

**MODELAREA COMPUTERIZATĂ A STRUCTURII
OSOASE FACIALE ÎN VEDEREA PROTEZĂRII ȘI
INTERVENȚIILOR CHIRURGICALE REPARATORII**

Teză de doctorat

Autor:

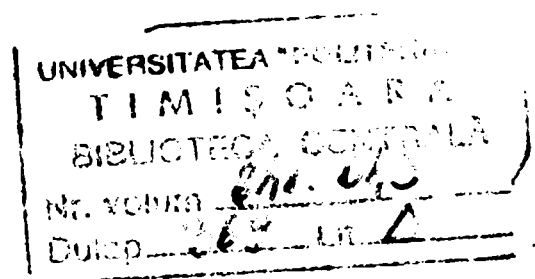
ing. Virgilius Stănciulescu

Conducător științific:

prof.dr.ing. Doina Drăgulescu

**m. c. al Academiei de Științe
Tehnice din România**

BIBLIOTECA CENTRALĂ
UNIVERSITATEA "POLITEHNICA"
TIMIȘOARA



Timișoara

2003

CUPRINS

1. INTRODUCERE	1
1.1 Modelarea și grafica computerizată în sisteme mecanice și biomecanice	1
1.2 Considerații generale	4
2. CONCEPTUL DE MODEL ȘI MODELARE	10
2.1 Conceptul de model	10
2.1.1 Definiție	10
2.1.2 Calitatea și clasificarea modelelor	12
2.2 Conceptul de modelare	18
3. METODE PENTRU REALIZAREA FORMELOR COMPLEXE ÎN MODELAREA GEOMETRICĂ	19
3.1 Tehnici de descriere a modelelor 3D	19
3.1.1 Reprezentarea geometric constructivă a solidelor (RGC)	21
3.1.2 Reprezentarea prin frontiere (RF)	22
3.1.3 Reprezentarea prin enumerare spațială (RES)	23
3.1.4 Reprezentarea prin baleiere (RB)	24
3.2 Reprezentarea obiectelor utilizând curbe	25
3.2.1 Curbe spline	26
3.2.2 Curbe Bezier	27
3.2.3 Curbe B-spline	30
3.3 Obținerea de suprafețe pornind de la curbe	32
3.3.1 Aplicație biliniară	33
3.3.2 Lofting	34
3.3.3 Suprafețe de tip Coons	35
3.3.4 Petice bicubice	36
3.3.5 Petice Bézier	37
3.3.6 Suprafețe capac	37
3.3.7 Suprafețe de revoluție	38

3.3.8 Suprafețe extrudate	39
3.3.9 Sweeping	39
3.3.10 Suprafețe offset	40
3.3.11 Suprafețe booleene și ajustate/tăiate	41
4. VIZUALIZAREA ȘI TRANSFORMAREA MODELELOR 3D	43
4.1. Procedura de vizualizare	44
4.2. Proiecții	46
4.2.1. Proiecția perspectivă	48
4.2.2. Proiecția paralelă	50
4.3 Transformări grafice aplicate modelelor	52
4.4. Redarea realistă a imaginilor	55
4.5 Interfața software OpenGL	58
4.5.1. Descrierea structurii unui program bazat pe OpenGL și a comenzilor de bază	60
5. RECONSTITUIREA ȘI MODELAREA 3D A STRUCTURILOR ANATOMICE COMPLEXE	64
5.1 Importanța și aplicabilitatea	64
5.2 Principiul și etapele soluției propuse	65
5.3. Citirea datelor	68
5.4 Selectarea zonelor de interes	71
5.4.1 Detecția conturului	72
5.4.2 Detecția regiunilor omogene	79
5.5 Filtrul volumetric de detecție a izosuprafețelor 3D	82
5.5.1 Divizarea volumului	83
5.5.2 Determinarea intersecțiilor dintre suprafață și voxelul curent	86
5.5.3 Determinarea suprafeței	97
5.5.4 Coerența	98
5.5.5 Rezoluția	100
5.5.6 Valori de prag multiple	101
5.5.7 Disecție virtuală	102
5.6 Reconstrucția 3D a structurilor ilor anatomice	104

6. CONSIDERAȚII ANATOMICE	128
6.1 Oasele capului	128
6.2 Articulațiile	132
6.3 Proprietăți mecanice ale sistemului osos	132
6.4 Mușchii aparatului dentomaxilar	133
7. MODELAREA ȘI ANALIZA STRUCTURILOR MECANICE REPARATORII UTILIZATE ÎN CHIRURGIA MAXILO-FACIALĂ	140
7.1 Considerații asupra modelării structurilor mecanice și analiza lor cu elemente finite	140
7.2 Metoda analizei cu elemente finite	143
7.3 Considerații asupra mecanismului de producere a fracturilor și a structurii de rezistență a craniului	145
7.4 Intervenții chirurgicale reparatorii la nivel osos cu ajutorul placulelor metalice și implantelor	146
7.5. Modelarea 3D a implanturilor pornind de la modelul reconstruit al structurii osoase	147
7.6 Analiza cu elemente finite a structurilor utilizate cu scop reconstructiv	149
7.6.1 Fractura de mandibula în plan vertical	149
7.6.2 Imobilizarea unei fracturi cu două plăcuțe diferite	156
7.6.3 Fractură cominutivă	161
7.6.4 Utilizarea miniplăcuței 3D	164
7.6.5 Cazul liniilor de fractură neparalele	167
7.6.6 Placa pentru fracturi cominutive majore dispersate	170
7.6.7 Placă pentru fracturi cominutive minore dispersate	173
8. CONTRIBUȚII ORIGINALE	177
BIBLIOGRAFIE	183

1. INTRODUCERE

1.1 Modelarea și grafica computerizată în sisteme mecanice și biomecanice

Obiectivul oricărui sistem de modelare computerizată este acela de a crea modele cât mai aproape de lumea reală, atât prin caracteristici vizuale cât și tehnice și funcționale.

Modelarea computerizată și grafica 3D au ca scop realizarea de modele ale corpurilor reale, prin proiectare asistată de calculator, modele utilizabile ulterior pentru studiu, și reprezentarea acestora prin imagini realiste, care să redea cât mai fidel proprietățile modelului.

Astfel, modelarea computerizată și grafica tridimensională sunt puncte unde se întâlnesc în slujba noastră majoritatea științelor pentru a studia lumea fizică, formele și reprezentările din natură. Modelarea computerizată, acest domeniu al tehnologiei informației, transformă în mod substanțial modul de studiu, lucru, modul de luare a deciziilor în urma simulărilor și chiar modul de fabricație a ansamblelor. Utilizatorul interacționează cu o lume tridimensională, mai aproape de lumea reală.

Modelarea tridimensională și reprezentarea grafică a modelelor constituie practic nucleul programelor pentru CAD/CAM. Având la dispoziție imagini tridimensionale, inginerul poate proiecta subansamble și poate verifica modul în care se potrivesc diferitele componente ale unui subansamblu, modul în care mai multe echipamente funcționează într-un complex de echipamente, fără a fi necesară realizarea fizică a acestora. În proiectarea asistată de calculator, inginerii proiectanți pot realiza, studia, modifica, testa în mod foarte eficient produsele la care lucrează. Tehnici specifice sunt folosite intens și în industria cinematografică pentru efecte speciale de un realism deosebit.

Un domeniu aparte, prezentând o mare importanță, este cel al aplicațiilor de simulare și modelare în domeniul biomedical și biomecanic, care permit evitarea investigațiilor invazive și oferă în același timp și soluții de reconstrucție a organelor deteriorate.

Aplicabilitatea în medicină se bazează pe prelucrarea informațiilor obținute prin metode specifice. Aria de utilizare se întinde de la statistica medicală până la prelucrări complexe de imagini ce permit extragerea de informații suplimentare de o mare valoare. În domeniul biomedical, sistemele de prelucrare automată a imaginilor au pătruns cu un succes deosebit, impunându-se tehnici din ce în ce mai avansate și mai performante, care își aduc aportul în analiza și diagnosticarea corectă. Dintre tipurile de investigații medicale bazate pe imagistică (figura 1-1), ajutate în mod deosebit de tehnica de calcul (figura 1-2) pot fi menționate[32][43][86][99][124][197]:

- explorările funcționale scintigrafice;
- ecografia;
- tomografia computerizată;
- radiografia;
- microscopia cantitativă;
- rezonanța magnetică nucleară.

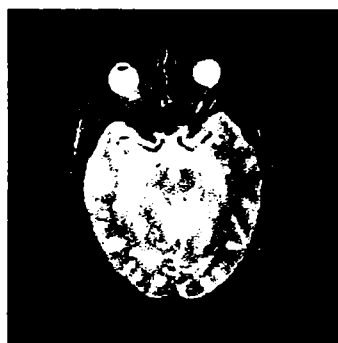


Fig.1-1: Tipuri de imagini din imagistica medicală

În general noile sisteme medicale sunt interfațate cu un calculator (figura 1-2), iar dezvoltarea de aplicații rămâne deschisă oricăror noi idei ce pot fi puse în slujba sănătății.



Fig.1-2: Sistem medical de investigație integrat cu tehnica de calcul

S-a ajuns la rezultate deosebite și chiar la diferențierea pe domenii specifice a aplicațiilor în domeniul biomedical[86][197][124]:

- imagistica medicală;
- imagistica funcțională cardiovasculară;
- imagistica computerizată în domeniul maxilofacial;
- telemedicina;
- diagnoza asistă de calculator, prin *sistemele expert*;
- terapia radiantă asistată de calculator;
- instrumentație și robotică chirurgicală;
- simularea și instruirea în domeniul chirurgical;
- chirurgie ghidată imagistic;
- proiectarea de proteze externe;
- modelarea deficiențelor sistemelor osoase prin dispozitive specifice (figura 1-3);

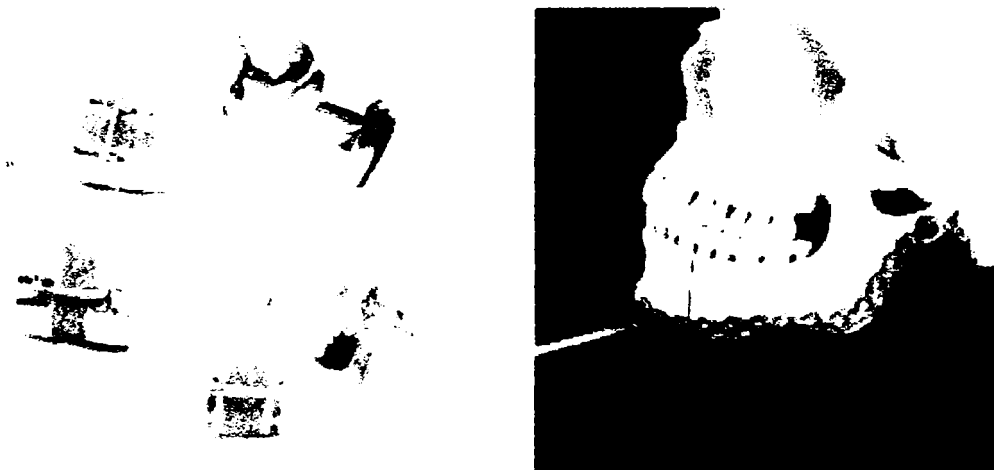


Fig.1-3: Mijloace de corectare a deficiențelor la nivelul mandibulei

Utilizarea tehnicii de calcul în aceste domenii permite evaluări calitative și cantitative ale unor procese fiziologice, iar dezvoltarea în ritm alert a sistemelor de calcul și a tehnicilor de prelucrare și interpretare a imaginilor constituie o garanție pentru obținerea de rezultate din ce în ce mai performante în domeniile de aplicație.

Dar, unul din cele mai spectaculoase și utile domenii în care calculatorul și proiectarea asistată își găsește aplicația este cel al modelării structurilor biologice, țesuturi dure sau moi, os sau piele, sisteme osoase ce se supun legilor fizicii și mecanicii, organe investigate, precum și cel al analizei și proiectării implantelor.

Cu ajutorul tomografiei asistate de calculator (figura 1-4) se poate acum aborda domeniul chirurgiei reconstructive: pacientul este supus analizelor, rezultând un set de tomografii computerizate, care nu mai sunt interpretate ca atare ci supuse unui șir de prelucrări specifice cu ajutorul calculatorului generându-se o reprezentare cât mai aproape de realitate a organului investigat. Această reprezentare poate fi utilizată într-un mediu virtual de către chirurghi pentru vizualizarea preoperativă a deficiențelor organului asupra căruia urmează să se intervină, planificarea intervenției, și, eventual modelarea și producerea unui implant sau substituent dacă este cazul.



Fig. 1-4: Tomograf computerizat

1.2 Considerații generale

Teza de doctorat *Modelarea computerizată a structurii osoase faciale în vederea protezării și intervențiilor chirurgicale reparatorii* încearcă să îmbine specificul proiectării asistate de calculator a corpurilor și subansamblelor mecanice cu specificul utilizării calculatorului în domeniul medical pentru prelucrare de informații și imagini, extragere de date esențiale și modelare.

Teoriile aplicate și algoritmi utilizați, rezultatele obținute, atestă interdisciplinaritatea lucrării aflată la frontiera dintre ingineria software și ingineria mecanică cu aplicații în domeniul biomedical. Tehnicile utilizate cuprind un domeniu extins de la algoritmi software clasici și originali (proprii autorului) din domeniul graficii computerizate și modelării, până la metoda analizei cu elemente finite a structurilor

mecanice, conducând la stabilirea definitivă a propunerilor de implante pentru anumite deficiențe ale scheletului facial.

S-a urmărit dezvoltarea unui sistem de reconstrucție și vizualizare 3D a structurilor de țesut, a organelor pentru care există descrieri 2D rezultate în urma analizelor, sistem ce va oferi posibilitatea unei analize mult îmbogățite din punct de vedere al informației pentru orice organ sau sistem anatomic uman, precum și proiectarea, modelarea și analiza implantelor și structurilor mecanice specifice necesare în intervențiile chirurgicale reparatorii și de protezare la nivelul feței.

Aceste obiective implică mai multe etape. În esență este vorba despre:

- realizarea analizelor specifice, tomografii;
- extragerea de informație utilă din acestea;
- interpretarea și prelucrarea datelor în mod specific conform obiectivelor propuse, pe baza unor algoritmi originali;
- generarea unui model al corpului de proiectat care să răspundă cerințelor;
- reprezentarea grafică pe monitor în mod foarte realist și cu cele mai fine și exacte detalii;

Caracteristica cea mai importantă este aceea că metoda nu numai că permite reconstrucția tridimensională a unui organ sau sistem anatomic, ci permite chiar disecția “virtuală” a acestuia, prin posibilitatea de vizualizare la diferite nivele de “adâncime”, și prin posibilitatea de a face diferența între diferitele părți componente ale unui sistem anatomic (figura 1-5).



Fig.1-5: Modele 3D ale capului evidențiind tipuri diferite de țesuturi

Un sistem anatomic remodelat tridimensional poate fi ușor identificat ca o structură mecanică extrem de complexă. Această caracteristică este utilizată atât pentru proiectarea de mici structuri mecanice reparatorii, cât și pentru analiza comportării lor prin metoda elementelor finite.

Lucrarea propune deci metode software moderne și originale pentru modelarea 3D a corpurilor pentru care există informații bidimensionale specifice imagisticii biomedicale, și metode de analiză și proiectare de structuri mecanice reparatorii pentru corpuri astfel modelate.

Lucrarea este structurată în opt capitole care prezintă în mod gradat problema, precum și tehnologiile ce intervin în rezolvarea ei, soluția aleasă, optimizări, experimente și concluzii.

Capitolul 1, *Introducere*, încearcă o scurtă trecere în revistă a posibilităților oferite de proiectarea asistată de calculator. Se poate observa că, dacă inițial calculatorul a fost folosit în mod special în inginerie, existând o mulțime de aplicații în cele mai variate domenii ingineresti și ale științelor exacte, în timp au apărut aplicațiile utilizate în domenii delicate cum ar fi cel biomedical și biomecanic. Dar acest lucru nu este surprinzător și poate fi considerat chiar o consecință a faptului că ființele vii reprezintă, dintr-un anumit punct de vedere, sisteme ce se supun legilor mecanicii clasice. În momentul în care aceste legi aplicate ajută la studiul ingineresc al sistemelor ce alcătuiesc ființa vie (de exemplu sistemul osos) și la recuperarea acestora pe baze științifice, atunci devine clar că trecerea de la domeniul ingineresc la cel biomedical devine o necesitate.

Capitolul 2, *Conceptul de model și modelare*, prezintă pe larg conceptele de model și modelare, deoarece obiectivul lucrării este acela de a modela părți osoase, sau moi, aparținând diverselor organe, modele ce trebuie să satisfacă cerința de a fi cât mai aproape de realitate. Definind modelul, se poate spune că acesta este o reprezentare a aspectelor esențiale ale unui sistem. Sunt definite noțiuni de bază și aspectele de care trebuie să se țină seama în definirea unui model, avantajele utilizării modelelor și modul de exprimare și apreciere a calității acestora. Este abordată și o clasificare a modelelor în funcție de materialitate, de forma de prezentare și aspect. Tot în acest capitol se definește conceptul de modelare și se realizează o trecere în revistă a etapelor parcurse în

activitatea de modelare. Un element de noutate este abordarea modelării părților anatomice, sisteme extrem de complexe, la care fiecare detaliu contează și este dificil să se realizeze o abstractizare.

Capitolul 3, *Metode pentru realizarea formelor complexe în modelarea geometrică*, abordează teoretic și practic, pe rând: tehnicile de descriere a modelelor geometrice tridimensionale, tipurile de reprezentări, metodele de creare a modelelor solide. Se observă că în domeniul biomecanic o combinație a tehnicii de reprezentare prin enumerare spațială și a unei metode booleene, constituie o soluție viabilă.

Capitolul 4, *Vizualizarea și transformarea modelelor 3D*, prezintă succesiuni de proceduri ce pot fi aplicate modelelor pentru a fi vizualizate după dorință. Sunt abordate tipurile de afișări și procedura de vizualizare. Sunt explicate detaliat transformările ce se aplică modelelor în spațiul tridimensional: scalare, rotire, translatare. Proiecțiile sunt explicate deasemenea pe larg: proiecția perspectivă cu unul, două sau trei puncte de convergență, proiecții paralele ortografice, proiecții paralele oblice. Este prezentată în detaliu metoda *ray-tracing* de redare realistă a scenelor 3D. În finalul capitolului, este descrisă librăria grafică OpenGL, care stă la baza tuturor algoritmilor importanți utilizați în teză. Această interfață și librărie software reprezintă o colecție de câteva sute de proceduri și funcții care permit programatorului să specifice obiectele și operațiile pe care le dorește, pentru a produce scene tridimensionale proiectate în imagini de înaltă calitate. Orice sistem modern actual folosește o librărie standardizată ce oferă suport multiplatformă. În acest fel efortul, poate fi concentrat pe esența algoritmilor și pe partea de cercetare propriu-zisă, favorizându-se obținerea de progrese reale.

Capitolul 5, *Reconstituirea și modelarea 3D a structurilor anatomice complexe*, descrie pe larg metoda de obținere a modelelor tridimensionale din seturi de date specifice imagisticii medicale. Sunt prezentate caracteristicile imagisticii medicale și rezultatele ei, ce constituie date de intrare în sistemul de reconstituire. Sunt descrise etapele ce trebuie urmate: achiziția imaginii, stabilirea necesității operației de segmentare și metodele de segmentare și clasificare a obiectelor din imagini, construirea setului volumetric. Pe acest set volumetric, în urma unor prelucrări specifice se va aplica un filtru de contur spațial original ce detectează izosuprafețele, adică suprafețele caracterizate de aceeași valoare a unui parametru (în acest caz densitatea țesutului). Algoritmul filtrului de detecție a izosuprafețelor este descris în amănunt, împreună cu

codul programului și optimizările originale ce îi conferă viteză, exactitate, versatilitate. Sunt descrise divizarea volumului și modul de decizie a apartenenței sau nu a unei entități la suprafața căutată, adică cea care se dorește a fi extrasă și modelată. Deasemenea, sunt prezentate experimente și rezultate obținute în reconstrucția de părți anatomice complexe prin metodele prezentate. Sunt prezentate imaginile sursă, datele reale avute la dispoziție, modul în care s-a extras informația utilă din ele și suprafața de țesut osos sau moale reconstituită, suprafață ce poate fi un anumit organ sau o porțiune de schelet: craniul în ansamblu, mandibula, maxilarul, coloana vertebrală, coastele, pelvisul, oasele mâinilor sau ale picioarelor. Sunt prezentate modelele reconstruite în diverse ipostaze inaccesibile altfel, precum și posibilitățile de export în programe de prelucrare CAD/CAM specifice pentru operațiuni ulterioare. Practic de la imagine se ajunge la un corp tridimensional posibil a fi analizat și utilizat cu perspective mult îmbunătățite.

Capitolul 6, *Considerații anatomice*, este rezervat considerațiilor de ordin anatomic și descrie oasele și mușchii capului. Justificarea este aceea de a înțelege pe parcurs că de fapt, capul, ca și restul trupului reprezintă un sistem mecanic, care interacționează cu mediul și asupra căruia se acționează dinamic și static. Acesta este de fapt și elementul de originalitate, care va fi reliefat în capitolele următoare, acela de a asimila un sistem biologic unui sistem mecanic.

Capitolul 7, *Modelarea și analiza structurilor mecanice reparatorii utilizate în chirurgia maxilo-facială*, prezintă consecințe ale rezultatelor obținute, în sensul că, având la dispoziție modele 3D, se pot analiza metodele de remediere a defecțiunilor acestora, înainte de a fi aplicate pe obiectul real. Particularizarea este făcută pe oasele craniului, capitolul cuprinzând considerații privind analiza și modelarea implantelor și a structurilor mecanice reparatorii utilizate în chirurgia maxilo-facială, asupra mecanismelor de producere a fracturilor și a structurii de rezistență a craniului. Deasemenea, sunt prezentate caracteristici ale reparațiilor chirurgicale la nivel osos cu miniplăcuțe metalice și o metodă de modelare a implantelor de formă naturală. Este prezentată analiza cu elemente finite a structurilor utilizate cu scop reconstructiv. Analiza este realizată cu ajutorul unui program specializat, fiind abordate diverse situații reale. Sunt analizate comportările diferitelor tipuri de plăcuțe aplicate pe os, la interacțiunea cu diverse forțe.

Lucrarea se încheie cu capitolul 8, *Concluzii și elemente de originalitate*, în care sunt sintetizate rezultatele obținute și elementele originale aduse de autor în cadrul tezei.

Părțile experimentale ale tezei au fost posibile datorită accesului asigurat pentru doctorand în cadrul Bazei de Cercetare cu Utilizatori Multipli *Centrul de Modelare a Protezării și Intervențiilor Chirurgicale asupra Scheletului Uman*, și a colaborării sale ca membru în colectivele mai multor granturi C.N.C.S.I.S. și P.N.C.D.I. [216][217][218].

2. CONCEPTUL DE MODEL ȘI MODELARE

2.1 Conceptul de model

2.1.1 Definiere

Modelul se consideră a fi un sistem abstract sau material cu ajutorul căruia pot fi studiate indirect proprietățile unui alt sistem mai complex (sistemul sursă), cu care modelul prezintă o analogie limitată și orientată [10][46][71][115][116][117][118].

Modelul este o reprezentare a aspectelor esențiale ale unui sistem (existent sau virtual).

Necesitatea construcției modelului este determinată, de cele mai multe ori, de necesitatea studiului, construcției sau conducerii unui sistem (real sau virtual), în condițiile imposibilității efectuării acestor activități asupra sistemului dat, din motive economice, ecologice, de complexitate, de accesibilitate, de periculozitate etc.

Sistemul real sau virtual, de la care se pornește și care reprezintă obiectul investigației este sursa informațiilor necesare modelării și este numit **sistem sursă**.

Definirea modelului trebuie să țină seama de următoarele:

- oglindirea trebuie făcută printr-o reprezentare *simplificată*, oferind deci și posibilități de studiu mai simple pentru beneficiarul modelului; gradul de simplificare trebuie privit prin prisma posibilității de *construire* a modelului, precum și prin cea a posibilității de *utilizare* a acestuia în scopul propus;
- în cadrul acestei reprezentări, deseori se consideră necesară relevarea *trăsăturilor caracteristice*, care diferențiază sistemul studiat de alte sisteme;
- reținerea doar a acelor trăsături care sunt *adecvate scopului*.

În definirea modelelor biomecanice (figura 2-1) trebuie luate în calcul toate aceste aspecte în funcție de situație. Modelul unei proteze sau al unui implant trebuie să pună la

dispoziție informații legate de formă (în cazul unui implant de ordin estetic), de funcționare și interacțiune cu alte organe (în cazul unui aparat care va ajuta la funcționarea corectă a unui organ), sau atât de formă cât și de funcționalitate. Însă, în general, este indicat să se evidențieze toate aspectele legate de: formă, proprietăți fizice, chimice și proprietăți funcționale.

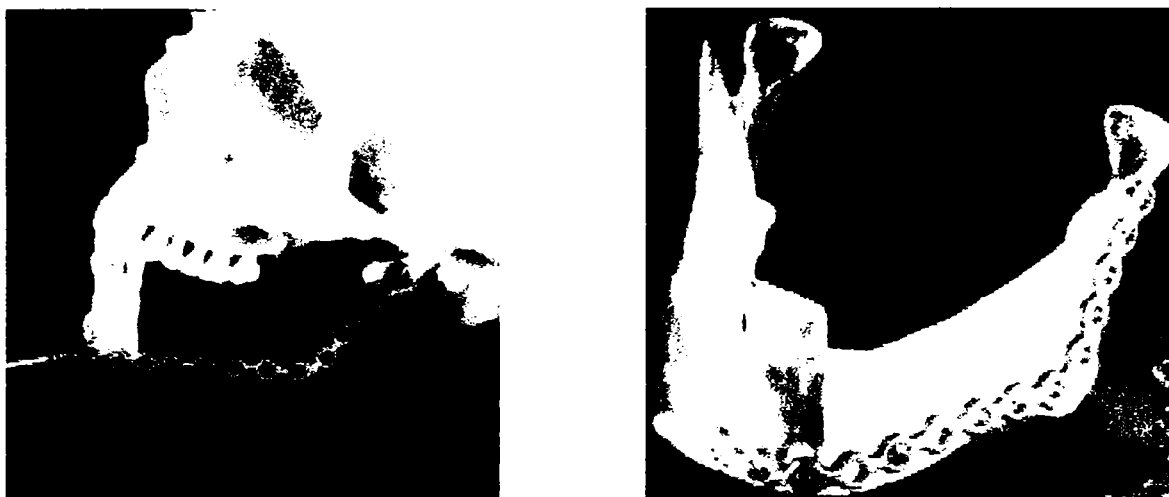


Fig.2-1 Modele de implante corective pentru mandibulă

Construirea modelelor și simularea se constituie în metode moderne, cvasigeneralizate de cunoaștere a unor sisteme sursă, cu tehnici, procedee și mijloace specifice.

Modelele și tehnicile de modelare sunt utilizate în toate domeniile științifice, cu atât mai pregnant în cele interdisciplinare, unde de multe ori rezolvarea problemelor se poate realiza doar prin modelare. Generalizarea utilizării modelelor se datorează, în principal, următoarelor avantaje:

- oferă studiului o accesibilitate mai ridicată decât a sistemelor sursă;
- necesită costuri mai mici pentru realizare și testare;
- necesită mai puțin timp pentru realizare și testare;
- schimbările în structura modelului sunt ușor de implementat;
- schimbările de comportament sunt mai simplu de izolat, înțeles și comunicat;
- permit, fără probleme, experimentări, indiferent de pericolozitatea mediului în care ființează sistemul sursă;

- pot oferi informații și asupra unor sisteme imaginate care nu se pot realiza sau executa prin tehnologiile existente; astfel operațiunea de modelare este de cele mai multe ori deschizătoare de drumuri în domenii de complexitate ridicată.

2.1.2 Calitatea și clasificarea modelelor

Calitatea unui model este exprimată, în primul rând, de **fidelitatea** cu care acesta reproduse comportamentul și aspectele cunoscute ale sistemului modelat. Comportamentul modelului este relevat prin funcționarea sa, care reprezintă **simularea** funcționării sistemului original. Fidelitatea simulării depinde de următoarele aspecte [165][115][116][117][135][139][140]:

- corectitudinea cu care s-a efectuat aprecierea aspectelor esențiale, relevante pentru scopul urmărit;
- procedeele și tehnicile de constituire a modelului;
- calitatea și cantitatea cunoștințelor disponibile posibil a fi utilizate de către modelator.

Cele prezentate se pot constitui în criterii de fidelitate, cu ajutorul cărora pot fi diferențiate calitativ diferitele modele ale aceluiași sistem sursă. Aceste criterii influențează modul de parcurgere a etapelor construirii modelului și validarea sa.

Calitatea unui model este determinată și de alte aspecte suplimentare: simplitatea, inteligibilitatea, costul etc.

Astfel, sunt prezentate câteva exemple de modelări care reliefează anumite trăsături ale originalului.

a) Exemplu de modelare a mandibulei din punct de vedere mecanic: mandibula este un os nepereche, constituie singură maxilarul inferior și se compune dintr-un corp și două ramuri, care formează cu corpul un unghi de circa 120° . Se poate modela simplificat ca o bară curbă (figura 2-2), pentru calculul centrului de greutate și al momentelor de inerție cu ajutorul mărimilor de intrare: unghiurile α , β , γ și lungimile l_1 , l_2 și R [177].

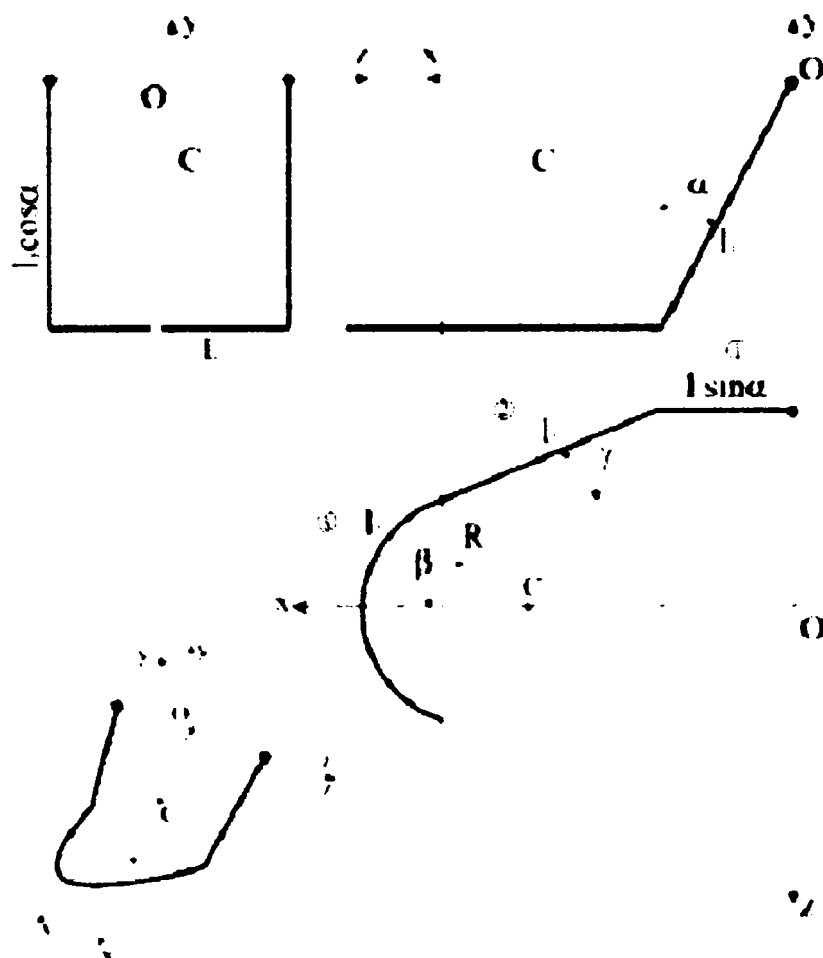


Fig.2-2 Exemplu de modelare a mandibulei din punct de vedere matematic-mecanic

b) Exemplu de modelare a mandibulei din punct de vedere al forțelor ce acționează asupra sa. În mod uzual mandibula se comportă ca o pârghie de gradul 3. Articulația temporo-mandibulară este punctul fix al pârghiei, forța motoare este dată de rezultanta acțiunii mușchilor (maseter, pterigoidian și temporal), iar la nivelul dinților se manifestă forța rezistentă (figura 2-3). La o închidere normală, relaxată a gurii, contactul dentar se face la nivelul incisivilor, astfel obținându-se pârghia de gradul 3. În această situație, forțele asupra incisivilor sunt destul de mici. Dacă însă apare un contact prematur între dinții laterali (premolari sau mai ales molari) sau dacă subiectul mușcă voit (cazul uzual al masticăției) întâi cu acești dinți, punctul de aplicație al forței rezistente se va muta pe ei, fapt care creează o pârghie de gradul 2[177].

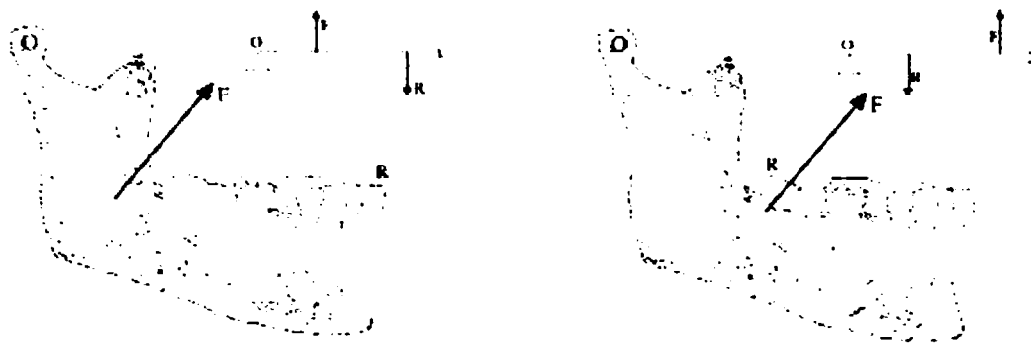


Fig.2-3: Mandibula asimilată cu pârghii de gradul 2 și 3

c) Exemplu de modelare fizică a mandibulei, realizată cu ajutorul metodelor ce vor fi descrise în teză (figura 2-4):

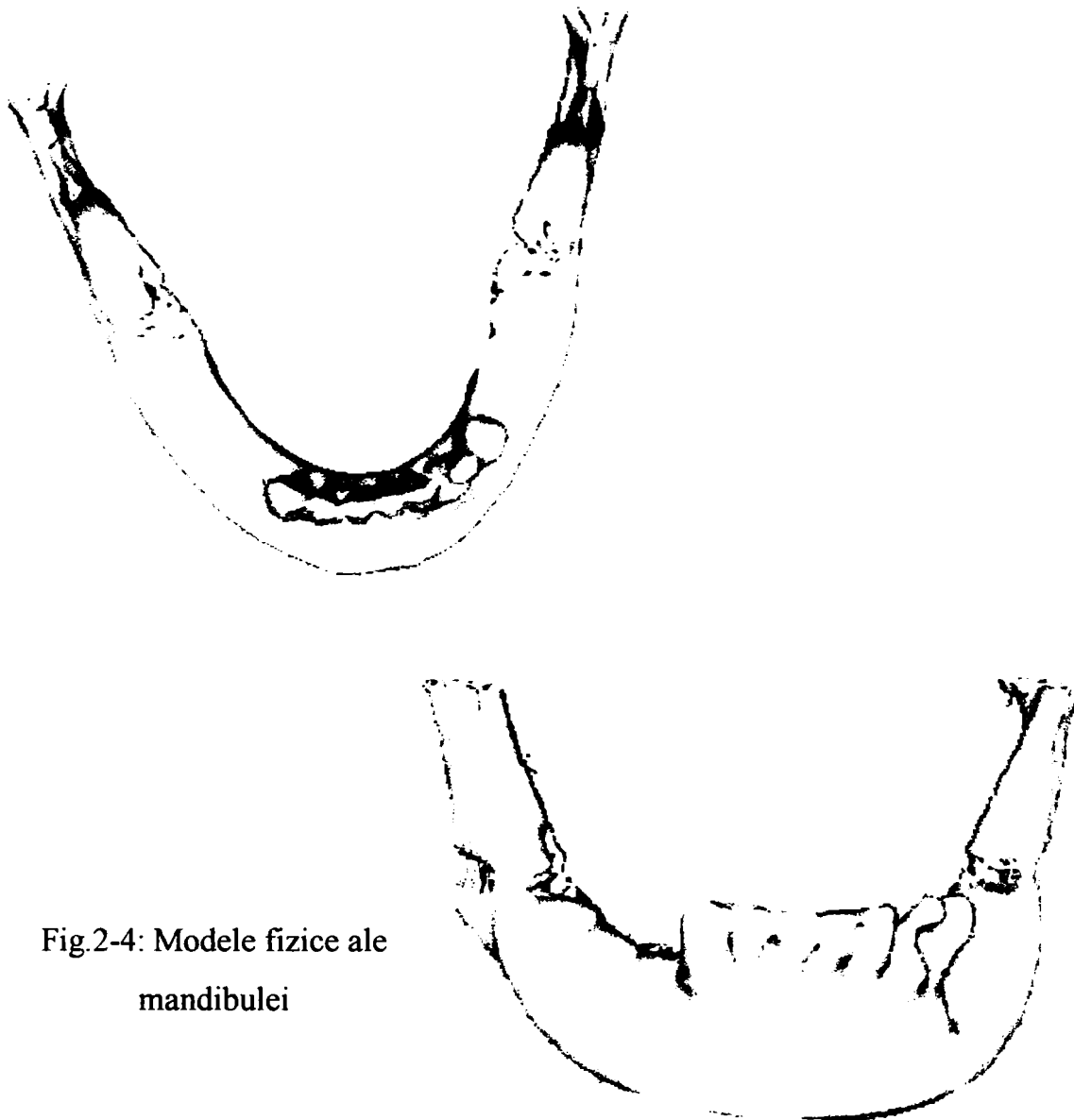


Fig.2-4: Modele fizice ale mandibulei

Se observă că modelul obținut prin această tehnică este real din punct de vedere anatomic, putând fi supus, ulterior obținerii sale, la orice acțiune exterioară, la fel ca o mandibulă reală.

Orice model poate fi apreciat prin prisma posibilităților de reprezentare a sistemului sursă. Pentru modelarea structurii osoase faciale este necesară abordarea configurațiilor geometrico-spațiale craniene ce au forma unor structuri de rezistență pentru susținerea componentelor funcționale active din grupa mușchilor și ligamentelor. Definitivarea completă a acestor configurații geometrico-spațiale se realizează prin procedeele și tehnicile de modelare geometrică. Modelul geometric rezultat face parte dintr-o mulțime de modele atașate celui care relevă esența funcțional-comportamentală.

Criteriile de clasificare cunoscute se referă la această ultimă trăsătură și diferențiază modelele, cu observația că, în afara celor prezentate în teza, pot exista și altele, în funcție de specificitatea domeniului din care face parte sistemul sursă.

În general, drept criterii de clasificare se acceptă[165][163][115][116]:

A. **Criteriul materialității**, funcție de care modelele pot fi **abstracte** (teoretice, matematice) sau **materiale** (fizice, replici).

Modelele **abstracte** pot fi diferențiate funcție de modul de abordare de către cel care modelează și de profunzimea cunoștințelor sale despre sistemul sursă:

a) în funcție de *forma de prezentare*, care poate fi:

- clasică (a relațiilor matematice):
 - analitice, la care sunt luate în considerare cunoștințe despre proprietățile sistemului sursă atât din punct de vedere calitativ, cât și cantitativ; pentru a construi modelele se pornește de la legi fizice care se pot concretiza în relații matematice;
 - analogice, care utilizează o presupusă analogie cu un sistem funcționând după legi fizice cunoscute, caz în care modelul se bazează pe un ansamblu de ipoteze din care pot fi deduse consecințe comportamentale ale modelului;
- mulțimi de date corelate (de intrare și de ieșire);
- descriptivă, legată într-o anumită măsură de o prognoză comportamentală care este destinată să explice comportamentul sistemului sursă (răspunde la

întrebarea *cum este?* sau *cum va fi?*); această formă este cel mai des utilizată în medicină;

b) în funcție de *aspectul* sistemului sursă *care interesează*:

- modele la care se urmărește compoziția, structura internă a sistemului de modelat; modelul este numit **conceptual** în unele surse bibliografice; se referă la variabilele caracteristice ale sistemului sursă;
- modelele la care interesează doar comportamentul, în sensul cunoașterii intrărilor și ieșirilor corespunzătoare ale sistemului (sistem considerat *cutie neagră*); este denumit și model **informațional**, pentru că la construirea modelului se ține seama doar de informații date sub forma valorilor concrete ale unor variabile de intrare/ieșire (pe baza acestora se pot desfășura operații de conducere ale sistemului sursă, modelul servind acestui scop);
- modele **hibride**, la care se iau în considerare ambele aspecte.

Modelul conceptual este, deci, o construcție de legături logico-matematice între variabilele definitorii ale sistemului sursă, pe când cel informațional se bazează pe informații care sunt oferite în exterior în timpul funcționării. În practică se preferă modelele hibride, în care se realizează un echilibru adecvat între aspectul informațional și aspectul conceptual.

Modelele **materiale** (fizice, concrete, substanțiale, replici) permit rezolvarea pe cale experimentală a unor probleme care nu pot fi rezolvate pe cale analitică sau logico-matematică, fie datorită necunoașterii metodelor de calcul adecvate, fie datorită gradului mare de complexitate a acestora, conducând la activități mult prea laborioase și costisitoare.

Funcție de natura acestor modele, se disting:

- a) modele **similare**, de aceeași natură cu sistemul sursă, care diferă de acesta din urmă doar prin numărul sau valoarea dimensiunilor sau caracteristicilor; se bazează pe *teoria similitudinii*, fiind utilizate frecvent în inginerie. Nu se poate concepe nici un studiu serios fără a se apela la astfel de modele, care poartă uzual denumirea de *machete*.

În domeniul abordat prin teză modelele de acest tip sunt fie sub formă de preparate anatomice din țesuturi prelevate de pe cadavre, fie realizate din

materiale speciale ce au proprietăți mecanice și funcționale aproape identice cu ale țesutului viu.

- b) modele **analoage**, bazate pe fenomene diferite de cele din sistemul sursă, dar la care relațiile matematice care descriu comportamentul modelului au aceeași formă cu cele care guvernează sistemul sursă.

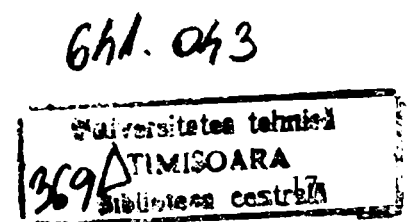
B. Criteriul posibilității de structurare conduce la modele:

- **sintetice**, la care sunt prezentate proprietățile relevante într-un tot cvasiunitar;
- **structurate (modulare)**, constituite dintr-unul sau mai multe modele elementare, acestea din urmă relevând și considerând determinant doar un singur aspect al structurii sau comportării.

În aplicațiile biomedicale sunt extrem de importante modelele materiale virtuale realizate cu ajutorul calculatorului (figura 2-5), și cele materiale similare. Cu ajutorul lor, dar și cu ajutorul celor matematice, se pot realiza studii prețioase cu privire la comportarea în ansamblu a unui anumit aparat anatomic, se pot pune diagnostice și se poate perfecționa o intervenție chirurgicală sau stabili o strategie de lucru în acest sens.



Figura 2-5: Model virtual al scheletului labei piciorului realizat cu ajutorul calculatorului



2.2 Conceptul de modelare

În sens restrâns, **modelarea** reprezintă activitatea de elaborare propriu-zisă a modelului unui sistem sursă. Activitățile desfășurate în acest scop sunt materializate prin:

- tehnici și proceduri de identificare;
- tehnici de simulare;
- tehnici și proceduri complementare.

Etapele prin care se realizează activitatea de modelare sunt, în general, următoarele[10][46][117][118][163][165]:

a) construirea modelului pe baza:

- analizei preliminare a sistemului sursă în vederea evidențierii parametrilor relevanți și a legăturilor funcționale dintre ei;
- stabilirea unei structuri a modelului;
- stabilirea valorilor parametrilor definatorii ai modelului;

b) analiza funcționării modelului prin simulare;

c) compararea rezultatelor analizei cu datele de comportare a sistemului sursă în condiții echivalente;

d) corectarea modelului, în sensul apropierii comportamentului de cel al sistemului sursă.

În sens mai larg, prin **modelare** se înțelege metoda de studiu bazată pe utilizarea modelelor, metodă la care se apelează practic în ansamblul cercetărilor din domeniile tehnice, mai ales în cele interdisciplinare.

3. METODE PENTRU REALIZAREA FORMELOR COMPLEXE ÎN MODELAREA GEOMETRICĂ

3.1 Tehnici de descriere a modelelor 3D

Funcție de structurile de date și de informațiile de model stocate, sistemele 3D pot fi [115][117][164]:

- Orientate pe muchii;
- Orientate pe suprafețe;
- Orientate pe volume.

Dintre acestea, modelele solide (orientate pe volume) înmagazinează cea mai completă descriere geometrică.

Modelele solide pot fi obținute prin următoarele tehnici mai frecvente [116][163][164][165]:

- reprezentarea geometric-constructivă (RGC)
- reprezentarea prin frontiere (RF)
- reprezentarea prin enumerare spațială (RES)
- reprezentarea prin baleiere (RB)

Formele modelelor solide pot fi gândite în termenii unor forme geometrice elementare (primitive de tipul prismă, cilindru, sferă, con etc.), sau în termenii unor forme elementare din punct de vedere al generării tehnologice (de rotație, extrudate etc.).

Formele geometrice elementare necesare construirii unor obiecte complexe, pot fi obținute utilizând accesul la forme **primitive standard** puse la dispoziție de sistemul de proiectare (modelator). Aceste forme primitive standard trebuie să fie parametrizate, în sensul că utilizatorul poate modifica una sau mai multe dimensiuni.

Utilizarea doar a formelor primitive standard poate limita construcția modelului. De aceea sistemele CAD oferă, de regulă, utilizatorului posibilitatea de a defini după necesități anumite entități geometrice elementare. Se pot crea astfel suprafețe pornind de la curbe, sau volume pornind de la suprafețe. Cea mai simplă metodă este atribuirea de grosimi formelor plane (procedeu echivalent prelucrării prin extrudare).

Metoda de generare prin primitive prezintă dezavantajul numărului mare de operații booleene standard necesare, aceasta implicând o complexitate crescută și un cost ridicat.

Operația manuală de grupare a unor fețe orientate coerent pentru a obține un solid este laborioasă și împotriva spiritului general al CAD, astfel încât s-au elaborat operatori speciali pentru a asista crearea modelelor solide.

Un astfel de operator special, aplicat unor suprafețe convenabil create, este cel de îngroșare. Acest operator ia o suprafață de bază, definește pentru ea o suprafață *offset* și apoi închide spațiul dintre ele cu o bandă laterală. Acest operator este potrivit pentru obiectele care au baze similare. Banda care închide corpul, poate fi riglată sau poate fi complexă, de tip *loft*, ce ține seama și de direcțiile normalelor celor două suprafețe inițiale (de bază și *offset*), pentru asigurarea continuității ceea ce va evita muchiile ascuțite.

Indiferent de tehnica de obținere a modelului, acesta trebuie să satisfacă anumite cerințe[139][140][165][164][116]:

- **Validitatea:** să existe un obiect real corespunzător oricărui model posibil creat;
- **Universalitatea:** să se poată crea un model pentru orice obiect 3D;
- **Unicitatea interpretării:** unei reprezentări concrete îi corespunde doar un singur obiect;
- **Unicitatea expresiei (neambiguitatea):** unui obiect 3D îi corespunde doar un singur model;
- **Caracterul complet:** toate operațiile suportate să fie aplicabile tuturor reprezentărilor (să genereze în toate cazurile entități valide ca operanzi pentru toate operațiile);
- **Concizia:** informația conținută în model să fie minimă.

3.1.1 Reprezentarea geometric constructivă a solidelor (RGC)

Utilizând această tehnică, modelele complexe se pot crea grupând componente mai simple, denumite **primitive**. Exemplele clasice de primitive geometrice sunt: cubul, conul, cilindrul, sfera, etc. În tehnica de reprezentare prin geometrie constructivă, modelul este reprezentat printr-un arbore (sau graf pentru modelele complexe). La capetele ramurilor arborelui se află obiecte primitive parametrabile, iar în noduri operatori pe mulțimi. Fiecărui nod îi corespunde o primitivă [165][139][140][164][115].

Tehnica RGC este avantajoasă în cazul modificării geometriei, dar nu este adaptată vizualizării. În figura 3-1 (b) este prezentat modelul RGC al obiectului din figura 3-1 (a). Modelul rezultă printr-o succesiune de reuniuni aplicate pornind de la forme geometrice simple.

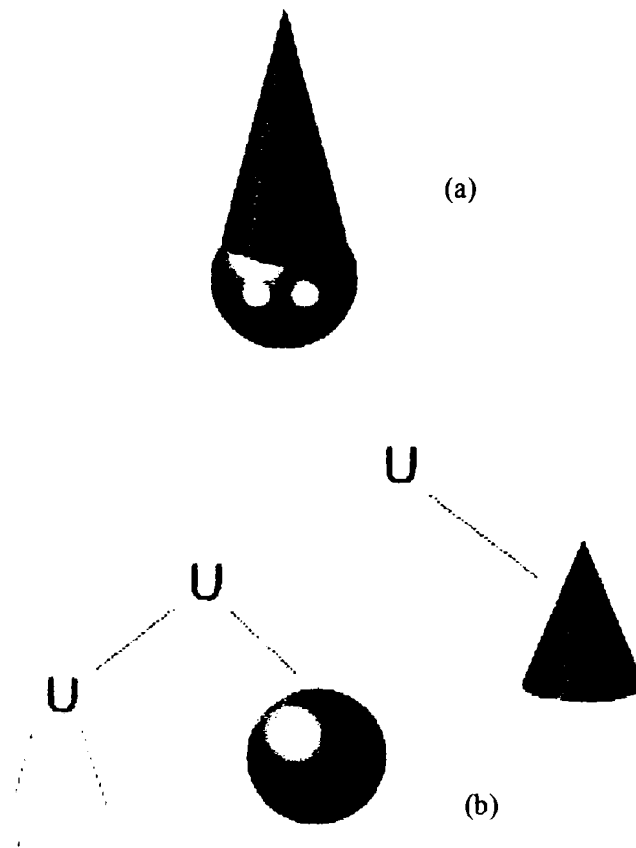


Fig.3-1: Model simplu (a) rezultat din succesiunea operațiilor de reuniune ale primitivelor cerc, sfera, con (b)

Această tehnică nu este avantajoasă pentru modelarea biomFiecanică, deoarece formele ce intervin în acest domeniu sunt extrem de complexe. Poate fi folosită în cazul

modelelor simple, dar și în cazul celor mai complexe destinate unor calcule simplificate în cazul în care sunt aproximare prin primitive, ca în exemplul din figura 3-3 (b).

Principalele avantaje ale modelării RGC sunt validitatea inerentă, unicitatea interpretării, caracterul complet, consecvența și concizia.

Principalul dezavantaj este complexitatea calculelor pentru evaluarea arborelui RGC în vederea vizualizării. Un alt dezavantaj este imposibilitatea reprezentării tuturor obiectelor fizice, de exemplu a celor cu suprafețe curbe cu formă liberă, precum cele întâlnite în domeniul biomedical.

3.1.2 Reprezentarea prin frontiere (RF)

Modelele 3D construite prin enumerarea fețelor ce separă obiectele de restul spațiului se numesc modele reprezentate prin frontiere [163][139][140][165]. Modelul memorează și poziția interiorului obiectului față de fețe. Tehnica este avantajoasă pentru vizualizări dar nu este bine adaptată pentru unele operații analitice, cum ar fi calculul poziției centrului de greutate sau al momentelor de inerție.

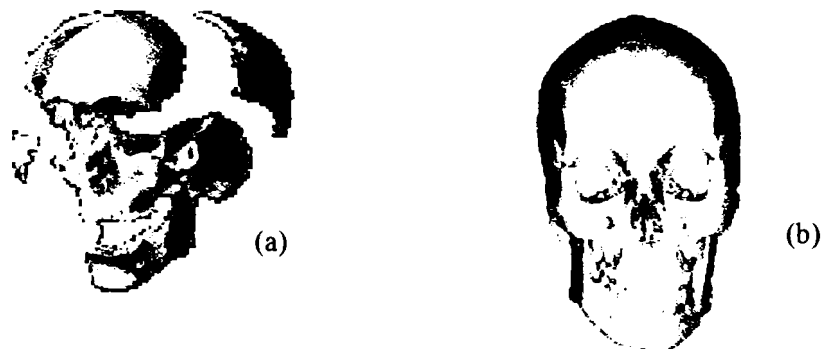


Fig.3-2: Model 3D descris prin frontierele sale

Astfel, figura 3-2 (a) prezintă un model RF al obiectului din figura 3-2 (b).

Modelarea prin frontiere prezintă dificultăți în respectarea cerințelor impuse modelelor solide valide. Unul dintre dezavantajele frecvente constă în asigurarea **continuității** geometrice, care la aplicarea unor modificări modelului este uneori dificil de obținut. În aceste cazuri pot să apară fețe flotante, cu muchii ce aparțin doar unei fețe. Într-un model corect, o muchie reprezintă întotdeauna concurența a doar două fețe. O altă

problemă posibilă este **compatibilitatea** modelului: suprafața generată trebuie să fie fizic posibilă. Sunt excluse auto-intersecțiile și ambiguitățile de orientare, normalele a două fețe ce se intersectează după o muchie trebuie să fie orientate compatibil.

3.1.3 Reprezentarea prin enumerare spațială (RES)

Reprezentarea prin enumerare spațială divizează spațiul modelului în elemente de volum, considerate unitare, similar cu tehnica de descriere a imaginilor 2D, prin mulțimea de pixeli constituenți (*bit-mapped*). Elementul de volum unitar echivalent unui pixel este denumit **voxel** (*volume element*) [67][115][116][163][165][140].

Un obiect solid este descris prin fanioane care indică elementele de volum ocupate. Deoarece enumerarea tuturor punctelor volumice ar consuma memorie enormă, se utilizează obiecte unitare speciale. S-a dezvoltat o metodă de divizare recursivă, cunoscută sub numele de *quadtree* 2D și *octree* în 3D. Un mare avantaj al acestei metode este posibilitatea de descriere a obiectelor complexe neregulate, de tipul celor existente în natură.

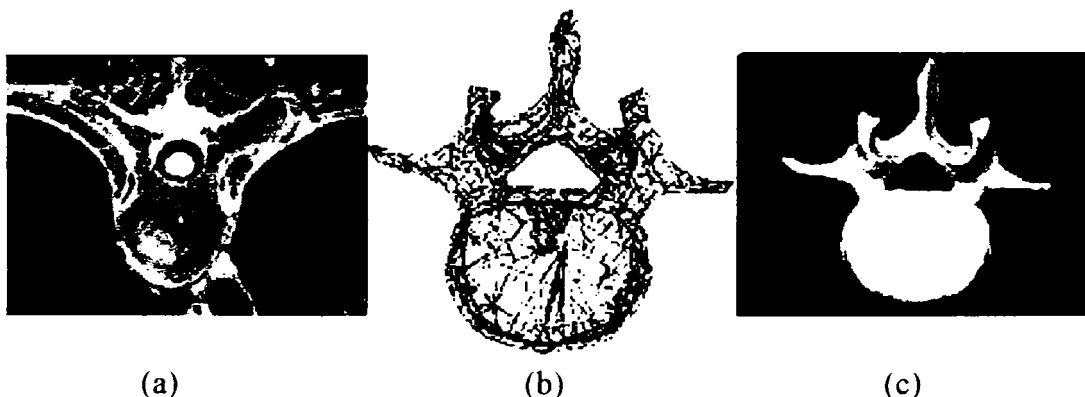


Fig.3-3: Modele ale unei vertebre

Spre exemplu, vertebra din figura 3-3(a) este reprezentată prin enumerare spațială în figura 3-3(b). Diferitele tipuri de modele servesc la aplicații distincte. Astfel, reprezentarea din figura 3-3(b) servește la analiza numerică a distribuțiilor de tensiuni și deformații prin metoda elementelor finite. Reprezentarea din figura 3-3(c) servește la modelarea comportării mecanice în vederea reconstrucției chirurgicale.

3.1.4 Reprezentarea prin baleiere (RB)

Este o tehnică utilă în primul rând, pentru obiecte solide cu grosime constantă sau cu simetrie axială. Solidele din prima clasă rezultă prin baleiere rectilinie (tip extrudare), iar cele din a doua clasă – prin baleiere circulară (corpuri de revoluție).

Baleierea constă în deplasarea unui punct, a unui segment sau a unei suprafețe, pe o anumită traiectorie. Pentru forme complexe, traiectoria este o curbă complexă [204][185][115][116][165][133][135].

Metoda este utilizată mai ales pentru obținerea unor elemente ce urmează a fi incluse în scheme RGC sau RF. Nu există modelatoare pure prin baleiere din cauza limitelor metodei.

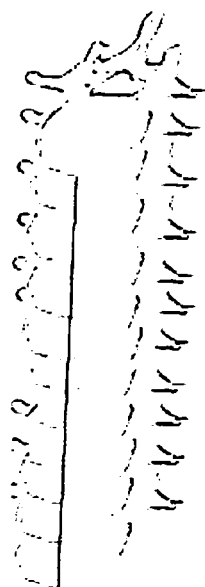


Fig.3-4 Exemplu de reprezentare prin baleiere a unei porțiuni din coloana vertebrală

Deplasând de-a lungul unei traiectorii stabilite o secțiune a unei vertebre, se poate obține o reprezentare aproximativă a coloanei vertebrale, așa cum se poate observa în figura 3-4. Pentru obținerea unui rezultat cât mai aproape de realitate trebuie luate în considerare particularitățile individuale ale vertebrelor precum și ale întregului ansamblu, coloană vertebrală, vertebrele nefiind identice, iar traiectoria fiind o linie curbă. Deci o sumă a unor reprezentări prin baleiere distincte ca obiect deplasat și ca traiectorie poate genera cu succes un obiect destul de complex cum sunt modelele structurilor biologice.

3.2 Reprezentarea obiectelor utilizând curbe

Una din principalele probleme de proiectare a produselor este proiectarea formei acestora. Pentru aceasta este necesară o tehnică simplă de descriere a formei oricărui obiect, oricât de complex ar fi el.

În cazul formelor simple, este suficientă utilizarea unor entități analitice ca: puncte, segmente de dreaptă, cercuri și arce de cerc, conice și arce de conice (elipse, parabole, hiperbole), etc. Pentru forme mai complexe, este necesară utilizarea unor entități sintetice (curbe spline și Bezier).

Este nerațională descrierea obiectelor complexe prin mulțimi de puncte (enumerare). Acest mod de reprezentare ar necesita o capacitate enormă de memorare. Dezavantajul specific este următorul: o modificare a obiectului (de exemplu prin rotire, scalare, deformare) ar fi extrem de dificilă, dacă nu chiar imposibilă. Este mult mai avantajos să se fixeze niște puncte de referință și să se genereze (sintetizeze) curbe între ele (figura 3-5), care să aproximeze forma obiectului de reprezentat.

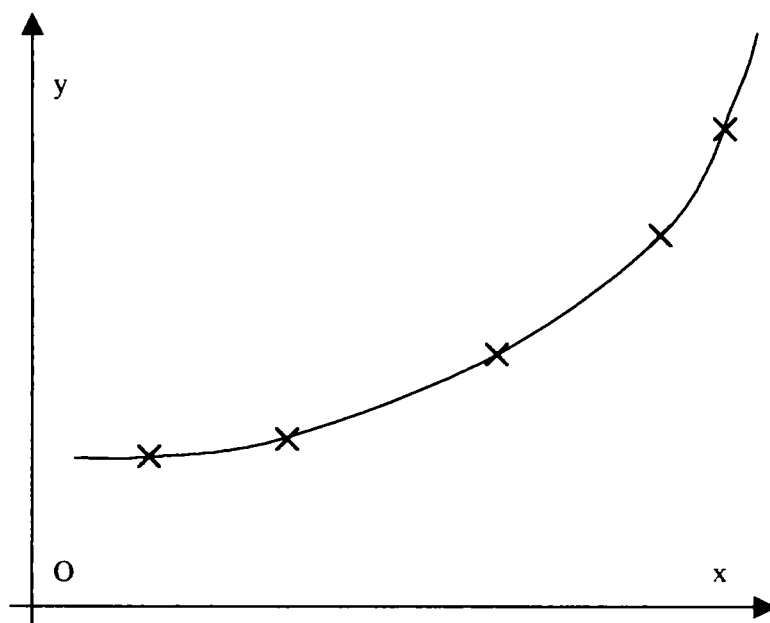


Fig.3-5 Interpolare polinomială

3.2.1 Curbe spline

Ideea entităților sintetice a condus la aplicarea metodele de interpolare: se caută o funcție continuă și netedă, care în anumite puncte (de *control* sau de *sprijin*), ia valori date. Pe lângă condițiile de a trece prin puncte (sau cât mai aproape de ele), se mai introduce și condiția ca cele două curbe ce se întâlnesc să realizeze un anumit grad de netezime, pentru a evita colțuri sau salturi nedorite. Matematic, se pune condiția ca primele n derivate a celor două funcții care se întâlnesc să fie continue în acel punct. Continuitatea de ordinul zero (notată C^0) asigură continuitatea pozițiilor, continuitățile de ordinul întâi (C^1) și al doilea (C^2) asigură continuitatea pantei tangentei într-un punct, respectiv a curburii[8][133][163][164][165][135][185].

De exemplu, pentru cazul a trei puncte se poate utiliza un polinom de gradul doi, deci o funcție de tipul:

$$f(x) = a \cdot x^2 + b \cdot x + c \quad (3-1)$$

Punerea condițiilor pentru funcție de a conține cele trei puncte conduce la un sistem de trei ecuații cu trei necunoscute: a , b și c . Acest sistem este determinat: există un singur polinom cu proprietățile cerute. De regulă, acest polinom are un mare dezavantaj: gradul crește cu creșterea numărului de puncte. Prin aceasta se ajunge la o comportare nedorită a funcției între punctele de control: tendința spre oscilații (figura 3-5, curba Lagrange/Newton). Această tendință puternică de oscilație a curbei de interpolare între punctele date nu este de neglijat la metodele clasice precum Newton sau Langrange. În cazul utilizării funcțiilor **spline** acest fenomen nu apare, din cauză că acestea sunt polinoame definite pe subdomenii. Între fiecare două puncte vecine, numite noduri, se definește câte o funcție de grad mic.

S-a dovedit că **funcțiile spline cubice** sunt corespunzătoare pentru majoritatea aplicațiilor. Polinoamele utilizate pe intervale între două puncte succesive sunt de grad maxim 3, independent de numărul punctelor. Pentru a obține polinoamele (coeficienții lor), trebuie rezolvate sisteme de ecuații. Condițiile impuse (trecerea prin punctele date și netezimea la traversări), generează ecuațiile sistemului. Pentru asigurarea unicității soluțiilor mai sunt necesare încă două condiții, care se pot obține pe baza unor condiții la limită. De exemplu, se pot descrie valorile primei derivate (tangenta), la capetele

domeniului (în primul și în ultimul punct). Mai frecvent se întâlnește curba *spline naturală*, în cazul căreia derivatele de ordinul doi sunt nule la capetele intervalului. Curbele spline pot fi astfel continuate cu segmente de dreaptă. În a treia variantă, a curbei *spline periodică*, se pune condiția ca derivatele de ordinul întâi și cele de ordinul doi să aibă aceleași valori la cele două capete ale intervalului. Astfel de curbe spline vor putea fi alăturate succesiv generând o curbă periodică [8][133][165][185][193][204].

3.2.2 Curbe Bezier

Un tip mai interesant de curbe pentru interpolare sunt curbele Bezier. Aceste curbe nu mai au impusă condiția de a trece prin punctele date, cu excepția primului și ultimului punct. Celelalte puncte servesc la controlul formei curbei. Curbele Bezier au, spre deosebire de curbele spline, avantajul că permit scrierea mai simplă a condițiilor de netezime. Panta tangentei în primul, respectiv în ultimul punct, depinde de al doilea respectiv de penultimul punct de control. Întreaga curbă are forma:

$$p(t) = p_0 \cdot b_{0n}(t) + p_1 \cdot b_{1n}(t) + \dots + p_n \cdot b_{nn}(t) \quad (3-2)$$

în care $p_0 \dots p_n$ sunt punctele de control, iar $b_{0n}(t), \dots, b_{nn}(t)$ sunt polinoamele Bernstein de ordinul n , definite prin:

$$b_{in}(t) = \frac{n!}{[i!(n-i)!]} \cdot t^i \cdot (1-t)^{n-i} \quad t \in [0,1] \quad (3-3)$$

Există o formulă de recurență care leagă polinoamele Bernstein de diferite grade:

$$b_{in}(t) = t \cdot b_{i-1,n-1}(t) + (1-t) \cdot b_{i,n-1}(t) \quad (3-4)$$

și care este utilizată de algoritmul *de Casteljau* [165] pentru a obține puncte pe segmentele Bezier prin interpolări liniare repetate.

Forma unei curbe Bezier poate fi ușor controlată. Ea încearcă să se apropie de poligonul ce trece prin punctele de control, denumit poligon de control. Curba va fi întotdeauna în interiorul înfășurătoarei convexe a tuturor punctelor (figura 3-6).

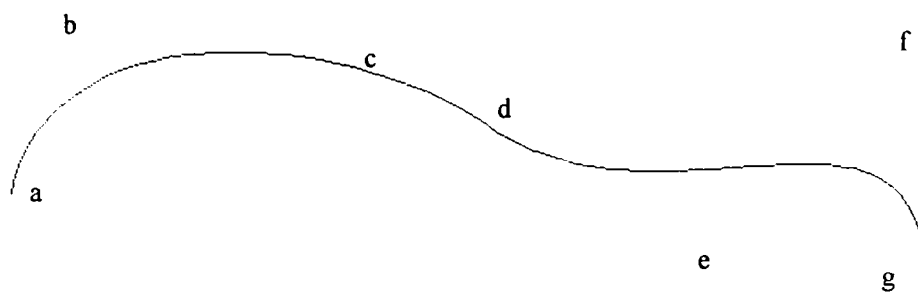


Fig.3-6 Curbă Bezier

Creșterea numărului de puncte de control duce la creșterea numărului și gradului polinoamelor necesare. Acest lucru face ca mici modificări ale punctelor de control să producă mari modificări ale întregii curbe. De asemenea, cresc resursele necesare calculării polinoamelor. Pentru eliminarea dezavantajelor menționate, se concatenează mai multe curbe Bezier de ordin scăzut. Această alăturare este foarte simplă la curbele Bezier, putându-se menține netezimea (figura 3-7)[8][133][165][209].

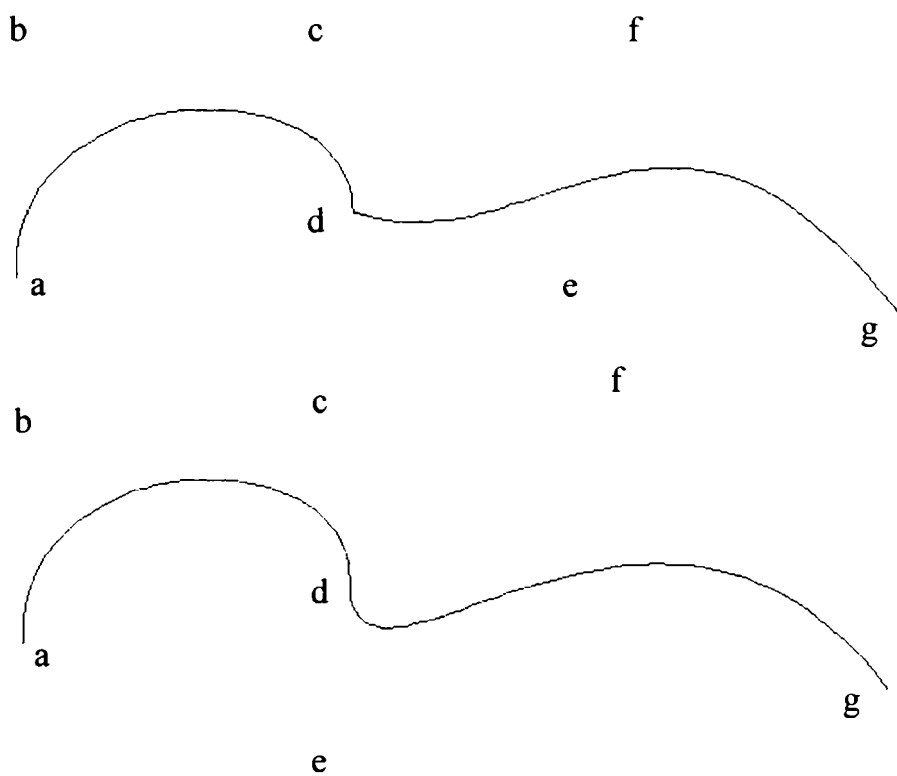


Fig.3-7 Netezirea curbelor succesive

Ambele curbe din figura 3-7 sunt ansamblate din două segmente de gradul trei. Punctele de control a, b, c, și d realizează segmentul 1, punctele d, e, f, și g – segmentul

2. Se observă că se obține o trecere lină între segmente în punctul d, dacă punctele c, d și e sunt coliniare. Acest lucru se întâmplă pentru că tangenta la curba Bezier în punctul de capăt se suprapune peste segmentul de capăt.

Utilizarea unei succesiuni de curbe Bezier de ordin mic reduce volumul de calcule, dar se menține influența modificării unui punct de control asupra întregii curbe, chiar dacă efectul este mai mic. Motivul rezidă în natura polinoamelor Bernstein (figura 3-8 pentru ordinul 3). Cu excepția punctelor de capăt ($t = 0$ și $t = 1$), valorile polinoamelor sunt nenule. Curba rezultantă se obține prin suprapunerea tuturor polinoamelor Bernstein, multiplicată cu coordonatele x și y ale punctelor de control. Prin aceasta o modificare a unui punct de control are efect asupra întregii curbe. Din figură se mai observă și faptul că rezultanta va trece prin punctele de capăt. Pentru $t = 0$ există $b_{03} = 1$, celelalte valori fiind nule, deci intervine doar primul punct, cu ponderea 1. De asemenea, $b_{33} = 1$ pentru $t = 1$ celelalte trei valori fiind nule, deci influență are doar ultimul punct, cu ponderea 1.

Dacă vectorii punctelor de control sunt exprimați în coordonate omogene, se obțin **curbele Bezier raționale**, având coordonatele punctelor exprimate ca rapoarte de polinoame în t . Se suplimentează astfel cu un grad obținerea formei curbelor, prin alegerea convenabilă a coordonatei omogene a fiecărui vector de control. De exemplu, reprezentarea conicelor (arce de cerc, de elipsă etc.) nu se poate face exact utilizând curbe Bezier obișnuite, dar este posibilă prin curbe Bezier raționale.

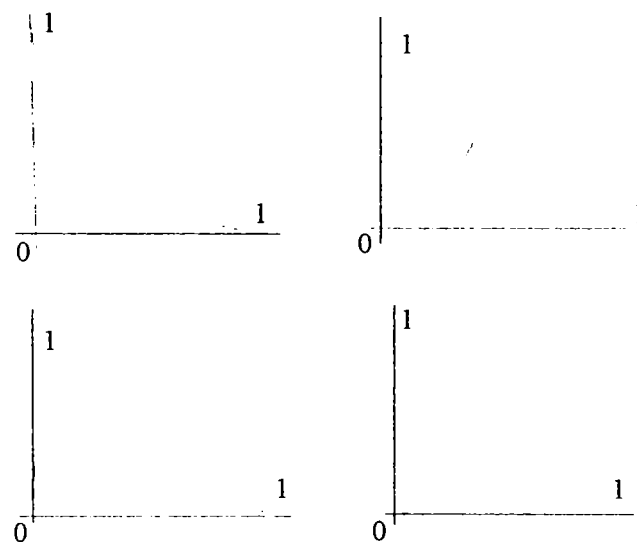


Fig.3-8 Funcțiile de bază Bernstein de ordinul 3

3.2.3 Curbe B-spline

Cele două tipuri de curbe analizate nu sunt convenabile în totalitate pentru reprezentări geometrice. Idealul ar însuma avantajele curbelor spline cubice cu cele ale curbelor Bezier și ar evita dezavantajele acestora.

O soluție practică o constituie curbele *B-spline*, a căror formă amintește curbele Bezier[8][165][133][209]:

$$p(t) = p_0 \cdot B_{0,k}(t) + p_1 \cdot B_{1,k}(t) + \dots + p_n \cdot B_{n,k}(t), \quad (3-5)$$

unde $B_{0,k}(t), \dots, B_{n,k}(t)$ sunt funcțiile de bază B-spline, definite recursiv:

$$B_{i,1}(t) = 1 \text{ dacă } t_i \leq t < t_{i+1} \text{ (} t_n \leq t \leq t_{n+1} \text{ dacă } i = n) \quad (3-6)$$

$$B_{i,1}(t) = 0 \text{ în caz contrar;} \quad (3-7)$$

$$B_{i,k}(t) = \frac{(t - t_i) \cdot B_{i,k-1}(t)}{(t_{i+k-1} - t_i)} + \frac{(t_{i+k} - t) \cdot B_{i+1,k-1}(t)}{(t_{i+k} - t_{i+1})}, \quad (3-8)$$

cu:

$$t_i = 0 \text{ dacă } i < k; \quad (3-9)$$

$$t_i = i - k + 1 \text{ dacă } k \leq i \leq n; \quad i = 0, \dots, n + k \quad (3-10)$$

$$t_i = n - k + 2 \text{ dacă } i > n; \quad (3-11)$$

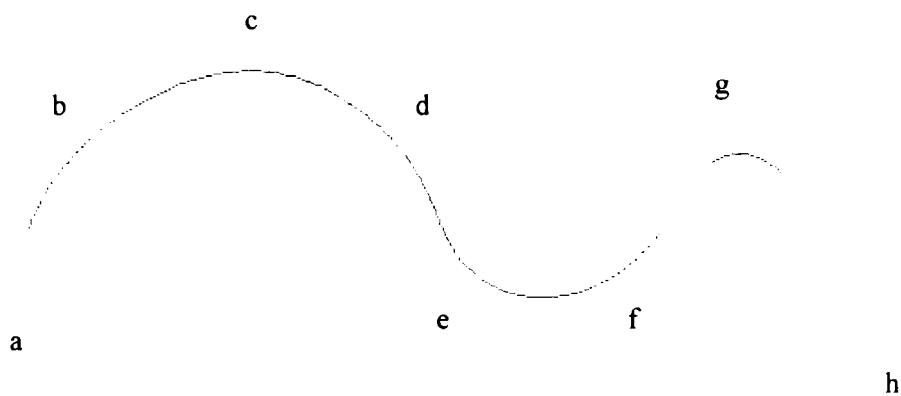


Fig.3-9 Exemplu de curbă B-spline

Ordinul k al curbei este independent de numărul n al punctelor, ceea ce rezolvă un dezavantaj al curbelor Bezier. Modificarea unui punct va avea efect doar în porțiunile de curbă din vecinătatea sa.

Uneori este necesară renunțarea la netezime, în anumite puncte ale curbei dorindu-se prezența unor vârfuri (colțuri). Curbele Bezier permit acest lucru dacă se concatenează mai multe curbe. Curbele B-spline permit o cale mai simplă de introducere a unor vârfuri, puncte de control multiple, adică puncte succesive cu coordonate identice. Aceste puncte multiple generează în jurul lor un domeniu cu o curbură foarte mare (figura 3-10).

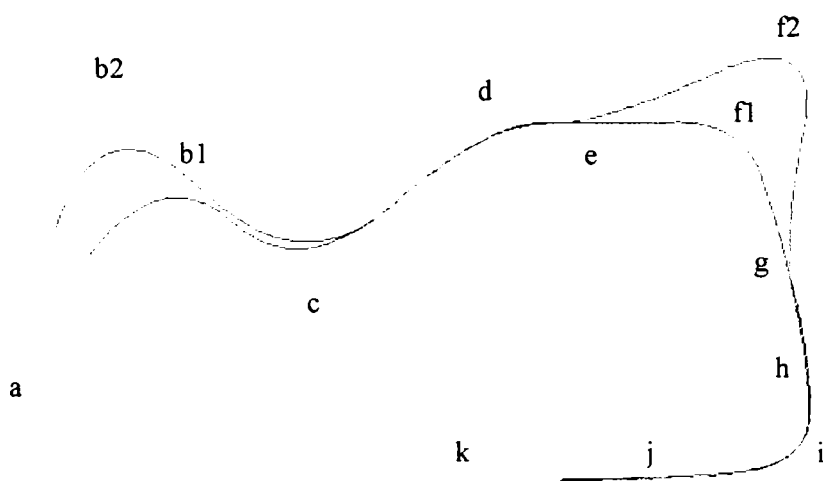


Fig.3-10 Modificarea curbelor B-spline

Prin modificarea ordinului k al unei curbe B-spline se poate influența forma întregii curbe, de exemplu, pentru $k = 2$, curba este poligonul ce leagă punctele ce definesc curba. Creșterea ordinului face curba B-spline tot mai netedă, dar se depărtează tot mai mult de punctele de control. Pentru $k = n+1$, curba B-spline este identică cu curba Bezier corespunzătoare, curbele Bezier fiind un caz particular al curbelor B-spline.

În mod obișnuit, nodurile sunt uniform distribuite, obținându-se curbele B-spline uniforme. Alegând în alt mod distribuția nodurilor, se obțin **curbe B-spline neuniforme**, care aduc un grad de libertate suplimentar în obținerea formei curbelor.

Dacă vectorii punctelor de control sunt exprimați în coordonate omogene, se obțin curbele **B-spline raționale** (cu coordonatele punctelor exprimate ca rapoarte de polinoame). Se suplimentează numărul gradelor de libertate în obținerea formei curbelor, prin alegerea convenabilă a coordonatei omogene a fiecărui vector de control.

Aplicând simultan o distribuție neuniformă a nodurilor și exprimarea vectorilor de control în coordonate omogene, se obțin cele mai complexe **curbe B-spline neuniforme raționale**[165][8][133][199][209][204].

Din cauza multelor proprietăți pozitive, în principal controlul local, curbele B-spline au intrat puternic în domeniul modelării pentru CAD. Calculul lor nu este dificil, ele pot fi ușor extinse la mai multe dimensiuni (cum sunt suprafețele).

3.3 Obținerea de suprafețe pornind de la curbe

Există două clase de suprafețe utilizate în proiectarea asistată de calculator:

- Suprafețe plane
- Petice

Suprafețele plane, cu toate că au limitări, permit realizarea de modele robuste și rezonabil de complexe. Modelele fațetate utilizează poligoane plane pentru a reprezenta inclusiv suprafețe curbe prin discretizare cu un grad de aproximare acceptat. Astfel, se realizează generarea informațiilor pentru interiorul unui petic, adică generearea unor puncte ce pot fi, apoi, unite pentru a rezulta un model *wireframe*, sau grupate pentru a forma poligoane (fațete), de obicei triunghiulare pentru o precizie suficient de bună.

Peticele realizează o descriere geometrică a unor suprafețe strâmbe. Peticele pot fi definite implicit, prin funcții de forma $f(x,y,z) = 0$, sau parametric. Reprezentarea parametrică este mai utilă în CAD. Fiecare coordonată este tratată ca o funcție de doi parametri, s și t sau u și v , de regulă cu limite normalizate: $s, t \in [0, 1]$.

Din punct de vedere al modului de generare, se evidențiază două clase de suprafețe:

- obținute prin interpolare
- obținute prin baleiere.

În prima clasă se înscriu: suprafețele generate prin aplicație bilineară, *lofting* simplu sau dublu (*Coons*), peticele Bezier și cele bicubice, etc., iar în a doua clasă se încadrează: suprafețele obținute prin extrudare, *sweeping* și cele de revoluție [8][33][34][209][199][173][171][110].

Suprafețele complexe ale modelelor vor fi realizate, de obicei, din mai multe petice alăturate. În acest caz se pun mai multe probleme pentru ca suprafețele să fie fizic realizabile:

- **Compatibilitatea (coerența):** normalele la suprafață de-a lungul muchiilor comune să fie orientate corespunzător pentru cele două petice;
- **Autointersectarea:** peticele să nu se autointersecteze;
- **Continuitatea:** funcționalitatea și estetica cer continuitate de ordin superior (și pentru derivate).

3.3.1 Aplicație biliniară

Generarea unei suprafețe pornind de la patru puncte se poate obține prin aplicație biliniară [8][33][34][110][209][171][165][173]. Punctele Q ale peticului rezultat utilizând punctele P_{00} , P_{01} , P_{10} , și P_{11} se pot calcula, parametric, din:

$$Q(s,t) = \left\{ \begin{matrix} 1-s & s \end{matrix} \right\} \begin{bmatrix} P_{00} & P_{01} \\ P_{10} & P_{11} \end{bmatrix} \left\{ \begin{matrix} 1-t \\ t \end{matrix} \right\} \quad (3-12)$$

Suprafața rezultantă este riglată. O față plană se poate obține prin interpolare, dacă toate vîrfurile poligonului ce definesc fața respectivă sunt coplanare. Figura 3-11 prezintă o suprafață obținută utilizînd patru din vîrfurile unui cub.

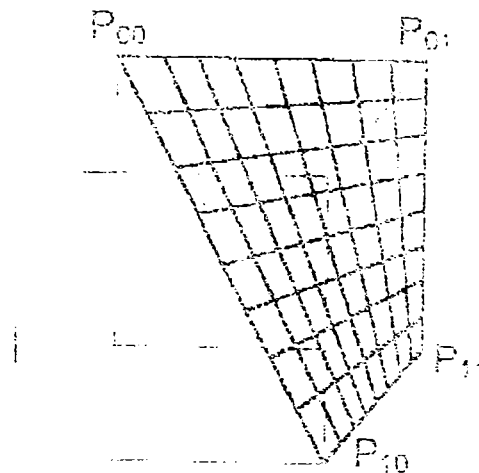


Fig.3-11 Suprafață tip aplicație biliniară

3.3.2 Lofting

Prin lofting se realizează potrivirea unei porțiuni de suprafață, pe două curbe parametrice limitatoare (frontierele), utilizînd interpolarea între două funcții parametrice opuse, utilizate pentru sprijinirea pe cele două frontiere [8][33][34][110][171][165][133].

Un caz particular este cel al **suprafețelor riglate** (figura 3-12), cînd cele două funcții opuse depind de s respectiv $1 - s$. Se realizează practic, o trecere lină de la o curbă limită la cealaltă. Punctele Q ale suprafeței rezultate prin *lofting* linear pornind de la curbele C_1 și C_2 (oricît de complexe) se pot calcula, în formă parametrică, din:

$$Q(s,t) = C_1(t) \cdot (1-s) + C_2(t) \cdot s \quad (3-13)$$

O generalizare simplă conduce la o operație ce generează o suprafață care interpolatează o familie de curbe parametrice paralele.

Este o operație de proiectare la nivel înalt, o metodă eficientă de a specifica mai mulți coeficienți simultan, printr-o operație coerentă unică și geometric semnificativă și intuitivă.

$$Q(s,t) = \begin{Bmatrix} 1-s & s & 1 \end{Bmatrix} \begin{bmatrix} -P_{00} & -P_{01} & C(0,t) \\ -P_{10} & -P_{11} & C(1,t) \\ C(s,0) & C(s,1) & 0 \end{bmatrix} \begin{Bmatrix} 1-t \\ t \\ 1 \end{Bmatrix} \quad (3-14)$$



Fig.3-13 Suprafață Coons

În multe situații de proiectare, caracteristicile critice ale unei suprafețe sunt curbele ce o limitează, interiorul fiind relativ sensibil în specificații. Aici se pot utiliza suprafețe Coons.

3.3.4 Petice bicubice

Din cauza faptului că, la construirea modelelor, expresiile analitice ale curbelor utilizate sunt rareori cunoscute (deseori rezultând din operații asupra entităților geometrice) și că, de regulă, se folosesc aproximări cubice (spline), se poate considera o clasă de petice cubice [8][33][34][110][165], ale căror puncte Q se pot calcula parametric din:

$$Q(s,t) = \sum_{i=0}^3 \sum_{j=0}^3 a_{ij} s^i t^j \quad (3-15)$$

Coeficienții a_{ij} depind de coordonatele vîrfurilor petecului, de tangentele și de curburile în vîrfuri.

3.3.5 Petice Bézier

Peticele Bézier sunt o extensie a curbelor B-spline [110][33][34][165], utilizînd, pentru interpolare, două seturi de funcții de bază Bernstein (Φ):

$$Q(s,t) = \sum \sum \Phi_i(s) \cdot \Phi_j(t) \cdot P_{ij} \quad (3-16)$$

De exemplu, o bază quadratică va conduce la:

$$Q(s,t) = \left\{ \begin{matrix} (1-s)^2 & 2s(1-s) & s^2 \end{matrix} \right\} \begin{bmatrix} P_{00} & P_{01} & P_{02} \\ P_{10} & P_{11} & P_{12} \\ P_{20} & P_{21} & P_{22} \end{bmatrix} \left\{ \begin{matrix} (1-t)^2 \\ 2t(1-t) \\ t^2 \end{matrix} \right\} \quad (3-17)$$

Se observă că matricea de control conține doar puncte. Acestea formează ceea ce se cheamă *rețeaua de control*.

3.3.6 Suprafețe capac

Problema generării suprafețelor de tip capac apare natural, constînd în definirea unei suprafețe de închidere într-o regiune definită de o singură curbă închisă. Un exemplu simplu este închiderea la capete a unei construcții cilindrice.

Modelele suprafețelor capac se pot obține prin generalizarea suprafețelor *Coons*. Există cîteva cazuri particulare interesante. Un astfel de caz apare cînd capacul este plan. Soluția standard va fi un disc [110][135][165][33][34][77].

Pentru forme mai complicate ale limitelor capacului, chiar concave, soluțiile se obțin mai dificil. Se poate apela, dacă este permis, la fațetarea regiunii capacului, încercînd apoi construirea unui sistem spațial de elemente triunghiulare (tesselare).

Aceste suprafețe se dovedesc utile în modelarea ce ajută chirurgia reparatorie, un caz fiind acela al substituirii de porțiuni ale craniului:



Fig.3-14 Modelare incompletă, a carei soluție poate fi adăugarea unei suprafețe de tip capac

În figura 3-14 este prezentat un model al capului unei femei în vârstă, reconstruit pe baza datelor reale, obținute prin tomografiere, însă, în partea superioară, lipsind plane cu informații, se poate observa că soluția ar putea fi dată de o suprafață “capac”. Evident, în cazul unor intervenții chirurgicale pe craniu, “peticele” sunt mult mai mici, curburile nu sunt atât de accentuate, și deși generarea modelelor suprafețelor capac nu este simplă, utilitatea este foarte mare.

3.3.7 Suprafețe de revoluție

Definirea suprafețelor ca suprafețe de revoluție este larg utilizată în situații practice, metoda fiind considerată una din cele de bază și naturală. Generarea se face prin baleiere circulară.

Utilizatorul specifică o curbă **profil** și o **axă de rotație** atașată, apoi cere realizarea unei suprafețe de revoluție. Deoarece cercul este reprezentabil ca o curbă B-spline rațională, suprafața de revoluție poate fi reprezentată de tip B-spline în cazul în

care curba profil este curbă B-spline. Prin aceste reprezentări, posibilitățile proiectantului CAD sunt mari. El poate specifica un unghi de rotire a profilului mai mic decât 2π rad, poate apoi combina suprafața sa cu alte suprafețe pentru a obține un obiect mai complex. Flexibilitatea are dezavantajele ei, deoarece, de exemplu, proiectantul poate specifica un profil ce intersectează axa de rotație, rezultând un corp tip dublu con, având un singur punct în secțiunea corespunzătoare intersecției. Un astfel de obiect este, evident, nerealizabil [110][33][34][77][165].

3.3.8 Suprafețe extrudate

Operația de *extrudare*, inspirată de procesul de prelucrare obișnuit, este înrădăcinată ca metodă de descriere a anumitor clase de suprafețe. În sistemele de modelare care permit doar forme primitive și combinațiile lor booleene, nepermițând suprafețe cu formă liberă, operația de extrudare se poate implementa în manieră directă. O operație de extrudare este o baleiere rectilinie, caracterizată de o formă **profil**. În general, profilul este definit de operații booleene între forme 2D. Formal, suprafața extrudată este **produsul vectorial** al formei secțiunii transversale cu un segment ortogonal [110][165].

O posibilitate de obținere a suprafețelor extrudate este definirea profilului la fiecare capăt al liniei de extrudare și apoi conectarea punctelor corespunzătoare prin segmente de dreaptă. Dacă profilul original este închis, rezultatul va fi un solid închis (lateral), cu secțiune constantă în orice plan perpendicular pe axa extrudării. Pentru profiluri curbe, suprafața extrudată se poate obține și prin *lofting*, de obicei riglat, între cele două curbe profil de la capete.

O generalizare permite ca profilul să fie definit într-un plan ce nu este perpendicular pe curba traiectorie.

3.3.9 Sweeping

Există multe posibilități de generalizare a operației de extrudare. Se poate generaliza direcția de baleiere astfel încât suportul său să fie curbe arbitrare în locul unei drepte normale, rezultând operația *sweeping* [110][33][34][165][133].

Operația este o abstractizare a deplasării unei șaibe de grosime infinit de mică pe o bară curbă, astfel să rămână perpendiculară pe tangenta la curba definită de bară în punctul respectiv (figura 3-15). Acest tip de operație este foarte util pentru definirea unor forme care în esență pot fi gândite ca țevi sau piese extrudate îndoite, cu profiluri complicate.

...

Fig.3-15 Suprafață generată prin sweeping

O altă generalizare utilă este cea în care profilul poate varia în funcție de poziția pe linia de *sweeping*. Astfel se poate defini, de exemplu, o conductă sau un canal care se modifică de la secțiune pătrată la una circulară între cele două capete (figura 3-12 b). Pentru traiectorii rectilinii suprafața se poate obține și prin *lofting*.

3.3.10 Suprafețe offset

Operatorul *offset* este important pentru procesul de definire a unui model *spline* al unui reper solid. O *suprafață offset* în cea mai simplă formă (figura 3-16), este translația unei suprafețe cu valoare dată de exemplu, pe direcția normalei. Astfel, cele două capete ale unei bare profilate pot fi considerate, reciproc, obținute prin offset. De notat că suprafața offset se autointersectează și formează bucle, dacă originalul are raze de curbură mai mici decât distanța offset. Dacă o suprafață se autoînvecinează (nelocal), cu mai puțin decât de două ori distanța offset, suprafața ei offset se va autointersecta [110][133][165].

Suprafața offset este o noțiune naturală în întocmirea tehnologiei pentru multe procedee de prelucrare. De exemplu, pentru frezarea unei suprafețe cu freză sferică, operația de frezare este specificată relativ la o suprafață pe care se plimbă centrul sculei, care este definită ca o suprafață paralelă cu suprafața finală (o distanță constantă între cele două suprafețe – pe direcția normală la suprafața origine). Suprafețe offset apar și în procesul de injectare de piese din mase plastice, când trebuie să se țină cont de contracția materialului în faza de răcire.

Fig.3-16 Suprafețe offset

Ambele procedee de prelucrare menționate sunt utilizabile la producerea de implantate metalice sau din material compozit.

3.3.11 Suprafețe booleene și ajustate/tăiate

Modele complexe se pot obține prin operații booleene între primitive sau realizate prin curbe B-spline. De exemplu, un tub poate fi modelat ca diferență între doi cilindri, din care unul cu raza puțin mai mică decât celălalt. Pentru implementarea algebrei volumetrice într-un sistem simplu, se poate memora o formă neevaluată a unei expresii ce descrie operațiile, urmînd a se calcula intersecțiile analitice și efectua evaluările expresiilor pe măsură ce sunt necesare. Sarcina nu este simplă, dar este mai ușor de lucrat cu corpuri ale căror limite nu sunt suprafețe *spline* [110][8][33][34][135][214].

În cazul modelelor B-spline, este posibil să fie necesare operații booleene pe corpuri incomplet specificate (neînchise, neregularizate). Ambiguitățile care apar, ridică probleme în rezolvare.

În cazul aplicării operațiilor booleene pe corpuri definite de funcții spline, de obicei, se presupune, că se lucrează cu o reprezentare a obiectelor prin frontiere. Acest lucru complică stabilirea poziției unui punct în spațiu față de un obiect.

Implementarea completă a operațiilor booleene este complexă. Se pornește de la faptul că într-o reprezentare prin frontiere se caută suprafețele limită ale obiectului rezultat.

Rezultatul poate fi calculat efectuând operațiile booleene de bază asupra elementelor limită ale fiecărui obiect implicat, conectând apoi, în mod corespunzător, elementele selectate.

Principala operație elementară în efectuarea operațiilor booleene este *tăierea*. Această operație este utilă și utilizată independent, nu în cadrul unei operații booleene. Devine necesară acceptarea unui tip de date *suprafață tăiată ajustată* (incompletă). Matematic, pentru suprafețe spline, modelul unei suprafețe tăiate prezintă un domeniu restrâns al parametrilor. Restricția impusă este uzual de forma unei curbe (B-spline), eventual un lanț de segmente de dreaptă aproximând curba reală. În esență operația de tăiere este echivalentă cu determinarea curbei de intersecție a două suprafețe. Se folosesc metode prin subdivizare sau incrementale pentru rezolvarea ecuațiilor rezultante [8][33].

Deoarece operațiile booleene între modelele corpurilor cu suprafețe curbe necesită calcule foarte complexe, mari consumatoare de timp, unele modelatoare de solide din programe CAD folosesc, în locul suprafețelor curbe, suprafețe fațetate. În acest caz, suprafețele curbe sunt approximate prin mulțimi de fațete plane.

4. VIZUALIZAREA ȘI TRANSFORMAREA MODELELOR 3D

Pentru obținerea modelelor anatomice, se prelucrează imagini, iar informația extrasă devine parametru de intrare pentru tehnicile de grafică ce permit reprezentarea modelului. De aceea, este necesar să fie tratate aspecte legate de vizualizarea și transformarea modelelor 3D, aspecte teoretice și practice ce stau la baza rezultatelor obținute în această lucrare.

Modalitatea cea mai simplă de a recepta caracteristicile unui obiect este vizualizarea modelului acestui obiect. Un obiect tridimensional poate fi vizualizat prin:

- afișarea de vederi **ortogonale** (ca în desenul tehnic);
- afișarea prin reprezentarea tuturor muchiilor (**wireframe**);
- afișarea cu efect de **perspectivă**: se definesc un punct de vedere și o direcție, mărimea depinzând de distanța față de observator;
- afișare cu simularea **adâncimii**: intensitatea luminoasă scade cu creșterea distanței față de observator;
- afișarea cu eliminarea **părților ascunse**;
- afișarea cu **nuanțe**: fețele obiectelor au nuanțe diferite în funcție de poziția și orientarea elementelor de suprafață relativ la sursele de lumină și la punctul de privire;
- afișare **stereoscopică**: se redau simultan câte două imagini, câte una pentru fiecare ochi, care sunt privite prin dispozitive speciale;
- afișarea de tip **realitate virtuală** [12][14][30][31][35][44][47][49][67][74].

Extrem de des utilizate sunt reprezentările wireframe și afișarea cu elemente de realism. Reprezentările wireframe au avantajul că toate liniile corpului sunt reprezentate, dar au dezavantajul că permit ambiguități privind identificarea formei reale a obiectului (figura 4-1).

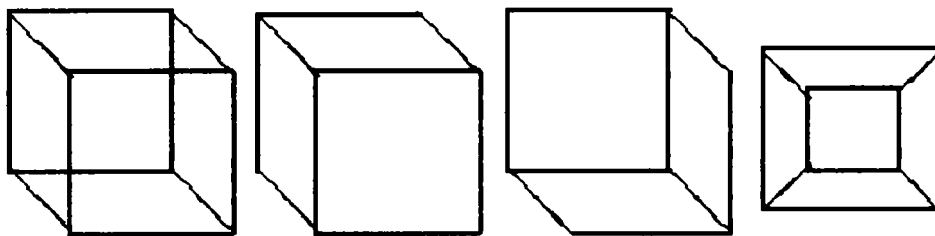


Fig.4-1 Reprezentări wire-frame. Ambiguități.

Aceste ambiguități pot fi eliminate prin trecerea la afișarea cu elemente de realism, adică eliminarea părților ascunse, nuanțare, proces care este numit *rendering* (figura 4-2).

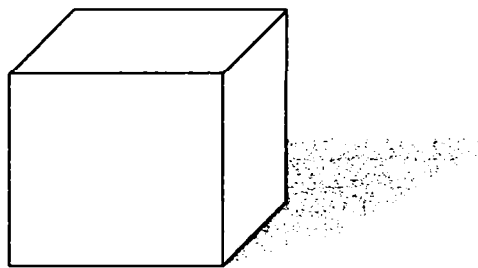


Fig.4-2 Reprezentare realistă a unei scene 3D, ce cuprinde un corp 3D și o sursă de lumină.

4.1. Procedura de vizualizare

Vizualizarea scenelor 3D introduce o proiecție care transformă obiectele din spațiu în proiecții plane. Transformarea de vizualizare 3D se determină pe baza unui volum de vizualizare în spațiul utilizator, o proiecție în plan și o poartă în spațiul dispozitiv. Obiectele 3D ale scenei se decupează la limitele volumului de vizualizare și se proiectează într-un plan. Imaginii proiectate i se aplică apoi o transformare de vizualizare 2D. Modelul conceptual al procesului de vizualizare 3D este reprezentat în figura 4-3.

Firește, diverse sisteme grafice implementează diferit acest model al procesului de vizualizare 3D. Modelul este utilizat în reprezentarea wireframe a obiectelor 3D. La

acesta se adaugă și proceduri pentru creșterea gradului de realism, în această lucrare fiind utilizate metode specifice sistemului OpenGL.

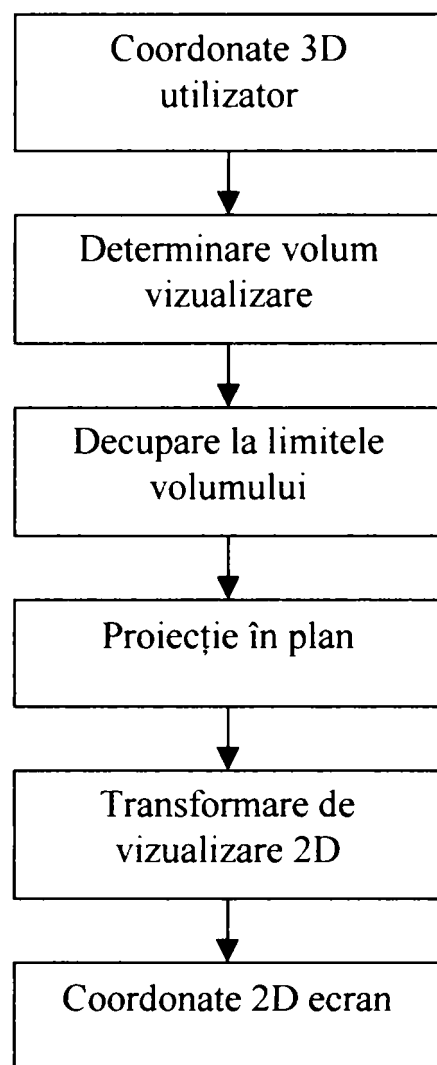


Fig.4-3 Modelul conceptual al procesului de vizualizare 3D

Dezvoltarea algoritmilor pentru manipularea și transformarea unui obiect 3D într-o fereastră de vizualizare pe ecran necesită informații despre:

- poziția obiectului (coordonatele: x , y și z);
- poziția ecranului, caracterizat prin punctul de referință al ferestrei de vizualizare (v_x , v_y , v_z);
- poziția planului normal (care este orientarea ecranului față de planul normal);
- dimensiunea ferestrei de vizualizare și poziția centrului de proiecție;
- direcția de proiecție;
- modul de reprezentare a obiectului.

Prin urmare, transformarea unui obiect 3D poate necesita translații, rotații, modificarea axei Z, înclinări, scalări, toate aceste transformări fiind precedate de proiecții.

4.2. Proiecții

În general proiecțiile transformă spațiul R^n în spațiu R^p , cu $p < n$. În grafică, se utilizează proiecții din spațiul 3D în spațiul 2D. Proiecția unui obiect 3D este dată de mulțimea punctelor de intersecție între planul de proiecție și razele de proiecție, care pleacă dintr-un punct fix, numit *centru de proiecție*, și trec prin fiecare punct al obiectului.

Proiecțiile plane se clasifică în funcție de poziția centrului de proiecție față de planul de proiecție [133][165][44][135][141][141]. Astfel, există:

- **proiecții perspectivă** - în care centrul de proiecție este la distanță finită față de planul de proiecție (figura 4-4);
- **proiecții paralele** - în care centrul de proiecție este la infinit (figura 4-4). În acest caz razele de proiecție sunt linii paralele care trec prin punctele obiectului proiectat. Direcția lor reprezintă *direcția de proiecție*.

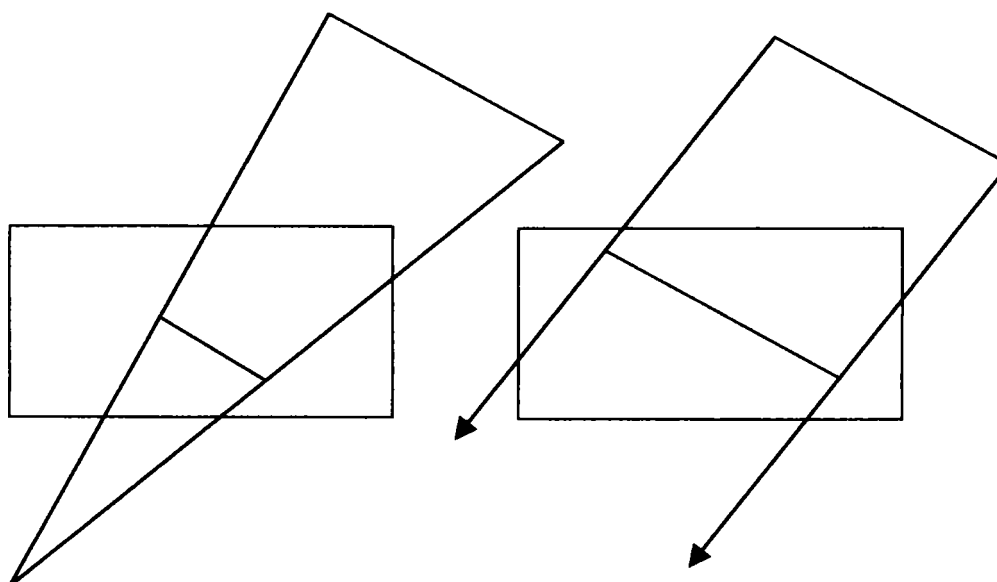


Fig.4-4. Proiecție perspectivă și proiecție paralelă

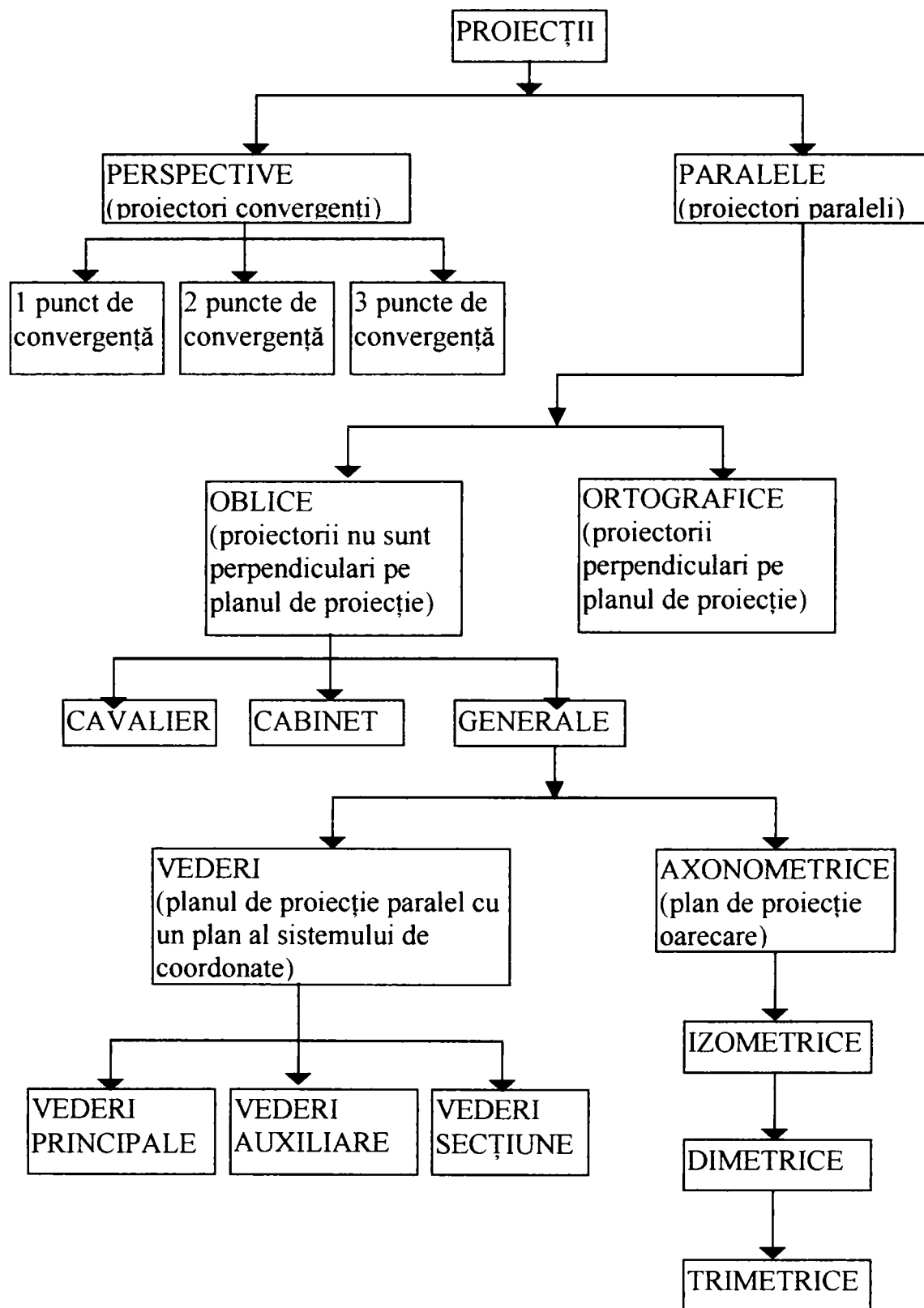


Fig.4-5. Clasificarea proiecțiilor plane

4.2.1. Proiecția perspectivă

Proiecția perspectivă creează imagini asemănătoare celor obținute cu un aparat de fotografiat. Este caracterizată de poziția centrului de proiecție și de planul de proiecție. Astfel, liniile paralele între ele și neparalele cu planul de proiecție converg spre un *punct de fugă*. Dacă liniile sunt paralele cu axele principale, punctul de fugă este *punct de fugă axial*. Într-o asemenea proiecție pot exista cel mult trei puncte de fugă. Spre exemplu, dacă planul de proiecție intersectează axa Oz atunci doar această axă are punct de fugă, întrucât liniile paralele cu axele Oy și Ox sunt paralele și în planul de proiecție (nu au puncte de fugă). Din acest motiv, numărul punctelor de fugă reprezintă un criteriu important de clasificare a proiecțiilor perspective [133][165][44][164][159] (figura 4-6).

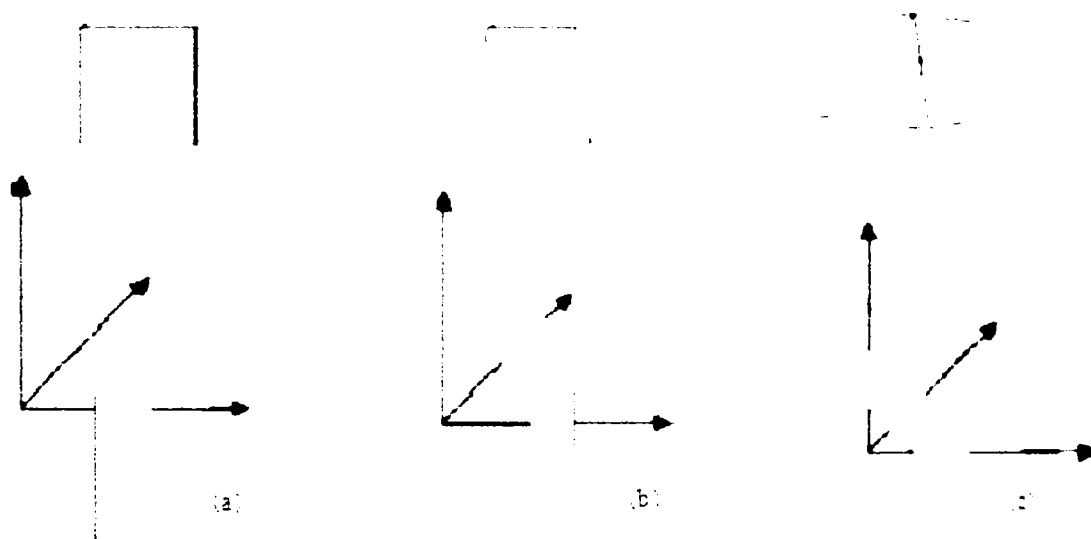


Fig.4-6. Proiecții perspectivă cu unul (a), două (b) și trei (c) puncte de fugă.

Caracteristic acestor proiecții este efectul de *zoom*. Dimensiunea obiectului proiectat este invers proporțională cu distanța de la centrul de proiecție la obiect. Deci, cu cât obiectul este situat la o distanță mai mare de centrul de proiecție, cu atât reprezentarea sa în plan este mai mică. De exemplu, în figura 4-7, aceeași sferă are o proiecție diferită în planul $z=0$, în funcție de distanța la care se află față de centrul de proiecție..

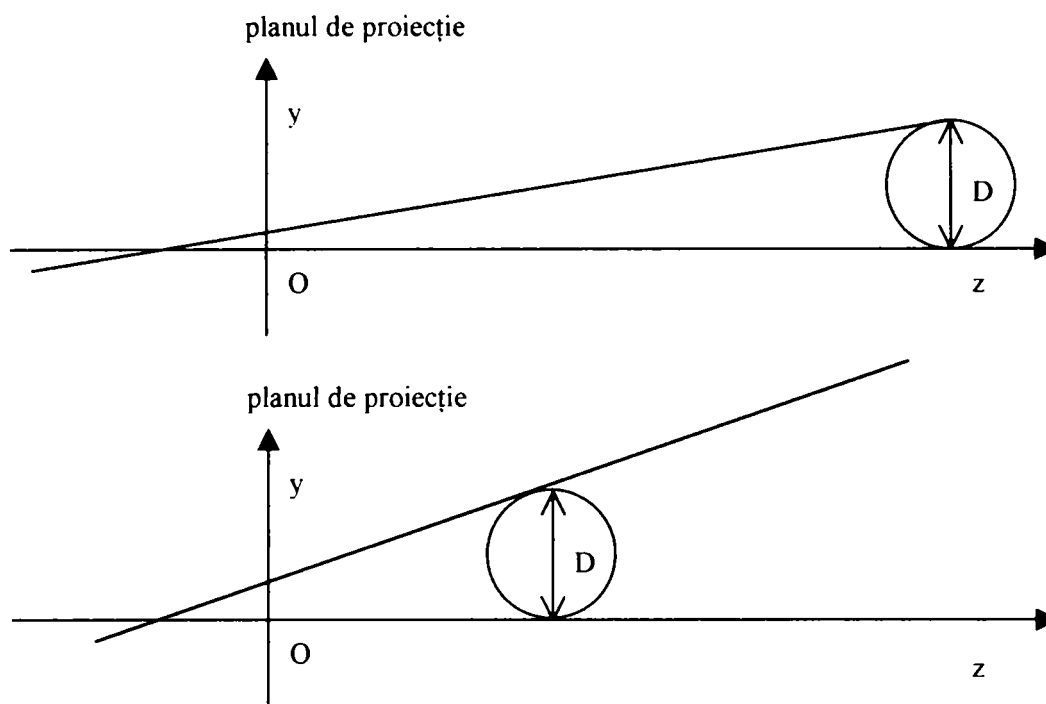


Fig.4-7 Dimensiunea proiecției este cu atât mai mare cu cât obiectul se află mai aproape de centrul de proiecție.

În funcție de înclinarea planului de proiecție față de centrul de proiecție și de obiect, pot fi identificate două tipuri de proiecții perspective:

- perspectiva ascendentă** (punctul de fugă este situat deasupra obiectului): este cazul centrului de proiecție privind obiectul de jos și aproape.
- perspectiva descendentă** (punctul de fugă este situat sub obiect): este cazul centrului de proiecție privind obiectul de sus și aproape.

Pentru determinarea coordonatelor în perspectivă pentru un punct se rezolvă sistemul format de ecuația planului de proiecție și ecuația razei vizuale. Acest mod de procesare conduce la calcule greoaie, necesitând un efort de calcul semnificativ.

Deși proiecția perspectivă prezintă o imagine realistă, nu oferă o reprezentare exactă a formelor și dimensiunilor obiectelor.

4.2.2. Proiecția paralelă

Proiecția paralelă oferă o imagine mai puțin realistă, fiind utilizată pentru o reprezentare exactă a formei și dimensiunii obiectului. Această proiecție păstrează paralelismul liniilor, iar unghiurile se conservă doar pentru fețele obiectului paralele cu planul de proiecție [133][165][44][164][159].

Proiecția ortografică este cea mai simplă proiecție paralelă. Ea prezintă corect forma și dimensiunea fețelor plane ale obiectului care sunt paralele cu unul din planele sistemului de coordonate (figura 4-8). Proiecțiile ortografice sunt proiecții în planele sistemului de coordonate $x=0$, $y=0$, $z=0$.

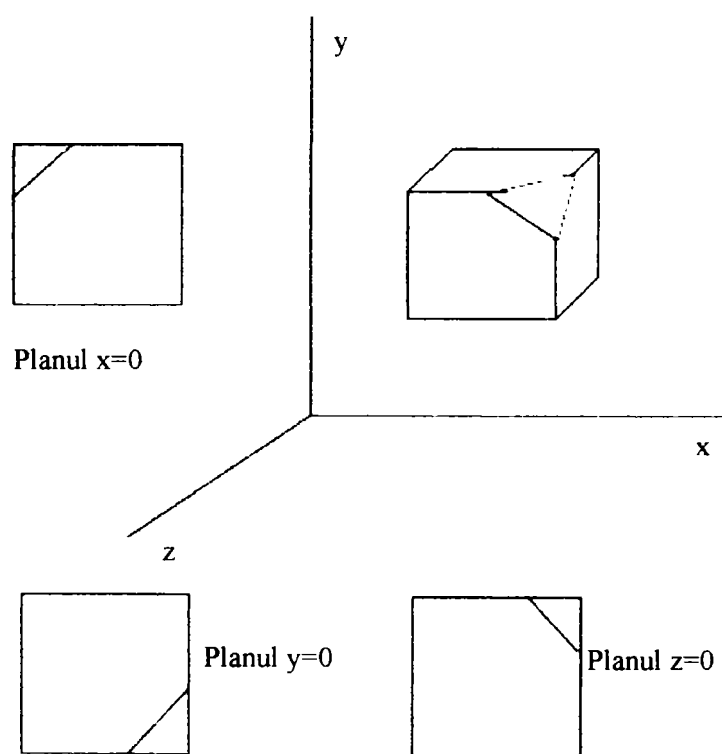


Fig.4-8. Proiecția ortografică

Proiecția axonometrică este o proiecție paralelă ortografică (cu raze de proiecție perpendiculare pe planul de proiecție) în care planul de proiecție nu este perpendicular pe nici una din axele sistemului de coordonate. În funcție de poziția normalei la planul de proiecție există trei tipuri de proiecții axonometrice: izometrică, dimetrică, trimetrică. Un exemplu de proiecție trimetrică este dat în figura 4-9.

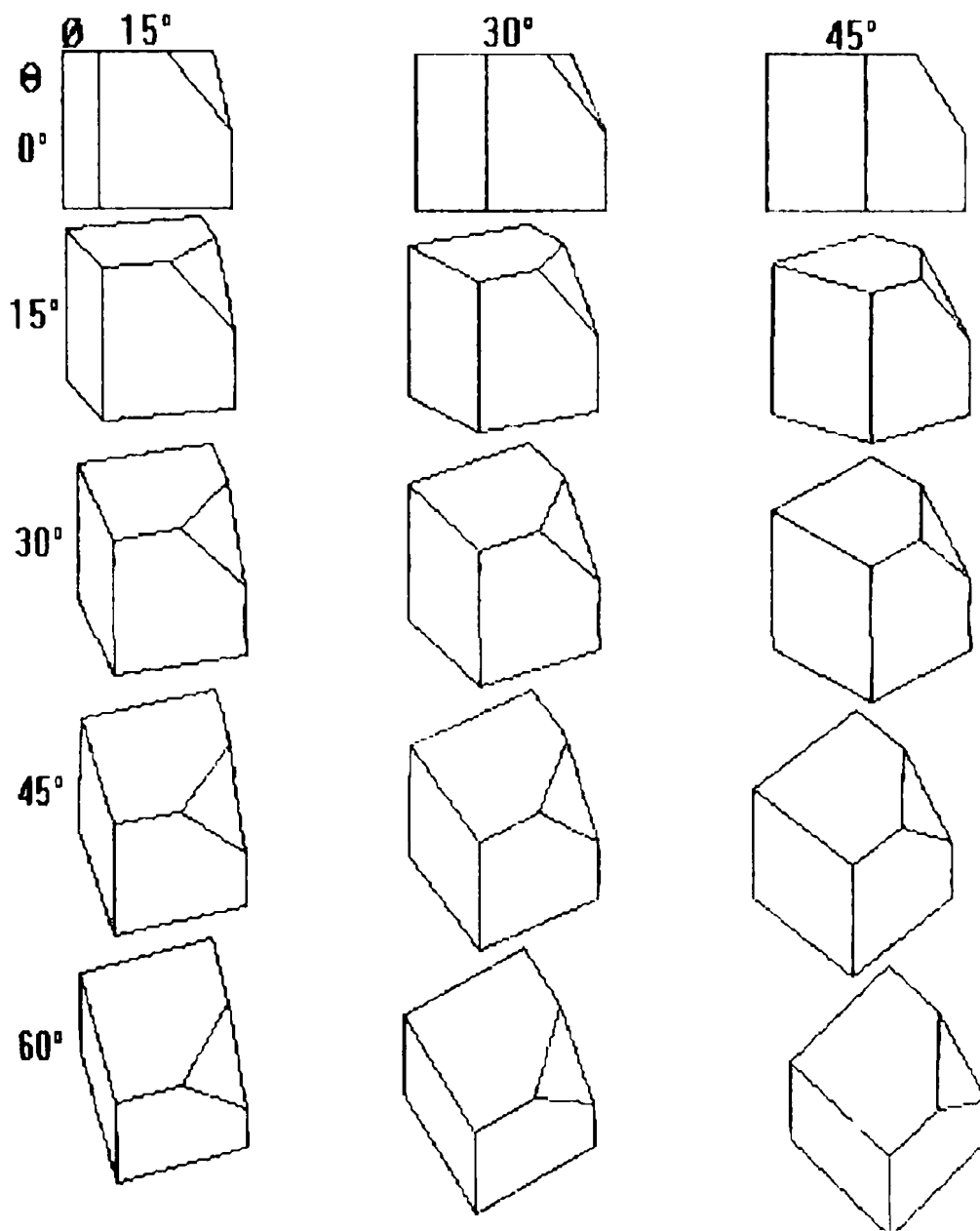


Fig.4-9 Proiecție axonometrică trimetrică

O proiecție trimetrică se obține prin una sau mai multe rotații în jurul axelor de coordonate, urmate de o proiecție în planul $z=0$. În figura 4-9, proiecțiile sunt formate prin rotație cu unghiul ϕ în jurul axei y , rotație cu unghiul θ în jurul axei x și apoi proiecție ortografică în planul $z=0$ [133].

Proiecția oblică este tot o proiecție paralelă, în care direcția de proiecție nu este perpendiculară pe planul de proiecție. Planul de proiecție este perpendicular pe una din axele sistemului de coordonate. De regulă, planul de proiecție este $z=0$. Proiecțiile oblice reprezintă forma generală a obiectelor 3D. Doar fețele paralele cu planul de proiecție sunt redată la forma și dimensiunile lor reale. Se conservă astfel unghiurile și lungimile numai

pentru aceste fețe. Fețele neparalele cu planul de proiecție sunt distorsionate. Există două tipuri de proiecții oblice: proiecția **cavalier**, unde direcția de proiecție formează un unghi de 45 de grade cu planul de proiecție, și proiecția **cabinet** unde respectivul unghi este de 63.43 grade [133][165][164]179][180][181].

4.3 Transformări grafice aplicate modelelor

Pentru a beneficia de cât mai multe informații pe care un model le furnizează, acesta trebuie să poată fi *învârtit pe toate fețele* adică fiecare detaliu al său să poată fi explorat din toate unghiurile. Din acest motiv modelelor li se aplică transformări geometrice tridimensionale: scalare, translație, rotație, toate urmate, evident, de proiecție.

Un punct din spațiu (x,y,z) se reprezintă în coordonate omogene prin vectorul

$[x_w, y_w, z_w, w]$ unde w este parametru real, iar $x_w = x w$, $y_w = y w$, $z_w = z w$. Punctele pentru care w este zero sunt puncte la infinit.

Matricea de transformare generalizată pentru coordonate omogene 3D are forma:

$$[T] = \begin{bmatrix} a & b & c & p \\ d & e & f & q \\ g & h & i & r \\ l & m & n & s \end{bmatrix} \quad (4-1)$$

Această matrice poate fi împărțită în patru, astfel[181][178][179][180][182][185]:

$$\begin{bmatrix} 3 & \times & 3 & 3 \\ & & & \times \\ & & & 1 \\ \dots & \dots & \dots & \\ & 1 \times 3 & & 1 \times 1 \end{bmatrix}$$

unde:

- matricea 3x3 include transformări de scalare locală și rotație;
- matricea 1x3 reprezintă transformarea de translație;

-matricea 3x1 reprezintă transformarea perspectivă;

-matricea 1x1 reprezintă transformarea de scalare generală.

Translația. Dacă (x,y,z) sunt coordonatele unui punct P în spațiu, el este dus prin translație în punctul $P1$ de coordonate $(x1, y1, z1)$, unde:

$$x1 = x + t_x \quad y1 = y + t_y \quad z1 = z + t_z \quad (4-2)$$

Sau, în forma matriceală:

$$[x1 \ y1 \ z1 \ 1] = [x \ y \ z \ 1] [T] \quad (4-3)$$

unde $[T]$ este matrice de translație 3D și are forma [GR10][GR11][GR12]:

$$[T] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ t_x & t_y & t_z & 1 \end{bmatrix} \quad (4-4)$$

Scalarea. Dacă (x,y,z) sunt coordonatele unui punct P în spațiu, el este dus prin scalare în punctul $P1$ de coordonate $(x1, y1, z1)$, unde:

$$x1 = x \cdot s_x \quad y1 = y \cdot s_y \quad z1 = z \cdot s_z$$

Sau, în forma matriceală:

$$[x1 \ y1 \ z1 \ 1] = [x \ y \ z \ 1] [S] \quad (4-5)$$

unde $[S]$ este matrice de scalare locală și are forma [133][179][180][181]:

$$[S] = \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4-6)$$

Matricea de scalare globala are forma:

$$[S] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & s \end{bmatrix} \quad (4-7)$$

Daca factorul de scalare globala este subunitar se produce o marire a vectorului de pozitie, iar daca este supraunitar atunci se produce o micșorare a vectorului de pozitie.

Rotația în jurul unei axe. În cazul rotației în jurul axei x , coordonatele vectorilor de poziție nu se schimbă. Rotația apare în plane perpendiculare pe axa Ox . În mod similar, în cazul rotației în jurul axei Oy sau Oz , coordonatele y , respectiv z ale vectorilor de poziție nu se schimbă; rotația se efectuează în plane perpendiculare pe axa y , respectiv pe axa z . Matricele de rotație în jurul axelor Ox , Oy , respectiv Oz cu unghiul θ sunt:

$$[R_x] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad [R_y] = \begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad [R_z] = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4-8)$$

Pentru cazul general, al rotației cu unghiul θ față de o axă de versor $[w_1 \ w_2 \ w_3]^T$ ce trece prin origine, matricea de transformare are expresia:

$$[R] = \begin{bmatrix} c + (1-c)w_1^2 & -w_3s + (1-c)w_1w_2 & w_2s + (1-c)w_1w_3 & 0 \\ w_3s + (1-c)w_1w_2 & c + (1-c)w_2^2 & -w_1s + (1-c)w_2w_3 & 0 \\ -w_2s + (1-c)w_1w_3 & w_1s + (1-c)w_2w_3 & c + (1-c)w_3^2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4-9)$$

în care $c = \cos(\theta)$ și $s = \sin(\theta)$ [133][164][165][179][180][181].

4.4. Redarea realistă a imaginilor

Redarea realistă a imaginilor cu ajutorul efectelor de iluminare și umbră se poate face prin *ray-tracing*. Algoritmul folosește un model de iluminare care calculează culoarea unui punct al unei suprafețe ținând cont de iluminarea locală și iluminarea globală. Iluminarea locală este dată de lumina care vine direct de la surse. Iluminarea globală într-un punct al unei suprafețe este dată de lumina care se reflectă sau se transmite prin refracție de la alte obiecte în acel punct. Algoritmul generează imaginea în funcție de poziția observatorului, urmărind traseul unor raze de lumină imaginare care pornesc de la observator la obiectele scenei 3D. Deci este un algoritm dependent de poziția observatorului.

Algoritmul *ray-tracing* realizează implicit operația de eliminare a părților ascunse (invizibile) observatorului. Operația de bază în *ray-tracing* o reprezintă calculul intersecției între o rază și un obiect.

În figura 4-10 este prezentat cazul unei mingi care se reflectă într-o oglindă.

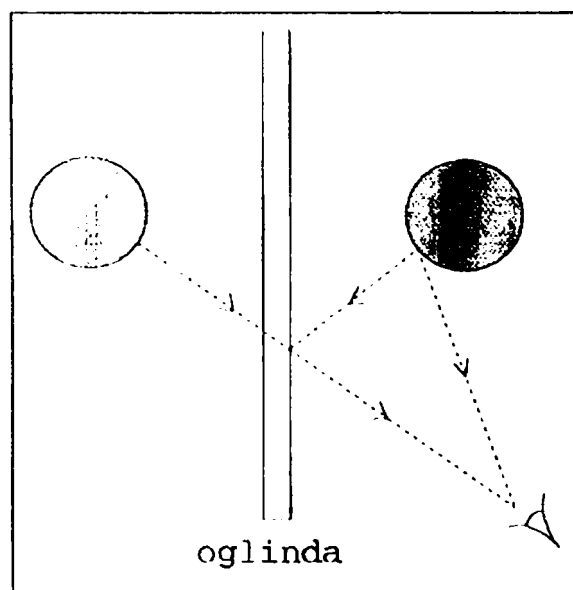


Fig.4-10 Razele de lumină pot ajunge pe două direcții diferite

Razele de lumină provenind de pe suprafața mingii pot ajunge la observator pe două direcții diferite: direct sau reflectate de suprafața oglinzii. O asemenea situație este reprezentată în figura 4-10. Se creează impresia că există două mingi, cea reală și imaginea ei în oglindă, care dă senzația că se află în spatele oglinzii.

Se poate acum imagina o scenă cu mai multe oglinzi și cu obiecte cu suprafețe lucioase; toate oferă mai multe căi pentru ca razele de lumină să ajungă la observator. Se vor vedea mult mai multe imagini, ca urmare a oglinzirilor repetate. Este dificil de reprodus exact ceea ce se întâmplă în natură. Ar trebui să fie trimise raze de lumină din toate sursele punctiforme de lumină din scenă, în toate direcțiile, și să se urmărească modul în care ele se reflectă la întâlnirea unei suprafețe lucioase. Dacă o rază de lumină cade pe o suprafață care nu este chiar lucioasă, ar trebui împrăștiată într-un număr de alte raze mai reduse ca intensitate, și să fie urmărite și acestea. Fiecare rază poate să se piardă în spațiu sau să cadă pe un obiect care nu reflectă, sau să ajungă la observator.

Atunci când o rază de lumină ajunge la ochi și se sesizează culoarea. Această culoare este ceea ce a rămas din starea inițială a razei. Inițial, raza are mai multe componente de culoare. Atunci când cade pe o suprafață lucioasă și se modifică direcția și, mai puțin, componența. Dacă ea cade pe o suprafață absorbantă, unele componente sunt parțial sau total absorbite de către suprafață. Componentele rămase sunt împrăștiate în toate direcțiile. Una din aceste raze - cea care ajunge la observator - poate să mai cadă și pe alte obiecte care absorb și astfel și se modifică spectrul. În final, se poate întâmpla să fie sesizate doar câteva componente rămase.

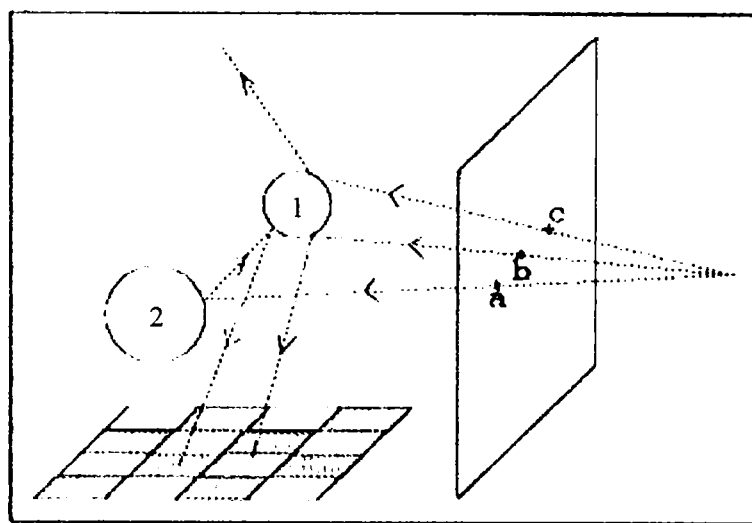


Fig.4-11 Ray-tracing

Practic ar fi imposibil să se imagineze direcția unei raze de lumină pornind dintr-un anumit punct, astfel încât după mai multe reflexii, refracții, absorbții, etc., ea să ajungă la persoana care privește o scenă. Este mult mai simplu să se urmărească traseul invers parcurs de o rază în drumul său spre observator. Se poate realiza aceasta plasând

observatorul în centrul de proiecție, deoarece numai razele care ajung în acest punct contribuie la perceperea imaginii. Contribuția la imagine a unei asemenea raze este intensitatea pe care o are atunci când ajunge la observator.

Pentru exemplificarea modului în care se realizează imaginea se presupune o scenă formată din două sfere (fig.4-11), complet lucioase, care plutesc deasupra unei table de șah, și se urmărește calea parcursă de o rază de lumină înainte de a ajunge la observator. Această operație se face în sens invers, și se aplică o singură dată pentru fiecare pixel al ecranului. Raza care trece prin pixelul a lovește sfera 2 apoi sfera 1 apoi cade pe tabla de șah pe un pătrat negru, deci pixelul va fi negru. Raza care trece prin b cade pe sfera 1, apoi pe tabla de șah pe un pătrat alb, deci pixelul b va fi alb. Raza care trece prin c cade pe sfera 1 și iese în spațiu, deci pixelul va lua culoarea fondului. Raza este o dreaptă care trece prin centrul proiecției $(0,0,-d)$ și prin centrul unui pixel cu locația (x,y) în coordonate absolute și (x_s, y_s) în coordonate ale spațiului obiect. Fie dreapta descrisă prin:

$$x = x_s \cdot u \quad y = y_s \cdot u \quad z = d \cdot u - d \quad (4-10)$$

Această rază va cădea pe o sferă și trebuie determinat pe care din ele. Ecuțiile sferelor sunt:

$$(x - x_{c1})^2 + (y - y_{c1})^2 + (z - z_{c1})^2 - r_1^2 = 0 \quad (4-11)$$

$$(x - x_{c2})^2 + (y - y_{c2})^2 + (z - z_{c2})^2 - r_2^2 = 0 \quad (4-12)$$

unde (x_{c1}, y_{c1}, z_{c1}) și r_1 sunt centrul, respectiv raza primei sfere, și similar pentru sfera a doua. Se rezolvă ecuația de gradul doi în u :

$$u^2(x_s^2 + y_s^2 + d^2) - 2u(x_s x_{c1} + y_s y_{c1} + d^2 + d z_{c1}) + x_{c1}^2 + y_{c1}^2 + (d + z_{c1})^2 - r_1^2 = 0 \quad (4-13)$$

La fel pentru a doua sferă. Dacă ecuațiile nu au soluții reale, înseamnă că nu există intersecții cu sferile. În acest caz se caută intersecții cu fiecare obiect din scenă. Dacă

există soluții reale, cea mai mică valoare a lui u corespunde unui punct de intrare a razei în cea mai apropiată sferă, deci un punct vizibil [80][98][103][104][4][14][31][35][49].

Metoda prezentată stă la baza mai multor sisteme de rendering a scenelor 3D, fiind cuprins și în sistemul grafic OpenGL, utilizat în teză.

4.5 Interfața software OpenGL

La baza algoritmilor și programelor pentru metodele utilizate în teză pentru modelarea și reprezentarea grafică a structurilor anatomice tridimensionale, se află biblioteca **Open Graphics Library (OpenGL)**, tehnologie a **Silicon Graphics**[18][85][90][101][131][172][174][175][194][204][205][210][211]. Aceasta este o interfață software ce constă dintr-o colecție de proceduri și funcții, care permit programatorului să specifice obiectele componente ale unei scene 3D și operațiile necesare a fi efectuate cu această scenă. În cazul tezei având ca obiectiv reprezentarea structurilor anatomice complexe: furnizarea sistemului de bază de rendering, toate metodele, algoritmi dezvoltati în lucrare pentru prelucrarea imaginilor, detecția zonelor de interes, segmentarea și identificarea obiectelor din imagini, detecția izosuprafețelor și reconstrucția 3D a structurilor de interes detectate având ca finalitate afișarea și posibilitatea de manipulare a modelului pe ecran, reprezintă contribuții personale ale autorului implementate în cadrul oferit de OpenGL.

OpenGL este portabilă pe mai multe platforme software, fiind independentă de sistemul de operare. Ea rulează la fel de bine pe un sistem Windows, ca și pe un sistem Unix. În plus, fiind un standard, beneficiază de suportul producătorilor de hardware, ce oferă acceleratoare grafice cu suport OpenGL, performanța și compatibilitatea fiind împinse la maximum în acest mod.

La baza reprezentărilor cu ajutorul OpenGL stau primitivele grafice: puncte, segmente de linii, poligoane, fiecare având propriile “stări”, moduri de afișare sau de lucru selectabile. Fiecare stare poate fi controlată independent, dar există și moduri de lucru ce permit setarea stărilor primitivelor grafice la modul global, orice modificare fiind vizibilă în framebuffer. Nu există comenzi predefinite pentru crearea de obiecte complexe. Orice obiect complex va fi definit numai prin primitive. În schimb, OpenGL cuprinde tehnici complexe pentru obținerea efectelor de spațialitate, de adâncime, de netezire, umbrire, iluminare, texturare, transformări geometrice, etc. Pentru specificarea

primitivelor, caracteristicilor acestora, precum și a altor operații specifice, comenzile sunt sub forma apelurilor de funcții. Primitivele sunt definite prin unul sau mai multe puncte (vertex). Un vertex definește un punct, un capăt al unui segment sau un colț al unui poligon unde se intersectează două muchii. Datele, constituite din coordonatele vertexurilor, culori, normale, coordonate texturi și stări ale muchiilor, spre exemplu, sunt asociate primitivei. Comenzile sunt întotdeauna executate în ordinea în care sunt furnizate, acest lucru însemnând că fiecare primitivă este desenată complet înainte de a se lua în considerare orice altă comandă ulterioară. Comenzile de interogare a stărilor primitivelor vor returna în acest mod întotdeauna informații consistente, în concordanță cu operațiile executate complet până la momentul interogării.

Legarea datelor se face la apel. Aceasta înseamnă că datele trimise unei comenzi sunt interpretate atunci când comanda este primită. Chiar dacă comanda face referire la un pointer la o dată, și această dată este interpretată atunci când apare apelul.

OpenGL permite controlul direct al operațiilor fundamentale pentru grafica 3D și 2D. Aceasta include specificarea parametrilor matricilor de transformare, coeficienții ecuațiilor de iluminare, metode de antialiasing, și operații pe pixel.

Modelul interpretării comenzilor GL este **client - server**. Aceasta înseamnă că un program (clientul) dă comenzi, și aceste comenzi sunt interpretate și procesate de GL (server). Serverul poate sau nu să opereze pe același computer ca și clientul. Un server poate menține un număr de contexte GL, fiecare fiind o încapsulare a unei stări GL curente. Un client poate alege să se conecteze la una dintre aceste contexte.

Un program tipic ce utilizează OpenGL deschide mai întâi o fereastră în framebuffer în care programul va desena. Apoi se declară și se alocă un context specific OpenGL care va fi asociat cu fereastra de desenare. Odată alocat contextul GL programatorul este liber să înceapă descrierea scenei și să comande ce se va întâmpla pe scenă. Se pot crea modele extrem de complicate la fel ca în lumea reală, bazate pe iluminări, umbriri, mapări de imagini, colorări, texturări, având la dispoziție un mecanism de **rendering** extrem de eficient și de rapid. Se pot crea obiecte complicate pornind de la unele simple asupra cărora se aplică transformări și diverse efecte sau prin citirea unor fișiere speciale în care sunt memorate datele ce descriu în detaliu corpul. Etapele parcurse în formarea unei imagini sunt:

- specificarea geometriei (coordonatele reale);
- specificarea camerei (coordonatele camerei);
- proiectarea (coordonate plan);

-maparea în viewport (coordonate ecran).

4.5.1. Descrierea structurii unui program bazat pe OpenGL și a comenzilor de bază

Pentru ca un program OpenGL să poată fi compilat corect, compilatorul trebuie să facă legături cu bibliotecile OpenGL, deci este necesară specificarea acestora la opțiunile de compilare și linkeditare ale mediului de lucru, Visual C++ 6.0: **OpenGL32.lib** **GLu32.lib** și **GLaux.lib**[202][127][128][129][130][131].

Primul pas este declararea fișierelor header:

```
#include <windows.h>
#include <gl\gl.h>
#include <gl\glu.h>
#include <gl\glaux.h>
```

Aceste fișiere sunt necesare pentru programarea în mediul Windows (**windows.h**) și OpenGL (**gl.h**, **glu.h**, **glaux.h**). Urmează declararea variabilelor folosite la crearea ferestrei aplicației, și câteva variabile necesare pe parcursul programului de inițializare.

```
HDC hDC=NULL;
HGLRC hRC=NULL;
HWND hWnd=NULL;
HINSTANCE hInstance;
```

```
bool keys[256];
bool active=TRUE;
bool fullscreen=TRUE;
```

Prima variabilă, **keys[256]**, este un vector ce va monitoriza tastele apășate de utilizator. Este o metodă eficientă, deoarece poate lucra și cu mai multe taste simultan. A doua variabilă, **active**, ce are valoarea initială **TRUE**, va fi folosită pentru a testa dacă programul este sau nu activ; astfel el va putea fi suspendat în cazul în care a fost minimizat. Ultima variabilă, **fullscreen**, va memora starea programului, modul fullscreen sau fereastră.

Pentru ca aplicația OpenGL să poată rula în orice rezoluție și în orice adâncime de culoare se vor defini 3 constante:

```
#define xrez 800
#define yrez 600
#define color_depth 16
```

Apoi se declară prototipul funcțiilor pentru a evita erorile de compilare:

```
LRESULT CALLBACK WndProc(HWND, UINT, WPARAM, LPARAM);
int DrawGLScene(void);
```

Funcția ReSizeGLScene

```
GLvoid ReSizeGLScene(GLsizei width, GLsizei height)
{
if (height==0) // Evitarea unei impartiri la 0 (zero)
{
height=1; // Daca inaltimea era 0, acum va lua valoarea 1
}
glViewport(0,0,width,height); // Reseteaza viewport-ul curent
glMatrixMode(GL_PROJECTION); // Selecteaza matricea pentru proiectii
glLoadIdentity(); // Reseteaza matricea curenta (cea selectata anterior)
gluPerspective(45.0f, (GLfloat)width/(GLfloat)height, 0.1f, 100.0f);
//Calculeaza aspectul ferestrei
glMatrixMode(GL_MODELVIEW); // Selecteaza matricea pentru modele
glLoadIdentity(); // Reseteaza matricea curenta (cea selectata anterior)
}
```

Această funcție va fi apelată în momentul în care fereastra este redimensionată. Dacă sistemul de operare ar face redimensionarea automat, scena afișată în acel moment pe ecran nu va fi recalculată în funcție de dimensiunile ferestrei după redimensionare, deci va fi afișată incomplet.

Funcția acceptă 2 parametri, *width* și *height*, care reprezintă lungimea și înălțimea ferestrei. Întâi, ea se asigură că înălțimea ferestrei nu este 0 (urmează o împărțire), apoi resetează viewport-ul curent în funcție de dimensiunile ferestrei prin funcția **glViewport**. Următoarea linie selectează matricea pentru proiecții, care este resetată prin **glLoadIdentity()**. Perspectiva este calculată cu un unghi de 45 de grade în funcție de dimensiunile ferestrei.

Funcția InitGL

```
int InitGL(GLvoid)
{
glShadeModel(GL_SMOOTH);
glClearColor(0.0f, 0.0f, 0.0f, 0.5f);
// Seteaza culoarea fundalului (negru)
glClearDepth(1.0f); // Curata buffer-ul de adancime
glEnable(GL_DEPTH_TEST); // Activeaza testarea in adancime
glDepthFunc(GL_LEQUAL); // Seteaza tipul testarii
glHint(GL_PERSPECTIVE_CORRECTION_HINT, GL_NICEST);
// Seteaza tipul de calcul pentru perspectiva
return TRUE; // Initializarea s-a incheiat cu succes
}
```

Aici vor fi inițializate cele mai multe elemente OpenGL, cum ar fi texturile, sursele de lumină, transparentă, etc.

Prima linie este o funcție prin care OpenGL setează modul de afișare a culorilor. Următoarea funcție **glClearColor** setează culoarea fundalului, iar următoarele funcții, **glClearDepth** și **glDepthFunc** setează buffer-ul de adâncime. Aceste funcții au grijă ca

obiectele să fie desenate corect în perspectivă. În final, ultima funcție, **glHint**, alege modul în care OpenGL va face calculele pentru afișaj.

Funcția DrawGLScene

```
int DrawGLScene(GLvoid)
{
glClearColor(0.0f, 0.0f, 0.0f, 0.5f);
glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
glLoadIdentity(); // Setează poziția în originea sistemului de referință
return TRUE; //
}
```

Aici vor fi scrise toate liniile de cod ce vor afișa imaginile OpenGL pe ecran. Practic această funcție descrie scena 3D pe care programatorul dorește să o afișeze pe ecran. Rezultatele tuturor transformărilor, calculelor, vor avea un corespondent în această funcție ce va fi apelată oricând este nevoie de randarea scenei dorite.

Alternativ, se pot utiliza, funcții similare din librăria GLUT (Graphics Library Utility Tools), care execută aceleași operațiuni.

```
void main( int argc, char** argv )
{
int mode = GLUT_RGB|GLUT_DOUBLE;
glutInitDisplayMode( mode );
glutCreateWindow( argv[0] );
init();
glutDisplayFunc( display );
glutReshapeFunc( resize );
glutKeyboardFunc( key );
glutIdleFunc( idle );
glutMainLoop();
}

void display( void )
{
glClear( GL_COLOR_BUFFER_BIT );
glBegin( GL_TRIANGLE_STRIP );
glVertex3fv( v[0] );
glVertex3fv( v[1] );
glVertex3fv( v[2] );
glVertex3fv( v[3] );
glEnd();
glutSwapBuffers();
}
```

Funcția **glutKeyboardFunc(keyboard)** este tot callback și răspunde de interacțiunea cu utilizatorul.

```
void keyboard( char key, int x, int y )
{
switch( key ) {
case 'q' : case 'Q' :
exit( EXIT_SUCCESS );
break;
case 'r' : case 'R' :
rotate = GL_TRUE;
break;
}
}
```

Funcția **glutIdleFunc(idle)** reprezintă mecanismul de realizare a animației. Funcția **idle()** este cea care schimbă valorile variabilelor de mișcare în program, de obicei, variabile globale, iar **glutIdleFunc(idle)** efectuează cereri de refresh a scenei.

```
void idle( void )  
{  
t += dt;  
glutPostRedisplay();  
}
```

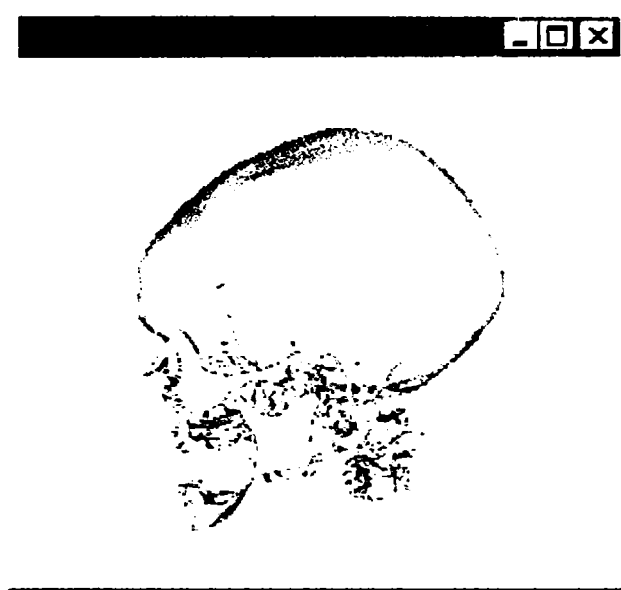


Fig.4-12. Imagine afișată prin tehnologia OpenGL

5. RECONSTITUIREA ȘI MODELAREA 3D A STRUCTURILOR ANATOMICE COMPLEXE

5.1 Importanța și aplicabilitatea

Se prezintă un ansamblu complet de metode ce se constituie într-o soluție originală și eficientă de reconstrucție a suprafețelor tridimensionale și generare de modele tridimensionale ale structurilor anatomice complexe.

Reprezentarea tridimensională a suprafețelor anatomice oferă o metodă de investigare medicală valoroasă. Imagini ale acestor suprafețe, construite pe baza planelor multiple bidimensionale rezultate din investigații prin tomografie computerizată, rezonanța magnetică, respectiv tomografie computerizată cu emisie de fotoni, ajută medicii să stabilească un diagnostic corect pe baza imaginilor redade de echipamente sub formă 2D. Interpretarea imaginilor medicale bidimensionale necesită cunoștințe speciale și experiența specialiștilor care trebuie să colaboreze și să comunice interpretările lor medicilor care vor aplica tratamentul. În multe cazuri, există dificultăți în interpretarea și imaginarea structurii tridimensionale reale pe baza imaginilor 2D furnizate de echipamentele de investigare imagistică.

Tomografia computerizată furnizează imagini ce reprezintă atenuarea radiațiilor X în țesuturile vii. Imaginile sunt achiziționate ca o succesiune de plane paralele care, interpretate împreună, reprezintă o distribuție tridimensională bazată pe aprecierea densității relative a țesuturilor. Astfel, poate fi investigată structura internă a corpului, indiferent că este urmărit aparatul osos sau țesuturi moi.

Rezonanța magnetică furnizează un contrast foarte bun între varietățile de țesuturi moi investigate. Rezonanța magnetică măsoară trei proprietăți fizice. O proprietate este distribuția nucleilor de hidrogen mobili care furnizează structura de ansamblu a planului de informație. Celelalte două proprietăți măsoară timpul de relaxare al nucleilor, obținându-se astfel diferența între diferitele tipuri de țesuturi.

Tomografia computerizată cu emisie de fotoni măsoară emisia de radiații gamma. Sursa acestor raze este un radioizotop introdus în corp. Pe lângă structură, acest tip de investigație poate semnala prezența sângelui în structura organului, cu o acuratețe mai mare decât în tomografia computerizată.

5.2 Principiul și etapele soluției propuse

Soluția propusă se bazează pe un ansamblu de metode ce includ tehnici specifice analizei și prelucrării de imagini, extragerii de informație utilă din acestea, reprezentării cu ajutorul graficii 3D a modelului construit pe baza informațiilor utile extrase și manipularea acestui model în scopuri asociate studiului sau prelucrării ulterioare.

Tehnicile de prelucrare prezentate în teză iau în considerare tomograme obținute pe Computerul Tomograf Somatom Plus 4, de producție Siemens, care are propriul său software de reconstrucție 3D, Magic View 300. Rezultatele obținute de autor în cadrul tezei vor fi comparate cu performanțele proprii ale echipamentului menționat.

Se prezintă:

- citirea și organizarea datelor de intrare;
- tehnicile de segmentare, identificare și clasificare automată și manuală a obiectelor din imagini (utile atunci când se dorește detectarea și punerea în evidență a unei regiuni de interes);
- filtrul de volum pentru detecția izosuprafețelor dorite, ce are la bază algoritmul de deplasare a cuburilor;
- construirea și vizualizarea modelului rezultat;
- modalități de utilizare;
- avantajele utilizării modelelor generate;
- rezultatele obținute;
- posibilitatea de export a modelului în formate de fișiere 3D accesibile programelor clasice de CAD.

Ideea de bază a tezei este de a crea o reprezentare poligonală a suprafețelor de densitate constantă, pornind de la un tablou tridimensional de date. Modelul rezultat este afișat utilizând algoritmi de rendering sau exportat în fișiere VRML (realitate virtuală), STL (stereolitografie) sau DXF (AutoCAD).

Etapele ce sunt parcurse sunt prezentate schematic în figura 5-1:

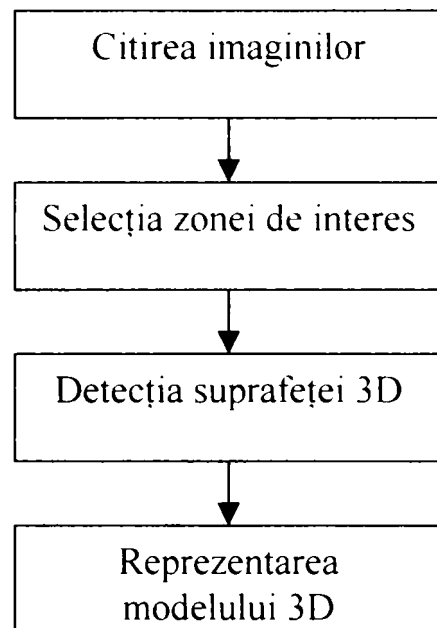


Fig.5-1 Etapele soluției propuse

Se realizează:

- citirea și organizarea datelor, imagini ale tomografiilor computerizate (figura 5-2):

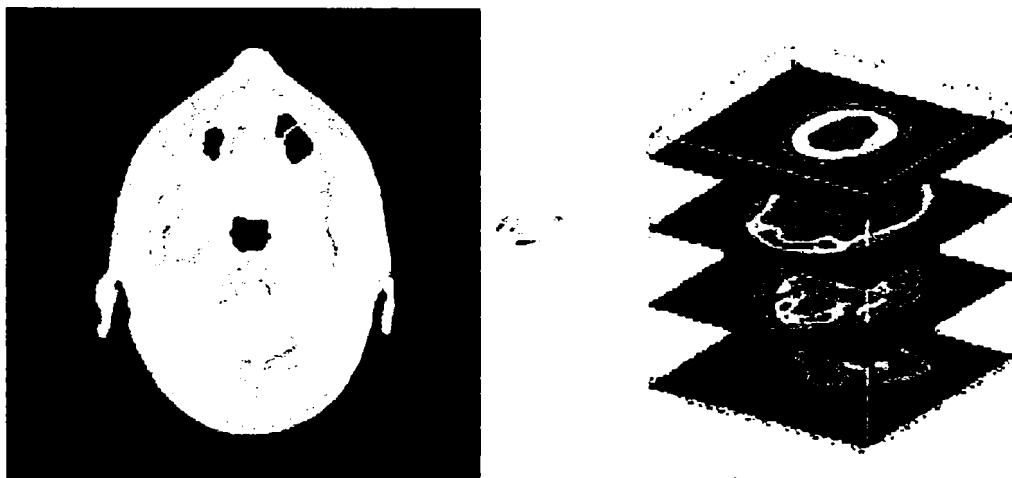


Fig.5-2 Sursa datelor de intrare o reprezintă tomogramele ce vor fi transformate într-o structură de memorare tridimensională

- segmentarea imaginilor și identificarea regiunilor de interes, dacă este cazul (figura 5-3):



Fig.5-3 Pe tomogramă se aplică o selecție a regiunii de investigat dacă aceasta nu este suficient reliefată prin scanarea inițială.

- detecția izosuprafeței prin aplicarea filtrului de detecție pe volum, și obținerea modelului (figura 5-4):



Fig.5-4 Prin aplicarea filtrului de detecție pe volum se obține o izosuprafață 3D

5.3. Citirea datelor

Datele de intrare ale procesului [213][113] sunt reprezentate de fișierele conținând tomogramele rezultate în urma analizelor specifice. Ca variabile de intrare se furnizează numărul de fișiere de citit, dimensiunea lor, spațierea acestora și prefixul acestor fișiere. Acest prefix împreună cu numărul fișierului va forma denumirea întreagă a acestuia, astfel încât să poată fi accesat. Practic, la un pacient va fi încărcat un set de fișiere.

Se utilizează următoarele apeluri de funcții:

```
// citirea volumului
Reader *v = Reader::New();
v->DataDimensions(64,64);
v->FilePrefix (".../.../.../data/img");
v->Range(1, 100);
v->Spacing(1,1,1);
```

care vor avea ca rezultat citirea fișierelor: img1, img2, img3, ...img100 fiecare având dimensiunea de 64x64 pixeli (figura 5-5).

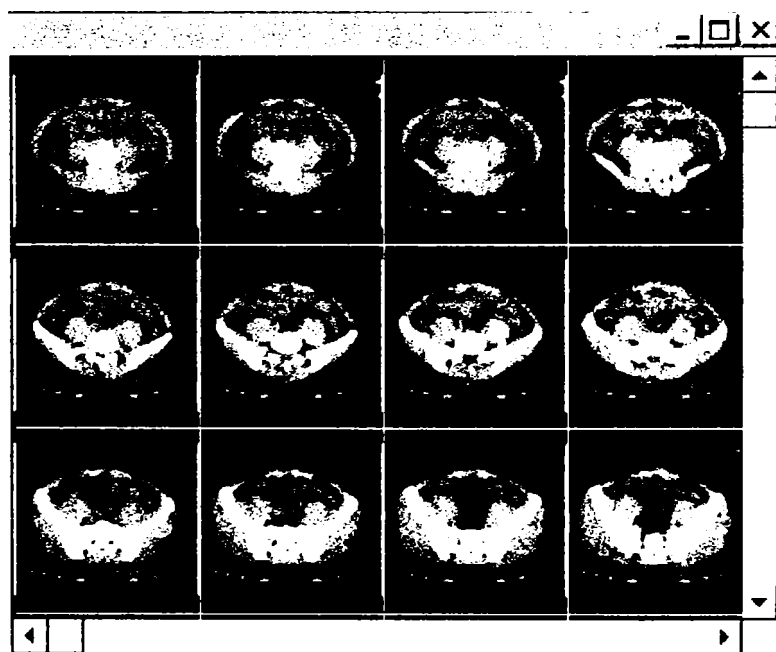


Fig.5-5 Exemplu de succesiune de imagini 64 x 64 pixeli dintr-o tomogramă abdominală, achiziționată prin tomograful Somatom Plus 4

Nucleul funcțiilor ce realizează citirea datelor este:

```
#include "Reader.h"

// Descriere:
// Construiește și initializează obiectul Reader
Reader::Reader()
{
    this->HeaderSize = 0;
    this->SwapBytes = 0;
    this->DataDimensions[0] = this->DataDimensions[1] = 0;
    this->Transform = NULL;
}

void Reader::Execute()
{
    Scalars *newScalars;
    int first, last;
    int numberSlices;
    int *dim;
    int dimensions[3];
    float Spacing[3];
    float origin[3];

    Points *output=(Points *)this->Output;

    // Validările datelor de intrare
    if (this->FilePrefix == NULL)
    {
        ErrorMacro(<< "FilePrefix is NULL");
        return;
    }
    if (this->HeaderSize < 0)
    {
        ErrorMacro(<< "HeaderSize " << this->HeaderSize << " must be >= 0");
        return;
    }
    dim = this->DataDimensions;
    if (dim[0] <= 0 || dim[1] <= 0)
    {
        ErrorMacro(<< "x, y dimensions " << dim[0] << ", " << dim[1]
            << "must be greater than 0.");
        return;
    }
    if ( (this->ImageRange[1]-this->ImageRange[0]) <= 0 )
    {
        numberSlices = 1;
        newScalars = this->ReadImage(this->ImageRange[0]);
    }
    else
    {
        first = this->ImageRange[0];
        last = this->ImageRange[1];
        numberSlices = last - first + 1;
        newScalars = this->Read(first, last);
    }

    // Descriere:
    // Citeste un volum de date
    Scalars *Reader::Read(int first, int last)
    {
        Scalars *scalars = NULL;
        unsigned short *pixels;
        unsigned short *slice;
        FILE *fp;
        int numPts;
        int fileNumber;
        int status=0;
    }
}
```



```

int numberSlices = last - first + 1;
char filename[1024];
int dimensions[3];
int bounds[6];

// calculeaza nr.de puncte per imagine
numPts = this->DataDimensions[0] * this->DataDimensions[1];
// aloca memorie
slice = new unsigned short[numPts];
scalars = Scalars::New();
scalars->Allocate(numPts * numberSlices);

pixels = scalars->Write(0, numPts*numberSlices);

// construiesc numele fisierului si citeste datele
for (fileNumber = first; fileNumber <= last; fileNumber++)
{
    // construiesc numele fisierului.
    if (this->FilePattern)
    {
        sprintf (filename, this->FilePattern, this->FilePrefix, fileNumber);
    }
    else
    {
        sprintf (filename, this->FilePattern, fileNumber);
    }
    if ( !(fp = fopen(filename,"rb")) )
    {
        ErrorMacro(<<"Can't find file: " << filename);
        return NULL;
    }

    // citeste datele
    status = ReadBit (fp, slice, this->DataDimensions[0],
                    this->DataDimensions[1], this->HeaderSize, this->SwapBytes);
    //inchide fisierul
    fclose (fp);

    if (status == 0) break;
}

delete (slice);
// verifica citirea si returneaza rezultatul
if (status == 0)
{
    scalars->Delete();
    return NULL;
}
else return scalars;
}

int Reader:: ReadBit (FILE *fp, unsigned short *pixels, int xsize,
                    int ysize, int askip, int swapBytes)
{
    unsigned short *shortPtr;
    int numShorts = xsize * ysize;

    if (skip) fseek (fp, skip, 0);

    shortPtr = pixels;
    shortPtr = shortPtr + xsize*(ysize - 1);
    for (int j=0; j<ysize; j++, shortPtr = shortPtr - xsize)
    {
        if ( ! fread(shortPtr,sizeof (unsigned short),xsize,fp) )
        {
            ErrorMacro(<<"Error reading raw pgm data!");
            return 0;
        }
    }
    return 1;
}

```

5.4 Selectarea zonelor de interes

Deși în metoda propusă în teză, filtrul de detecție a izosuprafețelor, nu necesită, în principiu, o identificare prealabilă a diferitelor suprafețe caracterizate de anumite densități ale țesuturilor, fiind o metodă suficient de puternică și flexibilă în identificarea acestora, în cazul în care se dorește totuși o izolare a unui anumit tip de țesut, acest lucru se va realiza printr-o operație de preprocesare numită *segmentare* [20][29][37][55][66][147][184][196]. Bazele teoretice și soluția propusă de autorul tezei sunt descrise mai jos.

Segmentarea reprezintă o categorie de tehnici de prelucrare a imaginilor ce permit extragerea din contextul imaginii a unor elemente de interes în scopul analizei lor ulterioare și, eventual, al clasificării lor. Extragerea conturilor este o etapă necesară pentru a realiza caracterizarea regiunilor din imagine în vederea identificării obiectelor de interes. În plus această operație asigură o reducere importantă a volumului de informație de la sute de mii sau milioane de pixeli, la mii de puncte de contur.

Sunt aplicate două strategii majore în cadrul procesului de segmentare:

a) Detecția conturului: sunt depistate zonele cu variații bruște ale proprietăților (muchii) prin a căror unire se construiește conturul obiectului ce trebuie delimitat (de exemplu, țesut osos).

b) Detecția regiunilor omogene: Sunt depistate zonele cu proprietăți uniforme, prin a căror analiză, se obține porțiunea corespunzătoare de imagine ocupată de obiectul cercetat (zona cu valori apropiate ale densității, ceea ce corespunde unui anumit țesut).

O metodă simplă presupune următorii pași succesivi:

- Se filtrează trece - jos imaginea și se elimină zgomotele care sunt asociate frecvențelor înalte. Operația este în principiu o mediere, dar poate introduce și o pierdere a detaliilor și o estompare a conturilor, care sunt și ele asociate tot frecvențelor înalte;
- Se filtrează trece - sus imaginea;
- Se aplică o detecție cu un prag sau două praguri pe imagine pentru a filtra imaginea și a izola doar obiectele de care este nevoie.

Datorită problemelor ce apar la alegerea pragurilor, adică pragurile prea mici nu furnizează rezultate satisfăcătoare iar pragurile prea mari filtrează prea puternic imaginea, în lucrare s-a optat pentru o metodă interactivă de segmentare care permite alegerea pragurilor în timp real și observarea imediată a influenței lor asupra imaginii.

5.4.1 Detecția conturului

Lucrând pe imagini numerice, filtrarea trece – sus care este asociată cu operația de derivare se realizează numai prin diferențe între nivelele de gri. Măștile asociate au și ponderi pozitive și ponderi negative. Derivarea imaginii rămâne în esență o operație de accentuare a detaliilor dintr-o imagine numerică. O modalitate de detecție a conturilor este calculul gradientului în fiecare punct al imaginii [66][20][55][82][133].

$$\vec{G} = (G_x, G_y) \quad G_x = \frac{\partial I_r(x, y)}{\partial x} \quad G_y = \frac{\partial I_r(x, y)}{\partial y} \quad (5-1)$$

Astfel se pot obține informații despre direcția conturului (gradientul este perpendicular pe contur). Compararea valorilor normei gradientului cu un prag ales va selecta punctele de pe contur (figura 5-6). Măștile asociate au o simetrie în raport cu o anumită direcție, care în acest caz este reprezentată de direcția muchiilor detectate.

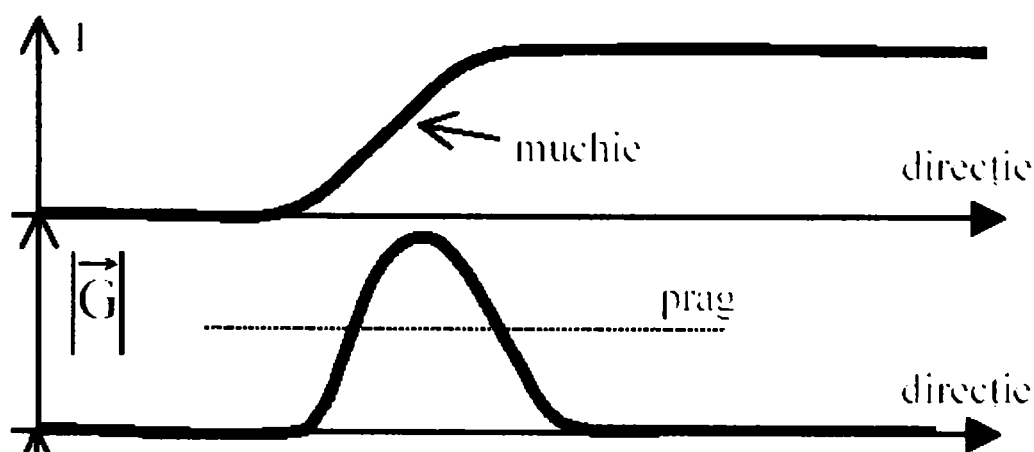


Fig.5-6 Variația intensității luminoase (nivelul de gri al pixelilor) și derivatelor corespunzătoare unei muchii

Imaginea numerică In este alcătuită din pixeli cu nivele de gri $g(i,j)$, care corespund la intensități luminoase $In(x,y)$. Pixelii sunt despărțiți de distanțe unitare și deci se poate scrie:

$$|\vec{G}| = \sqrt{[g(i,j) - g(i+1,j)]^2 + [g(i,j) - g(i,j+1)]^2} \quad (5-2)$$

sau:

$$|\vec{G}| = |g(i,j) - g(i+1,j)| + |g(i,j) - g(i,j+1)| \quad (5-3)$$

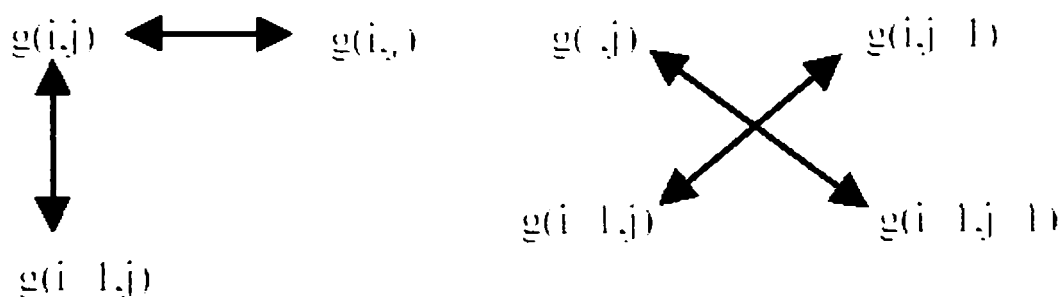


Fig.5-7 Modalități de calcul rapid al gradientului.

Se poate utiliza și aproximarea Roberts care face apel la diferențele încrucișate (figura 5-7) și este utilă pentru că detectează mai bine conturul atunci când gradientul variază puțin. În această aproximație se scrie :

$$|\vec{G}| = |g(i,j) - g(i+1,j+1)| + |g(i+1,j) - g(i,j+1)| \quad (5-4)$$

Ca alternativă și mai simplă, se pot deriva imaginile folosind ca formulă de tip gradient:

$$|\vec{G}| = \max \{ |g(i,j) - g(i+1,j)|, |g(i,j) - g(i,j+1)|, |g(i,j) - g(i+1,j+1)|, |g(i+1,j) - g(i,j+1)| \} \quad (5-5)$$

Avantajele obținute în acest caz sunt: volumul relativ redus de calcul, obținerea direcției conturului (codificat Freeman) dacă se iau în calcul și semnele diferențelor din formula (5-5), relativa independență a pragului în raport cu tipul imaginii, ceea ce evită căutarea lui dinamică și obținerea unui contur în general subțire cu poziția determinată de poziția pixelului din dreapta sus dintr-o vecinătate de patru pixeli.

Contururile dintr-o imagine se pot obține comparând $|G|$ cu un prag T :

$|G| \geq T$, punct care se află pe contur;

$|G| < T$, punct care nu se află pe contur; (5-6)

Astfel, gradientul se aplică sub formă de măști 3x3 sub diferite forme:

$$\begin{bmatrix} 0 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \text{ sau } \begin{bmatrix} 0 & 0 & 0 \\ -1 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \text{ pentru muchii verticale; (5-7)}$$

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 0 \end{bmatrix} \text{ sau } \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix} \text{ pentru muchii orizontale; (5-8)}$$

$$\begin{bmatrix} -1 & -1 & 2 \\ -1 & 2 & -1 \\ 2 & -1 & -1 \end{bmatrix} \text{ pentru muchii înclinate; (5-9)}$$

Conturul este reunirea muchiiilor ce separă un obiect de spațiul înconjurător și reprezintă forma obiectului (sau a proiecției obiectului în planul imaginii). Determinarea conturului depinde prin urmare nu numai de corectitudinea detecției muchiiilor, ci, în mod esențial, de conectarea lor într-o frontieră continuă. Urmărirea conturului este un algoritm care trasează conturul prin ordonarea succesivă a punctelor muchiiilor anterior determinate. Se pornește de la un punct cunoscut și se analizează vecinii pentru detecția

punctului următor și așa mai departe. Un instrument extrem de util pentru conectarea muchiilor într-un contur este transformarea Hough [66][84].

Metoda presupune calculul pentru fiecare punct $M_i(x_i, y_i)$ aparținând conturului obiectului, a ansamblului de parametrii a care verifică $f(x_i, y_i, a) = 0$. Aceasta revine la a evalua în spațiul parametrilor a , suprafața reprezentată de toți parametrii tuturor curbelor care trec prin $M_i(x_i, y_i)$.

Dacă $P(x, y)$ este un pixel alb dintr-o imagine cu fundal negru, ecuația unei drepte care trece prin acest punct este $y = ax + b$. Această ecuație definește în spațiul (a, b) , spațiul parametrilor, o dreaptă. Fiecărui punct de pe această dreaptă îi corespunde câte o dreaptă în spațiul (x, y) (figura 5-8). Și reciproc, fiecărui punct din spațiul (x, y) îi corespunde câte o dreaptă în spațiul (a, b) . Căutarea unei drepte în planul imagine se reduce la căutarea unui punct în planul parametrilor. Procedura poate fi prezentată astfel:

- Pentru fiecare punct din planul parametrilor (a, b) se definește câte un contor $C(a, b)$;
- Pentru fiecare punct alb $P(x, y)$ al imaginii se incrementează $C(a, b)$ corespunzător lui $D(a, b)$ care verifică ecuația $y = ax + b$;
- Dacă există într-adevăr o dreaptă $y = ax + b$ în imagine, atunci $C(a, b)$ corespunde la o valoare mare la sfârșitul procedurii. Identificarea dreptelor prezente în imagine revine la a identifica contoarele C cu valori ridicate.

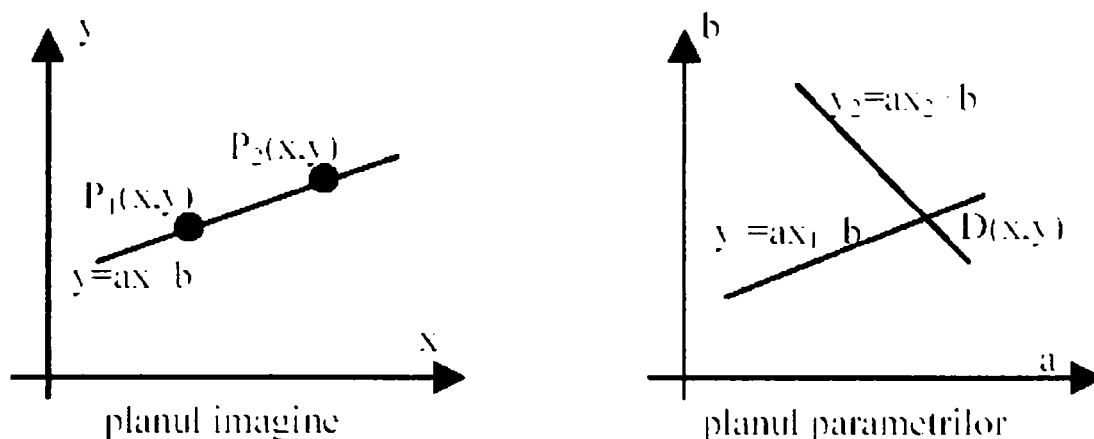


Fig.5-8 Corespondența variabile –parametrii

Nucleul funcțiilor care realizează transformarea Hough este:

```

FILE *sursa;
int x,y,xo=100,yo=100;

typedef struct lu{int alfa;          // structura unghi alfa
                 float sin;
                 float cos;
                 struct lu *urmat;}lu;
typedef struct th{int theta;        // structura punct in spatiul r,θ
                 float r;
                 struct th *urm;}th;

th *e1;
lu *e2;

unsigned char *aloca_byt (char DIM_X,char DIM_Y);//functie de alocare
int *af_bit (unsigned char *ptim,int x,int y,int xo,int yo);
int *preluare (unsigned char *ptim,int x,int y,int xo,int yo);
// preluare imagine de pe ecran intr-un pointer
int *af_oct (unsigned char *ptim,int x,int y,int xo,int yo);
// afisare imagini cu pixelii codati fiecare cu câte un octet
void adaug1(int unghi,float dist);          // adauga noduri
void adaug2(int unghi,float sinus,float cosinus);// in listele lu si th

void main(void)
{
  unsigned int *X,*Y;
  int j,k,lin,lin0,col,col0,cp,f;
  float a,s,c,d;
  int tab[10][18];
  int err,gd=DETECT,gm;
  unsigned char i,*ptim,*P;
  char nume[20];
  lu *u,*q;
  th *t,*w;
  clrscr();
  scanf("%s",&nume);
  sursa=fopen(nume,"rb");          // se deschide fisierul sursa
  X=malloc(sizeof(int));
  Y=malloc(sizeof(int));
  fread(X,1,2,sursa);
  fread(Y,1,2,sursa);
  P=malloc(sizeof(char));
  fread(P,1,1,sursa);          // se citeste dimensiunea paletei
  x=*X; y=*Y;          // se citesc dimensiunile imaginii
  free(X);free(Y);free(P); // eliberez X,Y,P
  ptim=aloca_byt(x/8,y);
  fread(ptim,1,(x*y)/8,sursa);
  fclose(sursa);          // se inchide fisierul citit
  initgraph(&gd,&gm,"c:\\tcpp\\bgi");
  if((err=graphresult())!=grOk){
    printf("%s",grapherrormsg(err));
    getch();
    exit(1);
  }
  for(i=0;i<16;i++)          // setare paleta gri-uri
    setpalette(i,i);
  af_bit(ptim,x,y,xo,yo);          // se afiseaza imaginea
  free(ptim);          // eliberare *ptim
  ptim=aloca_byt(x,y);          // alocare spatiu ptim ptr. reutilizare
  preluare(ptim,x,y,xo,yo) // preiau imaginea in *ptim
  lin0=100;col0=100;
  for(j=0;j<=9;j++)          // se initializeaza tab
    for(k=0;k<=17;k++)
      tab[j][k]=0;
  u=(lu*)malloc(sizeof(lu));          // se initializeaza lista lu
  u->urmat=NULL;
  e2=u;
  t=(th *)malloc(sizeof(th));          // se initializeaza lista th
  t->urm=NULL;
  e1=t;
  for(j=0;j<=170;j+=10){          // calculez sin si cos pentru
    a=j*3.141592654/180;          // toate unghiurile din 10° in 10°
    s=sin(a);

```

```

        c=cos(a);
        adaug2(j,s,c);
    }
for(lin=lin0;lin<=lin0+31;lin++){ // scanare imagine pentru efectuarea
    for(col=col0;col<=col0+31;col++){ // transformarii Hough
        cp=getpixel(col,lin);
        if(cp==15)
            for(q=u->urmat;q!=NULL;q=q->urmat){
                j=q->alfa;
                d=(col-115)*(q->cos)+(lin-115)*(q->sin);
                adaug1(j,d);
            }
    }
}
for(w=t->urm;w!=NULL;w=w->urm){ // sortare rezultate
    f=(w->theta)/10;
    if((w->r)>=0){
        if((w->r)<=4){ tab[4][f]++; continue;}
        if((w->r)<=8) {tab[3][f]++;continue;}
        if((w->r)<=12) {tab[2][f]++;continue;}
        if((w->r)<=16) {tab[1][f]++;continue;}
        if((w->r)<=22) {tab[0][f]++;}
    }
    else{
        if((w->r)>=-4) {tab[5][f]++;continue;}
        if((w->r)>=-8) {tab[6][f]++;continue;}
        if((w->r)>=-12) {tab[7][f]++;continue;}
        if((w->r)>=-16) {tab[8][f]++;continue;}
        if((w->r)>=-22) {tab[9][f]++;}
    }
}

void adaug2(int unghi,float sinus,float cosinus)
{
    extern lu *e2;
    lu *l;
    l=(lu *)malloc(sizeof(lu)); // alocare spatiu in memorie
    if(l==NULL){ // pentru un nou nod in lista lu
        printf("Memorie insuficienta! \n");
        getch();
        exit(1);
    }
    l->alfa=unghi; // definire elemente nod l
    l->sin=sinus; // in lista lu
    l->cos=cosinus;
    l->urmat=NULL; // nodul urmator in lista
    e2->urmat=l;
    e2=l;
}

void adaug1(int unghi,float dist)
{
    extern th *e1;
    th *l;
    l=(th *)malloc(sizeof(th)); // alocare spatiu in memorie
    if(l==NULL){ // pentru nod in lista th
        closegraph();printf("Memorie insuficienta! \n");
        getch();
        exit(1);
    }
    l->theta=unghi; // definire elemente nod l
    l->r=dist; // in lista th
    l->urm=NULL;
    e1->urm=l; // nodul urmator
    e1=l;
}

```


Rezultate ale detecției conturului regiunii de interes sunt prezentate în figura 5-9:

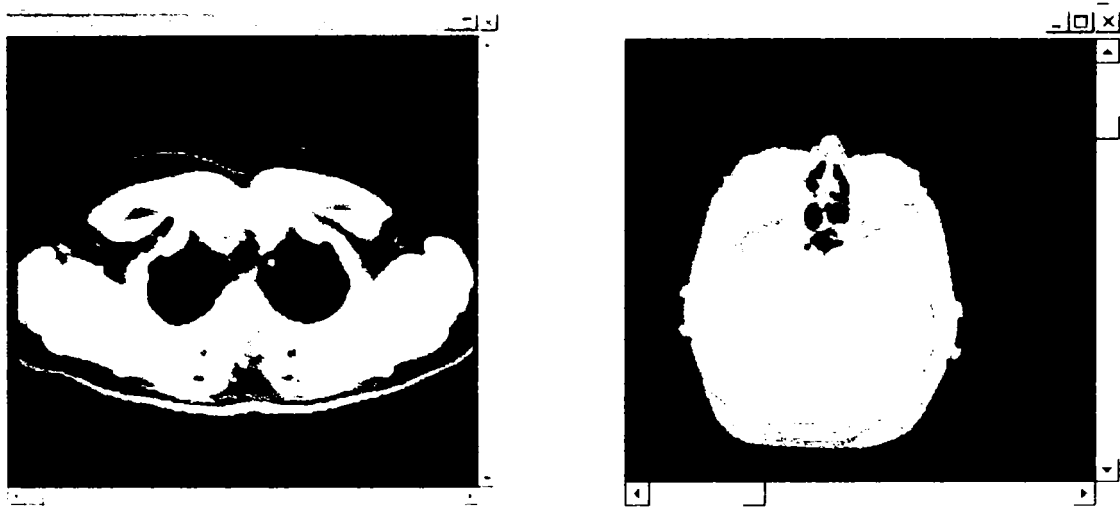


Fig.5-9 Detecții de contur aplicate pe tomograme pentru identificarea diferitelor zone de interes

Prima imagine este o tomogramă în care sunt identificate coloana vertebrală și coastele. A doua imagine prezintă o tomogramă cu țesut osos cranian identificat. Detecția conturilor este realizată în întreaga stivă de imagini (figura 5-10), astfel punându-se la dispoziția pasului următor regiunea dorită a fi reconstruită.

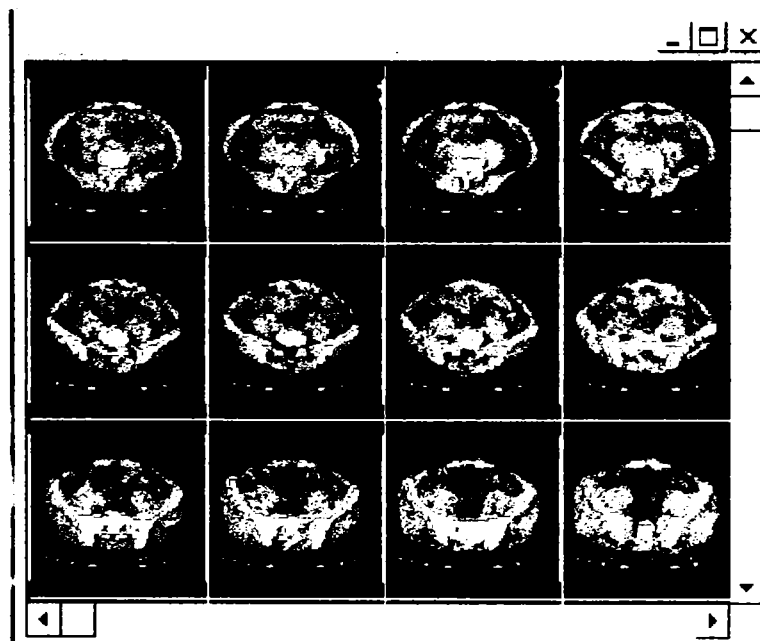


Fig.5-10 Detecția zonelor de interes se realizează în întreaga stivă de imagini

5.4.2 Detecția regiunilor omogene

Prin detecția regiunilor omogene se înțelege gruparea pixelilor dintr-o imagine în categorii distincte în funcție de proprietățile lor (de exemplu nivelul de gri) [147][151][173][196]. Această selecție duce la apariția unor porțiuni caracterizate printr-o relativă uniformitate. Zonele astfel determinate permit în ultimă instanță o separare a obiectului ce trebuie analizat de fondul imaginii și de eventuale alte obiecte aflate în cadru. Metodele care se bazează pe această procedură sunt mai flexibile decât cele de conturare și mult mai ușor de implementat asigurând și rezultate foarte bune.

Tehnicile din categoria segmentării histogramelor se dovedesc foarte utile ori de câte ori există o separare relativ clară a nivelelor de gri între obiectul analizat și fundal (histograme bimodale). În multe situații, pentru a ajunge la acest caz favorabil, sunt necesare preprocesări ale histogramei imaginii care permit aducerea acesteia la o formă avantajoasă.

Segmentarea cu prag unic reprezintă o metodă de bază fiind în același timp o tehnică simplă, rapidă și eficientă în multe cazuri. Etapele sunt:

- se examinează minimele și maximele locale;
- se selectează pragul t între cele două maxime locale dominante;
- se alocă pixelilor cu nivel de gri $n < t$ valoarea 0 (negru) și celor cu $n > t$ valoarea 1 (alb). Rezultă astfel imaginea binarizată.

În cazul în care histograma imaginii originale sau preprocesate nu are structura ideală (bimodală, cu două maxime locale dominante și un minim pronunțat între ele), este necesară aplicarea unei strategii diferențiate funcție de situația practică existentă.

În teză a fost implementată și utilizată o metodă de segmentare interactivă cu două praguri, pentru că oferă mai multă flexibilitate, cazurile de identificare a structurilor anatomice fiind mai complexe. S-a optat pentru varianta interactivă și cu propagare în întreaga stivă de imagini pentru a se putea selecta atât țesuturi cu densități mici cât și țesuturi cu densități mari (practic se poate selecta orice interval). Această selecție se realizează o singură dată pe o singură imagine după care ea se propagă pe întreaga stivă de imagini pentru a reduce efortul utilizatorului.

Segmentarea cu prag multiplu se face în funcție de valorile alese a două praguri $t_1 \neq t_2$. Astfel :

- pentru pixelii cu nivel de gri mai mic decât t_1 se adoptă valoarea 0;

- pentru pixelii cu nivel de gri mai mare decât t_2 se adoptă valoarea 0;
- pentru pixelii intermediari se adoptă valoarea 1.

În acest mod se izolează doar regiunea cu caracteristica dată; o abordare asemănătoare, mai elaborată este:

- pentru pixelii cu nivel de gri mai mic decât t_1 se adoptă valoarea 0;
- pentru pixelii cu nivel de gri mai mare decât t_2 se adoptă valoarea 1;
- pentru pixelii intermediari se adoptă valoarea 0 sau 1 funcție de valoarea majoritară în vecinătate (se realizează prin scanarea pixel cu pixel a imaginii în mai multe treceri succesive). Vecinătatea în care se face evaluarea se mărește pâna la eliminarea tuturor pixelilor de valoare intermediară.

Implementarea software a unei astfel de funcții este:

```
int *segm(unsigned char far *ptim,int *ldr,int *lst,int lgx,int lgy){
    int ALB,NEGRU,GRI;
    int ln,cl,s,sumax,c;
    unsigned char far *ini;
    ALB=240;
    NEGRU=0;
    GRI=120;
    ini=ptim;
    for(ln=0;ln<lgy;ln++){
        for(cl=0;cl<lgx;cl++){
            // testare pixel cu pixel
            // fata de pragul stanga
            if(*ptim<=*lst)
                *ptim=NEGRU;
            else
                if(*ptim>=*ldr) // si fata de pragul dreapta
                    *ptim=ALB;
                else
                    *ptim=GRI;
            ptim++;}
        getch();
        do{ // ciclul eliminare pixeli GRI
            c=0;
            ptim=ini;
            for(ln=0;ln<lgy;ln++){
                for(cl=0;cl<lgx;cl++){
                    if(*ptim==GRI){
                        s=0;
                        sumax=0;
                        if(*(ptim+1)==GRI)
                            sumax+=1;
                        s+*(ptim+1);
                        if(*(ptim-1)==GRI)
                            sumax+=1;
                        s+*(ptim-1);
                        if(*(ptim-256)==GRI)
                            sumax+=1;
                        s+*(ptim-256);
                        if(*(ptim-255)==GRI)
                            sumax+=1;
                        s+*(ptim-255);
                        if(*(ptim-257)==GRI)
                            sumax+=1;
                        s+*(ptim-257);
                        if(*(ptim+256)==GRI)
                            sumax+=1;
                        s+*(ptim+256);
```

```

    if (*(ptim+255) = =GRI)
        sumax+=1;
    s+ (*(ptim+255);
    if (*(ptim+257) = =GRI)
        sumax+=1;
    s+ (*(ptim+257);
    if (sumax<4){
        if (s<GRI*4)
            *ptimd=NEGRU;
        else
            *ptimd=ALB;
        c=1;}
        ptim++;}
    afis(ptim,SIZEX,SIZEY,180,60);}
while (c= =1);
return NULL;}

```

Astfel, în imaginile din figura 5-11 se poate observa identificarea de țesut moale și osos cu pragul minim $t1=100$ și cel maxim $t2=255$ (zona albă), din tomografia zonei toracelui. În schimb la aplicarea unui prag minim $t1= 170$, cel maxim rămânând neschimbat s-a obținut reliefarea țesuturilor osoase (coaste și vertebre).

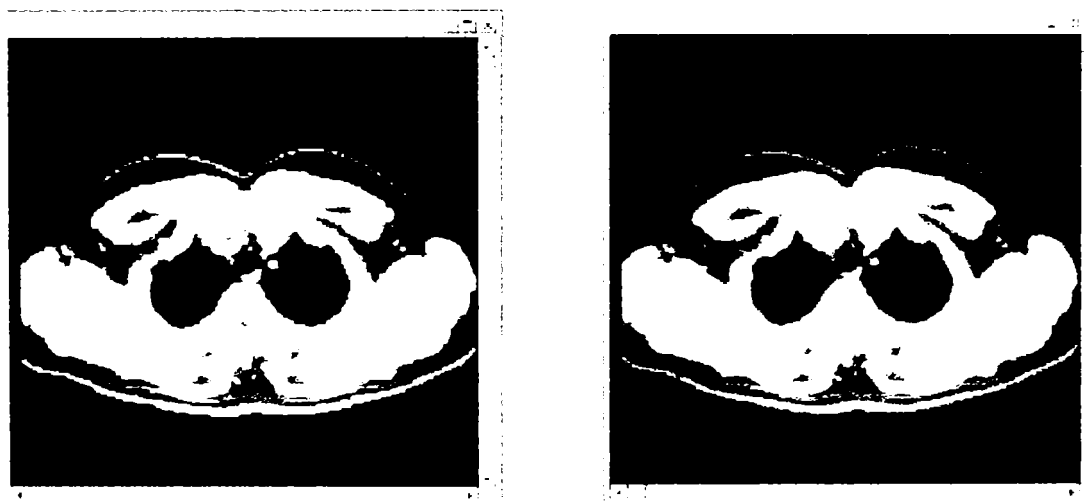


Fig.5-11 Variația pragurilor determină selecția diferitelor regiuni de interes.

În teză, metoda de bază pentru prelucrări este cea a segmentării histogramei cu două praguri. Deși filtrul de detecție a izosuprafețelor nu necesită în principiu segmentare, fiind suficient de puternic și flexibil în identificarea acestora, în cazul în care este nevoie de segmentare și se dorește totuși o izolare a unui anumit tip de țesut, se vor utiliza cele două valori ale pragurilor, corespunzătoare intervalului de densitate ce caracterizează țesutul.

5.5 Filtrul volumetric de detecție a izosuprafețelor 3D

Algoritmul este utilizat pentru a construi contururi din seturi de date volumetrică [92][100][102] prin aplicarea unui mecanism de detecție a *izosuprafețelor* (suprafețele care au asociate aceleași valori ale densității) specificate printr-o valoare de prag a densității. Aceste contururi determină limitarea regiunilor diferite de date între ele. De exemplu, având la dispoziție un set de date conținând densități provenind de la tomografierea computerizată a capului unui pacient, se pot genera imagini ce permit vizualizarea unei porțiuni a pielii sau imagini ce permit vizualizarea unei porțiuni a oaselor (figura 5-12).

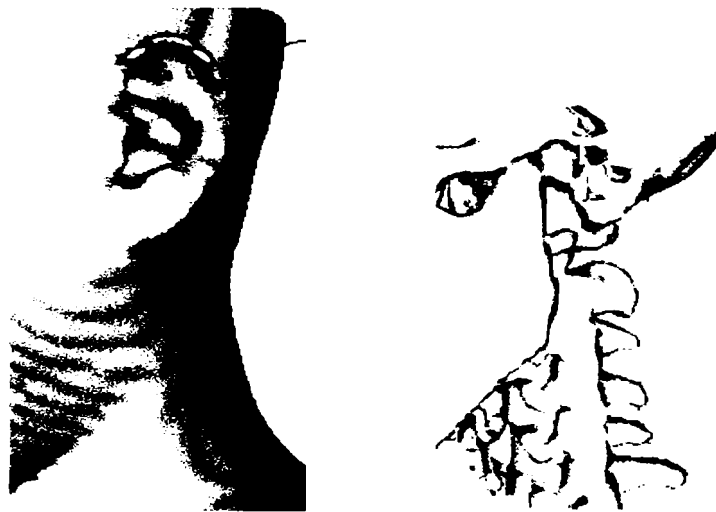


Fig.5-12 Suprafețe diferite extrase din același set de date: țesut moale (piele), și țesut osos

Ideea de bază este aceea de a diviza volumul format de stiva de imagini într-o rețea de celule unitare tridimensionale de formă cubică (voxeli). Suprafața care se dorește a fi extrasă se află în interiorul acestui volum iar filtrul va căuta, din mulțimea de puncte din interiorul volumului, doar pe acelea care se află pe suprafață [21].

Un contur 3D poate trece printr-o celulă doar într-un număr finit de moduri. Se poate deci construi un tabel care să cuprindă toate stările topologice posibile ale celulei elementare (modurile posibile în care suprafața poate intersecta muchii unei celule). Numărul stărilor topologice depinde de numărul vârfurilor celulelor și de faptul că acestea pot fi în interiorul sau în exteriorul conturului. Un vârf este considerat interior conturului 3D (izosuprafeței) dacă valoarea scalară asociată lui este mai mare decât

valoarea scalară asociată izosuprafeței. Vârfurile cu valoare scalară mai mică decât valoarea asociată suprafeței căutate sunt considerate exterioare.

Deoarece într-o celulă cubică există 8 noduri, se pot realiza 256 combinații topologice diferite. Cu ajutorul operațiilor de oglindire și simetrie, numărul lor poate fi redus la 15. Odată determinate muchiile intersectate pentru fiecare celulă, se calculează exact punctul de intersecție. Mulțimea punctelor rezultate se află pe suprafața căutată și o definesc, ceea ce înseamnă că ele pot fi *triangulate* și apoi, afișate 3D prin *rendering*.

Diagrama din figura 5-13 descrie mecanismul filtrului volumetric de detecție a izosuprafețelor utilizat în teză.

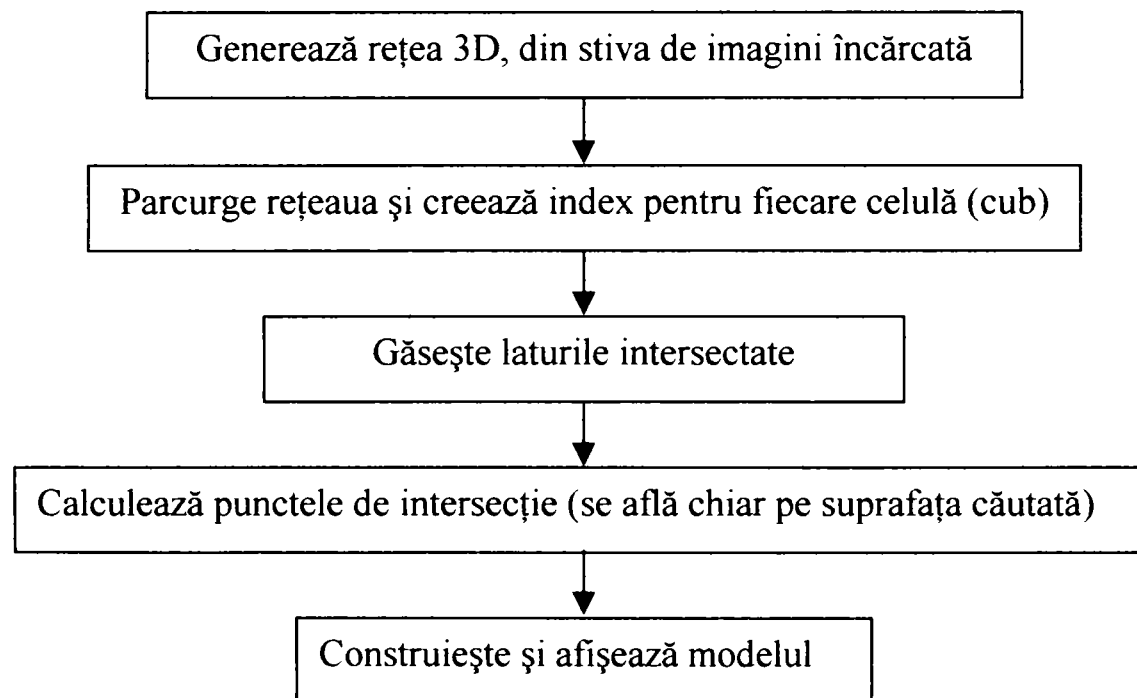


Fig.5-13 Etapele aplicării filtrului de volum

5.5.1 Divizarea volumului

În faza de citire, stiva de imagini este memorată într-o structură de tip *puncte structurate* [21][113][213], valorile pixelilor fiind organizate sub forma unei matrice tridimensionale (figura 5-14).

În prima fază a algoritmului se generează o rețea tridimensională de cuburi și fiecare vârf al fiecărui cub este asociat cu o valoare a densității din setul de date astfel memorat.

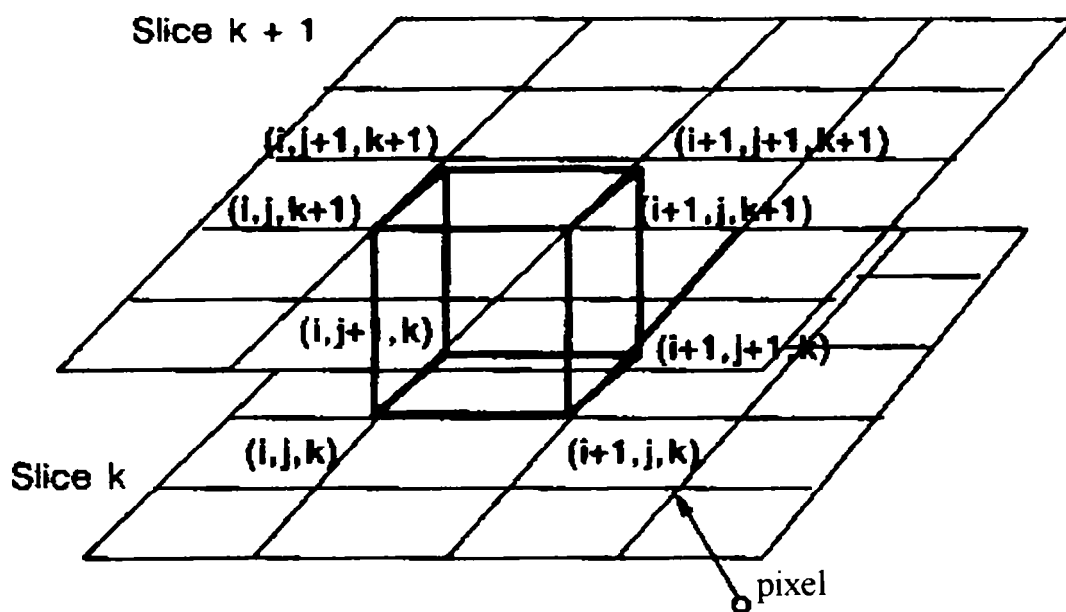


Fig.5-14 Unitatea de volum a rețelei tridimensionale

Prin examinarea, pe rând, a fiecărui cub (figura 5-15), algoritmul determină în ce mod suprafața care se dorește a fi găsită și modelată intersectează muchiile cubului. Acest lucru este necesar pentru a aproxima porțiunea de suprafață pe care se află cubul prin seturi de triunghiuri operație numită *triangulare*.

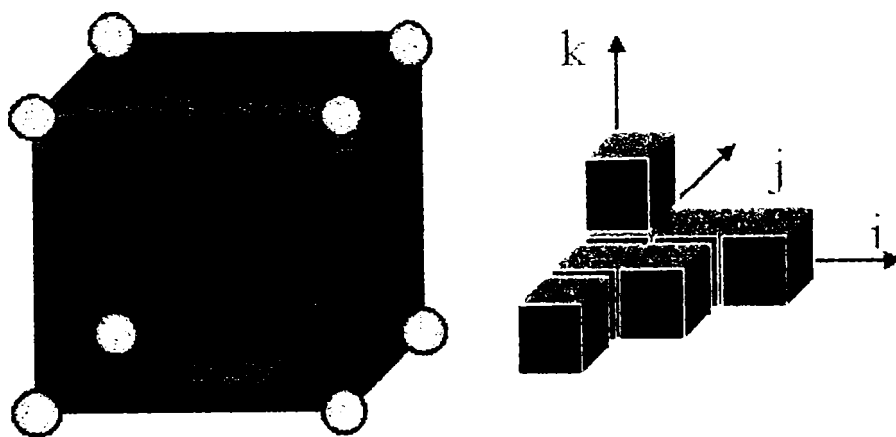


Fig. 5-15 Divizarea volumului.

Prin parcurgerea întregii rețele tridimensionale (figura 5-15), fiecare porțiune din suprafață va fi descompusă și aproximată prin triunghiuri, totalul acestora furnizând o aproximare destul de bună a suprafeței care se dorește a fi extrasă din setul de date inițial. Suprafața astfel determinată ce va fi vizualizată, poate fi de exemplu o anumită porțiune a unui os, o porțiune a țesutului ce acoperă osul sau o porțiune din piele (figura 5-16).



Fig.5-16 Zona cu țesut epitelial și zona osoasă

Codul sursă al funcției ce realizează traversarea volumului [113] este:

```
// Traversarea volumului
sliceSize = dims[0] * dims[1];
for ( k=0; k < (dims[2]-1); k++)
{
    kOffset = k*sliceSize;
    pts[0][2] = origin[2] + k*Spacing[2];
    zp = origin[2] + (k+1)*Spacing[2];
    for ( j=0; j < (dims[1]-1); j++)
    {
        jOffset = j*dims[0];
        pts[0][1] = origin[1] + j*Spacing[1];
        yp = origin[1] + (j+1)*Spacing[1];
        for ( i=0; i < (dims[0]-1); i++)
        {
            //incarca valorile scalare
            idx = i + jOffset + kOffset;
            s[0] = scalars[idx];
            s[1] = scalars[idx+1];
            s[2] = scalars[idx+1 + dims[0]];
            s[3] = scalars[idx + dims[0]];
            s[4] = scalars[idx + sliceSize];
            s[5] = scalars[idx+1 + sliceSize];
            s[6] = scalars[idx+1 + dims[0] + sliceSize];
            s[7] = scalars[idx + dims[0] + sliceSize];

            // test ca sa determin daca e posibil conturul
            if ( (s[0] < min && s[1] < min && s[2] < min && s[3] < min &&
            s[4] < min && s[5] < min && s[6] < min && s[7] < min) ||
            (s[0] > max && s[1] > max && s[2] > max && s[3] > max &&
            s[4] > max && s[5] > max && s[6] > max && s[7] > max) )
            {
                continue; // nu e posibil nici un contur
            }
            // creaza punctele voxelului
            pts[0][0] = origin[0] + i*Spacing[0];
            xp = origin[0] + (i+1)*Spacing[0];

            pts[1][0] = xp;
            pts[1][1] = pts[0][1];
        }
    }
}
```



```

pts[1][2] = pts[0][2];

pts[2][0] = xp;
pts[2][1] = yp;
pts[2][2] = pts[0][2];

pts[3][0] = pts[0][0];
pts[3][1] = yp;
pts[3][2] = pts[0][2];

pts[4][0] = pts[0][0];
pts[4][1] = pts[0][1];
pts[4][2] = zp;

pts[5][0] = xp;
pts[5][1] = pts[0][1];
pts[5][2] = zp;

pts[6][0] = xp;
pts[6][1] = yp;
pts[6][2] = zp;

pts[7][0] = pts[0][0];
pts[7][1] = yp;
pts[7][2] = zp;
// | urmeaza construirea indexului
// | determinarea laturilor intersectate si calculul punctelor de intersectie
    } //for i
  } //for j
} //for k
}

```

5.5.2 Determinarea intersecțiilor dintre suprafață și voxelul curent

Pentru a determina în ce mod intersectează suprafața cubul curent, se iau ca referințe vârfurile cubului care pot avea poziții diferite relativ la suprafața care intersectează sau nu cubul. În acest sens se asignează fiecărui vârf al cubului, în care valoarea densității este mai mare decât valoarea densității suprafeței ce se dorește a fi construită, o constantă aleasă convențional. Acestea sunt vârfurile din interiorul suprafeței. Vârfurile cubului cu valori mai mici decât valoarea densității suprafeței sunt asociate cu o altă constantă și sunt situate în afara suprafeței (figura 5-17).

Suprafața intersectează acele muchii ale cubului formate din vârfuri cu asignări diferite, adică acele muchii care au un vârf situat în afara suprafeței și un vârf situat în interiorul suprafeței. În acest mod se determină topologia suprafeței în interiorul cubului și de aici se pot determina punctele de intersecție muchie-suprafață.

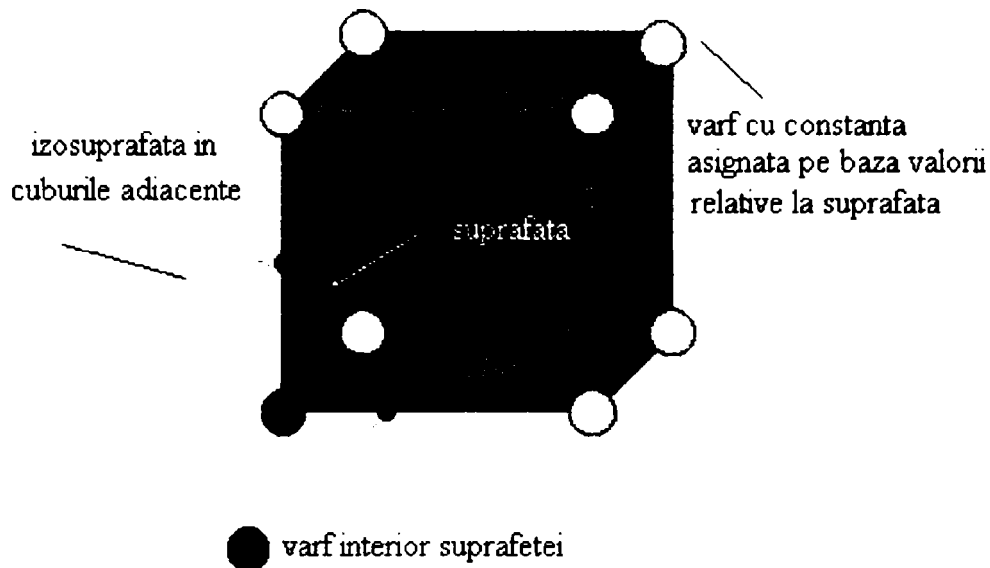


Fig. 5-17 Laturi ale cubului sunt intersectate de suprafață

Deoarece fiecare cub are 8 vârfuri și fiecare vârf poate avea două stări (în afara suprafeței sau în interiorul suprafeței), rezultă că în total există 256 de posibilități în care suprafața poate intersecta cubul. Prin considerarea tuturor acestor cazuri, și etichetarea vârfurilor cubului se creează o tabelă de consultare ce conține pentru fiecare caz intersecțiile dintre muchii și suprafață.

Pentru înlăturarea acestui neajuns [21], s-a ținut seama de cele două tipuri de simetrii ce apar în cele 256 de cazuri. În acest mod, numărul topologiilor distincte posibile și care vor fi luate în calcul se reduce de la 256 la 14.

În primul rând, topologia suprafeței triangulate rămâne neschimbată dacă relația dintre valoarea suprafeței și vârfurile cubului este inversată (figura 5-18). Cazurile complementare, în care punctele cu valori mai mari sau egale cu ale suprafeței sunt interschimbate cu punctele cu valori mai mici, sunt echivalente. În acest fel numărul cazurilor este deja redus la jumătate, adică la 128.

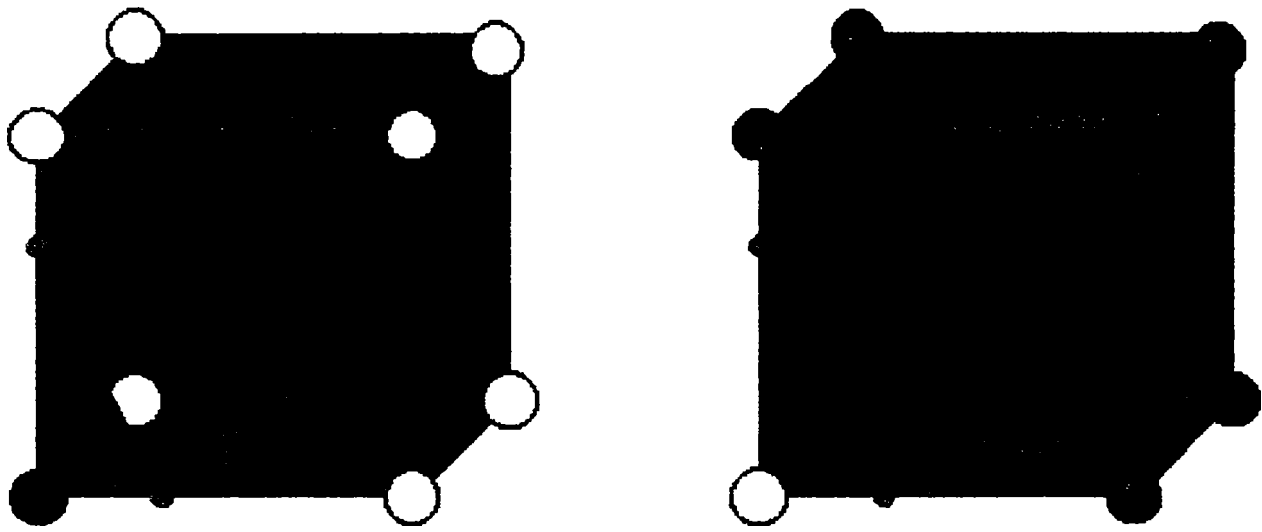


Fig. 5-18 Exemplu de simetrie prin inversiunea valorilor

Utilizând al doilea tip de simetrie, de rotație, pe cele trei direcții, problema se reduce la 16 cazuri (figura 5-19).

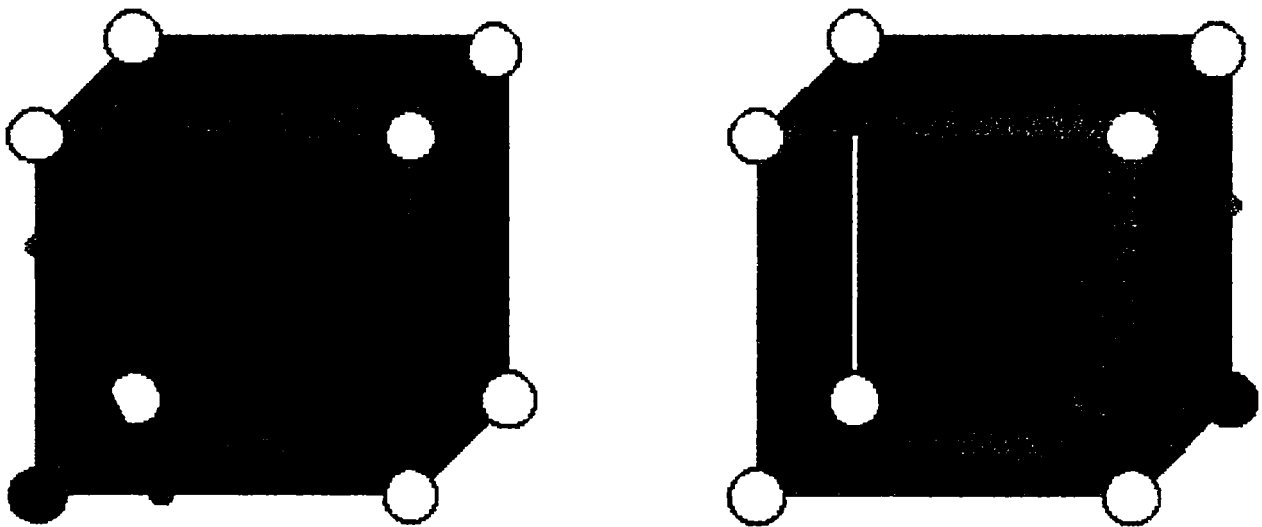


Fig.5-19 Exemplu de simetrie de rotație.

Dintre acestea există două cazuri particulare: când toate vârfurile cubului sunt în afara suprafeței și când toate vârfurile cubului sunt în interiorul suprafeței. Excluzând și aceste două cazuri particulare rămân practic 14 cazuri complet distincte care reprezintă modalități diferite în care suprafața poate intersecta această unitate tridimensională de volum. Deci vor fi 14 topologii diferite, conținând triunghiuri care vor aproxima suprafața (figura 5-20).

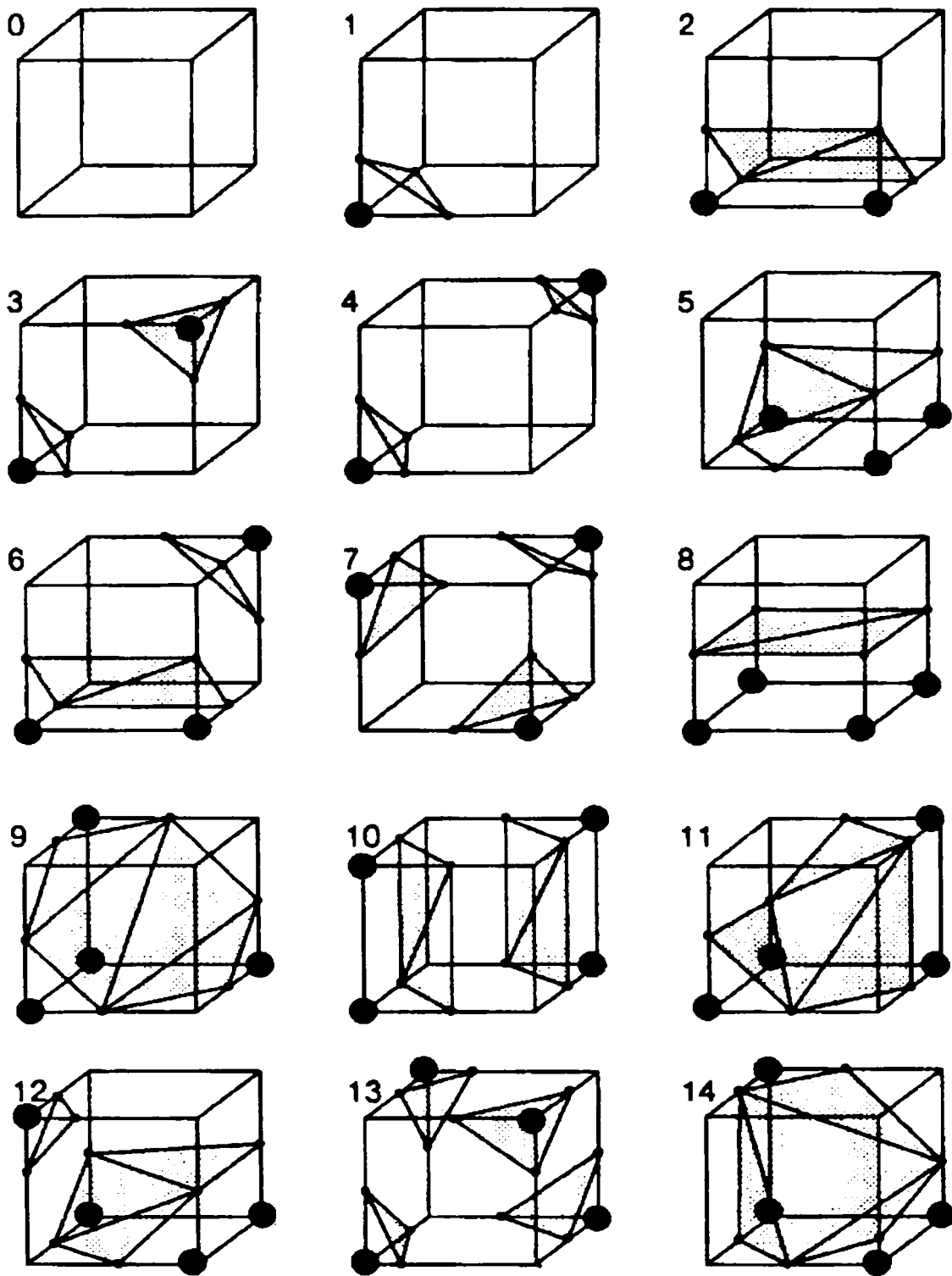


Fig. 5-20 Topologiile distincte posibile.

Cazul particular 0, este și el reprezentat în figura 5-21 și apare atunci când toate vârfurile au valori peste (sau sub) valoarea densității suprafeței de extras.

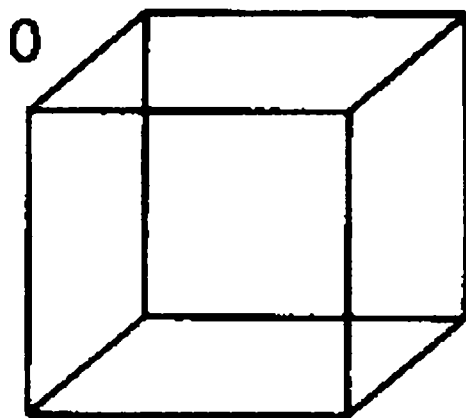


Fig.5-21 Cazul particular 0

Cazul nr.1, apare dacă suprafața separă un vârf al cubul de celelalte șapte, rezultând astfel un triunghi definit de intersecțiile suprafeței cu trei muchii adiacente (figura 5-22).

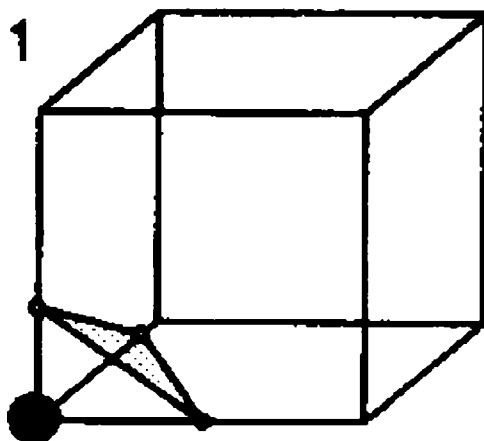


Fig.5-22 Topologia nr.1

Pentru o topologie cu 2 vârfuri de aceeași parte a suprafeței, există 3 configurații unice, date de poziția relativă a celor două colțuri: pe aceeași latură, pe aceeași fațetă, sau diagonal opuse (figura 5-23).

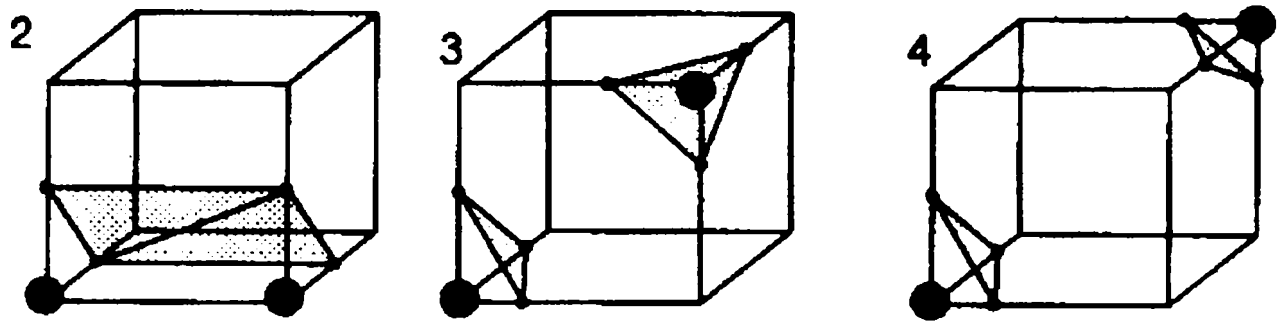


Fig.5-23 Topologiile 2,3 și 4: pentru 2 vârfuri în interiorul suprafeței există 3 configurații unice.

Pentru o topologie cu 3 vârfuri de aceeași parte a suprafeței, există tot trei configurații distincte, depinzând de existența a 0, 1, sau 2 laturi comune pentru cele trei vârfuri (figura 5-24).

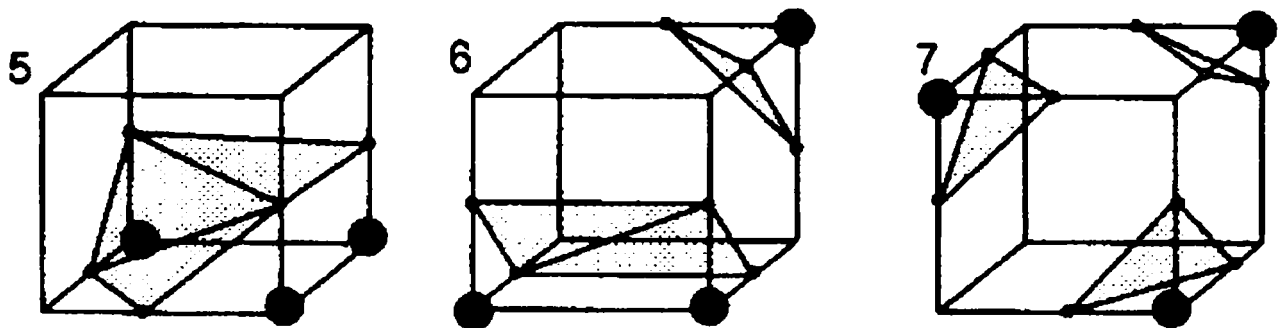


Fig.5-24 Topologiile 5,6 și 7: pentru 3 vârfuri în interiorul suprafeței există deasemenea 3 configurații distincte.

Pentru o topologie cu 4 vârfuri ale cubului de aceeași parte a suprafeței, există 7 configurații unice, date de existența a 0, 2, 3, sau 4 laturi comune (figura 5-25).

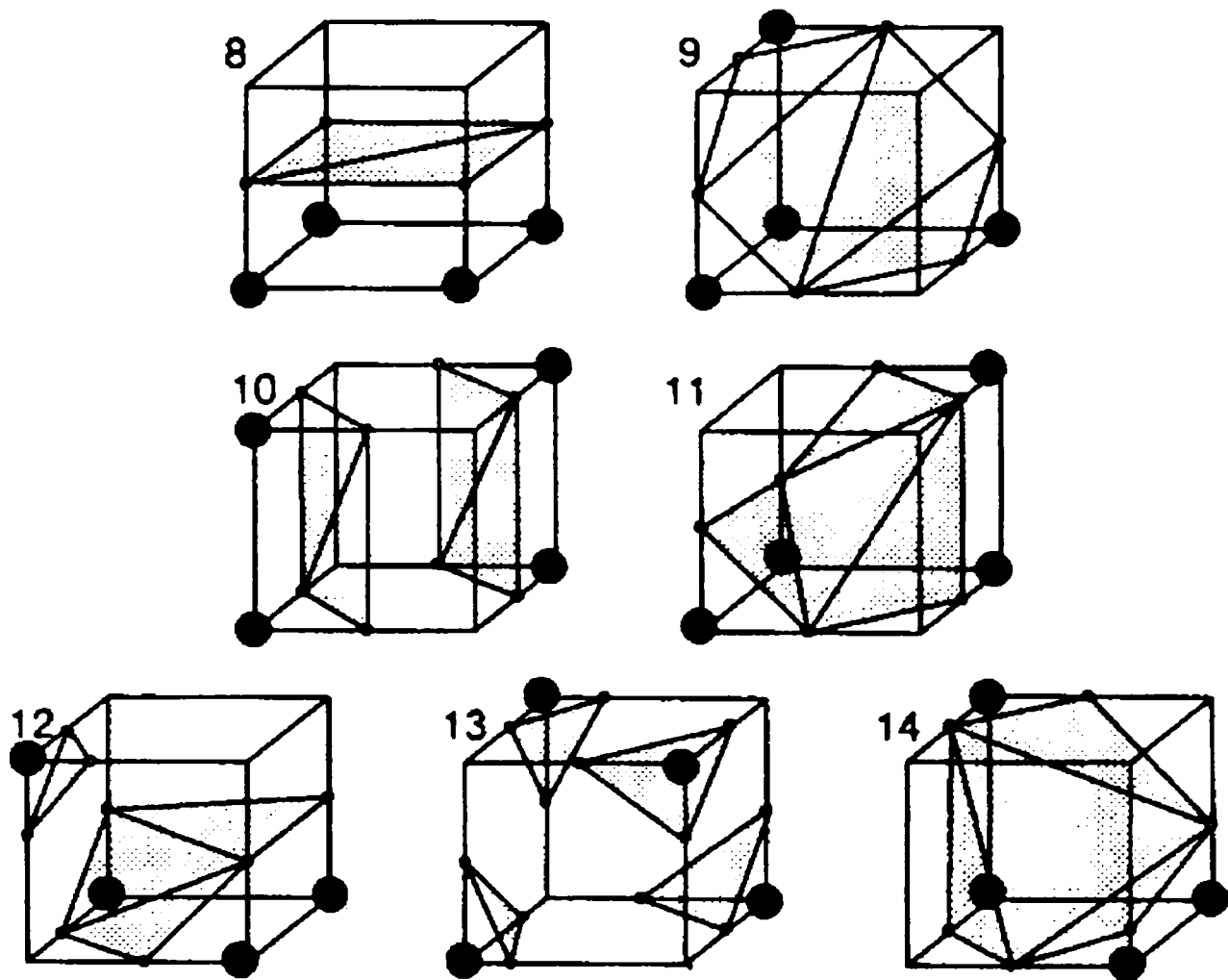


Fig.5-25 Configurațiile distincte cu 4 puncte
în interiorul suprafeței.

Permutările acestor 14 configurații de bază utilizând simetria complementară și simetria de rotație pe cele trei direcții, produc cele 256 de cazuri posibile.

Pe baza stării vârfurilor, adică a constantei asignate, se creează un index pentru fiecare caz. Utilizând numerotarea vârfurilor, indexul pe opt biți conține, câte un bit pentru fiecare vârf, care reflectă starea vârfului (figura 5-26).

Acest index servește ca pointer într-o tabelă ce furnizează toate laturile pe care le intersectează izosuprafața pentru o configurație dată, punctele de intersecție servind ca vârfuri ale triunghiurilor de aproximare.

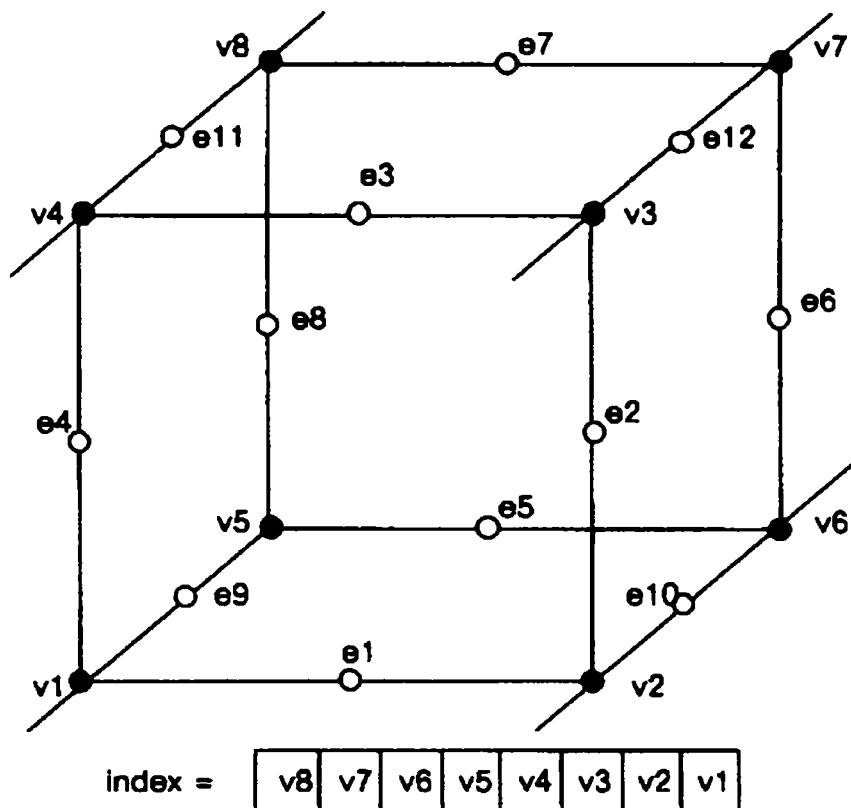


Fig.5-26 Indexul format pe baza constantelor asignate vârfurilor

Implementarea software a acestei tabele ("look-up table"), este realizată prin intermediul unei structuri de memorie TRIANGLE_CASES [113] ce conține un șir de variabile întregi (EDGE_LIST). De fapt, tabela este un șir de 256 de astfel de structuri, câte o structură pentru fiecare caz în parte. Șirul de variabile de tip EDGE_LIST va conține laturile intersectate pentru cazul pe care îl reprezintă. Din această tabelă se vor determina pentru fiecare voxel, (pe baza unui algoritm) laturile intersectate de suprafața ce se dorește a fi aproximată.

Modul de implementare software a tablei de consultare este:

```
// tabela de cazuri pentru generarea izosuprafetelor
//
typedef int EDGE_LIST;
typedef struct {
    EDGE_LIST edges[16];
} TRIANGLE_CASES;

//
// Laturile de intersectie. Trei laturi consecutive formeaza un triunghi.
// Comentariul de la sfarsitul fiecarui sir indica
// numarul cazului (0->255) si numarul cazului de baza (0->15).
static TRIANGLE_CASES triCases[] = {
{{-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1}}, /* 0 0 */
{{ 0, 3, 8, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1}}, /* 1 1 */
{{ 0, 9, 1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1}}, /* 2 1 */
{{ 1, 3, 8, 9, 1, 8, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1}}, /* 3 2 */
{{ 1, 11, 2, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1}}, /* 4 1 */

```



```

{{ 8. 7. 4. 3. 5. 10. 3. 1. 5. 5. 6. 10. -1. -1. -1. -1}}. /* 92 12 */
{{ 5. 10. 1. 5. 6. 10. 1. 10. 0. 7. 4. 10. 0. 10. 4. -1}}. /* 93 7 */
{{ 0. 9. 5. 0. 5. 6. 0. 6. 3. 10. 3. 6. 8. 7. 4. -1}}. /* 94 6 */
{{ 6. 9. 5. 6. 10. 9. 4. 9. 7. 7. 9. 10. -1. -1. -1. -1}}. /* 95 3 */
{{11. 9. 4. 6. 11. 4. -1. -1. -1. -1. -1. -1. -1. -1. -1}}. /* 96 2 */
{{ 4. 6. 11. 4. 11. 9. 0. 3. 8. -1. -1. -1. -1. -1. -1. -1. -1}}. /* 97 7 */
{{11. 1. 0. 11. 0. 6. 6. 0. 4. -1. -1. -1. -1. -1. -1. -1. -1}}. /* 98 5 */
{{ 8. 1. 3. 8. 6. 1. 8. 4. 6. 6. 11. 1. -1. -1. -1. -1. -1}}. /* 99 14 */
{{ 1. 9. 4. 1. 4. 2. 2. 4. 6. -1. -1. -1. -1. -1. -1. -1. -1}}. /* 100 5 */
{{ 3. 8. 0. 1. 9. 2. 2. 9. 4. 2. 4. 6. -1. -1. -1. -1. -1}}. /* 101 12 */
{{ 0. 4. 2. 4. 6. 2. -1. -1. -1. -1. -1. -1. -1. -1. -1}}. /* 102 8 */
{{ 8. 2. 3. 8. 4. 2. 4. 6. 2. -1. -1. -1. -1. -1. -1. -1. -1}}. /* 103 5 */
{{11. 9. 4. 11. 4. 6. 10. 3. 2. -1. -1. -1. -1. -1. -1. -1. -1}}. /* 104 7 */
{{ 0. 2. 8. 2. 10. 8. 4. 11. 9. 4. 6. 11. -1. -1. -1. -1. -1}}. /* 105 10 */
{{ 3. 2. 10. 0. 6. 1. 0. 4. 6. 6. 11. 1. -1. -1. -1. -1. -1}}. /* 106 12 */
{{ 6. 1. 4. 6. 11. 1. 4. 1. 8. 2. 10. 1. 8. 1. 10. -1}}. /* 107 7 */
{{ 9. 4. 6. 9. 6. 3. 9. 3. 1. 10. 3. 6. -1. -1. -1. -1. -1}}. /* 108 11 */
{{ 8. 1. 10. 8. 0. 1. 10. 1. 6. 9. 4. 1. 6. 1. 4. -1}}. /* 109 7 */
{{ 3. 6. 10. 3. 0. 6. 0. 4. 6. -1. -1. -1. -1. -1. -1. -1. -1}}. /* 110 5 */
{{ 6. 8. 4. 10. 8. 6. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1}}. /* 111 2 */
{{ 7. 6. 11. 7. 11. 8. 8. 11. 9. -1. -1. -1. -1. -1. -1. -1. -1}}. /* 112 5 */
{{ 0. 3. 7. 0. 7. 11. 0. 11. 9. 6. 11. 7. -1. -1. -1. -1. -1}}. /* 113 11 */
{{11. 7. 6. 11. 7. 11. 1. 8. 7. 1. 0. 8. -1. -1. -1. -1. -1}}. /* 114 9 */
{{11. 7. 6. 11. 1. 7. 1. 3. 7. -1. -1. -1. -1. -1. -1. -1. -1}}. /* 115 5 */
{{ 1. 6. 2. 1. 8. 6. 1. 9. 8. 8. 7. 6. -1. -1. -1. -1. -1}}. /* 116 14 */
{{ 2. 9. 6. 2. 1. 9. 6. 9. 7. 0. 3. 9. 7. 9. 3. -1}}. /* 117 7 */
{{ 7. 0. 8. 7. 6. 0. 6. 2. 0. -1. -1. -1. -1. -1. -1. -1. -1}}. /* 118 5 */
{{ 7. 2. 3. 6. 2. 7. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1}}. /* 119 2 */
{{ 2. 10. 3. 11. 8. 6. 11. 9. 8. 8. 7. 6. -1. -1. -1. -1. -1}}. /* 120 12 */
{{ 2. 7. 0. 2. 10. 7. 0. 7. 9. 6. 11. 7. 9. 7. 11. -1}}. /* 121 7 */
{{ 1. 0. 8. 1. 8. 7. 1. 7. 11. 6. 11. 7. 2. 10. 3. -1}}. /* 122 6 */
{{10. 1. 2. 10. 7. 1. 11. 1. 6. 6. 1. 7. -1. -1. -1. -1. -1}}. /* 123 3 */
{{ 8. 6. 9. 8. 7. 6. 9. 6. 1. 10. 3. 6. 1. 6. 3. -1}}. /* 124 7 */
{{ 0. 1. 9. 10. 7. 6. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1}}. /* 125 4 */
{{ 7. 0. 8. 7. 6. 0. 3. 0. 10. 10. 0. 6. -1. -1. -1. -1. -1}}. /* 126 3 */
{{ 7. 6. 10. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1}}. /* 127 1 */
{{ 7. 10. 6. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1}}. /* 128 1 */
{{ 3. 8. 0. 10. 6. 7. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1}}. /* 129 3 */
{{ 0. 9. 1. 10. 6. 7. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1}}. /* 130 4 */
{{ 8. 9. 1. 8. 1. 3. 10. 6. 7. -1. -1. -1. -1. -1. -1. -1. -1. -1}}. /* 131 7 */
{{11. 2. 1. 6. 7. 10. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1}}. /* 132 3 */
{{ 1. 11. 2. 3. 8. 0. 6. 7. 10. -1. -1. -1. -1. -1. -1. -1. -1. -1}}. /* 133 6 */
{{ 2. 0. 9. 2. 9. 11. 6. 7. 10. -1. -1. -1. -1. -1. -1. -1. -1. -1}}. /* 134 7 */
{{ 6. 7. 10. 2. 3. 11. 11. 3. 8. 11. 8. 9. -1. -1. -1. -1. -1}}. /* 135 12 */
{{ 7. 3. 2. 6. 7. 2. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1}}. /* 136 2 */
{{ 7. 8. 0. 7. 0. 6. 6. 0. 2. -1. -1. -1. -1. -1. -1. -1. -1. -1}}. /* 137 5 */
{{ 2. 6. 7. 2. 7. 3. 0. 9. 1. -1. -1. -1. -1. -1. -1. -1. -1. -1}}. /* 138 7 */
{{ 1. 2. 6. 1. 6. 8. 1. 8. 9. 8. 6. 7. -1. -1. -1. -1. -1}}. /* 139 14 */
{{11. 6. 7. 11. 7. 1. 1. 7. 3. -1. -1. -1. -1. -1. -1. -1. -1. -1}}. /* 140 5 */
{{11. 6. 7. 1. 11. 7. 1. 7. 8. 1. 8. 0. -1. -1. -1. -1. -1}}. /* 141 9 */
{{ 0. 7. 3. 0. 11. 7. 0. 9. 11. 6. 7. 11. -1. -1. -1. -1. -1}}. /* 142 11 */
{{ 7. 11. 6. 7. 8. 11. 8. 9. 11. -1. -1. -1. -1. -1. -1. -1. -1. -1}}. /* 143 5 */
{{ 6. 4. 8. 10. 6. 8. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1}}. /* 144 2 */
{{ 3. 10. 6. 3. 6. 0. 0. 6. 4. -1. -1. -1. -1. -1. -1. -1. -1. -1}}. /* 145 5 */
{{ 8. 10. 6. 8. 6. 4. 9. 1. 0. -1. -1. -1. -1. -1. -1. -1. -1. -1}}. /* 146 7 */
{{ 9. 6. 4. 9. 3. 6. 9. 1. 3. 10. 6. 3. -1. -1. -1. -1. -1}}. /* 147 11 */
{{ 6. 4. 8. 6. 8. 10. 2. 1. 11. -1. -1. -1. -1. -1. -1. -1. -1. -1}}. /* 148 7 */
{{ 1. 11. 2. 3. 10. 0. 0. 10. 6. 0. 6. 4. -1. -1. -1. -1. -1. -1. -1}}. /* 149 12 */
{{ 4. 8. 10. 4. 10. 6. 0. 9. 2. 2. 9. 11. -1. -1. -1. -1. -1}}. /* 150 10 */
{{11. 3. 9. 11. 2. 3. 9. 3. 4. 10. 6. 3. 4. 3. 6. -1}}. /* 151 7 */
{{ 8. 3. 2. 8. 2. 4. 4. 2. 6. -1. -1. -1. -1. -1. -1. -1. -1. -1}}. /* 152 5 */
{{ 0. 2. 4. 4. 2. 6. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1}}. /* 153 8 */
{{ 1. 0. 9. 2. 4. 3. 2. 6. 4. 4. 8. 3. -1. -1. -1. -1. -1}}. /* 154 12 */
{{ 1. 4. 9. 1. 2. 4. 2. 6. 4. -1. -1. -1. -1. -1. -1. -1. -1. -1}}. /* 155 5 */
{{ 8. 3. 1. 8. 1. 6. 8. 6. 4. 6. 1. 11. -1. -1. -1. -1. -1}}. /* 156 14 */
{{11. 0. 1. 11. 6. 0. 6. 4. 0. -1. -1. -1. -1. -1. -1. -1. -1. -1}}. /* 157 5 */
{{ 4. 3. 6. 4. 8. 3. 6. 3. 11. 0. 9. 3. 11. 3. 9. -1}}. /* 158 7 */
{{11. 4. 9. 6. 4. 11. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1}}. /* 159 2 */
{{ 4. 5. 9. 7. 10. 6. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1}}. /* 160 3 */
{{ 0. 3. 8. 4. 5. 9. 10. 6. 7. -1. -1. -1. -1. -1. -1. -1. -1. -1}}. /* 161 6 */
{{ 5. 1. 0. 5. 0. 4. 7. 10. 6. -1. -1. -1. -1. -1. -1. -1. -1. -1}}. /* 162 7 */
{{10. 6. 7. 8. 4. 3. 3. 4. 5. 3. 5. 1. -1. -1. -1. -1. -1. -1. -1}}. /* 163 12 */
{{ 9. 4. 5. 11. 2. 1. 7. 10. 6. -1. -1. -1. -1. -1. -1. -1. -1. -1}}. /* 164 6 */
{{ 6. 7. 10. 1. 11. 2. 0. 3. 8. 4. 5. 9. -1. -1. -1. -1. -1. -1. -1}}. /* 165 13 */
{{ 7. 10. 6. 5. 11. 4. 4. 11. 2. 4. 2. 0. -1. -1. -1. -1. -1. -1. -1}}. /* 166 12 */
{{ 3. 8. 4. 3. 4. 5. 3. 5. 2. 11. 2. 5. 10. 6. 7. -1}}. /* 167 6 */
{{ 7. 3. 2. 7. 2. 6. 5. 9. 4. -1. -1. -1. -1. -1. -1. -1. -1. -1}}. /* 168 7 */
{{ 9. 4. 5. 0. 6. 8. 0. 2. 6. 6. 7. 8. -1. -1. -1. -1. -1. -1. -1}}. /* 169 12 */
{{ 3. 2. 6. 3. 6. 7. 1. 0. 5. 5. 0. 4. -1. -1. -1. -1. -1. -1. -1}}. /* 170 10 */
{{ 6. 8. 2. 6. 7. 8. 2. 8. 1. 4. 5. 8. 1. 8. 5. -1}}. /* 171 7 */
{{ 9. 4. 5. 11. 6. 1. 1. 6. 7. 1. 7. 3. -1. -1. -1. -1. -1. -1. -1}}. /* 172 12 */
{{ 1. 11. 6. 1. 6. 7. 1. 7. 0. 8. 0. 7. 9. 4. 5. -1}}. /* 173 6 */
{{ 4. 11. 0. 4. 5. 11. 0. 11. 3. 6. 7. 11. 3. 11. 7. -1}}. /* 174 7 */
{{ 7. 11. 6. 7. 8. 11. 5. 11. 4. 4. 11. 8. -1. -1. -1. -1. -1}}. /* 175 3 */
{{ 6. 5. 9. 6. 9. 10. 10. 9. 8. -1. -1. -1. -1. -1. -1. -1. -1. -1}}. /* 176 5 */
{{ 3. 10. 6. 0. 3. 6. 0. 6. 5. 0. 5. 9. -1. -1. -1. -1. -1. -1. -1}}. /* 177 9 */
{{ 0. 8. 10. 0. 10. 5. 0. 5. 1. 5. 10. 6. -1. -1. -1. -1. -1. -1. -1}}. /* 178 14 */

```


Pe baza tabelii de consultare sunt determinate laturile voxelului intersectate de suprafață, utilizând algoritmul:

```
static int CASE_MASK[8] = {1,2,4,8,16,32,64,128};
    TRIANGLE_CASES *triCase;
    EDGE_LIST *edge;
    int contNum, jOffset, kOffset, idx, ii, jj, index, *vert;

for (contNum=0; contNum < numValues; contNum++)
    {
    value = values[contNum];
    // Construiește indexul
    for ( ii=0, index = 0; ii < 8; ii++)
        if ( s[ii] >= value )
            index |= CASE_MASK[ii];

    if ( index == 0 || index == 255 ) continue; // nu e suprafata

    triCase = triCases + index;
    edge = triCase->edges;
    }
}
```

Tehnica adoptată în teză constă deci în construirea indexului voxelului, acesta asigurând deplasarea (*offsetul*) în tabelă, exact la poziția corespunzătoare intersecțiilor laturilor voxelului curent cu suprafața.

5.5.3 Determinarea suprafeței

Conform metodei adoptate descrise, suprafața este aproximată prin petice triunghiulare. Triunghiurile de aproximare sunt determinate de punctele de intersecție dintre laturile voxelului și suprafață. Punctul de intersecție, poate fi considerat la mijlocul laturii intersectate de suprafață. Cunoscându-se coordonatele la capetele segmentului, se determină geometric coordonatele mijlocului segmentului [136].

În teză, s-a utilizat o metodă eficientă de aproximare a punctului de intersecție, care permite determinarea cu exactitate a poziției acestuia ținându-se seama de valorile scalare ale densităților la capetele intervalelor. Rezultatele metodei se reflectă în realismul și calitatea imaginii. Astfel, punctul de intersecție se calculează prin interpolare liniară a densităților din vârfurile de la capetele laturii intersectate. Valoarea densității în punctul de intersecție este chiar valoarea densității izosuprafeței ce se dorește a fi extrasă. Dacă P_1 și P_2 sunt vârfurile ce delimitează o latură intersectată, iar V_1 și V_2 sunt valorile scalare ale acestor vârfuri, punctul de intersecție P este dat de relația:

$$P = P_1 + (\text{const} - V_1) \cdot \frac{P_2 - P_1}{V_2 - V_1} \quad (5-10)$$

Algoritmul pentru interpolare liniară și determinare a triunghiurilor de aproximare este:

```

for ( ; edge[0] > -1; edge += 3 )
{
    for (ii=0; ii<3; ii++) //insereaza triunghi
    {
        vert = edges[edge[ii]];
        t = (value - s[vert[0]]) / (s[vert[1]] - s[vert[0]]);
        x1 = pts[vert[0]];
        x2 = pts[vert[1]];
        for (jj=0; jj<3; jj++)
        {
            x[jj] = x1[jj] + t * (x2[jj] - x1[jj]);
        }
        // insereaza un nou punct
        if ( (ptIds[ii] = locator->PointInsert(x)) < 0 )
        {
            ptIds[ii] = locator->NextPointInsert(x);
        }
        // verifica sa nu fie triunghi degenerat
        if ( ptIds[0] != ptIds[1] &&
            ptIds[0] != ptIds[2] &&
            ptIds[1] != ptIds[2] )
        {
            newPolys->NextCellInsert(3,ptIds);
        }
    }
}

```

În urma aplicării acestui algoritm sunt determinate peticele triunghiulare care aproximează suprafața inițială. Înainte ca triunghiurile să fie memorate ca atare în tabloul *newPolys* se execută o verificare în urma căreia se elimină triunghiurile *degenerate*, adică acelea la care două din cele trei vârfuri coincid. În acest fel se economisește memorie, se elimină calcule suplimentare la afișare, rezultatul fiind o afișare rapidă fără erori.

5.5.4 Coerența

Această optimizare permite algoritmului să beneficieze de avantajul coerenței pixel cu pixel, linie cu linie și plan cu plan. Pentru cuburile interioare setului de date tridimensional original (cele care nu se află la limitele acestuia, adică nu includ planul 0, linia 0 sau pixelul 0), este suficient să se ia în calcul doar trei dintre laturi pentru fiecare cub. Celelalte două laturi se obțin de la planul, linia, sau pixelul anterior, adică de la cubul procesat într-o etapă anterioară. În figura 5-27, sunt marcate laturile disponibile prin antecalculare. Pentru noul cub este necesar să se calculeze doar laturile 6, 7 și 12.

Implementarea acestei optimizări, presupune implementarea software corespunzătoare.

Utilizarea coerenței mărește viteza algoritmului de 3 ~ 4 ori.

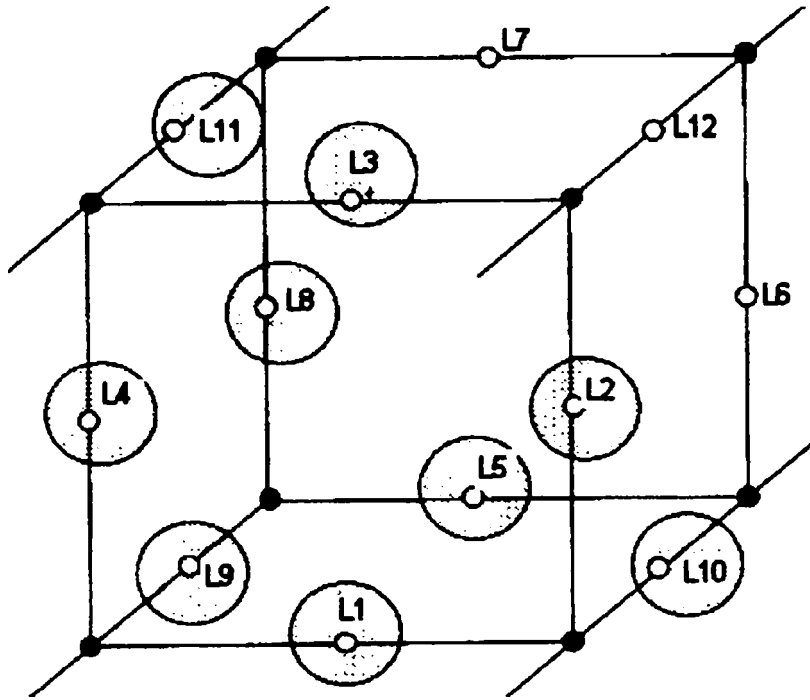


Fig.5-27 Coerența: laturile marcate sunt ‘disponibile’ din calcule anterioare. Trebuie procesate doar laturile 6,7 și 12

În secvența de cod următoare se exemplifică cele precizate, x_p , y_p , z_p reprezentând valorile menținute din calculul anterior, doar $pts[0,0]$, $pts[0,1]$, $pts[0,2]$ sunt puncte noi, ce determină laturi noi:

```
sliceSize = dims[0] * dims[1];
for ( k=0; k < (dims[2]-1); k++)
{
  kOffset = k*sliceSize;
  pts[0][2] = origin[2] + k*Spacing[2];
  zp = origin[2] + (k+1)*Spacing[2];
  for ( j=0; j < (dims[1]-1); j++)
  {
    jOffset = j*dims[0];
    pts[0][1] = origin[1] + j*Spacing[1];
    yp = origin[1] + (j+1)*Spacing[1];
    for ( i=0; i < (dims[0]-1); i++)
    {
      // creeaza punctele voxelului
      pts[0][0] = origin[0] + i*Spacing[0];
      xp = origin[0] + (i+1)*Spacing[0];

      pts[1][0] = xp;
      pts[1][1] = pts[0][1];
      pts[1][2] = pts[0][2];
    }
  }
}
```

```

pts[2][0] = xp;
pts[2][1] = yp;
pts[2][2] = pts[0][2];

pts[3][0] = pts[0][0];
pts[3][1] = yp;
pts[3][2] = pts[0][2];

pts[4][0] = pts[0][0];
pts[4][1] = pts[0][1];
pts[4][2] = zp;

pts[5][0] = xp;
pts[5][1] = pts[0][1];
pts[5][2] = zp;

pts[6][0] = xp;
pts[6][1] = yp;
pts[6][2] = zp;

pts[7][0] = pts[0][0];
pts[7][1] = yp;
pts[7][2] = zp;

```

5.5.5 Rezoluția

Reducerea rezoluției fiecărei imagini din stivă, prin medierea a patru pixeli într-unul singur, are ca rezultat scăderea numărului de triunghiuri, măbind eficiența construcției suprafeței și netezind imaginea. Deși pot apare pierderi de detaliu în aceste imagini, se poate aplica acest procedeu, în cazul în care ele sunt de mare rezoluție, rezultând suprafețe mult mai convenabile din punct de vedere al numărului de triunghiuri de aproximare.

Foarte important este și pasul rețelei, ca parametru ce exprimă rezoluția rețelei tridimensionale generate. Dacă acest pas este mare atunci apar pierderi de detaliu. Cu cât pasul este mai mic, adică rezoluția rețelei mai mare, cu atât rezultatul va fi mai aproape de realitate (figura 5-28). La rezoluții foarte mari ale rețelei, este însă necesară o mare putere de calcul [91][93][102][136].

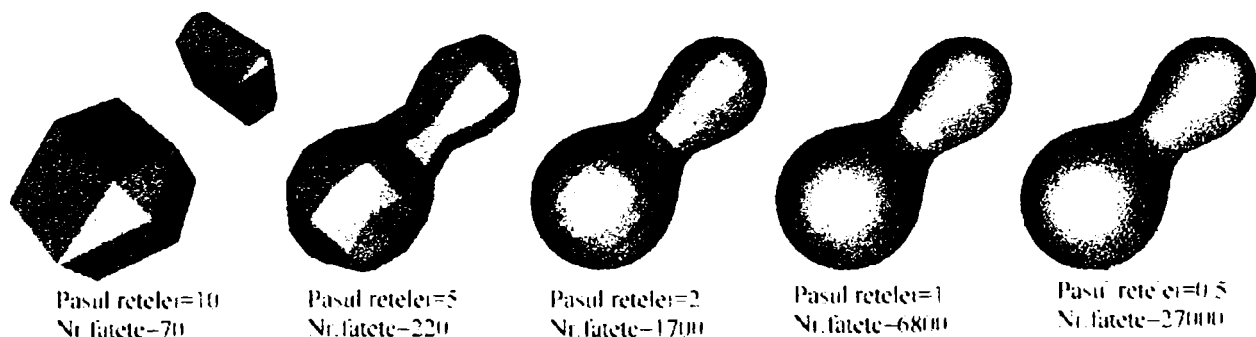


Fig.5-28 Influența pasului rețelei asupra calității imaginii

În figura 5-28 se poate observa că scăderea pasului rețelei duce la creșterea numărului de fațete și deci la creșterea calității suprafeței 3D rezultate.

În implementarea software, parametrul *Spacing* este cel care influențează rezoluția:

```
pts[0][2] = origin[2] + k*Spacing[2];
zp = origin[2] + (k+1)*Spacing[2];
for ( j=0; j < (dims[1]-1); j++)
{
  jOffset = j*dims[0];
  pts[0][1] = origin[1] + j*Spacing[1];
  yp = origin[1] + (j+1)*Spacing[1];
  for ( i=0; i < (dims[0]-1); i++)
  {
    pts[0][0] = origin[0] + i*Spacing[0];
    xp = origin[0] + (i+1)*Spacing[0];
    |
  }
}
```

5.5.6 Valori de prag multiple

O facilitate nouă, extrem de utilă, implementată în teză, o reprezintă faptul că algoritmul permite extragerea nu numai a unei suprafețe de o anumită densitate, ci se pot extrage suprafețe ale căror densități fac parte dintr-un interval de valori. Astfel, se poate vizualiza un organ în secțiune (figura 5-29), fiind detectabile atât țesutul exterior (pielea), cât și un țesut interior (mușchi sau os).



Fig.5-29 Structura de țesut moale și structura de țesut dur vizibile în același timp

Această facilitate a fost implementată prin execuția de mai multe ori a nucleului principal al algoritmului, pentru fiecare valoare dorită a densității:


```

// pentru fiecare valoare a densitatii ce face parte din intervalul de interes
for (contNum=0; contNum < numValues; contNum++)
{
    value = values[contNum];
    // executa filtrul
    // Construiește indexul
    for ( ii=0, index = 0; ii < 8; ii++)
        if ( s[ii] >= value )
            index |= CASE_MASK[ii];

    if ( index == 0 || index == 255 ) continue; // nu e suprafata

    triCase = triCases + index;
    edge = triCase->edges;
    //
    // aproximeaza suprafata si trece la urmatoarea
}

```

5.5.7 Disecție virtuală

Un element de originalitate în implementare îl reprezintă introducerea, în contextul realității virtuale, a termenului de *actor*. Un actor este o suprafață tridimensională extrasă și care prezintă capacități de manevrabilitate individuale. Într-un anumit context pot exista mai mulți *actori*. În aplicațiile ce fac obiectul tezei, un actor poate fi o suprafață extrasă la nivelul pielii și altul poate fi o extracție la nivelul osului, ele putând fi investigate separat sau împreună, în funcție de obiectivul urmărit. Această nouă metodă introduce posibilitatea *disecției virtuale* a organului de investigat (figura 5-30).

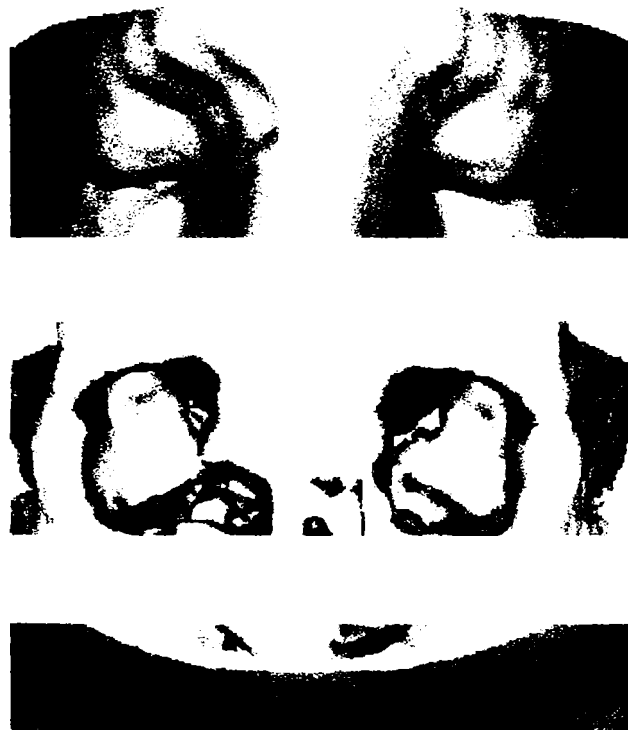


Fig.5-30 Reconstituire la nivelul pielii, al țesutului osos, și ambele deodată, separate prin culoare

În figura 5-30 este prezentată o reconstituire 3D a părții superioare a capului unui pacient. Mai întâi este evidențiat țesutul moale de la exterior, pielea, apoi, prin schimbarea parametrului ce exprimă densitatea dorită, este evidențiată structura osoasă ca actor separat. Cele două suprafețe pot fi reliefate și simultan, așa cum se observă în figura 5-30, unde regiunea frontală este reconstituită atât la nivelul pielii cât și al osului. Acest lucru este posibil prin reconstituirea separată a suprafețelor de densități diferite și prin tratarea lor ca *actori*, obiecte 3D de sine stătătoare, așa cum rezultă din codul prezentat:

```
// extrage suprafata pielii
Filter *piele = Filter::New();
piele->Input(v->Output());
piele->Value(0, 500);
// Iextrage suprafata, si apoi declara ca actor si seteaza culori
Actor *actor_piele = Actor::New();
actor_piele->Property()->Color(1, .49, .25);

// extrage suprafata osoasa
Filter *os =Filter::New();
os->Input(v->Output());
os->Value(0, 1150);
// Iextrage suprafata, si apoi declara ca actor si seteaza culori
Actor *actor_os = Actor::New();
actor_os->Property()->Color(1, 1, .9412);

// in acest moment avem doi "actori", suprafata pielii si cea osoasa
// afisarea va prelua separat cele doua suprafete
win->Actor(actor_piele);
win->Actor(actor_os);
```

5.6 RECONSTRUCȚIA 3D A STRUCTURILOR ANATOMICE

Se prezintă aplicații practice pe cazuri reale și rezultatele obținute cu soluția de calcul prezentată. Pornind de la seturi de tomografii obținute cu tomograful Somatom Plus 4 (figura 5-31), vor fi prezentate atât imagini inițiale (figura 5-32) cât și imagini prelucrate.



Fig.5-31 Succesiune de tomografii axiale

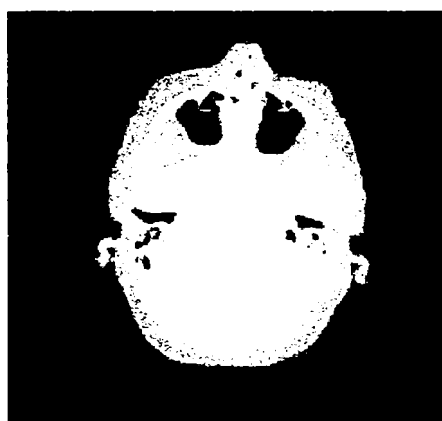


Fig.5-32 Secțiune tomografică prin craniu

Figura 5-32 reprezintă un plan dintr-o tomogramă la nivelul cutiei craniene, în care este evidențiat țesutul osos (porțiunea albă). Imaginea constituie un exemplu dintr-un set de date de intrare pe baza cărora au fost construite modele 3D.

Toate rezultatele prezentate, succesiunile de imagini cu suprafețe 3D modelate, sunt realizate pe baza următorului nucleu de funcții, care va fi apelat diferit, în funcție de obiectivul urmărit [113]:

```

// Functia constructor al obiectului Filter
Filter::Filter()
{
    this->ContVal = ContVal::New();
    this->CalculeazaNormale = 1;
    this->CalculeazaGradient = 0;
}

// Functia destructor al obiectului Filter
Filter::~Filter()
{
    this->ContVal->Delete();
}

// Calculeaza gradientul
void CalcGradPunct(int i, int j, int k, T *s, int dims[3],
                  int sliceSize, float Spacing[3], float
n[3])
{
    float sp, sm;

    // directia x
    if ( i == 0 )
    {
        sp = s[i+1 + j*dims[0] + k*sliceSize];
        sm = s[i + j*dims[0] + k*sliceSize];
        n[0] = (sm - sp) / Spacing[0];
    }
    else if ( i == (dims[0]-1) )
    {
        sp = s[i + j*dims[0] + k*sliceSize];
        sm = s[i-1 + j*dims[0] + k*sliceSize];
        n[0] = (sm - sp) / Spacing[0];
    }
    else
    {
        sp = s[i+1 + j*dims[0] + k*sliceSize];
        sm = s[i-1 + j*dims[0] + k*sliceSize];
        n[0] = 0.5 * (sm - sp) / Spacing[0];
    }

    // directia y
    if ( j == 0 )
    {
        sp = s[i + (j+1)*dims[0] + k*sliceSize];
        sm = s[i + j*dims[0] + k*sliceSize];
        n[1] = (sm - sp) / Spacing[1];
    }
    else if ( j == (dims[1]-1) )
    {
        sp = s[i + j*dims[0] + k*sliceSize];
        sm = s[i + (j-1)*dims[0] + k*sliceSize];
        n[1] = (sm - sp) / Spacing[1];
    }
    else
    {
        sp = s[i + (j+1)*dims[0] + k*sliceSize];
        sm = s[i + (j-1)*dims[0] + k*sliceSize];
        n[1] = 0.5 * (sm - sp) / Spacing[1];
    }

    // directia z
    if ( k == 0 )
    {
        sp = s[i + j*dims[0] + (k+1)*sliceSize];
        sm = s[i + j*dims[0] + k*sliceSize];
        n[2] = (sm - sp) / Spacing[2];
    }
    else if ( k == (dims[2]-1) )

```

```

{
    sp = s[i + j*dims[0] + k*sliceSize];
    sm = s[i + j*dims[0] + (k-1)*sliceSize];
    n[2] = (sm - sp) / Spacing[2];
}
else
{
    sp = s[i + j*dims[0] + (k+1)*sliceSize];
    sm = s[i + j*dims[0] + (k-1)*sliceSize];
    n[2] = 0.5 * (sm - sp) / Spacing[2];
}
}

// Filtrul...
void FiltruCalc(Filtru *self, T *scalars, int dims[3],
                float origin[3], float Spacing[3],
                Point *locator,
                Array *newScalars,
                Array *newGradients,
                Array *newNormals,
                CArray *newPolys, float *values,
                int numValues)
{
    float s[8], value;
    int i, j, k, sliceSize;
    static int CASE_MASK[8] = {1,2,4,8,16,32,64,128};
    FilterTriangleCases *triCase, *triCases;
    EDGE_LIST *edge;
    int contNum, jOffset, kOffset, idx, ii, index, *vert;
    Id ptIds[3];
    int CalculeazaNormale = newNormals != NULL;
    int CalculeazaGradient = newGradients != NULL;
    int CalculeazaValori = newScalars != NULL;
    int NeedGrad;
    float t, *x1, *x2, x[3], *n1, *n2, n[3], min, max;
    float pts[8][3], gradients[8][3], xp, yp, zp;
    static int edges[12][2] = { {0,1}, {1,2}, {3,2}, {0,3},
                                {4,5}, {5,6}, {7,6}, {4,7},
                                {0,4}, {1,5}, {3,7}, {2,6}};

    triCases = FilterTriangleCases::ReturnCase();

//
// Get min/max contour values
//
    if ( numValues < 1 )
    {
        return;
    }
    for ( min=max=values[0], i=1; i < numValues; i++)
    {
        if ( values[i] < min )
        {
            min = values[i];
        }
        if ( values[i] > max )
        {
            max = values[i];
        }
    }
//
// Traverseaza celulele, genereaza triunghiuri
//
    sliceSize = dims[0] * dims[1];
    for ( k=0; k < (dims[2]-1); k++)
    {
        kOffset = k*sliceSize;
        pts[0][2] = origin[2] + k*Spacing[2];
        zp = origin[2] + (k+1)*Spacing[2];
        for ( j=0; j < (dims[1]-1); j++)

```

```

{
jOffset = j*dims[0];
pts[0][1] = origin[1] + j*Spacing[1];
yp = origin[1] + (j+1)*Spacing[1];
for ( i=0; i < (dims[0]-1); i++)
{
//valorile scalare
idx = i + jOffset + kOffset;
s[0] = scalars[idx];
s[1] = scalars[idx+1];
s[2] = scalars[idx+1 + dims[0]];
s[3] = scalars[idx + dims[0]];
s[4] = scalars[idx + sliceSize];
s[5] = scalars[idx+1 + sliceSize];
s[6] = scalars[idx+1 + dims[0] + sliceSize];
s[7] = scalars[idx + dims[0] + sliceSize];

if ( (s[0] < min && s[1] < min && s[2] < min && s[3] < min &&
s[4] < min && s[5] < min && s[6] < min && s[7] < min) ||
(s[0] > max && s[1] > max && s[2] > max && s[3] > max &&
s[4] > max && s[5] > max && s[6] > max && s[7] > max) )
{
continue;
}

//creeaza punctele voxelilor
pts[0][0] = origin[0] + i*Spacing[0];
xp = origin[0] + (i+1)*Spacing[0];

pts[1][0] = xp;
pts[1][1] = pts[0][1];
pts[1][2] = pts[0][2];

pts[2][0] = xp;
pts[2][1] = yp;
pts[2][2] = pts[0][2];

pts[3][0] = pts[0][0];
pts[3][1] = yp;
pts[3][2] = pts[0][2];

pts[4][0] = pts[0][0];
pts[4][1] = pts[0][1];
pts[4][2] = zp;

pts[5][0] = xp;
pts[5][1] = pts[0][1];
pts[5][2] = zp;

pts[6][0] = xp;
pts[6][1] = yp;
pts[6][2] = zp;

pts[7][0] = pts[0][0];
pts[7][1] = yp;
pts[7][2] = zp;

NeedGrad = CalculeazaGradient || Calculeazanormale;

if (NeedGrad)
{
CalcGradPunct(i,j,k, scalars, dims, sliceSize, Spacing, gradients[0]);
CalcGradPunct(i+1,j,k, scalars, dims, sliceSize, Spacing,
gradients[1]);
CalcGradPunct(i+1,j+1,k, scalars, dims, sliceSize, Spacing,
gradients[2]);
CalcGradPunct(i,j+1,k, scalars, dims, sliceSize, Spacing,
gradients[3]);
CalcGradPunct(i,j,k+1, scalars, dims, sliceSize, Spacing,
gradients[4]);
CalcGradPunct(i+1,j,k+1, scalars, dims, sliceSize, Spacing,
gradients[5]);
}
}

```

```

        CalcGradPunct(i+1,j+1,k+1,  scalars,  dims,  sliceSize,  Spacing,
gradients[6]);
        CalcGradPunct(i,j+1,k+1,  scalars,  dims,  sliceSize,  Spacing,
gradients[7]);
    }
    for (contNum=0; contNum < numValues; contNum++)
    {
        value = values[contNum];
        // Construiește tabela de cazuri
        for ( ii=0, index = 0; ii < 8; ii++)
        {
            if ( s[ii] >= value )
            {
                index |= CASE_MASK[ii];
            }
        }
        if ( index == 0 || index == 255 ) //nu e suprafata
        {
            continue;
        }

        triCase = triCases+ index;
        edge = triCase->edges;

        for ( ; edge[0] > -1; edge += 3 )
        {
            for (ii=0; ii<3; ii++) //insereaza triunghi
            {
                vert = edges[edge[ii]];
                t = (value - s[vert[0]]) / (s[vert[1]] - s[vert[0]]);
                x1 = pts[vert[0]];
                x2 = pts[vert[1]];
                x[0] = x1[0] + t * (x2[0] - x1[0]);
                x[1] = x1[1] + t * (x2[1] - x1[1]);
                x[2] = x1[2] + t * (x2[2] - x1[2]);

                // verifica punct nou
                if ( loc->InsereazaPunct(x, ptIds[ii]) )
                {
                    if (NeedGrad)
                    {
                        n1 = gradients[vert[0]];
                        n2 = gradients[vert[1]];
                        n[0] = n1[0] + t * (n2[0] - n1[0]);
                        n[1] = n1[1] + t * (n2[1] - n1[1]);
                        n[2] = n1[2] + t * (n2[2] - n1[2]);
                    }
                }
            }
            // verifica sa nu fie triunghiuri degenerate
            if ( ptIds[0] != ptIds[1] &&
                ptIds[0] != ptIds[2] &&
                ptIds[1] != ptIds[2] )
            {
                newPolys->Insereaza(3,ptIds);
            }
        } //pentru fiecare triunghi
    } //pentru contururi
} //for i
} //for j
} //for k
}

```

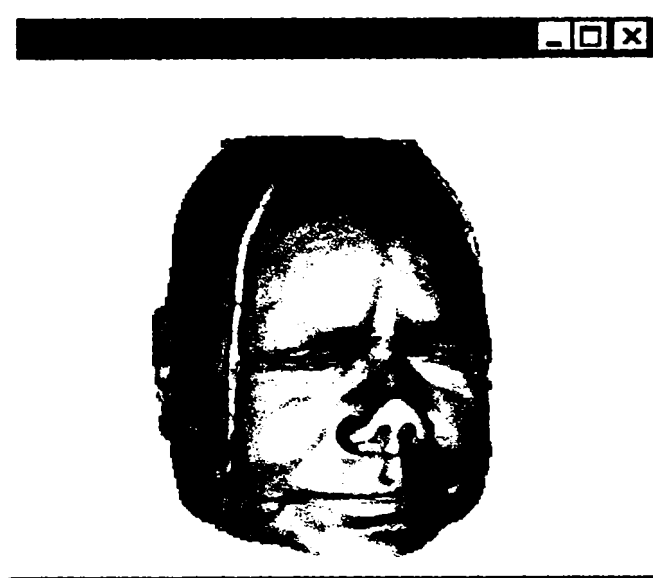


Fig.5-33 Model 3D al capului. Pacientul este o femeie în vârstă.

În figura 5-33 este prezentat rezultatul modelării dintr-un set de tomograme la nivelul capului. Reconstrucția 3D este realizată la nivel de țesut moale, prin apelul următoarelor succesiuni de funcții:

```
int main (int argc, char **argv)
{
    RenderingContext *r = RenderingContext::New();
    Window *Win = Window::New();
    Win->AddRenderingContext(r);
    RenderingInteraction *i = RenderingInteraction::New();
    i->SetWin(Win);

    //Pacient1
    Reader *R = Reader::New();
    R->Prefix("../.../.../.../pacient1/img");
    R->Format("%s%d.jpg");
    R->Spacing(1,1, 0.5);
    R->Dims(1,256,1,256,1,215);

    Filter *piele = Filter::New();
    piele->Input(R->Output());
    piele->Value(0, 50);

    TriangleArray *triSkin = TriangleArray::New();
    triSkin->Input(piele);

    PolyArray *skin = PolyArray::New();
    skin->Input(triSkin);

    Actor *askin = Actor::New();
    askin->Color(1, .49, .25);
    askin->Opacity(1);

    r->Actor(askin);
    r->Background(1,1,1);
    Win->Size(640, 480);

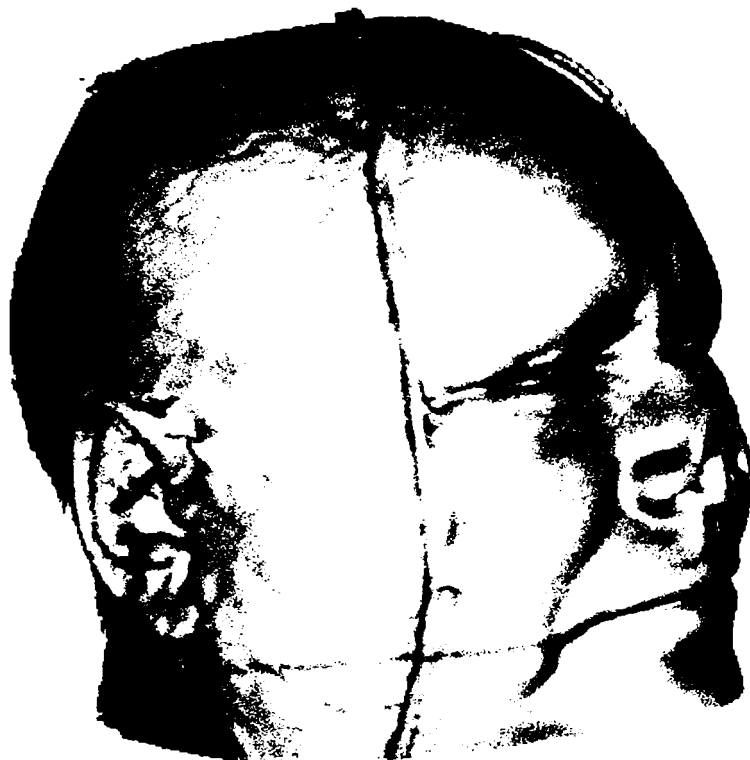
    return 0;
}
```


Pentru izolarea suprafeței 3D ce modelează țesutul moale al capului s-a apelat filtrul de volum cu un parametru de densitate a cărei valoare este 50, pentru exemplul prezentat.

Imaginile 5-34, 5-35, 5-36 sunt variante ale reconstrucției 3D din figura 5-33. Modelele sunt la rezoluție mare pentru a demonstra performanțele soluției propuse și sunt prezentate în poziții diverse, ca rezultat al posibilității de manipulare a lor.



Fig.5-34 Vedere din față a modelului 3D al capului, realizat la nivelul țesutului moale



Ffig.5-35 Vedere laterală a modelului 3D al capului, realizat la nivelul țesutului moale



Fig.5-36 Model 3D rotit, astfel încât se pot observa detalii (este evidențiată urechea internă)

Printre posibilitățile de manipulare independentă a modelelor este implementată și cea de vizualizare, prin poziționare cu ajutorul mouse-ului, din orice unghi (fig.5-36).

Metoda prezentată poate reconstrui deodată mai multe tipuri de țesut (figura 5-37).

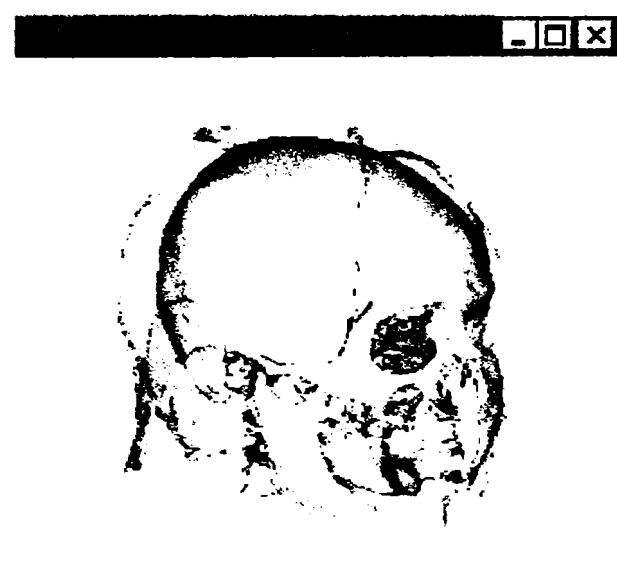


Fig. 5-37 Reconstrucție simultană a două tipuri de țesut

În fereastra program din figura 5-38, 5-39, 5-40 sunt prezentate craniul și pielea capului exact așa cum sunt în realitate, pielea având o nuanță transparentă pentru a se putea distinge prin ea.

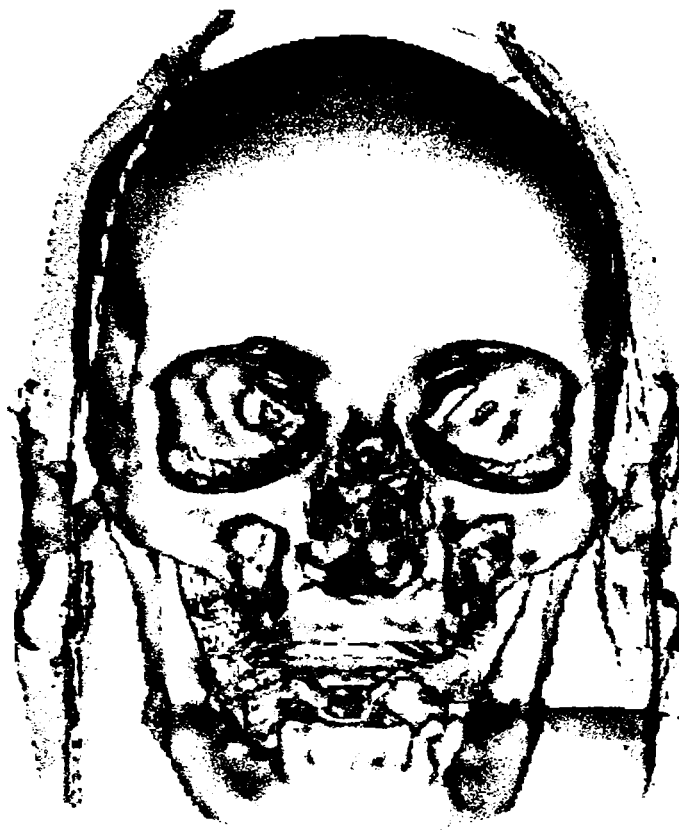


Fig.5-38 Reconstrucție 3D la nivelul țesutului moale și al celui osos, simultan.

Detecția suprafețelor 3D se face asemănător, prin specificarea de această dată a încă unui actor, asociat suprafeței osoase determinate de filtrul apelat cu altă valoare a parametrului de densitate.

```
int main (int argc, char **argv){
    RenderingContext *r = RenderingContext::New();
    Window *Win = Window::New();
    Win->AddRenderingContext(r);
    RenderingInteraction *i = RenderingInteraction::New();
    i->SetWin(Win);
    //Pacient1
    Reader *R = Reader::New();
    R->Prefix("../.../.../pacient1/img");
    R->Format("%s%d.jpg");
    R->Spacing(1,1, 0.5);
    R->Dims(1,256,1,256,1,215);

    Filter *piele = Filter::New();
    piele->Input(R->Output());
    piele->Value(0, 50);
    TriangleArray *triSkin = TriangleArray::New();
    triSkin->Input(piele);
    PolyArray *skin = PolyArray::New();
    skin->Input(triSkin);
    Filter *os = Filter::New();
    os->Input(R->Output());
    os->Value(0, 150);
    TriangleArray *triBone = TriangleArray::New();
```

```

triBone->Input(os);
PolyArray *bone = PolyArray::New();
bone->Input(triBone);
Actor *askin = Actor::New();
askin->Color(1, .49, .25);
askin->Opacity(0.3);
Actor *abone = Actor::New();
abone->Opacity(1);
r->Actor(askin);
r->Actor(abone);
r->Background(1,1,1);
Win->Size(640, 480);
return 0;}

```

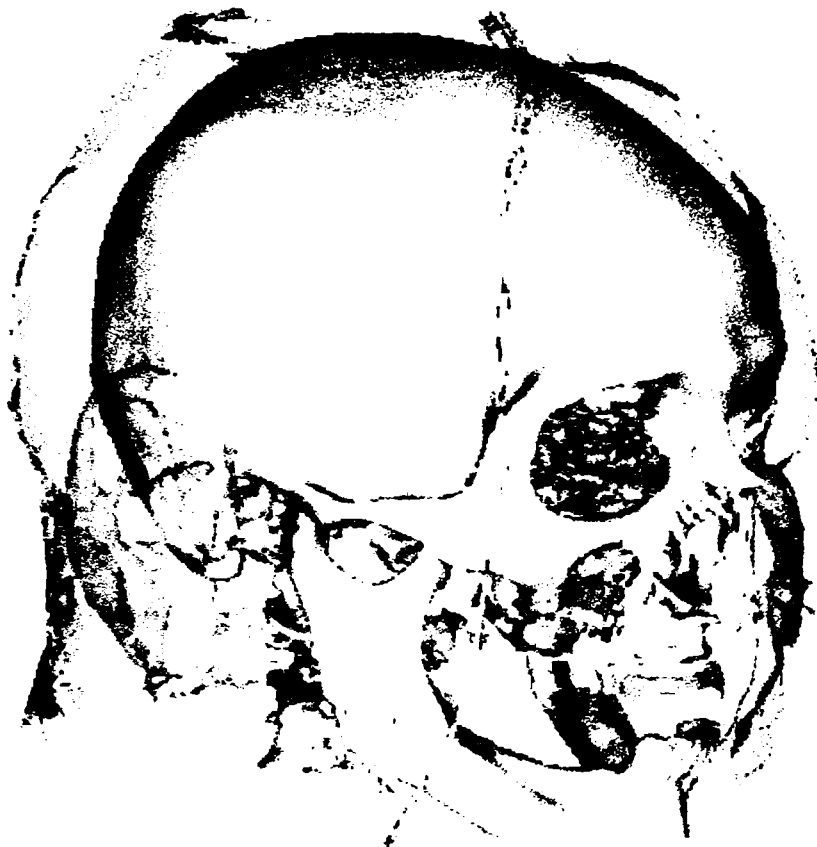


Fig.5-39 Reconstrucție 3D a craniului, ce poate fi observat datorită proprietății de transparență asociată suprafeței 3D ce modelează țesutul moale.

Proprietatea de transparență (figura 5-39) se poate asocia unei anumite suprafețe, așa cum se poate observa în nucleul de funcții prezentat, prin:

```

askin->Color(1, .49, .25);
askin->Opacity(0.3);

```

Unei suprafețe i se poate asocia și o anumită culoare, pentru a crește realismul imaginii și pentru a distinge ușor, diferitele tipuri de suprafețe reconstruite 3D și asociate diferitor tipuri de țesuturi (figura 5-40).

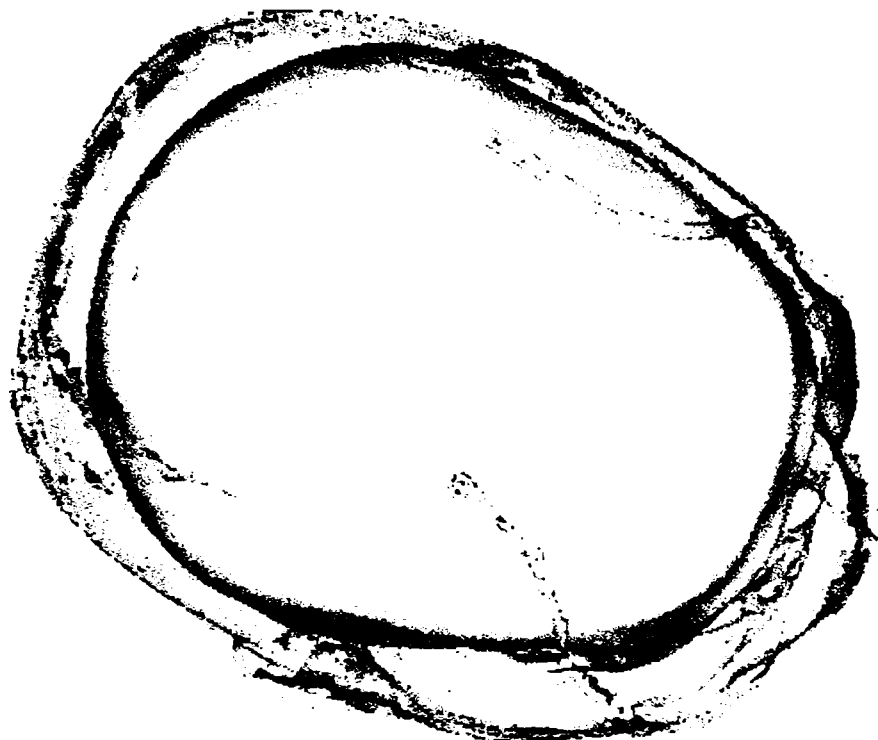


Fig.5-40 Asocierea de culori diferite și transparența/opacitate suprafețelor 3D permite modelarea și vizualizarea completă a unei părți anatomice.

În fereastra program din figura 5-41 este prezentat un model al craniului. Prin aplicarea filtrului pentru o densitate mai mare, corespunzătoare oaselor se obține suprafața 3D corespunzătoare.

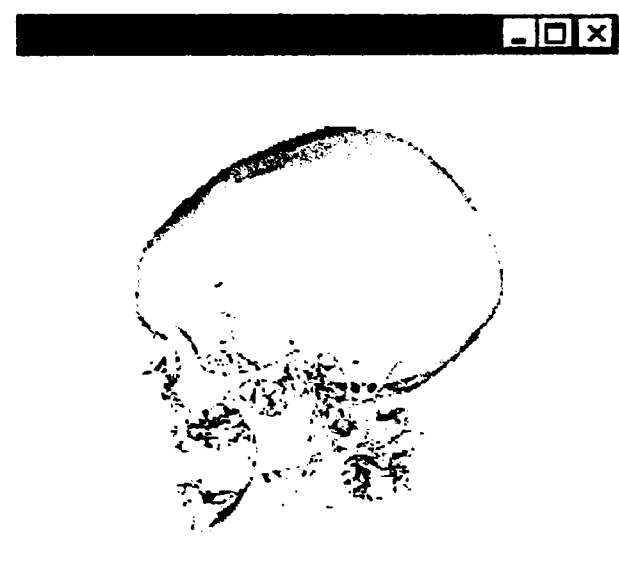


Fig.5-41 Model al craniului obținut pe baza tomogramei provenită de la tomograful Somatom Plus 4.

Modelul craniului este obținut prin apelul filtrului cu valoarea 150 a parametrului de selecție a suprafeței (figurile 5-42, 5-43, 5-44). Opacitatea este 100 % în acest caz pentru că nu se dorește transparența, iar culoarea este setată în gri pentru realism. Liniile de program care tratează acest aspect sunt:

```
Filter *os = Filter::New();
os->Input(R->Output());
os->Value(0, 150);
abone->Color(0.5, 0.5, 0.5);
abone->Opacity(1);
```

Figurile 5-43 și 5-44 prezintă variante din diferite unghiuri de vedere ale acestui model. La nivelul mandibulei poate fi observată o imperfecțiune cauzată de absența câtorva plane din setul de tomografii.

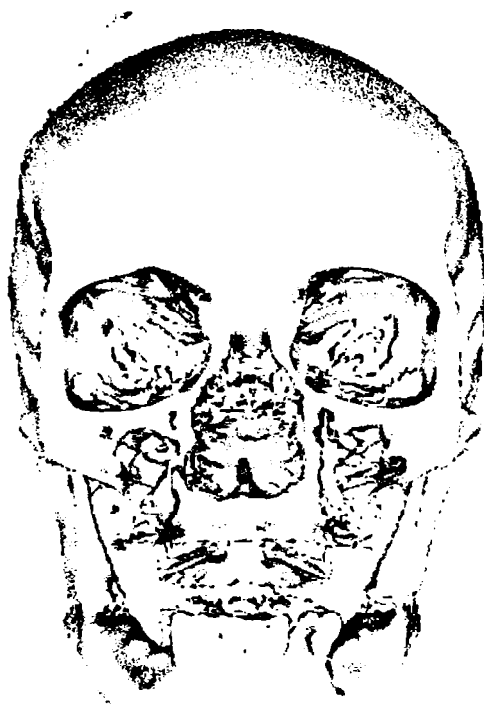


Fig.5-42 Modelarea 3D a craniului.



Fig.5-43 Vedere laterală a modelului 3D al craniului.

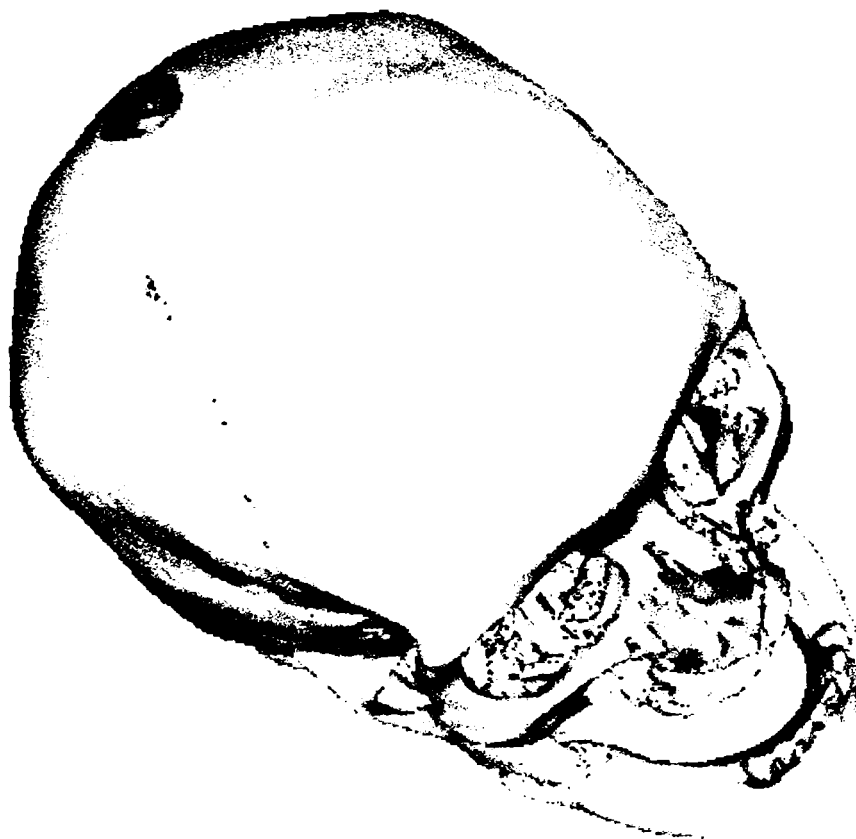


Fig.5-44 Vedere a modelului 3D al craniului obținută după câteva rotații în spațiu

Universitatea Tehnică
TIMIȘOARA
Biblioteca centrală

În fereastra din figura 5-45 se poate observa că dacă rezoluția de scanare este prea mică, acuratețea reconstrucției scade, dar cu toate acestea datorită interpolării, rezultatul este totuși bun. În realitate aceste cazuri sunt frecvente, nu întotdeauna dispunându-se de plane suficiente. În cazuri speciale este indicat să se scaneze cu rezoluție crescută pentru a putea fi puse în evidență toate detaliile.



Fig.5-45 Modelare pe baza unui set de date cu rezoluție de scanare mică

Imaginile din figura 5-46 prezintă modelări ale craniului bazate pe un alt set de date, tomograme în care nu a fost scanată partea superioară a capului, dar rezoluția de scanare a fost bună pentru partea inferioară, ceea ce permite studiul. Tubul care se poate observa că intră în cavitatea bucală a pacientului este necesar pentru administrarea anestezicului, acesta fiind în stare post traumatică.

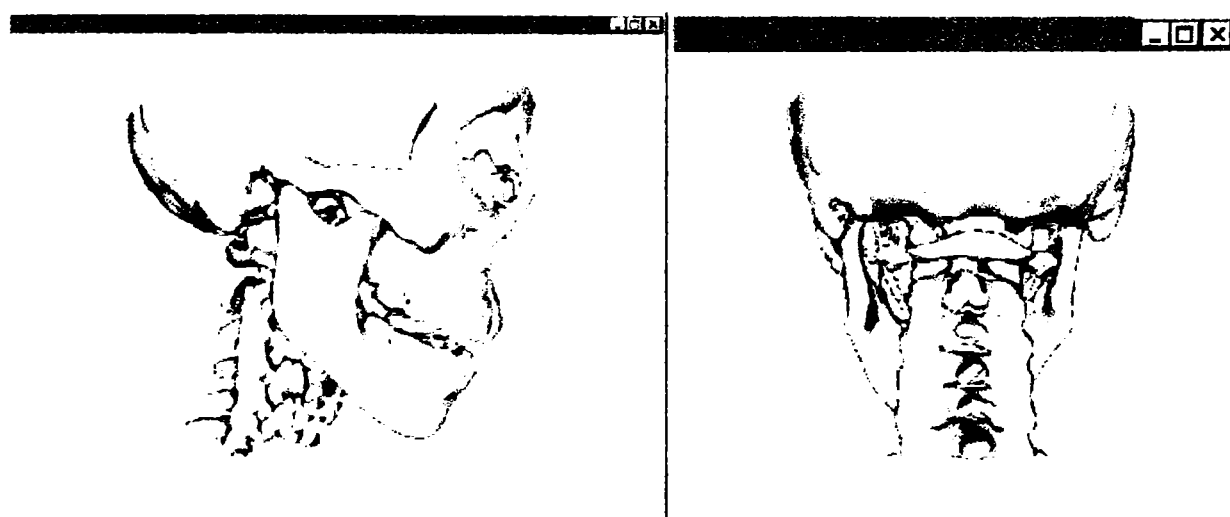


Fig.5-46 Craniul: vedere laterală și din spate. Se pot observa bine și mandibula și vertebrele cervicale

Modelarea 3D a suprafeței la nivelul țesutului moale (al pielii) este prezentată în figura 5-47. Suprafața pielii include detalii fine, cum ar fi chiar orificiul pentru cerceul purtat de pacient. Acest detaliu nu este de importanță clinică, ci constituie o dovadă a rezoluției și calității imaginii obținute. În zona gurii și a bărbiei se observă că în momentul scanării pacientul era intubat.

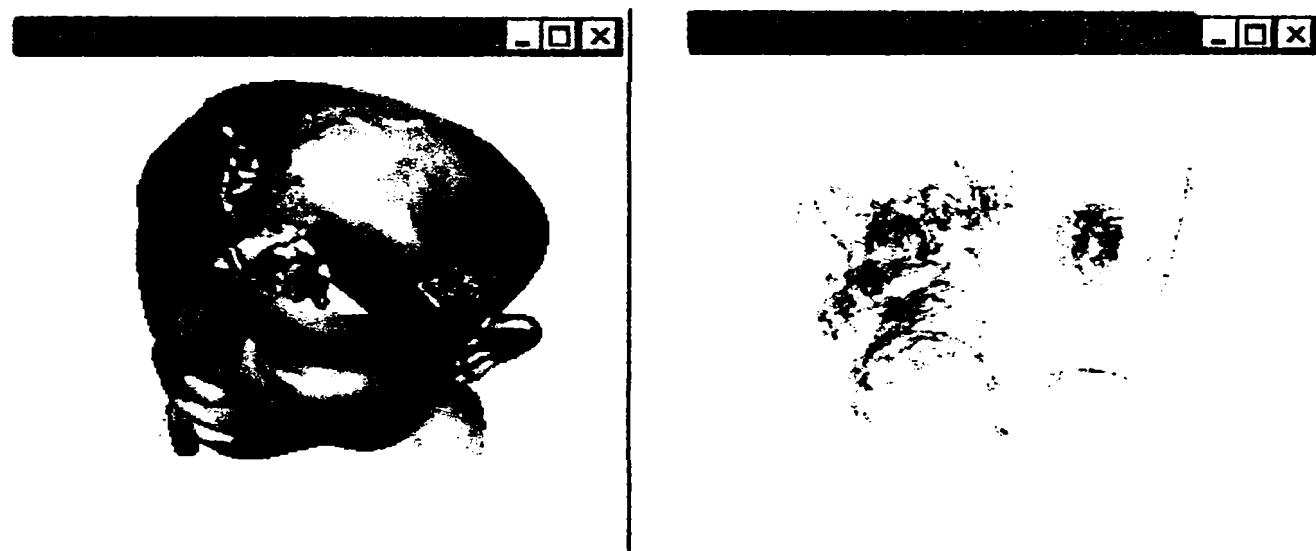


Fig.5-47 Modelarea 3D a suprafeței de piele. Densități apropiate de ale pielii au și țesuturile ce formează urechea internă

Având în vedere că în capitolele ce urmează vor fi tratate aspecte legate de analiza structurilor mecanice reparatorii cu aplicație pe structura osoasă facială, se exemplifică în continuare modelări ale mandibulei (figurile 5-48 și 5-49).

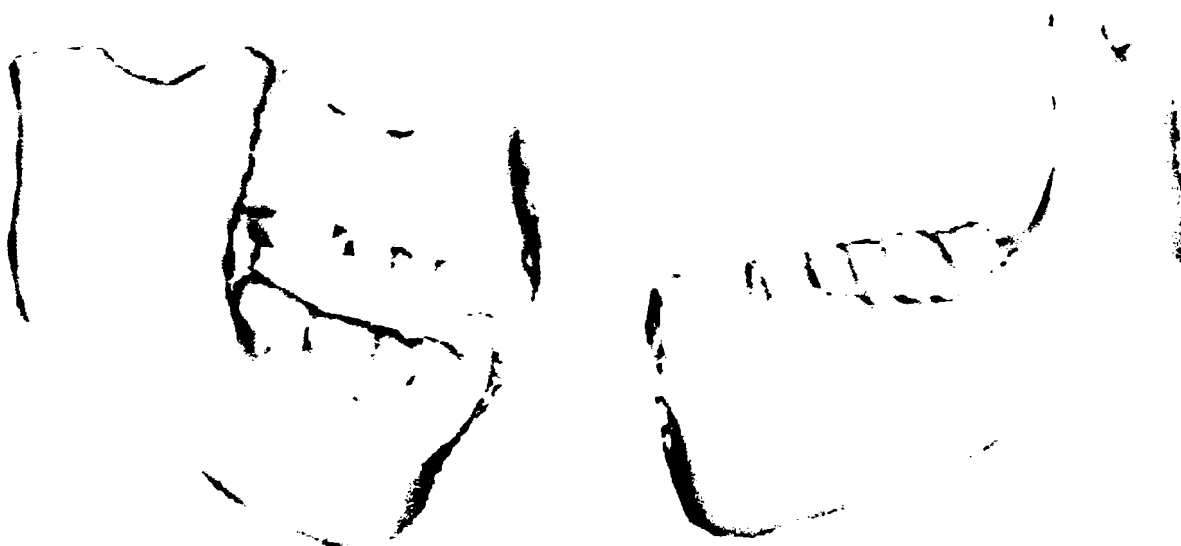


Fig.5-48 Modele 3D pentru maxilar și mandibulă

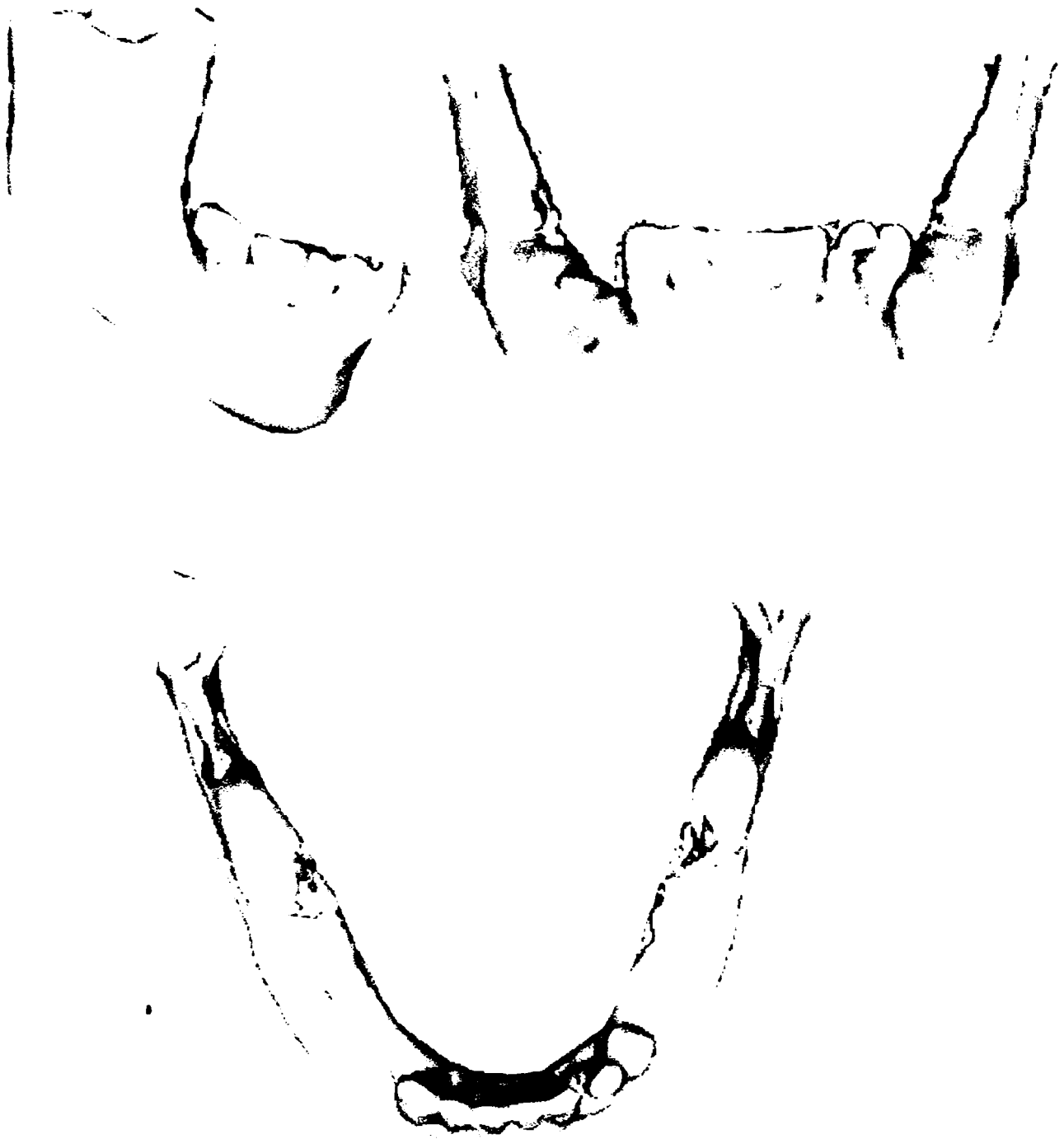


Fig.5-49 Modelări 3D ale osului mandibular realizate pe baza datelor provenite de la pacienți diferiți

Imaginile din figura 5-49 prezintă modelări 3D ale osului mandibular, fiind posibilă observarea osului, a gingiilor și a dentiției.

Cu aceeași tehnică se pot construi modele și pentru alte segmente ale corpului uman. Astfel, imaginile din figurile 5-50, 5-51, 5-52, 5-53, 5-54 prezintă reconstrucții și modele ale membrilor inferioare.

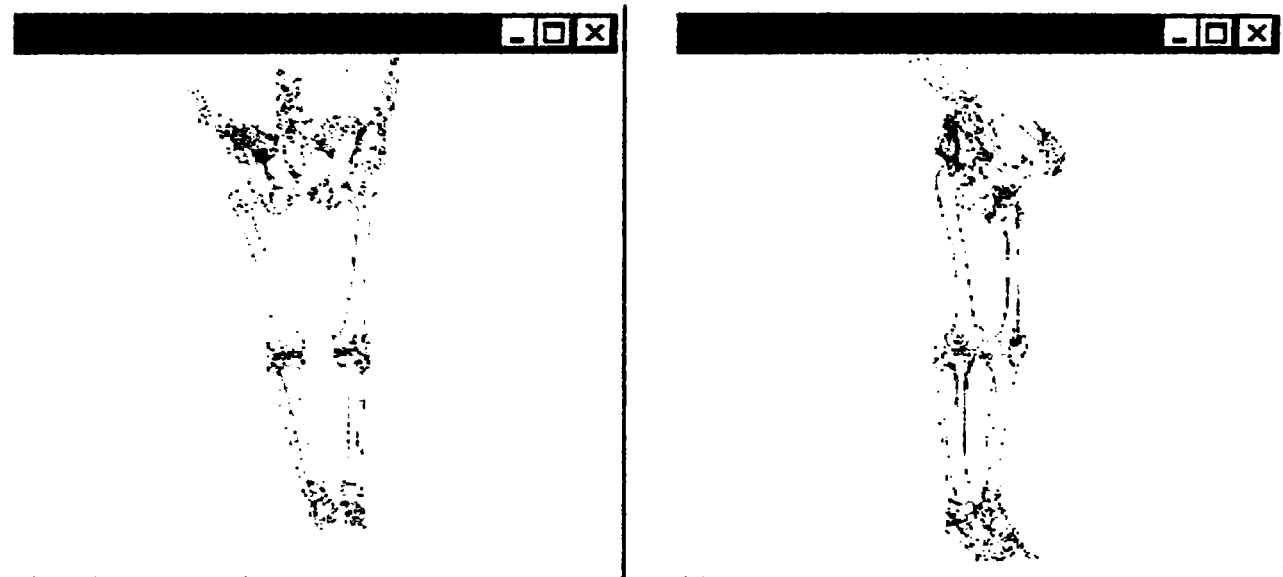


Fig.5-50 Modelarea 3D a scheletului membrilor inferioare. Vedere de ansamblu din unghiuri diferite

Schimbând parametrul asociat densității țesutului se scoate în evidență atât pielea, cât și mușchii sau oasele membrilor inferioare (figura 5-51).



Fig.5-51 Modelări ale membrilor inferioare, fiind reliefate diferite tipuri de țesuturi.

Prin alegerea în funcție de caz a secțiunilor plane din tomograma investigată, se realizează o modelare de ansamblu, sau, dimpotrivă, o modelare de detaliu, așa cum se poate observa în figura 5-52, unde este prezentată mai întâi o modelare complexă de ansamblu, urmată de o modelare detaliată a perechii tibie-peroneu și o modelare a osului femural.

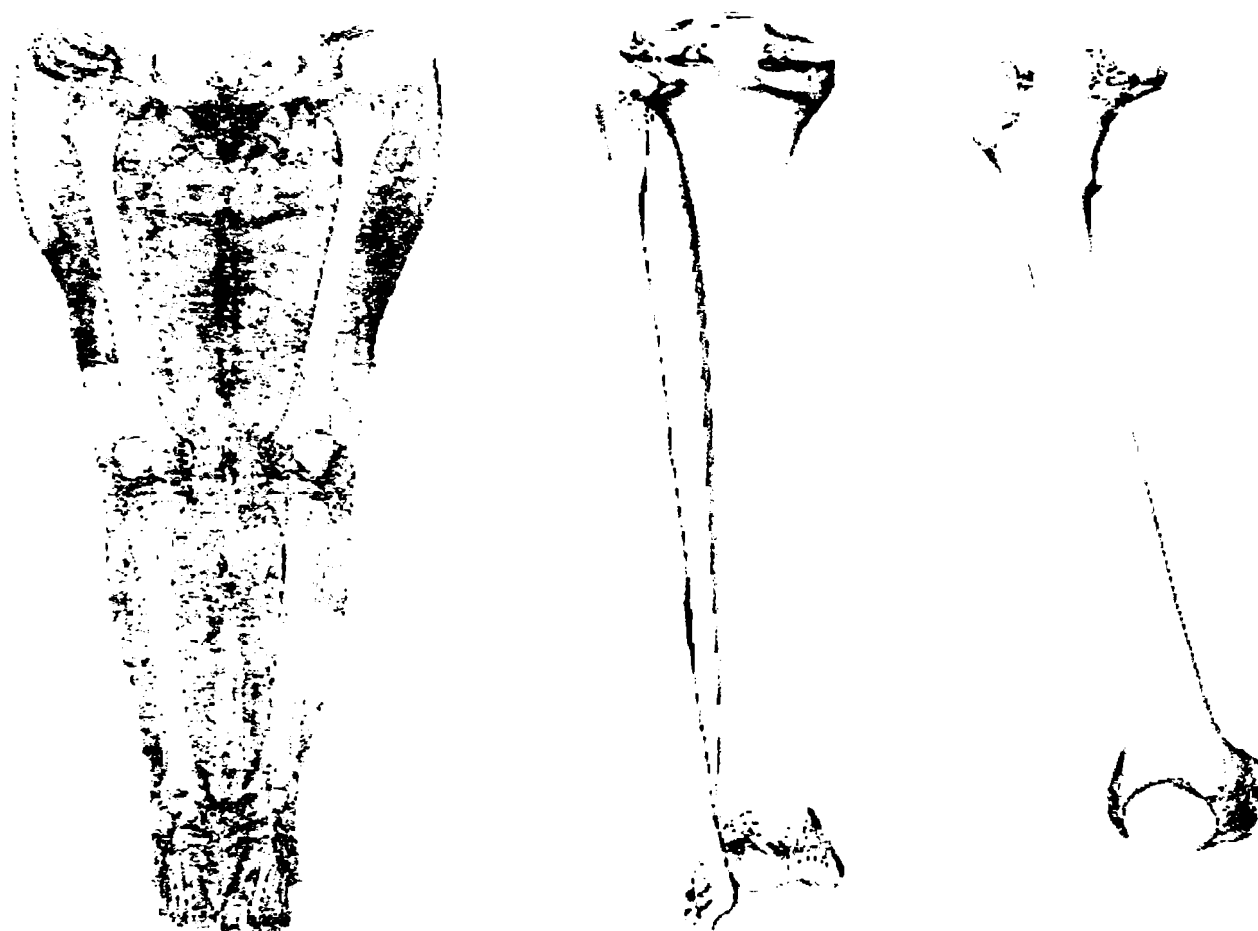


Fig.5-52 Modelări 3D ale membrilor inferioare (în ansamblu), ale perechii tibie-peroneu, precum și ale femurului.

Figurile 5-53 și 5-54 prezintă rezultate ale modelării 3D a labei piciorului. În figura 5-53 modelarea este realizată reliefându-se aspectul exterior. Figura 5-54 înfățișează aspecte interne, legate de scheletul labei piciorului.



Fig.5-53 Modelarea aspectului exterior al labei piciorului.

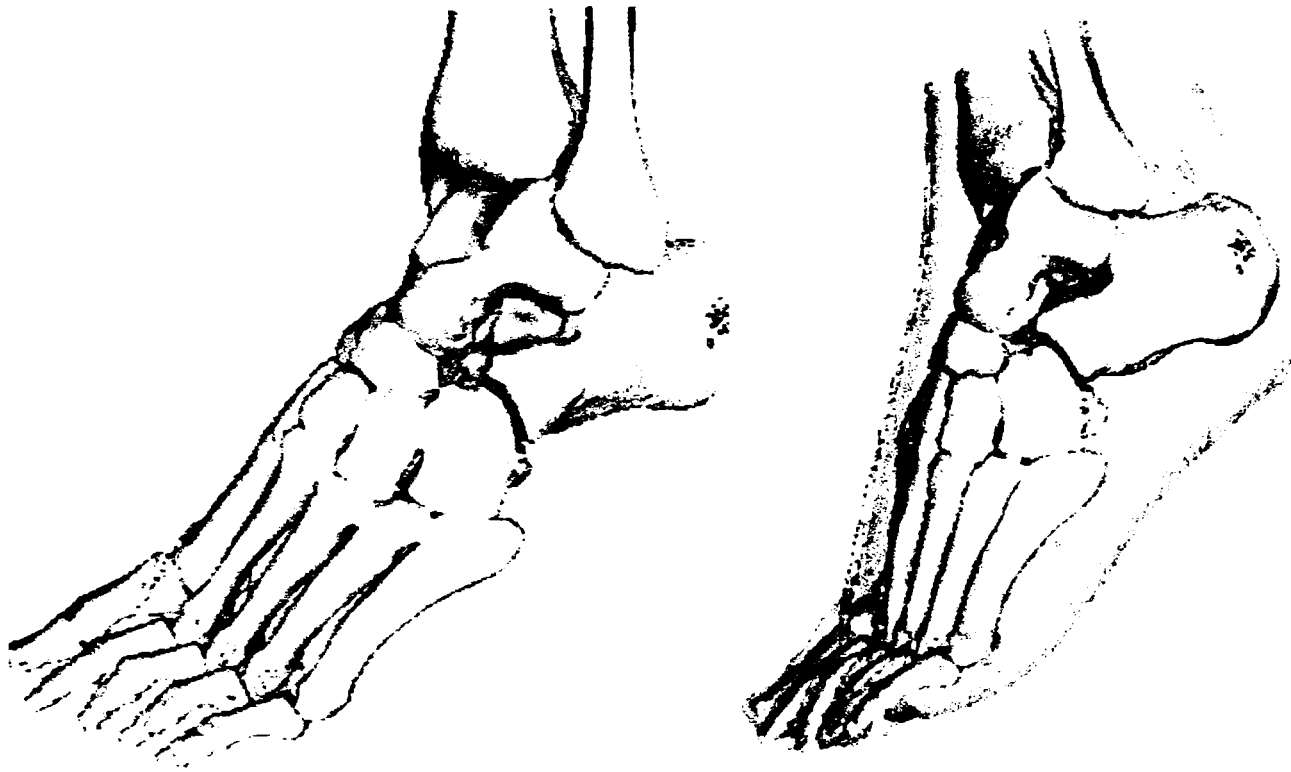


Fig.5-54 Modelarea 3D a scheletului labei piciorului.

Un alt exemplu îl reprezintă modelarea regiunii toracice. Se poate realiza modelarea de la exterior spre interior, reliefându-se atât aspectele exterioare, cât și cele legate de schelet sau organe interne (figurile 5-55 și 5-56).

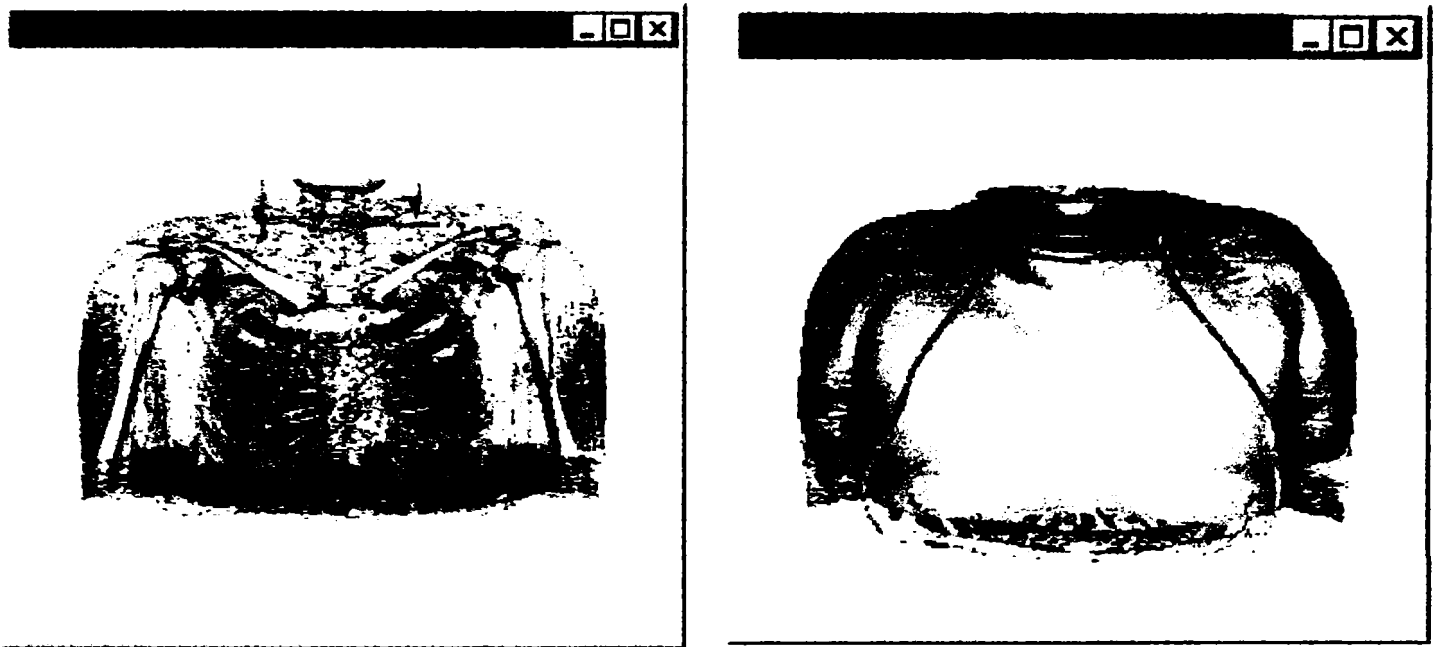


Fig.5-55 Reconstituiri 3D ale regiunii toracice în care se pot observa sternul, coastele și oasele brațelor, precum și o modelare de suprafață.

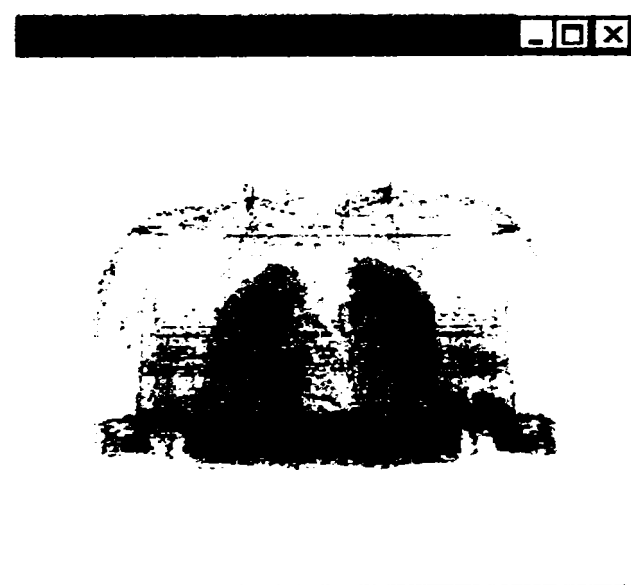


Fig.5-56 Prin aplicarea unei transparențe de 60% țesutului exterior se pot observa plămâni.

Figurile 5-57 și 5-58 prezintă modelări 3D ale scheletului cutiei toracice realizate pe baza a două seturi diferite de date, adică tomogramele a doi pacienți. Prima tomogramă cuprinde și oasele brațelor, iar a doua este focalizată doar asupra cutiei toracice.

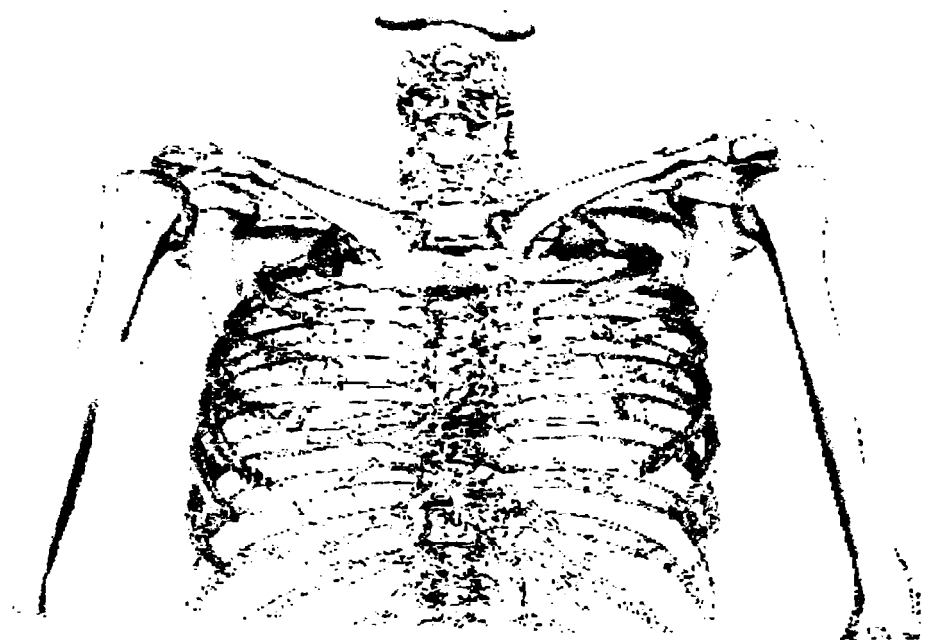


Fig.5-57 Modelarea scheletului cutiei toracice la un pacient de sex feminin cu evidențierea inclusiv a oaselor membrilor superioare

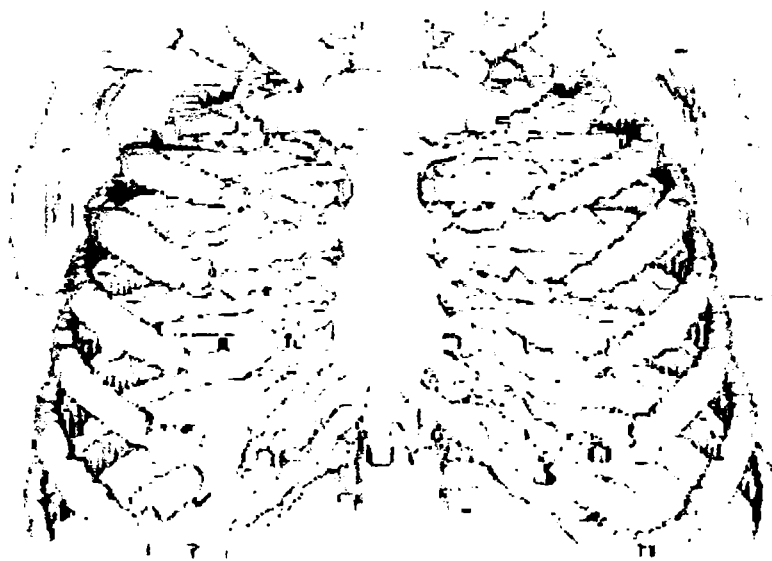


Fig.5-58 Scheletul cutiei toracice a unui pacient de sex masculin

Originalitatea și utilitatea metodei propuse rezultă, așa cum s-a arătat, din originalitatea algoritmilor, ce oferă:

- acuratețe deosebită a modelării 3D (figurile 5-33 până la 5-58);
- viteză de calcul, optimizări originale ce duc la posibilitatea utilizării metodei pe tehnica de calcul uzuală;
- modelare 3D ale diferitelor tipuri de țesuturi separat sau în ansamblu (figurile 5-39, 5-52, 5-54);
- asocierea de culori naturale diferitelor suprafețe modelate (figura 5-53, 5-57);
- operații de rotire și poziționare ce permit explorarea facilă pe ecran (figura 5-59);
- facilități de zoom (figura 5-59);
- posibilitatea disecției virtuale prin schimbarea interactivă a parametrilor ce selectează țesuturile care fac obiectul investigației (figurile 5-35, 5-37, 5-51, 5-52, 5-53, 5-54, 5-55, 5-56);
- posibilitatea de exportare a modelelor 3D în diferite formate grafice proprii programelor CAD, facilitate ce permite prelucrarea ulterioară cu alte programe specializate, cum sunt, de exemplu, cele de analiză cu elemente finite.

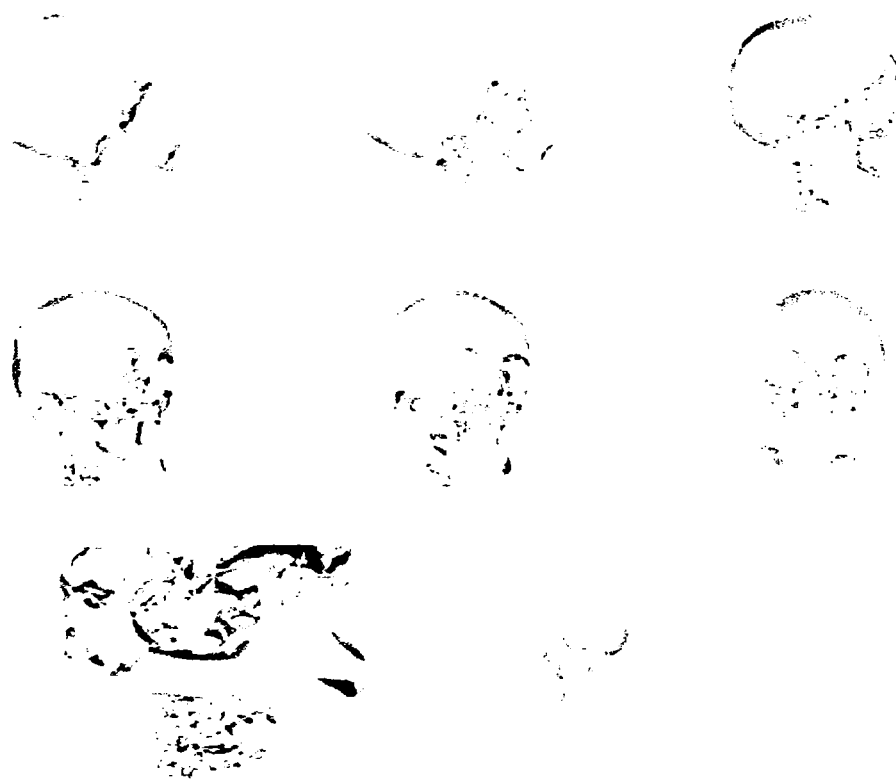


Fig.5-59 Cadre succesive ce demonstrează posibilitățile de investigație din orice unghi, precum și facilități de zoom.

Prin apelul funcțiilor de export, orice model poate fi disponibil în format STL (stereolitografie) sau VRML (Virtual Reality Markup Language):

```
Salvare in fisier STL
Triangles *triangles = Triangles::New();
triangles->Input(model);

STL *stl = STL::New();
stl->Input(triangles);
stl->Name("model1.stl");
stl->Write();

// Salvare in fisier VRML
VRML *exp = VRML::New();
exp->Input(model);
exp->Name("model2.vrml");
exp->Write();
```

Pentru comparație, în figurile 5-60 sunt prezentate câteva imagini preluate de la tomograful Somatom Plus 4, modele care sunt statice (fotografii), nu permit operații asupra lor, pot fi analizate doar din unghiuri standard, nu reliefează mai multe țesuturi simultan, nu permit disecția virtuală și mai ales nu permit utilizarea lor ulterioară în programe specializate.

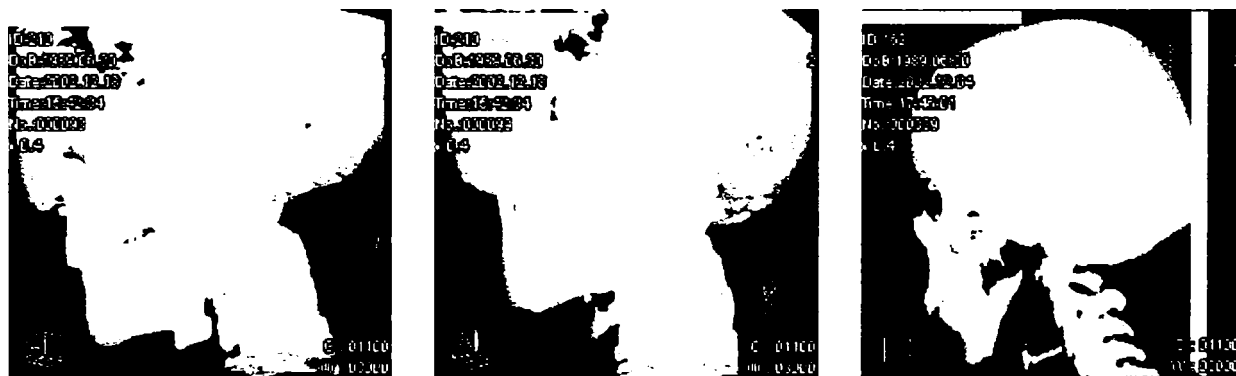


Fig.5-60 Imagini realizate cu Somatom Plus 4

6. CONSIDERAȚII ANATOMICE

Este importantă trecerea în revistă a unor informații anatomice generale despre sistemul osos și muscular, cu aplicații la craniul uman, acestea servind analizei numerice ulterioare. Astfel, s-a urmarit realizarea de studii prin metoda elementelor finite, asupra osului mandibular care nu ar fi putut fi realizate fără o bună cunoaștere a structurii acestuia.

6.1 Oasele capului

Scheletul capului este alcătuit din 22 oase dintre care 8 formează craniul *cerebral* (*neurocraniul*), care conține encefalul, iar celelalte 14 alcătuiesc craniul *visceral* sau al feței (*viscerocraniul*) [19][24][25][48][53][1][2][73][75][97][177].

În general oasele sunt alcătuite pe un principiu inteligent și optimal: *materie puțină și rezistență maximă*. Pe o secțiune longitudinală făcută printr-un os lung se observă la periferia diafizei *periostul*, o membrană conjunctivă vascularizată cu rol în creștere a osului în grosime și în refacerea țesutului osos la nivelul unei fracturi. Dedesubt se găsește os compact, iar spre interior, la nivelul diafizei, se găsește un canal central. În interiorul epifizelor, în locul canalului central apar lame osoase care se încrucișează formând osul spongios [177][19][24][25]. Aceste lame formează structuri adaptate pentru rezistență la tracțiune și presiune, fiind orientate pe direcția solicitărilor mecanice.

Craniul cerebral are formă ovoidală cu partea posterioară mai voluminoasă și prezintă o bază și o boltă. Dintre oasele care-l alcătuiesc, 4 sunt mediane și neperechi: *frontal*, *etmoid*, *sfenoid* și *occipital*, iar alte 4 sunt perechi și așezate lateral: oasele *temporale* și *parietale* (figura 6-1).

- *Osul frontal* are o porțiune verticală care formează fruntea și una orizontală, care include plafonul orbitei și arcadele orbitale. Cavitățile din grosimea osului este *sinusul frontal*.
- *Osul etmoid* conține *cornetele nazale*, care sunt cavități umplute cu aer.
- *Osul sfenoid*, în a cărui cavitate (numită „șaua turcească”) se află *hipofiza*.
- *Osul occipital*, amplasat la ceafă, face legătura cu prima vertebră cervicală, numit *atlas*.
- *Oasele temporale* (tâmpla) găzduiesc conductul auditiv extern și cavitățile urechii interne și urechii medii.
- *Oasele parietale* formează bolta craniană.

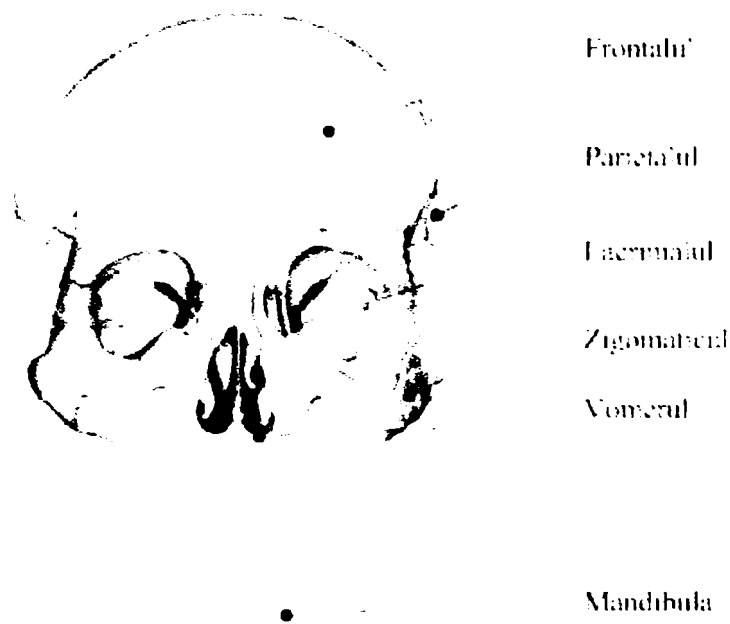


Fig.6-1 Oasele principale ale craniului.

Craniul visceral are 14 oase, dintre care 6 sunt perechi iar două neperechi: *vomerul* și *mandibula* (figura 6-1).

- *Osul vomer* pornește din etmoid și participă la formarea septului nazal.
- *Mandibula* este singurul os mobil al craniului și are forma unei potcoave. Ea este articulată cu oasele temporale prin *condili*. În partea superioară a corpului mandibulei se află alveolele dentare [19][24][25][177][114][105][106].

Cele 12 oase grupate în perechi sunt:

- *Oasele maxilare* sunt oase pneumatice care formează maxila (prin sudare).
- *Oasele palatine* formează *palatul dur*, care separă cavitatea bucală de cavitățile nazale
- *Oasele nazale* constituie partea osoasă a nasului.
- *Oasele lacrimale* se află în unghiul intern superior al orbitelor oculare.
- *Oasele zigomatice* sau „umerii obrazilor” se leagă de oasele temporale.

În afara craniului se mai află un os, *hioidul*, plasat în mușchii gâtului; de el se fixează o parte din mușchii limbii.

Arhitectura masivului facial este determinată de legile dinamicii funcționale ce dirijează organizarea osoasă în stâlpi și grinzi de rezistență, pereți, planșee și plafoane, toate constituind o structură unică ce delimitează cavități în forme și mărimi diferite, situate la etaje și laturi diferite [19][24][25][114][125].

Studiile experimentale clasice [19][24][25][114] precizează că întregul ansamblu arhitectonic al masivului facial este alcătuit din:

I. Zone de maximă rezistență, constituite din stâlpi și grinzi care formează cadrul sistemului. Au grosime și putere maxime în zonele de unire între treveele orizontale și verticale.

II. Zone de slabă rezistență constituite din pereții, planșeele și plafoanele lamelare subțiri situate între fasciculele verticale și orizontale din os compact.

Zonele de maximă rezistență sunt constituite din structuri multiple.

a) Stâlpii de rezistență [24][25][144], formați din travee de os compact, sunt în număr de 6, câte 3 de fiecare parte a masivului facial. Iau naștere prin presiunea funcțională în sens vertical a mandibulei pe maxilar în timpul actului masticator, maxilarele jucând un rol de nicovală pentru ciocanul mandibular. Traveele osoase ce iau naștere la baza maxilarelor se organizează în mănunchiuri de os compact de grosime variind între 3-5 mm ce conduc forțele spre baza craniului, pentru a le dispersa bilateral și simetric în masivul acestui complex de oase.

Prin stâlpii săi de rezistență masivul facial este fixat solid la baza craniului (figura 6-2).

b) Grinzile masivului facial [24][25][144], câte trei de fiecare parte a liniei mediane, sunt constituite din os compact și foarte puțin os spongios (figura 6-2).

c) Planșeele sau platformele orizontale sunt câte două de fiecare parte a liniei mediane. Numai planșeul inferior constituie zona de rezistență.

d) Cavitățile masivului facial sunt multiple, simetrice, de formă, volum și funcții diferite.

Grinzile și platformele orizontale inferioare și mijlocii creează zone de rezistență orizontale, iar pilierii constituie stâlpi de rezistență în sens vertical.

e) Sutura interosoasă atât cele ale masivului facial cât și între oasele acestuia și baza craniului, amortizează șocurile și creează o rezistență specială la acest nivel, făcând majoritatea liniilor de fractură să le ocolească.

Zonele de slabă rezistență se află între pilierii și grinzile orizontale, precum și între planșee. Sunt situate în osul lamelar subțire al pereților. Acestea dirijează în mare măsură direcția liniilor de fractură rezultate secundar prin iradiere, consecutiv traumatismelor de intensitate medie.

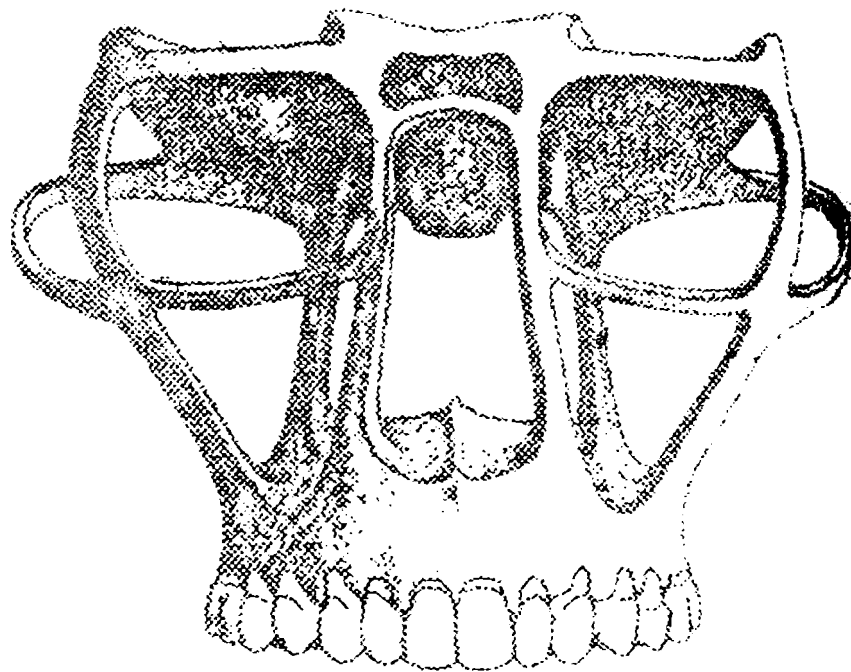


Fig.6-2 Stâlpii și grinzile de rezistență ale masivului facial

6.2 Articulațiile

Oasele sunt legate între ele prin articulații. Acestea pot fi mobile sau fixe, după gradul de mobilitate posibil între oasele respectivei articulații [24][137][153][157].

Articulațiile fixe sunt de obicei sub formă de *suturi*, așa cum apar la oasele parietale ale cutiei craniene, unde se evidențiază suturi dințate. La naștere, nu există încă sutură între oasele parietale, ci o zonă membranoasă (*fontanela*), care permite dezvoltarea dimensională a craniului; ulterior, se osifică și această zonă. La început toate articulațiile sunt mobile dar cele fixe devin ca atare pe măsura avansării în vârstă.

Articulațiile mobile pot avea un grad de mobilitate mai mic sau mai mare; cele cu mobilitate mică (numite *amfiartroze*) se întâlnesc la vertebre. Dacă există nepotriviri de formă între capetele oaselor dintr-o articulație, acestea sunt completate și corectate de către formațiuni fibro-cartilaginoase numite *meniscuri* [157][153][24]. Articulațiile cu mobilitate mare au în general mișcări de rotație și de alunecare.

O forțare ușoară a unei astfel de articulații poate produce o *entorsă*, (entorsele sunt întinderi de ligamente), iar o solicitare mai mare poate genera o *luxație*, când osul iese din articulație.

6.3 Proprietăți mecanice ale sistemului osos

Țesutul osos are o compoziție chimică de tipul: apă 20-25%, reziduu uscat 75-80%. Din reziduuul uscat, 40% este substanță organică (proteina numită *oseină*), restul de 60% reprezentându-l substanțele anorganice (săruri minerale, în special fosfați de calciu). Din păcate, în locul calciului pot fi asimilate substanțe înrudite (exemplu: stronțitul) care înglobate în masa osoasă îi modifică proprietățile mecanice [177][186][208].

Fiind compus din puțină apă și multe substanțe minerale, țesutul osos este dur și servește ca aparat de susținere. Tot fiindcă este rezistent, țesutul osos protejează zonele cele mai importante ale corpului: cutia toracică (care conține și inima și plămânii) și cutia craniană (unde se afla creierul).

Ca depozit de săruri minerale, țesutul osos menține echilibrul de calciu și fosfor din organism.

Forma și materialul osos al craniului sunt astfel constituite (figura 6-3) încât să confere rezistență mecanică și elasticitate deosebită; astfel, dacă se lasă să cadă liber de la înălțimea de 1 – 2 metri, pe o suprafață betonată, craniul uman va sări ca o minge și nu se va sparge decât în condiții anormale, adică dacă posedă deja amorse de ruptură. Craniul rezistă la mari presiuni pe axa boltă-bază, fără a se rupe ci doar micșorându-și eventual înălțimea cu 7-8% [208][177].

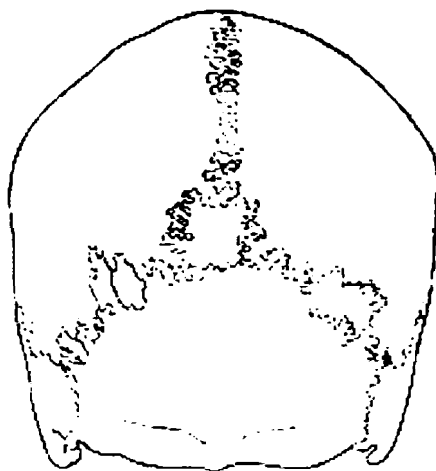


Fig.6-3 Structura de rezistență a craniului.

Rezistența la compresiune a craniului este de 30 de ori mai mare decât a cărămizii [208][177] și de 2,5 ori mai mare decât a granitului; ca valori ale rezistenței și modulului de elasticitate, poate fi comparat doar cu betonul armat, dintre materialele tehnice, și asta datorită arhitecturii sale, a structurii interne și a compoziției chimice.

6.4 MUȘCHII APARATULUI DENTOMAXILAR

Asupra oaselor craniului se exercită forțe care sunt dezvoltate în mușchi. Acestea pun în mișcare mandibula (unicul os mobil al craniului) și diferite porțiuni ale pielii, ceea ce conferă feței o anumită fizionomie. În poziția de echilibru static, cu mușchii în contracție izometrică, se exercită acțiuni diverse asupra oaselor scheletului, forțele maxime depinzând de arhitectura lor naturală.

Mușchii care intră în alcătuirea aparatului dentomaxilar participă de asemenea la menținerea echilibrului capului. Acești mușchi și acțiunea lor principală sunt prezentați în figurile 6-4 și 6-5.

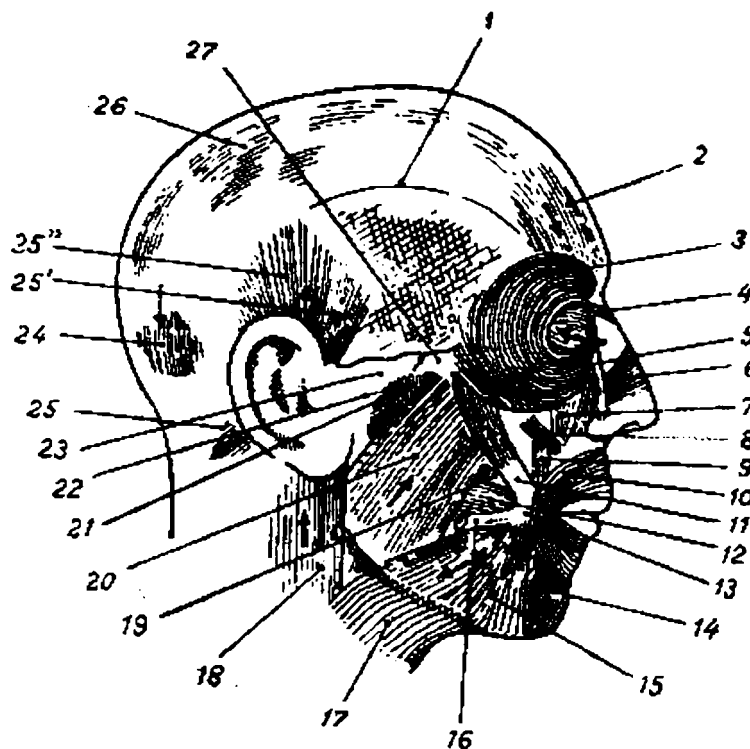


Fig.6-4 Mușchii feței și capului [19][24][25][186][187]

1-linia curbă temporală superioară; 2- mușchiul frontal; 3- m. Orbicular al pleoapei; 4-m. piramidal; 5- ridicător al buzei superioare și aripii nasului; 6- m. transvers al nasului; 7- m. ridicător al buzei superioare; 8- m. micul zigomatic; 9- m. ridicător al buzei; 10- m. orbicular al buzelor; 11- m. marele zigomatic; 12- m. buccinator; 13- modiolus; 14- m. pătrat al mentonului (depresorul buzei inferioare); 15- m. triunghiular al buzei inferioare; 16- m. rizorius; 17- m. pielos al gâtului; 18- m. sternocleidomastoidian; 19- canalul stenon; 20- m. maseter, fasciculul superficial; 21- m. maseter, fasciculul profund; 22- capsula articulară; 23- apofiza zigomatică a temporalului; 24- m. occipital; 25- m. auricular posterior; 25'- m. auricular anterior; 25''- m. auricular superior; 26- aponevroza epicraniană; 27- apofiza temporală a malarului.

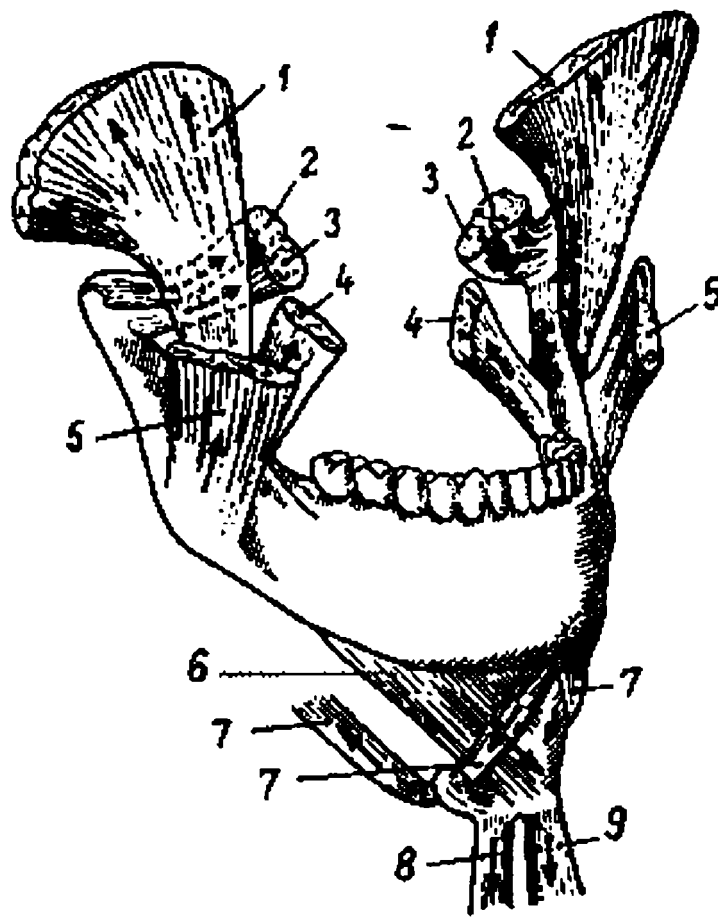


Fig.6-5 Musculatura mobilizatoare a mandibulei [19][24][25][186][187]

1-m. temporal (închidere, retropulsie); 2- fasciculul superior al m. pterigoidian extern (propulsie, deschidere, diducție); 3- fasciculul inferior al m. pterigoidian extern (propulsie asociată cu coborâre, mișcări diagonale); 4- m. pterigoidian intern (închidere); 5- m. maseter (închidere), fasciculul profund nu este reprezentat); 6- m. milohioidian (deschidere); 7- m. digastric (deschidere, retropulsie); 8- m. omohioidian; 9- m. sternocleidohioidian.

Un mușchi nu lucrează niciodată singur. Efectul mecanic, fie că este vorba de menținerea unei poziții de repaus sau de o deplasare, este rezultatul unei activități musculare complexe, în care se găsesc elemente protagoniste (cu acțiune în același sens), compensatorii sau complementare, precum și fenomene antagoniste.

Musculatura este în așa fel organizată în ceea ce privește conformația, dimensiunile, inserțiile, ca să corespundă cât mai bine din punct de vedere funcțional, pentru a se putea obține cu un minimum de efort, o eficiență cât mai mare. Mușchiul dă cel mai mare randament atunci când forța pe care o generează se aplică perpendicular pe

sarcina de deplasat. Deseori acest raport este obținut prin contracția concomitentă a mai multor mușchi, a căror rezultantă dirijează deplasarea.

Sistemul muscular este organizat pe principiul grupelor antagoniste, ceea ce oferă posibilitatea efectuării unor mișcări de amplitudini variate și asigurarea unui echilibru dinamic al elementului deplasat. Musculatura aparatului dento-maxilar oferă o foarte bună exemplificare a acestui principiu. La efectuarea unei mișcări mandibulare, de exemplu, intră în acțiune atât mușchii care o deplasează în sensul comenzii primite, cât și mușchii antagoniști, care-și reglează tonusul în urma comenzilor primite prin reflexele complementare antagoniste. De asemenea, se pot produce contracții simetrice sau asimetrice ale grupelor musculare omoloage.

Grupele antagoniste ale mușchilor mobilizatori ai mandibulei sunt:

a) În sens vertical:

1. mușchii ridicători: maseterul, pterigoidianul intern, fasciculul anterior și mijlociu al temporalului;
2. mușchii coborători: geniohioidianul, milohioidianul, pânțelele anterior al digastricului, pterigoidianul extern, în foarte mică măsură, mușchiul pielos al gâtului.

b) În sens sagital:

1. mușchii propulsori: pterigoidianul extern ajutat de pterigoidianul intern și fasciculul superficial al maseterului;
2. mușchii retropulsori: fasciculul posterior al temporalului, fasciculul profund al maseterului, geniohioidianul, pânțelele posterior al digastricului.

c) În plan transversal: mușchiul pterigoidian extern, în contracție unilaterală, ajutat parțial și de alți mușchi, care se contractă de aceeași parte cu el.

Se consideră [208][157][152][153][25][24][19] că menținerea în poziție echilibrată a craniului în vârful coloanei vertebrale, în sens antero-posterior, este asigurată de către două grupe musculare antagoniste:

- grupă posterioară alcătuită din mușchii cefei;
- grupă anterioară alcătuită din cele 3 etaje de mușchi: subhioidieni, supra hioidieni și mobilizatori ai mandibulei.

În plan transversal, craniul este menținut de către sternocleidomastoidian împreună cu mușchii precedenți cu care acționează simetric.

Asupra mandibulei acționează forțe active, care sunt generate de anumiți mușchi și realizează mișcarea sa, precum și forțe pasive (forțe rezistente, reacțiuni), care sunt date de greutatea proprie a mandibulei și de mușchii antagoniști. În figura 6-6 sunt prezentați principalii mușchi masticatori cu rol în dinamica la nivelul articulației temporomandibulare [19][24][25][186][187].

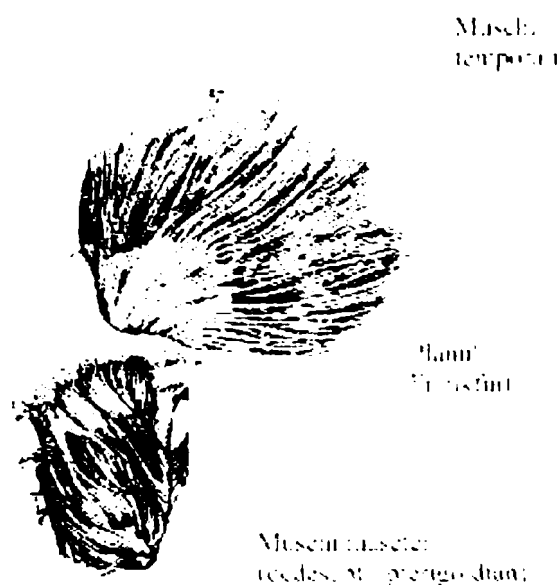


Fig.6-6 Principalii muschi masticatori cu rol în dinamica articulației temporomandibulare.

Forțele active care realizează ridicarea mandibulei și închiderea gurii sunt generate de către următorii mușchi (figura 6-6) ridicători (masticatori):

- Mușchii temporali, care trag mandibula în sus și înapoi (ridicare și retropulsie) sub un unghi de 60° față de planul orizontal Frankfurt, reușind să dezvolte maximum 700 N. În realitate, forța este mult mai mică decât valoarea maximă teoretică. Caracteristica dominantă a temporalilor este viteza de reacție. Dacă o forță mare este dezvoltată într-un timp foarte scurt, fenomenul este considerat ciocnire; percuția produsă în cadrul intervenției acestui mușchi este egală cu diferența impulsurilor și astfel, masa

mușchiului fiind constantă, rezultatul acțiunii sale este o variație mare a vitezei de mișcare;

- Mușchii pterigoidieni interni, care acționează pe direcția de 110° față de planul Frankfurt, dezvoltând forțe limitate superior la 400 N;
- Mușchii maseteri, înclinați la 97° față de planul Frankfurt, cu forțe de valori limitate superior la 800 N. Mușchii maseteri dezvoltă cele mai mari valori ale forței, fiind cei mai importanți mușchi masticatori.

Maseterii și pterigoidienii lucrează în tandem, pe fața exterioară, respectiv interioară a mandibulei (figura 6-7).

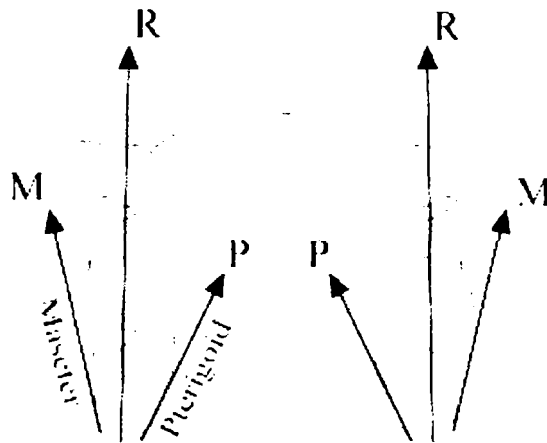


Fig.6-7 Tandemul maseter-pterigoid.

Rezultanta acestor 3 forțe acționează aproximativ perpendicular pe planul orizontal Frankfurt, modulul său atingând valori de până la 1800 N, cu valori uzuale de 100-200 N la nivelul incisivilor și 300-500 N la dinții laterali. Presiunea exercitată însă asupra incisivilor, este mai mare din cauza suprafețelor aflate în contact, care sunt mult mai mici.

Mărimea forțelor ocluzale poate fi calculată cu ajutorul *legii lui Weber* [177], care afirmă că forța dezvoltată de un mușchi este proporțională cu suprafața secțiunii sale transversale. Fiecare cm^2 de secțiune transversală a mușchiului uman poate dezvolta forțe de maximum 100 N. Valorile forțelor aferente celor 3 tipuri de mușchi, care au fost enunțate mai sus, verifică foarte bine această lege.

În timpul masticației, generatorul de forțe este aparatul muscular. Direcția principală verticală și sensul ei, în sus, sunt date de contracția mușchilor ridicători. Dinții și alimentele interpuse între arcade în timpul masticației, nu fac decât să divizeze forța, dându-i două sensuri: unul către maxilar și altul către mandibulă, care, sub forma de tracțiune, ajunge să se propage la nivelul oaselor respective și contribuie la structurarea funcțională a oaselor.

Șocurile mandibulare transmit o mare cantitate de forță la nivelele: maxilarului superior, oaselor faciale și chiar craniene. Deci, la nivelul oaselor maxilare se exercită forțe de tracțiune, a căror sursă este contracția mușchilor mobilizatori ai mandibulei. Și inserțiile directe osoase ale mușchilor exercită forțe de tracțiune care sunt generate tot de contracția musculară. Ele se transmit direct de la mușchi la os. Forțele de masticație, având tot o origine musculară, se transmit indirect prin intermediul sistemului dentar.

În procesul masticației, forțele au valori normale de 300-800 N, iar valorile maxime sunt de circa 3900 N la molari și 1500 N la incisivi.

Aceste exemple de forțe, indicate ca mărime, direcție de acțiune și orientare vor fi luate în considerare la modelarea numerică a implantelor osoase destinate scheletului cranian.

7. MODELAREA ȘI ANALIZA STRUCTURILOR MECANICE REPARATORII UTILIZATE ÎN CHIRURGIA MAXILO-FACIALĂ

7.1 Considerații asupra modelării structurilor mecanice și analiza lor cu elemente finite

Modelul de calcul trebuie să îndeplinească condiția de a fi adecvat scopului propus. În cazul în care se alege un model de calcul excesiv de complicat care își propune să aibă în vedere toate aspectele posibile ale fenomenului, acesta poate deveni costisitor, greoi sau chiar inoperant, iar un model simplist, prea sumar poate fi incorect ca urmare a neglijării unor fenomene și aspecte importante ale problemei.

După modul cum poate fi conceput un model de calcul, se disting următoarele tipuri de modele:

Modelul funcțional este un model structural teoretic sau experimental care pune în evidență diferitele componente ale fenomenului și ilustrează calitativ legăturile reciproce ale acestora din care să rezulte funcțiile globale fundamentale ale ansamblului.

Modelul de calcul este un model teoretic care pornind de la un ansamblu coerent de ipoteze, stabilește o schemă de calcul cu relații de calcul care descriu cantitativ și calitativ fenomenul. Trecerea modelului de calcul pe un calculator mărește considerabil eficiența acestuia.

Modelul experimental este un obiect fizic, un dispozitiv sau o instalație care reproduce în anumite condiții fenomenul care interesează.

Cele trei modele sunt complementare în descrierea unui fenomen, fiecare cu avantaje, dezavantaje și limitări. Pentru investigații complete se utilizează serii succesive de diverse modele. Pe măsură ce se culeg date despre fenomenul studiat, modelele se perfecționează, ceea ce nu înseamnă în mod neapărat complicarea lui ci poate fi foarte bine o simplificare a sa.

În vederea efectuării calculului de rezistență al unei structuri se elaborează un model de calcul. Acest lucru este posibil numai după ce s-au analizat condițiile generale și de principiu privind performanțele ansamblului, tehnologia, materialul, durata de viață, condițiile de lucru, etc.

Etapele de elaborare a unui model de calcul sunt următoarele [11][13][16][17][41][45][50][54]:

1. Stabilirea nivelului la care se face modelarea. Pentru o anumită fază a proiectării modelarea se poate face la nivelul întregii structuri, la nivelul substructurilor (ansamble) sau al componentelor acestora (elemente). Extragerea unor substructuri din structura inițială se poate face în etape ulterioare pentru un studiu amănunțit al unor zone de importanță mai mare ale structurii.

2. Alegerea metodei de calcul. Structura care se calculează trebuie să satisfacă cerințele de funcționalitate, siguranță și economicitate. De remarcat că metodele deterministe de calcul iau în considerație o siguranță absolută care acoperă toate incertitudinile printr-un coeficient de siguranță. Metodele moderne de calcul țin cont de caracterul aleatoriu al parametrilor care intervin în ansamblul structurii. În acest caz siguranța structurii este măsurată prin probabilitățile diferitelor comportări posibile pe o durată de timp dată.

3. Modelarea geometriei structurii. Aceasta constă în aproximarea structurii reale printr-un sistem echivalent de corpuri, suprafețe, etc. Modelul trebuie să reproducă cât mai fidel dimensiunile, rigiditățile și deplasările impuse structurii.

4. Determinarea sarcinilor sau încărcărilor și a caracterului acestora. Aceasta se realizează prin determinări experimentale sau prin calcule, pe baza datelor și parametrilor de exploatare cunoscuți sau impuși. Cunoașterea completă a sarcinilor presupune evidențierea următoarelor aspecte:

- cunoașterea modului de variație în timp.
- regimul de exploatare.
- analiza cazurilor de încărcare cu precizarea celor mai dezavantajoase situații.

5. Schematizarea distribuției sarcinilor sau schematizarea încărcării. Solicitarea structurii poate fi produsă de un sistem de sarcini concentrate sau distribuite după o anumită lege.

Faptul ca structurile mecanice sunt deformabile conduce in mod firesc la afirmatia că deformarea lor va creea în mod inevitabil, o altă lege de distribuție a sarcinilor diferită de distribuția initial acceptată. În foarte multe situații însă, întrucât determinarea distribuției reale a sarcinilor este foarte dificil de realizat, se apelează la schematizări, ceea ce înseamnă implicit acceptarea unor ipoteze simplificatoare ce conservă esența încărcării reale dar prezintă avantajul că simplifică considerabil volumul și dificultatea calculelor ulterioare.

Alegerea celei mai potrivite variante de calcul reprezintă în final de fapt o chestiune de experiență și ingeniozitate. Modelul de calcul trebuie să urmărească soluția constructivă, desi pentru soluții constructive similare există întotdeauna mai multe variante de modele ce conduc la rezultate diferite. Când există îndoieli în privința modelului de calcul, se va adopta acea schematizare care conduce la un calcul acoperitor.

6. Alegerea metodei de calcul. Aceasta operatie trebuie făcută în strictă dependență cu modelul de calcul căruia urmează să i se aplice. Un calcul oricât de exact nu poate compensa o modelare grosieră cum nici o modelare exactă, nu poate compensa un calcul aproximativ. Eficiența calculului crește considerabil prin armonizarea metodei de calcul cu cea a modelării.

7. Definitivarea soluției și perfecționarea modelului. Se pot face numai în măsura în care s-a trecut la interpretarea primelor rezultate obținute din calcule. Perfecționarea modelului se poate îmbunătăți prin continuarea procesului iterativ astfel început.

8. Interpretarea și prelucrarea datelor finale. Reprezintă o etapă tot atât de laborioasă în multe cazuri ca și crearea și elaborarea modelului de calcul în sine. Prezentarea rezultatelor sub forma unor liste de valori reprezintă o etapă necesară dar nu și suficientă pentru desprinderea unor concluzii privitoare la fenomenul studiat. Trebuie remarcat faptul că utilizarea unor calculatoare tot mai performante conduce la obținerea de multe ori a unui volum impresionant de date de ieșire. Se impune în acest context elaborarea și utilizarea unor programe de calcul specializate în post procesarea datelor de ieșire.

Sunt foarte multe situații în care volumul datelor de intrare și de ieșire se pot reduce considerabil prin utilizarea proprietăților de simetrie și antisimetrie ale structurilor. Se are în vedere pe de o parte simetria fizico-geometrică a structurilor în

raport cu un plan sau cu mai multe planuri, iar pe de altă parte, se are în vedere simetria sau antisimetria sarcinii. Ca o consecință a acestor proprietăți eforturile prezintă proprietăți similare de simetrie sau antisimetrice. Ca urmare a proprietăților de simetrie și antisimetrie unele eforturi / deplasări sunt nule sau egale între ele, formând perechi în planul de simetrie.

Este cunoscut faptul că **metoda analizei cu elemente finite** a devenit un instrument de lucru cu o largă utilizare în toate domeniile ingineresti. Principalul avantaj al acestei metode este acela de a fi înlăturat bariera legată de nivelul de complexitate al problemei studiate. Pot fi studiate sub aspect teoretic probleme oricât de complexe, cu luarea în considerație atât a fenomenelor liniare cât și al celor neliniare. În același timp MEF permite analize în regim dinamic, în regim termic, analize ale curgerilor de fluide sau studii ale câmpurilor electromagnetice, etc.

7.2 Metoda analizei cu elemente finite

Noțiunea de element finit a apărut prima dată într-o lucrare a lui R.W. Clough în anul 1960 intitulată “Elementul finit în analiza stărilor plane de tensiune”. Se poate spune că acesta a fost semnalul apariției și dezvoltării unei noi științe ingineresti și anume știința elementelor finite. Dezvoltarea metodei elementului finit după această dată a fost în strânsă dependență cu ritmul de dezvoltare și perfecționare al tehnicii de calcul. Limitele de aplicare ale metodei elementului finit sunt date de fapt de nivelul de dezvoltare al tehnicii de calcul.

Metoda analizei cu elemente finite a apărut ca o consecință a necesității de a calcula structuri de rezistență complexe pentru care metodele analitice de calcul nu sunt operabile. Ideea de bază este aceea că în cazul în care structura se împarte în mai multe părți numite “elemente finite” pentru fiecare dintre acestea se pot aplica teoriile de calcul corespunzătoare schematizării adoptate. Împărțirea întregului în părți de dimensiuni mai mici, operație care poartă denumirea consacrată de *discretizare* va avea drept efect obținerea de forme simple pentru elementele finite componente ale structurii. Modelul utilizat în analiza cu elemente finite este un model aproximativ, obținut prin ansamblarea elementelor finite componente, ținând cont de geometria structurii. Conectarea

elementelor finite se realizează numai în anumite puncte numite puncte nodale sau “noduri”. Nodurile reprezintă punctele de intersecție ale liniilor de contur rectilinii sau curbe ale elementelor finite. Elementele finite pot fi unidimensionale, bidimensionale sau tridimensionale în funcție de geometria structurii pe care o modelează [68][69][72][76][78][88][89][94][142].

Caracterul aproximativ al metodei elementului finit rezultă ca urmare a faptului că geometria reală este întodeauna înlocuită cu o rețea de elemente finite care urmărește forma reală, dar nu o poate reda cu exactitate, datorită numărului finit de elemente, iar mărimile necunoscute ale probleme sunt calculate numai în nodurile structurii. Rezultă de aici concluzia că precizia de calcul crește odată cu creșterea numărului de elemente finite. Pentru calculul marimilor necunoscute ale problemei pentru alte puncte, în afara nodurilor, pe domeniul de întregului element finit, se folosesc diferite funcții de aproximare. Trebuie subliniat faptul că în cazul metodei elementului finit este legată este legată foarte strâns de performanța de aproximare al acestor funcții. De asemenea continuitatea rezultatelor obținute depinde deasemenea de caracterul de continuitate pe care funcțiile de aproximare trebuie să-l asigure la nivelul zonelor de interelemente.

Metoda elementelor finite s-a aplicat pentru prima dată în calculul structurilor de rezistență. Datorită caracterului de generalitate al acestei metode, ea s-a extins cu rapiditate în aproape toate domeniile calculului ingineresc care au la bază metodele fizico-matematice de calcul. Formularea metodei elementului finit se bazează pe formularea condițiilor extreme pe care unele mărimi care intervin în fenomenul studiat trebuie să le satisfacă. Este deci o metodă numerică cu un vast domeniu de aplicabilitate care se bucură de avantajul unei formulări simple. Caracterul de generalitate al metodei îi conferă avantajul de a se adapta cu modificări simple celor mai complexe și variate probleme cum sunt problemele liniare și neliniare, solicitări statistice și dinamice structuri de bare, plăci plane sau curbe și masive, solicitări de contact, probleme de mecanica ruperii, etc.

7.3 Considerații asupra mecanismului de producere a fracturilor și a structurii de rezistență a craniului

Leziunile traumatiche ale masivului facial se produc ca rezultat al impactului dintre efector și receptor.

Leziunile rezultate prin acțiunea agentului traumatizant depind de receptor prin doi factori: părțile moi ale feței și scheletul osos, care recepționează diferit energia cinetică a agentului traumatizant.

- **Părțile moi ale feței.** Deși acoperă în strat subțire scheletul osos, acestea absorb o mare parte din energia cinetică a agentului traumatizat. Experimental [1][36][53][75][97] s-a aratat că părțile moi pe un craniu al unei persoane vii, cresc rezistența scheletului la agentul traumatizant cu 30%, în raport cu un craniu de cadavru de pe care tesuturile moi au fost îndepărtate.

Mușchii craniului fiind în mare majoritate pieloși, nu dezvoltă forta suficientă pentru deplasarea fragmentelor osoase. Singurii mușchi de forță sunt pterigoidianul intern inserat cu capătul cranial în foseta pterigoidă și maseterul, inserat pe arcada zigomatică, corpul malar și apofiza piramidală a maxilarului. Amândoi pot provoca deplasarea secundară a întregului bloc osos prin tracțiune în jos și posterior, doar când linia de fractură trece deasupra inserțiilor musculare superioare. Masa musculară poate însă proteja structurile osoase ale masivului facial în măsura în care agentul traumatizant o traversează. Ea poate absorbi o parte din energia pe care o primește.

- **Scheletul osos** recepționează diferit energia cinetică a agentului traumatizant. Absorbția, distribuția și rezistența scheletului la forța de impact sunt dependente de duritatea suportului osos și structura arhitecturală a întregului masiv osos. Duritatea suportului osos este dată de fasciculele de os compact distribuite neuniform în coloane verticale și orizontale de grosimi diferite ce realizează stâlpii și grinzile de rezistență ale masivului facial. Acestea sunt capabile să suporte forțe care exercită în sens longitudinal și nu perpendicular pe acest ax. Între stâlpii și grinzile de rezistență, pereții, plafoanele și planșeele sunt constituite din os lamelar subțire, nerezistent, ce orientează de regulă liniile de fractură.

Oasele feței sunt cu mult mai fragile decât alte oase ale scheletului uman. Studii experimentale [177][107][120] au arătat că acestea se fracturează la o forță variind între 100 și 500 daN la o energie de șoc de la 6 la 60 J. O forță mai mare de 800 daN poate fi suportată doar până la 4 ms. În ordine crescândă structurile etajului mijlociu al feței pot fi lezate cu energii ce depășesc anumite limite.

Cu o energie cinetică de 2-5 J sunt lezate părțile moi; de 6-30 J: se fracturează zonele fragile ale masivului facial (nas, dinți, os alveolar, arcade zigomatice); de 45-55 J (corpul malar și masiv facial respectând sau nu liniile de slabă rezistență).

7.4 Interventii chirurgicale reparatorii la nivel osos cu ajutorul plăcuțelor metalice și implantelor

La nivelul oaselor feței implantele se realizează din titan pur, aliaje de titan sau materiale plastice speciale. Un implant încearcă să suplinească lipsa osului într-o anumită porțiune. Forma unui implant diferă în funcție de cazul specific, ideal fiind ca implantul să reproducă întocmai forma porțiunii de os căreia îi ține locul, de obicei neregulată. Acest lucru este dificil, pe de o parte datorită complexității realizării unei copii fidele a porțiunii de os care lipsește. Pe de altă parte, de cele mai multe ori fracturile trebuie corectate în regim de urgență, când este esențial ca medicul chirurg să aiba deja implantul care se va utiliza.

De aceea, în chirurgia reparatorie maxilo-facială se folosesc foarte des plăcuțele metalice de forma regulată care se fixează prin suruburi de restul osului. Ele pot fi suporturi pentru osteosinteza fiind scoase după ce osul s-a refăcut sau pot rămâne permanent fixate pe os.

Grație unui contact strâns și de durată între fragmentele menținute aproape sub presiune, calusul osos se formează rapid scurtând cu mult perioada de vindecare. Plăcuțele permit și interpunerea, respectiv menținerea transplantelor osoase pentru reconstrucție tridimensională imediată, care se impune în fracturile cu pierdere de substanță. Este cazul mai ales al pilierilor sau grinzilor care mențin dimensiunile verticală, anteroposterioară și transversală a etajului mijlociu [2][7][9][15][27][28][38][39][58][61][62][63][105][111].

Plăcuțele (figura 7-1) oferă o excelentă stabilitate tridimensională fără risc de recidivă până la formarea calusului osos. De asemenea, ele constituie un bun suport pentru transplantele osoase interfragmentare în reconstrucția fracturilor cu pierdere de substanță.

Zonele de aplicare a șuruburilor pentru aplicarea plăcuțelor miniaturizate le constituie grinzile sau stâlpii de rezistență în care osul compact este mai abundent și grosimea osoasă mai mare.

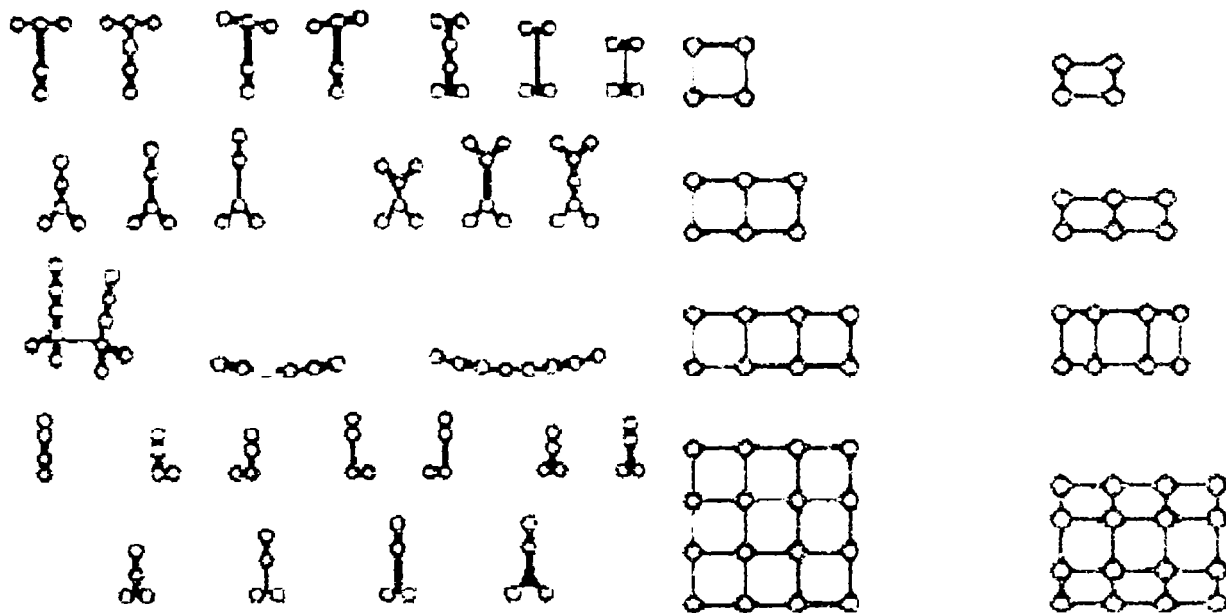


Fig.7-1 Tipuri de placute metalice

7.5. Modelarea 3D a implanturilor pornind de la modelul reconstruit al structurii osoase

Modelarea părților care lipsesc dintr-o structura osoasă este posibilă în anumite condiții și se poate realiza chiar cu metoda de reconstrucție descrisă în teză (capitolul 5), dar în doi pași.

Prin utilizarea aceluiași principiu ca la modelarea suprafețelor pline sau convexe, sistemul modelează corpuri ce sunt de fapt complementele unei concavități precizate, dând astfel naștere unei forme ce se potrivește exact în locul în care lipsește țesut osos. Această formă poate fi utilizată pentru realizarea unui implant din titan.

Datorită faptului ca părțile care lipsesc, evident, nu sunt evidențiate în imaginea tomografică, este necesară o prelucrare manuală sau semiautomată a planelor ce delimitează zona cu defecțiune. Preprocesarea presupune o segmentare și o identificare/izolare a zonei în care ar trebui de fapt să se afle țesut osos. Această identificare se face în fiecare plan imagine ce delimitează zona. După identificare, pixelilor din această zonă li se alocă valoarea maximă (255, în cazul imaginilor pe 8 biți). În acest fel, la aplicarea mecanismului de detecție cu un prag maxim, este detectată doar porțiunea care interesează, nu și restul osului.

În figura 7-2, după obținerea modelului 3D al craniului pacientului investigat se constată în partea stângă a mandibulei o porțiune de os lipsă.



Fig.7-2 Porțiunea de os care lipsește va trebui implantată

Rezultatul (figura 7-3) este o formă ce se va putea aplica drept implant, în locul dorit:



Fig.7-3 Modelul 3D al implantului.

Implantul se va potrivi în locul unde osul lipsește deoarece forma lui a fost construită ținându-se seama de suprafețele învecinate, însă reușita depinde foarte mult de acuratețea cu care se face delimitare zonei de interes. Acest tip de implant osos este actualmente fezabil prin prototipare din material plastic cu proprietăți apropiate de cele ale țesutului osos.

7.6 ANALIZA CU ELEMENTE FINITE A STRUCTURILOR UTILIZATE CU SCOP RECONSTRUCTIV

7.6.1 Fractura de mandibula în plan vertical

În cazul fracturilor de mandibulă (figura 7-4) porțiunea de os fracturată se imobilizează cu o plăcuță de titan.

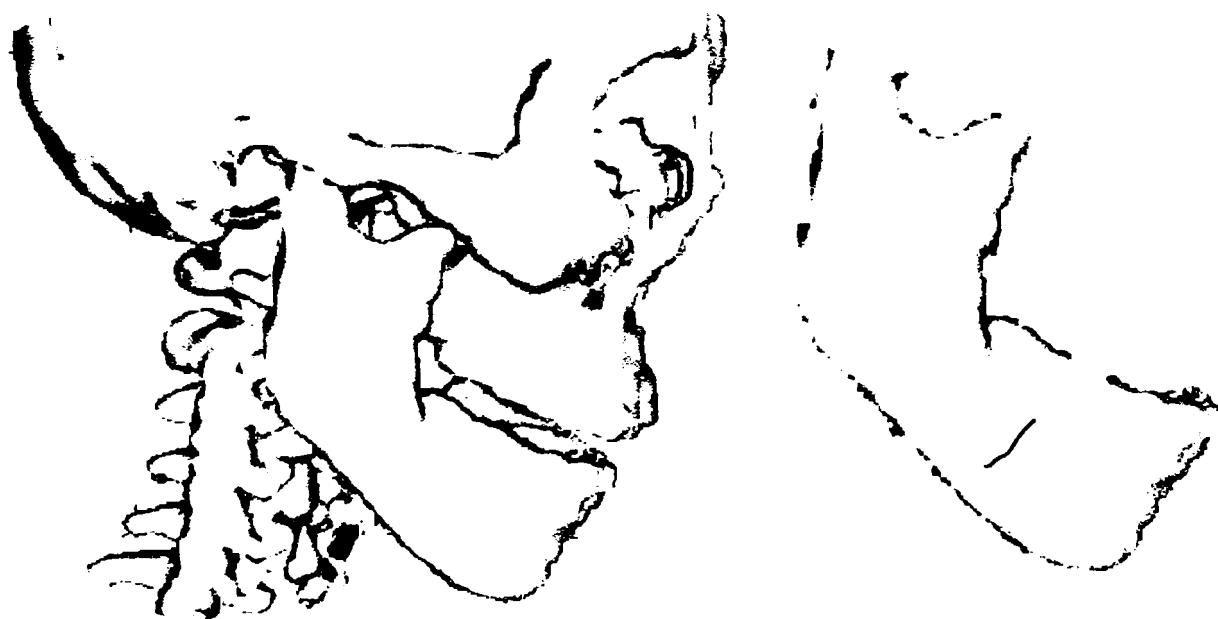


Fig.7-4 Craniu: vedere laterală și osul mandibular fracturat (se observă linia de fractură)

Aceste plăcuțe pot fi de diferite tipuri, și în general ele se aplică pe os fixarea realizându-se cu șuruburi, astfel încât fiecare capăt al plăcuței va fi de o parte respectiv de cealaltă a liniei de fractură (figura 7-5).

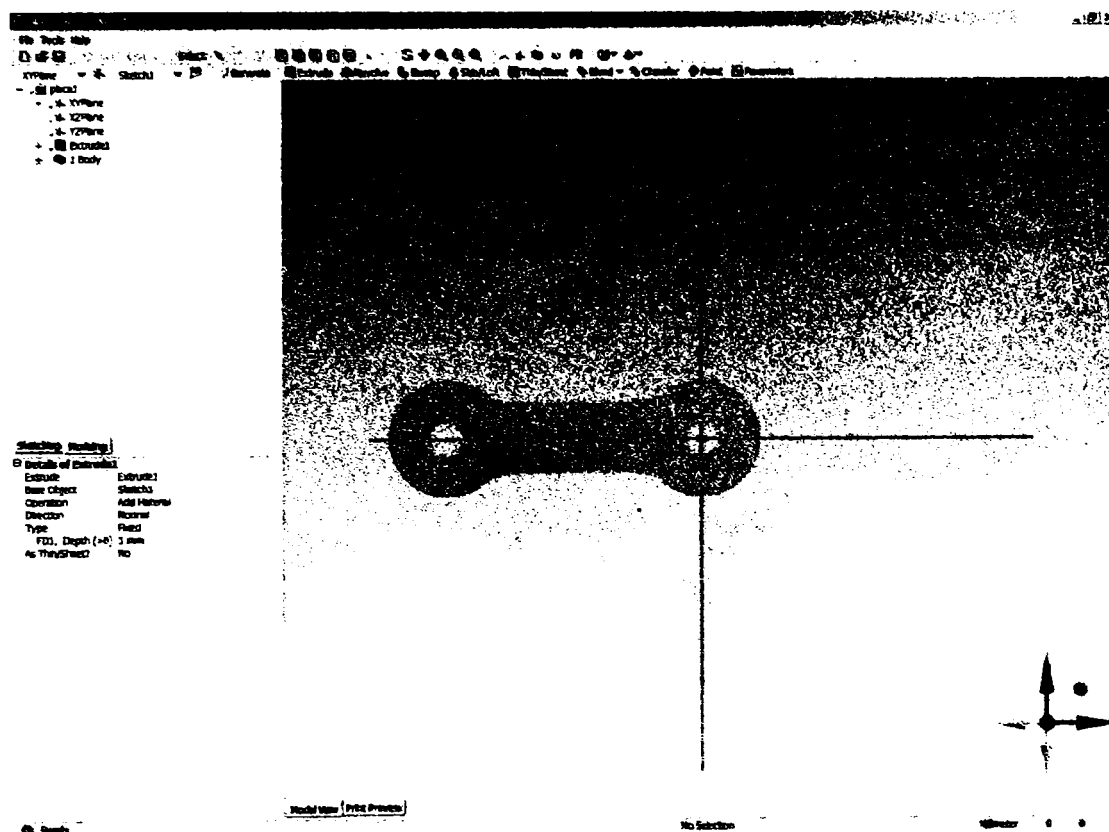


Fig.7-7 Proiectarea formei plăcuței

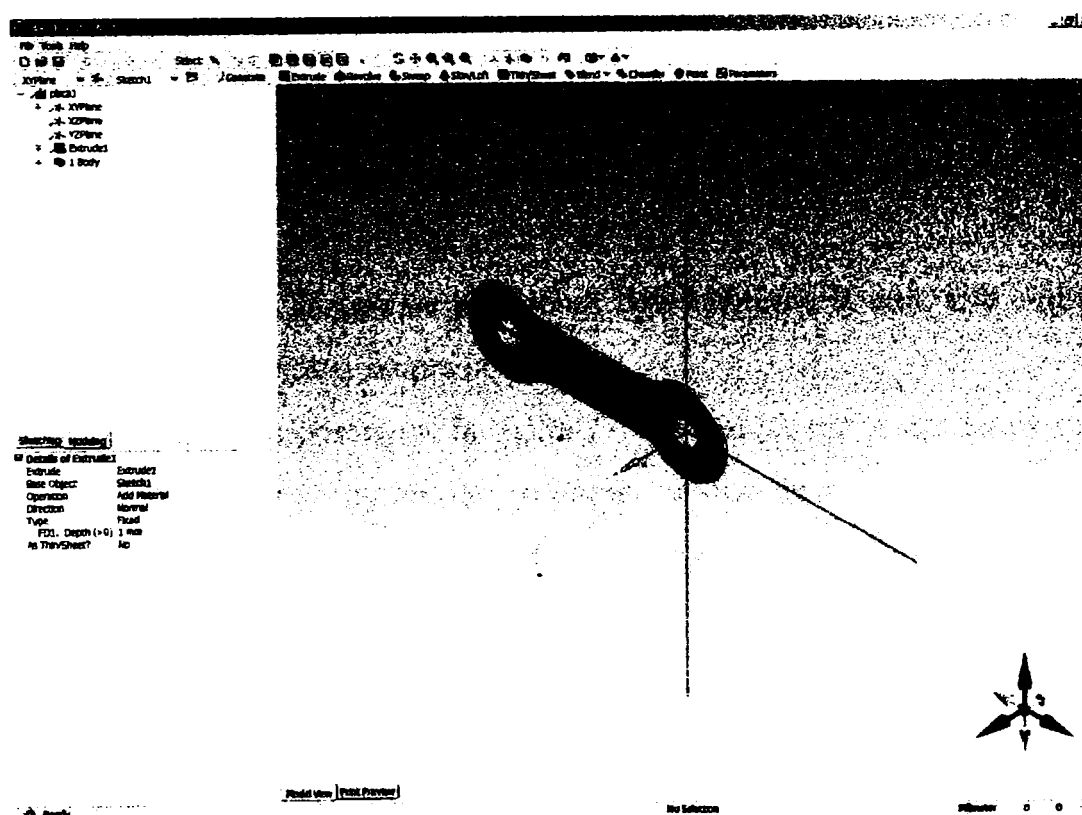


Fig.7-8 Forma finală a plăcuței

2. A doua etapă presupune generarea nodurilor și a elementelor finite. În cazul plăcuței analizate s-au obținut 2284 de noduri și 1070 de elemente (figura 7-9).

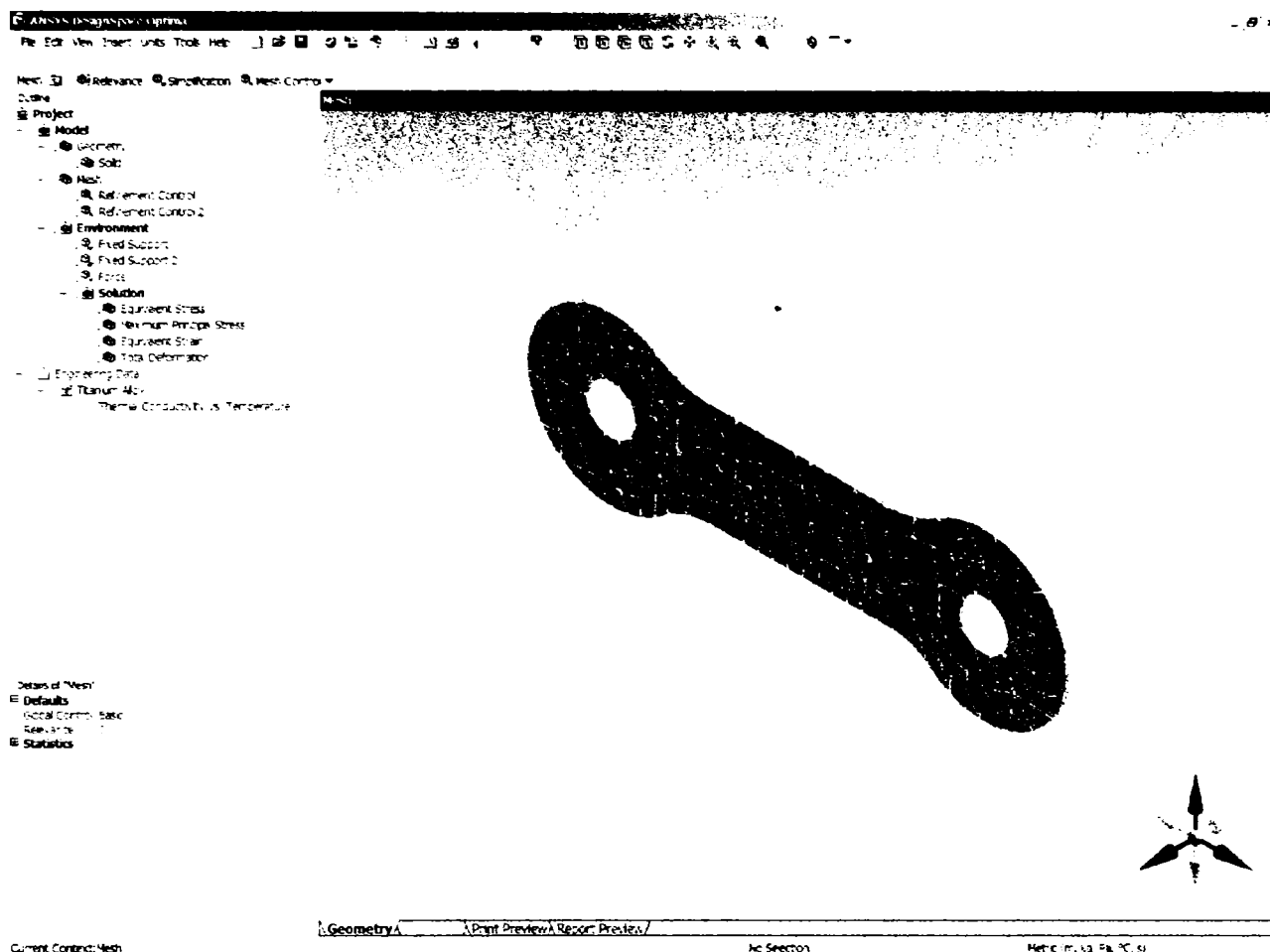


Fig. 7-9 Generarea rețelei de noduri și elemente finite

3. Etapa a treia este cea în care se precizează constrângerile la care este supusă placa, respectiv forțele ce acționează asupra ei.

Așa cum se poate observa și în figura 7-10, deoarece placa este fixată în șuruburi pe os, în locul de prindere a șuruburilor se impun condiții de suport fix. Pe de altă parte, deoarece forțele dezvoltate în timpul masticăției, de exemplu, apar de-a lungul liniei de fractură, se impune aplicarea unei forțe în planul plăcuței. Valoarea aleasă este de 600 N, fiind o valoare maximală pentru procesul de masticăție, și aproape de maximal pentru cazurile excepționale ale unor lovituri.

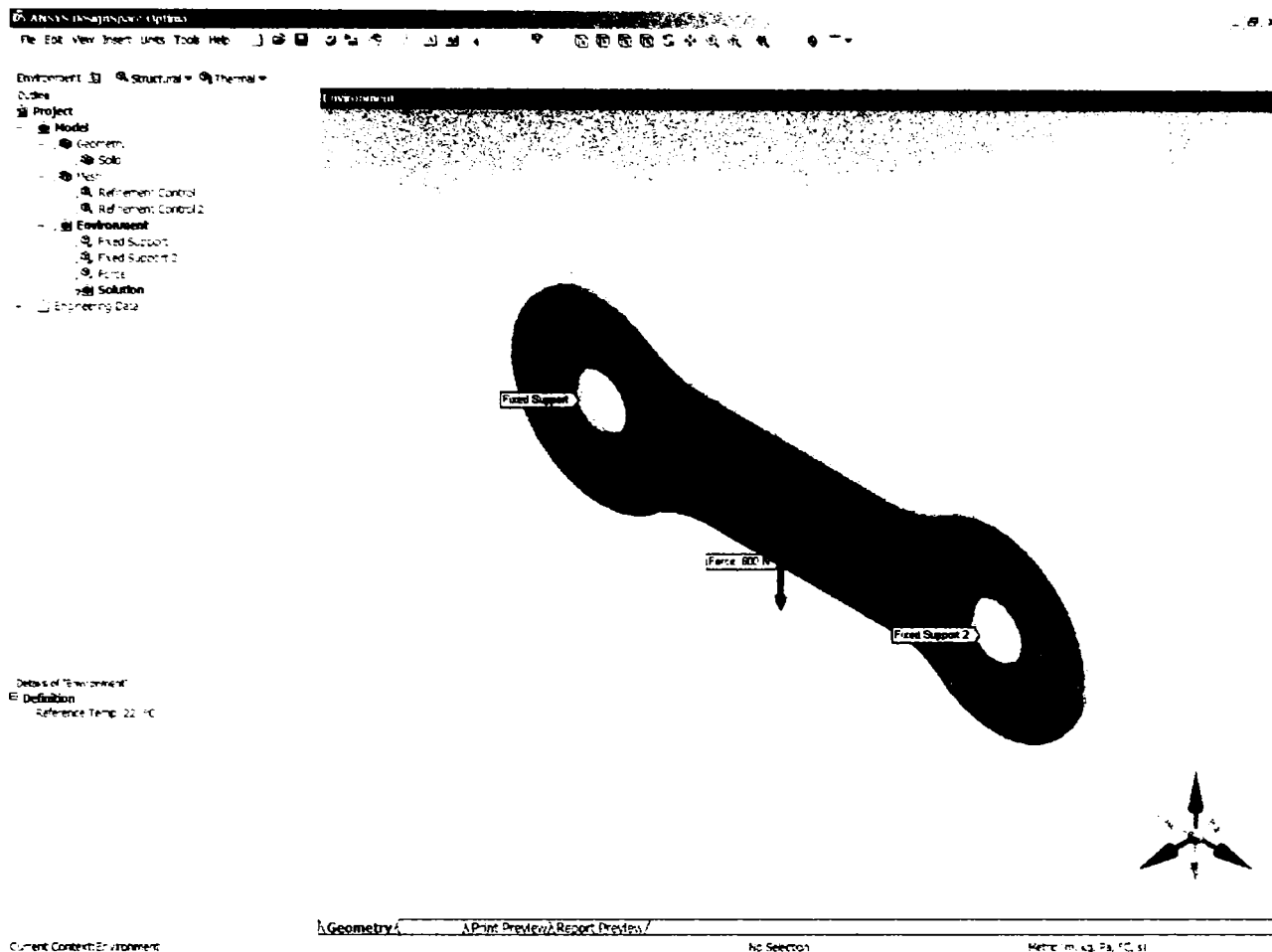


Fig.7-10 Impunerea constrângerilor și aplicarea unei forțe rezultante de 600 N

Materialul din care este confecționată placa este aliaj de titan, ale carui caracteristici sunt următoarele:

- modulul de elasticitate este de $1 \cdot 10^5 \text{ N/mm}^2$ sau $1 \cdot 10^{11} \text{ Pa}$
- rezistența la rupere (stress) este de 1200 N/mm^2 adică $1200 \cdot 10^6 \text{ Pa}$
- deformarea relativă maximă este de 6%

În urma aplicării condițiilor limită, asupra corpului acționează o forță descrisă de vectorul:

$$[2.83 \times 10^{-14} \text{ N } x, -600.0 \text{ N } y, 0.0 \text{ N } z]$$

precum și două forțe de reacție în punctele de fixare de 288.09 N respectiv 312.75 N descrise de vectorii:

$$[-15.86 \text{ N } x, 287.65 \text{ N } y, 2.21 \times 10^{-3} \text{ N } z]$$

$$[15.86 \text{ N } x, 312.35 \text{ N } y, -2.21 \times 10^{-3} \text{ N } z]$$

4. Ultima etapă este cea a analizei cu elemente finite (fig.7-11, 7-12, 7-13, 7-14).

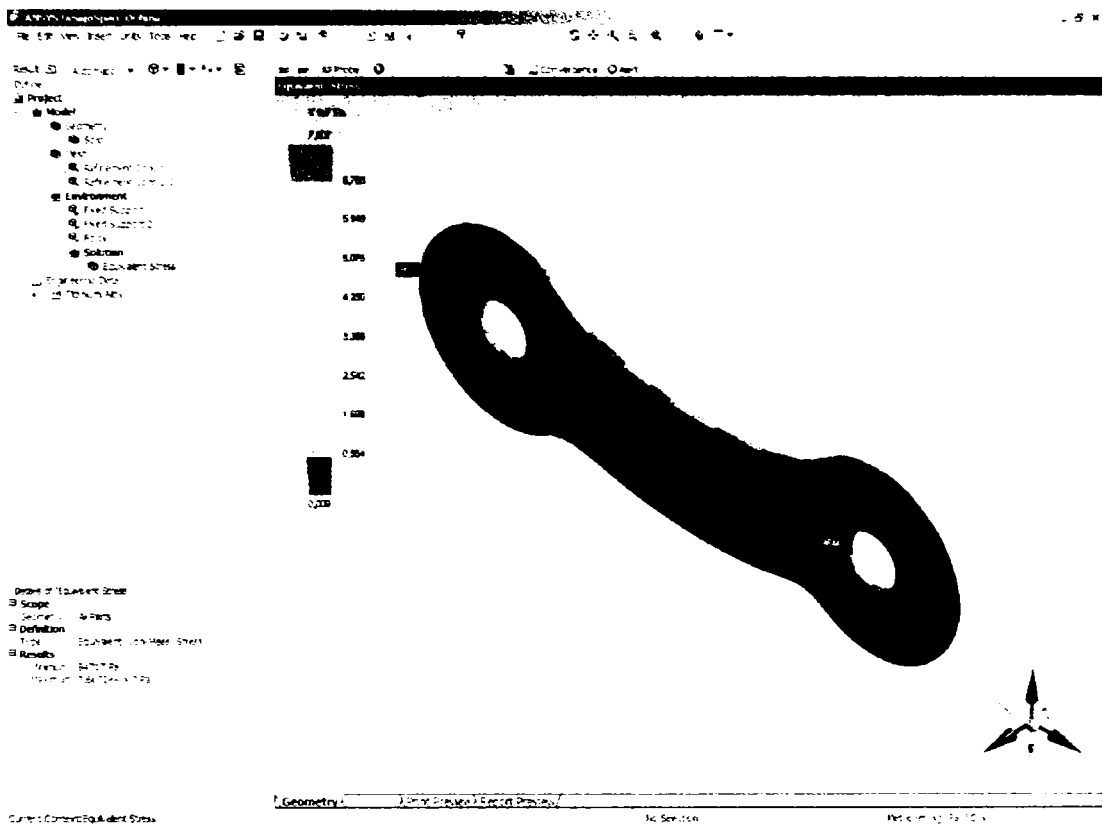


Fig.7-11 Rezultatul analizei cu elemente finite (tensiuni echivalente)

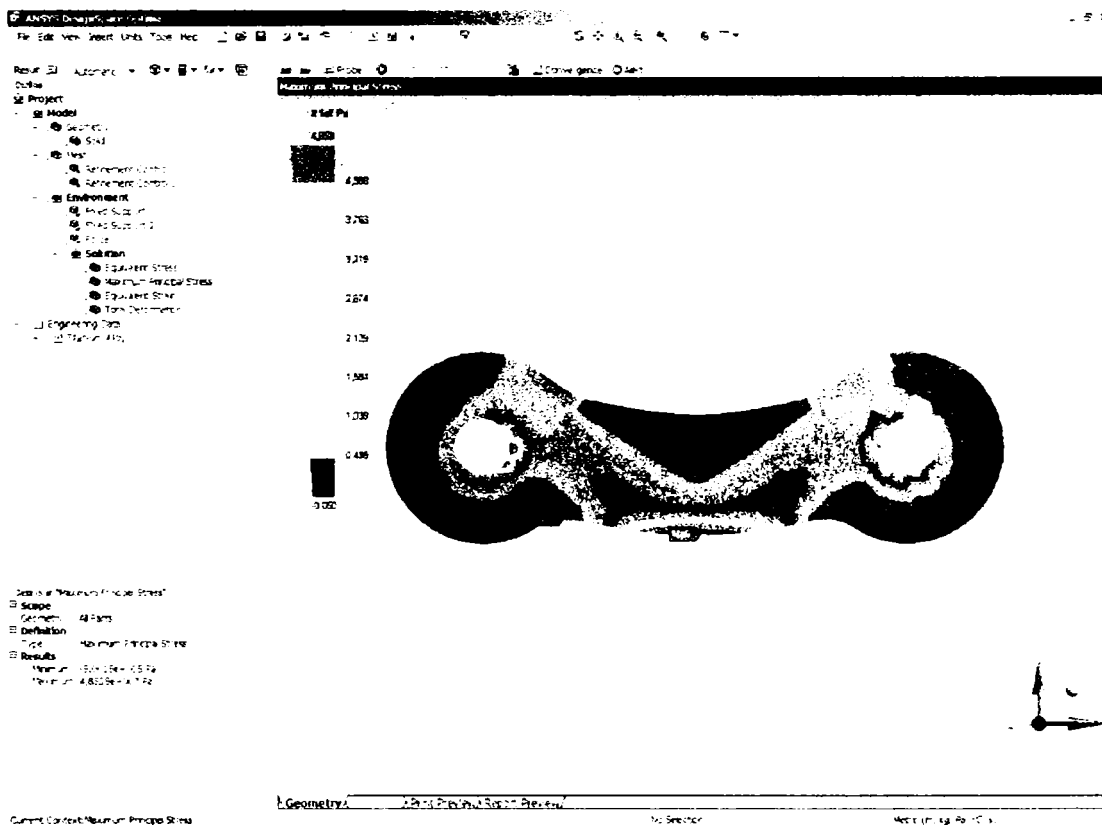


Fig.7-12 Rezultatul analizei cu elemente finite (tensiunea maximă principală)

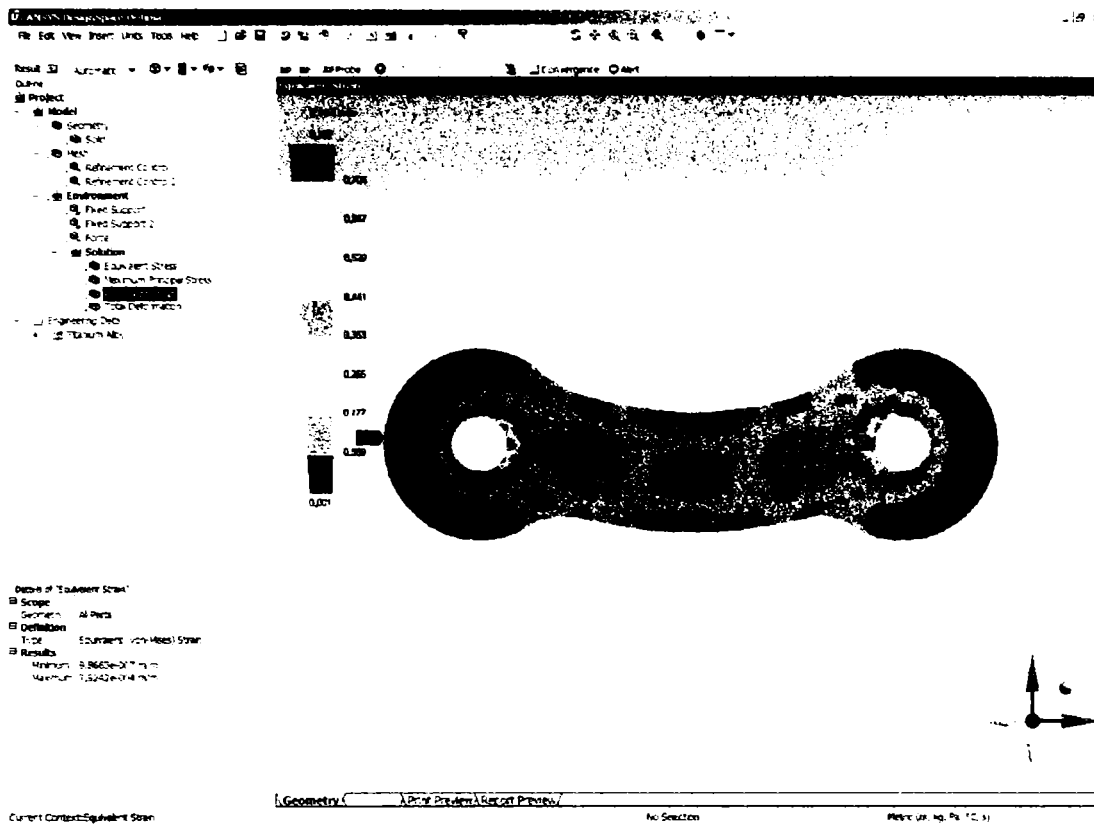


Fig.7-13 Rezultatul analizei cu elemente finite (deformarea relativă echivalentă)



Fig.7-14 Rezultatul analizei cu elemente finite (deformarea totală)

În urma analizei cu elemente finite a acestei plăcuțe de titan se obțin următoarele rezultate:

- tensiunea echivalentă minimă este 94,716.96 Pa, iar cea maximă 7.61×10^7 Pa
- tensiunea maximă principală este cuprinsă între -500,150.22 Pa și 4.85×10^7 Pa
- deformarea relativă echivalentă este între 9.87×10^{-7} m/m și 7.92×10^{-4} m/m
- deformarea totală minimă este 0.0 m iar cea maximă de 2.54×10^{-5} m

Concluzia este că în condițiile analizate, de utilizare normală placa se comportă bine, fixând puternic osul, neexistând deformări esențiale care să afecteze fractura.

7.6.2 Imobilizarea unei fracturi cu două plăcuțe diferite

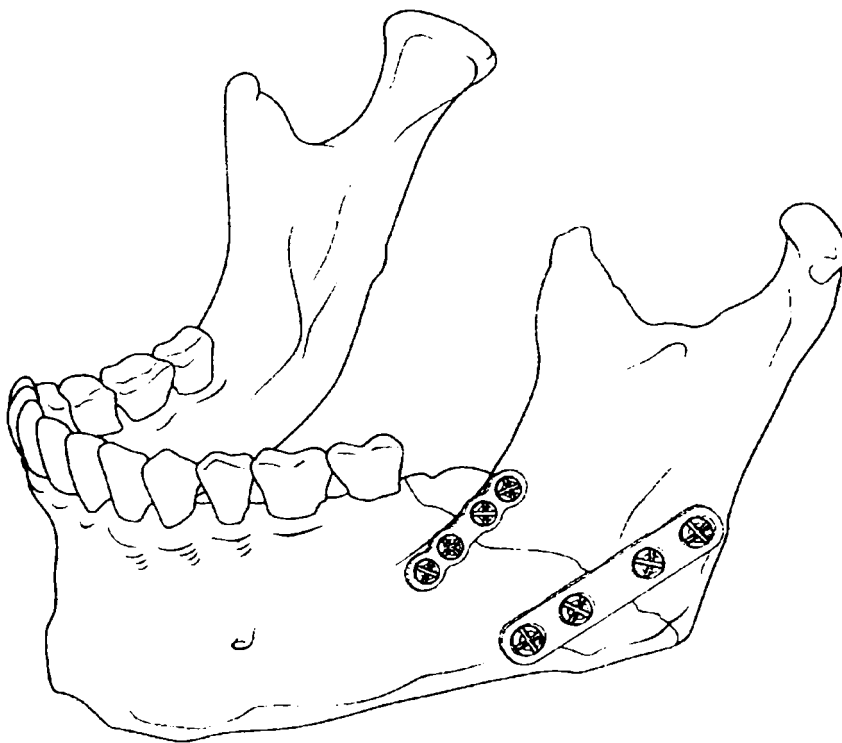


Fig.7-15 Fractură imobilizată cu 2 tipuri diferite de plăcuțe

În partea superioară a liniei de fractură este fixată o miniplăcuță din titan, iar în partea inferioară, pentru mărirea rezistenței la șocuri și pentru că osul permite, fiind mai gros, s-a apelat la o placă LC-DC de fixare. Este de remarcat că aceasta nu este perpendiculară pe linia de fractură deoarece condițiile de prindere pe os nu permit întotdeauna acest lucru.

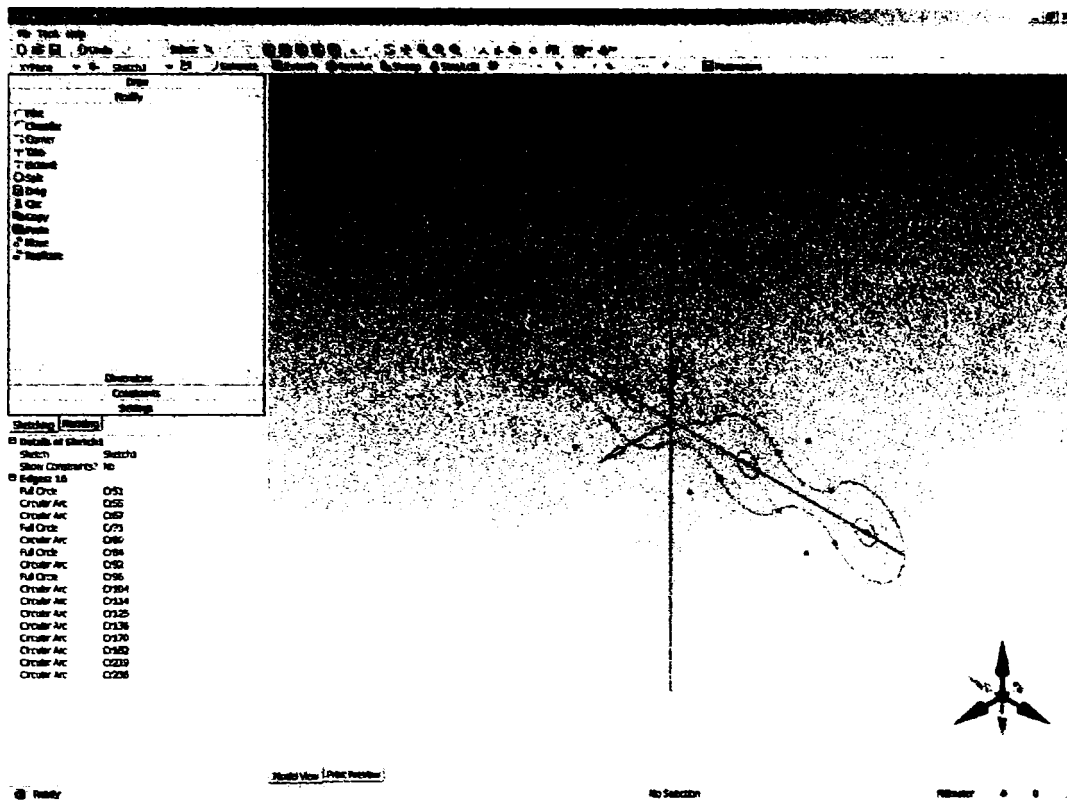


Fig.7-16 Forma plăcuței superioare

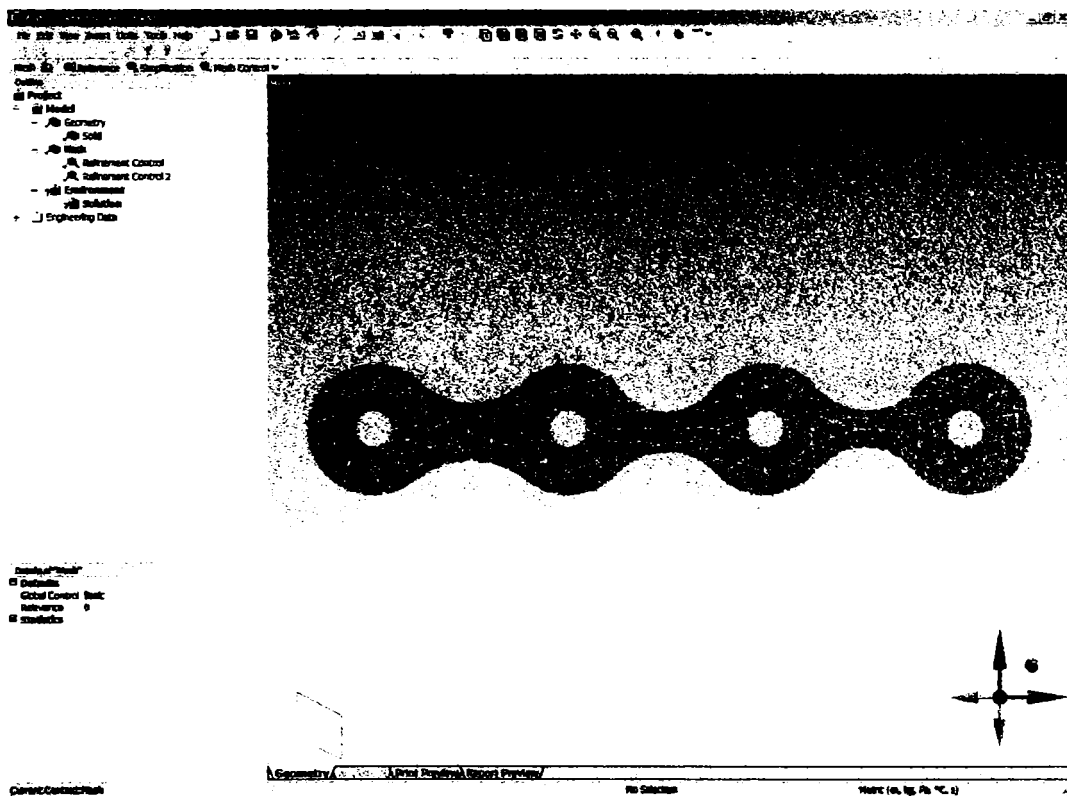


Fig.7-17 Plăcuța este descompusă în 3000 de noduri și 1329 elemente

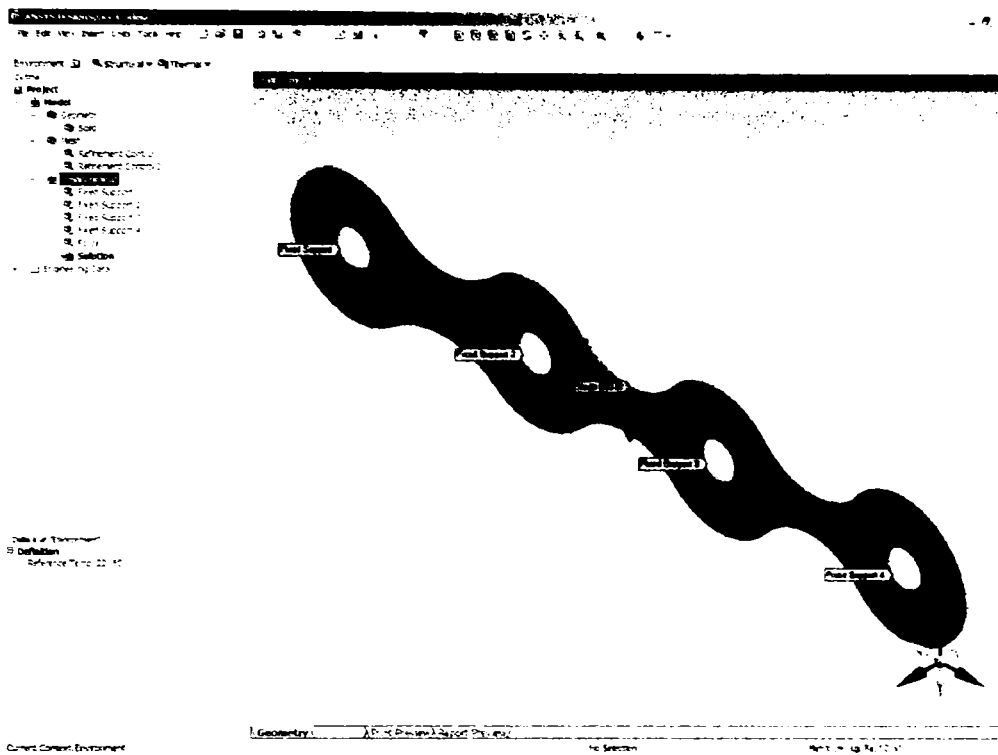


Fig.7-18 Impunerea constrângerilor în locul de fixare prin șuruburi, respectiv aplicarea forței în planul plăcuței, pe direcția liniei de fractură

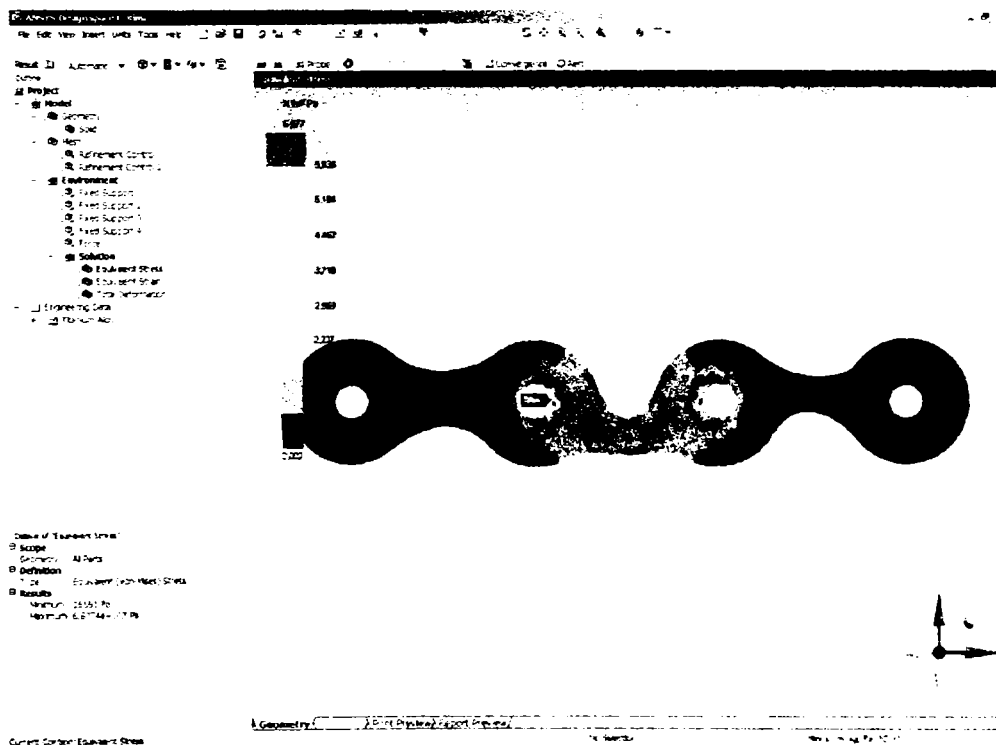


Fig.7-19 Distribuția tensiunilor

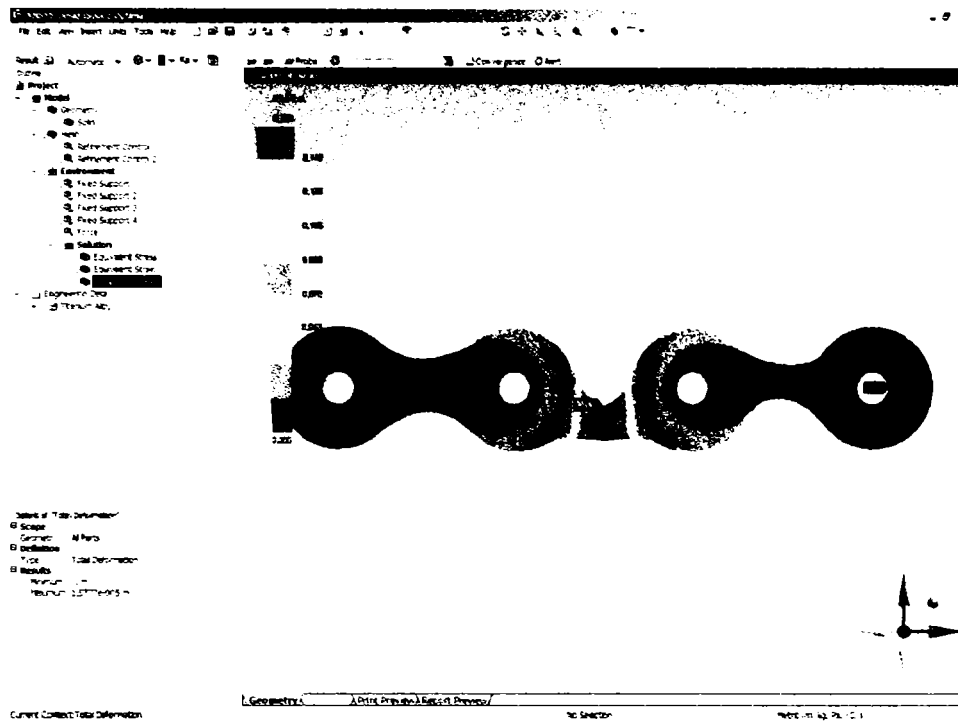


Fig.7-20 Deformarea totală maximă

Miniplăcuța fixează bine partea superioară a fracturii însă nu are rezistență la forțe maxime aplicate asupra sa. De aceea este utilă montarea plăcii LC-DC cu rol în mărirea rezistenței.

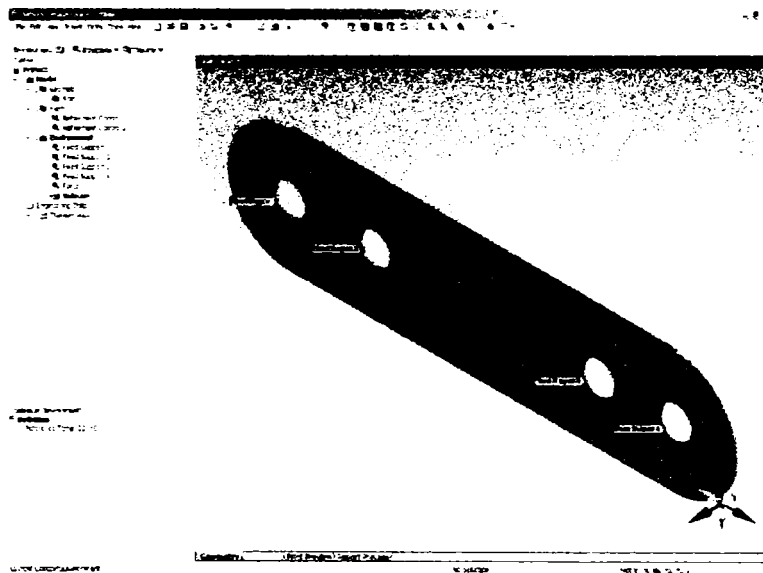


Fig.7-21 Placă de rezistență

În condiții de suprasolicitare această placă conferă o rezistență sporită ansamblului de imobilizare a fracturii (figurile 7-22 și 7-23)



Fig.7-22 Distribuția tensiunilor în cazul plăcii LC-DC

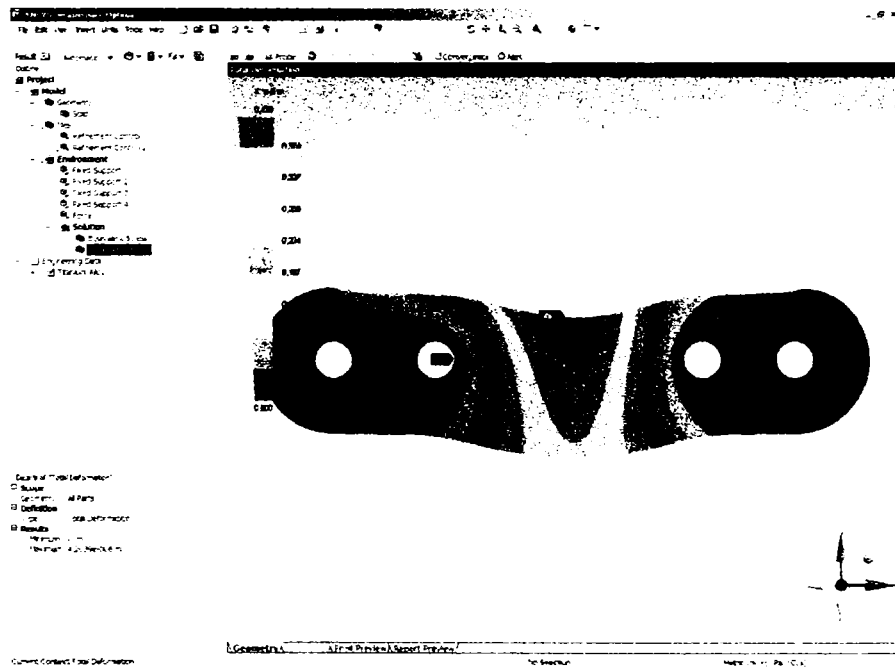


Fig.7-23 Deformarea totală a plăcii LC-DC

7.6.3 Fractură cominutivă

În cazul acestor fracturi, în care fracturile sunt mai mici dar multiple se poate utiliza o placă de fixare care poate imobiliza simultan mai multe fragmente osoase de-a lungul mai multor linii de fractură.

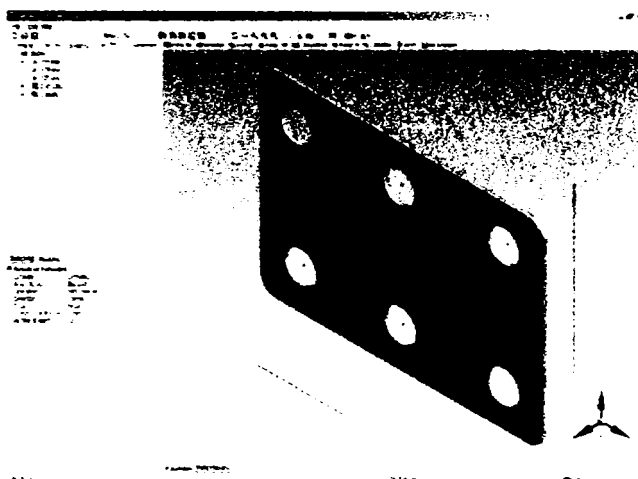


Fig.7-24 Placă pentru imobilizarea fracturilor multiple

În cazul a două linii de fractură este dificilă montarea plăcii perpendicular pe liniile de fractură, mai ales că liniile de fractură pot avea forme neregulate. Din acest motiv, forțele ce sunt aplicate în planul plăcii pot avea componente pe direcțiile celor 3 axe.

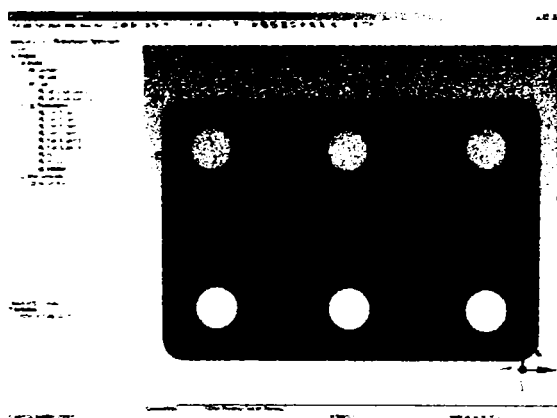


Fig.7-25 Constrângerile sunt aplicate la locul de fixare în șuruburi, iar rezultantele forțelor sunt orientate de-a lungul liniilor de fractură, existând și componente de-a lungul axelor.

Rezultatele analizei indică tensiuni doar în locurile de prindere în șuruburi, în restul plăcii acestea fiind ne semnificative. Deformarea este practic inexistentă, și acest lucru se datorează atât formei plăcuței cât și forțelor mici care intervin în astfel de cazuri.

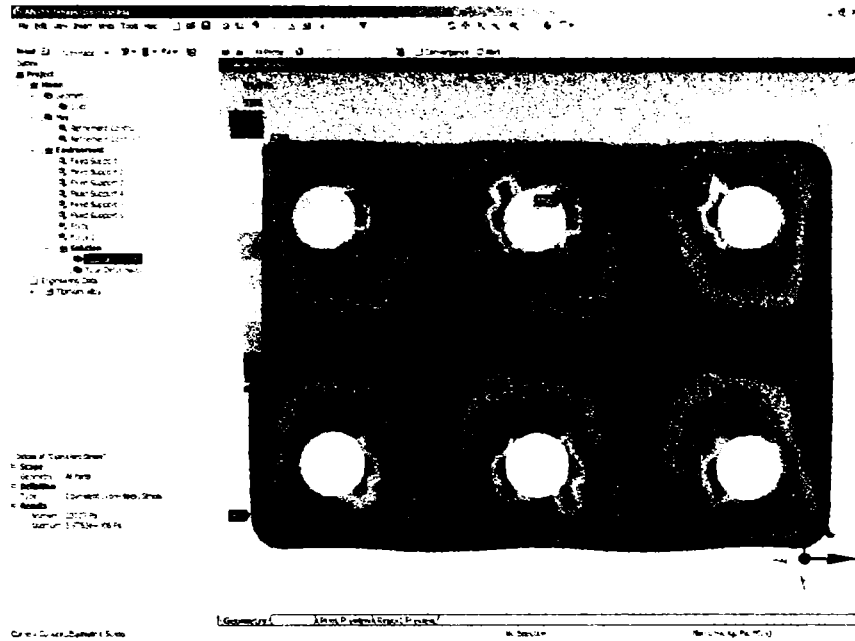


Fig.7-26 Distribuția tensiunilor în corpul plăcuței

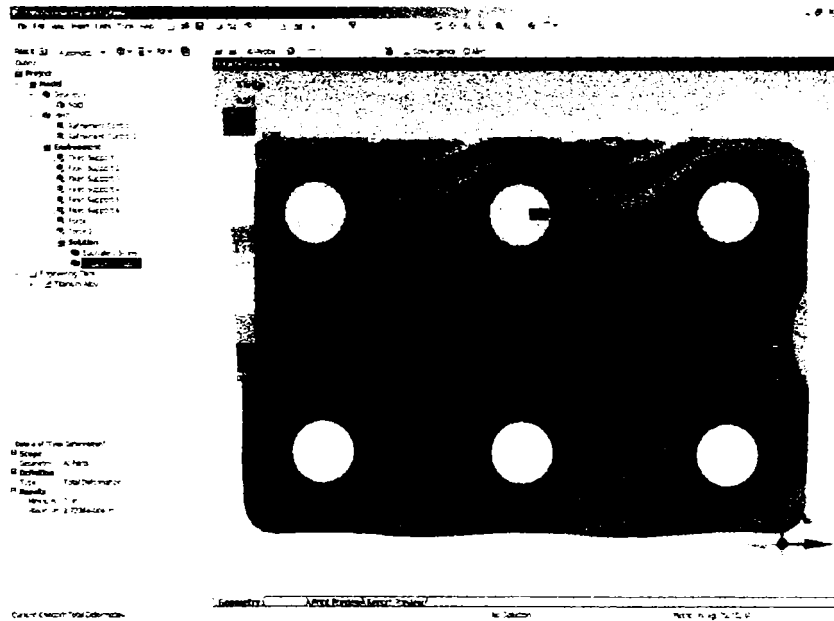


Fig.7-27 Deformarea totală

7.6.4 Utilizarea miniplăcuței 3D

În cazul unor fracturi de-a lungul mai multor linii, se impune utilizarea unei plăcuțe de fixare speciale, de lungime suficientă să acopere zona fracturată, urmarea fiind ca placa va trebui să aibă aceeași curbura ca mandibula. În cazul acestor tipuri de fracturi forțele ce intervin de-a lungul liniilor de fractură sunt mai mici, valorile ajungând până la 400 N.

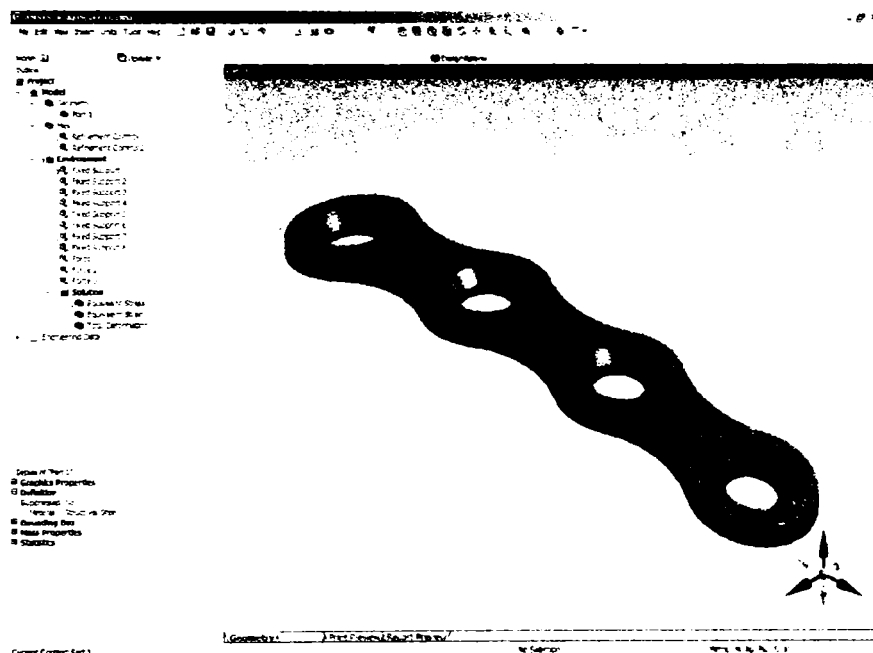


Fig.7-28 Forma miniplăcuței curbate

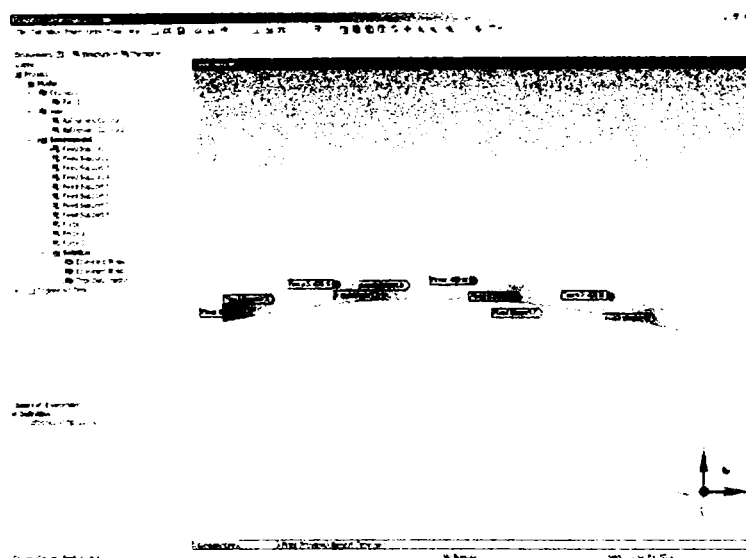


Fig.7-29 Forma miniplăcuței curbate. Constrângerile sunt impuse în punctele de fixare

Suprafața interioară a curbei va fi tangență la suprafața osoasă a mandibulei. Placa este fixată în șuruburi, deci se impun constrângeri în locurile de fixare. Forțele ce acționează în planul plăcii sunt de-a lungul celor 3 linii de fractură iar analiza este realizată pentru valoarea maximală de 400 N.

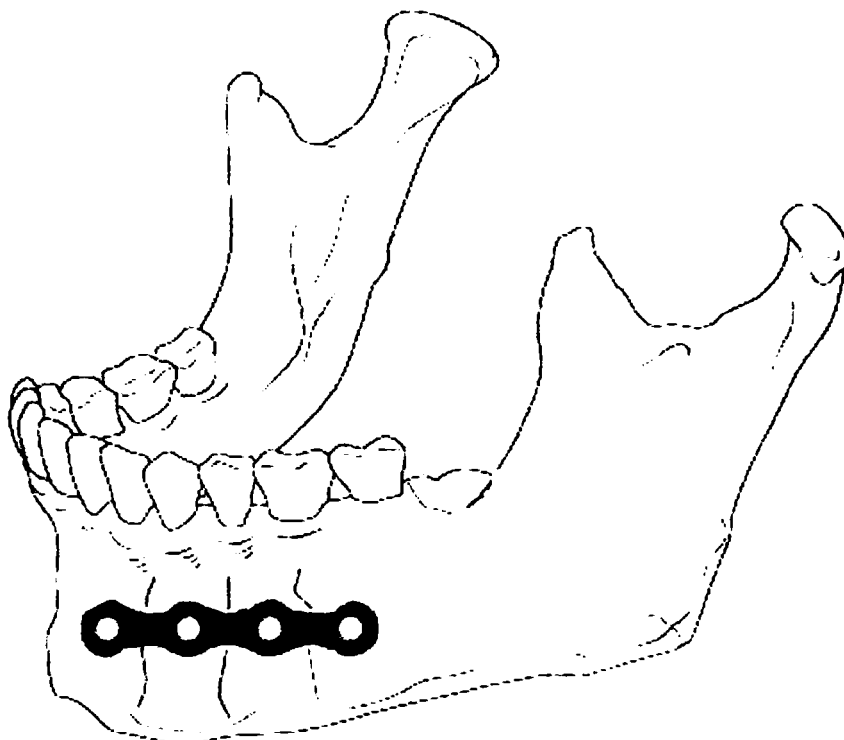


Fig.7-30 Placa imobilizează mandibula de-a lungul a 3 linii de fractură

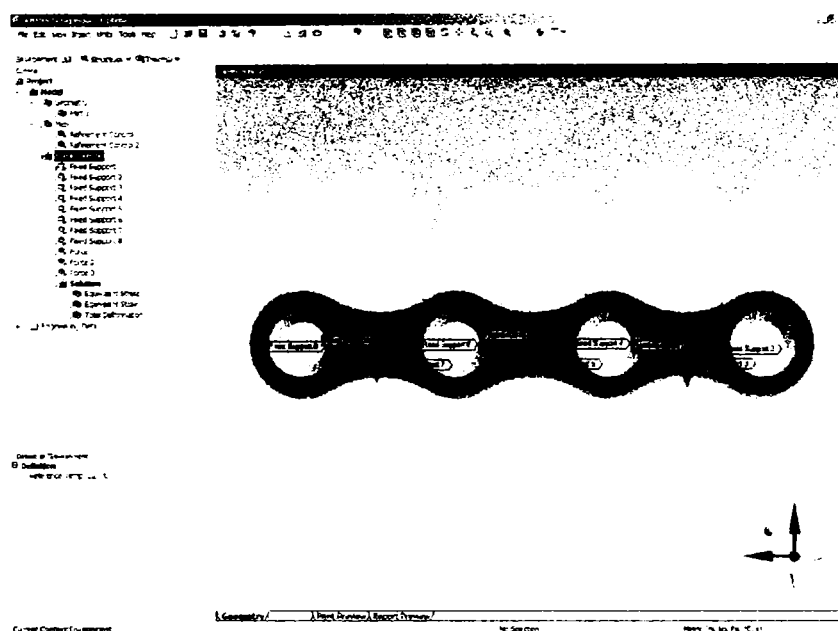


Fig.7-31 Constrângerile impuse și forțele ce acționează

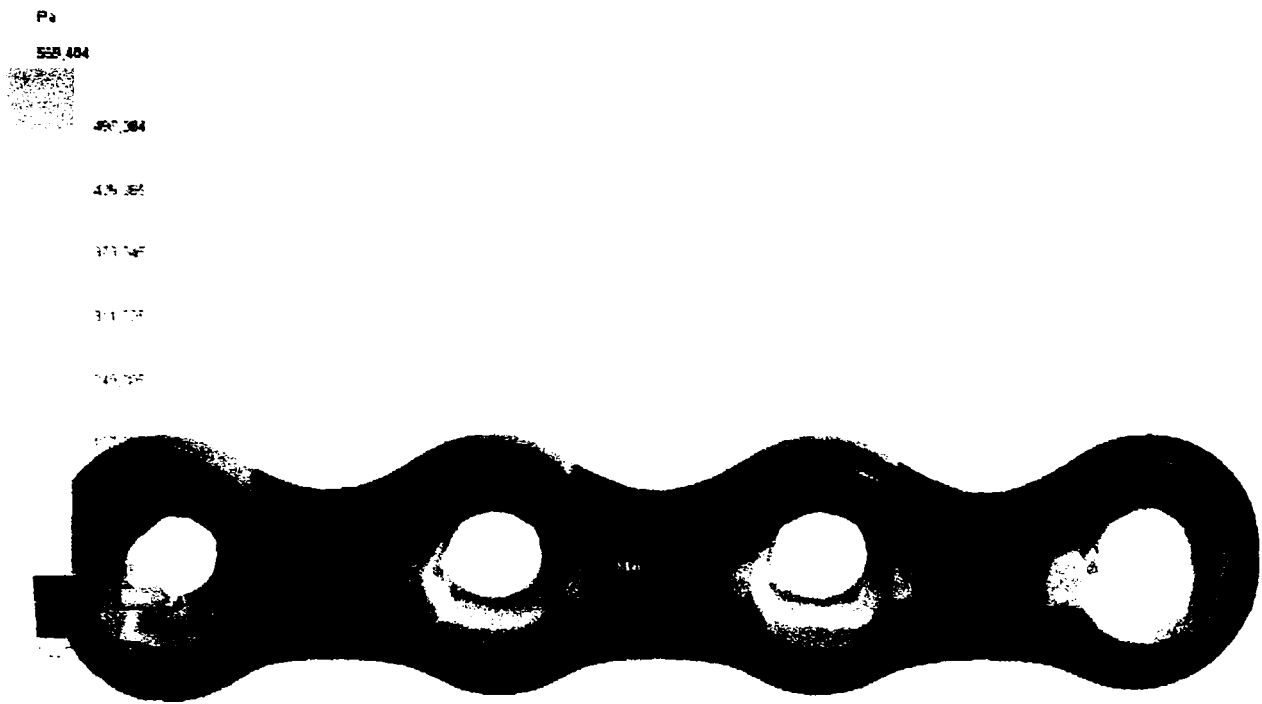


Fig.7-32 Tensiunea maximă este de 559.4 Pa, ceea ce atestă comportarea foarte bună a plăcii

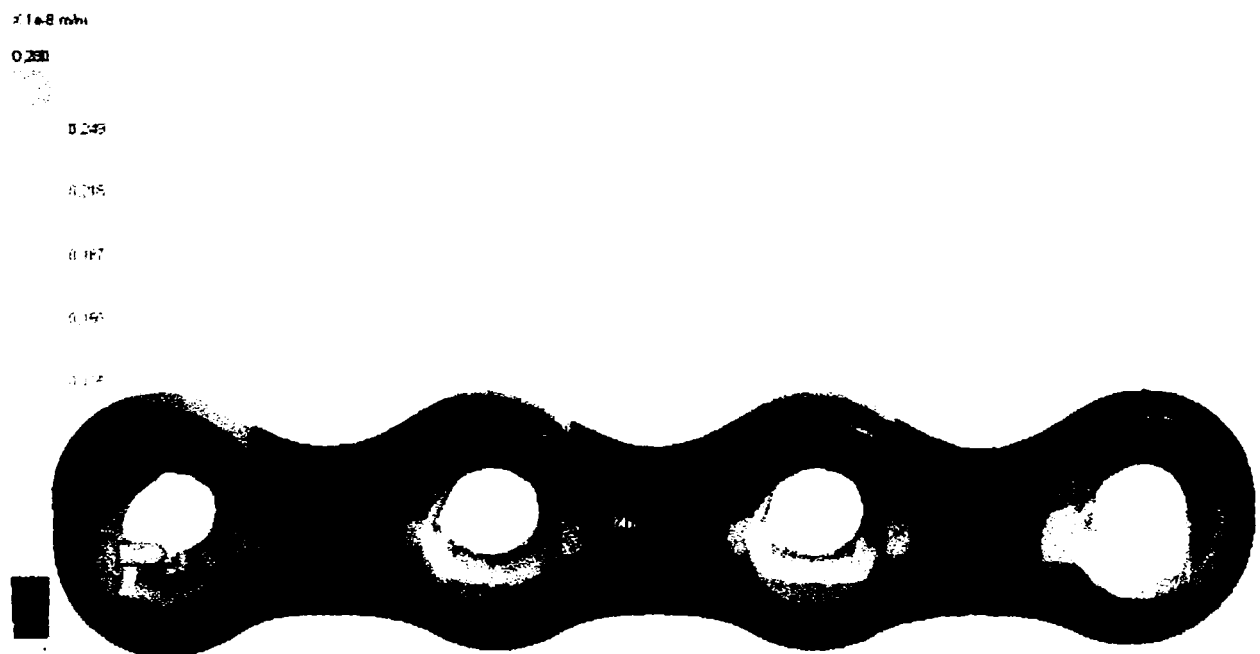


Fig.7-33 Deformarea relativă echivalentă, este de 2.8×10^{-9} m/m



Fig.7-34 Deformarea totală maximă este de 5.13×10^{-9} m

Constrângerile sunt descrise de vectorii:

- [0.0 N x, 0.0 N y, 0.0 N z]
- [1.74 N x, 2.08 N y, 78.77 N z]
- [-4.48 N x, -2.16 N y, 186.5 N z]
- [1.01 N x, -2.03 N y, 188.19 N z]
- [-0.87 N x, -0.23 N y, 170.99 N z]
- [-0.79 N x, -4.82 N y, 172.3 N z]
- [0.3 N x, -1.92 N y, 173.72 N z]
- [0.51 N x, 0.17 N y, 270.23 N z]

Rezultatul analizei prin metoda elementelor finite arată ca acest tip de placă este foarte rezistent, deformările fiind practic nesemnificative, tensiunile maxime apărând în punctele de fixare prin șuruburi, placa fiind indicată în cazul fracturilor multiple.

7.6.5 Cazul liniilor de fractură neparallele

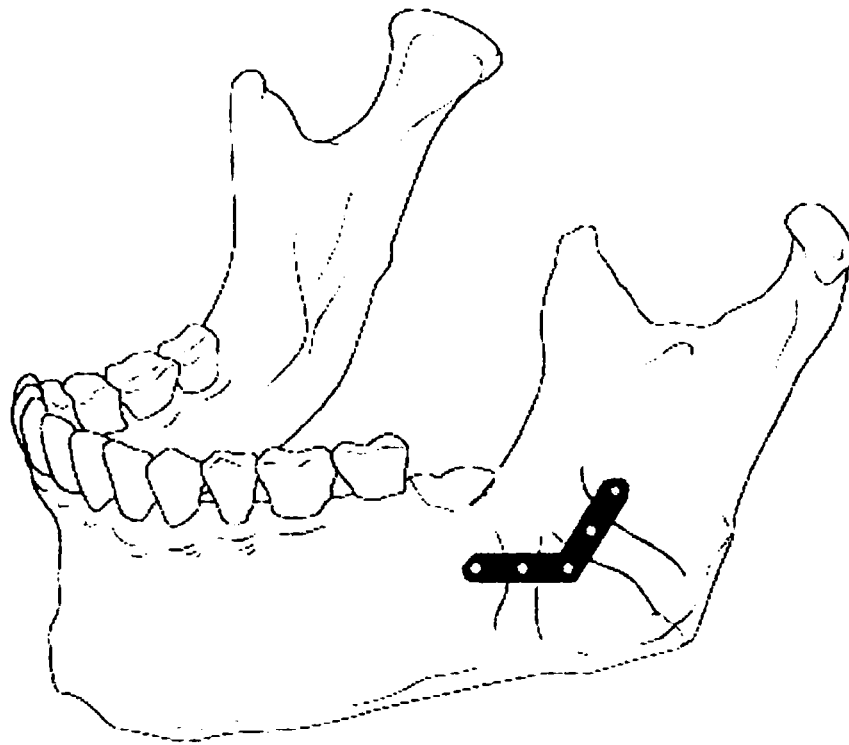


Fig.7-35 Placă L pentru imobilizarea fracturilor de mandibulă cu linii de fractură neparallele

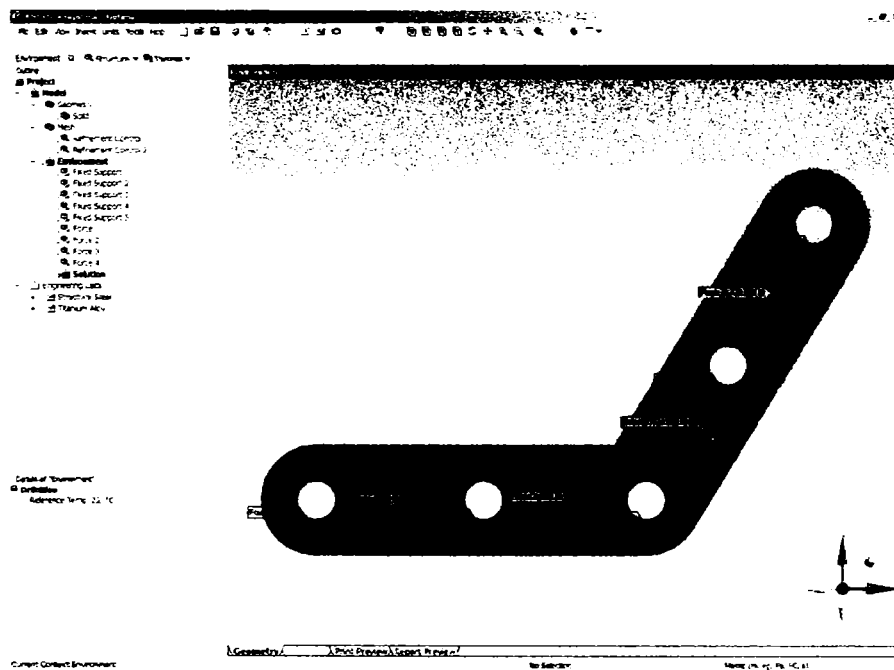


Fig.7-36 Constrângerile sunt impuse în locurile de fixare prin șuruburi

Forțele ce acționează în planul plăcii sunt de 600 N în partea inferioară a plăcii unde mușchii masticației își exercită puternic acțiunea și de 424 N în partea superioară a plăcii. Asupra plăcii, descompusă în 3735 de noduri și 1713 elemente, acționează forțe descrise de următorii vectori, în tabelele 7-1 și 7-2:

<i>Forța 1</i>	In planul plăcii	600.0 N	[0.0 N x, -600.0 N y, 0.0 N z]
<i>Forța 2</i>	In planul plăcii	600.0 N	[0.0 N x, -600.0 N y, 0.0 N z]
<i>Forța 3</i>	In planul plăcii	424.26 N	[300.0 N x, -300.0 N y, 0.0 N z]
<i>Forța 4</i>	In planul plăcii	424.26 N	[300.0 N x, -300.0 N y, 0.0 N z]

Tabelul 7-1 Forțe ce acționează asupra plăcii

<i>Support fix 1</i>	318.08 N	[-104.22 N x, 300.52 N y, -1.73 N z]
<i>Support fix 2</i>	432.84 N	[-136.79 N x, 410.66 N y, 1.52×10 ⁻² N z]
<i>Support fix 3</i>	409.04 N	[-131.04 N x, 387.44 N y, 5.28 N z]
<i>Support fix 4</i>	417.83 N	[-132.2 N x, 396.37 N y, -3.52×10 ⁻² N z]
<i>Support fix 5</i>	319.7 N	[-95.74 N x, 305.01 N y, -3.53 N z]

Tabelul 7-2 Constrângerile impuse

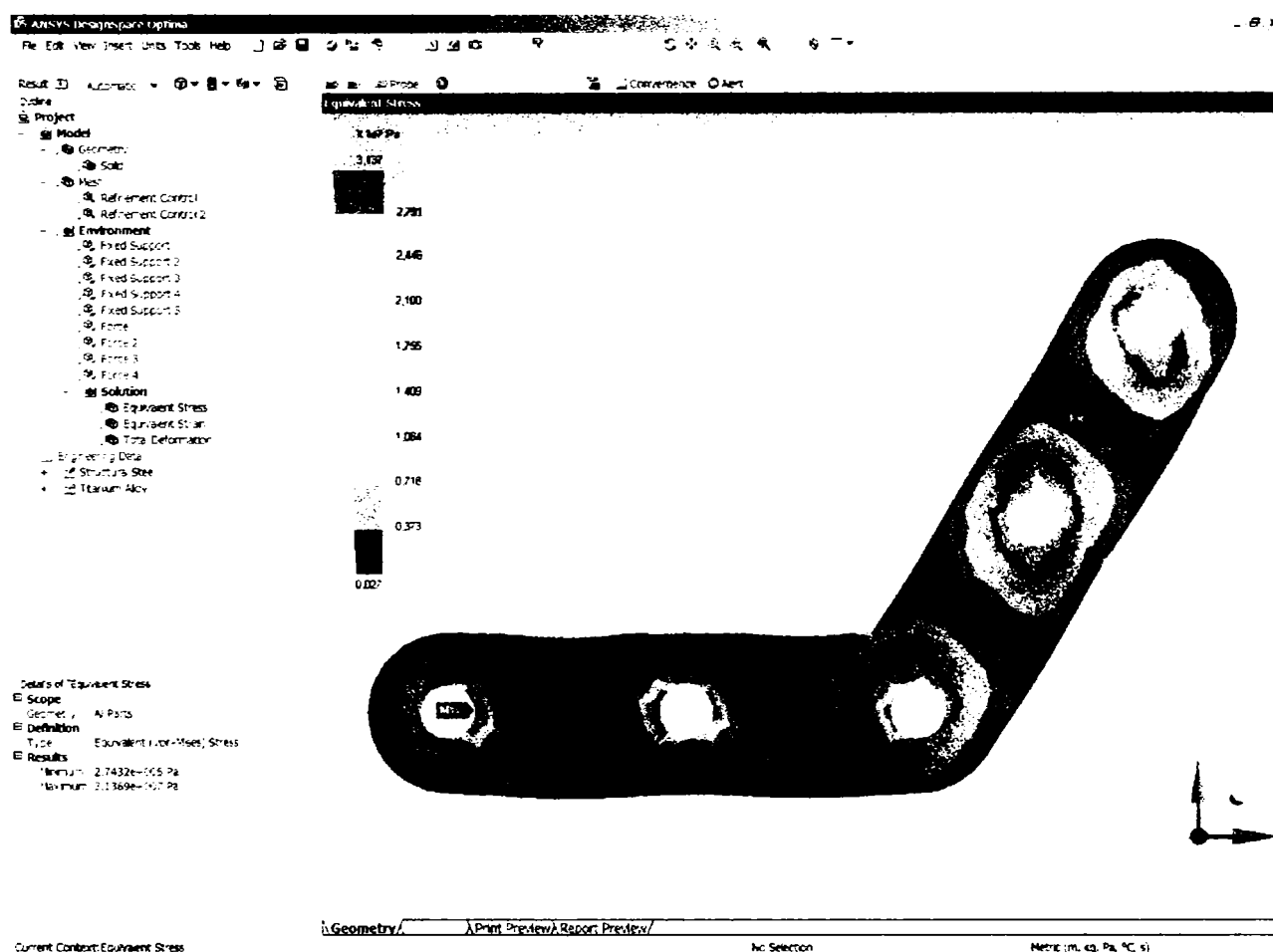


Fig.7-37 Tensiuni echivalente

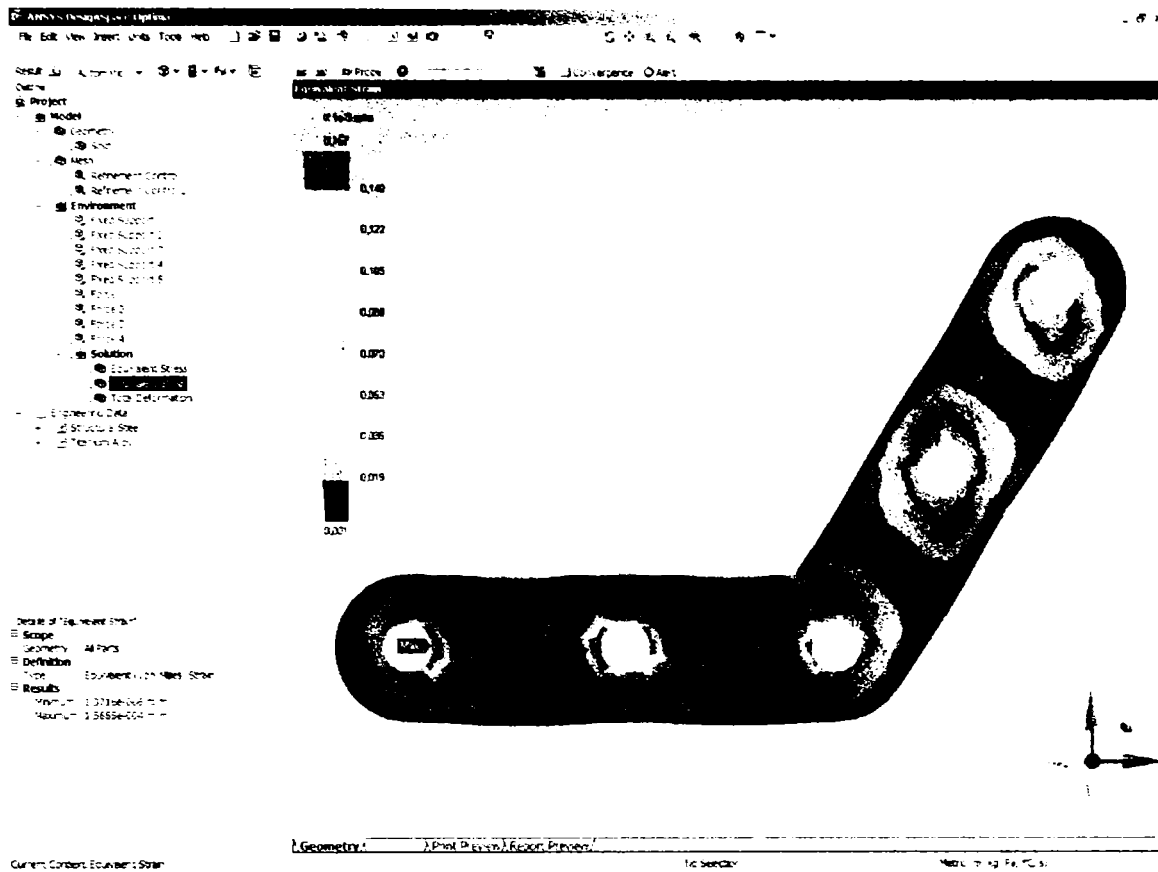


Fig.7-38 Deformarea relativă

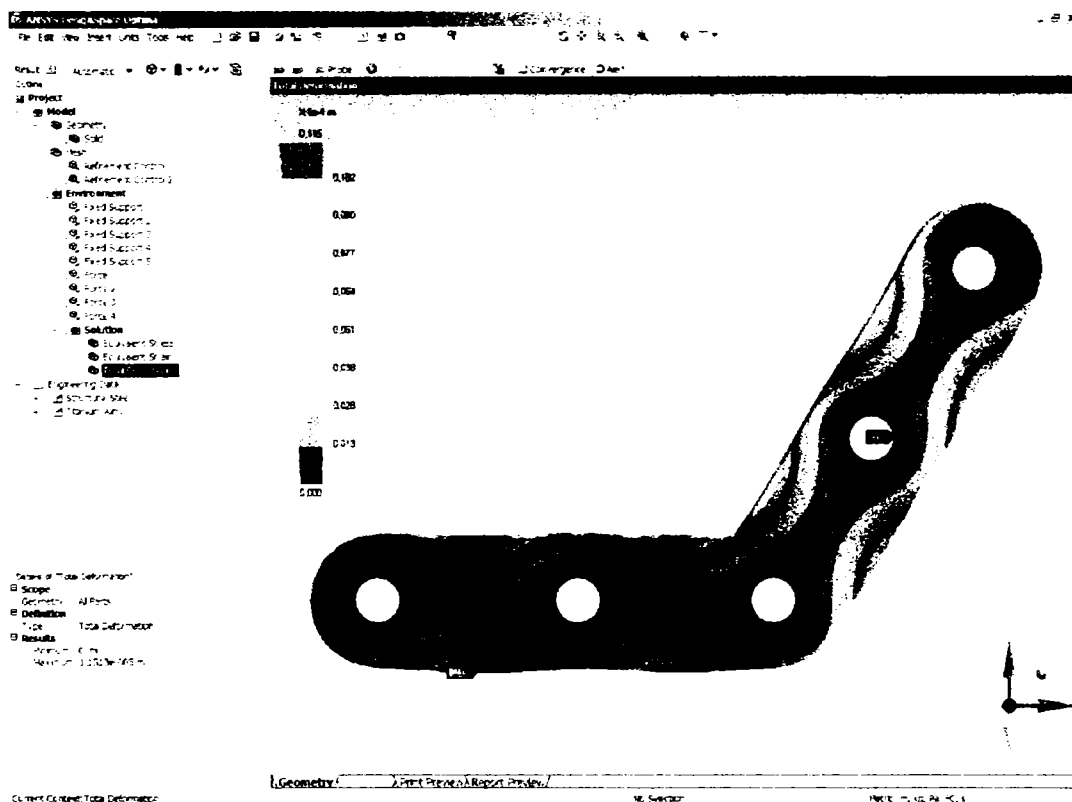


Fig.7-39 Deformarea totală maximă

În urma analizei prin metoda elementelor finite cu ajutorul ANSYS, se obțin rezultatele descrise în tabelul 7-3:

	Min	Max
" <i>Tensiuni echivalente</i> "	274,320.8 Pa	3.14×10^7 Pa
" <i>Deformare relativa</i> "	1.37×10^{-6} m/m	1.57×10^{-4} m/m
" <i>Deformarea totala</i> "	0.0 m	1.15×10^{-5} m

Tabelul 7-3 Rezultatele analizei cu elemente finite

În concluzie se poate spune ca placa se comportă bine la forțe medii, însă pentru a conferi siguranță se poate asocia cu montarea unei plăcuțe LC-DC de susținere, în partea inferioară unde forțele dezvoltate pot fi mai mari.

7.6.6 Placa pentru fracturi cominutive majore dispersate

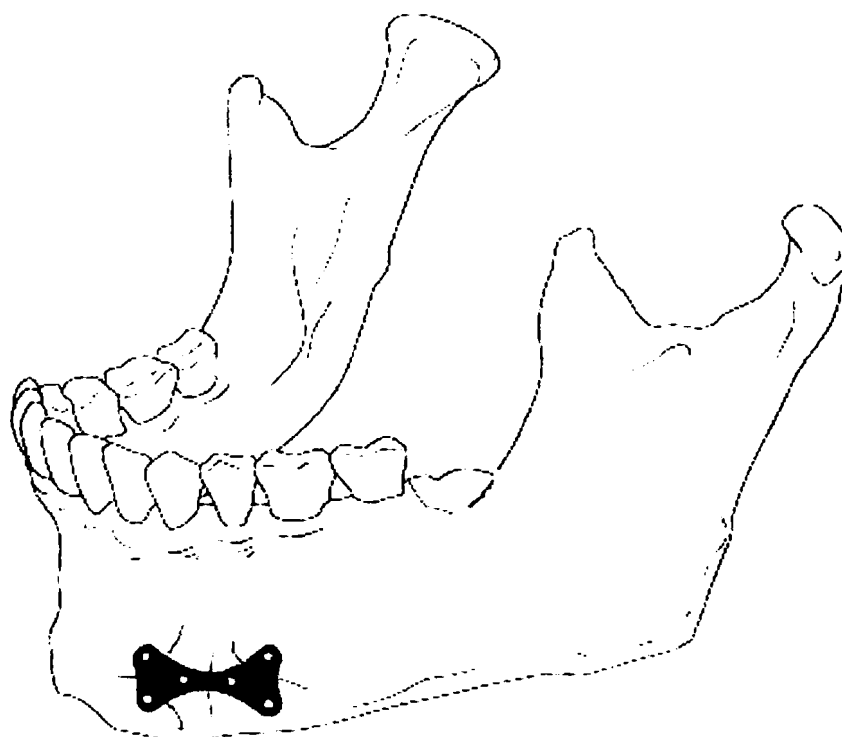


Fig.7-40 Imobilizarea unor fracturi cominutive majore

Placa din figura, având o structură solidă, fără nervuri, poate conferi rezistență fracturilor cominutive, pe care le fixează foarte bine.

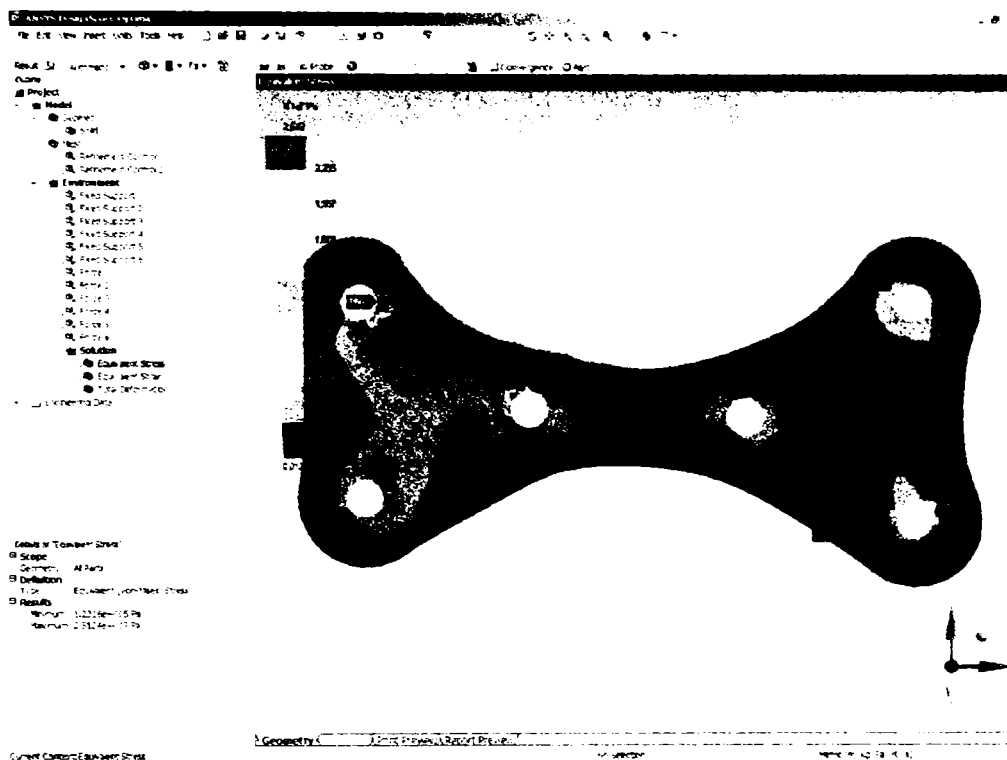


Fig.7-41 Tensiuni echivalente

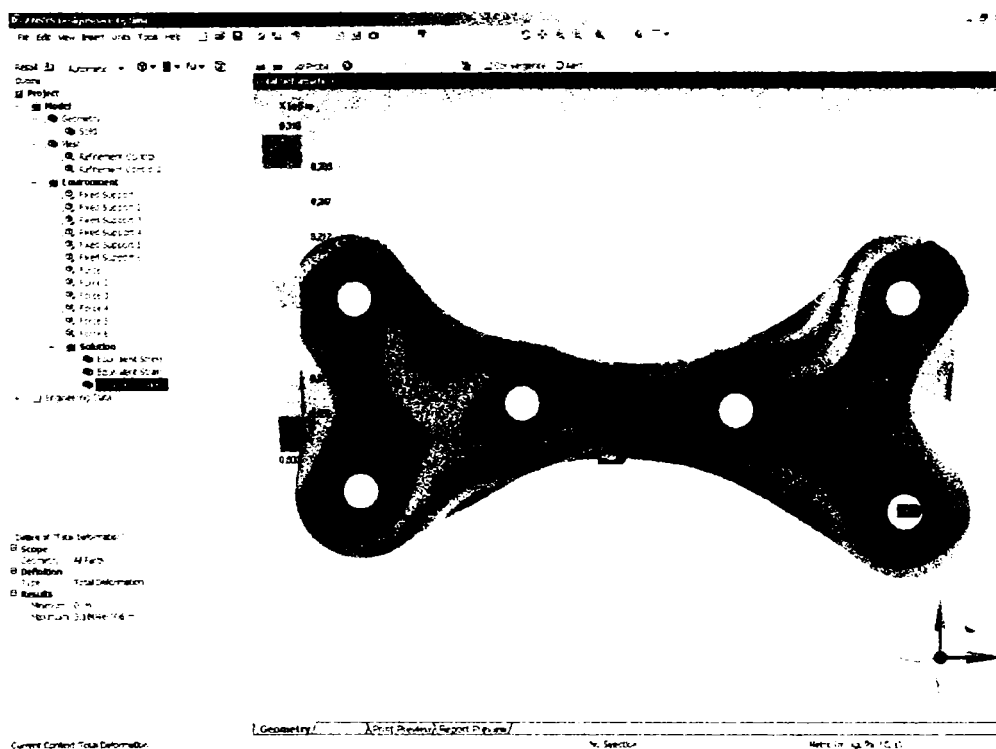


Fig.7-42 Deformarea totală

Modelul plăcii este descompus în 5146 noduri și 2421 elemente. Asupra plăcii acționează următoarele constrângeri și forțe, descrise în tabelul 7-4:

<i>Forța 1</i>	424.26 N	[-300.0 N x, -300.0 N y, 0.0 N z]
<i>Forța 2</i>	300.0 N	[-300.0 N x, 0.0 N y, 0.0 N z]
<i>Forța 3</i>	424.26 N	[300.0 N x, -300.0 N y, 0.0 N z]
<i>Forța 4</i>	200.0 N	[0.0 N x, -200.0 N y, 0.0 N z]
<i>Forța 5</i>	282.84 N	[200.0 N x, -200.0 N y, 0.0 N z]
<i>Forța 6</i>	200.0 N	[200.0 N x, 0.0 N y, 0.0 N z]

<i>Support fix 1</i>	243.07 N	[100.0 N x, 221.53 N y, -2.95 N z]
<i>Support fix 2</i>	180.41 N	[-9.53 N x, 180.15 N y, 1.48 N z]
<i>Support fix 3</i>	250.22 N	[2.8 N x, 250.2 N y, 0.73 N z]
<i>Support fix 4</i>	143.23 N	[-73.75 N x, 122.79 N y, -0.79 N z]
<i>Support fix 5</i>	119.43 N	[-51.67 N x, 107.67 N y, -1.45 N z]
<i>Support fix 6</i>	135.86 N	[-67.86 N x, 117.66 N y, 2.98 N z]

Tabelul 7-4 Forțe și constrângeri ce acționează asupra plăcii

Rezultatele analizei cu elemente finite a plăcii sunt prezentate în tabelul 7-5:

	Minimum	Maximum
<i>"Tensiuni echivalente"</i>	123,163.05 Pa	2.51×10^7 Pa
<i>"Deformarea relativă"</i>	6.16×10^{-7} m/m	1.26×10^{-4} m/m
<i>"Deformarea totală"</i>	0.0 m	3.18×10^{-6} m

Tabelul 7-5 Rezultatele analizei cu elemente finite

Concluzia este că placa asigură rezistență sporită comparativ cu un dispozitiv de fixare clasic cu nervuri, gen plasă de titan.

7.6.7 Placă pentru fracturi cominutive minore dispersate

Pentru acest tip de fracturi se utilizează o plăcuță din aliaj de titan, ușoară, cu nervuri. În continuare este analizat comportamentul acestui tip de placă în cazul unor fracturi ușoare, precum și comportamentul în cazul apariției de forțe mari accidentale.

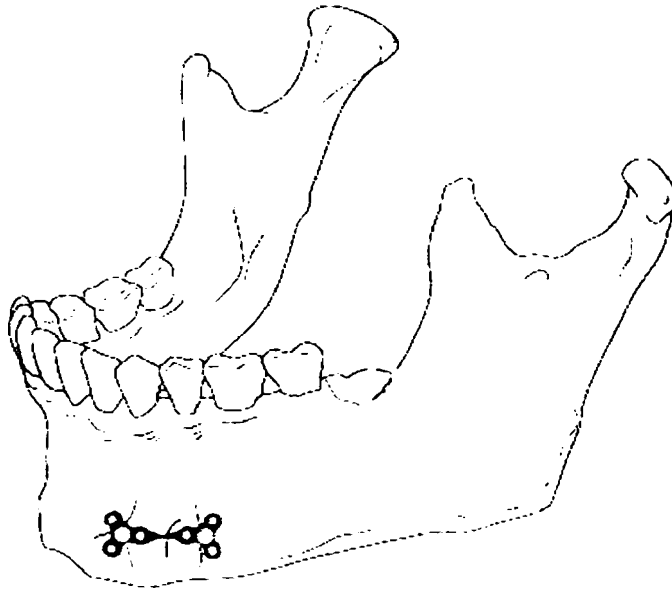


Fig.7-43 Plăcuță din titan, cu nervuri, pentru imobilizarea fracturilor cominutive minore

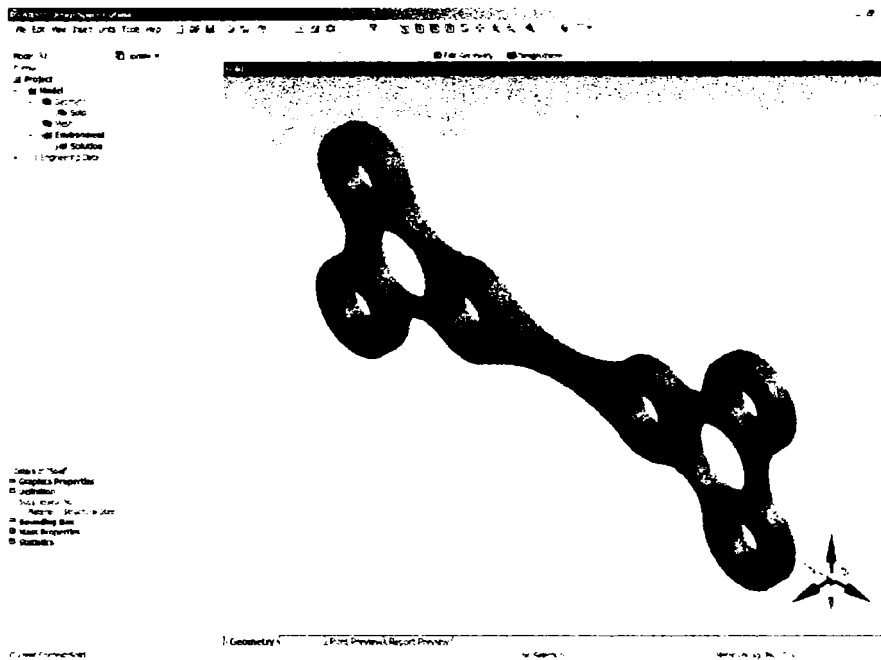


Fig.7-44 Modelul 3D al plăcuței

Modelul este descompus în vederea analizei prin metoda elementelor finite în 2615 noduri și 1266 elemente (figura 7-45):

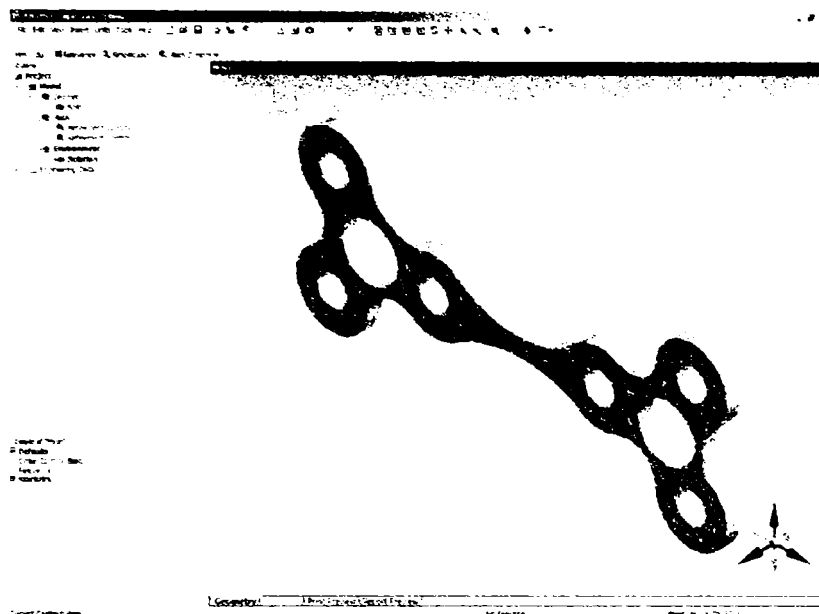


Fig.7-45 Modelul 3D descompus în elemente finite

Forțele la a căror acțiuni este supusă placa au caracteristicile descrise în tabelul 7-6, iar constrângerile și forțele de reacție ce iau naștere în punctele de fixare sunt descrise în tabelul 7-7:

<i>Forța 1</i>	141.42 N	[-100.0 N x, -100.0 N y, 0.0 N z]
<i>Forța 2</i>	100.0 N	[0.0 N x, -100.0 N y, 0.0 N z]
<i>Forța 3</i>	100.0 N	[0.0 N x, -100.0 N y, 0.0 N z]
<i>Forța 4</i>	100.0 N	[0.0 N x, -100.0 N y, 0.0 N z]

Tabelul 7-6 Forțe ce acționează asupra plăcii în cazul descris (fracturi ușoare)

<i>Support fix1</i>	64.03 N	[15.13 N x, 62.1 N y, -3.81 N z]
<i>Support fix2</i>	63.72 N	[16.63 N x, 61.35 N y, 4.47 N z]
<i>Support fix3</i>	83.27 N	[17.54 N x, 81.41 N y, 7.7×10 ² N z]
<i>Support fix4</i>	82.57 N	[21.46 N x, 79.73 N y, 0.49 N z]
<i>Support fix5</i>	60.79 N	[15.54 N x, 58.64 N y, -3.92 N z]
<i>Support fix6</i>	58.46 N	[13.71 N x, 56.77 N y, 2.7 N z]

Tabelul 7-7 Forțele de reacție ce iau naștere în punctele de fixare în care sunt impuse constrângeri

Rezultatele analizei cu elemente finite sunt reprezentate în figurile 7-46 și 7-47:

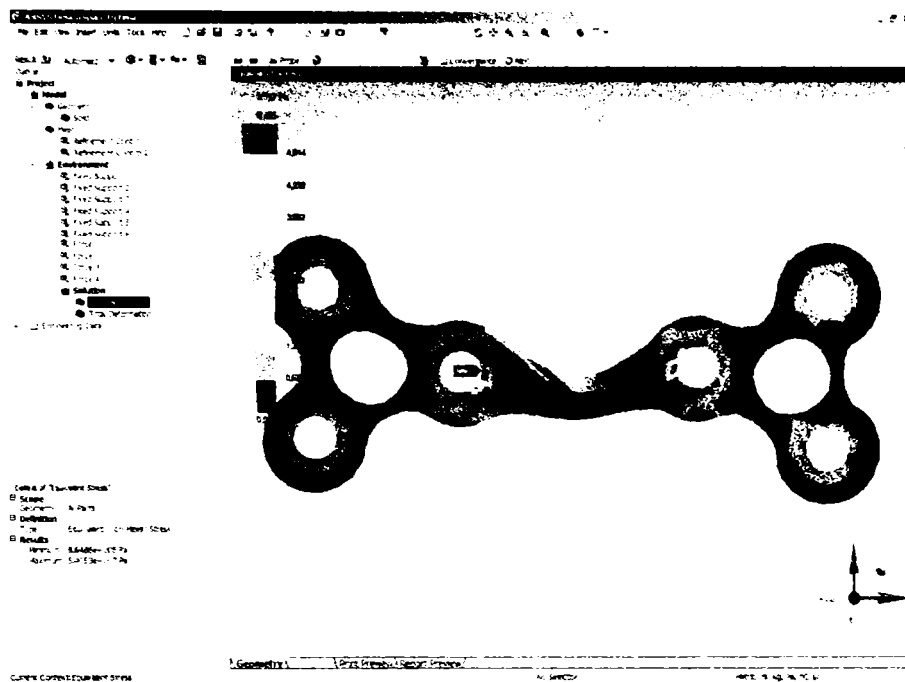


Fig.7-46 Distribuția tensiunilor în placă

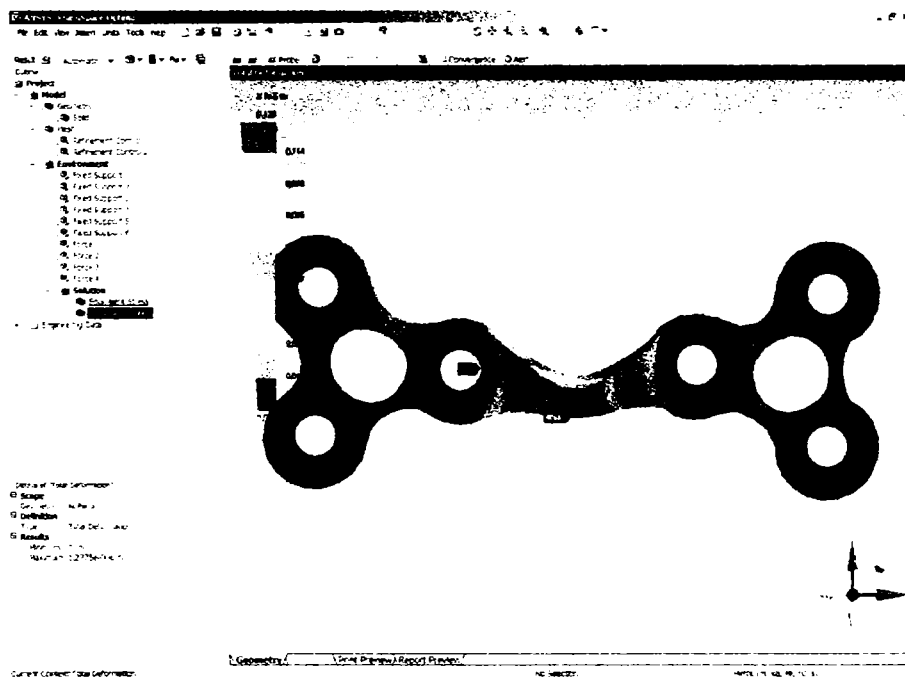


Fig.7-47 Deformarea totală

Tabelul 7-8 descrie rezultatele obținute:

	Minimum	Maximum
" <i>Tensiunea</i> "	864.854.48 Pa	5.41×10^7 Pa
" <i>Deformarea totala</i> "	0.0 m	1.28×10^{-6} m

Tabelul 7-8 Rezultatele obținute prin analiza cu elemente finite

Tensiunile ce intervin sunt sub limitele admise, deformarea există, fiind de valoare extrem de mică, practic doar la nivelul superior al părții din mijloc al plăcii unde s-a aplicat o forță mai mare. Concluzia este că acest tip de placă este utilă în cazurile fracturilor cominutive minore dispersate, în special la nivelul maxilarului. Dacă o linie de fractură prezintă un grad mai ridicat de periculozitate atunci se impune aplicarea unei plăci suplimentare de susținere. Dacă fracturile cominutive ce se doresc a fi immobilizate sunt la nivelul mandibulei, adică acolo unde forțele ce intervin sunt mai mari, se recomandă utilizarea modelului propus și analizat la capitolul 7.6.6.

Cazurile analizate în acest capitol reprezintă propuneri de tipuri de plăcuțe din titan pentru fixarea fracturilor la nivelul mandibulei. Rezultatele obținute și prezentate sunt utile în momentul alegerii tipului de placă necesar de la caz la caz, în special dacă sunt cazuri de risc, astfel încât intervenția să beneficieze de rezultate optime, atât pe baza analizelor de rezistență cât și pe baza experienței medicale.

8. CONTRIBUȚII ORIGINALE

Teza de doctorat *Modelarea computerizată a structurii osoase faciale în vederea protezării și intervențiilor chirurgicale reparatorii* îmbină specificul proiectării asistate de calculator a corpurilor și subansamblelor mecanice cu specificul utilizării calculatorului în domeniul medical pentru prelucrare de informații și imagini, extragere de date esențiale și modelare.

Teoriile aplicate și algoritmi utilizați, rezultatele obținute, atestă că teza cu temă interdisciplinară, aflată la frontiera dintre inginerie software și inginerie mecanică cu aplicații în domeniul biomedical, și-a atins scopul. Multitudinea de tehnici utilizate, algoritmi software clasici dar și cei originali fac posibilă modelarea 3D a structurilor anatomice complexe.

Contribuția autorului constă atât în punerea în valoare a unor tehnici moderne aplicate în domeniul proiectării de modele biomecanice, cât și în introducerea de soluții noi, originale, viabile, concretizate în programe, proceduri, algoritmi și tehnici ce formează un sistem complet de proiectare în domeniul biomecanic.

În capitolul 2, *Conceptul de model și modelare*, este definit modelul ca reprezentare a aspectelor esențiale ale unui sistem, precum și noțiuni de bază și aspectele de care trebuie să se țină seama în definirea unui model, avantajele utilizării modelelor și modul de exprimare și apreciere a calității acestora. Este abordată și o clasificare a modelelor în funcție de materialitate, de forma de prezentare și aspect. Tot în acest capitol se definește conceptul de modelare și se realizează o trecere în revistă a etapelor parcurse în activitatea de modelare.

Autorul aduce elemente de originalitate prin:

- sintetizarea caracteristicilor generale ale conceptelor de model și modelare;
- particularizarea caracteristicilor esențiale ale modelelor biomecanice, părți osoase, sau moi, aparținând diverselor sisteme organice, modele ce trebuie să satisfacă cerința de a fi cât mai aproape de realitate.
- reliefaarea aspectelor de bază și celor particulare ale modelelor biomecanice.
- crearea de modele virtuale originale ale unor părți osoase ale scheletului uman, modele realizate pe baza rezultatelor obținute în teză.

În capitolul 3, *Metode pentru realizarea formelor complexe în modelarea geometrică*, autorul abordează teoretic și practic:

- tehnici pentru descrierea modelelor geometrice tridimensionale;
- tipurile de reprezentări ale modelelor;
- metodele de creare a modelelor solide.

Elementele de originalitate constau în:

- prezentarea unor modele biomecanice originale, obținute prin tehnici diferite de reprezentare: geometric-constructivă, prin frontiere, prin enumerare spațială, prin baleiere;
- comparația între avantajele și dezavantajele care intervin în cazul acestor reprezentări, respectiv modele, având ca scop identificarea unei metode originale și adecvate proiectării de modele biomecanice;
- propunerea argumentată a utilizării combinației dintre tehnica de reprezentare prin enumerare spațială și a unei metode booleene, ca soluție viabilă pentru proiectarea în domeniul biomecanic.

În capitolul 4, *Vizualizarea și transformarea modelelor 3D*, autorul prezintă pe larg:

- tipurile de afișări;
- procedura de vizualizare;
- transformările ce se aplică modelelor în spațiul tridimensional: scalare, rotire, translatare;
- clasificarea tipurilor de proiecții și particularitățile acestora;
- metoda *ray-tracing* de redare realistă a scenelor 3D;
- librăria grafică OpenGL, care stă la baza tuturor algoritmilor importanți utilizați în teză.

Elementele originale introduse în acest capitol sunt:

- crearea unui cadru software original bazat pe OpenGL pentru întreaga aplicație dezvoltată în teză: un sistem de reconstrucție și vizualizare 3D a structurilor de țesut, a organelor pentru care există descrieri 2D rezultate în urma analizelor, sistem ce oferă posibilitatea unei analize mult îmbogățite din punct de vedere al informației pentru orice organ sau sistem anatomic uman,

precum și proiectarea, modelarea și analiza implantelor și structurilor mecanice specifice necesare în intervențiile chirurgicale reparatorii și de protezare la nivelul feței.

- proceduri de construire a unei scene 3D, bazate pe OpenGL, proceduri destinate reprezentării diferitelor tipuri de suprafețe 3D reconstituite;
- proceduri de manipulare facilă a componentelor scenei 3D (actorilor), prin care s-a realizat facilitatea de manevrare a modelelor doar cu ajutorul mouse-ului;
- programe pentru transformări geometrice aplicate modelelor în contextul OpenGL, ce permit investigarea din diverse unghiuri a modelelor afișate.

Capitolul 5, *Reconstituirea și modelarea 3D a structurilor anatomice complexe*, este cel care descrie pe larg metoda originală de obținere a modelelor tridimensionale din seturi de date specifice imagisticii medicale.

Contribuțiile originale sunt reprezentate de:

- prezentarea metodei de reconstituire și modelare 3D a suprafețelor aplicată structurilor anatomice complexe;
- filtrul de volum pentru detecția suprafețelor caracterizate de aceeași densitate de țesut;
- algoritm software de citire a datelor provenite de la echipamentele de imagistică medicală;
- algoritmul software de stocare a datelor citite;
- algoritm software de parcurgere optimizată a stivei de imagini;
- algoritm software de segmentare interactivă cu două praguri variabile;
- algoritm software de segmentare bazat pe detecția conturilor;
- metoda de construire a setului volumetric de date;
- metoda de filtrare a izosuprafeței dorite;
- algoritm de divizare a volumului (stivei de imagini) într-o rețea de elemente unitare, conținând valori semnificative;
- metoda de identificare a intersecțiilor unei suprafețe cu o altă suprafață;
- metoda de decizie a apartenenței unei entități la o suprafață căutată;
- metodă de optimizare a căutărilor tuturor posibilităților de intersecție, cu reducerea, pe baze matematice, a numărului de cazuri posibile.

- metodă de interpolare, pentru aproximarea cât mai exactă a suprafețelor 3D;
- metodă de optimizare a calculelor de obținere a izosuprafețelor din intersecțiile cu elemente unitate, bazată pe coerența pixel cu pixel, linie cu linie și plan cu plan;
- algoritm software de conversie a structurilor de memorie volumetrică în format STL (stereolitografie) și VRML (realitate virtuală), facilitate ce permite importul modelelor 3D astfel proiectate în alte programe specializate de tip CAD-CAM.

Totalitatea acestor elemente originale constituie nucleul unui sistem original de reconstrucție a suprafețelor 3D din seturi de tomograme, sistem ce permite:

- modelarea structurilor anatomice extrem de complexe, individualizate pe pacient;
- manipularea lor ca obiecte 3D;
- reconstituirea structurii interne ale organelor investigate;
- posibilitatea tratării separate a fiecărui model 3D organic rezultat chiar în cazul apartenenței la același sistem organic;
- posibilitatea disecției virtuale, oferindu-se facilitatea de investigare stratificată, bazată pe densitățile diferite ale diverselor țesuturi;
- posibilitatea vizualizării preoperative în cazurile complexe;
- obținerea de modele 3D reale ale structurilor anatomice complexe, modele ce sunt utilizate pentru proiectarea de implante și pentru analiza cu metoda elementelor finite a comportării reale a implantelor.

Sunt prezentate reconstituiri bazate pe date reale achiziționate cu ajutorul tomografului Somatom Plus 4 din dotarea Bazei de Cercetare cu Utilizatori Multipli *Centrul de Modelare a Protezării și Intervențiilor Chirurgicale asupra Scheletului Uman*, ale câtorva structuri anatomice: craniul în ansamblu, mandibula, maxilarul, coloana vertebrală, coastele, pelvisul, oasele mâinilor și ale picioarelor.

Practic a fost dezvoltat un sistem de reconstrucție, modelare și vizualizare 3D a structurilor biomecanice, a părților anatomice pentru care există descrieri bidimensionale rezultate în urma investigațiilor radiografice, sistem ce oferă posibilitatea unei analize mult îmbogățite din punct de vedere al informației pentru orice sistem anatomic uman,

precum și proiectarea, modelarea și analiza implantelor și structurilor mecanice specifice necesare în intervențiile chirurgicale reparatorii și de protezare.

Capitolul 6, *Considerații anatomice*, este rezervat considerațiilor de ordin anatomic și descrie oasele și mușchii capului.

Elementul de originalitate îl constituie abordarea biomecanică, ce consideră capul, ca și restul trupului un sistem mecanic, care interacționează cu mediul și asupra căruia se acționează static și dinamic. Rolul acestui capitol este acela de a încerca asimilarea unui sistem biologic cu un sistem mecanic, fiind prezentate elemente strâns legate de mecanica mandibulei, necesare analizei cu elemente finite a acesteia.

Prezentarea anatomiei capului din punct de vedere biomecanic a oferit posibilitatea calculării forțelor dezvoltate de mușchi, ale căror valori se iau în considerare ca date de intrare pentru analiza numerică a stărilor de tensiune și deformațiilor din implante.

Capitolul 7, *Modelarea și analiza structurilor mecanice reparatorii utilizate în chirurgia maxilo-facială*, aduce următoarele contribuții originale:

- prezintă consecințe ale rezultatelor modelării 3D obținute;
- analizează metodele de remediere a defecțiunilor modelelor, înainte de a fi aplicate pe obiectul real;
- considerații privind analiza și modelarea implantelor și a structurilor mecanice reparatorii utilizate în chirurgia maxilo-facială;
- metodă originală de modelare a implantelor de formă naturală, prin izolarea zonei de implantat și aplicarea metodei descrise la capitolul 5 dar cu parametri de calcul inversați;
- analiza cu elemente finite a structurilor utilizate cu scop reconstructiv utilizate în situații reale. Sunt analizate comportările diferitelor tipuri de plăcuțe aplicate pe os, la interacțiunea cu diverse forțe.

Experimentele efectuate pe cazuri reale și rezultatele prezentate dovedesc că soluția prezentată în teză furnizează rezultate foarte aproape de realitate, constituind un instrument de bază pentru luarea deciziilor în cazurile de diagnosticare și planificare a intervențiilor prin sistemul de vizualizare preoperativă 3D, pentru modelarea 3D a structurilor biomecanice, cât și pentru analiza structurilor reparatorii.

Perspectivile de viitor sunt legate realizarea fizică a modelelor proiectate cu ajutorul soluției propuse. Modelele 3D proiectate, disponibile atât în format STL (stereolitografie) cât și în format VRML (realitate virtuală), pot fi importate într-un program specific CAM, pentru fabricarea asistată de calculator. O direcție de dezvoltare a acestui sistem este implementarea unui modul CAM, care va gestiona totalitatea operațiilor necesare pentru obținerea pe baza modelului 3D, pentru care se dispune de toate datele necesare, a comenzilor specifice pentru comanda unor dispozitive speciale de prelucrare 3D.

O altă perspectivă de viitor o reprezintă realizarea fizică a implantelor de formă naturală, individualizate pe pacient, modelate prin metoda propusă în teză, implante ce pot fi realizate din materiale plastice speciale cu proprietăți asemănătoare osului, sau din titan.

Extrem de utilă este și integrarea soluției propuse în tomografele computerizate actuale, pentru extinderea capabilităților acestora cu facilități de investigare 3D în timp real, disecție virtuală, vizualizare preoperativă, facilități ce oferă o altă perspectivă investigațiilor medicale.

BIBLIOGRAFIE

1. Alling, C.C., Osbon, B.D., "Maxillofacial trauma", Lea and Febiger, 1988;
2. Anderson, T., Albert, B., „Experience with Rigid Fixation of Mandibular Fractures and Immediate Function”, J. Oral Maxillofac. Surg., 1992, 50, pag. 555;
3. Ansys Incorporated, "Ansys – User’s Guide”, 2002;
4. Atherton, P.R., Weiler, K., Greenberg, D., "Polygon Shadow Generation”, SIGGRAPH 78, pag. 275-281;
5. Barson, J.M., Rolfe, S.T., „Fracture & Fatigue Control in Structures”, Second Edition, Prentice-Hall;
6. Bathe, K.I., Wilson, F.L., „Numerical Methods in Finite Element Analysis”, Prentice-Hall, INC, Englewood, New Jersey, 1976;
7. Battersby, T.G., "Plating of Mandibular Fractures”, British Journal of Oral Surgery, 1967, 4, pag. 194;
8. Beach, R.C., "An Introduction to the Curves and Surfaces of CAD”, VNR, Computer Library, New York, 1991;
9. Becker, R., "Stable Compression Plate Fixation of Mandibular Fractures”, British Journal of Oral Surgery, 1974, 12, pag. 13;
10. Bennet, B.S., "Simulation Fundamentals”, Prentice Hall, Londra, Munich, 1995;
11. Berbente, C., Zancu, S., Mitran, S., Pleter, O., Tăranu, C., "Metode numerice de calcul și aplicații”, Ed. I.P.B., București, 1992;
12. Bezzel, B., "Graphics programming in C++”, Addison Wesley Publishing Company, 1990;
13. Bia, C., Ile, V., Soare, M.V., "Rezistența materialelor și teoria elasticității”, Ed. Didactică și Pedagogică, București, 1983;
14. Blinn, J., Newell, M., "Texture and reflection in computer generated images”, CACM, octombrie 1976;
15. Block, C., Boyne, P.J., "Use of a Metal Intraosseous Fixation Device for Treatment of Fractures of Atrophic Edentulous Mandibles”, J. South Calif. State Dent. Assoc., 1972, 40, pag.996;
16. Blumenfeld, M., "Introducere în metoda elementelor finite”, București, Ed. Tehnică, 1995;

17. Blumenfeld, M., Ioniță, A., Mares, C., “Metoda elementelor finite (Aplicații și programe introductive), București, Ed. I.P.B., 1992;
18. Blythe, David “Programming with OpenGL: Advanced Rendering”, 1997, SIGGRAPH;
19. Boboc, Gh., “Aparatul dentomaxilar-Formare și dezvoltare”, Editura Medicală, București, 1979;
20. Bolon, P., Mouhoub, M., “Une famille de detecteur de contours bases sur le filtrage d'ordre”, GRETSI, 1989;
21. Booth, Shanon “Volume Visualisation with Marching Cubes”, 1997;
22. Bracewell, R.N., “Two-Dimensional Imaging”, Prentice Hall, 1995;
23. Budynas, R.G., “Advanced Strength and Applied Stress Analysis”, McGraw-Hill, N.Y., 1977;
24. Burlibașa, C. și colab., „Chirurgie orală și maxilofacială”, Ediția a II-a, Editura Medicală, București, 1999;
25. Burlibașa, C. și colab., “Curs de chirurgie buco-maxilo-facială”, București, 1988;
26. Buzdugan, Gh., “Rezistența materialelor”, București, Ed. Academiei, 1986;
27. Campy, M., Lodde, J.P., “Synthèses mandibulaires. Localisation des syntheses en fonction de contraintes mandibulaires”, Rev. de Stom. Et de Chir. Max.-Fac., Paris, 1976, 77;
28. Campy, M., Lodde, J.P., Schmitt, R., Jaeger, J.H., Muster, D., “Mandibular Osteosynthesis by Miniature Screwed Plates Via a Buccal Approach”, Journal of Maxillo-Facial Surgery, 1978;
29. Canny, J.F., “A Computational Approach to Edge Detection”, IEEE-PAMI, 1986;
30. Carey, R., Bell, G., Marrin, C., ISO/IEC 14772-1:1997, “Virtual Reality Modeling Language”, URL: <http://www.vrml.org/Specifications/VRML97>, 1997;
31. Carpenter, L., “The A-buffer and Antialiased Hidden Surface Method”, SIGGRAPH 1984, pag.103-108;
32. Castleman, K.R., “Digital Image Processing”, Prentice Hall, 1996;
33. Catmull, E., “A Subdivision Algorithm for Computer Display of Curved Surfaces”, Ph.D.Thesis, Report UTEC-CSc, 1974;
34. Catmull, E., “Computer Display of Curved Surfaces”, IEEE Computer Graphics, mai 1975;

35. Catmull, E., "A Hidden Surface Algorithm with Anti-Aliasing", SIGGRAPH 78;
36. Cattone, A.C., "Complex Maxillofacial Injuries, from Maxillofacial Trauma", Lea and Febiger, 1988, pag. 419-494;
37. Challapa, R., "Digital Image Processing", 1993;
38. Champy, M., Pape, H.D., Gerlach, K.L., Lodde, J.P., "The Strasbourg Miniplates Osteosynthesis", Oral and Maxillo-Facial Traumatology, Berlin, Quintessence, 1986, pag. 19;
39. Chateau, M., "De la possibilité de modifier des bases osseuses", Orthodont. franç., 1956;
40. Choung, R., Piper, A.M., "Open Reduction of Condylar Fractures of the Mandible in Conjunction with Repair of Discal Injury", J. Oral Maxillofac. Surg., 1988;
41. Ciarlet, P.G., "The Finite Element Method", North-Holland, Amsterdam, 1978;
42. Clark, J., "A Fast Scan-Line Algorithm for Rendering Parametric Surfaces", SIGGRAPH 79;
43. Cohen, M.F., Greenberg, D.P., Immel, D.S., Brock, P.J., "An efficient radiosity approach for realistic image synthesis", IEEE Computer Graphics and Applications, 1986;
44. Cojocaru, D., "De la coordonate-imagine la coordonatele lumii reale", Scientific Seminar of Computer Engineering Dept., Automation, Computer and Electronics Faculty, ELSE Software, p165-171, nr. 9, Craiova, Romania, 1996;
45. Constantinescu, I., Dăneț, N., "Metode noi pentru calcule de rezistență", București, Ed. Tehnică, 1989;
46. Constantinescu, P., Negoită, V., "Sisteme informatice, modele ale conducerii și sistemelor de conducere", Editura Tehnică, București, 1975;
47. Cook, R.L., Torrance, K.E., "A Reflectance Model for Computer Graphics", ACM Trans. On Graphics, Vol.1, 1982, pag. 7-24;
48. Cox, H.N., Van Der Linden P.G.M., "Facial Harmony", Amer. J. Orthodont., 1971, 60, pag. 175;
49. Crow, F., "Shadow Algorithms for Computer Graphics", SIGGRAPH 77, pag.242-247;
50. Cuteanu, E., Marinov, A., "Metoda elementelor finite în proiectarea structurilor", Timișoara, Ed. Facla, 1980;

51. Cyrus, M., Beck, J., "Generalized Two-and Three-Dimensional Clipping", *Computers and Graphics*, 3(1), 1978, pag.23-28;
52. Dally, J.W., Riley, W.F., "Experimental Stress Analysis", Third Edition, McGraw-Hill, 1991;
53. David, J.M., Simpson, A.D., "Craniomaxillofacial Trauma", Churchill Livingstone, 1995;
54. Demidovitch, B., Maron, I., "Elements de calcul numerique", Moscove, Ed.Mir, 1973;
55. Deriche, R., "Optimal Edge Detection Using Recursive Filtering", Proc. First Conf. on Computer Vision, London, June, 1987;
56. Dippe, M., Swensen, J., "An Adaptive Subdivision Algorithm and Parallel Architecture for Realistic Image Synthesis", SIGGRAPH 84;
57. Doroga, H., "Suspensia prin ligaturi interne în tratamentul fracturilor maxilarului superior", *Consfătuirea Națională de Chirurgie*, Cluj-Napoca, 22-23 mai, 1976;
58. Doroga, H., "Contribuții la tratamentul chirurgical al fracturilor de maxilare", Teză de doctorat U.M.F., Cluj-Napoca, 7.07.1972;
59. Dumitru, I., Faur, N., "Elemente de calcul și aplicații în rezistența materialelor", Timișoara, Ed. Politehnica, 1999;
60. Dumitru, I., Faur, N., "Rezistența materialelor-Bazele teoretice în oboseala materialelor, mecanica ruperii, composite, metode de analiză numerică", Lito. Univ. Politehnica Timișoara, 1997;
61. Dym, H., Coro, C., Ogole, E.O., "Screw-Wire Osteosynthesis Technique for Intraoral Open Reduction of Mandibular Angle Fractures", *J. Oral Maxillofac. Surg.*, 1992, 50, pag. 1247;
62. Ellis, E., Reynolds, S.T., Park, H.S., "A Method to Rigidly Fix High Condylar Fractures", *Oral surg., Oral med., Oral pathol.*, 1989, 68, pag. 369;
63. Ellis, E., Tharanon, W., "Facial with Problems Associated with Rigid Fixation of Mandibular Fractures: Case Reports", *J. Oral Maxillofac. Surg.*, 1992, 50, pag. 87;
64. Encarnaçao, J., Schlenchtendahl, E.G., "Computer Aided Design", Springer Verlag, Berlin, 1983;
65. Enterprise Software Products Inc., "FEMAP – User's Guide";

66. Facultatea de Calculatoare, Automatică și Electronică Craiova, Dep.Robotică, curs, "Achiziția, prelucrarea, și recunoașterea imaginilor";
67. Faugeras, Ponces, J., "A Hierarchical Representation for 3D Objects", IJCAI, 1983;
68. Faur, N., "Elemente finite-Fundamente", Timișoara, Ed. Politehnica, 2002;
69. Faur, N., Dumitru, I., "Diferențe finite și elemente finite", Timișoara, Ed. Mirton, 1997;
70. Faur, N., Dumitru, I., "Metode numerice în rezistența materialelor", Timișoara, Lito. Univ. Politehnica, 1997;
71. Faux, I.D., Pratt, M.J., "Computational Geometry for Design and Manufacture", Ellis Horwood, Chichester UK, 1979;
72. Finite Element Method Universal Resource – <http://femur.wpi.edu>;
73. Firu, P., Milicescu, V., Diaconescu, M., "Dezvoltarea aparatului dentomaxilar", Stomatologia, București, 1971, 1, pag. 1;
74. Foley, J., Van Dam, A., Feiner, S., Hughes, J., "Computer Graphics – Principles and Practice", Addison Wesley Publ.Comp, 1992;
75. Fonseca, J.R., Walker V.R., "Oral and Maxillofacial Trauma", W.B. Saunders Company, 1991;
76. Free Finite Element Analysis, Web Resource, www.geocities.com/Athens/2099;
77. Fuchs,H., Kedem, Z.M., Naylor, B.F., "On Visible Surface Generation by A Priori Tree Structures", SIGGRAPH 80, pag. 124-133;
78. Gârbea, D., "Analiza cu elemente finite", București, Ed. Tehnică, 1990;
79. Ghinea, M., Fireteanu, V., "MATLAB; calcul numeric-grafică-aplicații", Editura TEORA, București, 1995;
80. Glassner, A., "An Introduction to Ray Tracing", Academic Press, 1989;
81. Gola, R., Chossergros, C., Waller, P.Y., Delmar, H., Cheynet, F., "Fractures de la région condylienne", Rev. Stom. Chir. Max., 1992, 93, pag. 70;
82. Gonzales, R.C., Woods, R., "Digital Image Processing", Addison-Wesley, 1993;
83. Gouraud, H., "Continuous shading of curved surfaces", IEEE Trans on Computers, 20(6), 1971, pag. 623-628;
84. Grimson, W.E., Huttenlocher, D.P., "On the Sensitivity of the Hough Transform for Object Representation", IEEE-PAMI, vol. 12, no. 3, March, 1990;
85. Hammersley, Tom "Viewing System for 3D Engines" Web Resource;

86. Hauenstein, P., "Applications de la tomographie à l'étude du problème de l'expansion transversale du maxillaire sup., *Orthodont. franç.*, 1963, 36, pag. 283;
87. Hauenstein, P., "Croissance et correction de la retromandibule, *Orthodont. franç.*, 1968, 39, pag. 457;
88. Hărdău, M., "Metoda elementelor finite (Curs)", Cluj-Napoca, Ed. Univ. Tehnică, 1995;
89. Heubner, K.H., "The Finite Element Method for Engineers", John Wiley and Sons, N.Y.-Toronto, 1975;
90. Hill, Francis "Computer Graphics Using Open GL (2nd Edition)";
91. Hope, Hugues, Eck, M., "Multiresolution Analysis of arbitrary meshes", *ACM SIGGRAPH*, 1995, pag. 173-182;
92. Hope, Hugues "Surface Reconstruction from Unorganized Points", Ph.D. Thesis, University of Washington, 1994;
93. Hope, Hugues, DeRose, T., "Mesh Optimization", *ACM SIGGRAPH*, 1993, pag. 19-26;
94. Hughes, T.J.R., "The Finite Element Method", Prentice-Hall, Englewood Cliffs, N.J., 1987;
95. ISO/IEC 14772 "The Virtual Reality Modeling Language Specification", URL: <http://vrm1.sgi.com/moving-worlds/spec/>, 1997;
96. Jamsa, K., Schmauder, P., Yee, N., "Biblioteca programatorului VRML", Editura All, București, 1998;
97. Jurkiewicz, M.J., Krizek, J.T., Mathes, J.S., Ariyan, S., "Plastic Surgery: Principles and Practice", C.V. Mosby Comp., 1990;
98. Kajiya, J., "Ray Tracing Parametric Patches", *Computer Graphics*, pag. 245-254;
99. Kapur Tina, "Knowledge Based 3D Medical Image Segmentation", Ph.D. Thesis, 1999;
100. Kari, P., "Surface Reconstruction and Display from Range and Colour Data", 1997;
101. Kari, P., "View Based Rendering", 8th Eurographics Workshop on Rendering, St. Etienne, France;
102. Kari, P., Segal, M., "Fast Rendering of Subdivision Surfaces";
103. Kay, D., Greenberg, D., "Transparency for Computer Synthesized Images", *SIGGRAPH 79*, pag. 158-164;

104. Kay, T.L., Kajiya, T.J., "Ray Tracing Complex Scenes", SIGGRAPH 86, pag. 269-278;
105. Kellman, R.M., Marrentette, L.J., "Atlas of Craniomaxillofacial fixation", Raven Press, New York, 1995;
106. Kruger, O.G., "Textbook of Oral and Maxillofacial Surgery", Sixth Ed., The C.V. Mosby Comp., St. Louis-Toronto, 1984;
107. Kruger, E., Schilli, W., Worthington, P., "Oral and maxillofacial traumatology", Quintessence Publishing Co, 1986;
108. Kufner, J., "A Method of Cranio-Facial Suspension", Journal of Oral Surg., 1970, 28, pag. 260;
109. Lachner, J., Clanton, T.J., Waite, P.D., "Open Reduction and Internal Rigid Fixation of Subcondylar Fractures Via an Intraoral Approach", Oral surg., Oral med., Oral pathol., 1991, 71, pag. 257;
110. Lane, J., Carpenter, L., Whitted, T., Blinn, J., "Scan Line Methods for Displaying Parametrically Defined Surface", CACM 23, ianuarie 1980;
111. Lars, M.V., "Procedures in Plastic and Reconstructive Surgery: How They Do It", Little Brown and Company, Boston-Toronto-London, 1991;
112. Liang, Y.D., Barsky, B., "A New Concept and Method for Line Clipping", ACM TOG, 3(1), 1984, pag.1-22;
113. Lorensen, W., Schroeder, W., Ken M., "Volume Visualisation Algorithms Implementation";
114. Lucchese, P., Bassani, S., "La croissance faciale étudiée par rapports aux types squelettiques", Orthodont. franç., 1968, 39, pag. 153;
115. Luchin, M., "Modelare geometrică. Referat doctorat nr.2, Timișoara, 1998;
116. Luchin, M., "Clasificarea și descrierea modelelor", AXXVIII-a Sesiune de comunicări științifice cu participare internațională, Academia Tehnică Militară, București, 1999;
117. Luchin, M., "Considerații asupra semnificației și conținutului conceptelor de model și modelare", A XXVIII-a Sesiune de Comunicări științifice cu participare internațională, Academia Tehnică Militară, București, 1999;
118. Luchin, M., "Tendințe actuale în abordarea teoriei modelării", Robotica&Management 1/1999, ISSN 1453-2069, Reșița, 1999;

119. Luhr, H.G., "Trauma of the Facial Skeleton: Procedures in Plastic and Reconstructive Surgery How They Do It", Editura By LarsM. Vistnes Little Brown and Comp., 1991, pag. 441;
120. Manson, N.P., "Facial Injuries in Plastic Surgery", vol. II, McCarthy part.1, W.B. Saunders Comp., 1990;
121. Manson, N.P., and coll., "Midface fractures: Advantages of immediate extendent open reduction and bone plastic and reconstructive surgery", 1985, 76, 1, 1-12;
122. Marin, D., Rotaru, Al., Orosz, K., "Posibilități de rezolvare în fracturile de mandibulă cu lipsă de substanță", U.S.S.M., Cluj-Napoca, 14.04.1984;
123. Marin, D., Doroga, H., Rotaru, Al., Marin, E., "Attitude thérapeutique en cas de fractures des maxillaires édentés", Archives de L'Union Médicale Balcanique, 1987, 1-4, pag. 352;
124. Martel, A.L.; Heid, O.; Slomczykowski, M.; Kerslake, R.; Nolte, "Assessment of 3D MRI images for computer assisted surgery", Comp. Aid. Surg. 3(1), 40-44, 1998;
125. McCarthy, G.J., "Plastic Surgery, vol.II, The Face, Part 1", W.B. Saunders Comp., 1990;
126. Michelet, F.X., Deymes, J., Dessus, B., "Osteosynthesis with Miniaturized Screwed Plates in Maxilo-Facial Surgery", J. of Max.-Fac. Surg., 1973, 1, pag., 79;
127. Microsoft Corporation, "Microsoft Developer's Network Library", 2003
128. Microsoft Corporation, "Microsoft Visual C++ User's Guide", 2003
129. Microsoft Corporation, "Microsoft Visual C++ Programmer's Guide", 2003
130. Microsoft Corporation, "Microsoft Windows Software Development Kit: Programmer's Reference"
131. Microsoft Corporation, "Microsoft's Introduction to OpenGL", Web Resource;
132. Mocanu, D.R., "Analiza experimentală a tensiunilor", Ed. Tehnică, București, 1977;
133. Moldoveanu, Florica, Racoviță, Zoea, Petrescu, Ș., Hera, G., Zaharia, M., "Grafica pe calculator", Editura Teora, București, 1996;
134. Moldoveanu, Florica, Hera, Gabriel "Programarea aplicațiilor Windows", Editura Teora, Bucuresti, 1995;
135. Mortenson, M.E., „Geometric Modelling”, John Wiley&Sons, USA, 1985;

136. Narkhede, Atul, Manocha, Dinesh, "Fast Polygon Triangulation based on Seidel's Algorithm", Dep.of Comp.Science, UNC Chapel Hill;
137. Naumann H.H., "Head and Neck Surgery", vol. 1, Georg Thieme Verlag, Stuttgart-New-York, 1995;
138. Newell, M.E., Newell, R.G., Sancha, T.L., "A solution to Hidden Surface Problem", Proceedings of the ACM National Conference, 1972;
139. Nicolau, Edm., "Analogie, modelare, simulare, cibernetică", Editura Științifică și Enciclopedică, București, 1977;
140. Nicolau, Edm., "Modelarea ca știință", Editura Științifică și enciclopedică, București, 1981;
141. Nishita, T., Eihachiro, N., "Continuous One Representation of Three-dimensional Objects Taking Account of Shadows and Inter-reflections", Computer Graphics, 1985, pag. 22-30.
142. Oden, I.T., Reddy, N.I., "An Introduction to The Mathematical Theory of Finite Elements", Wiley-Interscience, N.Y., 1982;
143. Olariu, V., Brătianu, C., "Modelarea numerică cu elemente finite", Ed. Tehnică, București, 1986;
144. Oprișiu, C., „Chirurgie buco-maxilo-facială”, Editura Didactică și Pedagogică, București, 1973;
145. Pascariu, I., "Elemente finite (Conceptii-Aplicații)", București, Ed. Militară, 1985;
146. Patel, M.F., Langdon, J.D., "Titanium Mesh (TiMesh) osteosynthesis: a fast and adaptable method of semi-rigid fixation", British Journal of Oral and Maxillo-Facial Surg., 1995, 29, pag. 316;
147. Pavlidis, T., "Integrated Region Growing and Edge Detection", IEEE Transactions on PAMI, March, 1990;
148. Pecheur, A., Reychler, H., "Evaluation à long terme du traitement fonctionnel des fractures condyliennes mandibulaires, Rev. Stom. Chir. Max., 1993, 94, 1, pag. 1;
149. Pedersen, W.G., "Oral surgery", W.B. Saunders Comp., 1988;
150. Peterson, L.J., Ellis, E.III, Hupp, J.R., Tucker, R.M., "Contemporary Oral and Maxillofacial Surgery", Second Ed., Mosby Comp., 1993;
151. Phong, B.T., "Illumination for Computer Generated Pictures", Comm. ACM, 1975, pag. 311-317;

152. Popescu, V., Burlibaşa, C., "Tehnici curente de chirurgie stomatologică", Editura Medicală, Bucureşti, 1961;
153. Popescu, V., Duţescu, N., Opreşiu, C., Timoşca, Gh., "Chirurgie buco-maxilo-facială", Editura Didactică şi Pedagogică, Bucureşti, 1967;
154. Przemieniecki, I.S., "Theory of Matrix Structural Analysis", McGraw-Hill, Sidney, 1968;
155. Rao, S.S., "The Finite Element Method in Engineering, Oxford, Ed. Pergamon Press, 1982;
156. Reddy, I.N., "An introduction to the Finite Element Method", McGraw-Hill, 1993;
157. Repciuc, E., "Mecanica articulaţiei mandibulare şi articuloarele", Stomatologia, Bucureşti, 1955, 1, pag. 64;
158. Richi, E., Carta, M., Bruzzone, A.A., Lonardo, P.M., Marinaro, E., Pastorino, A., "Experimental Analysis of internal rigid fixation osteosynthesis performed with titanium bone screw and plate systems", Journal of Cranio-Maxillofacial Surgery, 1996, pag. 53;
159. Rogers, D., "Procedural Elements for Computer Graphics", McGraw-Hill, New York, 1985;
160. Rotaru, Al., Doroga, H., Marin, D., Rotaru, M., "Notre experience dans les immobilisations monomaxillaires des fractures mandibulaires", Archives de L'Union Médicale Balcanique, 1987, pag. 357;
161. Rotaru, Al., Popa, M., Sima, V., "Mandibular osteosynthesis with a lower border far fixation: a simple, economical and efficient method", XIIth Congress E.A.C.-M.F.S., Haga, Olanda, 5-10 sept. 1994;
162. Rowe, N.L., Williams, J.L., "Maxillofacial injuries, vol.I", Churchill Livingstone, Edinburg-London, 1994;
163. Savii, G.G., "Bazele proiectării asistate de calculator. CAD", Editura Mirton, Timişoara, 1997;
164. Savii, G.G., "Grafica 3D pe calculatoare PC", Presa Universitară Română, 1997;
165. Savii, G.G., Luchin, M., "Modelare şi simulare", Editura Eurostampa, Timişoara, 2000;
166. Schultuz, C.R., "Facial injuries", Zear Book Medical Publishers, 1988;
167. Schwartz, M.D., Shires, C.T., Spencer, C.F., Jusser, C.W., "Principles of surgery", McGraw-Hill Inc., Health Profesions Division, 1994;

168. Sechrest, S., Greenberg, P.D., "A Visible Polygon Reconstruction Algorithm", ACM ianuarie 1982;
169. Segerling, L.I., Applied Finite Element Analysis", John Wiley, New York, 1976;
170. Seiller, A., "Mesures de la force de musculature de la bouche en rapport avec differents types d'occl., Orthodont. franç., 1956, 27, pag. 268;
171. Shantz, M., Lien, S., " Shading Bicubic Patches", Computer Graphics, pag. 189-196;
172. Shreiner, Dave "OpenGL Reference Manual: The Official Reference Document to OpenGL, Version 1.2 (3rd Edition)", Addison-Wesley Publishing;
173. Sid, Ahmed, M.A., Image Processing: Theory, Algorithms and Architecture, Prentice Hall, 1995;
174. Silicon Graphics Inc., "OpenGL's Command Structure", Web Resource
175. Silicon Graphics Inc., "Guide to OpenGL on Windows from Silicon Graphics", Web Resource
176. Souma, V.E., "Lecture Notes in Finite Element", CVEN 6525, Spring 1996;
177. Stanciu, Dan "Contribuții la studiul biomecanicii scheletului osos al feței umane", teza de doctorat, 2000, Universitatea "Politehnica" Timișoara;
178. Stanford Computer Graphics Laboratory, "Mathematical Methods for Computer Vision, Robotics and Graphics", Web Resource
179. Stanford Computer Graphics Laboratory, "Introduction to Computer Vision", Web Resource
180. Stanford Computer Graphics Laboratory, "Introduction to Computer Graphics", Web Resource
181. Stanford Computer Graphics Laboratory, "Computer Graphics: Geometric Modeling", Web Resource
182. Stanford Computer Graphics Laboratory, "Geometric Algorithm", Web Resource
183. Storey, E., "Growth and remodeling of bone and bones", Amer. J. Orthodont., 1972, 62, 2, pag. 142;
184. Tabbone, S., "Corner Detection Using Laplacian of Gaussian Operator", SCIA, Tromso, 1993;
185. Taylor, D.L., "Computer-Aided Design", Addison-Wesley, 1992;

186. Theiss, E., Grivu, O., “Cercetări antropologice asupra etajelor feței și importanța acestora în ortodontie”, *Stomatologia*, București, 1966, 4, pag. 317;
187. Timoșca, G., Burlibașa, C., “Chirurgie buco-maxilo-facială”, București, 1983;
188. Timoshenko, S.P., Woinowsky-Krieger, S., “Teoria plăcilor plane și curbe”, Editura Tehnică, București, 1968;
189. Tripa, P., Faur, N., “Metode teoretice și experimentale pentru determinarea stării de tensiune și deformație, Timișoara, Lito. U.T.T., 1994;
190. Tucatu, B., “Rapport d’harmonie dento-maxilaire”, *Orthodont. franç.*, 1967, pag. 119;
191. Tucker, R.M., Terry, C.B., White, P.R., Sickels, E.J., “Rigid fixation for maxillofacial surgery”, J.B. Lippincott company, 1991;
192. Ugural, A.C., Fenster, S.K., “Advanced Strength and Applied Elasticity”, Elsevier, 1987;
193. Umbaugh, S.E., “Computer Vision, and Image Processing”, Prentice Hall, 1998;
194. Vepstas, Linas “Programming with OpenGL”, AIXpert 1994;
195. Vertan, Constantin “Prelucrarea și analiza imaginilor”, 1999;
196. Visa, A., “Texture Classification and segmentation Based on Neural Network Methods”, Helsinki University of Technology, 1990;
197. Visarius, H.; Gong, J.; Scheer, C.; Haralamb, S.; Nolte, “Man-machine interfaces in computer assisted surgery”, *Comp. Aid. Surg.* 2(2), 102-107, 1997.
198. Voine, P.R., Voiculescu, C.V., Simion, Fl.P., “Introducere în mecanica solidului cu aplicații în inginerie”, București, Ed. Academiei, 1989;
199. Vlad Marin, colectiv “Grafica pe calculator in limbajele Pascal si C”, Editura Tehnica, Bucuresti, 1992
200. Wait, R., Mitchell, A.R., “Finite Element Analysis and Applications”, John Wiley, N.Y., 1985;
201. Waite, E.D., “Textbook of practical oral and maxillofacial surgery”, Lea and Febiger, 1987;
202. Walnum, Clayton “Windows 98 Programming Secrets”, IDG Worldwide, 1998
203. Warnock, J., “A Hidden Surface Algorithm for Computer Generated Half-Tone Picture”, Technical Report TR 4-15, University of Utah, iunie 1970;
204. Watt, Allan “3D Computer Graphics”, Third Edition, Addison Wesley, 2000
205. Watt, Allan, Watt, Mark, “Advanced Animation and Rendering Techniques : Theory and Practice”, 1992;

206. Watt, A., Policarpo, F., "The Computer Image", Addison – Wesley, 1998;
207. Weiller, K., Atherton, P., "Hidden Surface Removal Using Polygon Area Sorting", Computer Graphics, 11(2), pag. 214-222;
208. Wieslander, L., "The Effect of Force on Craniofacial Development", Amer. J. Orthodont., 1974, 65, 5, pag. 531;
209. Williams, L., "Casting Curved Shadows on Curved Surfaces", SIGGRAPH 78, pag. 270-274;
210. Woo, Mason "The OpenGL Programming Guide 3rd Edition"
211. Wright, Richard, Sweet, M. "OpenGL SuperBible, Second Edition";
212. Wylie, C., Romney, G.W., Evans, D.C., Erdahl, A.C., "Halftone Perspective Drawings by Computer", FJCC, Thompson Books, 1967;
213. Zaharia, M., "Structuri de date folosite în prelucrarea grafică a informației", Litografia Universității "Politehnica" București, 1994;
214. Zeid, I., "CAD/CAM Theory and Practice", McGraw-Hill, 1991;
215. Zienkiewicz, O.C., Taylor, R.L., "La méthode des éléments finits", AFNOR-Paris, 1991;
216. *** Contribuții la ameliorarea stării de sănătate a populației prin realizarea de aparate protetice, instrumentar chirurgical și truse de implantate pentru remedierea defectelor de schelet, Grant P.N.C.D.I., program Biotech 2002-2005, contract nr. 02-6-PA-346;
217. *** Studierea dinamicii mandibulei și modelarea de implantate pentru corectarea prin intervenții chirurgicale a ruperilor sale accidentale, Grant C.N.C.S.I.S., 2002-2004, tema 10, cod 11, contract nr. 33550 / 01.07.2003;
218. *** Modelarea, proiectarea și realizarea practică a unui sistem de implanturi medicale destinat chirurgiei maxilo-faciale și ortopedice, Grant C.N.C.S.I.S., 2002-2004, tema 11, cod 12, contract nr. 33550 / 01.07.2003;
219. *** Laborator de studii biomecanice și protezare a membrilor superioare și inferioare. *Modelarea cinematică și dinamică a membrilor superioare și inferioare*, Contract 5199/25.11.1999 Tema B1 Grant ANSTI;
220. *** Centru de modelare a protezării și intervențiilor chirurgicale asupra scheletului uman CM-PICUSU, Proiect BCUM inițiere, nr.2/1999;
221. *** Laborator de studii biomecanice și protezare a membrilor superioare și inferioare, *Cercetări privind modelarea aparatului locomotor și prelucrarea imaginilor radiografice*, Contract 5199/25.11.1999 Tema A1 Grant ANSTI

222. *** Modelarea aparatului locomotor și implanturi din materiale biocompatibile
Modelarea cinematică și dinamică a aparatului locomotor, Contract nr. 6153
Gr/19.10.2000 Tema B6
223. *** Laborator de studii biomecanice și protezare a membrilor superioare și
inferioare *Modelarea și determinarea stărilor de tensiune și deformare a
structurilor osoase prin metoda elementelor finite* Contract 5199/25.11.1999
Tema A1 Grant ANSTI
224. *** Centru de modelare a protezării și intervențiilor chirurgicale asupra
scheletului uman CM-PICUSU, Proiect BCUM, Cod CNCSIS 33, 2000-2002
225. *** Studierea dinamicii mandibulei și modelarea de implantate pentru corectarea
prin intervenții chirurgicale a ruperilor sale accidentale *Studiul dinamicii
mandibulei, pentru stabilirea prin metode numerice a comportării sale în timpul
unei masticații normale și în regim de șoc accidental*, Contract
33501/17.07.2002 Tema 14, cod CNCSIS 68;
226. *** Modelarea, proiectarea și realizarea practică a unui sistem de
implanturi medicale destinat chirurgiei maxilo-faciale și ortopedice
*Realizarea modelului pentru analiza cu element finit a unui set de
implanturi maxilo-faciale*, Grant C.N.C.S.I.S., 2002-2004, tema 15, cod 69,
contract nr. 33501 / 01.07.2003;
227. *** Studii numerice ale procedurilor restaurative directe și indirecte ale
aparatului dento-maxilar, cu materiale heterogene. *Modelarea prin metoda
elementelor finite a comportării mecanice a unor lucrări de protetică dentară*,
Nr. 3447/27.06.2002;

LISTA LUCRĂRILOR PUBLICATE DE AUTOR ÎN DOMENIUL TEZEI

- A1. Drăgulescu, D., Toth-Tașcău, M., Stănciulescu, V., Neagoe, C., “Metodă de identificare curentă a planelor autoascunse”, al XI-lea Simpozion Național de Roboți Industriali, Timișoara, Oct. 1992
- A2. Stănciulescu, V., “Analiza imaginilor scintigrafice achiziționate cu gamma camera”, Proiect de diplomă, Universitatea Politehnica Timișoara, 1996
- A3. Stănciulescu V., “Explorarea computerizată a interiorului unui organ”, Proiect de dizertație (master), Universitatea Politehnica Timișoara, 1997
- A4. Drăgulescu, D., Toth-Tașcău, M., Stănciulescu, V., “Computer-Aided Exploration inside Human Organs”, The 22nd National Conference on Medical Informatics, International Participation, MEDINF’99, Sibiu, 14-16 October 1999
- A5. Drăgulescu, D., Stănciulescu, V., Toth-Tașcău, M., “Method to rebuild the 3D configuration of human face's bones”, Buletin Științific al UPT, Automatic Control and Computer Science, IV CONTI 2000, Vol 45(59) 2000, No1, ISSN 1224-600X, p.57-62
- A6. Drăgulescu, D., Stănciulescu, V., Toth-Tașcău, M., “Rebuilding 3D Surfaces using Marching Cubes Algorithm for Maxillofacial Applications”, MEDINF 2001, Medical Informatics Europe, Healthcare Telematics Support in Transition Countries, Proceedings of MIE 2001 Special Topic Conference Bucharest, June 7 9, 2001, Publishing House EUROBIT Timișoara, p.91, ISBN 973-8181-91-7
- A7. Toth-Tașcău, M., Drăgulescu, D., Stănciulescu, V., “Analiza prin metoda Elementelor Finite a implantării scheletului facial”, Proceedings of the X-th Conference on Mechanical Vibrations with International participation, Vol. 3, Journal of Politehnica University Timisoara, Transactions on Mechanical Engineering, Tom 47(61), Supplement, Timisoara, 23-24 mai 2002, Editura Politehnica, ISSN 1224-6077, p.123-130
- A8. Stănciulescu, V., “3D Tomography. Computed Modelling of 2D Images. Results. Advantages”, Acustica și vibrațiilor structurilor mecanice. Biomecanica: Proceedings of VII-th Symposium, Timișoara, May, 2003