

Prefață

M-am străduit să am o abordare echilibrată între nevoia de a descrie/sintetiza aspecte actuale ale graficii pe calculator și necesitatea de a-i învăța pe studenți, la primul curs de grafică, aspectele/ideile de bază ale graficii pe calculator.

Am insistat pe conceptele matematice care stau la baza metodelor specifice graficii pe calculator prezentate în curs, pentru că, în mod sigur, direcțiile de dezvoltare viitoare ale acestui domeniu (direcțiile de creare de noi metode originale) se vor baza pe matematica/fizica evoluată, pe abstractizare și pe paralelism.

Aplicațiile majore din domenii ca: jocurile video, desene animate de sinteză, efecte speciale în filme, proiectare asistată de calculator, programe de simulare (simulatoare de zbor/navale) imagistică medicală, vizualizare științifică, realitate virtuală, prelucrări de imagini, interacțiunea cu utilizatorul, cer pentru implementare cunoștințe solide de grafică pe calculator, matematică (geometrie), fizică, structuri de date, tehnici de programare (care includ și programarea paralelă), metode numerice.

Exemplele (în principal rutine sursa C) din lucrare conțin implementări cât mai simple și clare ale algoritmilor/metodelor de bază care să aplice conceptele teoretice, acordând o atenție mai redusă procedurilor sau structurilor de date mai complicate/evolute care să producă algoritmi mai eficienți dar stufoși și mai greu de înțeles.

Studentul ar trebui să citească mai întâi partea a iii-a a lucrării, care conține în principal definiții ale unor concepte matematice folosite la cursul de grafică pe calculator, care sunt referite din ce în ce mai mult în cursurile de grafică actuale și care au cam început să fie neglijate/uite de studenții politehniști. Nu am făcut o simplă anexă cu “noțiuni de matematică” ci am pus descrierea lor în corpul principal al cursului, pentru a-i îndemna pe studenți și prin această “ridicare în rang” să acorde acestor aspecte atenția necesară. Dacă studenții stăpânesc deja aceste noțiuni nu trebuie să mai insiste. Dacă nu le știu suficient, e bine să citească mai multe despre ele (cel puțin de pe internet). Dacă le place, să nu se sfiască să aprofundeze, să pună întrebări. Se pune accent pe matematică pentru că aceasta îi poate face pe absolvenți în stare să se descurce în fața unor probleme noi care pot apare în dezvoltarea unor aplicații.

Cartea are și o parte de geometrie algoritmică. La universități precum Cambridge, cursul de geometrie algoritmică este un preliminar al cursului de grafică pe calculator. La Politehnică acest curs se desfășoară la o direcție specializată de master (GMRV – Grafică, Multimedia și Realitate Virtuală). În această lucrare, am selectat numai câteva aspecte și tehnici simple, urmând aceeași filozofie specifică unui curs introductiv: algoritmi simpli de înțeles chiar dacă nu sunt cei mai eficienți.

Din studiul evoluției în timp (în perioada 1992-2015) a cursurilor de grafică pe calculator, (publicate la edituri de renume) precum și al nivelului de pregătire la matematică al studenților români (de la Automatica) și al celor din străinătate cu care am intrat în contact până în 2004, (Olanda, Franța, Italia) îmi permit să estimez că actualmente cele două nivele sunt comparabile

(în anii <2000 nivelul mediu al pregătirii la matematică a studenților români era, după părerea mea, superior) deci încă nu ne-am pierdut pe deplin avantajul competitiv al pregătirii la matematică. Acum, cursurile bune (internaționale) de grafică pe calculator au început să utilizeze cunoștințe matematice, pe care un student mediu de la Politehnică nu prea le stăpânește.

Lucrarea cuprinde o selecție a aspectelor celor mai de bază din domeniul graficii pe calculator. Ea conține atât material cât este posibil de citit și învățat de un student mediu (care mai urmează și alte cursuri) într-un semestru. Nu este o abordare exhaustivă a domeniului graficii pe calculator în genul [Hug 14] sau [Glas 95]. Am preferat un stil de prezentare mai concis deși, din punct de vedere didactic ar fi fost poate mai bine dacă explicam mai mult fiecare concept. Am restrâns numărul de pagini pentru a putea fi accesibilă studenților.

În octombrie 2015, am recomandat studenților (cât am putut eu mai convingător) să își achiziționeze (sau să fotocopieze capitole din) [Hug 14] dar, spre dezamăgirea mea, nici un student nu a cumpărat această carte foarte bună de grafică pe calculator. Ca o comparație, când în 1979 profesorul Sipoș a recomandat studenților să își fotocopieze cartea de Sisteme de Operare a lui Donovan (întrucâtva echivalentă valoric, (în domeniul sistemelor de operare) în acea perioadă, a cărții de grafică actuale [Hug 14]), marea majoritate a studenților (pe atunci erau doar 60 de studenți la direcția de specializare software) l-a ascultat și a citit/fotocopiat această carte. Scuze că acum există documentație pe Internet, nu este după părerea mea, decât parțial valabilă. Efortul necesar pentru a ajunge la același nivel de cunoaștere (într-un semestru de studiu) prin citirea unei cărți fundamentale sau prin explorarea neghidată a documentației din domeniul graficii pe calculator disponibilă pe internet, este clar mai mic. Abia după ce citești cel puțin o carte fundamentală într-un domeniu, poți să aprofundezi folosind mult Internetul. Într-un singur semestru de studiu, învățarea tuturor cunoștințelor esențiale (care sunt evaluate la un examen) dintr-un domeniu, folosind numai Internetul (fără o cunoaștere prealabilă a domeniului) este practic imposibilă, ținând cont că în acel semestru se desfășoară mai multe cursuri. Există desigur și tutoriale pe Internet, de obicei ele dau o imagine parțială asupra unui subdomeniu al disciplinei, nu o vedere de ansamblu. Este adevărat că pot exista și studenți mai inventivi care pot ajunge uneori la rezultate originale, fără să citească prea multe cărți/documentație; de aceea trebuie spus că cele menționate mai sus sunt valabile în primul rând pentru “studentul mediu/generic”, dar acest tip de student este cel majoritar.

În lucrare sunt infățișați algoritmi reprezentativi, În general, fiecare algoritm prezentat aparține unei categorii distincte de algoritmi grafici și introduce concepte noi (daca sunt mai mulți algoritmi din aceeași categorie, fiecare din ei introduce principii noi de abordare a unei probleme (vezi de exemplu secțiunile 7.3.2.1, 7.3.2.2 sau 15.2, 15.3). Am prezentat și tehnicile de implementare ale apelurilor din bibliotecile (API) sistemelor grafice curente (OpenGL) (vezi capitolele 5 sau 9), pentru că am convingerea că adevărații specialiști trebuie să știe nu numai să folosească ceea ce au făcut alții și să producă aplicații (“loan”) cu valoare adăugată mai mare sau mai mică, ci să “știe/poată să facă” unelte de programare care să fie folosite de alți specialiști/programatori/dezvoltatori. Cu alte cuvinte să fie în stare să dezvolte procedee noi. Asta este mult mai apropiat de conceptul de specialist.

În capitolele finale ale părții întâi (7-11) se aplică noțiunile învățate în primele capitole. Există numeroase “referințe încrucișate” care indică cunoștințele din capitolele 1-6 respectiv 12-18 care sunt folosite.

În spiritul dedicației, am introdus în lucrare destule note de subsol, în care am prezentat personalitățile care au produs ideile (algoritmii, structurile de date, sau teoremele matematice) din curs. Pentru fiecare personalitate, am prezentat în general unde a studiat în principal (studiile majore cu relevanță în domeniul graficii) câteva realizări legate de grafică, originea și anii de viață. Astfel de oameni ar putea constitui modele pentru studenții de azi.

Voi prezenta pe scurt, câteva aspecte/amintiri referitoare la activitatea profesorilor pe care i-am menționat în dedicație, pentru a-i aduce și pe ei la același nivel de considerație cu specialiștii a căror activitate a fost menționată în note de subsol pe parcursul lucrării.

- **Ion Gh. Sabac** (1916-1996) A fost singurul profesor pe care l-am aplaudat vreodata (împreună cu întregul an IB al Facultății de Automatică și Calculatoare) la sfârșitul desfășurării cursului său (ianuarie 1978). Am urmat cu el cursul “Matematici speciale” (3 semestre în 1976-1978) În ultimul semestru, a avut un asistent (șef de lucrări) remarcabil, meticulous, numit Valter Rudner.
- **Octavian Stănașilă** (născut în 1939) absolvent al Facultății de Matematică Universitatea București, doctor în matematică (analiza complexă) al aceleiași universități, Premiul Academiei în 1974. Profesor la Universitatea Politehnica București. (șeful catedrei matematică ii). Am urmat cu el 3 semestre de curs (“Analiza matematică”) între 1976-1978. Poate mai mult decât la alte cursuri, autorul a acordat o atenție deosebită evidențierii aplicabilității conceptelor matematice (prezentate abundent și riguros), în diverse domenii ale ingineriei (în special în domeniul calculatoarelor).
- **Eugen Onofraș** s-a născut la Huși (aproximativ. în 1930), a urmat cursurile Facultății de Matematică a Universității din Iași, Un profesor de liceu care avea cu siguranță nivelul unui bun profesor universitar (dintre profesorii buni pe care i-am avut la Liceul Mihai Viteazul din Ploiesti în perioada 1972-1976, numai profesoara de franceză Speranța Râmnicianu (Poenu) se mai ridica la acest nivel înalt). În 1976, jumătate din lotul de la Internaționala de Matematică era din clasa profesorului Onofraș. Se pricepea ca nimeni altul să conceapă/selecteze probleme/studii de caz pe care le dădea spre rezolvare elevilor și de pe urma cărora aveai mereu câte ceva nou (principiu, metodă) de înțeles/învățat.
- **Vlad Ionescu** (1938-2000) absolvent (cu media 10) al Facultății de Energetică din Institutul Politehnic București în 1959, membru al Academiei Române din 1996. Premiul Academiei în 1975. Am urmat cu el cursul de “Teoria sistemelor” (un semestru în 1978).
- **Cristian Giumale** (născut 1946), a absolvit Facultatea de Automatică din cadrul Institutului Politehnic București, doctor al Universității Victoria din Manchester (1974). Premiul Traian Vuia al Academiei Române, profesor la Catedra de Calculatoare din Universitatea Politehnica București. Am urmat cu el cursul “Structuri de date și tehnici de programare” între 1978-1979. Un profesor pasionat de profesia sa și profund în gândire.

- **Mircea Petrescu**, (născut 1933), membru onorific al Academiei Romane din 2006, absolvent al Facultății de Electronică și Telecomunicații, Institutul Politehnic București, doctorat la Institutul Energetic din Moscova. A fost profesor invitat la Berkeley, premiul academiei 1983. Profesor la catedra de calculatoare a Universității Politehnica București. Am urmat cu el cursul “Analiza și Sinteza Dispozitivelor Numerice” (două semestre în 1978) și cursul “Baze de Date” (două semestre între 1980-1981). Mi-a fost coordonator al doctoratului. Un profesor de anvergură.
- **Dan Iordache**, (născut în 1939), a absolvit Facultatea de Fizică din cadrul Universității București în 1960, unde a susținut și doctoratul (1971). Am urmat cu el cursul de “Fizică” (două semestre 1976-1977). Este probabil, profesorul care m-a ținut cel mai mult “în priză” pe toata perioada desfășurării cursului său.

15 aprilie 2016

Marius Zaharia
București
<http://mdzaharia.eu>

Marius Dorian ZAHARIA
University POLITEHNICA Bucharest

Curs elementar de grafică pe calculator
An elementary course of computer graphics

CUVANT ÎNAINTE	13
Foreword	
I ELEMENTE DE GRAFICĂ PE CALCULATOR	17
I Elements of computer graphics	
1. Noțiuni introductive	17
Introduction	
1.1 Pipeline-ul grafic conceptual	18
The conceptual graphic pipeline	
1.2 Categoriile de aplicații grafice	27
Main categories of graphics applications	
1.3 Evoluția platformelor grafice 3D (OpenGL)	29
The evolution of OpenGL graphic platform	
1.4 Structura pipeline-ului grafic la un sistem cu funcții programabile (OpenGL 4.3)	30

The structure of the “programmable functions” graphics pipeline

2. Modelarea în grafica pe calculator	34
Geometric modelling for computer graphics	
2.1 Reprezentări generale bazate pe frontiere	37
Boundary representations, general notions	
2.2 Reprezentări prin frontiere bazate pe plase poligonale	37
Polygonal mesh boundary representations	
2.2.1 Procedee de determinare a normalelor	38
Methods for normal vectors determination	
2.2.1.1 Normala la planul unui triunghi	38
Normal to a (triangle) plane	
2.2.1.2 Normala la planul unui poligon (quasi-plan) oarecare	39
Normal to an arbitrary (quasi-plane) polygon	
2.2.1.3 Calculul normalei la vârf în cazul unei plase poligonale orientate	39
Finding vertex normals in an oriented polygonal mesh	
2.3 Reprezentari simple ale unei plase poligonale	40
Simple representations of a polygonal mesh	
2.4 Reprezentarea prin muchii înaripate (winged-edge) a unei plase poligonale	42
The winged-edge representation	
2.5 Operatii de bază asupra topologiei unui model prin frontiere care folosește o reprezentare bazată pe plase poligonale	47
Basic operations to modify the topology of a polygonal mesh model (Euler operators)	
2.6 Functii atașate unei plase poligonale	49
Functions attached to a polygonal mesh (barycentric coordinates of a point relative to a triangle or to a convex polygon)	
3. Reprezentarea prin frontiere folosind suprafețe de clasă C^1 sau superioară	55
Boundary representations with surfaces of class superior to C^1 .	
3.1 Reprezentări implicite	55
Implicit representations	
3.2 Reprezentări parametrice	56
Parametric representations	
3.3 Exemple de curbe/suprafețe definite implicit/parametric	56
Examples of implicitly/parametrically defined curves/surfaces	
3.3.1 Reprezentări analitice ale dreptei	56
Analytic representations of lines	
3.3.2 Reprezentări analitice ale planului	57
Analytic representations of plane	
3.3.3 Reprezentări analitice ale suprafețelor cuadrice	58
Analytic representations of quadric surfaces	
3.3.4 Reprezentari analitice ale suprafețelor supercuadrice	62
Analytic representations of superquadrics	
3.3.5 Suprafete polinomiale parametrizate	62

	Polynomially parametrized surfaces	
3.3.6	Calculul normalei într-un punct al unei curbe/suprafețe analitice	64
	Finding the normal vector in an arbitrary point of a surface patch	
3.3.7	Alte clase de suprafețe analitice utilizate în grafica pe calculator	66
	Another classes of analytically representable surfaces (explicit representations, sweeps)	
3.4	Curbe și suprafețe polinomiale	68
	Polynomially defined curves/surfaces	
3.4.1	Curbe/suprafețe de interpolare/aproximare	69
	Interpolation/approximation curves/surfaces	
3.4.2	Exemple de curbe de interpolare	70
	Examples of interpolation curves	
3.4.2.1	Interpolarea liniara	70
	Linear interpolation	
3.4.2.2	Interpolarea Lagrange. Algoritmul lui Neville	71
	Lagrange interpolation. Neville's algorithm.	
3.4.2.3	Racordarea arcelor de Curbă folosind interpolarea Hermite (curba de interpolare Hermite de grad 3)	73
	Connection of two arcs (of curve) using Hermite interpolation	
3.4.2.4	Curbe de aproximare Bézier. Algoritmul de Casteljeau.	74
	Bézier approximation curves. The de Casteljeau algorithm.	
3.4.3	Suprafețe de aproximare Bézier	78
	Bézier approximation surfaces	
4.	Transformari grafice	79
	Graphics transformations	
4.1	Descrierea generală a transformărilor grafice	79
	General description of a graphic transformation	
4.2	Transformări grafice în spațiul 2D	80
	2D graphics transformations	
4.2.1	Transformări elementare 2D	80
	Elementary 2D transformations (translation, rotation, scaling, symmetry, shearing, homogeneous representations)	
4.2.2	Determinarea matricei de transformare pentru o transformare liniară oarecare 2D	85
	Finding the transformation matrix for an arbitrary linear transformation in 2D space (2D vector system transformation)	
4.2.3	Exemple de determinare a transformărilor compuse 2D	87
	Composing 2D graphics transformations	
4.2.3.1	Determinarea transformării de rotație 2D, cu un unghi dat, în Jurul unui punct oarecare	87
	2D rotation relative to an arbitrary centre (point)	
4.2.3.2	Transformarea de vizualizare 2D (Transformarea Fereastra-Vizor)	88
	Window-Viewport transformation	

4.3	Transformari grafice in spațiul 3D	92
	3D graphics transformations	
4.3.1	Studiul rotațiilor în spațiul 3D	94
	Study of 3D general rotations	
4.3.1.1	Parametrizarea lui Euler a unei transformări generale de rotație	96
	Euler parametrization of a general rotation	
4.3.1.2	Calculul matricei unei transformări de rotație Generală, în funcție de rotațiile parametrizării Euler	96
	Finding the matrix of a general 3D rotation, from the elementary Euler rotations	
4.3.1.3	Parametrizarea (axă - unghi) a unei transformări generale de rotație	97
	(Axis-angle) parametrization of a general 3D rotation	
4.3.1.4	Reprezentarea rotațiilor generale folosind quaternioni	99
	Representing general rotations by quaternions	
4.4	Transformări 3D aplicate normalelor	102
	Transforming normal vectors	
4.5	Exemple de determinare a transformărilor 3D compuse	104
	Composing 3D graphics transformations (examples)	
4.6	Moduri de implementare a transformărilor grafice	106
	Implementing graphic transformations	
5.	Proiecția și vizualizarea	111
	Projection and View transform	
5.1	Transformarea de proiecție	111
	Projections	
5.1.1	Deducerea relațiilor pentru transformarea de proiecție paralelă în planul xOy având direcția de proiecție dată	112
	Mathematic description of a parallel projection with a given direction, in xOy plane.	
5.1.2	Deducerea relațiilor ce exprimă transformarea de proiecție perspectivă de centru dat, în planul xOy	113
	Mathematic description of a perspective projection with a given centre, in xOy plane	
5.1.3	Deducerea relațiilor ce exprimă transformarea de proiecție perspectivă în planul de ecuație $z=-1$ și având centrul de proiecție în origine	115
	Perspective projection with centre in origin, in plane $z=-1$.	
5.2	Transformări proiective	116
	Projectivities	
5.3	Clase de transformări ale spațiului 3D	117
	Classes of transformations in 3D space	
5.4	Vizualizarea 3D	119
	3D viewing transformation	
5.4.1	Transformarea de vizualizare în modelul camerei perspectivă	120
	Perspective camera model	

5.4.2	Transformarea de vizualizare în modelul camerei ortografice ... Ortographic camera model	125
5.5	Matricele de transformare folosite efectiv în pipeline-ul grafic OpenGL The graphics transformations matrices used in OpenGL graphics pipeline.	127
6.	Decuparea geometrică	129
	Clipping	
6.1	Generalități	129
	Generalities	
6.2	Algoritmul de decupare analitică Cohen-Sutherland în spațiul 2D	131
	2D Cohen-Sutherland clipping algorithm	
6.3	Algoritmul de decupare Cohen-Sutherland în spațiul 3D	134
	3D Cohen-Sutherland clipping algorithm	
6.4	Algoritmul de decupare Poligonală Sutherland-Hodgman în spațiul 2D	137
	2D Sutherland-Hodgman, polygonal clipping algorithm	
6.5	Algoritmul Sutherland-Hodgman în spațiul 3D	142
	3D Sutherland-Hodgman polygonal clipping algorithm	
7.	Metode clasice de sinteză realistă de Imagini:	
	Drumul optic (ray-tracing) și rasterizarea	143
	Methods for realistic image synthesis: ray-tracing and rasterization	
7.1	Descriere generală	143
	Generalities	
7.2	Implementarea nucleului unui sistem de sinteză realistă de imagini prin metoda ray-tracing	145
	A classical ray-tracing engine	
7.2.1	Fisierele header și rutina main()	146
	Headers and main()	
7.2.2	Rutine folosite în calculul de vizualizare 3D	152
	3D viewing routine	
7.2.3	Calculul general de intersecție (rază - obiect)	154
	General ray-object intersection	
7.2.4	Rutina de shading	156
	Shading routine	
7.2.4.1	Descrierea modelului de reflexie Phong	156
	The Phong empirical reflexion model	
7.2.5	Rutine asociate surselor luminoase	161
	Light sources routines	
7.2.6	Rutine pentru calculul umbrei, reflexiilor multiple interobiecte și transparenței (raze de ordin superior)	162
	Shadows computation, higher order reflexions and transmissions.	
7.2.6.1	Tratarea reflexiei multiple și a refracției	162
	Multiple reflexion/transmission treatment	
7.2.6.2	Tratarea umbrelor	165
	Shadow treatment	
7.2.7	Rutine de intersecție și calcul normale, asociate diferitelor categorii de obiecte	167
	Specialized ray-object intersection methods	

7.2.7.1	Obiectul sferă	167
	Sphere object	
7.2.7.2	Obiectul paralelipiped aliniat cu axele referențialului (box)	169
	Axis aligned box object	
7.2.7.3	Obiectul triunghi	172
	Triangle object	
7.2.8	Rutine pentru salvarea imaginii rezultat	174
	Saving the output image	
7.2.9	Cuantizarea culorilor	176
	Color quantization	
7.2.10	Rutine auxiliare	177
	Auxiliary routines	
7.3	Rasterizarea	178
	Rasterization	
7.3.1	Maparea continuu-discret (screen mapping)	178
	Screen mapping	
7.3.2	Algoritmi de generare a primitivelor grafice 2D în spațiul discret	179
	Rasterization of 2D primitives	
7.3.2.1	Algoritmul Bresenham pentru rasterizarea liniilor (vectorilor) în spațiul discret	179
	Bresenham's algorithm for lines/vectors drawing	
7.3.2.2	Algoritmul Gupta-Sproull pentru afișarea cu antialiasing a vectorilor în spațiul discret	183
	Gupta-Sproull's algorithm for antialiased lines drawing	
7.3.2.3	Rasterizarea triunghiurilor în spațiul 2D	187
	2D triangles rasterization	
7.3.3	Tratarea rasterizării 3D	189
	Treatment of 3D rasterization	
8.	Modele de culoare, iluminare și împrăștierea luminii	194
	Color models, lighting and light scattering	
8.1	Notiuni generale referitoare la culori	194
	General notions regarding colours	
8.2	Exemple de modele de culoare	198
	Colour models examples (RGB, CMY, YIQ, HSV, CIEXYZ, CIEL*u*v*)	
8.3	Împrăștierea luminii	206
	Light scattering	
8.3.1	Distribuții bidirecționale care descriu împrăștierea luminii	208
	Bidirectional distributions that model light scattering (types of scattering, parametrization of BSDFs)	
8.3.2	Descrierea modelelor de reflexie în termenii distribuțiilor BSDF	214
	Describing reflexion models in BSDFs terms	
8.4	Modele de iluminare, Surse luminoase	222
	Lighting models, light sources (types of light sources, lighting equation)	
9.	Textura	227
	Texturing	
9.1	Generalități	227

Generalities	
9.2 Maparea unei texturi bidimensionale pe suprafața unui obiect	229
Mapping bi-dimensional textures on a surface	
9.2.1 Maparea texturii pe suprafețe plase poligonale	231
Texture mapping on a polygonal map	
9.2.1.1 Metode de mapare globală (maparea în Două etape)	231
Two pass mapping, global mapping pass	
9.2.1.2 Maparea locală a texturilor în cazul plaselor poligonale	236
Two pass mapping, local mapping pass	
9.3 Texturarea prin aplicarea imaginii contextului unui obiect pe suprafața obiectului	237
Environment mapping	
9.4 Texturarea prin perturbarea direcției normalei la suprafața unui obiect	238
Bump mapping/Displacement mapping.	
9.4.1 Determinarea vectorilor bază ai spațiului tangent asociat unui vârf al unei suprafețe modelate printr-o plasă de triunghiuri	240
Finding the basis vectors of the tangent space associated to an arbitrary vertex of a triangle map.	
9.5 Filtarea texturilor	242
Texture filtering (nearest neighbour, bilinear interpolation, mip-map, trilinear interpolation)	
10. Eliminarea suprafețelor ascunse	247
Hidden surfaces removal	
10.1 Algoritmul back-face culling	248
Back face culling	
10.2 Algoritmul z-buffer	249
Z-buffer algorithm	
10.2.1 Codificarea informației stocate într-un z-buffer	252
Encoding the information in z-buffer's elements	
10.2.2 Algoritmul A-buffer	253
A-buffer algorithm	
10.3 Arborii BSP (partiționarea binară a spațiului)	255
BSP trees (hidden surfaces removal through back to front BSP tree traversal)	
10.3.1 Construirea unui arbore BSP	257
Building a BSP tree	
11. Studiul erorilor care apar în procesul de redare a imaginilor.	
Aliasing	262
Rendering errors due to aliasing	
11.1 Studiul teoretic al aliasingului	262
Theoretical study of aliasing (signal types, sampling/reconstruction, convolution, space/frequency domain, Shannon-Borel-Nyquist theorem)	
11.2 Metode de tratare a aliasingului în grafica pe calculator	

(procedee de antialiasing)	272
Methods for antialiasing in computer graphics	
II ELEMENTE DE GEOMETRIE ALGORITMICĂ	281
II Elementary computational geometry for computer graphics	
12. Structuri de date caracteristice geometriei algoritmice 2D	
folosite în grafica pe calculator	282
Data structures for computational geometry	
12.1 Tipul de date abstracte punct (2D)	283
The "2D point" abstract data type	
12.2 Tipul de date abstracte muchie (2D)	287
The "2D edge" abstract data type	
12.3 Tipul de date abstracte poligon (2D)	289
The "2D polygon" abstract data type	
13. Predicate geometrice	294
Geometric predicates	
13.1 Verificarea convexității unui poligon	294
Convexity of a polygon	
13.2 Pozitia unui punct în raport cu un poligon	295
Position of a point relative to a polygon (interior/exterior)	
14. Algoritmi de determinare a înfășurătorii convexe, asociate	
unei mulțimi de puncte	300
2D Convex Hull problem	
14.1 Metoda împachetării	301
Gift-wrapping (Jarvis) method	
15. Triangulații planare	305
2D Triangulations	
15.1 Noțiuni generale despre triangulații	305
General notions about triangulations	
15.2 Triangulația unui poligon simplu	309
Triangulation of a simple polygon	
15.3 Triangulația Delaunay și diagrama Voronoi	314
Delaunay triangulation and Voronoi diagram	
15.3.1 Triangulația Delaunay	314
Delaunay triangulation	
15.3.2 Diagrama Voronoi	319
Voronoi diagram	
16. Tipuri de date abstracte pentru modelare geometrică	
în spațiul 3D	325
Abstract data types for 3D geometric modelling	
16.1 Tipul de date abstracte punct3D	325
"3D point" abstract data type	

16.2	Tipul de date abstracte muchie3D "3D edge" abstract data type	327
16.3	Tipul de date abstracte triunghi3D "3D triangle" abstract data type	329
16.3.1	Determinarea intersecției unei muchii (dreaptă infinită) cu interiorul unui triunghi3D Finding the intersection between an infinite 3D edge and the interior of a 3D triangle	330
17.	Sugestii privind implementarea în C++ a TDA-urilor legate de modelarea geometrică 3D	333
	Suggestions regarding the C++ implementation of 3D geometric modelling abstract data types	
17.1	Clasa punct (2D) Class "2D point"	334
17.2	Clasa muchie (2D) Class "2D edge"	336
17.3	Clasa poligon (2D) Class "2D polygon"	337
17.4	Clasa punct3D Class "3D point"	339
17.5	Clasa muchie3D Class "3D edge"	340
17.6	Clasa triunghi3D Class "3D triangle"	340

..

III MATEMATICA

III Mathematics

18.	Noțiuni de matematică folosite în studiul graficii pe calculator	341
	Notions of mathematics used in computer graphics	
18.1	Structuri algebrice Algebraic structures	341
18.2	Spații matematice folosite în grafica pe calculator Mathematic spaces used in computer graphics	344
18.2.1	Spatiul vectorial Vector space	344
18.2.2	Spatiul metric Metric space	344
18.2.3	Spațiul Euclidian Euclidean space	348
18.2.4	Spațiul afin Affine space	349
18.2.5	Spațiul Grassmann și spatiul proiectiv Grassman space and projective space	351
18.2.6	Spatiul topologic Topological space	352

18.3	Relatie	353
	Relation	
18.4	Funcții și distribuții	354
	Functions and distributions	
18.4.1	Notiunea de distribuție	355
	The notion of distribution	
18.5	Tensor.....	358
	Tensor	
18.6	Reprezentarea funcțiilor (semnalelor) prin serii	361
	Representing signals through series	
18.6.1	Serii Fourier	366
	Fourier series	
18.7	Alte concepte/procedee matematice întrebuintate în acest curs	368
	Other mathematical concepts used in this course (inverse matrix, differential geometry, Jacobian, Plucker coordinates, lattice)	
IV ANEXE		373
IV Annexes		
A1. Exemplu de aplicație OpenGL simplă implementată în arhitecturile OpenGL 4.3 (pipeline-ul funcțiilor programabile) și OpenGL 1.1 (pipeline-ul funcțiilor fixe)		373
A simple application implemented in programmable functions pipeline (OpenGL 4.3) and fixed functions pipeline (OpenGL 1.1).		
A1.1	Aplicația OpenGL 4.3	373
A1.2	Aplicația OpenGL 1.1	383
A2. Shading interpolativ implementat în OpenGL 4.3 (pipeline-ul funcțiilor programabile)		386
Interpolative shading implemented in OpenGL 4.3		
A2.1	Implementare shading interpolativ conform metodei Gouraud	386
	Gouraud method (in vertex shader)	
A2.2	Implementare shading interpolativ conform metodei Phong	388
	Phong method (in fragment shader)	
A3. Subiecte de Examen 2015-2016		391
Exams subjects (2015-2016 summer session)		
A4. Noțiuni de analiza algoritmilor		400
Notions of algorithms analysis		
Bibliografie		404
References		

