-o *ierarhie de puncte (vertexuri)*, prin *secțiuni transversale*, *cadru de sârmă*, forme *poliedrale*, etc.