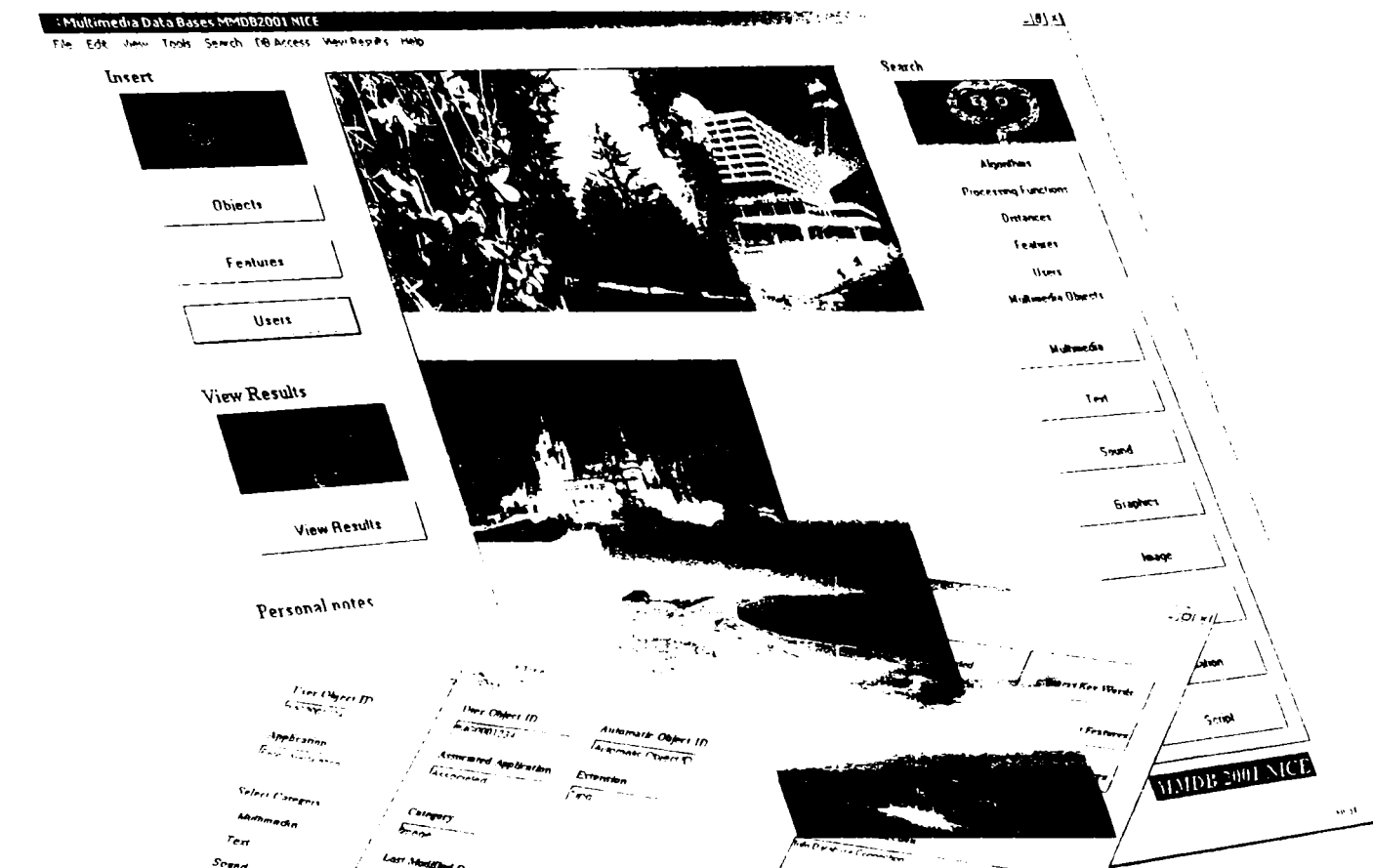


41. Obp
29 Lit:

litechnică" din Timișoara
tronică și Telecomunicații

Departamentul Comunicații



Contribuții la dezvoltarea aplicațiilor multimedia ce folosesc baze de date cu căutare bazată pe conținut

Teză de doctorat

Conducător științific
Prof.dr.ing. Toma I. Corneliu

Autor
Șl.ing. Mocofan Muguraș Daniel

2003

Cuprins

BIBLIOTECA CENTRALĂ
UNIVERSITATEA "POLITEHNICA"
TIMIȘOARA

UNIVERSITATEA "POLITEHNICA"
TIMIȘOARA
BIBLIOTECA CENTRALĂ
nr. volum 2/1999
După 1999

CAPITOLUL 1.	5
INTRODUCERE	5
1.1. Evoluția și stadiul actual al dezvoltării tehnicilor multimedia	8
1.2. Evoluția aplicațiilor multimedia și a suportului hardware	10
CAPITOLUL 2.	11
MEDIILE DE DEZVOLTARE ȘI APLICAȚIILE MULTIMEDIA ÎN CONTEXTUL NOILOR TEHNICI DE PROGRAMARE	11
2.1. Medii de dezvoltare ale aplicațiilor multimedia	11
2.1.1. Studiu comparativ privind mediile de dezvoltare a paginilor WWW	11
2.1.2. Studiu comparativ privind mediile de dezvoltare a aplicațiilor multimedia ce rulează de pe CD-ROM	12
2.1.3. Studiu comparativ privind mediile de editare video neliniară	14
2.1.4. Comparații și concluzii	16
2.2. Programarea aplicațiilor multimedia din perspectiva noilor tehnici de programare	17
2.2.1. Programare Orientată pe Obiecte	17
2.2.2. Noutăți aduse de limbajul Java în programarea orientată pe obiecte	20
2.3. Structurarea modului de programare conform conceptului Open Flexible Language (OFL), de hipergeneralizare în programare	22
2.3.1. Concepte și principii fundamentale	22
2.3.2. Conceptul relații (Concept-relations)	23
2.3.3. Conceptul descriere (concept-description)	25
2.3.4. Conceptul limbaj (Concept-language)	28
2.4. Programarea orientată pe aspecte (AOP) - Aspect J	30
2.4.1. Semantica Aspect J	30
2.4.2. Recomandări – Advice	31
2.4.3. Concluzii privind implementarea aspectelor în aplicațiile multimedia	32
2.5. JPVM - Java Parallel Virtual Machine	32
2.6. Concluzii și contribuții	33
CAPITOLUL 3.	35
MODELUL PROPUȘ PENTRU APLICAȚIILE MULTIMEDIA	35
3.1. Cerințe actuale pentru un mediu de dezvoltare pentru aplicațiile multimedia	35
3.2. Structura modelului propus pentru aplicații multimedia	37
3.3. Modelul descriptiv al obiectelor multimedia: <i>Objects</i>	39
3.4. Funcții și procesări	47
3.4.1. Procesări temporale: <i>Time_Line</i>	47
3.4.2. Funcții de interacțiune cu utilizatorul: <i>GUI, Events</i>	48
3.4.3. Procesări obiecte multimedia: <i>Processing_Functions, Transitions</i>	49
3.5. Operații de intrare – ieșire	50
3.5.1. Operații de intrare – ieșire: <i>Input_Output_Operations</i>	50
3.5.2. Conectarea în rețea (biblioteca <i>Network</i>)	50

3.5.3. Biblioteci de funcții și metode pentru dezvoltarea aplicațiilor de tip <i>Client/Server</i>	50
3.5.4. Gestionarea utilizatorilor – <i>Users</i>	51
3.6. Tehnologii integrate în cadrul modelului	51
3.6.1. <i>DDE</i>	51
3.6.2. <i>DLL</i>	52
3.6.3. <i>OLE</i>	52
3.6.4. Generalizarea modului de conectare la o bază de date (<i>G-ODBC</i>)	53
3.6.5. Funcții de control ale echipamentelor multimedia <i>MCI</i>	54
3.7. Facilități suplimentare integrate modelului	56
3.7.1. Funcții ce asigură calitatea unei aplicații: <i>QoS</i>	56
3.7.2. Funcții de căutare de obiecte: <i>Browse</i>	59
3.8. Concluzii și contribuții	59
3.8.1. Principalele avantaje ale modelului propus	59
3.8.2. Un nou mod de generalizare a aplicațiilor multimedia prin accesarea obiectelor utilizate din baze de date multimedia	60
CAPITOLUL 4.	63
BAZE DE DATE MULTIMEDIA	63
4.1. Evoluția bazelor de date.....	63
4.2. Situația actuală a bazelor de date	67
4.2.1. Baze de date pentru publicații multimedia	67
4.2.2. Access	68
4.2.3. DB2 – IBM	69
4.2.4. Oracle <i>interMedia</i>	69
4.2.5. Direcții de dezvoltare în domeniul bazelor de date clasice	71
4.3. Baze de date multimedia.....	73
4.3.1. Evoluția bazelor de date spre aplicații multimedia.....	73
4.3.2. Caracteristicile bazelor de date multimedia.....	74
4.3.3. Multimedia și bazele de date orientate pe obiecte	80
4.3.4. Moduri de creștere a performanțelor unei baze de date.....	82
4.3.5. Sisteme de gestionare a bazelor de date distribuite	83
4.4. Indexarea bazelor de date multimedia	84
4.4.1. Introducere	84
4.4.2. Structuri de indexare pentru BLOBs	85
4.4.3. Indexarea multidimensională pentru obiectele multimedia.....	86
4.4.4. Strategii alternative de stocare	88
4.4.5. Distanțe	90
4.4.6. Concluzii.....	91
4.5. Căutarea bazată pe conținut în bazele de date multimedia	91
4.5.1. Introducere	91
4.5.2. Modelarea obiectelor multimedia și interogarea bazată pe similaritate	92
4.5.3. Căutarea bazată pe conținut – abordare din punct de vedere al similarității modului de interpretare a unui obiect multimedia.....	93
4.5.4. Concluzii.....	95
4.6. Interogarea bazelor de date multimedia.....	95
4.6.1. Introducere	95
4.6.2. Accesul și stocarea.....	95
4.6.3. Dimensiunile procesului de interogare	96
4.6.4. Concluzii.....	97

CAPITOLUL 5. 98**CONTRIBUȚII LA INDEXAREA, INTEROGAREA ȘI CĂUTAREA IMAGINILOR ȘI A SECVENȚELOR VIDEO ÎN BAZE DE DATE MULTIMEDIA, ÎN CADRUL STUDIOURILOR DE TELEVIZIUNE..... 98****A. INDEXAREA, INTEROGAREA ȘI CĂUTAREA IMAGINILOR ÎN BAZE DE DATE MULTIMEDIA 98**

5.1. Caracteristici globale utilizabile la indexarea bazelor de imagini	98
5.1.1. Atributele stohastice ale unei imagini	98
5.1.2. Momentele și media spațială a unei regiuni.....	99
5.1.3. Matricea de coocurență	102
5.1.4. Lărgimea plajelor nivelurilor de gri sau isosegmentele	107
5.1.5. Spectrul	107
5.1.6. Contrastul asociat unei regiuni.....	109
5.1.7. Atribute de suprafață.....	110
5.2. Caracteristici locale ale imaginilor propuse pentru utilizarea la indexarea bazelor de imagini	113
5.2.1. Introducere	113
5.2.2. Texturi definite prin primitive	114
5.2.3. Texturi ce au un model structural de plasare a texelilor	115
5.2.4. Texelii	119
5.2.5. Dependența spațială a nivelelor de gri (SGLD).....	120
5.2.6. Energia texturii.....	121
5.2.7. Filtrele Gabor	123
5.2.8. Experimentări proprii referitoare la aplicații ale filtrelor Gabor și ale caracteristicilor texturilor.....	135
5.2.9. Caracteristici ale texturilor invariante la rotație, translație și scalare	136
5.2.10. Generarea texturilor folosind modele autoregresive.....	142
5.2.11. Gradientul texturii	145
5.2.12. Recunoașterea primitivelor texturii.....	150
5.3. Caracteristicile obiectelor	151
5.4. Tehnici de căutare în bazele de imagini.....	152
5.4.1. Modurile de utilizare ale caracteristicilor imaginilor folosite la căutare	152
5.4.2. Similaritatea imaginilor bazată pe proprietățile globale ale imaginilor.....	154
5.4.3. Evaluarea similarității folosind mai multe caracteristici	157
5.4.4. Căutarea în baze de date utilizând mai multe categorii de caracteristici (caracteristicilor globale, caracteristici stohastice ale imaginilor, caracteristici ale matricii de coocurență, descompuneri pe baza filtrelor Gabor)	158
Căutarea în baza de texturi folosind descompuneri realizate cu filtre Gabor	162
Sistem de căutarea în baza de imagini bazat pe similaritatea dintre texturi folosind descompuneri realizate cu filtre Gabor și clasificatori reprezentați de rețele neuronale	164
Metodă de căutare ce folosește pentru discriminare o tehnică bazată pe praguri	167
Sisteme ce interacționează cu utilizatorul.....	173
Forma obiectelor	175
5.4.5. Concluzii:.....	176

B. INDEXAREA, INTEROGAREA ȘI CĂUTAREA SECVENȚELOR VIDEO ÎN BAZE DE DATE MULTIMEDIA	177
5.5. Modelul materialelor video.....	177
5.5.1. Proprietățile secvențelor video	177
5.5.2. Nivelele semantice ce alcătuiesc modelul materialelor video	177
5.6. Tehnici de indexare a secvențelor video.....	179
5.7. Tehnici de căutare a secvențelor video	180
5.8. Utilizarea tehnicilor multimedia în televiziune	183
5.8.1. Tipuri de informații utilizate într-un studio de televiziune.....	183
5.8.2. Necesitatea introducerii tehnicilor multimedia într-un studio de televiziune.....	183
5.9. Concluzii și soluții propuse pentru îmbunătățirea activităților dintr-un post de televiziune	186
CAPITOLUL 6.	188
CONTRIBUȚII LA IMPLEMENTAREA BAZELOR DE DATE MULTIMEDIA.....	188
6.1. Analiza și proiectarea unei baze de date multimedia.....	188
6.1.1. Specificarea cerințelor	189
6.1.2. Contribuții la analiza orientată pe obiecte din punctul de vedere ale utilizatorului ...	189
6.1.3. Contribuții la analiza orientată pe obiecte din punctul de vedere al aspectului conceptual	191
6.2. Integrarea bazei de date multimedia implementate în cadrul modelului de aplicație multimedia propus.....	191
6.3. Modul de organizare al bazei de date propuse.....	193
6.3.1. Modul de indexare al bazei de date propuse.....	193
6.3.2. Algoritmii de căutare propuși	196
6.3.3. Implementare propusă pentru translatarea bazelor de date relaționale existente în bază de date cu facilități de manipulare a obiectelor multimedia.	200
6.3.4. Modalitate de mărire a vitezei de căutare	203
6.3.5. Implementarea software.....	203
6.3.6. Detalii referitoare la informațiile stocate în baza de imagini și reprezentarea caracteristicilor descriptive	208
CAPITOLUL 7.	211
CONTRIBUȚII PERSONALE ȘI CONCLUZII	211
Direcții viitoare de cercetare.....	214
BIBLIOGRAFIE	215

Capitolul 1.

Introducere

Mediile multimedia permit ca un calculator să interacționeze cu utilizatorul într-un mod plăcut și interesant, creând noi familii de produse, care conțin toate tipurile de informații, text, sunet, imagini, animații și secvențe video. Aceste procesări globale de informație, au făcut ca lumea afacerilor, a artei, a științei și tehnicii și a proceselor de producție să le considere, să le perceapă ca și o „revoluție informațională”. Scopul final al acesteia este de a avea toate informațiile digitizate și stocate pe un calculator conectat la o rețea globală de comunicare, care să asigure legătura informațională între utilizatori aflați în diverse părți ale globului. Trecerea de la variantele tradiționale de stocare a informației, hârtie, casete audio, video și film, a determinat o explozie tehnologică și în domeniul echipamentului. În acest domeniu au fost cunoscute cele mai rapide progrese tehnologice și au fost dezvoltate tehnologii din ce în ce mai performante.

Lucrarea are ca scop dezvoltarea de noi modele de aplicații multimedia, de noi algoritmi și tehnici multimedia, ce pot îmbunătăți activitățile de implementare a produselor multimedia. Finalitatea acesteia este implementarea unei baze de date multimedia (folosind noul model propus de aplicație multimedia) cu un grad mare de utilitate în cadrul unui post de televiziune prin utilizarea de tehnici de indexare și căutare bazate pe conținut.

Tehnicile multimedia își găsesc utilitatea pornind de la probleme de primire, gestionare și transmitere de informații referitoare la știri și până la prelucrări complexe pentru diverse medii informaționale (prelucrări complexe de imagini, compresii video, recunoaștere vocală, etc.) sau probleme de indexare, stocare și căutare în baze de date.

Evoluția aplicațiilor multimedia, modul în care se pot realiza aplicații multimedia folosind mediile de implementare ce există în prezent și contextul actual al tehnicilor de programare le-am prezentat sintetic în capitolul 2, „Mediile de dezvoltare și aplicațiile multimedia în contextul noilor tehnici de programare”. Am experimentat și dezvoltat aplicații multimedia folosind mai multe medii de dezvoltare pentru principalele categorii de aplicații multimedia (WWW, CD-ROM, Video). Experiența câștigată de-a lungul anilor mi-a permis să realizez o comparație a performanțelor acestora, iar concluziile rezultate m-au determinat să-mi concentrez atenția spre propunerea unui nou model de aplicație multimedia. Am sintetizat pentru acest domeniu și direcțiile viitoare de dezvoltare.

O atenție deosebită, am acordat noilor tehnici de programare orientate pe obiecte și a modului în care acestea lucrează cu obiecte multimedia. Se dezvoltă din ce în ce mai mult și conceptul programării orientate pe aspecte. Utilitatea acestui concept de programare se regăsește în special în cadrul aplicațiilor multimedia, ce pot fi mai bine structurate ca funcționare printr-o abordare orientată pe aspecte a proceselor cu restricții temporale.

Am implementat algoritmi de prelucrare a imaginilor ce rulează pe o rețea eterogenă de calculatoare. Astfel, sarcina de prelucrare, poate fi distribuită în rețea, fiecare dintre calculatoarele legate în rețea poate prelua o parte din sarcină, rezultatele fiind integrate de către

calculatorul ce a trimis operația de prelucrare. Rezultatele obținute, folosind o astfel de distribuire a calculelor în funcție de resurse existente, au demonstrat eficacitatea metodei prin exploatarea eficientă resurselor existente (mai multe calculatoare cu performanțe modeste, conectate în rețea pot depăși un calculator puternic și foarte scump) și scurtarea timpului de execuție. Au fost realizate mai multe teste privind eficacitatea metodei în cazul prelucrărilor de imagini. A fost verificat algoritmul pentru prelucrarea simultană pe mai multe calculatoare a mai multor imagini, dar și varianta de prelucrare paralelă a aceluiași imagini pe mai multe calculatoare.

Conceptele actuale de hipergeneralizare a programării, ce au fost dezvoltate în cadrul Laboratorului de Informatică, Semnale și Sisteme din cadrul Universității Sophia-Antipolis, Nice, Franța, le-am folosit în cadrul implementării modelului de aplicație multimedia propusă și a bazei de date multimedia cu posibilități de căutare bazate pe conținut. Am beneficiat de o foarte bună colaborare cu persoanele implicate în cercetare din cadrul acestui laborator.

„Modelul propus pentru aplicațiile multimedia” este titlul capitolului trei, în care am descris modelul propus pentru aplicațiile multimedia. Modelul propus prezintă:

- Optimizări din punct de vedere al structurării implementării prin orientarea pe obiecte a aplicațiilor. Prin definirea de noi clase, metode, obiecte și parametri s-a asigurat creșterea performanțelor aplicației ce lucrează cu obiecte multimedia de tip fluxuri de date condiționate temporal.
- Concepția modulară permite extinderi ulterioare ale aplicațiilor prin re folosirea unor părți din aplicație.
- Îmbunătățirea bibliotecilor de funcții și transferabilitatea acestora între aplicații, prin folosirea modelului propus, permite ca utilizatorii să poată contribui la îmbunătățirea performanțelor și la extinderea mediilor de dezvoltare ale aplicațiilor multimedia existente.
- Asigurarea unui anumit grad de calitate pentru serviciile oferite de aplicație și informarea utilizatorului din timp asupra problemelor ce pot apărea, existând chiar și posibilitatea sugerării unor soluții de remediere. Modelul propus permite folosirea unor biblioteci de funcții care să verifice înainte de rularea unei aplicații multimedia performanțele hardware ale sistemului.
- Includerea bibliotecilor de funcții și metode ce permit comunicarea cu bazele de date. Acest lucru face posibilă dezvoltarea aplicațiilor multimedia cu accesarea obiectelor direct din baze de date. Dezvoltarea unei aplicații multimedia se poate face prin introducerea în baza de date de noi obiecte, ne fiind necesară refacerea aplicației, rescrieri de cod sau alte complicații de acest fel. Modelul permite dezvoltarea de aplicații multimedia la care numărul de pagini este generat dinamic prin intermediul bazelor de date și nu este implementat în mod fix.
- Realizarea de aplicații care să lucreze în colaborare cu mai utilizatori simultan prin tehnologia client/server, inclusă în cadrul modelului.

Capitolul patru „Baze de date multimedia”, descrie evoluția bazelor de date până în prezent și se concentrează asupra bazelor de date multimedia orientate pe obiecte ce acceptă obiecte multimedia de tip flux de date cu constrângeri temporale. Am prezentat comparativ mai multe proiecte ce au dezvoltat baze de date multimedia, dar și situația celor mai cunoscute produse comerciale. Sunt prezentate:

- variantele clasice de indexare în bazele de date,
- avantajele ce rezultă din folosirea indexării bazate pe conținut în cazul utilizării de obiecte multimedia.
- o sinteză a avantajelor folosirii tehnicilor de programare orientate pe obiecte în cadrul bazelor de date multimedia.

Proprietățile corect alese ale obiectelor multimedia vor defini eficiența algoritmilor de căutare.

„Contribuții la indexarea, interogarea și căutarea imaginilor și a secvențelor video în baze de date multimedia, în cadrul studiourilor de televiziune” este titlul capitolului cinci ce prezintă tehnici de indexare, interogare și căutare în baze de date multimedia.

Am prezentat într-un subcapitol separat tehnicile folosite pentru indexarea imaginilor și căutarea bazată pe conținut. Aici, în mai multe paragrafe, sunt definite cele mai eficiente categorii de caracteristici ce sunt folosite în indexarea imaginilor. Din categoria caracteristicilor globale utilizabile la indexarea imaginilor recomand, ca rezultat al experimentărilor efectuate, o serie de proprietăți stohastice, momente de diverse ordine, proprietăți ale matricii de coocurență, isosegmentele, analiza spectrală a imaginii, contrastul asociat pe regiuni și atribute de suprafață.

În paragraful dedicat caracteristicilor locale ale imaginilor (5.2), am demonstrat eficiența folosirii ca element central în analiza imaginilor a tehnicilor bazate pe texturii. În cadrul paragrafului, sunt prezentate moduri de definire a texturilor în strânsă legătură cu modul de utilizare al acestora în cadrul unei baze de date multimedia. Sunt date modalități de generare și stocare a texturilor folosind metode structurale (5.2.2 – 5.2.5) dar și metode de generare pornind de la modele autoregresive (5.2.11). Aceste metode de generare a texturilor, oferă un grad mare de aplicabilitate în aplicațiile multimedia, prin faptul ca pornind de la un număr foarte redus de parametri se pot genera foarte multe tipuri de texturi, spațiul alocat stocării parametrilor fiind redus iar algoritmul de generare nu necesită un volum deosebit de calcule.

Sunt prezentate și aspecte ale analizei texturilor folosind metode din domeniul frecvență (5.2.7 – 5.2.8). Rezultate foarte bune am obținut în prelucrarea texturilor prin folosirea filtrelor Gabor. Am detaliat modul de proiectare a unui filtru Gabor, explicând majoritatea problemelor ce apar la definirea acestuia și în special la anticiparea eficienței acestuia pentru o problemă concretă pe care trebuie să o rezolve. Deoarece, am studiat în profunzime problema analizei imaginilor bazată pe texturi, am mai prezentat câteva aplicații de procesări de imagini bazate pe texturi, unde rezultatele experimentale obținute au fost deosebite prin combinare de mai multe tehnici de procesare: calcul de caracteristici folosind metode din domeniul frecvență sau caracteristici ale matricii de coocurență împreună cu morfologie matematică (5.2.9).

În imaginile reale, texturile se află într-o foarte mare varietate de orientări și la diverse rezoluții. Uneori este importantă detectarea texturii, și atunci se folosesc caracteristici invariante la rotație, translație și scalare (în 5.2.10 sunt descriși operatorii Circular-Mellin), dar se regăsesc și aplicații în care este importantă o anumită orientare a unei texturi, caz în care caracteristicile trebuie să fie foarte sensibile la orientare. În aplicații multimedia, problemele ce apar datorită existenței unui gradient sunt foarte frecvente. Am acordat un spațiu mai extins problemei de evaluare a orientării unei texturi. Am demonstrat existența unei legături de similaritate între rezultate obținute prin metode de descompunere folosind un set de filtre Gabor și folosirea proprietăților matricii de coocurență calculată pe ferestre alunecătoare de diverse dimensiuni.

Pentru descrierea obiectelor din imagini am folosit doar un set relativ redus de caracteristici. Se cunoaște complexitatea descrierii de obiecte foarte complexe, iar utilitatea în cazul unei baze de date multimedia poate fi atenuată de volumul mare de calcule.

Paragraful 5.4 „Tehnici de căutare în bazele de imagini” realizează o sinteză a modului de utilizare, în cadrul unei baze de date multimedia, a caracteristicilor prezentate în subparagrafele anterioare.

Un alt subcapitol l-am acordat indexării secvențelor video folosind în special tehnici de indexare automată după un model semantic de descriere a emisiunii conținute în acea secvență video. Autorul propune o metodă de segmentare automată a secvențelor video ce se bazează pe descrierea semantică și temporală a unei secvențe video.

Am avut un contact direct cu mai multe studiouri de televiziune locale și naționale, iar ca urmare am reușit identificarea problemelor ce există în cadrul acestora. Soluții de optimizarea a activităților sunt propuse prin intermediul introducerii tehnicilor multimedia în televiziune.

Implementarea software a bazei de date multimedia și algoritmi propuși de indexare și căutare bazați pe conținut, cu aplicabilitate în studiourile de televiziune sunt incluși în capitolul șase „Contribuții la implementarea bazelor de date multimedia”. Aici am descris pe larg modul în care această aplicație multimedia de tip baze de date respectă modelul propus în capitolul trei și avantajele rezultate în urma acestei implementări. Am descris implementarea orientată pe obiecte, clasele, metodele și funcțiile utilizate în concordanță cu modelul OFL de hipergeneralizării prezentat în capitolul doi. Un subcapitol l-am acordat performanțelor și utilității practice a acestei baze de date.

Explozia recentă a aplicațiilor pe Internet ce folosesc toate tipurile de medii, a avut ca rezultat dezvoltarea sistemelor de baze de date ce pot gestiona obiecte multimedia. Principala provocare a fost rezolvarea acestei gestiuni pornind de la conținutul obiectelor multimedia, care devin din ce în ce mai complexe, ce nu sunt întotdeauna bine structurate, au mare diversitate de formate și necesită asocierea unor informații suplimentare legate de aplicația care le-a generat.

Realizarea unor sisteme de baze de date distribuite și accesabile prin Internet care să rezolve problemele de transfer de informații între studiourile de televiziune (studiouri independente, sau filiale ale unui post central) este un domeniu ce necesită acordarea de atenție în continuare, mai ales din cauza problemelor complexe ce apar și care se situează în mai multe sfere de activitate.

Capitolul șapte prezintă o sinteză a tezei, a contribuțiilor personale și a rezultatelor obținute. Am descris și posibilitățile ulterioare de dezvoltare ale bazei de date implementate și a modului în care aplicațiile multimedia își vor continua dezvoltarea în cadrul posturilor de televiziune.

1.1. Evoluția și stadiul actual al dezvoltării tehnicilor multimedia

Într-o lume în continuă evoluție marcată de schimbări profunde, se poate distinge tendința generală de a edifica o societate informațională.

Ritmul schimbării se accelerează pe măsură ce tehnologia comunicațiilor înlătură privilegiul informațional, deschizând noi canale de informație cu o capacitate mai cuprinzătoare și un grad mai mare de sofisticare, anulând practic distanța dintre emițător și receptor, sporind viteza fluxului informațional.

Astăzi un sistem modern de prelucrare a informațiilor poate trata practic orice informație înregistrată pe un suport magnetic sau optic: date numerice, text, grafică, sunete, voce și imagini video. Nu mai este necesar ca în vederea introducerii documentelor într-o bază de date să fie codificate și structurate într-un fel anume, ele putând fi preluate, ca atare, prin scanare sau înregistrare audio sau video atașându-se doar un set de atribute, un index cu informații pentru regăsire, consultare și modificare. Asistăm astfel la un salt revoluționar în prelucrarea informațiilor.

Multimedia, un concept la modă dar în același timp controversat, în curs de clarificare, în curs de standardizare, dar mai ales în curs de răspândire, va schimba în curând societatea, la fel cum a făcut ziarul, telefonul și televizorul în perioade anterioare.

Multimedia combină trei mari informații ale secolului: telecomunicații, tehnică de calcul și audiovizualul. În anii '70, multimedia era doar un cuvânt care însemna audiovizual. În etapa actuală multimedia a devenit o tehnologie care a revoluționat domeniul calculatoarelor. Pornind de la ideea că omul însuși, prin esența lui, este "multimedia", aplicațiile de bază ale acestei tehnologii se canalizează spre informarea și formarea sa.

Multimedia reprezintă ansamblul tehnicilor de creare, stocare, transmitere și exploatare coordonată a informației sub formă de date numerice, text, grafică, sunete, voce, imagini și

video. În ultimii ani, o serie de factori au pregătit terenul pentru apariția conceptului de "Multimedia". Printre aceștia putem enumera :

- răspândirea calculatoarelor personale - se estimează ca astăzi în lume sunt instalate peste câteva sute de milioane de PC-uri;
- răspândirea unor interfețe grafice cu utilizatorul - au atras din ce în ce mai mulți utilizatori, arătând ce impact poate avea includerea imaginilor și cuvintelor în materialele care conțin cifre;
- răspândirea și standardizarea plăcilor de sunet și video - problema majoră ce a rămas un timp nerezolvată dar în acest moment există din ce în ce mai multe soluții disponibile pe piață.

Startul a fost dat de apariția unei tehnologii revoluționare și anume CD-ROM-ul, răspândit astăzi prin discurile cu muzica și care permite memorarea unei enciclopedii în multe volume pe un disc ce poate fi ținut în palmă. O secvență animată în timp real (25 imagini/sec). fiecare imagine are 800 x 600 puncte, în 16 milioane de culori, necesită circa 36 Mocteți / sec. Deci, un film de circa 20 minute are nevoie de 43,2 Gocteți. Pentru ca o aplicație de 20 de minute, să se afle pe un suport de tip CD-ROM trebuie adus volumul la maximum 700 Mb, printr-un sistem de comprimare. Deocamdată suportul amovibil care respectă o mare parte din aceste imperative este discul compact, cu o viteză de transfer între 150 și 7800 Kb/sec. A apărut astfel o nouă industrie reprezentată de producătorii de "cărți electronice" - discuri multimedia care acoperă cele mai diverse domenii : Mamiferele, Beethoven, Biblia, Cinefilia. Discul Guinness al recordurilor, Atlasul enciclopedic, etc. În general în aceste cărți pe lângă text există imagini statice, animație și sunet. Foarte impresionant este faptul că informațiile sunt structurate și se pot regăsi după diferite criterii.

Saltul este spectaculos, reprezentând un mod cu totul nou de receptare a informației. Aplicațiile acestei noi tehnologii pot fi limitate doar de lipsa de fantezie a realizatorilor și, deocamdată, de absența unor standarde unanim acceptate.

Voyager, o companie cu 60 de angajați, a realizat primul film ce a apărut pe CD-ROM, fiind vorba de " A Hard Day's Night", un film alb-negru înregistrat de formația Beatles în 1964. Pentru a stoca 90 de minute de film pe un CD-ROM frecvența de cadre pe secundă a fost redusă la 12 și pentru mai multe cadre au fost stocate doar modificările, obținându-se un raport de comprimare 4:1. Fără a ajunge la calitatea imaginii de televiziune, filmul arăta acceptabil pe ecranul unui calculator.

Compania Lotus a avut prima idee de a dota o versiune a vestitului său program de calcul tabelar cu un "help" multimedia, vorbit și animat. De fapt, tot produsul este un CD-ROM ce înlocuiește manualul de utilizare.

Teleconferințele multimedia "video desktop" permit utilizatorilor să se vadă unii pe alții și în același timp să își trimită documente multimedia în timp real.

Iată și câteva domenii de aplicații avute în vedere în viitorul apropiat pentru dezvoltări intensive: comerțul electronic, jocuri interactive, învățământ la distanță, arhitectura și îngrijirea sănătății. De asemenea, deosebit de spectaculoasă este realizarea de prezentări ale unor muzee pe echipamente multimedia. În acest fel, Muzeul Louvru de exemplu, se poate comprima pe câteva CD-uri, sau difuzate prin Internet, astfel încât să fie puse fără probleme la dispoziția publicului aflat la mii de kilometri distanță.

1.2. Evoluția aplicațiilor multimedia și a suportului hardware

Tendința actuală în domeniul echipamentelor este de a crea carduri specializate de captură audio-video de foarte bună calitate, care să poată fi montate pe un calculator cu performanțe medii. Pentru aceste plăci se asigură o compresie și decompresie hardware a informației, astfel ca puterea de calcul a procesorului să nu influențeze procesul de captură.

Pentru procesările ulterioare ale materialului captat se folosește procesorul. Utilizarea acestui mod de structurare a resurselor hardware a determinat ca prețul plăcilor de achiziție să scadă foarte mult, fiind accesibile utilizatorilor cu resurse financiare modeste. Este posibilă o prelucrare profesională, chiar dacă calculatorul are performanțe medii. Procesarea va dura mai mult, nu va fi în timp real dar este posibil de realizat.

Editarea neliniară oferă avantajul unei foarte mari libertăți de stocare a datelor într-o formă digitală pe orice tip de mediu (HDD, CD-RW, MO, casete digitale) și folosirea tehnicilor de procesare digitală folosind procesoare existente în calculatoarele obișnuite (nu pe procesoare specializate foarte scumpe). Imaginile pot provenii de pe peliculă de film sau pot fi în format video. De asemenea, se pot integra foarte ușor și imagini sintetizate pe calculator. Din punct de vedere estetic procesările sunt aceleași, diferă însă modul în care acestea sunt realizate, avantajul major fiind diversitatea mare de procesări ce pot fi folosite, numărul mare de formate de date acceptate și diversele variante de stocare a datelor [1].

În toate tipurile de editări scopul este de a alege o secvență de material, dorită de realizator și de a o poziționa acolo unde ea își atinge scopul artistic. Evaluarea eficienței unui sistem de editare neliniară se face prin prisma ușurinței în utilizare, a vitezei de lucru, a costului echipamentului și a calității materialului realizat din punct de vedere tehnic și artistic. Deci, pentru a avea succes, noile sisteme trebuie să ofere o cât mai mare flexibilitate a sistemelor.

Capitolul 2.

Mediile de dezvoltare și aplicațiile multimedia în contextul noilor tehnici de programare

2.1. Medii de dezvoltare ale aplicațiilor multimedia

2.1.1. Studiu comparativ privind mediile de dezvoltare a paginilor WWW

Macromedia Dreamweaver UltraDev este un program profesional pentru construirea aplicațiilor Web și de asemenea este un editor profesional pentru crearea și managementul site-urilor și a paginilor Web [2], [3]. UltraDev este un program pe deplin configurabil. Utilizatorul își poate crea propriile obiecte, comenzi, server-behaviours-uri, se pot modifica meniuri și chiar se pot scrie scripturi noi pentru a extinde UltraDev-ul cu noi acțiuni, comportamente și proprietăți.

Facilități oferite de Dreamweaver UltraDev:

- Dreamweaver UltraDev are suport ASP (Active Server Pages) pentru Microsoft și suport JSP (Java Server Pages) pentru Sun-uri, PHP, MySQL. Aceste tehnologii permit utilizatorului să creeze pagini dinamice prin executarea scripturilor direct pe serverul de Web.
- Nu este preferat nici un browser în mod special, indiferent de browser afișarea va fi aceeași.
- Programul lucrează cu baze de date, iar manipulările de date sunt oferite de către “serverul de comportamente”.
- Utilizatorii pot să creeze și să apeleze foarte ușor la librării de șabloane, script-uri, elemente și “comportamente”, salvând astfel o cantitate considerabilă de timp la producerea diferitelor site-uri.
- Dreamweaver folosește CSS (Cascading Style Sheets) și efecte HTML dinamice compatibile cu browsere de varianta 4.

NetObject Fusion este o unealtă puternică pentru realizarea unor site-uri Web performante. Acest mediu de dezvoltare a aplicațiilor multimedia se remarcă prin facilitățile sale deosebite în automatizarea producției de site-uri ce au un număr foarte mare de pagini și sunt organizate într-o structură de tip arborescent (este permisă orice tip de navigare, în schimb structura arborescentă de navigare se generează automat.).

Asemenea cazuri (site-uri) sunt caracteristice marilor firme, ce au un număr mare de produse și servicii, iar organizarea site-ului se face pe baza unui șablon ce definește stilul și grafica generală a aplicației.

Facilitățile oferite de NetObject Fusion [4], [5]:

- Sistem grafic interactiv și intuitiv de realizare a structurii site-ului, facil de utilizat de persoane cu diverse grade de experiență în domeniul producției de pagini Web.
- Permite dezvoltarea unei organizări ierarhice a site-ului, sub formă arborescentă, structură agreată de majoritatea persoanelor ce accesează site-urile, datorită intuitivității oferite de sistemul de navigare.
- Permite generarea automată a sistemului de navigare pornind de la structura site-ului ce se realizează într-o primă etapă.
- Generează butoanele și bannerele automat pornind de la un stil predefinit de un model de către implementator.
- Permite inserarea unui foarte mare număr de tipuri de obiecte multimedia.
- Permite lucrul cu baze de date, generarea automată a conținutului paginilor pornind de la o anumită bază de date.
- Asigură o performanță previzionare și publicare a site-urilor.

Macromedia Flash este un program pe care oricine, începător sau programator experimentat îl poate folosi cu ușurință. Pentru obținerea unor animații uimitoare și crearea interactivității în World Wide Web, Flash este neînvincibil ca editor de grafice vectoriale, animații și aplicații multimedia [6]. Cu Flash se pot crea cu ușurință grafice și efecte de animații într-un fișier care rămâne compact și foarte ușor de încărcat. Se pot adăuga acțiuni și control interactiv între animații fără a fi nevoie de scripturi. Flash exportă fișiere compresate numite "Shockwave Flash Movies". Pe Web aceste animații pot rula foarte ușor.

Facilități oferite de Flash:

- Pentru începători sunt metode foarte simple de creare a animațiilor și adăugare a scripturilor, prin simpla operațiune de "*drag and drop*". Pentru designeri, Flash propune mai mult control asupra uneltelor de desen Bezier și îmbunătățiri ale suportului de Illustrator și Freehand. Programatorii vor folosi noul ActionScript (similar cu Java Script) și suport pentru XML.
- *Exploratorul de Filme* este o nouă facilitate oferită de Flash, care permite utilizatorului să navigheze într-un film la fel de simplu ca într-un site Web.
- Animații sofisticate.
- Interactivitate.
- Aplicațiile tipice realizate cu Flash sunt asemănătoare unui film.

2.1.2. Studiu comparativ privind mediile de dezvoltare a aplicațiilor multimedia ce rulează de pe CD-ROM

Asymetrix Toolbook facilitează realizarea unor aplicații pe un calculator PC, care în trecut se puteau realiza doar de programatori bine instruiți, implicând un efort deosebit și necesitând un consum mare de timp [7], [8].

Facilitățile oferite de Asymetrix Toolbook:

- Aplicațiile tipice realizate cu Toolbook sunt asemănătoare unei cărți. Ele includ toate elementele specifice cărților, cum ar fi: paginile, fundalul, textul, grafica, posibilitatea trecerii de la o pagină la alta, precum și căutarea unei anumite pagini în interiorul cărții.
- Pe lângă acestea au fost introduse facilități suplimentare de animație, de integrare de sunete și secvențe video pentru a face prezentarea mult mai sugestivă și mai atractivă.
- Se poate construi foarte repede o interfață pentru dialogul cu utilizatorul.

- De asemenea, se pot crea baze de date noi care încorporează facilități de hypermedia: navigare, creare de legături, text, grafică, animație, sunet și secvențe video. Toate informațiile din baza de date sunt interconectate și ușor accesibile, dând posibilitatea creării rapide a oricărui tip de prezentare multimedia.
- Toolbook-ul poate fi folosit pentru gestionarea tuturor bazelor de date care au fost create anterior cu un alt produs (de ex. dBase, Lotus, Basic, Turbo Pascal), deoarece include o extensie a limbajului OpenScript care facilitează accesul la fișiere.
- Cu puțină imaginație se pot realiza chiar și desene animate și jocuri.
- Practic, singura limitare în dezvoltarea de aplicații multimedia este creativitatea și imaginația programatorului.

Aplicațiile create în mediul Toolbook rulează într-o fereastră client care este identică cu restul ferestrelor aplicațiilor existente sub Windows. De aceea, vom regăsi majoritatea elementelor standard ale meniurilor Windows, dar și elemente specifice pentru Toolbook. La un moment dat pot să fie lansate în execuție mai multe instanțe ale programului.

Macromedia Director definește termenul de animație ca fiind o referință generală la documente multimedia ce includ mai multe tipuri de informație, ce se află în general stocată sub diverse forme și sub diverse standarde. La fel ca toate filmele, animațiile realizate cu ajutorul programului Director conțin o secvență de imagini, sunete sau segmente muzicale, efecte speciale, animații realizate cu alte programe, secvențe video numerice. În plus animațiile realizate cu Director conțin o dimensiune absentă în filme :interactivitatea. Acest concept reprezintă capacitatea unei animații de a răspunde la unele acțiuni ale utilizatorului și modificarea derulării animației în funcție de opțiunile acestuia [9], [10]. În realizarea unei animații cu ajutorul programului Director, principalele elemente (meniuri și ferestre de dialog) utilizate sunt : *Scena, Scenariul, Distribuția, Actorii*.

Crearea unei animații presupune următoarele etape :

- Importarea sau crearea actorilor ce vor fi utilizați;

Actorii nu reprezintă în mod unic numai desenele sau animațiile afișate pe ecran, ci pot fi reprezentați și de texte, efecte sonore și muzica, palete de culori, butoane, secvențe video numerice și scripturi Lingo ce asigură interactivitatea și controlul animațiilor realizate cu programul Director. Fiecare actor creat sau importat apare în fereastra Distribuție activa.

- Modificarea actorilor

- Se poate de exemplu colora un desen ce a fost importat, de asemenea se poate desena un același actor dar în poziții diferite atunci când se dorește realizarea unei mișcări pentru un actor.
- Se pot modifica grafice și texte ce au fost importate în mod bitmap cu ajutorul utilităților din cadrul ferestrei Desen. Este posibilă și crearea de noi texte, forme și desene cu aceste utilitare.
- Modificarea altor tipuri de actori, cum ar fi fișierele sunet și animațiile QuickTime se poate realiza prin apelarea aplicațiilor proprii SoundEdit și Premiere. Pentru actorii importați se păstrează o informație de legătură între Director și programul în care au fost creați, astfel se poate apela automat un anumit program pentru realizarea modificărilor.
- Plasarea actorilor în scena sau în scenariu;
- Amplasarea unui actor în scena implică și inserarea automata în scenariu a informațiilor referitoare la actor.
- Scenariu va determina și introducerea automata în scena. Poziția unui actor în scena va determina și poziția acestuia pe ecran.

- Animarea actorilor.

Sub programul Director sunt posibile mai multe tipuri de animare a actorilor [11] :

- Înregistrarea în timp real – se înregistrează pozițiile ocupate de un actor atunci când acesta se deplasează în scena.
- Înregistrarea imagine cu imagine - se înregistrează poziția actorilor în scena, imagine cu imagine.
- Inserarea unei traiectorii – Director poate crea faze intermediare între un punct de plecare și un punct de sosire atunci când se dorește deplasarea unui actor. Sunt realizabile și modificări ale dimensiunilor, culorilor și transparenței.
- Realizarea unui ciclu pentru culori – permite stabilirea unui ciclu pornind de la o serie de culori selecționate dintr-o paleta de culori. Efectele ce se pot realiza sunt deosebite, de exemplu se pot modifica culorile unei scene în mod asemănător cu apusul soarelui, rotirea unei bile, etc.
- Tranziții de palete - se poate trece de la o paleta de culori la alta în mod gradat. Efectul obținut cu o tranziție între două palete de culori poate simula de exemplu un răsărit de soare.
- Autodeformații - creează automat pentru o secvența animată pozițiile intermediare ale unui actor ce suferă o deformare sau o rotație.
- Introducerea tranzițiilor – pentru trecerea de la o imagine la alta se pot folosi tranzițiile ce pot da un farmec artistic unui simplu document. Se poate trece, de exemplu, de la o imagine la alta în mod gradat sau utilizând efecte de deplasare a unei imagini peste cealaltă, wipe, etc.
- Introducerea efectelor sonore și a muzicii - un fundal sonor poate face mult mai atractivă o prezentare multimedia oferindu-i în acest mod o nouă dimensiune.
- Definirea elementelor de interactivitate - redarea interactivității unei animații se poate face prin utilizarea de butoane și legături de tip hypertext. Efectele de interactivitate elementare cum ar fi trecerea de la o secțiune animată la alta sunt foarte ușor de obținut prin utilizarea unor expresii Lingo simple. Dacă se complică scripturile Lingo se poate ajunge la orice interactivitate imaginată de realizatorul unei animații.

2.1.3. Studiu comparativ privind mediile de editare video neliniară

UleadMedia Studio este un pachet de programe format dintr-un editor video VideoEditor, un editor audio AudioEditor, un generator de generice și animare de text CG Infinity, VideoPaint care este un mediu de corectare și prelucrare la nivel de pixel cadru cu cadru a secvențelor video și o interfață de captură VideoCapture. Acest pachet oferă toate facilitățile necesare unei editări video sau pentru pregătirea acestuia pentru difuzarea pe Internet [12], [13].

Facilitățile oferite de pachetul de programe UleadMedia Studio Pro:

- Programele incluse în pachet oferă toate procesările necesare producției unui material audio-video destinat posturilor de televiziune.
- Un număr mare de producători de plăci de achiziție/redare a adoptat distribuția acestui pachet împreună cu partea hardware.
- Oferă un performant sistem de rendering ce folosește calcule realizate în timpul previzionărilor – “*smart rendering*”.
- Numărul de canale audio și video ce pot fi mixate simultan sunt suficiente pentru orice tip de aplicații – 99 de canale.
- Include o bibliotecă foarte mare de filtre ce se pot aplica global pe întreaga suprafață a unui material video.

- *VideoPaint* este un soft de procesare și corectare la nivel de pixel ce lucrează cadru cu cadru și permite corectarea unor materiale deteliorate.
- Permite folosirea unui sistem complex de transparențe pentru layere, inclusiv *alpha channel*.
- Producția unui material este organizată sub forma unui proiect, la care pot lucra mai multe persoane. Proiectul general, care include fișiere editate de diverși utilizatori, își actualizează automat legăturile de fiecare dată când unul dintre utilizatori și-a modificat sursa pe care lucrează.
- Poate lucra cu formate video digitale native DV, DVCAM, DVC Pro.

Adobe Premiere este un uimitor de simplu și puternic program de editare video, cu suport de DV pentru platforma Windows și suport multi-platformă pentru toate formatele video. Aplicația înlătură confortabil distanța dintre DV și Web păstrându-și în același timp, poziția, de altfel premiată, de cea mai accesibilă aplicație de pe piață.

Adobe Premiere integrează o mare varietate de funcțiuni și caracteristici devenind astfel cel mai bun program de editare video.

Facilitățile oferite de Adobe Premiere:

- Îmbunătățirea atât a calității cât și a vitezei de procesare la editarea filmelor, vizualizare, capturare și generare a aplicațiilor.
- *Fereastra de timp* oferă o mai bună fluiditate și intuitivitate a interfeței în adăugarea, vizualizarea și editarea diferitelor nivele într-un proiect.
- *Fereastra monitor* combină alte trei ferestre din trecut: *Clip*, *Preview* și *Trimming*. Această fereastră oferă o precisă și accesibilă egalizare (*trimming*), opțiuni flexibile de vizualizare, playback sincronizat.
- *Fereastra Project Management* permite crearea de structuri asemănătoare unei baze de date pentru categorisire, căutarea și sortare a clipurilor, de asemenea, se folosesc câmpuri definite de utilizator.
- *Keyframe-uri multiple pentru filtre* – Efectul filtrului se poate schimba în timp de-a lungul întregului clip sau poate fi definit doar pentru o anumită porțiune a acestuia.
- *Facilități audio* - o mai bună procesare audio;
 - multe filtre audio;
 - panning și fading.
- Capabilitățile programului Adobe Premiere în ceea ce privește echipamentul hardware (plăcile de achiziție și redare audio și video) și softul aferent acestora sunt din ce în ce mai puternice și eficiente.
- Interfața cu utilizatorul este dominată de *Fereastra Monitor* pentru editarea de clipuri, *Fereastra de Proiect* (care păstrează clipurile și capturile video importate) și o *Linie de Timp* care suportă la mixare până la 99 de canale audio și video.
- Adobe Premiere este OHCI (Open Host Controller Interface), ceea ce înseamnă că funcționează, în aceeași manieră, atât cu FireWire cât și cu camere video DV, DVCam și DVC Pro.

Aplicațiile create în Adobe Premiere rulează într-o fereastră client care este identică cu restul ferestrelor aplicațiilor existente sub Windows. Se regăsesc în acest program majoritatea elementelor standard ale meniurilor Windows, dar și elemente specifice Adobe Premiere.

2.1.4. Comparații și concluzii

Din punct de vedere al realizatorilor de aplicații multimedia, acestea vor deveni din ce în ce mai atractive, atât pentru realizatorii începători, cât și pentru cei mai experimentați, dacă mediile de dezvoltare a aplicațiilor multimedia își vor îmbunătăți performanțele în următoarele direcții [14]:

- Dezvoltarea de aplicații se va face având în vedere posibilitatea de rulare pe diverse tipuri de configurații hardware, acesta presupune:
 - includerea de algoritmi de detectare a configurației hardware
 - definirea modului de rulare pentru aplicație în funcție configurația hardware.
 - propunerea de diverse soluții de optimizare a modului de rulare pe configurația dată.
- Este necesar să se treacă la dezvoltarea de aplicații cu un grad mare de portabilitate – să ruleze pe diverse tipuri de calculatoare (PC, Mac etc.) și pe diverse sisteme de operare.
- Interfețele grafice cu utilizatorii trebuie să fie cât mai intuitive, cu un grad mare de interactivitate și manevrabilitate, ușor de învățat și de folosit.
- Să folosească tehnologii ODBC, OLE, DDE și DLL.

Îmbunătățirea performanțelor sistemelor de dezvoltare a aplicațiilor multimedia se poate face prin cercetarea în domeniul hardware pentru:

- Realizarea de echipamente de mare capacitate, rapide și eficiente de stocare a obiectelor multimedia.
- Obținerea unei calități deosebite la digitizarea diverselor medii – scannere, camere foto și video digitale, plăci de captură și prelucrare a sunetului și a semnalului video performante.
- Redarea rezultatului final trebuie făcută de asemenea pe echipamente performante – monitoare cu rezoluții mari, număr mare de culori, plăci video cu acceleratoare grafice puternice, etc.
- Creșterea puterii de calcul, implementarea hard a unor cât mai multe funcții multimedia în interiorul procesoarelor.
- Dezvoltarea rețelelor de comunicații de bandă largă care să asigure transmisii de pachete de date cu pierderi și întâzieri cât mai reduse.

Din punct de vedere software, performanțele mediilor de dezvoltare a aplicațiilor multimedia pot crește dacă se vor avea în vedere următoarele:

- Folosirea de structuri eficiente de reprezentare a obiectelor multimedia [15].
- Dezvoltarea de metode de contopire-sincronizare de obiecte multimedia omogene sau eterogene.
- Dezvoltarea de metode de indexare, interogare și căutare a datelor netradiționale (audio, video).
- Extragerea automată a caracteristicilor unui obiect multimedia.
- Dezvoltarea paralelizării proceselor, paralelizare pe calculatoare ce au mai multe procesoare, dar și paralelizarea proceselor prin transmiterea procesărilor în rețele eterogene de calculatoare [16].
- Dezvoltarea de sisteme de operare cât mai rapide.
- Protocoale de rețea care să satisfacă un cât mai mare set de servicii la o calitate cât mai mare.
- Algoritmi de compresie rapizi și adaptivi la situații concrete [17], [18], [19].

Lista prezentată este o sinteză a problemelor pe care de cele mai multe ori le-am întâlnit în practică.

2.2. Programarea aplicațiilor multimedia din perspectiva noilor tehnici de programare

În contextul actual, unul dintre principalele scopuri pentru producția de soft este micșorarea costurilor de producție. Costul unui soft depinde de două lucruri : programarea în sine și întreținerea.

În timpul acestor faze, trebuie găsit un echilibru între rapiditate și o bună calitate. Într-un context bine determinat cum ar fi interfață grafică sau site Web, capacitatea de a genera automat cod sursă aduce un important ajutor.

Eforturile făcute pentru a obține limbaje de programare mai ușor de citit (prin îmbunătățirii sintaxei) a determinat o creștere a codului scris în aceste limbaje. Totuși se reduce golul dintre faza de proiectare și faza de programare. Bibliotecile componentelor refolosibile permit ca fiecare nouă piesă software care este construită să nu se facă de la zero, iar cât privește modelele de proiectare ele oferă modele de arhitectură folosite pentru probleme de programare specifice.

2.2.1. Programare Orientată pe Obiecte

Programarea orientată pe obiecte este o metodă de implementare, în care programele sunt organizate ca ansamble de obiecte ce cooperează între ele, fiecare obiect reprezentând instanța unei clase; fiecare clasă aparține unei ierarhii de clase în cadrul căreia clasele sunt legate prin relații de moștenire.

Această definiție cuprinde trei părți importante și anume:

- obiectele (și nu algoritmi) sunt blocuri logice fundamentale;
- fiecare obiect este o instanță a unei clase. Clasa este o descriere a unei mulțimi de obiecte similare (caracterizate prin structură și comportament);
- clasele sunt legate între ele prin relații de moștenire.

Un limbaj de programare care oferă suport pentru utilizarea claselor și obiectelor, dar care nu are implementat mecanismul relațiilor de moștenire între clase este un limbaj de programare bazat pe obiecte. Programarea bazată pe clase și pe obiecte, care nu face uz de relația de moștenire se mai numește programare cu tipuri de date abstracte. Un obiect înglobează date și operații și reprezintă o abstracțiune a unei entități din lumea reală.

O clasă reprezintă descrierea unei mulțimi de obiecte care au aceeași structură și același comportament. Prin urmare într-o clasă vom găsi definițiile datelor și ale operațiilor ce caracterizează obiectele clasei respective. Datele definite într-o clasă se mai numesc date-membru, variabile-membru, atribute sau câmpuri, iar operațiile se mai numesc metode sau funcții membru. Valorile unui tip de clasă se numesc obiecte sau instanțe ale clasei respective.

Un obiect care este cărămida de bază a unui sistem orientat pe obiecte, include atât date, cât și metode (operații) care operează asupra datelor. Obiectul execută o operație când primește o cerere (mesaj) de la un client. Mesajele reprezintă singura cale prin care un obiect este determinat să execute o operație, în timp ce operațiile sunt singurul mod de a modifica datele interne ale obiectului. Din cauza acestor restricții starea internă a obiectului se spune că este încapsulată: ea nu poate fi accesată direct, iar reprezentarea ei este invizibilă din exteriorul obiectului. Pentru fiecare operație declarată într-un obiect se precizează numele, obiectele pe care le ia ca parametri și valoarea returnată; aceste elemente formează semnătura operației. Mulțimea tuturor semnăturilor corespunzătoare operațiilor dintr-un obiect, accesibile clienților

reprezintă interfața obiectului. Interfața unui obiect descrie complet setul mesajelor care pot fi trimise spre obiectul respectiv.

Un tip este un nume utilizat pentru a referi o anumită interfață. Astfel, un obiect este de tipul T dacă el acceptă toate mesajele corespunzătoare operațiilor definite în interfața numită T. Ca urmare, un obiect poate avea mai multe tipuri, adică o parte a interfeței poate fi de un tip, iar altă parte de alt tip. De asemenea, mai multe obiecte de tipuri diferite pot partaja un anumit tip comun, dacă interfețele lor includ tipul respectiv. Interfețele pot să conțină, la rândul lor, alte interfețe ca submulțimi. Interfețele sunt lucruri fundamentale în sistemele orientate pe obiecte. Obiectele sunt cunoscute doar prin intermediul interfețelor lor. O interfață nu dă nici un detaliu relativ la implementarea unui obiect, iar obiectele distincte pot implementa în mod diferit o aceeași cerere. Altfel spus, două obiecte având implementări complet diferite pot avea interfețe identice.

Implementarea unui obiect este definită prin intermediul clasei obiectului. Clasa unui obiect specifică datele interne ale obiectului și definițiile operațiilor pe care acesta le poate executa. Clasa definește cum este implementat obiectul, adică starea lui internă plus implementarea operațiilor. Tipul se referă doar la interfața obiectului, adică la setul mesajelor la care obiectul poate reacționa. Cu alte cuvinte, tipul ne spune ce face obiectul, în timp ce clasa ne spune cum face obiectul ceea ce face. Desigur că între clasă și tip există o strânsă legătură: prin faptul că o clasă definește operațiile pe care un obiect le poate executa, automat ea definește și tipul obiectului. Când spunem că un obiect este instanța unei clase, aceasta înseamnă că obiectul posedă interfața definită de clasa respectivă.

Principalele concepte care stau la baza programării orientate pe obiecte sunt:

- Abstractizarea exprimă toate caracteristicile esențiale ale unui obiect, care fac ca acesta să se distingă de alte obiecte; abstracțiunea oferă o definiție precisă a granițelor conceptuale ale obiectului din perspectiva unui privitor extern.
- Încapsularea este procesul în care are loc ascunderea implementării unui obiect față de majoritatea clienților săi. Deci, spre deosebire de abstractizare care definește interfața obiectului, încapsularea definește reprezentarea (structura) obiectului împreună cu implementarea interfeței.
- Modularizarea este realizată prin folosirea de clase și obiecte obținute în urma abstractizării și încapsulării ce trebuie grupate și apoi stocate într-o formă fizică, numită modul. Modulele formează arhitectura fizică a programului, existând două tipuri de nivele de modularizare: nivelul pachetelor (pachetele sunt colecții de clase înrudite din punct de vedere logic. Interfața unui pachet este constituită din clasele publice, care pot fi importate de alte pachete, iar implementarea este dată de restul claselor "nepublice") și nivelul fișierelor sursă: clasele care formează un pachet pot fi conținute în unul sau mai multe fișiere sursă și fiecare asemenea fișier poate fi compilat separat.
- Ierarhizarea este dată de faptul că adesea un grup de abstracțiuni formează o ierarhie, iar prin identificarea acestor ierarhii, putem simplifica substanțial înțelegerea problemei. Ierarhizarea reprezintă o ordonare a abstracțiilor. Cele mai importante ierarhii în programarea pe obiecte sunt: ierarhia de clase și ierarhia de obiecte.
- Moștenirea este unul din avantajele oferite de tehnologia orientării pe obiecte și reprezintă posibilitatea de a extinde comportamentul unei clase existente prin definirea unei clase noi care moștenește conținutul primei clase, adăugând la aceasta elemente specifice. Clasa existentă, care va fi moștenită se mai numește clasă de bază sau superclasă. Clasa care realizează extinderea se mai numește subclasă, clasă derivată sau clasă descendentă. În legătură cu relația de moștenire trebuie evidențiate două aspecte esențiale: reutilizarea de cod și polimorfismul. În ceea ce privește gradul de multiplicitate a relației de moștenire avem: o subclasă poate extinde o singură superclasă, adică avem de a face cu moștenire simplă;

aceeași superclasă poate fi extinsă de mai multe subclase distincte. O subclasă, la rândul ei, poate constitui superclasa pentru o altă clasă. O clasă de bază cu toate descendentele ei formează o ierarhie de clase. Numele unei superclase joacă rolul de supertip pentru oricare dintre descendentele ei directe sau indirecte. Polimorfismul este dat de proprietatea că o referință la superclasă poate indica și manipula obiecte ale oricărei clase descendente din ea. Manipularea se referă la faptul că prin intermediul acelei referințe se pot apela metodele definite în superclasă, indiferent dacă obiectul concret aparține superclasei sau uneia dintre subclase. Această facilitate constituie așa numitul polimorfism parțial.

Atunci când într-o subclasă apare o metodă având semnătura identică cu o metodă din superclasă, spunem că metoda din subclasă o redefinește pe omonima ei din superclasă. În acest caz, un obiect al subclasei practic are două exemplare ale metodei respective, unul moștenit și unul propriu. Se va executa întotdeauna acel exemplar de metodă care este definit în clasa la care aparține obiectul concret indicat de referința utilizată. Această facilitate, adăugată polimorfismului parțial conduce la polimorfismul total [20],[21].

În Java, clasa concretă la care aparține un obiect indicat de o referință se cunoaște însă abia la execuția programului. Deci, un apel de metodă nu este "rezolvat" (adică pus în corespondență) la compilare, ci doar la execuție. Acest procedeu se numește legare dinamică.

O interfață grafică conține elemente de control (widgets) care pot fi activate de utilizator cu ajutorul mouse-ului sau al tastaturii. Activarea unui element de control determină un anumit răspuns din partea programului, răspuns concretizat prin execuția unei anumite operații. Cele mai răspândite interfețe grafice sunt cele bazate pe ferestre. Principiul de lucru al unei asemenea interfețe presupune ca fiecărei aplicații active la un moment dat să i se aloce o zonă dreptunghiulară pe ecran, numită fereastră. În interiorul ferestrei respective se vor găsi toate elementele de control necesare dialogului utilizator - aplicație, precum și informațiile afișate de aplicație.

Se poate spune astfel că, dacă sistemul de operare folosit permite execuția concurentă a mai multor aplicații, prezența ferestrelor aplicațiilor face ca ecranul să devină transparent în raport cu memoria. Utilizatorul capătă o vedere de ansamblu asupra a ceea ce se întâmplă în "capul" calculatorului, dispărând senzația dată de lucrul cu o cutie neagră. Pentru ca o aplicație care folosește interfață grafică să poată funcționa, este necesară existența unei platforme sau server grafic care să determine intrarea monitorului în regim grafic și care să ofere funcțiile primitive necesare, între altele, pentru desenarea componentelor GUI și pentru receptarea semnalelor generate de mouse și tastatură.

Programarea aplicațiilor cu interfețe grafice ar fi însă o corvoadă dacă ne-am baza doar pe primitivele oferite de serverul grafic: ar fi cam ceea ce este un limbaj de asamblare pe lângă un limbaj de nivel înalt. De aceea, în sprijinul programatorilor au fost create o serie de suporturi software care permit ca o aplicație grafică să poată fi construită din "piese" standardizate, gata "fabricate". În felul acesta, programatorul este liber să se concentreze mai mult asupra logicii aplicației propriu-zise, munca lui constând în principal în a asocia elementelor de control ale interfeței, diferite funcțiuni și în a scrie codul ce realizează acele funcțiuni. Unei aplicații grafice îi corespunde o fereastră principală (frame) și, după caz, una sau mai multe ferestre subordonate celei principale. Orice acționare de către utilizator a unui element GUI prin intermediul mouse-ului - lui sau al tastaturii determină apariția unui așa - numit eveniment (concret, acest lucru înseamnă că se creează un obiect al unei clase predefinite). Pentru ca acționarea unui element GUI să se soldeze cu un anumit răspuns, programatorul trebuie să rezolve partea de tratare a evenimentului. Aplicațiile la care dialogul cu utilizatorul presupune generarea de evenimente și tratarea acestora se spune că fac parte din clasa aplicațiilor dirijate de evenimente (event driven).

Firele de execuție reprezintă porțiuni de cod ale aceluiași program care se pot executa în paralel una față de alta. Firele de execuție reprezintă modalitatea prin care de exemplu limbajul

Java suportă programarea concurentă. Lucrul acesta depinde practic de suportul de execuție a programelor, hardware și software. În general, dacă se dispune de un sistem cu mai multe procesoare, firele vor fi distribuite pe acestea. Dacă avem un singur procesor execuția paralelă se simulează folosind diferite tehnici. În principiu se cunosc două asemenea tehnici care constituie de fapt politici de planificare a firelor: un fir de execuție este lăsat să ruleze până când se termină sau până când ajunge în stare de așteptare, moment în care este preluat un fir cu prioritate mai mare pregătit pentru execuție sau în versiunile pentru 32 de biți se aplică tehnica numită time-slicing ("felierea" timpului). Practic fiecărui fir *i* se alocă câte un interval de timp în care poate ocupa procesorul. În felul acesta, fiecare fir ajunge, prin rotație, să ocupe câte puțin procesorul, iar utilizatorul are impresia că firele rulează în paralel.

Divizarea în fire este avantajoasă mai ales în cazul operațiilor care necesită timp mare de rulare sau care forțează procesorul să aștepte pentru a accesa discul sau rețeaua. Constituirea unor asemenea operații ca fire separate conduce la:

- creșterea vitezei de execuție a programului în ansamblu, deoarece procesorul nu mai pierde vremea așteptând după resurse, ci ia la rând alte fire până când resursele devin disponibile;
- Creșterea vitezei de răspuns a programului, deoarece interfața utilizator poate avea propriul ei fir de execuție care să preia intrările utilizatorului chiar în timpul cât celelalte fire sunt „ocupate”.

Desigur că toate aceste avantaje au un preț: întrucât mai multe fire de execuție aparțin aceluiași program, rezultă că ele pot accesa în comun anumite structuri de date ale programului. Aici se pune problema sincronizării accesului, astfel încât să se asigure consistența datelor, lucrul care complică puțin proiectarea programului.

De exemplu, dacă o metodă este declarată ca fiind sincronizată înseamnă că pe durata în care un fir execută această metodă, obiectul receptor implicat intră în starea blocat, astfel încât nici un alt fir nu mai poate executa vreo altă metodă sincronizată pentru obiectul respectiv. Dacă un fir apelează o metodă sincronizată pentru un obiect aflat în starea blocat, firul respectiv intră într-o coadă de așteptare atașată obiectului. Ieșirea din blocaj a obiectului se realizează la terminarea execuției metodei sincronizate.

O altă observație importantă este ca operațiile propriu – zise de citire / scriere se pot efectua numai în anumite condiții. De exemplu dacă tabloul este gol nu putem realiza citirea. Iar în acest caz, firul consumator trebuie să aștepte ca un eventual producător să introducă ceva în tablou. Aici intervine problema comunicării între fire. Comunicarea presupune un mecanism prin care un fir care a ținut ocupată o resursă să anunțe celelalte fire la momentul eliberării resursei.

2.2.2. Noutăți aduse de limbajul Java în programarea orientată pe obiecte

Java regrupează toate caracteristicile unui limbaj profesional: este simplu, dar în același timp flexibil și puternic. Caracteristicile sale cele mai importante sunt următoarele:

- Orientare pe obiecte;
- Multithread;
- Control static strict de tipuri;
- Cu legătură dinamică;
- Gestionarea memoriei;
- Simplitate;
- Securitate.

Java este un limbaj de programare profesional în termeni din punct de vedere al productivității, rentabilității, perenității și securitate, iar în plus poate fi considerat un limbaj dedicat client – server. În afară de caracteristicile intrinseci ale limbajului cea mai importantă caracteristică este portabilitatea [22], [23], [24].

Orientarea pe obiecte. Java este un limbaj obiectual. G. Masini definește limbajele pe obiecte ca fiind acelea care respectă principiile abstractizării, încapsulării și moștenirii, adică toată informația fiind definită în clase care îl obligă pe programator să-și structureze datele foarte bine. În acest sens, Java este un limbaj orientat pe obiecte oferind și o mai bună stăpânire a complexității, o reutilizare (potențială) mai ușoară, o administrare (evolutivă) ameliorată, un caracter esențial distribuit și rețeaua aplicațiilor.

Multithread. Limbajul permite dezvoltarea foarte ușoară ale fluxurilor de execuție concurente, în aceeași aplicație. Integrarea capacității de execuție concurente va putea, bine înțeles, servi la propunerea de noi posibilități pentru utilizator (a imprima sau a încărca imagini în timp ce se lucrează), dar scopul este de a permite programelor realizate în Java să funcționeze pe calculatoare multiprocesor care încep să apară tot mai frecvent pe piață, fără a exista probleme majore de adaptare. Alte limbaje propun mecanisme de multithreading dar folosesc librării externe nestandardizate. Integrând această capacitate direct în limbaj, programele care îl folosesc pot fi purtate de la o platformă la alta. Pentru a permite această transferabilitate, realizatorii limbajului Java au trebuit să integreze mecanismul de partiționare în timp (scheduling) în mașina virtuală (motorul de execuție a limbajului Java) și aceasta în detrimentul mecanismelor existente în sistemele de exploatare.

Legătura dinamică. Nu există adăugări de legături în Java (proces care intervine după compilare, care avea drept scop să lege librăriile externe la program). Verificarea existenței librăriilor se face în momentul compilării și încărcarea codului acestor librării se face în timpul execuției programului. Aceasta, are drept consecință o micșorare a mărimi executabililor și permite o optimizare în timpul încărcării librăriilor.

Gestionarea memoriei. Standardul Java dispune de un mecanism Garbage Collector care se ocupă cu ștergerea din memorie a tuturor obiectelor care nu sunt utile. Acest lucru, are un avantaj esențial de ușurare a programării, permite scrierea programelor unde fiecare linie are un sens în relațiile directe cu obiectivul programului.

Alt aspect important: noțiunea de pointer a dispărut în Java. Aceasta răspunde la două probleme: utilizarea pointerilor este sursa numeroaselor bag-uri în programe iar utilizarea pointerilor nu permite dezvoltarea securității programelor.

Încă, acesta este un lucru care ar putea penaliza dezvoltarea în acest sens pentru că el nu permite multe accese la memoria fizică a calculatorului, dar care aduce o fiabilitate fără precedent programului. Aceasta implică de asemenea că dezvoltarea drivere-lor sau gestionărilor comunicației la nivelul de bază nu pot fi efectuate direct în Java. Java nu are deci pretenții de schimbare a tuturor limbajelor de programare.

Securitate. Dezvoltarea unui limbaj care să posede caracteristici deja cunoscute cere un efort important din partea celor care îl concep, mai bine spus mașinii virtuale. De fapt înainte de execuția unui program Java este făcută o verificare strictă a codului. Câteva caracteristici importante legate de securitate în Java: codul nu poate să treacă de protecțiile impuse de limbaj (de exemplu un câmp declarat privat nu poate să fie accesat de o altă clasă), codul nu trebuie să încerce să construiască referințe directe la memorie, codul nu trebuie să încerce să utilizeze un obiect în loc de alt obiect (se poate spune că este un limbaj puternic tipizat), sau să utilizeze un tip incompatibil cu altul și codul nu trebuie să cauzeze deplasări în stivă.

Alte verificări de nivel mai înalt sunt făcute de medii în care se execută Java (de exemplu programele de navigare pe Internet), aceasta pentru a verifica dacă programele nu accesează resurse interzise (fișiere locale, servere exterioare, etc.). Acestea nu sunt impuse de limbajele în sine ci de utilitățile care execută programele în Java. În funcție de mediul de execuție aceste limite de securitate sunt mai mult sau mai puțin stricte.

Simplicitate. În primul rând se poate spune despre Java că are o sintaxă simplă. Un anumit număr de bariere a fost ridicat, de exemplu supraîncărcarea operatorilor a fost desființată.

Absența pointerilor, a variabilelor globale, obligativitatea unei bune structurări a surselor, obligativitatea folosirii obiectelor conferă o bună lizibilitate și o familiarizare mai rapidă cu programarea orientată pe obiecte.

Primul și cel mai important aspect este că Java facilitează utilizarea graficii interactive și a altor efecte speciale într-o pagină World Wide Web prin scrierea de aplicații speciale pentru Internet, denumite applets, care sunt executate în pagini de Web. În plus atunci când rulăm un applet, serverul aflat la distanță îl transmite către browser prin Internet [25], [26], [27], [28].

2.3. Structurarea modului de programare conform conceptului Open Flexible Language (OFL), de hipergeneralizare în programare

2.3.1. Concepte și principii fundamentale

De precizat ca OFL, e dezvoltat în cadrul Laboratorului de Informatica al IUT Nice din cadrul Parcului Tehnologic Sophia Antipolis, de către un colectiv de cercetatori francezi cu care autorul a colaborat timp de 2-3 ani. Rezultatele colaborării s-au concretizat într-un număr de 2 lucrări științifice publicate, prezentate în bibliografie [15], [29].

Acest capitol prezintă conceptele și principiile fundamentale ale modelului OFL. OFL este acronimul pentru Open Flexible Languages și numele unui Meta-model pentru limbajele de programare pe obiecte bazate pe clase. El se bazează pe trei concepte esențiale ale acestor limbaje, ce vor fi abordate în ceea ce urmează:

- Concept - limbaj;
- Concept - descriptor;
- Concept - relație.

În figura de mai jos este desenată legenda generală pentru OFL pentru a înțelege mai bine relațiile din acest mediu:

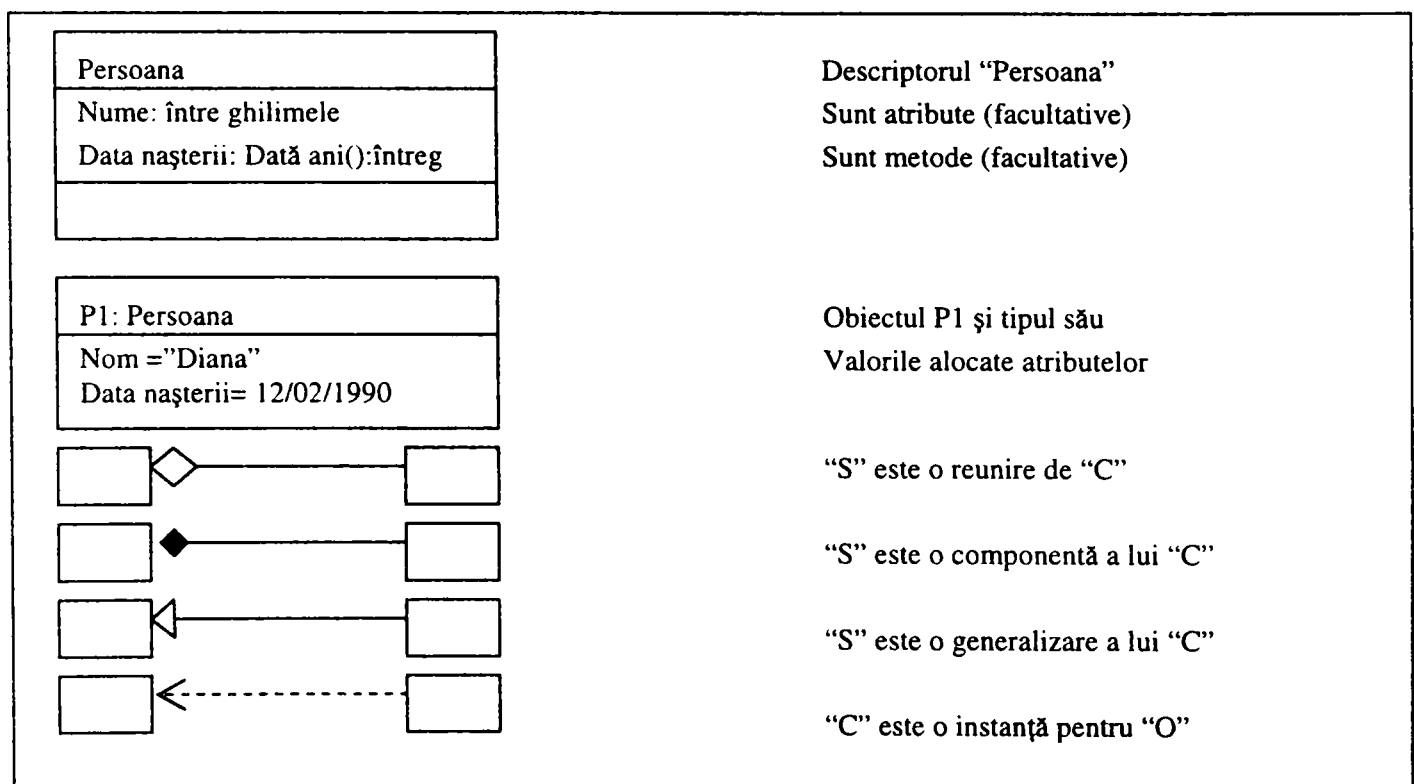


Figura 2-1 Relații directe sau indirecte

Prima noțiune fundamentală din model este considerată conceptul–relație. Se utilizează următoarea terminologie: concept-relație este un fel de relație abstractă a claselor, în limbaje de programare orientate pe obiecte. Un exemplu din Java, de concept-relație poate fi considerată moștenirea **implements**, iar ca exemplu de relație, relația **implements** dintre o interfață și o clasă de date.

Pentru fiecare relație, noi vom defini noțiunile:

- **Description-source:** provine de la descriptorul care declară relația;
- **Description-cible:** este un descriptor care definește și furnizează serviciul (într-un termen foarte general) cerut, de relația intermediară, de description-source.

Description-source este o clasă care declară un atribut, iar description-cible cel care descrie acest tip. O sursă poate fi directă sau indirectă și este exemplificată în imaginea de mai jos.

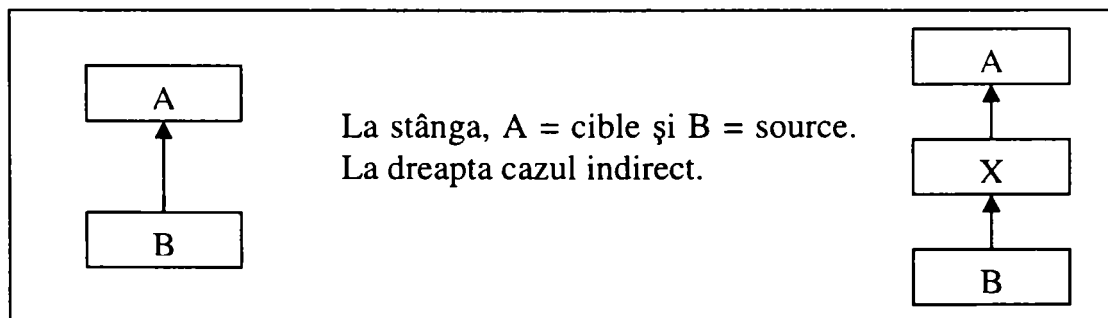


Figura 2-2. Relații directe sau indirecte

Modelul descrie relații între descriptori (exemplu: moștenire, reunire), relații între descriptori și obiecte (exemplu: instanță, extensie) și poate de asemenea defini relații între obiecte.

2.3.2. Conceptul relații (Concept-relations)

Caracteristicile pe care le are conceptul relații în limbajul Java. Se regăsesc trei concepte-relație importate (moștenire între clase, moștenire între interfețe și implementare) și patru concepte-relație utile (reunire, reunire de clase, compunere și compunere de clase).

- **Moștenirea între clase.** Clasa S moștenește clasa C, S și C pot fi concrete sau abstracte. S și C au fost definite în legenda de mai sus (figura 2-2). În Java putem avea o clasă concretă moștenită de o clasă abstractă. Acest concept poate fi:

- | | |
|---|--|
| <ul style="list-style-type: none"> ▪ Importat; ▪ Simplu (S nu poate moștenii altă clasă); ▪ Non-repetabil (S nu poate moștenii de mai multe ori C); ▪ Liniar (C nu poate S, direct sau indirect); ▪ Polimorfism; ▪ Invariant pentru redefinire (în S toate atributele, rezultatele funcțiilor și parametrii metodelor trebuie să fie de același tip ca omologul lor din C) și liber pentru reamplasare (în S, toate atributele și parametrii metodelor trebuie să fie de tip diferit fata de omologul său C); | <ul style="list-style-type: none"> ▪ Asimetric; ▪ Cu acces direct autorizat (declarația pentru constructorii și primitive este <i>private</i>) și cu acces indirect autorizat (aceasta în cazuri particulare și se folosește cuvântul cheie <i>super</i>); ▪ Concept –description aplicabil: <ul style="list-style-type: none"> ○ atât ca sursă directă: clasă, clasă abstractă, clasă membră statică, clasă membră, clasă locală, clasă anonimă; ○ cât și ca țintă directă: clasă, clasă abstractă, clasă membră statică, clasă membră, clasă locală. |
|---|--|

➤ **Moștenirea între interfețe.** Interfața S moștenește interfața C (cuvântul cheie utilizat este *extends*). Caracteristicile acestui concept sunt:

- | | |
|--|--|
| <ul style="list-style-type: none"> ▪ Importabilitate; ▪ Multiplicitate; ▪ Repetabilitate; ▪ Liniaritate; ▪ Acces direct și indirect interzis; ▪ Polimorfism; ▪ Invariabilitate; | <ul style="list-style-type: none"> ▪ Asimetrie; ▪ Concept – description aplicabil: <ul style="list-style-type: none"> ○ Atât ca sursă directă: interfață și interfață membră statică; ○ Cât și ca țintă directă: interfață și interfață membră statică. |
|--|--|

➤ **Implementarea.** Clasa S implementează interfața C și care are următoarele caracteristici:

- | | |
|---|---|
| <ul style="list-style-type: none"> ▪ Importabilitate; ▪ Multiplicitate; ▪ Repetabilitate (S nu poate implementa direct decât o dată C, iar indirect de mai multe ori); ▪ Liniaritate; ▪ Acces direct și indirect interzis; ▪ Polimorfism; ▪ Invariant; | <ul style="list-style-type: none"> ▪ Asimetric; ▪ Redefinirile și concretizările sunt autorizate; ▪ Concept – descriere aplicabil: <ul style="list-style-type: none"> ○ Atât ca sursă directă: clasă, clasă abstractă, clasă membră statică, clasă membră, clasă locală, clasă anonimă; ○ Cât și ca țintă directă: interfață și interfață membră statică. |
|---|---|

➤ **Reunirea.** Acesta este un concept utilizabil, mai mult el nu poate fi importabil. Caracteristicile acestui concept sunt:

- | | |
|---|---|
| <ul style="list-style-type: none"> ▪ Utilizabilitate; ▪ Multiplicitate; ▪ Repetabilitate; ▪ Circularitate; ▪ Acces direct interzis și acces indirect autorizat (pentru primitive avem declararea <i>public</i>); ▪ Independent; ▪ Atribut instanță; ▪ Atribut direct accesibil în citire; | <ul style="list-style-type: none"> ▪ Atribut direct accesibil în scriere; ▪ Asimetrie; ▪ Concept – descriere aplicabil: <ul style="list-style-type: none"> ○ Atât ca sursă directă: clasă, clasă abstractă, clasă membră statică, clasă membră, clasă locală, clasă anonimă; ○ Cât și ca țintă directă: clasă, clasă abstractă, interfață, clasă membră statică, clasă membră, clasă locală, clasă anonimă, interfață membră statică și tablou. |
|---|---|

➤ **Reunire de clase.** În acest caz, calificatorul de acces este *static* este precizat. Aceasta înseamnă că atributele din clasa C sunt asociate clasei S întotdeauna ca instanțele sale. Caracteristicile acestui nou concept sunt:

- | | |
|--|---|
| <ul style="list-style-type: none"> ▪ Utilizabilitate; ▪ Multiplicitate; ▪ Circularitate; ▪ Acces direct interzis și indirect autorizat (pentru primitive avem calificativul de acces <i>public</i>); | <ul style="list-style-type: none"> ▪ Independent; ▪ Atribut direct accesibil la citire; ▪ Atribut direct accesibil la scriere; ▪ Asimetric; ▪ Concept – descriere aplicabil: |
|--|---|

- Atât ca sursă directă: clasă, clasă abstractă, interfață, clasă membră statică, clasă membră, clasă locală, clasă anonimă și interfață membră statică;
- Cât și ca țintă directă: clasă, clasă abstractă, interfață, clasă membră statică, clasă membră, interfață membră statică și tablou.

➤ **Compunere.** Acest concept-relație descrie utilizarea atributelor de tip primitive (ex.: booleene) în clasa sursă. Caracteristicile acestui concept sunt:

- Utilizabilitate;
- Multiplicitate;
- Repetabilitate;
- Liniaritate;
- Acces direct și indirect interzis;
- Dependent;
- De atribut, deci de probleme de accesibilitate;
- Atribut instanță;
- Asimetric.
- Concept–descriere aplicabil:
 - Atât ca sursă directă: clasă, clasă abstractă, interfață, clasă membră statică, clasă membră, clasă locală, clasă anonimă și interfață membră statică;
 - Cât și ca țintă directă: tipul primitiv.

➤ **Compunere de clase.** Ea poate fi văzută ca un amestec de reuniuni de clase și compunere, iar caracteristicile pentru acest concept sunt:

- Liniaritate;
- Acces direct și indirect interzis;
- Dependent;
- De atribut, deci de probleme de accesibilitate.
- Concept–descriere aplicabil:
 - Atât ca sursă directă: clasă, clasă abstractă, interfață, clasă membră statică, clasă membră, clasă locală, clasă anonimă și interfață membră statică;
 - Cât și ca țintă directă: tipul primitiv.

2.3.3. Conceptul descriere (concept-description)

Se remarcă, că aceste concepte–relații, se aplică adesea între clase, dar nu întotdeauna. Conceptul-relații implementat de Java, dacă are o sursă pentru o clasă, el generează un ansamblu de interfețe ca ținte. Aceste diferite feluri de entitate sunt concepte-descriere.

Ca pentru concepts-relație, se dă un ansamblu de concepte-descriere în Java. Punctele considerate ca esențiale în prezentarea conceptelor-descriere sunt:

- Se spune despre un concept-descriere simplu că este asociat unui singur tip, sau despre mai multe că sunt asociate la mai multe tipuri;
- Acest concept descriere are capacitatea de a crea instanțe proprii;
- Primitivele lor sunt încapsulate în descriptori;
- Atributele și/sau metodele sunt permise de o instanță unică, de o instanță sau de descriptor sau de multe astfel de posibilități;
- Acest descriptor este vizibil și poate fi accesat.

Pentru fiecare concept-descriere, se asociază o listă de concepte-relație, care poate fi utilizată tot timpul de conceptul-descriere ca sursă. O a doua listă se realizează la fel și pentru țintă. În continuare vor fi prezentate nouă concepte-descriere pentru Java:

➤ **clasa:** luăm ca prim exemplu o clasă C. Voi descrie conceptul descriere corespunzător în felul următor:

- | | |
|--|---|
| <ul style="list-style-type: none"> ▪ simplu; ▪ generator; ▪ încapsulat; ▪ attribute și metode autorizate; ▪ autorizează attribute instanță sub-descriere; ▪ nedistrugător; | <ul style="list-style-type: none"> ▪ supraîncărcare autorizată, tipul valori de retur nu este luat în considerare; ▪ validează concepte-relație: <ul style="list-style-type: none"> ○ atât ca sursă directă: moștenire între clase, implementare, reunire, reunire de clase și compunere; ○ cât și ca țintă directă: moștenire între clase, reunire și reunire de clase. |
|--|---|

➤ **clasă abstractă:** Se spune că clasa poate conține în același timp metode abstracte și metode concrete. O clasă abstractă în Java poate fi descrisă prin identificatorul de acces *abstract* și poate fi:

- | | |
|--|--|
| <ul style="list-style-type: none"> ▪ simplă; ▪ non generatoare (dar poate poseda constructori care o pot activa indirect prin metoda <i>super(...)</i> într-o clasă concretă moștenită direct); ▪ nedistrugătoare; ▪ încapsulată; ▪ attribute și metode autorizate; | <ul style="list-style-type: none"> ▪ autorizează attribute instanță sau descriptori; ▪ supraîncărcare autorizată, tipul valori de retur nu este luat în considerare; ▪ validează concepte-relație: <ul style="list-style-type: none"> ○ atât ca sursă directă: moștenire între clase, implementare, reunire, reunire de clase și compunere; ○ cât și ca țintă directă: moștenire între clase, reunire și reunire de clase. |
|--|--|

➤ **interfața:** Acest concept- descriere este:

- | | |
|--|---|
| <ul style="list-style-type: none"> ▪ simplu; ▪ non generatoare; ▪ nedistrugător; ▪ attribute interzise dar metode autorizate; ▪ încapsulată; ▪ autorizează attribute instanță sau descriptori; | <ul style="list-style-type: none"> ▪ supraîncărcare autorizată, tipul valori de retur nu este luat în considerare; ▪ pachet local; ▪ validează concepte-relație: <ul style="list-style-type: none"> ○ atât ca sursă directă: moștenire între interfețe, reunire de clase și compunere; ○ cât și ca țintă directă: moștenire între interfețe, implementare, reunire și reunire de clase. |
|--|---|

➤ **clasă membră statică:** Clasele membre statice au ca principal interes sa apară în spațiul dintr-un alt descriptor. O clasă C poate accede la attributele și metodele clasei sale încapsulate. Caracteristicile unei clase membre statice sunt:

- | | |
|--|--|
| <ul style="list-style-type: none"> ▪ simplă; ▪ generatoare; ▪ nedistrugător; ▪ attribute și metode autorizate; ▪ încapsulată; | <ul style="list-style-type: none"> ▪ autorizează attribute instanță sau descriptori; ▪ supraîncărcare autorizată, tipul valori de retur nu este luat în considerare; |
|--|--|

- | | |
|---|---|
| <ul style="list-style-type: none"> ▪ descriptor local încapsulat (C poate fi accesibil dacă identificatorul său de acces nu este <i>private</i>); ▪ validează concepte-relație: | <ul style="list-style-type: none"> ○ atât ca sursă directă: moștenire între clase, implementare, reunire, reunire de clase și compunere; ○ cât și ca țintă directă: moștenire între clase, reunire și reunire de clase. |
| <p>➤ clasă membră: Este posibilă definirea în Java a claselor membre care sunt subdescriptori ai altor descriptori. Trebuie remarcat că fiecare instanță C este automat asociată unei instanțe din clasa sa încapsulată și deci are acces la toate atributele și metodele acestea. O clasă membră C poate fi descrisă astfel:</p> | |
| <ul style="list-style-type: none"> ▪ simplă; ▪ generatoare; ▪ nedistrugătoare; ▪ atribute și metode autorizate; ▪ încapsulată; ▪ autorizează atribute instanță sau descriptori; | <ul style="list-style-type: none"> ▪ supraîncărcare autorizată, tipul valori de retur nu este luat în considerare; ▪ descriptor local încapsulat; ▪ validează concepte-relație: ○ atât ca sursă directă: moștenire între clase, implementare, reunire și compunere; ○ cât și ca țintă directă: moștenire între clase, reunire și reunire de clase. |
| <p>➤ clase locale: Clasele locale sunt interne într-o metodă dintr-o clasă: ele sunt de asemenea similare variabilelor locale. Caracteristicile unei clase locale C sunt:</p> | |
| <ul style="list-style-type: none"> ▪ simple; ▪ generatoare; ▪ nedistrugătoare; ▪ atribute și metode autorizate; ▪ încapsulată; ▪ nu autorizează ca instanță atributele; | <ul style="list-style-type: none"> ▪ supraîncărcare autorizată, tipul valori de retur nu este luat în considerare; ▪ metodă local încapsulată; ▪ validează concepte-relație: ○ atât ca sursă directă: moștenire între clase, implementare, reunire și compunere; ○ cât și ca țintă directă: reunire. |
| <p>➤ clase anonime: Clasele interne anonime sunt ca și clasele locale. Ele sunt deci utilizate direct în expresii. O clasă anonimă C poate fi descrisă de următoarele caracteristici:</p> | |
| <ul style="list-style-type: none"> ▪ simple; ▪ generatoare (o singură instanță este generată pentru C și este automată); ▪ nedistrugătoare; ▪ atribute și metode autorizate; ▪ încapsulată; ▪ atribut instanță unic (instanța C este anonimă și creează sensul expresiei, ea nu este deci partajabilă); | <ul style="list-style-type: none"> ▪ supraîncărcare autorizată, tipul valori de retur nu este luat în considerare; ▪ obiect local provenit din expresia încapsulată; ▪ validează concepte-relație: ○ atât ca sursă directă: moștenire între clase, implementare, reunire și compunere; ○ cât și ca țintă directă: reunire. |
| <p>➤ interfață membră statică: Ele sunt echivalente din punctul de vedere al interfețelor cu clasele membre statice. Trebuie știut ca în Java interfețele membre statice sunt singurele</p> | |

interfețe servant subdescriptorilor: nu există interfețe membre (non statice), interfețe locale sau interfețe anonime. Caracteristicile unei interfețe membre statice sunt:

- | | |
|--|--|
| <ul style="list-style-type: none"> ▪ simplă; ▪ non generatoare; ▪ nedistrugătoare; ▪ attribute interzise dar metode autorizate; ▪ încapsulată; ▪ autorizează attribute instanță sau descriptori; | <ul style="list-style-type: none"> ▪ supraîncărcare autorizată, tipul valori de retur nu este luat în considerare; ▪ descriptor local încapsulat; ▪ validează concepte-relație: <ul style="list-style-type: none"> ○ atât ca sursă directă: moștenire între interfețe, reunire de clase și compunere; ○ cât și ca țintă directă: moștenire între interfețe, implementare, reunire și reunire de clase. |
|--|--|

➤ **tablou:** Se spune că este un caz particular. Tablourile sunt manipulate de clasa *java.lang.reflect.Array* și sunt modelate de pseudo-clasele lor create automat creând prima lor instanță. Caracteristicile unui descriptor tablou sunt:

- | | |
|--|---|
| <ul style="list-style-type: none"> ▪ simplu (un descriptor este creat pentru fiecare tip de element: un tablou întreg nu este de același tip cu un tablou boolean); ▪ generator; ▪ nedistrugător; ▪ attribute (de exemplu pseudo-atribut <i>length</i>) și metode autorizate; ▪ încapsulată; ▪ partajat; | <ul style="list-style-type: none"> ▪ global; ▪ nu există nici o supraîncărcare în această pseudo-clasă; ▪ validează concepte-relație: <ul style="list-style-type: none"> ○ atât ca sursă directă: nu este aplicabilă (nu este posibilă crearea unui nou tablou concept-descriere); ○ cât și ca țintă directă: reunire și reunire între clase. |
|--|---|

➤ **Tipul primitiv:** Este considerat cel de-al doilea caz particular pentru Java și sunt într-un număr de opt: boolean, caracter, octal, întreg scurt, întreg, întreg lung, float și double float. Contrar tuturor altor tipuri primitive nu sunt obiecte. Caracteristicile lui sunt:

- | | |
|---|--|
| <ul style="list-style-type: none"> ▪ Acest tip dispune de o interfață sintactic particulară: cuvintele lor cheie sunt asociate și este posibilă utilizarea directă a tipului constantelor lor (ex.: 16, true, etc.). ▪ Fiecare tip primitiv descrie o valoare și un nume de referință. A invoca o primitivă pentru el nu are deci nici un sens. | <ul style="list-style-type: none"> ▪ Fiecare tip primitiv este la fel descris de o clasă adevărată (exemplu: <i>integer</i> pentru <i>int</i>) care permite utilizarea de valori primitive atât ca obiect. ▪ validează concepte-relație: <ul style="list-style-type: none"> ○ atât ca sursă directă: nu este aplicabilă (nu este posibilă crearea unui nou tip primitiv); ○ cât și ca țintă directă: compunere. |
|---|--|

În Java, clasele reprezintă șiruri (de caractere), nu constituie un caz particular din punct de vedere sintactic.

2.3.4. Conceptul limbaj (Concept-language)

Conceptul limbaj este o noțiune importantă și simplă. El descrie, cum numele său indică, un limbaj modelat. Conform termenului, această modelare de limbaje trebuie să permită punerea în lucrare și controlarea interoperabilității între obiecte născute din concepts-language diferite.

În continuare, vor fi explicate elementele care sunt întâlnite în conceptul limbaj.

Caracteristici (caractéristiques)

Fiecare astfel de limbaj este format din două componente:

- un ansamblu de concepte de descriptori (concepts-descriptions)
- un ansamblu de concepte relații (concepts-relations), cu condiția ca, pentru fiecare concept-relație, cel puțin unu dintre conceptele-descriere selectate pot să servească drept sursă pentru descriptor și un altul (sau același) să servească ca destinație descriptor.

Ansamblul concept-relații poate sau să fie dedus din ansamblul concept-descriere, sau să fie dat explicit, ceea ce ne permite eventual reducerea, la acest nivel, specificațiilor de compatibilitate descrise în fiecare concept-descriere.

Parametrii (parameters)

Fiecare concept-limbaj dispune, ca și celelalte concepte de modelul nostru, și anume, un număr de parametrii care reprezintă principalele posibilități de comportament la nivel de limbaj. De exemplu:

- **Name** este vorba doar despre numele concept-limbaj, de exemplu, „MyLanguage” sau „Java-like”.
- **Persistence** acest parametru boolean indică dacă limbajul, concept-descriptorilor lui, concept-relațiilor lui trebuie să integreze și să țină cont de parametrii și acțiunile care privesc clasa de servicii persistence.
- **Concurrency** indică ca și cel precedent activarea unei clase de servicii, aici fiind vorba de concurrence. Pentru fiecare clasă de servicii avem deci un parametru boolean care specifică dacă parametrii și acțiunile asociate trebuie să fie luate în considerare.

Redefinire (redefinition)

În ideea de a permite utilizarea componentelor, reducând explicit capacitățile lor, se poate observa, de exemplu, utilizând un concept-relație (același lucru putându-se face și cu concept-descriptor) se pot limita capacitățile sale. Se poate pentru acesta defini o listă formată din tripleți:

<concept- relație, nume_parametru, valoare>

în care este posibilă specificarea unei noi valori pentru un parametru concept-relație. Este de înțeles deci că trebuie să se țină seama de a limita capacitățile unui concept-relație și în nici un caz de a le măria.

Această idee de limitare este de fapt dependentă de semantica fiecărui parametru. În prima aproximare, noi putem considera că, pentru toate tipurile desprinse dintr-o clasă (exemplu: integer) se limitează semnificația unei valori inferioare selecționate sau egale. Dar aceasta ne aduce la concluzia de exemplu că parametrii booleani nu pot fi redefiniți spre *false*, deoarece acesta poate fi un contrasens. Să ne imaginăm un parametru *Simple* care definește dacă un concept-relație este simplu sau multiplu. Este evident că pentru el limitarea constă în trecerea de la *false* la *true* și nu invers. Se preferă deci, pentru fiecare parametru specificarea în ce măsură poate fi redefinit, aceasta îmi permite a păstra toată libertatea pentru a asocia o semantică la valorile datelor.

Dacă ne imaginăm, pentru ilustrarea redefinirilor, un concept-relation având un parametru *Circularity* cu valoarea *allowed*, aceasta semnifică că are permisiunea să realizeze un ciclu cu relații din acest concept-relații.

Aserțiuni (assertions)

Aserțiunile, sunt formule logice adnotate modelului și codului său, sunt asociate noțiunii de concept-limbaj, ca de altfel la acele concept-descriere și concept-relație. Ele permit precizarea semanticii fiecărui program compozant și controlarea, în cursul utilizării sale, că folosirea se face conform specificațiilor date. Noi regăsim în OFL trei feluri principale de aserțiuni: invariante (condiție care trebuie verificată în fiecare moment), preconditionate (condiție care trebuie să fie verificată înainte de execuția unei metode) și postconditionate (condiție care trebuie verificată după fiecare execuție a unei metode). La fiecare concept-limbaj, este posibilă deci asocierea a două ansamble de invarianți.

Verificarea constrângerilor de utilizare a concept-descriptor și concept-relație asociate limbajului. În acest prim caz, se pot da următoarele două exemple:

Se caută verificarea regulii, că nici o circularitate nu trebuie să fie directă. Acest lucru se poate exprima simplu: Fie D1 și D2 descriptori și CR concept-relație, dacă D1 este sursa unui CR direct deci D2 este destinația, atunci D1 nu poate să fie destinația unui CR direct deci D2 este sursa. De notat că această condiție se aplică natural și fără precizarea suplimentară a relațiilor non-circulare.

Un alt exemplu de invarianță poate fi: Fie D1 și D2 descriptori și CR1 și CR2 concepte-relații opuse, dacă D1 este sursa lui CR1 deci D2 este destinație, atunci D1 trebuie să fie destinația lui CR2 deci D2 este sursa.

Într-o manieră mai generală, controlul validării constrângerilor structurale pentru tot limbajul. Și aici pot fi date cel puțin două exemple:

Acest invariant verifică că două relații opuse nu pot fi puse concomitent (un descriptor, de exemplu, nu poate în același timp să specializeze și să generalizeze altul). Parametrul care deține acest control este *Opposite*: Fie D1 și D2 descriptori și CR1 și CR2 concepte-relații opuse, dacă D1 este sursă în CR1 deci D2 este destinație, atunci D1 nu poate fi sursă în CR2 deci D2 este sursă.

Noi dorim de asemenea să verificăm că, fie orice limbaj descris, nici o destinație nu este realizată pentru concept-relație deci parametru *Circularity* este egal cu *forbidden*: Fie D1 și D2 descriptori și CR este concept-relație pentru care *Circularity* este egal cu *forbidden*, dacă D1 este sursă în CR deci D2 este destinație, atunci D1 nu poate fi destinație în CR deci D2 este sursă.

2.4. Programarea orientată pe aspecte (AOP) - Aspect J

Din ce în ce mai mulți dezvoltatori de aplicații sunt atrași de ideea folosirii programării orientate pe aspecte. Una din cele mai frecvente întrebări care se pune este de a integra această tehnologie în programele scrise anterior, ce beneficii ar oferi folosirea acestora, se pot descoperii ușor aspectele sau care sunt riscurile utilizării acestei tehnologii [30].

La ora actuală, de un succes deosebit, se bucură tehnologia programării bazată pe aspecte, dezvoltată pentru Java, de corporația Xerox împreună cu centre de cercetare din cadrul celor mai prestigioase universități [31].

2.4.1. Semantica Aspect J

O sugestivă reprezentare a semanticii programării bazate pe aspecte în cazul Java poate fi dată folosind un exemplu simplu pentru un simplu sistem de editare de figuri.

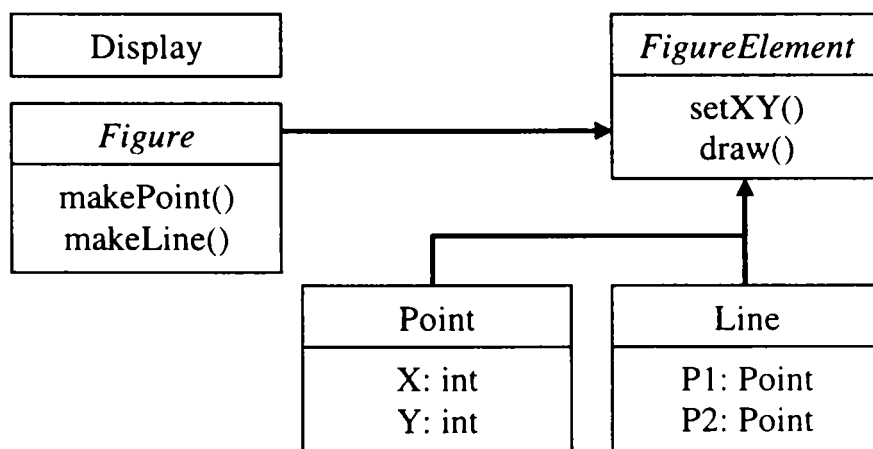


Figura 2-3 Exemplu de editare a unei figuri folosind tehnologia AOP

O figură *Figure* este formată dintr-un număr de elemente *FigureElement* care vor fi formate din puncte *Points* și linii *Lines*, iar clasa *Figure* asigură serviciile de construcție.

AspectJ realizează o modularizare a secțiunilor transversale ce folosesc puncte de legătură și recomandări. *Punctele de legătură* sunt puncte în fluxul programelor, bine definite iar recomandările definesc rutine de cod ce se execută în momentul apariției unui punct de legătură.

Modelul Join Point – al punctelor de legătură – un element critic în proiectarea oricărui limbaj de programare orientat pe aspecte este dat de definirea modelului punctului de legătură. Modelul punctelor de legătură oferă referința ce determină inserarea în execuție a unei structuri transversale de program. AspectJ oferă mai multe tipuri de puncte de legătură. Cele mai frecvente metode folosite și cele delicate sunt metodele de apelare a punctelor de legătură. Ele trebuie să asigure o corectă întrerupere a rulării programului, prin memorarea stării sistemului în momentul respectiv, astfel ca la reîntoarcerea la punctul de plecare, să se revină la starea inițială (cea anterioară apariției aspectului). În cazul execuției unei secțiuni transversale de program, pot să existe noi puncte de legătură spre alte secțiuni sau metode ce aparțin altor secțiuni.

Pointcut Designators – în cadrul AspectJ, *pointcut designator* permite identificarea punctelor de legătură din fluxul programului. Un exemplu, legat de ceea ce a fost prezentat în exemplul anterior este reprezentat de următorul *pointcut designator* :

```
call(void Point.setX(int))
```

Este astfel posibilă identificarea tuturor apelurilor ce utilizează metoda *setX* definită de obiectele *Point* (se pot folosi și combinații algebrice ale acestora). Programatorii pot defini propriile *pointcut designator* pentru mai multe puncte de legătură și pentru diferite clase de obiecte, se pot defini chiar clase pentru *pointcut designator*.

```
pointcut moves():
    call(void Point.setX(int))||
    call(void Point.setY(int))||
    call(void Line.setP1(point))||
    call(void Line.setP2(point));
```

2.4.2. Recomandări – Advice

Punctele de legătură sunt folosite pentru definirea recomandărilor. AspectJ are numeroase moduri de definire a recomandărilor, ce sunt definite ca și rutine adiționale ce vor rula la apariția punctelor de legătură.

Before advice ruleaz  c nd un punct de leg tur  este activat,  nainte de rularea unei sec iunii transversale. *After advice* ruleaz  c nd un punct de leg tur  este activat, dup  rularea unei sec iunii transversale. *Around advice* ruleaz  c nd un punct de leg tur  este activat,  i are controlul explicit peste toate calculele at ta timp c t punctul de leg tur   i permite acest lucru.

```
after() : moves () (System.out.println("Figura a fost mutat ");)
```

Dezvoltarea aplica iilor folosind aspecte ofer  o clar  modularizare  i face u oar  depanarea acestora. Astfel unele func ii pot fi validate sau invalidate ca func ionare foarte u or  n func ie de aspectele de rulare ale programului ce apar. Pot fi verificate, rulate  i compliate doar por iuni din program ceea ce faciliteaz  testarea unui produs software [32].

2.4.3. Concluzii privind implementarea aspectelor  n aplica iile multimedia

Implementarea aplica iilor folosind AspectJ ofer  o serie de avantaje  n compara ie cu o implementare standard:

- Structura sec iunilor transversale de program este dat   n mod explicit;
- Permite o foarte simpl  evolu ie a aplica iei;
- Se poate introduce foarte u or  n program un nou serviciu sau func ie prin intermediul punctelor de leg tur ;
- Se poate ob ine un bun control  i respectarea constr ngerilor de sincronizare prin producerea de aspecte bazate pe nume. Aceste aspecte sunt similare cu o schimbare a monitoriz rii controlului rul rii programului, cu precizarea c  func ioneaz  mai bine atunci c nd recomand rile devin foarte complexe, aceste aspecte folosesc o pereche de recomand ri *before – after* pentru  ndeplinirea sarcinilor de sincronizare;
- Transferul contextului  n cazul apari iei unui punct de leg tur  este foarte complex  n cazul unei aplica ii scrise  n Java, folosirea aspectelor permite modularizarea problemelor  i utilizarea doar a unui num r mic de metode ce gestioneaz  transferul contextului.
- AspectJ este o simpl   i practic  extensie a limbajului Java prin introducerea program rii orientate pe aspecte.. Cu doar c teva construc ii, AspectJ ofer  un suport pentru o implementare modular  de structuri transversale de cod.
- AspectJ folose te dou  tipuri de sec ionare: sec ionare bazat  pe nume  i sec ionare bazat  pe propriet i.
- Implementarea este mult mai sigur   i stabil .

2.5. JPVM - Java Parallel Virtual Machine

JPVM (Java Parallel Virtual Machine) este o libr rie ce permite unei aplica ii transferul mesajelor  ntr-un sistem bazat pe o teorie distribuit  MIMD (Multiple Instruction – Multiple Data)  n cazul program rii paralele  n Java. Libr ria suport  o interfa  similar  cu cea din C  i Fortran dezvoltate pe vechile sisteme Parallel Virtual Machine (PVM), dar cu sintax   i semantic  modificat   i adaptat  pentru Java.

JPVM este format  din dou  p r i – servere  i clien i. Clientul transmite procese spre servere sub form  de fire de execu ie. Serverele ruleaz  procesele  i returneaz  rezultatele c tre clien i. Clien ii  i serverele comunic  prin pachete de date de tip datagram. Clien ii re in IP-ul calculatoarelor pe care programele server lucreaz , transmit procese sub forma unor fire de execu ie  i recep ioneaz  r spunsul de la acestea. JPVM este implementat  n  ntregime  n Java. `jpvm-mEnvironment` Java class export  interfa a central  prin care JPVM interac ioneaz . Instan ele acestei clase sunt declarate ca o sarcina JPVM de conectare  i interac ionare cu

sistemul JPVM și cu alte procese ce se execută de sistem. Obiectele acestei clase reprezintă comunicații de tip end-points cu sistemul și sunt identificate de sistem prin intermediul unor identificatori opaci de tip `jpvmTaskId`. După ce o instanță `jpvmEnvironment` este alocată, pot fi invocate servicii de genul creare și transmitere de mesaje. Crearea unui task în JPVM este suportată de metoda `pvm_spawn()`, care folosește un parametru de tip șir de caractere pentru a indica un nume al unei clase valide Java vizibilă în mediul variabilelor CLASSPATH. Fiecare task creat prin `pvm_spawn()` este executat într-o instanță a Virtual Machine, evitându-se astfel conflicte de folosire a sistemului.

Transmiterea mesajelor în JPVM se realizează prin intermediul metodelor `pvm_send()` și `pvm_recv()` din `jpvmEnvironment` class. Întotdeauna înainte de a transmite datele, obiectul trebuie colectat în bufferul `jpvm-Buffer` object. Interfața `jpvmBuffer` conține două categorii de metode: cele care împachetează datele în buffer și cele care extrag datele din buffer. De exemplu, pentru a transmite o imagine clienților în scopul procesării, este necesar să împachetăm imaginea, considerând-o un vector de numere întregi (de exemplu, un vector format din nivelele de gri ale imaginii) folosind metoda publică `void pack(v[], int, int)` a `jpvmBuffer`. După ce conținutul a fost încărcat în `jpvmBuffer` object, bufferul poate fi trimis pentru procesare în sistemul JPVM folosind metoda `pvm_send()` [33], [34], [35], [36].

2.6. Concluzii și contribuții

În prezenta teză, am aplicat în practică conceptele de programare prezentate anterior. Am folosit acest mod de lucru, pentru a procesa un număr mare de imagini, într-o rețea de calculatoare eterogenă. Timpul de procesare a fost redus considerabil în acest fel. Nu au existat probleme legate de modul eterogen al configurațiilor calculatoarelor din rețea, deoarece distribuirea sarcinilor a ținut cont de faptul că răspunsurile de la calculatoarele client pot sosi cu anumite întârzieri. Am folosit doar puterea de calcul a acestora, fără a solicita resursele clienților (de exemplu, nu a fost stocat pe hardiscul clientului decât programul ce implementa serviciul client, rezultatele parțiale sau alte date ce rezultau în urma procesării fiind trimise direct serverului).

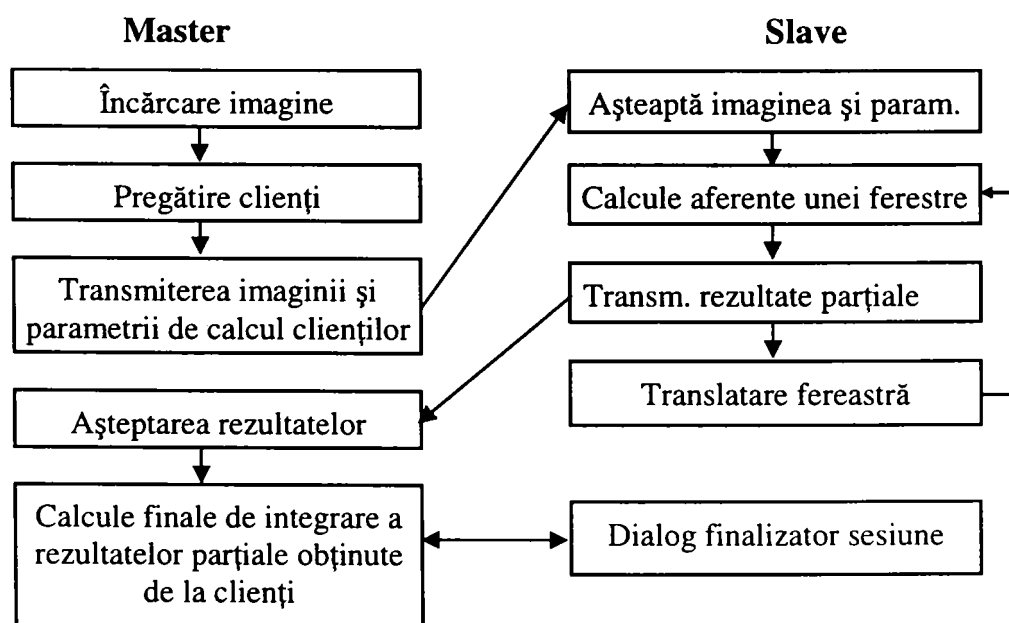


Figura 2-4 Exemplu de implementare a procesărilor de imagini în JPVM

Câștigul de viteză folosind acest mod de implementare depinde foarte mult de dimensiunea imaginii, calculatoarele folosite și tipul de rețea. Aplicațiile le-am rulat într-o rețea 100 MHz VG-ANY LAN, folosind 4 calculatoare HP PII 400 MHz, 64 Mb RAM sub Microsoft Windows NT 4.0, SUN JDK 1.2 și JPVM v 0.2. Detalii referitoare la rezultatele implementărilor au fost publicate de către autor la Conferința internațională "Communications 2000", de la București din 7 - 9 decembrie 2000, fiind prezentate în bibliografie detalii referitoare la rezultatele implementărilor [37].

Mediere pentru ferestre	Număr calculatoare	Timpul	Câștig de viteză
8x8, 10x10, 12x12	1	64''	1.06
	4	60''	
8x8, 12x12, 16x16	1	183''	3.12
	4	59''	
8x8, 16x16, 24x24	1	546''	3.47
	4	157''	

Tabelul 2-5 Rezultate experimentale privind extragerea caracteristicilor texturilor folosind o implementare sub JPVM

Folosind acest mod de implementare paralelă a procesărilor complexe, prin intermediul unor rețele de calculatoare eterogene, am realizat scurtarea duratei procesărilor, fără a fi necesară achiziționarea de calculatoare deosebit de puternice și foarte scumpe, cu alte cuvinte am realizat o eficientă exploatare a resurselor existente.

Capitolul 3.

Modelul propus pentru aplicațiile multimedia

3.1. Cerințe actuale pentru un mediu de dezvoltare pentru aplicațiile multimedia

Periada actuală poate fi caracterizată printr-o dezvoltare masivă a modurilor de prezentare și publicitate în care diverse companii își prezintă produsele și serviciile. Această dezvoltare a generat o nevoie acută de medii de integrare și de dezvoltare de aplicații multimedia care să satisfacă necesarul acestor companii în realizarea de medii publicitare.

Un alt element, ce poate defini situația actuală, caracterizează utilizatorii de medii de dezvoltare de aplicații multimedia, pentru aceștia, majoră este dorința de personalizare a mediului unde lucrează (adică, a interfeței grafice a mediului de dezvoltare) și mai ales de a introduce elemente personale în cadrul acestora. De exemplu, există dorința utilizatorilor de a defini efecte proprii, nu numai de a schimba parametrii unui model de efect gata definit (cazul unui mediu de editare neliniară video).

Din punct de vedere al companiilor ce realizează medii de dezvoltare a aplicațiilor multimedia, există o concurență deosebită de mare din partea altor companii dar și din partea unor dezvoltatori individuali. Pentru a contracara efectele concurenței, de foarte multe ori se apelează la utilizatorii individuali, care pot aduce importante îmbunătățiri pentru un mediu existent (cazul cel mai des întâlnit, este cel al utilizatorilor cu foarte multă experiență care ajung să cunoască foarte bine toate slăbiciunile mediului).

Pot exista și companii care dezvoltă diverși algoritmi de prelucrare a obiectelor multimedia, algoritmi care de foarte multe ori sunt mai performanți decât cei dezvoltați de compania ce realizează medii de dezvoltare a aplicațiilor multimedia. Majoritatea utilizatorilor doresc să folosească acești algoritmi, și, de aceea, se impune ca un mediu de dezvoltare, respectiv o aplicație multimedia să poată accepta introducerea acestora din exterior.

Utilizatorii de aplicații multimedia au tendința de a dezvolta o aplicație existentă fără a apela la dezvoltatorul inițial al aplicației. Cazul cel mai des întâlnit este cel al aplicațiilor ce au nevoie de foarte multe pagini care respectă în general un anumit model. De exemplu, o companie care dorește să își prezinte pe Internet un număr foarte mare de produse cu caracteristici asemănătoare (ex. produse chimice, la care informația vizuală asupra produsului este de mică importanță și unde contează mai mult componenta chimică a produsului), prin introducerea de noi produse în procesul de fabricație este necesară și prezentarea acestora. Se pot realiza prezentări ce folosesc ca tehnologie, generarea dinamică a paginilor, pornind de la un model pentru pagina respectivă și de la o bază de date ce stochează informațiile despre produse.

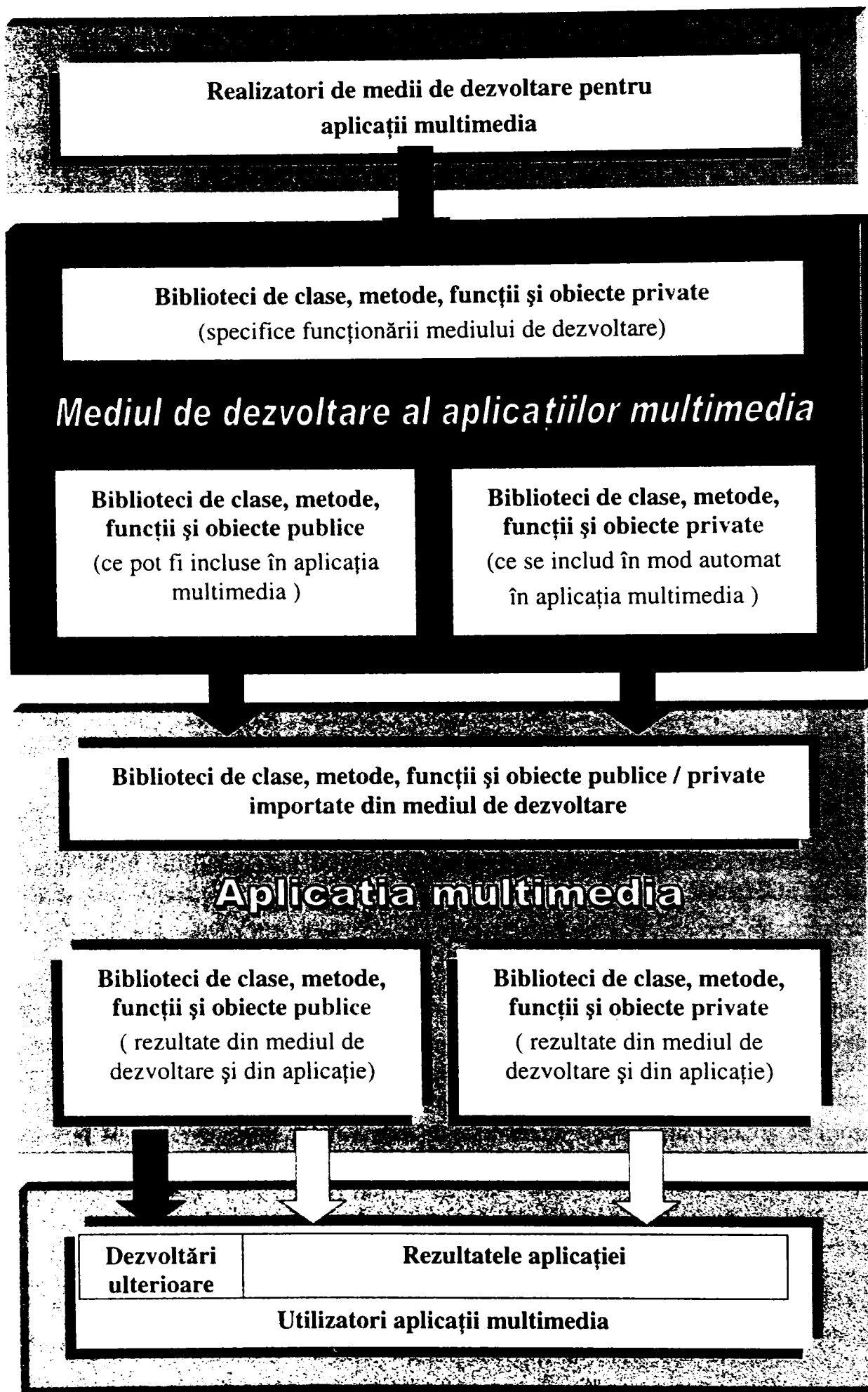


Figura 3-1 Încadrarea noului model de aplicație multimedia în relația producător medii de dezvoltare a aplicațiilor – dezvoltator aplicații – utilizator aplicații multimedia

O altă tendință actuală este reprezentată de facilitarea accesului la informații prin intermediul Internetului. Aplicațiile multimedia ce se dezvoltă în prezent, sunt dotate și cu facilități de accesare prin Internet.

În asemenea condiții este necesară folosirea de modele de aplicații *deschise* pentru dezvoltatorii de aplicații multimedia și pentru utilizatorii acestora. În continuare propun un model care să rezolve o bună parte din problemele existente.

3.2. Structura modelului propus pentru aplicații multimedia

Modelul propus pentru aplicațiile multimedia este constituit din mai multe module ce interacționează foarte strâns între ele, dar care oferă prin modularitatea lor și un grad mare de reutilizare, lucru ce se înscrie în tendințele actuale din domeniul aplicațiilor multimedia.

Modulele se structurează în patru mari categorii în jurul nucleului principal al aplicației. **Nucleul** - în general este o parte specifică fiecărei aplicații în parte și nu permite reutilizări și modificări din partea utilizatorilor. Rolul nucleului este de a gestiona întreaga aplicație, prin utilizarea de funcții, rutine și proceduri existente în cadrul modulelor. Nucleul are și rolul de verificare și validare, înainte de folosire, a unei noi funcții implementare de către utilizator.

Cele patru categorii de structurare a modulelor sunt:

- **Funcții și procesări obiecte** - procesările pot fi la rândul lor de două tipuri:
 - *Procesări temporale*
 - *Procesări obiecte* – există două tipuri de procesări de obiecte:
 - Procesări ce afectează un singur obiect
 - Procesări ce utilizează mai multe obiecte
- **Modul pentru interacțiunea cu utilizatorul** – se caracterizează prin interfața grafică și prin evenimentele generate în urma interacțiunii aplicație-utilizator.
- **Operații de intrare/ieșire** – operațiile de intrare/ieșire conțin și o componentă ce permite lucrul în rețea, dezvoltarea de servicii de tip client/server și un sistem de validare a utilizatorilor.
- **Tehnologii integrate modelului aplicației** – acest modul conține biblioteci dedicate lucrului cu baze de date (ODBC), biblioteci de integrare a obiectelor realizate de către alte aplicații (tehnologie OLE), implementări de servicii de tip client/server (DDE), biblioteci cu funcții ce optimizează funcționarea diverselor echipamente din interiorul unui PC (funcții MCI pentru placile de sunet, video, CD-ROM), biblioteci de evaluare a calității serviciilor oferite de către aplicație pe resursele hardware oferite de utilizator, biblioteci de funcții ce permit căutarea de obiecte și resurse ce sunt folosite de aplicația multimedia, etc.

Structurarea modulelor funcționale ale modelului propus, în funcție de modul lor de utilizare și de operare (interfațare) cu utilizatorul este prezentat în figura următoare:

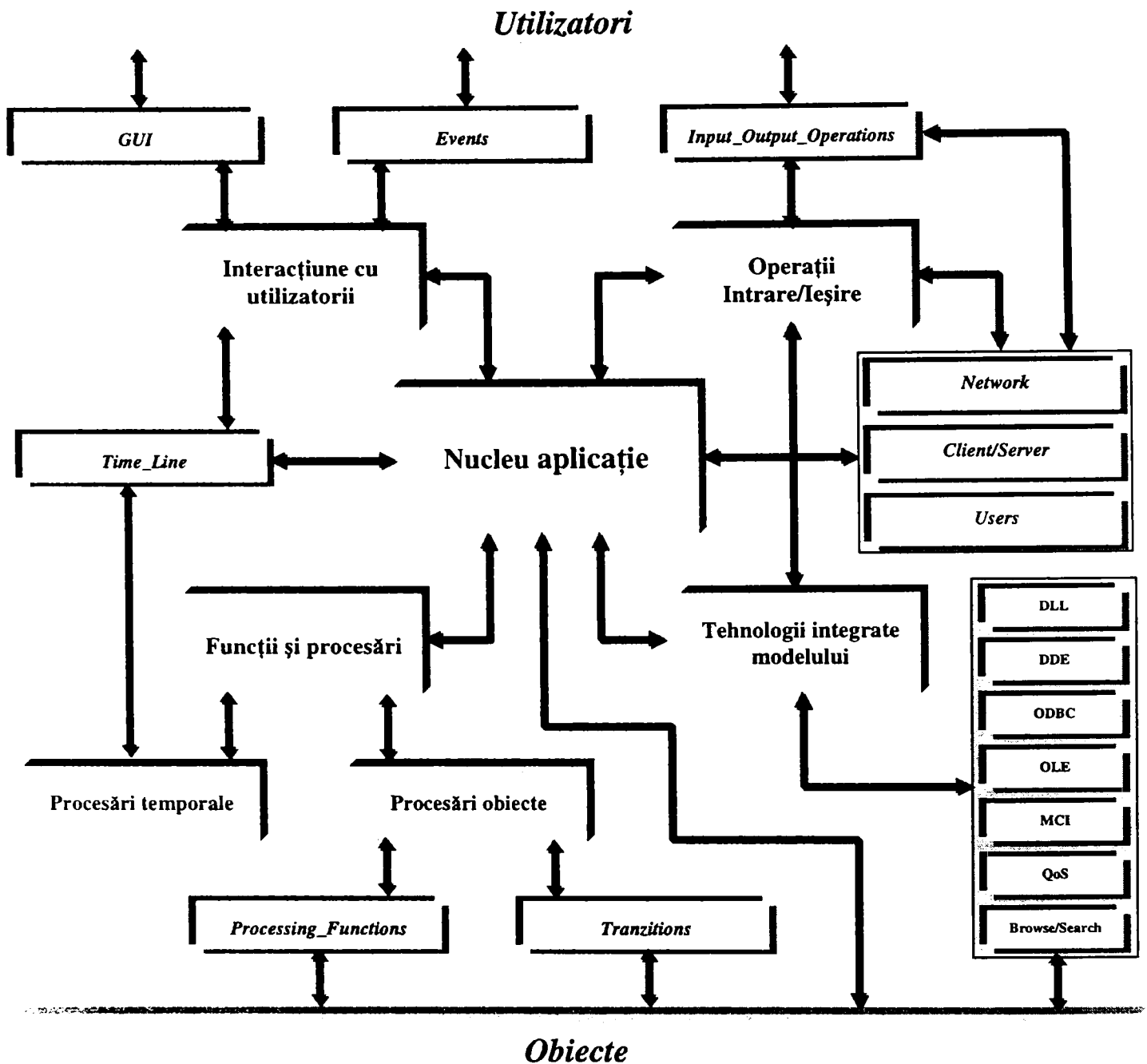


Figura 3-2 Interconectarea modulelor în cadrul modelului propus

În paragrafele următoare voi explica în detaliu modulele constituente ale modelului propus. Voi acorda mai multă atenție descrierii obiectelor multimedia ce se folosesc în aplicații deoarece există o legătură foarte strânsă între acestea și celelalte module ce formează modelul de aplicație multimedia.

3.3. Modelul descriptiv al obiectelor multimedia: *Objects*

Fiecare obiect multimedia are o valoare, o structură logică pentru acea structură și o interpretare a conținutului pentru acea valoare. Valoarea unui obiect multimedia, structura sa logică și modelul de reprezentare și stocare a acestuia se află în exprimată în modelul de descriere al obiectului multimedia.

Modelul de descriere multimedia conține informații referitoare la compoziția logică a acestui obiect, constrângeri de sincronizare și temporizare pentru componentele acestuia, parametri necesari pentru afișare, etc. Modelul de interpretare trebuie să descrie cât mai exact lumea reală prin prisma sub-obiectelor ce îl compun și prin relațiile ce există între acestea, respectiv, între fiecare sub-obiect component și lumea reală.

Un model de descriere al unui obiect multimedia este dat în figura următoare:

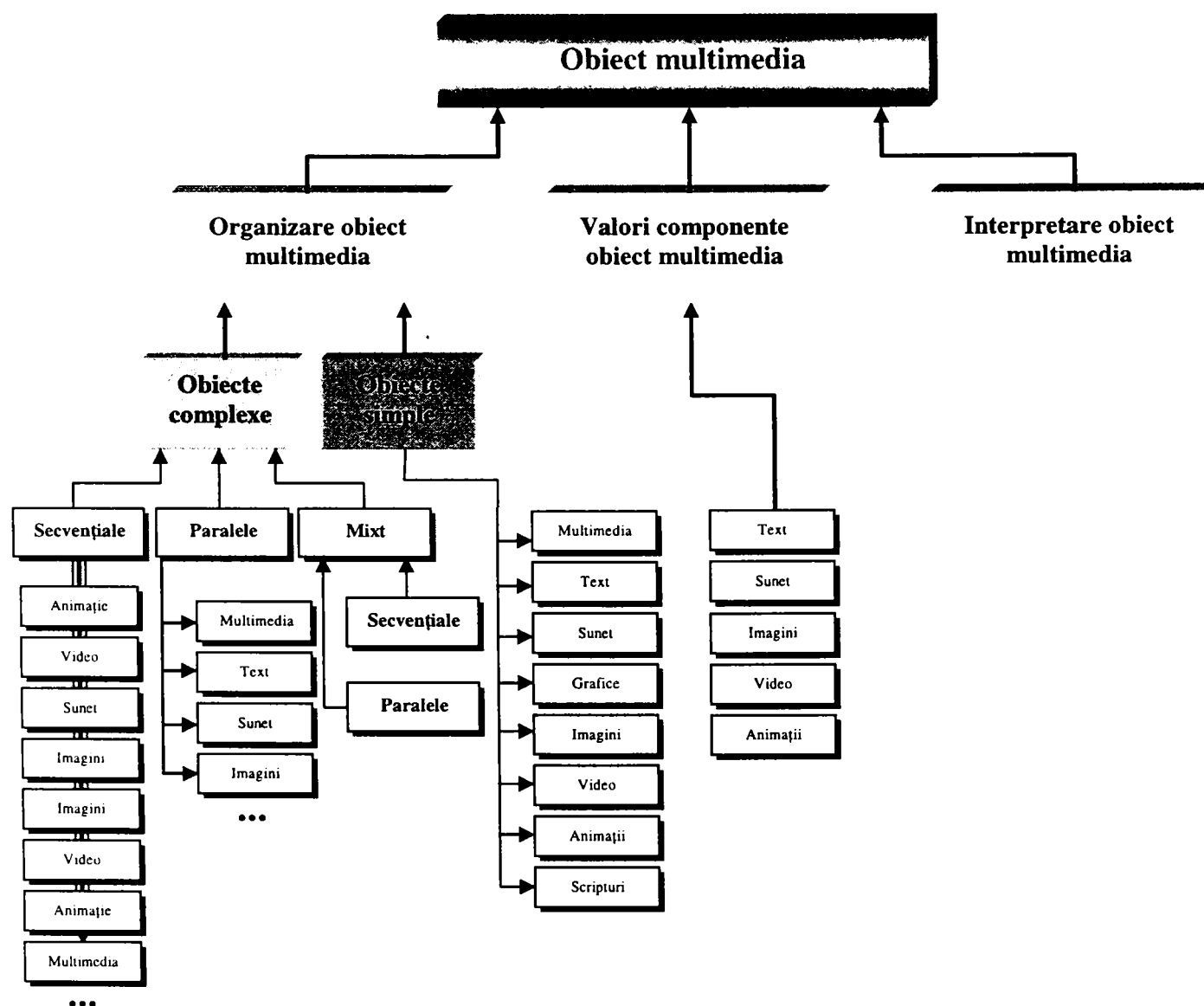


Figura 3-3 Descrierea unui obiect multimedia prin cele trei componente: organizare obiect, valoare componente obiect, interpretare

Pentru o descriere eficientă a obiectelor multimedia am propus o structurare a parametrilor de descriere în următorul mod:

1. Parametrii generali ce descriu obiectul – de exemplu, identificatorul obiectului, locația fizică de stocare a obiectului, flag care indică dacă obiectul este simplu sau complex și informații despre elementele din structura obiectului.
2. Parametrii ce descriu semantic obiectul prin cuvinte cheie.
3. Parametrii specifici de descriere a obiectului multimedia – descriu obiectul, dimensiuni, mod de compresie, aplicația ce a generat obiectul (de exemplu, număr de culori, dimensiune, rezoluție, în cazul imaginilor, număr de canale, rată de eșantionare, format, în cazul sunetului, etc.).
4. Parametrii globali de descriere ai obiectelor – descriu conținutul obiectelor (utili de exemplu în cazul unei baze de date ce acceptă o căutare bazată pe conținut).
5. Parametrii utilizator – specificați de fiecare utilizator, ce sunt caracteristici doar pentru acea aplicație.

Un obiect multimedia poate fi compus din alte obiecte multimedia sau din elemente simple (necompușe). Un obiect multimedia poate fi compus din următoarele elemente:

- texte
- sunete
- grafice
- imagini
- secvențe video
- animații
- scripturi

Modelul de descriere multimedia poate fi sintetizat astfel:

- Fiecare obiect multimedia este compus dintr-un număr de sub-obiecte simple sau complexe, care sunt organizate secvențial, paralel sau în mod mixt (secvențial și paralel). Un exemplu de asemenea obiect, poate fi de exemplu, un generic pentru un material video: genericul poate fi format dintr-o succesiune de imagini, deci o organizare secvențială, urmată apoi de o parte ce conține un text suprapus peste o compoziție de mai multe imagini statice (sau secvențe video), deci o secvență cu obiecte organizate într-o structură paralelă. Un dinamism deosebit se obține în momentul organizării mixte, combinate a celor două moduri prezentate mai sus.
- Elementele din structura obiectului pot lua o anumită valoare, valoare ce definește o instanță a aceluși obiect. Nu este necesar ca pentru un obiect să fie definite valori pentru toate componentele sale. Un obiect se definește prin valoarea doar a unor componente. Definind alte componente se poate obține un nou obiect. Se folosește conceptul de moștenire de la programarea orientată pe obiecte.

Pentru o descriere cât mai apropiată de lumea reală se poate stabili și o interpretare semantică pentru fiecare componentă a unui obiect multimedia. Descrierea semantică devine deosebit de utilă în cazul căutării unor obiecte în cadrul aplicației multimedia, sau atunci când numărul mare de obiecte constituite ale aplicației necesită o structurare, o grupare a elementelor după un concept semantic.

În continuare voi prezenta parametrii pe care îi consider importanți de mine pentru descrierea unui obiect multimedia și a elementelor simple. Am organizat descrierea obiectelor într-o structură de clase ce au proprietățile asemănătoare claselor ce se întâlnesc în cazul programării orientate pe obiecte folosind Java sau C++.

Structura ierarhiei claselor este prezentată în continuare:

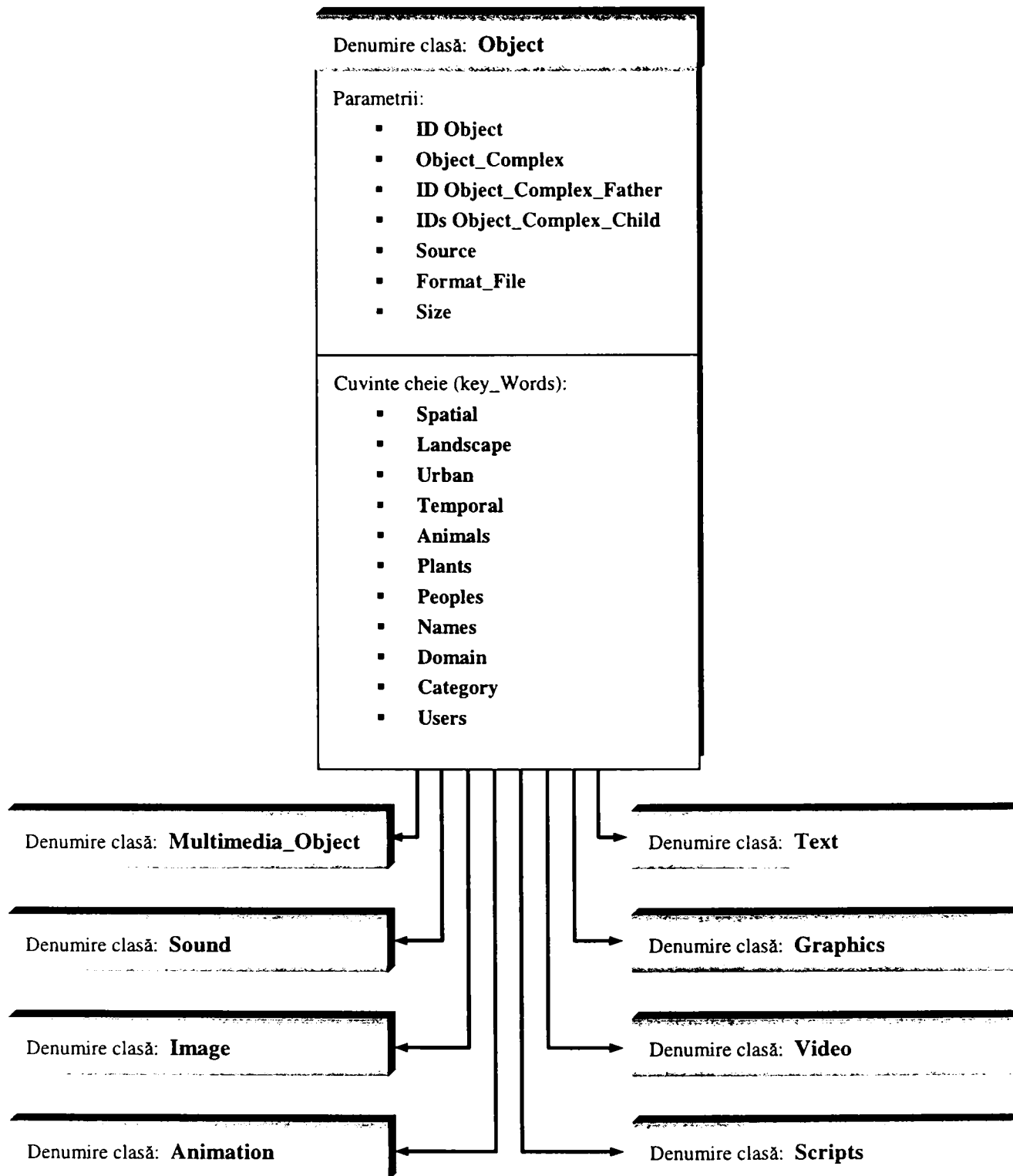


Figura 3-4 Structura ierarhică a obiectelor - organizarea pe clase

- Fiecare obiect dintr-o aplicație este caracterizat de o structură de bază de parametrii ce permit identificarea acestuia prin **ID Object**.
- Indicarea structurii complexe a obiectului se face prin flagul **Object_Complex**, iar explicitarea structurii prin intermediul:
 - ID Object_Complex_Father
 - IDs Object_Complex_Child

Folosirea primului identificator permite stabilirea obiectului părinte iar următorii identificatori arată care sunt sub-obiectele legate de obiectul în discuție.

- Am considerat necesară specificarea explicită a sursei, a locului și a modului de stocare fizică a obiectului. Pentru aceasta se folosesc parametrii:
 - **Source**
 - **Format_File**
 - **Dimension**

Un obiect multimedia poate avea în componența sa unul sau mai multe obiecte sau elemente primare simple sau complexe. Prin elementele primare ale unui obiect multimedia înțeleg elemente de tip: text, sunet, grafice, imagini, secvențe video, animații sau scripturi.

Descrierea obiectului multimedia din punct de vedere semantic se poate face prin cuvinte cheie structurate pe 11 categorii. O asemenea descriere semantică, permite utilizarea obiectului respectiv în mai multe tipuri de aplicații. De exemplu, o imagine care conține un peisaj cu munte, pădure și câteva animale poate fi utilizată în diverse situații, colecții de imagini cu natură sau colecții de imagini cu plante, cu animale, etc. Un rol deosebit îl are această descriere semantică în cadrul bazelor de date, unde o descriere semantică poate reduce semnificativ volumul căutărilor (în special în cazul căutărilor bazate pe conținutul obiectelor).

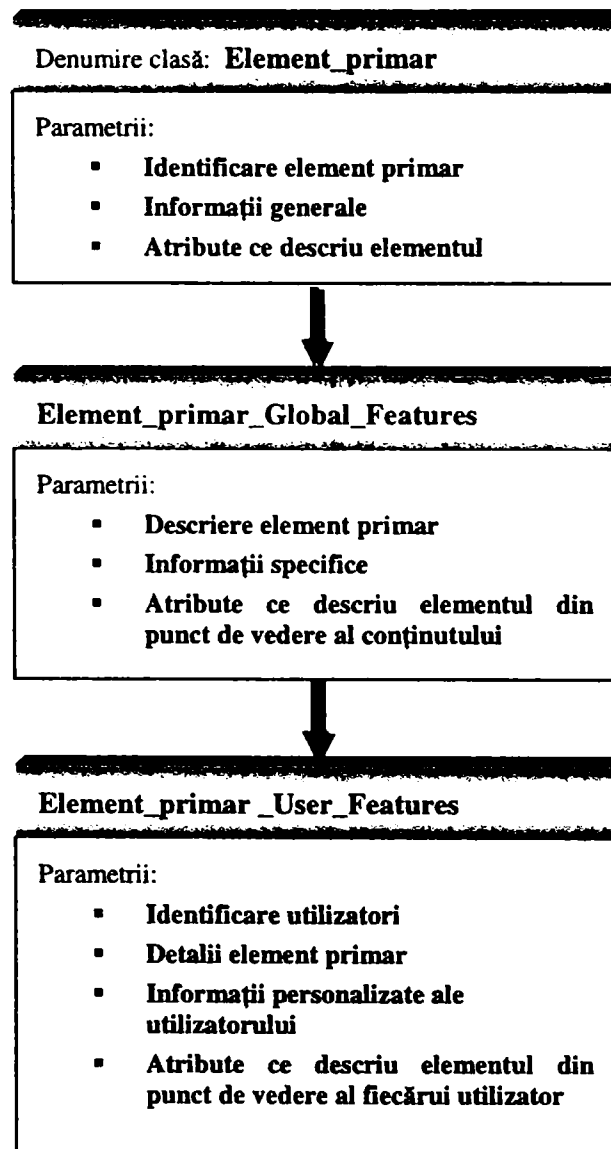


Figura 3-5 Structura ierarhică de descriere a unui element primar ce formează obiectul multimedia

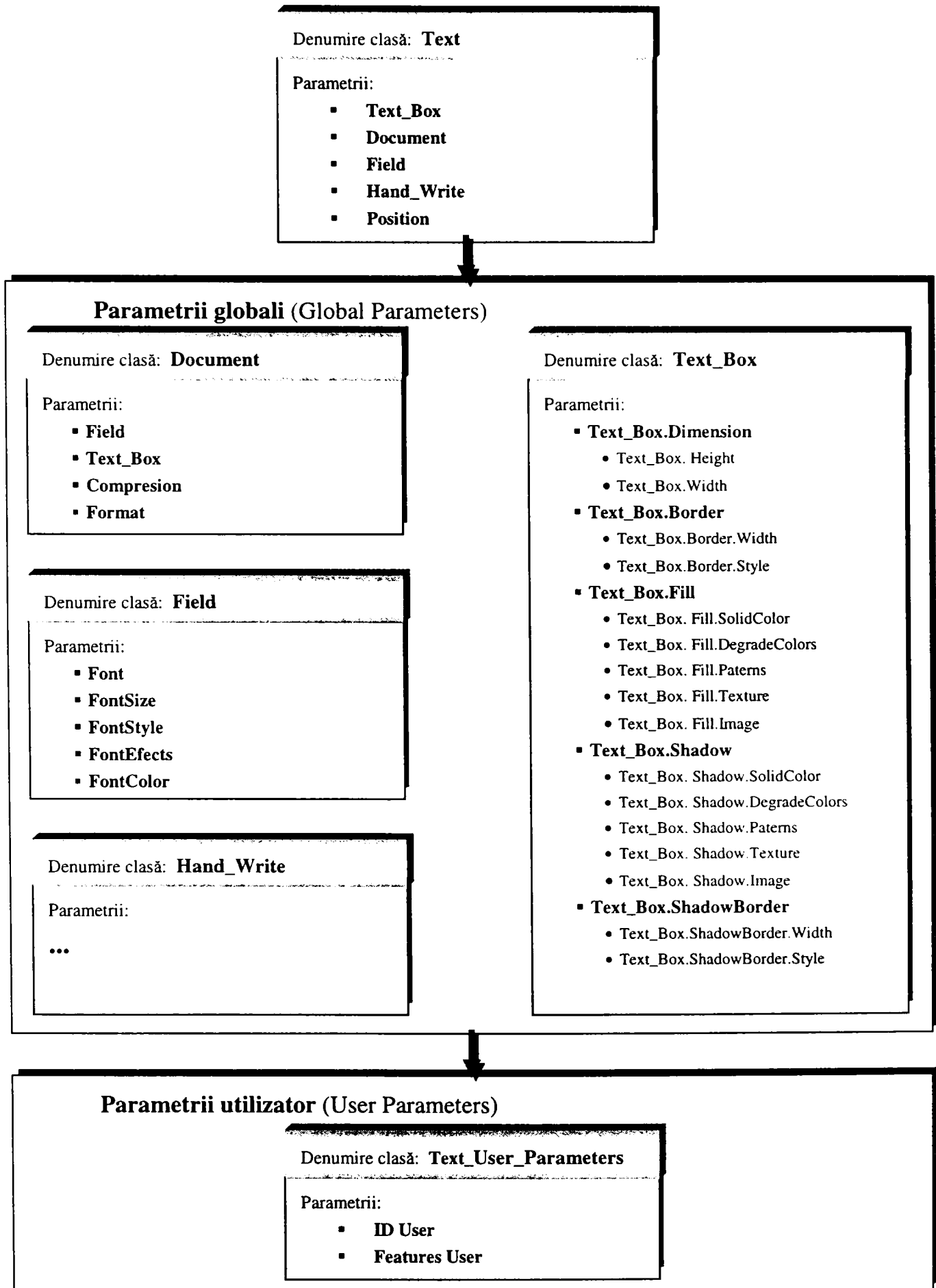


Figura 3-6 Detalierea structurii ierarhice pentru un elemet primar de tip text

După descrierea obiectului, urmează descrierea elementelor primare ce îl formează. Un element primar este constituit din descrierea sa generală, care îl identifică ca tip, urmată de o descriere a conținutului prin atributele globale și de o descriere specifică fiecărui utilizator în parte, prin intermediul atributelor utilizator. Astfel fiecare obiect poate avea o personalizare în funcție de utilizatorul ce îl folosește. Cazul general și detalierea pentru un element primar de tip text a fost dată în figurile de mai sus.

Parametrii generali de descriere a elementelor primare sunt:

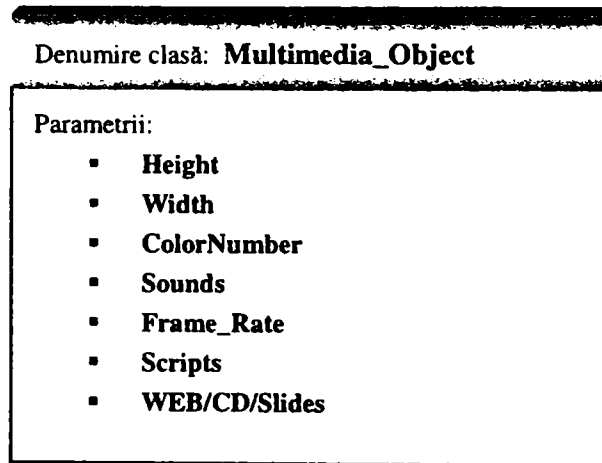


Figura 3-7 Multimedia_Object

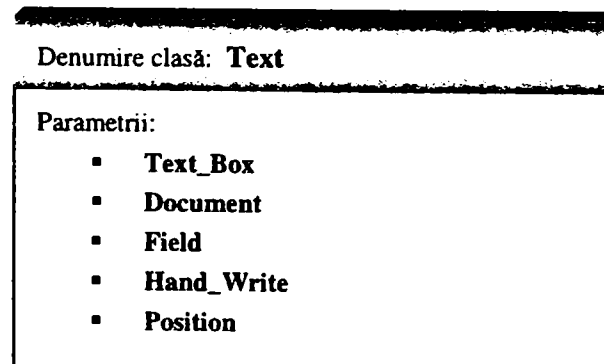


Figura 3-8 Text

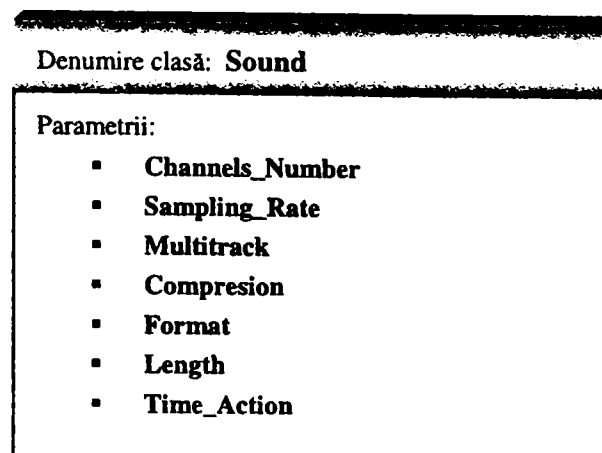


Figura 3-9 Sound

Denumire clasă: **Graphics**

Parametrii:

- **Dimension**
- **Color_Numbers**
- **Color_Model**
- **Layers**
- **Compresion**
- **2D/3D**

Figura 3-10 Graphics

Denumire clasă: **Image**

Parametrii:

- **Dimension**
- **Color_Numbers**
- **Color_Model**
- **Layers**
- **Compresion**
- **Real/PC**

Figura 3-11 Image

Denumire clasă: **Video**

Parametrii:

- **Dimension**
- **Color_Numbers**
- **Color_Model**
- **Sound**
- **Format**
- **Compresion**
- **Frame_Rate**
- **Lenght**

Figura 3-12 Video

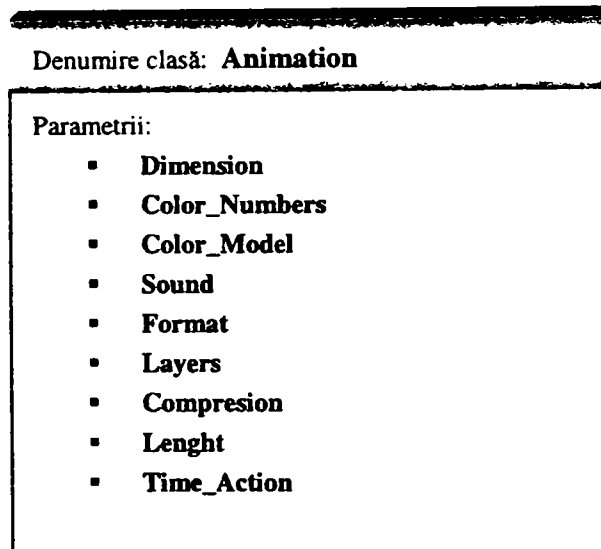


Figura 3-13

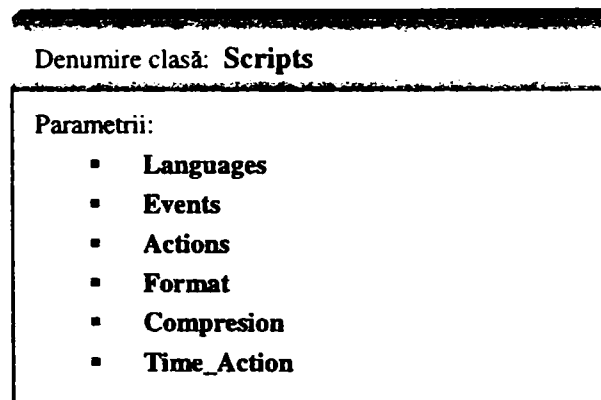


Figura 3-14

Parametrii de descriere a elementelor primare sunt aleși astfel încât să descrie elementul, dar se evită transmiterea de informații redundante sau cu semnificații caracteristice pentru o anumită aplicație.

Elementele **Graphics** includ imaginile generate vectorial, prin folosirea unor funcții matematice. Acestea la rândul lor sunt create din forme geometrice elementare (generate de anumite funcții).

Există animații care pornesc de la câteva obiecte și reguli de mișcare (traectorii, viteză de mișcare, rotații, redimensionări). Acestea nu există ca un simplu fișier, ci ele se generează la rularea unei mici aplicații. Pentru acest tip de animații am definit elementul primar **Animation**.

O aplicație multimedia, care prezintă un anumit grad de interactivitate cu utilizatorul, se folosește de rutine de program (scrise sub diverse limbaje de programare). Acestea definesc modul de rulare al aplicației în funcție de opțiunile utilizatorului. Aceste rutine sau biblioteci de rutine se regăsesc în cadrul elementelor primare de tip **Scripts**.

Caracteristicile globale ale conținutului elementelor primare depind în mare măsură de aplicația în care sunt utilizate, de aceea, nu voi detalia aceste aspecte în cadrul acestui capitol. În cadrul capitolului 6, la descrierea bazei de date implementate, voi trata în detaliu aceste aspecte.

3.4. Funcții și procesări

Funcțiile și procesările de obiecte pot fi clasificate astfel:

- *Procesări temporale* – procesări ce folosesc aspectele temporale ale obiectelor.
- *Funcții de interacțiune cu utilizatorul* – Interfața grafică cu utilizatorul *GUI* și sistemul de evenimente *Events* ce pot fi declanșate de către acesta.
- *Procesări obiecte* – există două tipuri de procesări de obiecte:
 - Procesări ce afectează un singur obiect
 - Procesări ce utilizează mai multe obiecte

3.4.1. Procesări temporale: *Time_Line*

Pentru o funcționare corectă, într-o aplicație multimedia, în care sunt rulate diverse sunete, secvențe video sau animații, trebuie să existe o coordonare a obiectelor media, care să țină cont și de resursele hardware ale sistemului pe care este aplicația.

Modul de rulare a aplicației este ghidat printr-o serie de referințe temporale stabilite de către realizatorul aplicației, dar trebuie să respecte și preferințele utilizatorului. Rutinele ce formează biblioteca de funcții *Time_Line*, trebuie să permită o ușoară implementare pentru dezvoltatorul de aplicații multimedia, iar pentru utilizator trebuie să reprezinte o anticipare a tuturor acțiunilor posibile ce vin din partea acestuia, astfel încât controlul obiectelor din aplicație să fie perfect.

Cazurile cele mai frecvente de greșeli de programare și de nerespectare a desfășurării temporale (nerespectarea constrângerilor temporale) se întâlnesc atunci când programatorul nu anticipează faptul că, dacă utilizatorul se hotărăște să părăsească o anumită scenă (în care se derulează o animație de exemplu și mai există și un sunet de fundal redat separat) și să treacă la o altă scenă, fără oprirea obiectelor de tip sunet, animație sau secvențe video și eliberarea resurselor hardware folosite de acestea, va conduce:

- reducerea resurselor calculatorului, ceea ce face ca aplicația să ruleze mai încet (cazul cel mai puțin deranjant);
- la redarea simultană de mai multe obiecte;
- blocarea calculatorului prin conflicte de redare a diverselor obiecte media.

Se impune astfel ca fiecare aplicație să conțină o așa zisă “linie a timpului”, adică rutine care să gestioneze corecta desfășurare a aplicației, indiferent de acțiunile utilizatorului.

Recomandarea rezultată în urma mai multor ani de experiență este de a implementa rutine care să verifice automat, la apariția unei acțiuni din partea utilizatorului, situația resurselor hardware (dacă sunt ocupate de către obiecte media ce rulează sau pot utilizate ca urmare a noilor opțiuni exprimate de utilizator).

Rezultatul rulării rutinelor din biblioteca *Time_Line* este:

- Gestionarea obiectelor cu constrângeri temporale și a resurselor care redau aceste obiecte;
- Verificarea în prealabil, înainte de rularea fiecărui obiect, dacă resursa hardware ce va reda este liberă, iar dacă nu să se forțeze eliberarea acesteia.
- Realizarea unei liste de referințe temporale care să fie utilizată și de bibliotecile de procesare a obiectelor ca și parametrii de procesare (de exemplu, stabilirea

momentului de începere și de finalizare a tranziției, printr-un efect, de la un obiect la altul, poate fi considerată ca definiție a parametrilor de procesare a efectului de trecere de la un obiect la altul).

- Realizarea de salturi la diverse referințe temporale în funcție de acțiunea utilizatorului – colaborarea cu bibliotecile de funcții *GUI* și *Events*.

Aici se poate apela la avantajele programării orientate pe aspecte (prezentate în capitolul 2). Rezultatele cele mai bune se obțin prin combinarea celor două tehnici, programare orientată pe obiecte și cea orientată pe aspecte (pentru gestionarea evenimentelor temporale).

3.4.2. Funcții de interacțiune cu utilizatorul: *GUI*, *Events*

Interfața grafică cu utilizatorul – *GUI*

Interfața grafică cu utilizatorul este formată din diverse elemente de grafică, obiecte ce oferă interactivitatea dintre utilizator și aplicație (butoane, legături, imagini, mici animații, etc.). Acestor obiecte li se atașează rutine ce se declanșează la apariția unuia dintre evenimentele definite în cadrul bibliotecii de funcții, metode și evenimente *Events*.

Interfața grafică poate conține elemente deja existente în cadrul sistemului de operare sau poate avea propriile elemente create de realizatorul aplicației. De foarte multe ori, sunt medii eterogene, ce conțin obiecte mixte, din ambele categorii. Folosirea obiectelor și a funcțiilor din cadrul sistemului de operare oferă o rapiditate în implementare. Rezultate superioare, din punct de vedere vizual, se obțin prin folosirea unor obiecte personalizate pentru o aplicație și scrierea propriilor rutine poate crește fiabilitatea aplicației.

Sistemul de evenimente *Events*

Modificarea modului de rulare a unei aplicații depinde de acțiunile efectuate de către utilizatorul aplicației. Acesta, declanșează un eveniment, iar ca urmare vom avea o anumită acțiune. Este foarte importantă definirea corectă a sistemului de evenimente ce pot să apară și a acțiunilor ce se realizează în urma acestora.

Este necesară definirea unei ierarhii de priorități de execuție:

- în cazul suprapunerii de mai multe evenimente în același timp
- în cazul în care un eveniment poate declanșa mai multe acțiuni.

Pot da exemple practice pentru ambele situații: un clic cu butonul de la mouse poate declanșa saltul la o nouă scenă, rularea unui sunet și oprirea unei animații. Ordonarea modului de execuție (ordinea de rulare a acestora) definește buna funcționare a aplicației. Deci, trebuie oprit sunetul și eliberate resursele hardware, realizat saltul la scena nouă și apoi rulat animația din noua locație.

În aplicațiile multimedia se lucrează cu obiecte ce se află pe diverse straturi. Obiectele de foarte multe ori se suprapun, existând zone comune. Pentru fiecare obiect, la o anumită acțiune – clic cu butonul de la mouse pe acel obiect – poate declanșa câte un eveniment. Pentru zonele suprapuse trebuie stabilită prioritatea de execuție: în funcție de ordinea straturilor pe care se află obiectele, dimensiunea obiectelor, etc.

Realizatorul aplicației trebuie să gestioneze foarte exact aceste situații și să creeze ierarhiile necesare, astfel ca un eveniment să lanseze în execuție rutina aferentă.

3.4.3. Procesări obiecte multimedia: *Processing_Functions, Transitions*

Procesări obiecte singulare: *Processing_Functions*

Pentru reducerea spațiului ocupat de aplicațiile multimedia, în contextul creșterii puterii de calcul, propun utilizarea conceptului “on-line processing” ce presupune procesarea obiectelor în timpul rulării aplicației. Astfel, prin procesarea obiectelor în timpul rulării aplicațiilor se poate folosi un singur obiect stocat fizic în mai multe ipostaze (în diverse scene el poate îmbrăca diverse forme – poate fi scalat, rotit, deformat, etc.). Dacă aplicația folosește un număr foarte mare de obiecte ce suferă aceleași procesări, gestionarea acestora poate fi simplificată folosind ideea de procesare în direct și nu de memorare a acestora.

Pot exista două situații:

- obiectul nou generat poate fi stocat (utilizatorul își poate personaliza obiecte multimedia pornind de la un obiect model);
- obiectul nou poate fi distrus după utilizare (la o nouă apelare el poate fi din nou regenerat).

Dacă procesările sunt foarte complexe, există riscul ca durata calculelor să fie mai mare, iar rezultatul să nu poată fi obținut instantaneu. În astfel de cazuri, recomand folosirea unor tehnici de “temperare a nerăbdării utilizatorilor”. Acest lucru se poate face prin folosirea de mici animații care să distragă atenția utilizatorului, afișarea unei clepsidre, sau a unui ceas care să indice timpul rămas până la terminarea procesării, anticiparea procesării și începerea ei cu câteva secunde înainte – realizarea în avans a unei preprocesări și terminarea ei doar dacă utilizatorul dorește acest lucru.

Funcțiile de procesare au un grad mare de diversitate și depind de elementul procesat. Uneori implementarea acestora poate fi anevoioasă. Recomand structurarea acestora, în cadrul unei biblioteci de funcții de procesare *Processing_Functions*, care să poată fi accesată ulterior. Pentru utilizatori experimentați se poate permite și personalizarea acestor funcții prin modificarea parametrilor de procesare.

Funcțiile din acest pachet (bibliotecă) folosesc în general un singur obiect în cadrul procesării.

Procesări între obiecte: *Transitions*

În timpul rulării unei aplicații multimedia, nu se cunoaște momentul în care utilizatorul hotărăște trecerea la o altă scenă a aplicației. Pentru a realiza o trecere cât mai interesantă de cele mai multe ori se folosesc tranziții sau efecte de trecere. În cazul în care în scene există obiecte de tip secvențe video sau animații, acestea au o altă stare în fiecare moment, de aceea nu se pot precalcula tranzițiile dintre cele două scene (tranzițiile dintre obiectele din scene). Procesările se vor face la momentul hotărât de utilizator, atunci când acesta dorește trecerea. Și în acest caz va fi o procesare *on-line*, ce va utiliza din plin resursele sistemului pe care rulează aplicația. Nu se pune problema, în acest caz, de stocare a rezultatului procesării.

Sunt însă și situații în care utilizatorul dorește diverse procesări între obiectele aplicației multimedia. Dacă aceste obiecte sunt foarte mari (ambele de tip video sau animație, etc.), și procesarea se face *on-line* este foarte probabil să avem probleme din cauza resurselor disponibile. De aceea, rutinele care prelucrează obiectele, trebuie să adopte o tehnică bazată pe încărcarea parțială în buffere de memorie a unor porțiuni doar din obiecte.

Pentru o reutilizare a procesărilor (efecte, tranziții) recomand organizarea în cadrul unei biblioteci de funcții de procesare distincte *Transitions*. Este necesar, ca pentru aceste funcții, să existe o transparență a parametrilor de procesare, utilizatorii reușind să își personalizeze aplicațiile. Pe lângă personalizarea parametrilor, este bine ca pentru dezvoltatorii experimentați

de aplicații multimedia, să existe instrucțiuni asupra modului de implementare de noi funcții, care ulterior să poată fi distribuite în versiuni ulterioare ale mediului de dezvoltare a aplicațiilor multimedia.

3.5. Operații de intrare – ieșire

3.5.1. Operații de intrare – ieșire: *Input_Output_Operations*

Operațiile de intrare/ieșire reprezintă modul de interacțiune al aplicației cu exteriorul (în special cu resursele hardware ale sistemului pe care rulează aplicația). Aceste funcții se regăsesc în toate limbajele de programare. Propun scrierea de mici rutine care să verifice starea perifericelor înainte de utilizarea lor, eliberarea unor resurse ocupate din cauza unor blocaje și determinarea performanțelor perifericelor. În colaborare cu funcțiile din cadrul bibliotecii *QoS* – Servicii de verificare a calității serviciului oferit (verificarea condițiilor de rulare corectă a unei aplicații multimedia pe un anumit sistem) se impune introducerea de rutine de captare a tuturor erorilor de funcționare ce apar. Împreună cu funcțiile *QoS* se vor genera mesaje care să informeze utilizatorul despre problemele apărute și să propună remedieri ale acestor probleme.

3.5.2. Conectarea în rețea (biblioteca *Network*)

Pentru aplicațiile ce relează în rețea se vor folosi funcțiile integrate în cadrul bibliotecii *Network*. Deoarece există o mare diversitate de tipuri de rețele, iar dezvoltarea acestora este permanentă, se va verifica pentru început tipul de rețea existentă, după care se vor apela rutinele specifice pentru acel tip de rețea. O asemenea structurare, ar permite o mai ușoară adaptare a aplicației în funcție de resursele hardware și software ale rețelei. Identificarea rețelei va determina încărcarea driverelor necesare pentru conectarea în rețea. Extinderi ulterioare presupun rescrierea unor drivere, iar pentru utilizator instalarea acestora nu prezintă în general probleme.

Problemele mai delicate apar atunci când aplicația este instalată pe mai multe calculatoare în rețea, se poate utiliza un server pentru instalarea celei mai mari părți ale aplicației iar pe calculatoare doar un minim. Traficul în rețea, numărul de utilizatori ce accesează aceeași aplicație influențează buna funcționare a unei aplicații multimedia distribuite în rețea.

Recomand, ca rutinele scrise pentru biblioteca *Network*, să permită verificarea stării rețelei, iar la apariția unor probleme (blocaje, întreruperea legăturii) să se încerce evitarea abandonării aplicației prin încercări repetate după un anumit algoritm, de reluare a conexiunii, iar dacă acest lucru eșuează în mod repetat, să fie informat utilizatorul de problema apărută și să fie oferite variante de continuare.

3.5.3. Biblioteci de funcții și metode pentru dezvoltarea aplicațiilor de tip *Client/Server*

Tot mai multe aplicații necesită implementarea serviciilor de tip client/server pentru comunicarea între aplicații. Astfel, se lărgeste gama aplicațiilor și schimbul de date dintre acestea va fi mult mai ușor.

Folosirea acestei tehnologii, permite distribuirea în rețea a unor sarcini, ce nu pot fi realizate de către server sau de pe calculatorul ce a lansat aplicația. Arta în programarea aplicațiilor multimedia ce înglobează și asemenea servicii este de a determina gradul de utilizare al diverselor calculatoare prezente în rețea și de a distribui o parte din sarcini altor calculatoare.

Am folosit în practică acest mod de lucru pentru a procesa un număr mare de imagini într-o rețea de calculatoare eterogenă. Timpul de procesare a fost redus considerabil în acest fel, după cum rezultă din două lucrări publicate de autor în anul 2000 [38], [39].

3.5.4. Gestionarea utilizatorilor – Users

Accesarea unei aplicații de către mai mulți utilizatori, simultan sau secvențial, necesită de foarte multe ori scrierea de rutine care să se ocupe de gestionarea acestora. Este necesar să se realizeze:

- Înregistrarea și ștergerea de utilizatori.
- Securizarea aplicației prin implementarea unui sistem de parole de acces și de protejare a datelor prin criptare.
- Implementarea unor algoritmi de criptare.
- Restricționarea accesului la anumite secțiuni din aplicație în funcție de utilizator.
- Personalizarea aplicației pentru fiecare utilizator în parte.
- Salvarea rezultatelor (ce sunt generate în urma rulării aplicației) în directoare securizate sau în cadrul unor baze de date.

3.6. Tehnologii integrate în cadrul modelului

3.6.1. DDE

Tehnologia Dynamic Data Exchange (DDE) permite construirea aplicațiilor de tip client-server. DDE permite actualizarea datelor dintr-o altă aplicație și facilitează executarea de comenzi “la distanță”.

DDE este un protocol de comunicare între aplicațiile Windows în modul client-server. Într-o aplicație client-server, o aplicație (clientul) trimite comenzi altei aplicații (server). DDE furnizează o metodă pentru controlarea unei alte aplicații, asemenea aplicației Excel sau Word, executând comenzi dintr-o altă aplicație. DDE permite schimbul de date între două aplicații.

Nu toate aplicațiile Windows suportă protocoalele DDE.

Există câteva comenzi DDE care pot fi executate de OS.

- Comanda *executeRemote* este folosită la executarea comenzilor înăuntrul aplicațiilor server. De exemplu, folosind DDE, o aplicație poate spune aplicației Excel să deschidă o foaie de calcul și să o tipărească, sau să exporte datele acesteia într-un fișier de format ASCII cu delimitatoare “,”.
- Comanda *getRemote* este folosită pentru obținerea informațiilor din aplicația server, de exemplu, valoarea din celula C5 din foaia de calcul din Excel.
- Comanda *setRemote* poate fi folosită să schimbe valoarea celulei C5 menționate mai sus.

Pentru ca schimbul de date să funcționeze trebuie ca aplicația server să fie mai întâi deschisă. O aplicație care suportă DDE, are un nume DDE prin care aceasta trebuie referită într-o comandă DDE. Numele DDE al aplicației se găsește în general în manualul de utilizare al acestuia. Datele pot fi obținute sau trimise de la aplicația server, sau se poate executa o comandă în aplicația utilizând comenzi DDE. Răspunsurile și erorile serverului sunt verificate examinând proprietatea *sysError*. Legăturile de comunicație între aplicațiile client și server rămân deschise până când programul termină execuția. Legătura poate fi forțată să rămână deschisă și după ce

programul s-a terminat utilizând comanda *keepRemote*. Comanda *closeRemote* este folosită pentru a închide legătura ce a fost menținută deschisă.

În tabelul următor sunt date câteva nume pentru aplicații ce rulează sub Windows.

Aplicație	Nume DDE
Lotus 1-2-3 for Windows	123w
Microsoft Excel	Excel
Microsoft Word	Word
Visual Basic	numele aplicației fără extensia “.exe”
ToolBook	Toolbook

Figura 3-15 Nume DDE pentru diverse aplicații

3.6.2. DLL

Dynamic Link Library (DLL) este un fișier extern care conține rutine (subrutine) sau funcții pe care aplicațiile Windows, le pot accesa. Aceste rutine, scrise de obicei în C sau C++, pot fi accesate de un program OpenScript extinzând astfel limbajul OS. Pachetele obișnuite includ în general câteva fișiere DLL pentru lucrul cu fișiere cum ar fi: copiere, ștergere fișiere, schimbarea driverelor, schimbarea directoarelor, lucrul cu baze de date, accesarea porturilor seriale de comunicare, completarea informațiilor Windows, etc. În plus, fișierele DLL sunt disponibile din alte pachete de programe, inclusiv DLL pentru Windows API (Application Programming Interface). DLL personalizate pot fi scrise de asemenea de un programator C/C++ și de asemenea să fie accesate din diverse aplicații.

Înainte ca o funcție DLL să fie accesată, fișierul DLL trebuie să fie mai întâi legat (similar cu deschiderea unui fișier) și să fie declarată funcția dorită. Funcția poate fi apoi accesată iar valoarea returnată de aceasta să fie utilizată în OpenScript. O dată ce funcția nu mai este apelată, fișierul DLL trebuie să fie închis (dezlegat).

Aplicațiile care folosesc fișiere DLL (adică librării cu legături dinamice exterioare unei cărți) trebuie să fie distribuite împreună cu aceste fișiere.

3.6.3. OLE

Object Linking and Embedding (OLE) este o tehnică ce permite crearea-exportarea și re folosirea obiectelor din aplicații multimedia. Refolosirea obiectelor folosind această tehnică presupune și un mecanism de căutare care să permită accesarea spre aceste obiecte multimedia. De exemplu, un asemenea mecanism care poate fi foarte util în editarea unui material video, din mai multe materiale care există se pot lua segmente care să constituie un nou element, un nou material video. Este necesar să se utilizeze și aspectele temporale ale unui astfel de obiect.

OLE este o extindere a conceptului DDE, permițând includerea de date dintr-o altă aplicație și actualizarea referințelor editate în aplicația respectivă atunci când editat documentul sursă. Când informația este “înglobată”, o copie a datelor este plasată în aplicație și memorată. Când informația este legată, este memorată doar referința la document. Aplicația examinează fișierul sursă pentru a vedea dacă acesta s-a modificat schimbând și fișierul aplicației corespunzător.

3.6.4. Generalizarea modului de conectare la o bază de date (G-ODBC)

Firma Microsoft a propus conceptul ODBC, adică *Open Database Connectivity*, pentru a generaliza modul de dialogare cu o bază de date relațională Microsoft Access prin intermediul unui protocol fixat. Propun o generalizare în aplicarea acestui concept, astfel aplicațiile multimedia să se folosească de acest mod de conectare pentru ași accesa obiectele necesare din baze de date. Este necesar să se permită interfațarea unui limbaj de programare cu bazele de date standard ce există. În momentul actual, pentru limbaje ca și C++ sau Java, dezvoltatorii de baze de date au realizat propriile drivere de interfațare.

Tehnologia ODBC pune la dispoziție o interfață comună pentru accesarea bazelor de date SQL eterogene, bazată pe limbajul SQL drept standard de accesare a datelor. Această interfață, oferă un grad înalt de interoperabilitate, o singură aplicație poate accesa diverse sisteme de gestiune a bazelor de date bazate pe limbajul SQL, prin intermediul unui set comun de coduri. Aceasta permite programatorilor să realizeze și să distribuie o aplicație client-server, fără a avea în vedere un sistem de gestiune a bazelor de date specific. Apoi, sunt adăugate driverele bazei de date, pentru a lega aplicația de sistemul SGBD ales de utilizator.

ODBC se evidențiază acum drept un standard industrial. Unul dintre motivele popularității standardului ODBC îl constituie flexibilitatea sa:

- aplicațiile nu sunt legate de un anumit comerciant care este proprietarul API;
- instrucțiunile SQL pot fi incluse explicit în codul sursă sau construite dinamic în timpul execuției;
- o aplicație poate ignora protocoalele de comunicații a datelor aflate la bază;
- datele pot fi expediate și recepționate într-un format convenabil pentru aplicație;
- standardul ODBC este proiectat în concordanță cu standardele X/Open și interfața ISO la nivel de apelare (CLI);
- astăzi există drivere ale bazelor de date ODBC pentru peste 50 dintre cele mai populare sisteme SGBD.

Structura unei interfețe ODBC se definește prin următoarele elemente:

- o bibliotecă de apelări de funcții care permite conectarea unei funcții la un sistem SGDB, executarea instrucțiunilor SQL și regăsirea rezultatelor.
- o modalitate standard de conectare și începere a unei sesiuni de lucru într-un sistem SGDB;
- o reprezentare standard a tipurilor de date;
- un set standard de coduri de erori;
- o sintaxă a limbajului SQL bazată pe specificațiile sistemului X/Open și interfața la nivel de apelare CLI.
- Arhitectura interfeței ODBC are 4 componente:
 - aplicația, care efectuează prelucrarea și apelurile la funcțiile ODBC, pentru a transmite instrucțiunile SQL sistemului SGDB și a regăsi rezultatele din acesta;
 - administratorul de drivere, care încarcă driverele în contul unei aplicații. Administratorul de drivere, furnizat de către compania Microsoft, este o bibliotecă cu legare dinamică (dll);
 - agentul pentru drivere și baza de date, care procesează apelările de funcții ODBC, transmite cererile ODBC către o anumită sursă de date și returnează rezultatele spre aplicație. Dacă este necesar, driverul modifică cererea unei aplicații, astfel încât aceasta să se conformeze sintaxei acceptate de către sistemul SGDB asociat. Driverele expun capacitățile sistemului SGDB aflat la bază; ele nu sunt necesare pentru implementarea capacităților care nu sunt acceptate de către sistemul SGDB. De exemplu, dacă sistemul SGDB aflat la bază nu acceptă uniuni externe, atunci nici driverul nu o face. Singura excepție importantă de la această

regulă este faptul că driverurile pentru sistemele SGDB care nu au motoare autonome ale bazei de date, cum ar fi Xbase, trebuie să implementeze unul care acceptă cel puțin un minimum de limbaj SQL.

- sursa de date, care este formată din datele pe care dorește să le acceseze utilizatorul și sistemul SGDB asociat, împreună cu sistemul de operare gazdă și platforma de rețea, dacă există.

În aplicația prezentată în capitolul 6 am folosit pentru comunicare JDBC (Java Database Connectivity) ce este o interfață API (Application Program Interface) realizată de Sun, care permite ca un program Java să acceseze un sistem de gestionare al unei baze de date. JDBC facilitează conectarea la baza de date, execută instrucțiunile specifice bazei de date (în general SQL) și preia rezultatele.

3.6.5. Funcții de control ale echipamentelor multimedia MCI

Un sistem de operare pentru aplicații multimedia (MMOS) trebuie să permită utilizarea resurselor multimedia de mai multe aplicații ce rulează simultan astfel încât fluxurile de date să fie sincronizate. Cel mai important pentru un sistem de operare multimedia este de a permite înregistrarea și redarea sunetului și a secvențelor video fără discontinuități. MMOS trebuie să permită unelte soft pentru creatorii de aplicații ce utilizează elemente standard de comandă pentru audio, MIDI, CD-ROM, afișare, animații sau alte periferice. O arhitectură generică pentru un sistem de operare multimedia este dat în figura următoare:

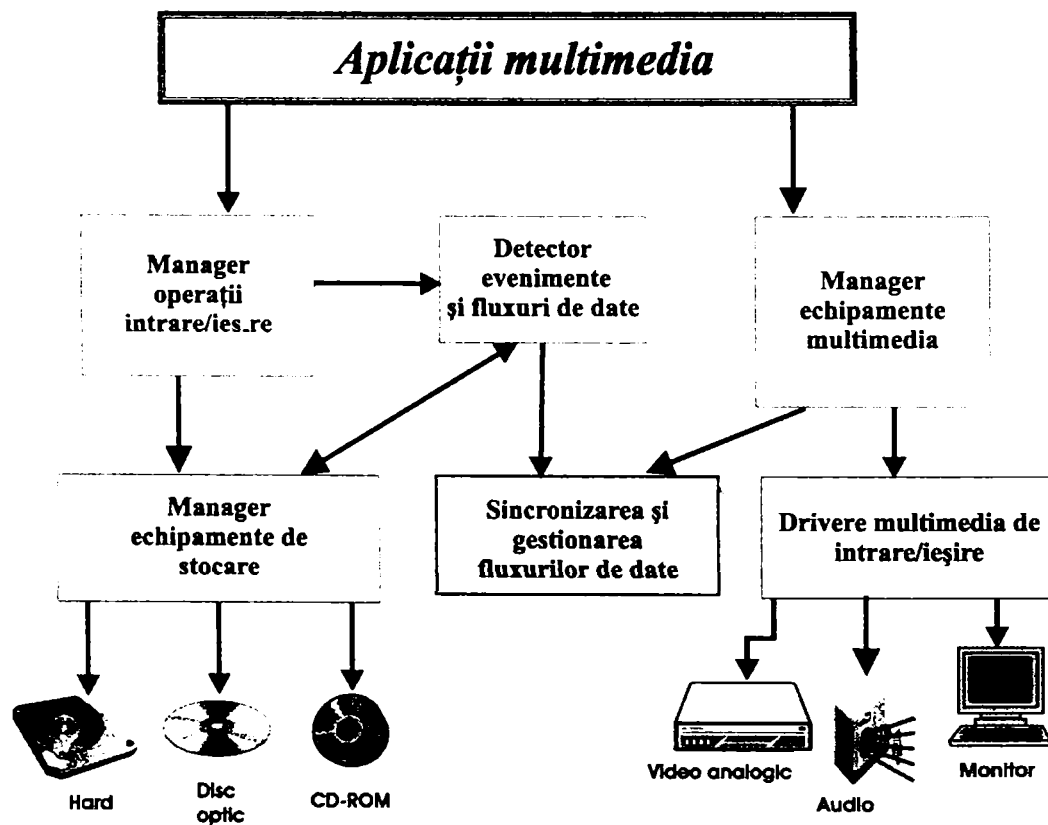


Figura 3-16 Reprezentarea simplificată a sarcinilor specifice unui sistem de operare cu capacități multimedia

Există pe piața dezvoltatorilor de sisteme de operare multimedia 3 mari producători ce includ sistemele IBM Multimedia Presentation Manager/2, Microsoft Windows Multimedia Extensions, Apple's QuickTime.

IBM Multimedia Presentation Manager/2 (MMPM/2) – reprezintă o extensie a sistemului de operare OS/2 pentru aplicațiile multimedia. MMPM/2 furnizează următoarele funcții:

- ✚ Interfață suport pentru controlul aplicațiilor media - aceasta permite pornirea și oprirea unor echipamentelor ce permit redarea sau înregistrarea sunetului sau a secvențelor video.
- ✚ Înregistrarea și redarea sunetului sau a secvențelor video.
- ✚ Manager al operațiilor de intrare ieșire destinate aplicațiilor multimedia ce permite o independență față de formatul obiectul de date.
- ✚ Drivere pentru echipamentele audio-video.
- ✚ Funcții pentru organizarea și sincronizarea fluxurilor de date.
- ✚ Acceptarea diverselor tipuri și formate de date

Arhitectura unui sistem de operare multimedia dezvoltat de IBM este prezentat în figura următoare:

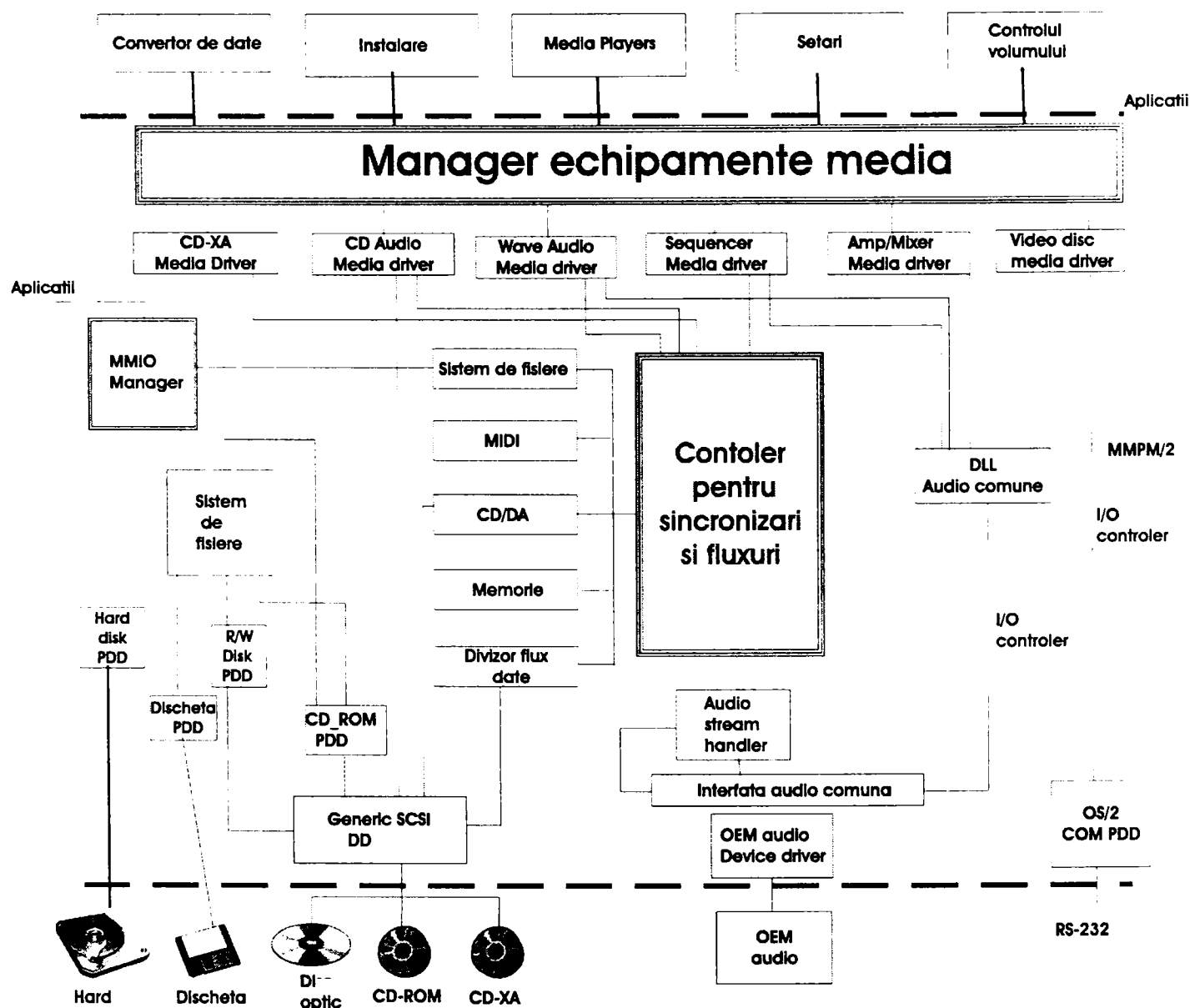


Figura 3-17 Reprezentarea unui manager de resurse hardware folosite pentru redarea obiecte media

Se remarcă în structura de mai sus cele două subsisteme ale MMPM/2:

- MDM (Media Device Manager)
- MMIO (Multimedia I/O Manager)

Aplicațiile comunică cu echipamentele hardware prin intermediu MDM și MMIO. Comenzile sunt trasmise de la aplicații prin MDM spre un driver al unui echipament multimedia și apoi unui driver fizic de echipament (PDD Physical Device Driver) sau prin intermediul managerului MMIO (atunci când se utilizează un eveniment al unui flux de date) și apoi se transmit spre PDD. Datele multimedia sunt tratate ca și fluxuri de date și sunt controlate prin intermediul unui set de evenimente de flux și prin intermediul managerul de sincronizare a fluxurilor (SSM - Sync-Stream Manager).

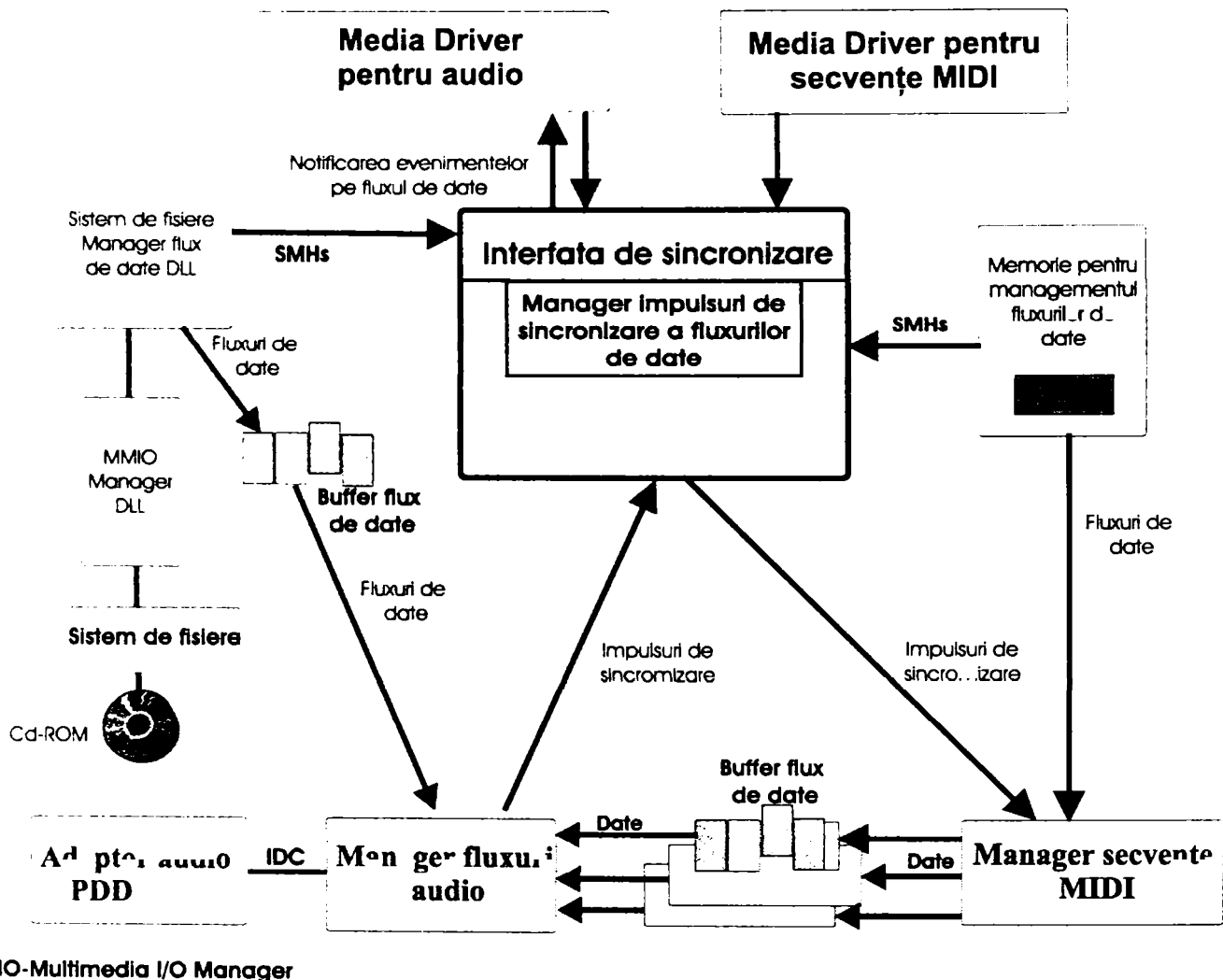


Figura 3-18 Modalitate de sincronizare a fluxurilor de date

MMPM/2 suportă semnale video ce provin din diverse surse: semnal video digital format (DV), analogic (videorecorder), videodisc și alte diverse animații. MMPM/2 implementează conceptul conectării prin soft a datelor sursă cu datele rezultante. Aici sunt 17 tipuri de conectări posibile, inclusiv fluxuri MIDI, fluxuri de la CD-ROM, semnal pentru căști, întrea microfonului și altele ce folosesc accesul la dispozitivele PC-ului prin diverse funcții. Aceste conexiuni sunt stabilite în mod automat atunci când un driver al unui dispozitiv multimedia este deschis.

3.7. Facilități suplimentare integrate modelului

3.7.1. Funcții ce asigură calitatea unei aplicații: *QoS*

Problemele cele mai mari în cazul aplicațiilor multimedia sunt date de constrângerile de sincronizare a diferitelor tipuri de medii ce rulează simultan într-o aplicație multimedia.

Impunerea constrângerilor de sincronizare, este necesară pentru a specifica care dintre operațiile, ce se execută la rularea unei aplicații multimedia, trebuie executată cu prioritate și ce resurse hardware necesită. Factorul *QoS* descrie cerințele unei aplicații multimedia din punct de vedere al funcționalității componentelor folosite prin prisma cerințelor de sincronizare a diverselor medii ce rulează [40].

Consider că modelul unei aplicații multimedia trebuie să conțină elemente de evaluare a performanțelor aplicației multimedia în primul rând și în plus trebuie să realizeze o verificare inițială a echipamentului pe care va rula. Se impune o informare adecvată a utilizatorului despre modul în care aceasta ar rula în condițiile date, eventual se pot propune soluții de ameliorare a performanțelor prin intervenția utilizatorului asupra unor elemente ce compun configurația sistemului pe care rulează aplicația.

Propun ca următorii parametrii ce descriu calitatea unui serviciu (a modului în care rulează o aplicație multimedia într-un anumit mediu hardware) să fie respectați și verificați de către realizatorul de aplicații multimedia:

- Media întârzierilor – descrie intervalul de timp după care sistemul reacționează la apariția unui eveniment prin executarea unei operații. De exemplu, întârzierea medie dintre momentul transmiterii unei comenzi de căutare într-o bază de date și momentul obținerii rezultatelor – este un element critic pentru utilizatori.
- Rata vitezei de derulare a aplicației – reprezintă raportul dintre viteza normală (specificată de realizator aplicației în anumite condiții) și viteza cu care aceasta rulează pe un anumit sistem. Viteza de derulare a unei aplicații multimedia depinde de resursele hardware ale sistemului pe care rulează, dar și de modul în care realizatorul aplicației a implementat optimizările de viteză, adică, este permisă rularea cu aceeași viteză pentru un interval mai mare de variație a resurselor hardware. Există mai multe tehnici prin care această optimizare poate fi realizată. Ele se împart în două mari categorii:
 - tehnici subiective, ce intuiesc modul de acțiune al utilizatorului și pregătesc aplicația spre acțiunea cu cea mai mare probabilitate de execuție din partea utilizatorului (ținând cont de acțiunile anterioare ale acestuia)
 - tehnici de implementare software și de gestionare a resurselor – sunt reprezentate de rutine implementate pentru optimizarea modului de rulare al aplicației și a modului de utilizare a resurselor hardware, aspect abordat de către autorul tezei în [41].
- Gradul de utilizare hardware – reprezintă cât la sută din resursele unui sistem sunt folosite de aplicația respectivă. De exemplu, o secvență video cu un debit de 4MB/sec poate reprezenta 50% din debitul care poate fi asigurat de un hardisk.
- Jitterul – reprezintă deviațiile temporale ce apar la rularea succesivă a aceleași prezentări pe același sistem.
- Eroarea temporală cumulată – reprezintă eroarea temporală cumulată pe o perioadă de timp specificată, de către toate mediile ce rulează în acea aplicație. De exemplu, suma tuturor decalajele ce apar într-o prezentare, pe perioada a 60 de secunde, între imagini și momentul când apare sunetul. Această eroare este foarte importantă, mai ales la prezentările de lungă durată, unde apar probleme de sincronizare a mediilor, cu efect cumulativ al decalajelor.

Un studiu efectuat de firma IBM, pe un eșantion considerabil de persoane, a dus la definirea următoarelor valori de referință pentru eroarea temporală cumulată pe perioada unei secunde, pentru ca o aplicație să se înscrie în parametrii unei calități acceptabile în sens global (din punct de vedere al sincronizării diverselor medii) [42]. O parte din concluzii sunt prezentate în următorul tabel:

Media		Aplicații, mod	QoS
Video	Animații	Corelate	120 ms
	Audio	Lipsa sincronizare – de exemplu, în cazul unui film	80 ms
	Imagine	Suprapusă – de exemplu, insert de imagine pe un material video (unde rămâne spațiu vizibil și pentru partea video)	240 ms
		Nesuprapusă – de exemplu, inserarea de imagine statică în loc de material video	500 ms
	Text	Suprapus – de exemplu, la subtitrarea unui material video	240 ms
		Nesuprapus – de exemplu, la inserarea unui generic de tip text simplu în cadrul unui material video	500 ms
Audio	Animații	Evenimente corelate – de exemplu, un dans animat sincronizat pe un sunet	80 ms
	Audio	Puternic cuplate – defazajul accidental dintre canalele audio pentru o transmisie stereo	11μs ms
		Mediu cuplate – de exemplu, discuții între mai multe persoane	120 ms
		Slab cuplate – de exemplu, coloana sonoră din fundalul unei aplicații multimedia	500 ms
	Imagini	Puternic cuplate – de exemplu, muzică sincronizată cu imagini, prezentări de tip diaporame	5 ms
		Slab cuplate – prezentări cu slide-uri	500 ms
	Text	Adnotări	240 ms
	Pointeri	Sunet folosit pentru a marca un eveniment	±400 ms

Tabel 3-1 Parametrii de sincronizare a mediilor dintr-o aplicație multimedia pentru satisfacerea criteriilor de calitate – studiu realizat de IBM

- Rata erorilor de mediu – este reprezentată în funcție de mediu, de către numărul de erori ce apar la rularea întregii aplicații sau numai pe o anumită perioadă de timp. Pot exista diverse moduri de măsurare, iar acestea depind în principal de elementul transmis. De exemplu poate fi evaluată calitatea transmisiei între un client și serverul de pe care rulează aplicația prin rata erorilor de bit (BER) sau prin rata pachetelor de date pierdute (la o transmisie video pe pachete), rata cadrelor pierdute la captura unui semnal video, etc.

Un exemplu de evaluare a performanțelor unui sistem pe care rulează o aplicație multimedia realizată în Asymetrix Toolbook folosind limbajul Lingo este:

```
to handle enterBook
  linkdll "mmsystem.dll"
  DWORD getTime()
end linkDLL
forward
  send leavePage
end enterPage
```

- Script ce se atribuie primei pagini de intrare în aplicație și care este bine să conțină majoritatea tipurilor de medii, la o solicitare maximă pentru sistem.

```
to handle LeavePage
  system startTime
  startTime = timeGetTime() – se inițializează pornirea cronometrării procesului
  forward
end
```

```
to handle enterPage
  system startTime, endTime
  endTime = timeGetTime()
  request "Durata a fost de:" &&(endTime—startTime) – se afișează durata
  clear startTime
  clear endTime
  forward
end
```

- Durata obținută poate fi comparată cu o valoare dată de realizatorul aplicației pentru o rulare optimă, în funcție de rezultat se poate recomanda utilizatorului ce trebuie să facă.
- Testul poate fi dezvoltat prin verificări pe componente ale calculatorului.

În mod asemănător prin scrierea de simple rutine, se pot testa diverse periferice și se pot evalua performanțele sistemului pe care va rula aplicația.

Un domeniu de actualitate îl reprezintă învățământul la distanță, acesta combină de cele mai multe ori videoconferința cu alte elemente media (sunet, imagini, text, secvențe video, animații). Succesul unei astfel de aplicații didactice depinde de cele mai multe ori de calitatea transmisiei, de modul de sincronizare a diverselor medii de transmitere a informației, de întârzierile ce apar în sistem [43].

3.7.2. Funcții de căutare de obiecte: *Browse*

Aplicațiile multimedia devin din ce în ce mai complexe, conținând mii sau chiar sute de mii de obiecte (marile servere de WWW). Este foarte dificilă parcurgerea întregii aplicații pentru găsirea unei informații, de aceea, se preferă soluții automate de căutare. Pentru asemenea aplicații este necesară introducerea unei biblioteci cu funcții de căutare a unor obiecte ce formează aplicația. Aceste rutine sunt foarte complexe, uneori fiind necesare metode complexe de căutare bazate pe conținut. O structurare corectă a obiectelor și implemetarea funcțiilor de căutare, oferă numeroase avantaje utilizatorilor, crescând atractivitatea aplicației.

Biblioteca *Browse* trebuie să fie deschisă spre dezvoltări ulterioare, spre introducerea de noi metode de căutare, aplicabile la noi tipuri de obiecte ce pot apare.

3.8. Concluzii și contribuții

3.8.1. Principalele avantaje ale modelului propus

Caracteristica generală a modelului propus de autor, constă în posibilitatea de reutilizare a unor elemente dezvoltate anterior și posibilitatea de introducere și dezvoltare din partea utilizatorilor, precum și un grad mare de portabilitate.

În concordanță cu cererile actuale modelul propus prezintă:

- Optimizări din punct de vedere al structurării, al implementării orientate pe obiecte a aplicațiilor, aplicații ce lucrează cu obiecte multimedia de tip fluxuri de date ce conțin

con condiționări temporale. Aici sunt definite obiecte, clase, metode și parametrii ce pot crește performanțele aplicației.

- O concepție modulară, ce permite extinderi ulterioare ale aplicațiilor, re folosirea unor părți din aplicații.
- Folosirea modelului propus permite ca utilizatorii să poată contribui la îmbunătățirea bibliotecilor de funcții și transferabilitatea acestora între aplicații.
- Modelul propune folosirea unor biblioteci de funcții care să verifice înainte de rularea unei aplicații multimedia mediul hardware, scopul acestora fiind asigurarea unui anumit grad de calitate pentru serviciile oferite de aplicație și informarea utilizatorului din timp asupra problemelor ce pot apare, pot fi propuse chiar și soluții de remediere în funcție de configurația detectată.
- Modelul propune includerea bibliotecilor de funcții și metode ce permit comunicarea cu bazele de date, ceea ce face posibilă dezvoltarea aplicațiilor multimedia cu accesarea obiectelor direct din baze de date. Dezvoltarea bazei de date cu noi obiecte duce mai departe noi facilități pentru aplicația multimedia, fără a fi necesare refacerea aplicației, rescrieri de cod sau alte complicații de acest fel. Este un model ce permite realizarea de aplicații multimedia, unde numărul de pagini ale aplicației este generat dinamic și nu este implementat în mod fix de către utilizator.
- Tehnologia client/server este inclusă în cadrul modelului pentru realizarea de aplicații care să lucreze în colaborare cu mai mulți utilizatori simultan.

Detalierea acestor avantaje o voi face pentru fiecare subcomponentă în parte în capitolele următoare.

3.8.2. Un nou mod de generalizare a aplicațiilor multimedia prin accesarea obiectelor utilizate din baze de date multimedia

Pentru moment, din ceea ce am studiat și din experiența practică ca și utilizator de medii de dezvoltare a aplicațiilor multimedia, am ajuns la concluzia, că realizatorii de medii de dezvoltare a aplicațiilor multimedia trebuie să își îndrepte atenția înspre a construi un sistem care să permită înglobarea tuturor tipurilor de media și în special a cât mai multor formate de reprezentare pentru același tip de media.

Trebuie să se asigure o deschidere a mediilor de dezvoltare a aplicațiilor multimedia, utilizatorii cu o anumită experiență să poată introduce noi tipuri de date multimedia, specifice unor cazuri ce se întâlnesc în practică, în formate ce optimizează aplicația pentru cazul respectiv. Ca și exemplu pot da situația din cadrul unui post de televiziune unde în producția unui material mai deosebit este necesară definirea unui efect video propriu.

Prin atașarea de informații de dezvoltare a uneltelor de lucru, ce se folosesc în cadrul unui mediu de dezvoltare a aplicațiilor multimedia, se poate dezvolta mediul respectiv. Astfel în modelul propus, pe lângă facilități de comunicare cu baze de date, propun existența în cadrul bazelor de date a unor biblioteci de funcții, de unelte de lucru, definite de către utilizator, care să poată fi accesate în continuare și de alți utilizatori.

Astfel vor exista următoarele relații:

- La nivel de mediu de dezvoltare a aplicațiilor multimedia vor exista biblioteci private și publice. Cele private sunt necesare funcționării mediului și asigură posibilitatea creării aplicației multimedia. Bibliotecile publice sunt obiecte, metode și funcții ce se integrează în interiorul aplicației multimedia.
- La nivelul aplicației multimedia vom avea biblioteci importate de la mediul de dezvoltare și biblioteci de funcții care se creează în mod special pentru acea aplicație. Din aceste biblioteci

existente se vor crea bibliotecile publice și private pentru utilizatorii aplicației multimedia. Este nevoie de această nouă regrupare a accesului la biblioteci pentru a fi permise dezvoltări ulterioare a aplicațiilor. De exemplu, se realizează o aplicație comercială de prezentare a unui studio de televiziune. Dacă se concepe prezentarea în mod *static*, cu număr fix de obiecte și acțiuni avem o aplicație închisă ce nu permite dezvoltări ulterioare. Dacă conceptul prezentării este *dinamic*, adică obiecte și acțiuni să fie generate dinamic în funcție de obiectele ce se găsesc într-o bază de date, aplicația noastră va fi o aplicație deschisă pentru dezvoltări ulterioare.

- La nivel utilizator vom avea din nou două categorii de biblioteci, publice și private. Bibliotecile private sunt elemente ce determină funcționarea aplicației. Dezvoltarea ulterioară a aplicației se face prin intermediul bibliotecilor publice, care permit accesul utilizatorului să le personalizeze în funcție de nevoile dinamice ale acestuia.

Includerea în baze de date a elementelor ce formează bibliotecile unei aplicații multimedia permite o generalizare a aplicațiilor create și dezvoltarea acestora în funcție de necesitățile utilizatorilor fără eforturi deosebite.

Un avantaj major ar putea fi reducerea dimensiunii fișierului executabil prin încărcarea de obiecte numai atunci când este nevoie din baza de date și nu includerea inițială a tuturor obiectelor în acesta.

Numărul de pagini al aplicației nu este fix el poate varia în funcție de acțiunile utilizatorului, altfel spus, ele se generează dinamic în urma interacțiunii utilizator – aplicație multimedia.

Din punct de vedere al efortului de implementare al unei asemenea aplicații, am ajuns la următoarele concluzii:

- Timpul și volumul de muncă necesar conceperii și modelării aplicației crește.
- Timpul și volumul de muncă necesar implementării fizice a aplicației scade foarte mult.
- Timpul și volumul de muncă global de realizare a aplicației se reduce.

Este nevoie de mai mult timp de proiectare și concepere a modelului, dar apoi apare o reducere la nivel de implementare fizică, nu va fi nevoie, de exemplu, pentru prezentarea unei companii, să se realizeze pagini pentru fiecare produs în parte (numărul acestora poate fi uneori foarte mare), ci se poate realiza un model de pagină, obiectele din această pagină fiind importate dinamic dintr-o bază de date. Deci, va fi necesar de făcut un model și structurarea datelor într-o bază de date și nu de a face sute de pagini.

Structura generală a modelului de aplicație multimedia, propusă de autor, ce permite obținerea de performanțe mai bune și care explică modul de accesare a diverselor tipuri de biblioteci este prezentată în figura următoare:

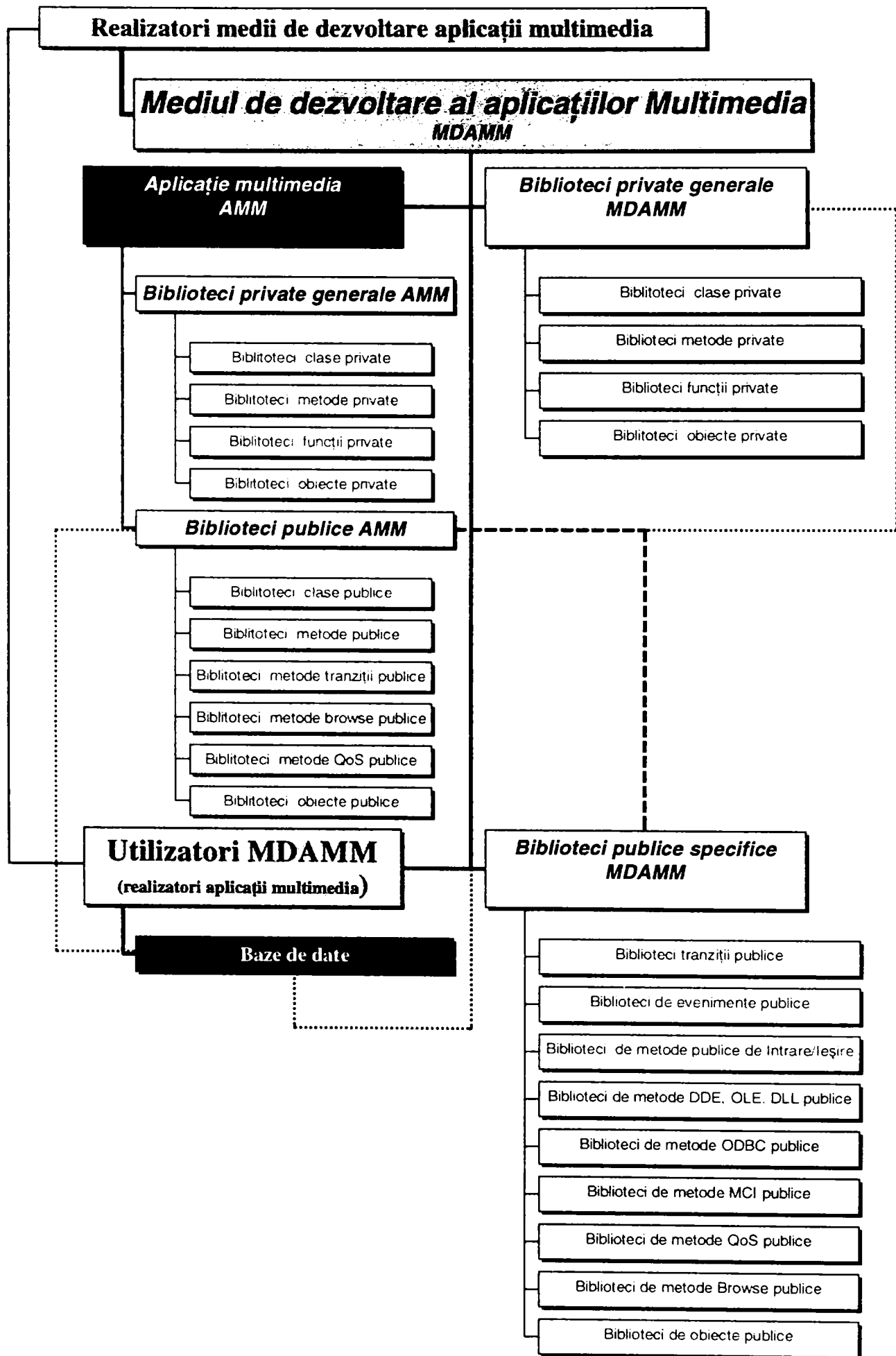


Figura 3-19 Partajarea bibliotecilor în relația producător medii de dezvoltare – dezvoltator aplicație – utilizator aplicație multimedia

Capitolul 4.

Baze de date multimedia

4.1. Evoluția bazelor de date

În perioada anilor 1950-1960, mari companii ca IBM, General Electric și Honeywell au început dezvoltarea conceptului de baze de date. Acestea trebuiau să permită accesul mai multor utilizatori simultan la informații.

Realizatorii limbajului COBOL au început prin implementarea, în cadrul unui departament DATA DIVISION, rutine care să lucreze cu baze de date sau să permită definirea de noi tipuri de date ce pot să acceseze datele și să le actualizeze.

Fundamentarea modelor ierarhice și rețea pentru bazele de date. În 1969 Data Base Task Group definește Data Description Language (DDL) și Data Manipulation Language (DML) ce permit lucrul cu baze de date și fundamentează începuturile managementului bazelor de date în rețele. Utilitatea dezvoltării de baze de date a fost confirmată cu ocazia programului spațial de explorare de pe Lună: Apollo. Aici s-a folosit o primă baza de date ce folosea conceptul de management de sistem ierarhic ce a fost implementat în cadrul Information Management System (IMS) dezvoltat de IBM. Ideea partajării în rețea și în timp a resurselor a fost introdusă de către compania System Development Corporation în cadrul produsului Time-Shared Database Management System.

Modelul relațional a fost introdus în ani 1970 de către E. Codd și rezolvă o parte din problemele primelor baze de date oferindu-le o mult mai mare flexibilitate [44]. Limbajul de interogare relațional ca Structure Query Language (SQL) este mult mai declarativ decât primele limbaje navigaționale. Cu SQL utilizatorii trebuie să specifice numai ce doresc să obțină de la baza de date și nu cum se obține acea informație. Modelul relațional este în general simplu și elegant în utilizare bazându-se pe o teorie matematică bine fundamentată și ușor de înțeles ce folosește algebra relațională și calculul predicatelor de prim ordin [45]. Algebra relațională constă într-un număr redus de seturi de operații: reuniuni, intersecții, diferențe, produs Cartesian, selecții și proiecții. Cel mai mare succes comercial cu acest tip de baze de date l-a avut firma IBM cu produsul DB2 ce a avut ca rezultat și impunerea și standardizarea dialectului SQL. Alți producători de baze de date relaționale sunt: Informix Corporation, Oracle Corporation, Sybase Inc., Microsoft Corporation, Borland International.

Modele semantice și funcționale - sistemele de baze de date ce au apărut ulterior bazelor de date relaționale s-au bazat o modelare semantică a datelor, motivarea acestui tip de modelare a constat în modelarea într-un mod cât mai apropiat de lumea reală apropiindu-se de modelarea orientată pe obiecte. Cel mai popular mod de exprimare semantică a datelor a fost introdus de Chen în 1976 prin faimosul mod ER (Entity-Relationship - entitate-relație) unde o entitate este un obiect sau un lucru care există și care este distinct față de alte obiecte [46]. Entitatea este descrisă prin attribute sau proprietăți. În termenii construcției unei baze de date,

entitățile sunt înregistrări clasice în baza de date plus atribute ale acestora ce sunt reprezentate ca și câmpuri ale acestor înregistrări. O schemă (structură) a unei baze de date se proiectează pornind de la un model ER.

O schemă semantică este mapată în general peste o schemă relațională folosind un limbaj de definire a datelor (DDL) sau un limbaj relațional (SQL). Ulterior conceptul de descriere semantică a datelor a fost dezvoltat în cadrul bazelor de date orientate pe obiecte care introduce conceptul de abstractizare a obiectelor, proprietăți de moștenire și interfețe ale unui tip ce sunt independente de implementare.

Au fost propuse mai multe modele, cel mai cunoscut fiind modelul DAPLEX dezvoltat de Shipman în 1981 [47], acesta tratează atributele ca și funcții, iar valorile operației de căutare sunt obținute prin aplicarea acestor funcții entităților. Sub conceptul acestui model au fost dezvoltate două sisteme comerciale : SIM dezvoltat de Unix și OpenODB dezvoltat de Hewlett-Packard.

Modele relaționale extinse sau modele obiect-relaționale - ca prim pas de trecere la bazele de date orientate pe obiecte au fost bazele obiect-relaționale care încorporează câteva din caracteristicile acestora. Proprietățile incluse sunt: introducerea tipului de date abstracte, moștenirea, identitatea obiectelor, posibilitatea de a include structuri sau proceduri în interfața de programare a bazei de date. Rezultatul a fost extinderea facilităților multimedia oferite de o bază de date. Sunt posibile în acest mod de organizare a informației să înregistreze tipuri de date cum ar fi sunetele, imaginile prin asigurarea suportului pentru memorarea unor date de mari dimensiuni (BLOB) și reduse posibilități de căutare a datelor după conținut.

Propunerile concrete de astfel de sisteme au fost făcute de către Stonebraker, Rowe și Hiohama în cadrul sistemului POSTGRES în 1990 [48]. O continuare a extinderii modelului obiect-relațional a fost realizată de către Khoshafian în 1991 prin sistemul Intelligent SQL [49].

Implementările comerciale sunt Illustra, UniSQL și Matisse. Proprietățile incluse sunt: introducerea tipului de date abstracte, moștenirea, identitatea obiectelor, posibilitatea de a include structuri sau proceduri în interfața de programare a bazei de date. Aceste proprietăți au fost incluse ca și dezvoltare a limbajului SQL în cadrul limbajului variantei SQL3.

Modele relaționale plate (Nested relational) - în spațiul obiectelor, modelele relaționale sunt constituite dintr-o colecție de tabele plate bidimensionale. Fiecare tabel poate fi considerat ca o mulțime de rânduri, coloanele fiecărui rând pot fi considerate ca fiind instanțe ale tipurilor atomice de bază – întregi, numere în virgula flotantă, șiruri de caractere, etc. Un tabel plat este cunoscut ca fiind primul form normal. Modelele de obiecte complexe încearcă să relaxeze, să slăbească restricțiile impuse de primul form normal, menținându-se în limitele impuse de solidă fundamentație matematică ce există în modelul relațional.

Cu un model relațional plat utilizatorii vor avea o „relație” (mulțime de tuples) descrisă de atribute și reprezentări ale acestora înregistrate în baza de date. Modelele relaționale nested sunt reprezentate de două importante propuneri. În 1983 a fost propus de către Bancilhon un prim model de baze de date relaționale nested denumit VERSO [50].

Cel de al doilea model propus a fost cel oferit de către Schek and Scholl în 1986 [51].

Mai general bazele de date ce folosesc modele de obiecte complexe se bazează pe construcția obiectelor într-un spațiu ce are la bază o colecție de tipuri atomice de bază și folosind doi constructori de obiecte: mulțimile și „tuples”, model prezentat în 1989 de către Khoshafian [52][53]. Modelele post-relaționale bazate pe obiecte complexe menționate mai sus au permis o structurare a spațiului obiectelor numai sub formă de arbore. Nu a fost permis ca același obiect să aibă de exemplu părinți multipli, deci nu suportă un spațiu al obiectelor structurat sub formă de graf. Pentru a avea această abilitate, modelele bazate pe obiecte trebuie să suporte conceptul de identitate a obiectului propus de Khoshafian [54].

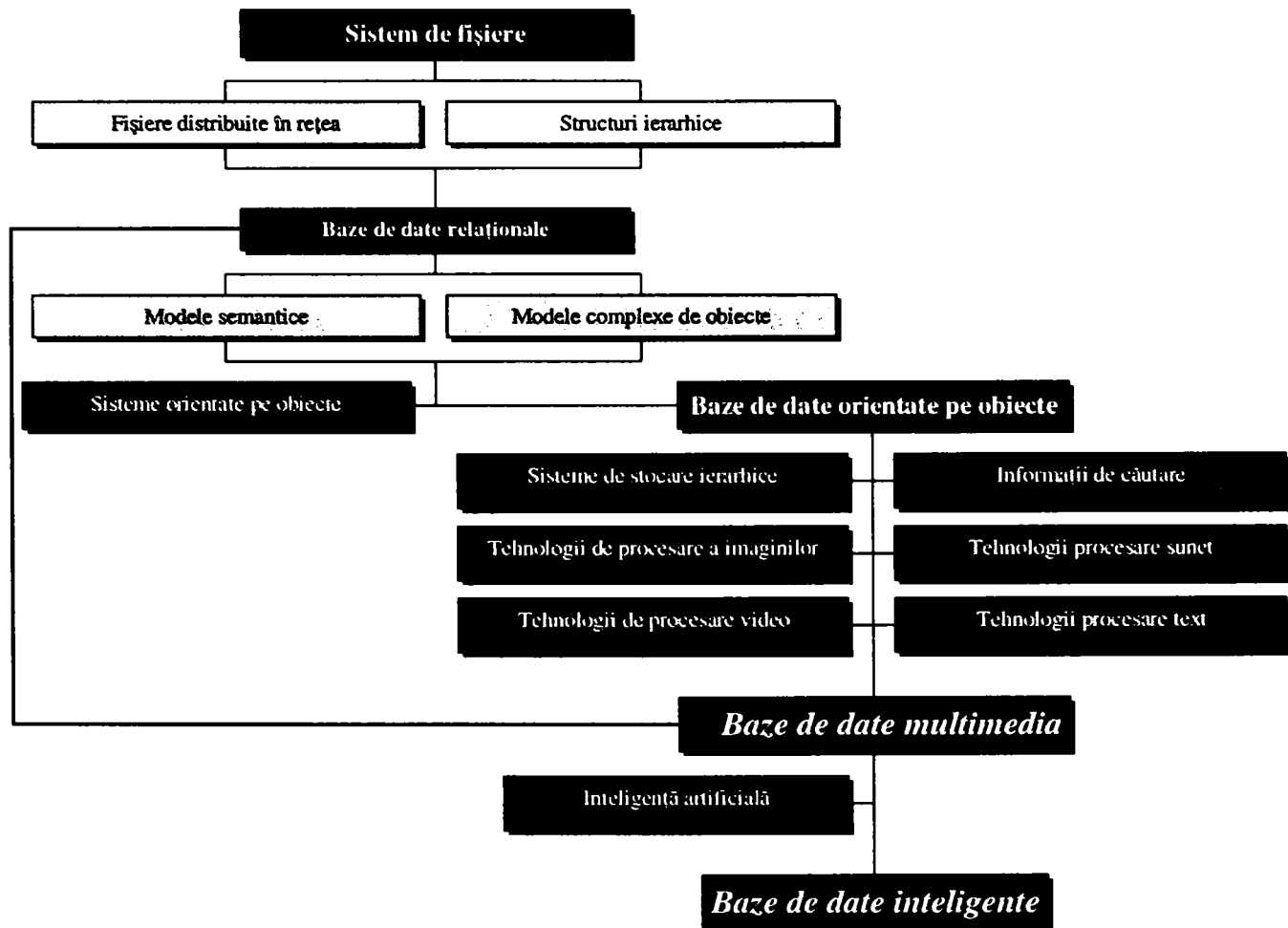


Figura 4-1 Evoluția spre bazele de date multimedia

Baza de date comercială implementată pe baza acestor concepte este: Franco-Armenian Data Language (FAD) implementat de consorțiul de certare Microelectronic and Computer Corporation (MCC) din Austin, Texas [55].

Pasul spre noua generație de baze de date ce au apărut a fost dat de către Show și Zdonik prin introducerea fundamentelor algebrice necesare dezvoltării bazelor de date orientate pe obiecte [56]. Continuarea și implementarea acestor fundamente matematice în cadrul bazelor de date, în special a modului de interogare a unei baze de date a fost realizată de către Straube și Ozsú prin orientarea accentuată a cercetărilor înspre bazele de date orientate pe obiecte [57].

Baze de date orientate pe obiecte au apărut ca variante comerciale după 1986, cele mai cunoscute sunt GemStone de la ServioLogic, Gbase de la Graphael și Vbase de la Ontos. Următorii ani au fost caracterizați de o dezvoltare impresionantă a aplicațiilor bazate pe orientarea pe obiecte: SIM de la Unisys, Versant de la Versant Object Technology, ObjectStore de la Object Design Inc. și altele.

Bazele de date orientate pe obiecte își dovedesc performanțele în aplicațiile de genul proiectare asistată de calculator (CAD), fabricare asistată de calculator (CAM), ingineria programării asistată de calculator (CASE), sisteme informaționale de birou (OIS), sisteme multimedia, editare digitală și sistemele informaționale geografice (GIS), ca și aplicațiile cu obiecte complexe intercorelate și cu date procedurale.

Următoarele proprietăți ale conceptului de programare orientată pe obiecte sunt exploatate cu succes în cadrul bazelor de date:

- conceptul de încapsulare: obiectul conține atât structura de date cât și setul de operații care pot fi utilizate pentru manipularea sa.

- conceptul de ascundere a informațiilor: anumite aspecte externe ale unui obiect sunt separate de detaliile lui interne.
- un obiect este o entitate unic identificabilă, care conține atributele care descriu starea unui obiect din lumea reală, dar și acțiunile asociate acesteia. Obiectele pot conține alte obiecte.
- comportamentul unui obiect este definit prin metode. Acestea sunt utilizate pentru verificarea valorii atributelor sau pentru modificarea stării obiectului prin modificarea valorii atributelor.
- mesajele sunt mijloacele prin care comunică obiectele, un mesaj este o cerere a unui obiect (emitent) către alt obiect (receptor), prin care cel de-al doilea obiect este solicitat să execute una din metodele sale.
- obiectele care au aceleași atribute și răspund la aceleași mesaje pot fi grupate împreună într-o clasă
- moștenirea permite ca o clasă să fie definită ca un caz special al unei clase mult mai generale. Se pot contrui clase generalizatoare, adică super clase, sau clase specializate, adică subclase.
- supraîncărcarea permite ca numele unei metode să fie reutilizat într-o definiție a unei clase sau pe parcursul definițiilor
- supracopierea este un caz special de supraîncărcare, care permite ca numele unei proprietăți să fie redefinit într-o subclasă.

Atkinson propune în 1989 [58], “manifestul sistemelor bazelor de date orientate spre obiecte”. Acest manifest este format din 13 caracteristici obligatorii pentru un SGBD OO:

1. Trebuie oferit suport pentru obiectele complexe.
2. Trebuie oferit suport pentru identitatea obiectelor.
3. Trebuie oferit suport pentru încapsulare.
4. Trebuie oferit suport pentru tipuri sau clase.
5. Trebuie oferit suport pentru ca tipurile sau clasele să poată moșteni atribute sau metode de la supertipurile sau superclasele lor.
6. Trebuie oferit suport pentru legătura dinamică.
7. Limbajul de manipulare al datelor DML trebuie să ofere posibilități complexe de calcul.
8. Mulțimea tipurilor de date trebuie să fie extensibilă.
9. Trebuie asigurată persistența datelor.
10. Sistemele de gestiune ale bazelor de date trebuie să administreze baze de date de dimensiuni foarte mari.
11. Sistemele de gestiune ale bazelor de date trebuie să ofere suport pentru utilizatorii concurenți.
12. Sistemele de gestiune ale bazelor de date trebuie să fie capabile de refaceerea datelor în cazul unor întreruperi de funcționare din motive hardware sau software.
13. Sistemele de gestiune ale bazelor de date trebuie să asigure interogarea datelor de o manieră cât mai simplă posibilă.

În tabelul următor sunt date comparativ avantajele și dezavantajele sistemelor de gestiune a bazelor datelor orientate pe obiecte:

Capacitate de modelare sporită	Lipsa unui model de date universal
Extensibilitate	Lipsa de standarde
Limbaj de interogare mai expresiv	Optimizarea interogării compromise încapsularea
Suport pentru evoluția schemei	Zăvorârea la nivelul obiectului poate influența performanțele
Suport pentru tranzații de lungă durată	Complexitatea
Aplicabilitate în domenii cu un grad mare de complexitate a aplicațiilor	Lipsa de suport performant pentru securitate
Performanțe îmbunătățite	

Tabel 4-1 Avantajele și dezavantajele sistemelor de gestiune a bazelor de date orientate pe obiecte

4.2. Situația actuală a bazelor de date

4.2.1. Baze de date pentru publicații multimedia

GMD-IPSI - este una dintre primele implementări prototip pentru o revistă multimedia denumită *MultiMedia Forum* [59]. Particularitatea unui asemenea mediu constă în faptul că oferă un mediu de editare, unde utilizator introduce informațiile și își realizează propriile prezentări multimedia, și un mediu pentru utilizatori care permite accesarea acestor informații de către diverși utilizatori prin Internet.

Continuarea acestei idei poate fi întâlnită acum foarte des mai ales în cazul mediilor de dezvoltare a cursurilor pentru învățământul la distanță World Wide Web Course Tools (WebCT) [60], WebBoard [61], sau Lotus Notes (care poate lucra foarte bine cu baze de date ca DB2, Oracle) [62].

În cazul cercetării efectuate, am lucrat cu primele două medii – WebCT [63],[64] și WebBoard [65] – dezvoltând cursuri pentru învățământul la distanță, facilitățile oferite sunt deosebite pentru realizarea paginilor de web, gestionarea studenților, realizarea de diverse tipuri de exemple și exerciții. Nu oferă însă un suport adecvat pentru secvențe video și nu au prevăzute facilități de gestionare a acestora. Se pot include secvențe video în paginile cursurilor, dar nu există un management adecvat care să gestioneze accesul sau să garanteze calitatea fluxului de date. Rezultatele experimentale obținute au fost publicate de autor pe web, în conformitate cu bibliografia prezentată [62],[63],[64].

MμSE – Multimedia and data system support for System Engineering – și reprezintă un mediu de gestionare a aplicațiilor din domeniul ingineriei de sisteme. Folosește o bază de date orientată pe obiecte care poate include text, grafice, sunet, imagini și semnal video [66]. Stocarea și manipularea hiperstructurilor și a hiperdocumentelor conținute de hipernoduri sunt modelate la nivelul bazei de date. Modelul orientat pe obiecte al documentelor conține metode ce facilitează accesul spre uneltele necesare editării acestora și a afișării de rezultate. Sistemului tradițional de gestionare a datelor i s-a adăugat un sistem de management al bazei de date de tip VODAK Data Manipulation Language, care să permită utilizare facilităților multimedia [67], [68].

QBIC - unul dintre primele sisteme de baze de date ce folosesc capabilități de căutare în baze de date bazată pe conținut este sistemul QBIC – Query By Image Content [69]. Această tehnologie poate fi folosită cu succes în foarte multe domenii de activitate, cum ar fi: domeniul

medical (căutarea în diverse imagini a tumorilor canceroase pe baza texturilor), domeniul industrial, domeniul turistic, jurnalism, artă, etc.

Realizarea unui astfel de sistem diferă foarte mult de varianta clasică ce folosește cuvinte cheie pentru căutare. Realizarea unui sistem ce folosește tehnologia asemănătoare cu cea a sistemului QBIC este compusă din trei etape:

- Aducerea imaginilor în baza de date, pregătirea unor imagini de mici dimensiuni care să fie folosite pentru facilitarea căutării în procesul iterativ de căutare.
- Cea de a doua etapă este calculul caracteristicilor imaginilor (caracteristici de textură, de culoare, formă etc.).
- Procesul de interogare reprezintă pasul de calcul al distanței dintre caracteristici.

MINOS este o aplicație multimedia ce permite stocarea documentelor complexe [70]. Acest sistem introduce o modelare orientată pe obiecte a documentelor multimedia ce folosește și o căutare axată pe conținut. Un document multimedia în MINOS este un obiect multimedia compus din atribute, o colecție de segmente de text, o colecție de segmente vocale, o colecție de imagini și informații despre alte obiecte multimedia. Fiecare obiect multimedia are de asemenea o informație despre obiectele multimedia care mai sunt în legătură cu el. Un text, de exemplu, poate fi segmentat logic în în titlu, rezumat, capitole și referințe. Fiecare capitol se subdivide în secțiuni, paragrafe, fraze și cuvinte. Extragerea componentelor unui document complex este permisă prin intermediul mecanismului de interogare.

4.2.2. Access

Microsoft Access este una din cele mai populare baze de date relaționale, fiind răspândită în general la utilizatori ce folosesc baze de date de dimensiuni relativ reduse cu un număr mic de utilizatori.

O aplicație Microsoft Access este formată din următoarele tipuri de obiecte: tabele, module de interogare, formurile, rapoartele, macrourile și modulele, ce sunt legate împreună în mod coerent. La baza acestei baze de date se Visual Basic.

Avantajele acestor aplicații, apreciate pe deplin de utilizatori, sunt legate de ușurința de implementare a formurilor și a rapoartelor. Setarea proprietăților obiectelor se realizează într-un mod plăcut, folosind din plin facilitățile oferite de interfața grafică. Utilizatorul beneficiază de o serie de automatizări, ce sporesc viteza de implementare, prin verificările realizate în mod automat pentru stările în care se pot afla obiectele, execuția mai multor proceduri la apariția unui eveniment și dialogarea cu sistemul de operare pentru capturarea erorilor datorate unui eșec în comunicarea cu perifericele sau din cauza epuizării resurselor sistemului.

Se pot personaliza o serie de elemente, un exemplu de personalizare a răspunsului la un eveniment este execuția unui macro sau a unei proceduri eveniment (procedură Visual Basic ce se atașează unui form, raport sau unui element de control).

Funcționalitatea bazei de date poate fi extinsă prin folosirea de librării externe Visual Basic. Librările unei baze de date sunt colecții de proceduri și obiecte ce pot fi chemate din orice aplicație. Se pot utiliza în scopul de a stoca unele rutine de program pe care dorim să le folosim și la alte aplicații.

Baza de date poate fi accesată de mai mulți utilizatori simultan prin intermediul unei rețele, administratorul bazei de date având suficiente unelte de gestionare a drepturilor pentru fiecare utilizator în parte.

Limbajul QBE a fost dezvoltat inițial de către IBM, în anii 1970, pentru a ajuta utilizatorii în regăsirea informațiilor din bazele de date. Succesul limbajului QBE a fost atât de mare, încât această facilitate este furnizată - într-o formă sau alta - de către cele mai populare

sisteme SGDB, inclusiv Microsoft Access. Facilitatea Access QBE este ușor de utilizat și are capacități foarte puternice. Limbajul QBE reprezintă un mod de tratare virtual, pentru accesarea informațiilor dintr-o bază de date, prin utilizarea șabloanelor de interogare. Se utilizează limbajul QBE introducând exemple de valori direct într-un șablon de interogare, pentru a reprezenta ce se va obține prin accesul la baza de date - cum ar fi răspunsul la o interogare.

Limbajul QBE poate fi utilizat pentru a pune întrebări referitoare la informațiile conținute în unul sau mai multe tabele și pentru a specifica câmpurile care dorim să apară în răspuns. Se pot selecta înregistrări conform unor criterii specifice sau nespecifice și se pot efectua calcule cu informațiile cuprinse în tabele. De asemenea, limbajul QBE se poate utiliza pentru a efectua operații utile asupra tabelor, cum ar fi inserarea și ștergerea de înregistrări, modificarea valorilor câmpurilor sau crearea de noi câmpuri și tabele. Atunci când se creează o interogare cu ajutorul limbajului QBE, sistemul Microsoft Access construiește în fundal instrucțiunea din limbajul de interogare structurată SQL

Dezavantajul sistemului Microsoft Access constă în principal în numărul limitat de tipuri de date ce pot fi stocate. Nu permite stocarea de date multimedia și nu poate gestiona fluxuri de date ce sunt apelate de către utilizatori.

4.2.3. DB2 – IBM

DB2 Universal Database este una dintre cele mai populare baze de date ce se găsesc pe piață în acest moment. DB2 este disponibilă pentru un număr foarte mare de sisteme de operare cum ar fi UNIX, Microsoft Windows, OS/2, AS/400 și OS/390. Este dezvoltată pentru a lucra ca și bază de date distribuită în rețea, dar sunt permise mai multe instanțe să ruleze pe un singur calculator. Este posibil ca baza de date să fie partiționată pe mai multe calculatoare și să ruleze în mod concurent prin intermediul unei rețele sau a unui cluster de servere. Acest lucru permite divizarea bazelor de date de dimensiuni foarte mari și de creștere a eficienței prin rularea în paralel a partițiilor din baza de date, rezultând o reducere a timpului de exploatare.

DB2 și-a extins suportul de la datele tradiționale și înspre tipurile de date multimedia (imagini, video, audio) sau tipuri de date utilizator distincte și flat files. *Control Center* este o interfață grafică ce permite managementul mai multor stații de lucru, iar prin interfața *Data Warehouse Center* se poate simplifica procesul de proiectare, realizare și management al datelor din cadrul sistemului.

Modul de protejare al datelor a făcut ca DB2 să fie remarcată de utilizatori prin sistemul de control al pierderilor de date, accesul neautorizat și intrările invalide. Administrarea sistemului fiind posibilă de pe oricare din stațiile de lucru, existând și posibilitatea de administrare prin Web. *Satellite Administration Center* permite o administrare a serverelor exterioare de tip DB2 *Satellite servers*. Există o unealtă, *DB2 Everywhere Administration Center*, ce permite sincronizarea mai multor baze de date

Colaborarea cu alte sisteme de baze de date este realizată prin intermediul *Data Warehouse Center* ce permite o mare varietate de date relaționale sau non-relaționale (surse de date pot fi oferite de orice bază de date din familia DB2, Oracle, Sybase, Informix, Microsoft SQL Server, Flat files, IBM IMS și Virtual Storage Access Method (VSAM), IBM DB2 *DataJoiner*, sunt posibile peste 100 de translații din diverse formate de baze de date).

4.2.4. Oracle *interMedia*

Oracle *InterMedia* poate gestiona date multimedia (sunet, imagini, secvențe video) și poate oferi un suport deosebit pentru aplicațiile WWW deoarece asigură o viteză deosebită a tranzacțiilor, eficiență, scalabilitate și securitate. Oferă un set de tipuri de obiecte pentru sunete,

imagini și secvențe video suficiente pentru cele mai populare aplicații pe Internet, inclusiv formate Web. Aceste tipuri de obiecte pot fi extinse pentru a suporta mai multe aplicații specifice, de exemplu:

- adăugarea de noi formate;
- noi metode de codare, criptare și decodare, respectiv decriptare;
- noi metode de compresie și decompresie;
- scheme de specifice de indexare;
- metode de optimizare a modului de interogare;
- algoritmi de procesare a datelor;
- noi surse de date.

Oracle InterMedia permite deschiderea unor aplicații create anterior și care folosesc SQL-ul pentru accesarea obiectelor de tip sunet, imagine și video. Suportă de asemenea cele mai populare servere WEB care să acceseze baza de date prin Internet.

Moduri de accesare Oracle Intermedia - aplicațiile pot accesa Oracle Intermedia folosind ambele interfețe: relațională și de tip obiect. Aplicațiile pentru baze de date scrise în Java, C++ sau 3GL pot utiliza Oracle InterMedia prin intermediul unei interfețe moderne de librării de clase sau folosind PL/SQL sau Oracle Call Interface.

Oracle InterMedia oferă o gamă largă de utilitare legate de utilizarea obiectelor multimedia prin încărcarea acestora în cadrul unor tabele relaționale, cum ar fi:

- Clipboard (oferă o reducere a volumului de resurse grafice).
- API (application programming interface);
- Database table replication (posibilități de transferare de date între baze de date Oracle);
- Annotator (un sistem sofisticat de parsing multimedia);
- SQL*Loader (sistem de încărcare directă a datelor);
- Oracle Internet File System. Oracle InterMedia extrage metadata (informația despre datele multimedia) și asociază datele cu această informație în momentul introducerii acestora în baza de date.
- BLOBs (binary large objects - suportă date în format binar de foarte mari dimensiuni);
- BFILEs (file based large objects - suportă date stocate în formatul specific al sistemului de operare);

Dezvoltatorii de aplicații beneficiază de mediul de dezvoltare Jdeveloper de la Oracle.

Tipuri de obiecte sub Oracle InterMedia - tipurile de obiectele sub Oracle InterMedia sunt similare cu clasele din Java și C++, pentru a descrie obiecte de tip sunet, imagine și secvențe video. Aceste tipuri de obiecte se numesc ORDAudio, ORDImage și ORDVideo. O instanță a acestor tipuri de obiecte (date) este constituită din attribute, metadata, conținutul multimedia și metode. Conținutul multimedia este chiar informația propriu-zisă. Metadata este informația ce descrie conținutul informației multimedia cum ar fi: dimensiunea obiectului, modul de compresie, informații despre realizatorul obiectului, etc. Metodele sunt proceduri ce permit realizarea de diverse operații cu obiectele, cum ar fi stocarea acestora, extragerea metadatelor, compresia sau convertirea formatului imaginilor.

Aplicațiile folosesc Oracle InterMedia prin adăugarea unei sau a mai multor coloane tabelor existente sau prin adăugarea de noi tabele care conțin informația. Obiectele multimedia gestionate de Oracle InterMedia (ORDAudio, ORDImage și ORDVideo) folosesc un model comun de stocare [71]. Componenta multimedia a acestor obiecte poate fi stocată controlat în baza de date în BLOBs în timpul operației de tranzacție sau poate fi stocată în afara bazei de date fără un control în urma unei tranzacții. Din practică a rezulat în că multe cazuri este utilă

stocarea exterioară, deoarece facilitează schimburi mai rapide fără transferul efectiv al informației primare ci doar a caracteristicilor obiectului.

Metadata aferentă obiectului și metodele sunt întotdeauna memorate în interiorul bazei de date sub controlul Oracle InterMedia. Metadata este gestionată global pentru toate tipurile de obiecte multimedia, iar extragerea acestora se face automat pentru fiecare tip în parte. Metadata include:

- informații despre data stocată: tipul sursei, locația, nume etc.;
- informații despre ultima actualizare a informației;
- formatul datelor;
- tipul MIME;
- dimensiunea imaginilor, numărul de culori, formatul de reprezentare a culorilor, tipul de compresie etc.
- tipul de codare audio, numărul de canale, rata de eșantionare, compresia, durata, descriere;
- dimensiunea cadrului video, rezoluția, rata cadrelor, durata, numărul de cadre, compresia, numărul de culori, descriere, etc.
- tipul aplicației ce a creat obiectul, autorul, etc.

4.2.5. Direcții de dezvoltare în domeniul bazelor de date clasice

Se cunosc atât avantajele bazelor de date orientate pe obiecte, cât și a celor relaționale. O soluție ar fi îmbinarea celor două modele total diferite, într-un model comun care să cumuleze avantajele fiecărui model și să minimizeze dezavantajele. Stonebraker, în 1996, a prezentat următorul mod de clasificare a sistemelor de gestiune a bazelor de date [72]:

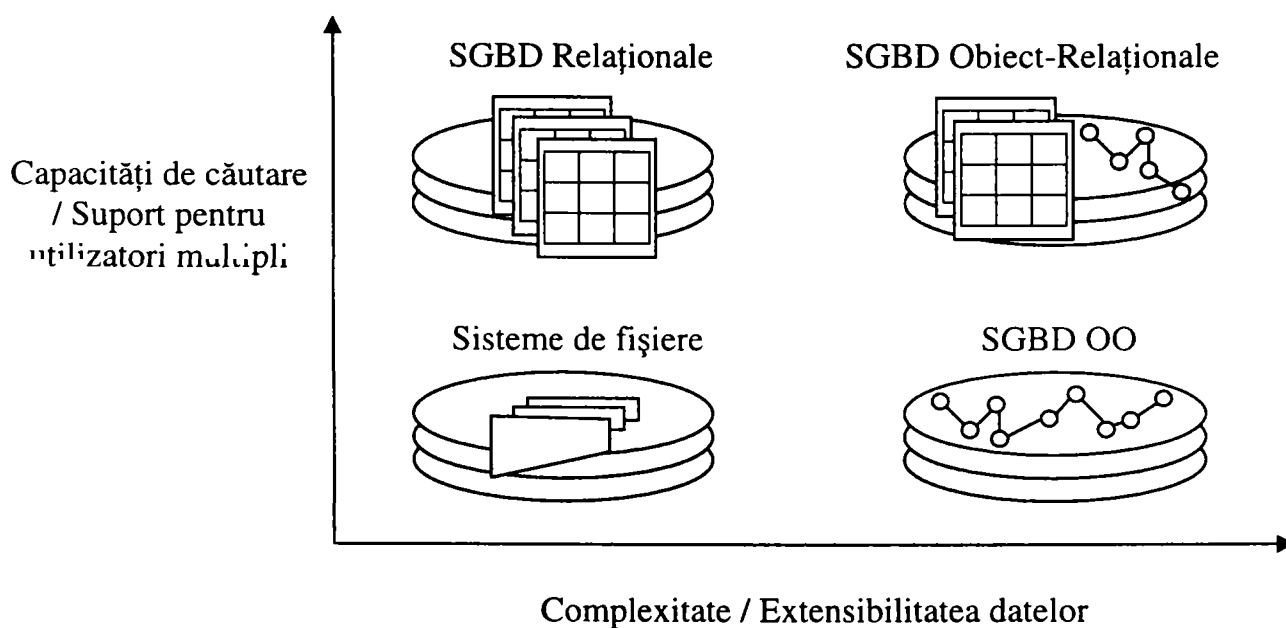


Figura 4-2 Clasificarea SGBD din punct de vedere al performanțelor

Avantajele principale ale extinderii modelului de date relațional provin de la caracteristicile de reutilizare și partajare. Reutilizarea derivă din posibilitatea de a extinde serverul SGBD, pentru a furniza centralizat funcționalitatea standard, în loc ca aceasta să fie codificată în fiecare aplicație. De exemplu, aplicațiile pot necesita tipuri de date spațiale care reprezintă puncte, linii și poligoane cu funcții asociate care calculează printre altele, distanța dintre două puncte, distanța dintre un punct și o linie, care stabilesc dacă un punct este conținut

în interiorul unui poligon și dacă două poligoane se intersectează. Dacă am putea încapsula, aceste funcționalități în server, atunci s-ar elimina necesitatea de a le defini pentru fiecare aplicație care are nevoie de ele și, în consecință, s-ar permite ca funcționalitățile să fie partajate de toate aplicațiile. De asemenea, aceste avantaje duc la creșterea productivității, atât pentru constructorul de software, cât și pentru utilizatorul final.

Un alt avantaj evident constă în faptul că abordarea relațională extinsă păstrează masa semnificativă de cunoștințe și experiență care a fost investită în dezvoltarea aplicațiilor relaționale. Acest avantaj este semnificativ, deoarece, multe organizații vor considera costul schimbării prohibitiv. Dacă noua funcționalitate este proiectată corespunzător, atunci această abordare trebuie să permită organizațiilor să profite în mod evolutiv de avantajele noilor extensii, fără pierderea beneficiilor oferite de caracteristicile și funcțiile bazelor de date curente. Astfel, un sistem SGBDOR ar putea fi introdus într-un mod integrat, sub formă de proiecte de verificare de concepte. Standardul SQL3 în curs de apariție, este proiectat astfel încât să fie compatibil cu standardul SQL curent, așa că orice sistem SGBDOR care se supune standardului SQL3 trebuie să asigure această capacitate.

Abordarea prin sisteme SGBDOR are dezavantaje evidente în ceea ce privește complexitatea și costurile ridicate ale acestora. Mai mult, unii dintre susținătorii abordării relaționale sunt de părere că simplitatea și puritatea modelului relațional, sunt pierdute în aceste tipuri de extensii. De asemenea, există alții care cred că sistemul SGBDR este extins pentru ceea ce va reprezenta o minoritate de aplicații, care nu realizează performanțe optime cu tehnologia relațională curentă.

În plus, partizanii orientării spre obiecte, nu sunt atrași nici de aceste extensii. Ei argumentează că terminologia sistemelor obiect-relaționale este revelatoare în acest sens. Sunt utilizate tipuri de date cum ar fi cele definite de utilizatori, în loc să se utilizeze modele de obiecte. Totuși, producătorii de sisteme SGBDOR sunt tentați să descrie modelele de obiecte ca extindere ale modelului relațional cu unele complexități adiționale. Prin aceasta, potențial, se trece peste trăsătura distinctivă de orientare spre obiecte, punând în evidență prăpastia semantică mare dintre aceste două tehnologii. Pur și simplu, aplicațiile de obiecte nu sunt atât de centrate pe date ca cele relaționale. Modelele și programele orientate spre obiecte combină profund relațiile și obiectele încapsulate, pentru a reflecta cât mai fidel lumea reală. Acest model definește mulțimi de relații mai largi decât cele din limbajul SQL, și presupune programe funcționale incluse în definițiile obiectelor. De fapt, obiectele nu sunt, în esență, extensii de date, ci un concept complet diferit, cu o putere mult mai mare de a exprima relațiile și componentele din lumea reală.

Manifestul bazelor de date de a treia generație - în mod asemănător, cu sintetizarea trăsăturilor bazelor de date orientate pe obiecte prin intermediul manifestului lansat de Atkinson în 1989, în 1990 Stonebraker propune "Manifestul sistemelor de baze de date din a treia generație" [73] în care sunt definite un număr de principii care trebuie îndeplinite de un sistem de gestiune al bazelor de date.

1. Un sistem de gestiune a bazelor de date din generația a treia trebuie să ofere un număr foarte mare de tipuri.
2. Moștenirea este proprietate ce oferă multiple avantaje, se recomandă a fi exploatată.
3. Funcțiile, procedurile, metodele ce implementează, precum și încapsularea în cadrul unei baze de date sunt idei bune de folosit.
4. Sistemul de gestiune al bazelor de date trebuie să atribuie indentificatori unici pentru înregistrări, numai dacă nu este disponibilă o cheie primară definită de utilizator.
5. Regulile (declanșatoarele, constrângerile) vor constitui o caracteristică principală în cadrul sistemelor viitoare. Ele trebuie asociate cu o funcție sau colecție.

6. Accesul programatic la baza de date trebuie să se efectueze prin intermediul unui limbaj neprocedural, de nivel înalt.
7. Trebuie să existe cel puțin două modalități de specificare a colecțiilor: prin utilizarea enumerării membrilor și prin folosirea limbajului de interogare pentru specificarea acestora.
8. Vederile reactualizabile sunt esențiale.
9. Indicatorii de performanță nu au aproape nimic de a face cu modele de date și nu trebuie să apară în cadrul acestora.
10. Sistemele de gestiune ale bazelor de date de a treia generație trebuie să fie accesabile prin intermediul unor multiple limbaje de nivel înalt.
11. Formele persistente ale unui limbaj de nivel înalt, pentru o diversitate de limbaje de nivel înalt, constituie o idee bună. Toate vor fi susținute deasupra unui singur sistem SGBD, prin extensii ale compilatoarelor și un sistem complex din timpul execuției.
12. Trebuie să se mențină limbajul SQL.
13. Interogările și răspunsurile rezultate din acestea trebuie să constituie nivelul de comunicație cel mai de jos dintre un client și un server.

Darwen și Date au publicat "Al treilea manifest" în 1995 și 1998 [74]. Acesta încearcă o redefinire a modelului relațional, dar care să accepte caracteristici specifice orientării înspre obiecte. Autorii consideră modelul relațional ca fiind suficient. Este respins limbajul SQL și este propusă utilizarea limbajului **D**. Manifestul de fapt se concentrează asupra limbajului **D**. Sunt prezentate sistematic recomandările și interdicțiile modelelor relaționale și a modelelor orientate pe obiecte, iar ca urmare propune câteva sugestii pentru ambele modele.

4.3. Baze de date multimedia

4.3.1. Evoluția bazelor de date spre aplicații multimedia

Atât bazele de date relaționale cât și cele orientate pe obiecte suportă ca tipuri de date numere întregi, în virgulă mobilă, caractere și șiruri de caractere. Pentru a suporta aplicații multimedia este necesar ca baza de date să suporte și tipuri de date foarte lungi bit, octet sau șiruri de caractere. Majoritatea bazelor de date stochează numai calea de acces înspre imagini sau secvențele video din baza de date, fiind mai mult orientate spre referințele acestora.

O altă funcție care lipsește de la bazele de date convenționale este căutarea bazată pe conținut. Tipic, interogarea și căutarea în baze de date convenționale sau baze de date orientate pe obiecte se bazează pe atributele obiectelor. Căutarea bazată pe conținut este relativ ușor de implementat în cazul obiectelor de tip text. În schimb definirea tuturor atributelor care caracterizează un obiect de tip video sau imagine este foarte dificilă. Complexitatea operației de indexare și căutare este dată de numărul mare de categorii de imagini și secvențe video ce există, pentru acestea fiind dificil de a se defini un număr de caracteristici redus care să permită o bună clasificare a acestora.

Caracteristic pentru bazele de date multimedia este și structurarea obiectelor sub diverse forme ierarhice, organizare ce se regăsește și în cazul sistemelor de stocare. Cercetările în acest domeniu au demonstrat că se poate utiliza un nivel ierarhic superior al bazei de date, restrâns ca dimensiune, pentru identificarea informației stocate. Astfel de sisteme folosesc arbori de căutare de tip B. Aplicațiile acestor baze de date se regăsesc în sistemele geografice informatizate [75], CAD [76], căutare în baze de date ce conțin fețe (aplicații de criminalistică) [77], etc. Ulterior au apărut prototipuri de baze de date multimedia generalizate ca și domenii de aplicații, dar care au o serie de similarități legate de cerințele utilizatorilor ce trebuie satisfăcute. Exemple de astfel de

sisteme sunt: VIMSYS ce folosește patru nivele ierarhice în definirea arhitecturii bazei de date orientate pe obiecte și este cu precădere orientată pentru stocarea de imagini [78]; GRIM-DBMS este o bază de date orientată pe obiecte, ce oferă facilități pentru stocarea, indexarea, interogarea și căutarea imaginilor, sunt incluse și câteva facilități de analiză a imaginilor, recunoaștere de obiecte și grupare pe baza caracteristicilor [79]; etc.

Există o mare varietate a prototipurilor sistemelor de baze de date [80], [81], [82], [83], [84] din mai multe puncte de vedere:

- Varietate a modelelor – baze de date relaționale extinse, orientate pe obiecte, bazate pe sisteme de căutare a conținutului, etc;
- Varietate a categoriilor de aplicații – aplicații ce folosesc tipuri de date spațial reprezentate, aplicații orientate pe conținutul informațional al înregistrărilor din baza de date, aplicații orientate pe descrierea de obiecte complexe sub formă ierarhică, etc;
- Varietate a scopului final al sistemului – interfețe grafice cu utilizatorul pentru aplicații multimedia, sisteme de căutare inteligente, modele de interogare, reprezentări de arhitecturi informaționale complexe, etc;
- Varietate a tipurilor de date multimedia suportate.

Ca și concluzie, pot spune că toate sistemele de baze de date studiate, au ca scop integrarea unui cât mai mare număr de tipuri de informații și diversificarea serviciilor oferite utilizatorilor: sisteme de interogare inteligente, motoare de căutare cât mai eficiente bazate și pe conținut.

4.3.2. Caracteristicile bazelor de date multimedia

Pentru a determina caracteristicile unei baze de date multimedia este bine ca pentru început să prezint principalele categorii de aplicații:

- Aplicații ce rulează de pe CD-ROM – există tendința la multe persoane să considere CD-ROM-ul asemănător unei baze de date multimedia deoarece permite stocarea unui număr foarte mare de informații, texte, sunete, imagini, secvențe video. De fapt, nu reprezintă decât un suport de stocare a informației.
- Aplicații ce permit organizarea și vizualizarea conținutului fișierelor și directoarelor de pe un calculator – există din ce în ce mai multe soft-uri ce permit o ușoară organizare a conținutului directoarelor prin afișarea de mici pictograme ce reprezintă de exemplu, o imagine stocată la mică rezoluție. Integrarea facilităților de copiere, mutare, ștergere îi face pe mulți utilizatori să considere aceste aplicații ca și baze de date.
- Programe video la cerere – „video on demand” – este cea mai spectaculoasă aplicație multimedia și se bazează pe transmiterea de programe video spre utilizator atunci când acesta le solicită. Are la bază o rețea de comunicații bine dezvoltată de mare viteză și servere de mare capacitate ce pot stoca cantități imense de programe video și care pot transmite simultan mai multe programe video [85], [86].
- Managementul documentelor – există mari cantități de informație ce sunt stocate sub formă de imagini, de exemplu diverse publicații mai vechi, ziare, microfilme, etc, unde este foarte important modul de indexare al informației, cum se poate ajunge la un anumit material iar dacă este posibil pe lângă tehnicile obișnuite de indexare să existe și variante de căutare bazate pe conținut – în general se folosesc tehnici de recunoaștere automată a caracterelor (OCR). Uzual se folosesc baze de date relaționale cu referințe la adresele de stocare ale imaginilor.
- BLOB – Binary Large Objects – de foarte multe ori bazele de date multimedia sunt reprezentate, sau considerate de unii ca fiind baze de date relaționale ce permit stocarea de

date de mari dimensiuni. Un BLOB poate conține sunet, imagine sau secvențe video, iar managementul de sistem al acestor baze de date relaționale (RDBMS) poate lucra cu acest tip de date și permite tranzacționarea acestora, în multe cazuri este permisă, de exemplu, afișarea imaginilor, redarea unui sunet sau rularea unei secvențe video direct de către interfața bazei de date.

- Bazele de date în sine pot fi considerate ca fiind aplicații multimedia mai ales dacă ele stochează obiecte multimedia. Se întâlnesc frecvent baze de date relaționale orientate pe obiecte ce suportă diverse tipuri de informații multimedia și care folosesc de asemenea ierarhii de clase atât pentru obiectele memorate cât și pentru structura sistemului (a bazei de date intrinseci). O categorie aparte de baze de date este reprezentată de către bazele de date orientate pe obiecte unde diversele câmpuri ale acestora sau atributele cele caracterizează obiectele multimedia sunt mult mai ușor de accesat printr-o astfel de reprezentare a lor. Cea mai des întâlnită utilizare a acestora este în cadrul bazelor de date multimedia ce folosesc pentru stocare obiectele aplicațiilor CAD sau CAM.
- CAD/CAM (Computer Aided Design / Computer Aided Manufacturing) sunt aplicații ce lucrează cu obiecte 3D. Mai mulți utilizatori pot accesa simultan același bibliotecă de elemente 3D.
- Agenții de știri, posturi de radio și televiziune – toate posturile de radio și televiziune folosesc medii informatizate de gestionare a informațiilor, atât pentru cele care sosesc de la diverse surse cât și pentru cele ce se transmit sau se stochează pe diverse suporturi. Pentru adunarea informațiilor și știrilor, într-un post de televiziune, de exemplu, se folosesc medii ca: publicații în format electronic, email, pagini Web, faxuri, date transmise de agenții specializate, etc. Toate acestea sunt sub formă numerică și pot fi stocate în baze de date adecvate. Tot ce este transmis este în general stocat pentru a exista o istorie a ceea ce a fost, sau pentru a se retransmite secvențe sau programe întregi. Aceste date pot servi și drept referință pentru emisiuni ulterioare. Se pot realiza și sisteme de știri și baze de date despre diverse evenimente ce pot fi accesate de către utilizatori direct prin Internet. De cele mai multe ori utilizator dorește ca aceste informații să fie cât mai variate, să conțină imagini, sunete și secvențe video, adică se fie reprezentate de o bază de date multimedia.
- Aplicații specifice se pot întâlni și în industrie unde sunt necesari algoritmi de căutare orientați pe aplicația respectivă.
- Sisteme educaționale – societatea actuală se caracterizează printr-o nevoie de perfecționare continuă și de înalt nivel a individului. De cele mai multe ori, aceasta se face după terminarea studiilor obișnuite, iar problemele familiare și de servicii impun utilizarea altor moduri de studiu, variantele tradiționale ne fiind suficiente și satisfăcătoare. În acest context s-au dezvoltat foarte mult în ultimul timp cursuri la distanță sau prin corespondență care apelează din plin la facilitățile oferite de aplicațiile multimedia, atât pe CD-ROM, cât și pe Internet. Este imperativ necesar să se folosească baze de date pentru stocarea cursurilor, pentru evidența cursanților și comunicația cu aceștia.
- Publicitate și marketing – majoritatea companiilor folosesc diverse medii pentru a prezenta produsele sau serviciile, apelându-se în general la baze de date multimedia pentru stocarea și accesarea acestora atât de clienți cât și de utilizatori.

Aplicațiile prezentate mai sus sunt doar câteva din domeniile unde este întâlnit conceptul de „multimedia” și în special aplicații ce se folosesc de facilitățile oferite de către bazele de date multimedia.

Respectând argumentele prezentate în cele câteva tipuri de aplicații prezentate, se poate trage o primă concluzie referitoare la bazele de date multimedia, din perspectiva utilizatorilor în primul rând: este nevoie ca baza de date să suporte cât mai multe formate și tipuri de date și să

ofere în același timp posibilitatea de stocare de obiecte de foarte mari dimensiuni caracteristice aplicațiilor multimedia.

Sistemul de management al bazelor de date multimedia se bazează pe integrarea a trei tehnologii fundamentale:

- Baze de date cu capabilități convenționale;
- Sisteme de stocare ierarhice;
- Capacitate de căutare a informație bazată pe conținutul acesteia.

Schematic o bază de date multimedia este reprezentată în figura următoare:

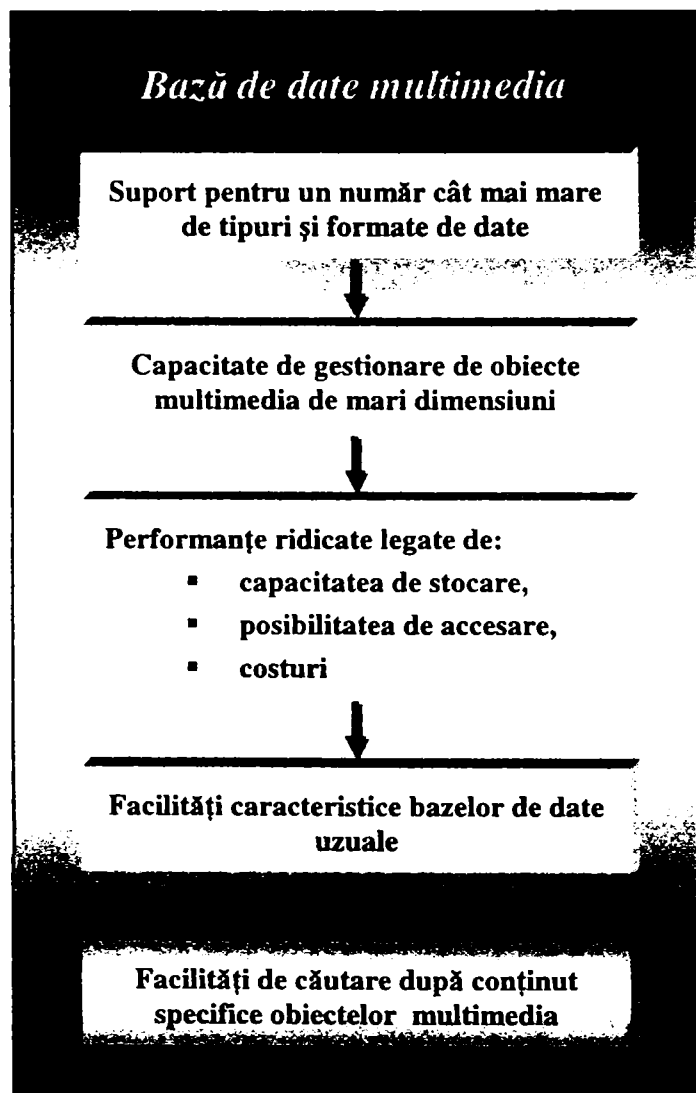


Figura 4-3. Cerințele utilizatorilor pentru baze de date multimedia

Sistemul de management al unei baze de date multimedia trebuie să suporte următoarele tipuri de date: text, sunet, obiecte grafice (2D și 3D), imagini, secvențe video, animații, scripturi și reprezentări pentru obiecte multimedia complexe [87], [88].

Prin faptul ca o baza de date trebuie să „suporte”, înțeleg, că anumite caracteristici și tipuri de date sunt recunoscute automat de către baza de date. Un tip este reprezentat de structura datelor și operațiile care se pot realiza cu acestea. Interfețele bazelor de date trebuie să permită invocarea de operații cu aceste tipuri de date, să permită rularea de predicate, să actualizeze substructuri ale acestor tipuri de date [89], [90].

Prin capacitatea de stocare o bază de date nu trebuie să fie limitată, ea trebuie să reprezinte un sistem deschis pentru extinderi ulterioare. Din acest punct de vedere trebuie conceput de la început sistemul de indexare astfel încât să nu apară limitări, recomandabil este să

se folosească structuri modulare organizate pe diverse ierarhii. Problemele de limitare a capacității de stocare se transmit și înspre partea hardware a sistemului. Sistemul hard trebuie prevăzut de la început cu structuri paralele atât pe partea de tratare a conectărilor din partea utilizatorilor la baza de date, cât și din punct de vedere al accesului la partea de stocare efectivă a informațiilor (folosirea de sisteme de hardisk-uri paralele).

Tipuri de date	Format	Volum date	Rată de transfer
Text	ASCII	1MB / 500 pagini	2 KB / pagină
Imagine alb-negru	G3/4 – fax	32 MB / 500 imagini	64 KB / pagină
Imagine color	TIFF	1.6 GB / 500 imagini	3.2 MB / imagine
	JPEG	0.2 GB / 500 imagini	0.4 MB / imagine
Voce	μ-law liniar	2.4 MB / 5 minute	8 KB / sec
	ADPCM	0.6 MB / 5 minute	2 KB / sec
	MPEG audio	0.2 MB / 5 minute	0.67 KB / sec
CD - muzică	CD-DA	52.8 MB / 5 minute	176 KB / sec
Video – calitate medie	PAL	6.6 GB / 5 minute	22 MB / sec
Video – calitate bună	HDTV	33 GB / 5 minute	110 MB / sec

Tabel 4-2 Tipuri de date multimedia și volumul ocupat de acestea

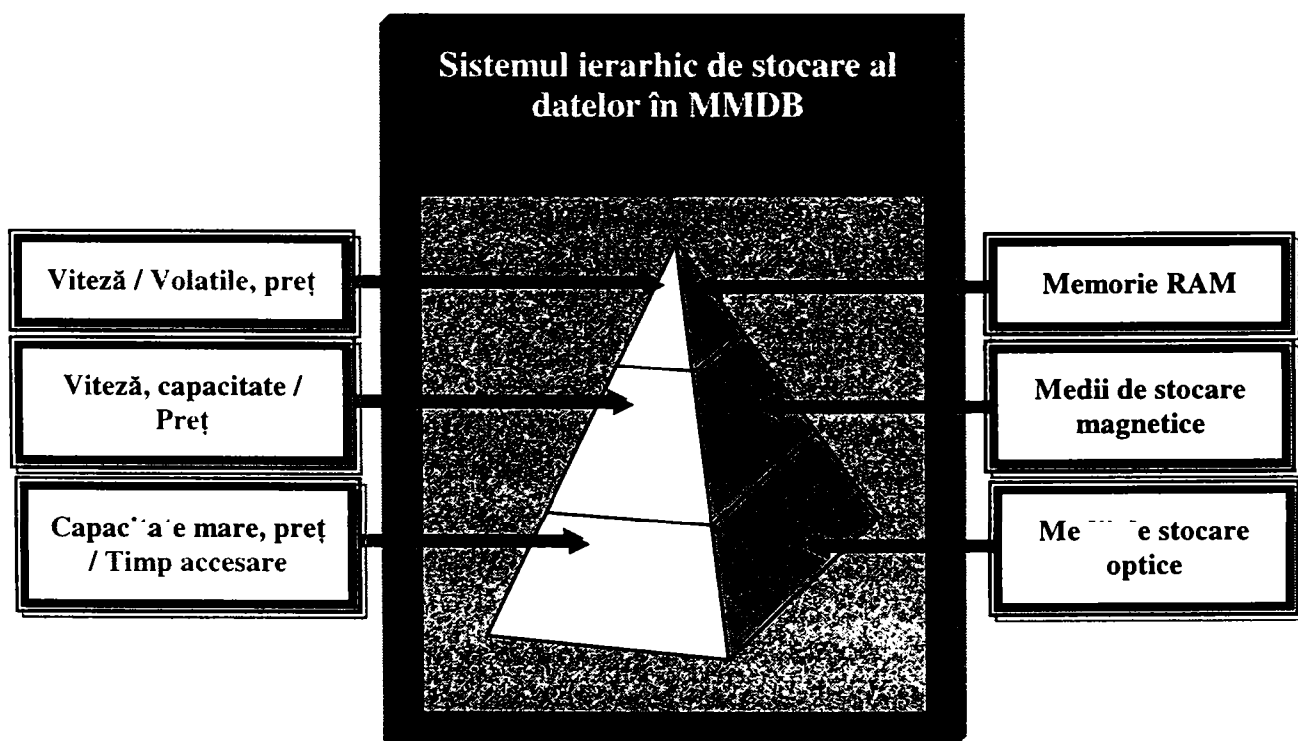


Figura 4-4 Sistemul ierarhic de stocare al datelor

Se pune problema și a unei bune gestionări și organizări a resurselor din punct de vedere financiar. O viteză mare de lucru pentru o bază de date se obține dacă datele acestea sunt stocate în memorii de mare viteză RAM. Prețul acestora este relativ mare și asigură o persistență a datelor, întreruperea alimentării de exemplu poate determina pierderea datelor. De aceea se preferă încărcarea temporară a datelor în memorii RAM de pe suporturi magnetice sau optice, iar pentru siguranță, în funcție de modificările apărute să se asigure o actualizare a datelor pe

hardisk-uri. În cazul datelor ce nu se modifică, se recomandă folosirea de suporturi optice pentru date, fiind dat prețul mai scăzut al acestora. Se poate realiza din acest punct de vedere o ierarhizare a modului de stocare și manipulare a datelor pentru îmbunătățirea performanțelor raportată la prețul de cost.

Capabilitățile unei baze de date sunt caracterizate de următoarele 9 elemente [91].

Persistența reprezintă abilitatea unui obiect de a exista (persista) la diverse invocări ale programului. În multe aplicații multimedia este văzută ca fiind abilitatea de stocare fișierele multimedia de către sistemul de operare. În aplicații de genul bazelor de date acestea pot fi de două feluri:

- Tranzitorii – care sunt valabile doar în interiorul programului sau pe perioada tranzacției. După încheierea tranzacției aceste date sunt șterse.
- Persistente – sunt datele memorate (stocate) în afara aplicației, rămân și după încheierea tranzacțiilor. Ele rezistă chiar dacă sistemul a avut probleme („a căzut” – probleme de hardisk, de exemplu) sau dacă au existat evenimente ce au blocat aplicația ce rula. Cu alte cuvinte pot spune că sunt obiecte ale bazei de date recuperabile.

Tranzacțiile sunt secvențe de rutine ce sunt executate în întregime sau nu. Tranzacțiile sunt atomice. Dacă un utilizator realizează o actualizare a unei baze de date persistente cu o tranzacție, atunci toate actualizările vor fi vizibile pentru exterior sau nici o actualizare nu va fi văzută spre exterior. O tranzacție poate fi realizată (comisă - committed) sau abandonată (aborted). În cazul bazelor de date multimedia tranzacțiile trebuie să garanteze consistența datelor din punct de vedere al locației elementelor și din punct de vedere al relațiilor ce există între aceste date.

Pot spune, că pentru aplicația implementată, a fost urmărit ca atât conținutul unei date, cât și atributele acesteia să fie inserate, șterse sau actualizate, asigurându-se astfel condiția de consistență a datelor din baza de date.

Controlul concurenței - în medii tipice managementului bazelor de date, tranzacțiile se realizează în mod concurențial. Pentru a se garanta integritatea bazei de date și consistența tranzacției, managementul de sistem al bazei de date impune serializarea ordini de execuție a operațiilor, deci ele se realizează una după alta pentru a nu exista riscul că mai multe tranzacții modifică în același timp un obiect din baza de date. Pentru a se garanta serializarea tranzacțiilor managementul de sistem al bazei de date folosește strategii de control concurențial. Există trei tipuri de algoritmi de control al concurenței:

- Time stampe ordering
- Optimistic algorithm
- Pessimistic or locking algorithm

Aceștia la rândul lor pot fi de mai multe tipuri. În multe aplicații multimedia este necesar să se folosească tranzacții numai de citire a datelor (read-only). Pentru aceste tipuri de aplicații se pretează a se folosi algoritmi de tipul time-stamp ordering și optimistic algorithms deoarece oferă performanțe mult mai bune [92].

Refacerea datelor - sistemul de management al bazei de date trebuie să garanteze că rezultate parțiale sau actualizări parțiale ale tranzacțiilor ca rezultat al unor eșecuri de procesare nu se vor propaga înspre baza de date persistentă. Sunt trei mari categorii de eșecuri de sistem ce trebuie corectate:

- Eșecuri datorate erorilor de tranzacție;
- Eșecuri datorate sistemului;
- Eșecuri datorate erorilor diverselor medii utilizate.

În funcție de posibilitățile de refacere a datelor, oferite de un sistem de management al bazelor de date, se pot caracteriza performanțele bazei de date. În cadrul bazelor de date multimedia, prin folosirea unui sistem ierarhic de stocare (on-line, near-line, off-line), procesul de refacere al datelor poate devenii foarte complex.

Sistemul de interogare - prin interogarea unei baze de date se obțin submulțimi sau subobiecte din cadrul unei baze de date ce conține diverse mulțimi și obiecte. Interogarea poate fi considerată ca fiind expresii de înalt nivel care prin modul lor declarativ permit utilizatorilor construcția cerințelor lor de la baza persistentă de date. Câteva din limbajele de interogare au fundații teoretice solide, de exemplu relații de calcul sau algebra booleană, alte baze de date folosesc facilități de interogare dezvoltate local doar pentru acea aplicație. În aplicațiile multimedia se folosesc de foarte multe ori metode de interogare interactive și cu un conținut vizual bogat, intuitiv. Pentru interogare se folosesc atributele obiectelor multimedia, cuvinte cheie sau proprietăți ale conținutului obiectului. Deoarece sunt și proprietăți ale obiectelor multimedia care la extragere nu sunt foarte exacte, nu oferă rezultate foarte precise (recunoașterea obiectelor de exemplu), se folosesc de foarte multe ori metode probabilistice de evaluare a rezultatelor sau chiar prezentarea de rezultate intermediare din care utilizatorul să poată extrage ceea ce dorește sau caracteristica relevantă pentru acea căutare. Cu alte cuvinte se realizează o căutare prin interogări iterative pentru obținerea de rezultate cât mai exacte.

Utilizarea de mai multe versiuni de date – o altă proprietate a sistemului de management al bazelor de date, ce se dovedește a fi utilă în practică, este posibilitatea de stocare a unor date mai vechi din baza de date, date care în general au fost actualizate și modificate. Este bine ca această operație de salvare a datelor intermediare să se facă automat, fără eforturi din partea utilizatorului. Sunt foarte multe cazuri în care mai sunt necesare stări anterioare ale datelor persistente.

Integritatea - managementul de sistem al bazei de date urmărește consistența bazei de date de la o tranzacție la alta. Predicatele sau condițiile care caracterizează baza de date la un moment dat, reprezintă consistența acelei baze de date. Predicatele se pot aplica și obiectelor sau valorilor atributelor din baza de date. Predicatele ce capturează consistența unei baze de date sunt numite constrângeri de integritate ale bazei de date.

Securitatea – sistemul de management al bazelor de date trebuie să conțină și elemente de securitate a datelor și a accesului la date, la obiectele persistente. Securizarea unei baze de date multimedia este de multe ori dificil de realizat, deoarece se lucrează cu diverse tipuri de date, pentru care există numeroase tabele cu atribute ale acestora, de multe ori comune pentru diverse obiecte. Acest lucru face ca atribuirea proprietăților de accesare să de multe ori o sarcină foarte dificilă. Este necesară de multe ori organizarea accesului pe mai multe nivele ierarhice și diverse grupuri de utilizatori.

Performanțele - optimizarea unei baze de date constituie un proces bazat pe mai multe etape:

- organizarea și structurarea categoriilor de obiectelor stocate;
- structurarea modului de stocare ierarhic, asigurarea persistenței;
- optimizarea tranzacțiilor;
- rezolvarea problemelor concurențiale;
- asigurarea refacerii datelor în cazul eșecului unei tranzacții (asigurarea refacerii datelor pentru un număr cât mai mare de tipuri de eșecuri posibile);
- optimizarea interogării, stabilirea de interogări multiple, ierarhice cu ordonare probabilistică și interactive;
- optimizarea și automatizarea procesului de salvare a stărilor bazei de date;

- dezvoltarea sistemului de verificare al integrității bazei de date;
- organizarea sistemului de securizare, definirea de utilizatori și drepturi pentru aceștia.

Cea mai importantă operație pentru utilizatorii de baze de date este operația de căutare și regăsire a informațiilor (Information retrieval - IR) în baza de date. De aceea, o atenție deosebită trebuie să se acorde pentru optimizarea acestui proces, în mod special dacă căutarea unei informații se bazează pe conținut.

4.3.3. Multimedia și bazele de date orientate pe obiecte

Bazele de date orientate pe obiecte sunt considerate ca fiind ideale pentru a suporta aplicații de tipul baze de date multimedia. Orientarea pe obiecte și multimedia au foarte multe elemente comune. Scopul orientării pe obiecte este de a realiza programe pentru calculatoare care să modeleze cât mai bine lumea reală. Scopul aplicațiilor multimedia este de a prezenta lumea înconjurătoare cât mai fidel posibil. Deci între punctul de vedere al realizatorilor de modele și punctul de vedere al realizatorilor de reprezentări există foarte multe asemănări.

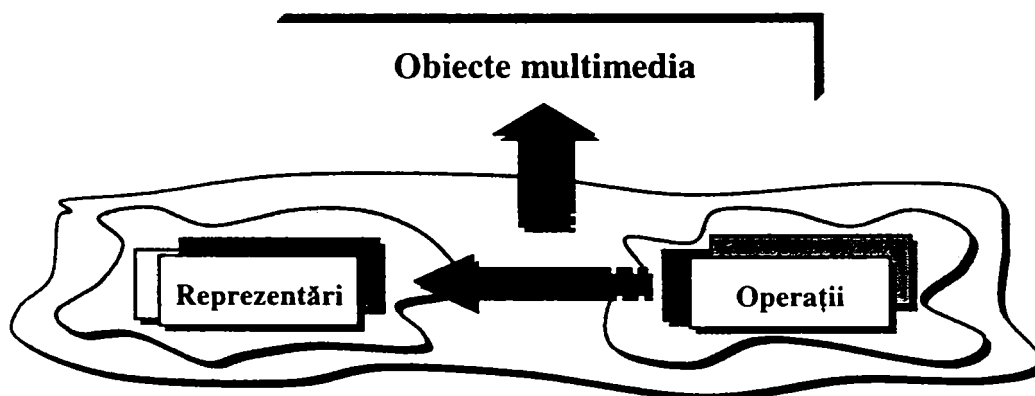


Figura 4-5 Reprezentarea unui obiect multimedia dintr-un punct de vedere comun pentru realizatorii de aplicații multimedia și pentru realizatorii de programe orientate pe obiecte.

O importantă similaritate este prezentată în figura de mai sus unde un obiect este văzut prin prisma reprezentării acestuia la care se adaugă operațiile care se pot realiza cu acesta.

Fiecare tip de date multimedia este un obiect cu structură proprie de stocare și un set particular de operații ce definesc starea unor obiecte specifice multimedia.

Caracteristici model	Model bază de date relațională	Model bază de date orientată pe obiecte	Model bază de date multimedia
Structură	Relații	Obiecte, attribute	Fluxuri continue de date
Mod de operare	SQL	Metode	Operații dependente de timp
Constrângeri	Chei primare	Integritate referențială	Parametrii ce calitate ai serviciului <i>QoS</i>

Tabel 4-3 Diferențe între modelele de baze de date

În tabelul 5-2 sunt prezentate sintetic trei modele de baze de date și diferențele ce apar între acestea. Se observă din tabel că este necesar ca pentru o bază de date multimedia ce are în general o structură orientată pe obiecte să se introducă parametrul *timp*, de care depind o parte

din date și operațiile care se realizează cu ele, iar ca și constrângere apare noțiunea de *calitate* a serviciului oferit [93].

Avantajele folosirii tipuri de date abstracte în bazele de date multimedia pot fi concluzionate în modul următor:

- Permit o mai bună conceptualizare și modelare a lumii reale;
- Măresc robustețea sistemului – prin folosirea acestui tip de date este permisă la folosirea variabilelor să se verifice de fiecare dată tipul datei respective pentru a nu avea erori în timpul rulării, verificarea în acest mod a integrității și a operațiilor efectuate permit creșterea corectitudinii programelor;
- Performanțele sistemului cresc – cunoașterea tipurilor de obiecte permite creșterea vitezei de compilare; sunt posibile de asemenea strategii de grupare a obiectelor persistente în vederea stocării; folosirea unei exprimări abstracte a obiectelor face posibilă ascunderea unor detalii de proiectare, la implementare folosindu-se doar obiectul ca o singură entitate;
- Permite o mai bună captură a semanticii tipurilor – folosirea tipurilor de date abstracte grupează sau localizează mai bine operațiile și reprezentarea atributelor. Acesta a permis o mai ușoară asociere a operațiilor cu obiectelor multimedia cu care acestea lucrează.
- Separă implementarea de specificații – sunt permise modificări și îmbunătățiri ale aplicației fără a se modifica interfața publică a datelor de tip abstract.
- Permite extensibilitatea sistemului – prin re folosirea unor componente software, fiind astfel mai ușoară realizarea aplicației și depanarea acesteia.

Aceste considerente sunt foarte utile din punct de vedere practic, mai ales că aplicațiile multimedia suportă conceptul de introducere - *plugged* - de noi tipuri de date, ceea ce permite sistemelor complexe să reutilizeze anumite unități.

Folosirea moștenirilor în cadrul bazelor de date multimedia prezintă două mari avantaje atât pentru cei care implementează aplicația cât și pentru utilizatori:

- Oferă un model natural de organizare a informației – de exemplu, elemente multimedia, elemente grafice, periferice multimedia pot fi folosite într-o organizare naturală și directă.
- Permite partajarea codului și a reprezentării, reducând foarte mult efortul de dezvoltare al aplicației. Este permisă definirea de noi clase în vârful ierarhiei. Instanțe sau ierarhii de clase ce implementează de exemplu importul sau exportul unei imagini, afișarea sau editarea unui grafic, acestea pot fi reutilizate sau pot fi extinse pentru aplicarea lor în cadrul bazelor de date multimedia.

Identitatea unui obiect oferă o serie de avantaje pentru bazele de date multimedia:

- Identitatea unui obiect ne permite reprezentarea directă a obiectului în spațiul obiectelor structurate sub forma unui graf. În multe aplicații este necesar ca unele instanțe ale unor clase definite de utilizator să fie partajată referențial.
- Folosirea identității unui obiect permite utilizatorilor să nu mențină integritatea referențială. În aplicațiile multimedia tipic se folosesc reprezentări ale obiectelor sub formă de graf, cum ar fi de exemplu documentele complexe cu multiple pagini și în fiecare pagină se găsesc diverse elemente. Aceste tipuri de obiecte pot fi suportate de baza de date fără a fi necesară o suprascriere a constrângerilor de integritate referențială (diferit față de bazele de date referențiale).
- Prin folosirea aceluiași *token* se rezolvă probleme de referință în cadrul memoriei sau al bazei de date.
- Asocierea diverselor operații cu identitatea obiectelor oferă o unealtă foarte puternică de manipulare a funcționalității obiectelor din cadrul aplicațiilor de tip baze de date multimedia.

- Medii pentru obiecte distribuite și platforme interoperabile – OLE și OpenDoc – încorporează conceptul de identitate a obiectului.
- Încadrarea suportului de identitate a obiectului în bazele de date multimedia oferă o uniformitate în suportarea obiectelor independente de aplicație sau locație.

4.3.4. Moduri de creștere a performanțelor unei baze de date

Pe lângă stocarea ierarhică a informației, posibilitate de căutare după conținut și alte proprietăți prezentate mai sus, unei baze de date multimedia i se mai pot adăuga și alte caracteristici secundare cum ar fi:

- Tipuri de date spațiale și sistem de interogare spațial – când se manipulează tipuri de date de tip imagine ce au sub-obiecte iar în imagine există relații spațiale între ele, de exemplu în cazul bazelor de date ce conțin informații geografice, cartografie, etc. Pentru indexarea unor astfel de obiecte este bine să se integreze și relațiile dintre sub-obiecte, pentru astfel de reprezentări se va dezvolta și un sistem de interogare adecvat.
- Sistem de interogare interactiv, cu redefinire a parametrilor de căutare în mod iterativ, bazat pe răspunsul utilizatorilor la rezultatele parțiale ale căutării afișate – acest tip de interogare presupune o bună interfață grafică cu utilizatorul, care să permită utilizatorului formarea interogării în mod iterativ și intuitiv. Se poate apela și la o primă etapă de interogare prin exemple, care să ghideze utilizatorul spre unul din domeniile definite de implementatorul bazei de date, domeniu care conține informațiile căutate. În cazul în care există incertitudini în modul de interogare este bine să fie afișate mai multe rezultate în ordinea relevanței. Prin alegerea unuia dintre răspunsuri se poate redefini, mult mai fin, mai exact, parametrii de căutare.
- Sistem de extragere automată a caracteristicilor și indexare – în cazul imaginilor, de exemplu, la introducerea unei imagini în baza de date este bine ca unele caracteristici să se calculeze la momentul respectiv (calculul valorii medii, a dispersiei, diverse momente și histograme). Integrarea acestor caracteristici în baza de date o dată cu obiectul inițial poate scurta timpul de căutare ce folosește acele caracteristici, ne mai fiind necesară calcularea lor în momentul căutării, ele existând în baza de date.
- Sistem de indexare multidimensional – se știe că folosirea unei singure chei pentru accesul la informații, cazul structurilor sub formă de arbore de tip B , este mai puțin performantă decât sistemele de indexare sub formă de arbore de tip R sau K , ce folosesc indexare spațială (cu mai multe chei de indexare) – se recomandă deci, folosirea mai multor chei de indexare pentru aplicațiile multimedia.
- Gruparea după obiecte complexe – în cazul aplicațiilor CAD/CAM este bine ca obiecte complexe constituite din mai multe sub-obiecte să se păstreze legătura cu obiectul principal care le încadrează și de la acestea să se meargă mai departe la căutarea sub-obiectelor.
- Într-o bază de date multimedia trebuie să existe posibilitatea definirii de noi tipuri de date care să fie transparente în relația aplicație-utilizator.
- Conceperea software a bazei de date multimedia trebuie să pornească de la idea modularității și dezvoltării ulterioare a bazei de date.
- Sistemul de afișare al rezultatelor căutării trebuie să permită afișarea acestora în diverse moduri. De multe ori relevanța căutării și a rezultatelor obținute depinde de modul în care acestea sunt prezentate. De exemplu, dacă se caută imagini similare cu o imagine dată, nu este relevantă afișarea unei liste cu sursele fișierelor ce conțin imagini, ci este mult mai bună o afișare de pictograme la scara redusă a imaginilor sau afișarea unei liste de imagini în

ordonate după parametrii de căutare. O ordonare după parametrii de căutare, poate ajuta la redefinirea acestora în scopul îmbunătățirii rezultatelor obținute la căutare.

- O bază de date multimedia poate funcționa bine doar dacă sunt îndeplinite și anumite condiții legate de resursele hardware, astfel se recomandă o deplină transparență a echipamentelor necesare de exemplu pentru accesarea bazei de date, dar și ale resurselor serverului, sistemului de comunicare client/server, etc. De exemplu, dacă se caută secvențe video, numărul de utilizatori ce accesează serverul poate fi destul de limitat atât din partea serverului care nu poate reda fizic decât un anumit debit de date (nu pot fi redade simultan decât un anumit număr de fluxuri de date video, limitări din partea sistemului de stocare, debitul harddisk-urilor) sau limitare a rețelei de comunicare (debitul asigurat de acestea este limitat). Aceleași resurse sunt partajate de mai mulți utilizatori. De aceea recomand utilizarea de mesaje de eroare care să informeze utilizatorul de motivul insatisfacerii cerinței acestuia, eventual o posibilă soluție (de exemplu să aștepte reducerea numărului de utilizatori sau să folosească echipament cu alte performanțe) și în nici un caz să nu se ajungă la blocarea aplicației. O soluție pentru cazul secvențelor video, de exemplu, ar fi folosirea unui sistem de indexare ierarhic care să nu transmită spre utilizator de la început o listă de secvențe video, ci o serie de imagini statice (ce ocupă mai puțin spațiu și debit) care să reprezinte câteva cadre cheie din secvența video.
- Din punct de vedere hardware, echipamentul trebuie să fie structurat modular, astfel încât să se poată realiza dezvoltări ulterioare fără o reproiectare a suportului software. Trebuie să existe o legătură între modularitatea software și cea hardware.
- Pentru bazele de date multimedia o caracteristică importantă o reprezintă dependența datelor de timp. De aceea, trebuie luate măsuri care să evite probleme de incompatibilitate a modului de reprezentare a informației de timp. O persoană care accesează baza de date din SUA va avea o altă referință temporală față de un utilizator din Europa, este deci necesară folosirea unui mod de referință temporală care să includă refacerea informației temporale la un sistem de referință fix, bine precizat (se va folosi o reprezentare pentru oră de forma: *10:48:55 – GMT: +02:00* ceea ce permite exacta identificare temporală a unui eveniment de exemplu). O problemă similară poate apare și în cazul altor unități de măsură.

Ca și concluzie, pot spune, că folosirea recomandărilor de mai sus, permite realizarea unui mecanism de indexare a datelor performant, bazat pe o modelare semantică, pe o modelarea abstractă a datelor dar și pe o modelare bazată pe conținutul informației.

Sunt permise dezvoltări ulterioare din punct de vedere conceptual (adăugarea de noi module și tipuri de date la baza de date multimedia existentă), din punct de vedere software și din punct de vedere hardware.

4.3.5. Sisteme de gestionare a bazelor de date distribuite

Posibilitatea de a realiza o structură de baze de date de mare complexitate care să fie localizate în diverse locații a fost o provocare pentru toți dezvoltatorii de aplicații din acest domeniu. În 1987, Date a sintetizat un set de obiective pentru sistemele de gestionare a bazelor de date distribuite [94]:

- Autonomia locală - siturile dintr-un sistem distribuit trebuie să fie autonome: datele locale sunt deținute și gestionate local; operațiile locale rămân pur locale; toate operațiile dintr-un sit sunt controlate de către situl respectiv.
- Nici un fel de dependență de un sit central - nu trebuie să existe nici un sit fără de care să nu poată opera sistemul. Aceasta implică faptul că nu trebuie să existe servere centrale pentru servicii, cum ar fi administrarea tranzacțiilor, detectarea erorilor, optimizarea interogărilor și administrarea globală a sistemului.

- Operarea continuă - ideal este ca nu fie necesare opriri planificate ale sistemului pentru operații de adăugare sau eliminare a unui sit din sistem, crearea și ștergerea dinamică a fragmentelor dintr-unul sau mai multe situri.
- Independența de locație - utilizatorul trebuie să aibă posibilitatea de a accesa baza de date din interiorul oricărui sit. Utilizatorul trebuie să aibă posibilitatea de a accesa toate datele ca și cum ar fi stocate în situl său, indiferent de locul unde acestea sunt stocate din punct de vedere fizic.
- Independența de fragmentare - utilizatorul trebuie să aibă posibilitatea de a accesa datele, indiferent de modul în care sunt fragmentate.
- Independența de reproducere - utilizatorul trebuie să nu fie conștient de faptul că datele au fost reproduse. Astfel, utilizatorul nu trebuie să aibă posibilitatea de a accesa direct o anumită copie a unui articol de date și nici nu trebuie să reactualizeze special toate copiile unui articol de date.
- Prelucrarea distribuită a interogărilor - sistemul trebuie să aibă capacitatea de a prelucra interogări care se referă la date aflate în mai multe situri.
- Prelucrarea distribuită a tranzacțiilor - sistemul trebuie să accepte tranzacția ca unitate de refacere. Sistemul trebuie să garanteze că atât tranzacțiile globale, cât și cele locale se conformează regulilor ACID pentru tranzacții și anume: caracterul atomic, coerența, izolarea și durabilitatea.
- Independența de hardware - este necesar ca sistemul de gestiune al bazelor de date distribuite să poată fi rulat pe o diversitate cât mai mare de platforme hardware.
- Independența de sistemul de operare - este necesar ca sistemul de gestiune al bazelor de date distribuite să poată fi rula pe o diversitate cât mai mare de sisteme de operare.
- Independența de rețea - este necesar ca sistemul de gestiune al bazelor de date distribuite să poată fi rula pe o diversitate cât mai mare de rețele de comunicație separate.
- Independența de baza de date - este posibil ca local să existe diverse sisteme de gestiune a bazelor de date ce rulează local, eventual pot exista și diferite modele de baze de date.

Ultimele reguli sunt foarte dificil de îndeplinit.

Pentru aplicația propusă în această lucrare, adică optimizarea activităților dintr-un studio de televiziune, realizarea unui sistem de gestionare a bazelor de date distribuit este un imperativ ce trebuie avut în vedere de la începerea proiectării aplicației. Recent, se remarcă, implementarea de mici studiouri locale ce aparțin de un studio general. Gestionarea și transferarea informațiilor dintre acestea fiind una dintre cele mai importante probleme ce trebuie rezolvate într-un mod cât mai performant.

4.4. Indexarea bazelor de date multimedia

4.4.1. Introducere

Problema pe care trebuie să o rezolve un sistem de indexare, într-o bază de date multimedia, este creșterea vitezei de căutare. Acest lucru se poate realiza prin intermediul modului în care se localizează informației în baza de date, prin modul în care este caracterizată, cuantificată, distanța dintre două obiecte[95], [96].

Tehnicile de indexare și grupare sunt folosite cu succes în cadrul bazelor de date relaționale și a celor orientate pe obiecte [97]. Folosind un index, sau o cheie pentru operația de căutare, timpul de căutare se reduce foarte mult în cazul bazelor de date de foarte mari

dimensiuni. În general, indiferent de tipul bazelor de date se folosesc sisteme de indexare a înregistrărilor ce au aceeași structură sau attribute comune.

O colecție de obiecte sau de înregistrări poate fi organizată pentru o căutare eficientă utilizând un sistem de indexare cu o sigură cheie. Cele mai des întâlnite sisteme de indexare cu o singură cheie sunt sistemele **B Trees**.

Sistemul de indexare B Trees - fișierele ce sunt sortate folosind un câmp de tip cheie sunt în mod obișnuit indexate astfel încât să permită o creștere dinamică a bazei de date, posibilități de actualizare și o eficientă căutare. Un arbore **B Trees** este un arbore echilibrat care în fiecare nod are între $k+1$, și $2k+1$ pointeri și între k și $2k$ chei în fiecare nod. Nodul unui astfel de arbore este format din secvențe alternative de *<Pointeri de pagină>* *<Valori ale cheii>* [98]. Parametrul k depinde de dimensiunea paginii și de dimensiunea cheii, iar acestea se aleg în funcție de aplicație.

Pentru o secvență $P_1, K_1, P_2, K_2, \dots, P_i$, unde P_j este un pointer la o pagină iar K_i este cheia, algoritmul de căutare **B Trees** funcționează astfel:

- P_1 va fi pointerul pentru o pagină care are toate cheile mai mici sau egale cu K_1 .
- P_2 va fi pointerul pentru o pagină care are toate cheile mai mari decât K_1 și mai mici sau egale cu K_2 .
- Și așa mai departe...

Căutarea, inserarea și ștergerea în cazul algoritmului de indexare de tip **B Trees**, au fost foarte bine tratate în literatură, în acest moment existând mai multe variante pentru acest algoritm [99], [100], [101]. Varianta **B+** folosește stocarea valorii cheilor în cadrul frunzelor arborelui. Dacă arborele **B** are N chei, atunci lungimea maximă a căii de la rădăcină la frunze este $\log_k N$, lungimea timpului de căutare având o expresie logaritmică în funcție de mărimea bazei de date. Se poate folosi și o variantă de indexare cu indici secundari, frunzele arborelui în acest caz conținând perechea *<Valoarea cheii, Identificator obiect>*. Rădăcina și nivelele intermediare sunt folosite pentru localizarea paginii (frunză) ce conține o valoare specifică pentru cheie. Iar apoi identificatorul obiectului va identifica înregistrarea ce are valoarea specificată de valoarea cheii.

HSF (Hashed File Structures) - o strategie alternativă pentru varianta prezentată mai sus este schema de transformare cheie-adresă. Funcțiile de mapare ale spațiului cheilor $h(K)$, în spațiul de valori al tabelelor HSF, au în general valori finite, fiind propuse în literatură un număr foarte mare de variante (metode ce folosesc gruparea valorilor existente într-un tabel HSF, metode ce folosesc structuri de tip HSF doar pentru partea de indexare secundară). Considerentele principale de folosire a acestora sunt rezumate în posibilitatea de a converti o distribuție neuniformă a cheilor în una uniformă și posibilitatea de mapare a unui spațiu de chei de foarte mari dimensiuni într-un spațiu redus de adrese de stocare și un timp relativ redus de căutare a unei înregistrări.

4.4.2. Structuri de indexare pentru BLOBs

BLOBs - reprezintă date în format binar de foarte mari dimensiuni nestructurate. Având dat un BLOB, operațiile ce se efectuează la modul general cu acesta sunt: crearea unui nou BLOB, poziționarea în cadrul unui BLOB pentru scrierea sau citirea din acesta, inserarea și extragerea unei secvențe de biți în cadrul unui BLOB, extragerea proprietăților unui BLOB.

Liste de legătură - o variantă folosită frecvent pentru simplitatea ei, este definirea indexării în cadrul unui BLOB prin intermediul unei liste de legături. Astfel, un BLOB este împărțit în mai multe pagini ce pot fi accesate direct. Operații de inserare, ștergere sau înlocuire de pagini din cadrul unui BLOB sunt facil de realizat, operația de refacere a listei de legături

fiind foarte simplă. Principalul dezavantaj al unei asemenea scheme de indexare constă în faptul că nu toate accesările sunt necesare chiar de la începutul unei pagini, și sunt descărcate de exemplu de pe un server cantități enorme de date fără utilitate, de aceea în sistemele cu foarte multe accesări se poate întâmpla să apară și blocaje. De aceea au fost necesare scheme mult mai complexe de indexare și accesare a BLOBs.

Directoare pentru BLOBs de mărime impusă - majoritatea implementărilor presupun o dimensiune maximă pentru BLOBs, iar uneori dimensiunea acestuia poate fi de câțiva gigaocteți. Se preferă definirea unui director de o dimensiune fixă, iar fiecare BLOB are mai multe segmente (pagini) ce pot fi accesați prin intermediul unor pointeri. Un exemplu este sugestiv în acest caz: consider dimensiunea directorului de 2Gb (valoare des întâlnită ca restricție a diverselor sisteme de operare sau aplicații ce manipulează asemenea date). Dacă se consideră dimensiunea unui segment de 10kb, iar pentru fiecare segment se folosesc pointeri pe 8 biți, vor fi deci necesari 1600kb pentru gestionarea acestui BLOB. Dimensiunea segmentelor poate fi redusă, iar spațiul ocupat de pointeri pentru gestionare poate crește, mărindu-se și complexitatea procesului de gestionare a BLOB, dar se obține și o mai bună acuratețe în poziționare iar în cazul serverelor cu obiecte multimedia se reduce considerabil riscul de blocare și debitul de informație inutilă accesată. O alternativă de creștere a performanțelor este combinarea conceptului de directoare de mărime impusă cu cel de listă de legături prin realizarea unei liste care să conțină intrarea în directoare.

Scheme de indexare prin poziționare folosind arbori de tip *B Trees* - schema folosește o structură de bază a unui arbore de tip *B Trees*, indexarea făcându-se după poziție în BLOB și nu după chei.

Similar cu indexarea *B Trees*, fiecare nod sau rădăcina sistemului de indexare va conține perechea: <Adresa paginii, Contor octeți>. Toți octeții conținuți de BLOB ce au poziția: mai mică decât valoarea indicată de *Contor octeți* din perechea curentă și mai mare decât valoarea indicată de *Contor octeți* din perechea anterioară sunt conținuți în pagina a cărei referință este indicată de *Adresa paginii*.

Au fost dezvoltate variante care să dezvolte performanțele sistemului, una din variante folosește o structură pe trei nivele. Astfel se poate ajunge foarte ușor la factori de umplere de 67% pentru pagini [67-legat de cele 3 de sus].

4.4.3. Indexarea multidimensională pentru obiectele multimedia

Există foarte multe aplicații multimedia unde căutarea se face după mai multe atribute. De exemplu, pentru imagini sunt mai multe atribute care caracterizează aspectul general (dimensiune, număr de culori, tip compresie, etc.) și număr foarte mare de atribute care caracterizează conținutul imaginii (texturi, obiecte, forme, etc.). În aceste condiții, sistemele de indexare unidimensionale nu sunt suficiente.

Fie următorul exemplu, de exemplificare a avantajelor oferite de indexarea multicheie: se presupune că operația de căutare necesită N atribute și că avem o cheie secundară de indexare pentru fiecare dintre aceștia. Dacă se realizează o căutare exactă pentru toate cele N atribute, vom avea:

- Pentru fiecare atribut A_i , se caută indexul secundar și se identifică identificatorii obiectelor $OID_i = \{O_i, \dots, O_m\}$ pentru toate obiectele calificabile în urma procesului de căutare după un anumit atribut.
- Intersectarea tuturor mulțimilor $OID_1, OID_2, \dots, OID_n$ va determina o mulțime $OID_{rezultate}$ ce conține identificatorii (adresele) pentru toate obiectele ce îndeplinesc aceste condiții.
- Se obțin obiectele pornind de la adresele identificate în pasul anterior.

Concluziile ce se pot extrage de la acest mod indexare sunt următoarele:

- Timpul de căutare, în cazul în care se utilizează mai multe atribute și o indexare după o singură cheie, devine foarte mare.
- Folosirea unui sistem multicheie poate crește performanțele căutării multiatribut.
- Spațiul multidimensional al atributelor (dacă avem k – atribute, vom avea un spațiu cu k dimensiuni) se împarte în spații mai mici k – dimensionale.

Dimensionalitatea și localizarea - o caracteristică fundamentală care se regăsește pentru toate structurile de date uni sau multidimensionale este dată de necesitatea acestora de a se adapta la numărul variabil de obiecte și chei. Trebuie să se adapteze la modificările valorilor cheilor și ștergerii obiectelor care determină și ștergerea valorii cheilor din structură. Când se adaugă valori cheilor și obiecte, spațiul (unidimensional pentru structuri cu o singură cheie și multidimensional pentru structuri cu mai multe chei) necesită o partiționare. Pentru această partiționare a spațiului de căutare, structurile multidimensionale de acces au două caracteristici suplimentare: *dimensionalitatea* și *localizarea*. Dimensionalitatea apare atunci când partiționarea spațiului se face prin intermediul unei linii, plan, unul sau mai multe hiperplane într-un spațiu multidimensional. Localizarea apare atunci când un subspațiu este partiționat prin împărțirea în subspații multidimensionale disjuncte.

Structurile cu orientare matricială sunt mult mai flexibile decât variantele structuri de tip arbore. Dezvoltarea acestora în ultimul timp a fost exponențială. Evoluția sistemelor actuale, au avut ca punct de start, structuri ca: *point-quad trees* propus de Finkel și Bentley în 1974 [102], *k-d trees* propus de Bentley în 1975 [103], versiunea extinsă prin paginare a algoritmului anterior a fost propusă de Robinson în 1981 [104]. Versiunea multidimensională a arborelui B a fost dată de către Scheuermann și Ouksel în 1982 [105], iar arborele R de către Guttman în 1984 [106]. Alte soluții apărute pentru indexarea multidimensională sunt date de Liou și Yao în 1977 prin folosirea unui sistem multidimensional de directoare [107], iar noi formate multidimensionale pentru arbori au fost propuse în 1980 - *Quintary trees* de către Lee și Wong [108], iar în 1982 Orenstein a propus o nouă variantă multidimensională [109].

Multikey Hashing este o structură de indexare multidimensională asociativă ce oferă o foarte bună dinamică în cazul căutărilor cu o mapare exactă, au fost propuse mai multe variante ce s-au dezvoltat în timp. Primele variante au fost propuse în 1974 de către Rohnie și Lozano [110] și dezvoltate de Lloyd și Romamohanarao în 1982 [111] și de către Otoo în 1984 [112].

Indexarea pentru căutarea bazată pe conținut - într-o bază de date multimedia nu sunt suficiente numai atribute (cheile) pentru căutare, ci este nevoie de o căutare bazată pe conținut mult mai complexă. Sistemele de indexare bazate pe structuri multidimensionale se dovedesc a fi utile, dar nu acoperă întreaga arie de aplicații.

Scopul unui sistem de indexare este de a îmbunătăți timpul de răspuns al sistemului, atunci când are loc o interogare a bazei de date. Deoarece fiecare tip de date, poate avea o structură aparte, este relativ dificil de a standardiza metodele utilizate. Se preferă utilizarea unei structuri comune, care să fie personalizată pentru fiecare tip de date în parte.

Dintre metodele secundare (acestea se introduc pentru creșterea eficienței la căutare, pe lângă variantele obișnuite de indexare) folosite pentru indexare pot aminti: metoda inversării indexilor și metoda ce folosește semnătura de indexare. Metoda de indexare, ce folosește inversarea indexilor, este o metodă secundară de stocare, reprezentată de un form, ce asociază o mulțime de obiecte cu un cuvânt cheie. Semnătura de indexare este o tehnică alternativă pentru indexarea conținutului și se folosește de construcția unei semnături pentru fiecare obiect indexat [113].

Gruparea - există două tehnici principale de îmbunătățire a performanțelor de căutare pentru un sistem de stocare a informațiilor de tip bază de date:

- Indexarea – am prezentat în paragrafele anterioare pe scurt câteva metode ce se folosesc, avantajele și dezavantajele acestora.
- Gruparea – *clustering* – se folosește pentru organizarea obiectelor compozite, complexe, în segmente de stocare continue, cu optimizarea accesului la subcomponente și componente.

Gruparea trebuie să permită ca obiectele cele mai des accesate, să fie înregistrate cât mai aproape unele de celelalte, pentru a se reduce timpul de acces. Aplicațiile multimedia tipice, ce folosesc ierarhii de obiecte sau obiecte complexe, sunt:

- Aplicațiile CAD;
- Aplicațiile de grafică pe mai multe straturi și grafica vectorială;
- Site-urile Web;
- Aplicații ce rulează de pe CD-ROM;
- Structuri video ierarhice.

Sunt mai multe strategii ce permit gruparea obiectelor. În genera, spațiul obiectelor dintr-o bază de date este controlat prin intermediul unui graf. Un spațiu gestionat de un graf trebuie mapat pe modul de stocare fizic. Unitățile de stocare în general sunt *paginate* iar fiecare pagină poate conține mai multe *segmente*. Gruparea poate fi realizată:

- static – pe baza tipului sau structurii obiectelor;
- dinamic – programatorul poate specifica în care segment să fie realizată înregistrarea. De exemplu, programatorul poate specifica că obiectul O_1 trebuie plasat cât mai aproape de obiectul O_2 . Dacă dezvoltatorul de aplicații cunoaște modul de accesare al obiectelor, atunci, este recomandabilă utilizarea variantei dinamice, deoarece poate îmbunătăți enorm performanțele de căutare (se reduce considerabil de mult timpul de accesare).

Considerente ce impun modul de grupare al obiectelor:

- Dimensiunea obiectelor - una din problemele ce apare în cazul bazelor de date, este varietatea dimensiunii obiectelor ce sunt stocate. Tipic sistemele de management al bazelor de date impun dimensiuni minime și maxime pentru obiecte. Folosirea conceptului de obiect complex, format din mai multe subobiecte poate ajuta în multe cazuri la eliminarea limitărilor de dimensiune.
- Există aplicații unde numărul de elemente este imens, o aplicație în domeniul fizicii unde se realizează de exemplu 10 măsurători pe minut, în timpul câtorva ani, numărul obținut este foarte mare.

4.4.4. Strategii alternative de stocare

Metoda de stocare directă presupune respectarea structurii obiectului complex cât mai fidel. Sub-obiectele sunt memorate în acest caz cât mai aproape de obiectul părinte. Cel mai important pas este descompunerea obiectului complex părinte și stocarea subcomponentelor împreună. Avantajul acestui model constă în eficiența de căutare a obiectelor complexe (este necesară și stocarea modului de compoziție a obiectului). Pot apărea probleme, dacă există sub-obiecte partajate de mai multe obiecte părinte și se folosește o schemă de indexare de tip arbore. Eliminarea acestor probleme se face folosind o schemă de indexare de tip graf ce permite partajarea sub-obiectelor (figura 5-6).

Modelul de stocare prin decompoziție pe componente se bazează pe construirea de perechi $\langle \text{Identificator obiect}, \text{Atribut obiect} \rangle$, pentru fiecare atribut în parte. În acest mod, se realizează o rapidă sortare, iar în plus căutarea după un anumit atribut este foarte rapidă.

Avantajul principal al acestui model este simplitatea și generalitatea, fiind permise accesări rapide ale obiectelor complexe. Dezavantajul constă în modalitatea relativ greoaie de reconstrucție a obiectelor complexe, ele fiind înregistrate pe componente (figura 5-7).

Modelul normalizat de stocare este similar cu maparea spațiului obiectelor complexe în tabelele normale ale unei baze de date relaționale [114]. Vor rezulta date omogene, caracteristice pentru fiecare tip de înregistrare. Relațiile dintre înregistrări pot fi *one-to-one*, *one-to-many*, *many-to-many*. Strategia de stocare a modelului normalizat constă în reprezentarea datelor din baza de date sub forma unor mulțimi de înregistrări de același tip. Fiecare clasă va avea o mulțime de înregistrări ce reprezintă instanțele acelei clase (figura 5-8).

Unul din avantajele modelului normalizat este accesul facil la diversele tipurile de obiecte ce sunt conținute de un obiect părinte, complex. Pentru instanțe, obiectele au referință în directe și pot fi accesate, iar obiectele din cadrul unui director pot fi interogate separat. Se oferă, de asemenea, un suport mai puțin complex, pentru relațiile dintre obiecte de tip *many-to-many*.

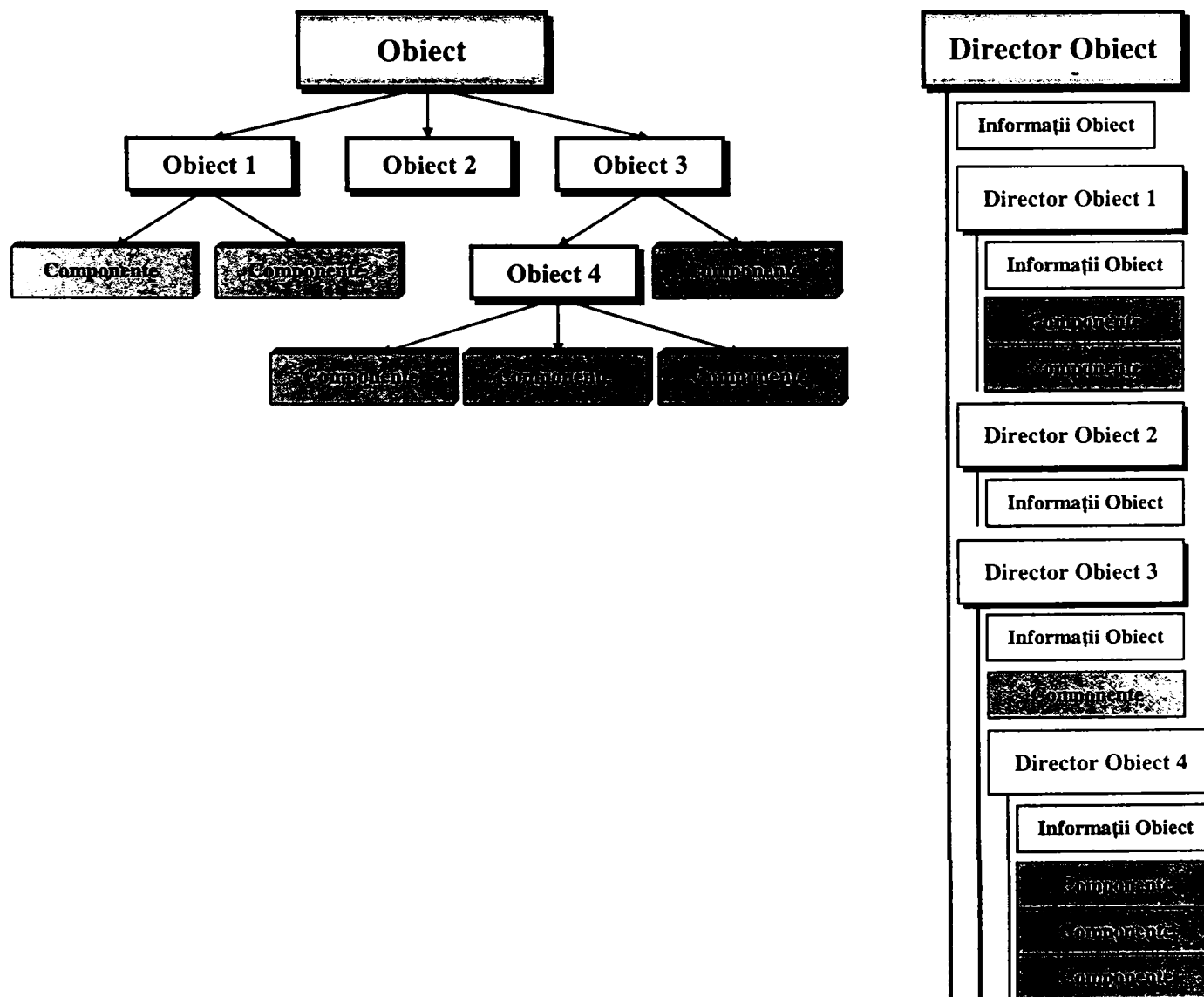


Figura 4-6 Variantă simplificată de prezentare a modului de stocare direct



Figura 4-7 Variantă simplificată de prezentare a modului de stocare cu descompoziție pe componente

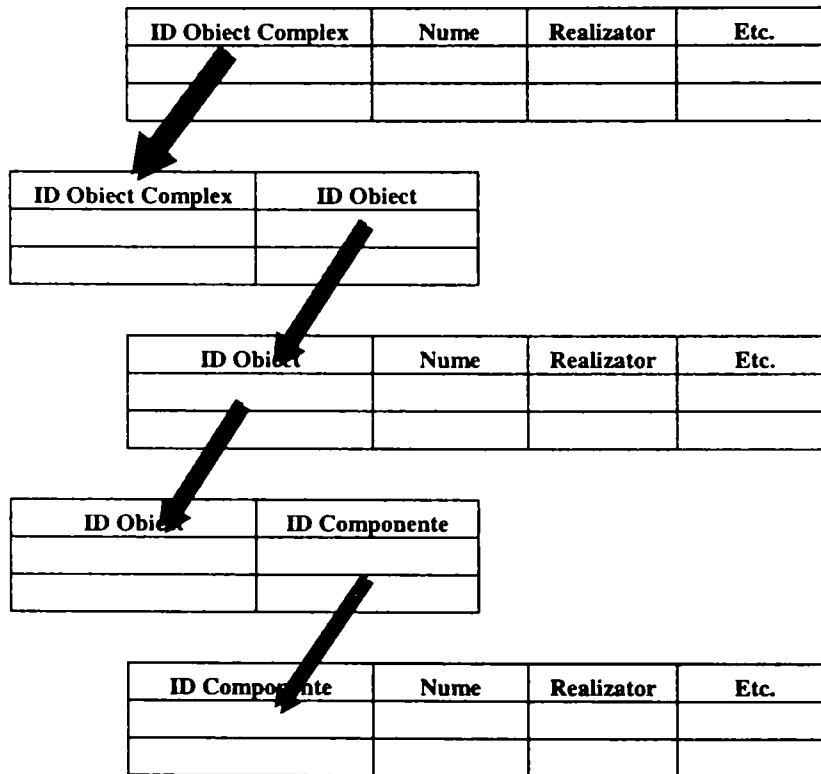


Figura 4-8 Variantă simplificată de prezentare a modului de stocare normalizat

4.4.5. Distanțe

Fiind dată o colecție de N obiecte: O_1, O_2, \dots, O_N , distanța (similaritatea) dintre două obiecte (O_i, O_j) este reprezentată de o funcție dată $D(O_i, O_j)$. Utilizatorii trebuie să specifice o interogare a bazei de date, adică un obiect Q , și o toleranță ε care este acceptată în raport cu acesta.

Scopul este de a găsi toate obiectele dintr-o bază de date, ce se află la o distanță mai mică decât toleranța dată ε față de obiectul de interogare. Soluția este de a scana întreaga bază de date iar pentru fiecare obiect se va calcula distanța acestuia față de obiectul de interogare. În cazul bazelor de date de foarte mari dimensiuni, este impractic să se calculeze pentru toate obiectele distanța față de obiectul de interogare și de aceea se recurge la folosirea unor condiții care să reducă cât mai mult posibil numărul elementelor candidate, calculul distanțelor realizându-se doar pentru această mulțime redusă.

În cadrul capitolului 6, am prezentat o detaliere a celor mai folosite distanțe.

4.4.6. Concluzii

Acest capitol a prezentat metode de accesare a informațiilor într-o bază de date prin intermediul mai multor tehnici de indexare și de grupare a informațiilor. Scopul acestora este de a crește performanțele de căutare ale obiectelor înregistrate. Tehnicile prezentate folosesc sisteme uni sau multi-cheie pentru indexare, având unul sau mai multe atribute folosite pentru căutare. Metodele de grupare constituie un mod de mărirea a performanțelor unei baze de date prin eficientizarea accesului la informație, indiferent dacă este vorba de baze de date relaționale sau orientate pe obiecte.

4.5. Căutarea bazată pe conținut în bazele de date multimedia

4.5.1. Introducere

Sistemul de management al unei baze de date multimedia orientată pe obiecte efectuează în general operații de stocare, manipulare a datelor și căutare a datelor existente. Unul dintre cele mai importante servicii ce trebuie să îl asigure este mecanismul de căutare a informației solicitate de un utilizator.

Partea de căutare bazată pe cuvinte cheie este la ora actuală foarte bine dezvoltată de majoritatea bazelor de date. Rezultate promițătoare sunt și în domeniul căutării bazată pe conținut a imaginilor și chiar în cazul secvențelor video. Problemele cele mai mari, apar atunci când avem obiecte complexe, formate din structurarea ierarhică sau conglomerarea de mai multe obiecte simple. Acesta este cazul obiectelor multimedia, unde pentru a efectua o căutare bazată pe conținut, trebuie în primul rând explicitată în baza de date modalitatea de formare a obiectului multimedia complex. Apoi folosind modul de organizare a unui asemenea obiect complex se pot identifica tipurile de obiecte componente și se poate realiza o căutare a unui element primar.

În general, un obiect multimedia poate fi văzut ca o colecție de secvențe de biți de mari dimensiuni (BLOBs), pentru acestea sistemul de management al bazei de date oferă un suport special de operare cu asemenea date (citire, inserare, ștergere, modificare, etc.). Interogarea și căutarea într-o secvență lungă de biți, fără o interpretare semantică, se rezumă doar la operații de mapare, ceea ce face practic imposibilă găsirea unui element dintr-un obiect multimedia fără a avea și informația de structurare.

În concluzie, se poate spune că sistemul de management al bazei de date trebuie să poată lucra cu structura internă a obiectului multimedia, cu partea de logică de formare a acestuia și deci să facă apel la partea semantică a unui BLOB. Indexarea (împărțirea) unui BLOB după modelul semantic al părților constituente ale obiectului multimedia va permite realizarea unei căutări bazate pe conținut.

Căutarea obiectelor multimedia se realizează în primul rând pe baza conținutului semantic al acestora. Conținutul semantic poate fi introdus manual, prin introducerea de texte descriptive sau poate fi extras automat după o analiză semantică (se introduc pentru început descrieri ale domeniului pentru acel obiect). De exemplu, indexarea automată a unei emisiuni de știri poate avea ca descriere semantică modul de organizare al emisiunii: generic emisiune + prezentatoare + știre + prezentatoare + + generic știri externe + prezentatoare + știre +... + prezentatoare + generic final (se poate completa modelul cu descrierea unor limite temporale pentru fiecare submodul al emisiunii).

Utilizatorul trebuie să fie capabil să interogheze după conținut baza de date. Această interogare se face, în general, prin specificarea:

- Valorilor atributelor semantice ale obiectului multimedia;

- Cuvintelor sau frazelor care conțin descrieri textuale ale modelului semantic al obiectului multimedia;
- Caracteristicilor globale ale obiectului multimedia;
- Proprietăților vizuale și relațiilor spațiale ce apar între elementele primare conceptuale ale unui obiect multimedia;
- Proprietăților în limbajul curent utilizat de majoritatea utilizatorilor și interacțiunilor ce apar între elementele primare conceptuale ale unui obiect multimedia;
- Comportării temporale a elementele primare conceptuale ale unui obiect multimedia.

În concluzie, se poate spune că interogarea se adresează atât conținutului, cât și structurii obiectelor multimedia.

4.5.2. Modelarea obiectelor multimedia și interogarea bazată pe similaritate

După cum am prezentat detaliat în capitolul 4 modelul de aplicație multimedia, din punct de vedere al bazelor de date, se poate concluziona:

- Fiecare obiect multimedia este compus dintr-un număr de sub-obiecte simple sau complexe, care sunt organizate secvențial, paralel sau în mod mixt (secvențial și paralel).
- Fiecare obiect multimedia are
 - o structură logică;
 - o valoare asociată componentelor obiectelor complexe;
 - o interpretare a conținutului pentru acea valoare.
- Valoarea unui obiect multimedia, structura sa logică și modelul de reprezentare și stocare a acestuia se află exprimată în modelul de descriere al obiectului multimedia.
- Modelul de descriere multimedia conține informații referitoare la compoziția logică a acestui obiect, constrângeri de sincronizare și temporizare pentru componentele acestuia, parametrii necesari pentru afișare, etc.
- Modelul de interpretare trebuie să descrie cât mai exact lumea reală prin prisma sub-obiectelor ce îl compun și prin relațiile ce există între acestea, respectiv, între fiecare sub-obiect component și lumea reală.

Folosirea și a modelului de interpretare poate ajuta foarte mult în cadrul bazelor de date multimedia pentru individualizarea obiectelor:

- de exemplu, individualizarea unui obiect multimedia poate fi dată de un set de relații ce există între subcomponentele acestuia: o imagine și un text, într-o pagină de prezentare multimedia poate descrie un produs. În acest caz, interpretarea multimedia a „obiectului pagină” este reprezentată de descrierea produsului.
- un cuvânt scris (deci, sub formă de obiect de tip text) sau sub formă de sunet poate fi interpretat ca atare sau poate reprezenta de exemplu numele unei companii.

Numărul unor astfel de exemple poate fi foarte mare. Recunoașterea conceptuală a subobiectelor nu este întotdeauna sigură. De exemplu, într-o imagine compusă din mai multe obiecte se poate întâmpla ca unul din ele să acopere o parte din celelalte obiecte, iar identificarea acestora să fie foarte dificilă. În schimb, pentru a face obiectele din aceeași imagine identificabile (compusă din mai multe obiecte, care privite în ansamblu nu pot fi identificate), putem lucra cu o imagine compusă din mai multe nivele (dar nu din obiecte integrate pe același nivel – „layer”), se alocă deci, pentru fiecare obiect un layer nou, iar prin compunerea acestora se va obține aceeași imagine. Procesul de căutare se va desfășura de această dată pe staturi – layere, la nivel de sub-obiect multimedia.

Atributele interpretării unui obiect multimedia, obiect ce este o instanță a unei clase, includ:

- Un pointer care să definească obiectul multimedia în cadrul unei înregistrări de tip BLOB sau o porțiune din acesta, adică numai locul unde este stocat obiectul;
- Un pointer care să clasifice conceptual obiectul;
- Proprietăți globale necesare pentru o căutare de mapare exactă, pentru segmentări sau alte operații de acest tip;
- Proprietăți globale necesare pentru căutări bazate pe similaritate (descrierea conținutului sau histogramele descompunerii pe componente de culoare în cazul imaginilor);
- Un pointer spre fiecare obiect conceptual ce apare în cadrul obiectului multimedia primar, adică un pointer care să corespundă interpretării multimedia a fiecărui sub-obiect ce compune obiectul analizat (De exemplu, un nume descriptiv din punct de vedere al interpretării pentru fiecare componentă a obiectului multimedia, sau o listă de proprietăți dinamice conceptuale ale obiectului – lista este dinamică, deoarece același obiect folosit în diverse situații, poate avea diverse interpretări).
- Interacțiunile dintre descrierile interpretative ale subcomponentelor ce formează obiectul multimedia analizat. Un obiect format din mai multe subcomponente, fiecare cu propria interpretare, prin introducerea unei interacțiuni între interpretarea sub-obiectelor, poate genera un alt obiect din punct de vedere al interpretării. Cele mai dese interacțiuni sunt cele de cauzalitate.
- Un pointer care să permită legătura cu *super-obiectele*. Adică, să existe o legătură și între obiectul analizat și obiectul în a cărei structură acesta este regăsit ca și un sub-obiect (legătură cu obiectul părinte).

În concluzie, se poate spune că, identificarea unui obiect va depinde deci de calcularea unei distanțe minime între parametrii descriptivi ai obiectului dar și între partea de interpretare a obiectelor. Partea cea mai delicată și cea care depinde în cea mai mare măsură de utilizator, este definirea ponderii părții interpretativ-semantice în procesul de căutare.

4.5.3. Căutarea bazată pe conținut – abordare din punct de vedere al similarității modului de interpretare a unui obiect multimedia

Pot exista două variante de căutare în cazul obiectelor multimedia prin prisma modului de interpretare a acestora: căutare exactă a unui cuvânt, fraze, etc. sau o căutare bazată pe similaritatea descrierii.

Căutarea în cadrul modelului interpretativ al unui obiect multimedia poate fi realizată folosind cuvinte cheie (acestea se introduc manual sau pot fi generate automat) sau prin căutarea unui anumit text.

Căutarea după cuvinte cheie este foarte simplă deoarece se bazează pe a găsi un anumit cuvânt cheie într-o listă. Singurele probleme ce apar țin de alegerea cuvintelor cheie care descriu cel mai bine conținutul unui document și de faptul că nu întotdeauna cuvintele cheie pot reflecta exact conținutul documentului.

Căutarea exactă a unui text presupune efectiv căutarea în documentul respectiv dacă există acel text. În general, se mai folosesc în procesul de interogare diverse operații booleene între cuvintele folosite pentru interogare (SI, SAU, etc.) [115].

Căutarea bazată pe similaritate, va avea ca rezultat, o serie de documente ce conțin elementele căutate, dar care pot fi similare și din punct de vedere conceptual. În acest caz se poate aplica conceptul de similaritate a interpretării, similaritate a modelului semantic. În foarte

multe situații, folosirea unui model de obiect multimedia ce are în componența sa și elemente ce definesc interpretarea acestuia, aduce un surplus de eficiență algoritmilor de căutare în bazele de date.

Eficiența căutării se poate evalua folosind relațiile:

$$R = \frac{\text{Elemente relevante găsite}}{\text{Numarul de elemente relevante}}$$

$$P = \frac{\text{Elemente relevante găsite}}{\text{Numarul de elemente găsite}}$$

Pentru similaritate, de foarte multe ori, se folosesc vectori care să definească legătura dintre ceea ce a cerut utilizatorul și ceea ce se găsește în cadrul bazei de date [116]. Similaritatea dintre două documente sau dintre un document și o condiție de interogare este dată în modelul de procesare bazat pe vectori de relația:

$$Sim_{icr}(Q, D_i) = \sum_{k=1}^l w_{qk} \cdot w_{ik}$$

unde,

$Q = (w_{q1}, \dots, w_{qt})$ este vectorul de interogare;

$D_i = (w_{i1}, \dots, w_{it})$ este vectorul ce caracterizează documentul i ;

$$w_{ik} = \frac{tf_{ik} \times \log\left(\frac{N}{n_k}\right)}{\sqrt{\sum_{k=1}^l (tf_{ik})^2 \times \left(\log\left(\frac{N}{n_k}\right)\right)^2}}$$

tf_{ik} este frecvența de apariție a termenului T_k în documentul D_i ;

N este numărul de documente din colecție;

n_k este numărul de documente din baza de date ce conțin termenul T_k .

Hearst și Plaunt au propus următoarea variantă îmbunătățită pentru calculul similarității între două blocuri de document [117]:

$$Sim_{Bloc}(b_1, b_2) = \frac{\sum_{k=1}^n w_{b_1k} \times w_{b_2k}}{\sqrt{\left(\sum_{k=1}^n (w_{b_1k})^2\right) \times \left(\sum_{k=1}^n (w_{b_2k})^2\right)}}$$

unde, b_1, b_2 reprezintă blocurile unui document.

Conceptul de căutare bazat pe similaritate a fost dezvoltat în continuare prin introducerea de noi facilități, prin dezvoltarea vitezei și a performanțelor de regăsire, inclusiv pentru lucrul cu informații legate de interpretare. Dezvoltarea acestor algoritmi este benefică pentru toate tipurile de date din cadrul bazelor de date multimedia. Prin integrarea conceptului de descriere a obiectelor multimedia și din punct de vedere al interpretării semantice a informație, un algoritm performant de căutare trebuie să includă astfel de concepte. Descrierea semantică, în general, se face prin intermediul textului, deci implementarea unor astfel de algoritmi de căutare bazați pe similaritatea textului sunt necesari în toate domeniile [118].

4.5.4. Concluzii

Optimizarea operației de căutare este în strânsă legătură de etapele anterioare acestei operații: de modul de indexare a bazei de date și de modul de interogare al bazei [119].

Cea mai importantă proprietate a unei baze de date multimedia, din punct de vedere al utilizatorului, este facilitatea de căutare după conținut. În acest scop, este necesară extragerea unor caracteristici, care să permită discriminarea obiectelor, preferabil cu costuri de calcul cât mai reduse.

În cazul integrării de mai multe caracteristici în cadrul unei distanțe, trebuie avută în vedere alegerea caracteristicilor obiectelor, astfel încât, o foarte bună caracteristică discriminatorie, să nu fie mascată de celelalte caracteristici.

Este recomandabilă căutarea interactivă, bazată pe mai multe etape de restrângere a numărului de candidați cu o redefinire a parametrilor de căutare, ceea ce duce la reducerea sensibilă a timpului de căutare și creșterea eficienței căutării.

4.6. Interogarea bazelor de date multimedia

4.6.1. Introducere

Dezvoltarea unui bun sistem de interogare implică o cantitate mare de muncă inclusiv pentru sistemele clasice de baze de date. Procesul de interogare în cazul obiectelor multimedia este însă mult mai complex decât pentru sistemele obișnuite ce folosesc date ce au un format bine definit. Pentru obiectele multimedia, care sunt formate din mai multe sub-componente și dacă au integrat sistemul de interpretare, procesul se complică foarte mult. Problema principală este eterogenitatea datelor procesate (date cu diverse caracteristici, caracterizate de diverse tehnici de stocare și diverse structuri de acces). Se poate spune încă de la început că procesul de interogare trebuie să combine tehnicile caracteristice pentru fiecare tip de date într-un mod cât mai eficient.

În general, orice metodă de procesare multimedia constă în alegerea unor elemente din variantele de bază ce țin cont de:

- metodele de stocare și de accesare folosite;
- tipul de predicate folosite pentru interogare;
- utilizarea de elemente pasive sau active;
- căutarea exactă sau bazată pe similaritate;
- aproximările folosite pentru definirea structurilor de date sau aproximările de calcul;
- extensibilitatea gradului de procesare.

Procesul de interogare rămâne deci o mare problemă deschisă ce permite dezvoltări ulterioare.

4.6.2. Accesul și stocarea

Eficiențizarea procesului de interogare este strâns legată de:

- Metodele de stocare utilizate – modul în care un obiect multimedia este stocat poate influența foarte mult strategia procesului de interogare.
- Metodele de accesare – dacă se utilizează metode de stocare bine structurate pentru obiectele multimedia, aceleași date pot fi ușor găsite în urma procesului de căutare folosind câteva

clase de metode de interogare, acest lucru este posibil dacă se definesc structuri adecvate de accesare (metode de indexare optime).

Pentru bazele de date tradiționale, au fost introduse mai multe structuri ce se bazează pe un model dat de reprezentare a datelor. În general, aceste tipuri de date respectă în general un format bine stabilit. Pentru fiecare tip de date au fost implementate metode particulare, structurile de indexare folosesc atributele obiectului pentru calcularea adresei de stocare a acestuia.

Stocarea obiectelor multimedia diferă ca tehnologie de stocarea clasică, deoarece presupun stocarea de obiecte de mari dimensiuni (video, imagini, audio), obiecte ce au în structura lor alte sub-obiecte și pot exista puternice constrângeri temporale între acestea. În aceste situații este nevoie de sisteme de stocare de mare capacitate dar care să ofere și un debit deosebit, pentru rularea acestora direct din baza de date. De aceea, nu se recomandă folosirea de suporturi optice de stocare (acestea au în general timpi mari de acces și debit relativ redus) și se propun metode de stocare paralelă. De exemplu, poate exista cazul unei aplicații care folosește sunet, animații și secvențe video. Fiecare din aceste trei elemente poate avea un debit suficient de mare, iar redarea acestora simultan de pe un singur hardisk poate fi în multe cazuri imposibil de realizat. Se poate folosi împărțirea obiectului multimedia pe obiecte și stocarea acestora să se facă separat. Această soluție poate rezolva problema de debit, în schimb crește complexitatea algoritmului de indexare și a modului în care se poate realiza interogarea bazei de date [120].

Multe procese de interogare se referă doar la un număr redus de obiecte din baza de date, în general este inefficient pentru sistem să citească fiecare obiect și să verifice toate condițiile pentru acesta. Ideal ar fi ca sistemul să identifice cât mai rapid obiectele ce ar putea fi interesante.

Eficiența procesului de interogare poate fi crescută prin introducerea de date suplimentare în interiorul bazei de date care să structureze mai bine informațiile sub forma unor clase de funcții.

4.6.3. Dimensiunile procesului de interogare

Problema cea mai mare în cazul proceselor de interogare este eterogenitatea datelor. Procese care funcționează bine pentru un anumit tip de date nu pot fi folosite pentru alte tipuri de date. Se pot extrage câteva caracteristici generale, pentru procesul de interogare:

- Folosirea predicatelor de interogare;
- Folosirea de componente active și pasive în procesul de interogare;
- Folosirea tehnicilor de mapare parțială sau exactă în spațiul predicatelor;
- Folosirea de aproximări în procesul de căutare și redefinirea iterativă a predicatelor de căutare pentru obținerea de rezultate mai bune;

Caracteristica tehnologiilor multimedia în procesul de căutare este extensibilitatea. Procesele de căutare trebuie să permită dezvoltări permanente. De exemplu, prin introducerea (calculul), de noi caracteristici și atribute obiectelor stocate, este necesar ca puterea de interogare să crească, iar rezultatele obținute să se îmbunătățească. Aceste noi inserări de atribute nu trebuie însă să afecteze și să implice reprogramarea aplicației.

În sistemele multimedia, recomand ca interogarea să fie de două tipuri (detalii privind modul de implementare ale acestora în baza de date multimedia propusă sunt prezentate în capitolul 7): interogare bazată pe predicate și interogare bazată pe modele sau exemple.

Interogarea bazată pe predicate are o formă particulară pentru fiecare tip de date, dar sunt și două aspecte globale, deoarece se adresează:

- Modulul de accesare a datelor – predicatelor de interogare pot fi clasificate ca fiind structurale și semantice.
- Modulul de accesare a structurilor – procesul de interogare poate fi rezolvat pe baza unui mod de accesare a obiectelor ce folosește structuri de accesare.

După modul de accesare a datelor, predicatelor de interogare se clasifică în două clase: predicatelor structurale și predicatelor semantice.

Predicatul structurale se referă la structura datelor și pot fi folosite pentru:

- reprezentarea diverselor tipuri de medii. O imagine de exemplu poate fi descrisă de exemplu prin dimensiunea sa în pixeli, prin numărul de culori, prin modul de compresie, etc.;
- clasificarea datelor;
- reprezentarea unor tipuri de medii și a relațiilor dintre acestea;
- specificarea structurii logice a datelor;
- facilitarea localizării unui obiect;
- realizarea istoriei evoluției stărilor unui obiect.

Predicatul pot caracteriza obiecte individuale sau pot caracteriza grupuri de obiecte.

Predicatul semantice se referă la conținutul semantic al datelor. Aceste predicat se dovedesc foarte utile în cazul mediilor continue (sunet, video, animații). Clasificarea acestora se poate face în:

- predicat semantice statice;
- predicat semantice temporale.

Predicatul de interogare și accesare de structuri se referă la date ce au structuri particulare.

4.6.4. Concluzii

Procesul de interogare a bazelor de date multimedia nu este un lucru bine consolidat și poate suferi dezvoltări permanente în strânsă legătură cu modelele de obiecte multimedia și aplicațiile ce se dezvoltă permanent.

Implementarea unui bun sistem de interogare presupune anumite costuri de realizare a unui model și de estimare a performanțelor acestuia. Costurile de implementare depind de:

- Structura modelului implementat;
- Numărul de parametri folosiți în procesul de interogare și eficiența acestora la căutare;
- Modul de implementare și integrare a diverselor strategii de estimare și aproximare a parametrilor.

În modul de interogare al bazei de date multimedia, am folosit o combinație a metodelor exacte cu cele ce folosesc aproximații. Această combinație oferă utilizatorului un grad mare de libertate în procesul de interogare și în special când se realizează o interogare iterativă. Detaliile legate de modul de interogare propus sunt prezente în capitolul 6.

Capitolul 5.

Contribuții la indexarea, interogarea și căutarea imaginilor și a secvențelor video în baze de date multimedia, în cadrul studiourilor de televiziune

A. Indexarea, interogarea și căutarea imaginilor în baze de date multimedia

5.1. Caracteristici globale utilizabile la indexarea bazelor de imagini

Regiunea unei imagini poate fi definită ca un ansamblu de pixeli dintr-o imagine ce pot fi caracterizați de câteva atribute. Aceste atribute vor fi folosite în continuare pentru a realiza o clasificare a pixelilor dintr-o imagine după proprietățile conexiunilor ce există între aceștia.

5.1.1. Atributele stohastice ale unei imagini

În acest paragraf, vom considera imaginea ca un proces aleator discret $A[s]$ cu valori reale. Dacă s este fixat, atunci $A[s]$ este o variabilă aleatoare pentru care se pot defini, pentru diverse realizări [121], următoarele:

Media

Pentru un semnal continuu $A(x)$, speranța sau media ansamblului este definit prin:

$$\mu(x) = E\{A(x)\} = \int_{-\infty}^{\infty} a \cdot f(a, x) da \quad (5.1.)$$

Pentru un semnal discret media se definește în următorul mod:

$$\mu[s] = E\{A[s]\} = \sum_i p_i \cdot a_i \quad (5.2.)$$

unde a_i este o realizare a variabilei $A[s]$ și p_i este probabilitatea $P(A[s] = a_i)$

Momentul de ordinul k

$$E\{A^k[s]\} = \sum_i p_i \cdot a_i^k \quad (5.3.)$$

Momentul centrat de ordinul k

$$\mu_k[s] = E\{A[s] - \mu[s]^k\} \quad (5.4.)$$

Autocorelația

$$R[s_1, s_2] = E\{A[s_1] \cdot A[s_2]\} \quad (5.5.)$$

5.1.2. Momentele și media spațială a unei regiuni

Pentru o regiune R de K pixeli, se definesc diferite momente spațiale, centrate sau necentrate [122]:

Momentul spațial de ordinul I sau media

$$m_1 = \frac{1}{K} \sum_{s \in R} A[s] \quad (5.6.)$$

Momentul spațial centrat de ordinul doi sau varianța

$$m_2 = \frac{1}{K} \sum_{s \in R} (A[s] - m_1)^2 \quad (5.7.)$$

Momentul spațial centrat de ordinul k

$$m_k = \frac{1}{K} \sum_{s \in R} (A[s] - m_1)^k \quad (5.8.)$$

Autocovarianța sau momentul spațial compus de ordinul secund

$$M_2(t) = \frac{1}{N_c} \sum_{s \in R, s+t \in R} (A[s] - m_1) \cdot (A[s+t] - m_1) \quad (5.9.)$$

unde, N_c este numărul de cupluri $(s, s+t)$ din regiunea R . Autocovarianța este periodică dacă textura este periodică.

Momentul spațial compus de ordinul k

$$M_k(t_1, t_2, \dots, t_{k-1}) = \frac{1}{N_k} \sum_{s \in R, s+t_1 \in R, \dots, s+t_{k-1} \in R} (A[s] - m_1) \cdot (A[s+t_1] - m_1) \cdot (A[s+t_2] - m_1) \dots (A[s+t_{k-1}] - m_1) \quad (5.10.)$$

unde, N_k este numărul de cupluri $(s, s+t_1, \dots, s+t_{k-1})$ din regiunea R .

Procesul este staționar de ordinul I dacă $E\{A[s]\}$ este independent de site. În plus un proces este staționar de ordinul II dacă autocorelația nu depinde de translația t între două site.

$$R[s_1 \cdot s_2] = E\{A[s_1] \cdot A[s_2]\} = R[t] = E\{A[s] \cdot A[s+t]\} \quad (5.11.)$$

Când procesele sunt staționare, se pot exploata momentele spațiale, observând o realizare pe un ansamblu de site, o legătură va apare între diverse site pentru diverse realizări. În plus procesul este ergodic de ordinul doi când R tinde spre întreg planul N^2 , la:

$$m = \lim_{R \rightarrow N^2} \frac{1}{K} \sum_{s \in R} A[s] = E\{A[s]\} \quad (5.12.)$$

$$M_2(t) = \lim_{R \rightarrow N^2} \frac{1}{N_c} \sum_{s \in R, s+t \in R} A[s] \cdot A[s+t] = E\{A[s] \cdot A[s+t]\} \quad (5.13.)$$

Studiul celor două momente nu este suficient pentru a discrimina diverse texturi. A. Gagalowicz a reușit să sintetizeze texturi diferențiabile pentru ochi ce au momentele egale până la ordinul 4 [123].

Pentru $\varpi \in \Omega = \{0, \dots, L-1\}$ se definește:

$$h(\varpi) = \frac{1}{K} \sum_{s \in R} \delta(A[s] - \varpi) \quad (5.14.)$$

$$\text{unde, } \delta(A[s] - \varpi) = \begin{cases} 1, & \text{daca } A[s] = \varpi \\ 0, & \text{daca nu} \end{cases} \quad (5.15.)$$

Pentru un număr mare de realizări, se estimează $P(A(s)=\omega)$ prin $h(\omega)$. h nu este decât densitatea empirică de probabilitate a lui A .

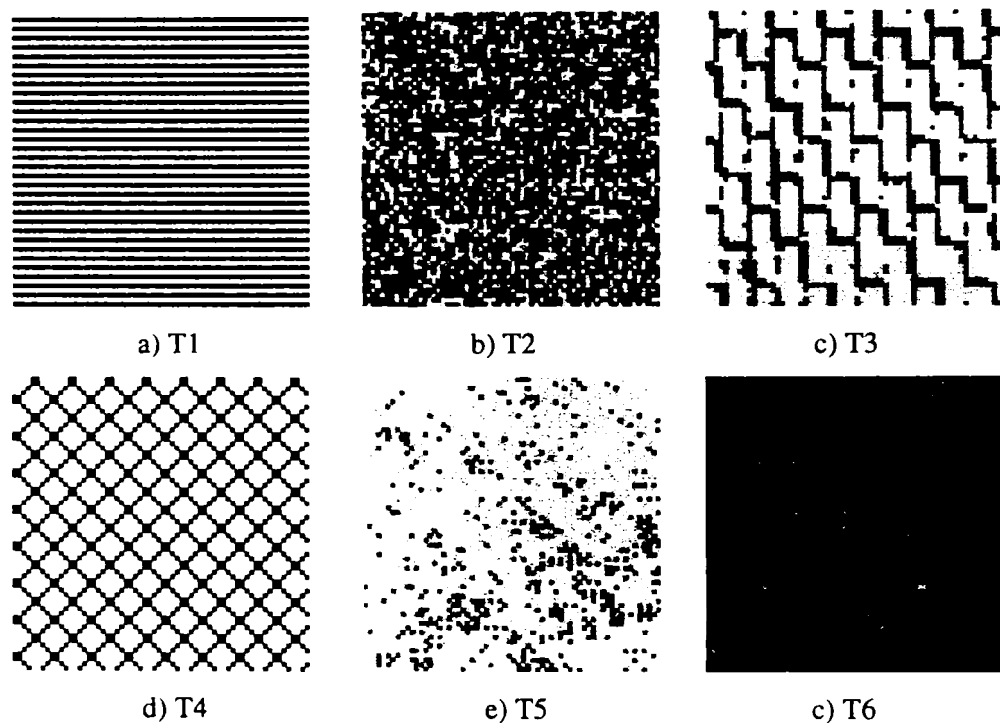


Figura 5-1 Texturile analizate

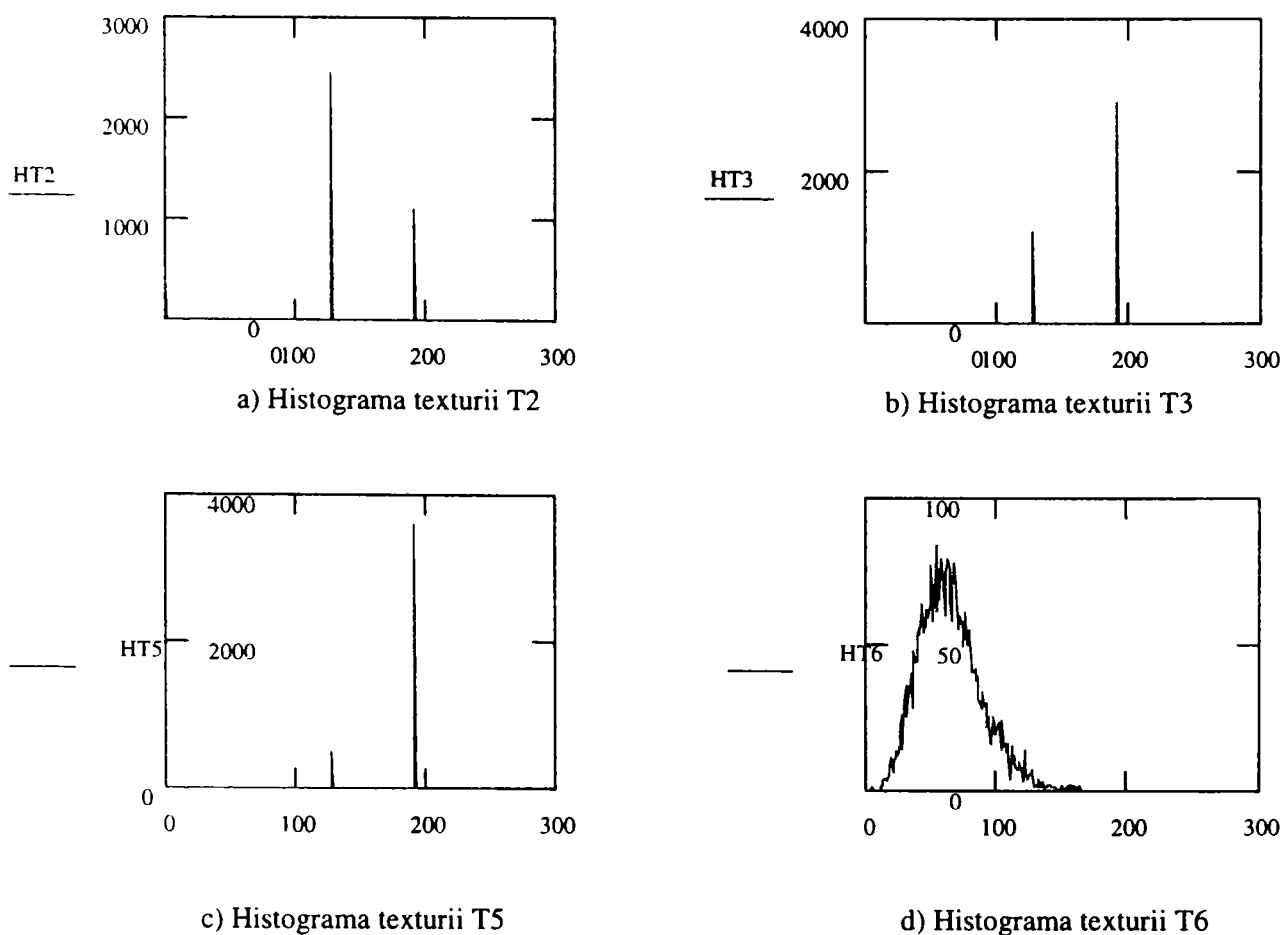


Figura 5-2 Histogramele texturilor analizate

Histograma de coocurență sau media spațială de ordinul doi

$$h_i(\varpi_i, \varpi_j) = \frac{1}{N_c} \sum_{s \in R} \delta(A[s] - \varpi_i) \cdot \delta(A[s+t] - \varpi_j) \quad (5.16.)$$

Ne permite să calculăm $P(A[s]=\omega_i, A[s+t]=\omega_j)$ care este coocurența unui pixel de nivelul de gri ω_i și un pixel de nivel de gri ω_j , separați de translația t .

Se poate introduce o generalizare a definițiilor precedente prin:

media spațială de ordinul k

$P(A[s]=\omega_1, A[s+t_1]=\omega_2, \dots, A[s+t_{k-1}]=\omega_k)$, pentru $k-1$ translații.

Autorul a efectuat experimentări pe imagini având diferite texturi, în conformitate cu figurile 5-1 și 5-2. Am constatat că sistemul vizual uman nu poate discrimina două texturi ce au aceeași distribuție de probabilitate de ordinul I și II. Pot exista contraexemple de texturi macroscopice ce permit discriminări datorită existenței unor elemente deterministe, texeli.

Acești parametri statistici au dezavantajul că sunt dificil de manipulat deoarece ocupă multă memorie și timp de calcul.

Pornind de la imaginea originală contrastul se definește în următorul mod:

$$\text{Contrastul} = \frac{\sigma}{\alpha_4^{1/4}}, \quad \alpha_4 = \frac{\mu_4}{\sigma^4} \quad (5.17.)$$

unde, σ este varianța, iar μ_4 este momentul de ordinul IV. Acesta poate fi calculat pentru întreaga imagine sau pe ferestre alunecătoare.

Granularitatea se bazează pe calculul mediei unor ferestre alunecătoare.

$$A_k(x, y) = \sum_{i=x-2^{k-1}}^{x+2^{k-1}-1} \sum_{j=y-2^{k-1}}^{y+2^{k-1}-1} \frac{\text{Img}[i,j]}{2^{2k}} \quad (5.18.)$$

unde, $\text{Img}[i,j]$ este valoarea nivelului de gri pentru pixelul de coordonate (i,j) ; k este între 0 și 5.

$$E_{k,h}(x, y) = |A_k(x + 2^{k-1}, y) - A_k(x - 2^{k-1}, y)| \quad (5.19.)$$

$$E_{k,v}(x, y) = |A_k(x, y + 2^{k-1}) - A_k(x, y - 2^{k-1})| \quad (5.20.)$$

Pentru fiecare pixel, se alege valoarea lui k care maximizează expresiile (5.19) și (5.20). Parametrul de optimizare va fi:

$$S(x, y) = 2^k \quad (5.21.)$$

Pentru o imagine de dimensiune $m \times n$, granularitatea se poate defini astfel:

$$G = \frac{1}{m \times n} \sum_{i=1}^m \sum_{j=1}^n S(x, y) \quad (5.22.)$$

Directivitatea este vectorul gradient calculat pentru fiecare pixel. Magnitudinea și orientarea acestui vector se definește în următorul mod:

$$|\Delta_D| = \frac{(|\Delta_v| + |\Delta_h|)}{2} \quad (5.23.)$$

$$\theta = \arctan\left(\frac{\Delta_v}{\Delta_h}\right) + \frac{\pi}{2} \quad (5.24.)$$

Diferențele Δ_V și Δ_H sunt calculate pe ferestre de dimensiunea 3 x 3 centrate pe fiecare pixel. Pentru toate valorile obținute pentru magnitudine și orientare se construiesc două histograme, pe baza cărora se evaluează directivitatea.

5.1.3. Matricea de coocurență

Matricea de coocurență și indicii de ordinul secund ai texturilor

Matricile de coocurență conțin mediile spațiale de ordin secund. Pentru o translație t , matricile de coocurență MC_t ale unei regiuni R sunt definite pentru toate cuplurile de nivele de gri (a, b) prin relația:

$$MC_t(a, b) = \text{card}\{(s, s+t) \in R^2 \mid A[s] = a, A[s+t] = b\} \quad (5.25.)$$

$MC_t(a, b)$ este deci numărul de cupluri de pixeli $(s, s+t)$ din regiunea considerată, separați prin vectorul de translație t .

Matricile de coocurență conțin o cantitate importantă de informație dar care este dificil de manipulat în întregime.

În general, se folosesc câțiva indici sau caracteristici ale imaginilor:

$$\text{Omogenitatea} : \frac{1}{N_c} \sum_a \sum_b (MC_t(a, b))^2 \quad (5.26.)$$

Acest indice, este cu atât mai mare, cu cât se regăsesc mai multe cupluri de pixeli de același nivel sau când nivelele de gri sunt constante sau când se descoperă o periodicitate în direcția translației.

$$\text{Contrastul} : \frac{1}{N_c(L-1)^2} \sum_{k=0}^{L-1} k^2 \sum_{|a-b|=k} MC_t(a, b) \quad (5.27.)$$

Fiecare termen al matricii MC_t este ponderat prin distanța sa față de diagonală. Se obține un indice ce corespunde noțiunii uzuale de contrast. El este mai mare când termenii de-a lungul diagonalei matricii sunt mari, acest lucru se datorează trecerii de la un pixel cu un nivel de gri mare și unul foarte redus.

$$\text{Entropia} : 1 - \frac{1}{N_c \cdot \ln(N_c)} \sum_a \sum_b MC_t(a, b) \cdot \ln(MC_t(a, b)) \cdot \mathbf{1}_{MC_t(a, b)} \quad (5.28.)$$

$$\mathbf{1}_{MC_t(a, b)} = \begin{cases} 1, & \text{daca } MC_t(a, b) \neq 0 \\ 0, & \text{daca nu} \end{cases}$$

Entropia este mică dacă întâlnim adesea același cuplu de pixeli și mare dacă cuplul de pixeli este slab reprezentat. Ea indică gradul de dezordine care este prezent în textură.

$$\text{Corelația} : \frac{1}{N_c \cdot \sigma_x \cdot \sigma_y} \left| \sum_a \sum_b (a - m_x) \cdot (b - m_y) \cdot MC_t(a, b) \right| \quad (5.29.)$$

$$\text{Media ponderată a liniilor matricii } MC_t : m_x = \frac{1}{N_c} \sum_a \sum_b a \cdot MC_t(a, b) \quad (5.30.)$$

$$\text{Media ponderată a coloanelor matricii } MC_t : m_y = \frac{1}{N_c} \sum_a \sum_b b \cdot MC_t(a, b) \quad (5.31.)$$

$$\text{Varianța liniilor matricii } MC_t: \sigma_x^2 = \frac{1}{N_c} \sum_a \sum_b (a - m_x)^2 \cdot MC_t(a, b) \quad (5.32.)$$

$$\text{Varianța coloanelor matricii } MC_t: \sigma_y^2 = \frac{1}{N_c} \sum_a \sum_b (b - m_y)^2 \cdot MC_t(a, b) \quad (5.33.)$$

$$\text{Omogenitatea locală: } \frac{1}{N_c} \sum_a \sum_b \frac{1}{1 + (a - b)^2} \cdot MC_t(a, b) \quad (5.34.)$$

$$\text{Directivitatea: } \frac{1}{N_c} \sum_a MC_t(a, a) \quad (5.35.)$$

Directivitatea este importantă mai ales atunci când există pixeli de același nivel de gri separați printr-o translație t [124].

$$\text{Uniformitatea: } \frac{1}{N_c^2} \sum_a MC_t^2(a, a) \quad (5.36.)$$

Este o caracteristică foarte utilă atunci când pixeli de același nivel de gri apar adesea în direcția de translație.

Indicii sau caracteristicile menționate mai sus sunt corelate, dar permit reducerea considerabilă a informației conținute în matricea de coocurență [125].

Din practică rezultă că în general sunt suficienți 5 indici pentru a discrimina texturile: omogenitatea, contrastul, entropia, omogenitatea locală și momentul de ordinul 2. Aceste aspecte au fost studiate de autor, care a propus un algoritm pentru segmentarea imaginilor folosind proprietățile texturilor [126].

Problemele apar însă atunci când prelucrăm imagini reale, cu un număr mare de texturi, texturi ce pot avea caracteristici asemănătoare, caz în care segmentarea cu praguri multiple devine o problemă serioasă.

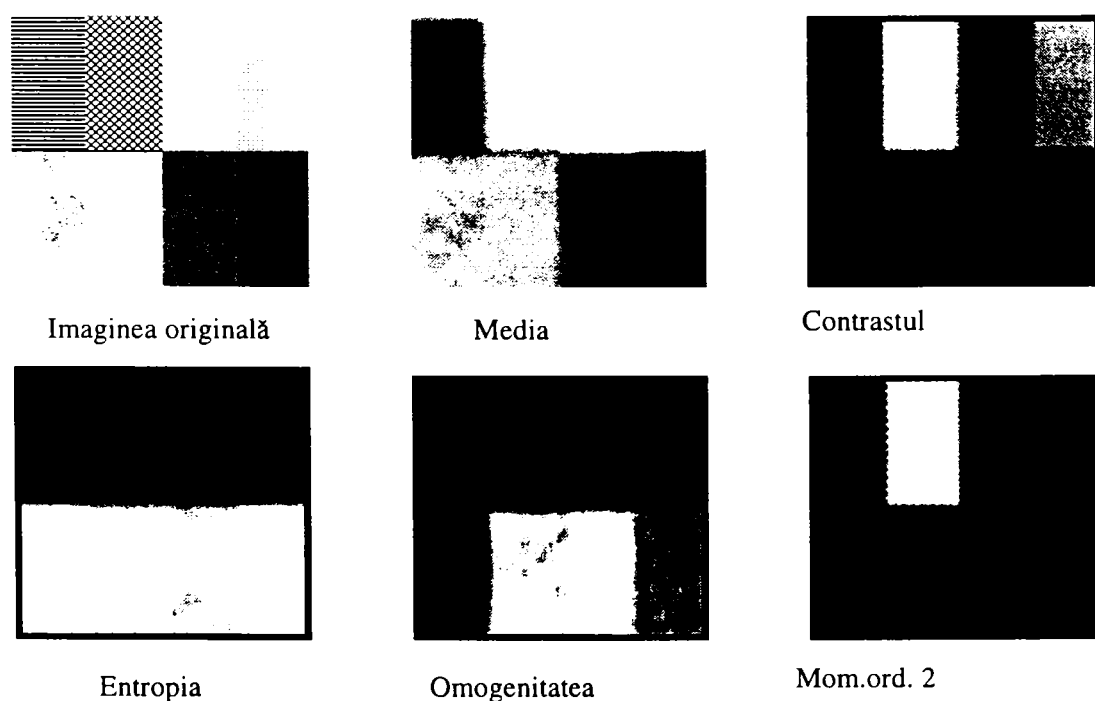


Figura 5-3 Aplicarea caracteristicilor matricii de coocurență pentru o imagine ce conține 8 texturi

În figura 5-3 sunt prezentate atributele unei imagini ce conține 8 texturi diferite. Se poate remarca faptul că una, două sau chiar trei texturi se pot identifica ușor, în schimb pentru celelalte texturi există o anumită confuzie. O segmentare multiprag nu ar putea oferi calitatea dorită.

O primă metodă propusă de autor presupune folosirea a două caracteristici și segmentarea imaginii rezultate cu o tehnică multiprag. Din figura 5-3 se observă că pentru o caracteristică rezultatul prelucrării unei texturi poate avea o valoare mare, iar pentru altă caracteristică pentru aceeași textură rezultatul prelucrării poate avea un nivel scăzut. O operație simplă de însumare ar putea media diferențele ce există și ar îngreuna din nou segmentarea. De aceea s-a urmărit să se accentueze diferențele, "contrastul" dintre două caracteristici, iar pe baza acestuia să se realizeze segmentarea. Contrastul propus, dintre două caracteristici, îl definesc în următorul mod:

$$C_{x,y} = \frac{|A_{x,y} - B_{x,y}|}{\max\{A, B\}} \quad (5.37.)$$

Rezultatul segmentării este prezentat în figura 5-4. Se observă o îmbunătățire considerabilă a calității segmentării.



Figura 5-4 Rezultatele segmentării multiprag a contrastului următoarelor perechilor de caracteristici: a) media-momentul de ordinul 2; b) entropia-media; c) media-momentul de ordinul 3; d) contrast-omogenitate.

Există o mică confuzie între două texturi deoarece media și dispersia acestora au valori foarte apropiate. În figura 5-5 este reprezentată histograma acestor texturi.

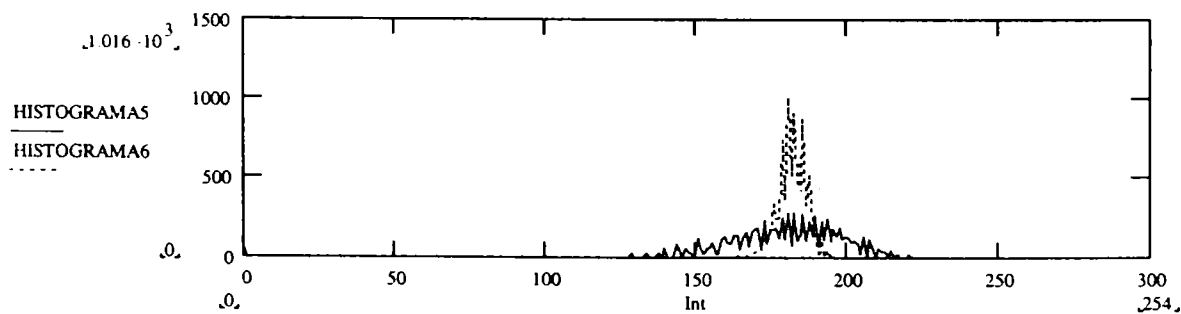


Figura 5-5 Histograma texturilor 5 și 6 din imaginea inițială.

Cea de a doua metodă propusă de autor folosește mai multe caracteristici simultan și se bazează pe definirea unei distanțe (metrice) care să permită o segmentare ulterioară multiprag. Am folosit folosită o metrică Euclidiană cu ponderarea termenilor ce intră în însumare.

$$DIST = \frac{\sqrt{\sum_{i \in \text{Nr caracteristici}} (a_i C_i)^2}}{\text{Nr. caracteristici}} \quad (5.38.)$$

a_i – coeficientul de ponderare al caracteristicii i . C_i – caracteristica i a texturii.

În figura 5-6 sunt prezentate mai multe rezultate experimentale. Pentru segmentarea imaginii sau folosit între două și cinci caracteristici diferite.

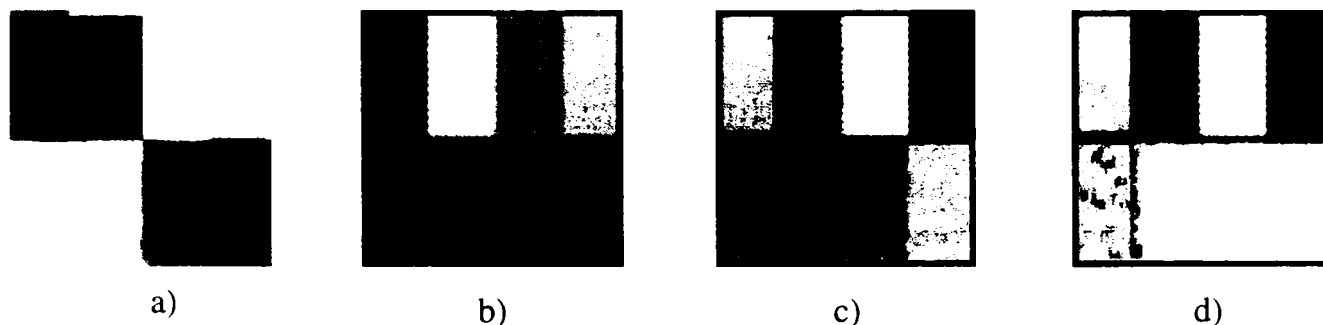


Figura 5-6 Rezultatele segmentării multiprag a imaginii “distanță” formată cu următoarele caracteristici: a) media-contrastul; b) media-contrastul-momentul de ordinul 3; c) media-contrastul-momentul de ordinul 3-entropia; d) media-contrastul-momentul de ordinul 3-entropia-omogenitatea.

În procesul de implementare și experimentare, am constatat că o dată cu creșterea numărului de texturi folosite a crescut considerabil de mult și complexitatea calculelor iar rezultatul obținut nu a fost considerabil mai bun, ci dimpotrivă, folosirea mai multor caracteristici a dus la înrăutățirea performanțelor de segmentare ale fiecărei caracteristici. Rezultatele bune ale unei caracteristici pentru o textură au fost diminuate de rezultatele defavorabile ale altor caracteristici pentru aceeași textură.

Periodicitatea calculată pornind de la coocurență

Multe imagini, în general cele ce conțin texturi, prezintă un aspect periodic. Perioada este un atribut important care poate fi estimat pornind de la spectru [127]. Poate servi de asemenea la extragerea primitivelor și regula lor de plasare în textură. Perioada elementară este în general un paralelogram. Acest paralelogram nu este unic, dar poate fi determinat cel care este definit de cele mai mici dimensiuni. Vom nota cu f și g vectorii directori pentru laturile paralelogramului.

Pentru studiul texturilor periodice, R. W. Connors și C. A. Harlow au propus utilizarea indicilor texturilor de ordinul doi pentru determinarea perioadei elementare și a regulilor de plasare a texturilor.

Dacă se consideră o textură periodică de perioadă f și nu ne interesează decât translațiile t , după aceeași direcție ca și f , matricea de coocurență verifică următoarele trei proprietăți :

- MC_t este o matrice diagonală.
- MC_t are elementele, care nu sunt pe diagonală, diferite de zero pentru orice t , atâta timp cât este îndeplinită relația $\|t\| < \|f\|$ (norma euclidiană).
- MC_{mf+tg} pentru orice valoare a lui m , ceea ce înseamnă că matricile de coocurență sunt de aceeași periodicitate ca și textura.

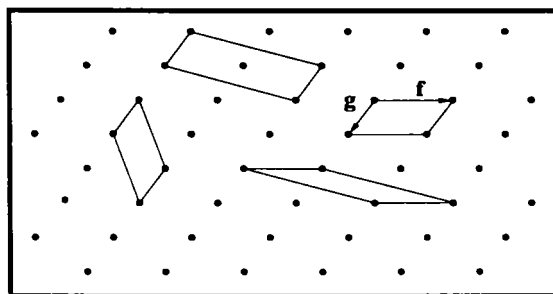


Figura 5-7 Exemple de perioade elementare pentru o textură periodică: cea mai mică perioadă este definită de vectorii f și g

Matricile de coocurență generalizate [128], [129] sunt pentru a descrie aranjamentul spațial al primitivelor imaginilor cum ar fi contururile sau liniile. Ele generalizează noțiunea de matrice de coocurență de la studiul nivelelor de gri la studiul distribuției spațiale a primitivelor texturilor.

Este necesar să se cunoască:

- Primitiva **P**.
- Un atribut p al acestei primitive.
- Un predicat spațial **Pred** de legare spațială a două primitive, acesta poate lua două valori adevărat sau fals.

Matricea de coocurență generalizată este definită pentru un atribut p și un predicat spațial **Pred**. Fie i și j două valori ale atributului p , termenul $G(i,j)$ al matricii de coocurență generalizată se exprimă în următorul mod:

$$G(i, j) = \frac{\text{card}\{(P_1, P_2) \mid p(P_1) = i, p(P_2) = j, \text{Pred}(P_1, P_2) = \text{adevarat}\}}{\text{card}\{(P_1, P_2) \mid \text{Pred}(P_1, P_2) = \text{adevarat}\}} \quad (5.39.)$$

De exemplu, primitivele pot fi pixeli, pixelii unui contur sau a unor porțiuni de contur. Dacă:

- Primitiva este pixelul – atunci atributul este nivelul de gri al pixelului și dacă predicatul este adevărat când cei doi pixeli sunt separați printr-o translație t fixată regăsită la matricea de coocurență.
- Primitiva este un pixel din contur – atributul poate fi orientarea conturului, contrastul lui, sau mai simplu poziția pixelului în imagine.

Au fost propuse diferite predicate spațiale cu atribut orientarea conturului:

- **Pred**₁(P_1, P_2, k) este adevărat dacă distanța între doi pixeli P_1 și P_2 de contur sunt inferioari sau egali cu k .
- **Pred**₂(P_i, P_j, k) este adevărat pentru primitivele P_i și P_j , dacă P_j unui sector de unghi θ limitat de k și având pentru axe vectorul f_i tangent la contur în P_i .
- **Pred**₃(P_i, P_j, k) este identic cu **Pred**₂(P_i, P_j, k) doar pentru direcția perpendiculară a vectorului f_i .
- **Pred**₂(P_i, P_j, k) și **Pred**₃(P_i, P_j, k) servesc pentru determinarea elongației și a mărimii dimensiunilor texturii.

Primitiva este o porțiune de contur – atributul poate fi poziția unei extremități, orientarea sau lungimea conturului.

Pentru matricile de coocurență se pot defini atribute care să permită clasificarea texturilor. Ele sunt calculate pentru un atribut p și un predicat spațial **Pred**:

$$\text{Omogenitatea} : \sum_i \sum_j G(i, j)^2 \quad (5.40.)$$

Acest indice este mare când toate elementele lui G sunt egale.

$$\text{Contrastul} : \sum_i \sum_j d(i, j) \cdot G(i, j) \quad (5.41.)$$

Unde $d(i,j)$ este o măsură care depinde de atributul p utilizat, de exemplu, dacă p este orientarea vom putea lua $d(i,j) = |\sin(i,j)|$.

$$\text{Entropia} : \sum_i \sum_j G(i, j) \cdot \ln(G(i, j)) \cdot \mathbf{1}_{G(i,j)} \quad (5.42.)$$

$$\text{unde, } \mathbf{1}_{G(i,j)} = \begin{cases} 1, & \text{daca } G(i, j) \neq 0 \\ 0, & \text{daca nu} \end{cases}$$

$$\text{Corelația : } \sum_i \sum_j \frac{i \cdot j \cdot G(i, j) - m_x^i m_y^j}{\sigma_x^i \cdot \sigma_y^j} \quad (5.43.)$$

Unde,

- m_x^i este media liniei i ;
- m_y^j este media coloanei j ;
- σ_x^i este dispersia liniei i ;
- σ_y^j este dispersia coloanei j .

5.1.4. Lărgimea plajelor nivelurilor de gri sau isosegmentele

O plajă a nivelelor de gri sau un isosegment este un ansamblu de pixeli consecutivi, într-o direcție dată, având aceleași nivel de gri [130]. Lungimea plajei "run length" este numărul de pixeli din acest ansamblu. La fiecare direcție, se poate asocia o matrice a lungimii plajelor $P_\theta = (p_\theta(i, j))$. Elementul $p_\theta(i, j)$ al acestei matrici reprezintă numărul de plaje de lungime j , în direcția θ , constituie de pixelii de nivel i .

Într-o regiune, vom putea extrage diferite atribute ale acestei matrici de L linii și n_θ coloane (L este numărul de nivele de gri din imagine și n_θ este lungimea maximă a corzii în direcția θ din regiune):

$$\text{Numărul de lungimi de plaje : } SPL = \sum_{i=0}^{L-1} \sum_{j=1}^{n_\theta} p_\theta(i, j) \quad (5.44.)$$

$$\text{Proporția de plaje mici : } RF_1 = \frac{1}{SLP} \sum_{i=0}^{L-1} \sum_{j=1}^{n_\theta} \frac{p_\theta(i, j)}{j^2} \quad (5.45.)$$

$$\text{Proporția de plaje mari : } RF_2 = \frac{1}{SLP} \sum_{i=0}^{L-1} \sum_{j=1}^{n_\theta} j^2 \cdot p_\theta(i, j) \quad (5.46.)$$

$$\text{Eterogenitatea nivelelor de gri : } RF_3 = \frac{1}{SLP} \sum_{i=0}^{L-1} \left(\sum_{j=1}^{n_\theta} p_\theta(i, j) \right)^2 \quad (5.47.)$$

$$\text{Eterogenitatea lungimii plajelor : } RF_4 = \frac{1}{SLP} \sum_{j=1}^{n_\theta} \left(\sum_{i=0}^{L-1} p_\theta(i, j) \right)^2 \quad (5.48.)$$

$$\text{Raportul plajelor : } RF_5 = SLP/K \quad (5.49.)$$

5.1.5. Spectrul

Transformata Fourier permite trecerea unei imagini din domeniul spațial în domeniul frecvențelor. Transformata Fourier a unei imagini corespunde unui semnal 2D $A[m, n]$ definit pe \mathbf{Z}^2 și se exprimă sub forma:

$$F(u, v) = \sum_{m=-\infty}^{+\infty} \sum_{n=-\infty}^{+\infty} A[m, n] \cdot e^{-2j\pi(mu+nv)} \quad (5.50.)$$

Această funcție este de perioadă 2π după u și v și reprezintă transformatele Fourier decalate ale semnalului continuu $A(x, y)$ [131].

În practică $A[s] = A[m, n]$, cu m, n întregi, $0 \leq m \leq M-1$, $0 \leq n \leq N-1$ este un semnal cu suport finit. Transformata Fourier discretă în acest caz este:

$$F(u, v) = \frac{1}{M \cdot N} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} A[m, n] \cdot e^{-2j\pi(\frac{mu}{M} + \frac{nv}{N})} \quad (5.51.)$$

unde, $0 \leq u \leq M-1$, $0 \leq v \leq N-1$

Pentru a evita problemele delicate de interpretare a fazei vom exploata adesea modulul transformatei Fourier sau spectrul Fourier.

În figura următoare am reprezentat o textură și spectrul acesteia, care este simetric în raport cu originea. Dinamica spectrului unei imagini este în general ridicată și de aceea pentru o bună afișare, de obicei se reprezintă $\ln|F(u, v)|$. Existența unor periodicități (pe orizontală, pe diagonală sau pe verticală) în imagine determină ca reprezentarea spectrului să prezinte periodicități asemănătoare, deci periodicitatea imaginii se traduce în periodicitatea spectrului ce este mult mai ușor de extras decât în plan spațial.

Spectrul este foarte bogat în informații. Se pot extrage componentele frecvențiale de maximă energie a imaginii.

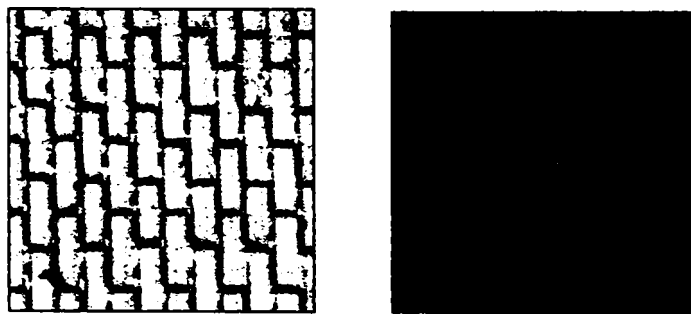


Figura 5-8. Spectrul Fourier al unei texturi 256 x 256

Densitatea spectrală de putere

Densitatea spectrală de putere, notată $S[u, v]$, este transformata Fourier a funcției de autocorelație:

$$S[u, v] = \sum_{p=-\infty}^{+\infty} \sum_{q=-\infty}^{+\infty} R[p, q] \cdot e^{-2j\pi(pu+qv)} \quad (5.52.)$$

Ea există pentru semnale de putere finită, poate fi calculată direct pornind de la transformata Fourier. Pentru un semnal ergodic $A[m, n]$, s-a demonstrat că:

$$S(u, v) = \lim_{M, N \rightarrow +\infty} E \left\{ \frac{1}{(2M+1) \cdot (2N+1)} \left| \sum_{m=-M}^{+M} \sum_{n=-N}^{+N} A[m, n] \cdot e^{-2j\pi(mu+nv)} \right|^2 \right\} \quad (5.53.)$$

În practică, nu se poate calcula spectrul cu ajutorul expresiei de mai sus datorită dimensiunii finite a imaginii. Se poate exploata atunci periodicitatea estimatorului spectral. Expresia sa este dată de relația:

$$\hat{P}[u, v] = \frac{1}{(2M+1) \cdot (2N+1)} \left| \sum_{m=-M}^{+M} \sum_{n=-N}^{+N} W[m, n] \cdot A[m, n] \cdot e^{-2j\pi(mu+nv)} \right|^2 \quad (5.54.)$$

unde, $W[m, n]$ este o funcție deterministă de ponderare (de exemplu o fereastră Hanning), introdusă pentru reducerea lobilor secundari ai ferestrei rectangulare impuse de dimensiunea finită a imaginii.

O metodă indirectă de calcul a spectrului de putere utilizează funcția de autocorelație ce poate fi calculată:

- Sub formă de medie statistică, dacă semnalul este staționar:

$$R[p, q] = E\{A[m, n] \cdot A[m + p, n + q]\} \quad (5.55.)$$

- Sub formă de medie spațială, dacă semnalul este ergodic

$$R[p, q] = \lim_{M, N \rightarrow +\infty} \frac{1}{(2M + 1) \cdot (2N + 1)} \sum_{m=-M}^{+M} \sum_{n=-N}^{+N} A[m, n] \cdot A[m + p, n + q] \quad (5.56.)$$

Periodicitatea calculată pornind de la spectru - Matsuyama propune o metodă de determinare a vectorilor f și g ce determină perioada minimală pornind de la periodicitatea calculată în plan Fourier.

Dacă textura este exact periodică, toată energia spectrului va fi concentrată pe frecvențele ce corespund perioadelor definite de : $h = mf + ng$, unde m și n sunt numere întregi. Chiar și în cazul texturilor cu neregularități în distribuția primitivelor sau afectate de zgomot, energia spectrului pentru frecvențele $mf + ng$ rămâne un criteriu discriminant.

Pentru o imagine de dimensiune $M \times M$, relațiile între perioadele spațiale f și g și perioadele spectrale u și v sunt:

$$|u| = \frac{M}{|f| \cos \theta} \quad |v| = \frac{M}{|g| \sin \theta} \quad (5.57.)$$

$$\arg u = \arg g + \pi/2 \pmod{\pi} \quad \arg v = \arg f + \pi/2 \pmod{\pi} \quad (5.58.)$$

$$\theta = \arg(g) - \arg(f) \quad (5.59.)$$

Există deci un decalaj de $\pi/2$ între direcțiile periodicității în planul imagine și planul frecvențelor. Perioadele sunt calculate mult mai exact în planul Fourier, prin extragerea maximului local al spectrului și supresia armonicilor. Aceste formule permit deducerea perioadelor f și g ale imaginii.

5.1.6. Contrastul asociat unei regiuni

Dacă regiunile în care lucrăm conțin zone omogene de gri și prezintă tranziții de nivele de gri foarte abrupte vis-a-vis de regiunile vecine adiacente, se poate utiliza noțiunea de contrast [132] [133]. Contrastul asociat pentru două regiuni s și t de nivele de gri $A[s]$ și $A[t]$ se definește în următorul mod în funcție de natura imaginii:

Pentru imaginile de natură logaritmică (cele care au nivelul de gri al pixelilor rezultați în urma unei compresii logaritmice a sistemului de captare), vom avea:

$$c(s, t) = \begin{cases} \frac{|A[s] - A[t]|}{\max(A[s], A[t])} & \text{daca } \max(A[s], A[t]) \neq 0 \\ 0 & \text{daca nu} \end{cases} \quad (5.60.)$$

Pentru imaginile de natură nelogaritmică, vom avea:

$$c(s, t) = \begin{cases} \frac{|A[s] - A[t]|}{L - 1} & \text{daca } L - 1 \neq 0 \\ 0 & \text{daca nu} \end{cases} \quad (5.61.)$$

Se pot calcula pentru o regiune R diferite contraste:

Contrastul interior este media contrastelor locale a punctelor din R :

$$C_{\text{int}}(R) = \frac{1}{K} \sum_{s \in R} \max(c(s,t), c \in V(s) \cap R) \quad (5.62.)$$

unde, K este numărul de pixeli din regiunea R , și $V(s)$ este o vecinătate a unei zone s , de exemplu o vecinătate de 3×3 .

Contrastul exterior este media contrastelor locale ale frontierei regiunii R . Vom denumi frontiera regiunii R ansamblul de puncte R conexe și unei ale regiunii decât R :

$$C_{\text{front}}(R) = \frac{1}{K_f} \sum_{s \in F} \max(c(s,t), t \in V(s), t \notin R) \quad (5.63.)$$

unde, F este frontiera regiunii R , și K_f este cardinalul acesteia. $V(s)$ este o vecinătate a unei zone s , de exemplu o vecinătate de 3×3 .

De remarcat, că regiunea R nu este omogenă dacă: $C_{\text{front}}(R) \leq C_{\text{int}}(R)$ (5.64.)

Contrastul unei regiuni este definit de relația:

$$C(R) = \begin{cases} 1 - \frac{C_{\text{int}}(R)}{C_{\text{front}}(R)} & \text{daca } C_{\text{int}}(R) < C_{\text{front}}(R) \\ 0 & \text{daca nu} \end{cases} \quad (5.65.)$$

Contrastul asociat unei regiuni de etichete Λ este definit de relația:

$$C(\Lambda) = \frac{1}{\sum K_i} \sum_{i=1}^{N_r} K_i \cdot C(R_i) \quad (5.66.)$$

unde, K_i este cardinalul regiunii R_i și N_r este numărul de regiuni din imagine.

5.1.7. Atribute de suprafață

Curburile - dacă modelăm imaginea ca o suprafață în spațiul 3-D, se poate studia local această suprafață folosind geometria diferențială. În cazul suprafețelor netede, toate derivatele parțiale secundare există și sunt continue, se folosesc două forme fundamentale L_1 și L_2 pentru calcularea atributelor de curbura.

Dacă un punct de pe suprafață este definit prin poziția sa $U = (x, y, z)$, unde x, y sunt coordonatele în imagine iar z este nivelul de gri al pixelului de coordonate (x, y) , prima formă fundamentală este:

$$L_1 = dU \quad (5.67.)$$

$$dU = E'dx^2 + 2F'dxdy + G'dy^2$$

unde,

$$E' = 1 + \left(\frac{\partial z}{\partial x} \right)^2, \quad F' = \frac{\partial z}{\partial x} \frac{\partial z}{\partial y}, \quad G' = 1 + \left(\frac{\partial z}{\partial y} \right)^2 \quad (5.68.)$$

L_1 reprezintă elementul de lungime al arcului du pentru curbele ce descriu suprafața și trec prin punctul de coordonate (x, y, z) . El nu depinde de translații și rotații efectuate, este deci un atribut intrinsec al suprafeței.

Forma fundamentală de ordinul doi, reprezintă curbura curbelor trasate la suprafață. Dacă \mathbf{n} este un vector normal la suprafață, ea este definită de:

$$L_2 = dU \cdot d\mathbf{n} = e'dx^2 + 2f'dxdy + g'dy^2 \quad (5.69.)$$

$$e' = \frac{1}{\Delta'} \cdot \frac{\partial^2 z}{\partial^2 x} \quad f' = \frac{1}{\Delta'} \cdot \frac{\partial^2 z}{\partial x \partial y} \quad g' = \frac{1}{\Delta'} \cdot \frac{\partial^2 z}{\partial^2 y} \quad (5.70.)$$

$$\Delta' = \sqrt{E'G' - F'^2} \quad (5.71.)$$

L_2 este un atribut extrinsec al suprafeței, el depinde de poziția acesteia în spațiu.

Funcția $k = L_1/L_2$ este curbura normală într-un punct U la suprafață. Extremele acestei funcții într-un punct definesc curbura principale k_1 și k_2 . Pornind de la acestea se definesc:

Curbura gaussiană :

$$K_1 = k_1 k_2 \quad (5.72.)$$

Curbura medie:

$$K_1 = (k_1 + k_2)/2 \quad (5.73.)$$

Curbura gaussiană și cea medie sunt invariante la rotație și translație, iar în plus curbura gaussiană este mai stabilă la schimbările de scară.

Cuadricele - se poate studia mult mai precis suprafața dacă se folosesc aproximări local prin suprafețe polinomiale de ordin secund.

Ecuția generală este:

$$q(x, y, z) = ax^2 + by^2 + cz^2 + 2dxy + 2eyz + 2fzx + 2gx + 2hy + 2iz + j \quad (5.74.)$$

Dacă se presupune că una din axele de inerție este direcția nivelelor de gri (Oz), ecuația (5.74) se scrie în coordonatele de inerție astfel:

$$q'(x, y, z) = \lambda x^2 + \mu y^2 + cz^2 + 2g'x + 2h'y + 2iz + j = 0 \quad (5.75.)$$

unde λ și μ sunt valorile proprii asociate celor două direcții diferite de (Oz).

În general autorii [134], [135] se limitează doar la formele bicuadrice deoarece imaginea este o funcție $z = f(x, y)$ și deci pentru orice punct (x, y) îi corespunde o valoare z . În acest caz ecuația (5.75) devine:

$$p(x, y) = ax^2 + by^2 + 2dxy + 2gx + 2hy + j \quad (5.76.)$$

Calculul celor 6 coeficienți $a, b, d, g, h,$ și j ai formei bicuadrice dau o bună aproximare a suprafeței pentru o vecinătate a pixelului (x, y) . Pentru a obține coeficienții trebuie să realizăm o convoluție a imaginii cu 6 măști de dimensiunea vecinătății dorite.

Pentru o vecinătate de 3x3 avem:

$$\mathbf{A} = \frac{1}{6} \begin{bmatrix} 1 & -2 & 1 \\ 1 & -2 & 1 \\ 1 & -2 & 1 \end{bmatrix} \quad \mathbf{G} = \frac{1}{12} \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad \mathbf{D} = \frac{1}{8} \begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{bmatrix} \quad \mathbf{J} = \frac{1}{9} \begin{bmatrix} -1 & 2 & -1 \\ 2 & 5 & 2 \\ -1 & 2 & -1 \end{bmatrix}$$

Pentru o vecinătate de 5x5:

$$\mathbf{A} = \frac{1}{70} \begin{bmatrix} 2 & -1 & -2 & -1 & 2 \\ 2 & -1 & -2 & -1 & 2 \\ 2 & -1 & -2 & -1 & 2 \\ 2 & -1 & -2 & -1 & 2 \\ 2 & -1 & -2 & -1 & 2 \end{bmatrix} \quad \mathbf{G} = \frac{1}{100} \begin{bmatrix} -2 & -1 & 0 & 1 & 2 \\ -2 & -1 & 0 & 1 & 2 \\ -2 & -1 & 0 & 1 & 2 \\ -2 & -1 & 0 & 1 & 2 \\ -2 & -1 & 0 & 1 & 2 \end{bmatrix}$$

$$\mathbf{D} = \frac{1}{200} \begin{bmatrix} 4 & 2 & 0 & -2 & -4 \\ 2 & 1 & 0 & -1 & -2 \\ 0 & 0 & 0 & 0 & 0 \\ -2 & -1 & 0 & 1 & 2 \\ -4 & -2 & 0 & 2 & 4 \end{bmatrix} \quad \mathbf{J} = \frac{1}{175} \begin{bmatrix} -13 & 2 & 7 & 2 & -13 \\ 2 & 17 & 22 & 17 & 2 \\ 7 & 22 & 27 & 22 & 7 \\ 2 & 17 & 22 & 17 & 2 \\ -13 & 2 & 7 & 2 & -13 \end{bmatrix}$$

Două exemple de măști pentru vecinătăți de dimensiunea 3x3 și 5x5 au fost prezentate mai sus. Masca **A** este folosită pentru calculul coeficientului a , masca **D** este folosită pentru calculul coeficientului d , masca **G** este folosită pentru calculul coeficientului g , masca **J** este folosită pentru calculul coeficientului j , coeficientul b va fi calculat cu ajutorul transpusei matricii **A**, coeficientul h va fi calculat cu ajutorul transpusei matricii **G**.

Atribute geometrice - pentru o regiune R dintr-o imagine ai cărei pixeli sunt de coordonate (x_i, y_i) putem defini câteva atribute pentru aceasta:

Aria $S(R)=K$ sau numărul de pixeli din regiune.

Perimetrul $P(R)$ în general se calculează ca fiind suma distanțelor între pixelii consecutivi ai conturului, poate fi 1 sau $\sqrt{2}$. Poate fi calculat pornind de la conturul interior sau cel exterior al regiunii.

Centrul de gravitate G este de coordonate:

$$x_G = \frac{1}{K} \sum_{i=1}^K x_i \quad \text{și} \quad y_G = \frac{1}{K} \sum_{i=1}^K y_i \quad (5.77.)$$

Diametru regiunii

$$D = \max\{d(s, t) | s, t \in R\} = \max\{d(s, t) | s, t \in F(R)\} \quad (5.78.)$$

$F(R)$ reprezintă frontiera regiunii R iar $d(s, t)$ distanța între punctele s și t , de exemplu poate fi distanța euclidiană.

Compactitatea sau factorul de circularitate:

$$\frac{4\pi(SR)}{P^2(R)} \quad (5.79.)$$

Se apropie de 1 dacă forma regiunii R este apropiată de un disc și tinde la 0 dacă aceasta ia o formă foarte alungită sau decupată.

Direcția principală de inerție se definește ca fiind primul vector propriu al matricii

$$\begin{bmatrix} a & c \\ c & b \end{bmatrix} \text{ unde } a, b, c, d \text{ sunt dați de relațiile:}$$

$$a = \frac{1}{K} \sum_{i=1}^K (x_i - x_G)^2 = \frac{1}{K} \sum_{i=1}^K x_i^2 - x_G^2 \quad (5.80.)$$

$$b = \frac{1}{K} \sum_{i=1}^K (y_i - y_G)^2 = \frac{1}{K} \sum_{i=1}^K y_i^2 - y_G^2 \quad (5.81.)$$

$$c = \frac{1}{K} \sum_{i=1}^K (x_i - x_G)(y_i - y_G) = \frac{1}{K} \sum_{i=1}^K x_i y_i - x_G y_G \quad (5.82.)$$

Direcția principală a regiunii R este definită prin unghiul α făcut față de axa x :

$$\tan 2\alpha = \frac{2c}{a - b} \quad (5.83.)$$

Dreptunghiul de încadrare este dreptunghiul a cărui laturi sunt paralele cu vectorii proprii ai matricii centrului de inerție a matricii R .

Paralelogramul de încadrare al regiunii R se calculează pornind de la poligonul ce aproximează conturul, prin calculul celor două direcții ce cumulează laturile cu direcțiile cele mai frecvente, ceea ce înseamnă direcțiile pentru care produsul vectorial este maxim. Paralelogramul de încadrare este paralelogramul circumscris regiunii R a cărui laturi sunt paralele cu cele două direcții. Elongația este raportul dintre lungimea și lățimea dreptunghiului de încadrare a regiunii R . O altă caracteristică ce poate caracteriza o regiune este raportul între suprafața regiunii și perimetrul său.

Atribute inter-regiuni - se pot studia relațiile dintre o regiune R și regiunile care îi sunt adiacente pentru a le clasifica. Pentru aceasta avem nevoie de o listă de atribute ale regiunii R , numărul de regiuni adiacente și regiuni incluse. Atributele ne pot da relații referitoare la:

- pozițiile relative ale regiunilor;
- frontierele dintre regiuni;
- diverse asemănări și diferențe ce pot fi asociate prin examinare vizuală.

5.2. Caracteristici locale ale imaginilor propuse pentru utilizarea la indexarea bazelor de imagini

5.2.1. Introducere

Texturile pot fi descrise în termeni lingvistici prin cuvinte ca: rugozitate, contrast, finețe, regularitate, etc. [136], [137]. Aceste caracteristici foarte ușor de interpretat de către ochiul uman sunt relativ greu de interpretat prin relații matematice.

Se întâlnesc două tipuri de definire a texturilor:

- Primul este determinist și se referă la repetițiile spațiale ale unui motiv de bază în diferite direcții. Această abordare structurală corespunde unei viziuni macroscopice a texturilor.
- Cel de al doilea tip este probabilistic și caută caracterizarea aspectului acesteia fără a ține seama de un motiv localizabil, nici de frecvența de repetiție a acestora. Această abordare corespunde unei analize microscopice.

Noțiunea de textură admite o descriere ce nu este foarte rigidă, iar definiția se poate da într-un mod generic [138]. Textura este compusă din câteva elemente de bază între care există o oarecare interacțiune. Elementele de bază se numesc și « texeli » [139], dimensiunea și forma acestora poate varia foarte mult, de la câțiva pixeli până la sute de pixeli.

Se remarcă, la câteva dintre ele, repetarea elementelor primitive cu câteva variații: «Boabe de cafea », «Zid din cărămizi», « Monede ». În schimb, pot exista și texturi sintetizate a căror texeli sunt amplasați exact, iar dimensiunile acestora sunt aceleași, dar pot exista și texturi cu caracteristici aleatoare.

În general, texturile reale nu au o distribuție uniformă și regulată a texelilor. Problema principală în recunoașterea unei texturi este identificarea elementului primitiv, texelul.

Texturile se pot clasifica în patru mari clase [140], după modul de definire:

- Texturi definite prin primitive;
- Texturi ce au un model structural;
- Texturi ce au un model statistic;
- Texturi gradient.

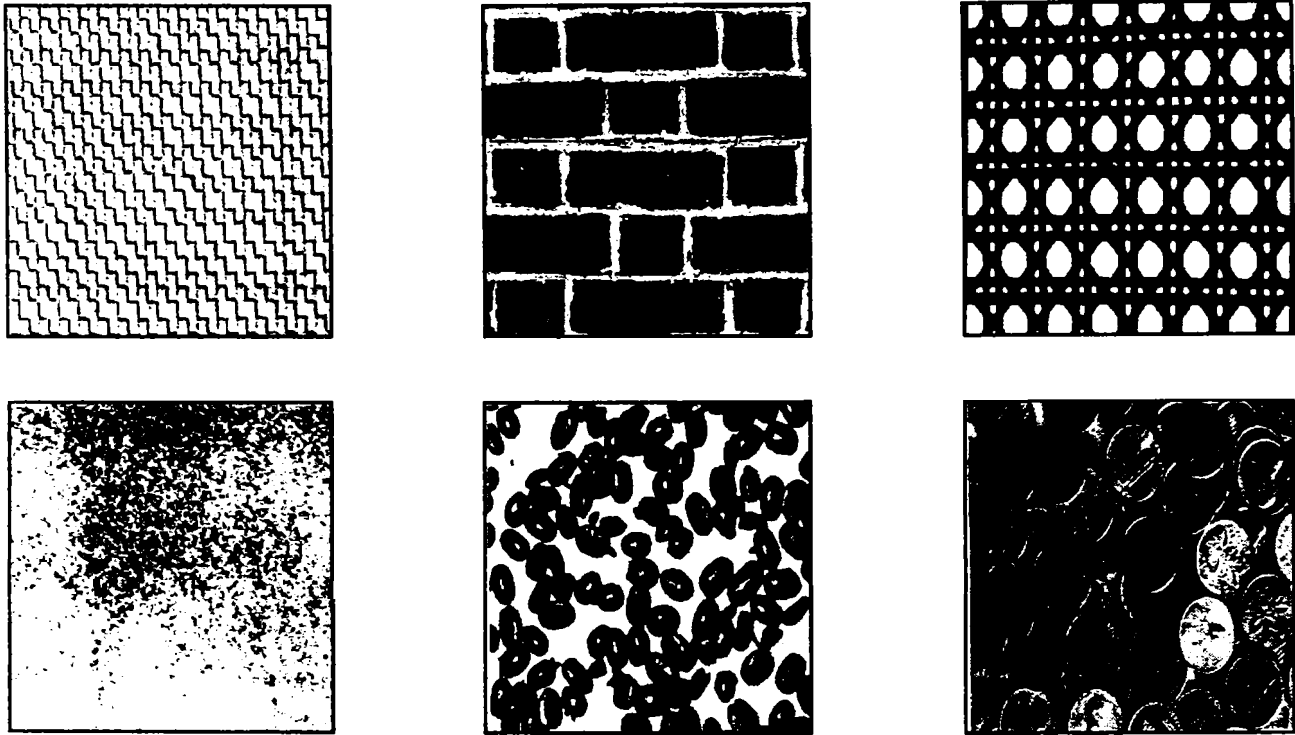


Figura 5-9 Exemple de texturi ce au primitiva repetată în mod regulat (rândul de sus)
Exemple de texturi ce au primitiva repetată în mod neregulat (rândul de jos)

5.2.2. Texturi definite prin primitive

Noțiunea de “*primitivă*” a unei texturi sau de “*texel*”, este elementul central în definirea unei texturi. Un texel este o primitivă vizuală care conține câteva proprietăți invariante și care apare în imagine în diferite poziții, deformații și orientări. O proprietate invariantă de bază a unui texel este faptul că media pixelilor ce îl formează dă un nivel de gri constant. Proprietăți mult mai elaborate sunt legate de descrierea formei.

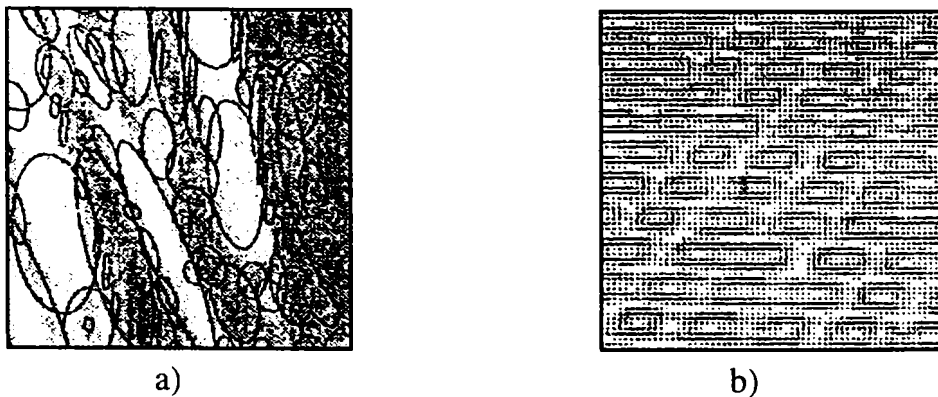


Figura 5-10 a) Textură ce are primitivele elipse de diverse dimensiuni. b) Textură ce are primitivele segmente de linii drepte

Primitivele texturii pot fi pixeli, grupări de pixeli (cum ar fi segmente de curbe) sau regiuni. Textura « boabe de cafea » ar putea fi descrisă și în următorul mod: elipse negre pe un fundal alb. În definirea unei texturi cu primitive este foarte important faptul că ele apar în mod repetat. Se pune întrebarea de câte ori? Acest lucru poate fi explicat calitativ prin imaginarea unei ferestre ce corespunde dimensiunii unui texel, și raportarea întregii suprafețe la suprafața ferestrei.

Este foarte importantă rezoluția la care se face analiza texturii. De exemplu, textura « zid din cărămizi », la o rezoluție scăzută elementul primitiv este « cărămida ». Dacă rezoluția crește, se poate lua în considerare textura fiecărei cărămizi, iar pentru aceasta să se facă analiza elementului primitiv. Din acest scurt exemplu se poate remarca faptul că ideea de primitivă poate fi văzută și sub o forma ierarhică. De aceea este foarte importantă alegerea unei rezoluții potrivite, adică alegerea dimensiunii ferestrei de analiză a texturii astfel încât să permită reperarea primitivelor dorite. Pentru analiza imaginii la nivel de grup de cărămizi este nevoie de o fereastră mult mai mare decât în cazul analizării texturii la nivel de cărămidă.

5.2.3. Texturi ce au un model structural de plasare a texelilor

Există o mare diversitate de modele structurale, ele ar putea fi grupate în câteva mari categorii:

- Modele structurale regulate – poligoanele corespunzătoare au toate același număr de laturi.
- Modele structurale semiregulate – unde avem două tipuri de poligoane, diferite ca număr de laturi, ce înconjoară un vertex.

Aceste structuri se descriu prin listarea în ordine a numărului de laturi ale poligoanelor din jurul unui vertex. Atunci, de exemplu, o construcție hexagonală poate fi descrisă prin (6,6,6). Este important de remarcat că în unele cazuri putem considera descrierea unei construcții ca fiind ea însăși o primitivă, în acest caz primitivele definesc o construcție și construcția descrie primitiva de plasare ca fiind duală acesteia.

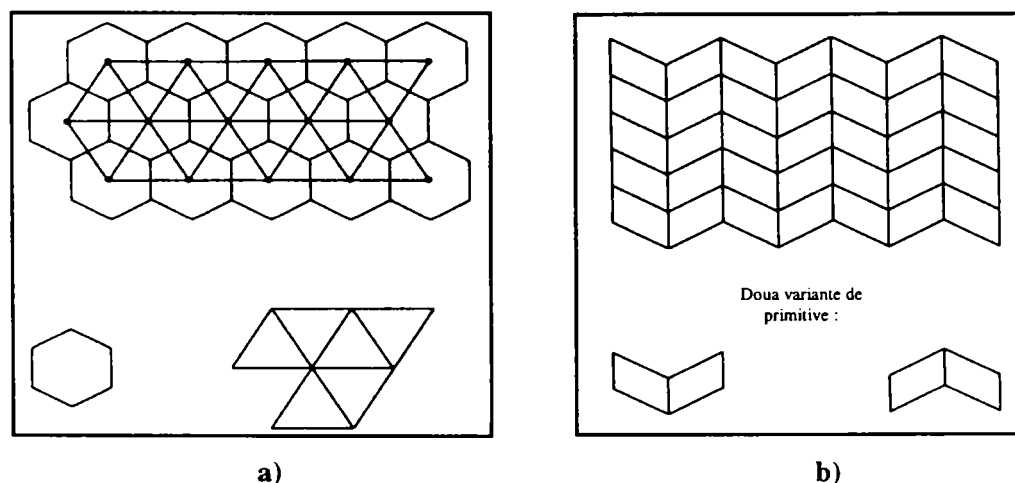


Figura 5-11 Exemplu structural regulat de plasare a primitivelor: a) Texel hexagonal într-o structură triunghiulară b) Textură ce poate fi realizată cu două primitive diferite

Gramatica modelelor

Un mod practic de descriere a regulilor ce guvernează structura unei texturi se poate face utilizând « gramatica texturii ». Gramatica descrie cum se poate genera o textura aplicând succesiv aceleași reguli pentru un număr restrâns de simboluri. Cu un număr restrâns de simboluri și de reguli, o gramatică poate genera un număr mare de texturi foarte complexe. În cazul nostru simbolurile sunt reprezentate de texeli.

Între prototipul de model pentru o textură și textura dintr-o imagine reală există diferențe ce pot fi atribuite unor reguli ce apar cu o anumită probabilitate. Gramaticile astfel formate au un caracter stohastic [141].

Aici nu este o gramatică unică pentru o textură dată, de fapt aici avem în mod uzual o infinitate de variante pentru reguli și simboluri. Deci, gramaticile texturii au o descriere sintactică ambiguă. În exemplul de mai sus, este arătată ambiguitatea sintactică a texturii și două

cazuri posibile de alegere a primitivelor. Această textură este de altfel și semantic ambiguă, ea poate fi privită și ca o structură tridimensională.

Există mai multe variante pentru ideea de *gramatică formală*:

- Gramatica formei;
- Gramatica arborelui;
- Gramatica ariei.

Gramatica formei se distinge de celelalte două prin faptul că are primitive de nivel înalt ce corespund formelor din textură. În exemplele cu gramatica arborelui și a ariilor, texeli sunt definiți ca pixeli și acest lucru implică o complexitate mult mai mare. Un caz particular de textură ce poate fi descris prin 8 reguli în gramatica formei, necesită 85 de reguli în gramatica arborelui.

Gramatica formei

Gramatica formei este definită prin cvadrupla:

$$\langle V_i, V_m, R, S \rangle \quad (5.84.)$$

unde: V_i este o mulțime finită de forme.

V_m este o mulțime finită de forme care respectă proprietatea:

$$V_m \cap V_i = \emptyset$$

R este o mulțime finită de perechi ordonate (u, v) , astfel u este o formă ce aparține elementelor mulțimii V_m^+ și v este o formă ce aparține elementelor mulțimii V_i^+ .

S este o formă ce constă într-un element al lui V_i^+ , combinat cu un element al lui V_m^+ .

Elementele mulțimii V_i sunt denumite elemente terminale de formă (sau terminale). Elementele mulțimii V_m sunt denumite elemente neterminale de formă (sau markeri). Mulțimile V_i și V_m trebuie să fie disjuncte. Elementele mulțimii V_i^+ sunt formate de un aranjament finit de unul sau mai multe elemente din mulțimea V_i în care oricare elemente și / sau imaginile acestora pot fi folosite de mai multe ori în orice locație, orientare sau scală.

Mulțimea $V_i^+ = V_i \cup \{\Lambda\}$, unde Λ este forma inițială. Mulțimile V_m^+ și V_i^+ sunt definite în mod similar. Elementele (u, v) ale lui R sunt denumite regulile formei și se scriu uv . Unde, u este partea stânga a regulii iar v este partea dreaptă a acesteia. S este forma inițială și conține în mod normal un u astfel ca perechea ordonată (u, v) să fie din R .

O textură este generată din gramatica formei începând cu forma inițială și aplicând în mod repetat regulile gramaticii. Rezultatul aplicării unei reguli din R unei forme S este o altă formă ce constă în forma S cu partea dreaptă a lui R substituită în S pentru a obține o ocurență cu partea stângă a lui R . Pentru aplicarea regulii, asupra unei forme, trebuie procedat în următorul mod:

- Se caută partea formei care este similară geometric cu partea stângă a regulii atât din punct de vedere al elementelor terminale cât și al celor neterminale. Aici trebuie să fie o corespondență unu la unu între terminalii și markerii din partea stângă a regulii și terminalii și markerii din partea formei căreia i se aplică regula.
- Se caută transformarea geometrică (de scală, translație, rotație, oglindire) ce face partea stângă a regulii identică cu partea corespunzătoare din formă.
- Se aplică aceste transformări părții drepte a regulii.
- Se înlocuiește partea dreaptă transformată a regulii pentru partea formei ce corespunde părții stânga a regulii.

Procesul de generare este terminat atunci când regula de gramatică nu mai poate fi aplicată. Un simplu exemplu pentru o textură hexagonală este următorul : $\langle V_i, V_m, R, S \rangle$

unde,

$$\begin{aligned} V_i &= \{ \text{hexagon} \} \\ V_m &= \{ \quad \} \\ R &: \text{hexagon} \rightarrow \text{hexagon-hexagon} ; \text{hexagon-hexagon} \text{ etc.} \\ S &= \{ \text{hexagon} \} \end{aligned} \quad (5.85.)$$

Texturile hexagonale pot fi generate prin aplicarea repetată a unei singure reguli în R . Aici poate fi reprodușă și prin aplicarea regulii în direcție opusă pentru o textură a cărei formă inițială, I , este dată. Desigur regula poate genera numai texturi hexagonale.

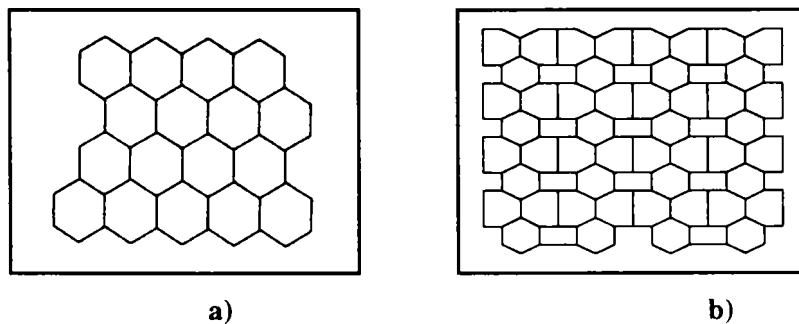


Figura 5-12a) Textură obținută prin aplicarea unei reguli în mod repetat; b) Textură ce nu poate fi obținută prin aplicarea regulii în două sensuri

Un exemplu mult mai dificil este dat prin textura « reptile ». Exceptând apariția ocazională a unui nou rând, o construcție de primitive de tipul $(3,6,3,6)$ poate modela foarte exact această textură.

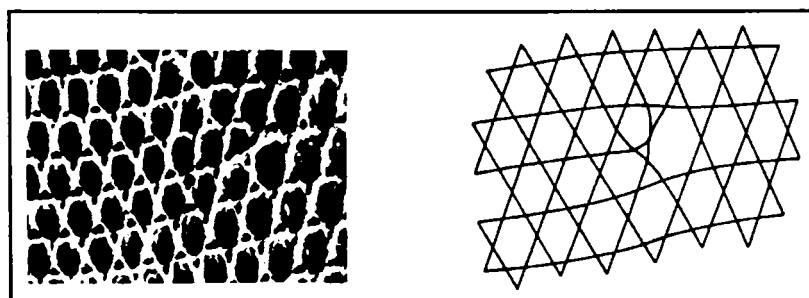


Figura 5-13 Textură reală „Reptile” cu reprezentarea amplasării primitivelor, textura prezintă și o iregularitate

În figură se poate remarca că noul rând este introdus prin împărțirea unui poligon cu 6 laturi, în două poligoane, dintre care unul are 5 iar celălalt 7 laturi. Cu ajutorul gramaticii formei, vom analiza aceste texturi și din punct de vedere al modului de amplasare a primitivelor. Acest graf ne dă o mai ușoară explicație a modului cum a fost introdus noul rând: are loc efectiv o împărțire în două a unei celule a grafului de amplasare a primitivelor.

Graful este în general format din poligoane cu 4 laturi, doar acolo unde este introdus noul rând apar două noi forme de poligoane formând construcții de tipul : $(4,4,4)$ și $(4,4,4,4,4)$.

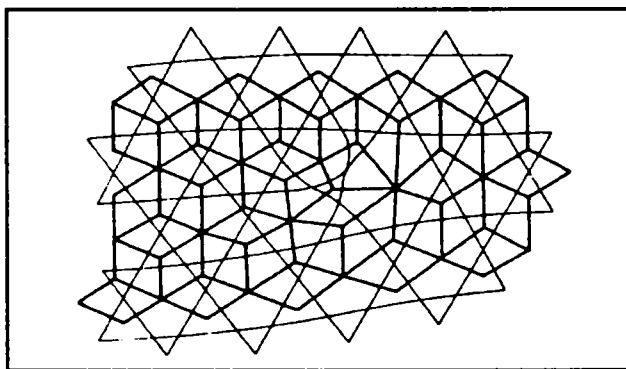


Figura 5-14 Reprezentarea texturi "reptile"

Gramatica arborelui

Simbolic gramatica arborelui este similară cu gramatica formei. O gramatică de tipul :

$$G_r = \langle V_t, V_m, r, R, S \rangle \quad (5.86.)$$

Este gramatica unui arbore, dacă:

V_t este o mulțime de simboluri terminale.

V_m este o mulțime de simboluri astfel încât să fie îndeplinită condiția:

$$V_m \cap V_t = \emptyset$$

$$r: V_t \rightarrow N$$

Unde, r este rangul asociat simbolului V_t , iar N este o mulțime formată din numere naturale.

S este simbolul de start.

R este o mulțime de reguli de forma: $X_0 \rightarrow x$ sau $X_0 \rightarrow x$

$$\begin{array}{c} x \\ / \quad \backslash \\ X_0 \dots X_{r(x)} \end{array}$$

cu x în V_t și $X_0 \dots X_{r(x)}$ în V_m .

Gramatica ariei

La fel ca și gramatica arborelui, gramatica ariei folosește o ierarhie a nivelelor de rezoluție [25] [26]. Ea are însă o altă formă față de gramatica arborelui, nu folosește încrustarea arborelui arie. O simplă gramatică a ariei ce generează primitiva unei « table de șah » este următoarea :

$$G = \langle V_t, V_n, R \rangle$$

unde,

$$V_t = \{0,1\}$$

$$V_n = \{b, S\}$$

(5.87.)

Simbolul blanc a fost notat cu b . O altă convenție de notație este de a folosi un subscript care să indice orientarea simbolurilor. De exemplu când vom descrie regulile R , vom folosi : $0_x b \rightarrow 0_x 1$

Unde, x este unul din elementele $\{ U D L R \}$, fiind astfel posibil să se indice următoarele reguli:

$$\begin{array}{cccccccc} 0 & 0 & b & 1 & 0b & 01 & b0 & 10 \\ b & 1 & 0 & 0 & & & & \end{array}$$

Atunci pentru tabla de șah vom avea:

$$R: S \rightarrow 0 \text{ sau } 1$$

$$0_x b \rightarrow 0_x 1$$

$$1_x b \rightarrow 1_x 0$$

Unde, x este unul din elementele $\{ U D L R \}$.

O codare compactă a primitivei texturii se obține prin organizarea nivelelor gramaticii ariei sub forma unei piramide. Simbolurile terminale ale unui nivel sunt simbolurile de start pentru următorul nivel gramatical ce se definește în piramidă. Această idee corespunde faptului că un nivel gramatical generează primitivele, iar celălalt modul de plasare a acestora.

În cazul bazelor de date și în special dacă sunt stocate texturi, aceste metode de reprezentare ocupă foarte puțin spațiu și pot genera un număr foarte mare de texturi regulate. Pentru aplicațiile din studiourile de televiziune ele oferă o mare diversitate de generare electronică a diverselor fundaluri, crearea de spații virtuale, etc.

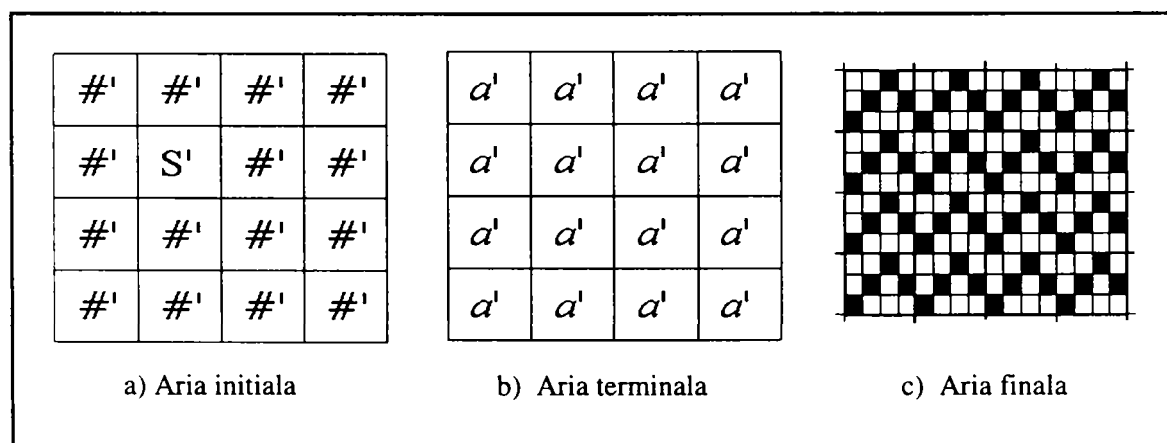


Figura 5-15 Textură realizată pornind de la gramatica unei arii

5.2.4. Texelii

Regiunea dintr-o imagine ce reprezintă un texel este acea porțiune ce conține o primitivă la nivel de pixeli. S-a constatat că este mult mai utilă în segmentarea imaginilor tratarea texturilor la nivel de primitive, texeli, decât la nivel de pixel, fiind mai utilă extragerea caracteristicilor pentru texeli decât pentru pixeli. Se pot folosi în segmentarea imaginilor utilizând texturile și elementele de muchii și contururi ce se folosesc în mod curent atunci când se realizează o segmentare la nivel de pixeli.

Începând cu 1971, au fost propuse mai multe metode de aproximare a texelilor. În funcție de textura ce se analizează forma acestora poate varia foarte mult.

Pentru figura de mai sus s-a ales ca formă a texelului elipsa. Pentru descrierea acestuia sunt necesari 5 parametri. Maleson a propus o metodă de segmentare în doi pași:

- în prima etapă se segmentează cu o metodă clasică ce ține cont de nivele de gri;
- cea de a doua etapă constă în acoperirea cu elipse a zonelor convexe detectate cu prima metodă.

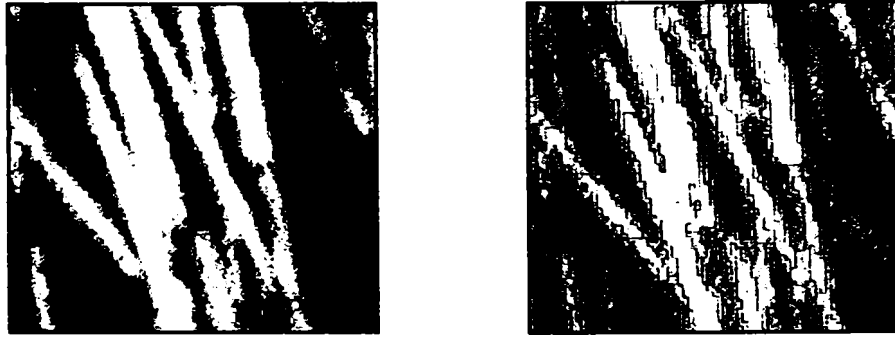


Figura 5-16 Segmentarea unei texturi folosind o descriere bazată pe nivele de gri

Parametrii elipselor sunt reprezentați de :

- coordonatele originii: x_0, y_0 .
- axele elipsei: a, b .
- orientarea elipsei: θ .

Prin medierea acestor parametri ce descriu texturi de formă eliptică se poate obține o informație cantitativă și calitativă asupra texturii.



Figura 5-17 Segmentarea unei texturi folosind o descriere bazată pe primitive de formă eliptică

5.2.5. Dependența spațială a nivelelor de gri (SGLD)

Matricile dependenței spațiale a nivelelor de gri sunt cele mai populare surse de extragere a caracteristicilor [142]. SGLD se calculează prin intermediul unei matrici intermediare de măsură și definește caracteristicile ca funcții ale acestei matrici intermediare.

Fiind dată o imagine f cu un număr discret de nivele N , vom defini pentru fiecare mulțime de valori discrete ale lui d și θ matricea intermediară $S(d, \theta)$ astfel că:

$S(i, j | d, \theta)$, o intrare în matrice, este numărul de ori a nivelului de gri i ce este în următoarea relație cu nivelul de gri j :

$$f(x) = i \text{ și } f(y) = j \text{ astfel încât} \quad (5.88.)$$

$$y = x + (d \cos \theta, d \sin \theta) \quad (5.89.)$$

Știind că:

$$S(d, \theta) = S(d, \theta + \pi) \quad (5.90.)$$

În practică se restricționează θ la multipli $\pi/4$. Vom redefini:

$$S(d, \theta) = \frac{1}{2}[S(d, \theta) + S(d, \theta + \pi)] \quad (5.91.)$$

Cele mai des întâlnite caracteristici folosite în practică sunt:

$$\text{Energia: } E(d, \theta) = \sum_{i=0}^K \sum_{j=0}^K [S(i, j|d, \theta)]^2 \quad (5.92.)$$

$$\text{Entropia: } H(d, \theta) = \sum_{i=0}^K \sum_{j=0}^K S(i, j|d, \theta) \log S(i, j|d, \theta) \quad (5.93.)$$

$$\text{Corelația: } C(d, \theta) = \frac{\sum_{i=0}^K \sum_{j=0}^K (i - \mu_x)(j - \mu_y) S(i, j|d, \theta)}{\sigma_x \sigma_y} \quad (5.94.)$$

$$\text{Inerția: } I(d, \theta) = \sum_{i=0}^K \sum_{j=0}^K (i - j)^2 S(i, j|d, \theta) \quad (5.95.)$$

$$\text{Omogenitatea locală: } L(d, \theta) = \sum_{i=0}^K \sum_{j=0}^K \frac{1}{1 + (i - j)^2} S(i, j|d, \theta) \quad (5.96.)$$

unde, $S(i, j|d, \theta)$ este elementul (i, j) al matricii $S(d, \theta)$, și:

$$\begin{aligned} \mu_x &= \sum_{i=0}^K i \sum_{j=0}^K S(i, j|d, \theta) \\ \mu_y &= \sum_{i=0}^K j \sum_{j=0}^K S(i, j|d, \theta) \\ \sigma_x^2 &= \sum_{i=0}^K (i - \mu_x)^2 \sum_{j=0}^K S(i, j|d, \theta) \\ \sigma_y^2 &= \sum_{i=0}^K (j - \mu_j)^2 \sum_{j=0}^K S(i, j|d, \theta) \end{aligned} \quad (5.97.)$$

Un aspect important, ce rezultă din relațiile de mai sus, constă în faptul că nu se obțin caracteristici ale texturii, care să fie corelate cu modul de percepere al ochiului uman. De exemplu, nici una din relații nu ne dă o informație valorică asupra gradului de “netezime” sau “rugozitate” al texturii. De asemenea, foarte greu de definit în termenii SGLD este noțiunea de gradient al texturii.

5.2.6. Energia texturii

Energia texturii în domeniul frecvență - dacă o textură este direcțional sau spațial periodică, puterea spectrului acesteia va avea vârfuri ce corespund frecvențelor spațiale [143]. Aceste vârfuri pot forma o bază a caracteristicilor unei primitive, ce permite recunoașterea acesteia. O altă cale este de a utiliza coordonate polare. Dacă F este transformata Fourier, puterea spectrului este dată de $|F|^2$. Caracteristicile radiale se exprimă în astfel :

$$V_{r_1, r_2} = \iint |F(u, v)|^2 dudv \quad (5.98.)$$

iar limitele de integrare se definesc astfel :

$$\begin{aligned} r_1^2 &\leq u^2 + v^2 < r_2^2 \\ 0 &\leq u, v < n-1 \end{aligned}$$

Unde,

- $[r_1, r_2]$ este zona dintre cele două cercuri

- V este un vector definit de diverse valori ale lui r_1 și r_2 .

Caracteristicile radiale sunt corelate cu gradul de netezime al texturii. O textură netedă va avea valori mari pentru raze mici, pe când texturile cu asperități, vor avea valori ale vârfurilor mai mari pentru raze largi.

Caracteristica ce măsoară orientarea unghiulară este dată de relația :

$$V_{\theta_1, \theta_2} = \iint |F(u, v)|^2 dudv \tag{5.99.}$$

iar limitele de integrare se definesc astfel :

$$\theta_1 \leq \tan^{-1}\left(\frac{v}{u}\right) < \theta_2$$

$$0 < u, v \leq n - 1$$

unde,

- $[\theta_1, \theta_2]$ este un sector de cerc;
- V este un vector definit de diverse valori ale lui θ_1, θ_2 .

Această caracteristică exploatează sensibilitatea puterii spectrale la direcționalitatea texturii. Dacă o textură are mai multe linii sau muchii după o direcție θ , $|F|^2$ va avea valori mai mari ce se grupează după direcția $\theta + \pi/2$ în spațiul frecvențelor.

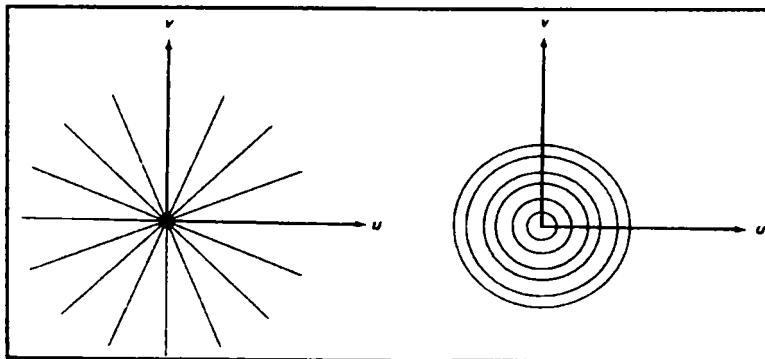


Figura 5-18 Partiționarea domeniului Fourier

Energia texturilor în domeniul spațial - transformata Fourier este folosită și în prelucrarea imaginilor. În 1980 Laws [144] a propus o bază de funcții, ce nu este o bază Fourier, dar care ne ajută să intuim caracteristicile texturilor. În continuare, sunt prezentate cele mai importante 4 funcții ale bazei, ce se folosesc în practică.

$$\begin{pmatrix} -1 & -4 & -6 & -4 & -1 \\ -2 & -8 & -12 & -8 & -2 \\ 0 & 0 & 0 & 0 & 0 \\ 2 & 8 & 12 & 8 & 2 \\ 1 & 4 & 6 & 4 & 1 \end{pmatrix}$$

$$\begin{pmatrix} 1 & -4 & 6 & -4 & 1 \\ -4 & 16 & -24 & 16 & -4 \\ 6 & -24 & 36 & -24 & 6 \\ -4 & 16 & -24 & 16 & -4 \\ 1 & -4 & 6 & -4 & 1 \end{pmatrix}$$

$$\begin{pmatrix} -1 & 0 & 2 & 0 & -1 \\ -2 & 0 & 4 & 0 & -2 \\ 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & -4 & 0 & 2 \\ 1 & 0 & -2 & 0 & 1 \end{pmatrix}$$

$$\begin{pmatrix} -1 & 0 & 2 & 0 & -1 \\ -4 & 0 & 8 & 0 & -4 \\ -6 & 0 & 12 & 0 & -6 \\ -4 & 0 & 8 & 0 & -4 \\ -1 & 0 & 2 & 0 & -1 \end{pmatrix}$$

Imaginea suferă o primă transformare prin egalizarea histogramei. Urmează obținerea celor 12 noi imagini prin convoluția imaginii principale cu cele 12 funcții ale bazei.

$$f_k = f * h_k \quad (5.100.)$$

unde, h_k este una din funcțiile bazei.

Deci fiecare imagine va reprezenta o "energie" a imaginii prin realizarea unor operații de convoluție: fiecare pixel al noii imagini va fi media valorilor absolute dintr-o fereastră de 15x15 pixeli centrată pe pixelul analizat.

$$f_k''(x, y) = \sum_{\substack{x', y' \\ \text{din fereastra}}} (|f_k'(x', y')|) \quad (5.101.)$$

Transformarea $f \rightarrow f_k''$ cu $k = 1 \dots 12$ este denumită și transformata de energie a texturii prin legile lui Laws ceea ce este analog cu puterea spectrală Fourier.

În figura de mai jos, sunt reprezentate mai multe texturi Brodatz ce au fost folosite de Laws în experimente. Se remarcă o bună clasificare în opt categorii.

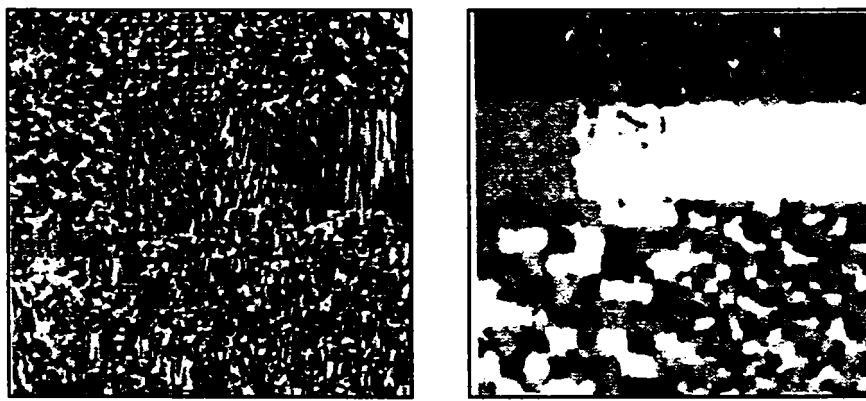


Figura 5-19 Segmentarea imaginii folosind funcții Law

5.2.7. Filtrele Gabor

Discriminarea texturilor în cadrul bazelor de date sau segmentarea acestora în cazuri de prelucrări de imagini, presupune o acuratețe și o abordare strâns legată de capacitatea robustă a ochiului uman de interpretare a texturilor și de modul de realizare a segmentării imaginilor bazată pe texturi.

Au fost propuse mai multe metode de interpretare și de segmentare a texturii, unele dintre ele se bazează pe o bancă de modele de filtre [145].

Metoda experimentată de autor, care va fi expusă în continuare, se bazează pe un set de filtre liniare ce operează în paralel în descompunerea imaginii într-o colecție de subimagini. Aceste subimagini vor fi combinate folosind una sau mai multe mecanisme neliniare pentru generarea imaginii segmentate.

Filtrele individuale sunt proiectate astfel ca acțiunea lor simultană să se concentreze asupra unui domeniu de frecvențe și asupra unor interacțiuni spațiale locale [146], [147], [148], [149]. Acesta este motivul pentru care se încearcă o descompunere spațiu-frecvență unitară. Analiza Fourier a descompunerilor unitare spațiu-frecvență permite extragerea modificărilor locale ale frecvenței, modificări ce depind și de poziția în imagine.

Utilizând diferite clase de operatori putem realiza o bună descompunere unitară spațiu-frecvență, una din clasele cele mai atractive este clasa funcțiilor elementare Gabor (GEF) [150], [151]. Aceste funcții sunt localizate optimal de principiul incertitudinii din domeniul frecvență și

din domeniul spațial (optimizarea incertitudinii se bazează pe metrica $\Delta x \cdot \Delta \omega$), ele putând fi foarte selective atât în domeniul frecvență cât și în domeniul spațial [152]. GEF sunt filtre trec-banda și refac caracteristicile celulelor vizuale cervicale.

Utilizarea de bănci de filtre determină o îngreunare a calculelor. Dacă o imagine de intrare conține două regiuni cu texturi diferite, atunci diferențele locale spațiale și de frecvență dintre cele două regiuni vor produce diferențe la ieșire pentru unul sau mai multe filtre. Deci, diferența dintre texturi este transformată în discontinuități ale imaginii de ieșire, atunci când există diferențe de textură între regiunile luate în discuție. Discontinuitățile pot fi folosite în continuare în procesare pentru partiția imaginii în diferite regiuni. Pentru scheme ce folosesc bănci de filtre GEF, fiecare filtru constă în parametrizarea GEF cu o neliniaritate. Neliniaritatea implică calcule foarte complexe, filtrele folosite sunt filtre Gabor.

Primele etape în proiectarea filtrelor au presupus :

- calcularea transformatei Fourier pentru textura în cauză și determinarea celor mai diferite frecvențe;
- folosirea unei căutări euristice pentru studiul sistemului vizual uman;
- realizarea unei decompoziții spectrale pe prototipuri de "texels" pentru fiecare textură de interes și notarea celor mai mari diferențe ce apar;

Dintre pionierii acestui domeniu îi putem aminti pe: A. Bovik și D.F. Dunn, ei au demonstrat că folosirea unui model matematic de definire a texturii și aplicarea filtrelor Gabor poate determina la ieșirea acestora un semnal *semnătură* ce corespunde conturului dintre două texturi diferite. Ei identifică patru tipuri de semnături: treaptă, vale, creastă, treaptă. Aceste semnături apar numai dacă parametrii de definire ai filtrului Gabor sunt bine aleși.

Pentru texturi naturale ce au un larg domeniu de variație, este nevoie de o metodă sistematică de selectare a parametrilor filtrului. Pentru a obține o discriminare mai robustă pentru diferite perechi de textură, se poate propune o schemă de segmentare ce utilizează un filtru multiplu, motivația fiind dată de modelul Rician (descriș într-un paragraf următor). Problema ce rămâne în continuare este cum combinăm acele filtre.

O problemă ce trebuie luată în considerare în cazul segmentării imaginilor este *supervizarea* segmentării, care nu se poate face în general automat atunci când imaginea ce este segmentată este compusă din foarte multe texturi, ea presupunând existența unei bănci mari de filtre sau a operatorului uman ce trebuie să indice eșantionul de textură pentru care se vor calcula parametrii filtrului.

Funcțiile elementare Gabor și filtrele Gabor

Voi începe cu o scurtă recapitulare a GEF și a filtrelor Gabor [153], [154]. O funcție elementară Gabor bidimensională este definită de către J. Daugman în următorul mod :

$$h(x, y) = g(x', y') \exp[j 2\pi(Ux + Vy)] \quad (5.102.)$$

$$(x', y') = (x \cos \theta + y \sin \theta, -x \sin \theta + y \cos \theta)$$

unde relația de mai sus reprezintă coordonatele rectilinii ale domeniului spațial rotite.

Fie (u, v) coordonatele rectilinii ale domeniului frecvență, iar (U, V) o frecvență anume din acest plan. Exponențiala complexă este o sinusoidă bidimensională de frecvență:

$$F = \sqrt{U^2 + V^2} \quad \text{iar} \quad \phi = \tan^{-1}(V/U) \quad (5.103.)$$

unde, ϕ reprezintă orientarea sinusoidii. Funcția $g(x, y)$ este funcția Gauss bidimensională:

$$g(x, y) = \frac{1}{2\pi \sigma_x \sigma_y} \exp \left\{ -\frac{1}{2} \left[\left(\frac{x}{\sigma_x} \right)^2 + \left(\frac{y}{\sigma_y} \right)^2 \right] \right\} \quad (5.104.)$$

unde, σ_x și σ_y caracterizează spațiul existent și banda filtrului. Atunci GEF este o Gaussiană ce este modulată de o sinusoidă complexă. Se poate arăta că transformata Fourier a lui $h(x, y)$ este

$$H(u, v) = \exp \left\{ -\frac{1}{2} \left[(\sigma_x (u - U)')^2 + (\sigma_y (v - V)')^2 \right] \right\} \quad (5.105.)$$

unde, $[(u - U)', (v - V)'] = [(u - U) \cos \theta + (v - V) \sin \theta, -(u - U) \sin \theta + (v - V) \cos \theta]$

Din relația (5.104) rezultă că avem pentru GEF un răspuns în frecvență de formă Gaussiană (lățimea este dată de σ_x și σ_y , iar unghiul de rotație de θ , centrarea fiind făcută pe frecvența (U, V)), de fapt GEF este un filtru trece bandă.

În cele mai multe cazuri se alege $\sigma_x = \sigma_y = \sigma$ și se consideră ca fiind o alegere rezonabilă din punct de vedere al proiectării. Întotdeauna, când se găsește o textură ce nu conține o aranjare a texel-ilor după un caroiaj pătratic, sunt mai utile filtrele asimetrice pentru care $\sigma_x \neq \sigma_y$.

În continuare, dacă vom presupune că: $\sigma_x = \sigma_y = \sigma$, parametrul θ nu este folosit și deci vom avea o formă simplificată:

$$h(x, y) = \frac{1}{2\pi \sigma^2} \exp \left\{ -\frac{(x^2 + y^2)}{2\sigma^2} \right\} \exp[j 2\pi(Ux + Vy)] \quad (5.106.)$$

Vom defini acum filtrul Gabor O_h în următorul mod :

$$m(x, y) = O_h(i(x, y)) = |i(x, y) * h(x, y)| \quad (5.107.)$$

unde, i este o imagine și m este ieșire acesteia.

Algoritm de proiectare al unui filtru optimal

Filtrul Gabor este determinat dacă se cunosc parametrii U , V și σ din relațiile (5.106). Cel mai bun filtru trebuie găsit din spațiul tuturor filtrelor Gabor posibile, determinat de U , V și σ . Pentru aceasta este nevoie de o metrică pentru a fixa calitatea filtrului. În cazul semnăturii de tip treaptă, calitatea filtrului Gabor este dată de panta și amplitudinea treptei produse. Din analize inițiale și experiențe a rezultat că imaginea de ieșire a filtrului Gabor $m(x, y)$ devine o treaptă ideală numai în anumite cazuri. Deoarece o structură aleatoare este inerentă în texturile tipice, treapta este însoțită de variații locale considerabile. Atunci folosirea directă a pantei și a amplitudinii treptei, ca unitate de măsură, este nefezabilă.

Ca și alternativă, se poate vedea segmentarea texturii ca și o problemă de detecție a saltului. Pentru un filtru particular definit prin U , V și σ se poate estima probabilitatea de asignare greșită a pixelilor din imagine la o anumită regiune. Această probabilitate de eroare este o posibilitate de măsurare a calității filtrului Gabor (U, V, σ) . De aici a rezultat ideea algoritmului de construcție a filtrului.

Algoritmul de construcție a filtrului presupune următoarele :

- eșantioanele prototip pentru texturile de interes A și B sunt cunoscute;
- ieșirile filtrului Gabor pentru texturile A și B pot fi modelate ca și variabile aleatoare independente.

Algoritmul de proiectare a filtrului

1. Se selectează valoarea lui σ în concordanță cu structura texturii.

2. Se aplică simultan un mare număr de filtre Gabor din vecinătatea (U, V, σ) pentru un set aleator de puncte din fiecare eșantion de textură. Mai explicit, pentru fiecare textură A și B avem:

a) se selectează aleator un set s de puncte dintr-o textură;

b) pentru fiecare punct $(X, Y) \in s$, calculăm :

$$|F_{X,Y}(U,V)|$$

unde,

$$F_{X,Y}(U,V) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} i(x,y)g(x-X, y-Y) \times \exp[-j2\pi(Ux+Vy)] dx dy \quad (5.108.)$$

iar, g este Gaussiană:

$$g(x,y) = \frac{1}{2\pi\sigma_x\sigma_y} \exp\left\{-\frac{1}{2}\left[\left(\frac{x}{\sigma_x}\right)^2 + \left(\frac{y}{\sigma_y}\right)^2\right]\right\}$$

având parametrii $\sigma_x = \sigma_y = \sigma$, iar i este eșantionul dat de textură. F este transformata Fourier a ferestrei pentru i centrat în (X, Y) și g funcția fereastră.

3. Calcularea calității filtrului prin măsurarea pentru fiecare candidat de filtru Gabor (U, V, σ) a următoarelor elemente:

a) Pentru fiecare textură și pentru fiecare pereche (U, V) , se calculează:

$$\bar{\mu}(U,V) = \frac{\sum_s |F_{X,Y}(U,V)|}{cards(s)} \quad (5.109.)$$

$$\bar{\sigma}^2(U,V) = \frac{\sum_s \left(|F_{X,Y}(U,V)| - \bar{\mu}(U,V)\right)^2}{cards(s)}$$

unde, $\bar{\mu}(U,V)$ și $\bar{\sigma}^2(U,V)$ sunt media și varianța eșantionului de textură pentru valorile lui F mediate peste toate punctele $(U,V) \in s$.

b) Pentru fiecare (U, V) , se calculează $P_E(U, V)$, probabilitatea ca punctul din imagine să fie incorect clasificat dacă se utilizează filtrul Gabor (U, V, σ) .

4. Ordinul de mărime al probabilității $P_E(U, V)$ ne dă o indicație efectivă despre filtrul Gabor, în legătură cu utilitatea acestuia atunci când se segmentează o imagine ce conține texturile A și B .

Pașul 2 al metodei presupune aplicarea în paralel a familiei de filtre Gabor pentru un punct din imagine. Acesta este un caz special al aplicării transformatei Fourier cu fereastră:

$$F_{X,Y}(U,V) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} i(x,y)w(x-X, y-Y) \times \exp[-j2\pi(Ux+Vy)] dx dy \quad (5.110.)$$

unde, w este funcția fereastră, i este imaginea ce va fi transformată, iar F este o funcție ce depinde atât de poziția ferestrei (X, Y) , cât și de frecvență (U, V) .

Fie p rezultatul convoluției dintre imaginea i și funcția de transfer h a GEF:

$$p(x,y) = i(x,y) * h(x,y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} i(\alpha, \beta) h(x-\alpha, y-\beta) d\alpha d\beta$$

Expandând funcția h și considerând un punct specific al convoluției (X, Y) , obținem:

$$p(X, Y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} i(\alpha, \beta) g(X - \alpha, Y - \beta) \times \exp[-j2\pi(U(X - \alpha) + V(Y - \beta))] d\alpha d\beta$$

După rearanjarea termenilor și luând funcția fereastră $w(x, y) = g(-x, -y)$ obținem :

$$p(X, Y) = K \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} i(\alpha, \beta) w(\alpha - X, \beta - Y) \times \exp[-j2\pi(U\alpha + V\beta)] d\alpha d\beta \quad (5.111.)$$

unde, $K = \exp[j2\pi(UX + VY)]$.

$$|p(X, Y)| = |F_{X, Y}(U, V)|$$

Dezvoltarea de mai sus se bazează pe funcții continue. Pentru cazul discret, transformata Fourier cu fereastră este aproximată la o transformată Fourier discretă. Atunci, (X, Y) se referă la pixelii din imagine, iar (U, V) la spațiul discret al frecvențelor. Pentru implementarea DFT, o imagine este multiplicată cu o funcție Gauss trunchiată axată pe pixelul (X, Y) . Amplitudinea transformatei Fourier discrete aproximează aplicarea unei familii de filtre Gabor unei imagini pentru pixelul (X, Y) , unde fiecare filtru este centrat pe o frecvență ce corespunde și uneia din frecvențele transformatei Fourier discrete. Atunci calculul unei singure transformate Fourier discrete cu fereastră este echivalent cu a determina ieșirea pentru o familie de filtre Gabor într-un singur punct, unde este centrul frecvențelor filtrului în domeniul de frecvență al imaginii. Este clar că algoritmul nostru de selecție nu trebuie să considere toate centrele de frecvență posibile pentru un rezultat formidabil. Experiența arată că acest fapt nu este necesar deoarece pasul de degradare crește foarte fin atunci când se modifică centrul frecvenței în filtrul Gabor ; filtrul Gabor ce are centrul frecvenței $(U + \delta U, V + \delta V)$ tinde spre același rezultat ca și în cazul când centrul frecvenței filtrului este (U, V) .

Modelul Rician

În continuare vom justifica faptul că ieșirea unui filtru Gabor poate fi modelată ca și o variabilă aleatoare Rician [155], [156]. Pentru această analiză, vom considera modelul texturii ca o colecție de similarități, dar nu neapărat identice, primitive geometrice denumite și "texels". Atunci când texelii sunt identici și se regăsesc într-o structură ordonată, repetându-se după o anumită regulă vom avea o textură uniformă, altfel ea se va numi neuniformă. Se poate spune că acest tip de model al texturii este mult prea diferit de cazurile practice în care există nerespectări ale regulilor de aranjare a texelilor, sau anumite deformări și zgomote ce se pot suprapune peste textura ideală. De aceea, este necesară includerea în structura modelului a unei componente care să reprezinte tocmai acest zgomot. Multe texturi conțin efecte aleatoare semnificative, de aceea modelele structurale nu pot prevedea aceste modificări aleatoare, dar poate în schimb să le introducă în calcule ca și un zgomot alb.

Pentru texturi uniforme se poate arăta că aplicarea unor GEF configurate adecvat poate produce la ieșire un semnal sinusoidal. În cazul texturilor neuniforme ce conțin perturbații ale texelilor, la ieșire vom avea suma unui semnal sinusoidal cu un zgomot Gaussian. Vom putea arăta în continuare că aplicarea unui filtru Gabor unei texturi neuniforme produce la ieșire tocmai distribuția funcției de probabilitate Rician

Simplificarea expunerii se poate face utilizând un caz unidimensional. În cazul unidimensional, aplicarea unui filtru Gabor unei imagini $i(x)$ poate fi reprezentată astfel:

$$m(x) = |i(x) * h(x)|$$

unde, $h(x)$ este o funcție Gabor elementară iar " * " reprezintă operația de convoluție. O funcție elementară Gabor poate fi scrisă astfel:

$$h(x) = g(x)e^{j\omega x}$$

unde, $g(x)$ este o Gaussiană iar ω este frecvența. Descompunând funcția h în parte reală și parte imaginară obținem :

$$h(x) = h_r(x) + jh_i(x)$$

unde, $h_r(x) = g(x)\cos(\omega x)$ și $h_i(x) = g(x)\sin(\omega x)$.

Atunci ieșirea filtrului Gabor are expresia :

$$m(x) = |A(x) + jB(x)| \quad (5.112.)$$

unde, $A(x) = i(x) * h_r(x)$ și $B(x) = i(x) * h_i(x)$ (5.113.)

Considerăm forma $A(x)$ atunci când i este o textură uniformă, adică texelii sunt identici și aranjați periodic. Atunci în domeniul frecvență, $I(\omega)$, transformata Fourier a lui i , constă într-o grupare periodică de impulsuri. h_r este modulată Gaussian de un cosinus. Deci, transformata Fourier a lui h_r restabilește o pereche Gaussiană centrată pe frecvențele $\pm\omega$ și aproximează un filtru trece bandă. Dacă ω coincide cu locația unuia dintre impulsurile cuprinse în $I(\omega)$, iar parametrul σ este ales astfel ca h_r să fie aproximabil cu o bandă îngustă, aplicând h_r lui i , obținem o pereche de impulsuri la frecvențele $\pm\omega$. În domeniul spațial, acestea corespund unui cosinus de ω . $A(x)$ poate fi exprimat sub forma:

$$A(x) \approx A_o \cos(\omega x + \Theta) \quad (5.114.)$$

unde, Θ este o fază arbitrară. O regiune mărginită produce la ieșire un cosinus trunchiat. Acest efect este nesemnificativ atunci când regiunea este relativ largă în raport cu dimensiunea efectivă a operatorului de filtrare Gabor.

Acum putem modela o textură neuniformă ca și o textură uniformă a cărei texeli au fost perturbați aleator. Într-un spațiu bidimensional variațiile aleatoare corespund modificărilor de poziție, formă și orientare. Vom presupune că texelii sunt independenți.

Dacă vom partiționa convoluția în regiuni disjuncte $[x_k, x_{k+1}]$ ocupate de un singur texel, vom avea relațiile:

$$A(x_l) = \sum_{k=-\infty}^{\infty} r_k ; \quad r_k = \int_{x_k}^{x_{k+1}} i(z)h_r(x_l - z) dz \quad (5.115.)$$

De notat că în spațiul 2-D, r_k este determinat de poziția, forma și orientarea texelului k . Știind că proprietățile texelului sunt aleatoare, va rezulta că r_k este o valoare aleatoare. Pe de altă parte deoarece s-a presupus că texelii sunt independenți, rezultă deci că și r_k este o variabilă aleatoare independentă. Teorema limitei centrale ne spune că $A(x_l)$ este Gaussiană și poate fi scrisă sub următoarea formă:

$$A(x_l) = \hat{A}_{x_l} + \delta_{x_l}$$

unde, \hat{A}_{x_l} este o constantă ce reprezintă valoarea prezisă pentru A_{x_l} și δ_{x_l} este o variabilă aleatoare ce are o distribuție normală $N_{x_l}(0, \hat{\sigma}^2)$. Deci ne sugerează că $A(x)$ poate fi modelat astfel:

$$A(x) = \hat{A}(x) + N(x) \quad (5.116.)$$

unde, $\hat{A}(x) = A_o \cos(\omega x + \Theta)$ este ieșirea funcției elementare Gabor pentru o textură neperturbată (uniformă) iar $N(x)$ este un proces aleator Gaussian, staționar în sensul larg, de medie nulă și de putere $E[N^2(x)] = \hat{\sigma}^2$.

Se poate afirma, că aplicarea unui filtru Gabor unei texturi dintr-o imagine produce la ieșire $m(x)$ ce are funcția de distribuție a probabilității $f_m(m)$ de tip Rician. Aceasta este :

$$f_m(m) = \frac{m}{b^2} \exp\left[-\frac{(a^2 + m^2)}{2b^2}\right] I_0\left(\frac{am}{b^2}\right) u(m) \quad (5.117.)$$

unde, I_o este funcția Bessel modificată de ordinul o , u unitatea de pas, $a \geq 0$ și $b > 0$ [157]. Este bine de știut că anvelopa unui semnal sinusoidal la care se însumează un zgomot Gaussian are o funcție de distribuție a probabilității de tip Rician. Vom arăta acum că ieșirea unui filtru Gabor este egală cu anvelopa lui $A(x)$ ce este de tip Rician.

Deoarece $A(x)$ este ieșirea unui filtru trece bandă, $N(x)$ poate fi reprezentat astfel:

$$N(x) = X(x)\cos(\omega x) - Y(x)\sin(\omega x)$$

unde, $X(x)$ și $Y(x)$ sunt procese trece-jos aleatoare Gaussiene, independente și de medie nulă (X, Y nu au nici o legătură cu punctele din eșantion (X, Y)). Înlocuim pentru $\hat{A}(x)$ dat:

$$\begin{aligned} A(x) &= [A_0 \cos(\Theta) + X(x)]\cos(\omega x) - [A_0 \sin(\Theta) + Y(x)]\sin(\omega x) = \\ &= R(x)\cos(\omega x + \Phi(x)) \end{aligned} \quad (5.118.)$$

$$\text{unde : } R(x) = \left[(A_0 \cos(\Theta) + X(x))^2 + (A_0 \sin(\Theta) + Y(x))^2 \right]^{\frac{1}{2}} \quad (5.119.)$$

este o anvelopă a amplitudinii ce are o variație lentă iar,

$$\Phi(x) = \tan^{-1} \left[\frac{(A_0 \sin(\Theta) + Y(x))}{(A_0 \cos(\Theta) + X(x))} \right] \quad (5.120.)$$

este o fază ce are o variație lentă. Este bine de știut că și $R(x)$ este de tip Rician. Ieșirea filtrului Gabor poate fi rescrisă în următoarea formă:

$$\begin{aligned} m(x) &= |A(x) + jB(x)| = |s(x)e^{j\omega x}| = \\ &= \left| [s_r(x)\cos(\omega x) - s_i(x)\sin(\omega x)] + j[s_i(x)\cos(\omega x) + s_r(x)\sin(\omega x)] \right| \end{aligned} \quad (5.121.)$$

unde, $s(x) = s_r(x) + js_i(x)$

Avem pentru $A(x)$ următorul mod de scriere:

$$A(x) = s_r(x)\cos(\omega x) - s_i(x)\sin(\omega x)$$

unde, $s_r(x) = A_0 \cos(\Theta) + X(x)$

$$s_i(x) = A_0 \sin(\Theta) + Y(x)$$

Dar,

$$m(x) = |s(x)e^{j\omega x}| = |s(x)| = \sqrt{s_r^2(x) + s_i^2(x)} = R(x)$$

Știind că $R(x)$ este de tip Rician rezultă că și $m(x)$ este de același tip, pentru orice x .

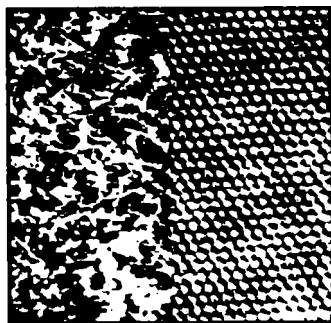


Figura 5-20 Imagine ce conține texturile Brodatz $d9$ și $d77$

În figurile următoare se pot vedea exemple de funcții de distribuție a probabilităților de tip Rician pentru texturile $d9$ și $d77$:

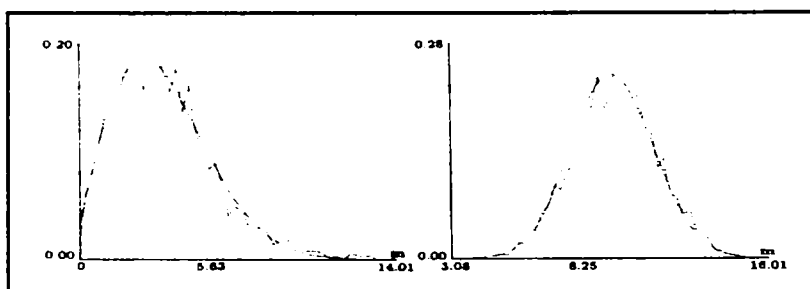


Figura 5-21 Funcțiile de distribuție Rician pentru texturile Brodatz $d9$ și $d77$

În funcție de parametrii a și b , funcția de distribuție a probabilităților de tip Rician se poate modifica de la o funcție de distribuție Rayleigh, la una Gaussiană, după cum se vede din figura următoare:

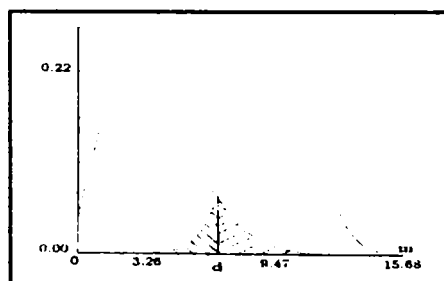


Figura 5-22 Modificarea distribuție Rician în distribuție Gaussiană

Măsurarea calității filtrului - Pasul trei al algoritmului de proiectare folosește o măsurare a calității filtrului bazată pe o aproximație dată de o decizie teoretică. Această măsurare este o formă de indicare a probabilității de asignare incorectă a unor pixeli din imagine unei anumite regiuni (vecinătăți). În continuare autorul va dezvolta metoda utilizată pentru calcul a probabilității (metodă de determinare a parametrilor ce definesc funcția de distribuție a probabilității).

Fie A și B texturile de interes și O_h filtrul Gabor cu frecvența centrală (U, V) . Fie p_A și p_B funcțiile de distribuție a probabilității de tip Rician ce descriu ieșirea filtrului Gabor cu O_h când se aplică A și B . Vom realiza următorul experiment: aplicăm O_h unei texturi necunoscute (A sau B) și înregistrăm în variabila aleatoare z ieșirea m considerată pentru un punct aleator (x, y) . Deci problema se rezumă la a decide dacă punctul (x, y) aparține texturii A (ipoteza H_0) sau texturii B (ipoteza H_1). Vom defini un punct de decizie d , astfel, dacă $z < d$, atunci

eșantionul aparține regiunii A (acceptăm H_0), în caz contrar el este presupus din regiunea B (acceptăm H_1). Se poate scrie și în următorul mod:

$$\begin{cases} (x, y) \in A, & \text{dacă } z < d \\ (x, y) \in B, & \text{în rest} \end{cases} \quad (5.122.)$$

Pentru acest experiment sunt posibile două erori: acceptăm H_1 când este adevărat H_0 (și avem primul tip de eroare P_I) sau acceptăm H_0 când este adevărat H_1 (și avem tipul doi de eroare P_{II}). Probabilitatea totală de eroare este:

$$P_E = P_I(U, V) + P_{II}(U, V) = \int_d^{\infty} p_A(x) dx + \int_{-\infty}^d p_B(x) dx \quad (5.123.)$$

Dacă P_E devine mic, probabilitatea de clasificare greșită a pixelilor va fi mică, deci P_E este un indicator rezonabil al calității filtrului Gabor. De menționat că în pasul trei al algoritmului de proiectare al filtrului, P_E este calculată pentru fiecare frecvență centrală de interes (U, V) a filtrului Gabor. Deci, P_E depinde de (U, V) , iar scopul nostru este de a minimiza valoarea acesteia.

Soluționarea acestei probleme implică găsirea punctului de decizie d astfel ca probabilitatea raportului

$$\lambda(z) = \frac{p_A(z)}{p_B(z)} \quad \text{să satisfacă} \quad \lambda(d) = \frac{P_o}{1 - P_o}$$

unde, P_o este probabilitatea anterioară ca regiunea să fie A .

Dacă se presupune că cele două regiuni au aceeași arie și probabilitatea lor de apariție este aceeași, atunci:

$$P_o = 1 - P_o = 1/2, \quad \text{și} \quad \lambda(d) = 1.$$

Problema se reduce la găsirea lui d astfel ca $p_A(d) = p_B(d)$.

Știm că $p_A(d)$ și $p_B(d)$ sunt funcții de distribuție a probabilității foarte complicate, o posibilitate de calcul pentru d ar fi abordarea numerică. Parametrii unei funcții de distribuție a probabilității pot fi estimați folosind metoda momentelor. Aceasta presupune rezolvarea următoarelor ecuații:

$$\begin{aligned} \bar{\mu} &= b \sqrt{\frac{\pi}{2}} e^{k^2/4} \left[\left(1 + \frac{k^2}{2} \right) I_0 \left(\frac{k^2}{4} \right) + \frac{k^2}{2} I_1 \left(\frac{k^2}{4} \right) \right] \\ \bar{\sigma}^2 &= b^2 (2 + k^2) - \bar{\mu}^2 \\ k^2 &= \frac{a^2}{b^2} \end{aligned} \quad (5.124.)$$

unde, a și b sunt parametrii funcției de distribuție a probabilității Rician, $\bar{\mu} = \bar{\mu}_A$ sau $\bar{\mu}_B$ este media eșantionului, $\bar{\sigma}^2 = \bar{\sigma}_A^2$ sau $\bar{\sigma}_B^2$ este varianța eșantionului. O dată cunoscuți parametrii pentru p_A și p_B vom putea calcula valoarea lui d , iar apoi să calculăm numeric P_E . Mai multe combinații de tipul textură/filtru vor da un larg răspuns pentru $m(x, y)$. Pentru acest caz când distribuția Rician are o medie largă se poate utiliza o aproximare a distribuției Rician cu Gauss. Această echivalare va simplifica foarte mult calculul lui d și P_E . Luând $p_A(d) = p_B(d)$ vom avea:

$$\frac{1}{\sigma_A \sqrt{2\pi}} \exp \left[\frac{-(d - \mu_A)^2}{2\sigma_A^2} \right] = \frac{1}{\sigma_B \sqrt{2\pi}} \exp \left[\frac{-(d - \mu_B)^2}{2\sigma_B^2} \right]$$

$$d = \frac{(\mu_A \sigma_B^2 - \mu_B \sigma_A^2)}{\sigma_B^2 - \sigma_A^2} \pm \frac{\sigma_B \sigma_A \sqrt{(\mu_B - \mu_A)^2 + 2(\sigma_B^2 - \sigma_A^2) \ln(\sigma_B / \sigma_A)}}{\sigma_B^2 - \sigma_A^2}$$

unde, d este ales astfel ca $\mu_A \leq d \leq \mu_B$.

Vom estima (μ_A, μ_B) prin media eșantionului $(\bar{\mu}_A, \bar{\mu}_B)$ iar (σ_A^2, σ_B^2) prin varianța eșantionului $(\bar{\sigma}_A^2, \bar{\sigma}_B^2)$. Media și varianța eșantionului constituie un estimatori suficient de consistenți. Dacă numărul de eșantioane disponibil tinde spre infinit, eroarea de estimare a lui P_E , folosind media și varianța eșantioanelor, tinde spre zero.

În unele cazuri, aproximarea Gaussiană este neadecvată, de exemplu în cazul în care texturile ce trebuiesc deosebite sunt ambele asemănătoare cu un zgomot de bandă îngustă, caz pentru care răspunsul filtrului Gabor este tipic de valoare mică. Se poate folosi, în unele cazuri, pentru aproximarea funcției de distribuție a probabilității de tip Rician distribuția Rayleigh ce este semnificativ mai asimetrică.

Specificarea parametrilor pentru filtrele Gabor - efectul dispersiei σ în calculul probabilității de eroare $P_E(U, V)$ este următorul: creșterea valorii dispersiei σ determină scăderea probabilității de eroare $P_E(U, V)$ deoarece scade dispersia ieșirii filtrului ($\bar{\sigma}_A^2$ sau $\bar{\sigma}_B^2$). Acest fapt apare deoarece, cu creșterea dimensiunii ferestrei (se lărgeste σ), iar valoarea pentru un punct a transformatei Fourier cu fereastră WFT este determinată de o vecinătate mai mare de pixeli. Extinderea vecinătății determină ca ieșirea WFT să fie mai puțin sensibilă la perturbațiile poziției ferestrei, reducând astfel variațiile ieșirii. Deci se sugerează calcularea lui σ pe o arie cât mai extinsă.

Trebuie să ne reamintim că într-o segmentare reală a texturii problema ce apare este faptul că nu se cunosc dimensiunile texturilor și nici conturul acestora. Dacă σ este luat pentru o arie mai largă, fereastra se poate suprapune semnificativ de mult peste alte regiuni, reducând acuratețea de determinare a conturului texturii. Deci se sugerează utilizarea unui σ calculat pe o arie cât mai mică.

De aici, alegerea lui σ este un compromis între variațiile ieșirii și localizarea conturului. Multe texturi constau în structuri cvasiperiodice de mică dimensiune denumite texeli. Dunn a arătat că pentru aceste texturi un bun compromis este alegerea lui σ de pe o suprafață aproximativ egală cu un texel. În unele cazuri ce vor fi discutate în continuare utilizarea unei arii mai mici pentru calculul lui σ poate fi mai avantajoasă. În practică algoritmul de realizare a filtrului dă în general aceleași rezultate pentru o scară de variație a lui σ suficient de mare. Deci este suficient să rulăm de câteva ori algoritmul, folosind câteva valori diferite pentru σ și să comparăm rezultatele.

Selecția eșantioanelor aleatoare din imagine - algoritmul de selecție al filtrului se face prin aplicarea unei familii de filtre Gabor unui set de puncte aleatoare. Selecția acestor puncte presupune câteva comentarii: în multe cazuri, textura poate fi modelată ca un proces aleator. Când vom eșantiona regiunea cu textură vom avea în eșantion numai o realizare a procesului aleator. În general, acesta este suficient pentru estimarea statistică a procesului. Ca efect este găsirea celui mai bun filtru pentru un eșantion particular al acestei texturi. Există însă și posibilitatea ca același filtru să fie total inefficient pentru o altă realizare (alt eșantion). Multe texturi pot fi modelate ca fiind periodice într-un sens mai larg, unde procesul statistic este unic numai într-o perioadă fixată (între doi texels adiacenți). Știind că pentru un exemplu din textură,

tipic conține mai multe perioade, eșantionarea unui astfel de exemplu de realizare a unui proces aleator acesta poate deveni un eșantion reprezentativ.

Spațiul dintre punctele de eșantionare se regăsește în acuratețea de exprimare a funcției de distribuție a probabilității. După cum s-a arătat, pentru un punct fixat (x, y) ieșirea filtrului Gabor poate fi modelată ca o variabilă aleatoare Rician. Valorile adiacente, întotdeauna sunt puternic corelate ceea ce dă o largă extindere spațială a operatorului de filtrare Gabor O_h . Atunci în realizarea unei eșantionări aleatoare a distribuției trebuie să luăm un spațiu considerabil între eșantioane. Acest lucru poate cauza probleme dacă dimensiunea eșantionului este foarte mare. În urma unor simulări a rezultat că eșantionul ce are o arie aproximativ egală cu $1/3$ din suprafața textelului produce parametri ai distribuției ce sunt similari cu cei determinați pe arii mult mai largi.

Optimizarea discriminării, pentru indexarea bazelor de imagini, în cazul imaginilor ce conțin mai mult de 2 texturi

Segmentarea unei imagini ce conține două texturi A și B , constă în realizarea unei bipartiții. Metoda prezentată pentru realizarea unui filtru ne dă funcția Gabor elementară optimă. Această metodă de segmentare poate fi folosită numai într-un singur pas de filtrare, ea nu este optimă pentru orice pereche de texturi. În continuare vor fi prezentate două metode de segmentare ce utilizează filtre Gabor.

Prima metodă combină rezultatul mai multor filtre Gabor. Se folosește teorema lui Bayes și se presupune că distribuțiile ieșirii pentru filtre diferite sunt statistic independente. Rezultă că asupra celor două texturi de interes se aplică n filtre Gabor. Atunci probabilitatea condiționată a unui pixel k de a aparține unei regiuni $r_i = A$ sau B este dată de relația:

$$P[k \in r_i | m_1, m_2, \dots, m_n] = \frac{P_i}{P_A + P_B} \quad (5.125.)$$

unde, m_j se referă la ieșirea filtrului Gabor j și avem următoarele definiții pentru probabilitățile P_i, P_A, P_B :

$$P_i = \prod_{j=1}^n P[m_j | k \in r_i]$$

$$P_A = \prod_{j=1}^n P[m_j | k \in A]$$

$$P_B = \prod_{j=1}^n P[m_j | k \in B]$$

De notat că: $P[m_j | k \in A]$ și $P[m_j | k \in B]$ corespund funcției de distribuție a probabilității p_A și p_B . Probabilitățile $P[k \in r_i | m_1, m_2, \dots, m_n]$ sunt folosite pentru asignarea pixelilor unor regiuni, obținând o imagine segmentată ce nu mai necesită alte prelucrări. Dealtfel această schemă se aplică numai pentru imagini ce conțin două texturi diferite, dar conceptul poate fi extins. Presupunând modelul Rician, această metodă teoretic este optimă. Pot însă să apară și rezultate suboptimale date de variații ale parametrilor considerați ideali.

Cea de a doua metodă presupune netezirea ieșirii filtrului Gabor cu o Gaussiană de o largă extindere O_g :

$$m'(x, y) = O_g(m(x, y)) = m(x, y) * g'(x, y) = O_g[O_h(i(x, y))] \quad (5.126.)$$

unde, $g'(x, y)$ este o Gaussiană, cu parametrul σ_g .

Este clar că semnătura de tip treaptă este foarte distorsionată, ceea ce va face foarte dificilă segmentarea. Distorsiunile sunt introduse de largă variație a orientărilor textelilor din fiecare regiune cu textură. Dacă structura texturilor este cvasiperiodică ea poate fi modelată ca un proces aleator periodic în sens larg. Dacă acest lucru este adevărat și perioada corespunde spațiului unui texel, atunci primele două momente ale distribuției nivelelor de gri pentru textură sunt periodice :

$$\mu(r) = \mu(r+T) \text{ și } K(r_1, r_2) = K(r_1+T, r_2) = K(r_1, r_2+T)$$

unde, μ este media și K este autocovarianța matricii, iar r este un vector de poziție bidimensional. Deci este de așteptat ca și compoziția locală spațiu-frecvență și ieșirea filtrului Gabor să fie aproximativ periodică (în sens stohastic). Atunci media local spațială a ieșirii unui filtru Gabor pentru o regiune cu textură este aproximativ constantă. Vom putea calcula această medie prin aplicarea unui filtru trece-jos.

Pragul de decizie teoretic nu poate măsura direct calitatea ieșirii decât după cele două procesări. Vom putea întotdeauna determina cel mai bun filtru Gabor pentru primul pas ce asigură o discriminare maximă. Pasul următor este operația de netezire, urmând ca mai apoi să efectuăm operația de detecție de muchii pentru a asigura acuratețe segmentării. Cu alte cuvinte operația de filtrare crește efectiv puterea de discriminare reducând varianța ieșirii.

O tehnică foarte populară se bazează pe o căutare heuristică ce ține cont de proprietățile neurofiziologice și psihologice ale sistemul vizual uman. Ea a fost folosită pentru a testa prototipuri de scheme de segmentare.

O altă metodă presupune o comparare a compoziției spectrale a celor două prototipuri de texturi din regiunile ce urmează a fi segmentate. Se poate arăta însă că pentru simple texturi ieșirea filtrului Gabor sub forma unei semnături de tip treaptă este obținut din diferențele dintre frecvențele corespunzătoare textelilor. Această metodă are două limitări:

- este restricționată de texturi cu texeli distincți;
- o dată cu scăderea dimensiunii textelilor, ei pot începe să interacționeze și să-și piardă propria lor identitate.

Dacă nu sunt respectate aceste restricționări metoda devine inefficientă.

Prioritară în cercetarea acestei probleme a fost găsirea unei metode efective de determinare a parametrilor filtrului Gabor care să se bazeze pe modul relativ ușor de calcul al transformatei Fourier discrete pentru un eșantion din textura regiunii studiate [158], [159], [160]. Componenta de frecvență bidimensională ce diferă cel mai mult între cele două regiuni este selectată ca frecvență centrală a filtrului Gabor (U, V). Aplicarea transformatei Fourier discrete este echivalentă cu aplicarea unei transformate Fourier cu fereastră unei regiuni, unde fereastra este rectangulară și egală cu dimensiunea regiunii. Atunci aplicarea transformatei Fourier discrete este echivalentă cu metoda prezentată când aria acoperită de filtrul Gabor se apropie de dimensiunea regiunii; cu alte cuvinte pentru un calcul al dispersiei făcut pe o arie cât mai extinsă.

În practică trebuie aleasă întotdeauna aria de calcul a dispersiei mai mică decât cea mai mică regiune cu textură din imagine. Din simulări a rezultat că pentru valori mici ale ariei de calcul a lui σ , acest algoritm oferă filtre cu o putere mai mare de discriminare decât metodele ce utilizează transformata Fourier discretă. Discriminarea poate fi măsurată relativ prin compararea probabilității de eroare $P_E(U, V)$ pentru diferite frecvențe centrale. Performanțele relativ mai slabe ale metodelor ce folosesc transformata Fourier discretă rezultă din faptul că ele nu iau în considerare banda de frecvență a filtrului atunci când se determină frecvența centrală.

Se găsesc în literatură mai multe articole ce folosesc în general aceleași principii ca cele de mai sus cu mici variații și îmbunătățiri [161], [162], [163], [164], [165].

5.2.8. Experimentări proprii referitoare la aplicații ale filtrelor Gabor și ale caracteristicilor texturilor

Una din cele mai importante aplicații ale filtrelor Gabor este detecția discontinuităților în cadrul texturilor. Această aplicație are un grad mare de răspândire în cazul proceselor tehnologice din industrie. În continuare, voi prezenta câteva dintre rezultatele propriilor experimentări referitoare la detecția discontinuităților folosind filtre Gabor [166]:

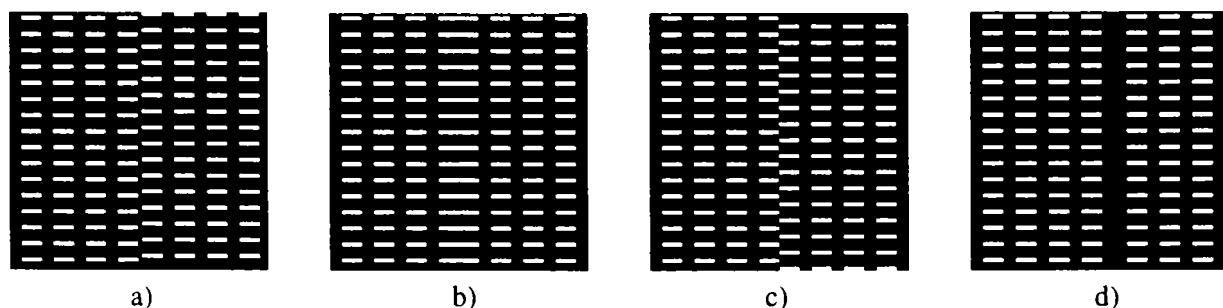


Figura 5-23 Textură cu discontinuități

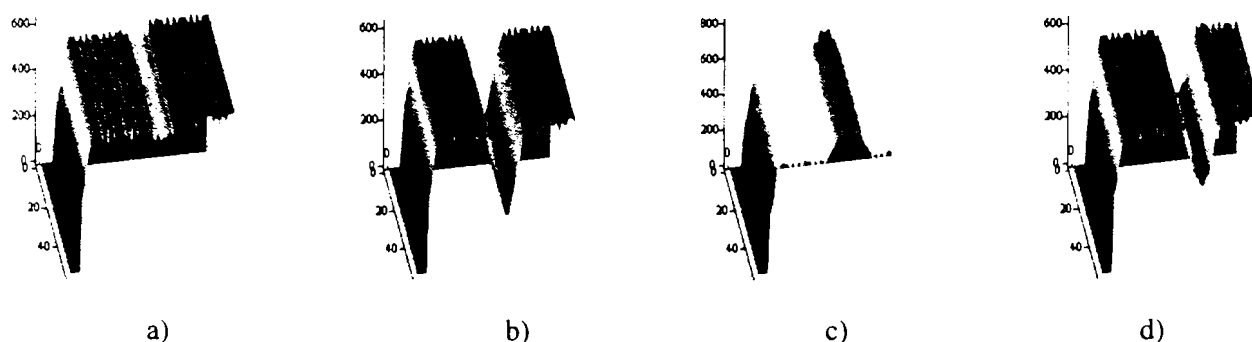


Figura 5-24 Rezultatul filtrării unei texturi cu discontinuități folosind filtre Gabor

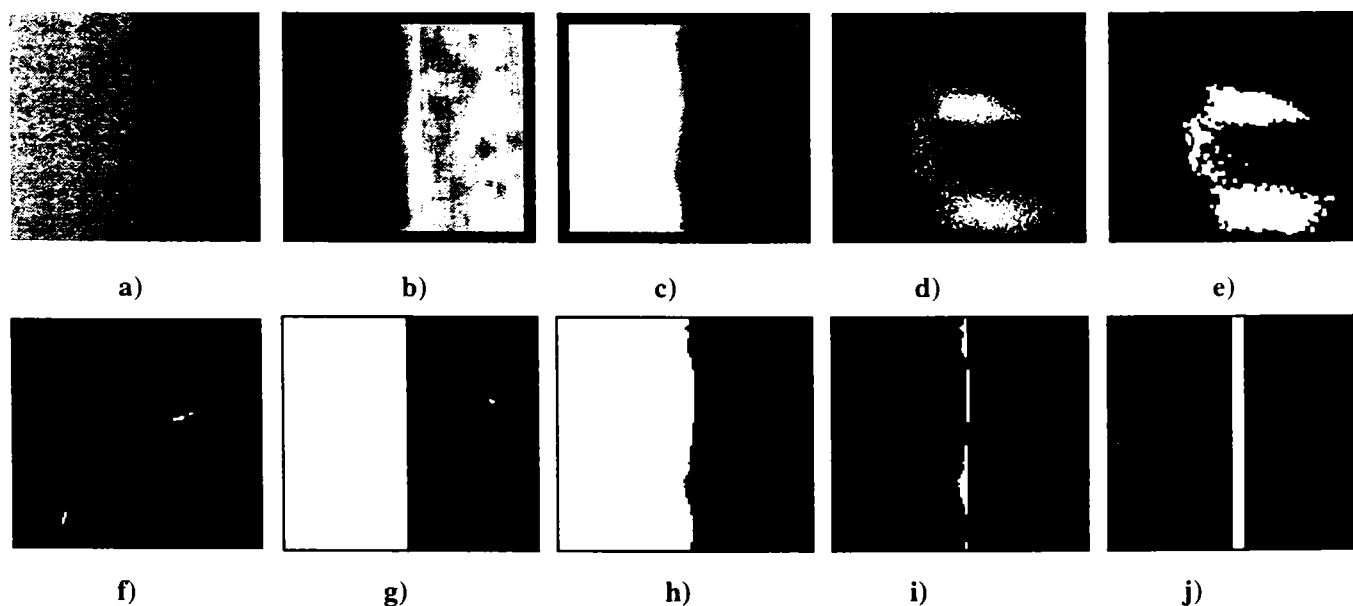


Figura 5-25 Segmentarea imaginilor bazată pe texturi folosind elemente de morfologie:

a) imaginea originală b) contrastul; c) omogenitatea; d) formarea unui spațiu bidimensional din cele două caracteristici ale texturii; e) Rezultatul operației morfologice "open-close"; f) rezultatul eroziunii imaginii din figura e) și formarea centroizilor; g) segmentarea contrastului pe baza pragului obținut din erodare; h) segmentarea omogenității pe baza pragului obținut din erodare; i) eroarea pentru segmentarea contrastului; j) eroarea pentru segmentarea omogenității [167], [168], [169], [170].

5.2.9. Caracteristici ale texturilor invariante la rotație, translație și scalare

Pentru început este bine de amintit că o imagine are un înalt grad de autocorelație. Aceste corelații ne permit să distingem imaginile ce conțin obiecte reale de zgomote și să reușim deosebirea imaginilor una de alta. Textura constituie baza pentru una dintre cele mai cunoscute metode de segmentare [171]. La analiza texturii se utilizează descrierea statistică a acestora (netede, aspre, etc.). O descoperire importantă în acest sens a avut Julesz [172], care a subliniat faptul că sistemul vizual uman este capabil să separe texturile după media lor statistică de ordinul I și chiar de ordinul II, dar nu distinge diferențele pentru un ordin superior. Folosirea funcțiilor Gabor pentru modelarea unei celule simple corticale a fost propusă în 1980 de Marcelja [173] într-o formă unidimensională și de Daugman [174], [175] într-o formă bidimensională. Eficacitatea funcțiilor Gabor include explicit parametrizarea poziției câmpurilor receptive pentru exprimarea bidimensională, orientarea și frecvența spațială. Această descompunere utilizată de aceste funcții Gabor 2-D parametrizate biologic conduce la obținerea unor imagini primitive și utilizarea lor în compunerea imaginilor, analiza texturilor și segmentare. În locul funcțiilor se pot utiliza și alte modele cum ar fi transformata undișoară, polinoamele Hermite, distribuția pseudo-Wigner, estimarea locală a puterii spectrale sau a momentelor complexe.

Pentru realizarea unei indexări care să fie independentă de rotația și scalarea texturii se utilizează caracteristicile *circular-Mellin*. Acestea corespund caracteristicilor Fourier în domeniul polar-logaritmice. Transformarea în coordonate polar-logaritmice este folosită deoarece permite realizarea invarianței față de rotație și scalare. Din studii neuropsihologice, a rezultat că răspunsul unei singure celule *monkey Macaque*, este mult mai sugestiv într-o reprezentare în coordonate polare decât în coordonate cartesiene. Corelatorul de arhitectură este folosit pentru a păstra invarianța la translație. Din moment ce diferite texturi au fost recunoscute, se poate folosi un singur set de caracteristici orientate pentru identificarea tuturor texturilor din imaginea de intrare.

Pentru problema dependenței caracteristicii Gabor de orientare au fost propuse mai multe soluții. Una dintre cele mai populare soluții este utilizarea unor măsuri, ce depind de suma caracteristicilor extrase din funcțiile Gabor la diferite orientări, iar rezultatul sumei fiind invarianța la rotație. Deci, s-ar sugera ideea că în "*spectrul orientării*" al unei texturi analizate s-ar putea ascunde o informație referitoare la rotația texturii. Rezoluția unghiulară este foarte mare și de aceea nu este întotdeauna posibil să realizăm o invarianță a orientării. Pentru texturile cu N înfășurări simetrice, momentele complexe au fost utilizate pentru o mai bună interpretare a rezultatelor. În general, toate aceste soluții propuse utilizează o bancă de filtre Gabor pentru realizarea invarianței.

O altă problemă similară cu cea a rotației texturilor pentru funcțiile Gabor este dependența de rezoluția texturilor. Dacă modelul texturilor este la o scară mult mai mare, adică la o rezoluție mai mică, este necesară o nouă funcție Gabor pentru realizarea discriminării a celor două texturi. Această problemă poate fi rezolvată prin utilizarea unei bănci de filtre Gabor pentru diferite rezoluții. Problema devine foarte complicată atunci când în imaginea de intrare există mai multe texturi cu diverse orientări și la diverse rezoluții, existența unei bănci de filtre nefiind o soluție. În concluzie, este necesar un nou mod de soluționare al problemei. Să presupunem că avem N_T modele de texturi diferite, care pot avea N_O orientări diferite și N_R rezoluții diferite. Scopul schemelor de segmentare a texturilor este de a distinge diferite modele ale texturilor dintr-o imagine. Calitatea unei scheme poate fi precizată prin acuratețea de estimare a orientărilor și rezoluțiilor pentru fiecare câmp de textură. Schemele de filtrare multicanal, ce utilizează bănci de filtre Gabor, au performanțe foarte bune în distingerea regiunilor texturate diferit. În general operația de indexare a diferitelor regiuni nu este foarte ușor de realizat dacă nu se folosesc explicit și estimările privind orientarea și rezoluția texturii. În prezența a: $N_O \times N_R$

diferite orientări/rezoluții sunt necesare mai mult de atâtea funcții Gabor într-o bancă de filtre pentru a se putea realiza o discriminare bazată pe proeminențele “spectrului orientare/rezoluție”.

Realizarea unei indexări bazată pe texturi poate fi împărțită în trei etape:

- identificarea texturii invariante la distorsiuni (orientare, rezoluție);
- estimarea orientării;
- estimarea rezoluției.

Dacă se utilizează operatori pentru textură invariante la rotație și scalare, va fi nevoie în mod practic de un număr redus de operatori Circular-Mellin pentru a se identifica un model de textură. Având în vedere cele prezentate, autorul a decis să folosească pentru început o metodă ce va identifica modelele de textură după care în funcție de necesități să treacă la estimarea orientării sau a scalării.

Operatori liniari invariante la rotație

Fiind dată o imagine $f(x,y)$, în coordonatele carteziane (x,y) putem să o reprezentăm în coordonate polar-logaritmice (λ,θ) în următorul mod:

$$\exp(\lambda) = r = \sqrt{x^2 + y^2} \quad \text{și} \quad \theta = \tan^{-1}(x, y) \quad \theta \in (-\pi, \pi] \quad (5.127.)$$

Lucrăm cu coordonate polare din motive de simplitate a calculelor și deci $f(x,y)$ este aceeași imagine cu $f(\lambda,\theta)$. Punctul de imagine pentru care se realizează conversia din coordonate carteziane în coordonate polar-logaritmice se referă la intrarea centrului de expansiune (COE_i). Locul tuturor punctelor cu aceeași coordonată λ este în jurul lui COE_i și în mod similar locul punctelor de coordonate θ este o rază exterioară lui COE_i . Obiectivul nostru este un filtru $h(x,y)$ care este invariant la rotație. Răspunsul corelației obținut cu o imagine de intrare este dat de relația:

$$c(x,y) = \iint_{x,y} f(x,y) \cdot h(x,y) dx dy \quad (5.128.)$$

Ecuția (5.128) poate fi exprimată în coordonate polare astfel:

$$c(COE_c) = \iint_{\lambda,\theta} f(\lambda,\theta) \cdot h^*(\lambda,\theta) \cdot \exp(2\lambda) d\lambda d\theta \quad (5.129.)$$

De remarcat că COE_i al imaginii coincide cu COE_i al filtrului și pentru fiecare COE_i la intrare este un punct corespondent COE_c în planul corelației pentru care răspunsul filtrului dorit este invariant la rotație.

Orice obiect (textură) poate fi descompus într-un număr finit de funcții circular armonice (CHF) care formează un spațiu ortogonal. Imaginea în coordonate polar-logaritmice poate fi descompusă în armonicele circulare cu ajutorul relației:

$$f(\lambda,\theta) = \sum_{m=-\infty}^{\infty} f_m(\lambda) \exp(j \cdot m\theta) \quad (5.130.)$$

unde $f_m(\lambda) \exp(j \cdot m\theta)$ este armonica circulară de ordinul m a imaginii și

$$f_m(\lambda) = \frac{1}{2\pi} \int_0^{2\pi} f(\lambda,\theta) \exp(-j \cdot m\theta) d\theta \quad (5.131.)$$

Pentru a sintetiza un filtru invariant la rotație $h(\lambda,\theta)$, vom separa filtrul după λ și θ folosind armonicele circulare:

$$h(\lambda, \theta) = \sum_{q=-\infty}^{\infty} h_q(\lambda) \exp(j \cdot q\theta) \quad (5.132.)$$

unde,

$$h_q(\lambda) = \frac{1}{2\pi} \int_0^{2\pi} h(\lambda, \theta) \exp(-j \cdot q\theta) d\theta \quad (5.133.)$$

unde, q este ordinul funcției armonice circulare. În notații se folosește litera m pentru ordinul CHF a imaginii de intrare și q pentru ordinul CHF al filtrului. Dacă filtrul este corelat cu o imagine de intrare rotită cu un unghi α , $f(\lambda, \theta + \alpha)$ folosind (5.129), valoarea în punctul COE_c din planul corelației este:

$$c(COE_c) = \iint_{\lambda, \theta} f(\lambda, \theta + \alpha) \cdot h^*(\lambda, \theta) \cdot \exp(2\lambda) d\lambda d\theta \quad (5.134.)$$

Înlocuind $h(\lambda, \theta)$ din (5.132) și $f(\lambda, \theta + \alpha)$ din relația (5.130) pentru obiectul rotit, vom obține:

$$c(COE_c) = \sum_{m=-\infty}^{\infty} \sum_{q=-\infty}^{\infty} \exp(j \cdot m\alpha) \times \int_{\lambda} f_m(\lambda) \cdot h_q^*(\lambda) \cdot \exp(2\lambda) d\lambda \times \int_{\theta} \exp(j \cdot [m - q] \cdot \theta) d\theta \quad (5.135.)$$

Deoarece integrarea peste θ este diferită de zero pentru $m=q$ deoarece armonicile sunt ortogonale, vom obține:

$$c(COE_c) = 2\pi \sum_{q=-\infty}^{\infty} \exp(j \cdot q\alpha) \times \int_{\lambda} f_q(\lambda) \cdot h_q^*(\lambda) \cdot \exp(2\lambda) d\lambda \quad (5.136.)$$

Deoarece însumarea după q , amplitudinea valorii $c(COE_c)$ va fi independentă de α dacă o armonică este folosită:

$$h(\lambda, \theta) = h_q(\lambda) \cdot \exp(j \cdot q\theta) \quad (5.137.)$$

Dacă filtrul este de forma din relația (5.137), răspunsul amplitudinii complexe este invariant la rotație. De notat, că rezultatul este independent de imaginea specifică $f(x, y)$, amplitudinea răspunsului corelației pentru toate obiectele indiferent de rotație este aceeași, iar valoare obținută poate fi folosită pentru a distinge un obiect de altul. Notăm factorul $\exp(\lambda) = r^2$ în integrala din relația (5.136).

Operatori liniari invarianți la scalare

Orice obiect poate fi descompus într-un număr infinit de funcții armonice Mellin (MHF), care formează un spațiu ortogonal al caracteristicilor. Imaginea în coordonate polare poate fi descompusă în armonici Mellin în următorul mod:

$$f(\lambda, \theta) = \exp(-\lambda) \sum_{n=-\infty}^{\infty} f_n(\theta) \exp(j \cdot 2\pi n\theta) \quad (5.138.)$$

unde, $\exp(-\lambda) f_n(\theta) \exp(j \cdot 2\pi n\theta)$ este funcția armonică de ordinul m a imaginii.

$$f_m(\theta) = \frac{1}{\lambda_{max} - \lambda_{min}} \int_{\lambda_{min}}^{\lambda_{max}} f(\lambda, \theta) \cdot \exp(\lambda) \cdot \exp(-j \cdot 2\pi n\lambda) d\lambda \quad (5.139.)$$

În relația (5.139), $\exp(\lambda_{min})$ și $\exp(\lambda_{max})$ sunt minimul și maximul ales pentru integrare și pentru respectarea condiției de ortogonalitate pentru diverse ordine ale funcțiilor MHF, și pot fi alese astfel încât $(\lambda_{max} - \lambda_{min})$ să fie un întreg.

Pentru a sintetiza filtrul $h(x,y)$, invariant la scalare, vom separa după λ și θ filtrul folosind armonicile Mellin:

$$h(\lambda, \theta) = \exp(-\lambda) \sum_{p=-\infty}^{\infty} h_p(\theta) \exp(j \cdot 2\pi p \theta) \quad (5.140.)$$

unde,

$$h_p(\theta) = \frac{1}{\lambda_{max} - \lambda_{min}} \int_{\lambda_{min}}^{\lambda_{max}} h(\lambda, \theta) \cdot \exp(\lambda) \cdot \exp(-j \cdot 2\pi p \lambda) d\lambda \quad (5.141.)$$

unde, p este ordinul funcției armonice Mellin. În notația noastră, n este ordinul MHF a imaginii de intrare, iar p este ordinul MHF a filtrului.

Deoarece însumarea peste p , amplitudinea acestei expresii va fi independentă de β numai dacă se folosește o singură armonică în filtru și vom avea:

$$h(\lambda, \theta) = \exp(-\lambda) \cdot h_p(\theta) \cdot \exp(j \cdot 2\pi p \lambda) \quad (5.142.)$$

Dacă filtrul este de forma arătată în relația (5.142), amplitudinea răspunsului corelației este dat de termenul din paranteza dreaptă din relația (5.141). Întotdeauna, constanta de scalare aflată în afara parantezei drepte conține un factor ce este dependent de β . Deci, valoarea amplitudinii răspunsului corelației va fi diferit pentru diferite scale ale obiectul vizat. De notat că raportul amplitudinilor răspunsurilor obținute cu o pereche de filtre (pentru armonici diferite) este invariant la scalare. Rezultatul este de asemenea independent de imaginea de intrare $f(x,y)$, amplitudinea răspunsului pentru toate corelațiile pentru toate imaginile scalate ale obiectului vizat este proporțională cu scala obiectului și deci poate fi folosită pentru a distinge obiectul la diferite scale.

Operatorii Circular-Mellin pentru texturi

După cum a rezultat din demonstrațiile de mai sus, un filtru de corelație invariant la rotație poate fi exprimat în coordonate polar-logaritmice în următorul mod:

$$h(\lambda, \theta) = h_q(\lambda) \cdot \exp(j \cdot q \theta) \quad (5.143.)$$

După cum a rezultat din demonstrațiile de mai sus un filtru de corelație invariant la scalare poate fi exprimat în coordonate polar-logaritmice în următorul mod:

$$h(\lambda, \theta) = \exp(-\lambda) \cdot h_p(\theta) \cdot \exp(j \cdot 2\pi p \lambda) \quad (5.144.)$$

Dacă se examinează structura filtrelor din ecuațiile (5.143) și (5.144) acestea sunt separabile în λ și θ . Întotdeauna invarianța la rotație va impune constrângeri asupra componentei θ a filtrului, pe când invarianța la scalare va impune constrângeri numai asupra componentei λ a filtrului. Deci este posibil să proiectăm un filtru care să fie invariant la ambele, forma acestuia fiind:

$$h(\lambda, \theta) = \exp(-\lambda) \cdot \exp(j \cdot 2\pi p \lambda) \cdot \exp(j \cdot q \theta) \quad (5.145.)$$

Pentru diverse valori ale lui p (ordinul MHP) și q (ordinul CHP) o clasă a filtrelor de corelație poate fi sintetizată, toate aceste filtre fiind invariante la rotație și la scalare.

Vom alege COE_i ce corespund centrului filtrului din planul Cartesian. Pentru fiecare x,y din spațiul Cartesian, se poate determina corespondentul din planul polar-logaritmice pornind de la relația (5.146).

O privire mai atentă a extractorilor de caracteristici din relația (5.146) remarcăm că aceste caracteristici provin de la o decompoziție în serii Fourier a imaginii de intrare $f(x,y)$ în coordonate polare, coeficienții Fourier din planul polar-logaritmice se calculează folosind:

$$\mathcal{F}(p, q) = \frac{1}{2\pi(\lambda_{\max} - \lambda_{\min})} \cdot \int_{\lambda} \int_{\theta} f(\lambda, \theta) \exp(-\lambda) \exp(j2\pi p\lambda) \exp(jq\theta) d\lambda d\theta \quad (5.146.)$$

Iar imaginea originală poate fi reconstruită din coeficienții Fourier cu ajutorul relației:

$$f(\lambda, \theta) = \sum_p \sum_q \mathcal{F}(p, q) \exp(-\lambda) \exp(j2\pi p\lambda) \exp(jq\theta) \quad (5.147.)$$

Coeficienții Fourier din relația (5.146) pentru toate valorile p și q ne dau o descriere completă a imaginii folosind o bază ortogonală.

Funcția fereastră exponențială $\exp(-\lambda)$ în relațiile (5.146) și (5.147) este Jacobianul transformării din coordonate cartesiene în coordonate polar-logaritmice și este necesar pentru a menține ortogonalitatea caracteristicilor Fourier polar-logaritmice. Este ceva similar cu fereastra gaussiană ce specifică regiunea spațială de interes în transformările Gabor.

Caracteristicile Gabor sunt date de o decompoziție spectrală pe o anumită fereastră a imaginii folosind o singură frecvență (u, v) , caracteristicile Gabor fiind o combinație liniară de caracteristici Fourier din planul Cartesian pentru o vecinătate (u, v) din domeniul frecvență.

Caracteristicile Circular-Mellin sunt o altă combinație liniară, diferită de cea Gabor, a caracteristicilor Fourier cartesiene pentru o fereastră de imagine, iar coeficienții combinației liniare se aleg astfel ca să se obțină o invarianță la rotație și scalare.

În figura 5-26 este arătată partea reală a unui extractor de caracteristici Gabor-orientat și a unui Circular-Mellin. Fereastra Gaussiană pentru funcții Gabor orientate are axele principale orientate după direcțiile orizontale și verticale, iar sinusoida care modulează gaussiană este de asemenea orientată în aceeași direcție. Funcțiile Circular-Mellin sunt compuse din sinusoida radiale și deci se anulează direcțiile.



Figura 5-26 a) Extractor de caracteristici Gabor; b) Extractor de caracteristici Circular-Mellin

În coordonate cartesiene, amplitudinea coeficienților Fourier este invariantă la deplasare, deci amplitudinea purtătoare de informație despre structura internă a zonei de interes, în timp ce localizarea obiectului este dată de informația de fază.

În coordonate polar-logaritmice, deplasarea de a lungul axei θ corespunde rotației obiectului iar deplasarea de a lungul axei λ corespunde scalării.

Știind că amplitudinea coeficienților Fourier este invariantă la deplasare, $|\mathcal{F}(p, q)|$ este invariant la deplasare în domeniul polar-logaritmice și deci este invariant la rotație și scalare. Vom considera acum o combinație de două sau mai multe caracteristici Fourier. Amplitudinea combinației liniare de coeficienți Fourier complecși nu este invariantă la deplasare. În mod

similar în domeniul polar-logaritmic, folosirea unei combinații liniare de două sau mai multe funcții Circular-Mellin care nu vor fi în mod cert invariante la rotație sau scalare.

Pentru reconstrucția imaginii avem nevoie de toate caracteristicile ce se obțin prin utilizarea transformatei Fourier în coordonate polare, pentru segmentare și detecția obiectelor se poate folosi un număr mai redus de caracteristici polar-logaritmice. Din testele realizate a rezultat că o caracteristică Circular-Mellin va fi adecvată pentru realizarea indexării, iar în cazuri complicate este bine ca numărul caracteristicilor să nu fie mare.

Atâta timp cât $|\mathcal{F}(p, q)|$ pentru toate valori lui p și q în relația (5.146) este invariant la rotație și scală nu este invariant la deplasare. O problemă similară apare și în cazul funcțiilor Gabor. Știind că atât funcțiile Gabor cât și funcțiile bazei Circular-Mellin sunt combinații liniare ale caracteristicilor Fourier carteziene, proprietatea de invarianță la deplasare a unei singure caracteristici Fourier nu mai este valabilă. Aceasta implică că localizarea zonelor de interes va fi cunoscută înainte ca acești extractori de caracteristici să poată fi folosiți pentru a clasifica o zonă de interes. În general, această simplificare nu este întotdeauna posibilă și localizarea zonelor de interes nu este cunoscută apriori. Deci vom considera un corelator de arhitectură, știind că mai multe obiecte în paralel pot fi afectate de zgomot. Filtrul folosit într-un corelator extrage un set de caracteristici ale pixelilor situați într-o fereastră și operația de corelație combină aceste caracteristici într-o valoare de ieșire. Arhitectura poate fi folosită pentru recunoașterea unui obiect, obținându-se o invarianță la distorsiune. Dacă filtrul de corelație este de forma dată în relația (5.147), atunci fiecare punct (x_0, y_0) din domeniul spațial, filtrul extrage trăsăturile ce sunt caracteristice pentru zona de interes din acel punct. Folosind un set de astfel de filtre se poate extrage un set de caracteristici invariante la rotație și la scală (folosind raporturi ale magnitudinilor răspuns al perechilor de filtre pentru invarianță la scală) pentru diferite puncte din imagine și folosind aceste caracteristici se poate clasifica fiecare punct dacă aparține zonei de interes sau fundalului.

Folosirea funcțiilor Circular-Mellin în indexarea imaginilor

Pentru segmentare se folosesc în general imagini de aceeași dimensiuni ce au fost normalizate din punct de vedere al energiei, pentru a ne asigura că nu sunt segmentate pe baza diferențelor de energie [176]. Algoritmul de selecție al caracteristicilor este un algoritm de căutare ce determină cea mai bună combinație a perechii p, q pentru diverse texturi. Vom restricționa căutarea la un număr redus de valori radiale și de frecvențe. Se poate folosi $p, q = 0 \div 9$ și $\lambda_{min} = 0$, $\lambda_{max} = 3$ astfel că $\lambda_{max} - \lambda_{min} = 3$ pentru a exista ortogonalitate. În toate testele a fost folosită o singură caracteristică pentru segmentare, cea mai reprezentativă, mai multe caracteristici folosindu-se numai în cazuri mai complexe [177].

Indexarea ce folosește operatorul Circular-Mellin presupune aceiași pași ca și în cazul utilizării funcțiilor Gabor [178], [179], [180].

Pentru fiecare extractor de caracteristică Circular-Mellin este generat pentru început planul de corelație al operatorului pentru textură. Aceste filtre convertesc variațiile texturii de la intrare în variații la ieșire ale amplitudinii în planul de corelație, în cazul când se aduc la intrare diferite texturi rezolvarea se găsește în folosirea unei scheme cu prag.

Amplitudinea modelului de corelație conține caracteristici ce sunt invariante la rotație și scalare, deci pot fi considerate reprezentative pentru un anumit model de textură ce este la intrare. În general, pentru aceeași textură, variațiile ieșirii planului de corelație sunt încă afectate de alegerea COE_i din modelul de plasare a texelilor. Pentru a elimina aceste diferențe ale valorilor caracteristicilor vom realiza o mediere locală pe segmente reduse în planul de corelație. Știind că amplitudinea la orice pixel din planul de corelație este invariantă la rotație și scalare, media amplitudinilor va fi și ea invariantă la rotație și scalare și mult mai robustă la variațiile din

modelul de plasare a texelilor. Alegerea dimensiunii regiunii pentru mediere depinde de textură și de rezoluția folosită. În final se poate folosi un simplu algoritm cu prag pentru a segmenta planul de ieșire după diferite texturi.

După ce au fost identificate diferitele texturi din imaginea de intrare este firesc să trecem la un alt nivel de procesare pentru a identifica orientări și rezoluții pentru modele de texturi. Este necesar să distingem de exemplu dacă o textură are o orientare orizontală sau verticală. Pentru aceasta avem nevoie de filtre orientate care să caute diferite orientări ale caracteristicilor texturilor. Funcțiile Circular-Mellin pot fi forțate să aibă o orientare specifică prin restricționarea unghiulară a acestora. De exemplu putem considera un operator Circular-Mellin cu $p = q = 2$ constrânge intervalul la $\pm \pi/4$ față de axa orizontală. Un răspuns mai mare al corelației se obține dacă modelul de textură din imaginea de intrare are aceeași orientare cu filtrul. În acest caz, suntem interesați să distingem numai între caracteristicile liniare orizontale și verticale. Când se dorește o fină clasificare este recomandabil să se folosească o bancă de filtre orientate. Se pot utiliza și alte metode ce folosesc alți operatori [181], [182], [183].

5.2.10. Generarea texturilor folosind modele autoregresive

În foarte multe domenii, se folosesc baze de date care să conțină diverse texturi. Texturile sunt de diverse tipuri și de aceea este foarte dificil de a crea un algoritm unic de generare a acestora.

În continuare prezint o metodă proprie de generare a unui set de parametri ce pot fi folosiți la generarea unei texturi, de un anumit tip, pornind de la un eșantion de textură cunoscut, metodă comunicată de autor în anul 2000, conform [184]. Eficiența metodei propuse constă în faptul că nu este necesară stocarea efectivă a unei texturi ci numai parametrii necesari generării acesteia. Un alt avantaj rezultă din metrica ce poate fi definită peste acest sistem de generare de texturi [185]. Metrica poate fi folosită în procesul de căutare, atunci când se dorește găsirea celei mai apropiate texturi de una existentă în baza de date sau când se dorește catalogarea unei noi texturi într-o categorie existentă [186].

Multe din texturile care se regăsesc în natură nu respectă foarte strict regulile ce rezultă din modelele ce folosesc diverse gramatici. De aceea, este nevoie de o nouă abordare care să concorde cu realitatea. Pentru generare am folosit un model autoregresiv bidimensional care oferă următoarele avantaje:

- La formarea bazei de texturi se calculează un set de coeficienți, nu foarte numeroși ce sunt stocați. Aceasta implică un spațiu foarte redus, în comparație cu memorarea unui eșantion din textură.
- Atunci când se folosește o anumită textură din baza de date, operația de generare este relativ simplă, nu implică o putere de calcul foarte mare și în plus poate genera o textură de dimensiunea dorită.
- Operația de identificare a unei texturi în baza de date se face pe baza comparării a câtorva coeficienți și nu pe baza comparării pixel cu pixel a texturii. Acest mod de comparare sporește foarte mult viteza căutare.

În continuare am făcut o descriere generică a proceselor autoregresive și modul de calcul al coeficienților modelului. În multe aplicații se dispune, în afara eșantionului de textură $x(n)$, $n=0,1,\dots,N-1$, de informații suplimentare despre procesul care generează secvența de date sau se pot face anumite presupuneri, altele decât considerarea procesului ca fiind periodic. Pe baza acestor informații sau presupuneri, se poate alege un model exact sau aproximativ al procesului care generează secvența de date. Având modelul, se procedează la estimarea (identificarea) parametrilor modelului utilizând eșantionul de textură. Estimarea spectrală se obține prin

înlocuirea parametrilor modelelor în relația de calcul a densității spectrale de putere corespunzătoare modelului.

Estimarea spectrală bazată pe tehnica modelării și identificarea parametrilor se realizează în trei etape: selecția modelului, estimarea parametrilor modelului și calculul estimatorului spectral.

Utilizarea tehnicii de modelare și identificare a parametrilor poate conduce la estimări spectrale îmbunătățite din punctul de vedere al rezoluției și fidelității, funcție de abilitatea cu care se face corelarea modelului și a parametrilor modelului cu informațiile despre textură și cu eșantionul de textură generată de proces. Se precizează că modelarea implică extrapolarea eșantionului de textură în afara intervalului de observare, altfel decât în variantă periodică, ceea ce, în multe aplicații și în special pentru secvențe de date scurte, conduce la estimări spectrale îmbunătățite față de cazul utilizării metodelor tradiționale.

Conform modelului funcției de transfer raționale, eșantionul de textură $x(n)$ se obține la ieșirea unui sistem liniar comandat la intrare cu o secvență $u(n)$, conform relației:

$$x(n) = -\sum_{k=1}^p a(k)x(n-k) + \sum_{k=0}^q b(k)u(n-k), \quad (5.148.)$$

unde, $b(0)=1$.

Acest model este cunoscut sub denumirea de model ARMA(p,q) [187], [188]. Parametrii modelului sunt coeficienții de regresie $a(k)$ coeficienții de mediere $b(k)$. Secvența de intrare $u(n)$ care face parte din model se consideră o realizare a unui proces aleator zgomot alb cu media zero și dispersia σ . Se pune problema determinării coeficienților de autoregresie (predicție) $a(k)$, $k=1, 2, \dots, p$ prin rezolvarea sistemului de p ecuații liniare rezultat din relațiile Yule-Walker. Sistemul de ecuații liniare rezultat are o structură particulară dată de matricea construită pe baza secvenței de autocorelație. Această structură permite dezvoltarea algoritmului Levinson care necesită un număr de operații de ordinul p^2 .

Algoritmul Levinson constă în calculul recursiv al coeficienților de predicție pentru modelul de ordinul k în funcție de coeficienții de predicție pentru modelul de ordinul $k-1$, pentru $k=2, 3, \dots, p$. Astfel, în cadrul algoritmului se calculează succesiv coeficienții

$a_1(1)$ pentru modelul de ordinul 1,

$a_2(1), a_2(2)$ pentru modelul de ordinul 2,

$a_p(1), a_p(2), \dots, a_p(p)$ pentru modelul de ordinul p ,

De asemenea, în cadrul algoritmului, se calculează succesiv puterile erorilor de predicție ρ_k pentru modelele de ordinul $k=1, 2, \dots, p$. Astfel, pe baza analizei valorilor puterilor erorilor de predicție se poate stabili ordinul modelului. Se precizează că puterea erorii de predicție scade la creșterea succesivă a ordinului modelului, astfel încât ordinul poate fi stabilit la valoarea de la care creșterea ordinului nu conduce la reducerea semnificativă a puterii erorii de predicție. Algoritmul Levinson permite calculul recursiv al parametrilor $a_k(1), a_k(2), \dots, a_k(k), \rho_k$ pentru $k = 1, 2, \dots$. Algoritmul demonstrat în bibliografie se inițializează prin

$$a_1(1) = -\frac{r_{xx}(1)}{r_{xx}(0)}, \quad \rho_1 = (1 - |a_1(1)|^2) r_{xx}(0) \quad (5.149.)$$

și cuprinde calcule recursive pentru $k=2, 3, \dots$ prin relațiile:

$$a_k(k) = - \frac{r_{xx}(k) + \sum_{l=1}^{k-1} a_{k-1}(l)r_{xx}(k-l)}{\rho_{k-1}}, \quad (5.150.)$$

$$a_k(i) = a_{k-1}(i) + a_k(k)a_{k-1}^*(k-i) \text{ pentru } i=1, 2, \dots, k-1, \quad (5.151.)$$

$$\rho_k = (1 - |a_k(k)|^2)\rho_{k-1}. \quad (5.152.)$$

Coeficienții $a_k(k), k = 1, 2, \dots$ se pot nota cu k_k și se numesc coeficienți de reflexie.

Pentru simplificarea scrierii nu au fost detaliate exprimările algoritmului pe cele două dimensiuni spațiale. preferându-se o redare sugestivă a modului de generare a texturilor. Câteva din rezultatele obținute de autor sunt date în figura următoare:

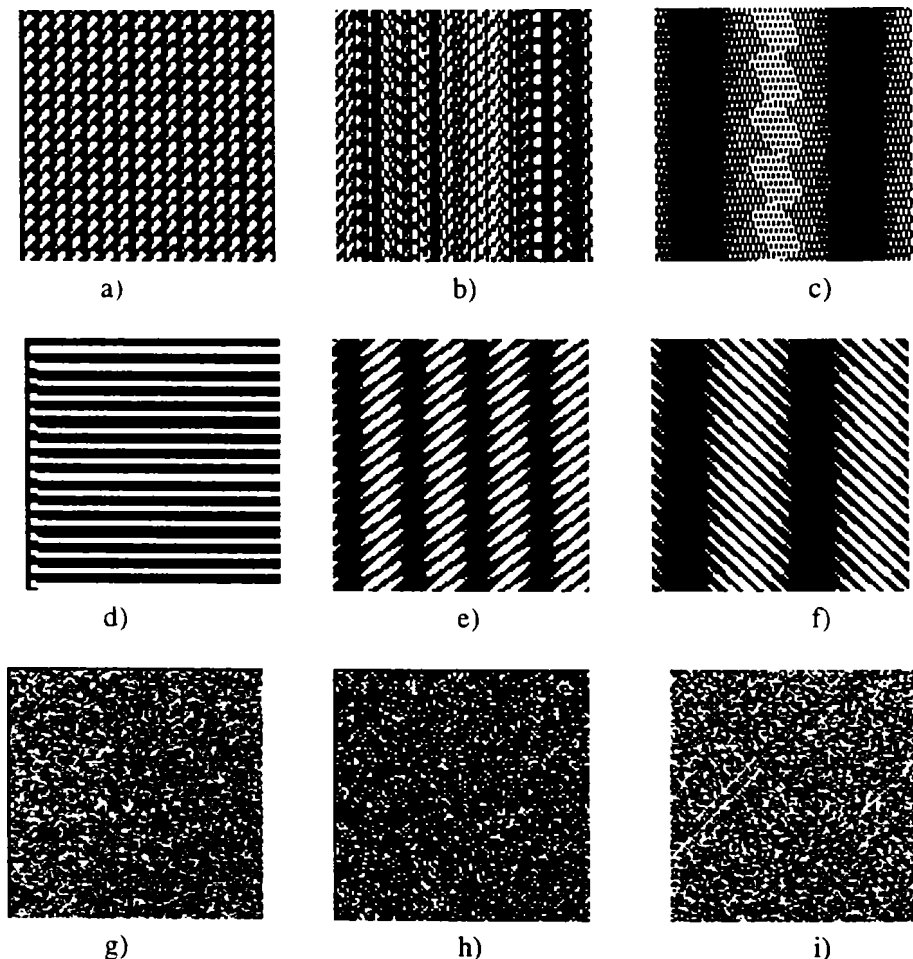


Figura 5-27 Texturi generate folosind modelul autoregresiv

Folosindu-se un număr relativ redus de coeficienți, maxim 20, au fost obținute texturile din figura de mai sus. Modelarea unei texturi poate fi realizată cu o precizie mult mai bună dacă se crește numărul coeficienților (valoarea maximă verificată a fost de 100), dar nu se justifică pentru toate tipurile de texturi.

Algoritmul propus se adresează bazelor de texturi de mari dimensiuni. Din punct de vedere al spațiului ocupat se realizează o foarte bună compresie, iar la căutare este mult mai rapidă o variantă de comparare a câtorva coeficienți decât o comparare pixel cu pixel. Generarea texturilor se face ușor, nu implică o putere mare de calcul și în schimb se pot genera texturi de orice dimensiune. Domeniul de aplicabilitate este foarte mare, aplicațiile multimedia și de grafică având cele mai mari avantaje. Aplicația poate fi folosită în diverse domenii cum ar fi managementul imaginilor medicale, librării multimedia, arhive de documente, muzee, informații geografice, etc.

5.2.11. Gradientul texturii

Evaluarea gradientului texturii folosind transformarea Hough

Importanța texturii în determinarea orientării suprafețelor a fost descrisă de Gibson încă din 1950. Există trei căi prin care aceasta poate fi făcută. Aceste metode pleacă de la premiza că textura aparține unei suprafețe plane. Metodele sunt sugerate intuitiv în figura de mai jos:

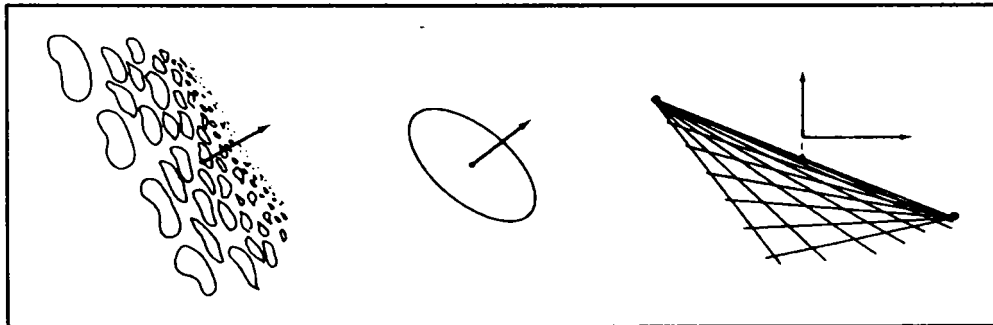


Figura 5-28 Modificarea formei textelului în funcție de gradient

Pentru început, dacă textura imaginii a fost segmentată în primitive, rata maximă de modificare a dimensiunii proiecției a acestor primitive constă în orientarea planului în următorul mod:

- Direcția ce corespunde maximului ratei de modificare a dimensiunilor primitive proiectate este gradientul texturii. Orientarea acestei direcții în raport cu coordonatele imaginii determină cât de mult este rotit planul imaginii texturate față de direcția de unde este privit acel plan.
- Magnitudinea gradientului ajută la determinarea unghiului cu care este rotit planul față de observator (cameră video, etc.). Este necesară însă și o bună cunoaștere a captorului de imagine.
- Rotația și înclinarea reprezintă coordonatele polare ale gradientului.

Cea de a doua cale de măsurare a orientării suprafeței utilizează cunoașterea apriori a formei textelului. De fapt o textură compusă din cercuri apare formată din elipse atunci când suprafața este înclinată. Orientarea axelor principale ale elipselor definesc rotația față de cameră, iar raportul axelor definește înclinarea.

În final, dacă textura este compusă din texeli aranjați după un grid regulat, vom putea calcula punctele de la infinit. Pentru perspectiva imaginii, punctele de infinit din planul P sunt proiecțiile în planul imaginii a punctelor de la infinit pentru a da direcția. Metoda se aplică în general texturilor ce au plasarea textelilor după linii drepte.

Liniile din imagine pot fi detectate cu ajutorul algoritmului Hough. De exemplu, prin utilizarea parametrizării liniei:

$$x \cdot \cos \theta + y \cdot \sin \theta = r \quad (5.153.)$$

și prin cunoașterea orientării liniei prin intermediul gradientului $g = (\Delta x, \Delta y)$ a unui segment de linie $(x, y, \Delta x, \Delta y)$ ce poate fi trecut în coordonate polare folosind relațiile:

$$r = \frac{\Delta x \cdot x + \Delta y \cdot y}{\sqrt{\Delta x^2 + \Delta y^2}} \quad (5.154.)$$

$$\theta = \tan^{-1} \left(\frac{\Delta x}{\Delta y} \right)$$

Aceste relații pot fi derivate și folosindu-ne de figura 5-29 și de proprietățile geometrice.

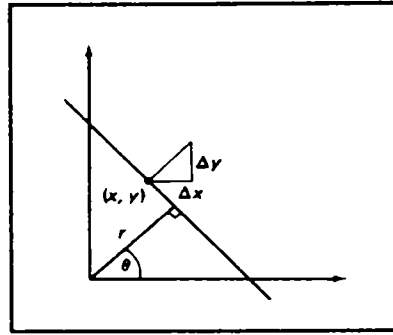


Figura 5-29 Măsurarea orientării unei suprafețe

Coordonatele Cartesiene ale spațiului vectorial $r - \theta$ sunt date de:

$$a = \left(\frac{\mathbf{g} \cdot \mathbf{x}}{\|\mathbf{g}\|^2} \right) \mathbf{g} \quad (5.155.)$$

Utilizând această transformare, mulțimea segmentelor de linie L_1 sunt mapate într-un singur punct în spațiul $r - \theta$. În plus, mulțimea liniilor L_2 ce are același punct de infinit (x_v, y_v) proiectat într-un cerc în spațiul $r - \theta$ ce are ca diametru segmentul de linie $((0, 0) (x_v, y_v))$.

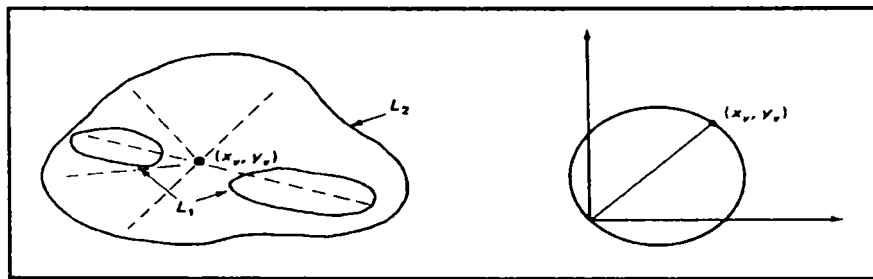


Figura 5-30 Transformarea spațiului prin aplicarea transformatei Hough

Această schemă are două dezavantaje:

- punctele de la infinit ce dispar sunt proiectate tot la infinit;
- cercurile necesită eforturi de detectare.

Deci este motivată utilizarea transformatei $(x, y, \Delta x, \Delta y) \rightarrow \left(\frac{k}{r}, \theta \right)$ pentru câteva constante k . Acum punctele ce dispar la infinit sunt proiectate în origine și în locul mulțimii L_2 avem acum o linie. Această linie este perpendiculară pe vectorul x_v și $\frac{k}{\|x_v\|}$ unități față de origine.

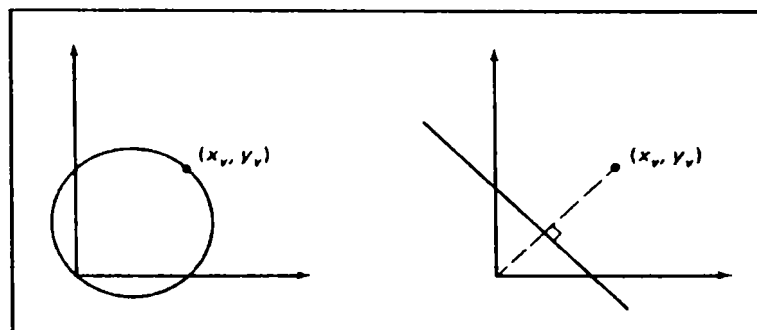


Figura 5-31 Revenirea la coordonate carteziene

Acest vector poate fi detectat de cea de a doua parte a transformatei Hough; fiecare punct a este mapat în spațiul $r' - \theta'$. Pentru fiecare a , vom calcula toate valorile r', θ' astfel încât să avem:

$$a \cos \theta' + b \sin \theta' = r' \quad (5.156.)$$

În acest spațiu secund, punctele ce dispar la infinit vor fi detectate cu relațiile:

$$r' = \frac{k}{\|x_v\|} \quad (5.157.)$$

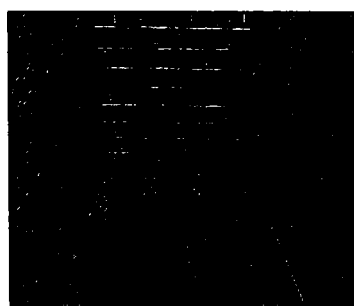
$$\theta' = \tan^{-1} \left(\frac{y_v}{x_v} \right)$$

Evaluarea gradientului prin descompunerea texturilor folosind Filtre Gabor

Detecția gradientului texturii depinde de detectarea schimbării dimensiunii texelilor din imagine. Această schimbare este dată atât de distanța dintre punctul de observație cât și de înclinarea planului în raport cu ecranul de proiecție. Detecția schimbărilor de scală a texelilor implică obținerea unor valori măsurabile care să fie calculate pentru fiecare punct din imagine, cu alte cuvinte este necesară o conversie care să convertească gradientul texturii în valori numerice [189].

Pentru găsirea perioadei fundamentale a texturii se pot folosi două metode: una care să estimeze perioada spațială pentru fiecare punct din imagine și o variantă care să calculeze o medie a valorilor obținute de fiecare filtru. Perioada fundamentală poate fi definită ca fiind perioada (cea mai mică frecvență) filtrelor Gabor ce produce un răspuns semnificativ pentru acea locație. Cu alte cuvinte, căutăm dimensiunea unor elemente de textură care se repetă cu o anumită perioadă. Se alege cea mai mică valoare a frecvenței și din considerente legate de modul de percepție a sistemului vizual uman. Calculul perioadei fundamentale spațiale pentru fiecare locație de imagine poate fi văzută ca o selectare din totalitatea filtrelor folosite [190], a aceluia care are cea mai mare lățime și are o contribuție importantă în raport cu celelalte. Deci, presupune folosirea unui prag de selecție.

În experimentările pe care le-am efectuat am folosit 25 de filtre. Răspunsurile acestora pentru o imagine ce combină mai multe texturi cu diverse orientări sunt date în figura următoare:



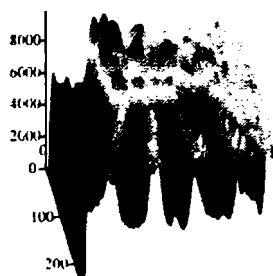
a)



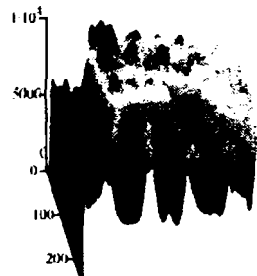
b. 1.)



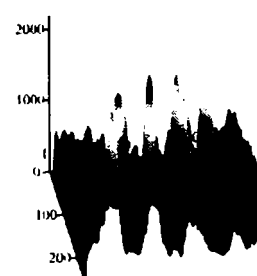
b. 2.)



b. 3.)



b. 4.)



b. 5.)

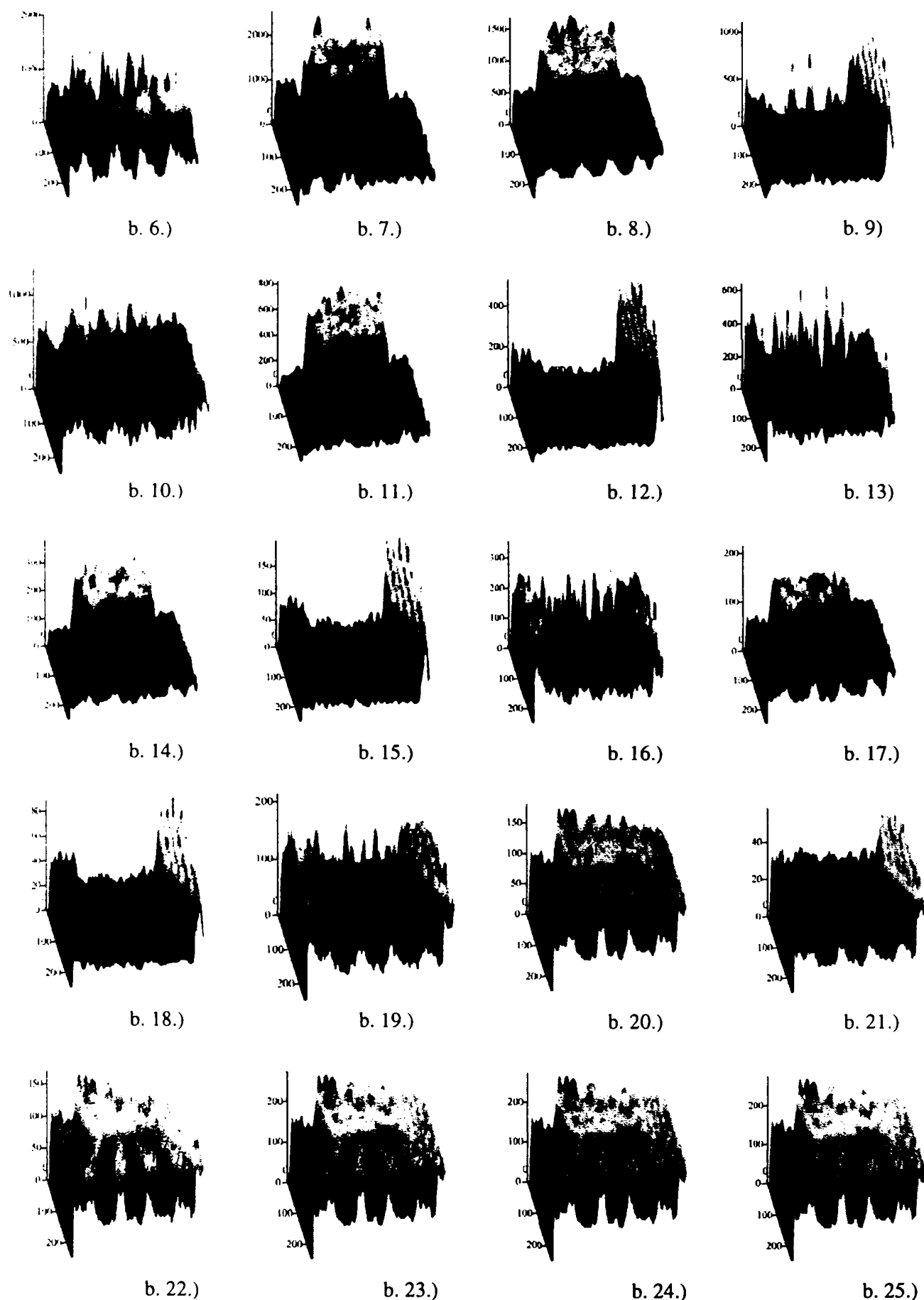


Figura 5-32 a) Imaginea originală ce conține plane texturate cu diverse orientări; b) Filtrarea imaginii folosind o familie de filtre Gabor cu următorii parametrii (u, v, σ):

b.1. (0, 0, 128); b.2. ($\pi/8, \pi/8, 64$); b.3. (0, $\pi/8, 64$); b.4. ($\pi/8, 0, 64$); b.5. ($\pi/4, \pi/4, 32$); b.6. (0, $\pi/4, 32$); b.7. ($\pi/4, 0, 32$); b.8. ($3\pi/8, 0, 16$); b.9. ($3\pi/8, 3\pi/8, 16$); b.10. (0, $3\pi/8, 16$); b.11. ($\pi/2, 0, 8$); b.12. ($\pi/2, \pi/2, 8$); b.13. (0, $\pi/2, 8$); b.14. ($5\pi/8, 0, 4$); b.15. ($5\pi/8, 5\pi/8, 4$); b.16. (0, $5\pi/8, 4$); b.17. ($3\pi/4, 0, 2$); b.18. ($3\pi/4, 3\pi/4, 2$); b.19. (0, $3\pi/4, 0$); b.20. ($7\pi/8, 0, 1$); b.21. ($7\pi/8, 7\pi/8, 1$); b.22. (0, $7\pi/8, 1$); b.23. ($\pi, 0, 0.1$); b.24. (0, $\pi, 0.1$); b.25. ($\pi, \pi, 0.1$);

În urma experimentărilor, am observat că rezultate foarte bune se pot obține pornind de la matricea de coocurență a imaginii originale.

Pentru o translație t , matricele de coocurență MC_t ale unei regiuni R , sunt definite pentru toate cuplurile de nivele de gri (a, b) prin relația următoare:

$$MC_t(a, b) = \text{card}\{(s, s+t) \in R^2 \mid A[s] = a, A[s+t] = b\} \quad (5.158.)$$

$MC_t(a, b)$ este deci numărul de cupluri de pixeli $(s, s+t)$ din regiunea considerată, separați prin vectorul de translație t .

Matricele de coocurență conțin o cantitate importantă de informație, dar aceasta este dificil de manipulat în întregime. În general, se folosesc doar câțiva indici sau caracteristici ale imaginilor.

În cazul existenței unei perspective sau a deformărilor datorate înclinării planului, se poate detecta gradientul acestuia pornind de la contrastul matricei de coocurență:

$$\text{Contrastul} : \frac{1}{N_c (L-1)^2} \sum_{k=0}^{L-1} k^2 \sum_{|a-b|=k} MC_t(a, b) \quad (5.159.)$$

unde, N_c este numărul de cupluri din imagine iar L este numărul de niveluri de gri din imagine. Fiecare termen al matricei MC_t este ponderat prin distanța sa față de diagonală. Se obține un indice ce corespunde noțiunii uzuale de contrast. El este mai mare când termenii de-a lungul diagonalei matricei sunt mari, acest lucru se datorează trecerii de la un pixel cu un nivel de gri mare la unul foarte redus. Rezultatele obținute sunt prezentate în figura următoare:



Figura 5-33 Reprezentarea contrastului pentru o fereastră de 10 x 10 și 22 x 22

Am experimentat calculul acestuia, folosind mai multe dimensiuni de ferestre de analiză. Răspunsul contrastului matricei de coocurență este asemănător cu cel al filtrelor Gabor. Dimensiunea ferestrei în cazul matricei de coocurență realizează legătura spațială dintre caracteristicile spațiale ale texturii și răspunsul contrastului. În cazul filtrelor Gabor parametrul σ ne dă corespondența spațială. Am încercat mai multe caracteristici ale matricei de coocurență, cum ar fi omogenitatea, entropia și momentele de ordin superior ale matricei de coocurență. Ele indică de asemenea prezența înclinării, în schimb rezultatele nu sunt satisfăcătoare. Cele mai bune rezultate au fost obținute de către autor pentru contrastul matricei de coocurență, fiind comunicate în cursul anului 2000 [191].

Înclinarea unui plan poate fi evaluată plecând de la efectele de perspectivă ce se observă pe textura acestuia. Problemele ridicate sunt date în special de numărul mare de filtre Gabor necesare (cu diverse orientări și diverse frecvențe). Precizie bună se poate obține numai în cazul folosirii unui foarte mare număr de astfel de filtre care să extragă cât mai exact perioada fundamentală a texturii. Calculele implicate sunt laborioase.

Alte metode, cum ar fi cele bazate pe proprietățile matricei de coocurență pot fi mai rapide dar nu asigură o foarte bună precizie. Pot indica prezența unui gradient (înclinări) dar nu

oferă o exprimare cantitativă precisă, ele pot fi folosite pentru rapide evaluări calitative ale orientării suprafețelor.

5.2.12. Recunoașterea primitivelor texturii

Multe texturi nu sunt geometric foarte exacte, de exemplu textura « reptile », variațiile acesteia nu pot fi descrise în mod satisfăcător cu ajutorul gramaticii formei, dar pot fi descrise foarte bine de modele statistice.

Recunoașterea statistică a primitivelor este o paradigmă ce poate clasifica statistic variațiile primitivelor. Problema recunoașterii primitivelor se transformă într-o problemă de clasificare a eșantioanelor de textură dintr-o imagine într-o mulțime de clase. De exemplu, pentru imaginile obținute dintr-un avion, ne interesează să putem face diferența între următoarele clase : « câmp », « apa », « clădiri ». Noțiunea de bază în recunoașterea primitivelor este vectorul caracteristic. Vectorul caracteristic v este un set de măsurători $\{v_1, v_2, \dots, v_n\}$ care condensează descrierea celor mai relevante proprietăți ale unei imagini texturate, într-un mic spațiu Euclidian, n -dimensional, al caracteristicilor. Fiecare punct din spațiul caracteristicilor reprezintă o valoare pentru vectorul caracteristic aplicat pe diverse imagini texturate. Valorile măsurate pentru caracteristici trebuie să fie corelate cu membrii acestor clase.

În figura 6-34 a) este prezentat un spațiu bidimensional în care caracteristicile sunt corelate. Valorile pentru vectorul caracteristicilor sunt grupate în concordanță cu textura din care face parte. În figura 6.34 b) este prezentat un spațiu bidimensional în care caracteristicile nu sunt corelate, se vede foarte clar că este o greșită alegere a caracteristicilor ce sunt măsurate, aici acestea nu pot fi separate în clase diferite.

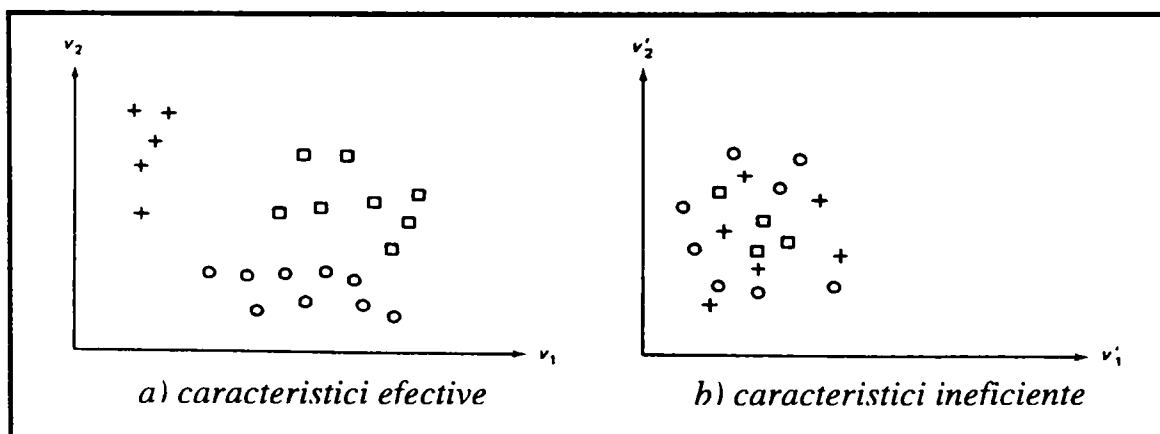


Figura 5-34 Spațiul caracteristicilor pentru discriminarea texturilor

Problema recunoașterii primitivelor se poate împărți în două faze: faza de antrenare și faza de testare. Uzual, pe durata fazei de antrenare, vectorii caracteristici din eșantioane cunoscute sunt folosiți pentru a partiționa spațiul caracteristicilor în regiuni ce reprezintă diferite clase. Selecția caracteristicilor se bazează pe modele parametrice sau nonparametrice ale distribuției punctelor în spațiul caracteristicilor. În unele cazuri parametrice se pot obține și soluții analitice, pentru cazul nonparametric, vectorii sunt grupați astfel încât să poată indica o anumită partiționare. Pe durata etapei de testare partițiile, spațiului caracteristicilor sunt folosite pentru clasificarea vectorilor caracteristici obținuți din eșantioane necunoscute. În figura 5-35 este arătat acest proces. Un mod obișnuit de a realiza o clasificare constă în utilizarea prototipurilor de puncte pentru fiecare clasă și o regulă de netezire a vecinătăților.

Un vector v aparține unei clase ω_i dacă pentru i se minimizează expresia :

$$\min_i d(v, v_{w_i}) \quad (5.160.)$$

unde v_{w_i} este prototipul punctelor din clasa w_i

Tehnicile parametrice presupun informații despre probabilitatea de distribuție a vectorului caracteristic pentru găsirea regulilor ce maximizează probabilitatea de clasificare corectă.

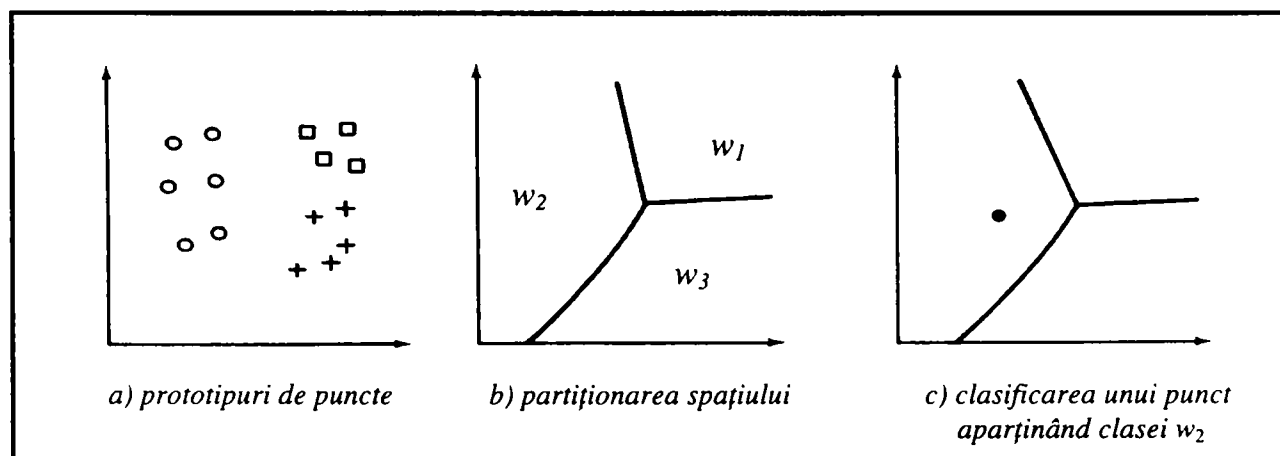


Figura 5-35 Clasificarea prin partiționarea spațiului

Un vector \mathbf{v} aparține unei clase ω_i dacă pentru i se maximizează expresia :

$$\max_i p(w_i|\mathbf{v}) \quad (5.161.)$$

Distribuția poate fi formulată și în sensul în care se minimizează probabilitatea de clasificare greșită.

În subcapitolul 5-4 sunt prezentate câteva tehnici de clasificare, ce pot fi folosite cu succes în clasificarea texturilor.

5.3. Caracteristicile obiectelor

Culorile și texturile sunt proprietăți utile care pot defini atât imaginile cât și obiectele prezente în acestea. Dar există proprietăți care se pot adresa în special obiectelor individuale. Pentru aplicații de indexare și căutare în baze de date este foarte important să cunoaștem proprietățile geometrice ale obiectelor, forma conturilor, dimensiunile și localizarea spațială în imagine. În cazul în care se folosește o căutare generală bazată pe obiectele din imagine cele mai importante proprietăți sunt legate de forma obiectului. Este foarte dificil să definim la modul general toate formele posibile, mai ales că există și un grad mare de subiectivism în această interpretare. De aceea se folosesc pentru caracterizare un set de momente invariante.

Fie un obiect R , reprezentat ca o imagine binară, atunci momentul de ordinul $(p+q)$ pentru forma obiectului se definește în următorul mod:

$$m_{p,q} = \sum_{(x,y) \in R} x^p y^q \quad (5.162.)$$

unde, p și q sunt numere naturale. Pentru a face aceste momente invariante la translație se definesc momentele centrate în următorul mod:

$$\mu_{p,q} = \sum_{(x,y) \in R} (x-x_c)^p (y-y_c)^q \quad (5.163.)$$

unde, x_c și y_c reprezintă coordonatele centrului de greutate. Dacă se dorește și o invarianță la scalare, se va normaliza momentul centrat în următorul mod:

$$\eta_{p,q} = \frac{\mu_{p,q}}{\mu_{0,0}^\gamma} \quad (5.164.)$$

unde, $\gamma = \frac{p+q+2}{2}$.

Pornind de la aceste momente au fost dezvoltate mai multe momente invariante la translație, rotație și diferențe de scală [192], [193].

Alte două caracteristici importante ale formei obiectelor sunt circularitatea și orientarea axei majore [194]. Se pot defini mai multe proprietăți geometrice ale obiectelor sau ale formelor acestora. Numărul acestora, prezentat în literatură este foarte mare, în special pentru forme specificate sau cazuri particulare.

Momentele geometrice sunt folosite cu succes în descrierea formelor geometrice. O importanță deosebită se acordă acestora, deoarece este posibilă descrierea formelor dar și reconstrucția unor imagini pornind de la acestea [195].

A. G. Mamistvalov propune mai multe momente invariante n -dimensionale pentru recunoașterea obiectelor. Suportul teoretic al acestora este teorema generalizată fundamentală a momentelor invariante (GFTMI), ea fiind adaptată pentru recunoașterea formelor geometrice regulate, sau a unor obiecte solide [196].

5.4. Tehnici de căutare în bazele de imagini

5.4.1. Modurile de utilizare ale caracteristicilor imaginilor folosite la căutare

Dezvoltarea rapidă a aplicațiilor ce folosesc baze de imagini a determinat și apariția unor probleme de căutare ale imaginilor atunci când dimensiunea bazei de imagini atinge valori foarte mari [197].

Algoritmii de indexare ce se bazează pe denumirea fișierului sau pe structurarea informațiilor în diverse directoare nu se dovedesc eficienți în procesul de căutare, datorită faptului că nu conțin nici o informație despre conținutul imaginilor din bază [198]. De aceea, algoritmul propus de autor va efectua o căutare bazată pe conținut. Într-un sistem de căutare a imaginilor bazat pe conținut, principalul obiectiv este de a găsi imaginile din baza de date ce au un conținut similar cu imaginea descrisă în procesul de interogare. Există mai multe moduri în care se poate realiza interogarea:

- Direct, prin folosirea limbajului de interogare - acest mod presupune că utilizatorul cunoaște care este sistemul de obiecte de tip imagine în cadrul bazei de date, schema bazei de date și vocabularul folosit. Similaritatea pentru cuvinte ce au aceeași ordine a literelor am evaluat-o folosind următoarea formulă:

$$Sim_{cuv}(C1, C2) = \sum_i w_i \left(\frac{l_i}{l_{cuv}} \right)^{p_i} \quad (5.165.)$$

unde, l_i - lungimea factorului comun dintre cele două cuvinte

w_i - coeficient de ponderare

p_i - parametrul ce crește ponderea elementelor comune

l_{cuv} - media lungimii celor două cuvinte

- Prin folosirea unor obiecte conceptuale în imagine - folosind o combinație între descrierea conceptuală a obiectelor din imagine și modul de plasare a acestora în imagine, putem

obține rezultate foarte bune.

- Prin folosirea de imagini exemplu - utilizatorul oferă o imagine exemplu, iar sistemul trebuie să determine imaginile cu un conținut similar.
- Prin asociere – o imagine este asociată cu texte ce descriu imaginea respectivă. În acest caz căutarea după conținutul imaginilor este echivalentă cu o căutare într-un document de tip text. Deși nu este un algoritm de căutare foarte puternic, uneori unele proprietăți din punct de vedere al imaginii sunt mai ușor de exprimat prin cuvinte decât prin expresii matematice deosebit de complexe.

După determinarea celor mai reprezentative proprietăți, ce definesc o imagine, este necesar să se definească un mod de măsurare a similarității dintre două imagini. Această similaritate trebuie să țină cont de modul de interpretare al sistemului uman, de multe ori relațiile matematice nu pot descrie perfect ceea ce sistemul de percepție uman definește a fiind “similaritate”.

Algoritmul implementat de autor conform descrierii din cadrul capitolului 6, propune calcularea unor caracteristici pentru fiecare imagine din baza de imagini, ce vor permite identificarea imaginii căutate sau găsirea celei mai apropiate asemănări dintre o imagine dată și o imagine din cadrul bazei. Alegerea celor mai reprezentative caracteristici permit creșterea vitezei de căutare, a eficienței și în special a gradului de clasificare corectă. Aplicația poate fi folosită în diverse domenii cum ar fi managementul imaginilor medicale, biblioteci multimedia, arhive de documente, muzee, informații geografice, identificarea criminalilor, etc.

Căutarea imaginilor într-o bază de date s-a dovedit a fi de cele mai multe ori o căutare bazată pe similitudini și nu neapărat pe o căutare exactă. S-a constatat că foarte important ar fi pentru un utilizator să existe o interfață interactivă care să-i permită selectarea progresivă a caracteristicilor imaginilor și o interpretare parțială a rezultatelor intermediare de căutare.

Ca și consecință a acceptării existenței interactivității dintre utilizator și baza de date este interfața grafică ce trebuie să suporte un sistem de interogare prin care utilizatorul să formuleze criteriile de căutare, iar baza de date să furnizeze răspunsuri care să fie evaluate de către utilizator. Algoritmii permit utilizatorului să atribuie fiecărui element obținut după o primă căutare un coeficient de relevanță ce va redefini parametrii de căutare [199]. După o evaluare sunt necesare redefiniri ale criteriilor de căutare pentru a crește finețea de căutare a imaginii dorite.

Limbajul de interogare trebuie să permită obținerea de informații de la utilizator legate de conținutul imaginilor. Se impune ca acesta să permită:

- selecționarea, derivarea și calculul caracteristicilor imaginilor și a obiectelor să se facă ușor și să fie expresive;
- utilizarea metodelor bazate pe similitudine;
- întrebări expresive care să caracterizeze corect și intuitiv ceea ce dorește utilizatorul. În accepția generală, o bază de date indexată constă într-un set de intrări (o intrare pentru fiecare dată) conținând valoarea atributului cheie pentru acele date și care sunt de fapt pointeri ce permit accesul imediat al datelor.

Se folosesc pentru schemele multidimensionale de indexare trei mari categorii de metode: R-trees, Liniar quad-trees, Grid files. Acestea pot fi compuse cu metode de comparare bazate pe rețele neuronale sau alte metode de discriminare (diverse distanțe și metrice). Dacă dimensiunea bazei de imagini este foarte mare, performanțele acestor trei mari categorii se apropie foarte mult de varianta unei scanări secvențiale. De exemplu, atunci când se folosește arborele liniar pătratic, timpul de căutare crește în progresie geometrică cu dimensiunea. Dacă se folosește modul “grid files” se constată aceeași creștere. Folosirea modului R-trees este eficientă

doar pentru valori mai mari de 2 ale divizărilor în noduri și pentru o dimensiune a arborelui sub 20. Aceste limitări sunt atribuite în general faptului că se folosește distanța Euclidiană pentru căutare, iar într-un spațiu cu foarte multe dimensiuni puterea de calcul implicată este foarte mare. În cazul indexărilor imaginilor, ținând cont de redundanța informației ce există în imagini, se poate folosi ca metrică de comparare între imagini și distanța Mahalanobis.

5.4.2. Similaritatea imaginilor bazată pe proprietățile globale ale imaginilor

O cale simplă de evaluare a similarității se bazează pe maparea exactă sau parțială. Se știe că în imagini pot să apară ușoare distorsionări din diverse motive, iar căutarea se va rezuma la identificarea unei porțiuni din imagine.

Se pot folosi pentru aceasta arbori de căutare ce folosesc ca principiu împărțirea imaginii succesiv în patru. Se obține astfel o descriere sub formă de arbore echilibrat al imaginii. Se va realiza căutarea prin parcurgerea unui bloc de o anumită dimensiune, corelată cu ceea ce se caută, peste întreaga imagine. La descoperirea unui bloc ce se apropie de cel căutat, se vor evalua vecinii descriși prin arborele imaginii, crescând astfel viteza de calcul.

Distribuția culorilor

Distribuția globală a culorilor este una dintre cele mai importante și utile caracteristici, în special atunci când în imagini sunt foarte multe texturi, iar tehnicile de segmentare nu sunt utile. Distribuția culorilor este cel mai bine reprezentată de către histogramă și se apropie cel mai mult de reprezentarea globală, ea nu ține cont de obiectele prezente în imagine sau cum este compusă această imagine. Histograma este invariantă la translație și rotație.

Histogramele pot fi comparate printr-o operație de intersecție [200]. Fie Q și P histogramele a două imagini, de exemplu imaginea ce se dorește a se verifica dacă există în baza de date și o imagine aleasă din baza de date pentru comparare. Dacă imaginea are N culori, se poate defini intersecția ca fiind:

$$I = \sum_{i=1}^N \min(Q_i, P_i) \quad (5.166.)$$

O altă cale de cuantizare a spațiului color este aplicarea unor tehnici de grupare - "clustering" cu scopul de a determina cele mai importante grupări dintr-o histogramă, iar comparația să se realizeze între grupările a două histogramme și nu între toate valorile histogrammei obținându-se astfel o reducere a timpului de calcul. Dezavantajul acestei metode constă în faptul că nu permite căutări pe subregiuni ale imaginilor. Se poate lucra și pe partiții ale imaginii pentru a dezvolta facilitățile de localizare. Volumul calculului va crește deoarece vor fi mai multe histogramme ce trebuie calculate și comparate.

Localizarea unor anumite obiecte din imagine de exemplu folosește proprietățile locale ale culorilor obiectelor. Pentru identificarea obiectelor în general se folosește o detecție bazată pe segmentarea color a imaginilor. Se poate realiza o detecție și folosind culori predefinite pentru căutare. Măsurarea similarității dintre două imagini prin evaluarea culorilor - ca metode folosesc histogrammele și momentele histogrammei [201], [202].

Similaritatea pornind de la histogramme este definită de Apers astfel:

$$Sim_{color}(Im1, Im2) = \frac{1}{N} \sum_{j=1}^N \left(1 - \frac{|Hist_{Im1}(c_j) - Hist_{Im2}(c_j)| \cdot N}{\max(N \cdot Hist_{Im1}(c_j), N \cdot Hist_{Im2}(c_j))} \right) \quad (5.167.)$$

$N \cdot Hist_{Im1}(c_j)$ - reprezintă normalizarea histogrammei în funcție de numărul de niveluri.

În cazul folosirii unei descompunerii a imaginii pe mai multe planuri de culoare, evaluarea similarității se face analizând global, pentru toate planurile de culoare sau pe fiecare plan independent. În multe aplicații, rezultate mai bune am obținut folosind pentru descompunere modelul de culoare CIELUV.

Un sistem de indexare propus și testat este prezentat în figura următoare:

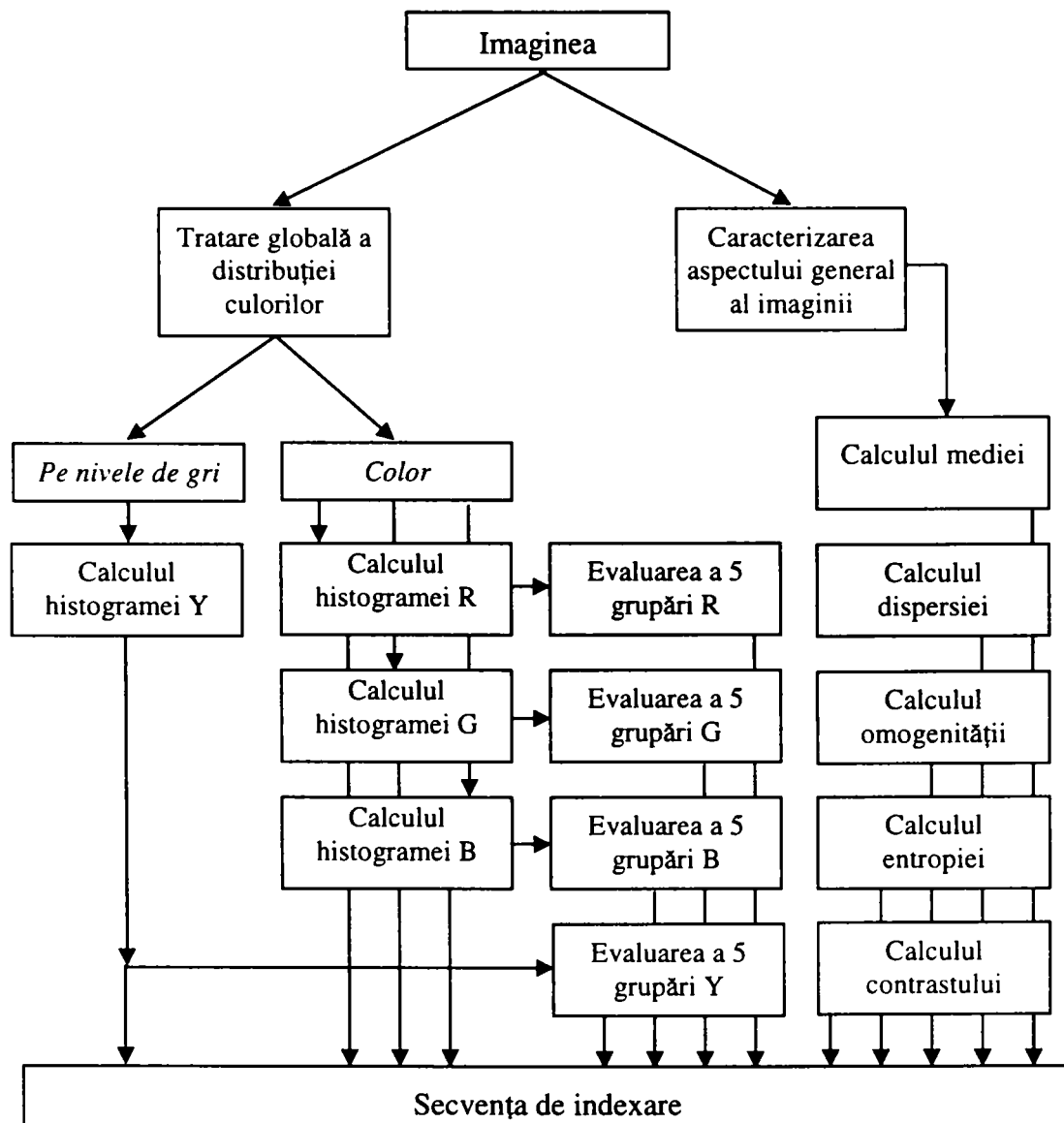


Figura 5-36 Schemă de indexare ce folosește numai caracteristicile globale ale imaginii

Sistemul de indexare propus prezintă următoarele avantaje:

- permite o căutare bazată pe conținutul global al imaginii;
- caracteristicile imaginii după care se face indexarea sunt alese în așa fel încât să fie sugestive pentru utilizator – omogenitate, entropie, contrast;
- este posibilă combinarea a mai multor caracteristici pentru a crește puterea de indexare;

Dezavantajele algoritmilor de indexare sunt legate de numărul mare de caracteristici extrase pentru fiecare imagine din baza de imagini, ceea ce presupune un timp mare de calcul.

Experimentările au fost realizate pe o bază ce conținea peste 400 de imagini. Pentru modulul de tratare globală diferența de viteză de calcul la căutare în comparație cu varianta comparării imagine cu imagine este dependent de dimensiunea imaginii. Pentru imagini de dimensiune 256 x 256, cu 256 de nivele de gri au fost necesare doar 256 de comparații plus 255 operații de însumare în loc de 65536 comparații. Pentru imagini de dimensiune și mai mare

sporul de viteză este considerabil mai mare. Dacă se folosesc numai 5 grupări reprezentative ale histogramei se va reduce numărul de operații la 5 detectări de grupări și 5 comparații.

Caracterizarea aspectului general al imaginii, prin folosirea a numai 5 caracteristici, sugestive pentru marea majoritate a imaginilor, permite lucrul într-un spațiu Euclidian cu relativ puține dimensiuni, fără a complica foarte mult calculele. Am preferat folosirea a doar 5 momente centrate pentru a oferi o structură modulară pe grupuri de câte 5 operații, grupare ce oferă o posibilă paralelizare pentru mărirea vitezei de indexare a imaginilor.

Deci, o secvență de indexare va fi formată din: un flag care să indice dacă este o imagine color sau pe nivele de gri, 3 histogramme pe 256 de nivele, pentru fiecare componentă de culoare vor fi 5 perechi de numere care să caracterizeze cele mai importante grupări, plus 5 valori pentru descrierea aspectului general al imaginii. Deci în total 1612 octeți, aceștia se vor adăuga pentru fiecare imagine din baza de date. În consecință, va rezulta o creștere a acesteia cu maxim 2% dacă imaginile sunt în jur de 100 kocteți. În general, se folosesc imagini mult mai mari și deci raportul în care va crește baza de date este nesemnificativ. Detalii privind acest mod de indexare sunt prezentate de autor în [203], [204].

Descompunere multirezoluție a imaginilor

Descompunerea imaginilor multirezoluție, și compararea imaginilor folosind sub-benzile, oferă posibilitatea implementării de algoritmi de căutare și indexare ce folosesc ca principiu de lucru, verificarea rapidă a similarității, pe un număr redus de sub-benzi. Dacă rezultatele sunt apropiate, se poate trece la o redefinire a rezoluției de lucru, folosindu-se tot mai multe benzi. Etapele parcurse pentru realizarea unei indexări prin descompunere multirezoluție sunt:

- Definirea spațiului color de lucru (RGB, HSV, YIQ) – pentru comparații, imaginile trebuie să folosească aceeași decompoziție a culorilor, dacă nu, trebuie realizată o conversie;
- Definirea modului de realizare a decompoziției – uni sau bidimensională;
- Alegerea tipului de undișoară folosită – se recomandă pentru performanțe și simplitate utilizarea undișoarei Haar;
- Se definesc modalitățile de trunchiere, cuantizare și normalizare.
- Se definește metrica folosită.

O metrică des întâlnită este cea Euclidiană, aceasta însă nu oferă viteze mari de lucru și rezultatele sale pot fi contradictorii. De aceea, pentru a mări viteza de căutare se folosește următoarea metrică:

$$\|Q, T\| = w_{0,0}|Q[0,0] - T[0,0]| + \sum_{i,j} w_{i,j} |\tilde{Q}[i, j] - \tilde{T}[i, j]| \quad (5.168.)$$

unde,

- Q, T – sunt descompunerile imaginilor comparate, în spațiul culorilor pe un singur plan.
- $Q[0,0], T[0,0]$ – sunt coeficienți funcției de scalare, ce corespund mediei planului color al imaginilor.
- $\tilde{Q}[i, j], \tilde{T}[i, j]$ – sunt coeficienți undișoară, trunchiați și cuantizați ai Q , respectiv T .

Pentru optimizarea și mărirea vitezei de lucru sunt disponibile în literatură mai multe metode [205].

5.4.3. Evaluarea similarității folosind mai multe caracteristici

Pentru evaluarea similarității dintre două imagini în spațiul caracteristicilor (spațiu cu un număr relativ mare de dimensiuni), autorul a ajuns la concluzia că se recomandă și folosirea tehnicilor fuzzy. Într-un spațiu fuzzy cu n dimensiuni, distanța dintre două imagini se definește astfel:

$$Dist_{fuzzy} = \sum_{i=1}^n \left(|Im1_i - Im2_i| \times \sum_{k=1}^n cor(M_i, M_k) \times |Im1_k - Im2_k| \right) \quad (5.169.)$$

unde, $cor(M_i, M_k)$ - este corelația dintre mulțimile fuzzy M_i, M_k .

M_1, M_2, \dots, M_n - spațiul mulțimilor fuzzy alocate caracteristicilor imagini; $Im1_i, Im2_i$ - valoarea caracteristicii i , din vectorii caracteristicilor $Im1$ și $Im2$ ce descriu imaginile.

În procesul de redefinire a parametrilor de căutare se poate folosi următoarea metodă de redefinire: fie T , valoarea curentă a unei caracteristici F , și σ dispersia caracteristicii F în baza de date. Schimbarea relativă a valorii caracteristicii F se definește astfel:

$$Sr = \frac{\sigma \times K \times \delta}{N} \quad (5.170.)$$

unde,

- K este numărul de imagini returnate utilizatorului,
- N numărul de occurențe ale caracteristicii F , în baza de date
- δ este mărimea schimbării specificată de utilizator

O nouă valoare pentru caracteristica folosită la căutare este dată de relația:

$$F' = F + Sr \times \sigma \quad (5.171.)$$

În foarte multe situații este necesar ca utilizatorul să acorde diverse grade de încredere pentru o anumită caracteristică, de aceea în procesul de căutare am folosit următoarea relație:

$$Dist_{inc}(Im1, Im2) = \sum_i \left(W_i \times Seg_i \times Inc_i \times \frac{|Im1_i - Im2_i|}{\sigma_i} \times C_i \right) \quad (5.172.)$$

- W_i este coeficientul caracteristicii i (este sau nu folosită caracteristica în calcul)
- Seg_i certitudinea de segmentare a unei caracteristici
- Inc_i gradul de încredere acordat de utilizator
- C_i coeficient de scalare pentru dispersia caracteristicii i
- $Im1_i, Im2_i$ caracteristicile i ale imaginilor
- σ_i dispersia caracteristicii i .

5.4.4. Căutarea în baze de date utilizând mai multe categorii de caracteristici (caracteristicilor globale, caracteristici stohastice ale imaginilor, caracteristici ale matricii de coocurență, descompuneri pe baza filtrelor Gabor)

Acest paragraf prezintă mai multe metode de căutare experimentate de către autor, ce combină în procesul de discriminare caracteristici globale și stohastice ale imaginilor, caracteristici ale matricii de coocurență și descompuneri bazate pe filtre Gabor (acestea simulează cel mai bine sistemul vizual uman). Pentru clasificare folosesc mai multe metode ce implică diverse distanțe (între caracteristici, sau histograme), diverse moduri de definire a similarității ce există între două imagini, metode cu prag, clasificator neuronal și unele combinații ale acestora în sisteme de căutare bazate pe structuri arborescente.

Textura este recunoscută ca fiind una dintre cele mai importante proprietăți ale unei regiuni. Textura poate fi analizată atât color cât și folosind nivele de gri, unul dintre avantajele folosirii texturilor este faptul că oferă informații importante despre imagine chiar și în varianta cu nivele de gri ce nu necesită un volum mult mai mic de calcule față de varianta color.

Analiza texturilor se face folosind două mari clase de metode: metode structurale și metode statistice. Metodele structurale urmăresc identificarea primitivelor texturii și modul lor de repetiție. Aceste metode nu pot fi folosite în cazul texturilor neregulate, de aceea este necesară și folosirea metodelor statistice ce urmăresc caracterizarea texturilor din punctul de vedere a distribuției spațiale ale nivelelor din imagine. Se folosesc metode bazate pe estimarea spectrului folosind analiza Fourier sau utilizând descompuneri unitare spațiu-frecvență. Alte subclase de analiză folosesc matricele de co-ocurență, modele MRF (Markov Random Field), modele fractale, etc. Cele mai importante caracteristici ce sunt extrase pentru texturi se referă în general la proprietăți ce pot avea o echivalență în acceptul uzual de zi cu zi, cum ar fi: granularitatea, contrastul, directivitatea, omogenitatea sau entropia. Acestea se pot calcula plecând de la imaginea originală sau pornind de la matricea de co-ocurență.

Pentru discriminarea și clasificarea texturilor autorul a folosit rețele neuronale, rezultatele experimentărilor fiind publicate în anul 1999 [206]. Acestea și-au demonstrat puterea de discriminare folosind și un număr relativ mic de caracteristici extrase.

Pentru realizarea unei căutări cu o discriminare cât mai performantă, într-un timp cât mai scurt, am folosit următoarele caracteristici extrase din matricea de coocurență. Prezentarea în detaliu a caracteristicilor ce se pot extrage prin intermediul matricii de coocurență (conține o cantitate foarte mare de informație referitoare la poziția pixelilor ce au valori similare pentru nivelurile de gri) a fost făcută în capitolul 5.1.3.

Caracteristicile derivate din matricea de coocurență și caracteristicile statistice folosite sunt :

$$\text{Omogenitatea: } \frac{1}{N_c} \sum_a \sum_b (\text{MC}_t(a,b))^2 \quad (5.173.)$$

$$\text{Entropia: } 1 - \frac{1}{N_c \cdot \ln(N_c)} \sum_a \sum_b \text{MC}_t(a,b) \cdot \ln(\text{MC}_t(a,b)) \cdot \mathbf{1}_{\text{MC}_t(a,b)} \quad (5.174.)$$

$$\text{Contrastul: } \frac{1}{N_c (L-1)^2} \sum_{k=0}^{L-1} k^2 \sum_{|a-b|=k} \text{MC}_t(a,b) \quad (5.175.)$$

$$\text{Dispersia: } m_2 = \frac{1}{K} \sum_{s \in R} (A[s] - m_1)^2 \quad (5.176.)$$

$$\text{Media: } m_1 = \frac{1}{K} \sum_{s \in R} A[s] \quad (5.177.)$$

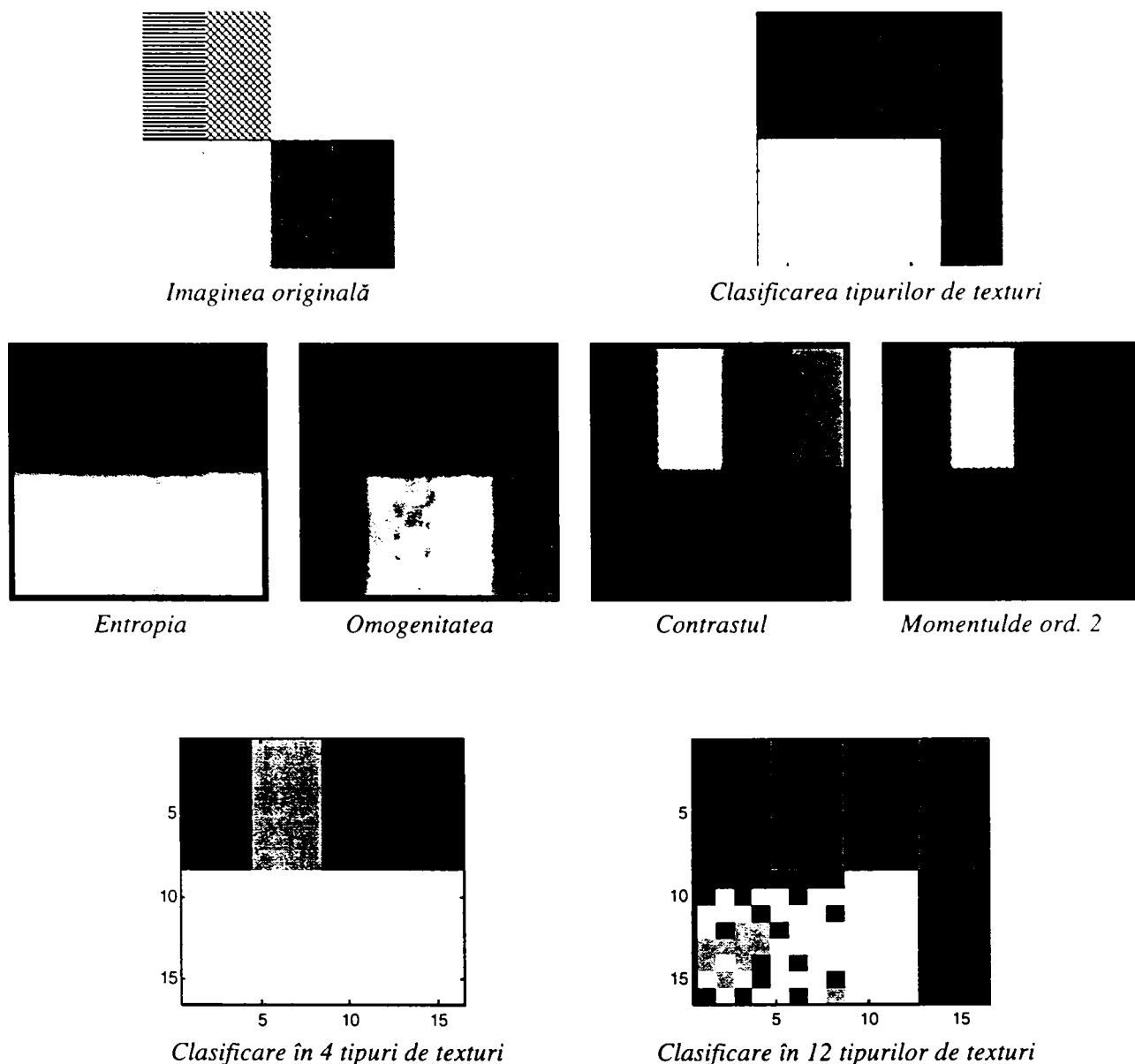


Figura 5-37 Rezultatele clasificării texturilor pe număr impus de clase

Metoda creează un vector din ferestrele locale ale caracteristicilor și ale imaginii originale concatenând valoarea fiecărui punct din fereastră. Acest vector se aduce la intrarea unei rețele neuronale de tip hartă de trăsături cu auto organizare (self-organising features map neural network – SOFM NN). Rețeaua neuronală va încerca să clasifice vectorul de intrare într-un număr apriori definit de clase. Pentru antrenare se folosește un algoritm de tip competitiv, ce forțează numărul de neuroni de ieșire spre un număr de clase specificat.

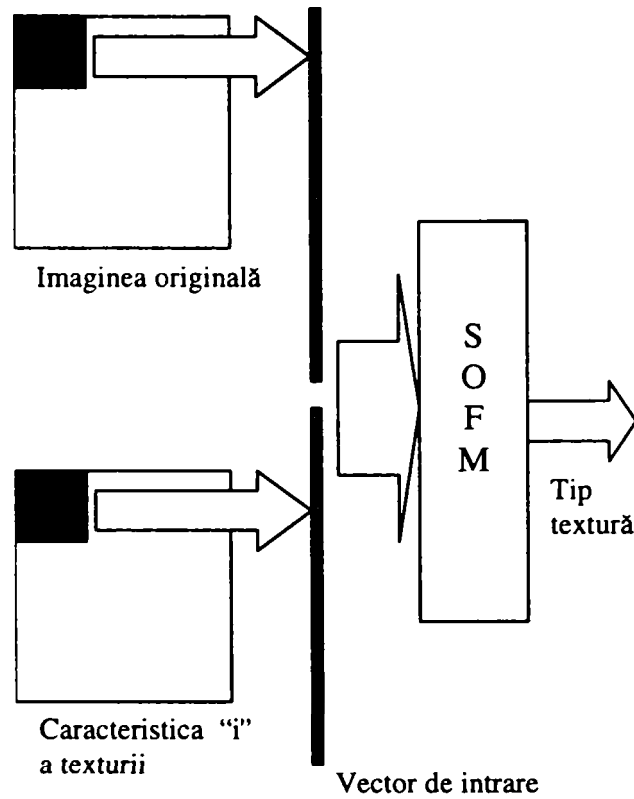


Figura 5-38 Diagrama sistemului pentru clasificarea texturilor pe diverse tipuri

Detalii privitoare la experimentările și implementările efectuate de autor au fost publicate de autor în țară [207] și în străinătate [208].

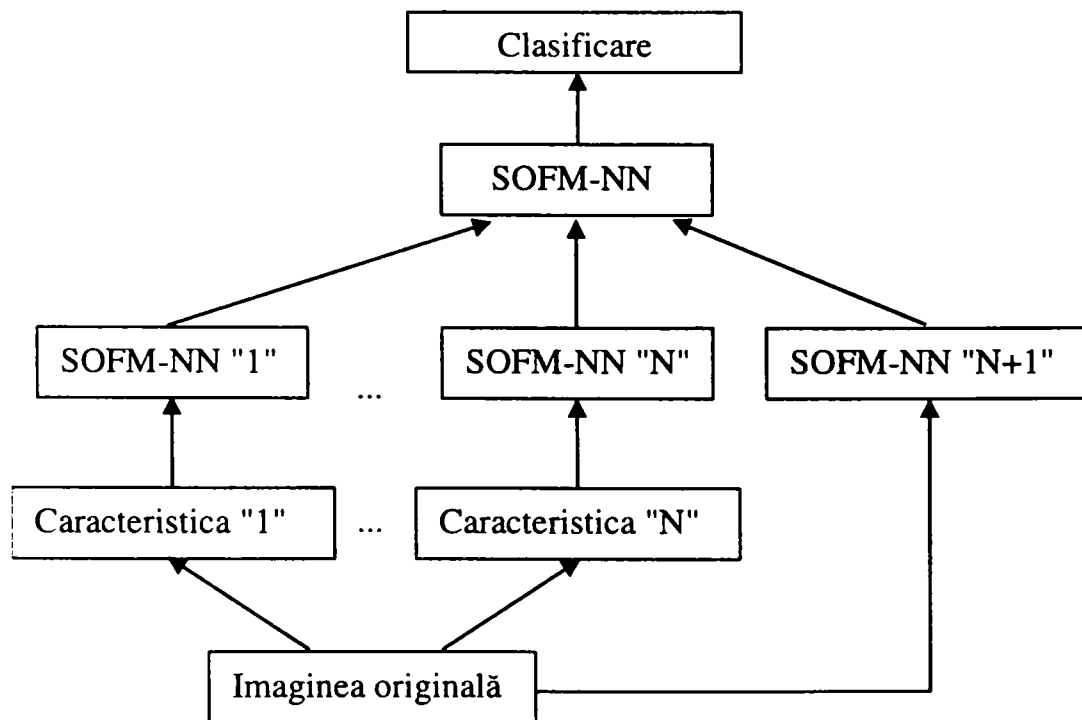


Figura 5-39 Diagrama sistemului pentru clasificarea texturilor pe diverse tipuri

În urma experimentărilor am ajuns la rezultate mult mai bune prin folosirea a două nivele de clasificatori. Se folosește un prim strat ce constă în aplicarea la intrarea rețelei neuronale a imaginii prelucrate (a unei caracteristici extrase din imagine), urmând ca decizia finală să fie

luată de către cel de al doilea nivel ce folosește la intrare rezultatele obținute la ieșirea primului nivel.

În tabelul următor sunt sintetizate câteva valori pentru caracteristici, pentru a scoate în evidență importanța caracteristicilor extrase din matricea de coocurență:

Textura	Media	Dispersia	Caracteristici matrice co-ocurență		
			C	O	E
T1	127.5	16.26	31	20	240
T2	130.85	3.3	193	121	205
T3	173.38	0.8	120	110	125
T4	191.25	12.19	224	235	181
T5	184.08	0.44	63	236	209
T6	65.185	0.57	203	114	218

În tabel sunt prezentate caracteristicile a 6 texturi exemplificate în figura 5-40. Se observă că sunt texturi care diferă ca aspect foarte mult, deși au aceleași nivele de gri, iar diferențele dintre nivele sunt relativ mici. Pot exista texturi care să aibă aceeași histogramă, de exemplu o textură formată din linii albe și negre care sunt orientate o dată pe verticală și o dată pe orizontală. De exemplu, textura T1 rotită cu 90 de grade. Pentru a diferenția cele două texturi un rol important îl vor avea caracteristicile matricei de co-ocurență [209],[210].

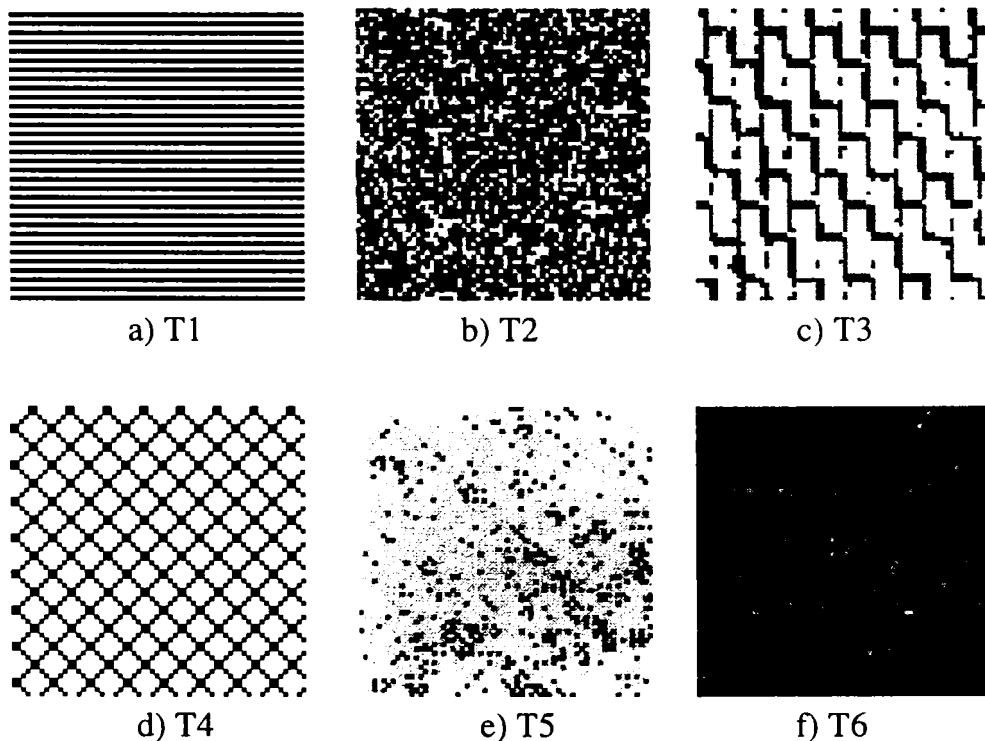


Figura 5-40 Texturi folosite ca imagini de test în procesul de căutare

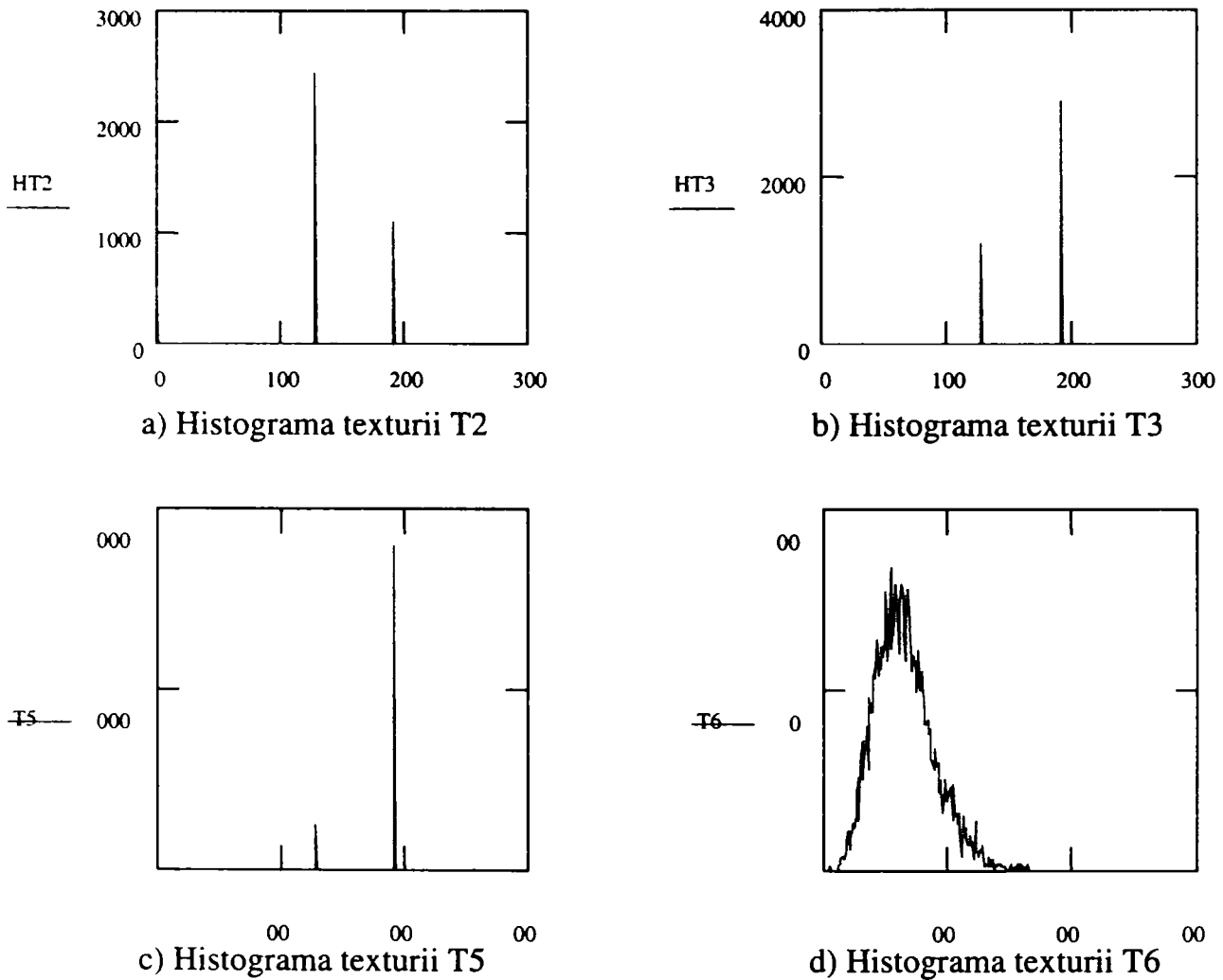


Figura 5-41 Histogramele texturilor T2, T3, T5 și T6

Căutarea în baza de texturi folosind descompuneri realizate cu filtre Gabor

Se formează un vector al caracteristicilor format din valorile medii și dispersiile decompunerii imaginii originale prin mai multe filtre Gabor [211].

$$V = [\mu_{00}, \sigma_{00}; \dots; \mu_{0N}, \sigma_{0N}; \mu_{10}, \sigma_{10}; \dots; \mu_{1N}, \sigma_{1N}; \dots; \mu_{MN}, \sigma_{MN}]$$

Distanța dintre două imagini se poate defini astfel:

$$d(i, j) = \sum_{m=0}^M \sum_{n=0}^N d_{mn}(i, j) \quad (5.178.)$$

$$\text{unde, } d_{mn}(i, j) = \left| \frac{\mu_{mn}^i - \mu_{mn}^j}{\alpha(\mu_{mn})} \right| + \left| \frac{\sigma_{mn}^i - \sigma_{mn}^j}{\alpha(\sigma_{mn})} \right| \quad (5.179.)$$

$\alpha(\mu_{mn}), \alpha(\sigma_{mn})$ - reprezintă dispersia.

O altă modalitate de selecție a filtrelor pornind de la domeniul frecvență poate fi realizată astfel:

$$D(u, v) = \frac{\left[|F_{in}(u, v)| - |F_{med}(u, v)| \right]^2}{F_{var}(u, v)} \quad (5.180.)$$

- $F_{in}(u, v)$ - transformata Fourier a imaginii de intrare
- $F_{med}(u, v)$ - media trasformatelor Fourier pentru toate imaginile din baza de date
- $F_{var}(u, v)$ - dispersia trasformatelor Fourier pentru toate imaginile din baza de date
- $D(u, v)$ - diferența de energie normalizată pentru fiecare frecvență (u, v)

Fiecare filtru este evaluat pe baza diferenței totale de energie, evaluare ce se realizează pentru întreg spectrul de frecvențe:

$$\xi_{mn} = \sum_u \sum_v D(u, v) \cdot |G_{mn}(u, v)|^2 \quad (5.181.)$$

unde $G_{mn}(u, v)$ este răspunsul în frecvență al filtrului $g_{mn}(x, y)$. Cea mai mare valoare ξ_{mn} reprezintă filtrul cu cele mai bune performanțe.

$$g(x, y) = \frac{1}{2\pi\sigma_x\sigma_y} \cdot \exp\left[-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2}\right) + 2\pi j W x\right] \quad (5.182.)$$

$$G(u, v) = \exp\left\{-\frac{1}{2}\left[\frac{(u-W)^2}{\sigma_u^2} + \frac{v^2}{\sigma_v^2}\right]\right\} \quad (5.183.)$$

$$\sigma_v = \frac{1}{2\pi\sigma_y} \quad \sigma_u = \frac{1}{2\pi\sigma_x}$$

Se poate utiliza cu succes o abordare cu undișoare a analizei texturilor folosind ca funcție de bază funcția Gabor.

$$g_{mn}(x, y) = a^{-m} g(x', y') \quad (5.184.)$$

$$x' = a^{-m}(x \cos \theta + y \sin \theta) \text{ iar } y' = a^{-m}(-x \sin \theta + y \cos \theta). \quad (5.185.)$$

unde, $\theta = \frac{n\pi}{K}$, K este numărul de orientări, $a > 1$, m, n - numere întregi..

$$a = \left(\frac{U_H}{U_L}\right)^{\frac{1}{S-1}}, \text{ unde } U_H \text{ și } U_L \text{ sunt frecvențele inferioare și superioare de interes}$$

$$\sigma_u = \frac{(a-1)U_H}{(a+1)\sqrt{2\ln 2}}, \quad \sigma_v = \tan\left(\frac{\pi}{2k}\right) \left[U_H - 2 \ln\left(\frac{\sigma_u^2}{U_H}\right) \right] \left[2 \ln 2 - \frac{(2 \ln 2)^2 \sigma_u^2}{U_H^2} \right]^{-\frac{1}{2}} \quad (5.186.)$$

unde $W = U_H$ și $m=0, \dots, S-1$.

Sistem de căutarea în baza de imagini bazat pe similaritatea dintre texturi folosind descompuneri realizate cu filtre Gabor și clasificatori reprezentați de rețele neuronale

Modalitatea de căutare în baza de date este prezentată structural în figura următoare:

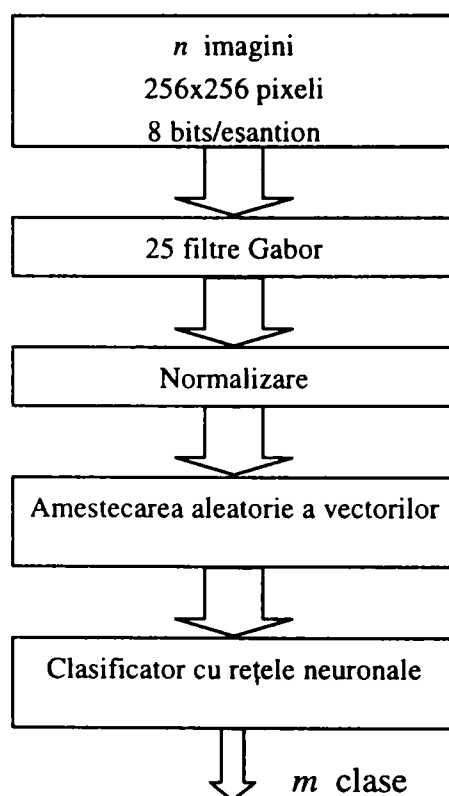


Figura 5-42. Diagrama sistemului de căutare în baza de imagini ce folosește o descompunere cu filtre Gabor și clasificatori ce folosește rețele neuronale

Primul pas este descumpunerea imaginilor prin intermediul a 25 de filter Gabor, alese în așa fel încât să acopere întreg spectrul de frecvențe, iar orientările filtrelor să fie orientări uzuale pentru sistemul vizual uman (orizontale, verticale și câteva diagonale).

Metoda se bazează pe un set de imagini caracteristici obținute prin descompunerea imaginii originale folosind filtre Gabor. Modul de alegere al frecvențelor de analiză și a dimensiunii spațiale a filtrelor a fost prezentată în subcapitolele anterioare.

Următorul pas este normalizarea datelor. Pentru o bună funcționare a rețelei neuronale este necesară normalizarea datelor obținute de la cele 25 de filtre Gabor. Am folosit două metode:

- $mnmx$, unde -1 este valoarea minimă iar 1 este valoarea maximă.
- $mn0std1$, unde media este zero iar deviația standard este 1.

Vectorii obținuți sunt împărțiți în două seturi, un set pentru antrenament și un set pentru testare.

Două rețele neuronale am folosit pentru clasificarea imaginilor pe baza similarității dintre imaginile comparate:

- Multilayer perceptron (MLP) este unul dintre cei mai des utilizați clasificatori neuronali. MLP poate învăța să genereze o mapare din spațiul vectorilor de la intrare spre spațiul vectorilor de ieșire prin minimizarea erorii dintre actuala ieșire și cea dorită.

- Radial Basis Function (RBF) este o metodă nesupervizată de clasificator neuronal ce folosește o bază de funcții radiale.

Arhitecturile celor două tipuri de rețele sunt date în figura următoare:

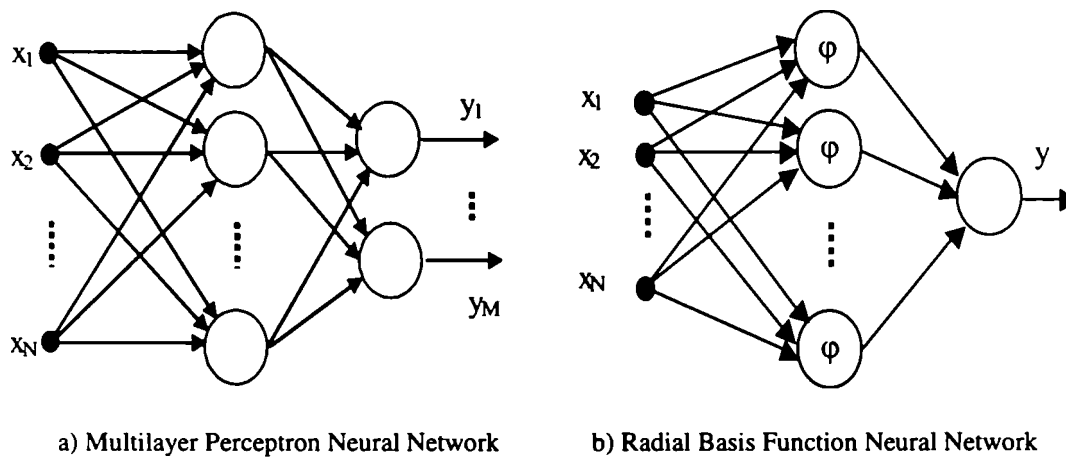


Figura 5-43. Structuri de rețele neuronale folosite pentru măsurarea similarității dintre două imagini comparate

Datorită complexității și subiectivismului ce caracterizează sistemul vizual uman ce trebuie modelat, o singură caracteristică nu poate fi folosită pentru discriminarea tuturor imaginilor, chiar dacă există cazuri particulare unde rezultatele pot fi foarte bune. Algoritmul propus reprezintă o combinație între caracteristicile filtrelor Gabor și un clasificator neuronal.

Pentru testare au fost folosite texturi Brodatz. Acestea au fost procesate folosind 25 de filtre Gabor. Am utilizat seturi de 56 de imagini pentru antrenare și câte 4 imagini pentru testare. Figura 5-44 ... 5-47 reprezintă rezultatele experimentale pentru texturile D52, D4, D44 și D26. Imaginile găsite de sistem ca fiind cu cea mai mare similaritate sunt prezentate în continuare:

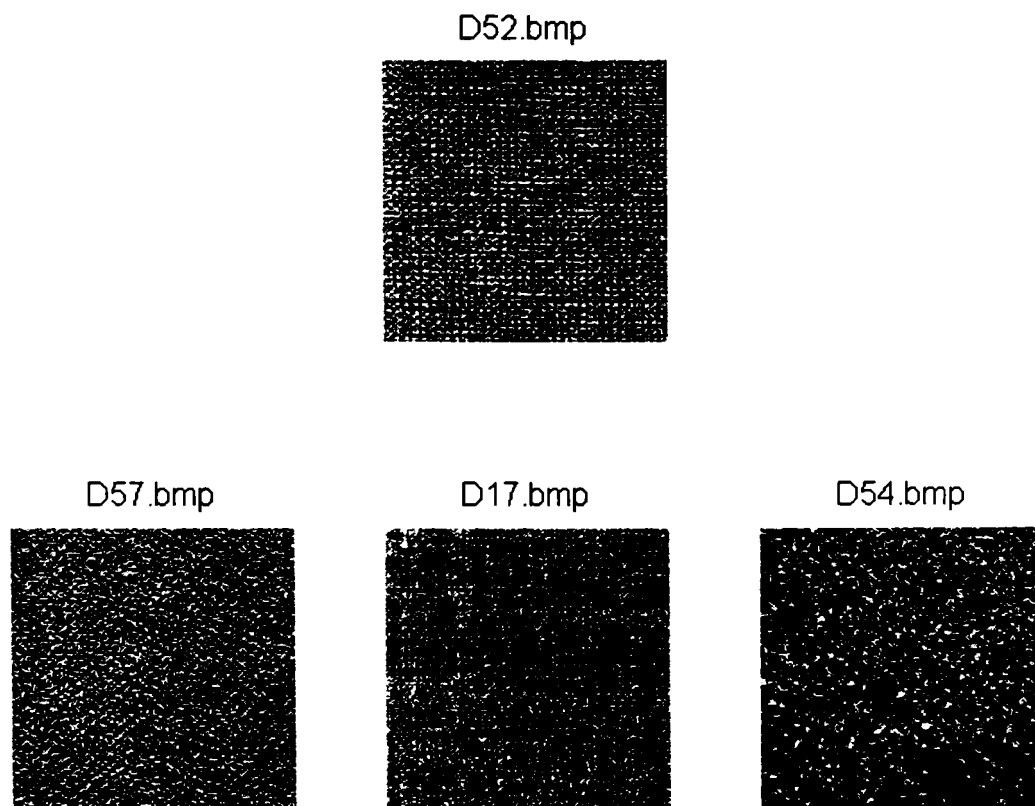


Figura 5-44. Rezultatele căutării pentru imaginea D52

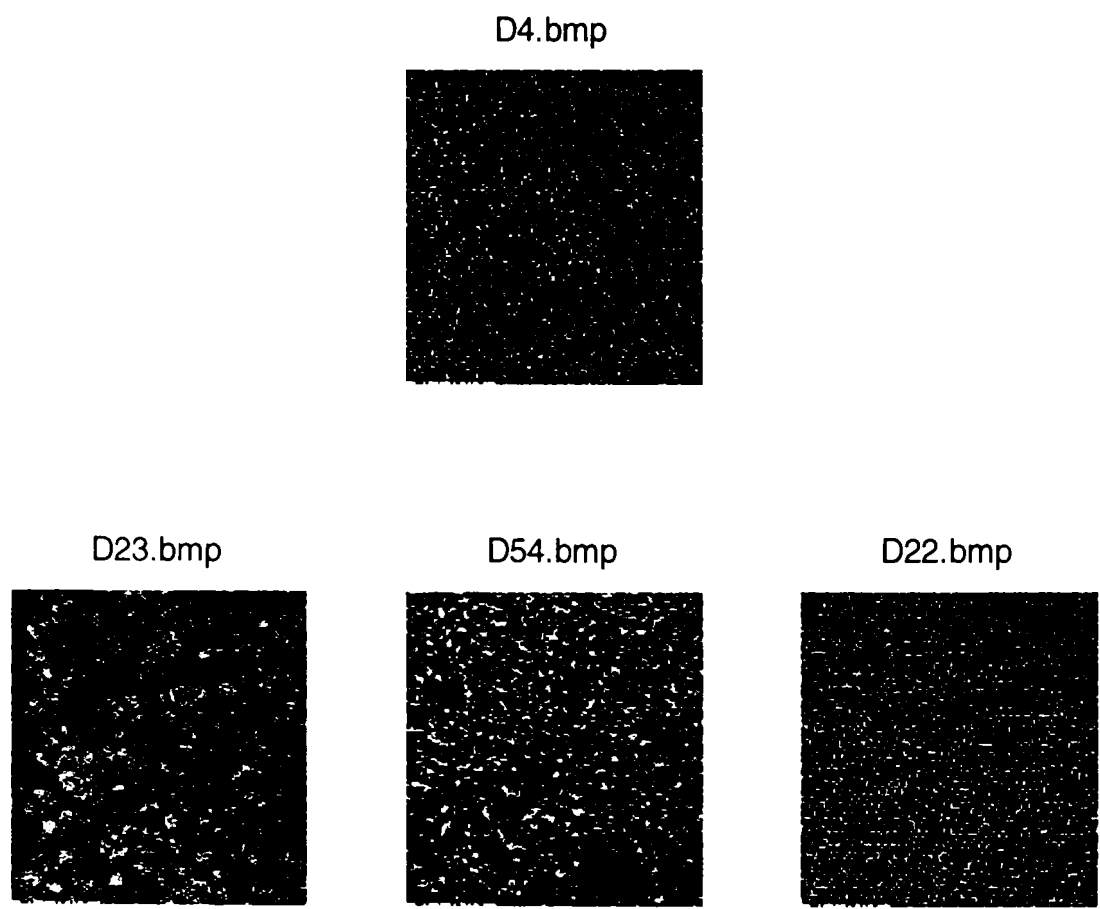


Figura 5-45. Rezultatele căutării pentru imaginea D4

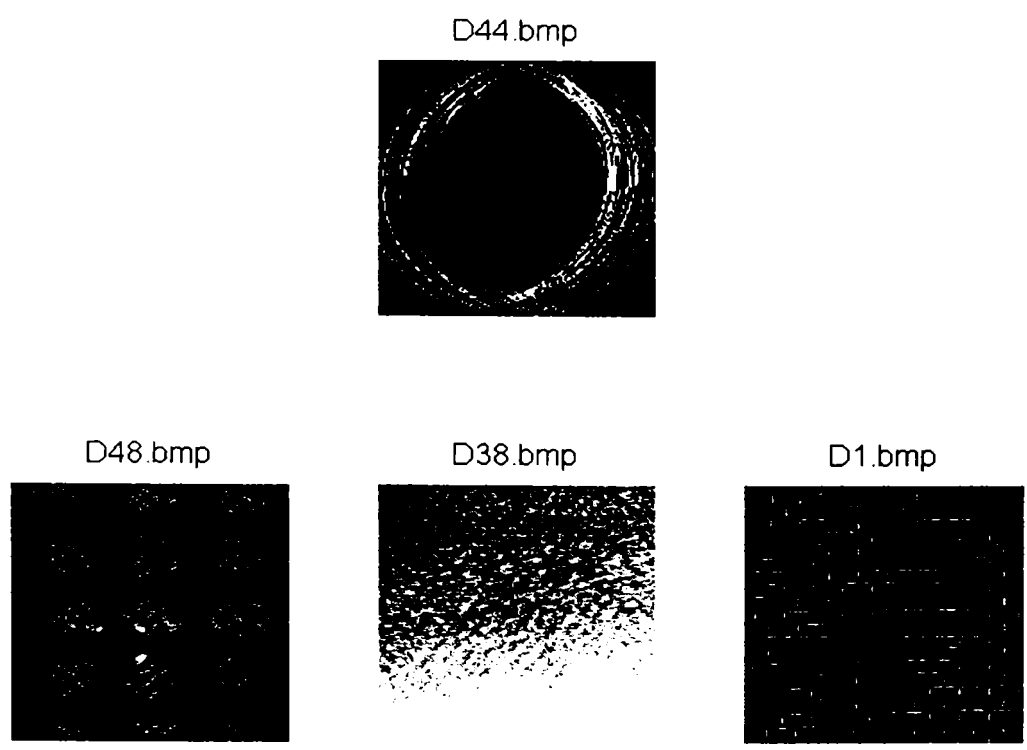


Figura 5-46. Rezultatele căutării pentru imaginea D44

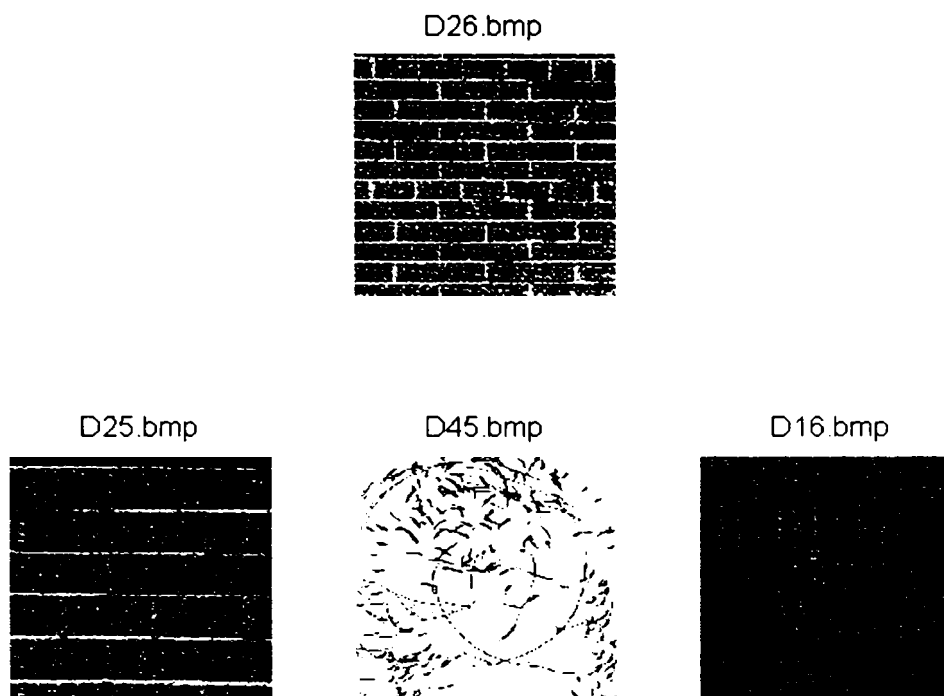


Figura 5-47. Rezultatele căutării pentru imaginea D26

Cele mai bune rezultate au fost obținute folosind normalizarea *mmmx* și o rețea de tip MLP, o dată ce procesul de antrenare a fost terminat (acesta având o durată considerabilă), timpul de procesare este foarte redus, iar rezultatele foarte bune.

O rețea de tip RBF are un timp mult mai scurt de antrenare, dar rezultatele căutării nu oferă o acuratețe bună [212].

Metodă de căutare ce folosește pentru discriminare o tehnică bazată pe praguri

Modul de lucru al metodei propuse este prezentat în continuare:

- Fiecare caracteristică a imaginii de test este comparată cu caracteristicile imaginilor din baza de imagini.
- Dacă caracteristica unei imagini din bază este cuprinsă între pragurile impuse, atunci acea imagine va primi un marker.
- Se va realiza o grupare a imaginilor după markerii asigurați pentru fiecare caracteristică.
- Se vor căuta apoi imaginile ce au aceeași markeri, în acest mod găsindu-se imaginile cu gradul cel mai mare de similaritate, cu cât sunt imagini cu mai mulți markeri, cu atât imaginea găsită se apropie mai mult de imaginea propusă.
- Am înlocuit noțiunea de imagini cu aceeași markeri cu cea de același număr de markeri, rezultând o creștere a numărului de imagini care au o anumită similaritate.

Am realizat comparații între diverse metode și am constatat că rezultatele metodei sunt foarte bune, viteza de căutare este foarte mare iar volumul și complexitatea calculelor este

redușă comparativ cu metodele ce folosesc matrice multidimensionale. Este foarte dificil de a lucra într-un spațiu cu multe dimensiuni, puterea de discriminare a unei caracteristici poate fi atenuată de un răspuns greșit al altei caracteristici implicate în acea metrică. Aceeași problemă există și dacă se folosesc histogramele imaginilor într-o formulă a similarității (5.167) prezentată în capitolele anterioare.

Histogramele sunt caracteristici globale ale imaginilor ce pot determina foarte bine gradul de similaritate dintre imagini, dar nu în toate cazurile. De exemplu, o imagine cu bare albe și negre verticale, are aceeași histogramă cu o imagine ce are acele-și bare dar orizontale. În acest caz diferența dintre cele două imagini fiind majoră din punct de vedere vizual

Am testat algoritmul prezentat pe o bază de date cu peste 300 de imagini, spectrul acestei baze fiind foarte larg: imagini reale din natură, animale, persoane, texturi, etc.

Figura 5-49 reprezintă rezultatele căutării dacă se folosește distanța Euclidiană dintre caracteristicile imaginii.

Figura 5-50 reprezintă rezultatele căutării atunci când se folosește distanța Euclidiană dintre histograme.

Pentru o mai rapidă căutare am folosit doar 5 valori reprezentative pentru histograme, rezultatele sunt redată în figura 5-51 (am folosit distanța Euclidiană ca metrică). Rezultatele nu sunt dintre cele mai bune, este rapidă ca metodă dar puterea de discriminare este scăzută.

Similaritatea dintre imagini definită în relația 5.167 are rezultatele căutării prezentate în figura 5-52.

Rezultatele metodei cu praguri sunt prezentate în figurile 5-53, 5-54 și 5-55, pentru următoarele valori ale acestora $t_1 = 1.5$, $t_2 = 1.1$, respectiv $t_3 = 1.05$.

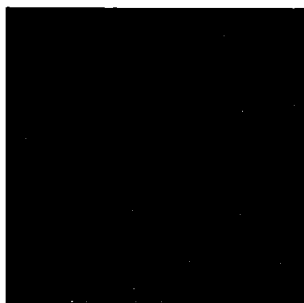


Figura 5-48. Imaginea de test

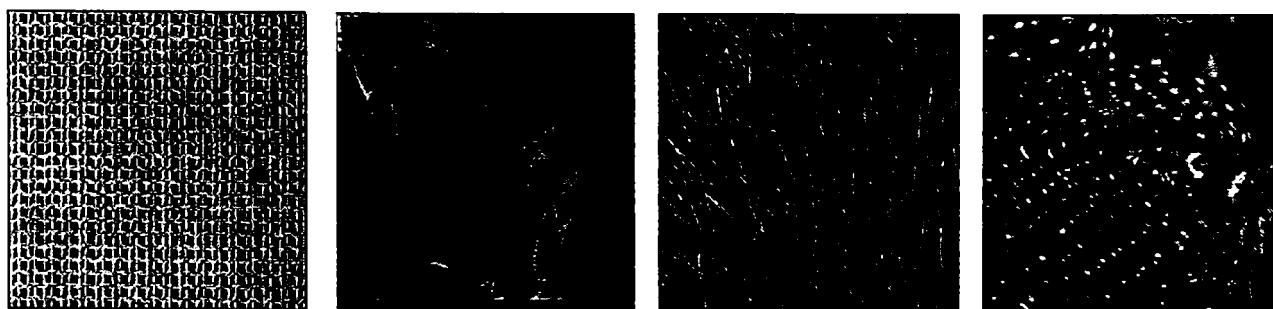


Figura 5-49. Rezultatele căutării folosind distanța Euclidiană dintre caracteristici

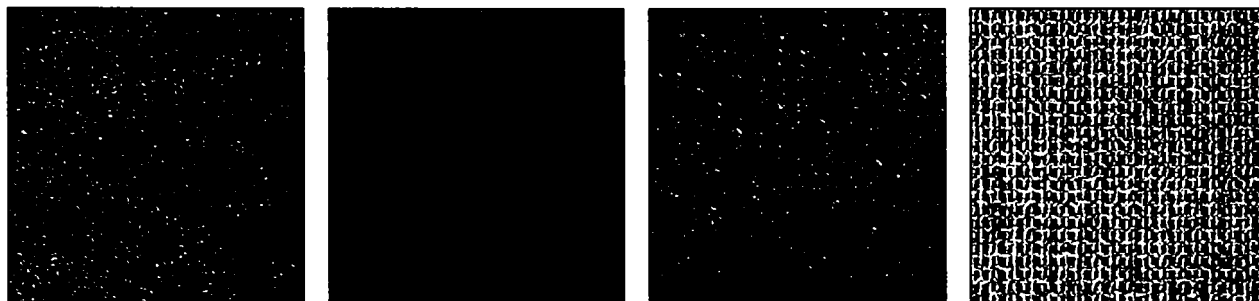


Figura 5-50 Rezultatele căutării folosind distanța Euclidiană dintre 5 momente ale histogramei

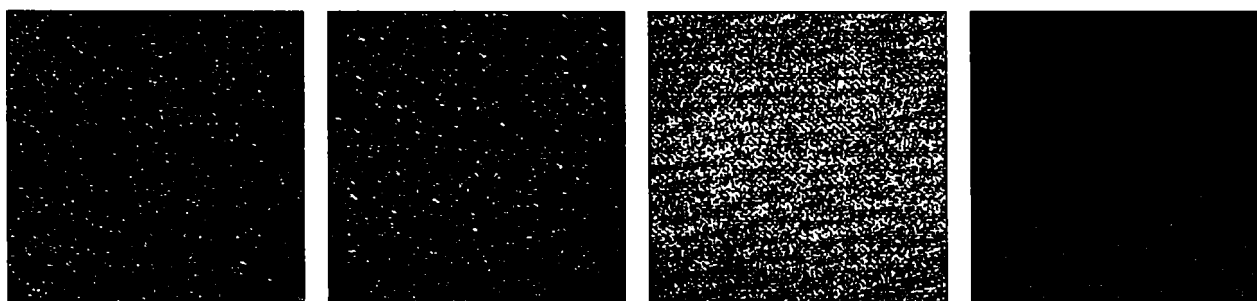


Figura 5-51. Rezultatele căutării folosind distanța Euclidiană dintre histograme

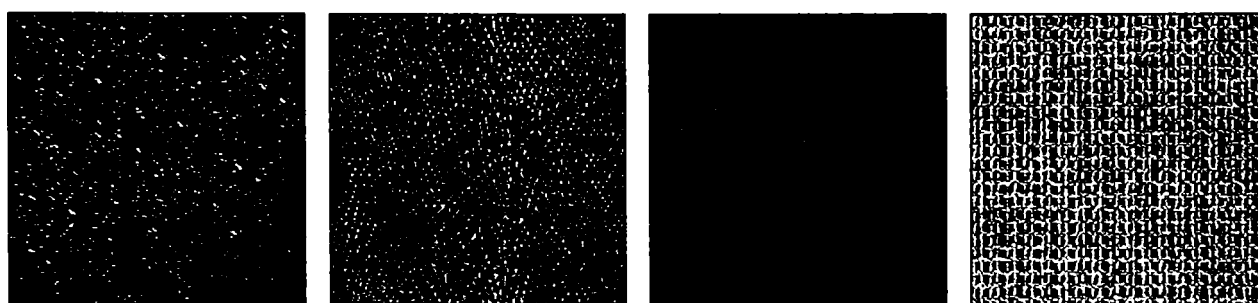


Figura 5-52. Rezultatele căutării folosind similaritatea dintre histograme

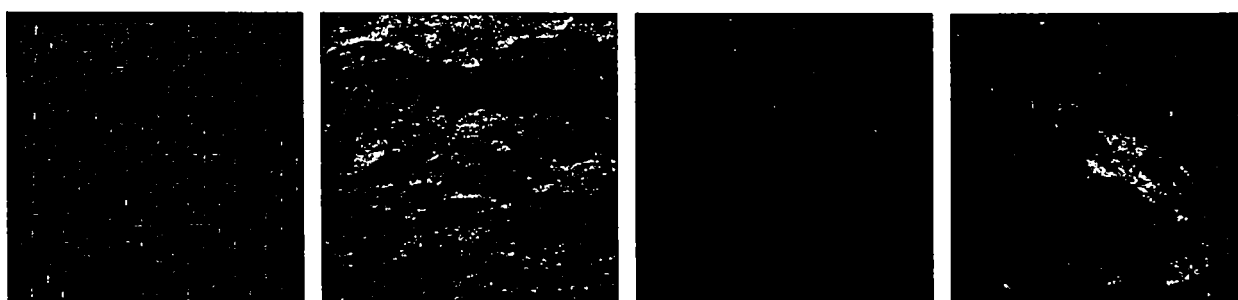


Figura 5-53. Rezultatele căutării pentru algoritmul propus ce folosește caracteristicile matricii de coocurență și o descompunere cu 25 de filtre Gabor pentru o valoare a pragului $t=1.5$

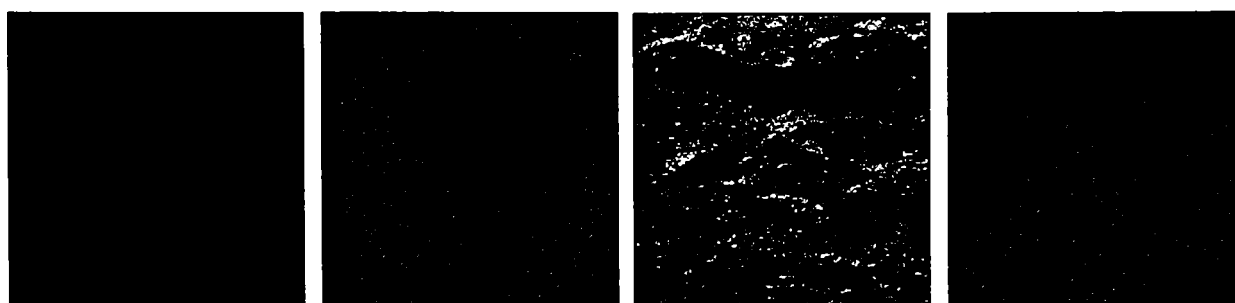


Figura 5-54. Rezultatele căutării pentru algoritmul propus ce folosește caracteristicile matricii de coocurență și o descompunere cu 25 de filtre Gabor pentru o valoare a pragului $t=1.1$

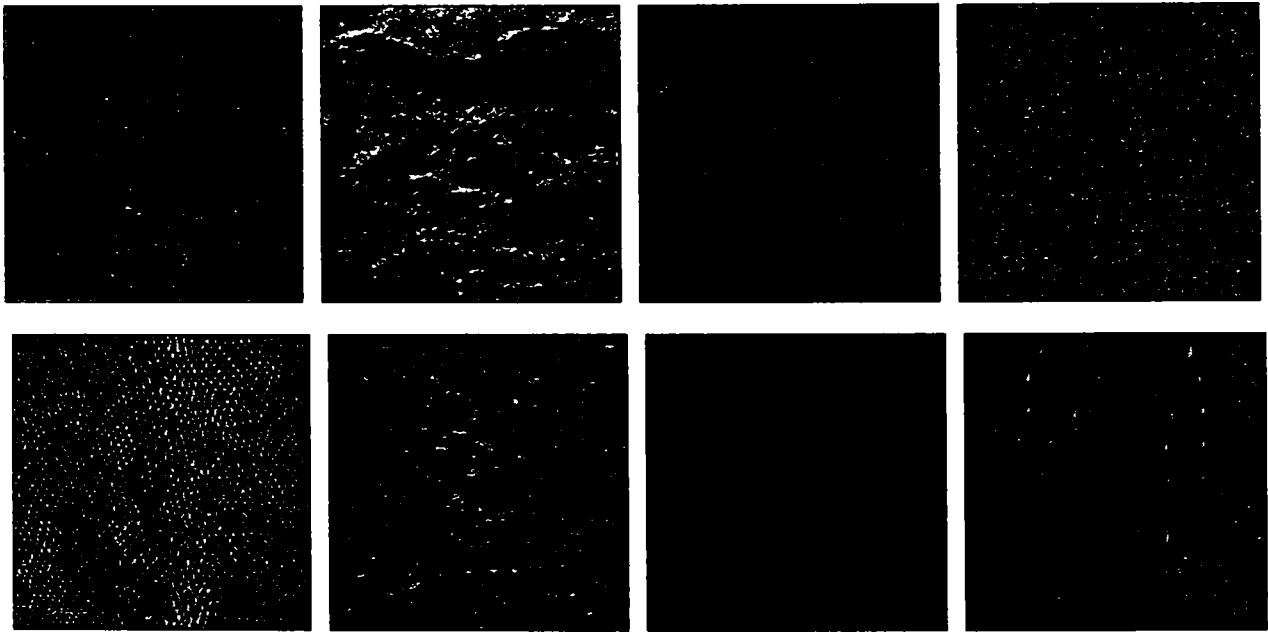


Figura 5-55. Rezultatele căutării pentru algoritmul propus ce folosește caracteristicile matricii de coocurență și o descompunere cu 25 de filtre Gabor pentru o valoare a pragului $t=1.05$

După mai multe experimente, am extras câteva concluzii ce au fost publicate la sesiuni de comunicări interne și internaționale [213], [214]. Modul de formare a vectorului general al caracteristicilor este prezentat în figura 5-56.

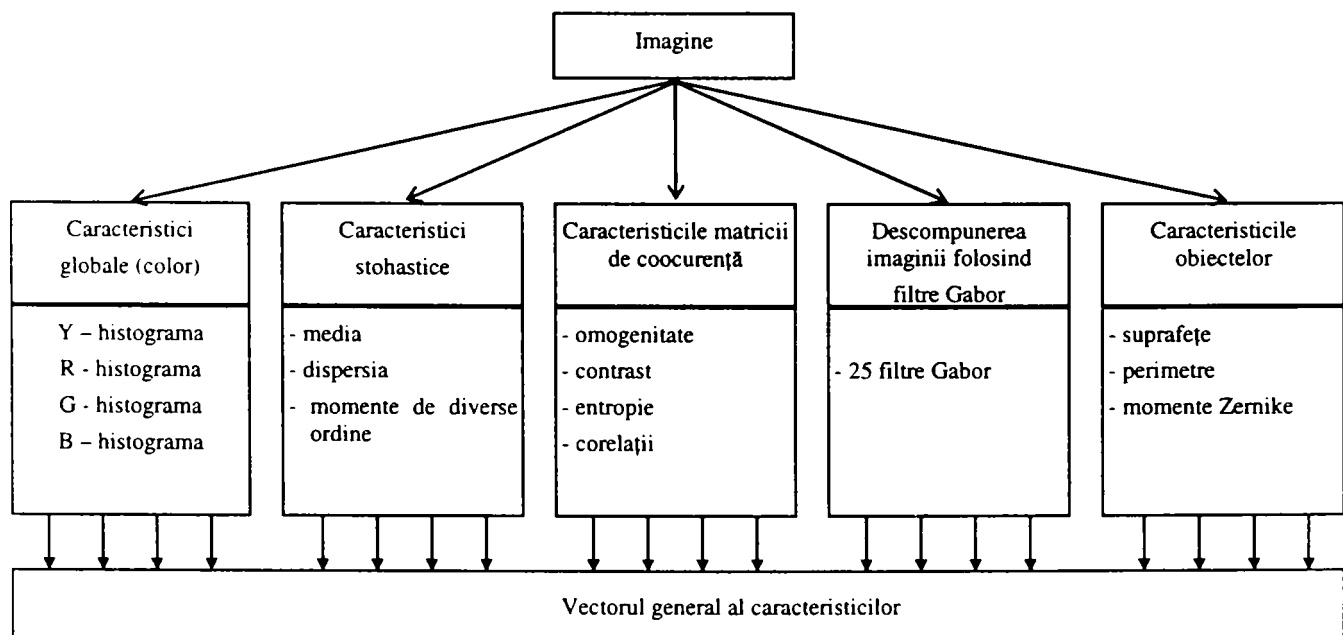


Figura 5-56. Formarea vectorului general al caracteristicilor

Vectorul caracteristicilor va fi format din următoarele categorii de caracteristici:

- Caracteristici globale – histograma semnalului de luminanță Y, histogramele componentelor primare color R, G, B;
- Caracteristici stohastice – media, dispersia, momente de diverse ordine;
- Caracteristicile matricii de coocurență – omogenitate, contrast, entropie, corelații, directivitate;
- Descompunerea imaginii cu ajutorul filtrelor Gabor – descompunere folosind 25 de filtre Gabor;
- Caracteristicile obiectelor – suprafețe, perimetre, contururi, momente Zernike.

Folosind tehnica cu prag am obținut similarități între imagini doar dacă se analizează doar dintr-un punct de vedere această similaritate. De exemplu, dacă se ignoră ca și caracteristică media și se folosesc caracteristici ale matricii de coocurență – legate de omogenitate, granularitate și contrast putem avea următoarele imagini similare (au anumite elemente descriptive comune):

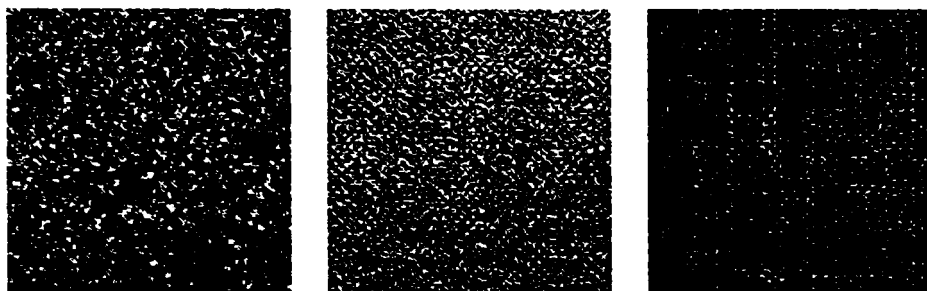


Figura 5-57. Imagini similare din punct de vedere al granularității, contrastului și omogenității ($t_3 = 1.05$)

Nu se pot utiliza numai anumite caracteristici, deoarece se obțin imagini fără similaritate din punct de vedere al sistemului vizual uman. Folosirea doar a caracteristicilor stohastice poate da de exemplu, că următoarele imagini sunt similare:

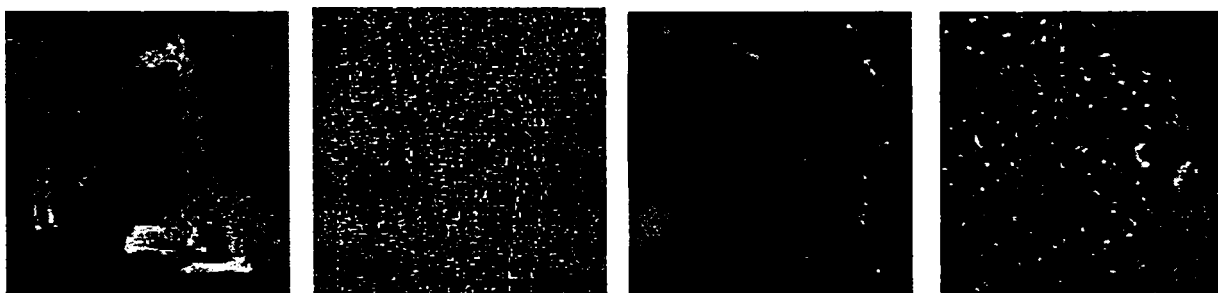


Figura 5-58. Imagini „similare” din punct de vedere al caracteristicilor stohastice



Figura 5-59. Imagini „similare” din punct de vedere al histogramelor

Din exemplele prezentate mai sus rezultă foarte clar că este nevoie de un număr mare de caracteristici care să descrie cu o anumită acuratețe similaritatea dintre imagini.

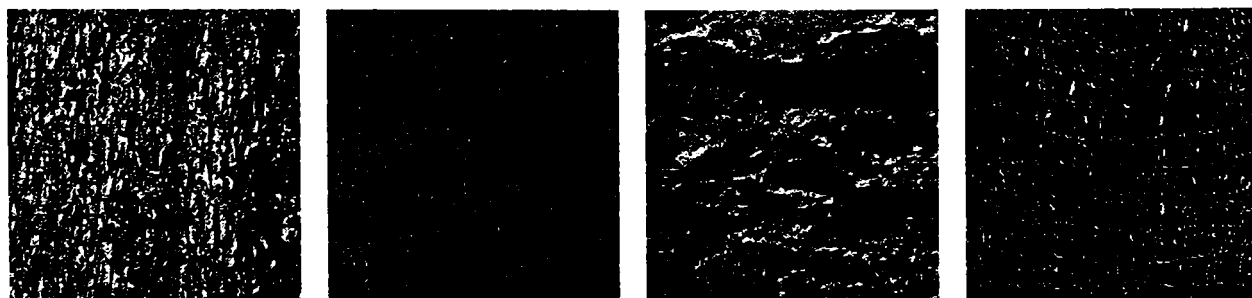


Figura 5-60. Imagini similare din perspectiva filtrelor Gabor și a caracteristicilor matricii de coocurență

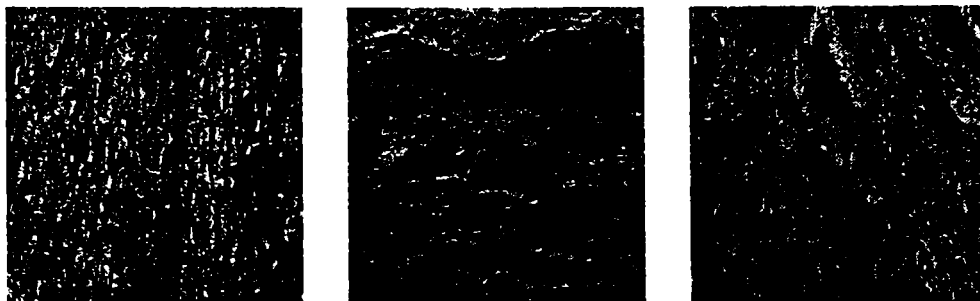


Figura 5-61. Imagini similare folosind toate caracteristicile si pragul $t_3 = 1.05$

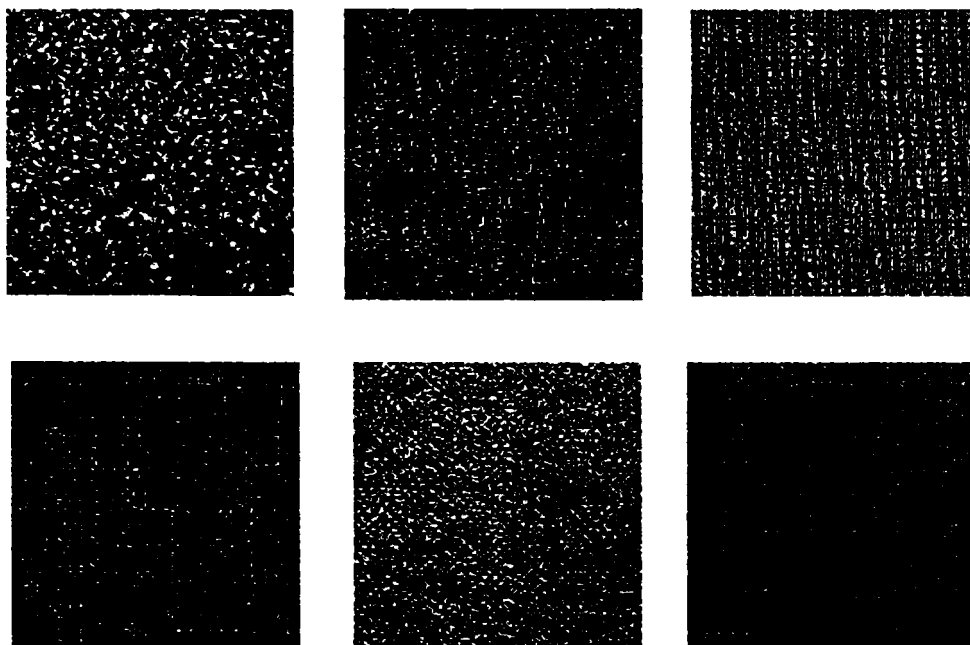


Figura 5-62. Imagini similare folosind toate caracteristicile si pragul $t_3 = 1.1$

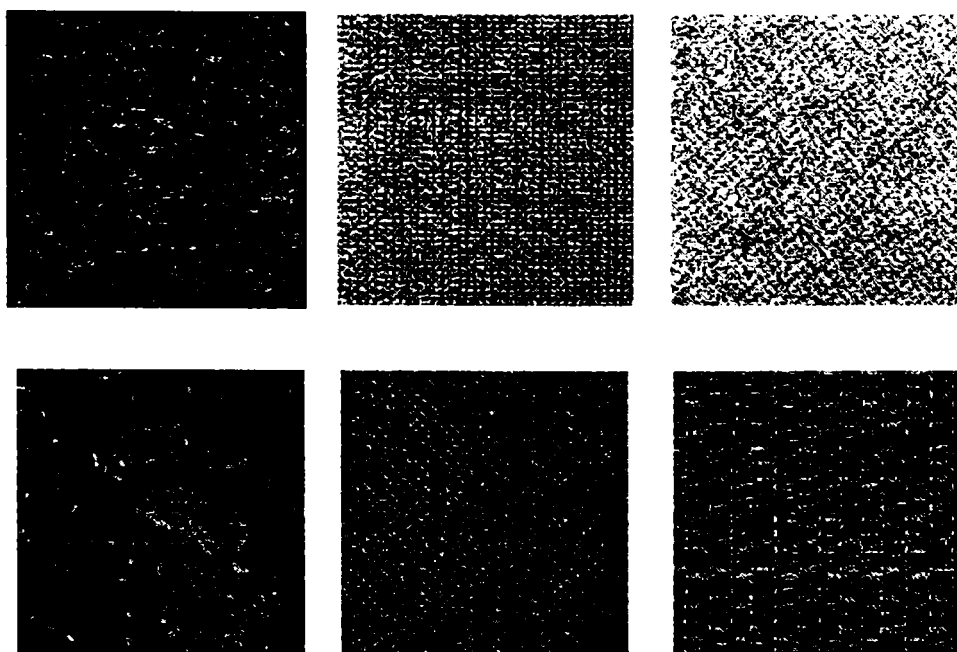


Figura 5-63. Imagini similare folosind toate caracteristicile si pragul $t_3 = 1.5$

Sisteme ce interacționează cu utilizatorul

Minka [215] presupune utilizarea interactivității cu utilizatorul, iar definirea unor clase de imagini se face pornind de la un grup de imagini, date de utilizator, pentru care sistemul extrage anumite categorii și încearcă extragerea de caracteristici globale pentru gruparea respectivă. Folosește rețele neuronale pentru clasificare și extragere de caracteristici globale.

Folosirea unei interactivități pentru reducerea numărului de calcule și supervizarea procesului de căutare a reprezentat una din direcțiile de dezvoltare a algoritmilor de indexare și căutare în bazele de date multimedia [216],[217],[218],[219],[220],[221].

Dezavantajele unui algoritm complex de indexare sunt legate de numărul mare de caracteristici extrase pentru fiecare imagine din baza de imagini, ceea ce presupune un timp mare de calcul. Pentru o bază de imagini ce nu a fost special concepută cu un algoritm de indexare bazat pe caracteristicile imaginilor este necesară extragerea caracteristicilor pentru toate imaginile parcurse. Aceasta presupune un număr foarte mare de calcule.

Interactivitatea permite utilizatorului să combine criteriile de căutare, ceea ce poate fi un dezavantaj în cazul în care utilizatorul este un începător în domeniu prelucrărilor de imagini.

Folosirea mai multor caracteristici necesită în multe cazuri ponderarea caracteristicilor cu diverse valori ce presupune la calculul distanței dintre imagini definirea metrice adaptive [222], [223]; modificabilă de către utilizator, ceea ce poate crea probleme unui utilizator fără experiență. Metrica adaptivă este impusă de diversitatea imaginilor, pentru utilizatori experimentați ea oferă un puternic instrument de discriminare.

Experimentările au demonstrat necesitatea folosirii unui număr mare de caracteristici, doar așa fiind posibilă o bună descriere a imaginii. Un număr mare de caracteristici presupune volum mare de calcule și implicit viteză redusă de căutare. De aceea mi-am îndreptat atenția spre a realiza un algoritm de căutare care să combine tehnica pragurilor cu o căutare bazată pe parcurgerea unui arbore care să permită o rapidă trecere de la o ramură la alta.

Pentru evitarea unui timp mare de căutare și calcule complexe, am ales un număr redus de caracteristici, 5 la număr, care să reprezinte principalele categorii de caracteristici.

Prima etapă este extragerea caracteristicilor pentru imaginea de test sau imaginea de interogare. Se va obține un vector cu următoarea structură:

$$F_{1,1}^{db} \dots F_{1,j}^{db}, F_{2,1}^{db} \dots F_{2,l}^{db}, F_{3,1}^{db} \dots F_{3,u}^{db}, F_{4,1}^{db} \dots F_{4,v}^{db}, F_{5,1}^{db} \dots F_{5,m}^{db}.$$

Unde în exprimarea $F_{l,j}^{qi}$ unde, l reprezintă categoria de caracteristici, j reprezintă ordinul caracteristicii în acea categorie, qi reprezintă imaginea de interogare iar db imagini din baza de date.

Cel de al doilea pas este de a verifica distanța pentru cele mai importante caracteristici dacă se înscriu sub un anumit prag θ_l .

Mixarea tehnicilor de căutare bazate pe arbori cu modul de interpretare al sistemului visual uman al caracteristicilor permite creșterea vitezei de căutare.

Cea mai importantă caracteristică, este media, fiind testată prima dată. Este o caracteristică globală bună, dar nu suficientă. Dacă distanța dintre o caracteristică a imaginii de interogare și imaginea din baza de imagini este mai mică decât un prag θ_l , atunci algoritmul va trece la evaluarea următoarei caracteristici. Dacă nu, se va abandona evaluarea pentru acea imagine și se va lua o nouă imagine din bază.

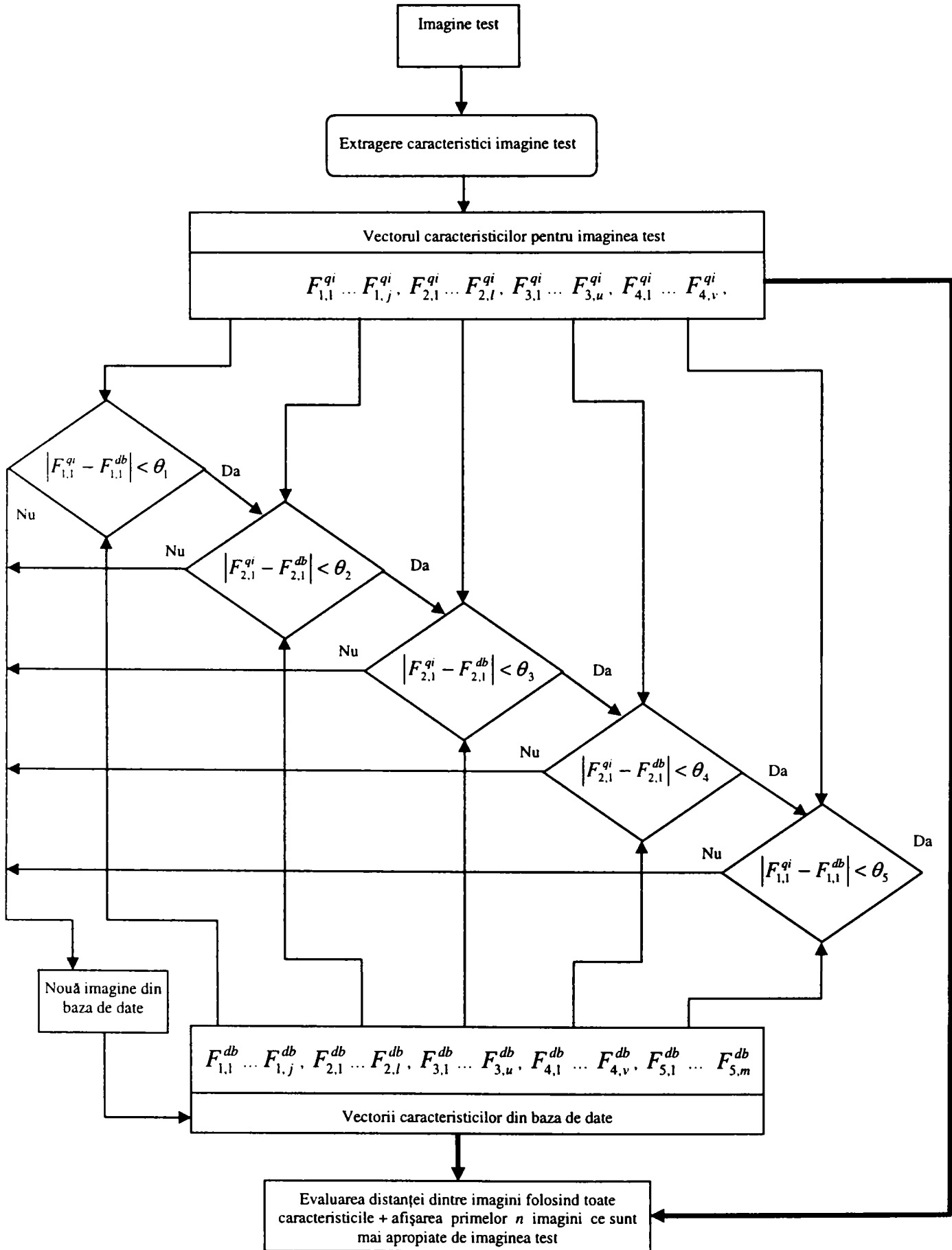


Figura 5-64. Algoritm de căutare bazat pe structura unui arbore

Cea de a doua caracteristică este histograma semnalului de luminanță. Histograma este o caracteristică globală pentru o imagine și în multe cazuri poate defini similaritatea dintre imagini. Sunt și situații particulare când nu rezultatele nu sunt foarte bune, de exemplu o imagine cu pătrate albe și negre și alta cu bare albe și negre în care există culorile alb/negru în aceeași proporție .

Din caracteristicile matricii de coocurență recomand utilizarea entropiei.

Pentru a caracteriza la modul global obiectele am ales ca măsură raportul arie/perimetru.

Pentru filtrele Gabor se pot alege doar câteva direcții mai importante: orizontală, verticală și diagonalele.

Distanțele dintre caracteristicile imaginii de interogare și cele ale unei imagini din bază pot fi evaluate folosind diverse valori pentru praguri.

După evaluarea celor mai importante caracteristici se va trece la finisarea procesului de căutare și integrarea tuturor caracteristicilor pentru evaluare. Partea finală poate fi una din metodele propuse la capitolele anterioare, foarte important pentru acest algoritm este implementarea arborescentă din partea de început ce poate reduce foarte mult calculele.

	<i>Caracteristica 1</i>	<i>Caracteristica 2</i>	<i>Caracteristica 3</i>	<i>Caracteristica 4</i>	<i>Caracteristica 5</i>
<i>Volu m calcule etapa I</i>	10 %	15 %	10 %	12 %	15 %
<i>Volu m calcule etapa II</i>	90 %	85 %	90 %	88 %	85 %
<i>Pondere a imaginilor din baza care au caracteristicile cuprinse între pragurile propuse</i>	30 %	20 %	20 %	15 %	25 %

Tabel 5-1 Evaluarea volumului de calcule pentru algoritmul propus

Procentul volumului calculelor introduse de prima etapă din volumul total dacă nu s-ar folosi acest algoritm este de 2,75%.

Procentul volumului calculelor introduse de prima etapă raportat la volumul calculelor pentru etapa a doua este de 12,5%.

Procentul volumului de calcule folosind algoritmul propus din volumul total dacă nu s-ar folosi acest algoritm (parcurgerea întregii baze de date) este de 22,05%.

Notă:

- Pentru calculul acestor rapoarte am folosit valorile cele mai dezavantajoase.
- Ponderile imaginilor ce au caracteristicile între anumite praguri poate varia foarte mult în funcție de imaginile conținute de baza de date.
- Valorile din tabel sunt rezultatul mai multor experimentări și reprezintă valori medii.
- Volumul calculelor depinde foarte mult de tipologia imaginii de interogare.
- De exemplu, dacă este o imagine care are media foarte diferită de valoarea medie a mediilor din baza de caracteristici, atunci timpul de căutare va fi foarte scurt.
- O imagine care are în baza de imagini, foarte multe imagini cu un grad mare de similaritate va determina creșterea timpului de căutare, respectiv a volumului de calcule.

Forma obiectelor

Culorile și texturile sunt proprietăți utile care pot defini atât imaginile cât și obiectele prezente în acestea. Dar există proprietăți care se pot adresa în special obiectelor individuale.

Pentru aplicații de indexare și căutare în baze de date este foarte important să cunoaștem proprietățile geometrice ale obiectelor, forma conturilor, dimensiunile și localizarea spațială în imagine. În cazul în care se folosește o căutare generală bazată pe obiectele din imagine cele mai importante proprietăți sunt legate de forma obiectului. Este foarte dificil să definim la modul general toate formele posibile, mai ales că există și un grad mare de subiectivism în această interpretare.

5.4.5. Concluzii:

Experimentările folosind algoritmi propuși au condus spre următoarele concluzii:

- Caracteristicile globale – histograme, media, dispersia imaginilor – nu asigură o clasificare decât în unele cazuri, pot exista foarte multe cazuri particulare ce nu sunt detectate de acestea.
- Matricea de coocurență conține o bogată informație despre relațiile spațiale dintre pixeli, dar necesită un volum foarte mare de calcul. Se pot folosi însă caracteristici derivate din matricea de coocurență, cum ar fi: entropia, contrastul, omogenitatea, directivitatea, granularitatea. Aceste caracteristici se apropie foarte mult de o descriere folosită în limbajul comun, fapt ce ajută utilizatori mai puțin experimentați în descrierea imaginii de interogare atunci când se realizează o operație de căutare într-o bază de imagini.
- Descompunerea imaginilor cu ajutorul filtrelor Gabor asigură o interpretare similară cu cea a sistemului vizual uman.
- Nu este necesară folosirea unui număr mare de filtre, în jur de 25 sunt suficiente. Un număr prea mare de filtre va îngreuna foarte mult calculele, volumul acestora fiind foarte mare, iar algoritmul poate deveni prea lent.
- Un pas foarte important este alegerea unei corecte decompoziții cu filtre Gabor care să acopere întregul spațiu.
- O toleranță prea mare acceptată de tehnica pragurilor nu va duce la o grupare adecvată, reprezentativă.
- Se poate accepta un sistem interactiv ce permite reducerea iterativă a valorii pragului în funcție de rezultatele dorite. Se vor obține imagini cu un grad de similaritate din ce în ce mai mare.
- O valoare prea mică a pragului nu reprezintă o soluție, deoarece imaginile pot avea anumite distorsiuni și zgomote iar imaginile să fie totuși similare. Reducerea pragului va determina și reducerea numărului de caracteristici cărora li se asignează un marker, iar un număr prea mic de caracteristici nu vor putea descrie cu o acuratețe prea mare o imagine.
- Este bine ca în procesul de căutare să se impună un număr minim de markeri pentru a defini o similaritate. Alegerea acestui număr depinde de caracteristicile folosite în descrierea imaginilor și de conținutul imaginilor din baza de imagini.
- Interactivitatea sistemului de interogare este un avantaj în procesul de căutare, astfel redefinirea parametrilor folosiți după o primă căutare permite creșterea acurateței de găsire a imaginilor în baza de imagini.
- Din imaginile inițial detectate, se pot alege un număr mai mic de imagini, prin intermediul cărora se vor evalua caracteristicile unei noi imagini de interogare, mai apropiată de cerințele utilizatorului. Folosindu-se un număr mai mare de imagini de model se pot determina mai exact caracteristicile imaginii căutate.
- Dintre clasificatorii neuronali testați care oferă o bună acuratețe în procesul de căutare sunt MLP și RBF.

B. Indexarea, interogarea și căutarea secvențelor video în baze de date multimedia

5.5. Modelul materialelor video

5.5.1. Proprietățile secvențelor video

Caracteristica principală a secvențelor video constă în redundanța existentă între cadre succesive. În general, între cadre adiacente modificarea informației nu este deosebit de mare. Dacă există o schimbare majoră între două cadre adiacente, acest lucru poate fi exploatat în procesul de căutare ca fiind momentul când are loc o schimbare a secvenței filmate (a locului unde s-a realizat filmarea). Problemele de detecție a momentului schimbării locației de filmare apar atunci când trecerea de la o secvență filmată la alta se realizează prin intermediul efectelor video. Efectul care maschează cel mai bine această trecere este efectul de mixare a celor două secvențe video. Pentru evitarea problemelor de detecție, se poate mări modelul de analiză, de la cadrele adiacente la o perioadă mai mare de timp (adică la un număr mai mare de cadre), comparația realizându-se atât între cadrele adiacente, cât și între cadre aflate la distanțe mai mari.

5.5.2. Nivelele semantice ce alcătuiesc modelul materialelor video

Similar cu textul (unde avem rânduri, paragrafe, capitole, etc.), o secvență video are o structură semantică ce corespunde unui model de material video.

Autorul a studiat modul de segmentare al unui material video în elementele constituente, rezultatul cercetărilor sale în acest domeniu fiind prezentate detaliat în [224]. Algoritmul propus de autor permite constituirea unui model ierarhic de căutare. Sintetic se pot defini elementele constituente – printr-o abordare semantică temporală astfel:

- Cadrul este reprezentat de o filmare propriu-zisă de durată relativ scurtă (2-5 secunde) și este caracterizat de faptul că nu există schimbări majore între elementele ce formează imaginea.
- Scena este formată dintr-o serie de cadre ce descriu aceleași elemente, între cadrele constituente există elemente de legătură ce interacționează foarte intens.
- Planul de filmare este reprezentat din momentul în care se pornește camera și se termină atunci când se oprește aceasta. În funcție de modelul emisiunii, un plan poate avea o dimensiune temporală variabilă.
- Secvența este reprezentată de un grup de scene și plane ce nu sunt în mod necesar legate între ele.
- Emisiunea este modelul complex ce conține o combinație cadre, scene, plane de filmare și secvențe. Pentru emisiuni se pot defini modele temporale ce se folosesc la căutare, dar și modele spațiale date de elementele stabile din imagine ce persistă o perioadă mai mare de timp.

Se remarcă în multe situații eficacitatea descrierii semantice a unui material video. Sunt situații în care indexarea materialului trebuie făcută la nivel de cadru, iar în interiorul cadrului să

se folosească descrierea unor obiecte. În funcție de imaginile existente, în realitate, între obiecte pot exista anumite reguli de apariție (de exemplu, o căprioară apare în general într-o imagine în care există și iarbă). Pentru definirea interacțiunilor dintre obiectele din imagini se poate utiliza cu succes și teoria grafurilor. Utilizarea grafurilor permite găsirea unor imagini pornind de la interacțiunile ce există între obiectul căutat și obiectele ce apar în general legate de acesta. Similaritatea dintre imagini este definită prin interacțiunile dintre obiecte [225].

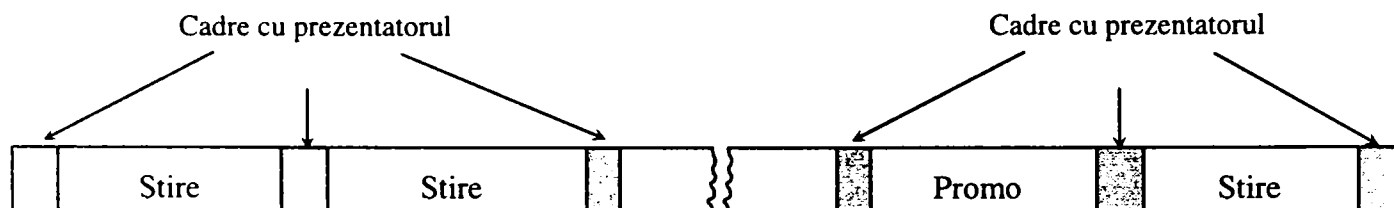


Figura 5-65 Descrierea semantică, din punct de vedere temporal, pentru un jurnal de televiziune

O emisiune poate fi caracterizată și din punct de vedere al organizării în interiorul unui cadru. Astfel, ca exemplu se poate considera jurnalul de știri. Jurnalul de știri începe cu un generic ce se menține neschimbat o perioadă îndelungată de timp (luni, uneori chiar și ani). Fiecare știre începe cu o scurtă prezentare ce se face din studio de către prezentator, după care se trece la difuzarea efectivă a știrii. Poziția, lumina și coloristica secvențelor în care apare prezentatorul este constantă pe întreaga durată a jurnalului de știri. În interiorul jurnalului există momente promoționale sau generice pentru subdiviziunile jurnalului de știri. Aceste momente delimitează noi momente temporale ce pot fi utilizate cu succes la definirea temporală a modelului semantic al emisiunii.

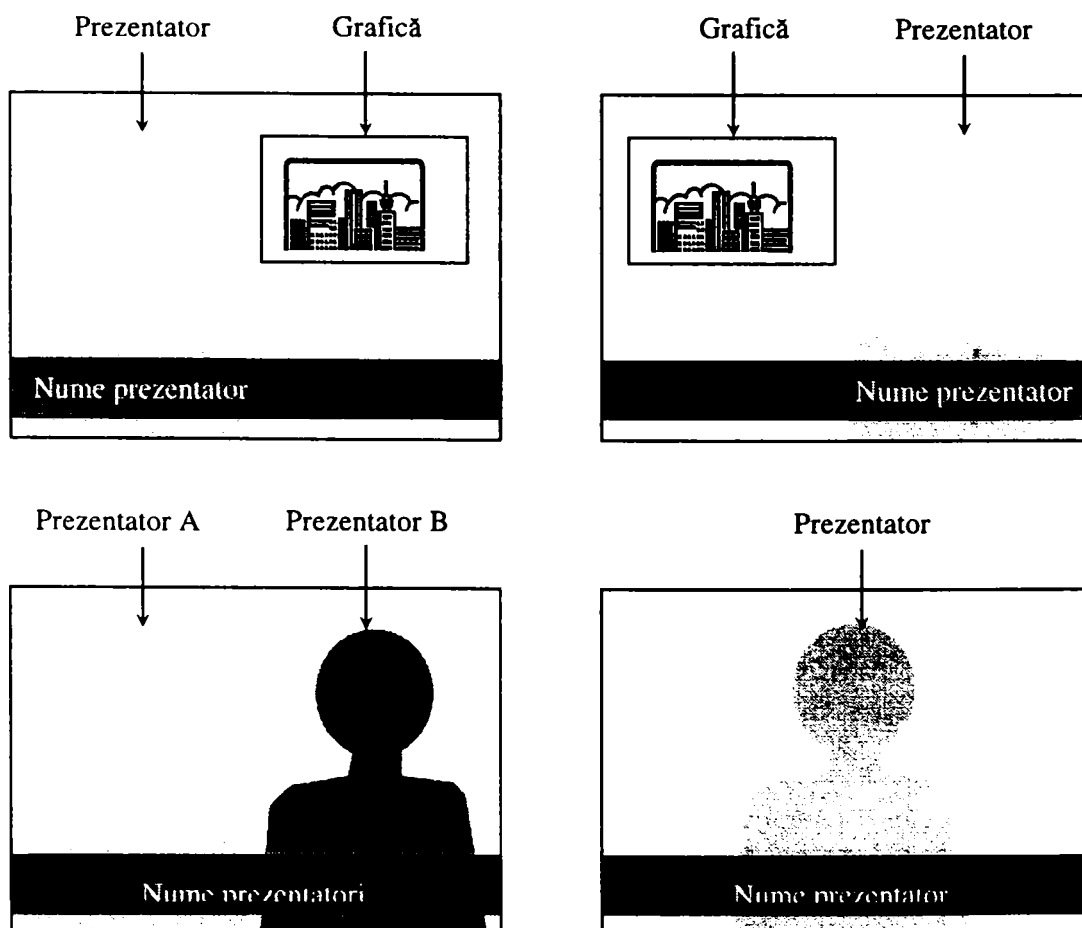


Figura 5-66 Descrierea semantică, din punct de vedere al organizării spațiale, pentru un jurnal de televiziune

5.6. Tehnici de indexare a secvențelor video

Tzanetakis G., Luc J. [226] prezintă o sinteză a sistemelor de căutare în bazele de date ce conțin ca obiecte secvențe video având ca finalitate o aplicație multimedia ce permite realizarea unei emisuni TV pornind de la secvențele video stocate într-o bază de date. Pentru indexarea secvențelor video se pot utiliza mai multe metode. Autorul a testat un număr de caracteristici ce sunt propuse în literatură, în care se folosesc caracteristici globale, bazate pe histograma imaginilor color precum și structuri semantice definite pentru model [227], [228], [229], [230].

Pentru segmentarea automată se pot utiliza cinci categorii principale de metode:

- metode ce lucrează la nivel de pixel;
- metode statistice;
- metode bazate pe diverse transformări;
- metode ce folosesc caracteristicile imaginilor;
- metode bazate pe histograme.

Pentru exemplul de emisiune de știri prezentată mai sus am folosit mai multe metode de detectare a prezentatorului:

- O primă metodă folosea o imagine tipică cu prezentatorul. Această imagine conține suficiente elemente de fundal care să caracterizeze emisiunea și să nu depindă de îmbrăcămintea prezentatorului.
- Pentru creșterea vitezei de căutare și reducerea volumului de calcul autorul folosit doar porțiuni din imagine. Acest lucru a fost posibil datorită persistenței obiectelor specifice în aceleași locații în imagine.
- În literatură sunt prezentate metode care folosesc caracteristici invariante pentru detectarea unui crainic [231]. Personal, consider mai importantă detectarea elementelor de fundal ce rămân neschimbate perioade mari de timp și nu a prezentatorilor care își pot schimba vestimentația, cofura sau machiajul zilnic, sau se pot schimba chiar prezentatorii.
- Pentru cazuri particulare de emisiuni, am folosit histograma componentei albastre de culoare ca element central de căutare a secvențelor cu prezentatorul (emisiunea de știri a postului de televiziune ProTV).
- Ca variantă adaptată la o anumită emisiune de știri, am considerat o imagine cu prezentatorul cunoscută apriori. Variațiile histogramelor imaginilor ce conțin prezentatorul în comparație cu histograma imaginii model s-au înscris în jurul valorii medii de 5 % cu câteva excepții în care procentul erorii a ajuns la 7%.
- Imaginile, definite apriori în procesul de căutare, pot fi utilizate pe perioada mai multor emisiuni și nu trebuie definite manual pentru fiecare emisiune în parte.
- Rezultate foarte bune, din punct de vedere al vitezei de lucru, au fost obținute de către autor prin detectarea benzii unde este trecut numele prezentatorului. Această bandă este în mod constant poziționată în aceeași locație. Problema este că în unele emisiuni aceasta poate fi „uitată”, adică nu se folosește.
- Criteriul divergenței Kullbach-Liebler [232] se folosește de vectorii caracteristici $x_{vi} \in \mathbf{R}^n$ generați prin intermediul histogramelor color și se definește astfel:

$$G_v(i) = - \sum_{k=1}^n x_{vi}^k \log \frac{x_{v(i-)}^k}{x_{vi}^k}$$

$$F_{com} = |\overline{G}_v(i) - G_v(i)| - \theta \geq 0$$

unde, $\overline{G}_v(i)$ reprezintă media divergenței.

Deoarece în emisiuni se pot folosi diverse efecte video ce fac ca tranzițiile dintre două filmări realizate în locații diferite să fie foarte line, astfel încât folosirea funcției de eroare între cadrele succesive să nu poată fi folosită, autorul propune ca pe lângă funcția de eroare calculată între cadre adiacente să se realizeze și calculul funcției de eroare între cadre ce se află la o anumită distanță (8-10 cadre). Folosind un model temporal se poate realiza o predicție de apariție a tranzițiilor. Fiecare emisiune poate avea propriul „ritm”. O analiză mai atentă a ritmului de schimbare a scenelor l-a condus pe autor la concluzia că nu este suficientă pentru o emisiune de știri folosirea doar a unui model simplu, ci este nevoie de adaptarea acestuia în funcție de fiecare submaterial video ce formează emisiunea. Există o primă divizare a emisiunii de știri realizată de către genericele ce definesc începutul fiecărei subdiviziuni. Aceste generice pot fi folosite adesea ca modele cunoscute apriori. Ritmul general de apariție a știrilor (acestea au o durată medie cuprinsă în intervalul 1-3 minute) poate fi detectat prin apariția prezentatorului între două știri succesive. Ritmul fiecărei știri depinde de conținutul materialului video. Ritmul unei știri despre un eveniment dintr-o discotecă diferă de ritmul unei știri ce redă un vernisaj într-o expoziție de picturi.

Pentru a realiza o mai bună descriere și grupare, se pornește de la considerentul că o anumită secvență este formată din mai multe imagini filmate de mai multe camere în aceeași locație din unghiuri diferite sau în mai multe locații. Deci există o anumită probabilitate ca la o anumită distanță temporală să existe asemănări între cadre. De aceea, după o primă indentificare a cadrelor prin metoda erorii între cadre succesive, susținută de verificarea erorii realizată la o distanță de 8-10 cadre, se poate trece la compararea cadrelor aflate la jumătatea intervalelor definite cu ajutorul erorilor temporale. Utilizarea unei astfel de metode permite o mai bună organizare a imaginilor în funcție de locațiile unde s-a realizat filmarea. Sau un alt caz concret poate fi dat de o emisiune ce are mai mulți invitați. Dacă se dorește o selecție automată a imaginilor ce conțin doar un invitat, este necesară o comparație între mijloacele intervalelor determinate prin metode ce folosesc eroarea dintre cadre succesive. Utilitatea unei baze de date care să conțină o astfel de structurare a informațiilor este deosebit de mare în cazul posturilor de televiziune.

5.7. Tehnici de căutare a secvențelor video

Metodele ce folosesc histogramele pentru comparație între cadrele adiacente sau la mică distanță se dovedesc suficient de rapide, iar calitatea discriminării suficient de bună. O dată extrase cadrele cheie (cadrele ce sunt determinate de apariția unor schimbări deosebite între cadrele adiacente), căutarea în secvențele video poate fi echivalată cu o căutare într-o bază de imagini formată din cadre cheie, iar dacă se determină o eroare mai mică decât un anumit prag, se poate trece la o căutare cadru cu cadru în secvența determinată de către acel cadru cheie. Pentru aceste căutări se pot folosi toate metodele prezentate în capitolele anterioare.

Se pot folosi pentru indexare metode ce extrag fiecare schimbare de cadru sau identificarea doar a unor elemente cheie, cadre cheie în interiorul unor emisiuni.

Se pot propune sisteme automate de indexare pentru cazuri concrete, practice din domeniul video conferințelor și a programelor de știri, în care clasificarea se face folosind metode bazate pe un model sau pe un scop general [233].

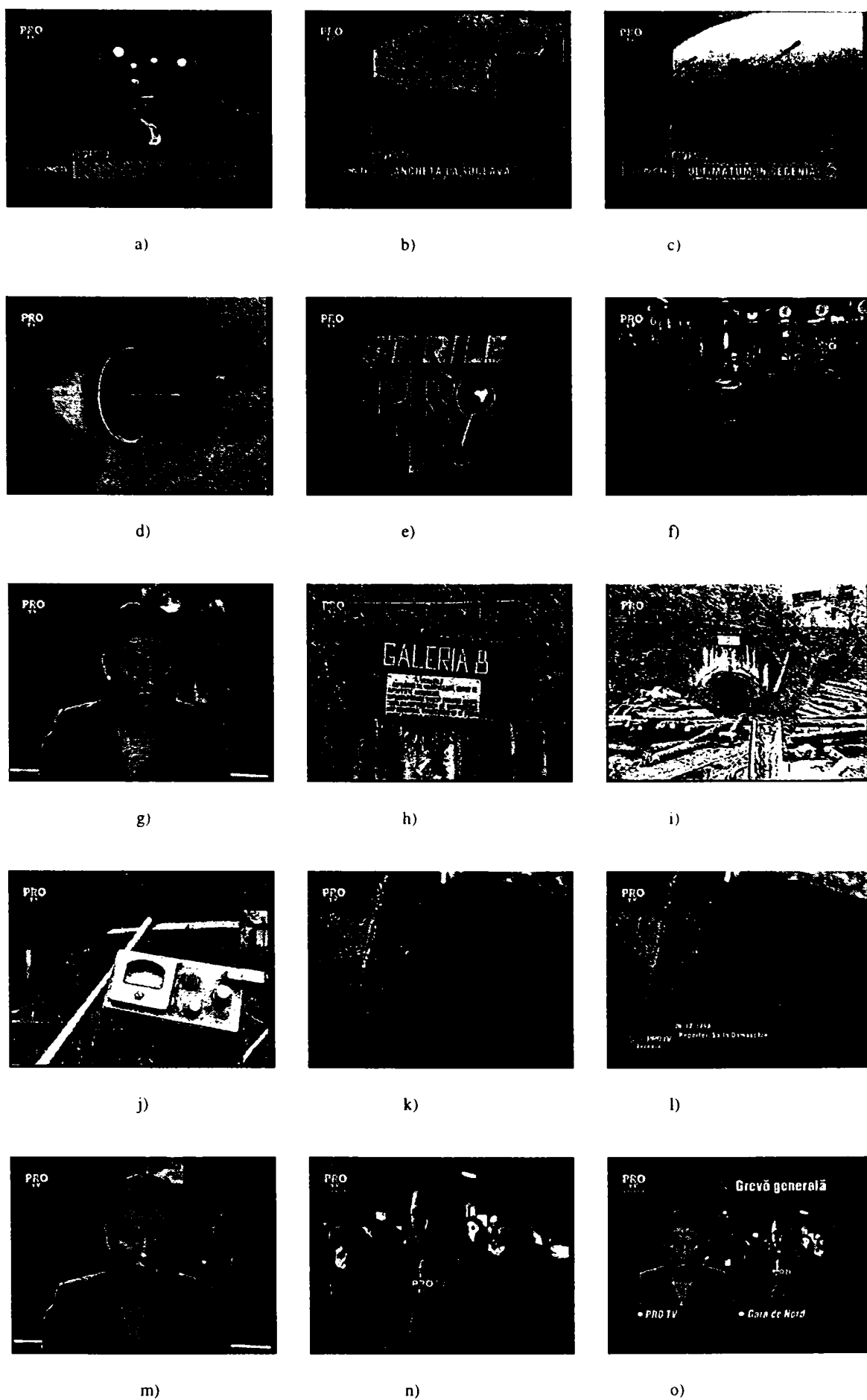


Figura 5-67 Exemple de cadre ce au fost detectate folosind un model de descriere spa io-temporal pentru jurnalul de  tiri de televiziune

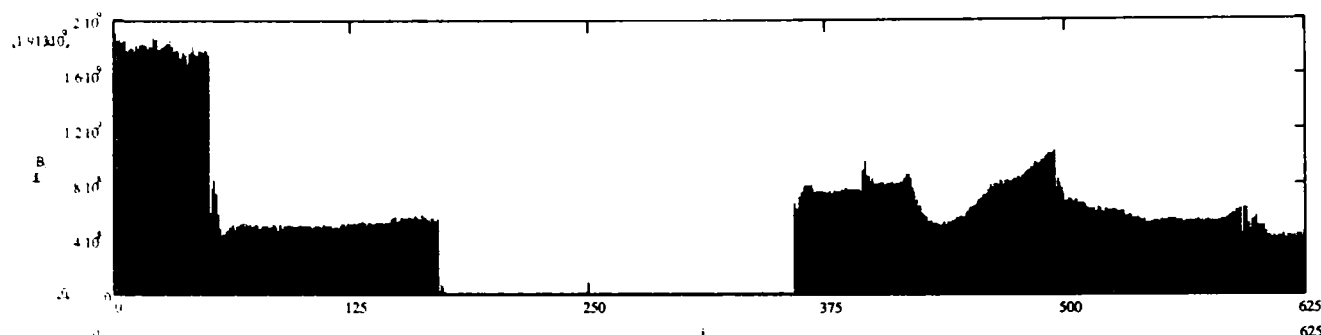


Figura 5-68 Funcția de eroare rezultată prin folosirea unui model cunoscut *apriori* pentru prezentator

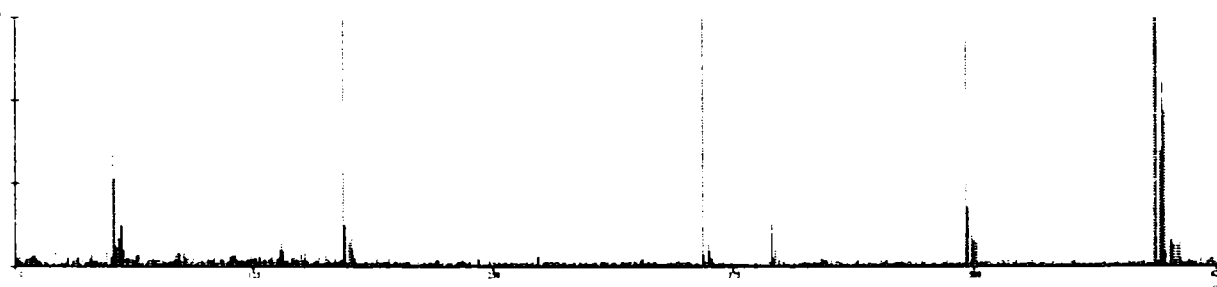


Figura 5-69 Funcția de eroare rezultată ce nu ține cont de modele cunoscute *apriori*

Din figurile de mai sus rezultă clar că folosirea unui model *apriori* poate determina cu precizie foarte bună momentul apariției imaginilor model. Pentru definirea modelului temporal al emisiunii, autorul recomandă această metodă.

Detecția schimbărilor planurilor de filmare, folosind cadre adiacente se dovedește eficientă, deși mai există mici vârfuri ce pot indica tranziții false precum și pierderea unor schimbări dacă trecerea se realizează folosind efecte video de durată mare. Comparațiile între cadre aflate la distanțe mai mari poate elimina imprecizia de detecție datorată efectelor video.

Accesarea bazelor de date prin Internet, în special a celor care conțin secvențe video, presupune o adaptare a modului de stocare a informației video, în special a modului de compresie a acesteia. Astfel, la transmiterea prin Internet trebuie să existe pierderi cât mai reduse din cauza blocajelor, pierderilor de pachete de date sau a alterării datelor transmise [234]. Se regăsesc mai multe metode de optimizare a acestor probleme, împărțite în două mari categorii: optimizarea rețelei de transmisie și metode de compresie ce țin cont de caracteristicile modului de transmisie. Soluția cea mai des utilizată este transmiterea semnalului video pe mai multe niveluri (straturi) ce redefinesc iterativ calitatea semnalului. Dacă în rețea există o aglomerare, se va transmite doar stratul de bază, cel de rezoluție scăzută, pentru asigurarea unui minim de imagine, la reducerea traficului în rețea se vor mări numărul de niveluri transmise [235]. Dacă se dorește căutarea într-o bază de date video prin intermediul unei conexiuni Internet, autorul recomandă folosirea unei indexări bazată pe cadre cheie. Utilizatorul poate lansa o căutare în baza de date video, pentru aceasta se pot folosi metode de comparație între imaginea de interogare și cadrele cheie obținute în mod automat la introducerea secvenței video în baza de date. Cadrele cheie rezultate în urma căutării, ca fiind cele mai apropiate de imaginea căutată, sunt mai ușor de transmis prin Internet, nu necesită un debit deosebit și permit o mai ușoară parcurgere din partea utilizatorului. Problemele legate de accesarea bazelor de date video prin Internet sunt probleme sensibile ce necesită cercetări susținute în domeniile bazelor de date, a transmisiei de date și a compresiei adaptive la modul de transmisie [236], [237].

5.8. Utilizarea tehnicilor multimedia în televiziune

5.8.1. Tipuri de informații utilizate într-un studio de televiziune

Universul în care trăim este puternic impregnat de cultura video și TV. Tehnologia actuală permite prelucrarea unui trafic foarte mare de informații. Interpretarea acestor informații și deciziile ce urmează a fi stabilite pe baza lor sunt efectuate în general de către factorul uman, dar pot fi realizate și din automatizări.

Studiourile de televiziune prelucrează cele mai diverse tipuri de informații, acestea provin sub diverse formate, ceea ce implică o dificilă gestionare a acestora. Ca surse de informații putem avea: telefonul, faxul, în viitor videotelefonul, emailul, informația video ce provine din materiale filmate sau cea generată pe calculatoare.

Diversitatea acestor date, trebuie integrată în materiale ce vor fi difuzate spre telespectatori. Corespondențele telefonice dau un plus de autenticitate materialelor, dar nu pot oferi informații video, de aceea de cele mai multe ori se folosesc ca suport de imagine poze, imagini mai vechi, sau imagini stocate pe CD-uri. Unele știri sunt primite de la agenții de presă cum ar fi: RomPress, MediaFax, prin fax sau telex sau chiar prin Internet, etc. Aceste date sunt recepționate în general cu ajutorul unor echipamente dedicate, dar există și posibilitatea recepționării prin intermediul unui calculator. Știrile primite pe această cale sunt citite și selecționate. Cele alese, vor fi transmise telespectatorilor prin intermediul crainicilor sau prin afișarea lor sub formă de text pe ecran. Există și posibilitatea de difuzare prin intermediul teletextului a unor informații diverse înspre telespectatori. Au fost experimentate cu succes sisteme de "data casting" în scopuri didactice, dedicate învățământului la distanță. Răspândirea videotelefonului și a videoconferințelor prin intermediul rețelelor de calculatoare au deschis noi frontiere comunicațiilor. Pot fi folosite în diverse scopuri: comerciale, de divertisment și pentru învățământ, reprezentând cea mai puternică îmbinare dintre televiziune, transmisie de date sub formă numerică și calculatoare, presupunând metode și tehnici comune.

Unele materiale video ce provin din filmări pot fi necorespunzătoare din punct de vedere al calității, necesitând prelucrări ulterioare, în general aceste prelucrări fiind digitale, pot fi realizate pe calculatoare performante sau pe echipamente dedicate care pot conține procesoare de semnal la fel de puternice ca și un procesor aflat într-un calculator.

O problemă suficient de delicată este și cea a stocării informației. În timp, suporturile magnetice își pierd proprietățile, dar mai dureroasă este pierderea informației înregistrate. Deși prin faptul că un suport magnetic, în general, poate fi reînregistrat face ca majoritatea publicului să-l îndrăgească, necesitatea ca informația să rămână nealterată în timp a determinat ca suporturile optice să câștige teren. Recent se folosesc discurile magneto-optice pentru stocări, deoarece, acestea permit și reînregistrări. Pentru stocarea de informații sunt folosite diverse tehnici de compresie ce se regăsesc atât în domeniul televiziunii și a prelucrării de imagini, cât și din domeniul calculatoarelor unde se utilizează stocarea de date.

5.8.2. Necesitatea introducerii tehnicilor multimedia într-un studio de televiziune

Voi încerca o demonstrație a utilității tehnicilor multimedia printr-un exemplu: unul dintre programele cu cea mai mare audiență la public este cel de știri

(actualități). Realizatorul unui astfel de program are o grea problemă de rezolvat : gestionarea știrilor sosite și mai ales alegerea celor ce merită a fi transmise.

Unele mesaje nu sunt însoțite de suport video, ele putând fi știri sosite prin fax. Pentru o ușoară selecție a acestora, un sistem de achiziție pe calculator a mesajelor de tip fax este foarte bine venit, deoarece există posibilitatea ca textul mesajului, după ce a fost selecționat, să fie convertit într-un format ce poate fi afișat pe monitorul crainiceii ce va citi această știre. În plus, dacă există conexiune în rețea a calculatoarelor, este posibil ca toți redactorii de la propriile calculatoare să afle despre informațiile sosite la redacție (se elimină des întâlnita situație că cineva ia faxul și pleacă cu el...uitând că ar putea fi util și pentru alte persoane).

Nu există de cele mai multe ori posibilități financiare de a avea echipe de filmare peste tot în lume. De aceea, se folosesc corespondenți de presă care pot lucra în același timp pentru un post de radio, pentru un ziar dar și pentru un post de televiziune. Soluția cea mai des adoptată este folosirea unor corespondențe telefonice. Pentru acestea nu există un suport de imagine, se folosesc de cele mai multe ori fotografiile ale corespondenților de presă sau imagini de „arhivă”.

Prin introducerea videotelefonului la scară largă într-un viitor nu foarte îndepărtat corespondențele telefonice nu vor mai fi lipsite de imagini. Conversia semnalului video furnizat de acesta nu va reprezenta o problemă majoră, convertirea imaginii pentru afișarea pe ecranul televizorului fiind numai o mică problemă de hard și soft.

De asemenea, se poate căuta foarte rapid cu ajutorul calculatorului în arhiva studioului dacă există material video, sau imagini ce ar putea reprezenta suportul video pentru o știre care nu are un suport de imagine (evenimentul respectiv nu a fost filmat). Arhiva poate fi foarte variată, de la simple casete video înregistrate cu diverse programe, CD cu muzică, colecții de ziare, documente primite la redacție, etc. Este bine să existe o bază de date cu referințe la acele materiale. Ideal ar fi ca tot sistemul să fie sub formă digitală. Aceasta presupune o digitizare a întregii arhive și indexarea informațiilor. O asemenea operație este deosebit de anevoioasă, și de aceea se impun tehnici de indexare automată a secvențelor video dintr-un material video. Operație care cu ajutorul unui calculator performant se poate realiza. Existența unei arhive sub formă digitală ar permite accesul simultan pentru mai mulți utilizatori la aceleași materiale, acces mult mai rapid, deci o creștere a eficienței de lucru. De exemplu, pentru o simplă alegere a unui fundal muzical pentru un material video pot fi parcurse mai mult de zece CD-uri cu muzică, acestea trebuie luate de la arhivă, în general probabilitatea ca cineva să mai aibă nevoie de acele CD-uri este foarte mare, rezultând astfel blocaje în desfășurarea activității.

Cu ajutorul calculatorului se pot genera imagini statice sau în mișcare, imagini ce se pot folosi pentru generice sau efecte speciale [238], [239]. Posibilitățile în acest domeniu fiind nelimitate, totul depinzând de imaginația factorului uman ce le utilizează.

Comanda aparatelor din dotarea unui studio poate fi sincronizată și gestionată cu ajutorul unui calculator ce dispune de o interfață adecvată [240], [241], [242], [243]. Complexitatea echipamentelor actuale, necesită performanțe deosebite, mai ales că există mai multe standarde de televiziune, de înregistrare și de compresie [244], [245], [246], [247]. Prin acordurile de colaborare între posturile de televiziune se realizează frecvent schimburi de programe, achiziționarea de echipamente dedicate pentru astfel de conversii poate fi costisitoare. Folosind însă un sistem, ce are în componență un calculator, aceste conversii pot fi realizate software, iar dezvoltarea sistemului, a numărului de tipuri de conversii posibile depinde de dezvoltarea unor noi drivere care să realizeze interfațarea [248].

Recent multe posturi de radio au trecut pe sisteme complet digitale, de la modul de adunare al reportajelor până la modul de stocare în arhivă și comunicarea informațiilor audio în interiorul studioului. Personal am vizitat stațiile de radio Ile din Oulu Finlanda și France3 din

Nice, Franța iar dintre producătorii de echipamente îi pot menționa pe cei de la Audio-Follow cu care am avut o colaborare excelentă.

Proiectele de dezvoltare a unor astfel de studiouri digitale de televiziune au demarat deja, problemele mai greu de rezolvat fiind sistemele de stocare de mare capacitate și accesarea simultană a unor astfel baze de date de către mai mulți utilizatori. Datorită debitului mare al semnalului video și a constrângerilor temporale răspândirea unor astfel de sisteme nu s-a realizat încă la scară largă.

Se întâlnește tot mai des idea de „televiziune la cerere” sau „video-on-demand” [249], [250], [251], [252], [253]. În acest moment cel mai des sunt întâlnite programe de televiziune transmise prin Internet la rate de compresie mare (calitate redusă). Un număr foarte mare de cercetători în ultimul timp s-au orientat spre rezolvarea problemelor ce apar la transmiterea semnalului video prin Internet:

- probleme legate de codarea semnalului video [254], [255], [256], [257];
- probleme de rezolvare a congestiunilor ce apar în rețea [258], [259], [260], [261]. Pentru a realiza acest lucru au fost dezvoltate diverse tehnici care să se adapteze modului de transmitere prin asincron prin Internet [262], [263], [264].

Se folosesc variante în timp real – „live” sau variante înregistrate. Dintre posturile românești care oferă astfel de servicii pot aminti: România 1, Antena 1, ProTv Internațional. Se pare că a avut de câștigat interactivitatea în detrimentul calității.

O altă problemă din cadrul unui studio de televiziune este cea a realizării traducerilor din diverse limbi. Există două variante:

- subtitrare (traducere cu scris peste imagini și păstrarea originală a coloanei sonore);
- overvoice – traducere cu voce (se poate elimina în totalitate sau parțial coloana sonoră originală).

Curent se folosesc variante de genul: se transmite textul ce trebuie tradus (pe hârtie sau în format electronic) și materialul video. Eventual se mai trimite separat și coloana sonoră. Problema ce apare este la sincronizarea acestora, atunci când se reface materialul tradus. Aplicațiile multimedia permit includerea unor referințe temporale care pot rezolva aceste probleme. Folosind un asemenea sistem se poate transmite pe un același suport, în general DVD, materialul video, și subtitrările pentru mai multe limbi cu referințele de timp incluse. Utilizatorul nu trebuie decât să aleagă limba dorită și materialul va fi subtitrat în momentul vizionării. Există și varianta transmiterii prin teletext a subtitrării, dar performanțele și calitatea subtitrării este scăzută. Deci, încă o situație în care folosirea tehnicilor multimedia îmbunătățesc activitatea din cadrul unui departament al unui studio de televiziune – cel de subtitrare și traducere.

Apariția televiziunii de înaltă definiție a adus noi probleme privind debitul mare de date. Rezolvarea a constat în combinarea reducerii debitului prin compresii atât hard cât și soft [265], [266], [267], [268]. S-au dezvoltat și tehnologiile de transmisie ceea ce a determinat demararea dezvoltării acesteia.

În concluzie, se poate spune că existența unui element activ care să permită accesul, prelucrarea, gestionarea și arhivarea informațiilor de tipuri și formate foarte diferite, ce sunt vehiculate într-un studio de televiziune, este o necesitate ce trebuie îndeplinită.

5.9. Concluzii și soluții propuse pentru îmbunătățirea activităților dintr-un post de televiziune

Prin această lucrare încerc să rezolv câteva din problemele existente în posturile de televiziune prin metode dedicate aplicațiilor multimedia. Problemele pentru care propun soluții sunt:

- Informatizarea sistemului de gestionare a informațiilor primite la redacție prin introducerea acestora într-o bază de date.
- Optimizarea procesului redacțional prin stocarea știrilor într-o baza de date ce poate fi accesată de un număr mare de utilizatori.
- Ameliorarea calității artistice a materialelor video:
 - Pentru cele ce nu au suport video filmat prin folosirea de imagini și secvențe video existente într-o bază de date multimedia. Se oferă algoritmi performanți de căutare bazați pe conținut.
 - Prin realizarea de generice și animații folosind aplicații multimedia.
 - Folosind proprietățile texturilor și a modului în care acestea pot fi generate calitatea și numărul fundalurilor generate electronic crește spectaculos.
 - Efecte speciale – colorarea filmelor alb-negru prin metode automate ce segmentează imaginile pe baza texturilor.
 - Folosirea modelului de aplicație multimedia propus în aplicațiile de editare neliniară video permite definirea de efecte video proprii.
- Modelul propus de aplicație multimedia include o clasă de referințe temporale, denumită *Time_Line* ce poate fi utilizată la subtitrări și traduceri.
- Optimizarea procesului de arhivare a materialelor video prin metode de indexare automată după modele semantice de emisiuni.
- Dezvoltarea metodelor de arhivare prin folosirea de metode de compresie video performante, care să reducă spațiul ocupat de programe și să asigure în același timp o calitate cât mai bună. Autorul propune mai multe metode de compresie pentru stocarea materialelor video [17], [18], [19].
- Dezvoltarea schimbului de date între stațiile afiliate, prin folosirea unei baze de date multimedia ierarhică comună, distribuită prin Internet.
- Optimizarea schimbului național și internațional de programe, prin folosirea de tehnici multimedia de conversie digitală a standardelor Tv folosite.
- Îmbunătățirea imaginii publicitare prin prezentarea postului de televiziune pe Internet (Site WWW), oferirea de secvențe video în direct – „live”, distribuirea de CD-uri publicitare cu detalii legate de postul de televiziune (prețul unui CD ca suport poate fi mult mai mic decât prețul unui pliant publicitar color de 25-50, iar cantitatea de informație pe care acesta poate să o conțină este de ordinul zecilor de mii de ori mai mare).
- Gestionarea eficientă a clienților și a colaboratorilor.
- Dezvoltarea permanentă a nivelului de pregătire profesională a angajaților folosind pe lângă variantele clasice de pregătire (bazate pe materiale scrise [269], [270], [271]), metode moderne de învățământ la distanță, bazat în principal pe aplicații multimedia ce sunt accesabile prin intermediul Internetului (cursuri on-line pe web) sau pe CD-ROM [62], [63], [64].

- Verificarea calității semnalului video compresat prin metode adecvate ce includ o combinație între tehnicile subiective și cele obiective. Autorul a propus modalități de evaluare în [272], [273].

În figura următoare este prezentat circuitul informatic al datelor în cadrul unui studio de televiziune. Tendința actuală este de a trece circuitul datelor interne și a informațiilor externe pe un circuit comun cu cel al semnalelor video și audio. Acest lucru este posibil prin intermediul rețelilor de date de foarte mare viteză, rețele ce permit vehicularea semnalului video în cadrul unui studio de televiziune în format digital. În plus, majoritatea sistemelor de editare actuale folosesc variante computerizate ce au un număr foarte mare de avantaje. Deci integrarea acestora într-o rețea comună este o operație relativ simplă. Folosirea acestui sistem asigură o optimizare a tuturor activităților din cadrul unui studio de televiziune.

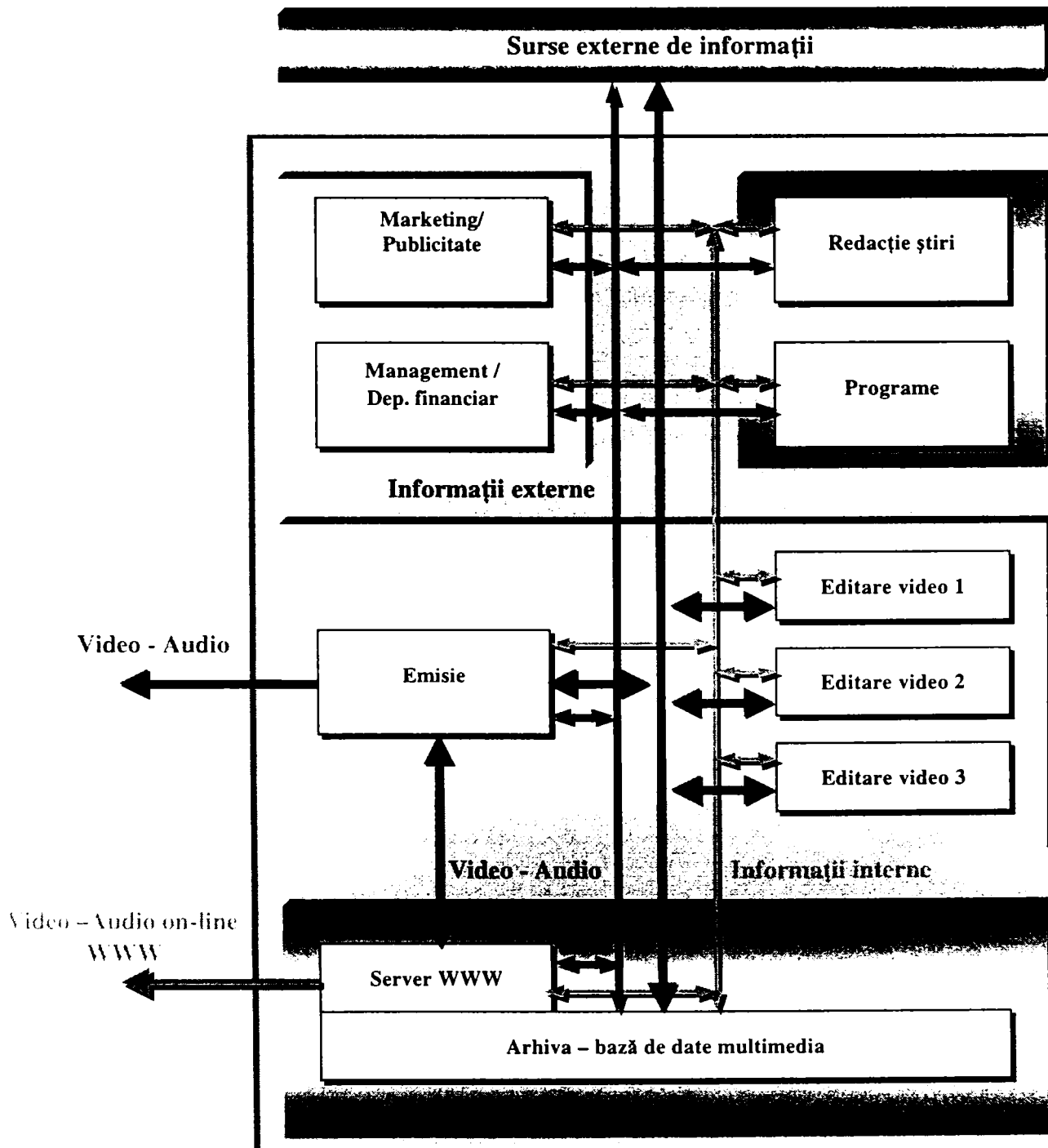


Figura 5 - 62 Circuitul informațional și audio-video într-un studio total digital

Capitolul 6.

Contribuții la implementarea bazelor de date multimedia

6.1. Analiza și proiectarea unei baze de date multimedia

În proiectarea bazelor de date multimedia se pot diferenția două mari etape:

- analiza cerințelor și dezvoltarea modelului conceptual al bazei de date – această etapă se referă în principal la studierea necesităților unui utilizator și la definirea modului de organizare și structurare a bazei de date la nivel conceptual.
- dezvoltarea modelului fizic de implementare – se referă la amplasarea fizică a datelor în mediile de stocare, la definirea și implementarea tuturor funcțiilor și serviciilor adiacente necesare [274].

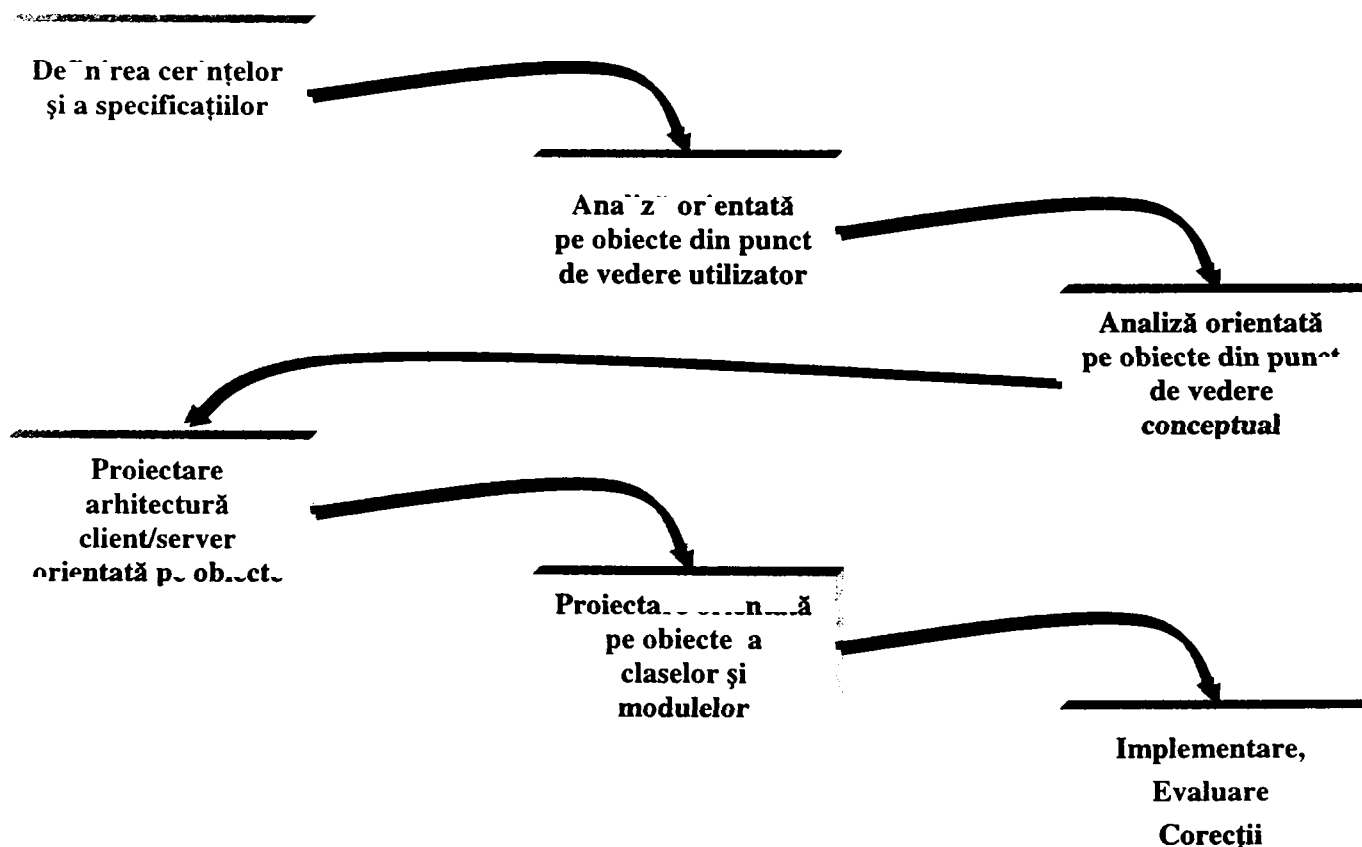


Figura 6-1 Structura analizei și a implementării bazei de date multimedia

6.1.1. Specificarea cerințelor

Sistemelor orientate pe obiecte li se cere în primul rând să modeleze și să implementeze lumea reală cât mai direct și mai ușor posibil [275]. În figura 6-1 am prezentat modul prin care am ajuns la implementarea bazei de date.

Funcționalitatea - baza de date implementată are ca principal scop stocarea de obiecte multimedia care să poată fi folosite de dezvoltatorii sau de aplicații multimedia. Ca scop secundar are extragerea de diverse obiecte dintr-o prezentare multimedia, obiecte care se pot folosi de exemplu în cadrul realizării de materiale audio-video din cadrul unui post de televiziune. Pentru îndeplinirea acestor principale cerințe și pentru a asigura funcționalitatea acesteia, trebuie ca baza de date să permită următoarele facilități, a căror implementare constituie contribuții originale ale autorului:

- Stocarea de diverse tipuri de obiecte, medii de transmitere a informației (text, sunet, grafice, imagini, video, animații);
- Stocarea unui număr mare de informații (atribute) ce descriu aceste obiecte;
- Utilizarea de funcții de prelucrare a obiectelor din baza de date multimedia. Aceste funcții se regăsesc în clasa *Processing_functions*;
- Utilizarea de metode și rutine ce se folosesc în aplicațiile multimedia (de exemplu, diverse tranziții și efecte de trecere de la o imagine la alta). Acestea sunt reprezentate de autor printr-o clasă aparte de obiecte denumită *Scripts* ;
- Autorul a realizat o lărgire a posibilităților de căutare în baza de date prin implementarea căutării bazate pe conținut, prin introducerea unei clase aparte denumită în implementare *Algorithms*. Folosind aceste facilități de căutare am mărit aplicabilitatea acestei baze de date într-un studio de televiziune;
- Dezvoltarea bazei de date am asigurat-o prin folosirea unor servicii prin care utilizatorii acceptați de către administratorul bazei de date pot introduce noi obiecte în cadrul bazei de date, noi funcții și metode de procesare a obiectelor, noi metode de interogare și căutare în cadrul bazei de date;
- Interfața grafică implementată respectă standardele și cerințele majorității utilizatorilor;
- Portabilitatea bazei de date pe diverse tipuri de calculatoare (PC, MAC) am asigurat-o prin implementarea în Java a bazei de date.
- Optimizare din punct de vedere a lungimii codului de program scris;
- Baza de date permite lucrul local dar și accesare prin rețea;
- Autorul a asigurat și facilități de accesare simultană de către mai mulți utilizatori;
- S-a asigurat o structurare ierarhică pentru optimizarea performanțelor și pentru extinderi ulterioare prin distribuirea bazei în rețele de calculatoare;
- Autorul a realizat și o variantă simplificată ce folosește mai puține resurse hardware prin combinarea dintre conceptul proiectării și reprezentării datelor orientat pe obiecte cu o stocare a informațiilor în baze de date tradiționale relaționale.
- Baza de date se înscrie în modelul de aplicații propus de autor în capitolul 3.

6.1.2. Contribuții la analiza orientată pe obiecte din punctul de vedere al utilizatorului

Principalul scop al acestei analize este identificarea entităților, relațiilor, operațiilor și constrângerilor de integritate ale bazei de date din punctul de vedere al utilizatorului. Principalele contribuții ale autorului în acest domeniu sunt sintetizate în cele ce urmează:

- Identificarea entităților (elemente multimedia, documente compuse, obiecte din lumea reală) cu care se lucrează, a relațiilor (de exemplu dacă există relații de moștenire sau relații de compoziție) dintre acestea și a operațiilor ce se pot aplica acestor entități.
- Identificarea aspectelor statice (am definit clase) și aspectelor dinamice (interacțiuni între obiecte, stări de tranziție) ale entităților folosite în aplicație.
- Identificarea tuturor operațiilor pe care un utilizator al bazei de date trebuie să le realizeze.
- Identificarea tuturor restricțiilor ce pot apărea pentru un utilizator (acest pas a ajutat la scrierea unei aplicații mult mai robuste, deoarece au fost căutate de la început restricțiile și situațiile, cazurile în care apar probleme).
- Realizarea unei interfețe grafice care să fie intuitivă și să nu pună probleme de exploatare pentru utilizator, urmărind și un grad mare de interactivitate.

În acest mod am căutat să identific modelul optim din punct de vedere al utilizatorului și care să rezolve cerințele acestuia.

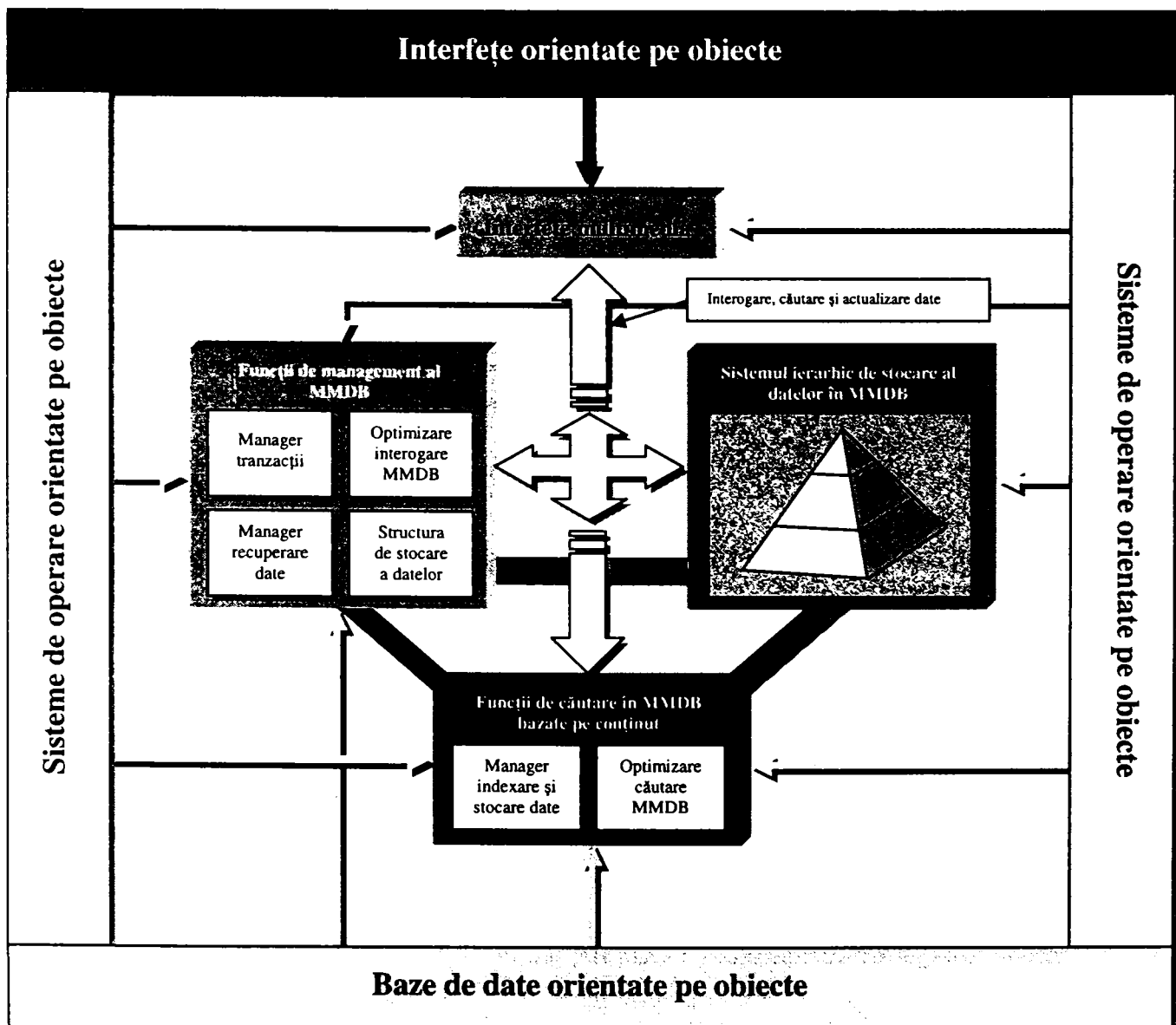


Figura 6-2 Integrarea bazei de date propuse într-un mediu orientat pe obiecte

6.1.3. Contribuții la analiza orientată pe obiecte din punctul de vedere al aspectului conceptual

Pentru definirea conceptului legat de modul de implementare a bazei de date am consultat bibliografie, dar am avut și discuții cu mai mulți utilizatori de aplicații multimedia, cu dezvoltatori de aplicații multimedia, cu persoane implicate în mai multe posturi de televiziune iar în plus m-au ajutat anii de experiență în care am realizat aplicații multimedia și materiale audio-video.

Modelul ce a rezultat a încercat să încorporeze foarte multe punctele de vedere individuale. Cunoașterea acestora mi-a permis să stabilesc mai exact problemele și cerințele din punct de vedere al utilizatorilor, dar și să elimin unele variante de implementare mai puțin performante prin anticiparea unor probleme ce puteau să apară. Consider că această etapă este foarte importantă și contribuie din plin la succesul aplicației finale.

O dată stabilite aceste elemente privind proiectarea orientată pe obiecte a bazei de date se poate trece la cea de a doua parte legată de implementarea fizică a bazei de date multimedia.

6.2. Integrarea bazei de date multimedia implementate în cadrul modelului de aplicație multimedia propus

Baza de date implementată se înscrie în modelul de aplicație multimedia propus, fiind o particularizare a acestuia prin slaba utilizare a unor elemente ale modelului.



Figura 6-3 Includerea bazei de date multimedia în modelul general de aplicații multimedia propus

Elementele modelului propus ce sunt bine dezvoltate în cadrul aplicației implementate sunt:

- *Objects* – fiind o bază orientată pe obiecte am acordat o atenție deosebită dezvoltării și definirii obiectelor multimedia folosite.
- *Browse/Search* – partea mai puțin dezvoltată a aplicațiilor multimedia curente (de exemplu, în cazul dezvoltării aplicațiilor ce rulează de pe CD-ROM, mediile de dezvoltare a aplicațiilor nu folosesc tehnologii de partajare de elemente, accesul la structura internă pentru dezvoltatorul de aplicații multimedia fiind puternic restricționată; aceleași „fenomene” se regăsesc în cazul sistemelor de editare video neliniare, unde introducerea unor noi efecte/tranziții personalizate de către un dezvoltator sunt imposibile) din perspectiva folosirii de obiecte comune din baze de date, partajarea bibliotecilor de funcții, am încercat să o extind și să permit ca un utilizator să își definească propriile tipuri de date pe care să le poată stoca în cadrul bazei de date și să le acceseze când este nevoie.
- *Input_Output_Operations* și *ODBC* – sunt intens folosite în acest tip de aplicație.
- *Client/Server, Users* și *Network* – pentru a fi posibilă utilizarea de către un număr cât mai mare de utilizatori a bazei de date, iar informația să fie accesibilă de la distanță, a impus folosirea unor astfel de tehnologii.

Elementele modelului propus ce sunt utilizate în mod obișnuit sunt:

- *GUI* – interfața grafică cu utilizatorul a fost dezvoltată și în tehnologie *applet* pentru ca baza de date să fie accesibilă prin intermediul paginilor de Web.
- *Processing_Functions* – sunt funcții necesare procesării obiectelor existente în baza de date.
- *Events* – evenimentele pe care trebuie să le trateze aplicația fac parte din categoria evenimentelor generale, nu am folosit evenimente particulare în aplicația implementată.
- *QoS* – se poate obține o părere greșită, falsă despre o aplicație, dacă acesteia nu i se asigură un minim de resurse hardware. De aceea, autorul propune ca la implementarea aplicațiilor să se realizeze înainte de utilizare un test al resurselor și să se avertizeze utilizatorul de rezultatul obținut.

Elementele modelului propus ce sunt mai slab utilizate (sunt prezentate în figura 6-3 pe un fundal de culoare mai închisă) sau lipsesc din cadrul bazei de date sunt:

- *Time_Line* – indicatorul de desfășurare temporală a acțiunilor în cadrul aplicației. Pentru o bază de date nu avem nevoie de o anumită desfășurare controlată de un parametru temporal. Acțiunile și funcțiile acesteia se desfășoară în ordinea firească și nu au durate impuse.
- *Transitions/Efects* – în general între obiectele cu care lucrează o bază de date nu se folosesc efecte și tranziții. Dacă se dorește în mod special, o ameliorare a prezentării grafice, se pot folosi și asemenea elemente.
- *DDE* - Dynamic Data Exchange permite construirea aplicațiilor de tip client-server. DDE permite actualizarea datelor dintr-o altă aplicație și facilitează executarea de comenzi “la distanță”. În cazul de față, aplicația în sine este de tip client/server și în general nu mai sunt necesare adăugarea de clase, metode și funcții care să permită utilizatorului dezvoltării suplimentare. Dacă utilizatorul dorește în mod special, acest lucru este permis.
- *DLL* – Dynamic Link Library am încercat evitarea acestor biblioteci de funcții pentru a facilita din plin portabilitatea aplicațiilor scrise în Java.
- *OLE* – *Object Linking and Embedding* este o extindere a conceptului DDE, ce permite includerea de date dintr-o altă aplicație și actualizarea referințelor editate în baza de date atunci când este editat documentul sursă. Când informația este “înglobată”, o copie a datelor este plasată în baza de date și memorată. Când informația este legată, este memorată doar referința la document. Am eliminat posibilitatea editării unui obiect din baza de date cu ajutorul unui program extern, din motive de securitate, dar am alocat un câmp (un

parametru) care să stocheze numele aplicației ce a creat obiectul. Astfel dacă se doresc modificări ulterioare, există referința la aplicația care a creat obiectul.

- *MCI – Media Control Interface* – aici sunt două situații posibile: elementele media să fie redade direct din baza de date sau acestea să fie descărcate local și după aceea să fie redade. În primul caz, am restricționat accesul utilizatorilor la gestionarea resurselor serverului.

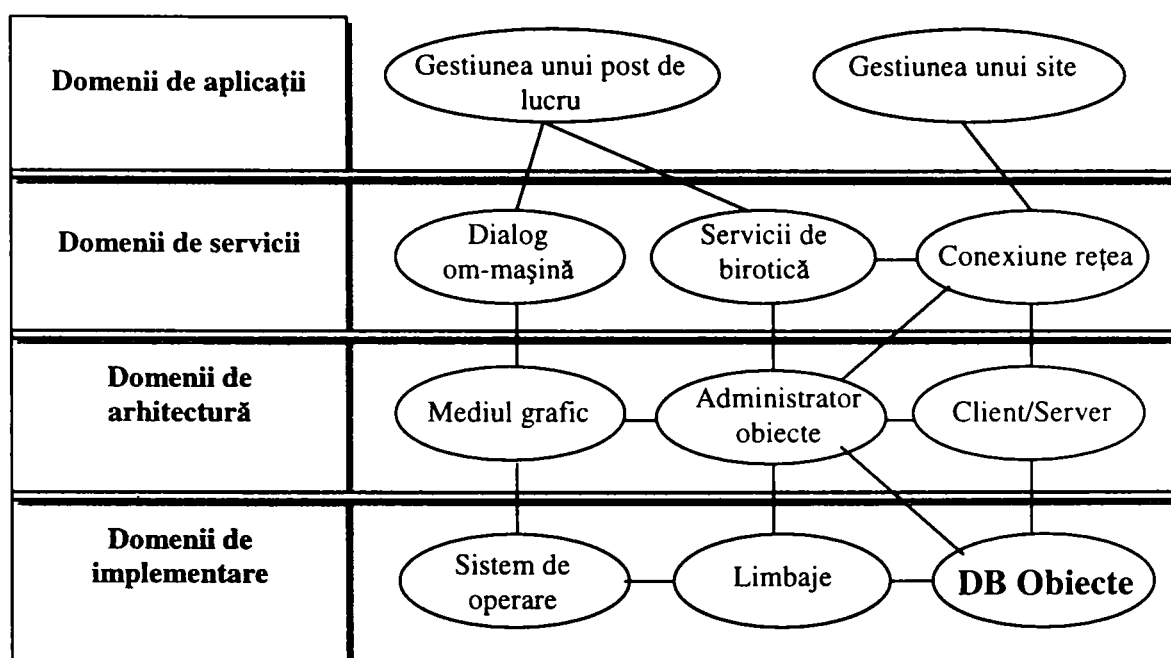


Figura 6-4 Legăturile unei baze de date în modelul de aplicație multimedia propus

6.3. Modul de organizare al bazei de date propuse

Sistemul folosește 4 tipuri generale de obiecte în organizarea generală a bazei de date. Acestea sunt stocate, respectiv utilizate pentru diverse operații ce se efectuează în cadrul aplicației:

- obiecte multimedia propriu-zise;
- algoritmi de extragere a proprietăților obiectelor multimedia;
- funcții de procesare a obiectelor multimedia;
- distanțe sau funcții de discriminare a obiectelor multimedia.

Pentru stocarea tuturor obiectelor se folosește o procedură unică, atât algoritmi de extragere a diverselor proprietăți, cât și funcțiile de procesare a obiectelor sau distanțele sau funcțiile de discriminare sunt înregistrate ca și obiectele propriu-zise. Prin utilizarea unui atribut pentru obiect se va realiza diferențierea apartenenței unui obiect la cele patru tipuri generale de obiecte.

6.3.1. Modul de indexare al bazei de date propuse

Pentru a realiza o bază de date performantă este necesar ca modul de indexare al datelor să permită diverse moduri de căutare, iar acestea să fie pe deplin optimizate. Am propus în aplicația realizată trei moduri principale de căutare:

- căutare după descrierea semantică a obiectului - folosesc 60 de câmpuri de înregistrare.

- căutare ce folosește o imagine ca model - utilizatorul trebuie să ofere o imagine drept model, iar algoritmul de căutare va determina cele mai apropiate imagini de acest model. Modul de căutare este interactiv și permite utilizatorului redefinirea parametrilor de căutare.
- căutare ce folosește parametrii de descriere ai obiectelor multimedia - acest mod de căutare presupune ca utilizatorul să aibă cunoștințe în domeniul prelucrărilor de imagini, pentru a cunoaște efectul fiecărei proprietăți în procesul de căutare.

Pentru indexarea bazei de date folosesc următoarele categorii de proprietăți:

- **Parametrii de identificare obiect:**

- Identificator obiect: *ID Object*
- Tip Obiect: *Object_Type*
- Categoria de obiecte: *Object_Category*
- Flag de confirmare a faptului că obiectul respectiv face parte dintr-o structură complexă a unui obiect: *Object_Complex*
- Identificatorul obiectului complex de care aparține obiectul: *ID Object_Complex_Father*
- Identificatorul obiectului care este legat de obiectul curent: *ID Object_Complex_Child*
- Fișierul sursă: *Source*
- Formatul fișierului: *Format_File*
- Dimensiunea obiectului: *Size*

- **Parametrii de descriere prin cuvinte cheie** - propun următoarele categorii care realizează:

- Descrierea localizării spațiale: *Spatial*
- Descrierea reliefului: *Landscape*
- Descrierea aspectului de urbanism: *Urban*
- Descriere temporală: *Temporal*
- Descriere a animalelor: *Animals*
- Descriere a plantelor: *Plants*
- Denumirea obiectelor: *Names*
- Domeniul de activitate: *Domain*
- Categoria: *Category*
- Categorie rezervată utilizatorilor: *Users*

Pentru fiecare categorie am rezervat 6 câmpuri ce vor detalia descrierea obiectului.

- **Parametrii de descriere a obiectului** - ce descriu proprietățile generale ale obiectului - acești parametrii sunt folosiți pentru descrierea obiectului, informații specifice și atribute ce descriu elementul din punct de vedere al conținutului. Parametrii folosiți ce descriu proprietățile obiectului au fost aleși în conformitate cu modelul original propus de autor în cadrul capitolului 4. Ei diferă în funcție de tipul obiectului.

- Pentru clasa de obiecte de tip **Objects** avem:

Parametrii globali ai obiectelor de tip *Multimedia_Object*:

- Height
- Width
- ColorNumber
- Sounds
- Frame_Rate
- Scripts
- WEB/CD/Slides/Video

Parametrii globali ai obiectelor de tip *Sound*:

- Channels_Number
- Sampling_Rate
- Multitrack
- Compresion
- Format
- Length
- Time_Action

Parametrii globali ai obiectelor de tip *Image*:

- Dimension
- Color_Numbers
- Color_Model
- Layers
- Compresion
- Real/PC

Parametrii globali ai obiectelor de tip *Animation*:

- Dimension
- Color_Numbers
- Color_Model
- Sound
- Format
- Layers
- Compresion
- Lenght
- Time_Action

Parametrii globali ai obiectelor de tip *Text*:

- Text_Box
- Document
- Field
- Hand_Write
- Position

Parametrii globali ai obiectelor de tip *Graphics*:

- Dimension
- Color_Numbers
- Color_Model
- Layers
- Compresion
- 2D/3D

Parametrii globali ai obiectelor de tip *Video*:

- Dimension
- Color_Numbers
- Color_Model
- Sound
- Format
- Compresion
- Frame_Rate
- Lenght

Parametrii globali ai obiectelor de tip *Scripts*:

- Languages
- Events
- Actions
- Format
- Compresion
- Time_Action

□ Pentru clasele de obiecte de tip **Algorithms**, **Processing_Functions**, și **Distances** avem posibilitatea de a defini următoarele informații la înregistrarea acestora în baza de date:

- Tipul obiectului: *Algorithms*, *Processing_Functions* sau *Distances*.
- Categoria de obiecte asupra căreia acționează: *Multimedia_Objects*, *Text*, *Sound*, *Graphics*, *Image*, *Video*, *Animation*, *Scripts*.
- 10 parametri ce definesc funcția respectivă. Pentru fiecare parametru se pot defini de către utilizatorul ce introduce funcția respectivă 3 valori ce descriu limitele de variație ale

parametrului: *Min_Value*, *Nom_Value*, *Max_Value* (valoarea minimă, nominală recomandată de utilizator și valoarea maximă).

- Un utilizator poate personaliza funcția prin introducerea pentru parametri a propriilor valori în cadrul câmpului *User_Value*.
- Fiecare parametru are un câmp de descriere a scopului în care acesta este utilizat: *Description*.
- Este eficient, din punct de vedere al utilizării, ca includerea parametrilor în funcția respectivă să fie confirmată prin intermediul unui flag *Use*.

Acești parametri sunt folosiți doar parțial în procesul de indexare, nu contează foarte mult pentru utilizatori în procesul de căutare valorile parametrilor, ci contează descrierea realizată în câmpurile *Description*.

- **Parametrii utilizator** - ce descriu proprietățile particulare introduse de utilizatori. Introducerea de noi parametri nu este o problemă pentru baza de date propuse. Calculul acestor parametrii poate fi realizat de către utilizator separat sau prin introducerea pentru început a unor algoritmi de calcul.

6.3.2. Algoritmii de căutare propuși

Se cunosc atât avantajele, cât și dezavantajele folosirii singulare a unei indexări bazate pe conținut sau a celei semantice. Printr-o interacțiune cu utilizatorul se poate alege unul din cele două moduri sau se folosesc împreună.

- **Căutare bazată pe o descriere semantică** - Pentru o descriere completă a unui obiect am folosit o descriere ce folosește **11** categorii pentru descrierea unui obiect. Fiecare categorie poate fi detaliată prin folosirea a **6** elemente descriptive.
- **Căutare după parametrii generali ai obiectului** - parametrii generali ai obiectului rezultă din caracteristicile sale ce descriu aspectul exterior, pentru o imagine de exemplu se folosesc proprietățile generale: dimensiunile imaginii, numărul de culori, modul de compresie folosit, tipul imaginii 2D/3D, numărul de nivele. Folosirea acestor parametri este folosită în general în cadrul procesului general de administrare al bazei de date
- **Căutare după proprietățile obiectului** - pentru căutarea obișnuită a utilizatorilor se folosesc proprietățile obiectului, proprietăți impuse de autor ca fiind cele mai reprezentative pentru procesul de căutare:

- *Proprietăți globale*
 - histograma luminanței
 - histograma pe componente de culoare
 - 5 momente ale histogramei.
- *Proprietăți statistice*
 - media
 - dispersia
 - momente de diverse ordine
- *Proprietățile texturilor*
 - contrast
 - entropie
 - omogenitate

- *Proprietățile obiectelor*

Foarte important în procesul de căutare este modul de utilizare a proprietăților folosite:

- propun metode de căutare ce utilizează în procesul de discriminare distanțe ce folosesc toate proprietățile dorite de utilizator
- metode de căutare structurate sub formă de arbori pentru evitarea parcurgerii întregii baze de date - avantajele acestei metode fiind viteza mare de parcurgere a bazei de date prin evitarea verificării întregii baze de date și simplificarea calculelor
- **Căutare după proprietățile utilizatorului** - pentru ca sistemul să se adapteze la nevoile utilizatorului, pe lângă proprietățile obiectului am introdus și posibilitatea ca acesta să își poată dezvolta propriile metode de căutare folosind propriile proprietăți ale obiectelor, eventual combinate cu cele generale.
- **Căutare complexă** - prin căutare complexă se înțelege o căutare care folosește o combinație de proprietăți generale și utilizator discriminate prin intermediul unei distanțe sau folosind rețele neuronale.
- **Căutare în cazul imaginilor folosind porțiuni dintr-o imagine** - în cazul folosirii unei imagini ca model se poate alege o anumită regiune dintr-o imagine pentru extragerea unor proprietăți, pentru ca mai apoi folosind valorile obținute să căutăm în baza de date imagini cu proprietăți similare.
- **Algoritm de căutare rapidă** - baza algoritmului de căutare rapidă folosește o structură ierarhică arborescentă a proprietăților, o dată ce un element din baza de date nu respectă una din proprietăți (parcurgerea proprietăților se face în funcție de opțiunea utilizatorului), el va fi abandonat, și nu se vor verifica pentru el celelalte proprietăți. Detalii sunt date în capitolul 5, structura algoritmului propus fiind prezentată în figura 5-64.
- **Sistem de căutare interactiv** - particularitatea sistemului de căutare, al aplicației implementate, constă în definirea unui număr mare de modalități de căutare propuse prin sistemul de reconfigurare a algoritmului de căutare:
 - folosirea unui număr mare de metode (distanțe, algoritmi) de discriminare;
 - parametrii folosiți în procesul de căutare se pot redefini iterativ de către utilizator;
 - există un grad mare de interactivitate cu utilizatorul;
 - posibilitatea de introducere de către utilizator a metodelor proprii de căutare;

Unul din avantajele sistemului propus de autor este interactivitatea sistemului de căutare, acesta permițând:

 - setarea priorităților în cadrul proprietăților folosite în sistemul de căutare;
 - setarea valorilor pentru anumite proprietăți;
 - redefinirea parametrilor de căutare în funcție de rezultatele obținute în urma ultimei căutări;
 - configurarea arborelui de căutare prin:
 - setarea modului de parcurgere;
 - setarea valorii de comutare;
 - setarea a două limite ce determină intervalul de comutare pentru arborele de căutare.
 - calcularea de noi proprietăți pentru obiecte.
 - folosirea de sisteme neliniare de discriminare (rețele neuronale);
 - prelucrări ajutătoare pentru obiectele implicate în proces;
 - posibilități de reconfigurare a modului de discriminare a proprietăților

“Nimic nu este fixat, totul este reconfigurabil și adaptabil la cerințele utilizatorului.”

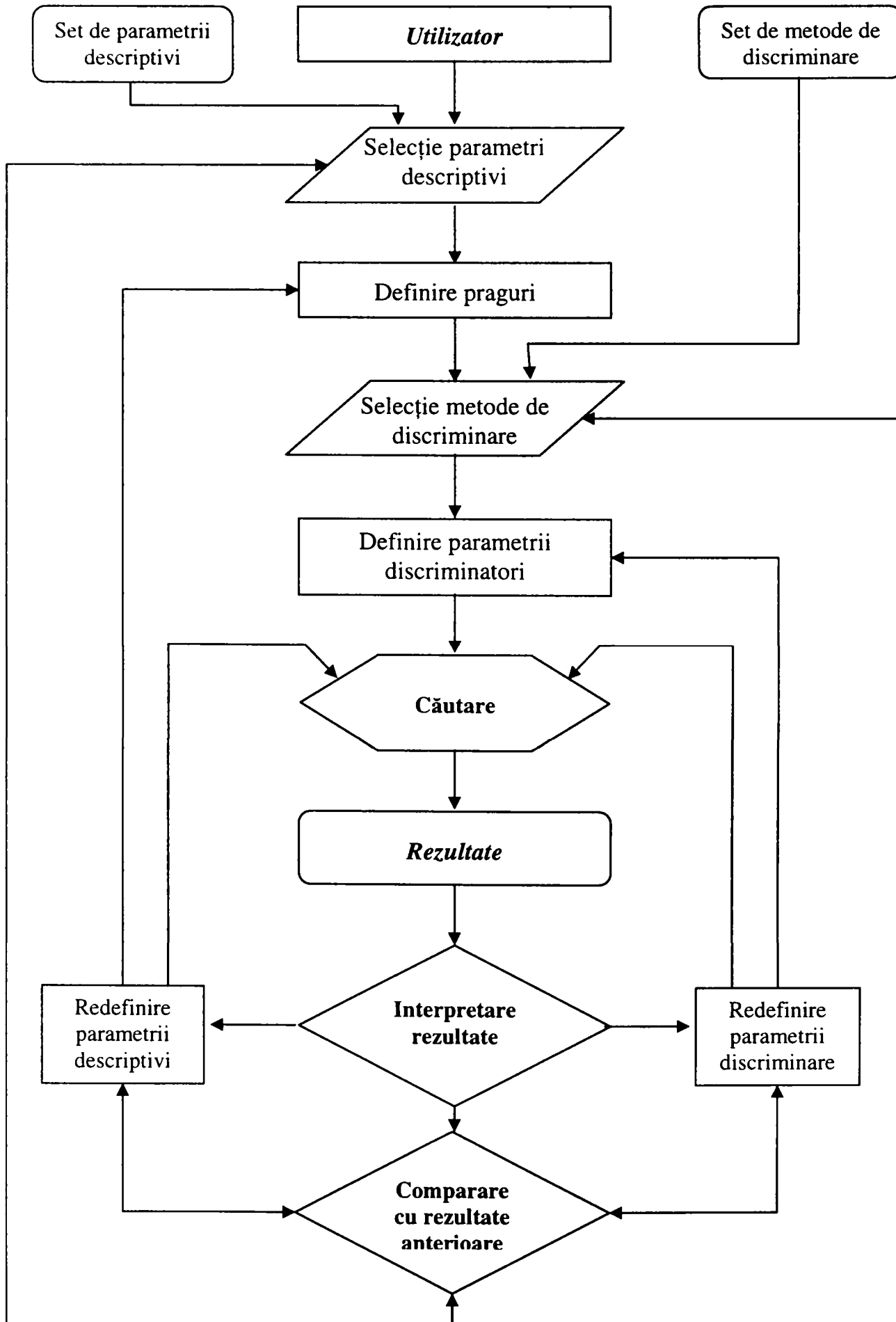


Figura 6-5 Legăturile unei baze de date în modelul de aplicație multimedia propus

Creșterea vitezei de căutare se poate face în mai multe moduri: utilizarea unui număr minim de caracteristici descriptive pentru imagini, utilizarea unor metode de discriminare rapide și robuste, iar pentru creșterea puterii de căutare se pot permite variante iterative de definire a parametrilor. Combinarea acestor idei, prin utilizarea unei interactivități umane a permis ca rezultatul căutării să poată fi obținut mai rapid prin parcurgerea următoarelor etape:

- *Definirea parametrilor descriptivi (alegerea celor mai potrivite caracteristici pentru descrierea imaginii)* – în această etapă utilizatorul este cel care știe ce dorește să găsească în baza de date. De exemplu, o imagine alb-negru sau color, o imagine ce conține texturi cu o anumită granularitate sau cu un anumit grad de eterogenitate, o imagine caracterizată de linii orizontale sau verticale – cunoscând caracteristica imaginii căutate, utilizatorul va alege acei parametri care descriu mai bine imaginea căutată. Modul în care au fost aleși acești parametri este deosebit de important deoarece, combinarea unui număr mare de caracteristici poate determina scăderea performanțelor în procesul de căutare, iar similitudinea dintre două imagini poate fi definită diferit în funcție de ce se dorește a se obține. De exemplu, căutarea unor imagini ce conțin preponderent dungi verticale poate fi realizată folosind doar filtre Gabor orientate corespunzător. Rezultatul este o serie de imagini care au dungi verticale dar la care nu contează în mod deosebit media imaginii (imaginile pot fi cu dungi negre pe un fundal alb sau gri – imagini similare din punct de vedere al dungilor verticale dar total diferite din punct de vedere al mediei).
- *Definirea pragurilor de toleranță pentru caracteristicile utilizate* – aici se vor tria imaginile ce se folosesc în procesul de discriminare prin impunerea unor toleranțele ale caracteristicilor ce trebuie respectate. Impunerea de praguri se transformă într-o selecție a imaginilor ce vor fi folosite la discriminare, metoda fiind foarte rapidă. Avantajul major constă în reducerea considerabilă a numărului de imagini folosite efectiv în procesul de căutare.
- *Selecția metodei de discriminare* – există mai multe tipuri de metode care oferă performanțe deosebite doar pentru un anumit tip de vectori de intrare (format din caracteristicile selectate anterior). Metodele de discriminare ce folosesc distanțe, suferă în general de un fenomen de integrare a caracteristicilor, astfel două caracteristici complementare ca valoare se pot compensa reciproc, în comparație cu o metodă cu praguri ce ar elimina imaginea respectivă din lista posibilelor imagini căutate. Acolo unde este posibil să existe un set de imagini de antrenare pentru o rețea neuronală, acestea din urmă pot oferi rezultate foarte bune.
- *Definirea parametrilor de discriminare* – este necesară și în acest caz o modelare a metodelor de discriminare în funcție de ceea ce se caută în baza de imagini.
- *Căutarea* – este procesul prin care vectorii de intrare selectați în prealabil se aplică metodei de discriminare, iar ca efect vom obține imaginile cele mai apropiate de cea dorită.
- *Interpretarea rezultatelor* - este realizată de către factorul uman. Mulțumirea sau nemulțumirea acestuia poate determina trecerea la etapele ulterioare de redefinire a algoritmului de căutare.
- *Selecția caracteristicilor / Redefinirea parametrilor descriptivi și a celor de discriminare*
- *Repetarea procesului de căutare* - are loc o după redefinirea parametrilor algoritmului, iar rezultatele obținute vor fi comparate cu cele obținute în iterația anterioară pentru a se decide în ce sens se vor ajusta parametrii de căutare.

Creșterea vitezei de căutare rezultă din:

- scăderea numărului de caracteristici folosite în procesul de căutare (lucru posibil prin selecția de către utilizator a caracteristicilor celor mai potrivite pentru tipul de căutare realizat).
- alegerea metodei de discriminare care definește cel mai bine modul de căutare al utilizatorului.
- redefinirea parametrilor până la optimizarea procesului.

6.3.3. Implementare propusă pentru translatarea bazelor de date relaționale existente în bază de date cu facilități de manipulare a obiectelor multimedia.

Am dezvoltat un model de implementare al unei baze de date multimedia care să se folosească de facilitățile oferite de bazele de date relaționale existente.

Implementarea propusă se bazează pe o structură compusă din 3 mari module funcționale:

- **Interfața cu utilizatorii** – acest modul este reprezentat în principal de partea de introducere a datelor și de partea de afișare a rezultatelor obținute. Sunt prevăzute 4 submodule pentru:
 - introducerea obiectelor multimedia;
 - introducerea parametrilor de procesare a obiectelor multimedia;
 - definirea parametrilor de indexare (ce vor fi extrași din obiectul ce va fi introdus în baza de date) ce vor fi folosiți ulterior în procesul de căutare.
 - introducerea parametrilor după care se dorește să se realizeze căutarea (fereastra de căutare).

Mai există un submodul pentru operațiile de afișare a rezultatelor obținute și trecerea la redefinirea parametrilor de căutare.

Comunicarea cu modulul de extensie multimedia a bazei de date relaționale se bazează pe trei fluxuri de ieșire (parametrii de procesare / caracteristicile ce vor fi extrase din obiectul introdus în baza de date și parametrii de căutare) și un flux de intrare (rezultatele obținute în urma procesului de căutare).

Comunicarea cu baza de date ce stochează efectiv obiectele multimedia se bazează pe flux unidirecțional prin intermediul căruia se introduc obiectele în baza de date într-un format de tip *BLOB*. Partea de afișare a obiectelor din bază va trece prin modulul de extensie multimedia a bazei de date, deoarece pe lângă informația primară se va mai adăuga și informația suplimentară ce permite manipulări de tip multimedia.

- **Extensia multimedia a bazei de date** – este modulul ce se atașează unei baze de date relaționale (obișnuite) și constă în principal din tabele ce memorează informații ce permit manipulări de date din punct de vedere multimedia (căutări bazate pe conținut / diverse structurări și clasificări, etc.). Pentru extensia multimedia a bazei de date propun utilizarea a 10 tabele în următorul mod:
 - *Parametrii de procesare a obiectelor* – este necesară memorarea modului în care a fost procesat un obiect din baza de date pentru prelucrări ulterioare. În plus, folosind un singur obiect și mai multe moduri de procesare se pot obține rezultate cu utilitate în diverse domenii. Aici pot apare două situații:
 - se stochează în baza de date obiectul inițial și în tabel parametrii de procesare;
 - se stochează atât obiectul inițial cât și rezultatele prelucrării obiectului inițial, plus parametrii de procesare;
 Prima situație prezintă avantajul unui spațiu mai mic de stocare utilizat, dar în procesul de utilizare a bazei de date pot apare situații cand sunt necesare refacerea procesării (afișarea obiectului procesat și nu a obiectului inițial), lucru ce poate fi un consumator mare de timp. Cel de al doilea caz permite o rapidă utilizare inclusiv a obiectelor procesate, dar va fi nevoie de un spațiu mai mare de stocare.
 - *Funcții de procesare a obiectelor* – aici vor fi stocate informații referitoare la funcțiile de procesare disponibile pentru un anumit tip de obiect multimedia, valori minime, maxime și recomandabile pentru parametrii acelor funcții, descrieri ale modului de utilizare și performanțe. Aceste informații permit utilizatorului ce nu

beneficiază de o specializare deosebită în domeniu să poată utiliza mai ușor sistemul propus.

- *Parametrii de extragere a caracteristicilor obiectelor* – este necesară memorarea modului în care au fost extrase caracteristicile descriptive ale unui obiect din baza de date. În plus, folosind un singur obiect și mai multe moduri de definire pentru aceleași caracteristici se pot obține variante de indexare diferite și diverse criterii de căutare în baza de date.
- *Algoritmi de extragere caracteristici obiecte* – aici vor fi stocate informații referitoare la algoritmi de extragere a caracteristicilor disponibili pentru un anumit tip de obiect multimedia, valori minime, maxime și recomandabile pentru parametrii acestor algoritmi, descrieri ale modului de utilizare și performanțe. Aceste informații permit utilizatorului ce nu beneficiază de o specializare deosebită în domeniu să poată utiliza mai ușor sistemul propus.
- *Obiecte procesate* – tabel ce va conține obiectele procesate. Dimensiunea acestuia poate fi foarte mare, deoarece conține obiectele procesate înainte de înscriere în baza de date.
- *Caracteristici* – acest tabel va conține caracteristicile obiectelor existente în baza de date și legături la celelalte tabele ce descriu modul de extragere a acestora. Aici vor fi folosite la maxim facilitățile bazelor de date relaționale.
- *Distanțe* – informațiile referitoare la algoritmi de discriminare a obiectelor (distanțele) pe baza caracteristicilor disponibile pentru obiectele multimedia sunt stocate în acest tabel. Sunt disponibile valori minime, maxime și recomandabile pentru aceste distanțe, descrieri ale modului de utilizare și performanțe. Aceste informații permit utilizatorului ce nu beneficiază de o specializare deosebită în domeniu să poată utiliza mai ușor sistemul propus.
- *Parametrii de căutare* – reprezintă un tabel cu parametrii după care s-a realizat o căutare. Recomand stocarea parametrilor pentru fiecare căutare, inclusiv o legătură la tabelul cu rezultatele obținute în scopuri administrative și de optimizare a bazei de date.
- *Rezultatul căutării* – rezultatele fiecărei operații de căutare sunt stocate aici, împreună cu legătura la tabelul ce conține parametrii de căutare. Tabelul conține doar identificatorii obiectelor și nu obiectele în sine.
- *Descriere globală a obiectelor din baza de date* – informații globale de descriere a obiectelor din baza de date se află în acest tabel (data când a fost realizată inserarea obiectului, utilizatorul care a realizat acest lucru, descriere semantică, etc.).
- **Baza de date relațională** (ce permite stocarea obiectelor în format BLOB). Aici se regăsesc 4 mari categorii de obiecte stocate în module separate:
 - *Funcții procesare obiecte;*
 - *Algoritmi de extragere a caracteristicilor;*
 - *Distanțe utilizate la căutare;*
 - *Obiecte multimedia.*

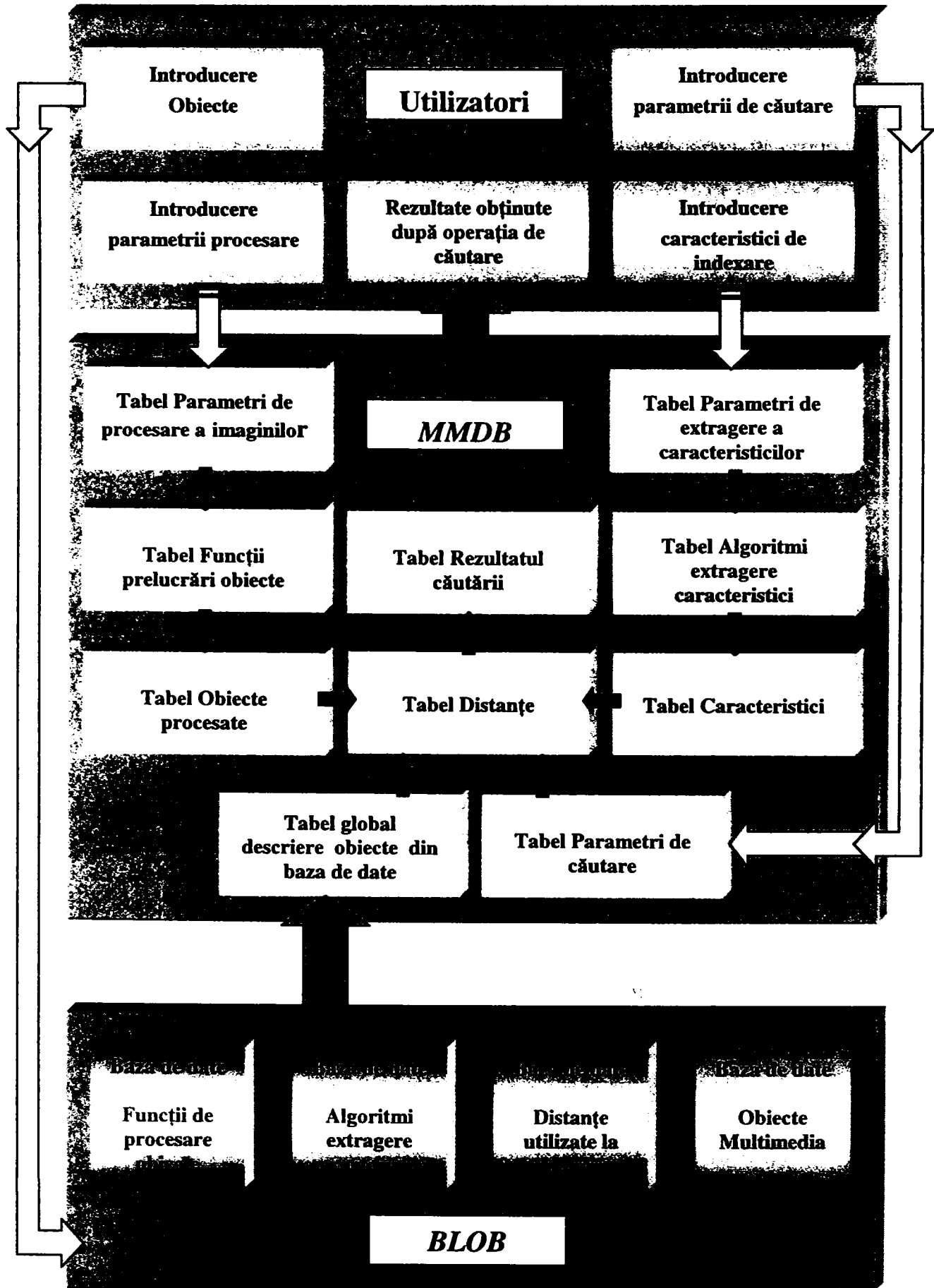


Figura 6-6 Detalii privind implementarea în cadrul bazei de date pentru obiecte de tip imagine

6.3.4. Modalitate de mărire a vitezei de căutare

Modalitate de creștere a vitezei propuse are la bază ideea de creare a unor grupări de obiecte pe diverse categorii, urmând ca la căutare să se evalueze rapid categoria din care face parte obiectul respectiv, iar numai pentru categoria determinată să se realizeze o căutare mult mai profundă (folosind mai multe proprietăți ale imaginilor). De exemplu, se pot împărți toate texturile dintr-o imagine în 10 categorii. Există un efort de calcul la realizarea partiționării în cele 10 categorii, în schimb la căutare se vor verifica doar 10% din texturile existente în baza de date. Pentru partiționare pot fi utilizate metode ce implică factorul uman, dar se poate folosi și metoda propusă în capitolul 5.4 ce folosește rețele neuronale de tip hartă de trăsături cu auto organizare. Deși foarte simplă ca și concepție, metoda se dovedește a fiind foarte eficientă.

6.3.5. Implementarea software

- **Orientarea pe obiecte** – oferită de limbajul de programare Java mi-a permis o optimizare a aplicației folosind avantajele programării orientate pe obiecte.
- **Structurare ierarhică** – în foarte multe situații am organizat informațiile sub o formă ierarhică ce permite o mai bună organizare. Pentru anumiți algoritmi de căutare am folosit ca formă de organizare o structură ierarhică arborescentă ce permite de multe ori scurtarea timpului de căutare. Algoritmul a fost detaliat în capitolul anterior.
- **Tehnologie *client-server*** – aplicația este concepută sub forma a trei module: client - server - baza de date.

Clientul poate fi:

- aplicația ce rulează local pe calculatorul utilizatorului și care are ca principală sarcină afișarea interfeței grafice de dialog.
- o pagină HTML ce conține un applet Java cu aplicația propriu-zisă.

Clientul va realiza legătura dintre utilizator și server. Clientul va prelua sarcina de afișarea a interfeței grafice și va ajuta utilizatorul la definirea interogării bazei de date. Prin intermediul clientului se vor realiza următoarele:

- introducerea de noi obiecte în bază, prin definirea de obiecte ce vor fi transferate serverului pentru înregistrarea în baza de date;
- stabilirea proprietăților utilizator ale obiectului care vor fi calculate suplimentar față de cele impuse, calculul acestora și stocarea lor fiind realizată de către server;
- stabilește cu utilizatorul modul de interogare al bazei de date;
- afișează rezultatele obținute de server în urma operației de căutare.

Serverul va realiza operațiile:

- identificarea utilizatorilor;
- coordonarea legăturii cu clientul, verificarea integrității datelor transmise de acesta;
- efectuarea de operații cu obiectele din baza de date (înregistrarea de noi obiecte, modificarea celor existente, ștergerea de obiecte);
- realizarea calculului de proprietăți generale pentru obiectele înregistrate în baza de date;
- introducerea de noi proprietăți utilizator, calculate ca urmare a cererilor ce vin din partea utilizatorilor.
- realizarea de căutări în baza de date în conformitate cu modul de interogare transmis de către utilizator prin intermediul clientului.

- transmiterea rezultatelor obținute în urma căutării clientului.
- **Facilitate de colaborarea cu sisteme de baze de date tradiționale** - Access, Oracle, DB2, MySQL – poate fi realizată prin intermediul ODBC sau JDBC. Folosirea unui asemenea mod de lucru, de partajare a sarcinilor, permite folosirea de mai multe sisteme de management a bazelor de date existente. Trecerea de la un sistem la altul implică scrierea câtorva drivere. Autorul a folosit ca mod de implementare limbajul Java pentru care se regăsesc o serie foarte mare de drivere ce pot comunica cu diverse tipuri de date.

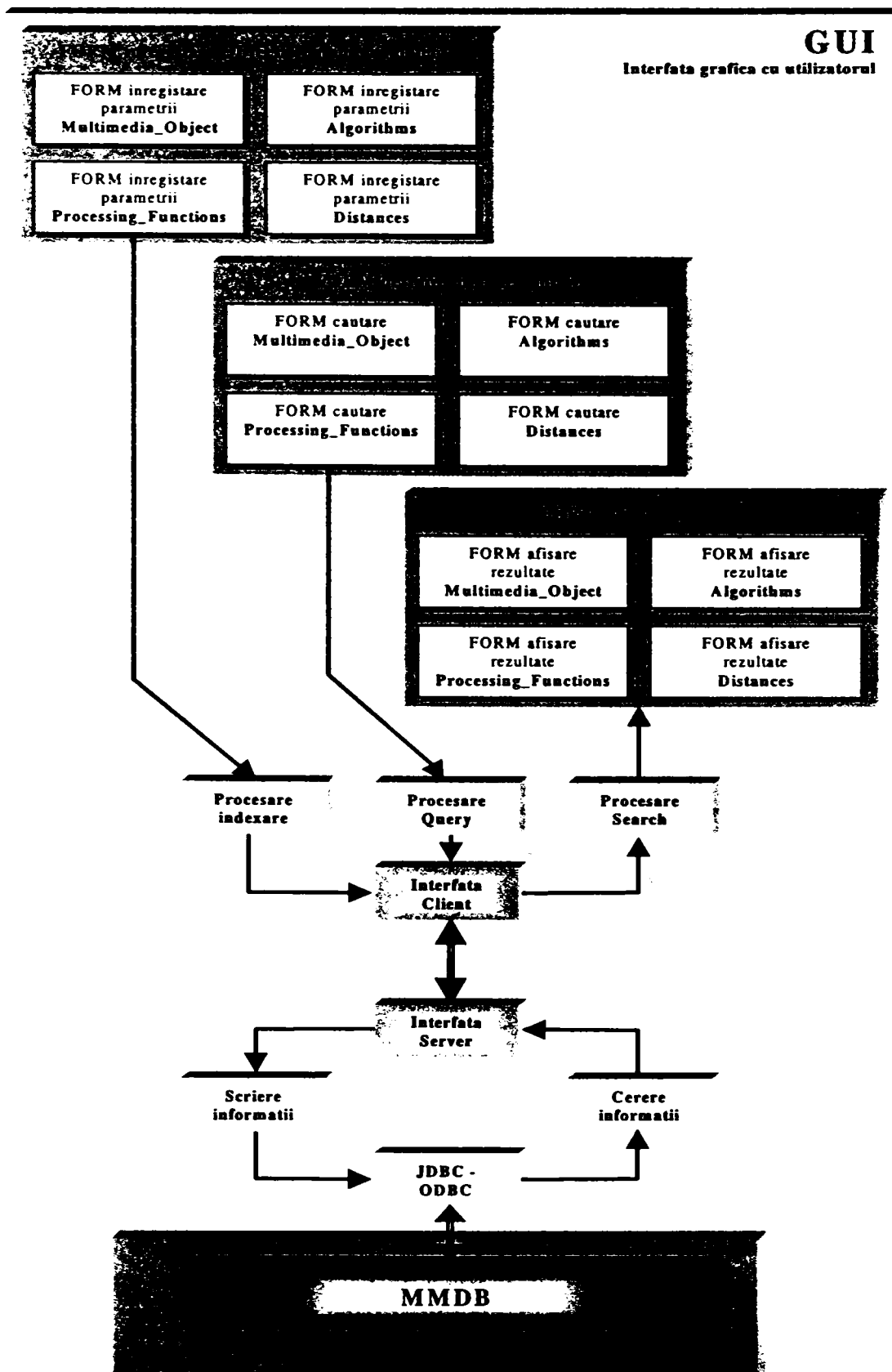


Figura 6-7 Modul de implementare modular al aplicației

În figura 6-7 este prezentată organizarea modulară a aplicației. Modificarea doar a unui modul poate permite conectarea sistemului la diverse tipuri de baze de date. Folosirea tehnologiei client server permite conectarea mai multor clienți simultan la baza de date [276].

- **Interfața cu utilizatorul – GUI** – principalele ferestre sunt prezentate în figurile următoare. S-a căutat o tratare generalizată a obiectelor ce vor fi inserate, diferențierile apărând abia atunci când sunt calculate caracteristicile obiectului. Caracteristicile globale sunt extrase în mod automat.

Am realizat interfața aplicației în ținând cont de cele trei tipuri de operații ce se realizează în general: *inserarea de obiecte, căutarea, afișarea rezultatelor*. În continuare am prezentat câteva din ferestrele mai importante ale aplicației.

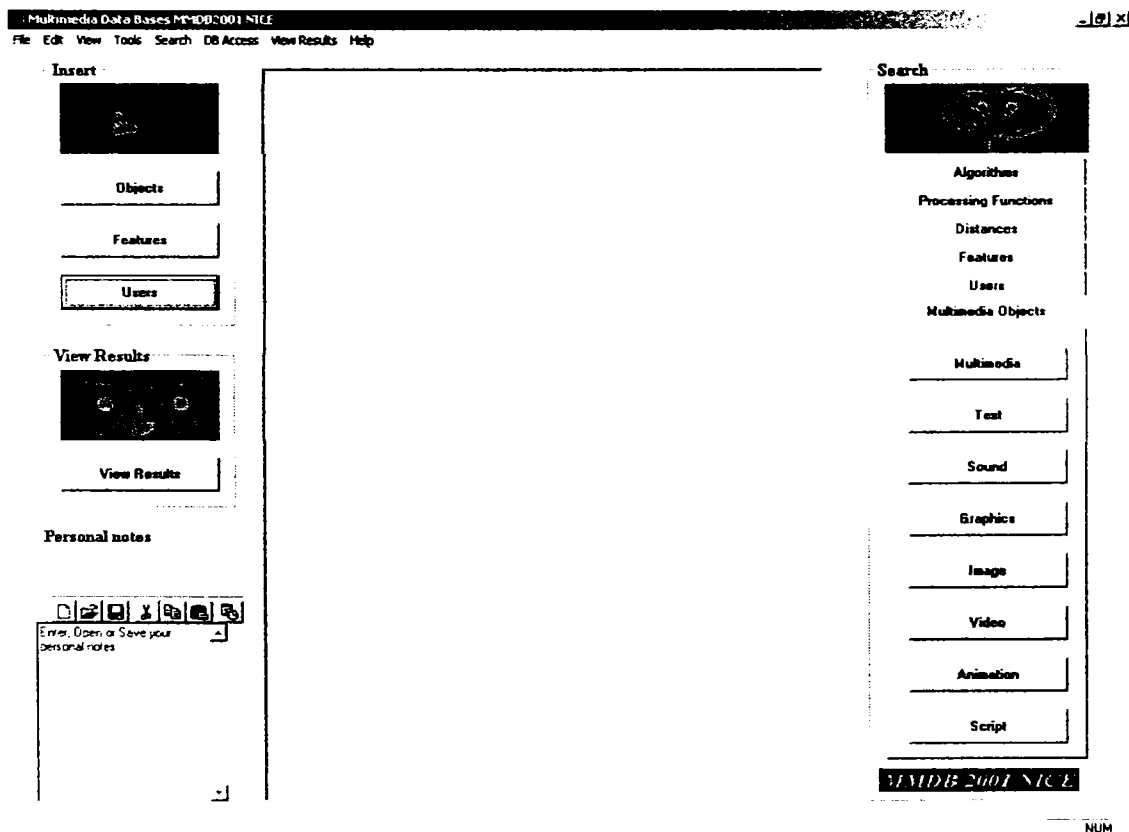


Figura 6-8 Meniul principal al aplicației

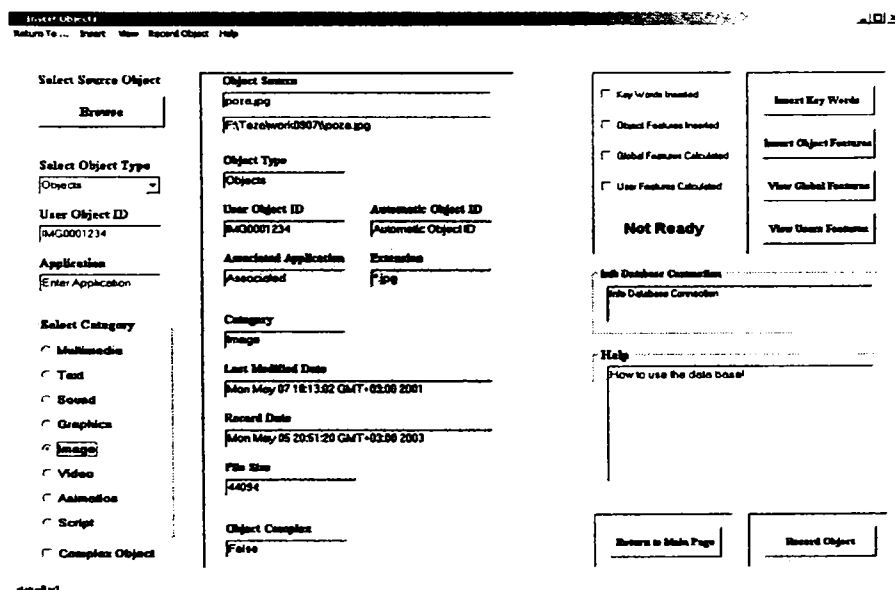
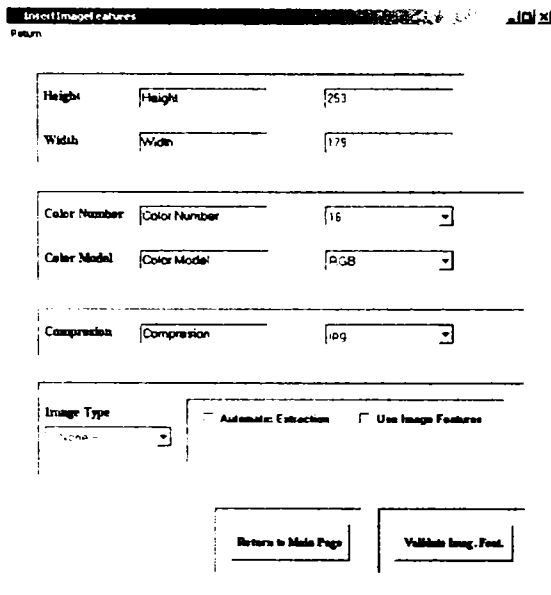
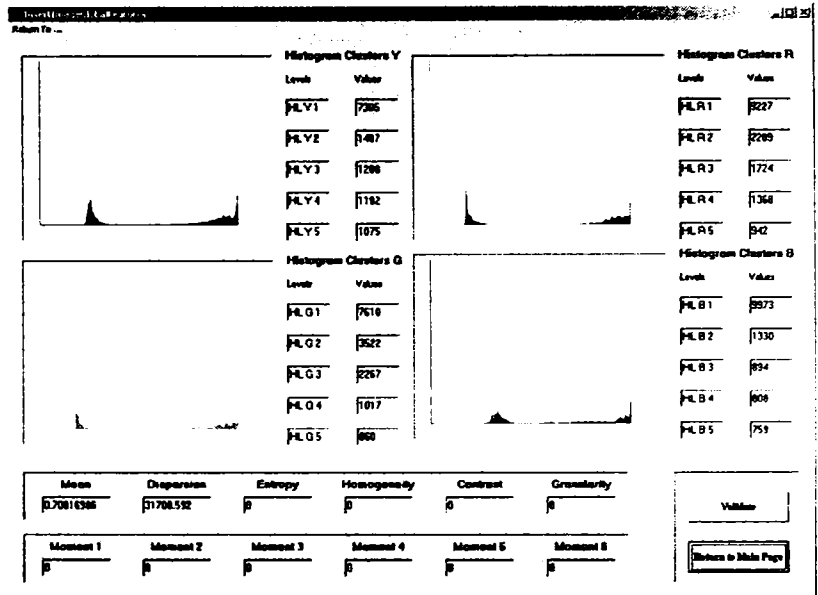


Figura 6-9 Interfața de inserare a unui obiect în baza de date

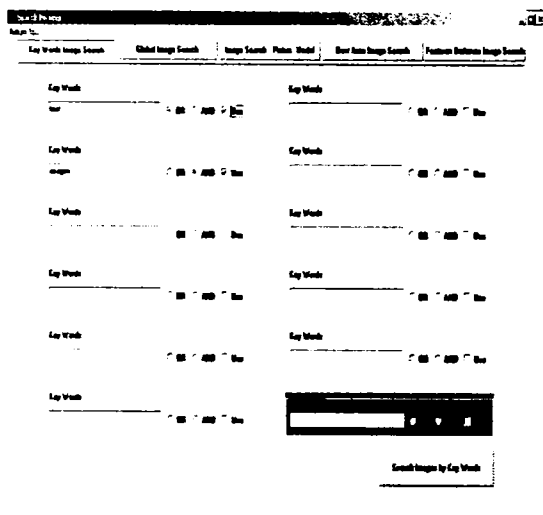


a)

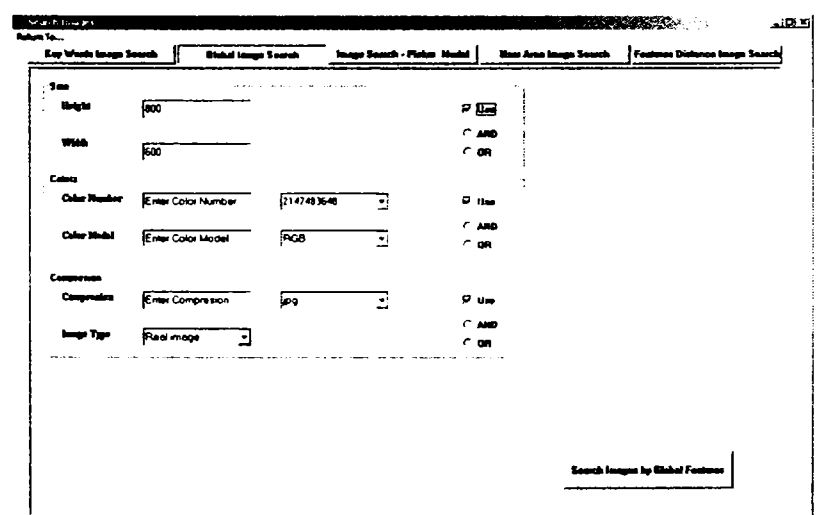


b)

Figura 6-10. a) Extragerea și afișarea caracteristicilor globale ale unui obiect de tip imagine
b) Extragerea și afișarea caracteristicilor unui obiect de tip imagine

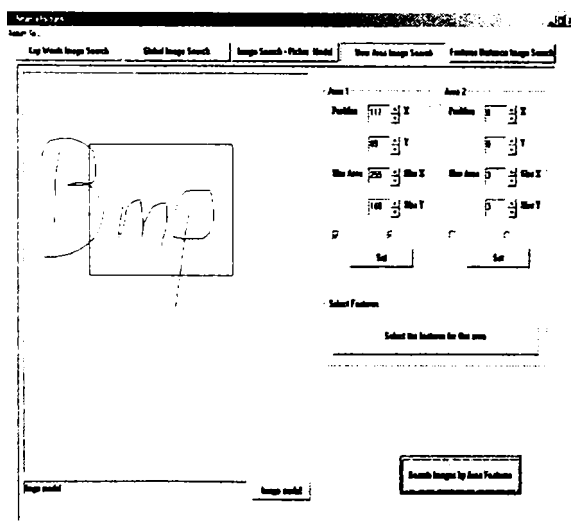


a)

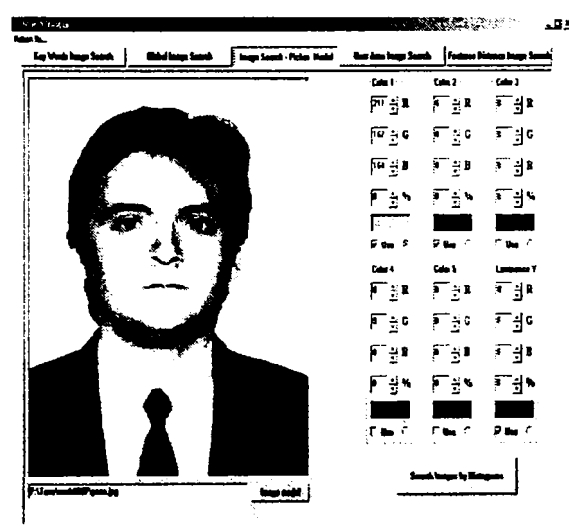


b)

Figura 6-11. a) Căutarea după cuvinte cheie b) căutare după caracteristicile globale ale unei imagini



a)



b)

Figura 6-12 a) Căutarea după caracteristicile unei zone din imagine b) căutare după culori predominante

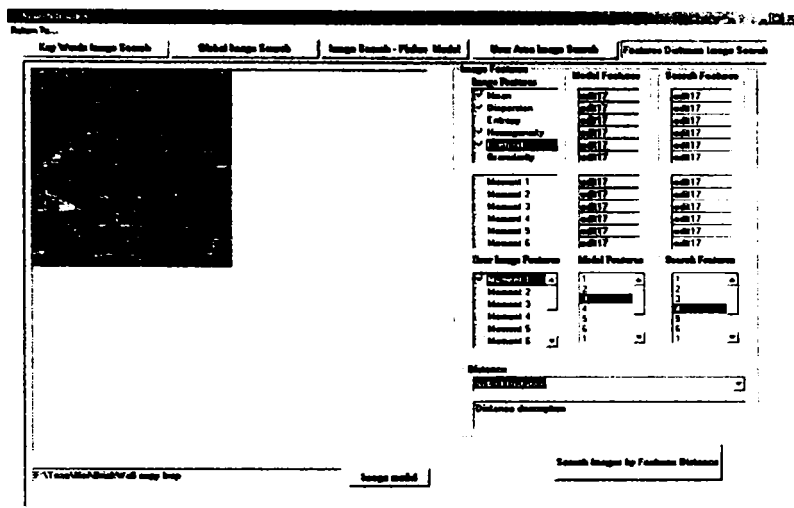


Figura 6-13 Căutarea după caracteristicile imaginii

- **Posibilități de implementare prin Web** - Nucleul aplicației poate fi adaptat cu ușurință pentru a rula aplicația prin Internet prin intermediul unor pagini Web, sunt necesare doar adaptările pentru interfața grafică, modulele aferente operațiilor de indexare, căutare și afișare rezultate rămân aceleași .

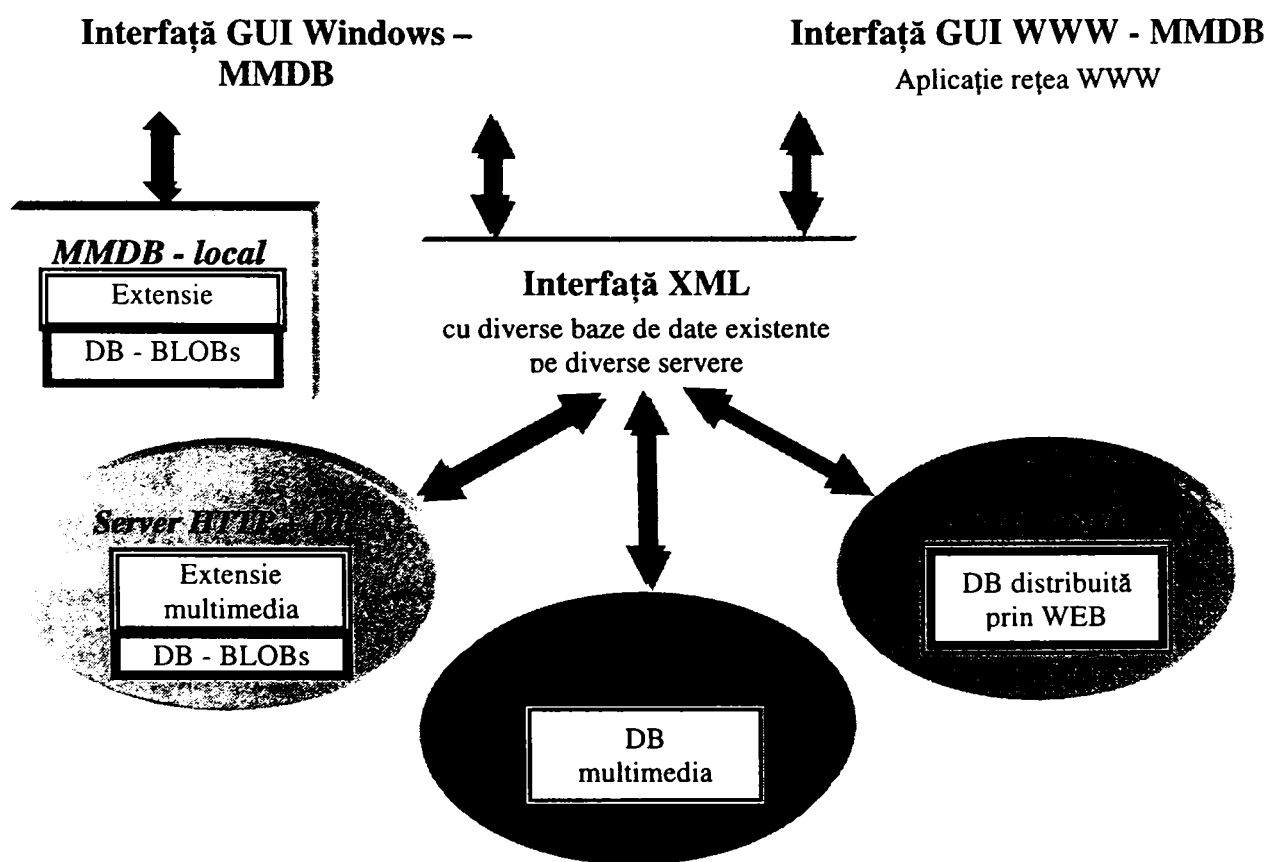


Figura 6-14 Posibilități de implementare a accesării prin Internet și facilități de distribuire

□ Sistem deschis pentru extinderi ulterioare ca bază de date distribuită

Sistemul conceput permite stocarea parțială a unor date în diverse locații pentru a mări viteza de răspuns la operațiile de căutare ale utilizatorilor. De exemplu, putem avea în diverse locații fragmente din baza de date. Un caz concret poate fi redacția unui ziar, ce folosește foarte frecvent, de exemplu, imaginile dintr-o bază de date multimedia. Pot fi stocate parțial informațiile legate de proprietățile semantice ale imaginilor, sau diverse proprietăți ce ne redau conținutul imaginilor, fără a fi necesară stocarea propriu-zisă a imaginilor. Ulterior după determinarea imaginii celei mai apropiate de ceea ce se caută, se va solicita descărcarea acesteia din baza de date globală. Astfel operația de căutare nu va mai depinde de serverul bazei de date, rularea algoritmilor de căutare realizându-se local, din date stocate local. Unui utilizator i se poate permite și căutarea altor tipuri de obiecte multimedia. Asemenea exemple pot fi regăsite în foarte multe domenii de activitate. Am constatat o redirectionare a interesului spre optimizarea aplicațiilor din domeniul bazelor multimedia folosind baze de date distribuite [277], [278], [279].

6.3.6. Detalii referitoare la informațiile stocate în baza de imagini și reprezentarea caracteristicilor descriptive

În urma experimentărilor am constatat marea diversitate de imagini ce există în natură și ceea ce înseamnă similaritatea dintre imagini în sensurile date de factorul uman. În continuare sunt date câteva caracteristici pentru numai 60 de imagini din baza folosită (folosesc acest număr redus de imagini din motive de afișare).

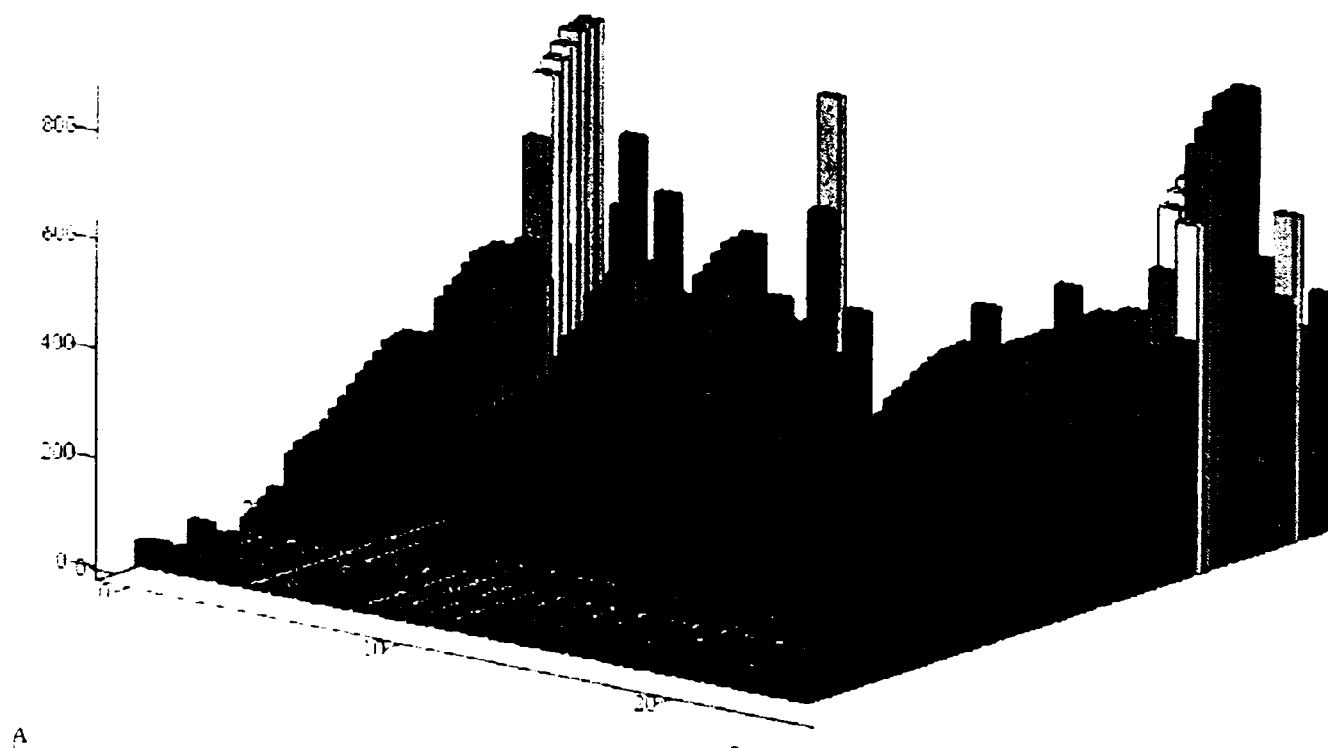


Figura 6-15 Caracteristicile a 60 de imagini pentru o descompunere cu 25 de filtre Gabor

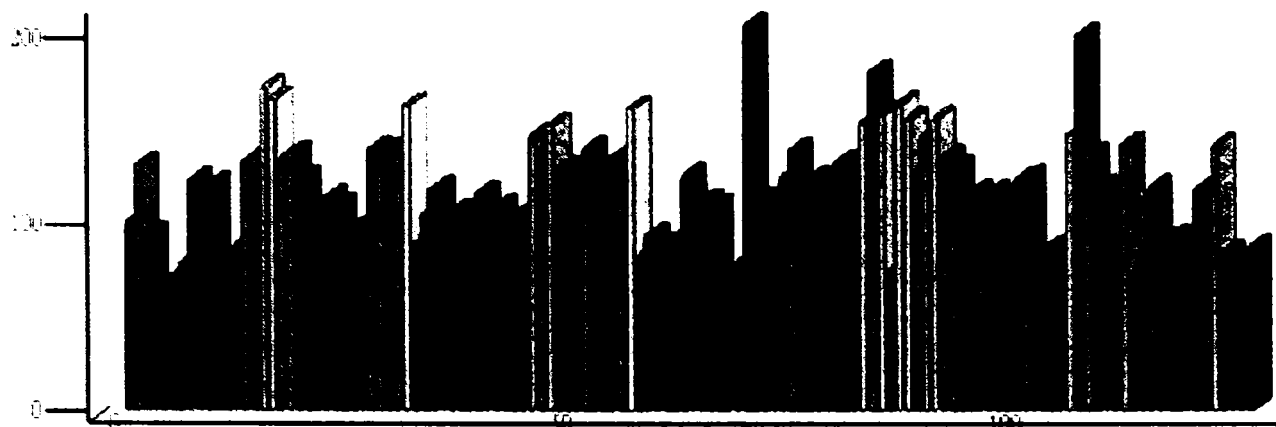


Figura 6-16 Contrastul a 60 de imagini din baza de imagini

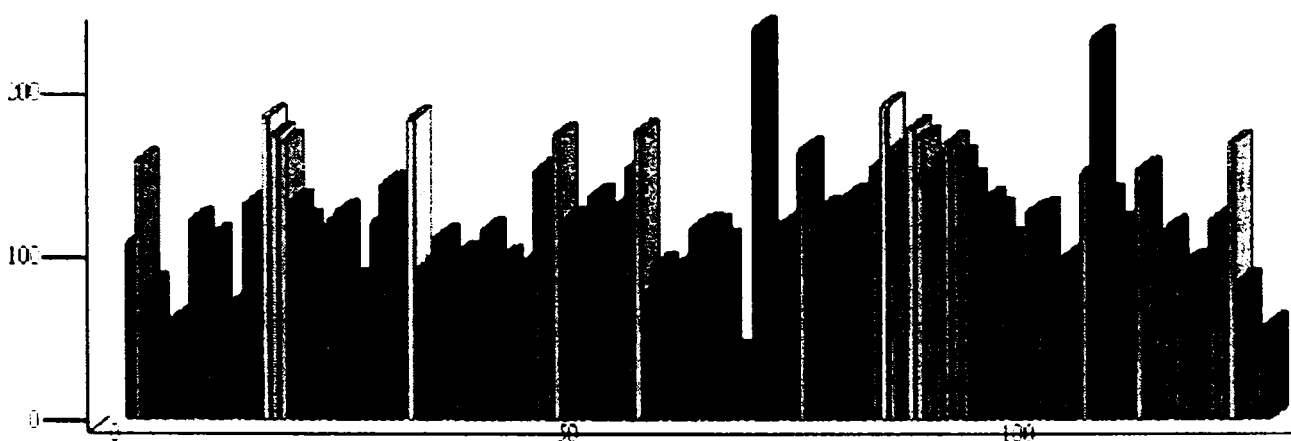


Figura 6-17 Omogenitatea a 60 de imagini din baza de imagini

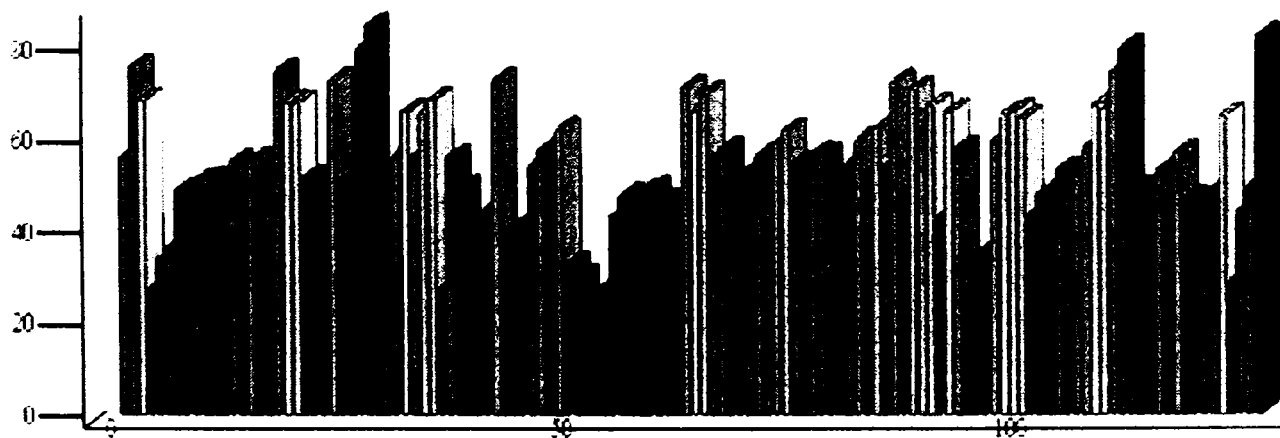


Figura 6-18 Entropia a 60 de imagini

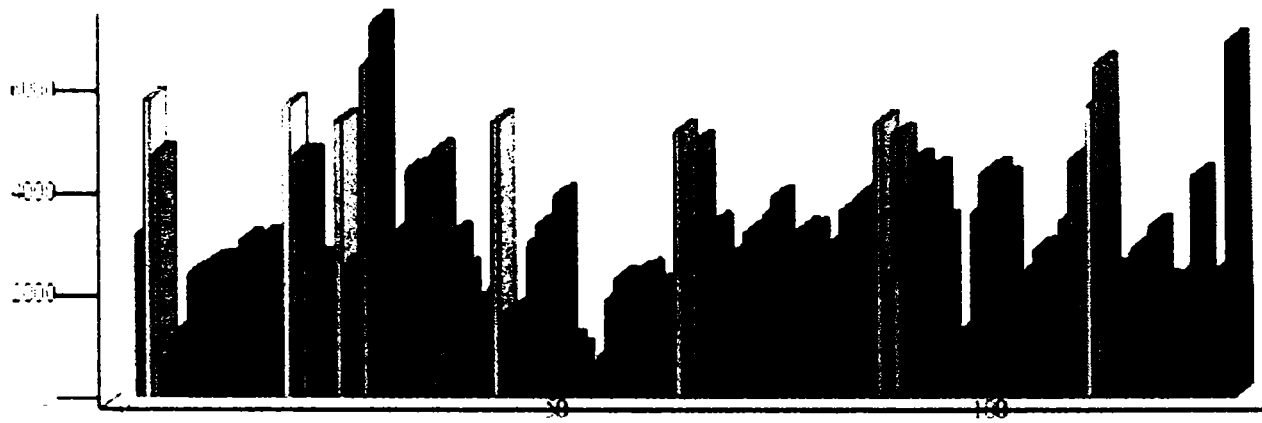


Figura 6-19 Directivitatea a 60 de imagini

Din imagini și din caracteristicile acestora a rezultat o mare diversitate de valori, dar pentru anumite imagini se poate constata o anumită regrupare în jurul unor caracteristici. Regrupări ce se regăsesc în cadrul interpretării umane a imaginilor ca și similitudini între acestea.

Capitolul 7.

Contribuții personale și concluzii

Lucrarea dezvoltă noi modele de aplicații multimedia, noi algoritmi și tehnici multimedia, ce îmbunătățesc activitățile de implementare a produselor multimedia: baze de date multimedia cu căutare bazată pe conținut, aplicații interactive pe CD-ROM, pagini web.

Folosind noul model de aplicație multimedia propus de autor, finalitatea tezei de doctorat asigură implementarea unei baze de date multimedia cu un grad mare de utilitate și flexibilitate în cadrul unui post de televiziune, prin utilizarea de tehnici de indexare și căutare bazate pe conținut.

În lucrarea de față se pot evidenția un număr de 13 contribuții personale, cu caracter de noutate în domeniul aplicațiilor multimedia, prezentate sintetic în continuare, fiind însoțite de referințe bibliografice la articolele publicate de doctorand.

➤ **Studiu comparativ privind mediile de dezvoltare a aplicațiilor multimedia.**

Acest studiu comparativ a permis autorului să definească direcțiile în care se pot aduce îmbunătățiri pentru creșterea performanțelor mediilor de dezvoltare a aplicațiilor multimedia, din punct de vedere software [11], [15], [16], [17], [18], [19].

➤ **Distribuirea programelor de simulare în rețele de calculatoare heterogene.**

Pentru a îmbunătăți timpul de procesare, autorul a propus și implementat algoritmi de prelucrare a imaginilor pe o rețea eterogenă de calculatoare folosind o procesare paralelă. În acest fel, sarcina de prelucrare este distribuită în rețea, fiecare dintre calculatoarele legate în rețea preluând o parte din sarcină; rezultatele parțiale sunt integrate de către calculatoarul ce a inițiat operația de prelucrare.

Rezultatele obținute, folosind o astfel de distribuție a calculelor ce trebuie realizate în funcție de resursele existente, au demonstrat eficacitatea metodei printr-o exploatare eficientă și prin scurtarea timpului de execuție (mai multe calculatoare cu performanțe modeste, conectate în rețea, pot depăși un calculator puternic și foarte scump). Au fost realizate mai multe teste privind eficacitatea metodei în cazul prelucrărilor de imagini. A fost verificat algoritmul pentru prelucrarea simultană pe mai multe calculatoare a mai multor imagini simultan, dar și varianta de prelucrare paralelă a aceleiași imagini pe mai multe calculatoare. Rezultatele obținute au fost publicate de autor [36], [37], [38].

➤ **Un nou model de aplicație multimedia.**

Modelul propus [15] prezintă următoarele avantaje:

- Optimizări din punct de vedere al structurării implementării prin orientarea pe obiecte a aplicațiilor. Prin definirea de noi clase, metode, obiecte și parametri s-a asigurat creșterea performanțelor aplicației ce lucrează cu obiecte multimedia de tip fluxuri de date condiționate temporal.

- Concepția modulară permite extinderi ulterioare ale aplicațiilor prin re folosirea unor părți din aplicație.
- Îmbunătățirea bibliotecilor de funcții și transferabilitatea acestora între aplicații, prin folosirea modelului propus, permite ca utilizatorii să poată contribui la îmbunătățirea performanțelor și la extinderea mediilor de dezvoltare a aplicațiilor multimedia existente.
- Asigurarea unui anumit grad de calitate pentru serviciile oferite de aplicație și informarea utilizatorului din timp asupra problemelor ce pot apare, existând chiar și posibilitatea sugerării unor soluții de remediere. Modelul propus permite folosirea unor biblioteci de funcții care să verifice înainte de rularea unei aplicații multimedia performanțele hardware ale sistemului.
- Includerea bibliotecilor de funcții și metode ce permit comunicarea cu bazele de date. Acest lucru face posibilă dezvoltarea aplicațiilor multimedia cu accesarea obiectelor direct din baze de date. Dezvoltarea unei aplicații multimedia se poate face prin introducerea în baza de date de noi obiecte, ne fiind necesară refacerea aplicației, rescrieri de cod sau alte complicații de acest fel. Modelul permite dezvoltarea de aplicații multimedia la care numărul de pagini este generat dinamic prin intermediul bazelor de date și nu este implementat în mod fix.
- Realizarea de aplicații care să lucreze în colaborare cu mai utilizatori simultan prin tehnologia cliet/server, inclusă în cadrul modelului.

➤ **Studiu comparativ privind evoluția bazelor de date multimedia.**

Acest studiu comparativ a avut la bază evoluția bazelor de date și se concentrează asupra caracteristicilor bazelor de date multimedia orientate pe obiecte, ce acceptă obiecte multimedia de tip flux de date cu constrângeri temporale. Autorul a prezentat comparativ mai multe proiecte ce au dezvoltat baze de date multimedia, precum și situația celor mai cunoscute produse comerciale [280].

➤ **Caracteristici ale imaginilor folosite pentru indexarea bazelor de imagini.**

În urma studiului și a simulărilor realizate de autor [120], [160], [161], [183] sunt evidențiate cele mai eficiente categorii de caracteristici folosite în indexarea bazelor de imagini, cu avantaje și dezavantaje. Principalele caracteristici recomandate sunt: proprietățile stohastice, momentele de diverse ordine, proprietățile matricii de coocurență, isosegmentele, analiza spectrală a imaginii (filtre Gabor), contrastul asociat pe regiuni și atribute de suprafață.

➤ **Extinderea metodei de modelare autoregresive cunoscută pentru generarea anumitor tipuri de texturi**

Autorul prezintă în [177] o extindere a metodelor de modelare autoregresive cunoscute [180], [181] pentru generarea texturilor. Metoda oferă un grad mare de aplicabilitate în aplicațiile multimedia, prin faptul ca pornind de la un număr foarte redus de parametri se pot genera foarte multe tipuri de texturi, spațiul alocat stocării parametrilor fiind redus. Metoda de generare a texturilor asigură un volum rezonabil de calcule.

➤ **Implementarea unor algoritmi de procesare a imaginilor ce folosesc ca element central caracteristicile texturilor.**

Sunt prezentate următoarele posibilități de procesare a texturilor în strânsă legătură cu modul lor de utilizare în cadrul unei baze de date multimedia.

- Generarea și stocarea texturilor folosind metode structurale și metode de generare bazate pe modele autoregresive.
- Rezultate practice bune privind prelucrarea și caracterizarea imaginilor prin intermediul texturilor s-au obținut prin folosirea filtrelor Gabor [38], [160], [161], [195]. În acest scop s-a prezentat modul de proiectare a unui filtru Gabor, evidențiindu-se problemele ce apar la

definirea acestuia și în special la anticiparea eficienței acestuia pentru o problemă concretă care trebuie să fie rezolvată.

- Întrucât în imaginile reale, texturile se află într-o mare varietate de orientări și la diverse rezoluții, am avut în vedere problema evaluării orientării, translației și a scalării unei texturi. În situațiile în care este necesară în detectarea texturii o anumită invarianță la rotație, translație și scalare recomand operatorii Circular-Mellin, iar atunci când este necesar un grad mare de sensibilitate la orientare recomand folosirea filtrelor Gabor și a caracteristicilor matricii de coocurență.
 - Din rezultatele privind evaluarea gradientului unei texturi au fost publicate în [183], autorul a găsit legături de similaritate între rezultate obținute prin metode de descompunere folosind un set de filtre Gabor și folosirea proprietăților matricii de coocurență calculată pe ferestre alunecătoare de diverse dimensiuni.
 - Calcul de caracteristici folosind morfologia matematică împreună cu metode din domeniul frecvență sau cu caracteristici ale matricii de coocurență [161].
- **Algoritmi de indexare și căutare în bazele de imagini prin combinare de mai multe tehnici de procesare.**
- Clasificarea unui număr mare de caracteristici este o problemă greu de rezolvat cu ajutorul unui clasificator bazat pe o metrică, într-un spațiu cu un număr mare de dimensiuni. Autorul prezintă un algoritm de indexare și căutare ce folosește caracteristicile un număr mare de caracteristici ale imaginii și două variante de clasificator bazat pe rețele neuronale [201].
 - Algoritm de căutare bazat pe o structura arborescentă prezentat de autor în [281] oferă o reducere considerabilă a timpului de căutare. Reducerea timpului de căutare se bazează pe organizarea modului de parcurgere și analiză a caracteristicilor. Autorul propune modul de alegere a caracteristicilor dominante dintr-un număr de 5 categorii de caracteristici, iar respectarea acestor dominante va determina parcurgerea în continuare a arborelui sau părăsirea unei ramuri. Algoritm se dovedește a fiind rapid, procentul volumului de calcule folosind algoritmul propus din volumul total dacă nu s-ar folosi acest algoritm (parcurgerea întregii baze de date) este de 22,05%. Rezultatele căutării sunt caracterizate de o apropiere strictă față de imaginea de interogare, similaritatea dintre imaginea de interogare și rezultate fiind puternică.
 - Combinarea unui număr mare de caracteristici (caracteristicile matricii de coocurență, descompunerea multicanal cu ajutorul filtrelor Gabor, caracteristici stohastice ale imaginii, histogramele) pentru a realiza un algoritm puternic de indexare și căutare este realizată de autor prin folosirea unui clasificator bazat pe asignarea unor markeri caracteristicilor dominante din imagine. Asignarea markerilor se realizează pe baza pe un algoritm cu praguri, asigurându-se o viteză considerabilă în procesul de clasificare [203]. Rezultatele algoritmului sunt caracterizate de o bună similaritate în sens larg. Pot fi regăsite imagini cu medii diferite dar caracterizate de o anumită orientare a texturii, generalizându-se conceptul de similaritate.

➤ **Metodă de segmentare automată și indexare a secvențelor video.**

Metoda propusă de autor se bazează pe descrierea semantică și temporală a unei emisiuni ce se află într-o secvență video [213]. Sunt evidențiate avantajele descrierii semantice și temporale ale emisiunii în raport cu metodele clasice de indexarea a secvențelor video. Metoda propusă reduce considerabil numărul de false detecții prin supervizarea metodei de indexare cu detecția schimbărilor din planul imagine cu un model spațial și temporal de descriere a emisiunii.

➤ Soluții de optimizarea activităților de producție în studiourile de televiziune.

Soluțiile de optimizare propuse au rezultat din contactul direct cu mai multe studiouri de televiziune locale și naționale și se bazează pe introducerea tehnicilor multimedia în televiziune. Soluțiile propuse vizează modul de organizare al circuitului informațional, arhivarea materialelor folosind metode avansate de compresie [213], [256], [257] și o bază de date multimedia cu un sistem complex de indexare și variante multiple de căutare [263].

➤ Proiectarea și implementarea bazei de date multimedia

- Proiectarea bazei de date este astfel concepută încât să existe și posibilitatea unei distribuiri optime prin Internet. Baza de imagini poate fi stocată într-un singur loc, iar variantele satelit ale acestora ce conțin doar caracteristicile imaginilor pot fi distribuite în mai multe locații prin Internet. Rezultatele căutării fiind afișate din baza de imagini principală. Soluțiile folosite de autor permit o bună re folosire a liniilor de cod, o bună portabilitate prin implementarea în Java a aplicației și accesare multiplă folosind o tehnologie client/server.
- Obiectele multimedia sunt descrise printr-o formă structurată ce permite căutarea inclusiv în interiorul obiectelor multimedia complexe, adică la nivel de subiect.
- Algoritmii propuși de autor pentru indexare și căutare combină varianta de căutare bazată pe o descriere semantică a imaginilor cu cea bazată pe conținutul acestora, sporind puterea de căutare. Sunt permise personalizări ale modului de extragere a caracteristicilor și a clasificatorilor. Noi metode de procesare a imaginilor, de extragere a caracteristicilor și de clasificare pot fi introduse de utilizatori, acestea fiind stocate în cadrul bazei de date ca și funcții ce pot fi apelate ulterior în procesele aferente lucrului cu baza de date.

Direcții viitoare de cercetare

- Direcțiile viitoare de cercetare includ dezvoltarea de noi modalități de interogare a utilizatorului, într-un limbaj cât mai apropiat de cel natural, folosit în vorbirea de zi cu zi [282].
- Se vor dezvolta metode de grupare după caracteristici care să înglobeze descrieri semantice ale căutării. Au fost începute astfel de metode, dar fără rezultate spectaculoase până în acest moment [283].
- În cazul secvențelor video, pe lângă detecția schimbării scenei [284] sau a obiectelor din scenă [285] se poate extinde domeniul de căutare la similaritatea de mișcare ce există într-o anumită secvență video [286].
- Prezentările multimedia compacte vor fi descompuse și stocate în baze de date dedicate.
- Pentru secvențele video ce conțin prezentări ale anumitor subiecte se poate încerca și descompunerea în obiecte multimedia ce vor fi stocate separat în baza de date multimedia pentru reutilizări ulterioare [287].

Bibliografie

-
- [1] Bhaskaran V., Konstantinides K. – *Image and Video Compression Standards: Algorithms and Architectures*, Ed. Kluwer Academic Publishers, London, 1997
 - [2] Schmeiser L. - *The Complete Website Upgrade and Maintenance Guide*, Ed. Sybex, San Francisco, 1999.
 - [3] Flanagan D. – *The Definitive Guide JavaScript*, Ed. O'Reilly, Cambridge, 1998.
 - [4] NetObject Fusion Press – *Building Business Web Sites*, RedWood, California, 2001.
 - [5] Official site of NetObject Fusion – <http://www.netobjects.com>
 - [6] Macromedia Press – *Using Macromedia Flash 5*, San Francisco, 2001
 - [7] Hustedde S. F. – *Developing with Asymetrix ToolBook*, Ed. Wadworth Publishing Company, Belmont, California, USA, 1996.
 - [8] Official site of Asymetrix ToolBook – <http://www.asymetrix.com>.
 - [9] Gross P. – *Macromedia Director and Lingo*, Ed. Macromedia Press, 1999.
 - [10] Official site Macromedia – <http://www.macromedia.com>
 - [11] Mocofan M. – *Dezvoltarea aplicațiilor multimedia folosind Macromedia Director – Ghid practic*, Program TEMPUS DRUM, 70 pagini, 1997.
 - [12] Official site Ulead – <http://www.ulead.com>
 - [13] Ulead Press – *MediaStudii Pro – Five in One*, Torrance, California, 2001.
 - [14] Cutler R. Candan K. – *Multimedia Authoring Systems*, Multimedia Database System, pp. 289-2296, Ed. Springer, Berlin 1996.
 - [15] Mocofan M., Toma C. I. – *New Interactive Multimedia Application Model Using Objects Storage in a Database*, Buletinul Științific al Universității “Politehnica” din Timișoara, Tom 46 (60), Fascicola 1, pp. 31-36, 2001.
 - [16] Mocofan M., Pescaru D. – *A parallel implementation of the Gabor image segmentation algorithm using textures*, Proceedings of International Symposium on System Theory, Automation, Robotics, Computers, Informatics, Electronics and Instrumentation, “SINTES-10” X-th edition, 25-26 May 2000, Craiova, Romania, pp. A201-204, 2000.
 - [17] Mocofan M., Alexa F., VasIU R. – *A New Prediction Model for Fast Block Motion Estimation*, Analele Facultății de Inginerie din Hunedoara, Tomul I, Fascicola 2, Proceedings of the International Symposium, 4-5 Nov. 1999, pp. 29-34, 1999.
 - [18] Mocofan M., VasIU R. – *Algorithm for Region-Based Image Compression Using Mathematical Morphology Tools*, Buletinul Institutului Politehnic din Iași, Tomul XLV (IL), Fasc. 5A, Proceedings of the International Conference on Electrical and Power Engineering, EPE'99, Iași, 4-6 Nov. 1999, pp. 114-117, 1999.
 - [19] Mocofan M., VasIU R. Gui V. – *Algoritm de codare a imaginilor folosind elemente de morfologie matematică și segmentare bazată pe texturi*, A doua Conferință Internațională de Sisteme Electromagnetice, SIELMEC'99, Chișinău, 8-9 Oct. 1999, pp. 39-42, 1999.

-
- [20] Rumbaugh J., Blaha M., Eddy F. – *Modélisation et conception orientées objet*, Ed. Masson, Paris, 1997.
- [21] Bouzeghoub M., Gardarin G., Valduriez P – *Les Objects*, Ed. Eyrolles, Paris, 1998
- [22] Mirecort A. – *Le développeur Java.2*, Ed. Eyrolles 1999;
- [23] Flamengan D. – *Java in a Nutshell*, Ed.O'Reilly 1999.
- [24] *Java Tutorial* – <http://java.sun.com/docs/books/tutorial>.
- [25] Chat M. C., Griffith S. W. – *Java 1001 secrete pentru programatori*, Ed. Jamasa Press, 2000;
- [26] Arnold K., Gosling J. – *The Java Programming Language Second Edition*, Ed. Addison Wesley, Decembrie 1997;
- [27] Nicolas C., Avare C. – *Java Client – Serveur*, Ed. Eyrolles, 1997;
- [28] Niemeyer P., Peck J. – *Java par la pratique*, Ed.O'Reilly 1996;
- [29] Pescaru D., Laire P. – *Modifiers in OFL: An Approach for Access Control Customization*, The 9th International Conferences on Object-Oriented Information Systems - OOIS'03, Geneva, Swizerland, september 2003.
- [30] Kiczales G., Hilsdale E., Hugunin J., Kersten M., Palm J., Griswold G. W. – *Getting Started with AspectJ*, Raport of Defence Advanced Research Projects Agency F30602-C-0246, 2001
- [31] Official site of AspectJ – <http://www.aspectj.org>
- [32] Kiczales G., Lamping J., Mendhekar A., Maeda C., Lopes V. C., Loingtier J., Irvin J. – *Aspect Oriented Programming*, Proceedings o European Conference on Object Oriented Programming ECOOP, Finland, Ed. Springer, 1997.
- [33] Cappello P, Christiansen B.O, Ionescu M.F, Neary M.O, Schauser K.E, and Wu D. – *Javelin: Internet-based Parallel Computing Using Java*, ACM Workshop on Java for Science and Engineering Computation, June, 1997.
- [34] Ferrari A. J. – *JPVM: Network Parallel Computing in Java*, Technical report CS-97-29, Department of Computer Science, University of Virginia, Charlottesville, VA 22903, USA, 1997.
- [35] Geist A, Beguelin A, Dongarra J, Jiang W, Manchek R, and Sunderam V.S. - *PVM: Parallel Virtual Machine*, MIT Press, 1994.
- [36] Thurman D. – *JavaPVM*, available from: <http://www.isye.gatech.edu/chmsr/JavaPVM/>, 1999.
- [37] Mocofan M., Pescaru D. – *A parallel implementation of a algorithm for content-based image indexing and retrieval using texture featres*, Proceedings of International Conference “Communications 2000”, 7 - 9 december 2000, Bucharest, Romania, pp. 85-88, 2000.
- [38] Mocofan M., Pescaru D. – *Parallelisation of a textured colour image segmentation alghorithm*, Transactions on Automatic Control and Computer Science, Fourth International Conference on Technical Informatics - “CONTI 2000”, 12-13 October, Timișoara, Romania, pp 139 – 144, 2000.
- [39] Mocofan M. Pescaru D. – *A parallel implementation of a algorithm for image segmentation using textures*, Proceedings of International Conference “Communications 2000”, 7 - 9 december 2000, Bucharest, Romania, pp. 81-84, 2000.
- [40] Ghinea G., Thomas J. P. – *QoS Impact on User Perception and Understanding of Multimedia Clips*, Proceeding of The 6-th ACM International Multimedia Conference, Bristol, England, 12-16 September, pp. 49-54, 1998.
- [41] Mocofan M., Pescaru D. – *Aplicații interactive multimedia. Programarea OpenScript*, Ed. Politehnica, Timișoara, 2001.
- [42] Steinmetz R., Engler C. – *Human Perception of Media Synchronization*, IBM European Networking Center Raport, 1993.
-

-
- [43] Watson A., Sasse A. – *Measuring Perceived Quality of Speech and Video in Multimedia Conferencing Applications*, Proceeding of The 6-th ACM International Multimedia Conference, Bristol, England, 12-16 September, pp. 55-60, 1998.
- [44] Codd E. F. – *A relational model for large shared data banks*, Communication of ACM 13, pg. 377-387, 1970.
- [45] Simon A. R. – *Strategic Database Technology: Management for the year 2000*, Morgan Kaufmann Publishers Inc., San Francisco, 1996.
- [46] Chen P. P. – *The entity- relationship model –Towards a unified view of data*, ACM Transactions on Databases Systems 1 - 1, 1976.
- [47] Shipman J. – *The functional data model and the data language DAPLEX*, ACM Transactions on Databases Systems 6 - 1, 1981.
- [48] Stonebraker M., Rowe L., Hiohama M. – *The implementation of POSTGRES*, IEEE Transactions on Knowledge and Data Engineering 2, 1 mars 1990.
- [49] Khoshafian S. – *Intelligent SQL*, Computer standards and Interfaces 12, X3/SPARC/DBSG/OODB Task Group Workshop, Atlantic City, 1991.
- [50] Bancilhon F. – *VERSO: A relational back end data base machine*, Proceedings of the second international workshop on database machines, 1983.
- [51] Schek H., Scholl M. H. – *The relational model with relational valued attributes*, Information Systems No 11, pg. 9-21, 1986.
- [52] Khoshafian S. – *A persistent complex object databases language*, Knowledge and Data Engineering, 1989.
- [53] Khoshafian S. – *A calculus for complex object*, Journal of Computers and System Sciences, No 38, 1989.
- [54] Khoshafian S., Copeland G. P. – *Object identity*, Readings in object-oriented databases systems, Ed. Morgan Kaufmann, San Francisco, 1990
- [55] Bancilhon F., Brigs T., Khoshafian S., Valduriez P. – *FAD, a powerful and simple database language*, Proceedings of the thirteenth international conference on VLDB, Brightonm Englandm 1987.
- [56] Show G., Zdonik B. – *A query algebra object-oriented databases*, TR-CS-89-19, Departament of Computer Science, Brown University, 1989.
- [57] Straube D., Ozsu M. T. – *Queries and query processing in object-oriented database system*, ACM Transactions on Information Systems, No 8, 1990.
- [58] Atkinson M. P., Bancilhon F., DeWitt D., Dittrich K., Maier D., Zdonik S. – *Object-Oriented Databases System Manifesto*, Proceedings 1st International Conference Deductive and Object-Oriented Databases, Kyoto, Japan, pp. 40-57, 1989.
- [59] Sullow K. – *MultiMedia Forum: an Interactive Online Journal*, Proceedings of the international Conference on Electronic Publishing, Document Manipulation, and Typography, EP 94, Darmstadt, Germany, John Wiley & Sons LTD, 1994.
- [60] WebCT Official Site – <http://www.webct.com>
- [61] WebBoard Official Site – <http://www.webboard.com>
- [62] Lotus Official Site – <http://www.lotus.com>
- [63] Mocofan M. – *Producție Audio Video*, Curs ODL, Universitatea Politehnica Timișoara, <http://odl.opendrum.utt.ro:8900>
- [64] Mocofan M. – *Echipamente de studio*, Curs ODL, Universitatea Politehnica Timișoara, <http://odl.opendrum.utt.ro:8900>
-

-
- [65] Mocofan M., Andone D. – *Study Skills Online*, Centru pentru Tehnologii Educaționale, Universitatea Politehnica Timișoara,
<http://odl.opendrum.utt.ro:8900/public/StudySkills>
- [66] Deegener M., Grobe G., John W., Kuhnappel B., Lohr M., Wirth H. – *Rapid Prototyping with M μ SE*, International Symposium on Automotive Technology and Automation, Dedicated Conference on Mechatronics, 1994.
- [67] Aberer K., Klas W. – *Supporting Temporal Multimedia Operation on Object-Oriented Databases Systems*, Proceedings of the IEEE International Conference on Multimedia Computing and Systems, Boston, USA, May 1994.
- [68] GMD – *VODAK V4.0 User Manual*, GMD Technical Report No 910, Sankt Augustin, Apr. 1995.
- [69] Niblack W., Barber R., Equitz W., Flickner M., Glasman E., Petrovik D., Yanker P., Faloutsos C., Taubin G. – *The QBIC Project: Querying Images By Content Using Color, Texture and Shape*, SPIE 1993 Int. Symposium on Electronic Imaging: Science and Technology, Conf. 1908, Storage and Retrieval for Image and Video Databases, Feb. 1993.
- [70] Christodoulakis S., Theodoridou M., Ho F., Papa M., Pathria A. – *Multimedia Document Presentation, Information Extraction, and Document Formation in MINOS: A Model and a System*, ACM Transactions on Office Information System, No. 4, pp. 345-383, 1996.
- [71] Loney K. - *Oracle - DBA - Handbook*, Ed. McGraw, London, 1994.
- [72] Stonebraker M. – *Object-Relational DBMSs: The Next Great Wave*, San Francisco, Ed. Morgan Kaufmann Publisher, 1996.
- [73] Stonebraker M., Rowe L., Lindsay B., Gray P., Carie Brodie M. L., Berstein P., Beech D., - *The third generation database system manifesto*, Proceedings of International Conference ACM SIGMOD, 1990.
- [74] Darwen H., Date C.J. - *Foundations for Object/Relational Databases: The third manifesto*, Ed. Addison Wesley, Harlow, 1998.
- [75] Egenhofer M. J. – *Extending SQL for graphical display*, Cartography and Geographic Information System, 1991.
- [76] Krishnan D., Kunii T. L. – *A visual database interface for CAD applications*, Visual database system, Ed. IFIP, Amsterdam, 1992.
- [77] Wu J. K., Nrasimhalu A. D. – *Identifying faces using multiple retrievals*, IEEE Multimedia, No. 1, 1994.
- [78] Gupta E. A., Weymouth T. E., Jain R – *Semantic queries with picture: The VIMSYS model*, Proceedings of the seventeenth international conference on VLDB, Barcelonam Spain, 1991.
- [79] Rabitti F., Stancev P. – *GRIM-DBMS: A graphical image database management system*, Visual DataBase Systems, Ed. Elsevier, New York, 1989 .
- [80] Hua K. A., Vu K., Oh J. H. - *SamMatch: A Flexible and Efficient Sampling-Based Image Retrieval Technique for Large Image Databases*, Proceeding of The 7-th ACM International Multimedia Conference, Orlando, SUA, 30 october - 5 November, pp. 225-234, 1999.
- [81] Aslandogan Y. A., Yu T. C. - *Multiple evidence combination in image retrieval: Diogenes searches for people on the WEB*, SIGIR 2000, Proceedings of the 23 rd Annual International ACM SIGIR Conference on Research an Development in Information Retrieval, Athens, Greece, pp. 88-95, 24-28 July 2000.
- [82] Nastar C., Mitschke M., Meilhac C., Boujemaa N. - *Surfimage: a Flexible Content-Based Image Retrieval System*, Proceeding of The 6-th ACM International Multimedia Conference, Bristol, England, 12-16 September, pp. 339-344, 1998.
-

-
- [83] Dimitrova N., Chen Y., Nikolovska L. - *Visual Associations in DejaVideo*, ACCV 2000 - Proceedings of the Fourth Asian Conference on Computer Vision, Taipei, Taiwan, pp. 353-361, 8-11 January 2000.
- [84] Porkaew K., Chakrabarti K. - *Query Refinement for Multimedia Similarity Retrieval in MARS*, Proceeding of The 7-th ACM International Multimedia Conference, Orlando, SUA, 30 october - 5 November, pp. 235-242, 1999.
- [85] Hua K. A., Cai Y., Sheu S. - *Patching: A Multicast Technique for True Video-on-Demand Services*, Proceeding of The 6-th ACM International Multimedia Conference, Bristol, England, 12-16 September, pp. 191-200, 1998.
- [86] Griwodz C., Merkel O., Dittmann J., Steinmetz R. - *Protecting VoD The Easier Way*, Proceeding of The 6-th ACM International Multimedia Conference, Bristol, England, 12-16 September, pp. 21-28, 1998.
- [87] Subrahmanian V. S., Jajodia S. - *Multimedia Database System*, Ed. Springer, Berlin 1996.
- [88] Marcus S., Subrahmanian V. S. - *Foundations of MMDS Systems*, Multimedia Database System, pp. 3-31, Ed. Springer, Berlin 1996.
- [89] Bohm K., Rakow T. C. - *Metadata for Multimedia Documents*, SIGMOD Record, Vol 23, No. 4, pp. 21-26, dec. 1994.
- [90] Jain R., Hampapur A. - *Metadata in Video Databases*, SIGMOD Record, Vol 23, No. 4, pp. 27-33, dec. 1994.
- [91] Khoshafian S. - *Object-Oriented databases*, Ed. John Wiley and Sons, New York, 1993.
- [92] Khoshafian S., Baker B. - *Multimedia and Imaging Databases*, Ed. Morgan Kaufmann Publisher, San Francisco, California, 1996.
- [93] Little T. D. C., Ghafoor A. - Network considerations for distributed multimedia objects composition and communication, IEEE Network Magazine No. 4, pp 32-49, 1990
- [94] Date C. J. - *Twelve rules for a distributed database*, Computer World, No. 21, pp. 75-81, 1987
- [95] O'Neil P., O'Neil E. - *Database: Principles, Programming, and Performance*, Ed. Kluwer Academic Publishers, London, 2001.
- [96] Steinmetz R., Nahrstedt K. - *Multimedia: Computing, Communications & Applications*, Ed. Prentice Hall, London, 1995.
- [97] Abiteboul S. Hull R., Vianu V. - *Fondements des Bases de Données*, Ed. Vuibert, Paris 2000.
- [98] Bayer R., McCreight E. - *Organization and maintenance of large ordered indexes*, Acta Informatica, No. 1, pp. 173-189, 1972.
- [99] Held G., Stonebraker M. - *B-Trees re-examined*, Communications on The ACM No. 21, pp. 139-143, 1978.
- [100] Bayer R., Unterauer K. - *Prefix B Trees*, ACM Transaction on Database Systems, No. 2, pp. 11-26, 1977.
- [101] Khoshafian S., Baker B. - *Multimedia and Imaging Databases*, Ed. Morgan Kaufmann Publisher, San Francisco, California, pp. 417-421, 1996.
- [102] Finkel R. A., Bentley J. L. - *Quad trees, a data structure for retrieval on composite keys*, Acta informatica, No. 4, pp. 1-9, 1974.
- [103] Bentley J. L. - *Multidimensional binary search trees used for associative searching*, Communications of the ACM, No.18, 1975.
- [104] Robinson V. B. - *The K-D-B Tree: A search structure for large multidimensional dynamic indexes*, VLDB, 1981
- [105] Scheuermann P., Ouksel M. - *Multidimensional B-Trees for associative searching in database system*, Inform System No. 7, pp. 123-137, 1982
-

-
- [106] Guttman A. – *R-trees: A dynamic index structure for spatial searching*, Proceedings ACM SIGMOD International Conference on Management of Data, 1984.
- [107] Liou J. H., Yao S. B. – *Multidimensional clustering for database organizations*, Inform System No. 2, pp. 187-198, 1982
- [108] Lee D. T., Wong C. D. – *Quintary trees: A file structure for multidimensional database system*, ACM Transaction on Database Systems, No. 5, pp. 339-353, 1980.
- [109] Orenstein J. A. – *Multidimensional tries used for associative searching*, Information Processing Letters, No. 14, pp. 150-157, 1982.
- [110] Rohnie J. B., Lozano T. – *Attribute based file organization in a paged memory enviroment*, Communications of the ACM, No.17, pp. 63-69, 1974.
- [111] Lloyd J. W., Romamohanarao K. – *Partial-match retrieval for dynamic files*, BITT 22, pp. 150-168, 1982.
- [112] Otto E. J. – *A mapping functions for the directory of a multidimensional extendible hashing*, Proceedings of the Tenth International Conference on Very Large Data Bases, 1984.
- [113] Faloutsos C. M. – *Signature files*, Information retrieval: Data Structures and Algoritms, Ed. Prentice-Hall, New York, 1992.
- [114] Khoshafian S., Chan A., Wong A., Wong H. K. – *A guide to developing client/server SQL application*, Ed. Morgan Kaufmann, San Francisco, 1992.
- [115] Witten I. H., Moffat A., Bell T. C. – *Managing Gigabytes: Compressing and Indexing Documents and Images*, Ed. Morgan Kaufmann Publishers, London, 1999.
- [116] Salton G. – *Automatic Text Processing – The Transformation, Analysis and Retrieval of Information by Computer*, Ed. Addison Wesley, New York, 1989.
- [117] Hearst M. A., Plaunt C. – *Subtopic sructuring for full-length document acces*, Proccedings of the ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 59-68, 1993.
- [118] Schaube P. – *Multimedia Information Retrieval – Content based Information Retrieval from Large Text and Audio Databases*, Ed. Kluwer Academic Publishers, Boston, pp. 4 – 60, 1997.
- [119] Naphade M. R., Huang T. S. - *A Probalistic Framework for Semantic Video Indexing, Filtering and Retrieval*, IEEE Transactions on Multimedia, Vol. 3, No. 1, pp. 141-151, March 2001.
- [120] Berson S., Ghandeharizadeh S. – *Staggered Striping in Multimedia Information System*, Proccedings of the ACM SIGMOID International Conference on Management of Data, pp. 79-90, Minneapolis, USA, May 1994.
- [121] Cocquerez J. P., Philipp S. – *Analyse d'images: filtrage e segmentation*, Ed. Mason, Paris, 1995.
- [122] Gagalowicz A. – *Vers un modele de textures*, These d'Etat, Universite Pierre et Marie Curie, Paris VI, 1983.
- [123] Connors R.W., Harlow C.A. – *Toward a structural textural analyser based on statistical methods*, CGIP, 1980.
- [124] Parker J. R. – *Algorithms for Image Processing and Computer Vision*, Ed. Wiley Computer Publishing New York, 1997.
- [125] Connors R.W., Harlow C.A. – *A theoretical comparaisn of texture algorithms*, IEEE Trans on PAMI, vol 2, 1980.
- [126] Mocofan M., VasIU R., Alexa F. – *Algoritm pentru segmentarea imaginilor folosind proprietățile texturilor*, Analele Facultății de Inginerie din Hunedoara, Tomul I, Fascicola 2, Proceedings of the International Symposium, pp. 23-28, 4-5 Nov. 1999.
- [127] Davis L.S., Johns S.A., Aggarwal J.K. – *Texture analysis using generalized co-occurrence matrices*, IEEE Trans on PAMI, vol 1, 1979.
-

-
- [128] Davis L.S., Clerman M., Aggarwal J.K. – *An empirical evaluation of generalized co-occurrence matrices*, IEEE Trans on PAMI, vol 3, 1981.
- [129] Galloway M.M. – *Texture analysis using gray level run lengths*, CGIP 4, 172-179, 1975.
- [130] Gonzales R., Woods R. – *Digital Image Processing*, Ed. Addison-Wesley Publishing Company, London, 1992.
- [131] Matsuyama T., Miura S., Nagao M. – *Structural analysis of natural textures by Fourier transformation*, CVGIP, 24, 347-362, 1983.
- [132] Jourlin M., Pinoli J.C., Zeboudj R. – *Contrast definition and contour detection for logarithmic images*, Journal of Microscopy 1989.
- [133] Zeboudj R. – *Filtrage, Seuillage, Contrast et Contours du pre-traitement a l'analyse d' images*, These de l'Universite de Saint-Etienne, 1988.
- [134] Yokoya N., Levine M. D. – *Range image segmentation based on differential geometry*, IEEE Trans on PAMI, vol 11, 1989.
- [135] Philipp S., Smadja M. – *Approximation of granular textures by quadric surface*, Pattern Recognition, No. 27, pp. 125-132, 1994
- [136] Rao A.R. – *A taxonomy for texture description and indentification*, Springer series in Perception engineering, Heidelberg, 1990
- [137] Brodatz P. – *Textures: A Photographic Album for Artists and Designers*, Dover Publishing Co., 1966.
- [138] Fu K.S. – *Syntactic methods in Pattern Recognition*, Academic Press, 1974
- [139] Zucker S.W. – *Toward a model of texture*, CGIP 5, 1973.
- [140] Ballard D., Brown C. – *Computer Vision*, Prentice-Hall, New York, 1992.
- [141] Pratt W., Faugeras O.D. – *Applications of Stochastic Texture Field Models to Image Processing*, IEEE Transactions on Image Processing, vol 69, 1981
- [142] Haralick R. M. – *Textural features for image classification*, IEEE Trans SMC 3, 1985
- [143] Dunn D. – *Design Gabor filters for texture segmentation*, PHD Tesis Penn. State Univ. 1992
- [144] Law I. K. – *Texture Image Segmentation*, Ed. University Southern California Press, 1980.
- [145] Bovik A., Clark M. – *Multichannel texture analysis using localized spatial filters*, IEEE Trans. Patt. Anal. Machine Intell vol. 12, 1990
- [146] Randen T. – *Filter and Filter Bank Design for Image Texture Recognition*, PHD Tesis 1997.
- [147] Dunn D., Higgins W. – *Optimal Gabor Filers for Texture Segmentations*, IEEE Transactions on Image Processing, july 1995.
- [148] Tuner A., Pichler O., Hosicka B. – *Unsupervised texture Segmentation of Image Using Tuned Matched Gabor Filers*, IEEE Transactions on Image Processing, june 1995.
- [149] Comer M. L., Delp E. J. – *Multiresoluion Image Segmentation*, Computer Vision and Image Processing, 1997.
- [150] Yao J., Krolak P., Steele C. – *The Generalized Gabor Transform*, IEEE Trans on Image Processing, vol 4, no.7, 1995.
- [151] Bastiaans M. – *Gabor's expansion of a signal into Gaussian elementary signals*, Opt.Eng. vol. 20, no.4, 1981.
- [152] Wexler J., Raz S. – *Discrete Gabor expansions*, Signal Processing vol. 21, no. 3, 1990.
- [153] Dunn D., Higgins W. – *2D analysis of Gabor filter output signatures for texture segmentation*, Proc. IEEE Int. Conf. Acoust. Speech Signal Processing, vol 3, 1992.
- [154] Dunn D., Higgins W. – *Texture segmentation using 2D Gabor elementary functions*, IEEE Transactions on Patt. Anal. Machine vol. 16, 1994.
-

-
- [155] Dunn D., Higgins W. – *Optimal Gabor Filters for Texture Segmentations*, IEEE Transactions on Image Processing, july 1995.
- [156] Dunn D. – *Design Gabor filters for texture segmentation*, PHD Tesis Pennsylvania State University 1992.
- [157] P. Peebles – *Probability, Random Variables and Random Signal Principles*, Ed. McGraw-Hill, 1987.
- [158] Francos J. – *Maximum Likelihood Parameter Estimation of Textures Using a Wavelet-Decomposition Based Model*, IEEE Transactions on Image Processing, december 1995.
- [159] Hall T. – *Bispectral Analysis and Model Validation of texture Image*, IEEE Transactions on Image Processing, july 1995.
- [160] Xiaolin W. – *A Segmentation-Based Predictive Multiresolution Image Coder*, IEEE Transactions on Image Processing, january 1995.
- [161] Unser M. – *Texture Classification and Segmentation Using Wavelet Frames*, IEEE Transactions on Image Processing, november 1995.
- [162] Kerivan C., Heitz F. – *A Markov Random Field Model-Based Approach to Unsupervised Texture Segmentation Using Local and Global Spatial Statistics*, IEEE Transactions on Image Processing, june 1995.
- [163] Porat M. – *Localized Texture Processing in Vision: Analysis and Synthesis in the Gaborian Space*, IEEE Transactions Biomed, january 1989.
- [164] Comer M. L., Delp E. J. – *Algorithm for segmentation of textured images*, Computer Vision and Image Processing, 1997.
- [165] Comer M. L., Delp E. J. – *Segmentation of textured image using a multiresolution Gaussian autoregressive model*, Computer Vision and Image Processing, 1997
- [166] Mocofan M., Alexa F. – *The automatic evaluation of two textured surfaces alignment*, Analele Facultății de Inginerie din Hunedoara, Anul 2, Fascicola 3, ISSN 1454-6531, pp 31-36, 2000.
- [167] Mocofan M., Vasiliu R. – *Morphological Texture Segmentation for Image Analysis*, Buletinul Științific al Universității “Politehnica” din Timișoara, Tom 44 (58), Fascicola 2, pp. 127-134, 1999.
- [168] Serra J. - *Image Analysis and Mathematical Morphology*, vol. I, Academic Press, 1982
- [169] Serra J. - *Introduction to Mathematical Morphology*, Computer Vision, Graphics and Image Processing, vol. 35, pp. 283-305, 1986.
- [170] Serra J. - *Image Analysis and Mathematical Morphology. Vol. II*, Theoretical Advances, Academic Press, 1988.
- [171] Gevers T., Smeulders A. W. M. - *Transformation Invariant Image Indexing and Retrieval for Image Databases*, Raport of Faculty of Mathematics & Computer Science, University of Amsterdam, 1995.
- [172] Julesz B. – *Texons, the fundamental elements in preattentive vision and perceptions of textures*, Bell Sys. Tech. vol.62, 1983.
- [173] Marcelja S. – *Mathematical description of the responses of simple cortical cells*, Soc. Amer, 1980.
- [174] Daugman J. G. – *Two dimensional spectral analysis of cortical receptive field profiles*, Vision res. Vol. 20 1980
- [175] Daugman J. G. – *Image analysis and compact coding by oriented 2-D Gabor primitives*, Image Understanding Man-Machine Interface, Vol. 758, 1987
- [176] Ravichandran G., Trivedi M. – *Circular-Mellin Features for Texture Segmentation*, IEEE Transactions on Image Processing, december 1995
-

-
- [177] Jain A. K., Farrokhnia F. – *Unsupervised texture segmentation using Gabor filters*, Pattern Recognition, vol.24, 1991.
- [178] Connors R. W., Harlow C. A. – *Toward a structural textural analyser based on statistical methods*, CGIP, 1980
- [179] Connors R. W., Trivedi M. M. – *Segmentation of a high-resolution urban scene using texture operators*, Computer Vision, Graphics, Image Processing vol. 25, 1984
- [180] Wen-Rong W. – *Rotation and Gray-Scale Transform-Invariant texture Classification Using Spiral Resampling, Subband Decomposition, and Hidden Markov Model*, IEEE Transactions on Image Processing, october 1996.
- [181] Olivier C., Quegan S. - *Understanding Synthetic Aperture Radar Images*, Ed. Artech House, London, 1998.
- [182] Richards A. J., Xiuping J. - *Remote Sensing Digital Image Analysis*, Ed. Springer, London, 1999.
- [183] Gimel'farb G. L. - *Image Textures and Gibbs Random Fields*, Ed. Kluwer Academic Publishers, London, 1999.
- [184] Mocofan M. – **Generarea bazelor de texturi, Sesiune Jubiliară de Comunicări Științifice cu participare internațională - 10 ani de la înființarea Universității “Aurel Vlaicu” din Arad, pp.119-124, 27-28 Octombrie 2000.**
- [185] Parker J. R. – *Algorithms for Image Processing and Computer Vision*, Wiley Computer Publishing, 1997.
- [186] Furht B., Smoliar S. W., Zhang H. J. – *Video and image processing in multimedia systems*, Kluwer Academic Publishers, London, 1995.
- [187] Steven M. – *Modern spectral estimation: Theory and Application*, Prentice Hall, Englewood Cliffs, New York, 1988.
- [188] Toma L. – *Metode și algoritmi de estimare spectrală*, Colecția Prelucrarea semnalelor, Editura Politehnica, 2000.
- [189] Francos J. M., Permuter H. - *Parametric Estimation of The Orientation of Textured Planar Surfaces*, IEEE Transaction on Image Processing, Vol. 10, No 3, pp. 413-418, March, 2001.
- [190] Swapp D. – *Estimation of visual textural gradient using Gabor function*, PHD Univ. Aberdeen, 1996.
- [191] Mocofan M. – **Anticiparea înclinării unui plan în conducerea roboților prin analiza texturilor, Al XV-lea Simpozion Național cu Participare Internațională “Robotica 2000”, Oradea, pp 233-236, 12-14 Octombrie 2000.**
- [192] Gose E., Johnsonbaugh R., Jost S. – *Pattern Recognition and Image Analysis*, Ed. Prentice Hall, New York, 1996.
- [193] Gonzalez R. C., Woods R. E - *Digital Image Processing*, Addison Wesley Publishing, New York, 1992
- [194] Pratt W. K. - *Developing Visual Applications*, Sun Microsystems Press, Prentice Hall, 1997
- [195] Liao S. X., Pawlak M. – *On Image Analysis by Moments*, , IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 18, No. 3, pp. 254-266, 1996.
- [196] Mamistvalov A. G. – *n-Dimensional Moment Invariants and Conceptual Mathematical Theory of Recognition n-Dimensional Solids*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 20, No. 8, pp. 819-831, 1998.
- [197] Pentland A.; Picard W. R., Sclaroff S. – *Photobook: Content-Based Manipulation of Image Databases*, M.I.T. Media Laboratory Perceptual Computing Technical Report No. 255, November 1993.
- [198] Thijs W. – *Image retrieval: Content versus Context*, Grant of Duch organization for Scientific Research NWO, 1999.
-

- [199] Rui Y., Huang T. S., Mehrotra S., Ortega M - *A Relevance Feedback Architecture for Content-based Multimedia Information Retrieval Systems*, NSF/DARPA/NASA DLI Rapport, 1996.
- [200] Pass G., Zabih R. - *Histogram refinement for content-based image retrieval*, IEEE Workshop on Applications of Computer Vision, pp. 96-102, 1996.
- [201] Adjero D. A., Lee M. C. - *On Ratio-Based Color Indexing*, IEEE Transaction on Image Processing, Vol. 10, No 1, pp. 36-48, January, 2001.
- [202] Paschos G. - *Perceptually Uniform Color Spaces for Color Texture Analysis: An Empirical Evaluation*, IEEE Transaction on Image Processing, Vol. 10, No 6, pp. 932-937, June, 2001.
- [203] Mocofan M. - *Algoritm de căutare în baze de imagini bazat pe conținutul imaginii*, Simpozionul Științific Internațional "Universitaria 2000", Petroșani, Romania, 19 - 20 octombrie 2000.
- [204] Mocofan M. - *Algoritm de indexare pentru baze de imagini*, Simpozionul Științific Internațional "Universitaria 2000", Petroșani, Romania, 19 - 20 octombrie 2000.
- [205] Jacobs C. E., Finkelstein A., Salesin D. H. - *Fast Multiresolution Image Querying*, Proceedings of SIGGRAPH 95, Los Angeles, 11-16 August 1995.
- [206] Mocofan M., Căleanu C., Toma C.I. - *Conscience algorithm based on neural network for unsupervised texture image segmentation*, Scientific Bulletin "Acta Universitatis Cibiniensis" of "Lucian Blaga" University of Sibiu, pp.97-101, 1999.
- [207] Mocofan M., Căleanu C., Lacrămă D., Alexa F. - *Unsupervised texture image segmentation*, Proceedings of the 5-th Seminar on Neural Network Applications in Electrical Engineering "Neurel 2000", 25-27 September, Belgrade, Yugoslavia, pp 101-104, 2000.
- [208] Mocofan M., Căleanu C. - *Hierarchical structure of modular self-organizing neural networks for unsupervised texture image segmentation*, Buletinul Științific al Universității "Politehnica" din Timișoara, Tom 44 (58), Fascicola 2, pp. 135-139, 1999.
- [209] Mocofan M. - *Algoritm de indexare și căutare în baze de texturi*, Sesiune Jubiliară de Comunicări Științifice cu participare internațională - 10 ani de la înființarea Universității "Aurel Vlaicu" din Arad, 27-28 Octombrie 2000, Arad, Romania, pp. 113-118, 2000.
- [210] Mocofan M. - *Content-based indexing scheme for texture image databases*, Proceedings of the Symposium on Electronics and Telecommunications "Etc. 2000", 23-24 November 2000, Timisoara, Romania, Tom 45(59), Fascicola 1, pp 210-213, 2000.
- [211] Manjunath B., Ma W. - *Texture Features for Browsing and Retrieval of Image Data*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vo. 18, No. 8, pp. 837-842, 1996.
- [212] Mocofan M., Căleanu C. - *Neural Network based Algorithm for Content Multimedia Databases Search*, Buletinul Institutului Politehnic Iași, Tomul XL VIII, Fasc. 5, pp. 196-200, 2002.
- [213] Mocofan M. - *Content search in an image database*, Proceedings of the Symposium on Electronics and Telecommunications "Etc. 2002", Timisoara, Romania, Tom 47(61), Fascicola 1-2, pp 171 - 174, 19-20 September 2002.
- [214] Mocofan M., Vasiu R. - *Algorithm based on image similarity for content search in an image database*, THIRD COST #276 WORKSHOP on Information and Knowledge Management for Integrated Media Communication, 11- 12 October, 2002, Budapest, Hungary.
- [215] Minka T. - *An Image Database Browser that Learns From User Interaction*, MIT Media Laboratory Technical Report 365, 1995.
- [216] Gudivada V. N., Raghavan V. V., Vanapipat K. - *A Unified Approach to Data Modelling and Retrieval for a Class of Image Database Applications*, Multimedia Database System, pp. 37-74, Ed. Springer, Berlin 1996.

- [217] Sistla P. A., Yu C. – *Retrieval of pictures Using approximate Matching*, Multimedia Database System, pp. 101-111, Ed. Springer, Berlin 1996.
- [218] Belussi A., Bertino E., Biavasco A., Rizzo S. – *Filtering Distance Queries in Image Retrieval*, Multimedia Database System, pp. 185-211, Ed. Springer, Berlin 1996.
- [219] Jagadish H. V. - *Indexing for Retrieval by Similarity*, Multimedia Database System, pp. 165-181, Ed. Springer, Berlin 1996.
- [220] Hu M.K. - “*Visual pattern recognitions by moments invariants*”, Computer Methods in Image Analysis, IEEE Computer Society, Los Angeles, 1977.
- [221] Jai A.K. – *Fundamentals of Digital Image Processing*”, Prentice Hall, New York, 1989.
- [222] Jahne B. – *Digital Image Processing, Concepts, Algorithms and Scientific Applications*, Ed Springer, 1997.
- [223] Lyon D. A. – *Image Processing in JAVA*, Ed. Prentice Hall, New York, 1999.
- [224] Mocofan M. – *An algorithm for image content-based retrieval in a video sequence*, Proceedings of the Symposium on Electronics and Telecommunications “Etc. 2000”, 23-24 November 2000, Timisoara, Romania, Tom 45(59), Fascicola 1, pp 204-209, 2000.
- [225] Naphade M. R., Kozintsev V. I., Huang T. S. – *A Factor Graph Framework for Semantic Video Indexing*, IEEE Transactions on Circuits and Systems for Video Technology, Vol 12, No. 1, pp. 40-52, january 2002.
- [226] Tzanetakis G., Luc J. – *Multimedia structuring trees*, Rapport of Computer-Human Interaction Center CA94025.
- [227] Rui Z., Huang S. T. – *A unified framework for video browsing and retrieval*, Grant of ARL Cooperative Agreement and CSE Fellowship No DAAL01-96-2-0003, 1999.
- [228] Rui Z., Huang S. T., Mehrotra S. – *Constructing Table of Content for Videos*, Grant of ARL Cooperative Agreement and CSE Fellowship No DAAL01-96-2-0003, 1999.
- [229] Uchihashi S., Foote J., Girgensohn A., Boreczky J. - *Video MANGA: Generating Semantically Meaningful Video Summaries*, Proceeding of The 7-th ACM International Multimedia Conference, Orlando, SUA, 30 october - 5 November, pp. 383-392, 1999.
- [230] Merialdo B., Lee T. K., Luparello D., Roudaire J. - *Automatic Construction of Personalized TV News Programs*, Proceeding of The 7-th ACM International Multimedia Conference, Orlando, SUA, 30 october - 5 November, pp. 323-331, 1999.
- [231] Milanese R., Cherbuliez M., Pun T. – *Invariant Content-Based Image Retrieval Using The Fourier-Melin Transform*, OFES 95.0493 Grant Raport, 1995.
- [232] Neti C., Maison A., Senior A., Iyengar G., De Cuetos P., Basu S., Verma A. - *Joint processing of audio and visual information for multimedia indexing and human-computer interaction*, IBM T. J. Watson Research Center Yorktown Heights Rapport, NY 10598, 1998.
- [233] Li C. F., Gupta A, Sanocki E., He L., Rui. Y. – *Browsing Digital Video*, Raport of Multimedia Microsoft Research and Group for User Interface Research EECS, 1999.
- [234] Ghandeharizadeh S. – *Stream-based Versus Structured Video Objects: Issues, Solutions, and Challenges*, Multimedia Database System, pp. 215-235, Ed. Springer, Berlin 1996.
- [235] De Cuetos P., Saporilla D., Ross K. W. – *Adaptive Streaming of Video in a TCP-Friendly Context: Multiple Versions or Multiple Layers*, Raport of Institute EUROCOM, France, 2000.
- [236] Schaube P. – *Multimedia Information Retrieval – Content based Information Retrieval from Large Text and Audio Databases*, Ed. Kluwer Academic Publishers, Boston, pp. 61 – 136, 1997.
- [237] Ozden B., Rastogi R., Silberschatz A. – *The storage and Retrieval of Continuous Media Data*, Multimedia Database System, pp. 237-260, Ed. Springer, Berlin 1996.

-
- [238] Mayer-Patel K., Rowe L. A. – *Exploiting Temporal Parallelism For Software-only Video Effects Processing*, Proceeding of The 6-th ACM International Multimedia Conference, Bristol, England, 12-16 September, pp. 161-169, 1998. Mayer-Patel K.,
- [239] Rowe L. A. – *A Multicast Control Scheme for Parallel Software-only Video Effects Processing*, Proceeding of The 7-th ACM International Multimedia Conference, Orlando, SUA, 30 october - 5 November, pp. 409-418, 1999.
- [240] Luther A. C.: *Digital Video in the PC Environment*, McGraw-Hill Book Company, New York, 1991
- [241] Watkinson J. – *An Introduction to Digital Video*, Focal Press, London, 1994.
- [242] Watkinson J. – *The Art of Digital Video*, Focal Press, London, 1994
- [243] Ozer J. – *Video Compression for Multimedia*, Academic Press, London, 1995.
- [244] Watkinson J. – *Compression in Video and Audio*, Focal Press, London, 1995.
- [245] Bojkovic Z., Toma C. I., Gui V., VasIU R. – *Advanced Topics in Image Compression Techniques*, Politehnica University of Timisoara Publishing House, 1997
- [246] Watkinson J. – *Compression in Video and Audio*, Focal Press, London, 1995.
- [247] Clarke R. J. – *Digital Compression of Still Images and Video*, Academic Press, 1995
- [248] Ozer J. - *Publishing Digital Video*, Academic Press, London, 1997.
- [249] Paris J. F., Long D. D. E., Mantey P. E. – *Zero-Delay Broadcasting Protocols for Video-on-Demand*, Proceeding of The 7-th ACM International Multimedia Conference, Orlando, SUA, 30 october - 5 November, pp. 189-198, 1999.
- [250] Eager D., Vernon M., Zahorjan J. – *Optimal and Efficient Merging Schedules for VoD Servers*, Proceeding of The 7-th ACM International Multimedia Conference, Orlando, SUA, 30 october - 5 November, pp. 199-202, 1999.
- [251] Pochueva J., Munson E. V. – *Optimizing Video-on-Demand Trough Requestcasting*, Proceeding of The 7-th ACM International Multimedia Conference, Orlando, SUA, 30 october - 5 November, pp. 207-210, 1999.
- [252] Al-Khatib W., Ghafoor A. – *An Approach for Video Meta-Data Modeling and Query Processing*, Proceeding of The 7-th ACM International Multimedia Conference, Orlando, SUA, 30 october - 5 November, pp. 215-224, 1999.
- [253] Cai J., Hua K. A. – *An Efficient Bandwidth - Sharing Technique for True VoD Systems*, Proceeding of The 7-th ACM International Multimedia Conference, Orlando, SUA, 30 october - 5 November, pp. 211-214, 1999.
- [254] Gao L., Zhang Z. L., Towsley D. – *Catching and Selecting Catching: Efficient latency Reduction Techniques for Delivering Continuous Multimedia Streams*, Proceeding of The 7-th ACM International Multimedia Conference, Orlando, SUA, 30 october - 5 November, pp. 203-206, 1999.
- [255] Servetto S. D., Nahrstedt K. – *Broadcast Quality Video over IP*, IEEE Transactions on Multimedia, Vol. 3, No. 1, pp. 162-173, March 2001.
- [256] Yang D. N., Liao W., Lin Y. T. – *MQ: An Integrated Mechanism for Multimedia Multicasting*, IEEE Transactions on Multimedia, Vol. 3, No. 1, pp. 82-97, March 2001.
- [257] Lombardo A., Schembra G., Maribito G. – *Traffic Specifications for the Transmission of Stored MPEG Video on the Internet*, IEEE Transactions on Multimedia, Vol. 3, No. 1, pp. 5-17, March 2001.
- [258] Puri R., Lee K. W., Ramchandram K., Bhargaran V. – *An Integrated Source Transcoding and Congestion Control Paradigm for Video Streaming in the Internet*, IEEE Transactions on Multimedia, Vol. 3, No. 1, pp. 18-32, March 2001.
-

- [259] Wu M., Joyce R. A., Wong H. S., Guan L., Kung S. Y. – *Dynamic Resource Allocation via Video Content and Short-Term Traffic Statistics*, IEEE Transactions on Multimedia, Vol. 3, No. 2, pp. 186-199, June 2001.
- [260] Gallardo J. R., Makrakis D., Angula M. – *Dynamic Resource Management Considering The Real Behavior of Aggregate Traffic*, IEEE Transactions on Multimedia, Vol. 3, No. 2, pp. 177-185, June 2001.
- [261] Ghosh D., Sarangon V., Acharya R. – *Quality of Service Routing in IP Networks*, , IEEE Transactions on Multimedia, Vol. 3, No. 2, pp. 200-208, June 2001.
- [262] Radha H. M., vander Schaar M., Chen Y. – *The MPEG 4 - Fine Grained Scalable Video Coding Method for Multimedia Streaming Over IP*, IEEE Transactions on Multimedia, Vol. 3, No. 1, pp. 56-68, March 2001.
- [263] Furini M., Towsley D. F. – *Real Time Traffic Transmission over the Internet*, , IEEE Transactions on Multimedia, Vol. 3, No. 1, pp. 33-40, March 2001.
- [264] Shin J., Kim J., Kuo C. C. J. – *Quality of Service Mapping Mechanism for Packet Video in Differentiated Services Network*, , IEEE Transactions on Multimedia, Vol. 3, No. 2, pp. 219-231, June 2001.
- [265] Sandbank C. P. – *Digital Television*, John Wiley & Sons, London, 1990.
- [266] Tarnai E. J. – *Digital Television*, Bell-Northern Research, 1994.
- [267] Murat T. – *Digital Video Processing*, Prentice Hall, 1995
- [268] Rosenfeld A., Kak A. C. - *Digital Picture Processing*, Academic Press, New York, 1982.
- [269] Mocofan M., Andone D. – *Dezvoltarea aplicațiilor multimedia*, Curs IDD, Centrul de multiplicare UPT, 2002.
- [270] Mocofan M. – *Producție Audio Video*, Curs IDD, Centrul de multiplicare UPT, 2003.
- [271] Mocofan M. – *Echipamente de studio*, Curs IDD, Centrul de multiplicare UPT, 2002.
- [272] Mocofan M., Vasiu R. – *Quality of MPEG coded video sequences*, THIRD COST #276 WORKSHOP on Information and Knowledge Management for Integrated Media Communication, 11- 12 October, 2002, Budapest, Hungary.
- [273] Mocofan M., Vasiu R. – *On Testing MPEG Compressed Digital Video*, Proceedings of the Symposium on Electronics and Telecommunications “Etc. 2002”, Timisoara, Romania, Tom 47(61), Fascicola 1-2, pp 175 – 179, 19-20 September 2002.
- [274] Castellani X. – *MCO: Méthodologie d'analyse et de conception des systèmes a objets*, Ingenierie des besoins, Ed. Masson, Paris, 1993.
- [275] Martin J., Odell J. – *Object Oriented Analysis and Design*, Ed. Prentice-Hall Inc., Englewood Cliffs, New York, 1992 .
- [276] Mocofan M. – *MMDB:NICE – Multimedia Image Database With Image Content Searching*, Proceedings of International Symposium on System Theory, Automation, Robotics, Computers, Informatics, Electronics and Instrumentation, “SINTES-11”XI-th edition, 22-23 October 2003, Craiova, trimisă spre publicare.
- [277] Powell A. L., French J. C. - *The Impact of Databases Selection on Distributed searching*, SIGIR 2000, Proceedings of the 23 rd Annual International ACM SIGIR Conference on Research an Development in Information Retrieval, Athens, Greece, pp. 232-239, 24-28 July 2000.
- [278] Arjen P. - *MIRROR: Multimedia Query Processing in Extensible Databases*, Raport of Centre for Telematics and Information Technology, University of Twente, Netherlands, 1999.
- [279] van Doorn M., de Vries A. - *The Psychology of Multimedia Databases*, Proceedings of the Fifth ACM Conference on Digital Library, San Antonio, Texas, pp. 1-10, 2 - 7 June 2000.

-
- [280] Mocofan M. – *Prevenirea și evaluarea catastrofelor naturale folosind metode moderne de prelucrare a imaginilor*, Raport contract CNCSIS nr 33501, Tema 4, Cod 4, 2003 Timișoara, spre publicare
- [281] Mocofan M, Căleanu C. Mărănescu V. – *Tree Content Based Search Algorithm For An Image Databases*, The Scientific Bulletin: "Acta Tehnica Napocensis Electronics and Telecommunications", Cluj, 2003 trimisă spre publicare.
- [282] Glockner I., Knoll A. - *Natural Language Navigation in Multimedia Archives: An Integrated Approach*, Proceeding of The 7-th ACM International Multimedia Conference, Orlando, SUA, 30 october - 5 November, pp. 313-322, 1999
- [283] Gholamhosein S., Wendy C., Aidong Z. - *Semantic Clustering and Queryng on Heterogeneous Features for Visual Data*, Proceeding of The 6-th ACM International Multimedia Conference, Bristol, England, 12-16 September, pp. 3-12, 1998.
- [284] Ponceleon D., Srinivasan S., Amir A., Petkovic D., Diklic D. - *Key to Effective Video Retrieval: Effective Cataloging and Browsing*, Proceeding of The 6-th ACM International Multimedia Conference, Bristol, England, 12-16 September, pp. 99-107, 1998.
- [285] Echigo T., Kurokawa M., Tomita A., Miyamoei H. Iisaku S. - *Video Enrichement: Retrieval and Enhanced Visualization Based on Object Behavior*, ACCV 2000 - Proceedings of the Fourth Asian Conference on Computer Vision, Taipei, Taiwan, pp. 384-399, 8-11 January 2000.
- [286] Delis V., Papadias D., Mamoulis N., - *Assessing Multimedia Similarity: A framework for Structure and Motion*, Proceeding of The 6-th ACM International Multimedia Conference, Bristol, England, 12-16 September, pp. 333-338, 1998.
- [287] Hurst W., Muller R., Mayer C. - *Multimedia information retrieval from recorded presentations*, SIGIR 2000, Proceedings of the 23 rd Annual International ACM SIGIR Conference on Research an Development in Information Retrieval, Athens, Greece, pp. 339-341, 24-28 July 2000.