

**UNIVERSITATEA "POLITEHNICA" din TIMIȘOARA  
FACULTATEA DE ELECTRONICĂ ȘI TELECOMUNICAȚII**

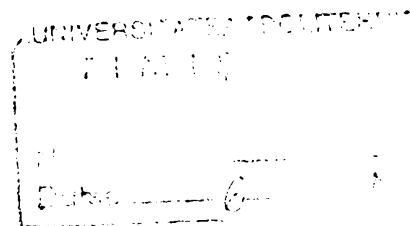
# **TEZĂ DE DOCTORAT**

**Studiul metodelor de compresie  
utilizate în prelucrarea și transmiterea  
numerică a imaginilor**

BIBLIOTECA CENTRALĂ  
UNIVERSITATEA "POLITEHNICA"  
TIMIȘOARA

**Autor:**  
**ing. Radu Adrian VASIU**

**Conducător științific:**  
**Prof.dr.ing. Corneliu I. TOMA**



**1998**

18321

## Mulțumiri

Doresc să exprim cele mai alese mulțumiri domnului prof.dr.ing. Corneliu I. Toma, conducătorul științific al tezei, pentru sprijinul și îndrumarea constantă realizată cu foarte multă delicatețe, care m-a încurajat mereu în momentele dificile de căutări ce au apărut pe parcursul realizării acestei teze. Discuțiile și încurajările primite m-au făcut să înțeleg că întotdeauna există loc de mai bine și să conștientizez necesitatea de a prezenta într-o formă cât mai accesibilă rezultatele cercetărilor mele.

O mențiune specială revine domnului profesor Zoran Bojković, de la Universitatea din Belgrad, pentru sprijinul și încrederea acordate. Schimbul de idei și îndrumarea în obținerea unor informații mi-au fost de un real folos și au condus la accelerarea ritmului de prelucrare a datelor necesare acestei teze de doctorat. Totodată, dumnealui a citit cu deosebită atenție varianta tezei redactată în limba engleză. Observațiile pertinente făcute au contribuit în mod esențial la îmbunătățirea variantei finale în limba română.

Mulțumiri speciale domnilor prof.dr.ing. Ioan Naforniță și prof.dr.ing. Alimpie Ignea, pentru sugestiile oferite pe parcursul elaborării tezei cât și pentru consecvența cu care au urmărit finalizarea acesteia, precum și pentru sprijinul acordat în activitatea de zi cu zi în cadrul Facultății de Electronică și Telecomunicații, Departamentul de Comunicații.

Mulțumesc domnului prof.dr.ing. Mircea Petrescu, de la Universitatea “Politehnica” din București, pentru sprijinul acordat și pentru acceptul de a deveni referent al tezei.

Domnului prof.dr.ing Radu Zaci, de la Universitatea “Politehnica” din București, îi mulțumesc pentru discuțiile fructuoase purtate încă de la începutul abordării temei de doctorat.

Sincere mulțumiri domnului conf.dr.ing. Vasile Gui, care mi-a deschis drumul spre fascinantul domeniu al compresiei imaginilor și care mi-a făcut o serie de sugestii valoroase pe parcursul elaborării tezei, contribuind la confirmarea unora din rezultatele obținute.

Domnului ing. Sorin Ivașcu, de la MobilRom, îi mulțumesc în mod deosebit pentru fructuoasa colaborare și pentru ajutorul acordat la implementarea software a unor algoritmi specifici morfologiei matematice.

Mulțumesc tuturor colegilor din Departamentul de Comunicații pentru crearea unei atmosfere colegiale de lucru și pentru încurajările permanente.

Pentru sprijinul acordat în obținerea multor materiale bibliografice le mulțumesc domnilor David Evans, de la Bilston Community College, Anglia și Andrew Marriott, de la University of Central Lancashire, Preston, Anglia.

Poate această teză ar fi fost dificil de realizat fără suportul familiei mele, care a crezut în permanență în mine.

Autorul

Timișoara, ianuarie 1998

## Cuprins

<b>Capitolul 1. Introducere</b>	11
<b>Capitolul 2. Bazele compresiei digitale a imaginilor</b>	14
<b>2.1. Introducere</b>	14
<b>2.2. Codarea imaginilor</b>	15
2.2.1. Fundamente	15
2.2.2. Măsura informației și entropia sursei	17
2.2.3. Entropia semnalelor video	18
<b>2.3. Codarea reversibilă a sursei</b>	20
2.3.1. Tehnici de codare cu lungime variabilă	20
2.3.1.1. Codul Shannon-Fano	21
2.3.1.2. Codul Huffman	22
2.3.1.3. Coduri cu virgulă	23
2.3.1.4. Coduri $B_n$	24
2.3.1.5. Coduri $S_n$	24
2.3.2. Codarea ariilor constante	25
2.3.3. Codarea “run-length”	25
2.3.3.1. Codarea “run-length” unidimensională	26
2.3.3.2. Codarea “run-length” bidimensională	26
2.3.4. Codarea pe plane de biți	27
2.3.4.1. Descompunerea planelor de biți	27
2.3.4.2. Codarea planelor de biți	28
<b>2.4. Codarea nereversibilă a sursei</b>	28
2.4.1. Codarea cu predicție	29
2.4.2. Codarea cu modulație delta	30
2.4.3. Codarea cu modulația diferențială a impulsurilor în cod	30
2.4.4. Codarea cu transformare	31
2.4.5. Codarea hibridă	34
<b>Capitolul 3. Metode de codare a imaginilor orientate pe contur și formă</b>	36
<b>3.1. Definirea problemei</b>	36
<b>3.2. Tehnici orientate pe contur</b>	37
3.2.1. Coduri lanț	37
3.2.2. Algoritm de codare a conturilor prin coduri lanț	39
3.2.3. Aproximări poligonale	43
3.2.4. Semnături	45
3.2.5. Segmente de contur	46
3.2.6. Scheletul unei regiuni	47
<b>3.3. Descrierea formei pe baza conturului</b>	51
3.3.1. Reprezentarea geometrică simplă a frontierei	51
3.3.1.1. Lungimea conturului	51
3.3.1.2. Diametrul conturului	51

3.3.1.3. Curbura	51
3.3.1.4. Distribuția corzilor	52
3.3.2. Numere ale formei	53
3.3.3. Descriptori Fourier	55
3.3.4. Momente	57
<b>Capitolul 4. Bazele morfologiei matematice</b>	59
<b>4.1. Operatori morfologici binari de bază</b>	60
4.1.1. Definiții de bază	60
4.1.2. Transformări morfologice	61
4.1.2.1. Dilatarea	61
4.1.2.2. Erodarea	64
4.1.2.3. Deschiderea și închiderea	66
4.1.2.4. Transformarea "Hit-or-Miss"	69
4.1.2.5. Principalele clase de elemente structurante	70
4.1.2.6. Proprietăți generale ale operatorilor morfologici	71
<b>4.2. Algoritmi morfologici de bază</b>	73
4.2.1. Extragerea conturilor	73
4.2.2. Umplerea regiunilor	74
4.2.3. Dilatarea condiționată	75
4.2.4. Extragerea componentelor conectate	75
4.2.5. Înfășurătoarea convexă	76
4.2.6. Scheletul	77
4.2.7. Subțierea	79
4.2.8. Îngroșarea	80
4.2.9. Metode practice de realizare a operațiilor de subțiere și îngroșare	81
4.2.10. Curățarea	83
<b>4.3. Morfologie pe scară de gri</b>	85
4.3.1. Introducere	85
4.3.2. Operatori morfologici de bază pe scară de gri	86
4.3.2.1. Dilatarea pe scară de gri	86
4.3.2.2. Erodarea pe scară de gri	87
4.3.2.3. Deschiderea și închiderea pe scară de gri	89
4.3.3. Aplicații ale morfologiei pe scară de gri	91
4.3.3.1. Netezirea morfologică	91
4.3.3.2. Gradientul morfologic	92
4.3.3.3. Segmentarea texturilor	92
4.3.3.4. Transformarea "top-hat"	94
4.3.3.5. Distribuția obiectelor după mărime	95
4.3.3.6. Morfologie color	95
<b>4.4. Utilizarea morfologiei matematice în compresia imaginilor</b>	96
4.4.1. Descompunerea morfologică adaptivă în subbenzi	96
4.4.2. Descompunerea piramidală morfologică	98
4.4.3. Codarea morfologică cu transformare distanță	101
4.4.4. Codarea morfologică bazată pe regiuni	101
4.4.4.1. Operatori conectați	102
4.4.4.2. Algoritmi de creștere a regiunilor	104
4.4.4.3. Scheletul geodezic	106
4.4.4.4. Interpolarea morfologică	108

<b>4.5. Descompunerea morfologică a elementului structurant</b>	110
4.5.1. Necesitatea descompunerii	110
4.5.2. Rezultate în descompunerea morfologică a elementului structurant	111
4.5.3. Contribuții la descompunerea morfologică a elementului structurant	114
4.5.3.1. Proprietăți ale reprezentărilor prin coduri lanț	115
4.5.3.2. Operatori morfologici pentru domenii convexe	118
4.5.3.3. Descompunerea domeniilor convexe	119
4.5.3.3.1. Definirea problemei	119
4.5.3.3.2. Setul de elemente de bază	119
4.5.3.3.3. Noua metodă de descompunere propusă	121
<b>Capitolul 5. Calitatea și performanțele tehnicilor de compresie a imaginilor</b>	132
<b>5.1. Măsurarea performanțelor compresiei</b>	132
<b>5.2. Măsurări cantitative pentru calitatea imaginii</b>	132
<b>5.3. Aprecierea subiectivă a calității imaginii pe baza sistemului vizual uman</b>	135
<b>5.4. Cerințele de calitate pentru informația video</b>	137
5.4.1. Calitatea imaginii	137
5.4.2. Erori vizibile	137
5.4.3. Întârzierea	138
5.4.4. Rata transmisiei de date	138
<b>5.5. Teoria rată-distorsiune</b>	140
5.5.1. Funcția rată-distorsiune	140
5.5.2. Funcția rată-distorsiune în termenii teoriei informației	141
5.5.3. Interpretarea spectrală a relației rată-distorsiune	146
5.5.4. Reprezentarea spectrală bidimensională și rata pentru sursa Markov de primul ordin	148
5.5.5. Măsura planității spectrale	149
5.5.6. Concluzii	151
<b>Capitolul 6. Implementarea algoritmului propus pentru compresia imaginilor</b>	152
<b>6.1. Introducere</b>	152
<b>6.2. Implementarea operatorilor morfologici de bază</b>	153
6.2.1. Dilatarea	153
6.2.2. Erodarea	154
6.2.3. Deschiderea și închiderea	156
6.2.4. Operatori morfologici pe scară de gri	157
<b>6.3. Descompunerea pe plane de biți a imaginilor pe scară de gri</b>	160
<b>6.4. Scheletul morfologic</b>	165
6.4.1. Conceptul de axă medială	165
6.4.2. Implementarea scheletului morfologic	166
<b>6.5. Reprezentarea conturului pentru imagini binare</b>	170
<b>6.6. Tehnici combinate contur-schelet</b>	175
6.6.1. Minimizarea scheletului condiționată de contur	175
6.6.2. Reconstrucția imaginii din scheletul minimal	178
<b>6.7. Codarea celor mai puțin semnificative plane de biți</b>	179
<b>6.8. Concluzii</b>	180

<b>Capitolul 7. Rezultate experimentale</b>	181
7.1. Imagini de test	182
7.2. Extragerea morfologică a conturilor și descompunerea în seturi de schelete	183
7.3. Rezultatele compresiei	189
7.4. Aprecierea calității	191
7.5. Imaginile de test decodate	195
7.6. Concluzii și perspective	196
<b>Concluzii</b>	197
<b>Bibliografie</b>	202



## Capitolul 1 Introducere

Această teză de doctorat este dedicată studierii efectului proprietăților morfologiei matematice în aplicații de compresia digitală a imaginilor. Teza prezintă metodele morfologiei matematice, integrarea lor în tehnicile de compresia imaginilor, utilitatea acestor unelte, precum și problemele pe care le pune utilizarea acestor metode în aplicații de acest gen. Se propune un nou algoritm pentru compresia imaginilor cu utilizarea metodelor morfologiei matematice, algoritm implementat și evaluat, care a condus la obținerea unor bune rezultate.

O prezentare a principalelor tehnici de codare numerică a imaginilor este inclusă în capitolul 2. Capitolul debutează cu o scurtă introducere a principiilor codării imaginilor din punctul de vedere al teoriei informației, cu scopul de a stabili limitele teoretice pe care le putem viza în codarea numerică a imaginilor. În continuare sunt prezentate principalele tehnici de codare fără pierderi a sursei: codarea cu lungime variabilă, codarea ariilor constante, codarea “run-length” și codarea planelor de biți. Se insistă mai mult asupra codării fără pierderi deoarece algoritmul propus pentru codarea imaginilor prin metode ale morfologiei matematice utilizează și descompunerea pe plane de biți, codarea “run-length” și codarea Huffman. O scurtă prezentare a metodelor de codare cu pierderi este de asemenea făcută, fără multe detalii, doar pentru facilitarea unor comparații.

Capitolul 3 tratează tehnicile de codare orientate pe formă și contur. După introducerea problemei, voi prezenta tehnicile de bază ale codării orientate pe contur: codurile lanț, aproximările polinomiale, semnăturile, segmentele de contur și scheletul unei regiuni. Principalele tehnici de descriere geometrică a imaginilor prin reprezentarea hotarelor, prin numerele forme (shape numbers), prin descriptorii Fourier și prin momente au fost considerate de interes în contextul compresiei morfologice a imaginilor, datorită similarității cu caracterul descriptiv geometric al morfologiei matematice. Scopul acestui capitol este de a pregăti elementele necesare implementării morfologice a algoritmului propus, care utilizează codarea conturilor și reprezentarea scheletului regiunii.

Capitolul 4 reprezintă o introducere destul de detaliată în morfologia matematică. Sunt prezentați principalii operatori morfologici (dilatarea, erodarea, deschiderea, închiderea și transformarea “hit-or-miss”) împreună cu proprietățile lor. În continuare sunt prezentați algoritmi morfologici de bază utilizați în prelucrarea imaginilor, ca de exemplu: extragerea conturilor (boundary extraction), umplerea regiunilor (region filling), dilatarea condiționată (conditional dilation), extragerea componentelor conectate (connected components extraction), înfășurătoarea convexă (convex hull), scheletul (skeleton), subțierea (thinning), îngroșarea (thickening) și curățirea (pruning). Următoarea secțiune tratează morfologia pe scară de gri, prezentând operatorii morfologici de bază și câteva aplicații ca: netezirea (smoothing), gradientul, segmentarea texturilor, transformarea “top-hat”, distribuția după mărime a obiectelor. Totodată sunt prezentate câteva noțiuni de morfologie color. De asemenea sunt prezentate câteva exemple de utilizare a morfologiei matematice în compresia

imaginilor: descompunerea morfologică adaptivă în subbenzi, descompunerea morfologică piramidală, codarea cu transformare distanță morfologică și codarea morfologică bazată pe regiuni prin operatori conectați, creșterea regiunilor, scheletul geodesic și interpolarea morfologică. În final, în secțiunea 4.5 voi discuta necesitatea descompunerii elementului structurant morfologic, precum și rezultatele cunoscute până acum în acest domeniu. În continuare voi prezenta câteva contribuții originale la descompunerea elementului structurant. Voi deduce câteva proprietăți interesante ale reprezentării domeniilor convexe prin coduri lanț run-length, utile pentru descompunerea morfologică a elementului structurant. După ce voi da o interpretare grafică a operatorilor de dilatare și erodare pentru domeniile convexe, voi propune o metodă grafică pentru realizarea descompunerii. Voi propune un set de 12 elemente structurante de bază care pot fi utilizate pentru descompunerea oricărui domeniu convex, demonstrând în același timp că acest set restrâns dă aceleași rezultate ca și setul de 13 elemente structurante care este uzual utilizat în literatura de specialitate [168], [169], [284], [291].

Un obiectiv important în compresia imaginilor îl constituie asigurarea unei calități a imaginii corespunzătoare cerințelor aplicației. Evaluarea calității nu este un lucru simplu în aplicațiile video. Capitolul 5 prezintă o serie de mărimi care pot caracteriza calitatea imaginilor, mărimi care vor fi utilizate în capitolul 7 pentru evaluarea rezultatelor. Capitolul 5 include și o discuție referitoare la teoria rată-distorsiune.

Capitolul 6 descrie algoritmul propus pentru compresia imaginilor prin utilizarea morfologiei matematice. În primul rând voi prezenta operatorii morfologici de bază care au fost implementați cu scopul de a putea fi utilizați în continuare în aplicațiile complexe. Operatorii de dilatare, erodare, deschidere și închidere au fost implementați atât în binar cât și pe scară de gri. În continuare este propus algoritmul de codare a imaginilor pe scară de gri. Pentru a simplifica implementarea și a obține un bun raport de compresie, utilizez o descompunere a imaginii pe plane de biți. Diversele plane vor fi codate prin metode diferite pentru a obține o eficiență crescută. Cele mai semnificative 4 plane de biți sunt codate fără pierderi, deoarece ele conțin cea mai importantă parte de informație. Există două posibilități, ambele fiind evaluate și testate. Prima posibilitate constă în utilizarea codării conturilor cu coduri lanț, iar a doua constă în utilizarea reprezentării imaginii prin schelete. Am constatat că utilizarea scheletelor poate conduce la rezultate mai bune pe o clasă mai largă de tipuri de imagini. Voi utiliza o metodă similară cu cea utilizată de Maragos și Schafer pentru imagini binare [130], dar folosind un element structurant având dimensiunea crescătoare exponențial, folosind totodată și metoda de minimizare a seturilor de schelete pe care ei au propus-o. Reuniunea seturilor de schelete obținute va fi minimizată în continuare prin așa numita *minimizare a scheletelor condiționată de contur*. Seturile de schelete obținute pentru cele mai semnificative 4 plane de biți sunt codate pentru transmisie prin metode de codare cu lungime variabilă. Pentru ultimele 4 plane de biți, cele mai puțin semnificative, am propus un algoritm de codare diferit, ținând cont de faptul că pentru aceste plane care nu conțin informație semnificativă este de dorit o rată de compresie cât mai bună. Pentru planul 0 voi determina densitatea de probabilitate de apariție a pixelilor albi/negri pentru întreg planul, acesta fiind singurul parametru care se transmite. La decodarea imaginii, se va genera local o imagine aleatoare cu aceeași densitate de probabilitate de apariție a pixelilor albi/negri, care va fi adăugată imaginii reconstituite. Pentru planele 1, 2 și 3 nu este de așteptat să se obțină o densitate de probabilitate cvasi-constantă, de aceea

imaginea va fi segmentată în blocuri de dimensiuni  $1/4$ ,  $1/8$  și  $1/16$ . Același parametru ca și pentru planul 0 este calculat și transmis pentru fiecare bloc de imagine.

Rezultatele obținute în urma implementării pe diferite clase de imagini sunt prezentate în capitolul 7. Aceste rezultate sunt evaluate prin intermediul raportului bit/pixel și prin aprecierea subiectivă a calității imaginii obținute. Este prezentată și o evaluare a calității imaginii prin intermediul raportului semnal/zgomot. Concluzia este că se poate obține un foarte bun raport bit/pixel pentru o clasă largă de imagini. Raportul total de compresie obținut este mai bun decât cele anterior prezentate în literatura de specialitate pentru aceeași clasă de tehnici de compresie. Imaginile decodate obținute sunt foarte bune prin prisma aprecierii subiective a calității imaginilor. Totuși, în urma studiului făcut, raportul semnal/zgomot a fost găsit ca nerelevant în acest caz. Chiar pentru rapoarte semnal/zgomot nu foarte bune, imaginea obținută în urma decodării arată mai bine decât așteptat, din punct de vedere al percepției vizuale, datorită proprietăților descriptive geometrice care stau la baza metodelor morfologiei matematice.

Scopul prezentului studiu a fost realizarea unui algoritm practic de codare a imaginilor prin utilizarea metodelor morfologiei matematice, algoritm care a fost testat și evaluat. Sugestii cu privire la posibile direcții de continuare a cercetărilor sunt prezentate în capitolul 7.

În final este prezentat un rezumat al contribuțiilor autorului și o listă bibliografică, incluzând rezultate anterioare ale autorului în domeniul abordat.

## Capitolul 2

### Bazele compresiei digitale a imaginilor

#### 2.1. Introducere

Atunci când o funcție intensitate luminoasă bidimensională este eșantionată și cuantizată pentru a crea o imagine digitală este produsă o cantitate enormă de date. Cantitatea de date generată poate fi atât de mare încât să ducă la imposibilitatea îndeplinirii cerințelor de stocare, prelucrare și transmitere. De aceea este necesară o prelucrare corespunzătoare a semnalului video digital care să conducă la reducerea debitului de informație care trebuie transmis. Acest deziderat, cunoscut sub denumirea de *compresia imaginilor*, este realizat în principal prin utilizarea altor modalități de codare față de simpla eșantionare bidimensională și cuantizare clasică a nivelurilor de gri [39], [59], [117].

Compresia imaginilor se referă la problema reducerii cantității de date cerute pentru reprezentarea unei imagini digitale. La baza acestui proces de reducere a cantității de date stă procesul de înlăturare a datelor redundante.

Din punct de vedere matematic, problema înseamnă transformarea unei arii bidimensionale de pixeli, într-un set de date statistic necorelate. Această transformare are loc înainte de stocarea sau transmiterea imaginii. Mai târziu, imaginea compresată va fi decompresată pentru reconstruirea imaginii originale sau a unei aproximări a ei.

Interesul privind compresia imaginilor datează de aproximativ 30 de ani. Eforturile de cercetare inițiale au fost în domeniul dezvoltării de metode analogice pentru reducerea benzii de transmisie video, proces numit de compresie a lățimii de bandă. Ulterior, dezvoltarea calculatoarelor numerice, precum și a circuitelor integrate avansate a condus la translatarea interesului către metode digitale de compresie.

De-a lungul anilor, necesitatea compresiei imaginilor a crescut în mod constant. Curent, ea este recunoscută ca fiind unul din factorii tehnologici care permit dezvoltarea unui număr tot mai mare și mai variat de aplicații.

De exemplu, compresia imaginilor a fost și continuă să fie crucială pentru dezvoltarea aplicațiilor multimedia. Totodată, compresia imaginilor constituie tehnologia actuală pentru satisfacerea cerințelor legate de rezoluția spațială în continuă creștere a senzorilor de imagine și a standardelor de transmitere în televiziunea de înaltă definiție. În plus, compresia imaginii joacă un rol important în multe alte aplicații ca: videoconferința, teleducția, imagini medicale, transmisii facsimil și controlul vehiculelor telecomandate în aplicații militare sau spațiale. Pe scurt, un număr crescând de aplicații depind de eficiența manipulare, stocare și transmitere a imaginilor digitale pe tonuri de gri sau color.

În prezentul capitol voi examina pe scurt aspectele teoretice și practice ale procesului de compresie a imaginilor. Într-o primă secțiune voi prezenta aspectele legate de redundanța informației care pot fi exploatate de algoritmi de compresie a imaginilor. Apoi voi prezenta un model general pentru procesul de compresie-decompresie, examinând totodată câteva noțiuni de teoria informației și rolul lor în stabilirea limitelor fundamentale în reprezentarea imaginilor.

În continuare voi prezenta principalele tehnici utilizate în compresia imaginilor, care se împart în două mari categorii: metode reversibile și metode nereversibile. Metodele din prima categorie sunt utile în principal în arhivarea imaginilor (de exemplu pentru stocarea înregistrărilor de imagini medicale și de documente legale). Aceste metode permit ca o imagine să fie compresată și decompresată fără pierdere de informație. Metodele din cea de-a doua categorie asigură un nivel mai înalt de compresie, dar implică o anumită pierdere de informație la reproducerea imaginii. Aceste metode se utilizează, de obicei, în aplicații ca: televiziune radiodifuzată, transmisii de videoconferințe și facsimil.

## 2.2. Codarea imaginilor

### 2.2.1. Fundamente

Conform teoremei lui Shannon privind codarea în prezența zgomotului, teoretic este posibilă transmiterea informației printr-un canal cu o probabilitate a erorii oricât de mică, cu condiția ca debitul de transmisie să fie inferior capacității canalului. Aceasta înseamnă că, prin utilizarea unor metode de codare care să reducă suficient de mult debitul informației, se poate realiza un canal de transmisie transparent, adică un canal care să nu introducă o deteriorare importantă a semnalului. Practic, cu condiția ca raportul său semnal/zgomot să nu scadă sub o anumită valoare, un semnal codat poate fi efectiv regenerat și transmis repetitiv pe distanțe lungi fără o acumulare importantă a zgomotului sau a distorsiunilor. Calitatea semnalului perceput de receptor este în mare măsură dependentă de tipul codării.

Procesul de codare într-un sistem de transmisie digitală a imaginilor, (ca de altfel în orice sistem de comunicație numeric) se compune din codarea sursei și codarea canalului. În figura 2.1 este prezentată schema simplificată a unui lanț de codare a semnalului de imagine.

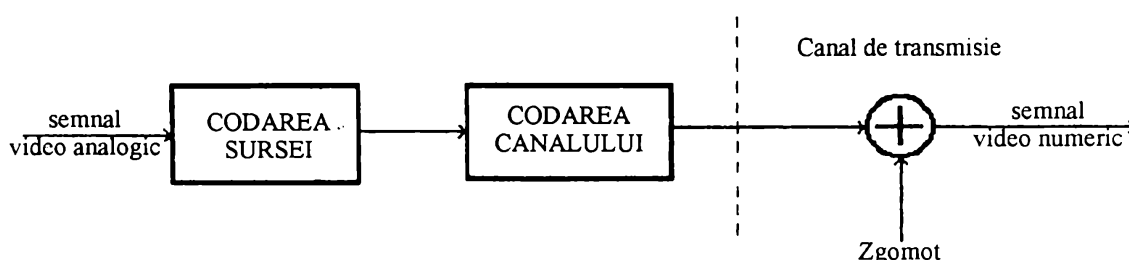


Fig. 2.1. Lanț de codare video

Blocul de codare a sursei efectuează compresia (reducerea) debitului binar prin micșorarea sau chiar suprimarea totală a redundanței prezente în semnalul de imagine. Blocul de codare a canalului realizează adaptarea semnalului la canalul de transmisie, introducând în general o redundanță cu scopul asigurării posibilității de corecție a erorilor introduse de zgomotul care apare pe canal. Global, însă, pentru sistemele video numerice, se poate obține o reducere semnificativă a debitului binar total (în general de

circa 50%) chiar în condițiile introducerii protecției contra erorilor, care determină o creștere a debitului binar de circa 5-10% [265].

În ceea ce privește codarea sursei, scopul principal al reducerii debitului binar îl constituie micșorarea lărgimii de bandă a canalului, permițându-se astfel reducerea costului legăturii. De exemplu, este inutilă transmiterea unui semnal video de calitate tradițională în cazul videoconferințelor deoarece, în acest caz, imaginile sunt lent variabile. Pentru acest tip de aplicații, tehnici complexe de codare a sursei, (inclusiv codarea intercadre) au permis obținerea unui raport de compresie a debitului binar de ordinul 40:1 și utilizarea unui debit binar de ordinul 1Mbit/s [195].

Principalele tehnici de codare în scopul reducerii debitului binar al semnalului de imagine se înscriu în două categorii: metode de codare reversibilă și metode de codare nereversibilă a sursei [265].

Codarea reversibilă, după cum arată și numele, prezintă o reversibilitate totală la nivelul decodurului, astfel încât nici o informație nu se pierde în timpul procesului de codare. Metoda mai poartă și denumirea de codare fără pierderi, codare statistică, codare cu lungime variabilă sau codare a entropiei, deoarece cuvântul de cod cel mai scurt corespunde evenimentului cel mai probabil. Un exemplu tipic pentru cazul transmisiei de date este în acest sens codul Morse. Aplicarea acestui tip de codare pentru semnalul de televiziune necesită cunoașterea probabilităților statistice ale imaginilor video și prin urmare cunoașterea entropiei sursei. Problema este însă foarte dificilă deoarece imaginile video se modifică permanent într-un mod arbitrar.

Codarea nereversibilă implică pierderea unei mici cantități de informație în cursul procesului de codare, informație care nu mai poate fi recuperată la nivelul decodurului. În această categorie se înscriu în principal metodele de codare cu transformare și cu predicție. Tot în această categorie se înscrie de fapt și eșantionarea sub-Nyquist, deoarece, nici în acest caz nu se poate evita o oarecare pierdere de informație datorată nerespectării condițiilor impuse de teorema eșantionării.

Este de menționat că există posibilitatea combinării mai multor tehnici de codare pentru obținerea compresiei dorite. Un exemplu îl constituie utilizarea eșantionării sub-Nyquist combinată cu modulația diferențială a impulsurilor în cod, obținându-se astfel un raport de compresie de 3:1 [183], [194].

În ceea ce privește codarea canalului (sau codarea liniei), aceasta poate fi considerată ca o adaptare a semnalului codat al sursei la canalul de transmisie. Procesul de codare a canalului poate să implice, în cazul cel mai general, următoarele procese: codarea pentru controlul erorilor, criptarea, translatarea codului și modularea purtătoarei numerice.

Cea mai simplă modalitate de protecție contra erorilor constă într-o tehnică de disimulare prin care se înlocuiește un eșantion eronat prin ultimul eșantion corect. Atunci însă când este necesară o veritabilă corecție a erorilor, se recurge la coduri corectoare de erori simple: codul Hamming sau codul Wyner-Ash [43], [139]. De exemplu, sistemele teletext utilizează codul Hamming pentru protejarea transmisiunilor importante ca numerele paginilor și cuvintele de adresă de rând. Dacă sistemul de transmisie trebuie să facă față unor erori multiple aleatoare, atunci se pot utiliza coduri mai complexe: de exemplu codul Bose-Chaudhuri-Hocquenghem (BCH) pentru sistemele de videoconferință. În cazul erorilor multiple în rafale este preferabilă utilizarea unor coduri corectoare de pachete de erori: de exemplu codul Reed-Solomon (RS). Disponând de un număr mare de coduri de protecție contra erorilor, alegerea se face în funcție de aplicație, urmărindu-se obținerea unor performanțe globale

satisfăcătoare, ceea ce pentru sistemele video înseamnă atingerea unei rate a erorilor binare în canalul de transmisiune de ordinul  $10^{-4}$  [203].

### 2.2.2. Măsura informației și entropia sursei

Teoria informației furnizează modele matematice pentru caracterizarea eficienței codării. În prezentul paragraf voi reaminti pe scurt conceptele de măsură a informației, entropie, debit de informație și capacitate a canalului.

Fundamentul teoriei informației îl constituie ideea că generarea informației poate fi modelată printr-un proces probabilistic, care poate fi apreciat în concordanță cu intuiția.

După cum s-a mai arătat, codarea reversibilă a sursei mai poartă și denumirea de codare statistică, codare cu lungime variabilă sau codare a entropiei. După cum arată și ultima denumire, această tehnică de codare face apel la conceptul de entropie. Prezentarea tehnicilor de codare reversibilă a sursei implică o descriere cantitativă a sursei discrete și, în particular, a conținutului de informație de transmis. Acestea sunt furnizate de entropia sursei și respectiv de debitul entropiei [69], [233].

Să considerăm o sursă discretă având un alfabet limitat  $A$ , format din  $M$  simboluri diferite  $a_j$  ( $j = 1, 2, \dots, M$ ), cu probabilitatea de apariție  $p(a_j)$ . Această probabilitate este numită probabilitate a priori pentru a o deosebi de probabilitatea a posteriori, care corespunde probabilității ca mesajul sosit la receptor să fie corect. Prin urmare

$$p(a_j) \geq 0, \quad j = 1, 2, \dots, M \quad (2.1)$$

și

$$\sum_{j=1}^M p(a_j) = 1 \quad (2.2)$$

Cu cât probabilitatea de apariție a unui simbol este mai mică, cu atât ne așteptăm mai puțin la apariția sa. Cu alte cuvinte, putem obține mai multă informație decât în cazul unui simbol cu apariție curentă. Având în vedere această remarcă, se poate defini autoinformația unei surse discrete, care pentru fiecare simbol  $a_j$  are valoarea:

$$I(a_j) = \log_2 \frac{1}{p(a_j)} = -\log_2 p(a_j) \quad (2.3)$$

unde folosind pentru logaritmul baza 2, s-a utilizat cuantizarea standard a informației în biți.

Este însă mult mai utilă descrierea sursei prin informația medie pe care o generează pe simbol decât prin informația instantanee pe simbol. Presupunând o sursă fără memorie (toate simbolurile sunt statistic independente), entropia sursei (informația medie pe simbol) este:

$$H(A) = -\sum_{j=1}^M p(a_j) \cdot \log_2 p(a_j) \quad [\text{biți/simbol}] \quad (2.4)$$

Entropia este o măsură a incertitudinii, a caracterului aleator al sursei. Valoarea sa este cu atât mai mică cu cât sursa are un caracter mai ordonat. La limită, dacă  $p(a_1) = 1$  și  $p(a_j) = 0$  pentru  $j=2, 3, \dots, M$ , atunci

$$H(A)_{\min} = 0 \quad (2.5)$$

În cazul unei surse foarte neordonate pentru care toate simbolurile sunt echiprobabile (adică  $p(a_j) = \frac{1}{M}$  pentru  $j=1, 2, \dots, M$ ) se obține valoarea maximă a entropiei:

$$H(A)_{\max} = \sum_{j=1}^M \frac{1}{M} \log_2 M = \log_2 M \quad (2.6)$$

Prin urmare avem:

$$0 \leq H(A) \leq \log_2 M \quad (2.7)$$

În legătură cu entropia, în cazul semnalului video, trebuie făcute următoarele observații:

- Entropia  $H(A)$  are doar mică semnificație dacă  $p(a_j)$  variază în timp, adică sursa nu este staționară. Prin urmare ne putem aștepta la dificultăți în cazul semnalelor video de exemplu.
- Entropia, calculată cu relația (2.4), nu reprezintă cu precizie conținutul mediu de informație decât dacă simbolurile sunt statistic independente, adică dacă sursa este fără memorie. Pentru surse cu memorie, la care apariția unui simbol dat depinde de simbolurile care îl preced, vom putea ține cont de această particularitate prin introducerea în expresia entropiei a probabilităților condiționate.

Expresia entropiei  $H(A)$  este importantă deoarece ea stă la baza teoremei lui Shannon de codare în absența zgomotului, prin care se demonstrează că  $H(A)$  (exprimată în biți/eșantion sau biți/simbol) poate fi considerată ca un obiectiv pentru schemele de compresie. O bună codare trebuie să aibă o lungime medie a cuvântului de cod apropiată de  $H(A)$ . Aceeași teoremă specifică faptul că debitul de entropie

$$R = r \cdot H(A) \text{ [biți/sec]} \quad (2.8)$$

unde  $r$  este debitul de simboluri, dă o limită cu privire la capacitatea canalului.

### 2.2.3. Entropia semnalelor video

Utilizarea relației (2.4) pentru calculul entropiei unui sistem video numeric poate conduce la rezultate false, din cauza redundanței mari pe care o prezintă un astfel de sistem.

Să considerăm pentru început problema transmisiei unei imagini statice monocrome, cuantizate uniform cu 64 niveluri de gri, adică codate cu 6 biți / eșantion. Se poate calcula densitatea de putere discretă a semnalului MIC prin măsurarea frecvenței relative de apariție a acestor niveluri de gri în imagine. Rezultatul unei astfel de determinări pentru o imagine oarecare este prezentat în figura 2.2.

În general, curba obținută este plată, dar, examinând -o în detaliu, se observă că ea nu este uniformă. Trebuie subliniat faptul că în această estimare nu s-a ținut seama de nici o dependență statistică (corelație între eșantioane). S-a demonstrat [107] încă din 1952 că pentru o imagine oarecare, înlocuind cele 64 valori pentru probabilitățile  $p(a_j)$ , se obține o valoare a entropiei  $H(A)$  de aproximativ 5 biți pe eșantion.



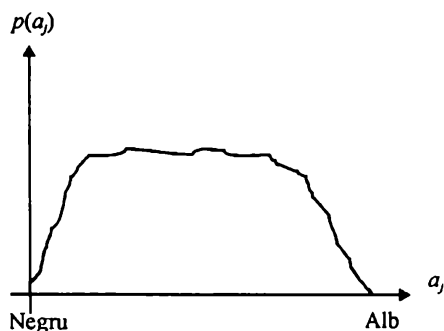


Fig. 2.2. Densitatea de putere pentru o imagine oarecare

Prin urmare există o redundanță statistică:

$$E = \log_2 M - H(A) = \log_2 64 - 5 = 1 \text{ bit/eașantion}$$

ceea ce indică posibilitatea utilizării unor forme de codare optimală în scopul obținerii unui debit binar apropiat de 5 biți pe eașantion.

Se știe însă că o imagine codată MIC prezintă o corelație importantă între eașantioane, așa încât, pentru realizarea unei transmisiuni realmente eficace, este necesar ca eașantioanele transmise să fie decorelate, obținându-se astfel o sursă discretă cu o entropie reală apropiată de entropia zero.

În figura 2.3 este prezentat un decorelator foarte simplu, dar eficient, realizat pe baza predicției pornind de la eașantionul precedent.

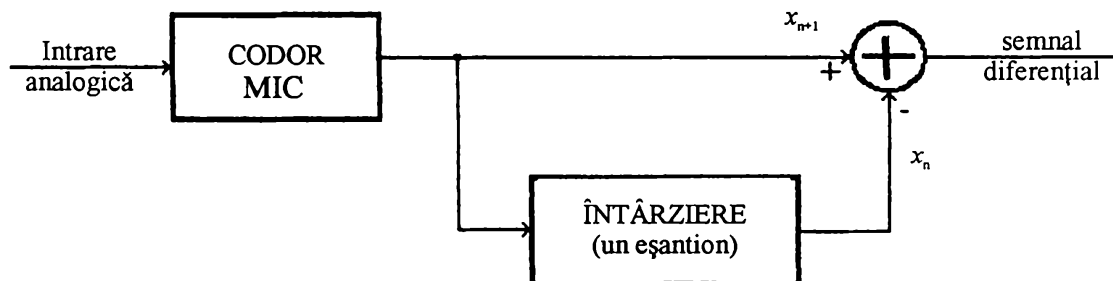


Fig. 2.3. Decorelator

Într-un astfel de sistem putem considera că probabilitatea condiționată ca două eașantioane învecinate să fie egale este aproximativ unitară. Prin urmare este rezonabil să considerăm că eașantioanele diferențiale din figura 2.3 sunt aproape statistic independente și că densitatea de putere discretă a semnalului diferențial are forma de variație prezentată în figura 2.4.

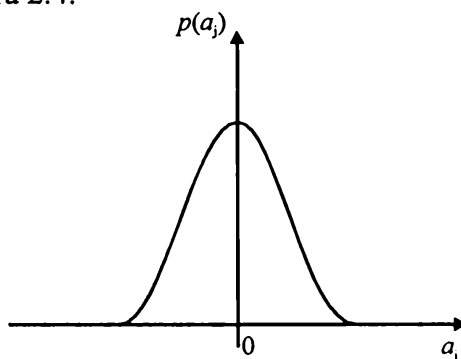


Fig. 2.4. Densitatea de putere discreta

S-a demonstrat de asemenea [107] că pentru aceleași imagini considerate anterior, dar utilizând 64 de valori  $p(a_j)$  pentru densitatea de putere condiționată din figura 2.4 se obține o entropie de ordinul zero de aproximativ 3 biți pe eșantion. Rezultă o redundanță de corelație de circa 3 biți/eșantion a transmisiunii MIC directe pe 6 biți/eșantion. Prin urmare trebuie să existe metode de codare optimală a semnalului diferențial prin care să se realizeze o compresie de 2:1.

În general entropia de ordinul zero a semnalului diferențial este inferioară entropiei de ordinul zero a semnalului MIC. Se poate deci realiza o reducere a debitului binar prin codarea independentă a fiecărui eșantion diferențial. Acest tip de transmisiune are o mare eficiență în codarea sursei, fiind totodată foarte bine adaptat la semnalele de televiziune în culori.

Considerând un semnal color PAL, eșantionat cu de  $N$  ori frecvența subpurtătoarei de culoare și aplicând la ieșirea MIC de 8 biți un decorelator ca cel din figura 2.3, prin efectuarea unor măsurători statistice asupra unor imagini diferite se obține o entropie cuprinsă între  $3,5 \div 6,5$  biți pe eșantion diferențial. Ținând cont și de faptul că frecvența de eșantionare se poate reduce la aproximativ 10 MHz prin suprimarea stingerilor, rezultă o estimare mai bună a capacității canalului necesar pentru difuzarea semnalelor video de calitate de circa  $35 \div 65$  Mbit/sec. Prin urmare o transmisiune PAL la debitul de 34 Mbit/sec recomandat de CCITT este perfect realizabilă prin utilizarea tehnicilor de compresie.

### 2.3. Codarea reversibilă a sursei

#### 2.3.1. Tehnici de codare cu lungime variabilă

Fiind demonstrată importanța transmisiunii diferențiale, rămâne problema codării semnalului diferențial (care conține toată informația utilă) printr-o tehnică optimală care să permită reducerea debitului binar. În cazul unui semnal MIC de 8 biți, eșantioanele diferențiale vor lua valori între  $-255$  și  $+256$ , ceea ce pentru transmitere directă ar implica utilizarea a 9 biți, deci o creștere a debitului binar. Pentru reducerea acestui debit binar există în principal două modalități de acțiune asupra semnalului diferențial, ambele bazate pe densitatea de putere neuniformă a semnalului diferențial: semnalele mici sunt mult mai probabile decât cele mari.

O primă metodă constă în utilizarea unui număr mai mic de niveluri de cuantizare și prin urmare a unui cuvânt de cod mai scurt pentru transmitere. Deși zgomotul de cuantizare crește, el poate fi menținut la valori acceptabile ținând cont de faptul că un circuit de cuantizare neuniformă este bine adaptat la o densitate de putere neuniformă. Cuantizarea va fi fină pentru nivelurile cele mai probabile și mai grosieră pentru nivelurile mai puțin probabile. Acest concept se utilizează la MDIC.

A doua metodă constă în translatarea codului, făcând să corespundă nivelurilor de semnal cele mai probabile cuvinte de cod scurte și celorlalte, cuvinte de cod mai lungi. Această metodă conduce la codarea cu lungime variabilă.

Procesul de codare cu lungime variabilă poate fi definit formal ca fiind conversia unei secvențe de simboluri sursă într-o secvență de simboluri cod. Fie  $A$  alfabetul sursei, definit prin probabilitățile a priori  $p(a_j)$  și  $C$  alfabetul codului, definit prin lungimile  $l_j$  ale cuvintelor de cod  $c_j$ .

$$A = \begin{cases} a_1 & a_2 & a_3 & \dots & a_M \\ p(a_1) & p(a_2) & p(a_3) & \dots & p(a_M) \end{cases} \Rightarrow C = \begin{cases} c_1 & c_2 & c_3 & \dots & c_M \\ l_1 & l_2 & l_3 & \dots & l_M \end{cases}$$

Lungimea medie a cuvântului de cod va fi:

$$\bar{L} = \sum_{j=1}^M p(a_j) \cdot l_j \quad (2.9)$$

iar randamentul codării poate fi definit ca:

$$\eta = \frac{H(A)}{\bar{L}} \quad (2.10)$$

În cele ce urmează vom prezenta succint câteva din cele mai des utilizate coduri cu lungime variabilă, care pot fi utilizate la reprezentarea scheletului unei imagini (vezi capitolul 7).

### 2.3.1.1. Codul Shannon-Fano

Să considerăm un set de  $M$  simboluri sursă  $a_j$ ,  $j = 1, 2, \dots, M$  care reprezintă o imagine digitală și fie  $p(a_j)$  probabilitățile de apariție a acestor simboluri. În cazul unei surse fără memorie și a unei codări binare, algoritmul de codare Shannon-Fano este următorul:

Pasul 1: Se stabilește lista simbolurilor sursă în ordinea descrescătoare a probabilităților de apariție.

Pasul 2: Se separă această listă în două grupe cu sumele probabilităților cât mai apropiate. Se asociază "0" tuturor simbolurilor dintr-o grupă și "1" tuturor simbolurilor din cealaltă grupă.

Pasul 3: Se repetă pasul 2 în cadrul fiecărei grupe, adăugând câte un nou bit la codul fiecărui simbol. Procedura se sfârșește atunci când fiecare subgrup conține doar un simbol sursă.

Pentru a exemplifica constituirea codului Shannon-Fano vom considera o sursă cu 8 simboluri  $a_1, a_2, \dots, a_8$  având probabilitățile de apariție:

$$\begin{array}{ll} p(a_1)=0,07 & p(a_5)=0,07 \\ p(a_2)=0,12 & p(a_6)=0,36 \\ p(a_3)=0,12 & p(a_7)=0,07 \\ p(a_4)=0,07 & p(a_8)=0,12 \end{array}$$

Practic, aceste probabilități pot fi estimate prin determinarea histogrammei imaginii digitale.

Tabelul 2.1 ilustrează construirea codului Shannon-Fano. O partiționare în subgrupe și o alocare a unui bit în cuvântul de cod constituie o etapă.

Tabelul 2.1. Construirea codului Shannon-Fano

$a_i$	$p(a_i)$	Etapa 1	Etapa 2	Etapa 3	Etapa 4
$a_6$	0,36	0	00		
$a_2$	0,12	0	01		
$a_3$	0,12	1	10	100	
$a_8$	0,12	1	10	101	
$a_1$	0,07	1	11	110	1100
$a_4$	0,07	1	11	110	1101
$a_5$	0,07	1	11	111	1110
$a_7$	0,07	1	11	111	1111

Lungimea medie a cuvântului de cod este  $\bar{L} = 2,80$  iar entropia sursei este  $H(A) = 2,71$  astfel încât randamentul codului este  $\eta = 96,6\%$ .

După ce codul a fost generat, procesul de codare și/sau decodare este realizat foarte simplu pe bază tabelară. Codul obținut este instantaneu unic decodabil, prin urmare nu apar ambiguități în procesul de decodare. Totuși, este de remarcat faptul că o anumită ambiguitate poate să apară în cursul construcției codului, în cazul în care există posibilitatea partiționării elementelor în mai multe moduri datorită unor sume de probabilități egale.

### 2.3.1.2. Codul Huffman

Metoda Huffman garantează obținerea unui cod optimal și nu este afectată de ambiguitatea amintită la algoritmul anterior. Metoda presupune parcurgerea următoarelor etape:

Pasul 1: Stabilirea listei simbolurilor sursei în ordinea descrescătoare a probabilităților.

Pasul 2: Combinarea celor mai mici două probabilități, prin adunare și reordonarea listei, urmată de repetarea operațiunii până la obținerea unei liste cu doar 2 probabilități.

Pasul 3: Generarea cuvintelor de cod pornind de la ultimul pas și revenind spre probabilitățile inițiale, prin atribuirea de "0" și respectiv "1" logic pe ramurile care apar.

Considerând aceeași sursă ca și pentru exemplul anterior, procedura de codare Huffman este ilustrată în tabelul 2.2.

Tabelul 2.2. Construirea codului Huffman

$a_i$	$p(a_i)$	Pasul1	Pasul2	Pasul3	Pasul4	Pasul5	Pasul6
$a_1$	0,36	0,36	0,36	0,36	0,36	0,38	0,62 0
$a_2$	0,12	0,14	0,14	0,24	0,26	0,36	0,38 1
$a_3$	0,12	0,12	0,14	0,14	0,24	0,26	0,26 1
$a_4$	0,12	0,12	0,12	0,14	0,14	0,14	0,14 1
$a_5$	0,07	0,12	0,12	0,12	0,12	0,12	0,12 1
$a_6$	0,07	0,07	0,12	0,12	0,12	0,12	0,12 1
$a_7$	0,07	0,07	0,07	0,07	0,07	0,07	0,07 1
$a_8$	0,07	0,07	0,07	0,07	0,07	0,07	0,07 1

Se obțin următoarele coduri finale:

$a_1$ : 00	$a_5$ : 0100
$a_2$ : 011	$a_6$ : 0101
$a_3$ : 100	$a_7$ : 110
$a_4$ : 101	$a_8$ : 111

Pentru acest cod avem:  $\bar{L} = 2,78$ ;  $H(A) = 2,71$  și  $\eta = 97,48\%$ .

Codul Huffman descris prezintă un dezavantaj. Pentru imagini cu multe niveluri de gri se vor obține unele cuvinte de cod foarte lungi. Cu prețul unei mici reduceri a gradului de compresie, codul Huffman poate fi modificat astfel încât să fie evitate cuvintele de cod lungi. Versiunea rezultată poartă denumirea de cod Huffman deplasat. Deși nu este un cod optimal, prezintă avantajul de a fi ușor de utilizat.

Procedura de generare a unui cod Huffman deplasat este următoarea:

Pasul 1: Stabilirea listei simbolurilor sursei în ordinea descrescătoare a probabilităților.

Pasul 2: Împărțirea listei globale a simbolurilor în blocuri cu același număr de simboluri sursă.

Pasul 3: Codarea identică a elementelor individuale din fiecare bloc, cu o procedură Huffman.

Pasul 4: Adăugarea de simboluri speciale pentru deplasare-sus și/sau deplasare-jos, pentru identificarea fiecărui bloc.

Atunci când un simbol de deplasare-sus sau deplasare-jos este recunoscut la nivelul decodorului, un bloc este deplasat sus sau jos față de un bloc de referință predefinit.

O altă implementare practică a procedurii Huffman este constituită de codul Huffman trunchiat, generat în conformitate cu următoarea procedură:

Pasul 1: Stabilirea listei simbolurilor sursei în ordinea descrescătoare a probabilităților.

Pasul 2: Codarea celor mai probabile  $k$  simboluri sursă după procedura Huffman originală.

Pasul 3: Codarea restului de simboluri sursă prin utilizarea unui prefix și a unui cod cu lungime fixă.

Cele două proceduri modificate de codare Huffman sunt mult mai simplu de implementat. Uneori este necesară o sacrificare a eficienței codării în favoarea simplității implementării.

### 2.3.1.3. Coduri cu virgulă

Un cod cu virgulă este un cod simplu cu lungime variabilă la care un simbol binar este utilizat pentru a semnifica sfârșitul fiecărui cuvânt de cod. Pentru anumite aplicații, este preferabilă utilizarea unui astfel de cod pentru simplificarea implementării și pentru creșterea rezistenței la erori. Randamentul codării este mai mic decât la codul Huffman, totuși în anumite situații se pot obține rezultate foarte bune [65].

Să considerăm exemplul unei surse cu 11 simboluri diferite. Tabelul 2.3 prezintă comparativ codul cu virgulă și codul Huffman corespunzător, împreună cu lungimea medie a cuvântului de cod.

Valoarea obținută pentru lungimea medie a cuvântului de cod este  $\bar{L} = 2,576$  pentru codul cu virgulă, în comparație cu  $\bar{L} = 2,499$  pentru codul Huffman.

Tabelul 2.3. Codul cu virgulă comparat cu codul Huffman

$a_i$	$p(a_i)$	Codul cu virgulă	Codul Huffman
$a_1$	0,4	0	0
$a_2$	0,175	10	111
$a_3$	0,175	110	110
$a_4$	0,09	1110	100
$a_5$	0,09	11110	1011
$a_6$	0,028	111110	10100
$a_7$	0,028	1111110	101011
$a_8$	0,005	11111110	1010100
$a_9$	0,005	111111110	10101011
$a_{10}$	0,002	1111111110	101010101
$a_{11}$	0,002	11111111110	101010100

Codul cu virgulă obținut este un cod unic decodabil. În condițiile unei eficiențe puțin inferioare codului Huffman, codul cu virgulă se dovedește uneori a fi o alegere mai bună datorită simplității și autoresincronizării sale imediate după fiecare eroare.

### 2.3.1.4. Coduri $B_n$

Codurile  $B_n$  sunt utilizate pentru codarea pixelului curent în funcție de probabilitatea de apariție a nivelului său. Codurile  $B_n$  presupun utilizarea așa-numiților "biți de continuare" pentru recunoașterea începutului unui nou cuvânt de cod.

De exemplu, pentru un cod  $B_1$ , succesiunea

00101010100 sau 100000011110

semnifică trei pixeli cu codurile 0, 001 și 10, biții de continuare fiind cei subliniați, restul fiind biți de informație. Codul  $B_1$  se caracterizează prin faptul că se utilizează un singur bit de informație pentru fiecare bit de continuare. La coduri  $B$  de ordin superior se utilizează  $n$  biți de informație pentru fiecare bit de continuare.

Vom exemplifica modul de formare a codurilor  $B_1$  și  $B_2$  în tabelul 2.4.

Tabelul 2.4. Construirea codurilor  $B$

$a_i$	$p(a_i)$	Codul $B_1$	Codul $B_2$
$a_1$	0,4	C0	C00
$a_2$	0,3	C1	C01
$a_3$	0,1	C0 C0	C10
$a_4$	0,1	C0 C1	C11
$a_5$	0,6	C1 C0	C00 C00
$a_6$	0,04	C1 C1	C01 C01

Lungimile medii ale cuvintelor de cod sunt  $\bar{L}_{B_1} = 2,6$  și  $\bar{L}_{B_2} = 3,3$  în comparație cu entropia sursei  $H(A)=2,14$ . Se poate demonstra [65] că aceste coduri sunt eficiente atunci când probabilitățile de apariție a nivelurilor de intrare satisfac cât mai bine o lege de forma  $p(a_j) = j^{-\gamma}$ , unde  $\gamma$  este o constantă pozitivă.

### 2.3.1.5. Coduri $S_n$

Codurile  $S_n$  sunt coduri de lungime variabilă prezentând avantajul ușurinței de implementare. Un cod  $S_n$  folosește cuvinte de cod de  $n$  biți, din care  $2^n - 1$  sunt utilizate pentru codarea unor simboluri ale sursei, iar ultimul pentru a semnifica că nivelul de intrare este în afara domeniului.

Tabelul 2.5 exemplifică modul de formare al codului  $S_2$  pentru aceeași sursă ca și în tabelul 2.4.

Tabelul 2.5. Construirea codului  $S_2$

$a_i$	$p(a_i)$	$S_2$ -code
$a_1$	0,4	00
$a_2$	0,3	01
$a_3$	0,1	10
$a_4$	0,1	11 00
$a_5$	0,06	11 01
$a_6$	0,04	11 10

Se obține  $\bar{L} = 2,4$  biți/eșantion și  $\eta = 89,1\%$ .

Codurile cu lungime variabilă prezentate sunt foarte utile pentru codarea numerică a imaginilor cu un număr mic de niveluri de gri. Acest lucru le face foarte utile în algoritmul de codare pe care l-am propus, pentru codarea seturilor de schelete, care va fi descrisă în capitolul 7.

### 2.3.2. Codarea ariilor constante

O metodă simplă, dar eficientă, pentru compresia imaginilor binare constă în utilizarea unor cuvinte de cod speciale pentru identificarea ariilor largi de "1" sau "0". Metoda poartă denumirea de *codarea ariilor constante CAC (Constant Area Coding)*.

Imaginea este divizată în blocuri de mărime  $m \times n$  pixeli, care sunt clasificate ca fiind complet albe, complet negre sau cu intensitate mixtă. Celei mai probabile sau celei mai frecvente categorii  $i$  se alocă cuvântul de cod de un bit 0, iar celorlalte categorii  $li$  se alocă codurile de doi biți 10 și 11, respectiv. Compresia este realizată datorită faptului că  $mn$  biți care ar fi în mod normal utilizați pentru a reprezenta fiecare arie constantă sunt înlocuiți cu cuvinte de cod de 1 bit sau 2 biți. Desigur, codul alocat categoriei cu intensitate mixtă este utilizat ca prefix, urmat de secvențele de  $mn$  biți pentru fiecare bloc.

Atunci când se pune problema memorării sau a transmiterii unor imagini reprezentând predominant documente tip text, o metodă relativ simplă este de a coda ariile compacte albe ca 0 și alte blocuri (inclusiv cele de negru compact) cu un 1 urmat de o structură de biți corespunzătoare blocului. Această metodă, numită *Omiterea blocurilor albe (White Block Skipping = WBS)*, folosește avantajele date de posibilitatea anticipării tendințelor structurale din imaginea care urmează a fi comprimată. Deoarece puține blocuri compacte de negru sunt așteptate să apară, ele sunt grupate împreună cu zonele de intensitate mixtă, permițând astfel utilizarea unui cuvânt de cod de un bit pentru blocurile albe cu probabilitate mare de apariție.

O modificare interesantă a acestei proceduri (cu blocuri de mărime  $1 \times n$ ) este de a coda liniile complet albe cu "0" și toate celelalte linii cu "1" urmat de secvența de cod *WBS* normal.

O altă modificare constă în utilizarea unei proceduri iterative în care imaginea binară este descompusă în subblocuri succesive din ce în ce mai mici. Pentru blocuri bidimensionale, o imagine complet albă este codată cu 0, iar toate celelalte imagini sunt descompuse în subblocuri cărora li se alocă prefixul 1 și sunt codate similar. Prin urmare, dacă un subbloc este complet alb el este reprezentat prin prefixul 1, indicând că este subbloc din prima iterație, urmat de 0, indicând că este un subbloc complet alb. Dacă subblocul nu este complet alb, procesul de descompunere este repetat până ce se ajunge la o mărime de subbloc predefinită și codul este fie 0 (dacă este complet alb) fie un 1 urmat de secvența de biți corespunzătoare blocului.

### 2.3.3. Codarea "run-length"

Codarea, "run-length" este o tehnică de codare relativ simplă în care sunt comparate nivelurile de gri ale elementelor de imagine adiacente de-a lungul unei linii de imagine. Metoda poate fi utilizată atât unidimensional cât și bidimensional.

Varianta unidimensională a codării "run-length" poate fi utilizată pentru reprezentarea conturilor, împreună cu codurile lanț, așa cum voi propune în algoritmul meu, descris în capitolul 6.

### 2.3.3.1. Codarea "run-length" unidimensională

În cazul codării "run-length" unidimensionale, secvența nivelurilor de gri ale elementelor de imagine este transformată într-o secvență de perechi de numere întregi  $(G_i, L_i)$ , unde  $G_i$  reprezintă nivelul de gri al direcției  $i$ , iar  $L_i$  reprezintă lungimea direcției  $i$  (adică numărul de pixeli adiacenți care au același nivel de gri).

Procedura de codare presupune următoarele etape:

- Se începe cu elementul de imagine din marginea stângă a liniei și se pune  $G_1$  egal cu nivelul de gri al acestui element, iar  $L_1$  egal cu numărul pixelilor cu același nivel de gri  $G_1$ ;
- La prima schimbare a nivelului de gri se pune  $G_2$  egal cu nivelul de gri al celei de-a doua direcții, iar  $L_2$  egal cu numărul pixelilor având nivelul de gri  $G_2$ ;
- Procedura se repetă până se ajunge la sfârșitul liniei respective, apoi se trece la linia următoare.

Se remarcă faptul că numărul direcțiilor de pe o linie este cuprins între 1 (toate elementele de imagine de pe linie au același nivel de gri) și  $N$  (nu există două elemente de imagine adiacente având același nivel de gri). Lungimile corespunzătoare ale "direcțiilor" sunt cuprinse între  $N$  și 1. Dacă lungimile sunt codate binar natural, atunci sunt necesari  $\log_2 N$  biți. Dar, deoarece cele mai multe lungimi sunt semnificativ mai mici decât  $N$ , nu este eficientă codarea în cod natural. Se poate utiliza spre exemplu codul  $B$ . În cazul imaginilor binare este posibil ca, pentru codul  $B$ , bitul de continuare  $C$  să semnifice chiar nivelul de gri (de exemplu, alb  $C=1$  și negru  $C=0$ ), iar biții de informație să fie folosiți pentru codarea lungimii "direcției".

### 2.3.3.2. Codarea "run-length" bidimensională

Principiul codării "run-length" bidimensionale este ilustrat în figura 2.5.

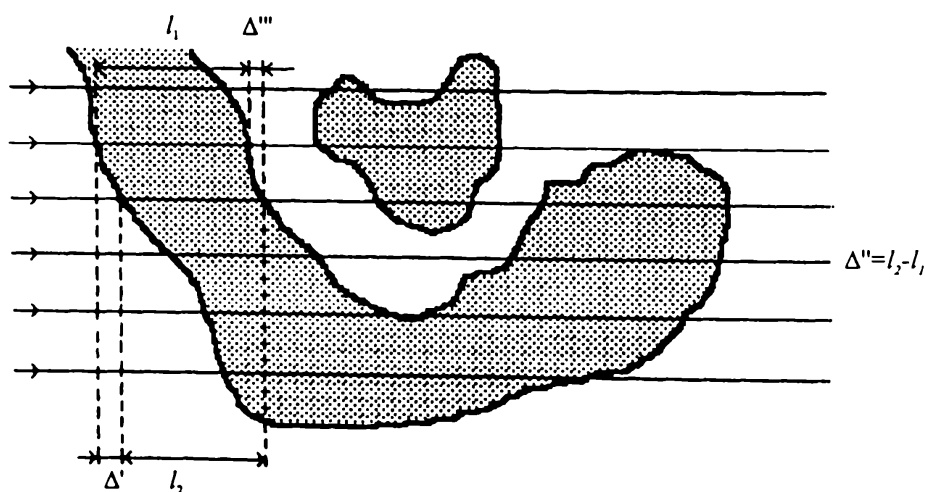


Fig. 2.5. Codarea "run-length" bidirecțională



O primă variantă de realizare constă în transformarea unui bloc de elemente de imagine într-o secvență de numere întregi  $(\Delta', \Delta'')$ , unde  $\Delta'$  reprezintă diferența între începuturile "direcției" pe două linii succesive, iar  $\Delta''$  reprezintă diferența dintre lungimile "direcției" pe două linii succesive. La secvența  $(\Delta', \Delta'')$  se adaugă indicatorii pentru începutul și sfârșitul zonei.

O altă variantă de realizare constă în transmiterea secvenței  $(\Delta'', \Delta''')$ , unde  $\Delta'''$  reprezintă diferența între sfârșiturile "direcției" pe două linii succesive.

Codurile  $B$  se pot utiliza pentru codarea parametrilor transmiși conform procedurii "run-length" bidimensionale.

Procedura "run-length" este foarte utilă pentru codarea imaginilor binare, dar în cazul unor tehnici combinate poate fi utilizată și pentru codarea imaginilor pe nuanțe de gri, așa cum voi propune în capitolul 6.

### 2.3.4. Codarea pe plane de biți

Codarea planelor de biți este o tehnică de compresie a informației video digitale bazată pe conceptul descompunerii unei imagini (monocrome sau color) codate cu mai mulți biți/pixel într-o serie de imagini binare și compresării fiecărei imagini prin folosirea unui cod cu lungime variabilă. În continuare voi prezenta cele mai utilizate modalități de descompunere și voi aminti principalele modalități de compresie. O versiune modificată a acestei tehnici de codare o voi utiliza în algoritmul pe care îl propun, algoritmul care este descris în capitolul 6, iar rezultatele sunt prezentate în capitolul 7.

#### 2.3.4.1. Descompunerea planelor de biți

Nivelurile de gri ale unei imagini reprezentate cu  $m$  biți/pixel pot fi caracterizate sub forma unui polinom binar:

$$a_{m-1}2^{m-1} + a_{m-2}2^{m-2} + \dots + a_12^1 + a_02^0. \quad (2.11)$$

Bazat pe această proprietate, o metodă simplă de descompunere a imaginii considerate într-un set de imagini binare este aceea de a separa cei  $m$  coeficienți polinomiali în  $m$  plane de biți de câte un bit. Planul de biți de ordinul zero este generat prin gruparea biților  $a_0$  ai fiecărui pixel, iar planul de biți de ordinul  $(m-1)$  este format prin gruparea biților  $a_{m-1}$  ai fiecărui pixel. În general, fiecare plan de biți este numerotat de la 0 la  $(m-1)$  și este construit prin gruparea biților corespunzători (sau a coeficienților polinomiali corespunzători) pentru fiecare pixel al imaginii originale. Dezavantajul inerent al acestei metode constă în faptul că modificări mici în nivelul de gri pot conduce la schimbări importante în complexitatea planelor de biți. De exemplu, dacă un pixel cu intensitatea 127 (0111111) este adiacent cu un pixel având intensitatea 128 (1000000) atunci fiecare plan de biți va conține o tranziție  $0 \rightarrow 1$  sau  $1 \rightarrow 0$ .

O modalitate alternativă de descompunere, care reduce efectul variațiilor mici ale nivelului de gri, este aceea de a reprezenta imaginea printr-un cod Gray pe  $m$  biți. Codul Gray pe  $m$  biți  $g_{m-1}g_{m-2} \dots g_2g_1g_0$  care corespunde polinomului (2.11) poate fi calculat prin

$$g_i = a_i \oplus a_{i+1}, \text{ pentru } 0 \leq i \leq m-2 \text{ și } g_{m-1} = a_{m-1}. \quad (2.12)$$

Având proprietatea că două cuvinte de cod succesive diferă doar prin valoarea unui bit, mici schimbări ale nivelului de gri este puțin probabil să conducă la afectarea tuturor celor  $m$  plane de biți. Pentru exemplul considerat anterior, când nivelurile de gri 127 și 128 sunt adiacente, doar planul de biți de ordinul 7 va conține o tranziție 1→0 deoarece codurile Gray corespunzătoare pentru 127 și 128 sunt 11000000 și 01000000, respectiv.

*Notă:* Aici,  $\oplus$  semnifică operațiunea logică SAU-EXCLUSIV, în timp ce în partea principală a acestei teze  $\oplus$  va semnifica operatorul morfologic de *dilatare*, care va fi introdus în capitolul 4.

#### 2.3.4.2. Codarea planelor de biți

După ce a fost realizată descompunerea imaginii în plane de biți, urmează procesul de codare a acestor plane cu scopul reducerii cantității totale de informație care urmează a fi transmisă. Ținând cont de importanta redundanță existentă în imagini, este sigur că nu toate planurile de biți vor avea aceeași complexitate. Întotdeauna planurile de biți de ordin superior sunt mult mai puțin complexe decât planurile de biți de ordin inferior, care conțin informația legată de detaliile fine ale imaginii.

Pentru planele de biți de ordin superior vor exista suprafețe mari cu luminanța constantă, motiv pentru care utilizarea unor metode de codare a ariilor constante, a conturilor sau a codării run-length unidimensionale sau bidimensionale sunt foarte potrivite, conducând la reducerea semnificativă a numărului mediu de biți/pixel utilizați pentru codare.

Tehnici de codare cu lungime variabilă pot fi utilizate în conjuncție pentru codarea tuturor planelor de biți.

Literatura de specialitate [69] menționează de asemenea faptul că planele de biți codate Gray sunt mai puțin complexe decât cele codate binar, rezultând o eficiență mai mare în aplicarea metodei.

#### 2.4. Codarea nereversibilă a sursei

Spre deosebire de codarea reversibilă a sursei la care se dorește reproducerea cu precizie a semnalului MIC original, au fost dezvoltate o serie de tehnici care realizează o compresie a semnalului MIC pe seama unei ușoare pierderi de informație înaintea transmisiei. Deși aceste tehnici sunt mai eficiente, în cazul sistemelor video se folosesc de multe ori combinații între codări reversibile și nereversibile.

În acest capitol voi descrie pe scurt cele mai reprezentative tehnici de codare nereversibilă. Deși algoritmul pe care îl propun în prezenta lucrare nu face apel la tehnici din această categorie, ele sunt utile din punct de vedere comparativ, deoarece stau la baza principalelor standarde de compresie numerică a imaginilor. În capitolul 7, voi compara rezultatele aplicării algoritmului pe care îl propun, cu rezultatele obținute prin utilizarea unor proceduri standard de compresie, utilizând aceleași imagini test.

Din rândul tehnicilor de codare nereversibilă cele mai adaptate pentru transmiterea imaginilor sunt: codarea cu predicție și codarea cu transformare.

### 2.4.1. Codarea cu predicție

După cum s-a arătat deja, puternica corelație existentă între elementele de imagine adiacente face să nu fie necesară transmiterea informației complete despre fiecare element. Se poate transmite eșantionul pentru un element și se pot estima celelalte prin predicție pe baza unor legi statistice, utilizând dispozitive potrivite la recepție. Totuși, indiferent cât de perfectă ar fi procedura de stabilire a corelației statistice într-o imagine, predicția elementelor de imagine de la un set de eșantioane va introduce o eroare datorată distribuției aleatoare a luminanței și crominanței într-o imagine. Această eroare trebuie estimată și corectată pentru fiecare element, deoarece în caz contrar imaginea reconstituită la recepție poate diferi foarte mult de scena originală.

Principiul codării cu predicție se poate explica pe baza schemei din figura 2.6.

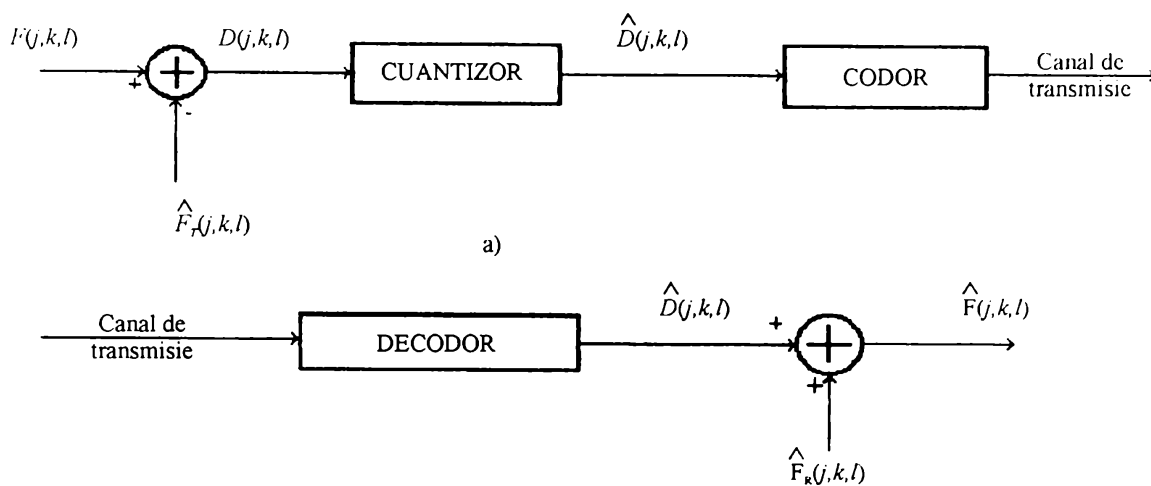


Fig. 2.6. Codarea cu predicție

La transmisie (figura 2.6.a), valoarea fiecărui eșantion de imagine  $F(j,k,l)$  se estimează pe baza cunoașterii valorilor eșantioanelor anterioare. Valoarea estimată  $\hat{F}_T(j,k,l)$  se scade din valoarea reală  $F(j,k,l)$  a eșantionului de imagine și se obține diferența:

$$D(j,k,l) = F(j,k,l) - \hat{F}_T(j,k,l) \quad (2.13)$$

Această diferență se cuantizează, se codează și se transmite prin canalul de comunicație.

La recepție (figura 2.6.b), în urma procesului de decodare, se obține diferența  $\hat{D}(j,k,l)$ , care se adaugă la valoarea estimată la recepție  $\hat{F}_R(j,k,l)$  a eșantionului de imagine, pentru a se obține valoarea reconstituită  $\hat{F}(j,k,l)$  a acestuia:

$$\hat{F}(j,k,l) = \hat{D}(j,k,l) + \hat{F}_R(j,k,l) \quad (2.14)$$

Codarea cu predicție se bazează pe folosirea atât a proprietăților statistice ale semnalului de imagine cât și a proprietăților sistemului de percepție vizuală. Detalii referitoare la tipurile de predicție, eroarea de predicție și optimizarea predicției pot fi găsite în [156] și [181].

### 2.4.2. Codarea cu modulație delta

*Modulația delta* reprezintă cea mai simplă formă de realizare a codării cu predicție. În acest caz, eroarea de predicție este cuantizată pe numai două niveluri, adică se ia în considerație doar semnul erorii. În cazul modulației delta predictorul este un integrator care primește la intrare o sumă liniară de semnale de eroare  $\pm \Delta$ , care sosesc cu frecvența de eșantionare. Principiul codării cu modulație delta este prezentat în figura 2.7.

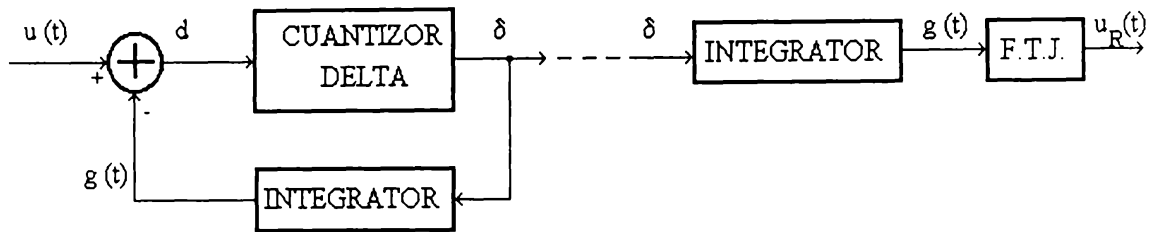


Fig.2.7. Principiul codării cu modulație delta

La transmisie, deoarece semnalele de eroare rămân neschimbate pe durata unui interval de eșantionare, însumarea lor dă naștere unei forme de undă în trepte,  $g(t)$ , care estimează semnalul inițial  $u(t)$ . Scăderea sa din semnalul original dă naștere, la ieșirea sumatorului, unui semnal diferență:

$$d_n = u(t_n) - g(t_n) \quad (2.15)$$

unde  $t_n = n \cdot T_E$  și  $T_E$  este durata dintre eșantioane. Acest semnal de diferență este limitat de către cuantizor la două niveluri, astfel încât semnalul de la ieșirea cuantizorului va fi sub formă de impulsuri:

$$\delta_n = \Delta \cdot \text{sgn } d_n = \begin{cases} +\Delta, & \text{dacă } d_n > 0 \\ -\Delta, & \text{dacă } d_n < 0 \end{cases} \quad (2.16)$$

Semnalul  $\delta$  se transmite în canalul de comunicație și în același timp se aplică integratorului la ieșirea căruia se obține semnalul de predicție  $g(t)$  sub formă de trepte:

$$g(t_n) = \sum_{k=-\infty}^{n-1} \delta_k \cdot \sigma(t_n - kT_E) \quad (2.17)$$

unde  $\sigma(t)$  este funcția treaptă unitate.

La această tehnică de codare este posibilă apariția distorsiunilor. O discuție pe această temă poate fi găsită în [156].

### 2.4.3. Codarea cu modulația diferențială a impulsurilor în cod

Codarea cu modulația diferențială a impulsurilor în cod (MDIC) exploatează densitatea de putere neuniformă a semnalului diferențial, prin cuantizarea grosieră a semnalelor rare și de o manieră mai fină a semnalelor mai probabile. Utilizarea cuantizării neuniforme se poate justifica statistic prin faptul că minimizarea erorii medii pătratică de cuantizare conduce la obținerea unei caracteristici de cuantizare neuniforme

[265]. Pentru obținerea unor performanțe superioare, în practică se utilizează un sistem de prelucrare cu reacție, în care semnalul diferențial este trecut printr-o buclă închisă.

În figura 2.8 este prezentată schema bloc de principiu a unui sistem MDIC.

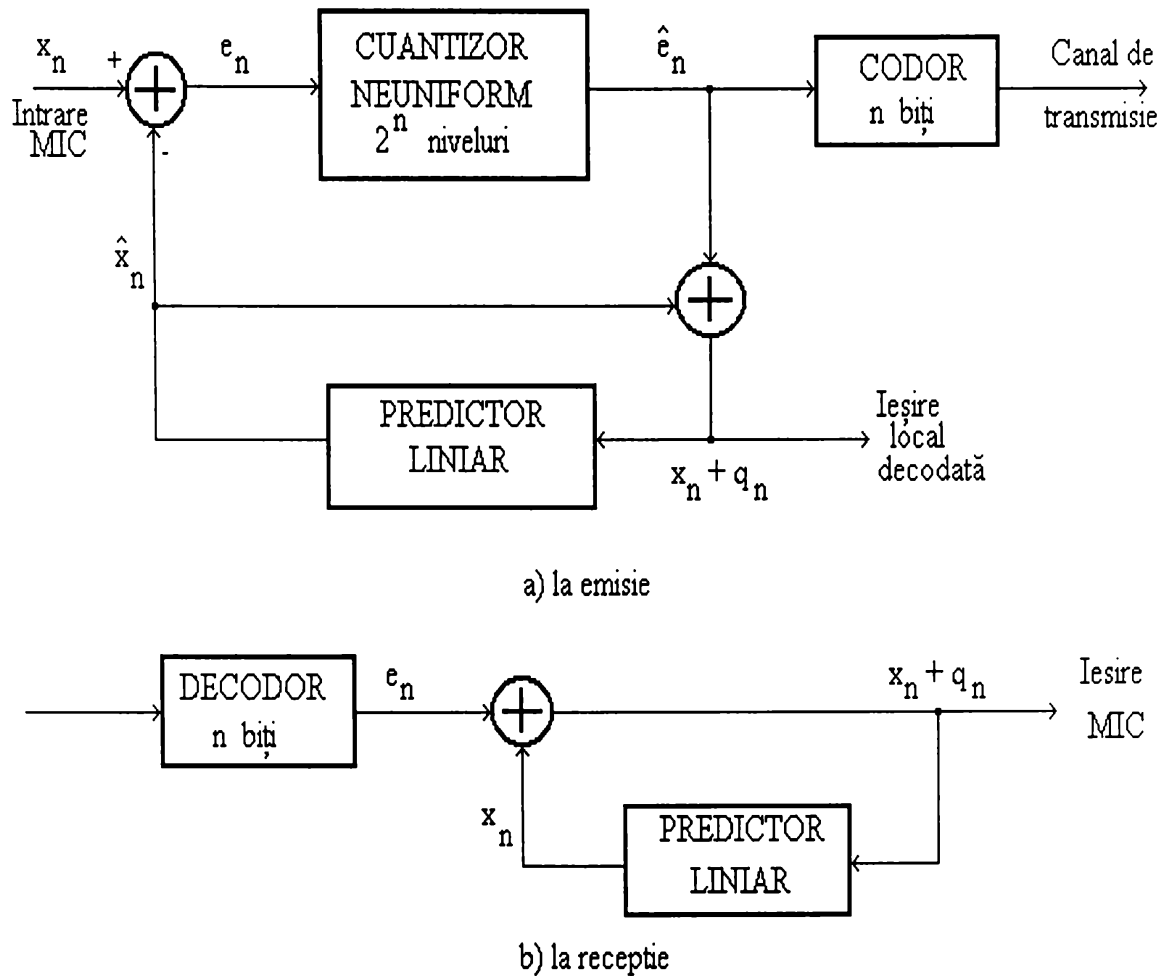


Fig. 2.8. Principiul codării MDIC

Discuții cu privire la sistemul de predicție, legea de cuantizare, pasul de cuantizare și erorile metodei se pot găsi în [91], [156] și [181].

#### 2.4.4. Codarea cu transformare

Codarea cu transformare este o codare de grup. Imaginea inițială de dimensiuni  $N \times N$  este împărțită în blocuri de imagine, de exemplu de formă pătrată, fiecare conținând  $n \times n$  eșantioane. Dacă fiecare eșantion este cuantizat cu  $2^m$  niveluri, numărul de biți necesari pentru transmisie este egal cu  $m \cdot n^2$ . Se urmărește realizarea unei transformări a blocurilor de imagine, din spațiul imaginii într-un alt spațiu, astfel încât printr-o codare corespunzătoare în noul spațiu să obținem o reducere a cantității de informație de  $m \cdot n^2$  biți.

Transformarea dintr-un spațiu în altul trebuie să conducă la slăbirea corelației dintre eșantioanele imaginii inițiale și, dacă este posibil, chiar anularea acesteia,

ajungând ca noile variabile să devină independente, iar distribuția dispersiei lor să fie foarte neomogenă. Existența unor asemenea transformări este confirmată de observația că energia în spectrul imaginii tinde să fie concentrată în jurul unui număr relativ mic de componente, astfel încât spectrul imaginii poate fi codat mai eficient decât imaginea însăși.

Să considerăm un bloc de imagine format din  $n \times n$  elemente discrete, fiecare element fiind caracterizat prin nivelul luminanței sale,  $L_{jk}$ . Fiecărui element îi corespunde un eșantion discret în semnalul video. Setul acestor eșantioane poate fi reprezentat printr-o matrice de distribuție a luminanței,  $X$ , care are elementele  $x_{jk}=L_{jk}$ . Imaginea originală este transformată într-un set de eșantioane care sunt caracterizate prin intermediul unei alte matrici,  $Y$ . Fiecare eșantion din noul set aduce o altă contribuție în conținutul final al imaginii. În acest caz va fi posibilă selectarea celor mai importante eșantioane în ceea ce privește conținutul de informație și transmiterea lor în modul cel mai potrivit. Celelalte eșantioane pot fi neglijate sau transmise cu un număr redus de biți.

$$X = \begin{bmatrix} L_{11} & L_{12} & L_{13} & \dots & L_{1n} \\ L_{21} & L_{22} & L_{23} & \dots & L_{2n} \\ L_{31} & L_{32} & L_{33} & \dots & L_{3n} \\ \dots & \dots & \dots & \dots & \dots \\ L_{n1} & L_{n2} & L_{n3} & \dots & L_{nn} \end{bmatrix} \quad Y = \begin{bmatrix} Y_{11} & Y_{12} & Y_{13} & \dots & Y_{1n} \\ Y_{21} & Y_{22} & Y_{23} & \dots & Y_{2n} \\ Y_{31} & Y_{32} & Y_{33} & \dots & Y_{3n} \\ \dots & \dots & \dots & \dots & \dots \\ Y_{n1} & Y_{n2} & Y_{n3} & \dots & Y_{nn} \end{bmatrix}$$

Schema bloc a codării cu transformare este prezentată în fig. 2.9

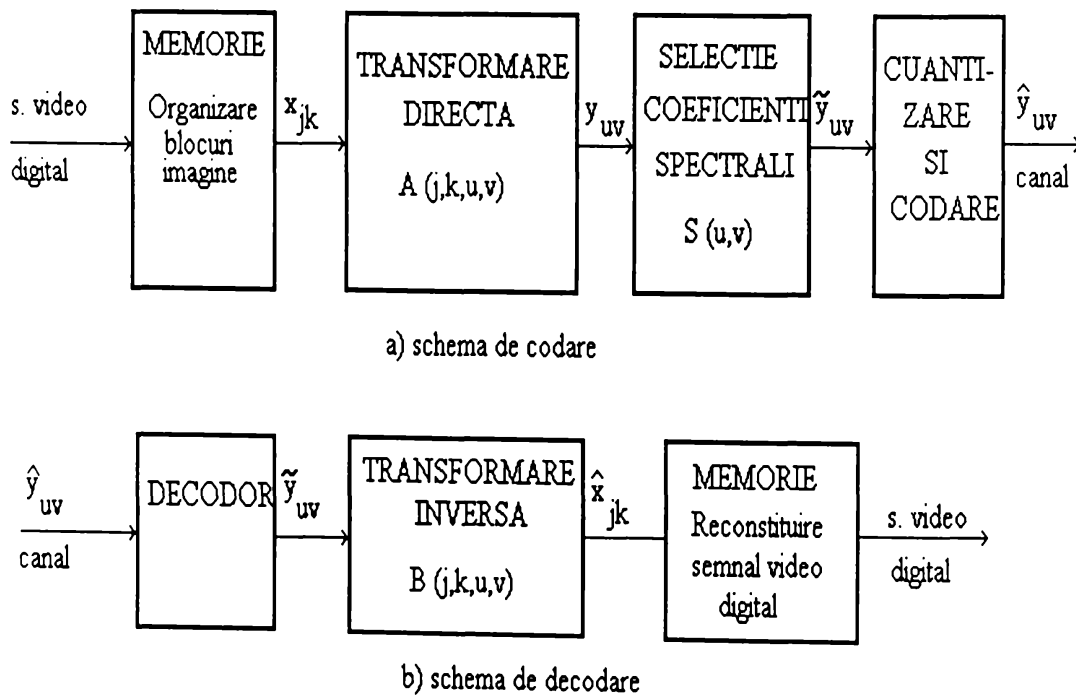


Fig. 2.9. Principiul codării cu transformare

La transmisie (fig. 2.9.a), prima etapă constă în stocarea eşantioanelor semnalului de imagine digital pentru a putea fi reorganizate în blocuri de imagine mai mici, de regulă de formă pătratică. Urmează procesul de transformare directă a blocurilor de imagine, printr-o lege de transformare bidimensională, unitară, ortogonală și inversabilă. Dacă  $x_{jk}$  ( $j, k = 1, 2, \dots, n$ ) sunt eşantioanele imaginii originale și  $y_{uv}$  ( $u, v = 1, 2, \dots, n$ ) sunt eşantioanele imaginii transformate, atunci operația de transformare se poate descrie sub forma:

$$y_{uv} = \sum_{j=1}^n \sum_{k=1}^n x_{jk} \cdot A(j, k, u, v) \quad (2.18)$$

unde  $A(j, k, u, v)$  este nucleul transformării.

Relația (2.18) poate fi scrisă și sub forma matricială:

$$Y = A \cdot X \quad (2.19)$$

unde  $A$  este matricea transformării directe. Matricea  $Y$  este transformata matricii  $X$ . Elementele sale sunt combinații liniare ale elementelor matricii  $X$ .

Următoarea operație constă în selecția coeficienților spectrali pentru a fi cuantizați și codați în vederea transmiterii prin canalul de comunicație. Există două metode de selecție a coeficienților spectrali:

- Metoda zonală*, în care se selectează coeficienții spectrali ai unei zone prestabilite, cum ar fi cei din spectrul de joasă frecvență. Coeficienții selectați sunt cuantizați și codați astfel încât numărul nivelurilor de cuantizare se alege proporțional cu dispersia estimată a coeficienților selectați.
- Metoda cu prag*, în care se selectează coeficienții care depășesc un anumit prag prestabilit.

Este de remarcat faptul că reducerea ratei de transmisie a semnalului video se realizează, în cazul codării cu transformare, tocmai prin eliminarea anumitor eşantioane din spațiul transformat, pe baza proprietăților statistice ale imaginii. În general, distribuția ariilor care urmează să fie transmise în transformată este cea prezentată în fig. 2.10. Pentru o largă clasă de imagini, frecvențele spațiale joase au amplitudini maxime. Corespunzător, informația este purtată în spațiul transformat de elementele matricii transformate  $Y$  care au indici mici de linie și coloană (colțul din stânga-sus din fig. 2.10). Pe de altă parte, definiția imaginii este determinată de frecvențele spațiale înalte (elementele din dreapta-sus și stânga-jos în fig. 2.10, care trebuie și ele transmise). Elementele rămase în matricea transformată pot fi neglijate, fără a fi afectată calitatea imaginii.

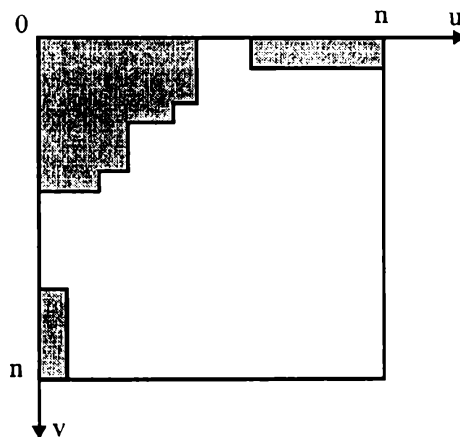


Fig. 2.10. Distribuția ariilor relevante pentru codarea cu transformare

Se observă din fig.2.10 că reducerea în cantitatea de informație care trebuie transmisă este destul de importantă. Pentru obținerea unei calități satisfăcătoare a imaginii este în general suficient să se transmită în medie 2 biți/eșantion, imaginea obținută fiind comparabilă cu cea din cazul MIC cu 7-8 biți/eșantion.

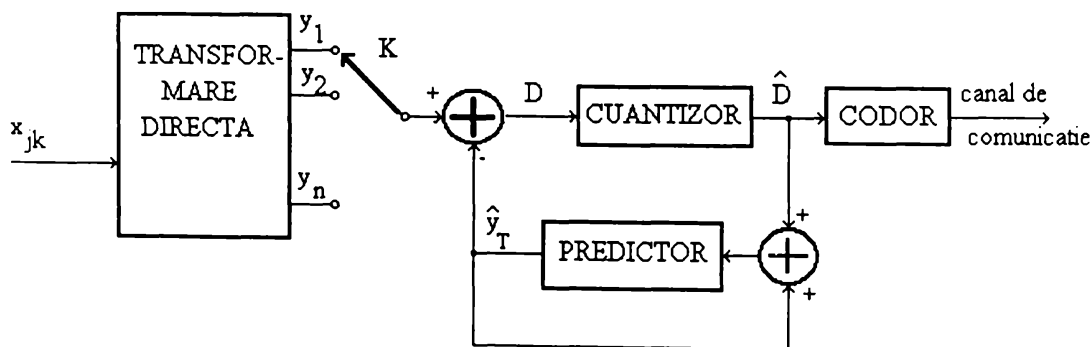
La recepție sunt realizate operațiile inverse, conform celor prezentate în fig. 2.9.b), unde  $B(j, k, u, v)$  reprezintă nucleul transformării inverse.

Au fost dezvoltate diferite legi de transformare, cu complexități de implementare diferite și cu diverse viteze de calcul. Discuții referitoare la acest subiect pot fi găsite în [39] și [66].

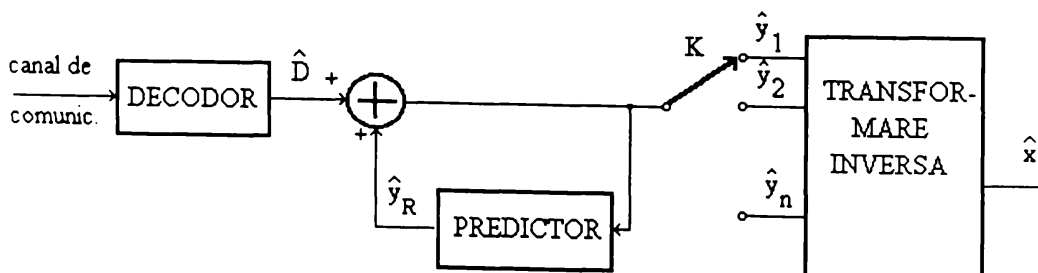
#### 2.4.5. Codarea hibridă

Sistemele de codare cu predicție și cu transformare au o mare eficiență, ambele bazându-se pe prelucrarea eșantioanelor corelate ale imaginii cu scopul de a obține semnale cât mai decorelate, care sunt apoi cuantizate și codate în vederea transmiterii pe canalele digitale de comunicație. Ambele metode prezintă avantaje și dezavantaje:

- codarea cu transformare asigură imagini de calitate la viteze scăzute de transmisie;
- codarea cu transformare este mai puțin sensibilă la schimbarea proprietăților statistice ale imaginii decât codarea MDIC cu predicție;
- codarea MDIC cu predicție dă rezultate foarte bune la viteze mari de transmisie;
- aparatura folosită pentru codarea MDIC este mai simplă decât cea folosită pentru codarea cu transformare.



a) la transmisie



b) la recepție

Fig. 2.11. Principiul codării hibride



Reunirea selectivă a celor două metode de prelucrare a condus la realizarea codării hibride (Habibi, în anul 1974), care rezolvă problema codării imaginilor de televiziune într-un mod mai eficient decât în cazul folosirii separate a celor două metode.

Schema bloc de principiu a codării hibride este prezentată în fig. 2.11. Se observă că prima etapă în realizarea codării hibride constă în efectuarea transformării directe. Coeficienții transformării sunt stocați în vederea prelucrării serie în cadrul codării MDIC cu predicție.

Există două principale moduri de implementare a metodei de codare hibridă: codarea hibridă spațială și codarea hibridă temporală.

În cazul codării hibride spațiale se folosește corelația pe orizontală și pe verticală a eșantioanelor de imagine. Eșantioanele de imagine se împart în blocuri de dimensiune  $1 \times n$ , de-a lungul liniilor de rastru. Procesul de codare cu transformare are loc în direcția orizontală, în interiorul fiecărui bloc, iar codarea MDIC cu predicție are loc în direcție verticală.

În cazul codării hibride temporale se folosește atât corelația spațială, cât și cea temporală a eșantioanelor de imagine. Acestea se împart în blocuri de dimensiune  $n \times n$  în interiorul unui cadru de imagine. Codarea cu transformare are loc în spațiul fiecărui bloc (codare intracadru), iar codarea MDIC cu predicție se folosește pentru codarea imaginii în direcția temporală (codare intercadre).

### Capitolul 3

## Metode de codare a imaginilor orientate pe contur și formă

### 3.1. Definirea problemei

După ce o imagine a fost segmentată pe regiuni, mulțimile de pixeli obținute în urma segmentării sunt de obicei reprezentate și descrise într-o formă adecvată pentru prelucrarea ulterioară. În general, reprezentarea regiunilor implică alegerea între două variante:

- reprezentarea unei regiuni prin caracteristicile sale externe (prin suprafața sa de separație);
- reprezentarea unei regiuni prin caracteristicile sale interne.

Alegerea unei metode de reprezentare este totuși doar o parte a sarcinii de a face datele disponibile cât mai utile pentru prelucrarea pe calculator. Următoarea etapă o constituie descrierea regiunii prin suprafața sa de separație, suprafață caracterizată prin mărimi ca lungime, orientare a liniilor drepte ce unesc puncte extreme și număr de concavități ale suprafeței de separație.

În general, o alegere a reprezentării caracteristicilor externe are loc atunci când primordiale sunt caracteristicile formei. O reprezentare a caracteristicilor interne este preferată atunci când accentul se pune pe proprietățile de reflexivitate, ca de exemplu culoarea și textura. În ambele cazuri, mărimile alese ca descriptori trebuie să fie cât mai sensibile la variații ca: modificări de mărime, translație și rotație.

Algoritmul de codare implică cel puțin trei pași:

1. Definirea unei partiții;
2. Codarea și transmiterea partiției;
3. Codarea și transmiterea valorii pixelilor corespunzători fiecărei regiuni care definește partiția.

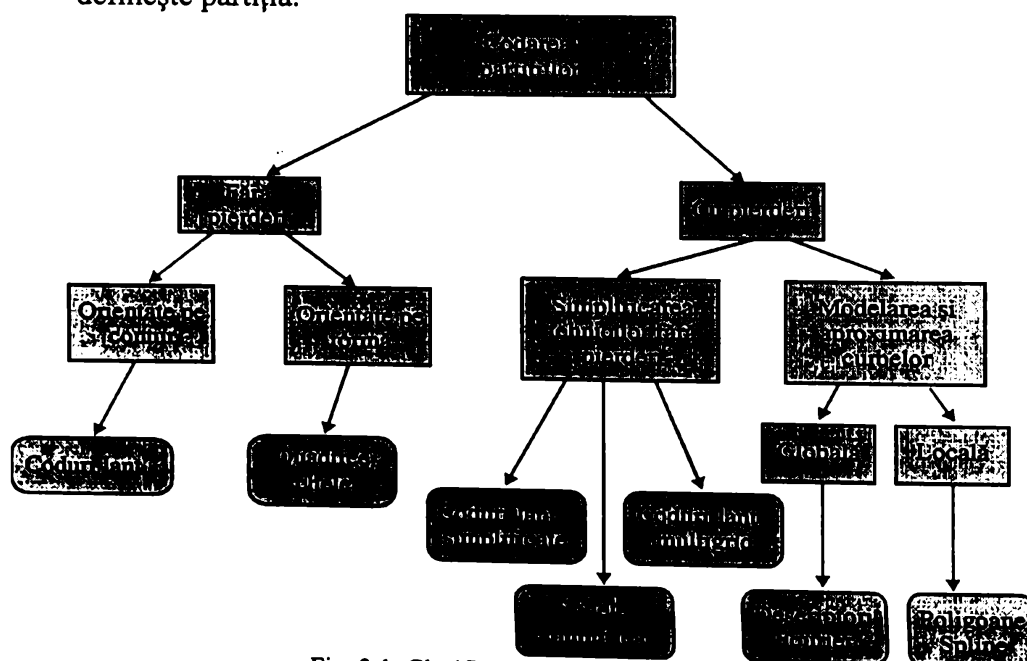


Fig. 3.1. Clasificarea codării partițiilor

Pentru definirea partițiilor pot fi utilizate mai multe criterii de segmentare. Printre cele mai des utilizate trebuie amintite omogenitatea valorii nivelului de gri, clasificarea în regiuni statice sau în mișcare precum și omogenitatea mișcării. Metode de segmentare a imaginilor sunt descrise în [201] și [278]. De asemenea sunt descrise posibile metode de segmentare orientate pe codarea imaginilor. În cele ce urmează mă voi referi doar la codarea imaginilor statice, bazat pe reprezentarea suprafeței de separație, metode care în general se regăsesc în literatură sub denumirea simplă de *codarea partițiilor*.

O clasificare a celor mai utilizate tehnici de codare a partițiilor este prezentată în fig. 3.1. În cazul tehnicilor de codare reversibilă pot fi utilizate două principale metode: *metode orientate pe contur*, care conduc la *coduri lanț* și *metode orientate pe formă*, care conduc la reprezentare de *schelete* și *descompuneri quadtree*. Tehnicile nereversibile pot fi obținute prin simplificarea sau extinderea unor metode reversibile de codare. Astfel pot fi obținute așa numitele *coduri lanț multirețea* (*multigrid chain codes*). În sfârșit, ultimul set de tehnici nereversibile se bazează pe aproximarea sau modelarea curburilor. În practică, aproximarea se poate face fie global pentru întreaga formă a unei regiuni (*descriptori Fourier*), fie local (aproximări *spline* sau *poligonale*).

### 3.2. Tehnici orientate pe contur

#### 3.2.1. Coduri lanț

Tehnicile orientate pe contur utilizează reprezentarea regiunilor prin suprafețele lor de separație, procedura de codare constând în urmărirea și codarea acestor suprafețe. Printre aceste tehnici se numără *codurile lanț*, care permit codarea fără pierderi a suprafețelor de separație. Această tehnică exploatează proprietatea că două puncte succesive de pe suprafața de separație, într-o rețea discretă, sunt vecine. De aceea, în vederea urmării unui contur, doar un număr mic de direcții sunt posibile. Un contur este codat prin înlanțuirea mișcărilor necesare pentru descrierea sa completă.

Tipic, această reprezentare se bazează pe o conectivitate de tip 4 sau 8 a segmentelor. Direcția fiecărui segment este codată prin utilizarea unei secvențe numerice de tipul celei descrise în figura 3.2.

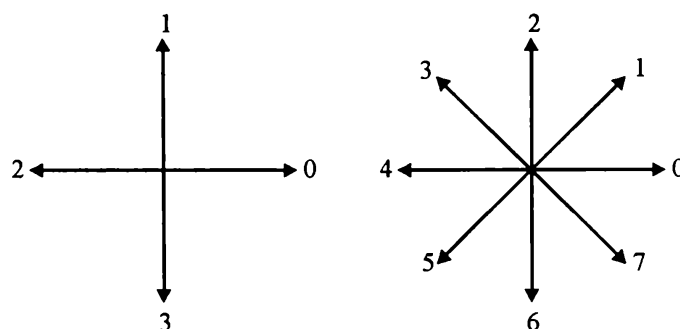


Fig. 3.2. Direcțiile pentru un cod lanț de tip 4 sau 8-direcțional

Imaginile digitale sunt în general obținute și prelucrate într-o rețea cu același pas după direcțiile  $x$  și  $y$ , astfel încât un cod lanț poate fi generat prin urmărirea unui contur, de exemplu, în direcția rotirii acelor de ceasornic și prin atribuirea unei direcții



formeze un număr întreg de valoare minimă. O altă posibilitate de normalizare o constituie utilizarea așa numitului cod lanț diferențial de ordinul unu, în locul codului propriu-zis. Codul diferențial de ordinul unu se obține simplu prin numărarea (în sensul invers acelor de ceasornic) a numărului de direcții care separă două elemente adiacente ale codului. De exemplu, codul diferențial de ordinul unu corespunzător codului lanț cu 4 direcții 10103322 este 3133030. Dacă tratăm codul ca pe o secvență circulară, atunci primul element al codului diferențial trebuie calculat prin utilizarea tranzițiilor între ultima și prima componentă a lanțului. Rezultatul este în acest caz 33133030. Normalizarea lungimii codului se poate face doar prin modificarea mărimii grilei de eșantionare.

Aceste normalizări sunt exacte doar dacă suprafețele de separație sunt ele însele invariante la rotație și schimbare a scalei, ceea ce în practică este rareori cazul. De exemplu, același obiect digitizat după două direcții diferite va avea în general forme diferite ale conturului, cu gradul de diferențiere proporțional cu rezoluția imaginii. Acest efect poate fi redus prin selectarea elementelor care definesc codul lanț, mai mari decât distanța dintre pixelii imaginii digitizate, sau prin orientarea grilei de reeșantionare de-a lungul principalelor axe ale obiectului care urmează să fie codat.

Reprezentarea prin coduri lanț propusă de Freeman nu include lungimile muchiilor în notația sa. Există totuși o posibilitate de a ține cont de această caracteristică în cadrul implementării. O versiune numită *cod lanț run-length* poate fi utilizată. Ea necesită reprezentarea explicită a lungimilor și direcțiilor muchiilor care formează suprafața de separație. Această schemă de codare utilizează o structură de date de tip listă.

Codul lanț run-length este o schemă de reprezentare a componentelor conectate prin punctele suprafeței lor de separație. Doar un punct de start al conturului este reprezentat explicit, restul punctelor suprafeței de separație fiind reprezentate prin deplasamentele succesive după una din cele opt direcții posibile. Dacă deplasamentele succesive sunt după aceeași direcție, ele sunt codate ca direcție urmată de numărul de deplasări după respectiva direcție. Notația formală utilizată pentru reprezentarea unei mulțimi de puncte conectate  $A$  este

$$A = \{(i_A, j_A)(d_1, n_1)(d_{l+1}, n_{l+1}) \cdots (d_m, n_m)\} \quad (3.1)$$

În relația (3.1),  $(i_A, j_A)$  reprezintă punctul de start al conturului, iar perechile ordonate  $(d_k, n_k)$  descriu fiecare deplasament. Numărul perechilor ordonate este egal cu numărul schimbărilor de direcție de deplasare. În aceste perechi,  $d_k \in \{d_0, d_1, \dots, d_7\}$  reprezintă direcțiile de deplasare, iar  $n_k$  reprezintă numărul de deplasări succesive în respectiva direcție.

Este evident că orice imagine binară care poate fi codată prin utilizarea codurilor lanț clasice poate fi de asemenea codată prin utilizarea versiunii run-length a codurilor lanț.

### 3.2.2. Algoritm de codare a conturilor prin coduri lanț

O imagine digitală monocromă poate fi considerată o funcție de două variabile. Variabilele sunt coordonatele spațiale, iar valoarea funcției este nivelul de gri în acel punct. Deoarece există un număr finit de niveluri de gri, putem considera funcția imagine ca o mulțime de platouri cu înălțimile egale cu nivelurile de gri corespunzătoare

(figura 3.4). Valorile spre negru ale luminanței corespund platourilor joase, iar valorile spre alb ale luminanței corespund platourilor înalte. Cunoașterea înălțimii, poziției și formei tuturor platourilor este echivalentă cu cunoașterea imaginilor monocrome respective. Prin urmare, dacă se reușește o codare eficientă a conturilor, prin reducerea imaginilor monocrome la o mulțime de contururi sau de platouri, se poate ajunge la o compresie importantă a datelor care reprezintă imaginea [65].

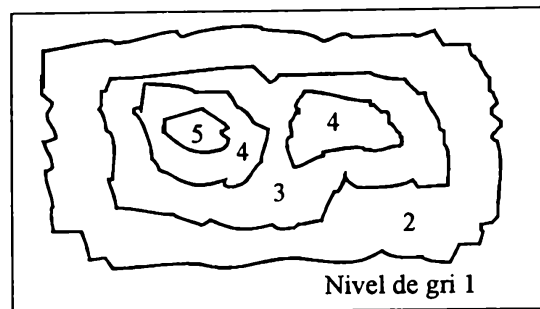


Fig. 3.4. Principiul codării conturilor

Algoritmul de codare a conturilor presupune parcurgerea a doi subalgoritmi: un algoritm *IP* (*Initial Points*) pentru localizarea de noi puncte inițiale corespunzând la noi contururi și un algoritm *T* (*Tracing*) pentru trasarea conturilor după localizarea lor. Cei doi algoritmi sunt utilizați secvențial. Algoritmul *IP* este utilizat pentru localizarea primului punct inițial, al primului contur și apoi algoritmul *T* este utilizat pentru trasarea conturului, algoritmul *IP* este utilizat din nou pentru localizarea punctului inițial pentru al doilea contur și algoritmul *T* pentru trasarea lui, ș.a.m.d. Algoritmul *IP* localizează toate conturile, nici unul nefiind însă localizat de două ori.

Algoritmul *T* trasează marginea exterioară al celui mai larg set de elemente având aceeași valoare a nivelului de gri ca și punctul inițial. Algoritmul *T* se încheie întotdeauna prin reîntoarcerea în punctul inițial. Pentru fiecare contur, algoritmul furnizează valoarea nivelului de gri, poziția punctului inițial și direcția de deplasare de-a lungul conturului.

#### Algoritmul *T*

Algoritmul *T* se folosește pentru trasarea unui contur după ce a fost localizat prin punctul său inițial *IP*. Trasarea unui contur implică definirea direcției de propagare între elementele de contur adiacente astfel încât să nu existe elemente situate în afara conturului și adiacente cu el, având aceeași valoare a nivelului de gri ca și elementele de pe contur. Regula utilizată pentru definirea direcției de propagare între elementele de contur adiacente este regula *LML* (*Left Most Looking* = regula privirii celei mai din stânga).

Conform regulii *LML* se verifică mai întâi elementul din stânga punctului inițial *IP* (relativ la direcția de intrare în punctul inițial, care se consideră întotdeauna pe orizontală de la stânga spre dreapta), dacă acest element are aceeași valoare ca și punctul inițial, atunci se consideră direcția de trasare a conturului spre el; dacă nu, atunci regula se repetă pentru elementele situate înaintea, în dreapta și în spatele punctului inițial *IP*. Procedura continuă, încheindu-se întotdeauna înapoi în punctul inițial *IP*. Dacă nici unul din elementele adiacente nu are aceeași valoare ca și punctul inițial *IP*, atunci conturul constă dintr-un singur punct (*IP*).

În figura 3.5 se prezintă contururile generate prin aplicarea repetată a algoritmului  $T$ , pentru o imagine cu patru niveluri de gri. Fiecare contur începe în punctul său inițial, indicat în figură.

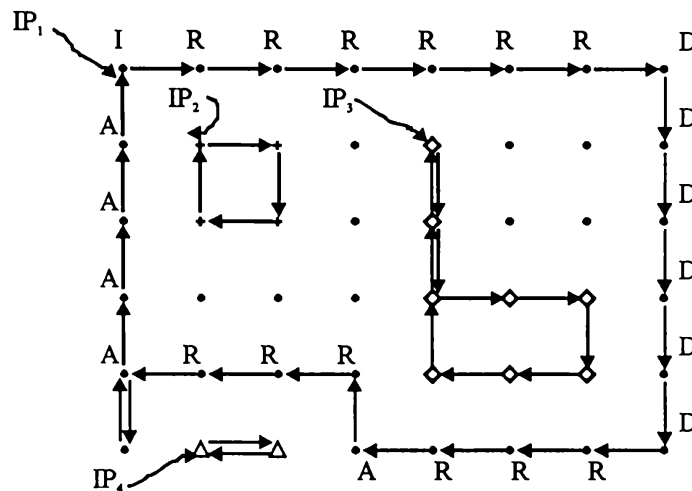


Fig. 3.5. Codarea contururilor

Algoritmul  $T$  realizează de asemenea și alocarea unui indicator din patru posibili ( $D$ ,  $A$ ,  $R$  și  $I$ ) pentru fiecare element de imagine. Acești indicatori sunt necesari pentru algoritmul  $IP$  de localizare a următoarelor puncte inițiale. De aceea, la memorarea imaginii, sunt alocați doi biți suplimentari pentru fiecare element de imagine pentru a memora indicatorul respectiv. Atunci când o arie de imagine este citită din memorie, fiecărui element  $i$  se atribuie inițial indicatorul  $I$ . Atunci când algoritmul  $T$  se deplasează de la un element la altul pentru trasarea conturului, el realizează totodată și înlocuirea fiecărui element de pe contur cu valoarea  $D$ ,  $A$  sau  $R$  conform regulii  $IA$  (*Indicator Assignment*).

Conform regulii  $IA$ , indicatorul atribuit fiecărui element de pe contur depinde de direcția propagării în acel element și de direcția propagării de la acel element, după cum este indicat în figura 3.6.

Direcție de propagare dinspre element Direcție de propagare înspre element	↓ sau →	↑ sau ←
↑ sau ←	A	R
↓ sau →	R	D

Fig.3.6. Alocarea indicatorilor pentru codarea contururilor

Prin unele elemente, după cum se vede și din figura 3.5, se trece de două ori în cursul trasării unui contur. Acest lucru este sesizat prin verificarea indicatorului său. La prima trecere indicatorul este întotdeauna  $I$ , iar la a doua trecere este întotdeauna diferit de  $I$ . În acest caz, la a doua trecere se atribuie din nou un indicator conform figurii 3.6. Indicatorul final se atribuie în funcție de indicatorii de la prima, și, respectiv, a doua trecere, conform figurii 3.7. Este de remarcat că prin nici un element nu se trece mai

mult decât de două ori. Singura excepție de la regula *IA* o constituie punctul inițial *IP*, al cărui indicator rămâne permanent *I*.

Indicator atribuit (prima trecere, a doua trecere)	(D, A)	(D, R)	(A, R)
	(A, D)	(R, D)	(R, A)
	(R, R)	(D, D)	(A, A)
Indicator atribuit în final	R	D	A

Fig.3.7. Alocarea indicatorilor la treceri multiple

Ca un exemplu, voi considera din nou conturul prezentat în figura 3.5. Pentru fiecare element al primului contur, algoritmul *T* aplică mai întâi regula *LML* pentru a determina direcția de propagare, apoi regula *IA* pentru a actualiza indicatorul atribuit fiecărui element. Indicatorul care va fi memorat pentru fiecare element al primului contur din figura 3.5 este prezentat alături de element. În acest moment, toate elementele care nu se găsesc pe acest contur rămân cu indicatorul *I*. După trasarea conturilor 2, 3 și 4, toate elementele de pe aceste conturi (cu excepția punctelor inițiale *IP*) vor avea atribuit unul din indicatorii *A*, *R* sau *D*. Punctele inițiale și toate elementele nesituate pe vreun contur rămân cu indicatorul *I*.

#### Algoritmul *IP*

Algoritmul *IP* se folosește pentru localizarea punctelor inițiale ale conturilor din imagine. Se utilizează o procedură de cercetare sistematică, începând cu elementul de imagine din stânga sus și terminând cu elementul de imagine din dreapta jos. Simultan cu trecerea de la un element la altul pe fiecare rând în vederea căutării punctelor inițiale *IP*, se completează o listă de comparare puncte *CPL* (*Comparison Point List*). Regulile de construire a acestei liste sunt următoarele:

- la începutul fiecărei linii din imagine lista este goală;
- la deplasarea de-a lungul unei linii se verifică indicatorul fiecărui element: dacă indicatorul elementului curent este *A*, atunci valoarea nivelului de gri al elementului este trecută în lista *CPL*; dacă indicatorul elementului este *D* atunci se șterge din listă ultima valoare de nivel de gri; dacă indicatorul este *I* sau *R*, atunci lista rămâne neschimbată;
- la sfârșitul parcurgerii fiecărei linii, lista *CPL* va fi goală, deoarece numărul intrărilor în listă este egal cu numărul ieșirilor de pe listă.

Un element este punct inițial *IP* dacă îndeplinește următoarele două cerințe:

- are indicatorul *I*;
- are valoarea nivelului de gri diferită de ultima valoare intrată pe lista *CPL*.

#### Realizarea codării

Algoritmul de codare a conturilor se realizează în următorul mod. Algoritmul este inițializat la elementul de imagine din stânga sus. Acest element este întotdeauna un punct inițial notat de exemplu *IP*<sub>1</sub>. În continuare, prin algoritmul *T* se trasează primul contur și se alocă indicatorii pentru elementele de pe contur. După revenirea în punctul



inițial  $IP_1$ , prin utilizarea algoritmului  $IP$  se găsește următorul punct inițial,  $IP_2$ . În continuare, se trasează al doilea contur, ș.a.m.d.

Fiecare contur este determinat în mod unic de următoarele elemente:

- nivelul său de gri;
- numărul rândului punctului său inițial;
- numărul coloanei punctului său inițial;
- o secvență de direcționale care indică direcțiile de propagare ale conturului.

O posibilitate simplă de codare a conturilor constă în utilizarea codului binar natural pentru specificarea valorii nivelului de gri, a rândului și coloanei punctului inițial și a codului lanț al lui Freeman pentru direcționale, ilustrat în figura 3.2.a). De exemplu, secvența de direcționale pentru al treilea contur din figura 3.5 este:

11110000111010010101

Pentru decodare este important să se cunoască când este atins sfârșitul fiecărui set de direcționale și începe un nou contur. Acest lucru nu necesită biți suplimentari, deoarece toate conturile trebuie să se termine înapoi în punctele lor inițiale. Trebuie memorate doar diferențele între numărul de direcționale dreapta și numărul de direcționale stânga, respectiv dintre numărul de direcționale sus și numărul de direcționale jos. Când ambele diferențe sunt nule, atunci suntem din nou în punctul inițial  $IP$ , prin urmare data care se memorează va corespunde unui nou contur.

Această regulă este valabilă pentru toate conturile de lungime mai mare sau egală cu două elemente. Pentru conturile formate dintr-un singur element (doar punctul inițial  $IP$ ) se utilizează ca primă direcțională valoarea 10, ceea ce semnifică o primă direcțională imposibilă, deoarece un punct inițial nu poate avea valoarea nivelului de gri egală cu cea a elementului din stânga sa.

Deoarece nivelurile de gri ale conturilor succesive sunt uzual foarte corelate, este posibilă o compresie prin codarea diferenței dintre nivelurile de gri ale conturilor succesive. Aceeași observație este valabilă și pentru numerele rândurilor și coloanelor punctelor inițiale.

O rezoluție ceva mai bună se poate obține uneori prin considerarea și a unor direcții diagonale, deci a opt diagonale diferite, codate tot cu ajutorul unui cod lanț Freeman, dar pe trei biți, conform figurii 3.2.b)

Numărul de biți necesari pentru codarea unei imagini particulare prin această metodă nu depinde doar de setul de coduri utilizate, ci și de numărul de contururi din imagine, care la rândul lui depinde de numărul de niveluri de gri. Astfel, pentru imagini care conțin un număr mare de contururi mici, eficiența metodei este scăzută, iar pentru imagini care conțin un număr mic de contururi mari, eficiența metodei este ridicată.

### 3.2.3. Aproximări poligonale

O suprafață de separație digitală poate fi aproximată cu o oarecare precizie printr-un poligon. În cazul unei curbe închise, aproximarea este exactă atunci când numărul de segmente ale poligonului este egal cu numărul de puncte ale suprafeței de separație, astfel încât fiecare pereche de puncte adiacente definește un segment al poligonului. În practică, scopul unei aproximări poligonale este de a reprezenta esențialul privind forma suprafeței de separație cu un număr cât mai mic de segmente poligonale. Această problemă nu este în general una trivială și se poate transforma rapid într-un algoritm complex de căutare iterativă, de lungă durată. Totuși, există mai multe

tehnicile de aproximare poligonală de complexitate moderată, care sunt bine adaptate aplicațiilor de prelucrarea imaginilor.

Una din metode constă în găsirea poligoanelor de perimetru minim. Să presupunem că includem suprafața de separație într-o mulțime de celule înlănțuite, așa cum se vede în figura 3.8.a). Poligonul de perimetru minim care se înscrie în geometria stabilită de aceste celule este prezentat în figura 3.8.b).

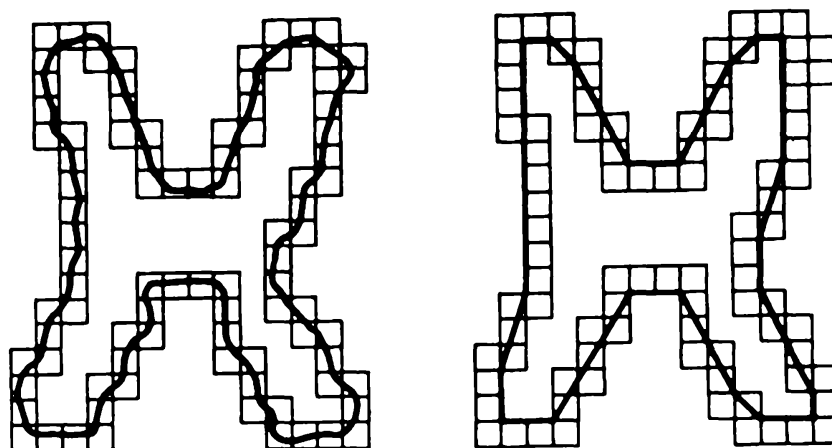


Fig. 3.8. Aproximarea prin poligoane de perimetru minim

O altă posibilitate este aceea de a utiliza tehnici de contopire, bazate pe criteriul de eroare. O tehnică constă în contopirea punctelor de-a lungul unui contur până ce eroarea pătratică determinată de linia care unește punctele contopite depășește o anumită valoare. Atunci când se ajunge la această condiție, parametrii liniei de unire sunt memorati, eroarea este readusă la zero și procedura se repetă prin contopirea a noi puncte de pe contur în aceleași condiții. La sfârșitul acestei proceduri, intersecțiile segmentelor liniare obținute formează vârfurile poligonului. Una dintre dificultățile aplicării acestei metode constă în faptul că, de obicei, aceste vârfuri nu corespund inflexiunilor conturilor, deoarece o nouă linie este trasată abia după ce se depășește o eroare prestabilită. Dacă, de exemplu, o linie dreaptă lungă a fost trasată și urmează un colț, un număr de puncte (care depinde de pragul de eroare prestabilit) sunt absorbite prin contopire înainte ca acest prag să fie depășit. Totuși, tehnici de divizare și contopire pot fi utilizate cu succes pentru a depăși această dificultate [278].

O tehnică de divizare a conturului constă în subdivizarea unui segment succesiv în două părți, până la satisfacerea unui criteriu prestabilit. De exemplu, un criteriu poate fi acela ca distanța perpendiculară maximă dintre un segment de contur și linia care unește punctele sale extreme să nu depășească o valoare de prag prestabilită. Dacă această valoare de prag este depășită, atunci punctul cel mai depărtat devine un vârf, subdivizând astfel segmentul inițial în două subsegmente. Această tehnică are avantajul de a căuta punctele de inflexiune proeminente ale conturului. De exemplu, figura 3.9.a) prezintă conturul unui obiect, iar figura 3.9.b) prezintă o subdivizare a acestui contur. Punctul marcat *c* prezintă cea mai mare distanță perpendiculară a segmentului superior față de linia *ab*. În mod similar, punctul *d* prezintă cea mai mare distanță perpendiculară a segmentului inferior. Figura 3.9.c) prezintă rezultatul utilizării acestei proceduri cu un prag stabilit la valoarea 0,25 din lungimea liniei *ab*. Deoarece nici un punct din noul

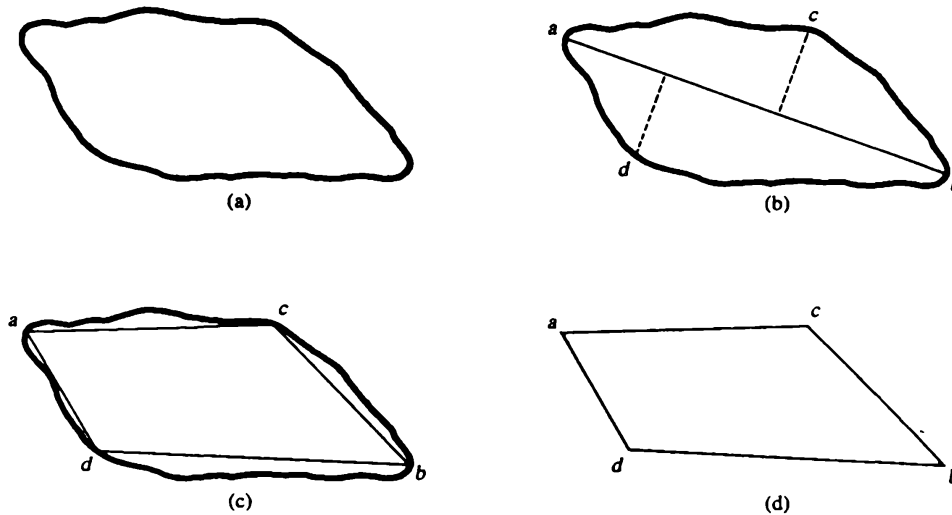


Fig. 3.9. Exemplu de divizare a conturului

contur segmentat nu are o distanță perpendiculară (referitoare la noile segmente de linie dreaptă) care să depășească pragul prestabilit, procedura se încheie la poligonul din figura 3.9.d).

### 3.2.4. Semnături

O semnătură este o reprezentare unidimensională a unei suprafețe de separație, care poate fi generată în mai multe moduri. Una dintre cele mai simple metode este de a trasa distanța dintre centroid și contur ca funcție de unghi, așa cum este ilustrat în figura 3.10. Ideea de bază pentru generarea semnăturii este de a reduce reprezentarea conturului la o funcție unidimensională, care este în principiu mai ușor de descris decât conturul original bidimensional.

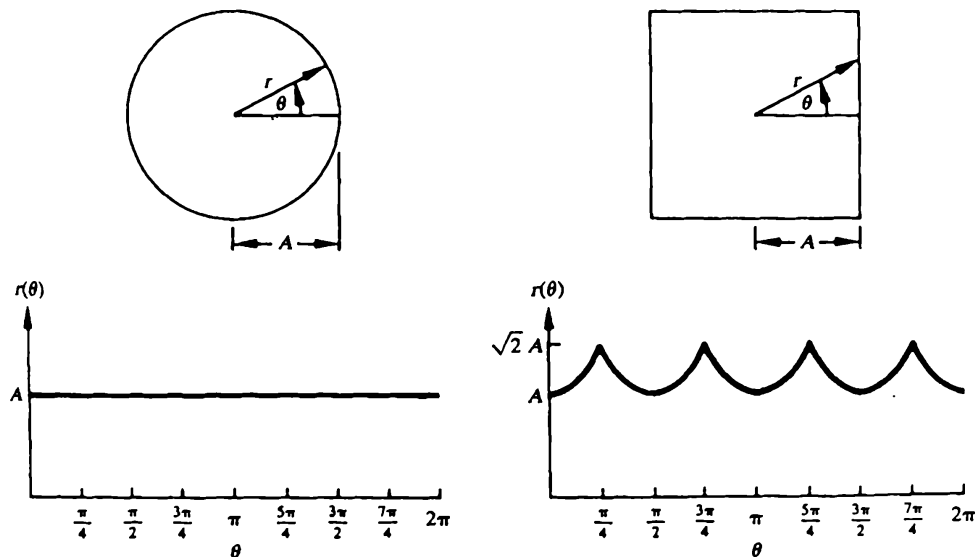


Fig. 3.10. Forme de contur și semnăturile corespunzătoare distanță-unghi

Semnăturile generate prin procedura descrisă sunt invariante la translație, dar ele sunt dependente de rotație și scalare. Normalizarea în raport cu rotația poate fi realizată prin găsirea unei modalități de selectare a punctului de start ca și cel mai depărtat punct de centroid, dacă acest punct este unic și independent de rotație.

Pe baza prezumpției de uniformitate la scalare față de ambele axe și a presupunerii că se utilizează intervale de eșantionare egale  $\theta$ , schimbările de dimensiune ale formei conduc la modificări de amplitudine a semnăturilor corespunzătoare. O metodă simplă de normalizare este aceea de a scala toate funcțiile astfel încât ele să varieze întotdeauna în același interval, de exemplu  $[0, 1]$ . Cel mai mare avantaj al acestei metode îl constituie simplitatea, dar ea prezintă și dezavantajul potențial că scalarea întregii funcții depinde de doar două valori: cea minimă și cea maximă. Dacă forma este afectată de zgomot, atunci această dependență poate fi o sursă de erori de la obiect la obiect. O altă metodă, cu o complexitate de calcul mai mare însă, constă în divizarea fiecărui eșantion prin varianța semnăturii, presupunând că această varianță nu este nulă, ca în cazul din figura 3.10.a). Utilizarea varianței semnăturii conduce la un factor de scalare variabil, invers proporțional cu schimbările de dimensiune. Indiferent însă de metoda utilizată, ideea de bază constă în eliminarea dependenței de mărime, dar cu păstrarea formei fundamentale.

Desigur că dependența distanță-unghi nu este singura modalitate de generare a unei semnături. O altă metodă constă, de exemplu, în utilizarea unghiului format de tangenta la contur într-un punct și o linie de referință, ca funcție de poziția punctului pe contur. Semnătura rezultată, deși diferită de curba  $r(\theta)$ , va conține și ea informațiile de bază referitoare la formă. De exemplu, liniile orizontale ale curbei vor corespunde liniilor drepte de-a lungul conturului, deoarece unghiul tangentei va fi constant pe aceste porțiuni. O variantă a acestei metode constă în utilizarea așa-numitei *funcții de densitate a pantei* ca și semnătură. Această funcție este de fapt o simplă histogramă a valorilor unghiului format de tangentă. În acest sens ea constituie o măsură a concentrației valorilor, funcția de densitate a pantei având valori mari în regiunile unde conturul prezintă unghiuri de tangentă constante (în zone cu segmente aproximativ drepte) și valori minime în regiunile cu unghiuri de tangentă rapid variabile (la colțuri sau puncte de inflexiune ascuțite).

### 3.2.5. Segmente de contur

Descompunerea conturului în segmente este deseori foarte utilă pentru reprezentarea și codarea imaginilor. Această descompunere reduce complexitatea conturului, simplificând procesul de descriere a acestuia. Această tehnică este deosebit de utilă acolo unde conturul conține una sau mai multe concavități semnificative, care poartă informație semnificativă referitoare la formă. În acest caz, utilizarea înfășurătoarei convexe a regiunii inclusă în contur este foarte utilă pentru descompunerea conturului.

*Înfășurătoarea convexă*  $H$  (convex hull) a unei mulțimi arbitrare  $S$  este cea mai mică mulțime convexă care conține pe  $S$ . Diferența mulțimilor  $H-S$  este numită *deficiență convexă*  $D$  a mulțimii  $S$ . Pentru a exemplifica cum pot fi utilizate aceste noțiuni pentru descompunerea unui contur în segmente, să considerăm figura 3.11.a), care prezintă un obiect (mulțimea  $S$ ) și deficiența sa convexă (regiunile hașurate). Suprafața de separație a regiunii poate fi descompusă urmărind conturul lui  $S$  și marcând punctele în care are loc o tranziție în sau dintr-o componentă a deficienței convexe.

Figura 3.11.b) prezintă rezultatul obținut. În principiu, această metodă este independentă de mărimea și orientarea regiunii.

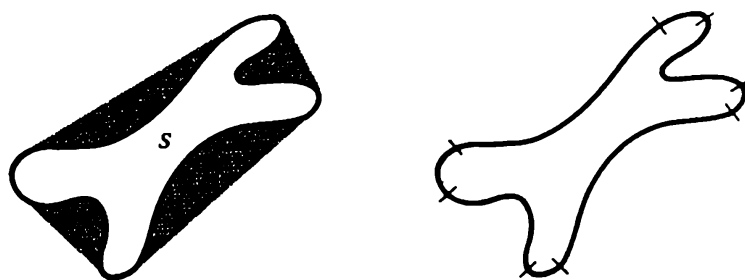


Fig. 3.11. Descompunerea conturului unei regiuni  $S$

Practic, suprafețele de separație digitale au tendința de a deveni neregulate din cauza procesului de discretizare, a zgomotului și a diferențelor de segmentare. Aceste efecte conduc de obicei la o deficiență convexă cu componente mici, fără înțeles, împrăștiate aleator de-a lungul conturului. Se obișnuiește ca, în loc de a elimina aceste neregularități prin postprocesare, să se realizeze o netezire a conturului înainte de a realiza partiția. Există mai multe modalități de a realiza acest lucru. O metodă este de a traversa conturul și de a înlocui coordonatele fiecărui pixel cu valoarea medie a coordonatelor a  $m$  vecini ai săi de-a lungul conturului. Această metodă este adecvată în cazul neregularităților mici, dar necesită un timp lung de calcul și este dificil de controlat. Valori mari ale lui  $m$  pot conduce la o netezire excesivă, în timp ce valori mici ale lui  $m$  pot să nu fie suficiente pe anumite segmente ale conturului. O metodă mai grosieră este de a utiliza aproximarea poligonală, după cum s-a descris în secțiunea 3.2.3, înainte de a găsi deficiența convexă a regiunii.

Conceptul de înfășurătoare convexă și deficiența sa sunt la fel de utile pentru descrierea întregii regiuni, dar și numai a conturului lor. De exemplu, descrierea unei regiuni poate fi bazată pe aria sa, respectiv a deficienței sale convexe, pe numărul de componente al deficienței convexe, pe localizarea relativă a acestor componente, ș.a.m.d. Un algoritm pentru găsirea înfășurătoarei convexe a unei regiuni, bazat pe tehnici ale morfologiei matematice, este descris în secțiunea 4.2.5.

### 3.2.6. Scheletul unei regiuni

Un mod de reprezentare a formei structurale a unei regiuni plane este de a reduce respectiva regiune la un graf. Această reducere poate fi realizată prin obținerea *scheletului* regiunii printr-un algoritm de subțiere (numit uneori și scheletizare). Procedurile de subțiere joacă un rol important într-un mare număr de aplicații de prelucrare a imaginilor.

Scheletul unei regiuni poate fi definit printr-o transformare axă medială MAT (medial axis transformation), propusă de Blum [11] în 1967. Transformarea axă medială a unei regiuni  $R$  cu marginea  $B$  poate fi descrisă astfel. Pentru fiecare punct  $p$  din  $R$ , vom determina cel mai apropiat vecin al său din  $B$ . Dacă  $p$  are mai mult de un singur astfel de vecin, se spune că el aparține *axei mediale* (scheletului) lui  $R$ . Conceptul de "cel mai apropiat" depinde de modul de definire a unei distanțe și de aceea rezultatele acestei

operații sunt influențate de alegerea unei măsuri pentru distanță. Figura 3.12 prezintă câteva exemple, prin utilizarea distanței euclidiene.

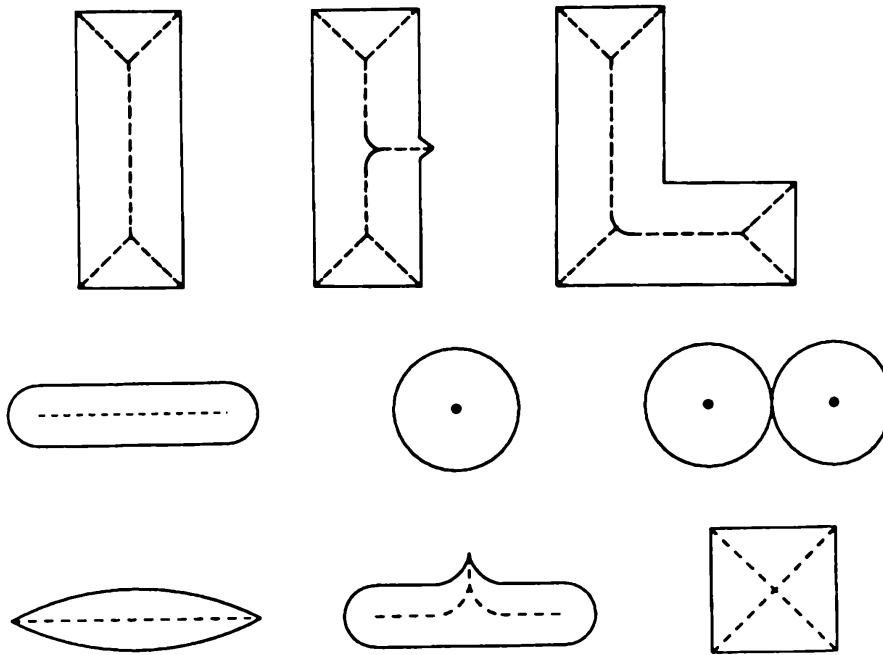


Fig. 3.12. Exemple de axă medială pentru regiuni simple

Deși conceptul de transformare axă medială conduce la un schelet intuitiv, implementarea directă a acestei definiții este de obicei prohibitivă din punct de vedere al complexității de calcul. Potențiala implementare presupune calcularea distanței de la orice punct interior regiunii la fiecare punct al conturului regiunii. Au fost propuși numeroși algoritmi pentru îmbunătățirea eficienței de calcul pentru determinarea axei mediale a regiunii. Tipic, este vorba despre algoritmi de subțiere care șterg iterativ puncte de pe muchiile regiunii cu condiția ca ștergerea acestor puncte:

- să nu înlăture punctele extreme;
- să nu modifice conectivitatea dintre puncte;
- să nu conducă la o erodare excesivă a regiunii.

În cele ce urmează voi prezenta un algoritm pentru subțierea regiunilor binare. Punctele aparținând regiunii sunt considerate a avea valoarea 1, iar punctele fundalului sunt considerate a avea valoarea 0. Metoda constă în parcurgerea succesivă a doi pași de bază aplicați punctelor conturului regiunii date, unde un *punct de pe contur* este reprezentat de orice pixel de valoare 1 care are cel puțin un vecin de valoare 0 într-o octoconectivitate. Cu referire la definiția octoconectivității prezentată în figura 3.13, pasul 1 indică un punct  $p$  de pe contur pentru ștergere dacă sunt satisfăcute următoarele condiții:

$$2 \leq N(p_1) \leq 6 \quad (3.2)$$

$$S(p_1) = 1 \quad (3.3)$$

$$p_2 \cdot p_4 \cdot p_6 = 0 \quad (3.4)$$

$$p_4 \cdot p_6 \cdot p_8 = 0 \quad (3.5)$$

unde  $N(p_1)$  este numărul de vecini diferiți de zero ai lui  $p_1$ , adică,

$$N(p_1) = p_2 + p_3 + \dots + p_8 + p_9 \quad (3.6)$$

iar  $S(p_1)$  este numărul de tranziții  $0 \rightarrow 1$  în secvența ordonată  $p_2, p_3, \dots, p_8, p_9$ .

$p_9$	$p_2$	$p_3$
$p_8$	$p_1$	$p_4$
$p_7$	$p_6$	$p_5$

Fig. 3.13. Organizarea vecinătății pentru algoritmul de subțiere

De exemplu,  $N(p_1)=4$  și  $S(p_1)=3$  în exemplul prezentat în figura 3.14.

0	0	1
1	$p_1$	0
1	0	1

Fig. 3.14. Ilustrarea condițiilor pentru algoritmul de subțiere

În pasul doi, condițiile (3.2) și (3.3) rămân aceleași, dar celelalte două condiții se schimbă în:

$$p_2 \cdot p_4 \cdot p_8 = 0 \quad (3.7)$$

$$p_2 \cdot p_6 \cdot p_8 = 0 \quad (3.8)$$

Primul pas este aplicat fiecărui pixel al conturului regiunii binare considerate. Dacă una sau mai multe din condițiile (3.2)-(3.5) nu sunt satisfăcute, valoarea punctului considerat nu este modificată. Dacă toate condițiile sunt satisfăcute, atunci punctul este marcat pentru ștergere. Totuși, punctul nu este șters până ce toți pixelii conturului au fost prelucrați. Această întârziere are rolul de a preveni schimbarea structurii datelor pe durata execuției algoritmului. După ce pasul întâi a fost aplicat tuturor punctelor de pe contur, toate punctele marcate sunt șterse (modificate la valoarea 0). În continuare, pasul al doilea este aplicat structurii de date rezultate în același mod ca și primul pas.

În acest mod o iterație a algoritmului de subțiere constă în:

1. aplicarea pasului 1 pentru a marca punctele de pe contur care urmează a fi șterse;
2. ștergerea punctelor marcate;
3. aplicarea pasului 2 pentru marcarea spre ștergere a punctelor de pe conturul nou rezultat;
4. ștergerea punctelor marcate.

Această procedură de bază este aplicată iterativ până ce nu mai rezultă noi puncte spre ștergere, moment în care algoritmul se încheie, el producând deja scheletul regiunii respective.

Condiția (3.2) nu este satisfăcută atunci când punctul  $p_1$  are doar unul sau șapte din vecini cu valoarea 1. Dacă există doar un astfel de vecin, atunci  $p_1$  este un punct extrem al scheletului și prin urmare nu trebuie șters. Dacă există șapte vecini cu valoarea 1, atunci ștergerea punctului va produce o erodare a regiunii. Condiția (3.3) nu este satisfăcută atunci când este aplicată pixelilor de pe un contur corespunzător unei regiuni cu grosimea de doar un pixel. De aceea această condiție previne deconectarea

segmentelor unui schelet pe durata procesului de subțiere. Condițiile (3.4) și (3.5) sunt satisfăcute simultan de setul minim de valori: ( $p_4 = 0$  sau  $p_6 = 0$ ) sau ( $p_2 = 0$  și  $p_8 = 0$ ). Astfel, în conformitate cu aranjamentul vecinilor din figura 3.13, un punct care satisface aceste condiții, precum și condițiile (3.2) și (3.3), este un punct situat pe contur la est sau sud sau un punct situat la colțul de nord-vest al conturului. În ambele cazuri,  $p_1$  nu este parte a scheletului și trebuie șters. În mod similar, condițiile (3.7) și (3.8) sunt satisfăcute simultan de următoarele seturi minime de valori: ( $p_2 = 0$  sau  $p_8 = 0$ ) sau ( $p_4 = 0$  și  $p_6 = 0$ ). Aceste valori corespund punctelor de contur nord sau vest, fie colțului de sud-est. Este de observat că punctele colțului de nord-est au  $p_2 = 0$  și  $p_4 = 0$ , satisfăcând condițiile (3.4) și (3.5), precum și condițiile (3.7) și (3.8). Aceeași observație este valabilă și pentru punctele colțului de sud-vest, care au  $p_6 = 0$  și  $p_8 = 0$ .

Figura 3.15.a) prezintă rezultatul aplicării pasului unu al algoritmului de subțiere la conturul unei regiuni simple. Sunt prezentate punctele marcate și apoi înlăturate la sfârșitul pasului unu. Figura 3.15.b) prezintă rezultatul obținut după pasul al doilea, iar figura 3.15.c) prezintă scheletul obținut după mai multe iterații a celor doi pași.

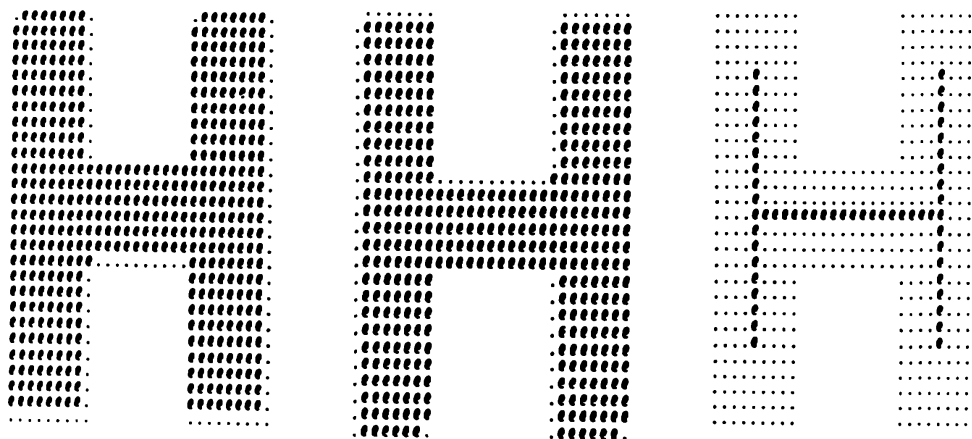


Fig. 3.15. Schelet regulat obținut prin subțiere

Scheletul unei regiuni cu proprietăți neregulate este prezentat în figura 3.16.

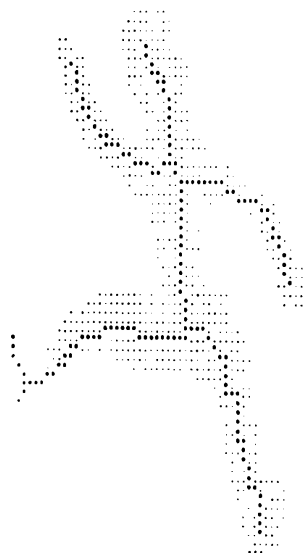


Fig. 3.16. Exemplu de schelet neregulat



### 3.3. Descrierea formei pe baza conturului

#### 3.3.1. Reprezentarea geometrică simplă a frontierei

Descriptorii pe care îi voi prezenta în continuare sunt bazați pe proprietăți geometrice ale regiunilor descrise. Datorită caracterului discret al imaginilor digitale, toți acești descriptori sunt sensibili la rezoluția imaginii.

##### 3.3.1.1. Lungimea conturului

Lungimea conturului este o proprietate elementară a regiunii, derivată în mod simplu din reprezentarea prin coduri lanț. Pașii verticali și orizontali au lungimi unitare, iar lungimea pașilor diagonali, prin utilizarea 8-conectivității, este  $\sqrt{2}$ . Lungimea conturului este mai mare în cazul 4-conectivității unde un pas diagonal este format din doi pași rectangulari cu o lungime totală 2. Lungimea unui contur închis (perimetru) poate fi ușor evaluată din codarea run-length [191] sau din reprezentările arborescente [39]. Lungimea conturului crește pe măsură ce crește rezoluția imaginii; pe de altă parte, aria regiunii nu este afectată de modificarea rezoluției, ea convergând spre o anumită limită. Pentru a asigura proprietăți continue în spațiu bazate pe perimetru (calculul ariei din lungimea conturului, proprietăți ale formei, etc.) este preferabilă definirea conturului unei regiuni ca fiind frontiera exterioară sau frontiera extinsă. Dacă se utilizează frontiera interioară, unele proprietăți nu sunt satisfăcute, de exemplu, perimetrul unei regiuni de un pixel este patru prin utilizarea frontierei exterioare și unu prin utilizarea frontierei interioare.

##### 3.3.1.2. Diametrul conturului

*Diametrul* unui contur  $B$  este definit ca

$$\text{Diam}(B) = \max_{i,j} [D(p_i, p_j)] \quad (3.9)$$

unde  $D$  este o măsură a distanței și  $p_i, p_j$  sunt puncte ale conturului. Valoarea diametrului și orientarea unei linii care leagă cele două puncte extreme care formează diametrul (linie numită *axă majoră* a conturului) sunt descriptori utili ai conturului.

##### 3.3.1.3. Curbura

În cazul continuu, curbura este definită ca viteză de variație a pantei. În spațiul discret, descrierea curburii trebuie ușor modificată pentru a depăși dificultățile rezultate din nerespectarea netezității curbei. Descriptorul scalar al curburii prezintă raportul dintre numărul total de pixeli ai conturului (lungime) și numărul de pixeli ai conturului în care direcția conturului se modifică semnificativ. Cu cât este mai mic numărul de schimbări a direcției, cu atât conturul este apreciat ca fiind mai drept. Algoritmul de evaluare este bazat pe detecția unghiurilor dintre pantele segmentelor de contur adiacente (care au fost reprezentate ca linii drepte). Unghiurile nu trebuie reprezentate

numeric, poziția relativă a segmentelor de dreaptă putând fi utilizată ca proprietate. Parametrul de curbură determină sensibilitatea la modificările locale ale direcției conturului. Calculul curburii pornind de la codurile lanț poate fi găsit în [191]. Calculul curburii se poate face și pornind de la reprezentarea conturului prin tangentele în fiecare punct al său. Valorile curburii în fiecare punct pot fi reprezentate printr-o histogramă; numerele relative pot în acest caz da informații cu privire la cât de frecvente sunt diversele schimbări specifice ale direcției conturului.

### 3.3.1.4. Distribuția corzilor

O linie care unește oricare două puncte ale conturului regiunii este o coardă, iar distribuția lungimilor și a unghiurilor tuturor corzilor pe un contur poate fi utilizată pentru descrierea formei. Să considerăm că  $b(x, y) = 1$  reprezintă punctele conturului, iar  $b(x, y) = 0$  reprezintă toate celelalte puncte. Distribuția corzilor poate fi calculată (figura 3.17.a) ca

$$h(\Delta x, \Delta y) = \iint b(x, y) b(x + \Delta x, y + \Delta y) \cdot dx \cdot dy \quad (3.10)$$

sau pentru imagini digitale, ca

$$h(\Delta x, \Delta y) = \sum_i \sum_j b(i, j) \cdot b(i + \Delta x, j + \Delta y) \quad (3.11)$$

Pentru a obține distribuția radială independentă la rotație  $h_r(r)$  se calculează integrala peste toate unghiurile (figura 3.17.b).

$$h_r(r) = \int_{-\pi/2}^{\pi/2} h(\Delta x, \Delta y) \cdot r \cdot d\theta \quad (3.12)$$

unde  $r = \sqrt{\Delta x^2 + \Delta y^2}$  și  $\theta = \arcsin\left(\frac{\Delta y}{r}\right)$ . Distribuția  $h_r(r)$  variază linear. Distribuția unghiulară  $h_u(\theta)$  este independentă de scară, în timp ce rotația produce o deviație proporțională

$$h_u(\theta) = \int_0^{\max(r)} h(\Delta x, \Delta y) \cdot dr \quad (3.13)$$

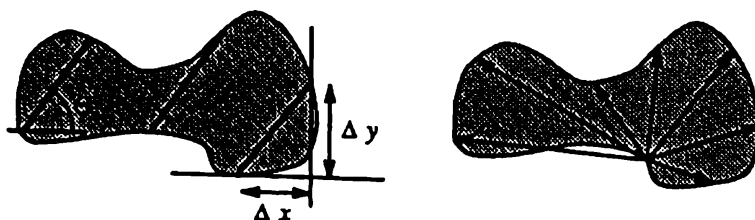


Fig. 3.17. Distribuția corzilor

Combinarea ambelor distribuții conduce la un descriptor al formei foarte robust [66].

### 3.3.2. Numere ale formei

După cum s-a prezentat în secțiunea 3.2.1, codul lanț diferențial de ordinul unu pentru reprezentarea conturului depinde de punctul de start. *Numărul formei* unui astfel de contur, bazat pe codul 4-direcțional din figura 3.2.a), este definit ca prima diferență de cea mai mică valoare. *Ordinul  $n$*  al unui număr al formei este definit ca numărul de digiți din reprezentarea sa. În plus,  $n$  este par pentru un contur închis, iar valoarea sa limitează numărul de forme diferite posibile. Figura 3.18 prezintă toate formele de ordinul 4, 6 și 8, împreună cu reprezentările lor prin coduri lanț, cu primele diferențiale și numerele formei corespunzătoare. Se remarcă faptul că primele diferențiale au fost calculate tratând codurile lanț ca secvențe circulare, așa cum s-a discutat în secțiunea 3.2.1.

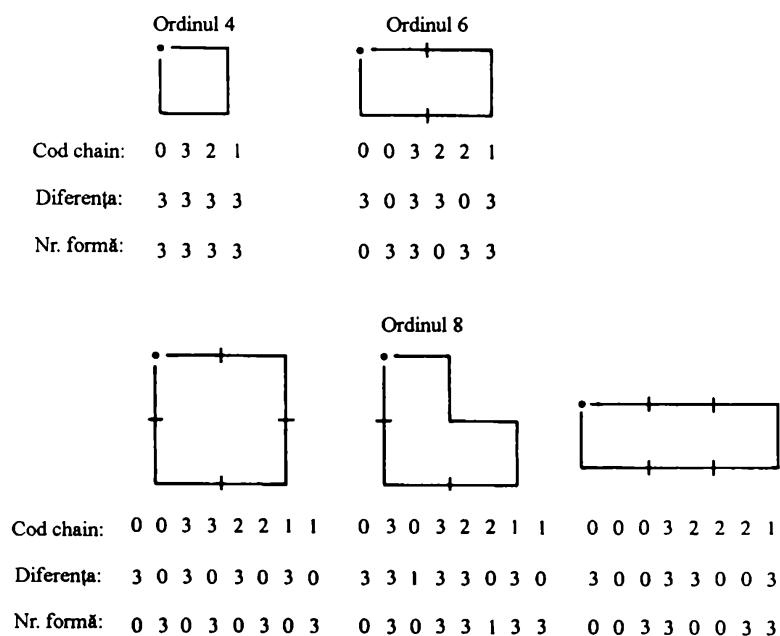


Fig. 3.18. Formele de ordinul 4, 6 și 8

Deși primele diferențiale ale unui cod lanț sunt independente de rotație, în general conturul codat depinde de orientarea rețelei. O metodă de normalizare a orientării rețelei este prezentată în cele ce urmează.

După cum s-a arătat anterior, *axa majoră* a unui contur reprezintă segmentul de linie dreaptă care unește cele mai depărtate dintre ele două puncte. *Axa minoră* este perpendiculară pe axa majoră și de o lungime corespunzătoare astfel încât să poată fi format un dreptunghi care să includă întregul contur. Raportul dintre axa majoră și axa minoră poartă denumirea de *excentricitate* a conturului, iar dreptunghiul menționat poartă denumirea de *dreptunghi de bază*. În cele mai multe cazuri, prin alinierea rețelei codului lanț cu laturile dreptunghiului de bază, se obține un număr al formei unic.

Practic, pentru un număr al formei dorit, se găsește dreptunghiul de ordin  $n$  a cărui excentricitate aproximează cel mai bine dreptunghiul de bază și se utilizează acest nou dreptunghi pentru stabilirea mărimii rețelei. De exemplu, dacă  $n = 12$ , toate dreptunghiurile de ordin 12 (adică cele care au lungimea perimetrului egală cu 12) sunt  $2 \times 4$ ,  $3 \times 3$  și  $1 \times 5$ . Dacă excentricitatea dreptunghiului  $2 \times 4$  aproximează cel mai bine

excentricitatea dreptunghiului de bază pentru un contur dat, atunci se va stabili o rețea  $2 \times 4$  centrată pe dreptunghiul de bază și se va utiliza procedura descrisă în secțiunea 3.2.1 pentru a se obține codul lanț. Numărul forme rezultă din prima diferențială a acestui cod, după cum s-a mai arătat. Deși ordinul numărului forme rezultat este uzual egal cu  $n$  datorită modului în care a fost selectată rețeaua, contururile cu o adâncitură comparabilă cu pasul rețelei pot conduce uneori la numere ale forme mai mari decât  $n$ . În acest caz, se specifică un dreptunghi de ordin mai mic decât  $n$  și se repetă procedura până ce numărul forme rezultat este de ordinul  $n$ .

Să considerăm un exemplu. Să presupunem că pentru conturul din figura 3.19.a) s-a specificat  $n = 18$ . Obținerea unui număr al forme de acest ordin presupune urmărirea pașilor prezentați anterior. Primul pas constă în determinarea dreptunghiului de bază, așa cum se vede în figura 3.19.b). Cel mai apropiat dreptunghi de ordinul 18 este dreptunghiul  $3 \times 6$ , care necesită subdivizarea dreptunghiului de bază ca în figura 3.19.c), unde direcțiile codului lanț sunt aliniată cu rețeaua rezultată. Pasul final constă în obținerea codului lanț și utilizarea primei sale diferențiale pentru a se calcula numărul forme, așa cum se vede în figura 3.19.d).

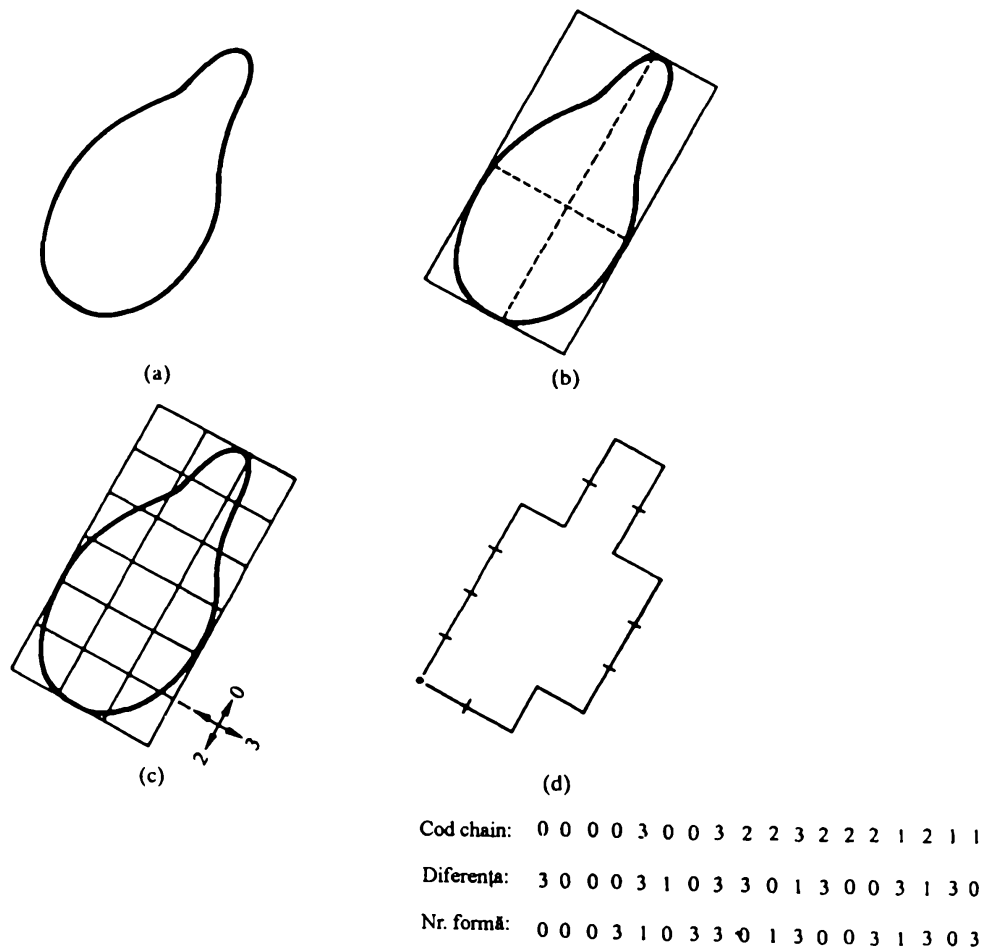


Fig. 3.19. Generarea unui număr al forme

### 3.3.3. Descriptori Fourier

Figura 3.20 prezintă un contur digital de  $N$  puncte în planul  $xy$ . Pornind dintr-un punct arbitrar  $(x_0, y_0)$ , perechile de coordonate  $(x_0, y_0)$ ,  $(x_1, y_1)$ ,  $(x_2, y_2)$ , ...,  $(x_{N-1}, y_{N-1})$  sunt întâlnite la traversarea conturului, de exemplu în sens invers acelor de ceasornic. Aceste coordonate pot fi exprimate în forma  $x(k) = x_k$  și  $y(k) = y_k$ . Cu această notație, conturul poate fi reprezentat ca o secvență de coordonate  $s(k) = [x(k), y(k)]$ , pentru  $k = 0, 1, 2, \dots, N - 1$ . În plus, fiecare pereche de coordonate poate fi tratată ca un număr complex, astfel încât

$$s(k) = x(k) + j \cdot y(k) \quad (3.14)$$

pentru  $k = 0, 1, 2, \dots, N - 1$ . Cu alte cuvinte, axa  $x$  este considerată ca axă reală, iar axa  $y$  este considerată ca axă imaginară pentru o secvență de numere complexe. Deși s-a schimbat interpretarea secvenței de perechi de coordonate, natura conturului în sine nu s-a modificat. Această interpretare prezintă marele avantaj de a reduce o problemă bidimensională la o problemă unidimensională.

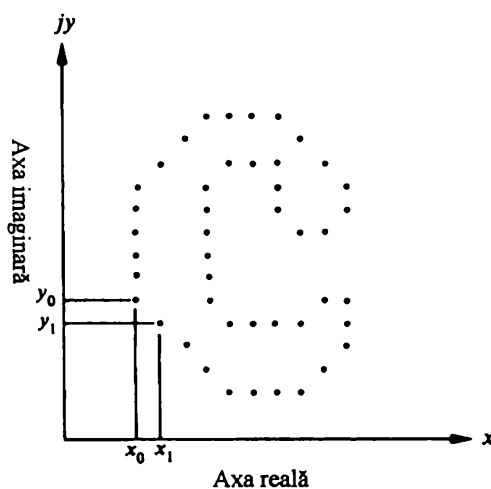


Fig. 3.20. Reprezentarea conturului digital prin secvențe complexe

Transformarea Fourier directă (DFT) a lui  $s(k)$  este

$$a(u) = \frac{1}{N} \sum_{k=0}^{N-1} s(k) \exp[-j2\pi uk / N] \quad (3.15)$$

pentru  $u = 0, 1, 2, \dots, N - 1$ . Coeficienții complexi  $a(u)$  sunt numiți *descriptori Fourier* ai conturului. Transformarea Fourier inversă a coeficienților  $a(u)$  permite restaurarea coeficienților  $s(k)$ .

Prin urmare,

$$s(k) = \sum_{u=0}^{N-1} a(u) \exp[j2\pi uk / N] \quad (3.16)$$

pentru  $k = 0, 1, 2, \dots, N - 1$ . Să presupunem însă, că în loc de a considera toți coeficienții  $a(u)$ , doar primii  $M$  coeficienți sunt utilizați. Acest lucru este echivalent cu a seta  $a(u) = 0$  pentru  $u > M - 1$  în ecuația (3.16). Rezultatul este următoarea

aproximare pentru  $s(k)$ :

$$\hat{s}(k) = \sum_{u=0}^{M-1} a(u) \exp[j2\pi uk / N] \quad (3.17)$$

pentru  $k = 0, 1, 2, \dots, N - 1$ . Deși doar  $M$  termeni sunt utilizați pentru obținerea fiecărei componente  $s(k)$ ,  $k$  continuă să ia valori de la 0 la  $N - 1$ . Aceasta înseamnă că în conturul aproximat există același număr de puncte, dar nu tot atâția termeni sunt utilizați pentru reconstrucția fiecărui punct. Dacă numărul de puncte ale conturului este mare,  $M$  este în general selectat ca o putere întreagă a lui 2, astfel încât algoritmul de transformare Fourier rapidă (FFT) să poată fi utilizat pentru calculul descriptorilor. Este important de reamintit că detaliile fine sunt aduse de coeficienții de înaltă frecvență, iar componentele de joasă frecvență determină forma globală. Prin urmare, cu cât  $M$  este mai mic, cu atât mai multe detalii se pierd din contur.

Figura 3.21 prezintă un contur de formă pătrată format din  $N = 64$  puncte și rezultatul utilizării ecuației (3.17) pentru reconstrucția acestui contur pentru diverse valori ale lui  $M$ . Este de remarcat faptul că valoarea lui  $M$  trebuie să fie cel puțin 8 pentru ca acest contur să înceapă să semene mai mult cu un pătrat decât cu un cerc. Se observă de asemenea că îmbunătățirea definiției colțurilor apare abia la valori ale lui  $M$  în jur de 56. În sfârșit, se observă că atunci când  $M = 61$ , curbele care reprezintă laturile încep să devină mai drepte, ceea ce conduce la o replică aproape exactă a originalului pentru valori mai mari ale lui  $M$ . Prin urmare, un număr mic de coeficienți de joasă frecvență pot da informații asupra formei grosiere, dar definirea precisă a colțurilor și liniilor drepte necesită mult mai mulți coeficienți de înaltă frecvență. Acest rezultat era însă de așteptat ținând cont de rolul componentelor de joasă și înaltă frecvență în definirea formei unei regiuni.

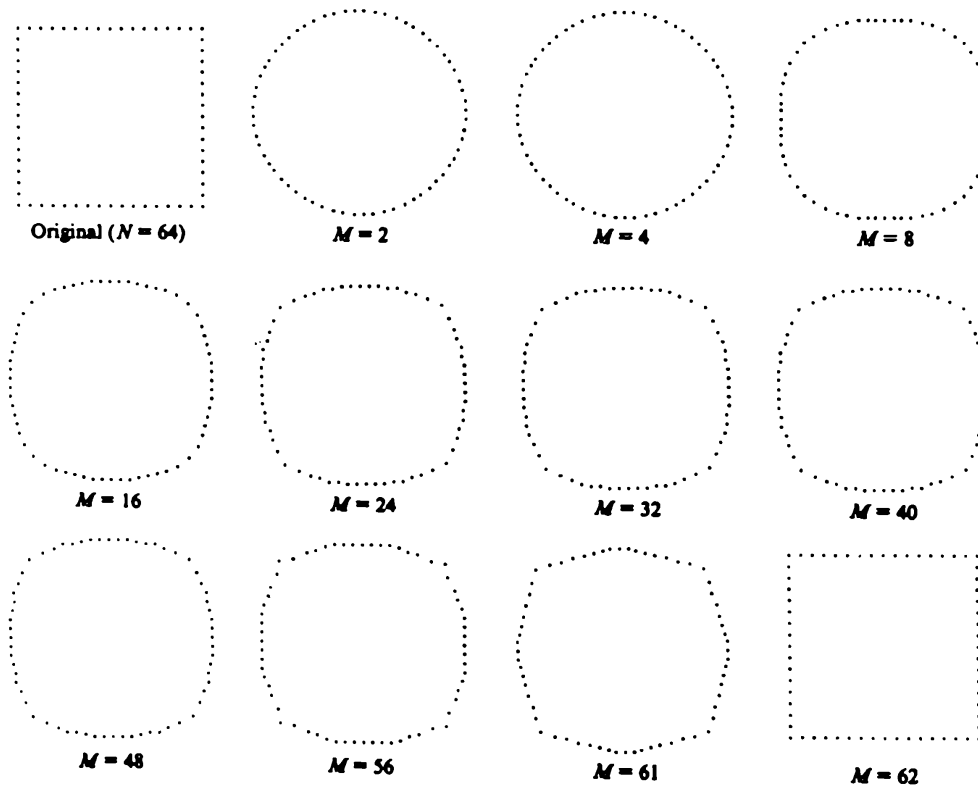


Fig. 3.21. Reconstrucția din descriptori Fourier

Descriptorii trebuie să fie cât mai invariabili la translație, rotație și scalare. În cazurile în care rezultatele depind de ordinea în care sunt prelucrate punctele, o condiție suplimentară este ca descriptorii să fie invariabili la punctul de start. Descriptorii Fourier nu sunt direct invariabili la aceste transformări geometrice, dar modificările pot fi modelate prin transformări simple ale descriptorilor. De exemplu, să considerăm rotația și să ne reamintim din analiza matematică că rotația unui punct cu un unghi  $\theta$  în jurul originii planului complex este realizată prin multiplicarea punctului cu  $e^{j\theta}$ . Multiplicând fiecare punct al secvenței  $s(k)$  se realizează rotirea întregii secvențe în jurul originii. Secvența rotită este  $s(k) \cdot e^{j\theta}$ , ai cărei descriptorii Fourier sunt

$$a_r(u) = \frac{1}{N} \sum_{k=0}^{N-1} s(k) \cdot \exp[j\theta] \cdot \exp[-j2\pi uk/N] = a(u) \cdot e^{j\theta} \quad (3.18)$$

pentru  $u = 0, 1, 2, \dots, N-1$ . Prin urmare rotația afectează toți coeficienții prin același coeficient de multiplicare constant  $e^{j\theta}$ .

### 3.3.4. Momente

Forma segmentelor de contur (și a semnăturilor) poate fi descrisă cantitativ prin utilizarea momentelor. Pentru a arăta cum este posibil acest lucru, să considerăm figura 3.22.a), care prezintă un segment al unui contur și figura 3.22.b), care prezintă segmentul descris ca o funcție unidimensională  $g(r)$  de variabilă arbitrară  $r$ .

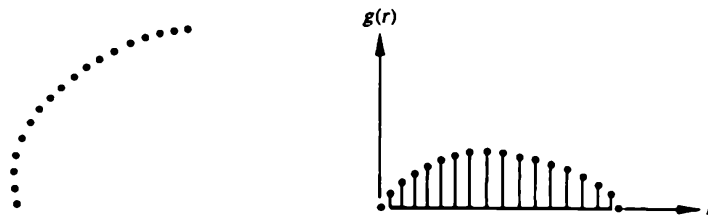


Fig. 3.22. Reprezentarea unui segment de contur ca funcție unidimensională

Să considerăm amplitudinea funcției  $g$  ca o variabilă aleatoare  $v$  și să formăm histograma amplitudinilor  $p(v_i)$ ,  $i = 1, 2, \dots, K$ , unde  $K$  este numărul de incremente ale amplitudinii discrete. Atunci, momentul de ordinul  $n$  al lui  $v$  este

$$\mu_n(v) = \sum_{i=1}^K (v_i - m)^n p(v_i) \quad (3.19)$$

unde

$$m = \sum_{i=1}^K v_i p(v_i) \quad (3.20)$$

Cantitatea  $m$  este cunoscută ca valoare medie a lui  $v$ , iar  $\mu_2$  ca varianța sa. În general, doar un număr mic de momente este necesar pentru a deosebi între ele semnăturile unor forme distincte.

O altă variantă este aceea de a normaliza  $g(r)$  la valoarea unitară și să o tratăm

ca pe o histogramă. În acest caz,  $r$  devine o variabilă aleatoare, iar momentele sunt

$$\mu_n(r) = \sum_{i=1}^L (r_i - m)^n g(r_i) \quad (3.21)$$

unde

$$m = \sum_{i=1}^L r_i g(r_i) \quad (3.22)$$

În acest caz,  $L$  este numărul de puncte ale conturului, iar  $\mu_n(r)$  este direct dependent de forma lui  $g(r)$ . De exemplu, momentul de ordinul doi  $\mu_2(r)$  măsoară împrăștierea curbei în jurul valorii medii a lui  $r$ , iar momentul de ordinul trei  $\mu_3(r)$  măsoară simetria sa față de medie. Ambele reprezentări ale momentelor pot fi utilizate simultan pentru descrierea unui segment de contur sau a unei semnături.

În esență, s-a realizat reducerea descrierii conturului la descrierea unor funcții unidimensionale. Deși utilizarea momentelor este cea mai populară metodă, ea nu este singura care poate fi utilizată în acest scop. O altă metodă constă în calcularea transformatei Fourier discrete unidimensionale, obținerea spectrului său și utilizarea primelor sale  $k$  componente pentru descrierea lui  $g(r)$ . Avantajul momentelor față de alte tehnici constă în ușurința implementării și în existența unei interpretări fizice a formei conturului. Invarianța la rotație a acestei tehnici rezultă din figura 3.22.



## Capitolul 4

### Bazele morfologiei matematice

Abordarea prelucrării numerice a imaginilor prin metode nemorfologice este axată pe tehnici de calcul, fiind bazată pe conceptul de distribuție a funcțiilor punctuale și pe transformări liniare ca, de exemplu, convoluția.

Prin morfologia matematică, descrierea imaginilor se face prin metode geometrice și de teoria mulțimilor. Morfologia matematică este o disciplină specifică în prelucrarea imaginilor, care a fost aplicată pentru analiza structurii materialelor în diverse domenii, ca: mineralogia, petrografia, angiografia, citologia, etc. Ea s-a născut practic în 1964, atunci când Matheron și-a început investigațiile privind relațiile dintre geometria mediilor poroase și permeabilitatea acestor medii, iar lui Serra [211] i s-a cerut să cuantifice petrografia minereurilor de fier pentru a putea prezice proprietățile lor de măcinare. Ei au început prin a observa că, dacă intensitățile sonore pot fi adunate pentru a obține sunete complexe, semnalele luminoase se combină într-un mod diferit. În cazul vederii linearitatea este absentă, imaginile putând fi adunate de exemplu pentru a reduce zgomotul. Prima lege a vederii arată că fiecare obiect ascunde tot ceea ce este situat în spatele său. De exemplu, dacă obiectul  $A$  este situat în fața unui obiect  $B$ , ochiul nu poate percepe obiectul  $B$ , ci doar diferența  $B-A$ . De aceea, pentru a exprima faptul că obiectele sunt opace, și nu translucide, precum și faptul că ele se văd în perspectivă, putem utiliza concepte ca incluziunea, intersecția, etc.

Cuvântul morfologie înseamnă “forma și structura unui obiect”, sau interrelațiile și condiționările dintre părțile componente ale unui obiect. Morfologia se ocupă cu forma obiectelor, iar morfologia digitală este o modalitate de descriere sau analiză a formei unui obiect digital. Ideea care stă la baza morfologiei digitale este: imaginile sunt constituite dintr-o mulțime de elemente de imagine (pixeli) care formează grupuri cu structură bidimensională (forme). Ideea de bază este de a compara obiectele care urmează a fi analizate cu alte obiecte având forma cunoscută, numite *elemente structurante*. Putem utiliza morfologia matematică cu scopul de a extrage componente de imagine utile pentru reprezentarea și descrierea formei unei regiuni, ca de exemplu contururi, schelete și înfășurătoare convexă. De asemenea sunt disponibile tehnici morfologice de pre și postprocesare, ca filtrarea morfologică a zgomotului, simplificarea formelor, subțierea (thinning), îngroșarea (thickening) și curățirea (pruning).

Limbajul morfologiei matematice este cel al *teoriei mulțimilor*. De aceea, morfologia matematică oferă o abordare puternică și unificată pentru diferite probleme de prelucrarea imaginilor. Mulțimile din morfologia matematică reprezintă formele obiectelor dintr-o imagine. De exemplu, mulțimea tuturor pixelilor negri dintr-o imagine digitală binară realizează o descriere completă a imaginii respective. În imaginile binare, mulțimile de pixeli sunt componente ale spațiului bidimensional  $Z^2$ , unde fiecare element al mulțimii este un vector bidimensional având coordonatele egale cu coordonatele  $(x,y)$  ale unui pixel negru al imaginii. Imaginile digitale pe scară de gri pot fi reprezentate ca mulțimi având componente în  $Z^3$ . În acest caz, două componente ale fiecărui element al mulțimii reprezintă coordonatele unui pixel, iar al treilea corespunde valorii discrete a intensității. Mulțimile din spații de dimensiuni mai mari pot conține alte atribute ale imaginii, ca de exemplu culoarea sau componentele variabile în timp.

În următoarele secțiuni ale prezentei lucrări voi ilustra mai multe concepte importante din morfologia matematică. Multe din aceste operații pot fi formulate ca probleme în spațiul euclidian  $n$ -dimensional,  $E^n$ . Totuși, mă voi referi în special la imagini binare cu componente în spațiul  $Z^2$ , care formează baza preocupărilor mele. Extinderea la imagini pe scară de gri va fi discutată ulterior.

#### 4.1. Operatori morfologici binari de bază

##### 4.1.1. Definiții de bază

Morfologia matematică utilizează proprietățile mulțimilor de puncte, rezultate ale geometriei și topologiei. Se presupun cunoscute și își păstrează semnificația operațiile clasice ca *incluziunea* ( $\subset$  sau  $\supset$ ), *intersecția* ( $\cap$ ), *reuniunea* ( $\cup$ ), *mulțimea vidă*  $\emptyset$  și *mulțimea complementară* ( $^c$ ).

Fie  $A$  și  $B$  două mulțimi în  $Z^2$ , cu componentele  $a = (a_1, a_2)$  și  $b = (b_1, b_2)$ , respectiv.

*Tranșlația* lui  $A$  cu  $x = (x_1, x_2)$ , notată  $(A)_x$ , este definită ca

$$(A)_x = \{c \mid c = a + x, \text{ pentru } a \in A\}. \quad (4.1)$$

*Reflectarea* mulțimii  $B$ , notată cu  $\hat{B}$ , este definită ca

$$\hat{B} = \{x \mid x = -b, \text{ pentru } b \in B\}. \quad (4.2)$$

*Complementara* mulțimii  $A$  este

$$A^c = \{x \mid x \notin A\}. \quad (4.3)$$

*Diferența* a două mulțimi  $A$  și  $B$ , notată  $A - B$ , este definită ca

$$A - B = \{x \mid x \in A, x \notin B\} = A \cap B^c. \quad (4.4)$$

Punctele care aparțin obiectelor dintr-o imagine reprezintă o mulțime  $X$ . Aceste puncte sunt pixeli cu valoarea egală cu unu. Punctele mulțimii complementare  $X^c$  corespund pixelilor fundalului, care au valoarea egală cu zero. Originea (marcată în exemplul considerat cu o cruce) are coordonatele  $(0,0)$ , iar coordonatele oricărui punct sunt interpretate ca (*rând*, *coloană*). Figura 4.1 prezintă un exemplu de astfel de mulțime, punctele aparținând obiectului fiind reprezentate prin pătrate înnegrite. Orice punct  $x$  al unei imagini discrete  $X$  poate fi considerat ca un vector raportat la originea  $(0,0)$ .

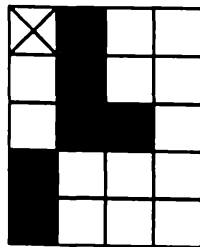


Fig. 4.1. Exemplu de mulțime de puncte

### 4.1.2. Transformări morfologice

Să considerăm un obiect bidimensional, definit prin mulțimea de puncte  $X$  care constituie respectivul obiect. Structura lui  $X$  este dată prin totalitatea relațiilor dintre aceste puncte. Ideea de a descrie structura prin conexiuni între forme similare aflate în poziții diferite este cuantizată în morfologia matematică prin conceptul de *element structurant*. Acesta este ales ca o mică mulțime de puncte  $B$ , cu o geometrie cunoscută. În continuare, elementul structurant este translatat astfel încât originea sa să se deplaseze în fiecare punct al spațiului. Pentru fiecare poziție se pune problema unei relații referitoare la reuniune, intersecție sau incluziune între mulțimile  $X$  și  $B$ . Răspunsul la întrebarea privitoare la această relație va fi pozitiv sau negativ, de unde apare denumirea de “transformări de tipul totul sau nimic”. Mulțimea punctelor corespunzătoare unui răspuns pozitiv va forma o nouă mulțime, numită *image transformată*. Figura 4.2 prezintă câteva elemente structurante tipice. Elementele structurante  $B$  sunt exprimate în raport cu originea locală  $O$ , numită *punct reprezentativ*. În figura 4.2 punctele reprezentative sunt marcate cu o cruce, fiind de asemenea ilustrată și posibilitatea ca punctul  $O$  să nu aparțină elementului structurant.



Fig. 4.2. Elemente structurante tipice

Vom considera imaginile boolene ca submulțimi ale unui spațiu euclidian sau digital  $E$ . Un element structurant este în esență o submulțime  $B$  a spațiului  $E$ . Presupunând că am fixat originea  $O$  a spațiului  $E$ , fiecărui punct  $p$  din  $E$  îi corespunde o translație care transformă  $O$  în  $p$ , și  $B$  în  $(B)_p$ , numită mulțimea  $B$  translatată prin  $p$ . Această proprietate se datorează faptului că spațiul  $E$  în care sunt reprezentate imaginile este omogen față de grupul translațiilor.

Cel mai simplu set de transformări morfologice este format din *dilatare*, *erodare*, *deschidere* și *închidere*.

#### 4.1.2.1. Dilatarea

Fiind dată o submulțime  $X$  a lui  $E$ , *dilatarea* lui  $X$  cu  $B$ , notată  $X \oplus B$ , este definită ca mulțimea tuturor sumelor vectoriale a perechilor de elemente, câte unul din mulțimea  $X$ , respectiv  $B$ .

$$X \oplus B = \{c \mid c=x+b, x \in X, b \in B\} \quad (4.5)$$

Această operație a fost de fapt introdusă prima dată de Minkowski, fiind de aceea numită și adunare Minkowski a mulțimilor.

Figura 4.3 ilustrează un exemplu de dilatare, prezentând în mod grafic efectul acestei transformări [232].

$$\begin{aligned} X &= \{ (0,1), (1,1), (2,1), (2,2), (3,0), (4,0) \} \\ B &= \{ (0,0), (0,1) \} \\ X \oplus B &= \{ (0,1), (1,1), (2,1), (2,2), (3,0), (4,0), (0,2), (1,2), (2,2), (2,3), (3,1), \\ &\quad (4,1) \} \end{aligned}$$

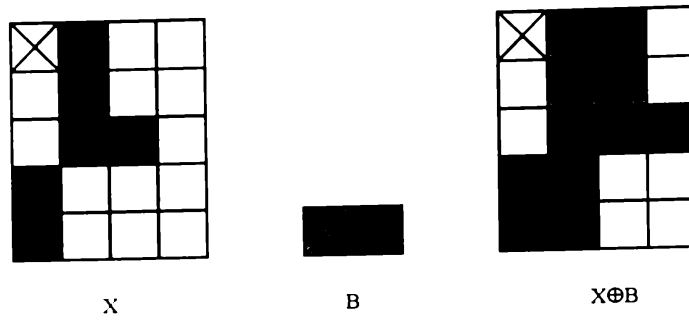


Fig. 4.3. Exemplu de dilatare

Este interesant de luat în considerare și cazul în care originea nu este inclusă în elementul structurant. Figura 4.4 prezintă un astfel de exemplu.

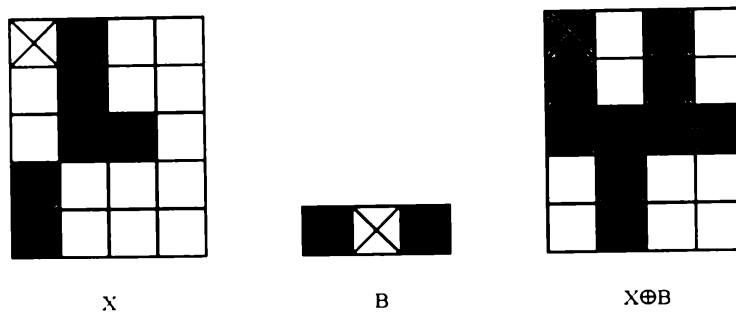


Fig. 4.4. Dilatarea cu element structurant care nu include originea

Modalitatea de calcul a dilatării conform relației (4.5) presupune considerarea dilatării ca pe o reuniune a tuturor translațiilor specificate de elementul structurant, prin urmare

$$X \oplus B = \bigcup_{b \in B} (X)_b \quad (4.6)$$

Deoarece dilatarea este comutativă, ea poate fi de asemenea considerată ca reuniunea tuturor translațiilor elementului structurant cu toți pixelii din imagine:

$$X \oplus B = \bigcup_{x \in X} (B)_x \quad (4.7)$$

Demonstrarea echivalenței acestor definiții poate fi găsită în [211] și [72]. Această echivalență furnizează o posibilă modalitate de implementare a operatorului de dilatare: considerarea elementului structurant ca un șablon, urmată de mișcarea acestui șablon peste imagine. Atunci când originea elementului structurant se suprapune peste un pixel negru al imaginii, toți pixelii imaginii care corespund pixelilor negri ai elementului structurant sunt marcați și vor fi transformați ulterior în pixeli negri. După ce întreaga imagine a fost baleiată de elementul structurant, calculul dilatării este complet. În general dilatarea unei imagini nu este calculată prin substituție, adică rezultatul nu este copiat peste imaginea originală. O a treia imagine, inițial complet albă, este utilizată pentru a memora rezultatul dilatării în timpul efectuării calculului.

O altă definiție echivalentă poate fi găsită în [63]:

$$X \oplus B = \{c | (\hat{B})_c \cap X \neq \emptyset\} \quad (4.8)$$

Conform acestei definiții, procesul de dilatare constă în obținerea mulțimii  $B$  reflectate față de origine și apoi în translatarea acesteia cu  $c$ . Dilatarea lui  $X$  prin  $B$  va fi

prin urmare mulțimea tuturor deplasamentelor  $c$  astfel încât  $\hat{B}$  și  $X$  să se suprapună cel puțin într-un element nenul. Pe baza acestei interpretări, ecuația (4.8) poate fi rescrisă ca:

$$X \oplus B = \{c \mid [(\hat{B})_c \cap X] \subseteq X\} \quad (4.9)$$

Totuși, definiția (4.7) are un mare avantaj față de celelalte formulări, fiind mai intuitivă prin faptul că elementul structurant  $B$  este văzut ca o mască de convoluție. Deși dilatarea se bazează pe operațiuni cu mulțimi, iar convoluția pe operații aritmetice, procesul de reflectare a lui  $B$  față de origine și apoi de deplasare succesivă a sa astfel încât să se suprapună peste imaginea  $X$  este analog cu procesul de convoluție.

Figura 4.5 prezintă două exemple de dilatare realizată cu elemente structurante diferite pe aceeași imagine de bază.

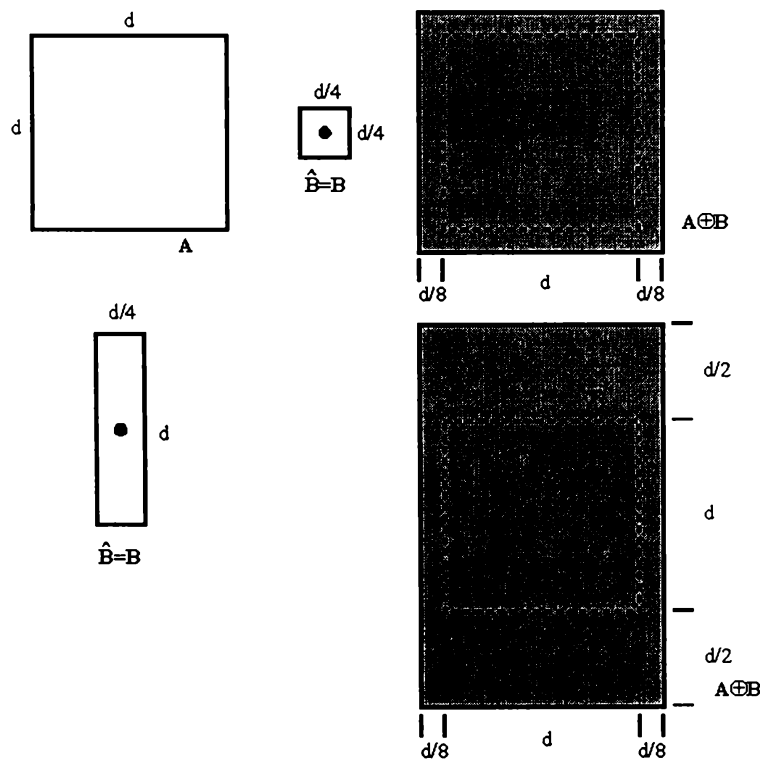


Fig. 4.5. Exemple de dilatare

Dilatarea are mai multe proprietăți interesante, care pot ușura implementarea sa hardware sau software. Voi prezenta câteva din aceste proprietăți, fără demonstrații, care pot fi găsite în [63] sau [66].

Dilatarea este *comutativă*,

$$X \oplus B = B \oplus X \quad (4.10)$$

*asociativă*,

$$X \oplus (B \oplus C) = (X \oplus B) \oplus C \quad (4.11)$$

*invariantă la translație*,

$$X \oplus B_h = (X \oplus B)_h \quad (4.12)$$

și totodată o *transformare crescătoare*,

$$\text{dacă } X \subseteq Y \text{ atunci } X \oplus B \subseteq Y \oplus B \quad (4.13)$$

Dilatarea este utilizată pentru umplerea unor mici goluri și a unor golfuri înguste din obiecte. Ea conduce la creșterea dimensiunilor obiectelor. Dacă este necesară păstrarea mărimii originale a obiectelor, atunci dilatarea este combinată cu operațiunea de erodare, care va fi descrisă în continuare.

#### 4.1.2.2. Erodarea

Fiind date mulțimile  $X$  și  $B$  din  $E^2$ , erodarea lui  $X$  cu  $B$ , notată  $X \ominus B$ , este definită prin:

$$X \ominus B = \{d \in E^2 \mid d + b \in X \quad \forall b \in B\} \quad (4.14)$$

Aceasta înseamnă că fiecare punct  $d$  al mulțimii (imaginii)  $X$  este testat. Rezultatul erodării este dat de acele puncte  $d$  pentru care toate valorile posibile  $d+b$  sunt incluse în  $X$ . Figura 4.6 prezintă un exemplu de mulțime de puncte  $X$  erodată cu elementul structurant  $B$ .

$$\begin{aligned} X &= \{(0,1), (1,1), (2,1), (3,0), (3,1), (3,2), (3,3), (4,1)\} \\ B &= \{(0,0), (0,1)\} \\ X \ominus B &= \{(3,0), (3,1), (3,2)\} \end{aligned}$$

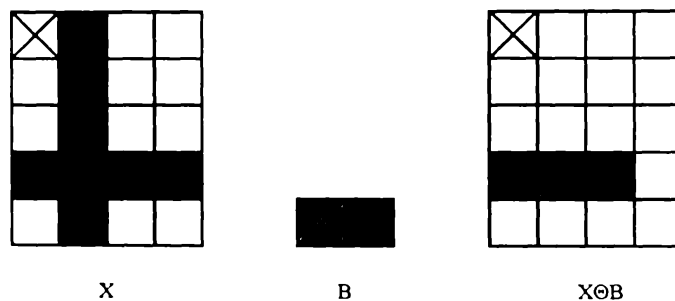


Fig. 4.6. Erodarea

Există o definiție echivalentă a erodării [63] care poate exprima mai bine proprietățile acestei operațiuni:

$$X \ominus B = \{d \in E^2 \mid B_d \subseteq X\} \quad (4.15)$$

ceea ce înseamnă că erodarea lui  $X$  cu  $B$  este mulțimea tuturor punctelor  $d$  astfel încât  $B$ , translatată cu  $d$ , să fie inclusă în  $X$ .

Implementarea erodării poate fi simplificată dacă se observă că o imagine  $X$  erodată cu elementul structurant  $B$  poate fi exprimată ca intersecție a tuturor translațiilor imaginii  $X$  prin vectorul  $-b \in B$ :

$$X \ominus B = \bigcap_{b \in B} X_{-b} \quad (4.16)$$

Figura 4.7 prezintă un exemplu [63] de erodare cu două elemente structurante diferite.

Erodarea este utilizată pentru a simplifica structura unui obiect. Obiectele sau părțile obiectelor având lățimea egală cu unu vor dispărea. Se poate astfel realiza descompunerea unor obiecte complicate în mai multe obiecte simple.

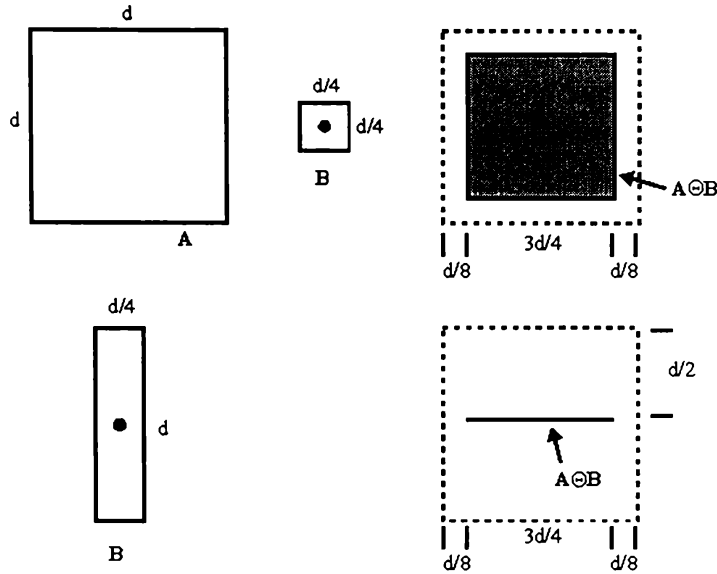


Fig. 4.7. Exemplu de erodare cu două elemente structurante diferite

Dacă punctul reprezentativ este parte a elementului structurant, erodarea este o transformare *anti-extensivă*, adică

$$\text{dacă } (0,0) \in B, \text{ atunci } X \ominus B \subseteq X. \quad (4.17)$$

Erodarea este de asemenea, ca și dilatarea, o *transformare invariantă*, ceea ce înseamnă că:

$$X_h \ominus B = (X \ominus B)_h \quad (4.18)$$

$$X \ominus B_h = (X \ominus B)_{-h} \quad (4.19)$$

și o transformare *crescătoare*:

$$\text{dacă } X \subseteq Y \text{ atunci } X \ominus B \subseteq Y \ominus B \quad (4.20)$$

Dacă  $B$  și  $D$  sunt elemente structurante, iar  $D$  este inclus în  $B$ , atunci erodarea cu  $B$  este mai puternică decât cea cu  $D$ , adică:

$$\text{dacă } D \subseteq B, \text{ atunci } X \ominus B \subseteq X \ominus D \quad (4.21)$$

Această proprietate permite introducerea unei relații de ordine pe mulțimea erodărilor, în conformitate cu elementele structurante de aceeași formă, dar de dimensiuni diferite.

Spre deosebire de dilatare, erodarea nu este nici comutativă:

$$X \ominus B \neq B \ominus X \quad (4.22)$$

nici asociativă. În schimb, erodările succesive ale imaginii  $X$ , mai întâi cu elementul structurant  $B$  și apoi cu elementul structurant  $D$  pot fi echivalate cu erodarea imaginii  $X$  cu dilatarea elementelor structurante  $B$  și  $D$ :

$$(X \ominus B) \ominus D = X \ominus (B \oplus D) \quad (4.23)$$

Este interesant de observat că erodarea și dilatarea nu sunt operații inverse una alteia. Deși există anumite situații în care o erodare poate reface exact rezultatul unei dilatări, acest lucru nu este în general adevărat. Această proprietate poate fi utilizată pentru a realiza operații utile asupra imaginilor. Totuși, erodarea și dilatarea sunt *duale* una alteia în raport cu complementarea și reflectarea mulțimilor. Prin urmare,

$$(A \ominus B)^c = A^c \oplus \hat{B} \quad (4.24)$$

Aceasta înseamnă că imaginea complementară a unei erodări este egală cu dilatarea imaginii complementare cu elementul structurant reflectat. Dacă elementul

structurant este simetric, atunci reflectarea nu schimbă elementul structurant și ecuația (4.24) poate fi interpretată în modul următor: complementarea erodării unei imagini este egală cu dilatarea fundalului respectivei imagini. Demonstrația dualității erodare-dilatare este destul de simplă și poate fi găsită în [63], [211].

#### 4.1.2.3. Deschiderea și închiderea

Practic, dilatarea și erodarea sunt utilizate de obicei împreună, fie prin determinarea dilatării unei imagini urmate de erodarea rezultatului obținut, fie prin erodarea imaginii, urmate de dilatare. În ambele cazuri, rezultatul aplicării iterative a dilatării și erodării este o eliminare a unor detalii specifice din imagine, mai mici decât elementul structurant, fără distorsionarea globală a caracteristicilor nesuprimate din imagine. Este în particular important faptul că transformările de imagine care utilizează în mod iterativ dilatarea și erodarea sunt *idempotente*, adică repetarea lor nu modifică în continuare rezultatele anterior obținute prin transformare. Importanța practică a transformărilor idempotente este că ele realizează etape de analiză a imaginilor, prin faptul că permit descrierea naturală a formelor prin specificarea elementelor structurante cu care, prin deschidere sau închidere, rămân nemodificate. Funcționalitatea lor este foarte apropiată de descrierea unui semnal prin banda sa. Filtrarea morfologică a unei imagini printr-o operație de deschidere sau închidere corespunde cu nerealizabila filtrare trece-bandă ideală în cazul filtrării lineare convenționale (după ce o imagine a fost filtrată trece-bandă ideal, o nouă filtrare ideală trece-bandă nu mai modifică rezultatul).

Această proprietate motivează importanța definirii operațiilor de deschidere și închidere, concepte studiate pentru prima dată de Matheron.

Cunoscând erodarea mulțimii  $X$  cu  $B$ , nu este în general posibil să se reconstituie mulțimea originală prin dilatarea lui  $X \ominus B$  cu  $B$ . În realitate, noua mulțime care se obține filtrează o submulțime a lui  $X$ , care conține o serie de detalii morfologice și proprietăți de mărime.

*Deschiderea* imaginii  $X$  cu elementul structurant  $B$  se notează cu  $X \circ B$  și se definește ca:

$$X \circ B = (X \ominus B) \oplus B \quad (4.25)$$

ceea ce înseamnă că deschiderea lui  $X$  cu  $B$  este simpla erodare a lui  $X$  cu  $B$ , urmată de o dilatare a rezultatului cu  $B$ .

*Închiderea* unei imagini  $X$  cu elementul structurant  $B$  este notată  $X \bullet B$  și este definită ca:

$$X \bullet B = (X \oplus B) \ominus B \quad (4.26)$$

ceea ce înseamnă că închiderea lui  $X$  cu  $B$  este simpla dilatare a lui  $X$  cu  $B$ , urmată de o erodare a rezultatului cu  $B$ .

Dacă o imagine  $X$  este neschimbată prin deschiderea cu elementul structurant  $B$ , ea se numește *deschisă în raport cu  $B$* . În mod similar, dacă o imagine  $X$  este neschimbată prin închiderea cu  $B$ , ea este numită *închisă în raport cu  $B$* .

Deschiderea și închiderea cu un element structurant izotrop este utilizată pentru eliminarea unor detalii specifice din imagine, mai mici decât elementul structurant, fără distorsionarea formei globale a obiectului. Închiderea realizează conectarea obiectelor apropiate unul de altul, umple micile goluri și netezește marginile obiectelor prin umplerea "golfurilor" înguste. Termenii "mici" și "înguste" sunt legați de dimensiunile elementului structurant.



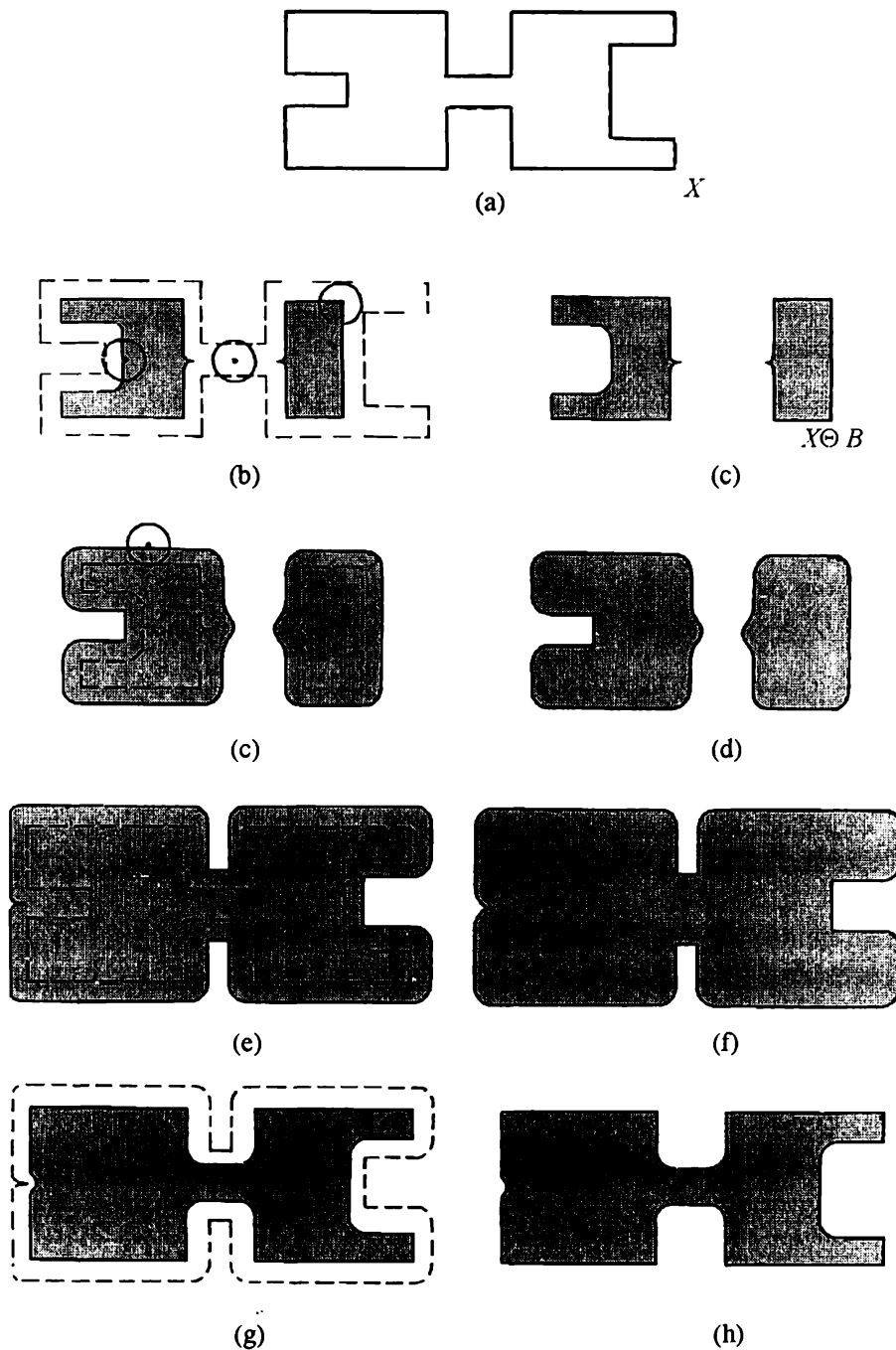


Fig. 4.8. Ilustrarea operațiilor de deschidere și închidere

Figura 4.8 ilustrează deschiderea și închiderea unei mulțimi  $X$  cu un element structurant disc [63]. Se poate observa din acest exemplu că operațiunea de deschidere duce la netezirea conturilor imaginii, la ruperea micilor istmuri și la eliminarea neregularităților fine. Operațiunea de închidere conduce de asemenea la netezirea unor porțiuni de contur, dar contrar deschiderii, ea contopește micile întreruperi și golurile lungi și înguste care descriu forma, elimină micile goluri și umple spărturile din contur.

Deschiderea și închiderea au o interpretare geometrică simplă [63]. Să ne imaginăm elementul structurant  $B$  ca pe o minge plată care alunecă peste obiectul considerat. Conturul lui  $X \ominus B$  este dat de punctele de pe conturul lui  $B$  care se găsesc la limită în interiorul lui  $X$  atunci când  $B$  alunecă de-a lungul interiorului acestui contur. Se

observă că toate colțurile exterioare sunt rotunjite, în timp ce colțurile interioare nu sunt afectate. Elementele ieșite în exterior cu dimensiunea mai mică decât a discului alunecător au fost eliminate. Această proprietate geometrică de “potrivire” a discului alunecător în conturul obiectului conduce la o formulare bazată pe teoria mulțimilor pentru operațiunea de deschidere. Deschiderea lui  $X$  cu  $B$  este obținută prin determinarea reuniunii tuturor translațiilor lui  $B$  care sunt cuprinse (care se “potrivesc”) în  $X$ . Aceasta înseamnă că deschiderea poate fi exprimată prin

$$X \circ B = \cup \{(B)_h \mid (B)_h \subset X\} \quad (4.27)$$

Figura 4.9 (adaptată după [63], [66] și [211]) ilustrează acest concept, utilizând un element structurant necircular.

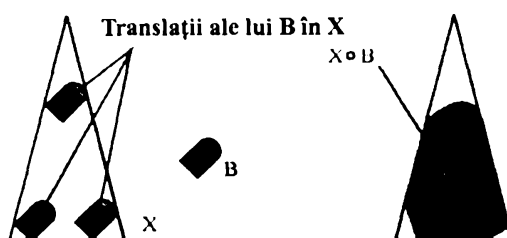


Fig. 4.9. Interpretarea geometrică a deschiderii

Închiderea are o interpretare geometrică similară, cu excepția faptului că de această dată, utilizând tot interpretarea geometrică cu discul alunecător, elementul structurant  $B$  este translatat pe exteriorul conturului. Considerând exemplul din figura 4.8, se observă că în acest caz colțurile interioare sunt rotunjite, în timp ce colțurile exterioare rămân nemodificate. Intrând din partea stângă a conturului lui  $X$  este redus semnificativ în mărime, datorită dimensiunilor mai mari ale elementului structurant. Operațiunea de închidere poate fi exprimată ca

$$X \bullet B = \cup \{(B)_h \mid (B)_h \cap X \neq \emptyset\} \quad (4.28)$$

Figura 4.10 ilustrează această proprietate.

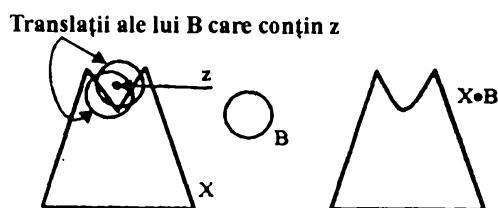


Fig. 4.10. Interpretarea geometrică a închiderii

Ca și în cazul dilatării și erodării, deschiderea și închiderea sunt *duale* în raport cu complementarea și reflectarea mulțimilor. Aceasta înseamnă că:

$$(X \bullet B)^c = X^c \circ \hat{B} \quad (4.29)$$

O altă proprietate semnificativă este aceea că deschiderile și închiderile utilizate iterativ sunt *idempotente*, adică reaplicarea acestor transformări nu modifică rezultatul anterior. Formal se poate scrie că:

$$X \circ B = (X \circ B) \circ B \quad (4.30)$$

$$X \bullet B = (X \bullet B) \bullet B \quad (4.31)$$

Rezultă din proprietatea de invarianță la translație a dilatării și erodării că atât deschiderea cât și închiderea sunt și ele *invariante la translație*.

Spre deosebire de dilatare și erodare, deschiderea și închiderea sunt *invariante la translatarea elementelor structurante*. Prin urmare,

$$X \circ (B)_h = X \circ B \quad (4.32)$$

$$X \bullet (B)_h = X \bullet B \quad (4.33)$$

#### 4.1.2.4. Transformarea “Hit-or-Miss”

Transformarea hit-or-miss este un operator morfologic proiectat pentru a localiza forme simple într-o imagine. Ea se bazează pe erodare, lucru natural deoarece erodarea unei imagini  $X$  cu un element structurant  $B$  constă doar din acei pixeli în care elementul structurant  $B$  este conținut în  $X$  sau se “potrivește” peste o mulțime de pixeli dintr-o mică regiune a lui  $X$ . Totuși, erodarea include și zone în care pixelii de fundal ai respectivei regiuni nu se suprapun peste cei ai lui  $B$ , locații care în mod normal nu sunt percepute ca o “potrivire”. De aceea era necesară o operație care să selecteze pixelii suprapuși ai lui  $B$  și  $X$  atât în prim-plan cât și în fundal.

“Potrivirea” pixelilor din prim-planul lui  $B$  peste cei ai lui  $X$  constituie o “suprapunere” (hit) și se realizează printr-o simplă erodare  $X \ominus B$ . Pixelii de fundal ai lui  $X$  sunt cei aflați în  $X^c$  și, deși poate fi utilizat  $B^c$  drept fundal al lui  $B$ , o abordare mai flexibilă constă în specificarea explicită a pixelilor de fundal printr-un nou element structurant  $B_2$ . O “potrivire” în fundal este numită “pierdere” (miss) și este determinată ca  $X^c \ominus B_2$ . Dacă dorim localizarea elementelor care aparțin atât de “hit” cât și de “miss”, va trebui să selectăm pixelii care aparțin lui  $(X \ominus B) \cap (X^c \ominus B_2)$ .

Dacă generalizăm notația punând  $B = (B_1, B_2)$ , unde  $B_1$  este mulțimea formată din elementele lui  $B$  asociate cu un obiect, iar  $B_2$  este mulțimea formată din elementele lui  $B$  asociate cu fundalul corespunzător. În acest caz  $B$  este numit *element structurant compus*. Cu această notație, putem defini transformarea hit-or-miss ca:

$$X \otimes B = (X \ominus B_1) \cap (X^c \ominus B_2) \quad (4.34)$$

Utilizând definiția diferenței mulțimilor și relația de dualitate între erodare și dilatare, putem rescrie ecuația (4.34) ca:

$$X \otimes B = (X \ominus B_1) - (X \oplus \hat{B}_2) \quad (4.35)$$

Astfel, mulțimea  $X \otimes B$  conține toate punctele în care, simultan,  $B_1$  realizează o suprapunere (“hit”) în  $X$ , iar  $B_2$  găsește una în  $X^c$ .

Figura 4.11 prezintă un exemplu de utilizare a transformării hit-or-miss pentru detectarea colțurilor din dreapta sus în imaginea originală  $X$ .

Imaginea originală  $X$  poate fi interpretată ca fiind suprapunerea a două pătrate. Un colț este definit ca fiind unghiul drept format de pixelul din colț și cei doi pixeli imediat vecini, unul mai jos și unul la stânga, după cum se vede din figura 4.11.b). Figura mai prezintă și porțiunea de “hit” a operațiunii (c), complementara imaginii (d) și elementul structurant utilizat pentru modelarea fundalului (e), porțiunea “miss” a operațiunii (f) și rezultatul intersecției porțiunilor “hit” și “miss”. Ambii pixeli ai rezultatului corespund colțurilor din imagine.

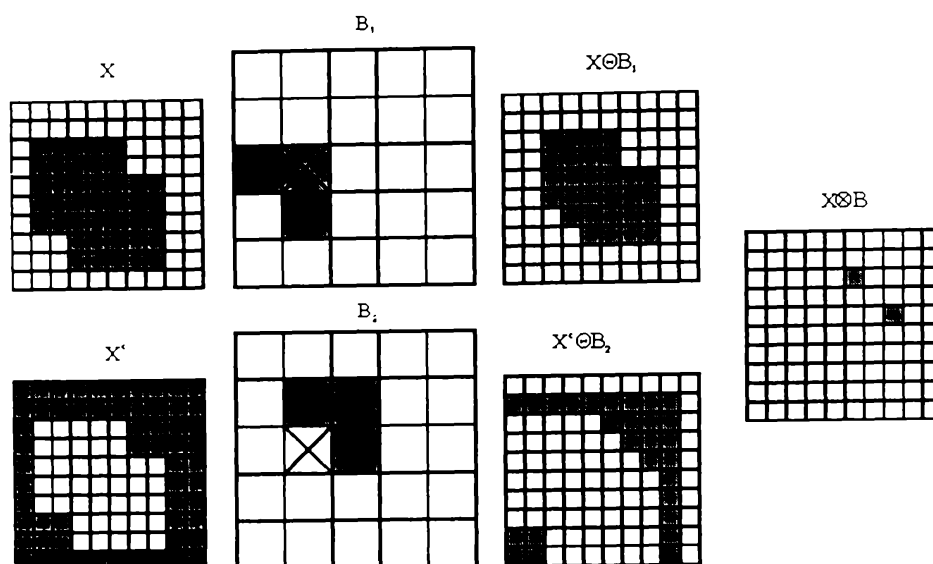


Fig. 4.11. Exemplu de transformare hit-or-miss

Este important de observat că elementul structurant al fundalului *nu este* complementul elementului structurant al prim-planului. Într-adevăr, dacă ar fi, atunci rezultatul ar fi o imagine goală deoarece nu există nici o suprapunere a formei sale specifice în imaginea complementară. Mulțimea pixelilor din elementul structurant al fundalului este formată din acei pixeli ai fundalului care *trebuie* să fie pixeli ai fundalului pentru ca o suprapunere să aibă loc. Supraspecificarea acestor pixeli conduce la mai puține suprapuneri, iar subspecificarea conduce la prea multe suprapuneri. Din acest motiv este necesară o selecție atentă a elementului structurant, posibil chiar prin recurgerea la experimentări.

#### 4.1.2.5. Principalele clase de elemente structurante

După cum s-a arătat, ideea de bază a morfologiei matematice este de a analiza o imagine cu un element structurant având o formă cunoscută. Fiecare element structurant dă informații în concordanță cu geometria și topologia sa.

În concordanță cu exemplele anterioare de erodare și dilatare, se poate observa că rezultatul transformării cu un element structurant de tip linie este foarte diferit de cel obținut printr-o erodare sau o dilatare cu un element structurant de tip cerc sau pătrat. Rezultatele pot fi comparate pe o singură direcție. Cazul utilizării unui element structurant format din două puncte este foarte diferit, rezultatul depinzând atât de distanța dintre cei doi pixeli ai elementului structurant și de dimensiunile componentelor imaginii  $X$ , cât și de orientarea celor două puncte.

Elementele structurante nu se aleg aleator. Din mai multe considerente, incluzând interpretarea și ușurința implementării, ele sunt alese cât mai simple posibil. Erodarea cu un segment de linie depinde de direcția acesteia, deoarece este vorba de un element anizotrop. Aceeași observație este valabilă și pentru elemente structurante eliptice, dar rezultatele sunt mult mai dificil de interpretat. Rezultatele erodării cu un cerc sau un segment de dreaptă pot fi comparate, ambele elemente fiind convexe. Acest lucru nu este valabil în cazul elementelor structurante de tip bi-punct, deoarece acestea nu sunt mulțimi convexe.

În cele mai multe cazuri putem alege elementele structurante în conformitate cu proprietățile lor: convexe sau ne-convexe, izotrope sau ne-izotrope. Figura 4.12 prezintă principalele elemente structurante care sunt utilizate în spațiul euclidian  $R^2$ , cel mai important pentru analiza imaginilor.





	convexe	neconvexe
izotrope	 suprafată	 contur
neizotrope	 segment	 bi-punct

Fig. 4.12. Tipuri de elemente structurante

Alegerea elementului structurant este o problemă care depinde de aplicația care trebuie rezolvată. De exemplu cercul, cel mai convex și mai izotrop element structurant, este foarte util pentru determinarea dimensiunilor obiectelor. Segmentul de dreaptă este cel mai anizotrop dintre toate elementele structurante convexe, fiind foarte util pentru măsurătorile granulometrice. Elementul structurant de tip bi-punct este cel mai anizotrop și cel mai puțin convex. Erodarea cu un astfel de element structurant poate caracteriza dispersia unei structuri.

#### 4.1.2.6. Proprietăți generale ale operatorilor morfologici

Există mai multe proprietăți generale interesante ale operatorilor morfologici. În primul rând, toate transformările morfologice sunt nereversibile (cu excepția unor cazuri particulare, pentru anumite subclase de mulțimi precum cele invariante). De fapt, ideea restaurării imaginilor nu este relevantă în acest caz; dimpotrivă, principala caracteristică a morfologiei matematice este de a stabili că imaginile studiate prezintă o cantitate prea mare de informație și că scopul oricărei prelucrări morfologice este de a *utiliza pierderea de informație* în urma transformărilor succesive.

Cele mai importante patru proprietăți generale, care stau la baza oricărei analize morfologice, sunt prezentate în continuare. Să considerăm  $\psi$  drept simbol generic al unei transformări morfologice.

##### a) Monotonie

Transformarea  $\psi$  este crescătoare atunci când ea conservă incluziunea, adică atunci când

$$X \subset Y \Rightarrow \psi(X) \subset \psi(Y), \quad \forall X, Y \in E \quad (4.36)$$

##### b) Anti-extensivitate

Transformarea  $\psi$  este anti-extensivă atunci când ea contractă pe  $X$ , adică atunci când

$$\psi(X) \subset X, \quad \forall X \in E \quad (4.37)$$

c) *Idempotența*

Transformarea  $\psi$  este idempotentă atunci când rezultatul  $\psi(X)$  rămâne neschimbat prin reaplicarea transformării, adică atunci când

$$\psi[\psi(X)] = \psi(X), \quad \forall X \in E \quad (4.38)$$

d) *Omotopia*

Mulțimea  $E$  este considerată în acest caz ca un spațiu topologic. Totuși, din motive de simplitate, ne vom limita la mulțimile mărginite ale planului. Fiecărei mulțimi mărginite  $K$  i se asociază *arborele omotopic*, al cărui trunchi corespunde fundalului imaginii, ramurile de primul ordin corespund obiectelor (regiunilor), iar ramurile de ordinul doi corespund golurilor din obiecte, ș.a.m.d.

O transformare este *omotopică* atunci când ea conservă arborele omotopic al mulțimii  $K$ , ceea ce înseamnă că ea nu modifică relația de proximitate între regiuni și goluri în imagine. Omotopia este o relație mai restrictivă decât conectivitatea: un disc și un inel sunt ambele mulțimi conexe, dar nu sunt omotopice.

Figura 4.13 prezintă un exemplu [232] de arbore omotopic corespunzător la două imagini diferite.

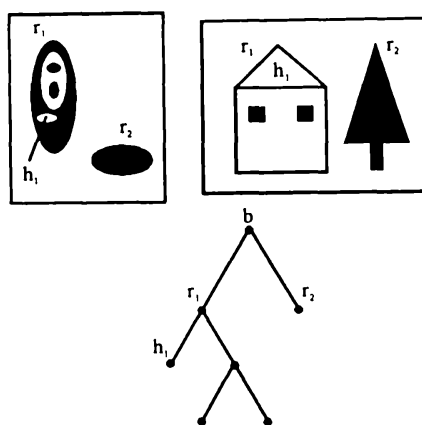


Fig. 4.13. Același arbore omotopic pentru două imagini diferite

Aceste patru proprietăți stau la baza clasificărilor pe criterii morfologice. Tabelul 4.1 prezintă situația celor patru proprietăți în cazul transformărilor morfologice euclidiene de bază, care au fost deja introduse, ca și pentru câteva alte transformări geometrice utile, din care unele vor fi prezentate mai târziu.

Tabelul 4.1. Transformările morfologice euclidiene și proprietățile lor de bază

Transformarea	Proprietăți			
	Crescătoare	Extensivitate	Idempotență	Conservarea omotopiei
Subțierea și îngroșarea	Nu	Da	Nu	Nu
Subțierea secvențială, scheletul, watershed, erodarea extremă	Nu	Da	Da	Nu
Proiecția, filtrarea morfologică	Da	Nu	Da	Nu
Deschiderea, închiderea, limitarea, înfășurătoarea convexă	Da	Da	Da	Nu
Subțierea și îngroșarea omotopică și secvențială	Nu	Da	Da	Da
Erodarea și dilatarea când originea nu este conținută în elementul structurant, filtrarea mediană	Da	Nu	Nu	Nu

Transformarea	Proprietăți			
	Crescătoare	Extensivitate	Idempotență	Conservarea omotopiei
Erodarea și dilatarea când originea este conținută în elementul structurant	Da	Da	Nu	Nu
Deplasarea, similaritatea, afinitatea, simetria	Da	Nu	Nu	Da
Complementaritatea, Hit-or-Miss, extragerea contururilor	Nu	Nu	Nu	Nu

## 4.2. Algoritmi morfologici de bază

Atunci când se lucrează cu imagini binare, principala aplicație a morfologiei constă în extragerea componentelor de imagine care sunt utile pentru reprezentarea și descrierea formei. În particular, au fost dezvoltati algoritmi morfologici pentru extragerea contururilor, a componentelor convexe, a înfășurătorii convexe și a scheletului unei regiuni. De asemenea există metode utilizate în conjuncție cu acești algoritmi, ca pași de pre- sau post-procesare, pentru umplerea regiunilor, subțiere, îngroșare și curățire. Toate exemplele prezentate în continuare se referă la imagini binare și sunt prelucrate pentru a ilustra și clarifica fiecare proces morfologic introdus.

### 4.2.1. Extragerea contururilor

În secțiunea precedentă am prezentat doar operatorii morfologici care micșorează sau măresc obiectele. În cele ce urmează voi prezenta modalități de extragere a contururilor unui obiect, care sunt foarte utile în tehnicile avansate de compresie a imaginilor. Punctele de pe contur sunt acele puncte cărora le lipsește cel puțin unul dintre vecini. Uzual se definește o vecinătate tetra-conexă sau octo-conexă pe o rețea dreptunghiulară. Punctele de pe contur pot fi înlăturate prin erodarea obiectului cu un element structurant care conține toți vecinii posibili ai pixelului central, adică, pentru cazul octo-conectivității și a tetra-conectivității, respectiv:

$$B_1 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \text{ și } B_2 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad (4.39)$$

Conturul este obținut prin diferența de mulțimi între obiect și erodarea sa. Dacă notăm cu  $\beta(X)$  conturul mulțimii  $X$ , atunci

$$\beta(X) = X - (X \ominus B) \quad (4.40)$$

unde  $B$  este elementul structurant adecvat.

Prin utilizarea unora dintre proprietățile deja prezentate ale erodării morfologice, putem obține pentru ecuația (4.40) următoarea formă:

$$\beta(X) = X - (X \ominus B) = X \cap (X \ominus B)^c = X \cap (X^c \oplus B) \quad (4.41)$$

Prin urmare, putem interpreta diferența de mulțimi ca intersecție între obiect și fundalul său dilatat. Linia de contur arată duala conectivitate cu obiectul erodat. Dacă erodăm un obiect cu un element structurant octo-conectat, conturul va fi tetra-conectat și invers.

Un exemplu de extragere a conturului [63] este prezentat în figura 4.14.

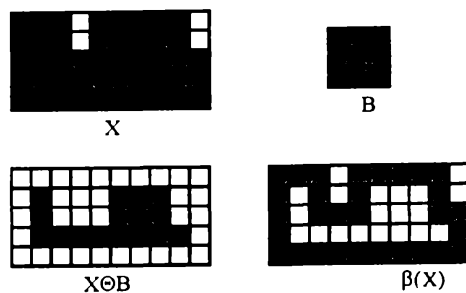


Fig. 4.14. Extragerea conturului

În acest exemplu este utilizat elementul structurant octo-conectat. Este interesant de menționat posibilitatea de utilizare și a unui alt element structurant, de exemplu un element de tip  $5 \times 5$  cu toate valorile pixelilor 1. Rezultatul va consta în obținerea unui contur cu grosimea între 2 și 3 pixeli.

#### 4.2.2. Umplerea regiunilor

Un algoritm simplu pentru umplerea regiunilor este bazat pe dilatarea, complementarea și intersecția mulțimilor, fiind prezentat în [66].

Figura 4.15 ilustrează acest algoritm.  $X$  semnifică o mulțime care conține o submulțime ale cărei elemente sunt punctele octo-conectate ale conturului unei regiuni. Scopul algoritmului este de a umple întreaga regiune cu pixeli de valoare 1, începând cu un punct  $p$  în interiorul conturului.

Deoarece, prin convenție, toate punctele care nu aparțin conturului au atribuită valoarea 0, vom da valoarea 1 punctului de start  $p$ . Procedura de umplere cu 1 a regiunii este următoarea:

$$Y_k = (Y_{k-1} \oplus B) \cap X^c \quad \text{unde } k = 1, 2, 3, \dots \quad (4.42)$$

iar  $Y_0 = p$  și  $B$  este elementul structurant prezentat în figura 4.15 (corespunzător tetra-conectivității). Algoritmul se încheie la iterația  $k$  dacă  $Y_k = Y_{k-1}$ . Reuniunea mulțimilor  $Y_k$  și  $X$  conține mulțimea umplută și conturul său.

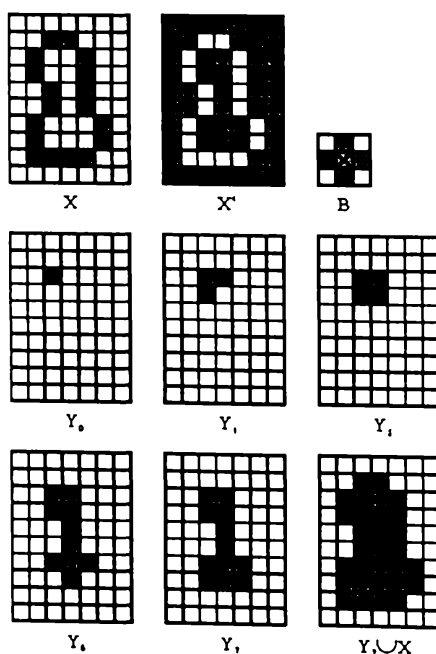


Fig. 4.15. Algoritm de umplere a regiunilor



Procesul de dilatare descris de ecuația (4.42) ar conduce la umplerea întregii imagini dacă s-ar desfășura nesupravegheat. Totuși, intersecția cu  $X^c$  făcută la fiecare pas limitează rezultatul la interiorul regiunii de interes (acest tip de procedură se numește uneori *dilatare condiționată* și va fi prezentat în următorul paragraf). Mecanismul de operare al ecuației (4.42) rezultă din figura 4.15.

Deși acest exemplu conține o singură submulțime, conceptul se poate aplica la orice număr finit de submulțimi, atâta timp cât este dat punctul interior fiecărui contur.

### 4.2.3. Dilatarea condiționată

Există anumite situații în care este de dorit ca dilatarea unui obiect să aibă loc în așa fel încât anumiți pixeli să rămână neafecțați de acest proces. Dacă, de exemplu, un obiect nu poate să ocupe anumite porțiuni ale unei imagini, atunci dilatarea obiectului trebuie să nu pătrundă în zona respectivă. Se spune că în acest caz a avut loc o *dilatare condiționată*. Aria interzisă a imaginii este specificată ca o a doua imagine, în care pixelii interziși sunt negri.

Notăția pentru dilatarea condiționată este  $X \oplus (B, Y)$ , unde  $B$  este elementul structurant utilizat în dilatare, iar  $Y$  este imaginea care reprezintă pixelii interziși. Expresia acestei operații este:

$$X \oplus (B, Y) = (X \oplus B) \cap Y \quad (4.43)$$

Prin iterarea operației se poate obține o secvență crescătoare a cărei limită este notată  $X \oplus (\{B\}, Y)$ :

$$X \oplus (\{B\}, Y) = \dots \left[ \left[ \left[ (X \oplus B) \cap Y \right] \oplus B \right] \cap Y \right] \oplus B \dots \quad (4.44)$$

Dilatarea condiționată este utilizată în două scopuri:

- Se poate dori excluderea din imaginea  $X$  a obiectelor care ating limitele  $R$  ale rastrului. Aceste obiecte sunt obținute prin dilatarea condiționată  $Z = R \oplus (B, X)$ .
- Dacă se dorește determinarea (și eventual umplerea) golurilor  $Y$  dintr-o imagine  $X$ , atunci se poate utiliza ecuația:  $Y = (R \oplus (B, X^c))^c$ .

### 4.2.4. Extragerea componentelor conectate

Conectivitatea dintre pixeli este un concept important utilizat în stabilirea conturilor obiectelor și determinarea componentelor regiunilor unei imagini, concept care a fost introdus în secțiunea 3.1. Practic, extragerea componentelor conectate dintr-o imagine binară este foarte importantă pentru multe aplicații de analiză automată a imaginilor. Fie  $Z$  o componentă conectată conținută într-o mulțime  $X$  și să presupunem că se cunoaște un punct  $p$  al lui  $Z$ .

În continuare, toate elementele lui  $Z$  sunt furnizate de următoarea expresie iterativă:

$$Y_k = (Y_{k-1} \oplus B) \cap X \quad \text{unde } k = 1, 2, 3, \dots \quad (4.45)$$

unde  $Y_0 = p$  și  $B$  este elementul structurant corespunzător, așa cum se vede din figura 4.16. Dacă  $Y_k = Y_{k-1}$ , algoritmul a ajuns la convergență și avem  $Z = Y_k$ .

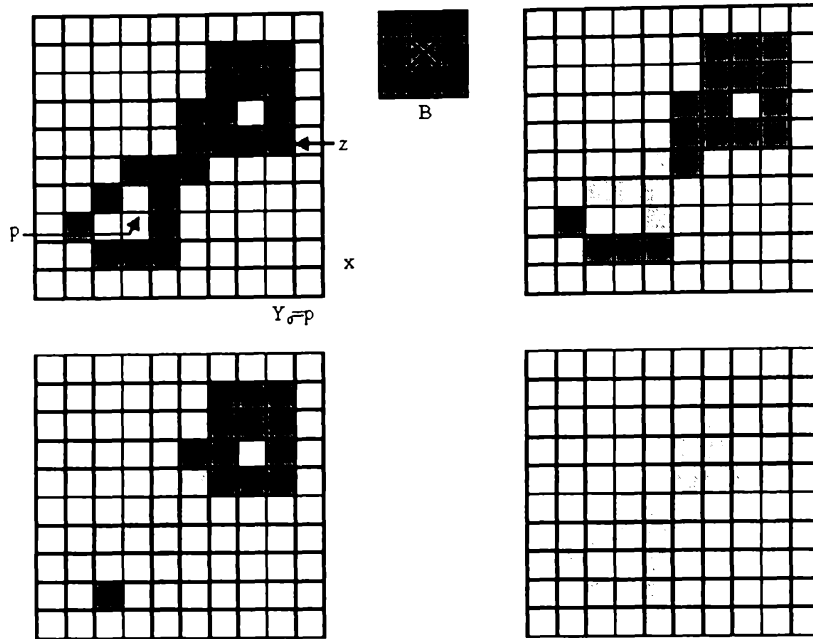


Fig. 4.16. Extragerea componentelor conectate

Ecuția (4.45) este similară ca formă cu ecuația (4.42). Singura diferență constă în utilizarea aici a mulțimii  $X$  în locul complementarei sale, deoarece toate elementele dorite sunt cu valoarea 1. Intersecția cu  $X$  la fiecare pas de iterație elimină dilatățile centrate pe elemente cu valoarea 0. Figura 4.16 ilustrează mecanismul ecuației (4.45). Este important de observat că forma elementului structurant presupune octo-conectivitatea între pixeli. Ca și în cazul algoritmului de umplere a regiunilor, rezultatul obținut poate fi aplicat la orice număr de mulțimi de componente conectate conținute în  $X$ .

#### 4.2.5. Înfașurătoarea convexă

Noțiunea de *înfașurătoare convexă* a unei mulțimi a fost introdusă în secțiunea 3.2.5 ca un descriptor util pentru imagine. În continuare voi prezenta un algoritm morfologic simplu pentru obținerea înfașurătorii convexe  $C(X)$  a unei mulțimi  $X$ . Fie  $B^i, i = 1, 2, 3, 4$  un număr de patru elemente structurante.

Procedura constă în implementarea ecuației:

$$Y_k^i = (Y \otimes B^i) \cup X \quad k = 1, 2, 3, \dots \quad (4.46)$$

cu  $Y_0^i = X$ . Fie acum  $D^i = Y_{\text{conv}}^i$ , unde indicele "conv" arată convergența în sensul că  $Y_k^i = Y_{k-1}^i$ . Înfașurătoarea convexă a mulțimii  $X$  este

$$C(X) = \bigcup_{i=1}^4 D^i \quad (4.47)$$

Procedura constă în aplicarea iterativă a transformării "hit-or-miss" asupra mulțimii  $X$  cu elementul structurant  $B^i$ . Atunci când nu mai apar modificări, se realizează reuniunea cu  $X$ , rezultatul fiind notat cu  $D^i$ . Procedura este repetată cu elementul structurant  $B^2$  ș.a.m.d. Reuniunea celor patru rezultate  $D$  va constitui înfașurătoarea convexă a mulțimii  $X$ .

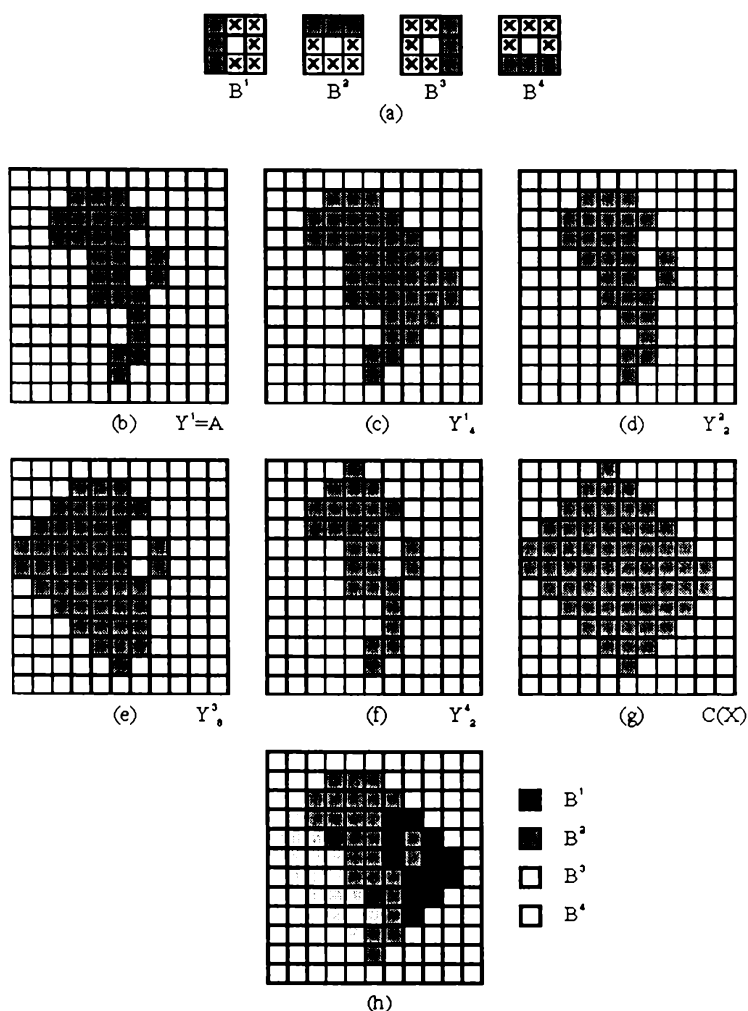


Fig. 4.17. Algoritm de extragere a înfășurătorii convexe

Figura 4.17 ilustrează procedura dată de ecuațiile (4.46) și (4.47). Figura 4.17.a) prezintă cele patru elemente structurante utilizate pentru extragerea înfășurătorii convexe (originea fiecărui element fiind în centrul său). Figura 4.17.b) prezintă mulțimea  $X$  a cărei înfășurătoare convexe se determină. Începând cu  $Y_0^1 = X$ , rezultatul a patru iterații ale ecuației (4.46) este prezentat în figura 4.17.c). În continuare, cu  $Y_0^2 = X$  și utilizând din nou ecuația (4.46) va rezulta mulțimea din figura 4.17.d), convergența obținându-se în doar doi pași. Următoarele două rezultate sunt obținute în mod similar. În final, prin reunirea mulțimilor din figurile 4.17.c), d), e) și f), se obține înfășurătoarea convexe prezentată în figura 4.17.g). Contribuția fiecărui element structurant este subliniată în mulțimea compusă prezentată în figura 4.17.h).

#### 4.2.6. Scheletul

Uneori este avantajos să se convertească un obiect în forma sa arhetip numită *schelet* (*skeleton*), introdusă în secțiunea 3.2.6. Conceptul de schelet poate fi abordat și prin prisma morfologiei matematice.

Serra [211], Maragos și Schafer [130] au demonstrat că scheletul unei mulțimi (regiuni)  $X$  poate fi exprimat prin operatori de erodare și deschidere. Dacă  $S(X)$  este scheletul lui  $X$ , atunci se poate demonstra [211] că:

$$S(X) = \bigcup_{k=0}^K S_k(X) \quad (4.48)$$

cu

$$S_k(X) = \bigcup_{k=0}^K \{(X \ominus kB) - [(X \ominus kB) \circ B]\} \quad (4.49)$$

unde  $B$  este un element structurant,  $(X \ominus kB)$  indică  $k$  erodări succesive ale lui  $X$ , iar  $K$  este ultima iterație înainte ca  $X$  să devină în urma erodării mulțimea vidă. Cu alte cuvinte,

$$K = \max\{k | (X \ominus kB) \neq \emptyset\} \quad (4.50)$$

Formularea dată de ecuațiile (4.48) și (4.49) arată că  $S(X)$ , scheletul lui  $X$ , poate fi obținut ca reuniune a *submulțimilor de schelete*  $S_k(X)$ . Se poate de asemenea arăta că  $X$  poate fi reconstruită din aceste submulțimi prin utilizarea ecuației:

$$X = \bigcup_{k=0}^K (S_k(X) \oplus kB) \quad (4.51)$$

unde  $(S_k(X) \oplus kB)$  reprezintă a  $k$ -a dilatare succesivă a lui  $S_k(X)$ , adică

$$(S_k(X) \oplus kB) = (\dots (S_k(X) \oplus B) \oplus B) \oplus \dots \oplus B \quad (4.52)$$

de  $k$  ori, iar limita de însumare  $K$  este cea precizată anterior.

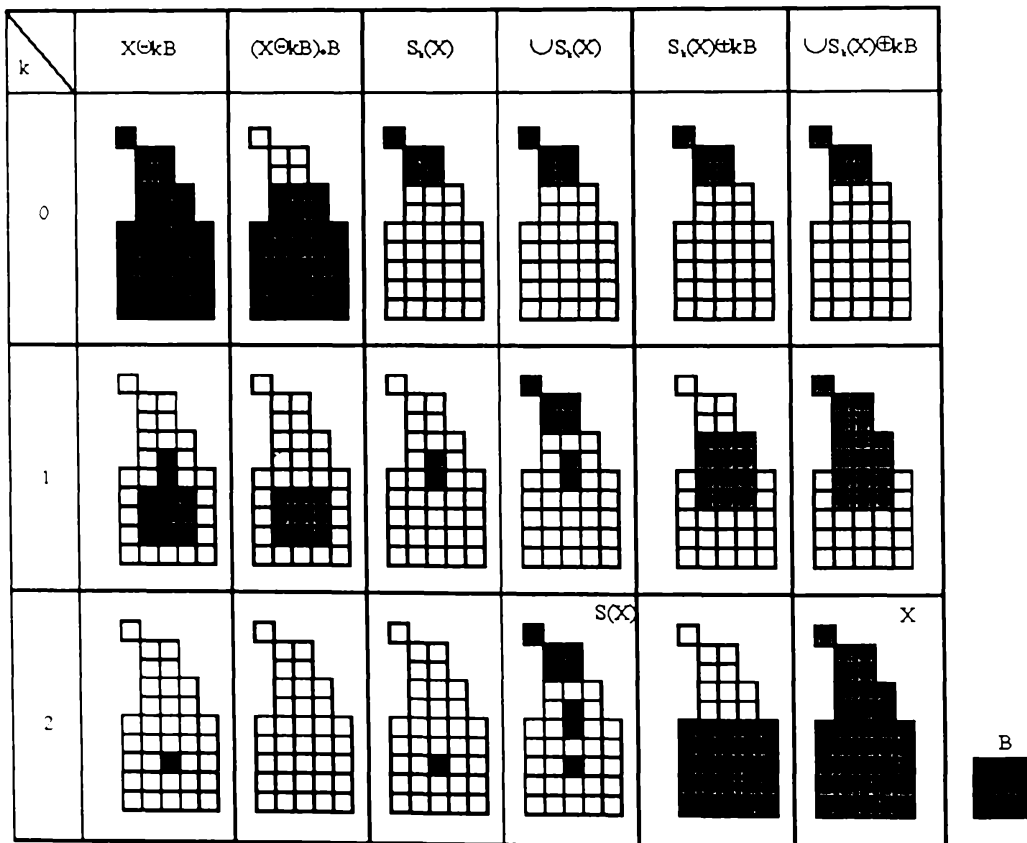


Fig. 4.18. Implementarea scheletului

Figura 4.18 ilustrează conceptul de schelet. Prima coloană prezintă mulțimea originală (sus) și rezultatele a două erodări cu elementul structurant  $B$ . Încă o erodare a

lui  $X$  ar conduce la mulțimea vidă, așa că în acest caz  $K=2$ . A doua coloană prezintă deschiderea cu elementul structurant  $B$  a mulțimilor din prima coloană. Aceste rezultate sunt ușor de explicat prin caracterizarea operatorului de deschidere făcută anterior. Cea de-a treia coloană conține diferențele dintre mulțimile din prima și a doua coloană.

Coloana a patra conține două schelete parțiale și rezultatul final (în partea de jos a coloanei). Acest rezultat nu este numai mai gros decât este necesar, dar totodată, lucru mai important, este și neconectat. Acest rezultat este de fapt normal, deoarece nici una din operațiile care definesc scheletul morfologic nu garantează conectivitatea. Morfologia permite o formulare elegantă a definiției scheletului, prin operatori de erodare și deschidere a mulțimii inițiale. Totuși sunt necesare formulări euristice, ca cea din capitolul 3, dacă scheletul trebuie să fie maximum de subțire, conectat și erodat minim.

Coloana a cincea prezintă pe  $S_0(X)$ ,  $S_1(X) \oplus B$  și  $(S_2(X) \oplus 2B) = (S_2(X) \oplus B) \oplus B$ . În sfârșit, ultima coloană prezintă reconstrucția mulțimii  $X$  ca reuniune a submulțimilor de schelete dilatate din coloana a cincea.

Din punctul de vedere al algoritmilor morfologici este important ca scheletele să poată fi obținute prin transformări de erodare și deschidere. Un astfel de schelet are însă două proprietăți dezavantajoase pentru aplicațiile practice: în primul rând, el poate să nu fie omotopic și în al doilea rând, el constă din linii care pot fi mai late de un pixel.

În paragraful 4.2.9 voi arăta că scheletul este deseori înlocuit prin subțierea secvențială omotopică, care nu are cele două proprietăți menționate anterior.

Pe de altă parte, calculul acestui înlocuitor al scheletului omotopic este mai lent (tipic de zece ori) decât calculul scheletului conform definiției [211].

#### 4.2.7. Subțierea

*Subțierea* unei mulțimi  $X$  cu un element structurant  $B$ , notată cu  $XOB$ , poate fi definită prin transformarea hit-or-miss:

$$XOB = X - (X \otimes B) = X \cap (X \otimes B)^c \quad (4.53)$$

O exprimare mai utilă pentru subțierea simetrică a mulțimii  $X$  se bazează pe o *secvență* de elemente structurante:

$$\{B\} = \{B^1, B^2, B^3, \dots, B^n\} \quad (4.54)$$

unde  $B^i$  este o versiune rotită a lui  $B^{i-1}$ . Utilizând acest concept, subțierea poate fi definită printr-o secvență de elemente structurante, ca

$$XO\{B\} = \left( \left( \dots \left( (XOB^1)OB^2 \right) \dots \right) OB^n \right). \quad (4.55)$$

Cu alte cuvinte, procesul constă în subțierea lui  $X$  în primul pas cu  $B^1$ , apoi în subțierea rezultatului în alt pas cu  $B^2$ , ș.a.m.d., până ce  $B$  este subțiat la un pas cu  $B^n$ . Întregul proces este repetat până ce nu mai apar modificări. Această operațiune este numită *subțiere secvențială*.

Figura 4.19 ilustrează procesul de subțiere. Figura 4.19.a) prezintă un set de elemente structurante frecvent utilizate pentru subțiere, iar figura 4.19.b) prezintă o mulțime  $X$  care urmează să fie subțiată. Figura 4.19.c) prezintă rezultatul subțierii într-un pas cu  $B^1$ , iar figurile 4.19.d)-k) prezintă rezultatele subțierii cu celelalte elemente structurante. Convergența a fost obținută după cea de-a doua subțiere cu  $B^4$ . Figura 4.19.k) prezintă rezultatul subțierii. În sfârșit, figura 4.19.l) prezintă rezultatul final.

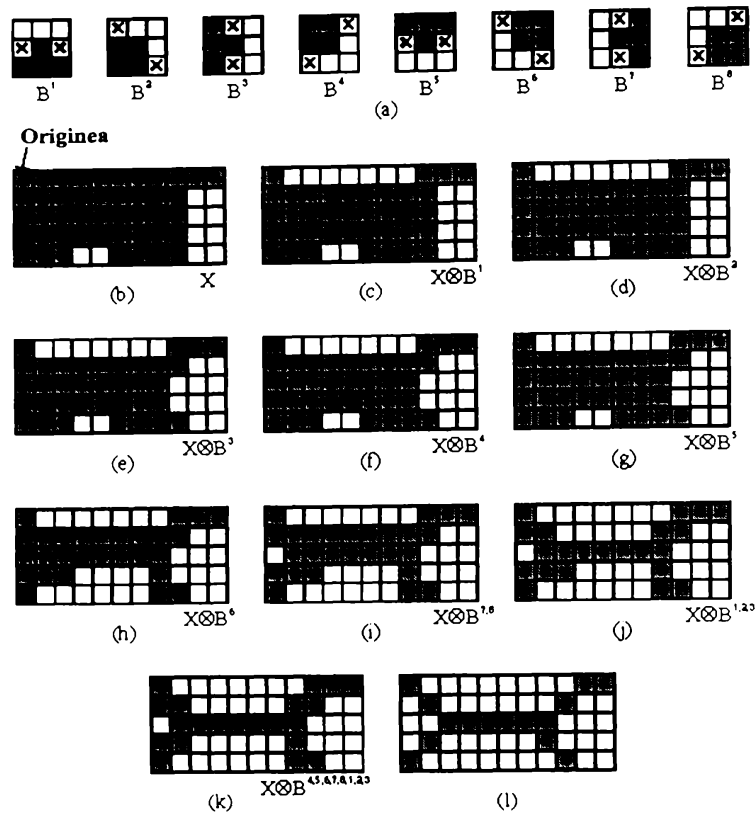


Fig. 4.19. Algoritm de subțiere

#### 4.2.8. Îngroșarea

Îngroșarea este operațiunea morfologică duală subțierii, este notată cu  $X * B$  și este definită prin expresia:

$$X * B = X \cup (X \otimes B) \quad (4.56)$$

unde  $B$  este un element structurant convenabil pentru îngroșare.

Ca și în cazul subțierii, îngroșarea poate fi de asemenea definită ca o operațiune secvențială, numită *îngroșare secvențială*:

$$X * \{B\} = \left( \left( \left( \left( X * B^1 \right) * B^2 \right) \dots \right) * B^n \right). \quad (4.57)$$

Elementele structurante utilizate pentru operațiunea de îngroșare au aceeași formă cu cele prezentate în figura 4.19.a) pentru cazul subțierii, dar cu valorile de "1" și "0" schimbate între ele. Totuși, în practică, se utilizează de multe ori un algoritm de îngroșare diferit. În locul procedurii normale, se subțiază fundalul mulțimii considerate și apoi se completează rezultatul. Cu alte cuvinte, pentru a îngroșa mulțimea  $X$ , se formează mulțimea complementară  $Y = X^c$ , se subțiază mulțimea  $Y$  și apoi se completează rezultatul. Figura 4.20 ilustrează această procedură. Figura 4.20.a) prezintă imaginea originală  $X$ , figura 4.20.b) prezintă complementara sa  $Y$ , figura 4.20.c) prezintă rezultatul subțierii lui  $Y$ , iar figura 4.20.d) prezintă mulțimea îngroșată obținută prin complementarea lui  $Y$  subțiat.

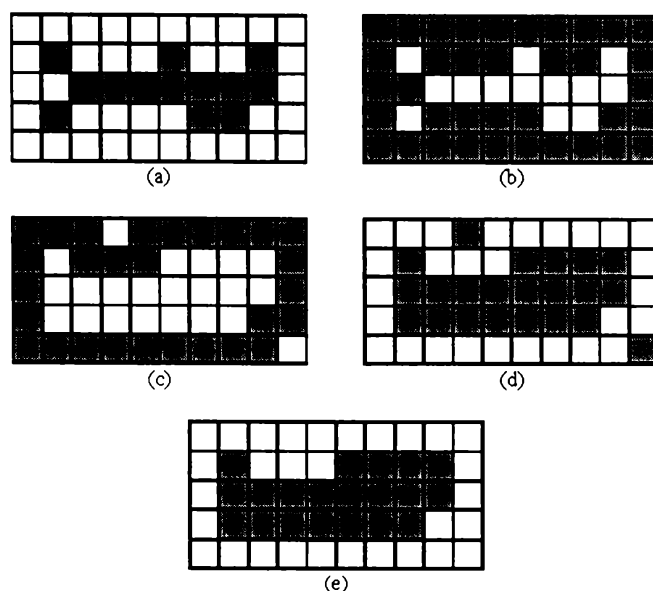


Fig. 4.20. Algoritmul de îngroșare

În funcție de natura mulțimii  $X$ , această procedură poate conduce la obținerea unor puncte neconectate, după cum se observă din figura 4.20.d). Din acest motiv, îngroșarea prin această metodă este urmată de obicei de un pas de post-procesare pentru înlăturarea punctelor neconectate. Se poate observa din figura 4.20.c) că fundalul subțiat formează o limită pentru procesul de îngroșare. Această proprietate utilă nu apare în implementarea directă a algoritmului de îngroșare conform ecuației (4.57) și constituie principalul motiv pentru utilizarea subțierii fundalului în scopul realizării îngroșării.

#### 4.2.9. Metode practice de realizare a operațiunilor de subțiere și îngroșare

În practică se utilizează mai multe secvențe de elemente structurante  $\{B'\}$ . Multe dintre ele sunt obținute printr-o rotație a elementului structurant în rastrul digital corespunzător (de exemplu hexagonal, pătrat sau octogonal). Aceste secvențe sunt numite uneori *alfabet Golay* și sunt prezentate pentru un rastru hexagonal în [211]. Voi prezenta în continuare elementele alfabetului Golay pentru un rastru pătrat și octogonal, indicele 4 semnificând un element pe rastrul pătrat, iar indicele 8 un element pe rastrul octogonal. Matricile  $3 \times 3$  vor fi prezentate doar pentru primele două rotații, din care se pot obține ușor și celelalte rotații.

Un element structurant compus va fi exprimat printr-o singură matrice. O valoare "1" în această matrice semnifică faptul că respectivul element aparține lui  $B^1$  (este o submulțime a obiectelor în transformarea hit-or-miss), iar valoarea "0" semnifică apartenența la  $B^2$ , adică la submulțimea pixelilor fundalului. Un "x" în matrice are semnificația unui element neutilizat în algoritmul de calcul, prin urmare valoarea sa este ne semnificativă.

Transformările secvențiale de subțiere și îngroșare converg către o imagine, numărul de iterații necesare depinzând atât de obiectele din imagine cât și de elementul structurant utilizat. Dacă două imagini succesive din secvență sunt identice, procesul de subțiere sau îngroșare este stopat.

*Subțierea secvențială cu elementul structurant L*

Acest tip de subțiere secvențială este important deoarece el se utilizează ca înlocuitor omotopic al scheletului. Imaginea finală subțiată constă doar din linii cu lățimea de un pixel sau din puncte izolate.

Elementul structurant  $L$  este dat de

$$L_{(4)}^1 = \begin{bmatrix} 0 & 0 & 0 \\ \times & 1 & \times \\ 1 & 1 & 1 \end{bmatrix}, \quad L_{(4)}^2 = \begin{bmatrix} \times & 0 & \times \\ 1 & 1 & 0 \\ 1 & 1 & \times \end{bmatrix}, \quad (4.58)$$

$$L_{(8)}^1 = \begin{bmatrix} 0 & 0 & 0 \\ \times & 1 & \times \\ 1 & 1 & 1 \end{bmatrix}, \quad L_{(8)}^2 = \begin{bmatrix} \times & 0 & 0 \\ 1 & 1 & 0 \\ \times & 1 & \times \end{bmatrix}, \quad (4.59)$$

*Subțierea secvențială cu elementul structurant E*

Să presupunem că a fost găsit un înlocuitor omotopic al scheletului prin elementul  $L_{(4)}$ . De obicei, scheletul este prea zimțat datorită punctelor ascuțite ale obiectului, dar este posibil ca scheletul să fie "netezit" prin subțiere secvențială cu elementul structurant  $E$ . Prin efectuarea a  $n$  iterații, mai multe puncte (al căror număr depinde de  $n$ ) de pe liniile cu lățime unu (precum și puncte izolate) sunt înlăturate de la capetele libere. Dacă subțierea cu elementul structurant  $E$  este făcută până ce imaginea nu se mai modifică, atunci vor rămâne doar contururile închise.

Elementul structurant  $E$  este dat prin

$$E_{(4)}^1 = \begin{bmatrix} \times & \times & \times \\ 0 & 1 & 0 \\ \times & 0 & \times \end{bmatrix}, \quad E_{(4)}^2 = \begin{bmatrix} \times & 0 & \times \\ 0 & 1 & \times \\ \times & 0 & \times \end{bmatrix}, \quad (4.60)$$

$$E_{(8)}^1 = \begin{bmatrix} \times & 1 & \times \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad E_{(8)}^2 = \begin{bmatrix} 0 & \times & \times \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad (4.61)$$

*Subțierea secvențială cu elementul structurant M*

Subțierea secvențială cu elementul structurant  $M$  este rareori utilizată pentru obținerea înlocuitorului omotopic al scheletului. Scheletul obținut este prea zimțat, de aceea elementul structurant  $L$  este mai des utilizat în acest scop. Singurul caz în care elementul  $M$  se comportă mai bine este acela în care fundalul conține puncte izolate.

Elementul structurant  $M$  este dat prin

$$M_{(4)}^1 = \begin{bmatrix} \times & 0 & \times \\ \times & 1 & \times \\ 1 & 1 & 1 \end{bmatrix}, \quad M_{(4)}^2 = \begin{bmatrix} \times & 0 & \times \\ 1 & 1 & 0 \\ 1 & 1 & \times \end{bmatrix}, \quad (4.62)$$

$$M_{(8)}^1 = \begin{bmatrix} \times & 0 & \times \\ 1 & 1 & 1 \\ \times & 1 & \times \end{bmatrix}, \quad M_{(8)}^2 = \begin{bmatrix} \times & 0 & 0 \\ 1 & 1 & 0 \\ \times & 1 & \times \end{bmatrix}, \quad (4.63)$$



*Subțierea și îngroșarea secvențială cu elementul structurant D*

Elementul structurant  $D$  este dat prin

$$D_{(4)}^1 = \begin{bmatrix} \times & 0 & \times \\ 0 & 1 & 1 \\ \times & 0 & \times \end{bmatrix}, \quad D_{(4)}^2 = \begin{bmatrix} 0 & 0 & \times \\ 0 & 1 & 1 \\ \times & 1 & 1 \end{bmatrix}, \quad (4.64)$$

$$D_{(8)}^1 = \begin{bmatrix} 0 & \times & \times \\ 0 & 1 & 1 \\ 0 & \times & \times \end{bmatrix}, \quad D_{(8)}^2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & \times \\ 0 & \times & 1 \end{bmatrix}, \quad (4.65)$$

Aplicațiile posibile ale elementului  $D$  sunt:

- Subțierea cu un element  $D$  înlocuiește toate elementele fără goluri cu un punct izolat. Doar contururile închise rămân neschimbate. Această operațiune este numită uneori *marcare omotopică*.
- Îngroșarea cu un element  $D$  crează o înfășurătoare pseudoconvexă a mulțimii  $X$ . Prefixul pseudo este utilizat deoarece arborele omotopic trebuie să fie modificat, procesul este stopat, iar mulțimea  $X$  este neconexă.

*Îngroșarea secvențială cu elementul structurant C*

Îngroșarea secvențială cu elementul structurant  $C$  crează înfășurătoarea convexă a mulțimii  $X$ . Această operațiune nu conservă omotopia.

Elementul structurant  $C$  este dat prin

$$C_{(4)}^1 = \begin{bmatrix} 1 & 1 & \times \\ 1 & 0 & \times \\ \times & \times & \times \end{bmatrix}, \quad C_{(4)}^2 = \begin{bmatrix} \times & 1 & 1 \\ \times & 0 & 1 \\ \times & \times & \times \end{bmatrix}, \quad (4.66)$$

$$C_{(8)}^1 = \begin{bmatrix} \times & 1 & \times \\ 1 & 0 & \times \\ \times & 1 & \times \end{bmatrix}, \quad C_{(8)}^2 = \begin{bmatrix} \times & 1 & \times \\ 1 & 0 & \times \\ 1 & \times & \times \end{bmatrix}, \quad (4.67)$$

**4.2.10. Curățarea**

Metodele de curățare sunt elemente ajutătoare esențiale pentru algoritmi de subțiere și scheletizare, deoarece aceste proceduri au tendința de a lăsa componente parazite care trebuie curățate printr-o postprocesare. Voi prezenta în cele ce urmează o problemă de curățare a imaginii [211] rezolvată prin morfologia matematică, pe baza mai multor tehnici din cele anterior descrise.

O metodă uzuală pentru recunoașterea automată a caracterelor scrise de mână constă în analiza formei scheletului fiecărui caracter. Aceste schelete sunt caracterizate de multe ori prin existența unor componente parazite, care apar în timpul procesului de erodare datorită neuniformităților de mișcare a mâinii la scrierea caracterelor. O tehnică morfologică pentru tratarea acestei probleme poate fi găsită în [211]. Ea este bazată pe presupunerea că lungimea componentelor parazite nu depășește trei pixeli.

Figura 4.21 prezintă scheletul caracterului "a" scris de mână. Componenta parazită din partea din stânga a caracterului este tipică pentru ceea ce trebuie înlăturat

din imagine. Soluția se bazează pe suprimarea unei ramuri parazite prin eliminarea succesivă a punctului său final. Desigur, acest proces conduce de asemenea la scurtarea (sau eliminarea) și a altor ramuri în caracter. În absența altor informații structurale, orice ramură cu lungimea de trei pixeli sau mai puțin este eliminată. Pentru o mulțime  $X$ , subțierea cu o secvență de elemente structurante proiectată să detecteze doar punctele finale, realizează rezultatul dorit. Fie, prin urmare

$$Y_1 = XO\{B\} \quad (4.68)$$

unde  $\{B\}$  semnifică secvența (4.54) prezentată în figura 4.21.b) și c). Secvența de elemente structurante constă din două structuri diferite, fiecare rotită cu  $90^\circ$  pentru a obține numărul total de opt elemente. Notația “ $\times$ ” din figura 4.21.b) semnifică o condiție de “nu contează”, în sensul că valoarea pixelului din respectiva locație poate fi “0” sau “1”, fără ca rezultatul să fie afectat.

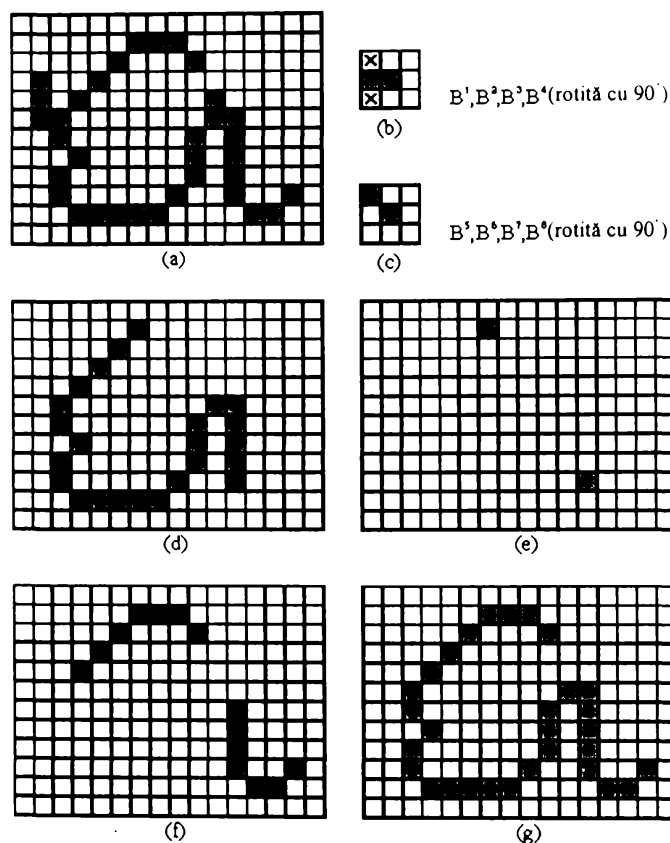


Fig. 4.21. Exemplu de curățare a imaginii

Aplicarea de trei ori a ecuației (4.68) asupra imaginii  $X$  conduce la mulțimea  $Y_1$  prezentată în figura 4.21.d). Următorul pas constă în “restaurarea” caracterului la forma sa originală, dar cu ramurile parazite înlăturate. Pentru a realiza acest lucru este necesar ca mai întâi să fie formată o mulțime  $Y_2$  care să conțină toate punctele extreme ale lui  $Y_1$ , așa cum rezultă din figura 4.21.e):

$$Y_2 = \bigcup_{k=1}^8 (Y_1 \otimes B^k) \quad (4.69)$$

unde  $B^k$  sunt elementele utilizate pentru detectarea extremelor care au mai fost prezentate anterior. Următorul pas este dilatarea de trei ori a punctelor extreme, prin utilizarea mulțimii  $X$  pentru delimitare:

$$Y_3 = (Y_2 \oplus H) \cap X \quad (4.70)$$

unde  $H$  este un element structurant de dimensiune  $3 \times 3$  format din pixeli de valoare 1. Ca și în cazul umplerii regiunilor și a extragerii componentelor conectate, acest tip de dilatare condiționată previne crearea de elemente cu valoarea 1 în afara regiunii de interes, așa cum rezultă din rezultatul prezentat în figura 4.21.f). În final, reuniunea mulțimilor  $Y_3$  și  $Y_1$  conduce la rezultatul final

$$Y_4 = Y_1 \cup Y_3 \quad (4.71)$$

așa cum rezultă din figura 4.21.g).

În cazurile mai complexe, utilizarea ecuației (4.70) conduce la selectarea unor capete de ramuri parazite. Acest lucru se poate întâmpla atunci când punctele extreme ale acestor ramuri sunt situate în apropierea scheletului. Deși ecuația (4.68) poate să le elimine, ele pot fi reselectate în timpul procesului de dilatare deoarece ele sunt puncte valide ale lui  $X$ . Cu excepția cazului în care toate elementele parazite sunt reselectate (rareori, dacă aceste elemente sunt scurte în comparație cu liniile utile), detectarea și eliminarea lor este ușoară deoarece ele constituie regiuni neconectate.

Este de presupus că ar trebui să existe și metode mai simple pentru rezolvarea acestei probleme. De exemplu, am putea ține evidența tuturor punctelor șterse pentru a putea reconecta punctele potrivite la punctele extreme rămase după aplicarea ecuației (4.68). Acest lucru este posibil, dar metoda anterioară are avantajul că soluția poate fi obținută prin metode morfologice simple. În situații practice, când există un set de operatori morfologici implementat, avantajul este că nu mai trebuie scriși alți algoritmi. Putem doar combina funcții morfologice în secvențe de operații.

### 4.3. Morfologie pe scară de gri

#### 4.3.1. Introducere

Teoria morfologiei matematice a fost extinsă de Sternberg [235] și Serra [211] pentru cazul imaginilor pe scară de gri. Operatorii binari de dilatare, erodare, închidere, și deschidere pot fi extinși în mod natural la imagini pe scară de gri prin utilizarea operațiunilor *min* și *max*. Acest tip de extindere poate fi vizualizată geometric cu ajutorul așa numitelor *umbre*, adică prin punctele de pe și de sub graful unei funcții.

O extensie a morfologiei matematice la laticelul complet a fost propusă recent de Serra [213], conducând la o abordare conceptual diferită față de morfologia pe scară de gri.

Utilizarea nivelurilor de gri multiple introduce o complicație enormă, atât din punct de vedere conceptual cât și din punct de vedere computațional. Un pixel poate lua în acest caz orice valoare întreagă, astfel încât o imagine nu mai poate fi considerată ca o mulțime. Conceptele introduse pentru morfologia binară se bazează pe mulțimi din spațiul euclidian bidimensional, care reprezintă imagini binare. Morfologia pe scară de gri ia în considerare mulțimi din spațiul euclidian tridimensional. Imaginile binare din spațiul euclidian bidimensional sunt văzute ca și tăieturi plane ale planului orizontal  $x,y$ . Mulțimile considerate în spațiul euclidian tridimensional au volume solide. De un interes particular în acest caz sunt *umbrele*, adică mulțimile solide care extind contururile neîntrerupte în jos, în direcția negativă a axei  $z$ . Detaliile privitoare la cea de-a treia dimensiune a umbrei pot fi determinate printr-un singur parametru, înălțimea  $z$  a umbrei la coordonatele  $(x, y)$  ale planului binar.

Umbra unei mulțimi  $X$  din spațiul tridimensional include atât mulțimea  $X$  cât și volumul punctelor de sub mulțime. Se consideră că umbra este indicată de o sursă de lumină punctuală situată la infinit pe direcția pozitivă a axei  $z$ . Umbra poate fi exprimată ca o transformare morfologică a mulțimii  $X$  cu elementul structurant  $Z$  compus din puncte ale axei  $z$ , incluzând originea. Umbra lui  $X$  este formată prin dilatarea mulțimii  $X$  cu  $Z$ .

O imagine pe scară de gri este o funcție pe scară de gri  $f(x,y)$ , definită în punctele spațiului euclidian bidimensional. O funcție pe scară de gri poate fi considerată în spațiul euclidian tridimensional ca o mulțime de puncte  $[x, y, f(x,y)]$ , imaginată ca o foaie subțire, ondulată, nu neapărat conexă. O imagine pe scară de gri  $f(x, y)$  este reprezentată în morfologia matematică printr-o umbră  $U[f]$  în spațiul euclidian tridimensional, unde un punct  $p = (x, y, z)$  aparține umbrei dacă și numai dacă  $z \leq f(x, y)$ . Semnificația umbrelor în prelucrarea imaginilor este aceea că ele rămân umbre în urma transformărilor morfologice uzuale ca reuniunea și intersecția, dilatarea și erodarea.

### 4.3.2. Operatori morfologici de bază pe scară de gri

În cele ce urmează vom utiliza funcții digitale de imagine de forma  $f(x, y)$  și  $b(x, y)$ , unde  $f(x, y)$  este imaginea de intrare, iar  $b(x, y)$  este un element structurant, el însuși o funcție subimagine. Se presupune că aceste funcții sunt discrete.

#### 4.3.2.1. Dilatarea pe scară de gri

Dilatarea pe scară de gri a lui  $f$  cu  $b$ , notată  $f \oplus b$ , este definită ca:

$$(f \oplus b)(s, t) = \max \{ f(s-x, t-y) + b(x, y) \mid (s-x), (t-y) \in D_f; (x, y) \in D_b \} \quad (4.72)$$

unde  $D_f$  și  $D_b$  sunt domeniile lui  $f$  și  $b$ , respectiv. Elementul structurant  $b$  este de această dată o funcție, nu o mulțime.

Condiția ca parametrii de deplasament  $(s-x)$  și  $(t-y)$  să fie conținuți în domeniul lui  $f$  este analoagă cu cea din definiția binară a dilatării, unde cele două mulțimi trebuie să se suprapună cel puțin printr-un element. În ecuația de definiție, operațiunea  $\max$  înlocuiește sumele de convoluție, iar adunarea înlocuiește produsul de convoluție.

Voi ilustra notațiile și mecanismul ecuației (4.72) prin intermediul funcțiilor simple unidimensionale. Pentru funcții de o variabilă, ecuația (4.72) se reduce la expresia:

$$(f \oplus b)(s) = \max \{ f(s-x) + b(x) \mid (s-x) \in D_f \text{ și } x \in D_b \} \quad (4.73)$$

Ca și în cazul convoluției, funcția  $f(s-x)$  realizează translația spre dreapta pentru  $s$  pozitiv și către stânga pentru  $s$  negativ. Condițiile ca valorile lui  $(s-x)$  să fie în domeniul lui  $f$ , iar valorile lui  $x$  să fie în domeniul lui  $b$ , implică ca  $f$  și  $b$  să aibă puncte de domeniu comune. Spre deosebire de cazul binar, de data aceasta este translatată în general funcția  $f$  în locul elementului structurant  $b$ . Ecuația (4.72) poate fi rescrisă astfel încât  $b$  să realizeze translația în locul lui  $f$ . Totuși, dacă domeniul  $D_b$  este mai mic decât  $D_f$  (condiție aproape întotdeauna întâlnită în practică), forma dată în ecuația (4.72) este mai simplă și conduce la același rezultat. Conceptual nu există diferență între translatarea lui  $f$  cu  $b$ , respectiv a lui  $b$  cu  $f$ .

Dilatarea este comutativă, astfel încât pentru calculul lui  $b \oplus f$  poate fi utilizată ecuația (4.72), cu interschimbarea lui  $f$  cu  $b$ .

Un exemplu de dilatare pe scară de gri este prezentat în figura 4.22.

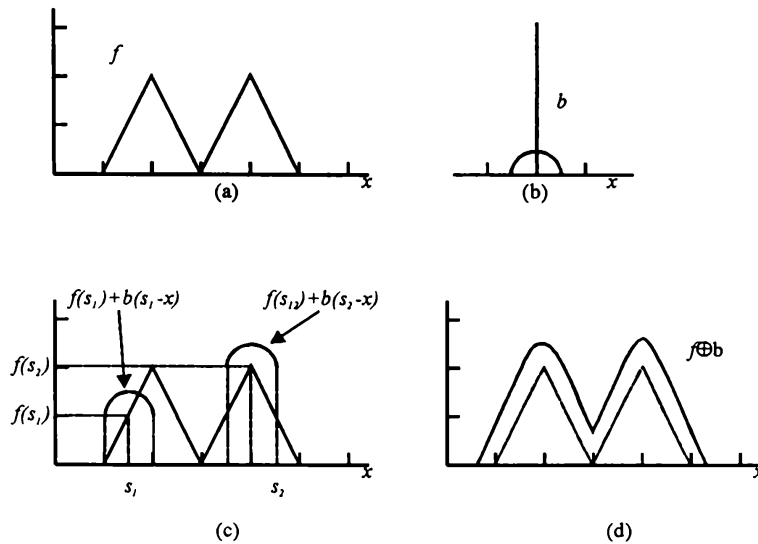


Fig. 4.22. Dilatare pe scară de gri

Deoarece dilatarea se bazează pe alegerea valorii maxime a lui  $f+b$  într-o vecinătate definită de forma elementului structurant, efectul general al aplicării unei dilatări pe o imagine pe scară de gri (figura 4.24.b) este:

- dacă toate valorile elementului structurant sunt pozitive, imaginea rezultată tinde să fie mai luminoasă decât imaginea originală;
- detaliile întunecate sunt fie reduse, fie eliminate, în funcție de modul în care valorile și formele lor sunt legate de elementul structurant utilizat pentru dilatare.

#### 4.3.2.2. Erodarea pe scară de gri

Erodarea pe scară de gri, notată cu  $f \ominus b$ , este definită prin

$$(f \ominus b)(s, t) = \min \{ f(s+x, t+y) - b(x, y) \mid (s+x, t+y) \in D_f; (x, y) \in D_b \} \quad (4.74)$$

unde  $D_f$  și  $D_b$  sunt domeniile lui  $f$  și  $b$ , respectiv. Condiția ca parametri de deplasament  $(s+x)$  și  $(t+y)$  să fie conținuți în domeniul lui  $f$  este analoagă cu condiția din cazul definiției binare a erodării, unde elementul structurant trebuia să fie conținut în întregime în mulțimea care urma să fie erodată.

Mecanismul este ilustrat în figura 4.23, prin erodarea funcției unidimensionale simple prezentată în figura 4.22.a) cu elementul structurant din figura 4.22.b).

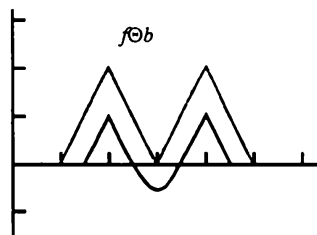


Fig. 4.23. Erodarea pe scară de gri

Pentru funcții de o variabilă, expresia erodării se reduce la:

$$(f \ominus b) = \min \{ f(s+x) - b(x) \mid (s+x) \in D_f \text{ și } x \in D_b \} \quad (4.75)$$

Funcția  $f(s+x)$  este trasată către stânga pentru  $s$  pozitiv și la dreapta pentru  $s$  negativ. Cerințele ca  $(s+x) \in D_f$  și  $x \in D_b$  implică ca domeniul lui  $b$  să fie complet conținut în deplasamentul lui  $f$ . În sfârșit, spre deosebire de cazul definiției binare a erodării, se translatează funcția  $f$  și nu elementul structurant  $b$ . Ecuația (4.74) poate fi scrisă în așa fel încât funcția trasată să fie  $b$ , rezultând o formă mai complicată a expresiei matematice, din punct de vedere al indicilor.

Așa cum arată ecuația (4.74), erodarea se bazează pe alegerea valorii minime a lui  $(f-b)$  într-o vecinătate definită de forma elementului structurant. Efectele generale ale aplicării operatorului de erodare asupra unei imagini pe scară de gri (figura 4.24.c) sunt:

- dacă toate componentele elementului structurant sunt pozitive, imaginea de ieșire va fi mai întunecată decât imaginea de intrare;
- efectul detaliilor luminoase din imaginea originală, mai mici ca suprafață decât elementul structurant, este redus; gradul de reducere este determinat de valorile nivelurilor de gri din jurul detaliilor luminoase, precum și de forma și amplitudinea valorilor componentelor elementului structurant însuși.



a)



b)



c)

Fig. 4.24. Exemplu de implementare a dilatării și erodării pe scară de gri

### 4.3.2.3. Deschiderea și închiderea pe scară de gri

Expresiile pentru deschiderea și închiderea imaginilor pe scară de gri au aceeași formă cu corespondentele lor binare. Deschiderea unei imagini  $f$  prin imaginea (elementul structurant)  $b$ , notată cu  $f \circ b$ , este

$$f \circ b = (f \ominus b) \oplus b \quad (4.76)$$

Ca și în cazul binar, deschiderea constă în erodarea lui  $f$  prin  $b$ , urmată de dilatarea rezultatului prin  $b$ .

Similar, închiderea lui  $f$  prin  $b$ , notată cu  $f \bullet b$ , este:

$$f \bullet b = (f \oplus b) \ominus b \quad (4.77)$$

Deschiderea și închiderea imaginilor pe scară de gri sunt duale în raport cu complementarea și reflectarea. Prin urmare,

$$(f \bullet b)^c = f^c \circ \hat{b}. \quad (4.78)$$

Deoarece  $f^c = -f$ , ecuația (4.77) poate fi scrisă și ca

$$-(f \bullet b) = (-f \circ \hat{b}). \quad (4.79)$$

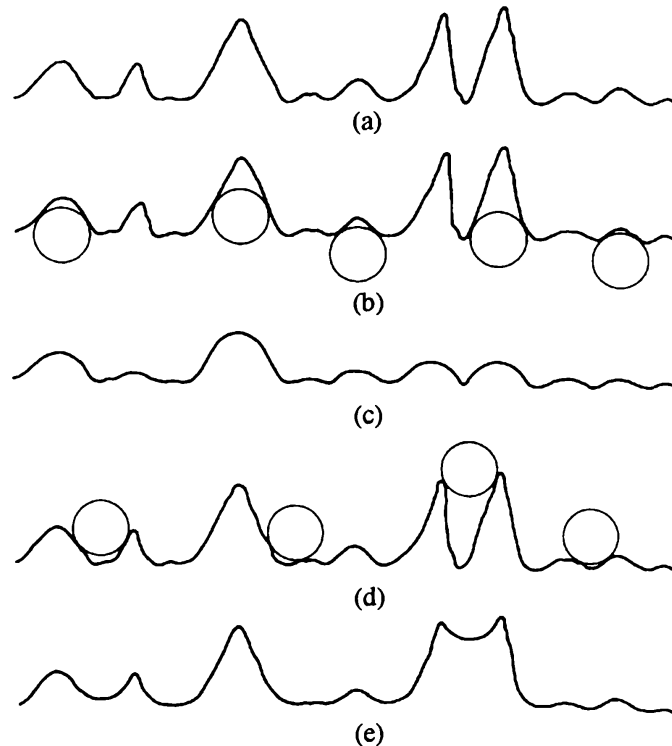


Fig. 4.25. Deschiderea și închiderea pe scară de gri

Operațiunile de deschidere și închidere a imaginilor au o interpretare geometrică simplă. Să considerăm o funcție imagine  $f(x, y)$  în perspectivă tridimensională, cu axele  $x$  și  $y$  fiind tradiționalele coordonate spațiale, cea de-a treia axă fiind luminozitatea (adică valoarea funcției  $f$ ). În această reprezentare, imaginea apare ca o suprafață discretă a cărei valoare în orice punct  $(x, y)$  este aceea a funcției  $f$  la coordonatele respective. Să presupunem acum că dorim să realizăm deschiderea funcției  $f$  printr-un element structurant de formă sferică,  $b$ , care poate fi considerat ca o “minge rostogolitoare”. Mecanismul deschiderii lui  $f$  prin  $b$  poate fi interpretat geometric ca proces de împingere a mingii de-a lungul părții inferioare a suprafeței și a rostogolirii ei

în așa fel încât întreaga față inferioară să fie traversată. Deschiderea,  $f \circ b$ , va fi prin urmare suprafața formată din punctele cele mai ridicate pe care le atinge orice parte a sferei, atunci când aceasta se deplasează pe întreaga față inferioară a lui  $f$ . Figura 4.25 ilustrează acest concept. Figura 4.25.a) prezintă o linie de rastru a unei imagini pe scară de gri ca pe o funcție continuă, pentru a simplifica prezentarea. Figura 4.25.b) prezintă mingea rostogolitoare în diverse poziții, iar figura 4.25.c) prezintă rezultatul deschiderii lui  $f$  cu  $b$  de-a lungul liniei de rastru. Toate vârfurile înguste, în raport cu diametrul mingii, au fost reduse în amplitudine și ascuțite. În aplicațiile practice, operațiunile de deschidere sunt utilizate de obicei pentru a înlătura micile (în raport cu dimensiunile elementului structurant) detalii luminoase, lăsând în același timp nemodificate nivelurile de gri globale și zonele luminoase mari. Erodarea inițială înlătură detaliile mici, făcând însă imaginea mai întunecată. Dilatarea ulterioară mărește luminozitatea imaginii, fără a reintroduce detaliile înlăturate prin erodare.

Figurile 4.25.d) și e) prezintă rezultatul închiderii lui  $f$  prin  $b$ . De data aceasta, mingea alunecă pe partea superioară a suprafeței, iar vârfurile sunt menținute la forma lor originală (atâta timp cât distanța care le separă de cei mai apropiați vecini similari este mai mare decât diametrul mingii). În aplicațiile practice, închiderea este utilizată în general pentru înlăturarea detaliilor întunecate dintr-o imagine, lăsând totodată elementele luminoase relativ nedistorsionate. Dilatarea inițială înlătură detaliile întunecate și face imaginea mai luminoasă, în timp ce erodarea ulterioară întunecă imaginea fără a reintroduce detaliile înlăturate prin dilatare.

Operațiunile de deschidere și închidere au proprietățile de *monotonie crescătoare* și *idempotență*.

Figura 4.26 prezintă rezultatul aplicării algoritmului pe care l-am implementat în cazul unei deschideri, respectiv a unei închideri pe scară de gri, pe aceeași imagine ca și în cazul dilatării și erodării. Trebuie remarcată mărimea scăzută a detaliilor luminoase, fără efect apreciabil asupra nivelurilor de gri ale zonelor întunecate, în cazul deschiderii, precum și mărimea scăzută a detaliilor mici, întunecate, cu un efect mic asupra detaliilor luminoase în cazul închiderii.

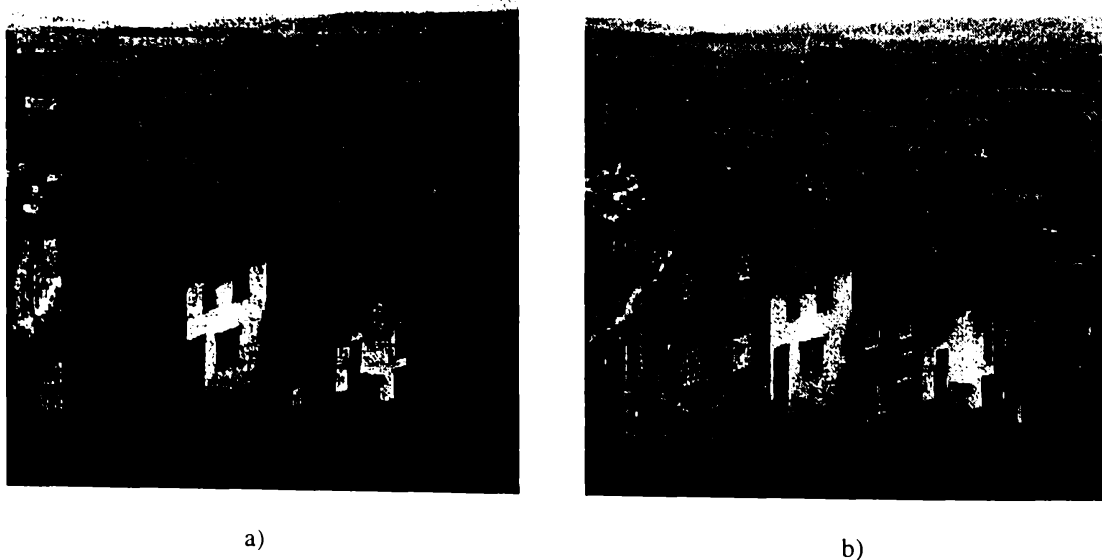


Fig. 4.26. Exemplu de implementare a deschiderii și închiderii pe scară de gri



### 4.3.3. Aplicații ale morfologiei pe scară de gri

Există numeroase aplicații ale morfologiei pe scară de gri, în special în domeniul inspecției vizuale a obiectelor. Voi prezenta în continuare câteva detalii legate de aplicațiile tipice, utilizând imagini cu rezoluția de  $512 \times 512$  pixeli.

#### 4.3.3.1. Netezirea morfologică

O posibilă modalitate de realizare a netezirii este de a efectua o deschidere morfologică pe scară de gri, urmată de o închidere. Aceasta va conduce la înlăturarea pixelilor excesiv de luminoși și excesiv de întunecați din imagine. Acești pixeli pot să apară ca rezultat al zgomotului, dar s-ar putea ca uneori ei să constituie valori reale ale imaginii. Prețul reducerii zgomotului este în general constituit de o oarecare neclaritate a imaginii.

Figura 4.27.a) prezintă o imagine afectată de zgomot Gaussian. Figura 4.27.c) prezintă rezultatul netezirii morfologice aplicate acestei imagini. La prima vedere nu este foarte clar care imagine este de preferat. Totuși, după o detecție de prag pe cele două imagini, prezentată în figurile 4.27.b) și d), se poate vedea că procesul de netezire a eliminat foarte mult din problemele de zgomot, care acum apare ca un zgomot de tip "sare și piper".

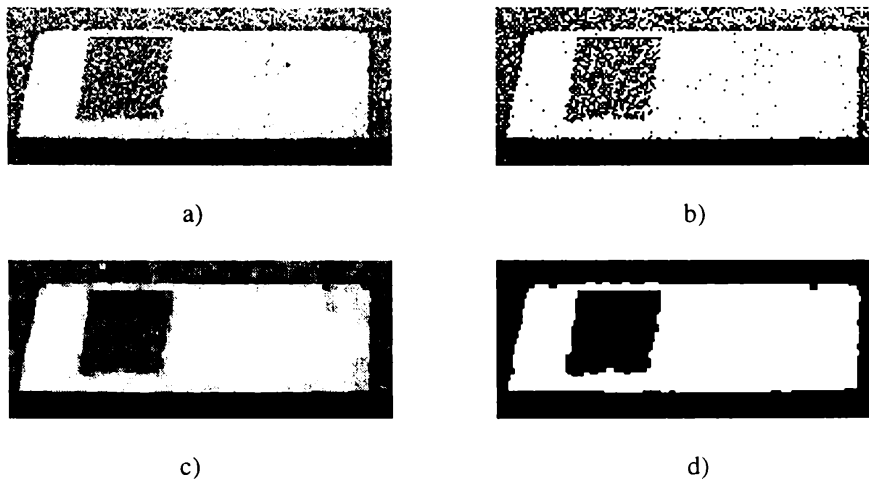


Fig. 4.27. Netezirea pe scară de gri

Elementul structurant utilizat pentru netezirea imaginii din figura 4.27.a) a fost elementul simplu (pătrat  $3 \times 3$ ), dar alegerea sa poate depinde de tipul zgomotului care trebuie curățat. O problemă comună care apare este aceea a apariției *liniilor de rastru* în imaginile obținute prin fotografierea unui ecran de televiziune sau video. Este posibilă o eliminare similară a acestui tip de zgomot structural. Elementul structurant trebuie construit în așa fel încât nivelurile sale de gri să fie diferențele dintre nivelul de gri în zona cea mai întunecată a liniei de rastru și nivelurile de gri ale fiecăruia din pixelii următori de pe aceeași coloană.

#### 4.3.3.2. Gradientul morfologic

În paragraful 4.2.1 au fost discutate metode de identificare a conturilor obiectelor bidimensionale. Ideea de bază constă în erodarea unei imagini prin utilizarea elementului structurant simplu, urmată de scăderea rezultatului din imaginea originală, ceea ce conduce la păstrarea doar a pixelilor erodați. Această procedură poate fi aplicată și în cazul imaginilor pe scară de gri. Operatorul de detecție a conturilor poate fi exprimat în același mod ca și pentru imaginile binare. Totuși, deoarece contrastul nu este la fel de mare într-o imagine pe scară de gri ca într-o imagine binară, rezultatele detecției conturilor nu sunt la fel de bune.

Se poate obține o îmbunătățire prin utilizarea *gradientului morfologic* al imaginii, notat cu  $g$  și definit prin:

$$g = (f \oplus b) - (f \ominus b). \quad (4.80)$$

În loc de a scădea imaginea erodată din cea originală, ecuația (4.80) realizează scăderea imaginii erodate dintr-o imagine dilatată. Această operație mărește contrastul și lățimea muchiilor extrase. Gradientul morfologic realizează detecția muchiilor într-un mod mai puțin dependent de direcție decât un operator gradient uzual.

Figura 4.28 prezintă rezultatele aplicării asupra aceleiași imagini atât a algoritmului de detecție a conturilor cât și a gradientului morfologic. Figura 4.28.a) prezintă imaginea originală, figura 4.28.b) prezintă rezultatul detecției conturilor pentru imagini pe scară de gri, urmat de detecția de prag. Figurile 4.28.c) și d) prezintă rezultatul aplicării gradientului morfologic înainte și după detecția de prag, respectiv.

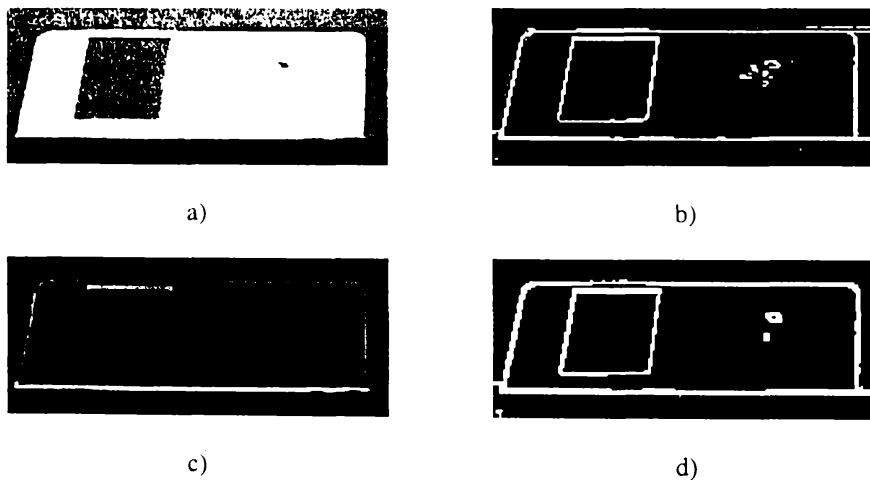


Fig. 4.28. Gradientul morfologic pe scară de gri

#### 4.3.3.3. Segmentarea texturilor

Operatorul de închidere conduce la înlăturarea detaliilor întunecate, iar operatorul de deschidere conduce la contopirea regiunilor întunecate. Aceste proprietăți sugerează o posibilă aplicație la texturi și identificarea regiunilor într-o imagine pe baza formei texturilor.

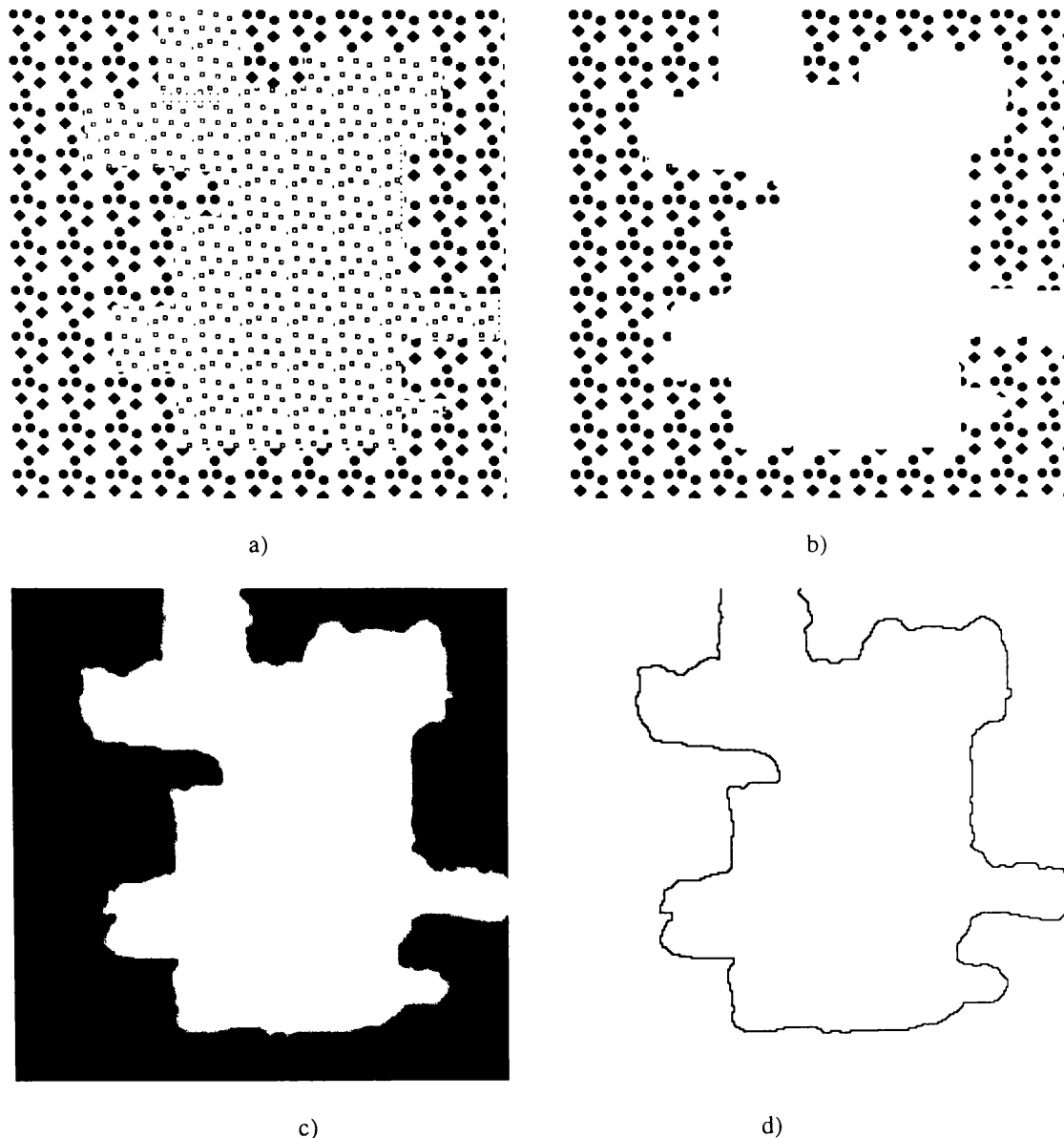
Dacă, de exemplu, o textură constă din pete mici întunecate și alta din pete întunecate mai mari, atunci obiectivul constă în găsirea hotarului dintre cele două

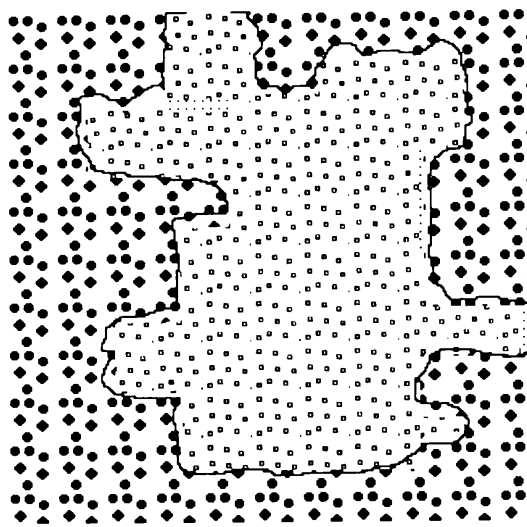
regiuni pe baza conținutului texturilor. Este foarte eficientă o abordare morfologică prin utilizarea operatorilor de închidere și deschidere.

Deoarece închiderea are tendința de a elimina detaliile întunecate dintr-o imagine, procedura de adoptat în această aplicație particulară constă în închiderea imaginii de intrare prin utilizarea succesivă a unor elemente structurante de dimensiuni mai mari. Atunci când mărimea elementului structurant corespunde cu cea a petelor mici, acestea sunt înlăturate din imagine, lăsând doar un fundal luminos în zona anterior ocupată de ele. În acest moment al procesului, în imagine mai rămân doar petele întunecate mari și fundalul luminos din zona anterior ocupată de petele mici, precum și ariile luminoase dintre petele întunecate mari.

În continuare se realizează o singură deschidere cu un element structurant mare în comparație cu ariile de separare dintre petele mari. Această operație înlătură petelele dintre petele întunecate, lăsând doar o regiune întunecată în zona petelor de dimensiuni mari. În acest moment, procedura a condus la obținerea unei regiuni luminoase în zona petelor mici și a unei regiuni întunecate în zona petelor mari. O simplă detecție de prag conduce în continuare la obținerea hotarului dintre cele două regiuni de textură diferită.

Figura 4.29 prezintă un exemplu de segmentare a texturilor.





e)

Fig. 4.29. Segmentarea texturilor

Metoda poate fi aplicată la o mare varietate de texturi, dar pot fi necesare anumite experimentări cu diferite elemente structurante pentru a obține rezultate bune.

#### 4.3.3.4. Transformarea “top-hat”

Așa numita transformare morfologică *top-hat* a unei imagini, notată cu  $h$ , este definită prin

$$h = f - (f \circ b) \quad (4.81)$$

Această transformare, care își datorează numele utilizării unei funcții cilindrice sau paralelipipedice având un vârf plat pentru elementul structurant, este utilă pentru îmbunătățirea detaliilor în prezența umbrelor. Figura 4.30 prezintă rezultatul aplicării unei transformări *top-hat* pe o imagine cu rezoluția de 256 x 256 pixeli, utilizând un element structurant paralelipedic de tip 3 x 3 pixeli.



Fig. 4.30. Transformarea morfologică top-hat

#### 4.3.3.5. Distribuția obiectelor după mărime

Utilizarea morfologiei pentru segmentarea regiunilor prin texturi sugerează o altă aplicație: clasificarea obiectelor după mărimea lor. Există un mare număr de obiecte care necesită o clasificare după dimensiuni, de la obiecte biologice sub microscop până la ouă, mere sau monede.

Vom exemplifica acest ultim caz. Monedele variază în dimensiune după valoarea lor. Figura 4.31 prezintă o imagine a unei mici colecții de monede pe un fundal întunecat. Deoarece o deschidere pe scară de gri conduce la descreșterea mărimii obiectelor, imaginea a fost deschisă cu un element structurant circular cu raza crescătoare în mod gradat. În momentul în care raza elementului structurant depășește dimensiunea unei monede, respectiva monedă este înlăturată din imagine. Monedele vor fi eliminate din imagine una câte una în ordinea crescătoare a mărimii lor.

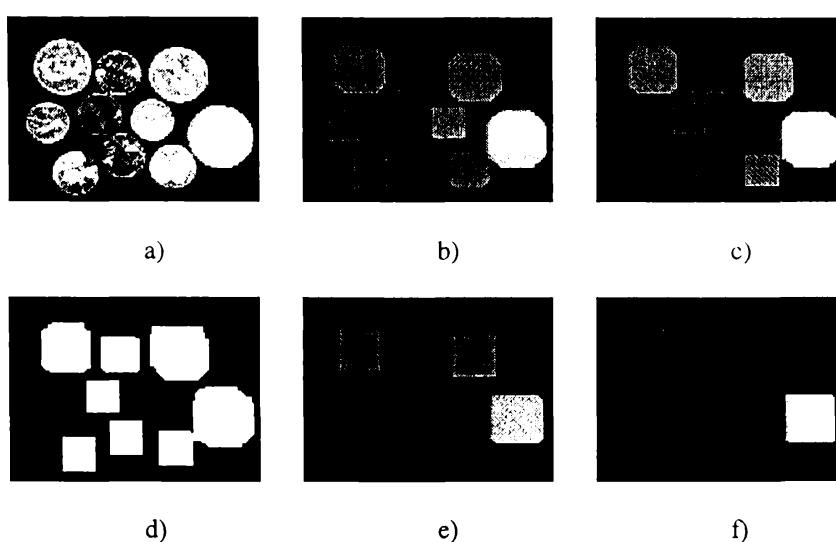


Fig. 4.31. Clasificarea monedelor după mărime

#### 4.3.3.6. Morfologie color

În contextul morfologiei matematice, culoarea poate fi utilizată în două moduri. Putem considera că existența a trei culori componente (roșu, verde, albastru) este o extensie a ideii de niveluri de gri, sau fiecare culoare poate fi considerată ca un domeniu separat care conține informații noi. În ambele cazuri, morfologia nu este aplicată în general imaginilor color, în principal datorită faptului că determinarea elementului structurant potrivit pentru realizarea unei anumite aplicații este o sarcină foarte complexă.

Voi prezenta un singur exemplu de utilizare a morfologiei pentru imagini color. Figura 4.32.a) reprezintă o imagine color a unei insecte așezate pe o frunză, în timp ce figurile 4.32.b), c), d) reprezintă componentele sale de roșu, verde și, respectiv, albastru.

Atât insecta cât și fundalul sunt esențial verzi, astfel încât localizarea automată a insectei poate fi eronată. La o examinare atentă a celor trei imagini componente color se poate observa că există ușoare diferențe în fiecare componentă: insecta pare mai luminoasă în componentele roșu și albastru, în timp ce fundalul este mai luminos în componenta verde.

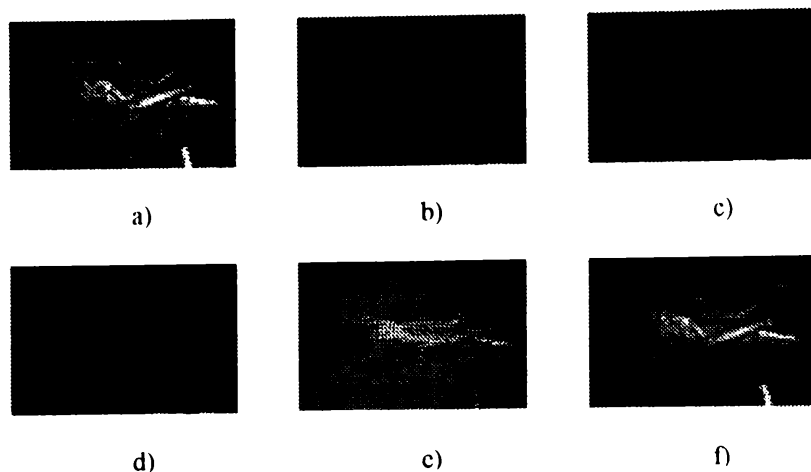


Fig. 4.32. Exemplu de morfologie color

Închiderea imaginilor componente roșu și albastru ar trebui să conducă la o strălucire mai mare a insectei, în timp ce deschiderea imaginii componente verzi ar trebui să atenueze fundalul. În ambele cazuri se utilizează un element structurant circular cu raza de 4. După realizarea închiderilor și deschiderii, cele trei imagini componente au fost recombinate pentru a forma o singură imagine color. Insecta poate fi văzută acum în contrast puternic cu fundalul întunecat. Figura 4.32.e) prezintă imaginea color rezultată, iar figura 4.32.f) prezintă imaginea originală mascată de cea rezultată prin prelucrarea morfologică, unde insecta poate fi recunoscută cu claritate (deteție de prag).

#### 4.4. Utilizarea morfologiei matematice în compresia imaginilor

Morfologia matematică a fost dezvoltată ca o alternativă geometrică pentru prelucrarea semnalelor, ceea ce a făcut-o foarte utilă de la început în aplicațiile care implicau analiza imaginilor. În prima fază nu s-a manifestat nici un fel de interes pentru cercetări în domeniul utilizării operatorilor morfologici pentru compresia imaginilor. Maragos și Schafer [135] au fost primii care au luat în discuție în 1986 posibilitatea utilizării metodelor morfologice în codarea imaginilor. Oricum, în ultimii ani a apărut un interes din ce în ce mai mare pentru utilizarea metodelor de codare bazate pe regiuni în compresia imaginilor, metode în care caracteristicile geometrice ale semnalului joacă un rol important. Metodele liniare clasice de prelucrare a semnalelor nu sunt foarte bine adaptate pentru o abordare geometrică, mai atractive fiind metode provenite din prelucrarea neliniară a semnalelor sau din domeniul vederii artificiale. Utilizarea metodelor morfologiei matematice pentru codarea imaginilor a devenit un domeniu foarte activ de cercetare [54], [162], [198], [201] și [245].

În cele ce urmează voi prezenta stadiul actual în acest domeniu prin rezumarea tehnicilor propuse, în scopul de a sublinia scopul și rezultatele cercetărilor mele.

##### 4.4.1. Descompunerea morfologică adaptivă în subbenzi

Codarea imaginilor cu un grad mare de compresie s-a dovedit un domeniu de mare interes în ultimii ani. În acest tip de codare sunt acceptate anumite distorsiuni față de imaginea originală cu scopul de a obține factori ridicați de compresie. Totuși, este

foarte important ca distorsiunile introduse să fie cât mai agreabile posibil pentru ochiul uman. Există două tipuri principale de distorsiuni în proiectarea coderelor de imagine. Primul tip este numit “efect de bloc” și apare atunci când prelucrarea se face la nivel de blocuri de imagine. Toate coderele cu transformare sunt afectate de acest tip de distorsiune. Din nefericire, ochiul uman este foarte sensibil la acest tip de distorsiune și din acest motiv codarea bloc nu este deloc potrivită pentru codarea imaginilor la rate mici de bit. Al doilea tip de distorsiune se datorează fenomenului Gibbs al filtrelor liniare și este denumit “efect de dedublare a imaginii”. Toate coderele subbandă cunoscute sunt afectate de această distorsiune. Acest efect este vizibil în jurul contururilor cu contrast mare și poate fi foarte deranjant. Deși este posibil să se reducă efectul de dedublare a imaginii printr-o proiectare adecvată a filtrelor subbandă, nu este posibil să se găsească filtre subbandă care să nu prezinte această distorsiune.

Pentru a evita aceste două tipuri de distorsiune, Murat Kunt, Olivier Egger și Wei Li au propus o tehnică de descompunere morfologică [54]. Ei propun o descompunere adaptivă, utilizând filtre liniare pentru regiunile texturate și filtre morfologice în rest. A fost de asemenea propus un algoritm pentru detecția texturilor.

Ideea cercetărilor pe care ei le-au făcut provine din descompunerea morfologică standard utilizată în analiza multirezoluție. Abordarea propusă presupune descompunerea imaginii în diferite subimagini, fiecare conținând obiecte cu mărimea specificată. Imaginile reziduale sunt obținute printr-o succesiune de operațiuni de deschidere și închidere morfologică. Reconstrucția este realizată prin adunarea tuturor subimaginilor cu diferite rezoluții. Totuși, dezavantajul principal pentru aplicațiile de codare constă în faptul că nu se realizează o subeșantionare și, de aceea, toate subimaginile au aceeași dimensiune ca și imaginea originală.

Avantajul bancului de filtre morfologice față de codarea liniară subbandă constă în lipsa oricărui efect de dedublare a imaginii. Totuși, regiunile texturate sunt slab reprezentate prin filtre morfologice în comparație cu varianta liniară. Este luată în considerație o descompunere structurată arborescent, pentru care fiecare subbandă trece-jos este descompusă în continuare, în timp ce subbenzile trece-sus nu mai sunt prelucrate în continuare. Sunt luate în considerație trei tipuri de filtre morfologice, și anume dilatarea, închiderea și filtrarea mediană. Ultima variantă este găsită ca fiind cea mai potrivită pentru această structură subbandă. Din punct de vedere vizual, imaginile reconstituite după descompunerea morfologică subbandă sunt găsite foarte agreabile pentru ochiul uman. Așa cum era de așteptat nu se manifestă nici un efect de dedublare a imaginii. Singurul dezavantaj este, așa cum s-a menționat deja, reprezentarea slabă a texturilor în imagine.

Pentru a evita această problemă a fost investigată posibilitatea utilizării filtrelor liniare pentru codarea texturilor. Pentru a beneficia de avantajele ambelor clase de filtre, cei trei autori au introdus o nouă metodă de descompunere, care alege în mod adaptiv ce clasă de filtre va fi utilizată pentru fiecare pixel de imagine. Este cunoscut că în general, la codarea subbandă, nu se poate comuta între două tipuri de filtre de sinteză deoarece condiția de reconstrucție perfectă nu va mai fi satisfăcută. Totuși, ei au demonstrat că în acest caz este posibilă comutarea între cele două filtre de sinteză deoarece filtru trece-jos de analiză este doar o transformare identică. De fapt, descompunerea subbandă liniară și morfologică realizează doar interpolarea între doi pixeli. Eroarea de interpolare este subbanda trece-sus, în timp ce imaginea subeșantionată este subbanda trece-jos. De aceea, în partea de sinteză este cunoscut ce tip de filtru a fost utilizat pentru interpolarea oricărui pixeli. Rezultă că este posibilă comutarea între cele două tipuri de filtre, atâta timp cât acest lucru este făcut în același mod atât în partea de analiză cât și în partea de

sinteză. Totuși în practică apare o problemă majoră: nu poate fi transmisă informația asupra modului în care au fost descompuși toți pixelii. De aceea, chiar dacă este utilizat același criteriu în partea de analiză ca și în partea de sinteză, filtrul nu va fi întotdeauna ales corect deoarece cuantizarea este realizată între partea de analiză și cea de sinteză. Acest lucru conduce la modificarea semnalului, iar criteriul considerat nu va da exact aceeași ieșire în partea de analiză și cea de sinteză. Kunt propune o metodă în care nu este necesară transmiterea informației despre tipul de filtrare efectuat pentru fiecare pixel.

În [54] se demonstrează că în cazul unei rate de compresie ridicate, imaginile reconstituite prin metoda propusă sunt mult mai agreabile pentru observatorul uman decât imaginile codate la aceeași rată de biți prin tehnici cunoscute, ca JPEG sau codarea liniară subbandă. Puterea tehnicii propuse este subliniată prin faptul că distorsiunea introdusă este mult mai puțin deranjantă decât efectul de bloc al codării JPEG sau efectul de dedublare a imaginii al sistemelor de codare liniară subbandă.

#### 4.4.2. Descompunerea piramidală morfologică

Structura piramidală pentru reprezentarea imaginilor furnizează o modalitate de prelucrare a imaginilor la rezoluții multiple. Reprezentarea piramidală a unei imagini constă într-o succesiune de imagini, derivate din imaginea originală, cu o rezoluție spațială descrescătoare. Dacă secvența de imagini este văzută ca o structură de date multinivel, cu imaginea originală pe nivelul de bază, structura este asemănătoare cu o piramidă. Această structură de date imagine a fost utilizată în mod frecvent pentru codarea imaginilor, pentru aplicațiile de vedere artificială și cele de transmitere progresivă a imaginilor. Algoritmii bazați pe reprezentarea piramidală sunt atractivi datorită structurii lor inerent paralele și faptului că se bazează pe principiile prelucrării multirezoluție. Algoritmii piramidali de bază implică mai multe etaje de filtrare și decimare la transmițător, urmate de mai multe etaje de interpolare la receptor. După obținerea imaginii cu rezoluția cea mai scăzută în urma pașilor de filtrare și decimare, algoritmul continuă cu interpolarea acestei imagini atât la transmițător cât și la receptor. Imaginea interpolată este scăzută din imaginea cu următoarea rezoluție de la transmițător, imaginea diferență este transmisă către receptor unde ea este adunată la imaginea interpolată la receptor. Această imagine de rezoluție mai mare este interpolată în continuare, întregul proces repetându-se de mai multe ori până ce se reconstituie complet imaginea originală.

Overturf, Comer și Delp au propus un algoritm de descompunere [162] care utilizează o structură de piramidă modificată în care cuantizorul este inclus în transmițător înainte de pasul de interpolare. Acest tip de reacție permite prelucrearea informației pierdute prin cuantizare. Un filtru morfologic de deschidere este utilizat în algoritmul de descompunere piramidală, având marele avantaj de a conserva structura geometrică și de a fi simplu de implementat.

Studiul a fost efectuat pe imagini color de 24 biți, conținând cele trei componente: roșu, verde, albastru (RGB), cu opt biți alocați pentru fiecare culoare. Înainte de descompunere, componentele RGB au fost transformate liniar în componente de luminanță ( $Y$ ) și crominanță ( $C_r$ ,  $C_b$ ). Cunoscând faptul că sistemul vizual uman este mai sensibil la detalii în componenta de luminanță decât la detalii în componentele de crominanță, cele două tipuri de componente sunt tratate diferențiat. Imaginea



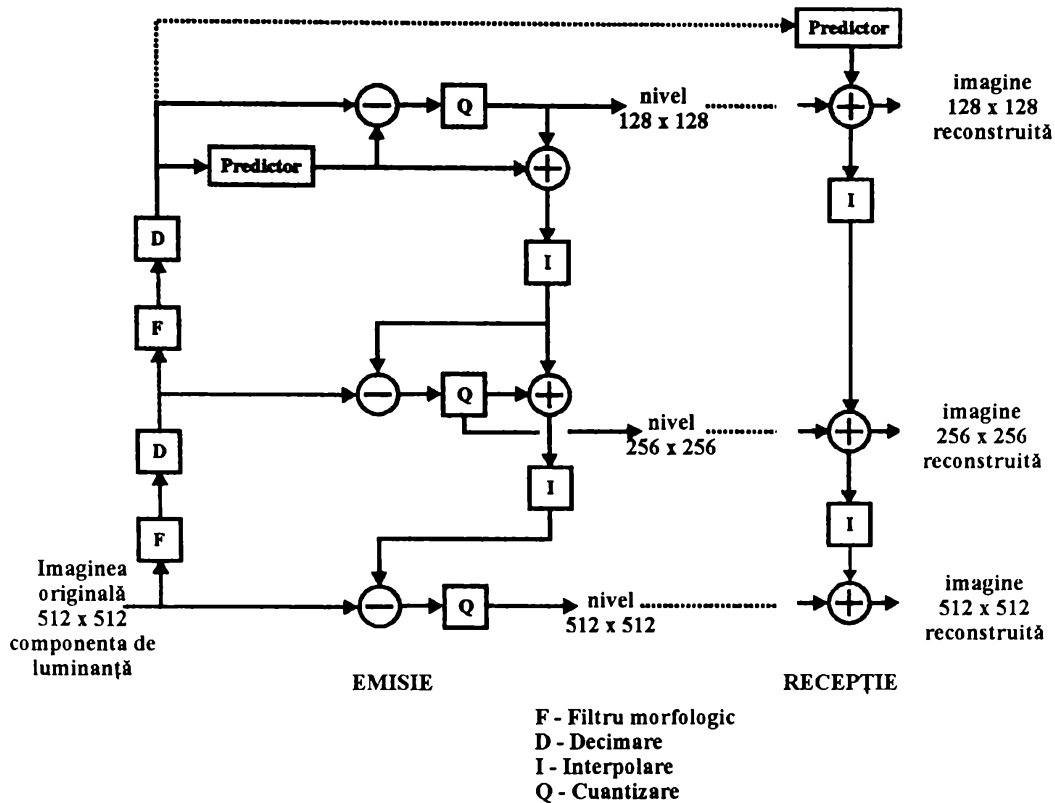


Fig. 4.33. Sistem de descompunere/transmisie piramidală pentru semnalul de luminanță

corespunzătoare luminanței a fost transmisă prin utilizarea unei descompuneri piramidale pe trei niveluri și a unui algoritm de cuantizare și codare care face apel la tehnici morfologice. Imaginile corespunzătoare crominanței au fost filtrate și decimate, apoi cuantizate, codate și transmise către receptor pentru interpolare. La receptor, componentele reconstituite sunt transformate înapoi în componente RGB. S-a observat că, pentru imaginile decodate, calitatea percepută depinde în principal de calitatea componentei de luminanță decodate. Totuși, pentru imaginile color de înaltă calitate, componentele de crominanță decimate trebuie transmise fără recuantizare. Figura 4.33 prezintă schema bloc a sistemului de descompunere piramidală propus și sistemul de codare utilizat pentru codarea semnalului de luminanță. Figura 4.34 prezintă schema bloc a sistemului de transmisie a componentelor de crominanță.

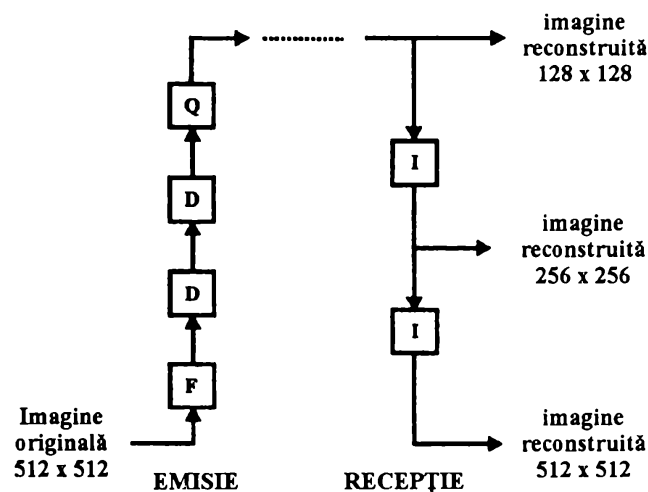


Fig. 4.34. Sistemul de codare pentru semnalele de crominanță

Una dintre componentele foarte importante pentru performanțele globale ale sistemului este filtrul de netezire utilizat înainte de decimare pentru a preveni efectul de aliere. Pentru netezire au fost utilizate filtre morfologice. Acestea implică transformarea imaginilor cu ajutorul unui element structurant. Forma și dimensiunile elementului structurant determină care dintre caracteristicile geometrice sunt conservate în imaginea filtrată. În algoritmul propus de Overturf, Comer și Delp este utilizată o deschidere morfologică pe scară de gri cu un element structurant simplu pentru filtrarea componentelor de luminanță și crominanță. Operatorul de deschidere a fost ales datorită entropiei mici care se obține cu acest filtru pentru semnalele diferențiale. Avantajele utilizării deschiderii morfologice constau în simplitate și în faptul că produce semnale diferențiale cu o gamă dinamică mai mică decât în cazul netezirii liniare. Necesitatea acestei reduceri de domeniu constă în faptul că, în urma unei deschideri morfologice, o imagine conține valori de gri strict mai mici sau egale cu valorile originale pentru fiecare pixel.

Alegerea elementului structurant utilizat pentru realizarea deschiderii este foarte importantă deoarece ea determină atât extinderea cât și “tipul” de netezire care se efectuează. Alegerea elementului structurant s-a făcut în mod empiric prin experimentarea cu mai multe forme și dimensiuni. Criteriul pentru evaluarea fiecărui element structurant a fost entropia semnalelor diferențiale rezultate și aspectul respectivei imagini netezite. Obiectivul l-a constituit obținerea unei entropii cât mai scăzute pentru a favoriza cuantizarea și codarea, cu menținerea integrității imaginii netezite. În final, a fost preferat elementul structurant de tip cruce  $3 \times 3$ , prezentat în figura 4.35. S-a constatat că acest tip de element structurant elimină imperfecțiunile de tip bloc care caracterizează elementul structurant de tip pătrat.

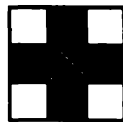


Fig. 4.35. Elementul structurant utilizat pentru deschiderea morfologică

Metodele morfologice au fost utilizate și în procesul de interpolare. Au fost experimentate atât filtre de interpolare liniară cât și filtre morfologice. Filtrele de interpolare liniară obțin valoarea pixelilor lipsă prin medierea celor ale celor patru vecini cei mai apropiați. Filtrele de interpolare morfologică obțin valoarea pixelilor lipsă printr-o operație de dilatare cu un element structurant de tip pătrat  $3 \times 3$ . Operațiunea de dilatare substituie valoarea maximă de pixel găsită în fereastră cu valoarea pixelului pe care este centrată fereastra. Prin compararea celor două metode de interpolare s-a constatat că operația de dilatare are tendința de a crea imagini interpolate mai structurate pe blocuri, în timp ce interpolarea liniară produce imagini mai netezite.

Algoritmul final de descompunere piramidală a imaginilor color utilizează deschiderea morfologică pentru netezire, codarea bloc trunchiată pentru cuantizarea semnalelor diferențiale la fiecare nivel al piramidei pentru luminanță și codarea bloc trunchiată pentru cuantizarea semnalelor de crominanță decimate. Algoritmul de compresie a fost testat pe mai multe imagini, conducând la o calitate acceptabilă a imaginilor decodate la rate de transmisie de 2,64 biți/pixel.

#### 4.4.3. Codarea morfologică cu transformare distanță

O altă metodă care poate fi utilizată pentru codarea morfologică a imaginilor este bazată pe modelarea imaginilor ca suprafețe definite prin valorile de gri ale fiecărui pixel. O suprafață poate fi modelată, de exemplu, prin funcții spline biliniare. Punctele de control corespunzătoare pot fi determinate prin utilizarea conceptului de distanță, așa cum a propus Toivanen [245].

*Transformarea distanță* (DT-distance transform) definește distanța dintre contururile imaginii și toate celelalte puncte. Distanța față de contururi poate fi calculată în două moduri diferite. În primul rând, distanța poate fi calculată prin erodări (sau dilatări) succesive printr-un element structurant. Transformarea distanță corespunzătoare a fost denumită de Toivanen *transformare distanță prin erodări succesive* SEDT (Successive Erosions Distance Transform). În al doilea caz, transformarea distanță poate fi privită ca o distanță euclidiană față de contururi. Pentru calcularea unei *transformări distanță euclidiene* EDT (Euclidean Distance Transform), trebuie păstrată poziția celui mai apropiat punct de contur.

Recent au fost definite două noi transformări distanță, aplicabile la imagini pe scară de gri, de către Preteux și Merlet: o distanță topografică și una diferențială. Distanța topografică a fost definită prin utilizarea unei funcții numite cost de conectare. Să presupunem că pentru imagine se efectuează o detecție de prag (cu o valoare de gri  $\alpha$ ) și că punctele pentru care trebuie calculată distanța sunt în aceeași conectivitate cu imaginea binară creată. Costul de conectare este definit ca infimum al acestor valori de gri. Distanța diferențială a fost definită prin utilizarea funcției cost de deviație. Această funcție controlează informația referitoare la cea mai mare pantă a suprafeței topografice.

Vepsalainen și Toivanen au definit *funcția distanță pe spații curbe* DFOCS (Distance Function On Curve Space). În termenii morfologiei matematice, ea poate fi definită ca

$$D_r = (X \ominus rB) - [(X \ominus rB) \circ drB] \quad (4.82)$$

unde  $rB$  semnifică discul deschis cu raza  $r$ , iar  $drB$  este discul închis de rază infinit mică  $dr$ . Contururile mulțimilor erodate  $(X \ominus rB)$  pot fi văzute ca și fronturi de undă care se propagă, timpul de propagare coincidând cu raza  $r$ . Scăzând din aceste valori erodate ale lui  $X$  deschiderea lor cu  $drB$ , se păstrează doar punctele unghiulare ale scheletului. Mulțimea originală  $X$  poate fi reconstituită ca reuniune, pentru toate valorile  $r > 0$ , a tuturor submulțimilor dilatate cu un disc deschis  $rB$ .

În continuare se utilizează transformarea distanță pentru a genera punctele de control pentru reprezentarea imaginii. Calitatea imaginii reconstituite depinde de algoritmul utilizat pentru interpolarea punctelor de control. Rezultatele experimentale au arătat că rapoartele de compresie obținute sunt bune, iar calitatea vizuală a imaginilor este comparativă cu cea obținută prin compresia JPEG. Rapoartele de compresie obținute sunt de ordinul 1:20 până la 1:35.

#### 4.4.4. Codarea morfologică bazată pe regiuni

Metodele de compresie bazate pe regiuni descriu imaginile prin intermediul unui set de regiuni, adică printr-o partiție, precum și prin informație corespunzătoare fiecărei regiuni care trebuie utilizate de receptor pentru reconstituirea imaginii. Salambier,

Torres, Meyer, Gu, Brigger, Casas și Pardas [198], [201] au propus un algoritm de codare care combină o analiză spațială a secvenței cu o compensare a mișcării în informația transmisă.

Într-o strategie de codare a imaginilor bazată pe regiuni, un rol important îl ocupă caracteristicile geometrice ale semnalului. De exemplu, definirea partiției implică o segmentare care ar trebui să realizeze extragerea ideală a obiectelor din imagine sau secvență. Evident că obiectele nu sunt caracterizate doar prin corelația dintre pixelii componenți, ci și printr-o serie de proprietăți geometrice. Un alt exemplu îl constituie codarea partițiilor. În acest caz informația care trebuie transmisă este pur geometrică. În sfârșit, pentru compresia foarte puternică a informației despre texturi, poate fi transmisă doar partea cea mai semnificativă a semnalului. Această parte semnificativă poate fi definită geometric, de exemplu prin valorile minime și maxime ale semnalului, prin liniile de curbura maximă sau minimă, etc. Aceasta înseamnă că sunt necesare metode geometrice pentru compresia imaginilor.

Din acest motiv, metodele morfologiei matematice sunt foarte interesante pentru tehnicile de codare. Cel puțin patru elemente morfologice au fost definite și s-a demonstrat că sunt utile pentru compresie:

- *operatori conectați*
- *algoritmi de creștere a regiunilor*
- *scheletul geodezic*
- *interpolarea morfologică*.

În cele ce urmează voi descrie pe scurt utilitatea acestor elemente în contextul compresiei imaginilor.

#### 4.4.4.1. Operatori conectați

Primul operator conectat care a fost luat în discuție în contextul codării imaginilor a fost cel de *deschidere prin reconstrucție*. Acest operator a fost denumit *deschidere* deoarece este un proces crescător, antiextensiv și idempotent, care posedă proprietățile fundamentale ale deschiderii algebrice. Operatorul este numit de asemenea *prin reconstrucție*, deoarece el implică un proces de reconstrucție a componentelor conectate care nu au fost complet înlăturate prin erodare. Se poate considera că *deschiderea binară prin reconstrucție* are proprietatea fundamentală de a simplifica semnalul, păstrând totodată informația despre contur. Componentele conectate ale imaginii binare sunt fie total eliminate (efectul de simplificare), fie perfect conservate (păstrarea conturului).

Idea originală a deschiderii prin reconstrucție constă într-o separare între un proces de analiză și unul de decizie, așa cum este ilustrat în figura 4.36. Simplificarea este un proces de decizie binară, care arată care dintre componentele conectate trebuie să fie păstrate și care trebuie înlăturate.

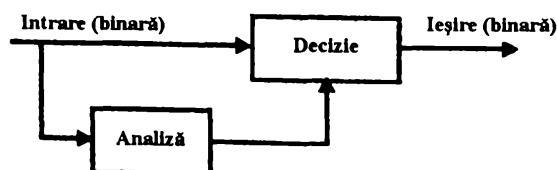


Fig. 4.36. Structura unui operator binar conectat

Conceptul de operator binar conectat este definit formal în următorul mod. În primul rând trebuie definită o conectivitate, care practic se reduce la definirea unui sistem de vecinătăți locale care să descrie conectivitățile dintre pixelii adiacenți. Variantele clasice presupun utilizarea tetra, hexa sau octo-conectivității. După selectarea tipului de conectivitate, se poate defini noțiunea de operator conectat în modul următor.

Un operator binar  $\psi$  este numit conectat atunci când pentru orice imagine binară  $X$ , diferența simetrică  $X \setminus \psi(X)$  este compusă exclusiv din componente conectate cu mulțimea  $X$  sau cu complementara sa  $X^c$ .

Extensia noțiunii de operatori binari conectați la operatori pe scară de gri conectați se bazează pe conceptul de partiție. Extensia nu se poate face direct deoarece noțiunea de conectivitate nu are un echivalent în cazul funcțiilor pe nivel de gri. O partiție a spațiului euclidian  $E$  este o mulțime de componente conectate  $\{A_i\}$ , care sunt disjuncte, iar reuniunea lor este întregul spațiu. Fiecare  $A_i$  poartă denumirea de clasă de partiție. Mai mult, se spune că o partiție  $\{A_i\}$  este mai fină decât o altă partiție  $\{B_i\}$  dacă orice pereche de puncte aparținând aceleiași clase  $A_i$  aparține în același timp și unei clase unice de partiție  $B_i$ . Considerând o imagine binară putem defini *partiția asociată* a sa ca partiția compusă din componentele conectate ale mulțimilor binare și ale complementarelor lor. Definiția operatorilor conectați poate fi exprimată și prin partiții asociate.

Un operator binar  $\psi$  este conectat dacă și numai dacă, pentru orice imagine binară  $X$  partiția asociată lui  $\psi(X)$  este mai puțin fină decât partiția asociată lui  $X$ .

Conceptul de operatori conectați pe scară de gri poate fi introdus dacă definim o partiție asociată unei funcții. În acest scop a fost propusă utilizarea așa numitelor *zone plate*. Mulțimea zonelor plate ale unei funcții pe scară de gri  $f$  este mulțimea cea mai mare a componentelor conectate din spațiul în care funcția  $f$  este constantă. Poate fi demonstrat [211] că mulțimea zonelor plate ale unei funcții constituie o partiție a spațiului. Această partiție este denumită *partiția zonelor plate* ale funcției pe scară de gri. Ea conduce la următoarea definiție.

Un operator  $\psi$  care acționează asupra imaginilor pe scară de gri este conectat dacă, pentru orice funcție  $f$ , partiția zonelor plate ale lui  $\psi(X)$  este mai puțin fină decât partiția zonelor plate ale lui  $f$ .

Cea mai simplă modalitate de a crea operatori conectați pe scară de gri este de a extinde operatorii binari conectați. După cum se arată în [78], orice operator binar poate crea un operator pe scară de gri prin *descompunere prin detecție de prag și stocare*. Această procedură este ilustrată în figura 4.37. Descompunerea prin detecție de prag generează o imagine binară  $X_\lambda$  pentru fiecare nivel de gri posibil  $\lambda$ , adică,  $2^N$  imagini binare dacă nivelurile de gri sunt cuantizate cu  $N$  biți. Fiecare imagine binară, asociată unui nivel de gri  $\lambda$ , este prelucrată printr-un operator binar conectat  $\psi$ . În final, stocarea constă în reconstituirea unei imagini pe scară de gri  $g = \psi(f)$  din mulțimea de imagini binare  $\psi(X_\lambda)$  după cum urmează:

$$g = \psi(f) = \sup_{\lambda} \left( \bigcap_{\mu < \lambda} \Psi(X_{\mu}) \right) \quad (4.83)$$

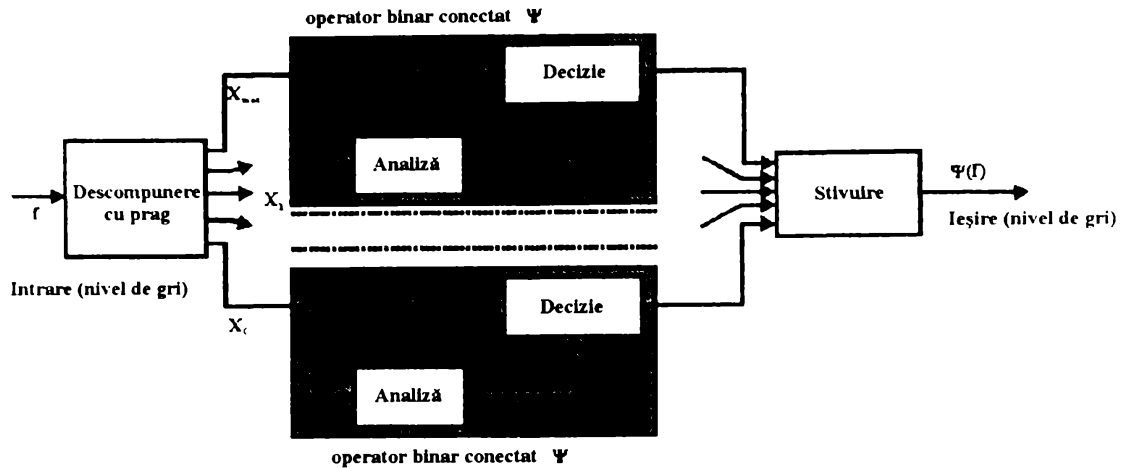


Fig. 4.37. Operator conectat pe scară de gri obținut prin descompunere cu detecție de prag

Dacă operatorul binar conectat  $\psi$  este crescător, stocarea poate fi simplificată după cum urmează:

$$g = \Psi(f) = \sup_{\lambda} (\Psi(X_{\lambda})) \quad (4.84)$$

Prin această procedură se poate demonstra [211] că operatorul pe scară de gri rezultat  $\psi$  este un operator conectat, deoarece partiția zonelor plate ale lui  $f$  este întotdeauna mai fină decât partiția zonelor plate ale lui  $\psi(f)$ . Structura de prelucrare ilustrată în figura 4.36 explică de ce operatorii conectați pe scară de gri realizează o simplificare, păstrând în același timp informația despre contur. Într-adevăr, ca și în cazul operatorilor binari conectați, un proces de decizie binară arată dacă o zonă plată trebuie păstrată sau nu. În plus, procesul de decizie este separat de procesul de reconstrucție.

Putem menționa ca exemple de operatori pe scară de gri: *deschiderea prin reconstrucție pe scară de gri*, *deschiderea zonală* și operatorul *h-max*.

Operatorii conectați sunt foarte utili pentru segmentare și, în particular, în domeniul codării. Într-adevăr, pentru aplicațiile de codare, segmentarea trebuie să fie condiționată și trebuie să extragă doar regiunile cele mai importante din imagine. *Deschiderea (închiderea) prin reconstrucție* sau operatorii *h-max (h-min)*, respectiv, elimină regiunile de întindere mică sau de contrast scăzut. Ei permit reprezentarea imaginii prin regiuni de întindere mare sau de contrast ridicat, care în general corespund componentelor de imagine cu importanță vizuală mare. În plus, ei sunt foarte potriviți pentru simplificare deoarece înlătură o parte din informație, conservând totodată informația de contur pentru componentele de imagine rămase.

#### 4.4.4.2. Algoritmi de creștere a regiunilor

Metoda clasică de segmentare constă în realizarea extragerii unor caracteristici și în utilizarea așa numitului *algorithm watershed* asupra gradientului imaginii care trebuie segmentată. Această modalitate nu este adecvată pentru aplicații de codare, deoarece utilizarea gradientului conduce la o pierdere de informație. În cazul imaginilor statice acest fenomen nu este deosebit de supărător. Totuși, în cazul imaginilor în mișcare, pierderea de informație este mai importantă, de aceea s-a demonstrat [198] că utilizarea

gradientului trebuie evitată, iar algoritmul watershed trebuie modificat pentru a acționa direct asupra semnalului și nu asupra gradientului său.

Algoritmul rezultat este o variantă de creștere a regiunilor, în sensul că pornește de la punctul de identificare a interiorului regiunilor și apoi le extinde până ce ocupă tot spațiul disponibil.

O implementare eficientă a algoritmului watershed modificat necesită o scanare inteligentă a imaginii, definită prin *cozi ierarhice*. O coadă ierarhică este formată dintr-un set de șiruri FIFO cu priorități diferite. Elementele prelucrate de aceste cozi sunt pozițiile pixelilor, cozile fiind utilizate pentru definirea scanării. Această structură permite reprezentarea unei duble ordonări: pixelii sunt dispuși într-una din cozi în funcție de o prioritate dată. Primul pixel extras din coadă este primul pixel intrat în coada cu cea mai mare prioritate. În continuare sunt extrași succesiv toți pixelii din coada de prioritate maximă. În final, dacă coada de cea mai mare prioritate este goală, următorul pixel extras este primul pixel al următoarei cozi nenule. Varianta de creștere a regiunilor pentru algoritmul watershed poate fi implementată simplu cu aceste cozi. Algoritmul lucrează în doi pași distincți: *inițializarea cozii și creșterea regiunilor*.

Pasul de inițializare constă în așezarea locațiilor tuturor pixelilor care corespund interiorului regiunilor în coada de prioritate maximă. Este utilizată coada de prioritate maximă deoarece prioritatea corespunde probabilității cu care un pixel aparține unei regiuni date.

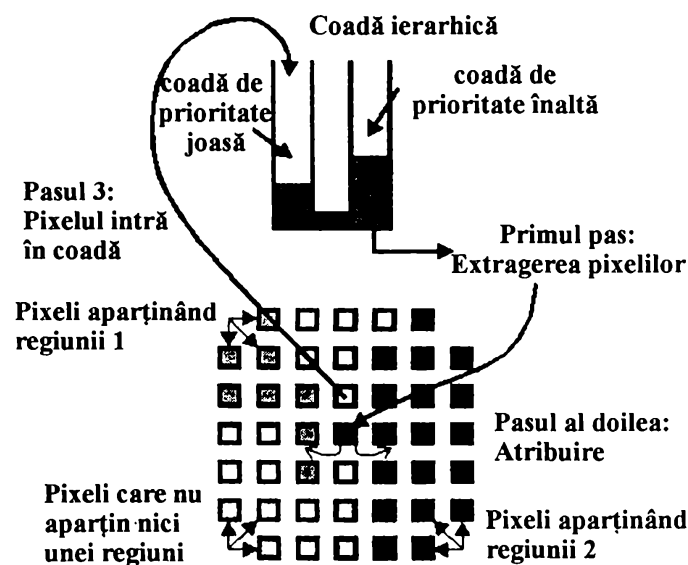


Fig. 4.38. Implementarea algoritmului de creștere a regiunilor

Pasul de creștere a regiunilor constă în extragerea unui pixel din coadă. Dacă pixelul respectiv nu aparține încă unei regiuni, acest lucru este cunoscut datorită procedurii de umplere care indică că el are cel puțin o regiune vecină. De aceea este calculată *distanța* dintre fiecare pixel și fiecare regiune vecină. Pixelul va fi asignat regiunii care corespunde distanței minime. În continuare, dacă pixelul curent are vecini care nu aparțin nici unei regiuni, aceștia sunt introduși într-o coadă cu prioritatea definită prin distanța la regiunea de care aparține pixelul curent. După cum se vede, fiecare pixel introdus într-o coadă, are cel puțin o regiune vecină. De aceea este posibil să se ia o decizie asupra pixelului respectiv în momentul în care el este extras din coadă. Procedura de creștere a regiunilor este ilustrată în figura 4.38 pentru cazul imaginilor

bidimensionale. Același algoritm poate fi utilizat și pentru secvențe de imagini, care vor fi interpretate ca semnale tridimensionale. Singura diferență constă în modul de definire a sistemului de vecinătate.

Un parametru important al algoritmului este funcția distanță. În mod ideal, această distanță estimează probabilitatea cu care un anumit pixel aparține unei regiuni. Totuși, pentru aplicații de codare, este util să se introducă în distanță un criteriu de estimare a complexității conturului, pentru a limita costurile de codare a conturului. Distanța poate fi definită, de exemplu, ca sumă ponderată a diferenței de nivel de gri între valoarea  $x$  a nivelului de gri al pixelului curent și valoarea medie a regiunii  $R$  și un termen proporțional cu lungimea incrementului conturului regiunii  $\Delta\partial R$ . Incrementul conturului regiunii poate fi calculat local prin considerarea numărului punctelor de contur care sunt adăugate/înlăturate de fiecare dată când un pixel este asignat unei anumite regiuni. De aceea funcția distanță este definită prin

$$d(x,R) = \alpha \left( x - \left( \sum_{y \in R} y \right) / \left( \sum_{y \in R} 1 \right) \right) + (1 - \alpha) \Delta\partial R \quad (4.85)$$

Dacă distanța este definită prin nivelul de gri  $x$  (adică dacă  $d(x, R)=x$ ), algoritmul devine clasicul "watershed".

#### 4.4.4.3. Scheletul geodezic

După cum s-a arătat deja, tehnicile de codare a partițiilor pot fi clasificate în metode bazate pe contur sau forme. Primul set de metode utilizează reprezentarea partițiilor prin descrierea conturilor, în timp ce al doilea set utilizează forma regiunilor.

Cele mai populare tehnici de codare bazate pe contur sunt reprezentate de codurile lanț derivative, aproximările poligonale și aproximarea geometrică a curbilor. Avantajul algoritmului bazat pe coduri lanț derivative constă în faptul că el permite codarea eficientă și fără pierderi a arcelor de contur conectate. Deficiența principală constă în codarea punctului de început al arcului de contur. Celelalte două tehnici sunt cu pierderi, performanțele lor depinzând mult de aplicație. Ele sunt dificil de aplicat în special în cazurile în care pierderea acceptabilă în poziționarea conturului este mică.

Poate fi introdusă o metodă morfologică pentru codarea orientată pe formă a conturului. Această metodă permite o reprezentare flexibilă a conturului, ceea ce îi conferă o mare eficiență în cazul mai multor contururi separate în formă de arc sau a conturilor cu puține detalii.

Scheletul euclidian a fost deja introdus în paragraful 3.2.6, iar cel morfologic în paragraful 4.2.6. În aplicațiile care implică studiul topologic al obiectelor (câte obiecte, câte goluri în interiorul obiectelor, câte ramuri) este foarte importantă conservarea conectivității. Pentru aplicațiile de codarea imaginilor nu este obligatorie conservarea conectivității. În aceste cazuri este necesară găsirea unei descompuneri în schelete care să îndeplinească următoarele trei condiții:

- existența și unicitatea scheletului pentru o mulțime  $X$
- echivalența între mulțimea  $X$  și mulțimea  $X'$  reconstituită din scheletul său
- o formulă explicită de calcul a scheletului.

În plus, este important ca formula explicită să conducă la obținerea unui schelet cu cât mai puține puncte posibil, pentru a obține o compresie importantă a informației despre contur. S-a demonstrat [130] că descompunerea în schelete conține puncte



redundante, care nu sunt necesare pentru o reconstrucție perfectă. Totuși scheletul nu a fost utilizat niciodată pentru codarea imaginilor segmentate care conțin mai multe regiuni. Motivul este probabil constituit de faptul că o aplicare directă conduce la o codare redundantă a conturilor. Într-adevăr, fiecare contur aparține la două regiuni vecine și va fi prin urmare reprezentat de scheletul fiecărei regiuni.

Punctele scheletului pot fi caracterizate ca o mulțime particulară rezultată din funcția distanță, ceea ce conduce la un algoritm de determinare extrem de rapid. Fiind dată o mulțime  $X$ , funcția distanță  $\rho$  într-un punct  $x$  al mulțimii  $X$  poate fi definită ca

$$\rho_x(x) = d(x, X^c) = \inf_{y \in X} d(x, y) \tag{4.86}$$

Se observă ușor că întotdeauna există cel puțin un punct  $y_0$  pe conturul mulțimii  $X$  astfel încât  $d(x, y_0) = \rho_x(x)$ . Începând cu un punct  $x$ , vom numi superiorul lui  $x$ , mulțimea de puncte  $y$  care satisfac relația

$$\rho(y) = \rho(x) + d(x, y) \tag{4.87}$$

Dacă superiorul lui  $x$  este format doar din elementul  $x$  însuși, atunci punctul  $x$  aparține scheletului. Se ajunge astfel la următoarea definiție. *Scheletul este mulțimea punctelor fără nici un alt superior decât punctul însuși.*

Conform acestei definiții, pentru fiecare punct  $x$ , toate punctele  $y \in X$  trebuie verificate pentru a ști dacă există un superior al lui  $x$ , caz în care  $x$  nu este un punct al scheletului. În cazul numeric este suficient să fie considerați toți vecinii lui  $x$ , care sunt definiți prin graful utilizat pentru calcularea distanței. Frecvent este utilizată masca de tip tablă de șah (pătrat  $3 \times 3$ ). Din această definiție a scheletului, poate fi dedus scheletul geodezic. Ideea constă în excluderea conturilor care sunt deja cunoscute. Aceasta înseamnă că punctul  $y_0$  este limitat la un număr specific de puncte ale lui  $X$ . Dacă notăm cu  $Z$  aria deja codată, putem defini o nouă referință prin

$$K = Z \cup X \tag{4.88}$$

care va fi utilizată în calculul distanței și conduce la

$$\rho_K(x) = d(x, K^c) = \inf_{y \in K} d(x, y), \quad \forall y \in E, x \in X \tag{4.89}$$

Scheletul este prin urmare format din punctele fără alt superior decât ele însele. În figura 4.39 este prezentat un exemplu, în care sunt prezentate scheletul normal și scheletul geodezic pentru o regiune la care regiunea vecină a fost deja codată.

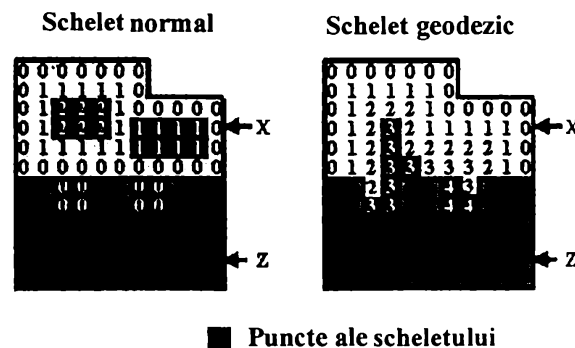


Fig. 4.39. Comparație între scheletul clasic și cel geodezic

După cum se arată în [130], scheletul conține puncte redundante. La fel se întâmplă și în cazul scheletului geodezic. Aceste puncte pot fi înlăturate, păstrându-se în

aceiași timp posibilitatea reconstrucției fără pierderi. Pentru detecția punctelor redundante poate fi utilizat următorul algoritm:

- Se calculează o matrice de redundanță. Pentru fiecare pixel se determină numărul de elemente structurante care îl acoperă.
- Se înlătură toate punctele scheletului care acoperă o arie care este deja complet acoperită de alte elemente structurante.

Această procedură implică dilatarea fiecărui punct al scheletului și scanarea fiecărei regiuni dilatate în vederea depistării prezenței pixelilor care provin de la alte puncte ale scheletului, ceea ce conduce la un timp de calcul extrem de mare. De aceea a fost propusă [198], [201] o procedură de dilatare pentru punctele scheletului, bazată pe cozi ierarhice, care permite o implementare mai rapidă. S-a arătat totodată că scheletul geodezic conține întotdeauna un număr de puncte mai mic sau cel mult egal cu cel al scheletului clasic. Rezultatele experimentale confirmă că se poate obține o reducere în medie cu 50% a numărului de puncte din scheletul unei imagini segmentate.

În aproape toate aplicațiile de codarea imaginilor, decodorul trebuie să funcționeze în timp real. Această cerință implică necesitatea unei structuri simple și eficiente. Scheletul geodezic permite o reconstrucție exactă a regiunilor descompuse prin următoarea relație:

$$X = \bigcup \{x \in r_k(X) : \delta_Z^{\rho_k(x)}(x)\} \quad (4.90)$$

unde  $r_k(X)$  semnifică scheletul geodezic, iar  $\delta_Z^{\rho_k(x)}$  semnifică dilatarea geodezică în raport cu  $Z$ , de dimensiune  $\rho_k(x)$ . În acest fel este garantată o reconstrucție simplă, bazată pe reuniunea dilatărilor.

#### 4.4.4.4. Interpolarea morfologică

Tehnicile de codare cu interpolare se bazează pe codarea și transmiterea unui subset de pixeli din imaginea originală, astfel încât la recepție este necesară interpolarea pixelilor rămași pornind doar de la informația transmisă. Imaginea reconstruită este aproximată prin funcții continue, cu o anumită eroare permisă la nivelul pixelilor interpolați. Utilizarea tehnicilor de interpolare se bazează pe proprietățile sistemului vizual uman, imaginea fiind percepută ca o alăturare de arii relativ netede, separate prin discontinuități sau muchii. Ținând cont de această proprietate este posibilă obținerea unei reconstrucții a imaginii originale pornind doar de la informația referitoare la structura geometrică a discontinuităților și la amplitudinea pixelilor de pe muchii.

În [198] este prezentată o metodă morfologică pentru rezolvarea problemei de interpolare bidimensională. Metoda se bazează pe dilatarea geodezică și se caracterizează printr-o mare eficiență față de metodele care realizează filtrarea liniară a imaginii inițiale. Algoritmul propus are un grad mare de generalitate, el putând realiza interpolarea din orice set de pixeli dați.

Scopul algoritmului de interpolare morfologică este de a aproxima amplitudinea pixelilor necunoscuți din imagine pornind de la un subset de pixeli cu valoarea cunoscută, care vor fi numiți în continuare *pixeli inițiali*. Pixelii obținuți prin interpolare formează prin valorile lor o suprafață, asupra căreia se pune condiția de a fi cât mai netedă față de valorile pixelilor cunoscuți. Această condiție este definită în sensul ca variațiile de la pixel la pixel să fie minimizate.

O strategie potrivită pentru interpolarea bidimensională este de a calcula în fiecare punct valoarea medie a amplitudinilor pixelilor inițiali ponderate prin inversul

distanței la fiecare pixel. Pixelii mai apropiați au o influență mai mare decât cei mai depărtați, iar amplitudinile interpolate se schimbă puțin în ariile cuprinse între pixelii inițiali. Măsura distanței este considerată ca *distanță geodezică* în cadrul setului de pixeli necunoscuți. Figura 4.40 ilustrează avantajul utilizării distanței geodezice.

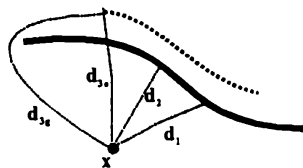


Fig. 4.40. Măsura distanței geodezice pentru interpolarea pixelului  $x$

Mulțimea pixelilor inițiali utilizați pentru interpolare este indicată prin linii continue groase și prin linii întrerupte. Să presupunem că linia întreruptă reprezintă limita superioară a unei tranziții spațiale, iar linia continuă reprezintă limita inferioară a tranziției. Influența valorilor amplitudinilor limitei superioare (linia întreruptă) asupra pixelului  $x$  este dată de inversul distanței geodezice  $d_{3g}$ , care este mai mare decât distanța euclidiană  $d_{3e}$ . De aceea, valoarea interpolată a pixelului  $x$  va fi influențată în principal de către pixelii inițiali de pe linia continuă. Ca rezultat, utilizarea distanței geodezice permite conservarea tranzițiilor.

Strategia de interpolare descrisă poate fi implementată eficient printr-un algoritm iterativ bazat pe operatori morfologici. Fiecare iterație a algoritmului realizează o *propagare geodezică* urmată de un *pas de netezire*. În loc de a se calcula distanțe geodezice de la fiecare pixel necunoscut la fiecare pixel inițial, valorile amplitudinilor pixelilor inițiali sunt propagate printr-o dilatare geodezică pentru a umple ariile goale ale imaginii, apoi sunt netezite tranzițiile care apar în afara mulțimii pixelilor inițiali. Atunci când se întâlnesc două sau mai multe fronturi de propagare cu originea în același pixel inițial, dar de amplitudini diferite, procesul este oprit și se crează o tranziție falsă. Pixelii de pe ambele fețe ale tranziției false compun mulțimea pixelilor *secundari*. Fiecărui pixel secundar  $i$  se atribuie o valoare de nivel de gri egală cu valoarea medie a nivelurilor de gri de pe ambele fețe ale tranziției. Acesta este pasul de netezire.

În continuare este realizată cea de-a doua iterație: pasul de propagare realizează propagarea nivelurilor de gri atât pentru pixelii inițiali cât și pentru cei secundari. Această propagare crează noi tranziții false, care crează un nou set de pixeli secundari cu niveluri de gri netezite. Acest nou set de pixeli secundari nu include în general primul set de pixeli secundari. Acest proces de propagare a valorilor pixelilor inițiali și secundari, urmată de netezirea nivelurilor de gri ale noilor pixeli secundari este iterat până la idempotență.

Aplicarea interpolării morfologice la codarea texturilor se bazează pe selectarea unui set de pixeli inițiali. Acest set de pixeli inițiali trebuie să permită simultan o bună restaurare a imaginii prin interpolare și să conducă la o reprezentare compactă. Pentru codarea texturilor pot fi utilizate două metode: în prima metodă, setul de pixeli inițiali este compus din linii de curbă maximă și minimă; a doua metodă utilizează punctele de maxim și minim local.

Interpolarea morfologică se dovedește a fi foarte eficientă pentru interpolarea rețelelor de curbe neregulate, deoarece permite restaurarea imaginii dintr-un număr redus de puncte. Metoda poate fi utilizată și în cadrul unor metode bazate pe regiuni sau pe blocuri.

## 4.5. Descompunerea morfologică a elementului structurant

### 4.5.1. Necesitatea descompunerii

Conceptele morfologiei matematice au fost utilizate pentru descompunerea formelor. Formele sau obiectele pot fi descrise prin intermediul unor părți mai simple și mai bine caracterizate. O descriere morfologică a formei exprimă de obicei acea formă prin descompunerea sa într-o serie echivalentă de dilatări a unor componente mai simple. Aceste componente mai simple pot fi, în cazul formelor binare, discuri, linii, dreptunghiuri, etc, de diferite dimensiuni. O formă este exprimabilă ca dilatare a două forme mai simple dacă forma originală poate fi descrisă ca o suprafață marcată în exterior atunci când o formă este păstrată fixă, iar cealaltă este trasată deasupra ei.

Formele binare sunt exprimate de obicei ca mulțime a tuturor punctelor care le constituie. Aceste forme sunt complet caracterizate prin conturul lor, multe scheme eficiente de reprezentare a informației despre contur fiind deja prezentate anterior în capitolul 3. Reprezentările conturilor fac explicite o serie de caracteristici importante, ca de exemplu vârfurile și lungimile muchiilor. Dacă aceste caracteristici sunt utilizate de algoritmi de descriere a formelor, utilizarea reprezentării conturilor va face mult mai simplă extragerea descriptorilor din reprezentare. Au fost propuși mai mulți algoritmi pentru realizarea operațiunilor morfologice prin utilizarea caracteristicilor obiectelor în domeniul bidimensional continuu.

Operațiunile morfologice sunt executate pe mașini specializate pentru realizarea acestora, dar dimensiunile maxime ale elementelor structurante care pot fi utilizate sunt limitate de structura hardware a acestor mașini. Dacă trebuie executată o operațiune morfologică cu un element structurant având dimensiuni mai mari decât permite hard-ul, atunci elementul structurant trebuie descompus în elemente cu dimensiuni mai mici. Condițiile pe care trebuie să le îndeplinească noile elemente structurante sunt următoarele:

- fiecare element trebuie să poată fi acceptat de structura hardware;
- dilatarea tuturor trebuie să conducă la elementul structurant original.

Mai multe calculatoare specializate în prelucrarea imaginilor pot realiza operațiunile morfologice de bază foarte rapid într-o configurație de tip paralel și/sau pipeline [87], [93], [135], [149]. Toate impun însă anumite restricții asupra dimensiunilor și formei elementelor structurante care pot fi utilizate într-o fază de execuție. În cazul citocalculatoarelor (CC-cyocomputers), care sunt mașini de tip pipeline, fiecare etaj poate realiza o operație morfologică de dimensiuni  $3 \times 3$ . Procesorul pe arii binare (BAP-Binary Array Processor) este format din celule octo-conectate. Pe acest tip de mașină pot fi implementate operațiuni morfologice  $3 \times 3$  prin utilizarea unor operațiuni simple. Procesorul masiv paralel (MPP-Massively Parallel Processor) este realizat pe baza celulelor tetra-conectate, care implementează operațiunile morfologice simple printr-un număr de translații și operațiuni logice. Calculatorul cu flux de imagini (IFC-Image Flow Computer) este o mașină de tip pipeline, care poate realiza la fiecare fază de lucru o operațiune morfologică pe orice pereche de pixeli.

Dacă elementul structurant utilizat într-o operație morfologică are un domeniu mai larg decât cel pe care îl acceptă structura hardware într-un pas de prelucrare, atunci respectivul element trebuie descompus în mai multe elemente structurante simple, așa cum s-a menționat deja. Problema descompunerii constă în a determina cea mai scurtă

succesiune de elemente structurante mici a căror compunere morfologică să fie elementul structurant dorit.

Aceasta înseamnă că un element structurant de dimensiuni mari trebuie descompus într-o dilatare de elemente structurante mai mici, adică

$$B = B_1 \oplus B_2 \oplus \dots \oplus B_n \quad (4.91)$$

unde fiecare element  $B_i$  poate fi acceptat de structura hardware dată.

Dilatarea unei imagini  $A$  cu  $B$  poate fi, prin urmare, implementată ca

$$A \oplus B = A \oplus (B_1 \oplus B_2 \oplus \dots \oplus B_n) = (((A \oplus B_1) \oplus B_2) \dots) \oplus B_n \quad (4.92)$$

iar erodarea lui  $A$  cu  $B$  va fi

$$A \ominus B = A \ominus (B_1 \oplus B_2 \oplus \dots \oplus B_n) = (((A \ominus B_1) \ominus B_2) \dots) \ominus B_n \quad (4.93)$$

Această descompunere furnizează o posibilă implementare în cazul elementelor structurante care nu pot fi utilizate direct pe hard-ul existent, iar pentru cele care nu pot fi utilizate eficient ea furnizează o soluție pentru o implementare mult mai eficientă.

#### 4.5.2. Rezultate în descompunerea morfologică a elementului structurant

Problema descompunerii elementului structurant a fost abordată în multe articole din revistele de specialitate, din mai multe puncte de vedere și pentru diverse arhitecturi de calcul. În conformitate cu evoluția hardware, mărimea elementelor structurante la care se face apel variază de la 2 pixeli până la arii de  $3 \times 3$  pixeli.

Problema a fost abordată pentru prima dată de către Serra [211], Giardina și Dougherty [63] dintr-un punct de vedere strict teoretic. Mai târziu, Zhuang și Haralick [291] au abordat problema descompunerii pentru structurile IFC. În lucrarea lor a fost dezvoltat un algoritm de căutare arborescentă care poate determina descompunerea optimă a unui element structurant într-o secvență de elemente de 2 pixeli, fiecare cu o mărime arbitrară a regiunii de suport.

Xu [284] a dezvoltat un algoritm optimal pentru descompunerea elementelor structurante cu *formă poligonală convexă*, pentru sisteme de calcul cu regiunea de suport de  $3 \times 3$  pixeli. Pentru descrierea formei unui element structurant dat sunt utilizate codurile lanț Freeman. El definește o imagine ca fiind simplă dacă este finită, conectată și fără goluri. Codul lanț pentru o astfel de imagine este determinat prin specificarea unui punct de start și prin definirea unei secvențe de mișcări în jurul conturului imaginii.

Pentru o imagine simplă (în spațiul  $Z \times Z$ ), dacă forma sa determinată prin linii de contur codate cu coduri lanț este un poligon convex (în spațiul  $R \times R$ ), este demonstrat că în acest caz imaginea poate fi descompusă. Demonstrația este bazată pe inducție.

Este de asemenea demonstrat că două imagini au aceeași formă dacă și numai dacă ele au același cod lanț, adică dacă și numai dacă o imagine poate fi obținută din cealaltă prin translație. De aceea, un cod lanț pentru un poligon convex reprezintă de fapt o întreagă clasă de imagini cu forma identică. Este important de observat că o translație poate fi convertită într-o succesiune de operații de vecinătate. De exemplu

$$B_{(2,1)} = (B_{(2,0)})_{0,1} = (B \oplus \{(1,0)\} \oplus \{(1,0)\}) \oplus \{(0,1)\} \quad (4.94)$$

De aceea, dacă una dintre imagini are o descompunere, atunci toate imaginile din clasa respectivă au descompunere. Se poate spune că respectiva clasă are descompunere.

Descompunerea lui Xu este efectuată prin utilizarea unei clase de 13 submulțimi, prezentate în figura 4.41. Submulțimile acestei clase au proprietatea că toate sunt mulțimi netriviiale (neunitare), nedecompozabile și de formă poligonală convexă. Orice

altă submulțime de elemente convexe poligonale netriviiale și nedecompozabile poate fi obținută prin translație din submulțimile clasei respective. Există o clasă minimală cu această proprietate, numită *clasă primitivă*. Fiecare submulțime dintr-o clasă primitivă poate fi considerată ca un element de bază pentru descompunere. Desigur, clasa primitivă nu este unică.

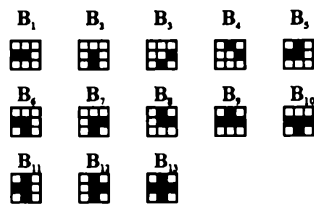


Fig. 4.41. Clasa primitivă pentru descompunere

În general, descompunerea unui poligon convex într-o submulțime de elemente de bază nu este unică. Descompuneri diferite pot conduce la implementări diferite și prin urmare la viteze diferite pe structuri de calcul diferite. În figura 4.42 este prezentat un exemplu, care arată două modalități de descompunere a unui poligon convex.

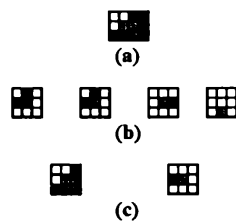


Fig. 4.42. Exemple de descompunere

Problema descompunerii pe calculatoare de tip MPP va fi prezentată cu mai multe detalii. Pentru calculatoarele MPP, cele mai eficiente de utilizat elemente structurante sunt cele care provin din elementul de tip romb (originea și cei patru vecini într-o tetra-conectivitate). Pentru a implementa cu eficiență operațiuni morfologice pe acest tip de mașină, va trebui realizată o descompunere a elementului structurant în submulțimi ale elementului romb. O astfel de descompunere este numită de multe ori *descompunere pe vecinătăți de tip romb*, iar o submulțime dintr-o astfel de descompunere este numită *submulțime de vecinătăți romb*. Descompunerea din figura 4.42.b) este o astfel de descompunere. Submulțimile din clasa primitivă prezentată în figura 4.41 sunt toate descompuneri pe vecinătăți de tip romb. Aceasta înseamnă că s-a demonstrat de fapt că toate poligoanele convexe au descompuneri pe vecinătăți de tip romb.

Park și Chin [168] au propus un algoritm optimal pentru descompunerea elementelor structurante convexe pe arii de prelucrare tetra-conectate, utilizând drept criteriu de optimizare numărul de translații efectuate. În același timp este prezentată o demonstrație algebrică pentru existența descompunerii. Pentru descompunere este luat în considerație un set similar cu cel din figura 4.41, format tot din 13 elemente structurante. Este prezentată de asemenea o comparație cu metoda de descompunere a lui Xu. Rezultatele obținute sunt considerate mai bune datorită utilizării criteriului de optimizare. Oricum elementele structurante de bază utilizate sunt obținute prin simple operații de translație și rotație, din cele folosite de Xu. Totuși, nu este demonstrată existența unui set optim de *elemente structurante de bază* care să furnizeze o

descompunere optimă independent de forma și dimensiunile elementului structurant care urmează să fie descompus.

Mai târziu, aceiași autori, Park și Chin [169], au prezentat rezultate referitoare la descompunerea formelor arbitrare (convexe sau concave) de elemente structurante în elemente cu dimensiunile maxime de  $3 \times 3$  pixeli. Descompunerea se bazează pe conceptul de factorizare a unui element structurant în factorii săi primi.

Pentru specificarea și rezolvarea problemei sunt introduse câteva definiții și notații.

Un *colț* al unei imagini binare conectate este un pixel al conturului care este 8-conectat cu cei doi pixeli de contur adiacenți pe direcțiile  $i$  și  $j$  din codul lanț Freeman, unde  $i \neq j$ . Un colț este definit de trei pixeli ai conturului care au două direcții distincte în codul lanț.

*Unghiul* unui colț este diferența dintre cele două direcții ale colțului format de trei pixeli, măsurat dinspre interiorul imaginii. Un *colț concav* este un colț al cărui unghi este mai mare decât  $180^\circ$ . Un *colț convex* este un colț al cărui unghi este mai mic decât  $180^\circ$ .

Un *segment de contur* este format dintr-o mulțime de pixeli conectați ai conturului, de la un colț convex la alt colț convex, inclusiv cele două colțuri convexe. Un *contur concav* este un segment de contur care conține unul sau mai multe colțuri concave. Un *contur convex* este un segment de contur care nu conține colțuri concave.

Orice contur concav sau convex are un pixel comun cu un alt contur concav sau convex învecinat. De aceea, codul lanț al unui contur dintr-o imagine binară este divizat în contururi concave sau convexe care nu se suprapun.

Există un număr infinit de contururi concave diferite, dar dacă imaginile sunt restricționate la mărimi de  $3 \times 3$ , atunci există doar 28 de contururi concave distincte, reprezentate în figura 4.43. Fiecare din cele 28 de contururi concave este notat cu  $Q_{i,T}$ , unde indicele  $T$  definește tipul de contur concav, iar  $i$  definește direcția de start a codului lanț al conturului. Există cinci tipuri, și anume  $T=U, J, L, V$  sau  $r$ .

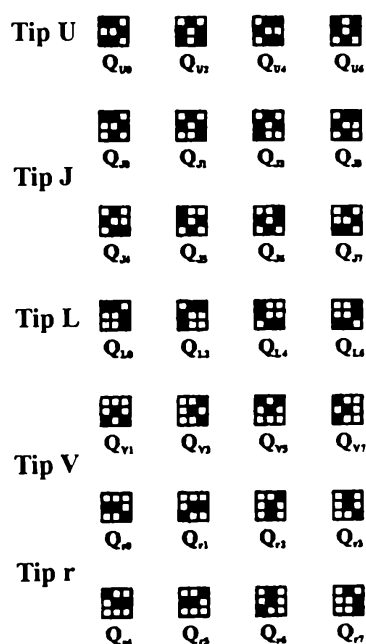


Fig. 4.43. Contururi concave pentru imagini de  $3 \times 3$  pixeli

Se spune că o imagine  $A$  este un factor al unei imagini  $X$  dacă și numai dacă  $X = A \oplus B$  pentru o imagine oarecare  $B$ . Un factor  $A$  al lui  $X$  este un *factor trivial* dacă și numai dacă  $A=X$  sau  $A$  este o imagine de un singur pixel. Un factor  $A$  al lui  $X$  este *factor prim* dacă și numai dacă fiecare factor al lui  $A$  este un factor trivial.

Pentru descompunere este utilizat conceptul de factorizare. Totuși, nu orice element structurant de formă arbitrară poate fi descompus. Park și Chin au dedus condiția necesară, dar nu și suficientă, pentru ca un element structurant  $X$  să poată fi descompus. Această condiție a fost dedusă din proprietățile codurilor lanț și arată că elementele structurante care pot fi descompuse nu trebuie să includă direcțiile 0 și 7 în codurile lanț, sau tipul  $Q_{r0}$  de contur concav.

*Chiar dacă Park și Chin prezintă această restricție ca pe una generală (atât pentru elemente concave cât și convexe), trebuie subliniat că rezultatele lor pot fi aplicate doar formelor concave. Pentru elementele convexe, a fost deja demonstrat că toate formele convexe pot fi descompuse.*

În [168] Park și Chin au propus un algoritm optimal pentru descompunerea elementelor structurante binare de formă arbitrară, simplu conectate, în elemente de tip  $3 \times 3$  pixeli. Descompunerea este optimizată în raport cu numărul elementelor de dimensiune  $3 \times 3$  pixeli. Lucrarea lor prezintă și multe exemple de descompunere a elementelor concave.

Deși rezultatele prezentate în [169] sunt foarte utile, pornind de la ideea că există un mare număr de elemente structurante neconvexe, eu consider că o soluție mai bună este aceea de a exprima elementul structurant concav  $B$  ca o reuniune a unui număr de poligoane convexe, ceea ce este *întotdeauna posibil*. Într-adevăr, dacă exprimăm

$$B = B_1 \cup B_2 \cup \dots \cup B_n \quad (4.95)$$

atunci dilatarea unei imagini poate fi implementată ca

$$X \oplus B = (X \oplus B_1) \cup (X \oplus B_2) \cup \dots \cup (X \oplus B_n) \quad (4.96)$$

în timp ce erodarea poate fi implementată ca

$$X \ominus B = (X \ominus B_1) \cap (X \ominus B_2) \cap \dots \cap (X \ominus B_n). \quad (4.97)$$

Rezultatele obținute pentru descompunerea elementelor structurante convexe pot fi, prin urmare, aplicate la orice formă arbitrară.

#### 4.5.3. Contribuții la descompunerea morfologică a elementului structurant

În cele ce urmează voi descrie câteva rezultate teoretice pe care le-am obținut în problema descompunerii morfologice a elementelor structurante.

Voi deduce câteva proprietăți interesante ale reprezentării domeniilor convexe prin coduri lanț de tip run-length, pe care le voi utiliza în continuare pentru a realiza descompunerea.

În continuare voi prezenta o interpretare grafică a dilatării și erodării domeniilor convexe.

În final, voi propune un set redus de 12 elemente structurante de bază care pot fi utilizate pentru descompunerea oricărui domeniu convex. Voi demonstra că acest set redus de elemente de bază pentru descompunere este capabil să dea rezultate similare cu cele furnizate de setul tradițional de 13 elemente prezentat în literatură ([168], [169], [284], [291]). În timpul demonstrației, voi propune de asemenea o metodă grafică simplă pentru realizarea descompunerii.



#### 4.5.3.1. Proprietăți ale reprezentărilor prin coduri lanț

După cum s-a arătat anterior, majoritatea cercetătorilor s-au axat pe problema descompunerii formelor convexe, care pot fi ulterior utilizate pentru descompunerea formelor arbitrare.

Clasele de obiecte care vor fi utilizate pentru rezolvarea problemei descompunerii sunt mulțimi discrete convexe, tetra-conectate, cu toate contururile orientate la unghiuri egale cu multipli de  $45^\circ$ . Lungimile laturilor de contur sunt multipli ai distanței dintre doi pixeli pentru direcțiile situate la  $0^\circ$  și  $90^\circ$ , respectiv multipli de  $\sqrt{2}$  ori distanța dintre acești pixeli pentru direcțiile situate la  $45^\circ$  și  $135^\circ$ . Obiectele acestei clase vor fi denumite în continuare simplu domenii convexe. Aceeași clasă de obiecte a fost studiată și de către Xu [284].

Câteva exemple de domenii convexe sunt prezentate în figura 4.44.

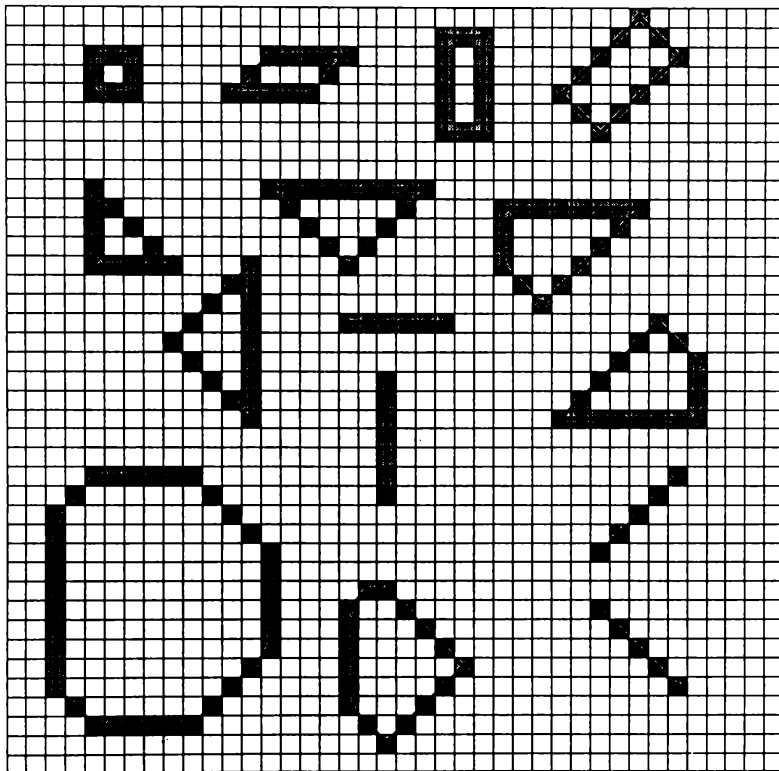


Fig. 4.44. Exemple de domenii convexe

În contextul morfologiei matematice, am considerat că una dintre cele mai eficiente căi de reprezentare a unui astfel de domeniu îl constituie utilizarea codurilor lanț de tip run-length, prezentate deja în paragraful 3.2.1.

În cele ce urmează voi deduce câteva proprietăți utile ale reprezentărilor formelor convexe prin intermediul acestor coduri, proprietăți pe care le-am găsit deosebit de interesante pentru calculul dilatării și erodării.

Fiind dată imaginea binară a unui domeniu convex  $B$ , reprezentarea sa prin coduri lanț de tip run-length nu este unică, deoarece singura restricție impusă punctului inițial este ca acesta să fie un vârf. Prin urmare, pentru un domeniu convex există atâtea reprezentări prin coduri lanț run-length câte vârfuri are domeniul respectiv. Pentru a se

evita posibilele ambiguități de reprezentare, voi utiliza în cele ce urmează convenția de mai jos:

*Punctul inițial pentru reprezentarea unui domeniu este întotdeauna vârful cel mai de jos și mai din stânga domeniului convex respectiv.*

Celelalte vârfuri sunt codate prin parcurgerea domeniului convex de-a lungul conturului său, în direcția inversă acelor de ceasornic, codând totodată lungimile muchiilor care formează domeniul  $B$ . Punctele interioare domeniului vor fi prin urmare întotdeauna punctele situate în stânga direcției de deplasare.

Reprezentarea prin coduri lanț obținută prin această convenție reprezintă o clasă de echivalență, clasa tuturor codurilor lanț de tip run-length care reprezintă domeniul convex considerat. Fiecare cod lanț de tip run-length din clasa de echivalență este o versiune rotită a altuia (printr-o modificare corespunzătoare a punctului inițial), dar el reprezintă cu toate acestea aceeași mulțime de puncte.

Pentru demonstrațiile următoare mă voi baza pe următoarele proprietăți, unele dintre ele fiind discutate mai pe larg în [284].

#### *Proprietatea 4.1.*

Orice domeniu convex poate fi reprezentat printr-un cod lanț run-length de următoarea formă

$$B = \{(i, j)(d_0, n_0)(d_1, n_1)(d_2, n_2)(d_3, n_3)(d_4, n_4)(d_5, n_5)(d_6, n_6)(d_7, n_7)\} \quad (4.98)$$

dând valorile corespunzătoare coeficienților  $n_i$ .

Trebuie precizat că  $(i, j)$  reprezintă punctul inițial, iar perechile ordonate descriu fiecare deplasament succesiv. Numărul de perechi ordonate este egal cu numărul de modificări ale direcției de deplasare. Într-o pereche ordonată  $(d_k, n_k)$ ,  $d_k \in \{d_0, d_1, \dots, d_7\}$  reprezintă direcția de deplasare, iar valorile întregi nenegative  $n_i$  reprezintă numărul de mișcări succesive pe respectiva direcție. Direcțiile  $d_0, d_1, \dots, d_7$  sunt aceleași ca și direcțiile  $0, \dots, 7$  din codul lanț original, care corespund unghiurilor de valoare  $\{0^\circ, 45^\circ, 90^\circ, 135^\circ, 180^\circ, 225^\circ, 270^\circ, 315^\circ\}$  față de axa pozitivă  $x$ . Aceasta înseamnă că  $d_0=(1,0)$ ,  $d_1=(1,1)$ ,  $d_2=(0,1)$ ,  $d_3=(-1,1)$ ,  $d_4=(-1,0)$ ,  $d_5=(-1,-1)$ ,  $d_6=(0,-1)$  și  $d_7=(1,-1)$ .

Prin urmare, în reprezentarea domeniilor convexe prin coduri lanț run-length, există întotdeauna opt vârfuri și opt deplasamente. Unghiurile acestor deplasamente cresc monoton de la  $d_0$  la  $d_7$ . Dacă nu există deplasament după una dintre direcții, atunci perechea respectivă poate fi eliminată din reprezentarea prin coduri lanț, valoarea parametrului  $n_i$  corespunzător fiind zero. În acest caz două vârfuri vor coincide.

Fiind dat un contur închis, deplasamentul total net la urmărirea completă a conturului este zero. Deoarece domeniile convexe reprezintă contururi închise, ele păstrează următoarele două proprietăți ale conturilor închise.

#### *Proprietatea 4.2.*

Suma deplasamentelor pe direcția pozitivă a axei  $x$  este egală cu suma deplasamentelor după direcția negativă a axei  $x$ :

$$n_0 + n_1 + n_7 = n_3 + n_4 + n_5 \quad (4.99)$$

*Proprietatea 4.3.*

Suma deplasamentelor pe direcția pozitivă a axei  $y$  este egală cu suma deplasamentelor după direcția negativă a axei  $y$ :

$$n_1 + n_2 + n_3 = n_5 + n_6 + n_7 \quad (4.100)$$

Din aceste proprietăți rezultă și următoarea:

*Proprietatea 4.4.*

Orice cod lanț de tip run-length de forma

$$B = \{(i, j)(d_0, n_0)(d_1, n_1)(d_2, n_2)(d_3, n_3)(d_4, n_4)(d_5, n_5)(d_6, n_6)(d_7, n_7)\}$$

ai cărui coeficienți  $n_i$  satisfac proprietățile reprezentate prin ecuațiile (4.99) și (4.100) este fie un domeniu convex, fie o linie la  $45^\circ$  sau la  $135^\circ$ . Liniile sunt cazuri speciale și sunt de forma  $B = \{(i, j)(d_1, n_1)(d_5, n_5)\}$  și  $B = \{(i, j)(d_3, n_3)(d_7, n_7)\}$ .

Fiind dată reprezentarea prin cod lanț run-length a unui domeniu convex  $B$ , toate cele opt vârfuri ale poligonului sunt unic definite și pot fi determinate în două moduri.

*Proprietatea 4.5.*

Să considerăm că vârful  $v_0$  este punctul inițial  $(i, j)$  al domeniului. Celelalte vârfuri pot fi obținute prin recurență. Fiind cunoscut al  $k$ -lea vârf  $v_k$ , coordonatele  $x$  și  $y$  ale celui de-al  $(k+1)$ -lea vârf sunt determinabile prin ecuațiile recursive:

$$x[v_{k+1}] = x[v_k] + n_k \cdot x[d_k] \quad (4.101)$$

$$y[v_{k+1}] = y[v_k] + n_k \cdot y[d_k] \quad (4.102)$$

pentru orice  $k \in N$ ,  $0 \leq k \leq 6$ . În această relație,  $x[v_0] = i$  și  $y[v_0] = j$  sunt coordonatele  $x$  și  $y$  ale punctului inițial al reprezentării prin cod lanț run-length.

În acest fel, coordonatele vârfurilor lui  $B$  pot fi calculate în funcție de punctul inițial al domeniului convex.

*Proprietatea 4.6.*

Coordonatele celui de-al  $k$ -lea vârf  $v_k$  pot fi calculate în funcție de poziția de start  $(i, j)$  și de lungimile  $n_l$ ,  $0 \leq l \leq k$ . Fie  $V_x$ ,  $V_y$ , și  $N$  matricile

$$V_x = \begin{bmatrix} x[v_0] \\ x[v_1] \\ x[v_2] \\ x[v_3] \\ x[v_4] \\ x[v_5] \\ x[v_6] \\ x[v_7] \end{bmatrix}, \quad V_y = \begin{bmatrix} y[v_0] \\ y[v_1] \\ y[v_2] \\ y[v_3] \\ y[v_4] \\ y[v_5] \\ y[v_6] \\ y[v_7] \end{bmatrix}, \quad N_x = \begin{bmatrix} i \\ n_0 \\ n_1 \\ n_2 \\ n_3 \\ n_4 \\ n_5 \\ n_6 \\ n_7 \end{bmatrix}, \quad N_y = \begin{bmatrix} j \\ n_0 \\ n_1 \\ n_2 \\ n_3 \\ n_4 \\ n_5 \\ n_6 \\ n_7 \end{bmatrix}. \quad (4.103)$$

Atunci

$$V_x = P_x \cdot N_x \text{ și } V_y = P_y \cdot N_y \quad (4.104)$$

unde

$$P_x = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & -1 & -1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & -1 & -1 & -1 & 0 & 0 \\ 1 & 1 & 1 & 0 & -1 & -1 & -1 & 0 & 0 \end{bmatrix} \quad (4.105)$$

și

$$P_y = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & -1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & -1 & -1 & 0 \end{bmatrix} \quad (4.106)$$

sunt matricile provenite din relațiile care definesc propagările pe axele  $x$  și  $y$ , respectiv pentru fiecare din direcțiile  $d_0, \dots, d_7$ .

#### 4.5.3.2. Operatori morfologici pentru domenii convexe

O altă modalitate de reprezentare a domeniilor convexe este utilizată de Haralick [73]. El consideră domeniile convexe ca rezultat al unor intersecții de semiplane discrete. Un domeniu convex  $B$  poate fi definit ca intersecție a opt semiplane discrete  $H_i$ ,  $0 \leq i \leq 7$ . Aceste semiplane discrete sunt funcții de direcțiile de bază ale deplasamentelor  $d_i$  și de vârfurile  $v_i$  ale domeniului convex. Fiecare semiplan discret  $H_i$  este definit în așa fel încât conturul său să treacă printr-un vârf  $v_i$ , iar muchia sa să fie de-a lungul direcției  $d_i$ . Semiplanul  $H_i$  este reprezentat de către toate punctele situate la stânga și pe contur la o parcurgere de-a lungul conturului pe direcția  $d_i$ .

Această reprezentare a fost utilizată în [73] pentru a deduce algoritmi pentru dilatare, erodare, deschidere, închidere, dilatare și erodare de ordinul  $n$  a domeniilor convexe. S-a demonstrat, de asemenea, că rezultatele la care s-a ajuns sunt echivalente cu cele obținute prin utilizarea morfologiei clasice.

Voi descrie în continuare pe scurt aceste rezultate, care sunt foarte utile pentru descompunerea domeniilor convexe.

Fie  $A$  și  $B$  două domenii convexe date prin reprezentarea lor prin cod lanț run-length:

$$A = \{(i_A, j_A)(d_0, n_0^A)(d_1, n_1^A)(d_2, n_2^A)(d_3, n_3^A)(d_4, n_4^A)(d_5, n_5^A)(d_6, n_6^A)(d_7, n_7^A)\} \quad (4.107)$$

$$B = \{(i_B, j_B)(d_0, n_0^B)(d_1, n_1^B)(d_2, n_2^B)(d_3, n_3^B)(d_4, n_4^B)(d_5, n_5^B)(d_6, n_6^B)(d_7, n_7^B)\} \quad (4.108)$$

și să notăm cu  $N_A$  și  $N_B$  vectorii lor coloană  $8 \times 1$ , cu elementele formate din lungimile de muchii respective.

Dacă  $C = A \oplus B$ , atunci s-a demonstrat că

$$(i_C, j_C) = (i_A, j_A) + (i_B, j_B) \quad (4.109)$$

și

$$N_C = N_A + N_B \quad (4.110)$$

Aceste relații furnizează o modalitate ușoară de lucru cu interpretarea geometrică a dilatării și erodării domeniilor convexe. Coordonatele originii domeniului dilatat sunt date de suma dintre coordonatele originii domeniului inițial și cele ale elementului structurant. Muchiile după fiecare direcție sunt egale cu sumele dintre muchiile domeniului original și cele ale elementului structurant.

Există o interpretare similară pentru operația de erodare, iar generalizările pentru dilatarea și erodarea de ordinul  $n$  sunt foarte ușor de obținut.

### 4.5.3.3. Descompunerea domeniilor convexe

#### 4.5.3.3.1. Definirea problemei

După cum s-a arătat în paragraful 4.5.2, există un număr mare de lucrări care raportează rezultate ale cercetărilor în domeniul descompunerii elementului structurant. În [284] s-a demonstrat că orice domeniu convex poate fi descompus în componente simple de bază. Același set de 13 elemente structurante de bază a fost prezentat, prin diferite metode de abordare, atât în [284] cât și în [291].

În cele ce urmează voi da o interpretare problemei descompunerii, prin intermediul reprezentării domeniilor convexe prin coduri lanț run-length. Voi introduce o soluție constructivă pentru descompunere și voi demonstra că, în realitate, *este necesar un set de doar 12 elemente de bază pentru realizarea descompunerii*. Elementul care nu este necesar pentru descompunere, din cele prezentate în lucrările anterioare, este elementul de tip romb.

#### 4.5.3.3.2. Setul de elemente de bază

Existența descompunerii pentru domeniile convexe fiind demonstrată, eforturile s-au concentrat pentru determinarea unui set de elemente de bază care să realizeze descompunerea. Independent unul de altul, Zhuang și Haralick [291], respectiv Park și Chin [168], au ajuns la un set similar de 13 elemente de bază, care diferă doar prin originea elementelor structurante (care nu afectează rezultatul final, datorită proprietăților de translație). Acest set de elemente de bază este prezentat în figura 4.45.

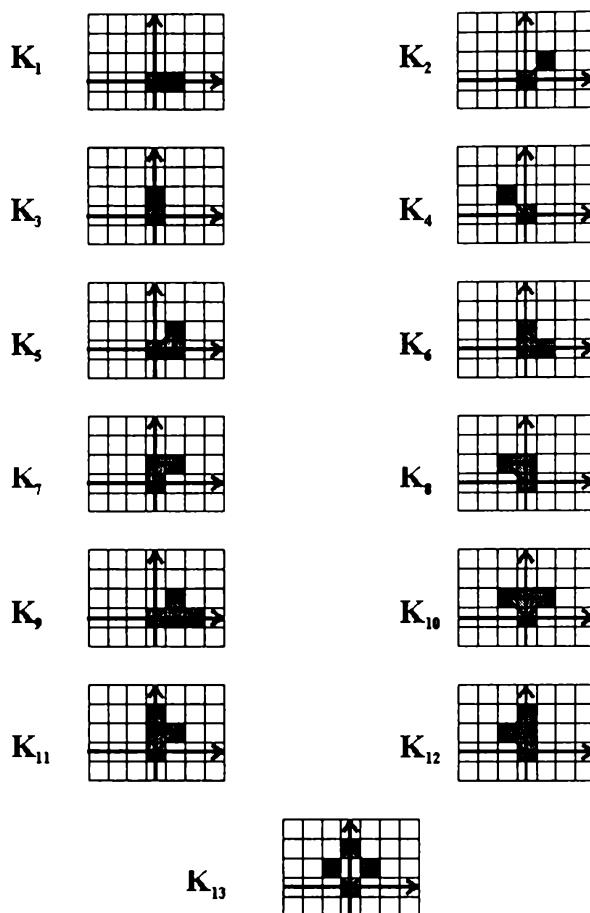


Fig. 4.45. Setul de 13 elemente structurante de bază

Lucrând cu setul de elemente structurante de bază  $\{K_1, K_2, \dots, K_{13}\}$ , orice domeniu convex poate fi descompus ca și combinație de dilatări de ordinul  $n$  a celor 13 elemente structurante de bază, printr-o relație de forma:

$$B = \left( \bigoplus_{k_1} K_1 \right) \oplus \left( \bigoplus_{k_2} K_2 \right) \oplus \dots \oplus \left( \bigoplus_{k_{13}} K_{13} \right) \quad (4.111)$$

unde  $\left( \bigoplus_{k_i} K_i \right)$  reprezintă dilatarea de ordinul  $k_i$  a elementului  $K_i$ .

Este foarte simplu de demonstrat că elementul structurant de bază  $K_{13}$  (rombul) nu este de fapt necesar pentru descompunere. Se poate demonstra că rombul este obținut printr-o dilatare a lui  $K_2$  cu  $K_4$ .

*Demonstrație.*

Dacă considerăm reprezentările lui  $K_2$ ,  $K_4$  și  $K_{13}$  ca mulțimi de puncte, obținem:

$$K_2 = \{(0,0), (1,1)\}$$

$$K_4 = \{(0,0), (-1,1)\}$$

$$K_2 \oplus K_4 = \{(0,0), (1,1), (-1,1), (0,2)\}$$

în timp ce rombul are reprezentarea

$$K_{13} = \{(0,0), (1,1), (0,2), (-1,1)\}.$$

Aceasta înseamnă că, de fapt,  $K_{13} = K_2 \oplus K_4$ , ceea ce trebuia demonstrat.

Figura 4.46 ilustrează acest rezultat.

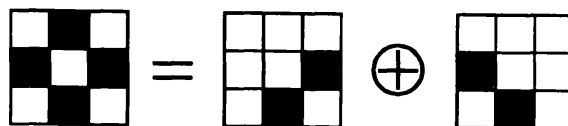


Fig. 4.46. Descompunerea rombului

Același rezultat poate fi obținut prin interpretarea geometrică a dilatării domeniilor convexe, prezentată în paragraful 4.5.3.2. Trebuie observat că liniile  $K_2$  și  $K_4$  nu sunt strict domenii convexe, deoarece ele nu sunt tetra-conectate. Totuși, ele satisfac toate regulile de operare cu domenii convexe și sunt în general utilizate de toți autorii, în problema descompunerii.

După ce am demonstrat acest lucru, voi utiliza în continuare un set de bază format din doar 12 elemente structurante. Voi demonstra că acest set este capabil să realizeze descompunerea și voi propune o nouă metodă de descompunere. Setul de bază  $\{K_1, K_2, \dots, K_{12}\}$  este reprezentat în figura 4.47.

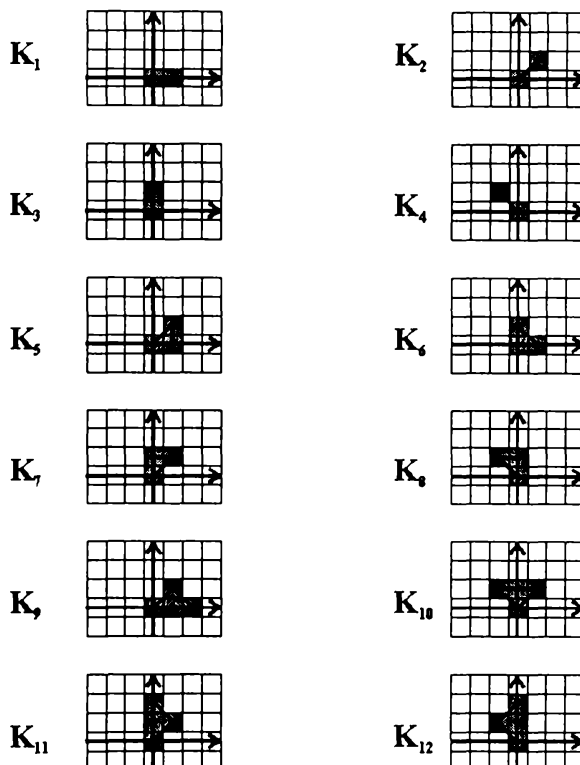


Fig. 4.47. Setul de bază de 12 elemente structurante

#### 4.5.3.3.3. Noua metodă de descompunere propusă

În cele ce urmează voi propune o nouă metodă pentru descompunerea domeniilor convexe ca o combinație de dilatări de ordinul  $n$  a celor 12 elemente structurante de bază reprezentate în figura 4.47. Reprezentarea prin coduri lanț run-length pentru aceste elemente structurante de bază este dată prin relațiile:

$$\begin{aligned}
K_1 &= \{(0,0)(d_0,1)(d_4,1)\} \\
K_2 &= \{(0,0)(d_1,1)(d_5,1)\} \\
K_3 &= \{(0,0)(d_2,1)(d_6,1)\} \\
K_4 &= \{(0,0)(d_3,1)(d_7,1)\} \\
K_5 &= \{(0,0)(d_0,1)(d_2,1)(d_5,1)\} \\
K_6 &= \{(0,0)(d_0,1)(d_3,1)(d_6,1)\} \\
K_7 &= \{(0,0)(d_1,1)(d_4,1)(d_6,1)\} \\
K_8 &= \{(0,0)(d_2,1)(d_4,1)(d_7,1)\} \\
K_9 &= \{(0,0)(d_0,2)(d_3,1)(d_5,1)\} \\
K_{10} &= \{(0,0)(d_1,1)(d_4,2)(d_7,1)\} \\
K_{11} &= \{(0,0)(d_1,1)(d_3,1)(d_6,2)\} \\
K_{12} &= \{(0,0)(d_2,2)(d_5,1)(d_7,1)\}
\end{aligned} \tag{4.112}$$

Aceasta înseamnă că orice domeniu convex  $B$  poate fi descompus ca o combinație de dilatări de ordinul  $k_i$  a celor 12 elemente structurante de bază  $K_i$ , prezentate în figura 4.47. În acest set de elemente structurante,  $K_i$  reprezintă doar triunghiuri sau linii.

$$B = \left( \bigoplus_{k_1} K_1 \right) \oplus \left( \bigoplus_{k_2} K_2 \right) \oplus \cdots \oplus \left( \bigoplus_{k_{12}} K_{12} \right) \tag{4.113}$$

După cum s-a arătat în paragraful 4.5.3.1, punctul inițial al reprezentării unui domeniu convex este considerat întotdeauna ca fiind vârful cel mai de jos și mai din stânga al domeniului convex. Această presupunere nu micșorează gradul de generalitate al metodei propuse, deoarece orice domeniu convex poate fi translatat cu originea în orice punct al spațiului printr-o simplă dilatare cu un element structurant de tip punct.

Comparând expresiile (4.98) și (4.113), se observă că, pentru ca egalitatea să fie adevărată, trebuie ca lungimile laturilor conturului domeniului convex reprezentat prin relațiile (4.98) și (4.113) să fie aceleași. Putem calcula dilatările din relația (4.113) prin algoritmul menționat în paragraful 4.5.3.2 și putem determina lungimile laturilor conturului domeniului convex rezultat în funcție de  $k_i$ ,  $1 \leq i \leq 12$ . Prin urmare, problema constă în a determina un set de valori  $k_i$  astfel încât lungimile laturilor domeniului convex rezultat să fie aceleași cu ale lui  $B$ . Deci, va trebui să determinăm valorile  $k_i$  astfel încât să fie satisfăcut următorul sistem:

$$\begin{aligned}
n_0 &= k_1 + k_5 + k_6 + 2k_9 \\
n_1 &= k_2 + k_7 + k_{10} + k_{11} \\
n_2 &= k_3 + k_5 + k_8 + 2k_{12} \\
n_3 &= k_4 + k_6 + k_9 + k_{11} \\
n_4 &= k_1 + k_7 + k_8 + 2k_{10} \\
n_5 &= k_2 + k_5 + k_9 + k_{12} \\
n_6 &= k_3 + k_6 + k_7 + 2k_{11} \\
n_7 &= k_4 + k_8 + k_{10} + k_{12}
\end{aligned} \tag{4.114}$$

unde  $n_i$  sunt lungimile laturilor domeniului convex  $B$ .



Setul de ecuații liniare (4.114) poate fi rescris, prin utilizarea reprezentării prin matrici, prin relația:

$$T \cdot K = N, \quad (4.115)$$

unde  $T$  este o matrice de transformare cu dimensiunile  $8 \times 12$ :

$$T = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 2 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}, \quad (4.116)$$

iar  $N$  este un vector cu valori întregi nenegative de dimensiuni  $8 \times 1$ , care reprezintă domeniul convex  $B$  prin intermediul lungimii laturilor sale:

$$N = [n_0 \ n_1 \ n_2 \ n_3 \ n_4 \ n_5 \ n_6 \ n_7]^T \quad (4.117)$$

și  $K$  este un vector de dimensiuni  $12 \times 1$  care reprezintă domeniul  $B$  prin descompunerea sa într-o succesiune de dilatări:

$$K = [k_1 \ k_2 \ k_3 \ k_4 \ k_5 \ k_6 \ k_7 \ k_8 \ k_9 \ k_{10} \ k_{11} \ k_{12}]^T \quad (4.118)$$

Având în vedere cele prezentate, problema descompunerii poate fi reformulată după cum urmează: fiind date matricea  $T$  și vectorul  $N$ , să se găsească un vector  $K$  cu elemente numere întregi și nenegative, care satisfac relația (4.115).

Matricea  $T$  este o transformare liniară de la un spațiu liniar 12-dimensional la un spațiu liniar 8-dimensional care transformă vectorul  $K$  în vectorul  $N$ . De aceea, pentru a găsi vectorul  $K$ , trebuie să găsim preimaginea vectorului  $N$  în transformarea  $T$ .

Deoarece dimensiunea spațiului nul  $K_h$  al transformării  $T$  nu este zero, vectorul  $N$  are mai mult decât o preimagine în  $T$  și există mai mult decât o singură posibilă descompunere a domeniului convex. În general putem exprima orice preimagine a lui  $N_B$ , vectorul  $N$  al domeniului convex  $B$ , ca sumă dintre o preimagine particulară  $K_p$  și un vector din spațiul nul  $K_h$ . În final, soluțiile generale  $K_g$  pot fi exprimate ca  $K_g = K_p + K_h$ . În cele ce urmează voi determina o soluție particulară  $K_p$ .

Interpretarea în spațiul vectorial este ilustrată în figura 4.48.

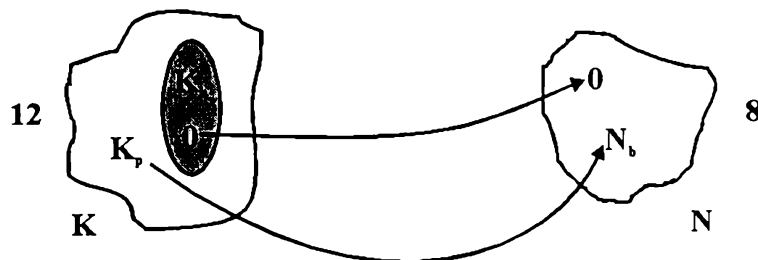


Fig. 4.48. Interpretarea în spațiul vectorial a problemei descompunerii

Voi determina o soluție particulară  $K_p$  a sistemului de ecuații (4.114) utilizând o interpretare geometrică a dilatării, așa cum am arătat în paragraful 4.5.3.2. Pentru

descompunere voi folosi același exemplu ca și Xu [284], dar voi utiliza setul redus de elemente structurante de bază, pe care l-am propus anterior. Descompunerea realizată de Xu se bazează pe inducție, în timp ce demonstrația mea se bazează pe determinarea unui algoritm constructiv. Găsirea unei soluții particulare a sistemului (4.114) demonstrează în același timp existența unei descompuneri pentru domeniile convexe, prin păstrarea gradului de generalitate a domeniului considerat.

Să considerăm domeniul convex prezentat în figura 4.49 (același domeniu utilizat și de Xu în exemplul său).

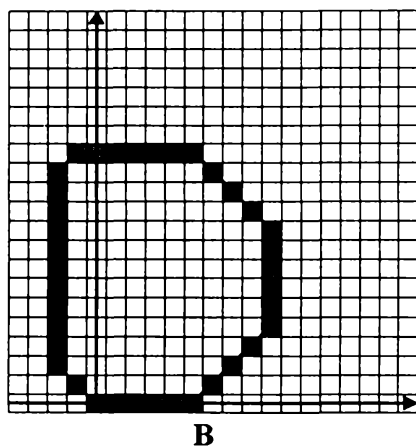


Fig. 4.49. Domeniul convex  $B$  care trebuie descompus

Demonstrația se bazează pe trei pași: descompunere pe direcțiile verticală și orizontală, descompunere după direcțiile diagonale la  $45^\circ$  și  $135^\circ$ , și respectiv, descompunere după triunghiuri.

*Pasul 1.*

Domeniul convex  $B$  poate fi descompus ca

$$B = B^{(1)} \oplus \left( \bigoplus_{k_1} K_1 \right) \oplus \left( \bigoplus_{k_3} K_3 \right), \quad (4.119)$$

unde

$$k_1 = \min[n_0, n_4], \quad (4.120)$$

$$k_2 = \min[n_2, n_6], \quad (4.121)$$

iar  $B^{(1)}$  este noul domeniu convex sau o linie diagonală, cu

$$(i_{B^{(1)}}, j_{B^{(1)}}) = (0, 0), \quad (4.122)$$

$$n_i^{(1)} = \begin{cases} n_i - k_1, & \text{daca } i \in \{0, 4\} \\ n_i - k_3, & \text{daca } i \in \{2, 6\} \\ n_i, & \text{daca } i \in \{1, 3, 5, 7\} \end{cases} \quad (4.123)$$

*Demonstrație*

Voi demonstra mai întâi că orice domeniu reprezentat printr-un cod lanț run-length cu valorile  $n_i$  definite prin relația (4.123) este un domeniu convex sau o linie diagonală, prin urmare  $B^{(1)}$  este un domeniu convex sau o linie diagonală. Apoi, voi

demonstra că domeniul convex obținut prin calculul dilatării din partea dreaptă a relației (4.119) este de fapt  $B$ .

Condiția ca  $B^{(1)}$  să fie un domeniu convex impune coeficienților  $n_i^{(1)}$ ,  $0 \leq i \leq 7$  satisfacerea următoarelor ecuații:

$$n_0^{(1)} + n_1^{(1)} + n_7^{(1)} = n_3^{(1)} + n_4^{(1)} + n_5^{(1)} \quad (4.124)$$

și

$$n_1^{(1)} + n_2^{(1)} + n_3^{(1)} = n_5^{(1)} + n_6^{(1)} + n_7^{(1)} \quad (4.125)$$

Prin utilizarea definiției lui  $n_i^{(1)}$  dată de ecuația (4.123), relația (4.124) poate fi rescrisă ca:

$$[n_0 - k_1] + n_1 + n_7 = n_3 + [n_4 - k_1] + n_5. \quad (4.126)$$

Simplificând, obținem relația

$$n_0 + n_1 + n_7 = n_3 + n_4 + n_5 \quad (4.127)$$

care este evident adevărată,  $B$  fiind un domeniu convex care satisface proprietatea 4.2 prezentată în paragraful 4.5.3.1.

Similar, se poate arăta că este adevărată și relația (4.127). Trebuie menționat că  $B^{(1)}$  este o linie diagonală dacă  $n_1^{(1)} = n_5^{(1)} \neq 0$  și  $n_i^{(1)} = 0$  pentru  $i \notin \{1,5\}$ , sau dacă  $n_3^{(1)} = n_7^{(1)} \neq 0$  și  $n_i^{(1)} = 0$  pentru  $i \notin \{3,7\}$ .

În continuare voi demonstra că dilatarea din partea dreaptă a relației (4.119) conduce la domeniul convex  $B$ . Din proprietățile dilatării prezentate în paragraful 4.5.3.2, rezultă că următoarele relații între  $B$  și  $B^{(1)}$  trebuie să fie adevărate:

$$(i_B, j_B) = (i_{B^{(1)}}, j_{B^{(1)}}) + k_1(i_{K_1}, j_{K_1}) + k_3(i_{K_3}, j_{K_3}) \quad (4.128)$$

și

$$n_i = n_i^{(1)} + k_1(n_i^{K_1}) + k_3(n_i^{K_3}), \quad 0 \leq i \leq 7. \quad (4.129)$$

Deoarece  $(i_{K_1}, j_{K_1}) = (i_{K_3}, j_{K_3}) = (i_{B^{(1)}}, j_{B^{(1)}}) = (0,0)$  și  $(i_B, j_B) = (0,0)$ , ecuația (4.128) este evident adevărată.

Prin extinderea părții din dreapta a relației (4.129) pentru  $0 \leq i \leq 7$ , vom obține următoarele:

Cazul  $i \in \{0,4\}$ :

$$n_i^{(1)} + k_1(n_i^{K_1}) + k_3(n_i^{K_3}) = [n_i - k_1] + k_1(1) + k_3(0) = n_i$$

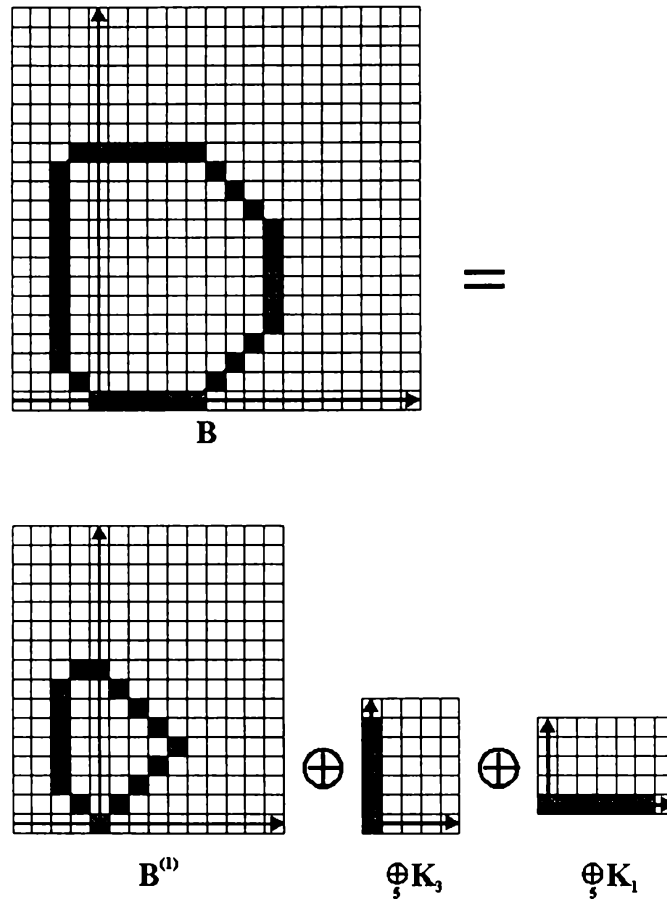
Cazul  $i \in \{2,6\}$ :

$$n_i^{(1)} + k_1(n_i^{K_1}) + k_3(n_i^{K_3}) = [n_i - k_3] + k_1(0) + k_3(1) = n_i$$

Cazul  $i \in \{1,3,5,7\}$ :

$$n_i^{(1)} + k_1(n_i^{K_1}) + k_3(n_i^{K_3}) = n_i + k_1(0) + k_3(0) = n_i.$$

Din acest motiv  $B$  este dilatarea lui  $B^{(1)}$ , de ordinul  $k_1$  cu  $K_1$  și de ordinul  $k_3$  cu  $K_3$ . Această descompunere este reprezentată în figura 4.50.

Fig. 4.50. Descompunerea lui  $B$  după direcțiile orizontale și verticale

*Pasul 2.*

Similar, putem reprezenta domeniul convex  $B^{(1)}$  ca dilatare după liniile diagonale la  $45^\circ$  și  $135^\circ$ . Putem exprima în general domeniul convex rezultat  $B^{(1)}$  ca fiind descompus după relația

$$B^{(1)} = B^{(2)} \oplus \left( \oplus_{k_2} K_2 \right) \oplus \left( \oplus_{k_4} K_4 \right) \quad (4.130)$$

unde

$$k_2 = \min[n_1^{(1)}, n_5^{(1)}], \quad (4.131)$$

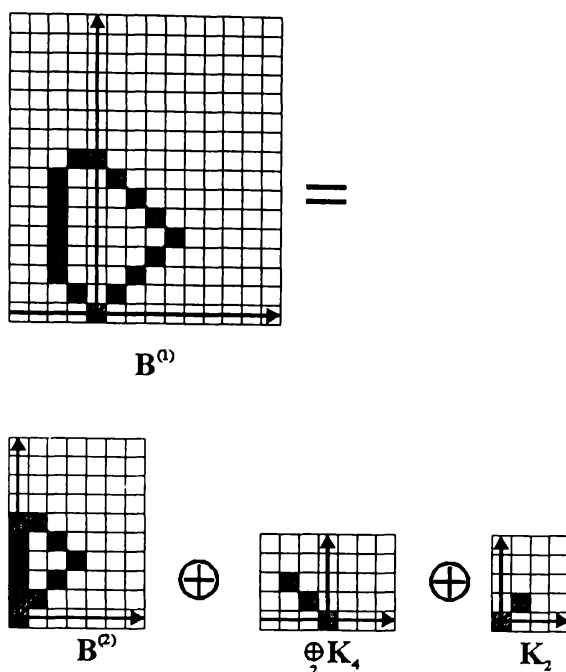
$$k_4 = \min[n_3^{(1)}, n_7^{(1)}], \quad (4.132)$$

și  $B^{(2)}$  este un domeniu convex cu

$$(i_{B^{(2)}}, j_{B^{(2)}}) = (0,0) \quad (4.133)$$

$$n_i^{(2)} = \begin{cases} n_i^{(1)} - k_2, & \text{daca } i \in \{1,5\} \\ n_i^{(1)} - k_4, & \text{daca } i \in \{3,7\} \\ n_i^{(1)}, & \text{daca } i \in \{0,2,4,6\} \end{cases} \quad (4.134)$$

Demonstrația acestei propoziții este similară cu cea anterioară, astfel încât nu voi prezenta detaliile demonstrației. Figura 4.51 prezintă descompunerea în continuare a domeniului convex  $B^{(2)}$ .

Fig. 4.51. Descompunerea lui  $B^{(1)}$  după linii diagonale

Rezultatul  $B^{(2)}$  este un domeniu convex cu patru sau mai puține laturi, care satisfac următoarele proprietăți:

$$n_0^{(2)} + n_1^{(2)} + n_7^{(2)} = n_3^{(2)} + n_4^{(2)} + n_5^{(2)}, \quad (4.135)$$

$$n_1^{(2)} + n_2^{(2)} + n_3^{(2)} = n_5^{(2)} + n_6^{(2)} + n_7^{(2)}, \quad (4.136)$$

$$n_0^{(2)} \text{ sau } n_4^{(2)} \text{ sau ambele} = 0, \quad (4.137)$$

$$n_1^{(2)} \text{ sau } n_5^{(2)} \text{ sau ambele} = 0, \quad (4.138)$$

$$n_2^{(2)} \text{ sau } n_6^{(2)} \text{ sau ambele} = 0, \quad (4.139)$$

$$n_3^{(2)} \text{ sau } n_7^{(2)} \text{ sau ambele} = 0, \quad (4.140)$$

și

$$n_i^{(2)} \geq 0, \text{ pentru } 0 \leq i \leq 7. \quad (4.141)$$

### Pasul 3.

Domeniul convex  $B^{(2)}$  poate fi descompus în continuare. În cazul în care  $B^{(2)}$  este un domeniu convex cu patru laturi, el poate fi descompus ca dilatare de ordinul  $n$  a două triunghiuri din setul de elemente de bază. Dacă  $B^{(2)}$  este un triunghi, atunci el poate fi exprimat ca dilatare de ordinul  $n$  a unuia din triunghiurile setului de bază. Altfel,  $B^{(2)}$  este doar un punct, originea, și nu mai trebuie descompus în continuare.

Voi demonstra în continuare că, dacă domeniul convex  $B^{(2)}$  are elemente  $n_i^{(2)}$  nenule, el poate fi descompus în continuare fie ca dilatare de ordinul  $n$  a unuia dintre elementele triunghiulare din setul de elemente structurante de bază, fie ca dilatare de ordinul  $n$  a două elemente structurante triunghiulare diferite din setul de bază.

*Demonstrație.*

Se poate verifica ușor din proprietățile lui  $B^{(2)}$  că există doar 8 domenii convexe cu patru laturi care satisfac condițiile impuse prin ecuațiile (4.135)-(4.141). Aceste domenii sunt următoarele:

$$a) \left\{ (0,0) \left( d_0, n_0^{(2)} \right) \left( d_1, n_1^{(2)} \right) \left( d_3, n_3^{(2)} \right) \left( d_6, n_6^{(2)} \right) \right\}, \quad (4.142)$$

$$b) \left\{ (0,0) \left( d_1, n_1^{(2)} \right) \left( d_2, n_2^{(2)} \right) \left( d_4, n_4^{(2)} \right) \left( d_7, n_7^{(2)} \right) \right\}, \quad (4.143)$$

$$c) \left\{ (0,0) \left( d_0, n_0^{(2)} \right) \left( d_2, n_2^{(2)} \right) \left( d_3, n_3^{(2)} \right) \left( d_5, n_5^{(2)} \right) \right\}, \quad (4.144)$$

$$d) \left\{ (0,0) \left( d_1, n_1^{(2)} \right) \left( d_3, n_3^{(2)} \right) \left( d_4, n_4^{(2)} \right) \left( d_6, n_6^{(2)} \right) \right\}, \quad (4.145)$$

$$e) \left\{ (0,0) \left( d_2, n_2^{(2)} \right) \left( d_4, n_4^{(2)} \right) \left( d_5, n_5^{(2)} \right) \left( d_7, n_7^{(2)} \right) \right\}, \quad (4.146)$$

$$f) \left\{ (0,0) \left( d_0, n_0^{(2)} \right) \left( d_3, n_3^{(2)} \right) \left( d_5, n_5^{(2)} \right) \left( d_6, n_6^{(2)} \right) \right\}, \quad (4.147)$$

$$g) \left\{ (0,0) \left( d_1, n_1^{(2)} \right) \left( d_4, n_4^{(2)} \right) \left( d_6, n_6^{(2)} \right) \left( d_7, n_7^{(2)} \right) \right\}, \quad (4.148)$$

$$h) \left\{ (0,0) \left( d_0, n_0^{(2)} \right) \left( d_2, n_2^{(2)} \right) \left( d_5, n_5^{(2)} \right) \left( d_7, n_7^{(2)} \right) \right\}. \quad (4.149)$$

Toate cele opt posibilități pentru  $B^{(2)}$  sunt prezentate în figura 4.52. Se poate observa că toate domeniile convexe sunt versiuni rotite unul altuia.

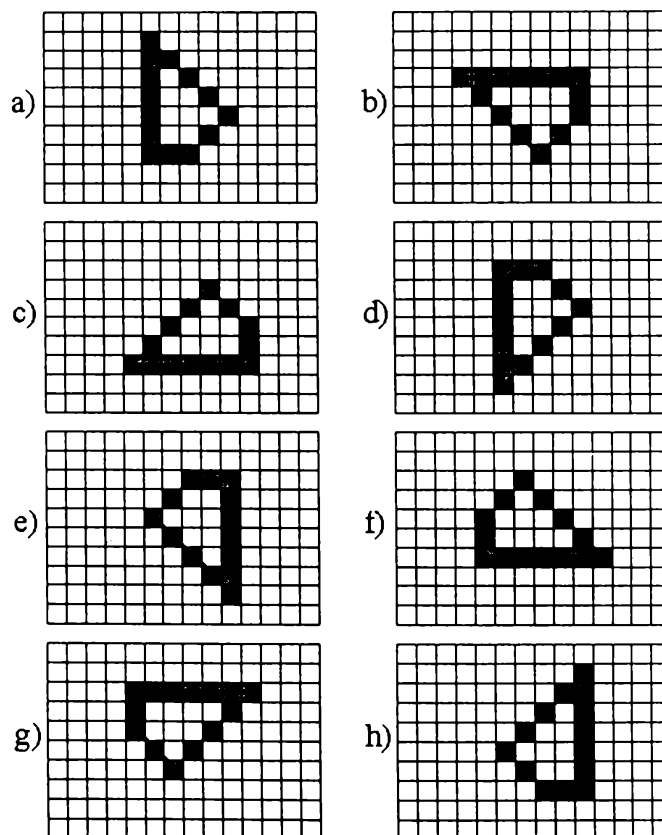


Fig. 4.52. Cele opt posibilități pentru  $B^{(2)}$

În plus, unii dintre coeficienții  $n_i^{(2)}$  pot fi zero, făcând astfel ca  $B^{(2)}$  să fie un triunghi sau un punct (originea), depinzând de faptul că la descompunerea anterioară s-a obținut un domeniu convex cu laturi opuse egale sau nu. Unul sau toți cei patru

coeficienți  $n_i^{(2)}$  pot fi zero. Nu este posibil ca doar doi coeficienți  $n_i^{(2)}$  să fie nenuli, deoarece în acest caz relațiile (4.135) și (4.136) nu vor fi respectate. Dacă toți coeficienții  $n_i^{(2)}$  sunt zero, atunci  $B^{(2)}$  este un punct singular, originea, prin urmare nu mai trebuie descompus în continuare. Cazul a doar trei coeficienți  $n_i^{(2)}$  nenuli poate fi tratat ca un caz special a patru coeficienți  $n_i^{(2)}$  nenuli. Prin urmare, va trebui demonstrat că domeniul  $B^{(2)}$  cu patru coeficienți  $n_i^{(2)}$  nenuli poate fi descompus ca dilatare de ordinul  $n$  a două triunghiuri.

Voi soluționa această problemă de descompunere pentru cele opt variante posibile de domeniu  $B^{(2)}$  pe baza unei abordări caz după caz.

*Cazul a).*

$$B^{(2)} = \left\{ (0,0) \left( d_0, n_0^{(2)} \right) \left( d_1, n_1^{(2)} \right) \left( d_3, n_3^{(2)} \right) \left( d_6, n_6^{(2)} \right) \right\} \quad (4.150)$$

În acest caz, domeniile convexe obținute după descompunere trebuie să aibă în mod necesar  $n_2 = n_4 = n_5 = n_7 = 0$ , deoarece în conformitate cu regulile dilatării, dilatarea unui domeniu convex, având un  $n_i$  nenul, cu orice alt domeniu convex, va avea un  $n_i$  nenul. Acest lucru reiese direct din relația (4.114). Singurele elemente structurante de bază care satisfac aceste condiții sunt  $K_6$  și  $K_{11}$ . Prin urmare va trebui determinat  $k_6$  și  $k_{11}$  astfel încât

$$B^{(2)} = \left( \bigoplus_{k_6} K_6 \right) \oplus \left( \bigoplus_{k_{11}} K_{11} \right). \quad (4.151)$$

Dacă calculăm dilatarea din membrul drept al acestei relații, se vede că  $k_6$  și  $k_{11}$  trebuie să satisfacă următoarele condiții:

$$n_0^{(2)} = k_6 \quad (4.152)$$

$$n_1^{(2)} = k_{11} \quad (4.153)$$

$$n_3^{(2)} = k_6 + k_{11} \quad (4.154)$$

$$n_6^{(2)} = k_6 + 2k_{11}. \quad (4.155)$$

Dacă  $k_6 = n_0^{(2)}$  și  $k_{11} = n_1^{(2)}$ , rezultă din (4.135) și (4.136) că relațiile (4.154) și (4.155) sunt îndeplinite. Prin urmare, a fost găsită o descompunere pentru  $B^{(2)}$ .

Toate celelalte cazuri pot fi abordate într-o manieră similară. Voi prezenta doar rezultatele finale.

*Cazul b).*

$$B^{(2)} = \left\{ (0,0) \left( d_1, n_1^{(2)} \right) \left( d_2, n_2^{(2)} \right) \left( d_4, n_4^{(2)} \right) \left( d_7, n_7^{(2)} \right) \right\}$$

va fi descompus ca

$$B^{(2)} = \left( \bigoplus_{k_8} K_8 \right) \oplus \left( \bigoplus_{k_{10}} K_{10} \right) \quad (4.156)$$

unde  $k_8 = n_2^{(2)}$  și  $k_{10} = n_1^{(2)}$ .

*Cazul c).*

$$B^{(2)} = \left\{ (0,0) \left( d_0, n_0^{(2)} \right) \left( d_2, n_2^{(2)} \right) \left( d_3, n_3^{(2)} \right) \left( d_5, n_5^{(2)} \right) \right\}$$

va fi descompus ca

$$B^{(2)} = \left( \bigoplus_{k_5} K_5 \right) \oplus \left( \bigoplus_{k_9} K_9 \right) \quad (4.157)$$

unde  $k_5 = n_2^{(2)}$  și  $k_9 = n_3^{(2)}$ .

*Cazul d).*

$$B^{(2)} = \left\{ (0,0) \right\} \left( d_1, n_1^{(2)} \right) \left( d_3, n_3^{(2)} \right) \left( d_4, n_4^{(2)} \right) \left( d_6, n_6^{(2)} \right) \right\}$$

va fi descompus ca

$$B^{(2)} = \left( \bigoplus_{k_7} K_7 \right) \oplus \left( \bigoplus_{k_{11}} K_{11} \right) \quad (4.158)$$

unde  $k_7 = n_4^{(2)}$  și  $k_{11} = n_3^{(2)}$ .

*Cazul e).*

$$B^{(2)} = \left\{ (0,0) \right\} \left( d_2, n_2^{(2)} \right) \left( d_4, n_4^{(2)} \right) \left( d_5, n_5^{(2)} \right) \left( d_7, n_7^{(2)} \right) \right\}$$

va fi descompus ca

$$B^{(2)} = \left( \bigoplus_{k_8} K_8 \right) \oplus \left( \bigoplus_{k_{12}} K_{12} \right) \quad (4.159)$$

unde  $k_8 = n_4^{(2)}$  și  $k_{12} = n_5^{(2)}$ .

*Cazul f).*

$$B^{(2)} = \left\{ (0,0) \right\} \left( d_0, n_0^{(2)} \right) \left( d_3, n_3^{(2)} \right) \left( d_5, n_5^{(2)} \right) \left( d_6, n_6^{(2)} \right) \right\}$$

va fi descompus ca

$$B^{(2)} = \left( \bigoplus_{k_6} K_6 \right) \oplus \left( \bigoplus_{k_9} K_9 \right) \quad (4.160)$$

unde  $k_6 = n_6^{(2)}$  și  $k_9 = n_5^{(2)}$ .

*Cazul g).*

$$B^{(2)} = \left\{ (0,0) \right\} \left( d_1, n_1^{(2)} \right) \left( d_4, n_4^{(2)} \right) \left( d_6, n_6^{(2)} \right) \left( d_7, n_7^{(2)} \right) \right\}$$

va fi descompus ca

$$B^{(2)} = \left( \bigoplus_{k_7} K_7 \right) \oplus \left( \bigoplus_{k_{10}} K_{10} \right) \quad (4.161)$$

unde  $k_7 = n_6^{(2)}$  și  $k_{10} = n_7^{(2)}$ .

*Cazul h).*

$$B^{(2)} = \left\{ (0,0) \right\} \left( d_0, n_0^{(2)} \right) \left( d_2, n_2^{(2)} \right) \left( d_5, n_5^{(2)} \right) \left( d_7, n_7^{(2)} \right) \right\}$$

va fi descompus ca

$$B^{(2)} = \left( \bigoplus_{k_5} K_5 \right) \oplus \left( \bigoplus_{k_{12}} K_{12} \right) \quad (4.162)$$

unde  $k_5 = n_0^{(2)}$  și  $k_{12} = n_7^{(2)}$ .

Singurul caz rămas este cel în care  $B^{(2)}$  este un triunghi, adică, trei din coeficienții  $n_i$  sunt nenuli. Problema descompunerii poate fi abordată și în acest caz prin



utilizarea soluției obținute pentru cazul domeniilor cu patru laturi. Singura diferență constă în faptul că de această dată unul din cei doi coeficienți  $k_i$  va fi zero.

Figura 4.53 prezintă descompunerea obținută pentru domeniul convex  $B^{(2)}$  considerat ca exemplu.

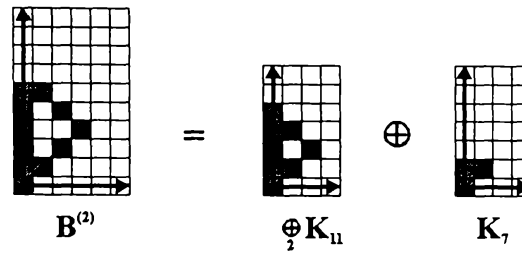


Fig. 4.53. Descompunerea domeniului  $B^{(2)}$

În concluzie, am demonstrat că orice domeniu convex poate fi descompus ca o combinație de dilatări de ordinul  $k_i$  a elementelor structurante din setul propus de 12 elemente  $K_i$  de bază. Totodată, am propus o metodă grafică de realizare a descompunerii.

## Capitolul 5

### Calitatea și performanțele tehnicilor de compresie a imaginilor

#### 5.1. Măsurarea performanțelor compresiei

Performanțele unei tehnici particulare de compresie a imaginilor sunt măsurate printr-o serie de parametri. Selectarea unei tehnici de compresie pentru o anumită aplicație trebuie să se bazeze pe clasificarea tehnicilor de compresie (compresie fără pierderi sau cu pierderi, codare cu rată de bit constantă sau variabilă), pe parametrii care definesc performanța și pe cerințele aplicației.

Primul parametru utilizat pentru măsurarea performanțelor compresiei este *raportul de compresie*. Acesta este definit ca raport între cantitatea inițială de date și cantitatea de date rezultată în urma compresiei. La prima vedere s-ar părea că o tehnică de compresie este cu atât mai bună cu cât raportul de compresie este mai mare. Această observație este valabilă pentru tehnicile de codare reversibile, fără pierderi. În cazul tehnicilor de compresie nereversibile, cu pierderi, se impune luarea în considerație a unui al doilea parametru: *calitatea medie a reconstrucției*. Acest parametru este apreciat de obicei prin raportul semnal-zgomot, el demonstrând că nu este util să avem un raport de compresie foarte bun, în condițiile unei calități inacceptabile a reconstrucției. În general este adevărat că pe măsură ce crește raportul de compresie scade calitatea reconstrucției pentru tehnicile de codare nereversibile.

Alți parametri importanți pentru evaluarea performanțelor sunt *complexitatea implementării* și *viteza de compresie*. Desigur, cu cât este mai simplă implementarea și cu cât este mai rapidă compresia, cu atât metoda este mai bună. Viteza de compresie este un parametru foarte important, în special pentru aplicațiile în timp-real, ca de exemplu sistemele de videoconferință. Pentru aplicații de tipul restabilire de informații sau prezentări multimedia, vitezele de compresie și decompresie trebuie luate în considerație separat. În aceste cazuri, compresia este făcută de obicei o singură dată, off-line, pe când decompresia se face de mai multe ori și trebuie realizată on-line. Din acest motiv, viteza de compresie nu este foarte importantă la aceste aplicații. Unele tehnici de compresie au viteze similare de compresie și decompresie. Ele sunt denumite *tehnici simetrice de compresie*. Alte tehnici sunt mai lente la compresie, dar sunt rapide la decompresie. Ele sunt denumite *tehnici asimetrice*.

#### 5.2. Măsurări cantitative pentru calitatea imaginii

Toate măsurătorile pentru aprecierea calității imaginii compresate au drept bază comună utilizarea erorii reziduale, adică diferența pixel cu pixel dintre imaginea originală și cea reconstruită după compresie. Pentru a exprima informația conținută în această imagine reziduală printr-un singur număr pot fi definite diferite valori. Numărul care reprezintă măsura erorii trebuie să satisfacă următoarele condiții:

- să fie un număr pozitiv;
- să fie zero, dacă imaginea compresată este identică cu cea originală;
- să crească monoton odată cu înrăutățirea calității imaginii.

Cea mai simplă măsură a erorii este *eroarea medie absolută MAE* (*mean absolute error*) sau eroarea/pixel, definită ca:

$$MAE = \frac{1}{mn} \sum_{x=1}^m \sum_{y=1}^n |f(x, y) - g(x, y)| \quad (5.1)$$

unde  $f(x, y)$  reprezintă imaginea originală, iar  $g(x, y)$  reprezintă imaginea reconstruită, ambele de dimensiuni  $m \times n$ .

O altă măsură a erorii o constituie *eroarea medie pătratică MSE* (*mean square error*), definită prin relația:

$$MSE = \frac{1}{mn} \sum_{x=1}^m \sum_{y=1}^n [f(x, y) - g(x, y)]^2 \quad (5.2)$$

Pentru a compara între ele cele două relații menționate anterior, ultima este înlocuită prin rădăcina sa pătrată, rezultând așa numita *RMSE* (*root mean square error*):

$$RMSE = \frac{1}{mn} \sqrt{\sum_{x=1}^m \sum_{y=1}^n [f(x, y) - g(x, y)]^2} \quad (5.3)$$

Aceste valori verifică primele două criterii pe care trebuie să le satisfacă măsura erorii, dar nu și pe a treia. De exemplu, dacă considerăm două imagini identice, având doar luminața diferită cu 40 pentru fiecare pixel, vom obține o valoare a RMSE egală cu 20, ceea ce indică o imagine de slabă calitate. De asemenea, dacă considerăm două imagini identice translatate ușor una față de alta, se obține o valoare mare a erorii, datorită faptului că diferențele nu se fac între pixeli identici.

Din aceste motive, măsurile definite anterior oferă doar o indicație relativă cu privire la calitatea imaginii, decizia finală trebuind însă să fie luată doar în concordanță cu caracteristicile percepției vizuale umane.

În afara măsurilor definite anterior, pot fi utilizate pentru caracterizarea erorii și așa numitele rapoarte semnal-zgomot. Ele indică raportul dintre puterea semnalului original și puterea erorii. În literatură sunt prezentate mai multe modalități de a defini astfel de parametri, cei mai uzuali fiind:

- raportul semnal/zgomot, care utilizează MSE ca și criteriu de eroare:

$$SNR_{ms} = \frac{\sum_{x=1}^m \sum_{y=1}^n f(x, y)^2}{\sum_{x=1}^m \sum_{y=1}^n [f(x, y) - g(x, y)]^2} \quad (5.4)$$

- raportul semnal/zgomot, care utilizează RMSE ca și criteriu de eroare:

$$SNR_{rms} = \sqrt{\frac{\sum_{x=1}^m \sum_{y=1}^n f(x, y)^2}{\sum_{x=1}^m \sum_{y=1}^n [f(x, y) - g(x, y)]^2}} \quad (5.5)$$

- definirea logaritmică:

$$SNR_{\log} = 10 \cdot \log_{10} \left[ \frac{\sum_{x=1}^m \sum_{y=1}^n f(x, y)^2}{\sum_{x=1}^m \sum_{y=1}^n [f(x, y) - g(x, y)]^2} \right] \quad (5.6)$$

- rapoarte semnal de vârf/zgomot:

$$PSNR_{\log} = 10 \cdot \log_{10} \left[ \frac{\max[f(x, y)]^2}{\frac{1}{mn} \sum_{x=1}^m \sum_{y=1}^n [f(x, y) - g(x, y)]^2} \right] \quad (5.7)$$

sau:

$$PSNR_{\log} = 10 \cdot \log_{10} \left[ \frac{[\max[f(x, y)] - \min[f(x, y)]]^2}{\frac{1}{mn} \sum_{x=1}^m \sum_{y=1}^n [f(x, y) - g(x, y)]^2} \right] \quad (5.8)$$

sau:

$$PSNR_{\log} = 10 \cdot \log_{10} \left[ \frac{\text{Maximum Dynamic Range}^2}{\frac{1}{mn} \sum_{x=1}^m \sum_{y=1}^n [f(x, y) - g(x, y)]^2} \right] \quad (5.9)$$

unde *Maximum Dynamic Range* reprezintă domeniul dinamic maxim al semnalului și are de obicei valoarea 255, corespunzător la 256 niveluri de gri.

Relațiile de legătură între *RMSE* și *PSNR* sunt:

$$PSNR = 10 \cdot \log_{10} \left[ \frac{255^2}{RMSE^2} \right] = 20 \cdot \log_{10} \left[ \frac{255}{RMSE} \right] \quad (5.10)$$

și

$$RMSE = \frac{255}{10^{\frac{PSNR}{20}}} \quad (5.11)$$

Tabelul 5.1 prezintă corespondența dintre valorile *RMSE* și ale *PSNR* calculate în conformitate cu definițiile anterioare.

Tabelul 5.1. Corespondența între *RMSE* și *PSNR*

<i>PSNR</i>	<i>RMSE</i>	<i>RMSE</i>	<i>PSNR</i>
20,00	25,50	1,00	48,33
21,00	22,73	2,00	42,11
22,00	20,26	3,00	38,59
23,00	18,05	4,00	36,09
24,00	16,09	5,00	34,15
25,00	14,34	6,00	32,57
26,00	12,78	7,00	31,23
27,00	11,39	8,00	30,07
28,00	10,15	9,00	29,05
29,00	9,05	10,00	28,13
30,00	8,06	11,00	27,30
31,00	7,19	12,00	26,55
32,00	6,41	13,00	25,85
33,00	5,71	14,00	25,21
34,00	5,09	15,00	24,61
35,00	4,53	16,00	24,05
36,00	4,04	17,00	23,52
37,00	3,60	18,00	23,03
38,00	3,21	19,00	22,56
39,00	2,86	20,00	22,11
40,00	2,55	21,00	21,69

Desigur, analiza cantitativă a calității realizată prin intermediul acestor mărimi nu poate înlocui evaluarea subiectivă făcută în concordanță cu caracteristicile percepției vizuale umane. Din acest motiv, literatura de specialitate prezintă o serie de teste de evaluare subiectivă a calității imaginii.

### 5.3. Aprecierea subiectivă a calității imaginii pe baza sistemului vizual uman

Unul dintre cele mai importante obiective în proiectarea unui sistem de comunicare vizuală este acela de a reprezenta, transmite, stoca sau afișa doar acea informație care este percepută de ochiul uman. A transmite sau reda caracteristici ale imaginii pe care oricum un observator uman nu le poate percepe constituie o risipă plătită prin bandă de transmisie, memorie, viteză sau resurse de afișare. De aceea este necesar să înțelegem cum putem reprezenta imaginile într-un mod cât mai economic și cum să le transmitem cu fidelitatea cerută de ochiul uman. Prin urmare trebuie să cunoaștem câteva proprietăți ale vederii umane, care sunt utile pentru evaluarea calității imaginilor codate.

Un element important în determinarea calității necesare a imaginii este stabilirea scopului în care imaginea este utilizată. Deoarece relația dintre calitatea necesară a imaginilor și scopul în care ele sunt utilizate este foarte complicată, atunci când este apreciată subiectiv calitatea imaginii se are în vedere un spectator normal de televiziune.

Deși înțelegerea sistemului vizual uman a progresat sensibil în decursul ultimului secol, au fost deslușite multe mecanisme complexe ale sistemului vizual, dar suntem încă departe de a dezvolta un model pentru vederea umană care să fie simplu și în același timp puternic, care să poată fi utilizat pentru optimizarea sistemelor de codare. Din acest motiv, de multe ori sunt necesare teste subiective pentru evaluarea calității imaginilor. În afară de a permite optimizarea sistemelor de codare, testele subiective permit și monitorizarea, respectiv asigurarea service-ului pentru un sistem aflat în operare.

Există două tehnici principale de măsurare a calității răspunsului uman la diferiți stimuli: metode primare sau explicite și metode secundare sau implicite [156]. În cazul *metodelor primare*, un grup de subiecți examinează un set de imagini și iau decizii subiective cu privire la calitatea acestora. În cazul *metodelor secundare*, caracteristicile imaginilor sunt măsurate obiectiv, iar rezultatele sunt convertite în măsuri ale calității. Metodele primare sunt mai utile dacă distorsiunile introduse de prelucrare sunt foarte complexe ca aparență, așa cum este în general cazul tehnicilor de codare digitală a imaginilor.

Evaluările primare subiective se bazează pe două metode diferite: *metode de clasificare pe categorii* și *metode comparative*. Metodele de clasificare pe categorii se mai numesc și *metode de clasificare pe scară valorică*.

Pentru a efectua o clasificare a calității, subiecților li se prezintă, în condiții normale de observare, o secvență de imagini test obținute prin prelucrare în modalități predeterminate, aceștia urmând să asocieze fiecărei imagini (sau secvențe de imagini) o categorie dintr-un set dat de categorii posibile. Categoriile se bazează în general pe calitatea totală, scara utilizată fiind:

5. Excelentă
4. Bună
3. Medie

## 2. Slabă

## 1. Foarte slabă.

Se mai utilizează și o scară bazată pe vizibilitatea imperfecțiunilor, în conformitate cu următoarele categorii:

5. Imperceptibile
4. Perceptibile dar nesupărătoare
3. Ușor supărătoare
2. Supărătoare
1. Foarte supărătoare.

Răspunsurile subiecților depind în mod uzual de mai mulți factori, ca de exemplu: luminața imaginii, contrastul imaginii, luminozitatea ambientală, dimensiunile imaginii, distanța de vizionare, experiența și motivația personală a subiecților. Asemenea factori trebuiesc eliminați cu multă grijă în timpul experimentelor, pentru a evita rezultatele părtinitoare. De asemenea, variabilitatea observatorilor poate fi minimizată prin alegerea unor observatori antrenați sau experți, familiarizați cu principiile televiziunii și, în special, cu imperfecțiunile uzuale ale imaginilor. Trebuie totuși avut în vedere faptul că acești experți sunt prea sensibili, reacționând la imperfecțiunile imaginilor altfel decât consumatorul normal de imagini. Multe dintre variabilele menționate, precum și influența lor asupra evaluării subiective a calității imaginilor, au fost cuprinse în recomandările ITU-T 500 [20]. Aceasta face posibilă compararea unor rezultate furnizate de observatori diferiți la momente de timp diferite.

Rezultatele procedurilor de clasificare pe scară valorică sunt prezentate în mod normal prin calcularea așa numitului *Scor Mediu al Opiniilor* MOS (Mean Opinion Score)

$$MOS = \frac{\sum_{i=1}^k n_i C_i}{\sum_{i=1}^k n_i} \quad (5.12)$$

unde  $C_i$  este valoarea numerică care corespunde categoriei  $i$ ,  $n_i$  este numărul de rezultate în respectiva categorie, iar  $k$  este numărul de categorii care formează scala. Metodele de clasificare pe scară valorică sunt foarte populare în televiziunea difuzată, deoarece pot fi utilizate pentru realizarea unui sistem eficient de supraveghere și service al echipamentului folosit.

*Metodele comparative* constituie cea de-a doua tehnică utilizată pentru aprecierea subiectivă a calității imaginilor. În acest caz, subiecții compară imaginea rezultată în urma compresiei (care este o imagine imperfectă, distorsionată) cu o imagine de referință peste care s-au suprapus distorsiuni standardizate (de exemplu, zgomot alb). Compararea poate fi făcută pe două monitoare așezate unul lângă altul sau pe un singur monitor care afișează secvențial cele două imagini. Peste imaginea de referință se suprapun distorsiuni până când cele două imagini sunt percepute de către observator ca având aceeași calitate. Cantitatea de distorsiune care este introdusă pe imaginea de referință poate fi controlată de către observator, sau sunt utilizate imagini precalculate cu grad variabil de distorsiune, memorate și afișate într-o ordine dată.

Compararea imaginii de test cu imaginile de referință se poate face cu destulă acuratețe dacă cele două tipuri de distorsiune prezente în imagini sunt similare din punct de vedere vizual, de exemplu, zgomot aditiv cu diferite caracteristici spectrale. Imaginii

de test evaluate și se poate atribui o calitate (categorie), prin utilizarea scalei anterior definite pentru imaginile cu imperfecțiuni.

O variantă modificată a acestei metode constă în utilizarea de către observatori a următoarei scale de comparare a imaginii test cu imagini având nivelul de distorsiune cunoscut:

- 3 Mult mai bună
- 2 Mai bună
- 1 Puțin mai bună
- 0 La fel
- 1 Puțin mai proastă
- 2 Mai proastă
- 3 Mult mai proastă

În acest caz observatorul trebuie să răspundă la întrebarea “cu cât este mai bună sau mai proastă imaginea de test în comparație cu imaginea de referință?”. Datele obținute sunt apoi prelucrate pentru a determina așa numitul “punct de egalitate subiectivă” între imaginea de test și imaginile de referință distorsionate.

#### 5.4. Cerințele de calitate pentru informația video

Este util să examinăm care sunt cerințele pentru o transmisie ideală a informației video digitale, adică nivelul de calitate pe care trebuie să-l aibă drept obiectiv sistemele de comunicație video. Desigur că multe dintre caracteristicile sistemelor ideale nu pot fi îndeplinite în practică, astfel încât trebuie realizat un compromis între dorința unei foarte bune calități video și limitările sistemelor și rețelelor practice de comunicație.

##### 5.4.1. Calitatea imaginii

Într-un sistem ideal de comunicații video digitale, calitatea informației video percepute de către spectator nu trebuie să fie limitată de sistemul de codare sau de mediul de transmisie. Acest lucru nu este posibil în practică, deoarece sistemele reale de codare și transmisie produc o degradare suplimentară a calității imaginii, față de limitările introduse de sistemele de înregistrare și redare.

Dacă acceptăm că un sistem practic furnizează un anumit nivel de calitate, atunci este de dorit ca această calitate să fie constantă de la cadru la cadru. Calitatea secvenței video afișate trebuie să fie independentă de conținutul vizual al secvenței sau de starea curentă și utilizarea sistemului de transmisie. În practică, conținutul diferitelor scene poate avea un mare efect asupra caracteristicilor datelor codate și a calității decodării. În plus, calitatea datelor video transmise pe o rețea de comunicație poate fi afectată și de alți factori, precum intensitatea traficului pe rețea.

##### 5.4.2. Erori vizibile

Ideal, o secvență video decodată trebuie să nu prezinte nici o degradare datorită erorilor sau pierderilor din sistemul de comunicație. Orice mediu de transmisie introduce zgomot și/sau pierderi în informația transmisă. De exemplu, informația digitală transmisă prin undă radio poate fi afectată de erori datorate zgomotului sau

interferențelor, în timp ce datele transmise printr-o rețea cu comutare de pachete, cum este Internet-ul, poate fi afectată de pierderi de pachete.

Sistemul vizual uman este sensibil la erori și degradări ale informației video. O eroare care apare într-un singur cadru decodat va fi aparentă pentru doar 1/25 sau 1/30 dintr-o secundă, neavând un mare impact asupra imaginii percepute dacă nu afectează o zonă spațială mare din cadrul respectiv. Erorile frecvente care afectează cadre singulare de imagine vor apărea ca o “pâlpăire” în secvența video decodată. O eroare sau o zonă degradată care persistă pe durata mai multor cadre este mult mai evidentă pentru privitor, fiind din acest motiv mai importantă de evitat.

Un sistem de comunicație video trebuie să aibă drept scop minimizarea apariției erorilor în secvențele video decodate și mai ales a acelor erori care persistă mai multe cadre. Se pune problema eliminării acestui ultim tip de erori.

### 5.4.3. Întârzierea

Întârzierea este un factor important în cazul sistemelor de transmisie în timp real a informației video digitale, iar întârzierea totală este esențială pentru sistemele de comunicație video bidirecționale. Ideal, ar trebui să nu existe o întârziere perceptibilă la nici unul dintre capetele de transmisie. Orice întârziere semnificativă conduce la dificultăți supărătoare în comunicație; de exemplu, conversația normală este dificilă sau chiar imposibilă dacă întârzierea pe o cale este mai mare de câteva sute de milisecunde. Dacă informația audio este decalată față de cea video, atunci se ajunge la perceperea pierderii sincronizării (de exemplu, desincronizarea mișcării buzelor).

Au fost propuse sau sunt deja în curs de dezvoltare o serie de servicii video interactive (de exemplu, video la cerere, cumpărături la domiciliu, etc.). Pentru toate aceste servicii este foarte important să nu existe o întârziere semnificativă între momentul în care utilizatorul cere un serviciu și momentul în care el primește răspunsul corespunzător.

Pentru transmisiunile video unidirecționale (de exemplu, televiziunea digitală) este mai puțin semnificativă întârzierea totală, un parametru mult mai critic fiind modificarea acestei întârzieri. Cadrele de imagine, împreună cu informația audio asociată, trebuie să fie prezentate utilizatorului la o rată constantă. Orice variație a întârzierii între cadre codate succesive trebuie să fie “netezite” înainte ca acestea să fie afișate.

### 5.4.4. Rata transmisiei de date

Rata disponibilă de transmisie a datelor are un efect semnificativ asupra calității datelor video codate. Tehnicile practice de codare video sunt, de obicei, nereversibile. Prin urmare o parte din datele utile ale secvenței originale se pierd pe durata procesului de codare și compresie. În general, pentru orice tehnică de codare nereversibilă, valoarea medie a ratei de bit pentru datele codate este proporțională cu calitatea decodării. Într-un codor bazat pe transformarea cosinus DCT (de exemplu MPEG-1, MPEG-2, MPEG-4, H.261 sau H.263 [21, 38, 156]) pierderea de calitate apare pe durata procesului de cuantizare. Un pas de cuantizare mai mare conduce la o calitate mai slabă a secvenței decodate, conducând însă totodată la o compresie mai puternică. Prin urmare, pasul de



cuantizare este unul dintre parametrii de bază prin care se poate controla rata de bit a secvenței video codate.

Mai multe sisteme de comunicații video existente utilizează un canal cu rata de transmisie constantă (de exemplu, sistemele de videoconferință pe canale ISDN, televiziunea digitală). Datele video codate trebuie transmise în canal cu o *rată de bit constantă* CBR (Constant Bit Rate). Totuși, acest lucru crează anumite probleme deoarece datele video codate cu oricare din standardele principale au, inevitabil, o mare variație a ratei de bit. Prin natura sa, codarea DCT presupune că acele cadre care conțin o mare cantitate de detalii spațiale vor genera mai multe date decât cadrele care conțin puține detalii spațiale. Predicția temporală cu compensarea mișcării lucrează mai bine pentru scenele care conțin doar o mișcare limitată și liniară. Pentru scenele mai complexe, predicția mișcării este mai slabă și se generează mai multă informație eronată. De aceea, scenele cu multă mișcare sau cu tipuri mai complexe de mișcare vor produce mai multă informație codată decât scenele cu puțină mișcare. În concluzie, diferitele tehnici de codare utilizate, de exemplu standardul MPEG [158], produc cantități variabile de date. Imaginile de tip *I* au o eficiență a compresiei relativ mică și generează o mare cantitate de date. Imaginile de tip *P* generează mai puține date, iar imaginile de tip *B* generează cea mai puțină informație dintre cele trei tipuri de imagini. O secvență MPEG tipică, care conține o mixtură de imagini *I*, *P* și *B* va avea un număr foarte variabil de biți pe cadru codat, prin urmare o rată de transmisie a datelor foarte variabilă.

Pentru a adapta această rată variabilă de date la un canal CBR (cu rata de biți constantă), datele furnizate de un codor video trebuie înmagazinate într-o memorie tampon înaintea transmisiei. Aceasta are drept efect o oarecare netezire a variațiilor pe termen scurt ale ratei de date. Variațiile pe termen lung (datorate modificărilor în conținutul spațial și temporal al secvenței) nu pot fi netezite pe această cale, decât dacă se utilizează un buffer foarte mare. În schimb, tehnica uzuală constă în transmiterea unei reacții de la ieșirea decodurului către codor ca măsură a ratei de biți, care să permită ajustarea factorului de compresie. Această acțiune poate avea loc prin modificarea pasului de cuantizare.

Prin urmare, calitatea secvenței decodate variază dependent de rata de bit de la ieșirea codurului. Această modificare a calității poate fi foarte supărătoare pentru unele aplicații, în special la transmisiunile cu rată de bit scăzută. De exemplu, datele video codate conform standardului H.261 [21] și transmise printr-o interfață standard ISDN vor cunoaște o degradare semnificativă a scenelor care conțin multă mișcare.

Rețelele cu comutarea pachetelor pot suporta transmisii de date cu rată variabilă. Rețelele ATM pot suporta trafic cu *rată de bit variabilă* VBR (Variable Bit Rate) în timp real. Transmisiile VBR a semnalului video codat pot avea o serie de avantaje față de transmisiile CBR. Tehnicile care utilizează codarea DCT pot asigura o calitate constantă pentru datele video transmise, dacă pasul de cuantizare este menținut la un nivel fix. Această metodă de control a codurului, unde nu există reacție de la ieșirea codurului către parametrii de codare, este cunoscută sub denumirea de *codare în buclă deschisă*.

Totuși, codarea video în buclă deschisă poate crea probleme chiar și în rețelele proiectate pentru a suporta trafic VBR (cum sunt rețelele ATM). Atunci când se realizează o conexiune printr-o rețea ATM, trebuie specificate caracteristicile sursei, inclusiv rata medie și rata de vârf pentru datele transmise. Codarea video în buclă deschisă prezintă variații ale ratei de bit atât pe termen scurt cât și pe termen lung. De exemplu, dacă o scenă se modifică pe durata unei secvențe video, atunci rata de date sau

rata de bit de vârf continuă să varieze și se poate ajunge la nerespectarea parametrilor de trafic conveniți la stabilirea conexiunii.

Din scurta descriere pe care am făcut-o se vede că rețelele practice de comunicație prezintă limitări ale calității serviciilor furnizate.

### 5.5. Teoria rată-distorsiune

În compresia imaginilor una dintre întrebările care se pun este: care este rata de bit minimă necesară pentru a coda o sursă, păstrând degradarea calității sub un anumit nivel? Răspunsul la această întrebare este subiectul de cercetare pentru o ramură a teoriei informației cunoscută sub numele de *teoria rată-distorsiune* [181], [209]. Teoria rată-distorsiune stabilește limite teoretice ale performanțelor pentru tehnicile nereversibile de compresie a datelor în conformitate cu un criteriu de fidelitate. Această teorie arată că rata maximă pentru o transmisiune fără distorsiuni este limitată de entropia sursei, iar pentru o distorsiune dată  $D$ , rata de transmisiune maximă este  $R(D)$ . Pentru transmiterea informației cu o distorsiune mai mică sau egală cu  $D$  este necesar un canal de comunicație cu capacitatea mai mare decât  $R(D)$ . Rata de transmisiune  $R(D)$  reprezintă numărul de biți per pixel necesari pentru transmisiunea respectivă.

Utilizatorul dorește să obțină o bună reconstrucție a sursei din datele recepționate. Implicit, utilizatorul are o măsură a distorsiunii și un criteriu de fidelitate care-i permite să compare diverse sisteme de comunicație a imaginilor. În mod natural, utilizatorul dorește să realizeze o reproducere perfectă, dar în majoritatea cazurilor acest lucru nu este posibil datorită zgomotului de pe canalul de comunicație, a cerințelor de cost, a limitărilor de putere, a întârzierilor maxime acceptabile, etc. Datorită acestor constrângeri, utilizatorul va trebui să accepte anumite degradări, iar rolul proiectantului va fi acela de a minimiza distorsiunile de reproducere în condițiile impuse de constrângerile menționate.

Disciplina teoretică care tratează compresia de date din punctul de vedere al teoriei informației este chiar teoria rată-distorsiune. Principala problemă pe care această disciplină urmărește să o rezolve este aceea de a determina cel mai mic debit cu care trebuie transmisă către utilizator informația despre sursă, în așa fel încât să se realizeze o fidelitate prescrisă. Dorința de a utiliza un canal cu capacitatea minim admisibilă pentru scopul propus provine din faptul că atât complexitatea cât și costul canalului variază în general direct proporțional cu capacitatea sa.

#### 5.5.1. Funcția rată-distorsiune

Pentru o clasă largă de măsuri ale distorsiunii și de modele ale sursei, teoria rată-distorsiune furnizează o funcție a ratei de transmisie dependentă de distorsiune  $R(D)$ . Aceasta reprezintă rata efectivă cu care produce sursa informații, cu condiția ca utilizatorul să accepte o distorsiune medie  $D$ . Rata cu care sursa produce informație în condiții de reconstrucție perfectă se numește entropie a sursei. Dacă măsura distorsiunii este definită astfel încât o reproducere perfectă să corespundă distorsiunii zero, atunci  $R(0)$  egalează entropia sursei  $H$ . Atunci când  $D$  crește,  $R(D)$  descrește monoton și devine în general zero la o anumită valoare finită a distorsiunii. Se poate demonstra că  $R(D)$  este o funcție convexă, continuă și strict descrescătoare de  $D$ . **Forma generală a funcției  $R(D)$  este prezentată în figura 5.1.**

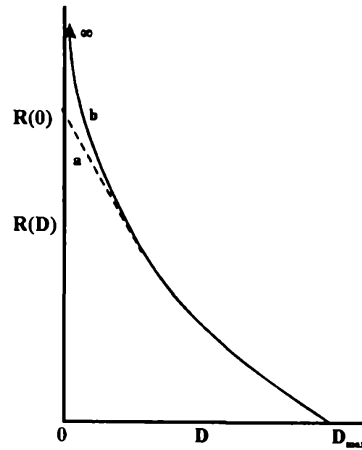


Fig. 5.1. Funcția tipică  $R(D)$ . a) cazul discret; b) cazul continuu

Regiunea centrală a curbei care exprimă rata de transmisie în funcție de distorsiune are o pantă negativă. La rate foarte mici de transmisie a informației, distorsiunea atinge o valoare maximă  $D_{max}$ , care, în cazul distribuției continue va fi chiar energia totală a semnalului. La cealaltă extremitate a scalei distorsiunii, pe măsură ce dorim o calitate tot mai bună a semnalului reprodus, adică o distorsiune tot mai mică, rata de transmisie trebuie să crească și, în cazul distribuției continue, devine infinită dacă se dorește o reproducție perfectă. Pe de altă parte, în cazul discret, se știe că reproducția perfectă este posibilă în principiu, cu condiția ca rata de codare să nu fie mai mică decât entropia sursei  $H(x)$ . Se obține o recepție corectă, adică un simbol  $j$  este transmis și recepționat ca atare. Atunci  $R(0) = H(x)$ . În cazul discret, există posibilități de decodare corectă a simbolurilor de intrare chiar în condițiile în care  $R(0) \leq H(x)$ .

Principalele proprietăți ale funcției rată-distorsiune pot fi rezumate astfel:

- Pentru orice nivel de distorsiune dat  $D$ , este posibilă găsirea unei scheme de codare cu rata de transmisie oricât de apropiată de  $R(D)$  și distorsiune medie oricât de apropiată de  $D$ .
- Este imposibil de găsit un cod care să realizeze o reproducere cu distorsiunea  $D$  sau mai bună, la o rată de transmisie sub  $R(D)$ .

### 5.5.2. Funcția rată-distorsiune în termenii teoriei informației

Să considerăm acum definiția funcției  $R(D)$  în termenii teoriei informației. Autoinformația conținută în evenimentul  $x = X$ , este

$$i(x) = -\log p(x) \quad (5.13)$$

Baza logaritmului este cea care definește unitatea pentru informație. Valoarea medie a lui  $i(x)$  este entropia

$$H(x) = -\sum_x p(x) \log p(x) \quad (5.14)$$

unde însumarea se face după toate valorile permise ale lui  $x$ . În mod similar,

$$H(y) = -\sum_y p(y) \log p(y) \quad (5.15)$$

unde  $y = Y$  reprezintă apariția, la recepție, a variabilei  $y$  care ia valoarea  $Y$ . Distribuția de două variabile  $x$  și  $y$  este  $p(x, y)$ , iar distribuțiile separate sunt obținute prin însumarea, după  $y$ , respectiv  $x$  din  $p(x, y)$ , adică,

$$p(x) = \sum_y p(x, y) \quad (5.16)$$

$$p(y) = \sum_x p(x, y) \quad (5.17)$$

Distribuțiile condiționate sunt definite prin

$$p(x/y) = \frac{p(x, y)}{p(y)} \quad (5.18)$$

$$p(y/x) = \frac{p(x, y)}{p(x)} \quad (5.19)$$

În termenii comunicării informației prin canal, sau al reconstrucției imaginii după codare și prelucrare, putem privi pe  $x$  și  $y$  drept cantitate transmisă sau originală, respectiv cantitate recepționată sau reconstruită, respectiv. Problema care se pune este: cât de mult spune apariția evenimentului  $y = Y$  despre starea intrării  $x = X$ ? Luând în considerație ecuația (5.13) și faptul că

$$i(x, y) = -\log p(x/y) \quad (5.20)$$

obținem informația mutuală ca

$$i(x, y) = i(x) - i(x/y) = \log [p(x, y) / p(x)] \quad (5.21)$$

Mediind această cantitate pentru distribuția  $p(x, y)$ , vom obține

$$I(x, y) = \sum_x \sum_y p(x, y) \log \frac{p(x, y)}{p(x)} \quad (5.22)$$

Am dori să maximizăm pe  $I(x, y)$ , adică să facem informația mutuală dintre emițător și receptor cât mai mare posibil. Mai mult, este de dorit ca, fiind dat un anumit criteriu de fidelitate, transmisiunea să aibă loc la o rată minimă. Problema va fi deci aceea de a minimiza ecuația (5.22) relativ la respectivul criteriu. Dacă acceptăm că distribuția la intrare este variabilă, atunci valoarea maximă a lui  $I(x, y)$  va fi capacitatea canalului  $C$  sau a sistemului care leagă pe  $x$  și  $y$ .

Să presupunem acum o sursă cu entropia  $H$  și un canal cu capacitatea  $C$  date. Dacă  $C < H$ , atunci teorema codării în prezența zgomotului arată că orice sistem de comunicație proiectat pentru transmiterea sursei pe canal trebuie să aibă o nesiguranță de cel puțin  $H - C$ . Cu alte cuvinte, o medie de cel puțin  $H - C$  biți de informație ai mesajului sursei sunt inevitabil pierduți în timpul traversării canalului. Să presupunem că suntem interesați să știm dacă, în aceste condiții, este posibilă reproducerea sursei la ieșirea canalului cu o fidelitate  $D$ . Pentru a fi posibil acest lucru, trebuie ca cel puțin  $R(D)$  biți de informație pe mesaj, din totalul de  $H$  biți pe mesaj produși de sursă, să ajungă la ieșire. Aceasta înseamnă că maxim  $H - R(D)$  biți pot fi pierduți. Prin urmare, este imposibil de obținut o distorsiune medie  $D$  dacă  $H - R(D)$  nu depășește  $H - C$ , sau echivalent dacă

$$R(D) \leq C \quad (5.23)$$

Aceasta este o condiție necesară pentru existența unui sistem de comunicație care operează cu fidelitatea  $D$ .

Să considerăm acum mai detaliat natura distorsiunilor care pot admise. În condiții ideale, simbolurile de intrare și ieșire vor fi întotdeauna identice, adică, dacă  $x = X$ , atunci și  $y = X$ . Pe de altă parte, atunci când este admisă o anumite distorsiune, acest lucru nu mai este adevărat.

Funcția distorsiune măsoară "distanța" dintre  $x$  și  $y$ , care poate fi scrisă sub forma  $d(x, y)$ . Dacă simbolurile sursă individuale  $x_1, x_2, \dots, x_n$  sunt grupate în cuvinte de lungime  $n$ , atunci un criteriu de fidelitate așa numit de "mesaj singular" este dat de

$$\rho_n(x, y) = \frac{1}{n} \sum_{k=1}^n d(x_k, y_k) \quad (5.24)$$

Acesta reprezintă media distorsiunilor simbolurilor individuale, făcută pe lungimea cuvântului. Ținând cont și de distribuțiile condiționate, distorsiunea va fi

$$D_{p(y/x)} = \sum_x \sum_y p(x, y) d(x, y) = \sum_x \sum_y p(x) p(y/x) d(x, y) \quad (5.25)$$

Distribuția sursei  $p(x)$  este fixă, în timp ce  $D_{p(y/x)}$  reprezintă distorsiunea medie pentru distribuția de probabilitate  $p(y/x)$ .

Rescriind ecuația (5.22), obținem

$$I_{p(y/x)} = \sum_x \sum_y p(x) p(y/x) \log \frac{p(y/x)}{p(y)} \quad (5.26)$$

în timp ce funcția rată-distorsiune  $R(D)$  este definită ca

$$R(D) = \min I_{p(y/x)}, p(y/x) \in p_D \quad (5.27)$$

Acesta este minimul informației mutuale medii pentru toate probabilitățile  $p(y/x)$  care conduc la o distorsiune care nu depășește pe  $D$ . Alternativ, se poate fixa valoarea informației mutuale și se pot determina valorile  $p(y/x)$  care minimizează distorsiunea. În acest caz  $R(D)$  ține locul entropiei sursei din teorema codării a lui Shannon, conducând la necesitatea ca, pentru a nu se depăși distorsiunea  $D$ , capacitatea canalului  $C$  să fie cel puțin egală cu  $R(D)$ . Aceasta înseamnă că  $R(D)$  este rata efectivă la care produce informații sursa.

Este important să se generalizeze conceptul de rată-distorsiune pentru a acoperi și cazul amplitudinilor continue. Nu pot fi aplicate direct ecuațiile de bază ale teoriei informației. Totuși, dacă considerăm mulțimea simbolurilor de intrare și ieșire ca fiind continuă, adică dacă are un număr infinit de membri, atunci entropia absolută definită prin ecuația (5.13) devine infinită. În locul acesteia se va defini entropia diferențială

$$h\{p(x)\} = - \int_{-\infty}^{\infty} p(x) \log p(x) dx \quad (5.28)$$

care nu mai este independentă de sistemul de coordonate.

Unii autori preferă să utilizeze o funcție numită entropie  $\epsilon$  a sursei de informație  $H_\epsilon$  în locul funcției rată-distorsiune. Problema unui inginer de sistem este de a optimiza sistemele de comunicație video digitale. Fiind dată o sursă, un utilizator și un criteriu de fidelitate sau o măsură a distorsiunii pentru calitatea imaginii, trebuie proiectat un codor și un decodor astfel încât capacitatea necesară a canalului să fie minimă. Sau, alternativ, fiind dată o sursă, un utilizator și un canal, trebuie să se proiecteze un codor și un decodor astfel încât calitatea semnalului care ajunge la utilizator să fie maximă. Cadrul matematic ideal pentru această problemă de optimizare este entropia  $\epsilon$  a lui Kolmogorov. Rezultatul principal al utilizării entropiei  $\epsilon$  constă în demonstrarea faptului că, pentru o sursă dată și un criteriu de fidelitate dat sau o măsură a distorsiunii dată, există o funcție numită entropie  $\epsilon$  a sursei de informație  $H_\epsilon$ , astfel încât putem transfera un semnal la utilizator cu o distorsiune cât de apropiată de măsura distorsiunii, atâta timp cât capacitatea canalului  $C$  este mai mare decât  $H_\epsilon$ , lucru imposibil dacă  $C < H_\epsilon$ . Pentru a calcula entropia  $\epsilon$ , trebuie să avem un model matematic realist al sursei, definit în termeni statistici. Fie variabila aleatoare  $\xi$  care ia valorile  $x_1, x_2, \dots, x_k, \dots$  și fie  $p_k$  probabilitatea ca  $\xi$  să fie  $x_k$ . Să notăm prin  $\xi'$  o variabilă aleatoare care ia doar aceeași valoare ca și  $\xi$ . Dacă este suficient să se reproducă  $\xi$  printr-o valoare  $\xi'$  pentru care distribuția de probabilitate condiționată  $P_{\xi\xi'}$  a variabilelor  $\xi$  și  $\xi'$  să satisfacă aceeași

condiție  $W$  asupra vecinătății lui  $\xi$  și  $\xi'$ , atunci "entropia- $W$ " începe să joace rolul entropiei  $H(\xi)$  unde

$$H_W(\xi) = \inf_{P_{\xi', \xi} \in W} I(\xi; \xi') \quad (5.29)$$

Aici,  $I(\xi; \xi')$  este cantitatea de informație referitoare la  $\xi$  conținută în  $\xi'$ . Conceptul de entropie- $W$ , absolut necesară pentru a putea lucra cu variabile distribuite continuu, pare să fie mai interesant pentru variabile discrete, deoarece pentru acestea există o condiție specifică  $W$ , care în cazul continuu nu va duce la micșorarea nedeterminării infinite a variabilei.

Să considerăm un sistem digital pentru transmiterea imaginilor, a cărui schemă bloc simplificată este prezentată în figura 5.2.



Fig. 5.2. Schema bloc a unui sistem digital de transmitere a imaginilor

Sursa de informație generează semnalul aleator  $\xi$  care este codat cu o precizie  $\epsilon$  ( $\epsilon > 0$ ). Semnalul aleator  $\xi'$  de la ieșirea codorului, precum și semnalul generat  $\xi$  aparțin aceluiași spațiu. Probabilitatea asociată lor aparține aceleiași clase de distribuție  $W_\epsilon$  care este determinată de  $\epsilon$ . Entropia lui  $\epsilon$ ,  $H_\epsilon(\xi)$ , este o măsură a informației mutuale a două semnale aleatoare, unul dintre ele dat, adică

$$H_\epsilon(\xi) = \inf_{P_{\xi', \xi} \in W_\epsilon} I(\xi; \xi') \quad (5.30)$$

Să presupunem acum că un semnal  $x$  provenit de la sursă este exprimat prin eșantioane de valori, precum și că eroarea medie pătratică a eșantioanelor de semnal nu depășește  $\epsilon^2$ . Atunci când  $x = (x_1, x_2, \dots, x_M)$  este o variabilă aleatoare  $M$ -dimensională, care are o funcție densitate  $p(x)$  suficient de netedă, entropia- $\epsilon$  a lui  $x$  este dată prin

$$H_\epsilon(x) = M \log_2 1/\epsilon + h\{p(x)\} - \frac{M}{2} \log_2 2\pi\epsilon \quad (5.31)$$

unde  $h\{p(x)\}$  este entropia diferențială a lui  $x$ .

O problemă care apare deseori în prelucrarea imaginilor este legată de numărul minim de biți per pixel care este necesar pentru a atinge o fidelitate prestabilită. Răspunsul la această întrebare este dat de entropia- $\epsilon$ .

Fie  $x$  o variabilă aleatoare unidimensională cu funcția densitate de probabilitate  $p(x)$  și dispersia  $\sigma_x^2$ .

Cele mai utile modele pentru caracterizarea semnalelor audio și video în schemele de codare sunt funcțiile de densitate de probabilitate Gaussiană

$$p_G(x) = \frac{1}{\sqrt{2\pi}\sigma_x} \exp\left(-\frac{x^2}{2\sigma_x^2}\right) \quad (5.32)$$

Laplaciană

$$p_L(x) = \frac{1}{\sqrt{2}\sigma_x} \exp\left(-\frac{\sqrt{2}}{\sigma_x}|x|\right) \quad (5.33)$$

și gamma

$$p_\gamma(x) = \frac{k}{2\Gamma(m)} (k|x|)^{m-1} \exp(-k|x|) \quad (5.34)$$

ale intrării cuantizorului. Densitățile Laplaciană și gamma sunt cele mai interesante, în timp ce densitatea Gaussiană este utilizată în principal pentru motive de comparație. În ecuația (5.34),  $\Gamma(\cdot)$  definește funcția gamma, parametrul  $m$  definește forma, valoarea  $m = 0,5$  producând cele mai potrivite date, în timp ce

$$k^2 = \frac{0.75}{\sigma_x^2} \quad (5.35)$$

Intrările cuantizorului în schemele de codare predictivă a imaginilor sunt specificate de o funcție densitate de probabilitate Laplaciană, în timp ce intrările pentru schemele de codare a vorbirii și pentru eșantioanele erorilor de predicție pot fi reprezentate printr-o densitate gamma. Pentru funcția densitate de probabilitate Gaussiană, entropia diferențială este

$$h \{p_G(x)\} = - \int_{-\infty}^{\infty} p_G(x) \log_2 p_G(x) dx = \frac{1}{2} \log_2 2\pi e \sigma_x^2 \quad (5.36)$$

Pentru densitatea Laplaciană, entropia diferențială este obținută similar

$$h \{p_L(x)\} = \frac{1}{2} \log_2 2e^2 \sigma_x^2 \quad (5.37)$$

în timp ce pentru modelul gamma obținem

$$h \{p_\gamma(x)\} = \frac{1}{2} \log_2 \frac{4e\pi\sigma_x^2}{3\lambda} \quad (5.38)$$

unde  $e = 2,718$ , în timp ce  $\lambda = 1,781$  este constanta lui Euler.

Entropia- $\epsilon$  corespunzătoare distribuției Gaussiene a lui  $x$  este obținută ținând cont de ecuațiile (5.31) și (5.36):

$$H_{G_\epsilon}(x) = \frac{1}{2} \log_2 \frac{\sigma_x^2}{\epsilon^2} \quad (5.39)$$

unde eroarea medie pătratică a eșantioanelor de semnal nu trebuie să depășească  $\epsilon^2$ .

Pentru distribuția Laplaciană a lui  $x$ , obținem din ecuațiile (5.31) și (5.37):

$$H_{L_\epsilon}(x) = \frac{1}{2} \log_2 \frac{\sigma_x^2}{\epsilon^2} - \frac{1}{2} \log_2 \frac{\pi}{e} \quad (5.40)$$

în timp ce din ecuațiile (5.31) și (5.38) obținem, pentru funcția de densitate de probabilitate gamma:

$$H_{\gamma_\epsilon}(x) = \frac{1}{2} \log_2 \frac{2\sigma_x^2}{3\lambda\epsilon^2} \quad (5.41)$$

Chiar dacă ecuația (5.39) este aplicabilă doar datelor Gaussiene, ea furnizează totuși o limită superioară pentru performanțele care pot fi atinse de orice sursă ne-Gaussiană.

Dacă  $x$  este o variabilă aleatoare Gaussiană cu dispersia  $\sigma^2$ , iar  $y$  este valoarea sa reprodusă, și dacă distorsiunea este măsurată prin valoarea medie pătratică a diferenței  $(x - y)$ , adică

$$D = E[(x - y)^2] \quad (5.42)$$

atunci funcția rată-distorsiune a lui  $x$  este

$$R(D) = \begin{cases} \frac{1}{2} \log_2 \frac{\sigma^2}{D}, & D \leq \sigma^2 \\ 0, & D > \sigma^2 \end{cases} = \max \left[ 0, \frac{1}{2} \log_2 \frac{\sigma^2}{D} \right] \quad (5.43)$$

unde  $\sigma^2$  este dispersia lui  $x$ . Acest rezultat este aplicat frecvent codării coeficienților transformatelor, care sunt foarte apropiați de distribuția Gaussiană. Rezultă că, așa cum distribuția Gaussiană este cea care are entropia diferențială maximă, valoarea lui  $R(D)$

pentru o sursă Gaussiană este limita superioară pentru sursele cu distribuție diferențială, dar cu dispersii egale.

### 5.5.3. Interpretarea spectrală a relației rată-distorsiune

Spectrul sursei Markov de ordinul întâi, cu coeficientul de corelare interelemente  $\rho$ , determinat prin

$$S(\omega) = \frac{1 - \rho^2}{1 + \rho^2 - 2\rho \cos \omega} \quad (5.44)$$

este un model foarte utilizat pentru a descrie proprietățile spațiale simple ale surselor de imagine [209]. Acest model poate fi utilizat pentru a determina rata teoretică a sursei. În primul rând trebuie calculate valorile maxime și minime ale funcției spectrale  $S(\omega)$ . Deoarece valorile maxime și minime ale lui  $\cos \omega$  sunt  $+1$  și  $-1$ , respectiv, vom obține

$$S(\omega)_{\max} = \frac{1 + \rho}{1 - \rho} \quad (5.45)$$

în timp ce

$$S(\omega)_{\min} = \frac{1 - \rho}{1 + \rho} \quad (5.46)$$

În cazul în care nivelul de distorsiune  $\theta < S(\omega)_{\min}$ , adică,  $\theta < S(\omega)$  pentru orice  $\omega$ , distorsiunea este

$$D(\theta) = \int_{-\pi}^{\pi} \theta d\omega = \theta \quad (5.47)$$

iar

$$R(D) = \frac{1}{4\pi} \int_{-\pi}^{\pi} \log \frac{S(\omega)}{D} \quad (5.48)$$

Efectuând calculele, putem scrie că

$$R(D) = \frac{1}{2} \left( \log \frac{1 - \rho^2}{D(1 + \rho^2)} - \log \frac{1 + \sqrt{1 - k^2}}{2} \right) \quad (5.49)$$

unde  $k = \frac{2\rho}{1 + \rho^2}$ . În final, prin integrare

$$\int_0^{\pi} \log(1 - k \cos \omega) = \pi \log \left( \frac{1 + \sqrt{1 - k^2}}{2} \right), \quad (5.50)$$

în regiunea observată, și vom obține [21], [209]:

$$R(D) = \frac{1}{2} \log \left[ \frac{1 - \rho^2}{D} \right] \quad (5.51)$$

În cazul general, unde  $S(\omega)_{\max} \geq \theta \geq S(\omega)_{\min}$ , vom avea

$$D(\theta) = \frac{1}{\pi} \int_0^{\pi} \min[\theta, S(\omega)] d\omega \quad (5.52)$$



Interpretarea spectrală a relației rată-distorsiune este prezentată în figura 5.3.

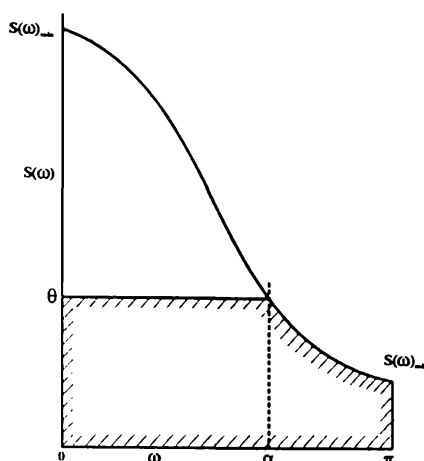


Fig. 5.3. Interpretarea spectrală a relației rată-distorsiune

$S(\omega)$  reprezintă densitatea de putere spectrală,  $\omega$  este frecvența normalizată,  $\theta$  este nivelul de distorsiune, în timp ce  $\alpha$  este frecvența corespunzătoare lui  $\theta$ . Domeniul spectral se divide în două părți. Astfel, vom avea

$$D(\theta) = \frac{1}{\pi} \int_0^{\alpha} \theta d\omega + \frac{1}{\pi} \int_{\alpha}^{\pi} S(\omega) d\omega \quad (5.53)$$

Vom reține toate componentele pentru care  $S(\omega) > \theta$ , implicând o distorsiune  $\theta$ , și vom omite restul de componente, care implică o distorsiune  $S(\omega)$ .

Relația rată-distorsiune pentru sursa Markov unidimensională de ordinul întâi, pentru diferiți coeficienți de corelație interelemente  $\rho$  este prezentată în figura 5.4. Liniile întrerupte reprezintă limitele de mică distorsiune. Trebuie observat că aceste curbe sunt de fapt drepte într-o bază logaritmică, până la limitele de mică distorsiune. Desigur, toate curbele trebuie să treacă prin punctul  $D/\sigma^2 = 1$ , unde  $R(D) = 0$ .

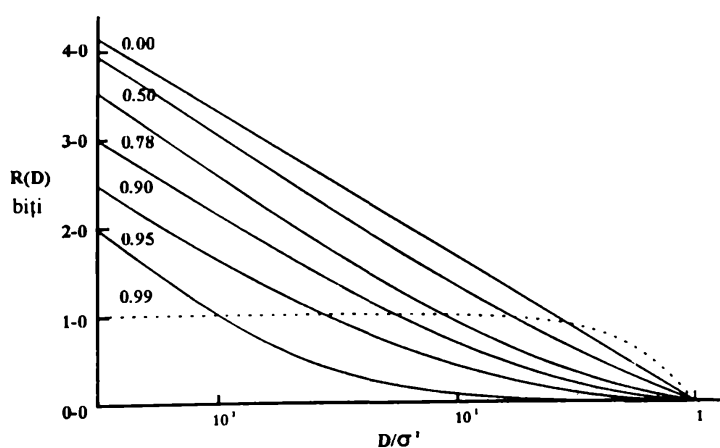


Fig. 5.4. Relația rată-distorsiune pentru surse Markov 1D de ordinul întâi, pentru diferiți coeficienți de corelație

#### 5.5.4. Reprezentarea spectrală bidimensională și rata pentru sursa Markov de primul ordin

Pentru datele de tip imagine, reprezentarea rată-distorsiune este bidimensională, trebuind totodată luată în considerație și interdependența suplimentară între elemente după direcțiile verticală și orizontală. Adică, reprezentarea spectrală generală unidimensională a funcției rată-distorsiune furnizează două rezultate apropiate de distorsiune și funcția rată-distorsiune, respectiv

$$D(\theta) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \min[\theta, S(\omega)] d\omega \quad (5.54)$$

$$R(D) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \max\left[0, \frac{1}{2} \log \frac{S(\omega)}{\theta}\right] d\omega \quad (5.55)$$

Aici,  $S(\omega)$  reprezintă densitatea puterii spectrale a sursei. Dacă dorim să utilizăm un criteriu de eroare medie pătratică ponderată WMSE (Weighted Mean Square Error), de exemplu în codarea psihovizuală, putem utiliza funcția de ponderare spectrală  $W(\omega)$  pentru a modifica  $S(\omega)$  [21], [209].

Extensia bidimensională este prezentată, pe scurt, în [21], o tratare mai detaliată putând fi găsită în [102]. Se obține

$$D(\omega) = \frac{1}{4\pi^2} \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} \min[\theta, S(\omega_h, \omega_v)] d\omega_h d\omega_v \quad (5.56)$$

$$R(D) = \frac{1}{4\pi^2} \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} \max\left[0, \frac{1}{2} \log \frac{S(\omega_h, \omega_v)}{\theta}\right] d\omega_h d\omega_v \quad (5.57)$$

unde  $S(\omega_h, \omega_v)$  este densitatea puterii spectrale bidimensionale. În general, prin multiplicarea cu funcția de ponderare spațială  $W(\omega_h, \omega_v)$  se obține răspunsul bidimensional. Determinarea lui  $R(D)$  și  $D(\theta)$  depinde de presupunerile făcute în legătură cu  $S(\omega_h, \omega_v)$ . Aici,  $\omega_h$  și  $\omega_v$  sunt frecvențele normalizate orizontală și verticală, respectiv.

Să presupunem, pentru început, că proprietățile de corelație ale imaginii sunt separabile. Funcția de covarianță pentru modelul Markov de ordinul întâi poate fi scrisă, în acest caz, sub forma

$$\psi(x, y) = \sigma^2 \rho_h^{|x|} \rho_v^{|y|} \quad (5.58)$$

unde  $\rho_h$  și  $\rho_v$  sunt coeficienții de corelație inter-elemente pe orizontală și verticală [21]. Pentru procesul cu dispersie unitară, spectrul bidimensional poate fi definit ca

$$S(\omega_h, \omega_v) = \frac{(1 - \rho_h^2)(1 - \rho_v^2)}{(1 + \rho_h^2 - 2\rho_h \cos \omega_h)(1 + \rho_v^2 - 2\rho_v \cos \omega_v)} \quad (5.59)$$

Acesta este produsul dintre distribuțiile spectrale orizontală și verticală. În regiunea de mică distorsiune  $\theta < S(\omega_h, \omega_v)_{\min}$  pentru toate valorile lui  $\omega_h, \omega_v$  și pentru o analiză unidimensională, se poate arăta că

$$D(\theta) = \theta$$

precum și că [21]

$$R(D) = \frac{1}{2} \log \frac{(1 - \rho_h^2)(1 - \rho_v^2)}{D} \quad (5.60)$$

În acest caz vom obține

$$S(\omega_h, \omega_v)_{\min} = \frac{(1 - \rho_h)(1 - \rho_v)}{(1 + \rho_h)(1 + \rho_v)} \quad (5.61)$$

Putem să deducem acum și avantajele prelucrării bidimensionale asupra prelucrării unidimensionale.

Ținând cont de ecuațiile (5.51) și (5.60), se poate trage concluzia că avantajul prelucrării bidimensionale poate fi exprimat de relația

$$R_{2D} = -\frac{1}{2} \log(1 - \rho_v^2) \quad (5.62)$$

Pentru o valoare practică a lui  $\rho_v = 0,95$  se obține 1,68 biți/element.

Cea de-a doua modalitate constă în considerarea spectrului de densitate de putere ca fiind izotrop [21], [73]. În acest caz, funcția covarianță are forma

$$\psi_i(x, y) = \sigma^2 \exp[-\alpha(x^2 + y^2)] \quad (5.63)$$

unde  $\alpha = -\ln \rho_h$ . Densitatea de putere spectrală bidimensională de dispersie unitară este

$$S(\omega_h, \omega_v) = \frac{2\pi\alpha}{(\alpha^2 + \omega_r^2)^{3/2}} \quad (5.64)$$

unde  $\omega_r^2 = \omega_h^2 + \omega_v^2$  și prezintă o simetrie circulară, definită de frecvența radială  $\omega_r$ . Pentru această reprezentare spectrală, regiunea care trebuie codată este conținută în interiorul unui cerc cu raza definită prin relația

$$S(\omega_h, \omega_v) = \theta \quad (5.65)$$

### 5.5.5. Măsura planității spectrale

Măsura planității spectrale este o consecință importantă a modelelor pentru semnale aleatoare. Pentru un semnal  $x(n)$  cu puterea spectrală  $S_{xx}(e^{j\omega})$ , măsura planității spectrale este definită prin [21]

$$\gamma_x^2 = \frac{\exp\left[\frac{1}{2\pi} \int_{-\pi}^{\pi} \ln S_{xx}(e^{j\omega}) d\omega\right]}{\frac{1}{2\pi} \int_{-\pi}^{\pi} S_{xx}(e^{j\omega}) d\omega} \quad (5.66)$$

Din această definiție rezultă că  $0 \leq \gamma_x^2 \leq 1$ .

Un semnal zgomot alb  $\omega(n)$  cu  $R_{\omega\omega}(n) = \sigma_\omega^2 \delta(n)$  are  $S_{\omega\omega}(e^{j\omega}) = \sigma_\omega^2$  ceea ce conduce la  $\gamma^2 = 1$ . Pe de altă parte, un semnal color  $x(n)$  va avea  $\gamma_x^2 < 1$ . Aceasta este justificarea utilizării denumirii de planitate spectrală. Cu cât măsura este mai mică, cu atât  $S_{xx}(e^{j\omega})$  va fi mai departe de a fi plat.

În abordarea rată-distorsiune, distorsiunea medie maxim admisibilă este condiționată, iar rata de transmisie cerută (prin urmare și capacitatea canalului) este minimizată. În abordarea distorsiune-rată, rata de transmisie este cea condiționată, posibil datorită constrângerilor impuse asupra canalului de către echipamentul de transmisie și nivelurile de cuantizare disponibile, iar distorsiunea medie rezultată este minimizată [21], [108]. Pentru surse de semnal staționare și ergodice, funcția distorsiune-rată  $D(R)$  specifică distorsiunea medie pătratică minimă  $D$  care ar trebui să fie atinsă la o rată de transmisie medie de  $R$  biți per variabilă aleatoare. De aceea, funcția distorsiune-rată furnizează o limită inferioară a fidelității care poate fi atinsă de orice codor pentru un canal de comunicație dat. După cum este de așteptat intuitiv,  $D(R)$  este o funcție monotonă necresătoare de  $R$ , cu  $D \rightarrow 0$ , atunci când  $R \rightarrow \infty$ . Inversa sa este funcția rată-distorsiune  $R(D)$ .

Pentru o sursă staționară și ergodică cu o funcție densitate de probabilitate Gaussiană și dispersie  $\sigma_x^2$ , funcția distorsiune-rată este dată de [21]:

$$D(R) = 2^{-2R} \gamma_x^2 \sigma_x^2 \quad (5.67)$$

Cunoașterea măsurii planității spectrale  $\gamma_x^2$  este utilă atunci când se proiectează un sistem de codare a semnalului, deoarece ea va da informații asupra rezervelor de îmbunătățire existente într-un sistem particular.

Funcția rată-distorsiune pentru o sursă staționară Gaussiană bidimensională discretă în timp  $x(m, n)$ , cu puterea spectrală  $S_{xx}(e^{j\omega_1}, e^{j\omega_2})$  este dată parametric prin:

$$R(\theta) = \frac{1}{4\pi^2} \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} \max\left[0, \frac{1}{2} \log_2 \left( \frac{S_{xx}(e^{j\omega_1}, e^{j\omega_2})}{\theta} \right)\right] d\omega_1 d\omega_2 \quad (5.68)$$

$$D(\theta) = \frac{1}{4\pi^2} \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} \min[\theta, S_{xx}(e^{j\omega_1}, e^{j\omega_2})] d\omega_1 d\omega_2 \quad (5.69)$$

Aici,  $R(\theta)$  și  $D(\theta)$  sunt rata de transmisie și distorsiunea per pixel a unui semnal codat optimal, ca funcție de un parametru de codare, adică de nivelul de distorsiune  $\theta$ .  $R(\theta)$  reprezintă cea mai scăzută rată de transmisie care poate fi obținută de orice metodă de codare cu distorsiunea  $D(\theta)$ . Parametrul  $\theta$  acționează ca un prag, după cum se vede în figura 5.5. Adică, toate componentele de frecvență cu  $S_{xx}(e^{j\omega_1}, e^{j\omega_2}) > \theta$  sunt codate și transmise la rata  $R(\theta)$ , în timp ce componentele de frecvență cu  $S_{xx}(e^{j\omega_1}, e^{j\omega_2}) < \theta$  nu sunt transmise. Distorsiunea rezultată, prezentată hașurat,  $D(\theta)$  constă din două părți: (a) zgomotul din interiorul benzii, care constă din zgomotul de codare rezultat din frecvențele codate, și (b) zgomotul din afara benzii, rezultat din frecvențele care nu sunt transmise.

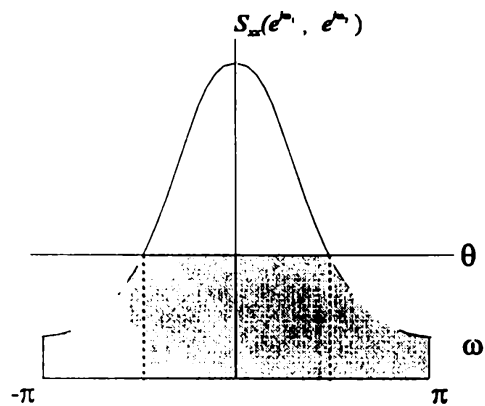


Fig. 5.5. Spectrul de putere pentru o sursă Gaussiană staționară discretă în timp

În acest fel, spectrul erorii unei surse Gaussiene codate optimal este de forma:

$$S_{qq}(e^{j\omega_1}, e^{j\omega_2}) = \begin{cases} \theta, & \text{pt. } S_{xx}(e^{j\omega_1}, e^{j\omega_2}) \\ S_{xx}(e^{j\omega_1}, e^{j\omega_2}), & \text{pt. } S_{xx}(e^{j\omega_1}, e^{j\omega_2}) \end{cases} \quad (5.70)$$

Utilizând această definiție, ecuația (5.69) poate fi rescrisă sub forma:

$$D(\theta) = \frac{1}{4\pi^2} \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} S_{qq}(e^{j\omega_1}, e^{j\omega_2}) d\omega_1 d\omega_2 \quad (5.71)$$

Dacă un spectru de eroare nu poate fi descris prin ecuația (5.70), rezultă că sursa corespunzătoare nu a fost codată optimal. Forma spectrului de putere poate fi descrisă compact prin măsura planității spectrale

$$\gamma_x^2 = \frac{\exp\left\{\frac{1}{4\pi^2} \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} \ln[S_{xx}(e^{j\omega_1}, e^{j\omega_2})] d\omega_1 d\omega_2\right\}}{\frac{1}{4\pi^2} \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} S_{xx}(e^{j\omega_1}, e^{j\omega_2}) d\omega_1 d\omega_2} \quad (5.72)$$

Dacă  $\gamma_x^2 = 1$ , atunci spectrul de putere  $S_{xx}(e^{j\omega_1}, e^{j\omega_2})$  este plat, iar sursa este necorelată. Pentru  $\gamma_x^2 < 1$ , semnalul este corelat și poate fi codat cu un număr mai mic de biți la o distorsiune dată, în comparație cu semnalul necorelat. Pe de altă parte, atunci când pragul de codare  $\theta \leq \min S_{xx}(e^{j\omega_1}, e^{j\omega_2})$ , adică atunci când în spectrul erorii nu este prezent zgomot în afara benzii, ecuațiile (5.68) și (5.69) pot fi introduse în funcția distorsiune-rată exprimată prin ecuația (5.67), unde  $\sigma_x^2$  reprezintă dispersia sursei.

### 5.5.6. Concluzii

Pentru a caracteriza funcția  $R(D)$ , este necesar un model pentru sursă și un criteriu de distorsiune. Problema devine tratabilă matematic dacă sursa este modelată ca o sursă DMS, iar măsura distorsiunii este distorsiunea unui mesaj singular. Aceasta implică faptul că distorsiunea este funcție de simbolul original și de valoarea sa reprodusă, ea nedepinzând de ceilalți termeni din secvența de simboluri sursă și de reproducerea acestora. În aceste condiții, au fost determinate soluții pentru surse cu diferite distribuții, utilizând criteriile de distorsiune pornind de la eroarea medie pătratică. Pentru acele distribuții unde o soluție nu există, se poate determina o soluție numerică prin utilizarea unui algoritm iterativ sugerat în [21]. Din păcate, situațiile concrete de prelucrare a imaginilor sunt caracterizate printr-o puternică corelație, neputând fi modelate adecvat ca surse DMS. De asemenea, în cele mai multe aplicații, imaginea comprimată este percepută de un subiect uman și, pentru fiecare pixel, distorsiunea percepută este o funcție complexă de valoarea aceluși pixel și a valorilor pixelilor înconjurători. Aceasta limitează utilitatea măsurii distorsiunii unui mesaj singular. Mai mult, chiar în cazurile cele mai simple, implementarea schemei optimale de codare care realizează limita teoretică poate fi foarte complexă. Deși rezultatele teoretice disponibile sunt limitate pentru sursele cu memorie, un caz important pentru care există o soluție este cel al unei surse Gausiene cu o măsură a distorsiunii dată de eroarea medie quadratică. Prin modelarea unei imagini ca o sursă bidimensională Gauss-Markov cu un coeficient de corelație apropiat de unu, teoria rată-distorsiune poate fi utilizată pentru stabilirea unei referințe pentru performanțe. În sfârșit, este important să se observe că determinarea funcției rată-distorsiune  $R(D)$  pentru sursele care descriu situațiile reale de prelucrare a imaginilor rămâne un foarte activ domeniu de cercetare.

## Capitolul 6

### Implementarea algoritmului propus pentru compresia imaginilor

#### 6.1. Introducere

În acest capitol voi prezenta implementarea algoritmilor pe care i-am propus pentru compresia imaginilor prin utilizarea metodelor morfologiei matematice.

Mai întâi am implementat modulele de prelucrare morfologică de bază, necesare pentru elaborarea aplicațiilor mai complexe, cum sunt cele de compresia imaginilor. Printre aceste module se numără operatorii de dilatare, erodare, deschidere și închidere, pe care i-am implementat atât în variantă binară cât și în variantă pe scară de gri.

În continuare voi descrie algoritmul pe care l-am propus pentru compresia imaginilor pe scară de gri.

Pentru a simplifica implementarea și a obține în același timp un bun raport de compresie prin utilizarea metodelor morfologiei matematice, în algoritmul propus am utilizat o descompunere pe plane de biți. Am codat apoi diferitele plane prin metode diferite, pentru a obține o eficiență mai mare.

Cele mai semnificative 4 plane de biți sunt codate fără pierderi, deoarece ele conțin cea mai importantă informație cuprinsă în imaginea respectivă. Am testat două posibilități de codare a acestor plane și am efectuat o comparație, care este prezentată în paragraful 6.5. Prima posibilitate constă în utilizarea codării conturilor cu ajutorul codurilor lanț, iar cea de-a doua constă în utilizarea reprezentării unei imagini prin scheletul său. Am constatat că utilizarea scheletelor poate conduce la rezultate mai bune pe o clasă mai largă de imagini. Am utilizat o metodă similară cu cea propusă de Maragos și Schafer pentru aplicații binare [130], dar am utilizat un element structurant cu creștere exponențială a dimensiunii. Pentru minimizarea setului de schelete rezultate am utilizat aceeași metodă pe care au propus-o Maragos și Schafer. Reuniunea seturilor de schelete obținute a fost minimizată în continuare prin așa numita *minimizare a scheletelor condiționată de contururi*. Pentru codarea în vederea transmiterii setului de 4 schelete rezultate în urma minimizării, am utilizat o codare cu lungime variabilă.

Pentru ultimele 4 plane, mai puțin semnificative, am propus un algoritm de codare diferit. Ținând cont de faptul că aceste plane poartă puțină informație semnificativă, am dorit să obțin pentru aceste plane un foarte bun raport de compresie. Pentru planul 0 am determinat o densitate de probabilitate pentru apariția pixelilor albi/negri pe suprafața întregului plan, urmând ca acesta să fie singurul parametru transmis pentru caracterizarea planului. În cursul procesului de decompresie a imaginii, se generează local o imagine aleatoare cu aceeași densitate de probabilitate, care este apoi adăugată imaginii reconstruite. Pentru planele 1, 2 și 3, nefiind de așteptat o densitate de probabilitate cuasi-constantă pe toată suprafața planului, am efectuat o segmentare a acestor plane în blocuri de dimensiuni 1/4, 1/8, respectiv 1/16. Am calculat apoi același parametru ca și în cazul planului 0 pentru fiecare bloc, acești parametri fiind transmiși pentru caracterizarea planului.

Rezultatele obținute după implementare pentru diferite clase de imagini sunt prezentate în capitolul 7.

## 6.2. Implementarea operatorilor morfologici de bază

În primele capitole ale acestei teze au fost deja demonstrate importanța și eficiența metodelor morfologice în analiza imaginilor. Pentru a studia utilitatea acestor metode în aplicații de compresia imaginilor, a fost necesar ca mai întâi să realizez implementarea unor module morfologice de bază (în primul rând a operatorilor morfologici elementari), pe care să le pot utiliza ulterior în realizarea unor algoritmi mai complexi cum sunt cei de compresia imaginilor.

Operațiunile morfologice de bază sunt erodarea, în care pixelii care se suprapun pe o anumită grilă sunt îndepărtați din imagine și dilatarea, în care unei forme date i se adaugă o arie mică de aproximativ un pixel de-a lungul conturului.

Acești operatori de bază sunt elementul central al metodelor pe care le-am propus cu scopul de a dovedi că morfologia matematică poate fi utilă și în compresia imaginilor. Am implementat toți operatorii morfologici de bază sub forma unor module software realizate în limbaj C.

### 6.2.1. Dilatarea

Implementarea generală a dilatării imaginilor binare constă în parcurgerea a două etape:

- un program care creează o imagine dilatată pornind de la o imagine de intrare dată și de la un element structurant dat;
- o funcție care realizează același obiectiv, dar care poate fi apelată dintr-o altă funcție, care permite ca dilatarea să fie încorporată în programe mai complexe de prelucrarea imaginilor.

Programul a fost denumit **bindil** (de la *binary dilation*). El acceptă trei argumente într-o linie de comandă:

**bindil input\_file SE\_file output\_file**

Toate cele trei argumente sunt nume de fișiere: **input\_file** reprezintă numele fișierului de imagine care conține imaginea care urmează să fie dilatată, **SE\_file** conține datele pentru elementul structurant, iar **output\_file** este numele fișierului care va fi creat pentru a memora imaginea dilatată rezultată.

Deoarece dilatarea este comutativă, elementul structurant poate fi considerat ca fiind, la rândul său, o imagine. O mică cantitate de informație trebuie adăugată la fișierul inițial care descrie elementul structurant, pentru a se defini originea acestuia. De exemplu, elementul structurant din figura 6.1 este reprezentat sub forma:

```
#origin 1 1
3 3
0 1 0
1 1 1
0 1 0
```

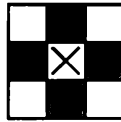


Fig. 6.1. Element structurant

Programul **bindil** realizează citirea imaginilor de intrare și apelarea funcției **bin\_dilate**, definită ca:

**IMAGE bin\_dilate (IMAGE im, IMAGE p);**

Funcția implementează dilatarea unei imagini de intrare **im** cu un element structurant reprezentat prin **p**, prin translatarea originii lui **p** în fiecare din pozițiile pixelilor negri ai imaginii **im** și prin copierea pixelilor negri din **p** în pozițiile corespunzătoare din imaginea de ieșire. Figura 6.2 ilustrează acest proces, care este cel descris matematic prin relația (4.6).

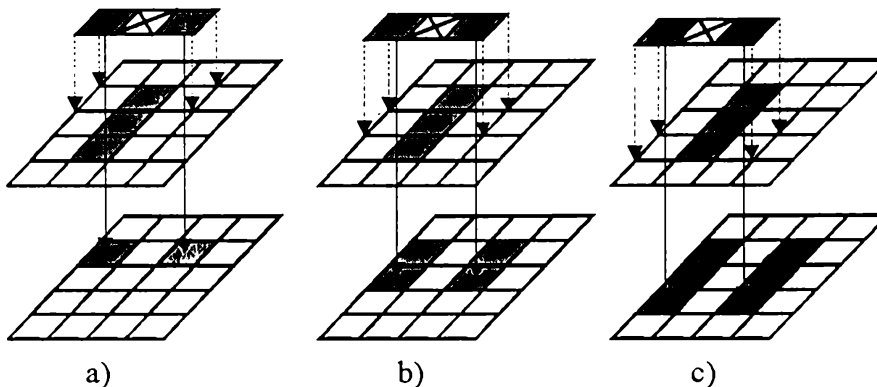


Fig. 6.2. Implementarea dilatării

Figura 6.2.a) ilustrează primul pas al algoritmului de dilatare, în care originea elementului structurant este plasată pe poziția primului pixel negru din imagine, iar pixelii elementului structurant sunt copiați pe pozițiile lor corespunzătoare din imaginea rezultată. Figura 6.2.b) prezintă elementul structurant plasat pe poziția următorului pixel negru, iar procesul se repetă. Figura 6.2.c) prezintă rezultatul final al procedurii de dilatare.

### 6.2.2. Erodarea

Dacă dilatarea poate fi văzută ca un proces care adaugă pixeli unei imagini, făcând-o mai mare, erodarea face imaginea mai mică. În cazul cel mai simplu, o erodare binară va elimina stratul exterior de pixeli care formează un obiect. Dacă elementul structurant este o arie de  $3 \times 3$  pixeli negri, atunci erodarea poate fi implementată simplu prin marcarea tuturor pixelilor negri care au cel puțin un vecin alb, și prin setarea la valoarea de alb a tuturor pixelilor marcați.

După cum am procedat și anterior în cazul dilatării, implementarea erodării constă într-un program care crează o imagine erodată pornind de la o imagine de intrare dată, un element structurant dat și o funcție care realizează prelucrarea.



Programul este similar cu **bindil** și poartă numele **binerode** (de la binary erosion):

**binerode input\_file SE\_file output\_file**

Funcția care realizează erodarea este numită **bin\_erode** și este implementată într-o manieră similară cu **bin\_dilate**. Procesul de erodare este ilustrat în figura 6.3.

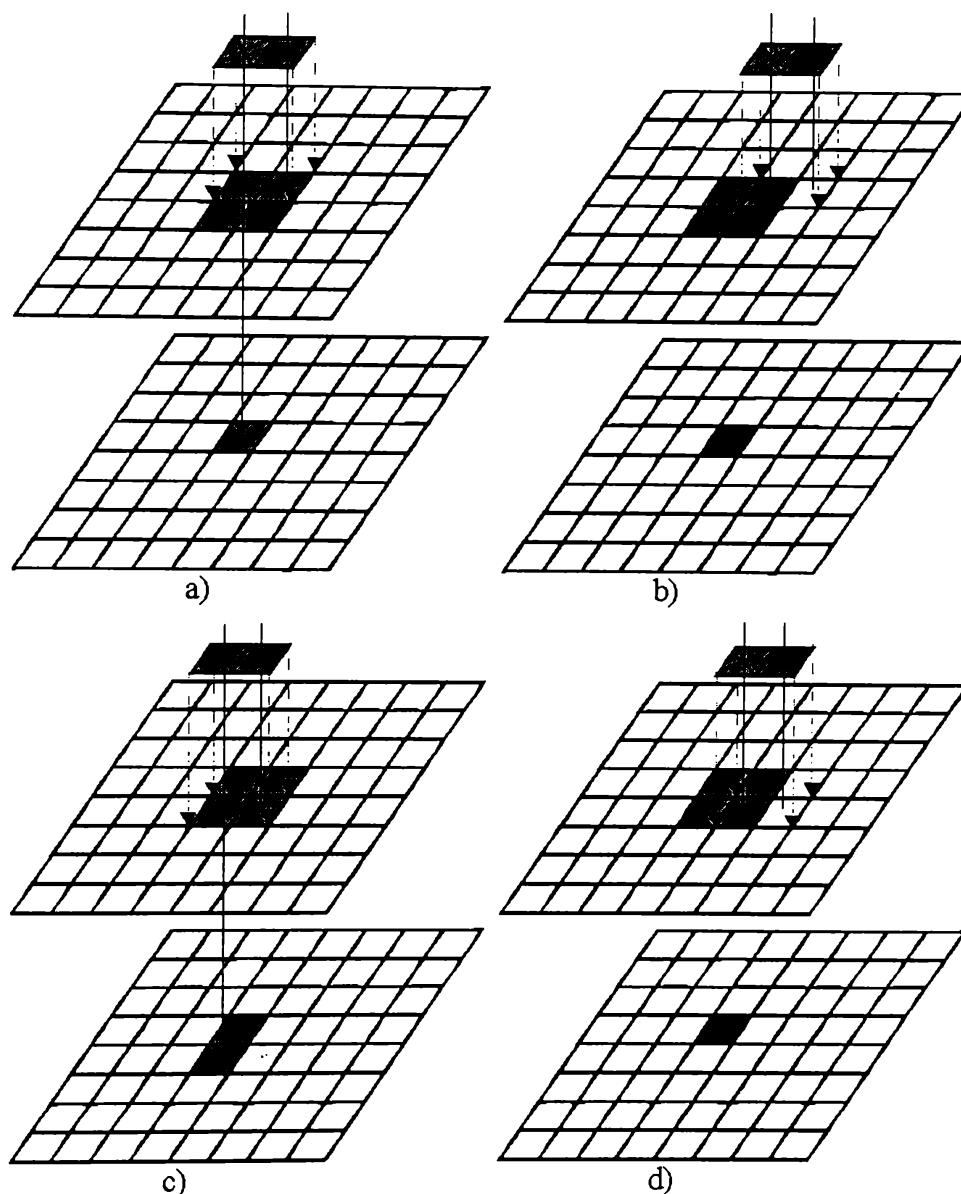


Fig. 6.3. Implementarea erodării

În figura 6.3.a), elementul structurant este translatat pe poziția unui pixel negru din imagine. În acest caz, toate componentele elementului structurant corespund unor pixeli negri din imagine, astfel încât rezultatul este un pixel negru. În figura 6.3.b), elementul structurant este translatat pe poziția următorului pixel negru din imagine, existând un singur pixel care nu se suprapune. Rezultatul este un pixel alb. Figura 6.3.c) prezintă următoarea translație, pentru care imaginile se suprapun, astfel încât, din nou, pixelul din imaginea de ieșire care corespunde originii translate a elementului

structurant este setat pe valoarea de negru. Ultima translație este ilustrată în figura 6.3.d), care nu corespunde unei suprapunerii, astfel încât pixelul rezultat în imaginea de ieșire este alb. Restul de pixeli din imaginea originală sunt albi, astfel încât nu au cum să se suprapună ca nivel de negru cu originea elementului structurant, prin urmare nu este necesar ca ei să fie luați în considerație pentru efectuarea unei translații.

### 6.2.3. Deschiderea și închiderea

După cum rezultă din ecuațiile (4.25) și (4.26), deschiderea și închiderea pot fi ușor implementate printr-o serie de dilatări și erodări, deja descrise în paragraful anterior.

Programele pe care le-am realizat sunt similare cu **bindil** și poartă numele **open**, respectiv **close**:

```
open input_file SE_file
close input_file SE_file
```

Aceste programe utilizează funcțiile anterior menționate **bin\_erode** și **bin\_dilate**.

Este important de observat că, pentru a realiza netezirea unor obiecte dintr-o imagine, poate fi utilizată operațiunea morfologică de închidere. Totuși, poate să fie necesară utilizarea mai multor elemente structurante, deoarece elementul structurant simplu este util doar pentru înlăturarea sau netezirea neregularităților datorate unor pixeli singulari. O altă posibilitate constă în aplicarea repetată a dilatării, urmate de același număr de erodări. Aplicarea a  $N$  dilatări/erodări conduce la netezirea neregularităților cu dimensiunea de  $N$  pixeli.

În practică, cel mai utilizat element structurant pentru efectuarea operațiunilor de deschidere și închidere este elementul simplu (pătrat de  $3 \times 3$  pixeli). Cea mai uzuală modalitate de efectuare a unei deschideri cu dimensiunea  $N$  constă în realizarea a  $N$  erodări binare succesive, urmate de  $N$  dilatări binare. Aceasta înseamnă că, pentru calcularea tuturor deschiderilor unei imagini până la dimensiunea zece, este necesară efectuarea a 110 erodări și dilatări. Dacă erodarea și dilatarea sunt implementate în varianta de bază, aceasta înseamnă că sunt necesare 220 de translații pe suprafața imaginii. O alternativă constă în salvarea fiecăreia din cele zece erodări ale imaginii originale. Fiecare dintre ele este ulterior supusă unui proces de dilatare printr-un număr adecvat de iterații, pentru a obține cele zece deschideri ale imaginii. Din păcate, cantitatea de memorie necesară pentru salvarea tuturor acestor variante poate fi foarte mare, limitând astfel aplicabilitatea ei.

Am implementat și *deschiderea de ordinul  $n$* , respectiv *închiderea de ordinul  $n$* . În cele ce urmează voi da un exemplu de cod sursă C pentru realizarea deschiderii de ordinul  $n$ :

```
#include <stdio.h>
#include "morph.h" //header file for
                  morphological operators
main(int argc, char *argv[])
{
  int i,n;
  IMAGE A,B,C ;
  scanf("%d",&n); //read the number of
                  dilation-erosions
  A=Input(argv[1]); //read input image
```

```

B=Input(argv[2]); //read structuring element
for(i=0;i<n;i++)
    binerode(A,B,C); //erode n times
for(i=0;i<n;i++)
    bindil(A,B,C); //dilate n times
Output(C,argv[3]); //write output image file
}

```

Închiderea de ordinul  $n$  este implementată similar.

#### 6.2.4. Operatori morfologici pe scară de gri

Utilizarea nivelurilor de gri multiple introduce mai multe complicații, atât din punct de vedere conceptual cât și din punct de vedere al complexității calculului. În acest caz, un pixel poate avea practic orice valoare întreagă, imaginea nemaiputând fi considerată ca o mulțime.

Să considerăm imaginea liniei prezentate în figura 6.4.a). Aceasta este o imagine cu două niveluri, putând fi calculată o dilatare a sa cu un element structurant de tip pătrat (așa cum se vede în figura 6.4.b). Să considerăm acum că în loc de a avea doar nivelurile "0" și "1", pixelii care formează linia au valoarea 20, iar fundalul este de nivel "0". Problema care se pune este de a vedea cum se va manifesta în imaginea finală dilatarea pe nivel de gri, în comparație cu dilatarea binară. Dacă, în cazul dilatării binare, linia este îngroșată devenind de lățime trei pixeli în loc de unul singur, în cazul morfologiei pe scară de gri ar trebui să avem o aparență asemănătoare a liniei după dilatare. Diferența dintre pixelii planului principal și cei ai fundalului trebuie să fie aproximativ aceeași ca și în imaginea originală, în timp ce linia trebuie să devină lată de aproximativ trei pixeli. În figura 6.4.d) este prezentat un exemplu privitor la cum ar trebui să apară linia dilatată pe scară de gri, în comparație cu cea originală, prezentată în figura 6.4.c).

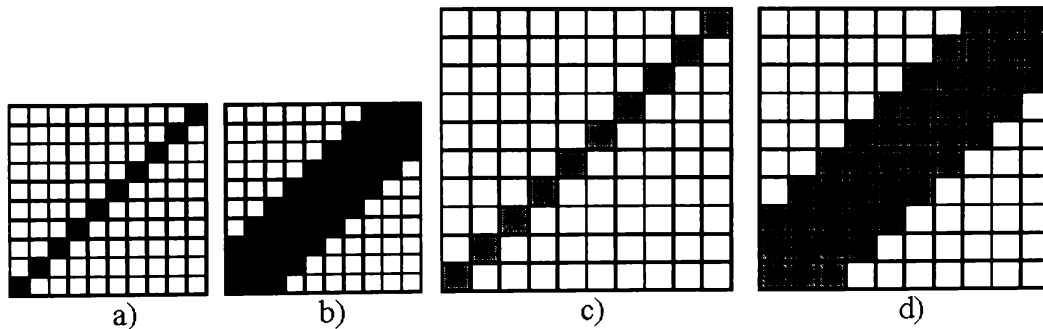


Fig. 6.4. Exemplu de dilatare pe scară de gri

Considerând dilatarea pe scară de gri definită prin

$$(f \oplus b)(s, t) = \max \{ f(s-x, t-y) + b(x, y) \mid (s-x, t-y) \in D_f; (x, y) \in D_b \} \quad (6.1)$$

ea poate fi calculată conform următorului algoritm:

1. Se poziționează originea elementului structurant peste primul pixel al imaginii care trebuie dilată.
2. Se calculează suma valorilor pentru fiecare pereche de pixeli corespunzători din elementul structurant și din imagine.

3. Se determină valoarea maximă a tuturor acestor sume și se alocă această valoare pixelului corespunzător din imaginea de ieșire.

4. Se repetă acest proces pentru fiecare pixel din imaginea care trebuie dilatată.

Valorile pixelilor din elementul structurant sunt și ele valori pe scară de gri, putând fi chiar negative. Deoarece pixelii cu valori negative nu pot fi afișați, există două modalități de lucru cu pixelii negativi din rezultat: ei pot fi toți setați la valoarea zero (depășire negativă), sau se translatează valorile pixelilor pentru toată imaginea în așa fel încât cea mai mică valoare să devină zero, iar restul pixelilor să aibă aceeași valoare relativă unul față de altul ca și anterior. Metoda cea mai simplă (și mai des utilizată) este prima.

Erodarea pe scară de gri poate fi definită în mod similar cu relația (6.1), dar, în acest caz, se ia în considerație valoarea minimă a sumelor perechilor de pixeli corespunzători din elementul structurant și imagine.

Programele C pe care le-am realizat pentru a realiza dilatarea, respectiv erodarea pe scară de gri sunt:

```
dilg input_gray_image SE_gray_image output_gray_image;
erog input_gray_image SE_gray_image output_gray_image;
```

Figura 6.5. prezintă imaginea unei legături de chei (fig. 6.5.a) după o dilatare pe scară de gri (figura 6.5.b, prin utilizarea programului **dilg**), respectiv după o erodare pe scară de gri (figura 6.5.c, prin utilizarea programului **erog**). Elementul structurant este un pătrat de 3×3 pixeli, toți pixelii având valoarea 1.

Se poate observa că dilatarea face ca orificiile din partea de sus a cheilor să devină mai mici, în timp ce erodarea le face mai mari, exact ca și în cazul binar.

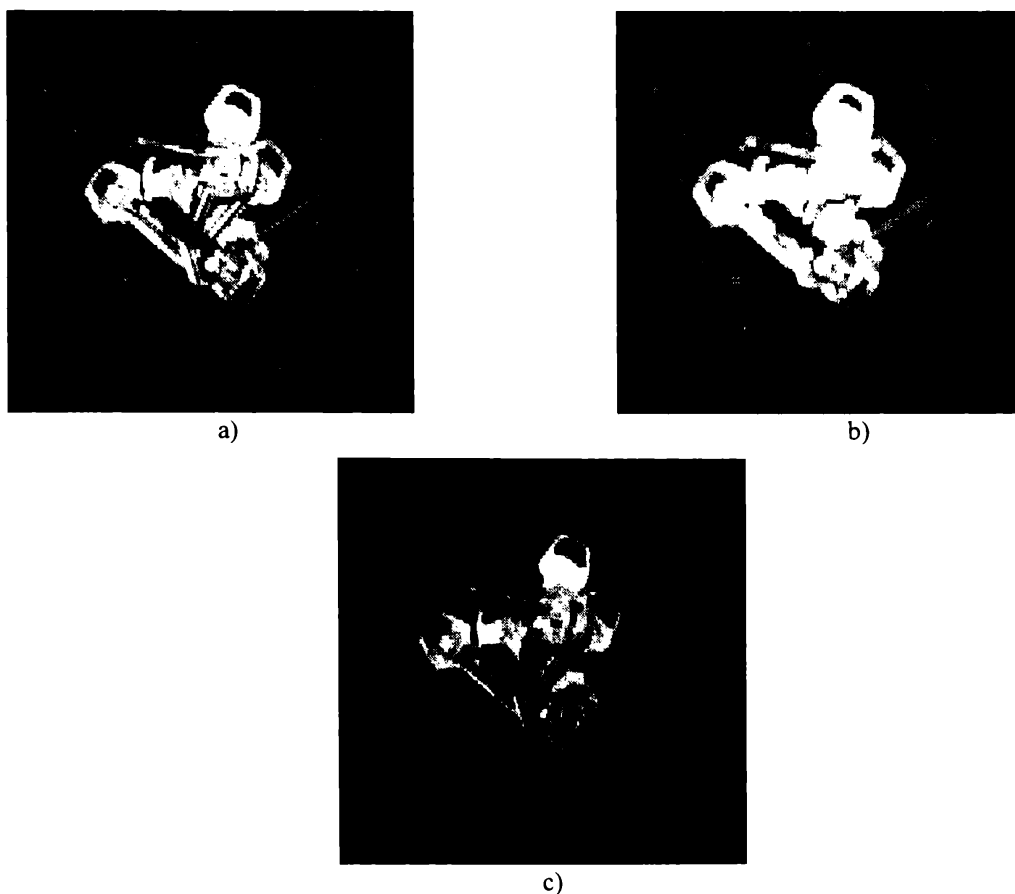


Fig. 6.5. Dilatarea și erodarea pe scară de gri pentru imaginea “legătură de chei”

Deschiderea și închiderea unei imagini pe scară de gri se fac în același mod ca și în cazul binar, dar prin utilizarea operatorilor de erodare și dilatare pe scară de gri.

Programele C pe care le-am realizat sunt **gopen** (de la gray-scale opening) pentru deschiderea pe scară de gri, respectiv **gclose** (de la gray-scale closing) pentru închiderea pe scară de gri. Rezultatele pentru imaginea de intrare “legătură de chei” (figura 6.6.a) sunt prezentate în figura 6.6.b) (deschidere cu element structurant sferic), figura 6.6.c) (închidere cu element structurant sferic) și 6.6.d) (închidere cu element structurant pătrat de 3×3 pixeli).

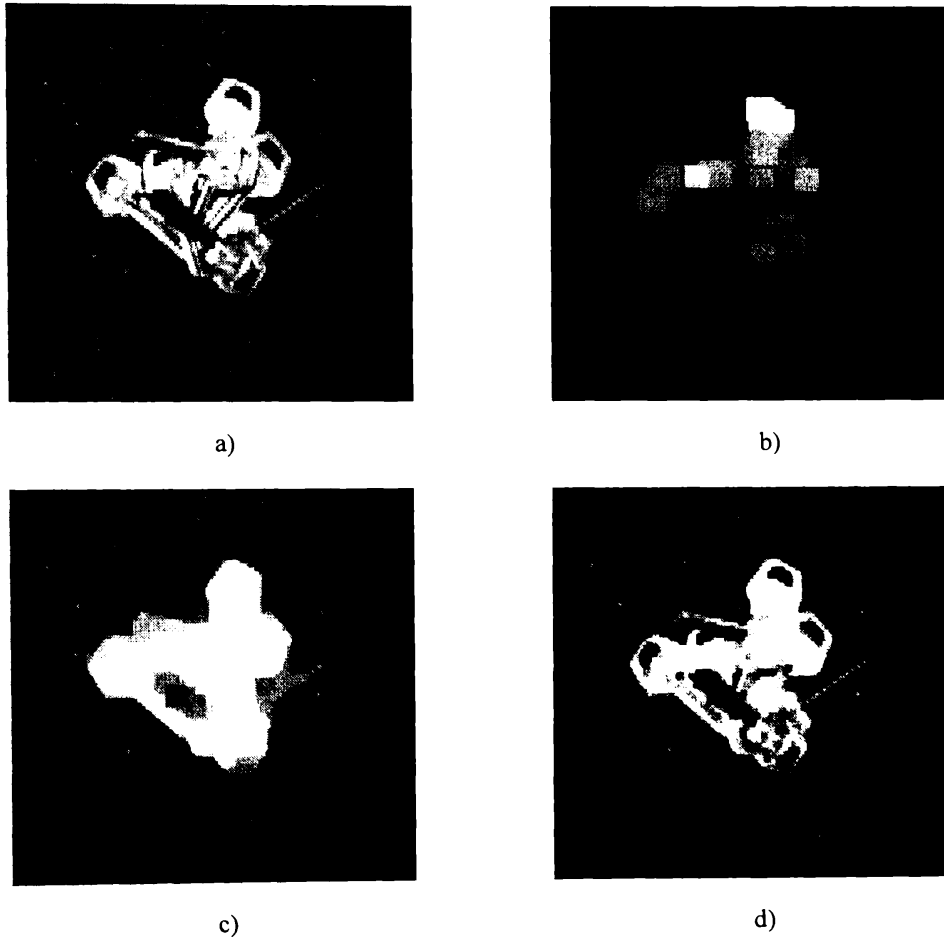


Fig. 6.6. Exemple de deschidere și închidere pe scară de gri cu diferite elemente structurante

Elementul structurant sferic utilizat în exemplul anterior este tot o imagine pe scară de gri, după cum urmează:

```

0 0 0 0 0 0 0 0 0 0 0
0 0 1 1 1 2 1 1 1 0 0
0 1 1 2 2 3 2 2 1 1 0
0 1 2 3 3 4 3 3 2 1 0
0 1 2 3 5 5 5 3 2 1 0
0 2 3 4 5 6 5 4 3 2 0
0 1 2 3 5 5 5 3 2 1 0
0 1 2 3 3 4 3 3 2 1 0
0 1 1 2 2 3 2 2 1 1 0
0 0 1 1 1 2 1 1 1 0 0
0 0 0 0 0 0 0 0 0 0 0

```

### 6.3. Descompunerea pe plane de biți a imaginilor pe scară de gri

Să considerăm o imagine cu dimensiunea de  $N \times M$  pixeli, în care fiecare valoare de pixel este reprezentată prin  $k$  biți. Prin selectarea unui singur bit situat pe aceeași poziție în fiecare pixel, se poate forma o imagine binară de dimensiune  $N \times M$ , numită *plan de biți* [181]. De exemplu, se poate selecta bitul cel mai semnificativ din fiecare valoare de pixel pentru a genera o imagine binară de dimensiune  $N \times M$  care reprezintă cel mai semnificativ plan de biți. Prin repetarea acestui proces pentru celelalte poziții de bit, imaginea originală poate fi descompusă într-un set de  $k$  plane de biți, de dimensiune  $N \times M$ , numerotate de la 0 pentru cel mai puțin semnificativ plan de biți până la  $k-1$  pentru cel mai semnificativ plan de biți.

Dacă considerăm, de exemplu, o imagine cu  $k = 8$  (256 niveluri de gri), o valoare de pixel  $x[i, j]$  poate fi scrisă ca:

$$x[i, j] = a_0[i, j] \cdot 2^0 + a_1[i, j] \cdot 2^1 + \dots + a_7[i, j] \cdot 2^7 \quad (6.2)$$

unde  $a_n[i, j] \in \{0, 1\}$ ,  $n \in \{0, 1, \dots, 7\}$  reprezintă cel de-al  $n$ -lea plan de biți.

Motivația pentru o astfel de descompunere constă în faptul că fiecare plan de biți poate fi codat de data aceasta într-un mod mai eficient prin utilizarea tehnicilor de compresie binară. Deoarece, în general, planele de biți cele mai semnificative conțin cea mai mare parte a informației structurale majore și pot fi compresate foarte puternic, poate fi utilizată o tehnică de reconstrucție progresivă a imaginii pornind de la planele de biți.

Am utilizat această proprietate a descompunerii pe plane de biți în contextul morfologiei matematice binare, pentru a reduce semnificativ redundanța existentă în planele de biți cele mai semnificative, simultan cu asigurarea reconstrucției fără pierderi a imaginii originale.

Algoritmii de codare a planelor de biți codează aceste plane, în mod tipic, independent, ținând cont de avantajul existenței unor arii uniforme de suprafață relativ mare în fiecare plan, ceea ce conduce la rapoarte de compresie mari. Având în vedere acest lucru, este de dorit ca planele de biți să fie formate în așa fel încât complexitatea lor să fie minimizată. Din păcate, construirea planelor de biți pornind de la reprezentarea binară convențională a valorilor pixelilor nu realizează acest scop.

Pentru a demonstra acest lucru, să considerăm o imagine pe 8 biți, ai cărei pixeli iau valori fluctuante aleator între codurile 127 și 128. Deși entropia unei astfel de imagini este de doar 1 bit/pixel, codarea independentă a planelor sale de biți necesită utilizarea în total a 8 biți/pixel. Acest lucru se datorează faptului că reprezentările binare pentru valorile întregi 127 și 128 sunt 0111111 și, respectiv, 1000000. Deoarece cele două reprezentări diferă pe fiecare poziție de bit, planelor de biți rezultate le lipsește coerența, fiind necesară găsirea unei metode care să le mărească coerența. O soluție simplă constă în formarea planelor de biți pe baza reprezentării valorilor pixelilor cu ajutorul codului Gray.

Codul Gray se caracterizează prin faptul că realizează un alfabet binar astfel încât valorile numerice succesive diferă între ele printr-un singur bit al reprezentării. Astfel, atunci când doi pixeli vecini diferă doar printr-o unitate de cod, un singur plan de biți este afectat. Figura 6.7 prezintă reprezentările pe 4 biți a numerelor de la 0 la 15 în cod binar, comparativ cu codul Gray. Valorile de bit 0 și 1 sunt reprezentate prin arii albe și negre, respectiv.

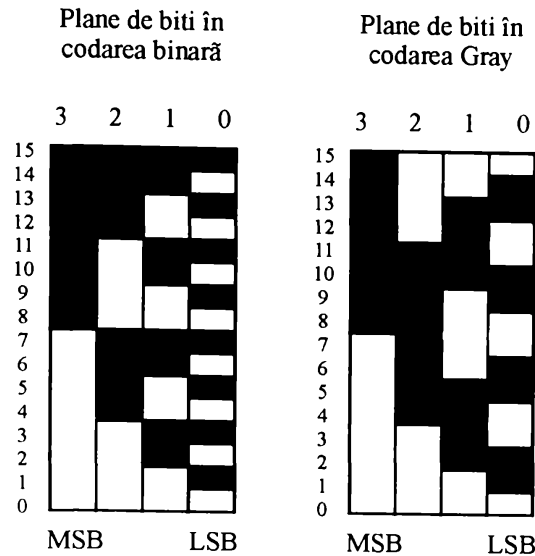


Fig. 6.7. Codurile binar și Gray pe 4 biți

Algoritmul de conversie din cod binar în cod Gray este definit de următoarele reguli:

1. Începând cu bitul cel mai semnificativ (MSB) al reprezentării în cod binar, toți biții "0" sunt lăsați nemodificați până ce se întâlnește un "1".
2. Primul "1" este lăsat nemodificat, dar toți biții care urmează sunt complementați până ce se întâlnește un "0".
3. Bitul "0" este complementat, dar toți biții care urmează sunt lăsați nemodificați până ce se întâlnește un nou "1".
4. Se revine la pasul 2.

Conversia inversă se face după următoarele reguli:

1. Începând cu bitul cel mai semnificativ (MSB) al reprezentării în cod Gray, toți biții "0" sunt lăsați nemodificați până ce se întâlnește un "1".
2. Primul "1" este lăsat nemodificat, dar toți biții care urmează sunt complementați până ce se întâlnește un nou "1".
3. Bitul "1" este complementat, dar toți biții care urmează sunt lăsați nemodificați până ce se întâlnește un nou "1".
4. Se revine la pasul 2.

Algoritmii menționați anterior nu sunt foarte eficienți, deoarece pentru fiecare pixel al imaginii trebuie parcurși pașii 1 până la 4, astfel încât timpul de calcul este destul de lung. O alternativă o constituie calcularea o singură dată a reprezentării în cod Gray a numerelor binare, urmată de găsirea reprezentării corespunzătoare în cod Gray pentru fiecare pixel de imagine ca un element cu indicele  $i$  într-un tabel, unde  $i$  este reprezentarea binară a pixelului considerat.

Generarea tabelului de conversie din cod binar în cod Gray se face recursiv, conform următoarei relații:

$$Gray(i) = 2^k + Gray(i - 2^k) - 1 \quad (6.3)$$

unde

$$k = \lceil \log_2 i \rceil - 1, \quad Gray(0) = 0 \quad \text{și} \quad Gray(1) = 1$$

Un exemplu de cod sursă C pentru calculul reprezentării în cod Gray pe 8 biți este:

```

int i,k;
unsigned char gray[256]={0};
gray[0]=0;
gray[1]=1;
k=1;
for(i=2;i<=256;i++)
{
    if(i-(2<<k)>=0)
        k++;
    gray[i]=(1<<k)+gray[i-2*(i-(1<<k))-1];
}

```

Problema inversă, de determinare a codului binar care corespunde unui cod Gray dat, poate fi rezolvată prin utilizarea tabelului rezultat conform algoritmului descris anterior: pentru o reprezentare în cod Gray,  $i$ , reprezentarea în cod binar este dată de indicele elementului  $i$  din tabel.

Am dezvoltat două programe în C pentru descompunerea unei imagini de intrare pe scară de gri în planele sale de biți (utilizând reprezentarea în cod Gray), respectiv pentru reconstrucția imaginii inițiale din reprezentarea lor prin planele de biți corespunzătoare. Numele acestor programe sunt:

```

bplan input_image_file;
bplanrec reconstructed_image_file;

```

Pentru a avea posibilitatea efectuării unor comparații, am dezvoltat și variante ale acestor programe care nu utilizează reprezentarea în cod Gray. Planele de biți rezultate pentru imaginea Lena sunt prezentate în figura 6.8 (planele cele mai semnificative până la planele cele mai puțin semnificative).

Cod Gray



a)

Cod binar



b)





c)



d)



e)



f)



g)



h)



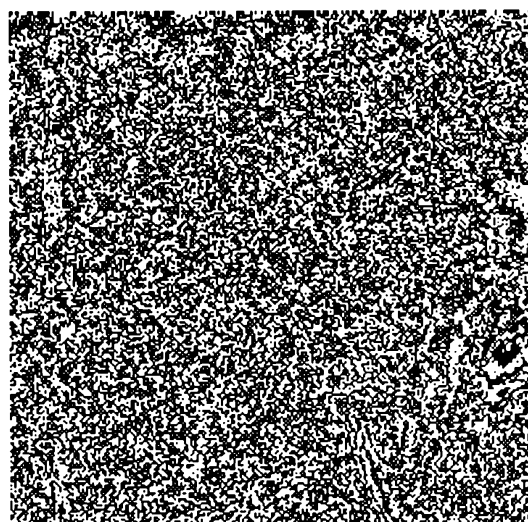
i)



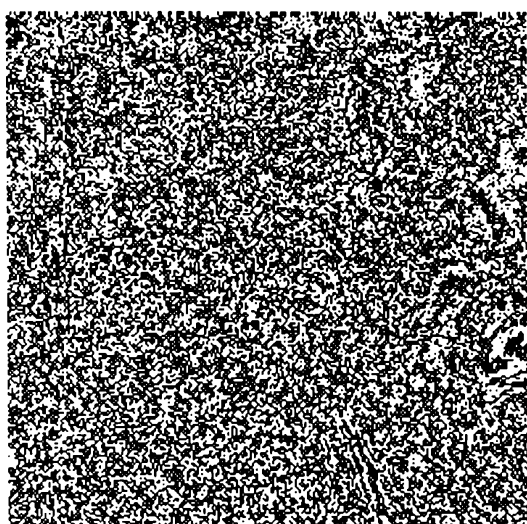
j)



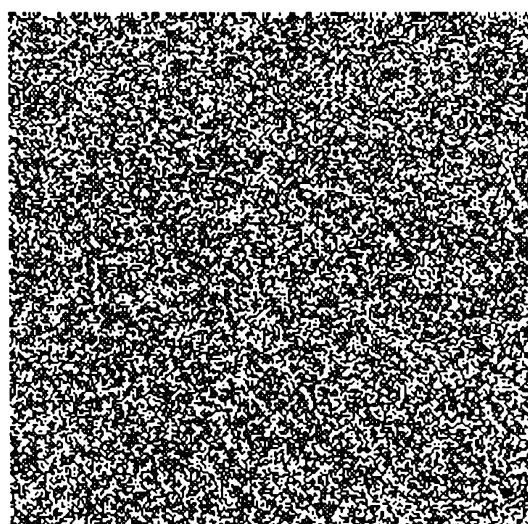
k)



l)



m)



n)

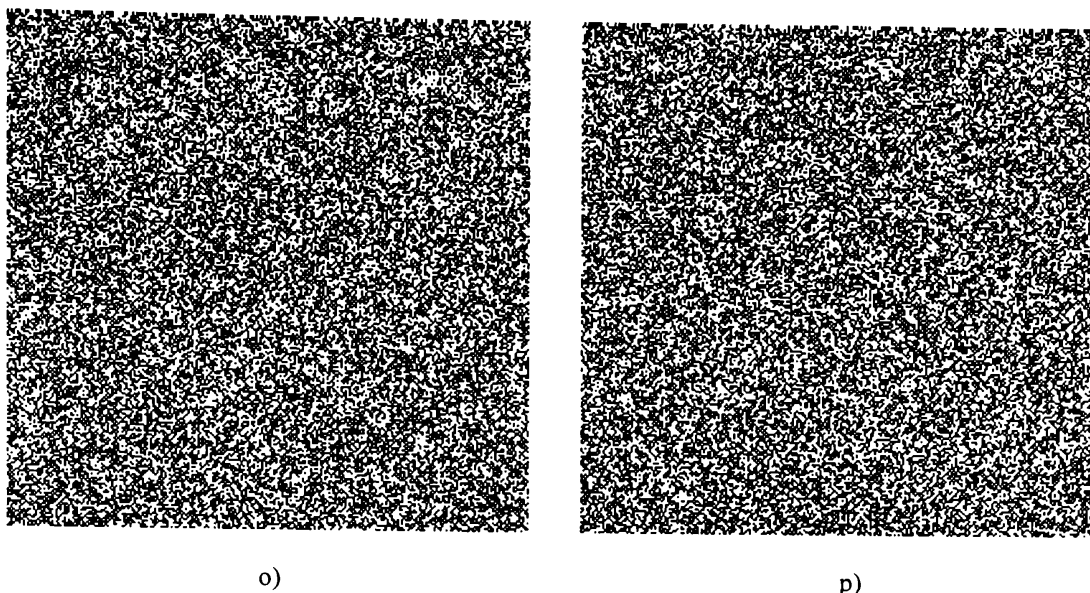


Fig. 6.8. Descompunerea pe plane de biți prin reprezentare în cod binar și cod Gray

#### 6.4. Scheletul morfologic

Bazele teoretice pentru majoritatea cercetărilor în domeniul codării digitale a imaginilor din ultimul deceniu au fost furnizate de modelarea autoregresivă și transformările ortogonale, cum sunt transformările Fourier sau Karhunen-Loeve. Ambele abordări au avut drept punct de plecare exploatarea proprietăților structurii algebrice a semnalelor. Totuși, în cazul semnalelor binare de imagine, care sunt percepute în primul rând ca forme geometrice, este necesară o reprezentare care să accentueze mai degrabă structura geometrică decât cea algebrică. O astfel de reprezentare este *scheletul imaginii*.

În general, termenul de schelet a fost utilizat pentru a descrie o reprezentare sub formă de linie subțiată a imaginii binare, care să simbolizeze forma și să furnizeze informații despre mărime, orientare și conectivitate. Scheletul a fost deja aplicat în descrierea formelor biologice, în recunoașterea formelor, în metalografia cantitativă și inspecția industrială automată, precum și în codarea imaginilor.

##### 6.4.1. Conceptul de axă medială

Noțiunea de schelet al unei imagini binare continue a fost introdus pentru prima oară de Blum [11], care inițial l-a denumit "*axă medială*", iar mai târziu "*axă simetrică*". Procedura inițială propusă de Blum pentru obținerea axei mediale s-a bazat pe definirea unor "unde" la momentul  $t=0$  de-a lungul punctelor conturului, urmată de propagarea acestor "unde" ca fronturi către centrul obiectului cu o viteză constantă, conform principiului lui Huygens. Punctele în care aceste unde se întâlnesc și se "sting" formează axa medială a obiectului. Punctele axei mediale împreună cu timpul lor de sosire definesc *funcția axă medială (simetrică)*. Una dintre proprietățile importante ale acestei funcții este abilitatea sa de a reconstrui conturul obiectului prin propagarea inversă a fronturilor de undă.

Ulterior, o serie de cercetători au dezvoltat o teorie matematică pentru schelet: Kotelly [106] și Calabi [29] pentru imagini binare continue, Rosenfeld și Pfaltz [193], Mott-Smith [153] și Montanari [148] pentru imagini binare discrete. Influențat de aceste contribuții, Blum a luat în considerație două noi metode de determinare a axei mediale. Prima metodă utilizează *distanța simetrică a punctelor*, de la un punct al scheletului până la contur. În cadrul celei de-a doua metode, Blum a demonstrat că axa simetrică este locul geometric al centrelor *discurilor maximale* inscriptibile într-un obiect al imaginii.

În paralel și independent de această evoluție a noțiunii de schelet, s-a dezvoltat și morfologia matematică, ca fundamentare teoretică pentru analiza imaginilor pe baza descrierii cantitative a structurilor geometrice. Una dintre proprietățile importante ale morfologiei matematice este aceea că realizează o unificare a modului de realizare a mai multor sisteme invariante la translație, liniare sau neliniare. Ca aplicație, Lantuejoul [111] a demonstrat că scheletul poate fi obținut printr-o serie de transformări morfologice. Pentru a face distincția între scheletul obținut prin metode morfologice de scheletul obținut prin alte metode, acesta este denumit în literatura de specialitate *schelet morfologic*.

#### 6.4.2. Implementarea scheletului morfologic

*Scheletul morfologic*  $SK(X)$  al imaginii continue a obiectului  $X$ , văzut ca un subset al lui  $R^2$ , este definit ca mulțimea a centrelor discurilor maximale inscriptibile în  $X$ . Un disc este *maximal* dacă nu este conținut în orice alt disc complet inclus în  $X$ . Din acest motiv, un disc maximal trebuie să atingă conturul unui obiect  $X$  în cel puțin două puncte diferite. De exemplu, scheletul unui cerc este chiar centrul său, iar scheletul unui sector unghiular este bisectoarea acestuia.

Fie  $S_r(X)$ ,  $r > 0$ , cel de-al  $r$ -lea subset de schelete, adică, mulțimea centrelor discurilor maximale cu raza egală cu  $r$ . Aceste subseturi de schelete pot fi obținute prin utilizarea operatorilor morfologici de erodare și deschidere. În aceste condiții, Lantuejoul a demonstrat [111] că scheletul global al obiectului  $SK(X)$  există și este egal cu:

$$SK(X) = \bigcup_{r>0} S_r(X) = \bigcup_{r>0} [(X \ominus rB) - (X \ominus rB)_{drB}] \quad (6.4)$$

unde  $rB$  semnifică discul deschis cu raza  $r$  și  $drB$  este un disc închis de rază infinit mică  $dr$ . Contururile mulțimilor erodate  $(X \ominus rB)$  pot fi văzute ca fronturile de undă care se propagă ale lui Blum, unde timpul de propagare coincide cu raza  $r$ . Prin scăderea din aceste versiuni erodate ale lui  $X$  a deschiderilor lor cu  $drB$ , vor rămâne doar punctele unghiulare, care sunt puncte ale scheletului. Mulțimea originală  $X$  poate fi reconstruită ca reuniune pentru toate valorile  $r > 0$  a subseturilor  $S_r(X)$  dilatate prin discurile deschise  $rB$ , respectiv.

Deși scheletul nu este un concept ușor de digitizat, se poate defini totuși scheletul morfologic al imaginilor binare eșantionat pe o rețea rectangulară sau hexagonală. Să considerăm că o submulțime  $X$  a lui  $Z^2$  reprezintă o imagine digitală binară. Presupunând că  $X$  este nenulă și mărginită, Serra [211] a furnizat un algoritm pentru determinarea scheletului morfologic  $SK(X)$  al unei imaginii binare discrete eșantionate pe o grilă hexagonală:

$$S_n(X) = (X \ominus nB^s) - (X \ominus nB^s)_B, \quad n = 0, 1, 2, \dots, N \quad (6.5)$$

$$SK(X) = \bigcup_{n=0}^N S_n(X) \quad (6.6)$$

unde  $S_n(X)$  semnifică cel de-al  $n$ -lea subset de schelete ale lui  $X$ , iar  $B$  este un element structurant discret format din șapte pixeli pe o grilă hexagonală.

Imaginea discretă  $X$  poate fi reconstruită *exact* ca reuniune finită a celor  $N+1$  subseturi de schelete dilatate printr-un element structurant hexagonal de dimensiuni potrivite.

$$X = \bigcup_{n=0}^N [S_n(X) \oplus nB] \quad (6.7)$$

Deși cercurile și hexagoanele sunt elemente structurante de dorit, datorită gradului lor înalt de simetrie, Maragos și Schafer [130] au considerat că nu este necesară limitarea elementelor structurante la forme circulare. Ei au extins definiția scheletului morfologic dată de Serra, prin utilizarea oricărui element structurant simetric convex. Figura 6.9 prezintă elementele structurante CERC, PĂTRAT și ROMB de rază egală cu 1 pe o grilă rectangulară.

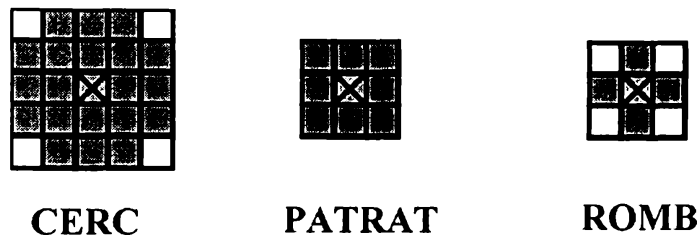


Fig. 6.9. Elemente structurante de rază 1

Indiferent care este elementul structurant utilizat pentru obținerea scheletului, subseturile de schelete rezultate pot reconstrui exact imaginea originală prin utilizarea relației (6.7). Astfel, informația totală din imaginea originală finită  $X$  este echivalentă cu cea din mulțimea finită a tuturor subseturilor de schelete  $S_n(X)$ , împreună cu cunoașterea indicelui corespunzător  $n$ .

Maragos și Schafer [130] au exploatat anumite proprietăți algebrice ale operatorilor morfologici pentru a dezvolta doi algoritmi pentru descompunerea, respectiv reconstrucția scheletelor.

Erodarea lui  $X$  cu  $nB$ , care necesită  $f(n^2)$  operații paralele logice ȘI pe întreaga imagine, se poate efectua mult mai rapid prin erodarea succesivă a lui  $X$  cu  $B$  de  $n$  ori, ceea ce implică doar  $f(n)$  operații paralele ȘI. Pentru a putea avea o bază de comparație pentru metoda propusă în această teză, am implementat și acest algoritm propus de Maragos și Schafer, care exploatează proprietatea de mai sus. Pașii implementării sunt:

```

pas1:      n:=0; EROS1=X;
pas2:      EROS2:=EROS1⊖Bs;
pas3:      IF(EROS2=∅) THEN
            N:=n; SN(X):=EROS1; STOP;
pas4:      OPEN:=EROS2⊕B;
pas5:      Sn(X):=EROS1-OPEN;
pas6:      n:=n+1; EROS1:=EROS2;
            GO TO pas 2;

```

unde  $EROS1$ ,  $EROS2$  și  $OPEN$  sunt trei mulțimi suficient de largi pentru a menține imaginea obiectului și fundalul său.

Algoritmul de mai sus, ilustrat în figura 6.10 necesită doar  $N$  erodări și  $N$  dilatări ale imaginii cu  $B$  și, din acest motiv, are o complexitate liniară.

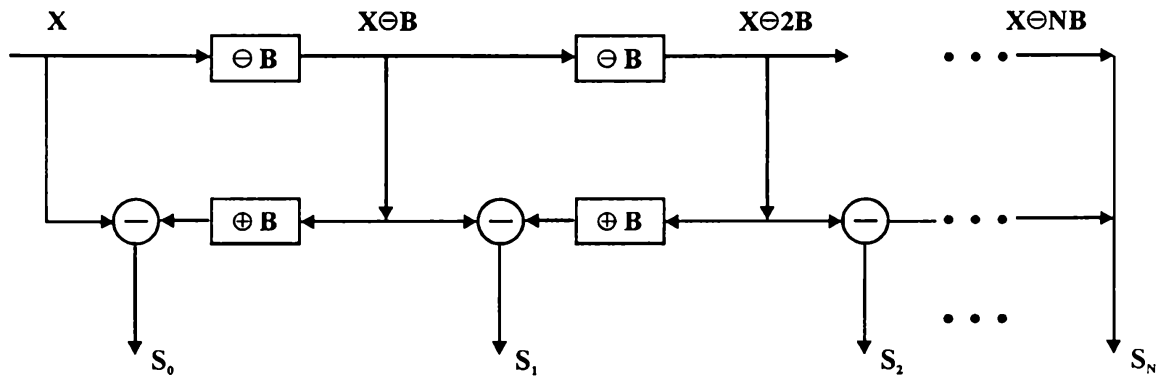


Fig. 6.10. Algoritmul de descompunere în subșeturi de schelete

În cazul celui de-al doilea algoritm propus de Maragos și Schafer, ținând cont că dilatarea este distributivă față de reuniunea mulțimilor și asociativă, ecuația (6.7) poate fi rescrisă ca:

$$X = \left[ \left[ \left[ S_N(X) \oplus B \right] \cup S_{N-1}(X) \right] \oplus B \cup S_{N-2}(X) \dots \right] \oplus B \cup S_0(X) \quad (6.8)$$

Am implementat și acest algoritm, tot din motive de comparație. El utilizează ecuația (6.8) pentru reconstrucție:

```

pas1:      n:=N; A:=0;
pas2:      A:=A ∪ Sn(X);
pas3:      IF (n=0) STOP;
           ELSE A:=A ⊕ B;
pas4:      n:=n-1;
           GO TO pas 2;

```

unde  $A$  este o mulțime suficient de largă pentru a menține imaginea obiectului și fundalul său.

Figura 6.11 ilustrează acest algoritm.

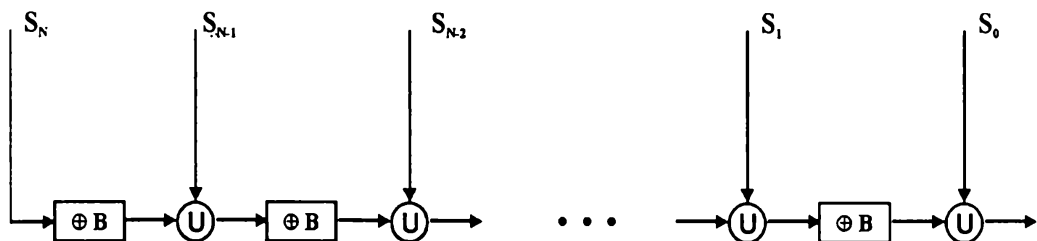


Fig. 6.11. Algoritmul de reconstrucție

Scheletul morfologic prezentat mai sus este calculat cu ajutorul unui element structurant  $B$ , cu o formă fixă, dar cu dimensiunea crescând liniar ( $nB$  utilizat în ecuațiile (6.5) și (6.6)). Sapiro și Malah [204] descriu un algoritm care calculează scheletele morfologice cu ajutorul unor elemente structurante având dimensiunea crescătoare *exponențial* prin dublarea mărimii la fiecare pas al scheletului  $n$ , adică,

dimensiunea elementului structurant la pasul  $n+1$  al calculării scheletului este dublă față de dimensiunea celui utilizat la pasul  $n$  (forma rămânând neschimbată).

Această reprezentare este motivată prin faptul că, atunci când se utilizează elemente structurante de dimensiuni mari, se obțin mai puține seturi de schelete, ceea ce face posibilă obținerea unui raport de compresie ridicat.

În varianta modificată de schelet morfologic al lui  $X$ , fiecare subset de schelete este calculat prin intermediul celui mai mare element structurant posibil. Scheletul obținut prin această metodă poate fi definit ca:

$$MS(X) = \bigcup_{n=0}^{N_M(B)} M_n(X) \quad (6.9)$$

unde

$$\begin{aligned} M_0(X) &= S_0^B(X) \\ M_n(X) &= S_1^{B(n)}(X), \quad n = 1, 2, \dots, N_M(B) \\ B(n) &= 2^{n-1} B, \quad n = 1, 2, \dots, N_M(B) \\ N_M(B) &= \max \{n \mid X \ominus 2^{n-1} B \neq \emptyset\} \end{aligned} \quad (6.10)$$

Subsetul de schelete  $M_n(X)$  ( $n \geq 1$ ) conține toate punctele  $z \in X$  (și doar aceste puncte) astfel încât elementul  $(2^{n-1}B)_z$  este maximal în  $X$ . De aceea,  $MS(X)$  conține toate centrele  $z$  ale elementelor structurante maximale  $(2^{n-1}B)_z$ ,  $n=1, 2, \dots, N_M(B)$ , precum și subsetul  $M_0(X)$  care conține punctele scheletului de "rază zero". Se observă că prin utilizarea acestui schelet morfologic modificat, se obțin mai puține subseturi de schelete datorită utilizării elementului structurant cu dimensiunile crescând exponențial, deoarece

$$N_M(B) = \lceil \log_2 N(B) \rceil < N(B) \quad (6.11)$$

unde  $N(B)$  este numărul de subseturi de schelete obținut prin algoritmul lui Maragos și Schafer.

Imaginea poate fi reconstruită din scheletul morfologic modificat prin relația

$$X = \bigcup_{n=0}^{N_M(B)} [M_n(X) \oplus B(n)] \quad (6.12)$$

Algoritmul lui Maragos și Schafer poate fi rescris prin utilizarea celor două relații anterioare după cum urmează:

```

pas1:      n:=0; EROS1=X;
pas2:      EROS2:=EROS1⊖B†;
pas3:      IF(EROS2=∅) THEN
              N:=n; SN(X):=EROS1; STOP;
pas4:      OPEN:=EROS2⊕B;
pas5:      Sn(X):=EROS1-OPEN;
pas6:      n:=n+1; EROS1:=EROS2;
              B:=B⊕B; // dublarea elementului structurant B prin dilatare
              GO TO pas 2;

```

iar pentru reconstrucție:

```

pas1:      n:=N; A:=0;
pas2:      A:=A ∪ Sn(X);
pas3:      IF (n=0) STOP;
              ELSE A:=A⊕B;
              B:=B⊕B; // dublarea lui B prin dilatare;

```

```

pas4:      n:=n-1;
           GO TO pas 2;

```

Am realizat implementarea acestor algoritmi prin două programe C, pe care le-am denumit **skel.c** și **skelref.c**. Primul program calculează  $N$  subseturi de schelete ale unei imagini binare de intrare, subseturile rezultate fiind înscrise în  $N$  fișiere separate cu numele **skel0**, **skel1**, ..., **skelN**. Cel de-al doilea program realizează refacerea imaginii inițiale din cele  $N$  subseturi de schelete.

### 6.5. Reprezentarea conturului pentru imagini binare

Conturul unei imagini binare constă din acele puncte ale imaginii care au cel puțin 4 vecini în regiunea care constituie fundalul obiectului. Orice punct al conturului poate fi ales ca punct inițial pentru trasarea conturului. Un *lanț de contur* pornește dintr-un punct inițial al conturului și se sfârșește în același punct, conectând toate punctele de pe contur și constă dintr-o succesiune de pași făcuți într-una din cele opt direcții posibile. Cele opt direcții sunt codate cu numerele 0 până la 7, după cum se vede din figura 6.12.

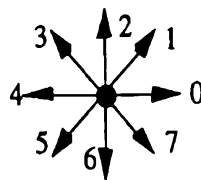


Fig. 6.12. Direcții de codare a conturului într-o octo-conectivitate

În acest fel, o secvență de direcționale  $r_1, r_2, \dots, r_N$ , cu  $0 \leq r_i \leq 7$ , descrie fără nici o nedeterminare conturul imaginii (cu ajutorul a 8 componente). Specificarea suplimentară a coordonatelor absolute ale punctului inițial ales permite o poziționare exactă a conturului pe rastrul imaginii.

Algoritmul de codare a conturului este o procedură fundamentală în domeniul prelucrării imaginilor. El este utilizat foarte des în contextul analizei formelor, a extragerii caracteristicilor geometrice sau al compresiei imaginilor. Codurile pentru contur au avantajul că permit calcularea caracteristicilor geometrice ale obiectului prin manipulari numerice ale codului lanț al conturului.

În general, algoritmi pentru codarea conturilor sunt foarte eficienți în ceea ce privește timpul de calcul, un cod lanț pentru imaginile transformate geometric sau morfologic putând fi calculat direct pornind de la secvența originală. Un cod lanț al conturului permite reconstrucția exactă a obiectului.

Întregul proces de determinare a codului pentru contur constă în două faze, una de localizare a obiectului și cealaltă de trasare a conturului. Prima fază explorează imaginea linie după linie până ce se determină punctul inițial al următorului obiect. Cea de-a doua fază trasează conturul acestui obiect, memorând secvența de coduri rezultată într-un fișier. Cele două faze ale algoritmului alternează pe durata întregului proces până ce conturile tuturor obiectelor sunt codate.

În cele ce urmează sunt descrise sumar principalele obiective ale algoritmului și soluțiile posibile:

1. Se presupune că în imagine sunt conținute mai multe obiecte. Conturile corespunzătoare trebuie memorate ca secvențe de numere într-un fișier în



- ordine secvențială. Trebuie asigurat ca punctele de start și sfârșit ale fiecărui cod lanț al conturului să poată fi detectate în momentul decodării.
2. Obiectele detectate anterior pot de asemenea intersecta liniile de scanare în faza de căutare a unui nou obiect. Din acest motiv, obiectele detectate trebuie să fie etichetate ca “invizibile” pentru următoarele faze de căutare a obiectelor.
  3. Golurile din interiorul obiectelor trebuie să fie recunoscute și etichetate ca și goluri. Algoritmul trebuie să determine mai întâi conturul exterior al obiectului, urmând ca abia ulterior să prelucreze golurile din obiect. Nu este permis ca punctul inițial al unui contur interior să fie în același timp un punct al conturului exterior obiectului.

Faza de trasare a conturului conduce la determinarea secvenței binare a codului conturului și la memorarea acestei secvențe într-o arie  $E[1, \dots, L_{max}]$ , unde  $L_{max}$  definește lungimea maxim admisibilă a conturului unui obiect. Pentru o imagine de dimensiune  $N \times N$ , se poate obține, în cazul cel mai defavorabil, un contur format din  $N^2/2$  puncte. În cele mai multe cazuri această valoare este prea mare, o valoare  $L_{max}=2000$  putând fi de obicei acceptabilă pentru  $512 \times 512$  pixeli.

Codul conturului, așa cum a fost generat în  $E[1, \dots, L_{max}]$ , este apoi stocat într-un fișier rezultat după completarea fazei de trasare a conturului pentru un obiect sau un gol. Pentru aria  $E$  este utilizat un format special de date, obținându-se pentru lanțul generat și datele adiționale:

$E(1)$	numărul total al pașilor de contur (2 bytes)
$E(2)$	coordonata $x$ a punctului inițial (1 byte pentru imagini mai mici sau egale cu $256 \times 256$ puncte dimensiune)
$E(3)$	coordonata $y$ a punctului inițial (1 byte pentru imagini mai mici sau egale cu $256 \times 256$ puncte dimensiune)
$E(4)$	etichetă: obiect sau gol (1 bit)
$E(5)$	direcția $r_1$ a primului pas de contur (3 biți)
$E(6)$	direcția $r_2$ a celui de-al doilea pas de contur (3 biți)
...	...
$E(Z+4)$	direcția $r_z$ a ultimului pas de contur (3 biți).

În fișierul rezultat, aceste date sunt stocate consecutiv.

Figura 6.13 prezintă organigrama algoritmului de trasare.

Operațiunea de scanare a imaginii pentru detectarea noilor obiecte are loc din partea de sus a imaginii spre partea de jos, adică de la  $y=Y_{max}$  la  $y=Y_{min}$ ; și de la stânga la dreapta, adică de la  $x=X_{min}$  la  $x=X_{max}$ . Valoarea simbolică de gri  $G-1$  etichetează conturul obiectului tratat anterior.

De aceea, valorile de gri ale imaginii originale trebuie să fie anterior limitate la maxim  $G-2$ . Aceste contururi de obiecte etichetate nu sunt detectate din nou în fazele ulterioare de detecție a obiectelor. O tranziție de la o valoare de gri mai mică decât  $S_2$  la o valoare de gri mai mare decât  $S_1$  indică apariția unui nou obiect. Atunci se revine din nou la faza de trasare a conturului.

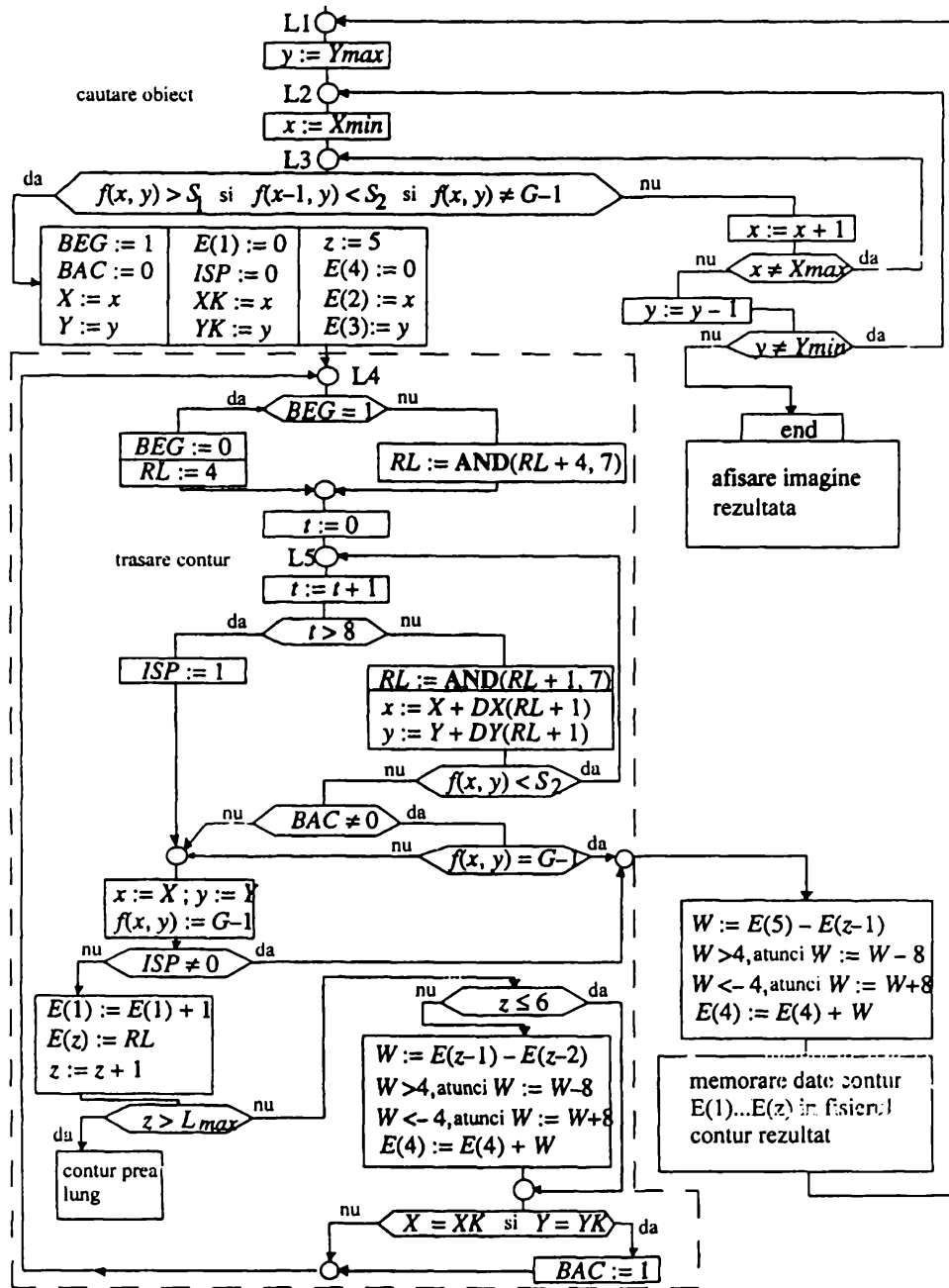


Fig. 6.13. Organigrama algoritmului de trasare

Unele variabile sunt inițializate la începutul fazei de trasare a conturului. Ele sunt reactualizate pe durata acestei faze. Semnificația acestor variabile este:

- $X, Y$  sunt coordonatele punctului curent al conturului,
- $XK, YK$  sunt coordonatele punctului inițial al conturului,
- $BEG$  este egală cu 1 dacă punctul de pe conturul curent este totodată și punctul inițial,
- $ISP$  este egală cu 1 dacă obiectul (degenerat) este doar un punct izolat,
- $z$  este indicele următoarei locații libere din aria  $E$ ,
- $BAC$  este egală cu 1 dacă punctul curent al conturului nu reajunge din nou în punctul inițial. (Această situație nu implică faptul că întregul contur este deja trasat. De exemplu, dacă punctul inițial  $(XK, YK)$  este pe o zonă îngustă

situată între două părți ale unui obiect, atunci acest punct va fi parcurs de două ori pe durata trasării conturului. Totuși, scanarea imaginii în ordinea crescătoare a liniilor de imagine evită în mod normal această situație deoarece punctul din stânga sus al obiectului este considerat ca punct inițial. Acest lucru este fals doar în cazul în care valoarea de gri a punctului din stânga-sus al obiectului este în domeniul cuprins între  $S_1$  și  $S_2$

- $E(1) \dots E(4)$  sunt inițializate sau reactualizate cu datele corespunzătoare ale conturului curent.

Faza de trasare a conturului utilizează și alte variabile pentru realizarea pașilor respectivi. Semnificația lor este următoarea:

- $RL$ , cu  $0 < RL < 7$ , definește direcția pasului de contur anterior. Punctul curent al conturului  $(X, Y)$  este atins prin pasul curent al conturului. Toți cei 8 vecini ai punctului  $(X, Y)$  sunt testați consecutiv pentru a vedea dacă valorile lor de gri depășesc pragul  $S_2$ . Văzută dinspre punctul  $(X, Y)$ , prima direcție testată este  $RL + 5$  (modulo 8). Aceasta corespunde direcției lui  $RL$ , rotită cu  $180^\circ + 45^\circ$ . La începutul trasării conturului se utilizează o valoare prestabilită  $RL = 4$ . Variabila  $BEG$  este setată la 0, adică următorul punct al conturului este presupus a nu fi punctul inițial. (Ar putea fi doar în cazul unui obiect degenerat în punct izolat.)
- $DY$  definește incrementele  $x$  și  $y$  cu care se efectuează deplasarea din punctul  $(X, Y)$  pe direcția selectată  $RL$  către următorul punct al conturului. Două tabele de căutare  $DX(1 \dots 8)$  și  $DY(1 \dots 8)$  specifică valorile acestor două incremente. Valoarea lui  $RL$  este situată între 0 și 7, iar  $RL + 1$  este utilizat ca și indice de tabel.
- Variabila  $t$  contorizează numărul vecinilor lui  $(X, Y)$  testați până în acel moment, fără a se detecta noul punct al conturului. Dacă valoarea lui  $t$  este egală cu 8, atunci obiectul este un punct izolat.
- $W$  definește unghiul dintre direcția pasului anterior de contur și direcția pasului curent de contur. Acest unghi este un multiplu întreg de  $45^\circ$ , putând avea semn pozitiv sau negativ. Suma acestor  $W$  valori este memorată în  $E(4)$ . Valoarea  $W$  a ultimului pas către  $(XK, YK)$  este adăugată la sfârșitul procesului de trasare a conturului. Valoarea lui  $E(4)$  rezultată este un multiplu întreg de 8 (deoarece  $8 \cdot 45^\circ = 360^\circ$ ), semnul său specificând dacă conturul mărginește un obiect ( $E(4) > 0$ ) sau un gol ( $E(4) < 0$ ).
- Variabila  $z$  contorizează numărul de poziții ocupate de conturul curent în aria  $E$ .

Conținutul  $E(1) \dots E(Z+4)$  al ariei  $E$  este adăugat la fișierul rezultat de coduri ale conturului, la sfârșitul fazei de trasare a conturului. În continuare începe faza de detectare a următorului obiect sau gol din imagine. Problema este în ce poziție a imaginii trebuie început procesul de căutare. Obiectele și golurile pot să apară indiferent de complexitatea topologică, de exemplu, un obiect are un gol, acest gol conține un obiect, acesta are un gol, ș.a.m.d. Algoritmul presupune cazul cel mai defavorabil, faza de căutare pornind întotdeauna din punctul  $(X_{min}, Y_{max})$ .

În sfârșit, ultima fază de căutare a obiectului conduce la punctul  $(X_{max}, Y_{min})$  fără detectarea unui nou contur. Punctele conturului sunt etichetate cu  $G-1$ , ele apărând ca

linii luminoase în imaginea rezultată. Această etichetare este făcută pe durata procesului de trasare a conturului.

Programul C pe care l-am elaborat pentru implementarea algoritmului de trasare, în conformitate cu organigrama din figura 6.13 este prezentat în continuare:

```

Label1:    y=Ymax;
Label2:    x:=Ymin;
Label3:    if((f[x,y]>S1) && (f[x-1,y]>S2) && (f(x,y) !=G-1))
           {
           BEG=1; ISP=0; BAC=0; E[1]=0; z=5; X=x; XK=x;
           E[2]=x; Y=y; YK=y; E[3]=y; E[4]=0;
Label4:    if(BEG=1)
           {
           BEG=0; RL=4;
           }
           else
           RL=(RL+4) & 7;
           t=0;
Label5:    t++;
           if(t>8) ISP=1
           else
           {
           RL=(RL+1) & 7; x=X+DX[RL+1];
           y=Y+DY[RL+1];
           if(f[x,y]<S2) goto Label5;
           if(BAC!=0)
           if(f[x,y]==G-1) goto Label6;
           }
           x=X; y=Y;
           if(ISP!=0) goto Label6;
           E[1]++; E[z]=RL; z++;
           if(z>Lmax)
           {printf("contour too long"); exit(1);}
           if(z>6)
           {
           W:=E[z-1]-E[z-2];
           if(W>4) W-=8;
           if(W<=-4) W+=8;
           E[4]+=W;
           }
           if(X==XK && Y=YK) BAC=1;
           goto Label4;
Label6:    W=E[5]-E[z-1];
           if(W>4) W-=8;
           if(W<=-4) W+=8;
           E[4]+=W;
           for(i=0; i<z; i++)
           fprintf(out, "%d", E[i]);
           goto Label1;
           }
           else
           {
           x++;
           if(x!=Xmax) goto Label3;
           y--;
           if(y!=Ymin) goto Label2;
           };

```

## 6.6. Tehnici combinate contur-schelet

### 6.6.1. Minimizarea scheletului condiționată de contur

Prin exploatarea suprapunerilor subseturilor de schelete dilatate, Maragos și Schafer [130] au demonstrat că subseturile de schelete definite în 6.4.1 nu sunt minimale. Ei au definit metode de înlăturare a anumitor puncte din schelet, care nu afectează posibilitatea reconstrucției exacte a imaginii originale. De asemenea, ei au propus algoritmi pentru determinarea scheletului minimal.

Un *subset minimal* al scheletului este definit ca parte a scheletului original, ale cărui puncte sunt suficiente pentru reconstrucția exactă, dar pentru care înlăturarea unui singur punct conduce la o reconstrucție parțială.

Să considerăm că  $X$  este imaginea binară originală, iar  $S_n(X)$ ,  $n=1,2,\dots,N$ , sunt subseturile sale de schelete în raport cu un element structurant  $B$ . Pentru fiecare subset de schelete cu indicele  $n$ , se definește o funcție binară:

$$k_n(i, j) = \begin{cases} 1 & \text{dacă } (i, j) \in nB \\ 0 & \text{dacă } (i, j) \notin nB \end{cases} \quad (6.13)$$

numită *funcția caracteristică* a mulțimii  $nB$ .

Pentru fiecare  $n$ , vom deplasa funcția  $k_n$  în toate punctele lui  $S_n(X)$ , apoi vom aduna algebric contribuțiile acesteia pentru toate punctele lui  $S_n(X)$  și pentru toate valorile lui  $n$ . Apoi vom crea o funcție *pseudo-gri*,  $pgf(X)$ , al cărei domeniu de suport este identic cu al imaginii originale  $X$  și ale cărei valori în fiecare punct al imaginii sunt  $\geq 1$ ; această funcție pseudo-gri este egală cu:

$$[pgf(X)](i, j) = \sum_{n=0}^N \sum_{(r,t) \in S_n(X)} k_n(i-r, j-t) \quad (6.14)$$

pentru orice  $(i, j) \in Z^2$ .

În continuare, pentru a decide dacă un anumit punct  $(r, t) \in S_n(X)$  poate fi înlăturat din subsetul inițial de schelete, trebuie verificat mai întâi dacă valoarea funcției pseudo-gri  $pgf(X)$  este  $\geq 2$  în toate punctele domeniului de suport al funcției respective translate  $k_n(i-r, j-t)$ . În acest caz, acest punct  $(r, t)$  este înlăturat din schelet, iar  $k_n(i-r, j-t)$  se scade algebric din  $[pgf(X)](i, j)$ . În caz contrar, procedura se repetă pentru următorul punct al scheletului până ce sunt verificate toate punctele.

Algoritmul de minimizare propus de Maragos și Schafer presupune modificarea și memorarea fiecărui subset de schelete separat. Pentru a reduce capacitatea de memorie necesară și complexitatea calculului, ei au propus un nou algoritm de minimizare a scheletului. Acesta utilizează, pentru minimizare, proprietățile structurale ale scheletului final și structura conturilor calculate pentru o imagine originală. Pe scurt, algoritmul propus utilizează doar două imagini binare în locul unei arii de subseturi de schelete, pentru a ajunge la un schelet morfologic unic, minimizat. Pentru calculul funcției pseudo-gri sunt necesare doar  $N$  dilatări cu un element structurant inițial. Figura 6.14 prezintă organigrama algoritmului pentru calculul funcției pseudo-gri.

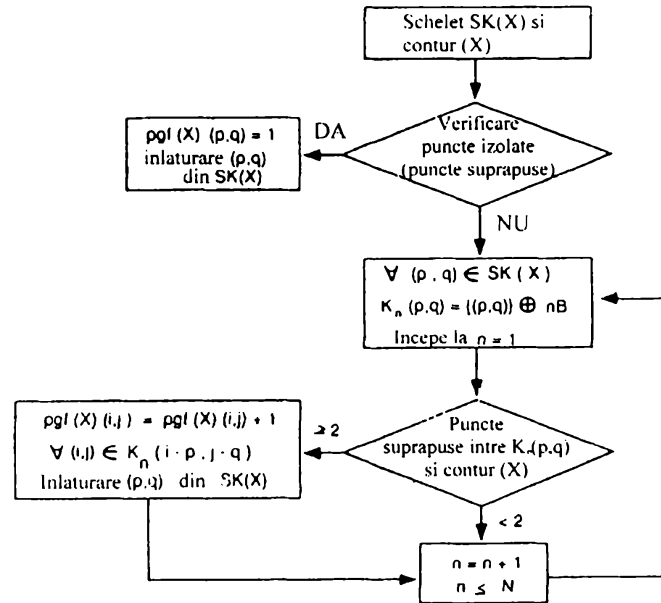


Fig. 6.14. Organigrama de calcul a funcției pseudo-gri

În continuare voi descrie procedurile de creare a funcției pseudo-gri și de minimizare a structurii scheletului. În primul rând este formulată funcția pseudo-gri prin însumarea algebrică a tuturor elementelor maximale. Un element maximal este calculat prin efectuarea unei operații de dilatare pe fiecare punct al scheletului, cu condiția de limitare a marginii, dată de informația morfologică asupra conturului.

Informația despre contur poate fi obținută prin efectuarea diferenței dintre imaginea erodată și cea originală:

$$\text{Boundary}(X) = X - (X \ominus B) \quad (6.15)$$

În acest algoritm de minimizare, sunt examinate în primul rând punctele care se suprapun între structura scheletului și structura conturului. Aceste puncte constituie un subset al lui  $SK_0$  și nu pot fi înlăturate ca puncte redundante. Ele sunt prin urmare memorate separat, fiind apoi efectuate iterații succesive pentru determinarea elementelor maximale pentru fiecare punct al scheletului. Elementele maximale sunt însumate algebric pentru a obține funcția pseudo-gri  $pgf(X)$ .

După stabilirea funcției  $pgf(X)$ , fiecare punct redundant al scheletului este înlăturat prin examinarea regiunii elementelor maximale ale fiecărui punct al scheletului. Punctele scheletului care aparțin lui  $SK_0$  sunt în mod garantat aparținătoare scheletului minim final, astfel încât aceste puncte nu mai sunt luate în considerație la următoarea examinare. Dacă valoarea funcției pseudo-gri în toate punctele regiunii elementelor maximale este mai mare sau egală cu 2, aceasta confirmă că regiunea este suprapusă cel puțin de două ori pe durata procedurii de reconstrucție. Prin urmare, acest punct corespunzător al scheletului poate fi înlăturat, deoarece această regiune a elementelor maximale poate fi acoperită de alte elemente maximale. După ce elementul redundant al scheletului a fost înlăturat, elementul maximal corespunzător este scăzut algebric din funcția pseudo-gri. Reprezentarea scheletului minimal final este obținută după examinarea tuturor elementelor maximale ale scheletului original.

Figura 6.15 prezintă organigrama algoritmului de minimizare a scheletului:

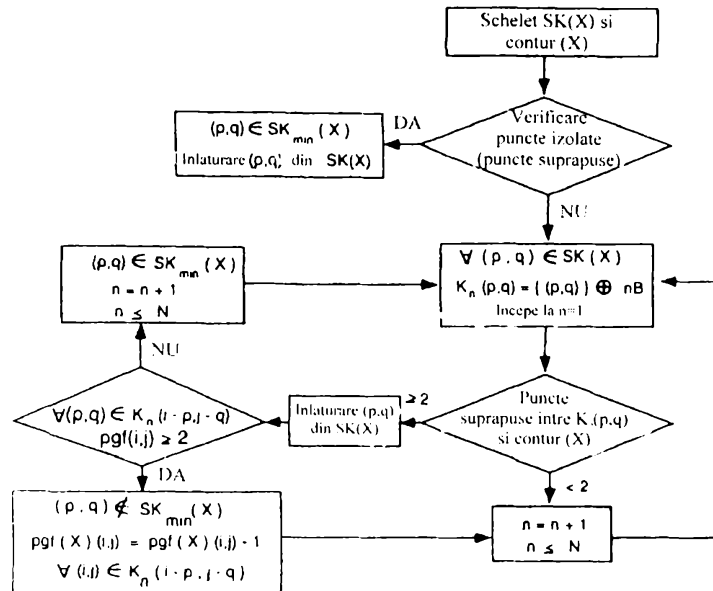


Fig. 6.15. Organigrama algoritmului de minimizare

Pentru implementarea algoritmilor descriși anterior, am elaborat două programe C. Ele sunt **pgf.c**, care calculează o imagine pe scară de gri corespunzătoare funcției pseudo-gri, și **skelmin.c** care calculează scheletul minimal.

Rezultatele obținute sunt prezentate în figura 6.16.

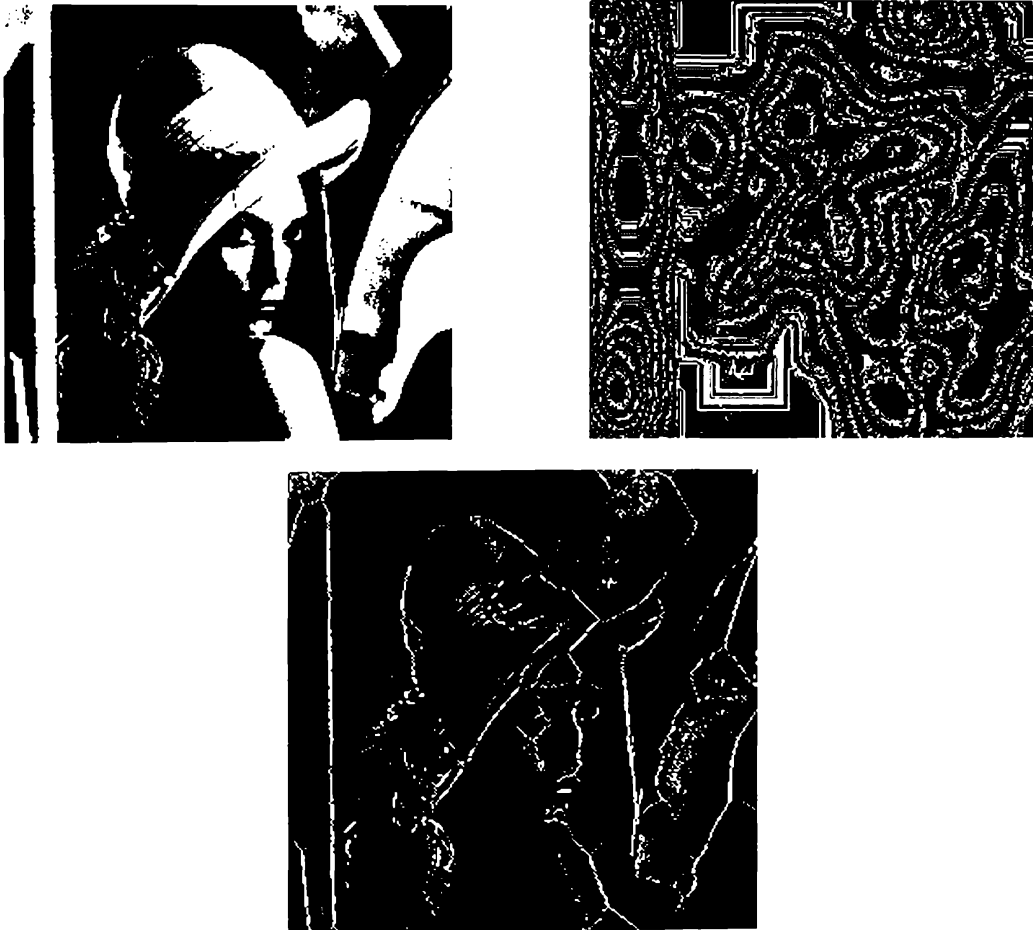


Fig. 6.16. Rezultatele algoritmului de minimizare

### 6.6.2. Reconstrucția imaginii din scheletul minimal

Pe baza teoremelor de analiză a scheletului morfologic, am elaborat următorul algoritm care furnizează o tehnică nouă de reconstrucție a imaginii prin utilizarea scheletului binar morfologic și a structurii conturului binar în locul mai multor seturi de schelete separate.

Scheletul morfologic al unei regiuni nenule și finite dintr-o imagine binară eșantionată pe o grilă dreptunghiulară se obține prin efectuarea reuniunii tuturor subseturilor de schelete.

Pentru cel de-al  $n$ -lea subset de schelete ale imaginii  $X$ , fie  $(nB)_z$  cel mai mare element centrat pe  $Z$  astfel încât  $(nB)_z \subseteq X$ . Atunci  $Z$  este un punct al scheletului celui de-al  $n$ -lea subset de schelete ale lui  $X$  dacă nu există un alt element  $(mB)_y$  în  $X$  cu  $m > n$  astfel încât  $(nB)_z \subseteq (mB)_y \subseteq X$ . Cel mai mare element  $(nB)_z$  este definit, ca și anterior, ca un element maximal, dar pentru imaginea discretă  $X$ . Punctul  $Z$  poate fi un punct al scheletului care aparține celui de-al  $n$ -lea subset de schelete ale lui  $X$  dacă și numai dacă elementul  $(nB)_z$  este elementul maximal din  $X$ .

O altă condiție este că un element maximal trebuie să atingă conturul unei imagini  $X$  cel puțin în două puncte diferite. Alternativ, un punct  $Z$  al scheletului lui  $X$  există dacă există unul și doar un subset maximal corespunzător cu proprietatea  $(n_{\max}B)_z \subseteq X$  și  $((n_{\max} + 1)B)_z \not\subseteq X$ . Algoritmul de reconstrucție rezultat poate fi reprezentat după cum urmează:

$$X = \bigcup_{z \in SK(X)} (n_{\max}B)_z \quad (6.16)$$

unde  $n_{\max}$  este dimensiunea maximă elementului structurant care depinde de punctul scheletului  $Z$ , iar  $(n_{\max}B)_z$  este chiar elementul maximal față de  $Z$ . Deoarece un element maximal întâlnește conturul obiectului în cel puțin două puncte diferite și trebuie să fie o submulțime a obiectului, informația despre contur este necesară pentru reconstrucție. Informația despre contur a unei imagini binare finite se poate obține prin extragerea morfologică a conturului (6.15) cu ajutorul unui element structurant având aceeași formă și dimensiune cu elementul structurant inițial utilizat în algoritmul de descompunere a scheletelor.

Structura conturului  $Boundary(X)$  furnizează conturul necesar al imaginii, iar scheletul morfologic  $SK(X)$  detaliază poziția fiecărui punct al scheletului. Reuniunea tuturor elementelor maximale în raport cu fiecare punct al scheletului realizează reconstrucția exactă a imaginii originale. Organigrama pentru algoritmul de reconstrucție este ilustrată în figura 6.17.

Primul pas al noului algoritm de reconstrucție constă în examinarea oricărui punct izolat din imaginea originală. Dacă obiectul original conține puncte izolate cu mărimea mai mică decât cea a elementului structurant, atunci aceste puncte izolate vor face parte totodată și din conturul obiectului, respectiv din structura scheletului său (ele fac parte din primul subset de schelete  $SK_0(X)$ ).

După prima iterație, sunt observate și salvate punctele izolate ale imaginii originale, iar următoarele iterații caută elementele maximale pentru toate punctele scheletului. După ce a fost obținut elementul maximal al unui punct al scheletului, acesta va fi salvat ca parte a obiectului original, iar acest punct al scheletului este dezactivat înaintea următoarei iterații. Procedura se repetă până la determinarea tuturor elementelor maximale.



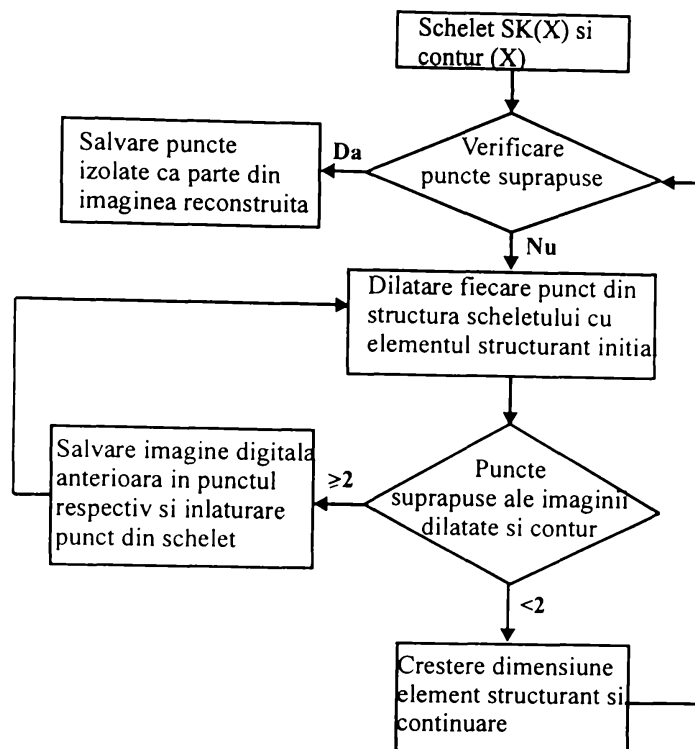


Fig. 6.17. Organigrama algoritmului de reconstrucție

Imaginea originală poate fi reconstruită exact prin reunirea tuturor elementelor maxime și a punctelor izolate.

### 6.7. Codarea celor mai puțin semnificative plane de biți

Cele mai puțin semnificative patru plane nu conțin informație foarte semnificativă. Ele conțin doar detalii fine. Dacă se utilizează o tehnică de transmisie progresivă, atunci aceste plane mai puțin semnificative pot chiar să lipsească. Mai multe detalii și o îmbunătățire a calității se pot obține fără a utiliza tehnici la fel de complicate ca și în cazul primelor patru plane de biți, mai semnificative, permițându-se în același timp și o compresie mai bună.

Eu voi propune un algoritm de codare diferit pentru cele patru plane de biți mai puțin semnificative.

Pentru planul cel mai puțin semnificativ (planul 0), am observat că pentru majoritatea imaginilor, aspectul este acela de zgomot (după cum se vede din figura 6.8). Din acest considerent am propus determinarea densității de probabilitate pentru apariția pixelilor negri/albi pentru întregul plan și transmiterea doar a acestui parametru. La decompresia imaginii, se generează local o imagine aleatoare cu aceeași densitate de probabilitate, care este apoi adăugată peste imaginea reconstruită.

Pentru următoarele trei plane de biți mai puțin semnificative (planele 1, 2, și 3), nu ne putem aștepta la o densitate de probabilitate cvasi-constantă pe întregul plan. Am propus segmentarea acestor plane în blocuri cu dimensiunile  $1/4$ ,  $1/8$  și  $1/16$  din imaginea originală, urmată de calcularea densității de probabilitate a apariției pixelilor negri/albi pentru fiecare bloc în parte. Urmează să se transmită valoarea corespunzătoare pentru fiecare bloc și aceasta să fie regenerată la decodarea imaginii.

### 6.8. Concluzii

Am propus un algoritm de codare a imaginilor care utilizează o descompunere a imaginii pe plane de biți și care tratează în mod diferit cele mai semnificative patru plane de biți și cele mai puțin semnificative patru plane de biți. Algoritmul utilizează metode ale morfologiei matematice pentru codarea celor mai semnificative plane, prin intermediul unor tehnici combinate contur-schelet. Algoritmul asigură reconstrucția perfectă a informației esențiale conținute în planele de biți cele mai semnificative.

Utilizarea metodelor morfologice permite o abordare geometrică orientată pe formă pentru reprezentarea imaginii prin contururi și schelete. Am constatat că această abordare este mai adecvată din punctul de vedere al caracteristicilor percepției vizuale umane.

Un alt avantaj al algoritmului pe care l-am propus constă în posibilitatea de a fi utilizat în sisteme de transmisie și stocare progresivă a imaginii, lucru de mare importanță, în special pentru aplicațiile de arhivare. Modul de manifestare este similar cu cel al aplicațiilor Internet. În timpul în care se decodează doar partea cea mai relevantă a informației din planele de biți, utilizatorul poate localiza imaginea recepționată și poate decide dacă procesul de decodare trebuie continuat sau se poate trece la căutarea unei noi imagini. Dacă se dorește acest lucru, atunci rezoluția imaginii obținute poate fi crescută gradat prin decodarea succesivă a tuturor celor patru plane de biți mai semnificate și a unu până la patru plane de biți mai puțin semnificative.

## Capitolul 7

### Rezultate experimentale

În acest capitol voi prezenta rezultatele obținute prin aplicarea algoritmului pe care l-am propus pentru compresia imaginilor prin utilizarea metodelor morfologiei matematice.

Voi prezenta în primul rând imaginile de test utilizate pentru evaluarea performanțelor algoritmilor. Imaginile utilizate sunt imagini standard, pentru a asigura posibilitatea comparării rezultatelor cu cele furnizate de algoritmi similari descriși în literatura de specialitate.

Algoritmul pe care l-am propus utilizează metode ale morfologiei matematice binare pentru codarea imaginilor pe scară de gri. Pentru ca acest lucru să fie posibil, este efectuată mai întâi o descompunere a imaginii pe plane de biți. Voi prezenta rezultatele obținute pentru descompunerea pe plane de biți a mai multor imagini de test. Primele patru plane de biți, care conțin cea mai mare parte din informația semnificativă, vor fi prelucrate cu metode morfologice, în timp ce ultimele patru plane de biți, care conțin în cea mai mare parte doar detalii fine, vor fi prelucrate separat pentru a obține un raport de compresie cât mai bun. Datorită complexității lor foarte ridicate, aceste plane nu pot fi prelucrate eficient prin metode ale morfologiei matematice.

În continuare, voi prezenta rezultatele obținute în extragerea conturilor și reprezentarea prin metodele morfologiei matematice, precum și calcularea seturilor de schelete, comparativ, prin diferite metode: metoda lui Maragos și Schafer [130], metoda lui Maragos și Schafer modificată cu creșterea exponențială a dimensiunii elementului structurant și reprezentarea scheletului condiționată de contur. Aceste metode sunt comparate din punctul de vedere al ratei bit/pixel.

Am implementat toți algoritmi prin utilizarea de elemente structurante diferite, comparând între ele rezultatele obținute. Am ajuns la concluzia că cele mai bune rezultate le-am obținut într-adevăr cu elementele structurante cel mai des menționate în literatura de specialitate: elementul SIMPLU (pătrat de  $3 \times 3$  pixeli) și elementul ROMB.

În continuare voi prezenta rezultatele obținute prin aplicarea algoritmului pe care l-am propus, rezultate apreciate prin prisma ratei de transmisie biți/pixel. Pentru a facilita și aprecierea subiectivă a calității (prin intermediul scorului mediu al opiniilor), voi prezenta și imaginile decodate.

Este prezentată de asemenea și o apreciere mai obiectivă a calității, prin intermediul raportului semnal/zgomot. Totuși, această apreciere nu este foarte relevantă, datorită caracterului aleator al reconstrucției ultimelor patru plane de biți. Totuși, rezultatele testelor subiective sunt mai bune decât cele indicate prin aceste măsuri relative la zgomot, acest lucru datorându-se faptului că metodele morfologiei matematice sunt foarte apropiate de caracteristicile percepției vizuale umane.

### 7.1. Imagini de test

Unele dintre imaginile de test utilizate au fost obținute prin Internet, de la adresa (<ftp://links.uwaterloo.ca:/pub/BragZone/GreySet1.>)

Imaginea ELECTRO, care reprezintă clădirea principală a Facultății de Electronică și Telecomunicații din Universitatea "Politehnica" din Timișoara, a fost capturată la rezoluția de 256×256 pixeli, cu 8 biți/pixel, pe scară de gri, la fel cu toate celelalte imagini de test.

Imaginile de test utilizate au fost alese astfel încât să prezinte o varietate mare a conținutului, de la imagini geometrice regulate (PĂTRAT, CERCURI), la fețe umane (LENA, GIRL) și fotografii (GOLDHILL, ELECTRO).

În cele ce urmează sunt prezentate originalele acestor imagini.



LENA



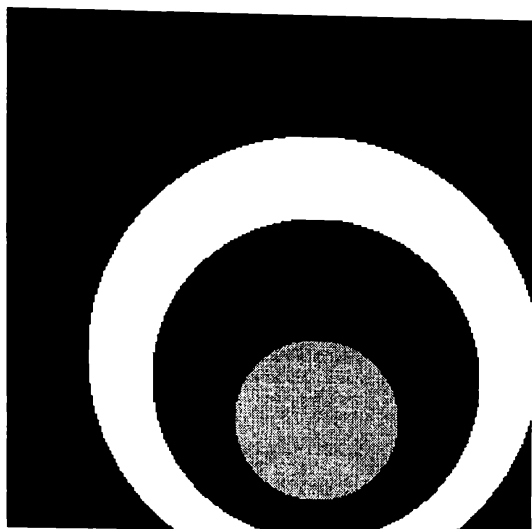
GIRL



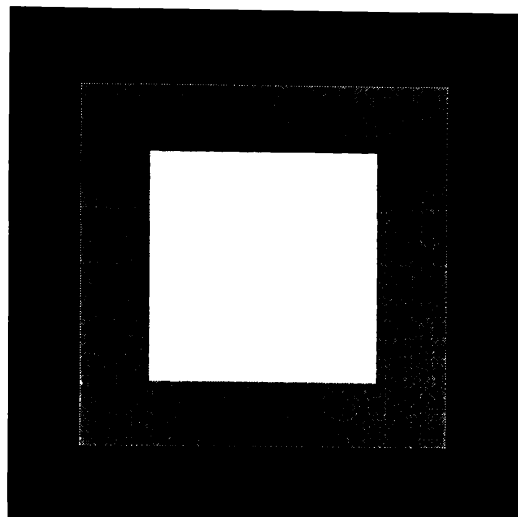
GOLDHILL



ELECTRO



CERCURI



PĂTRAT

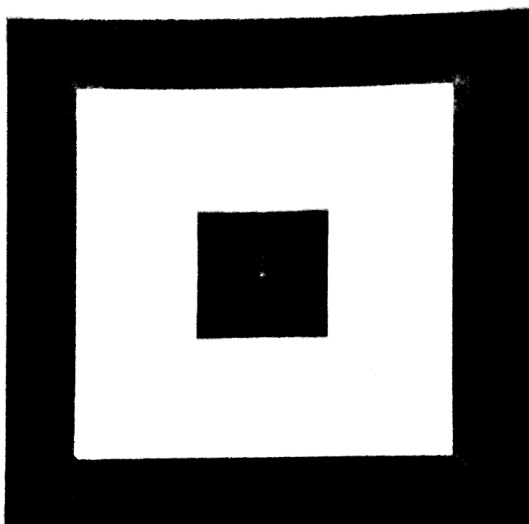
### 7.2. Extragerea morfologică a conturilor și descompunerea în seturi de schelete

Acest paragraf prezintă conturile și seturile de schelete rezultate pentru imaginile de test prezentate anterior. Din motive de spațiu, ilustrarea este prezentată doar pentru un plan de biți (vezi paragraful 6.3). Pentru fiecare imagine de test sunt prezentate comparativ rezultatele obținute cu elementele structurante SIMPLU și ROMB.

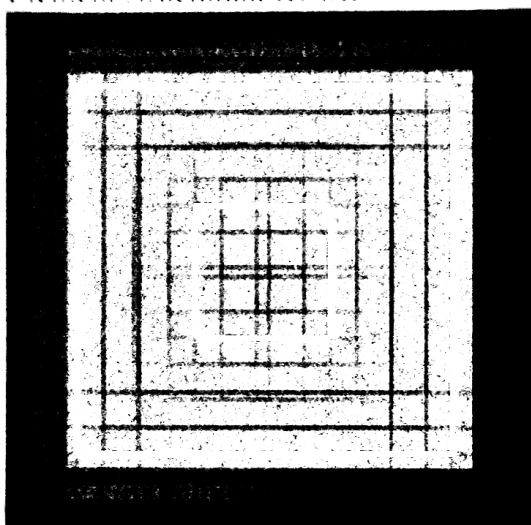
În ceea ce privește conturile obținute, nu se pot observa diferențe semnificative între rezultatele obținute cu cele două tipuri de elemente structurante.

Elementul structurant ideal, din punct de vedere teoretic, pentru obținerea seturilor de schelete, trebuie să fie discul. Datorită dimensiunilor finite care pot fi utilizate pentru realizarea elementului structurant inițial, acesta nu poate aproxima foarte bine forma circulară. Din acest motiv apar diferențe între rezultatele obținute cu cele două tipuri de elemente structurante. Se poate observa că, în general, elementul structurant ROMB dă rezultate mai bune, în special în cazul imaginilor cu formă geometrică regulată.

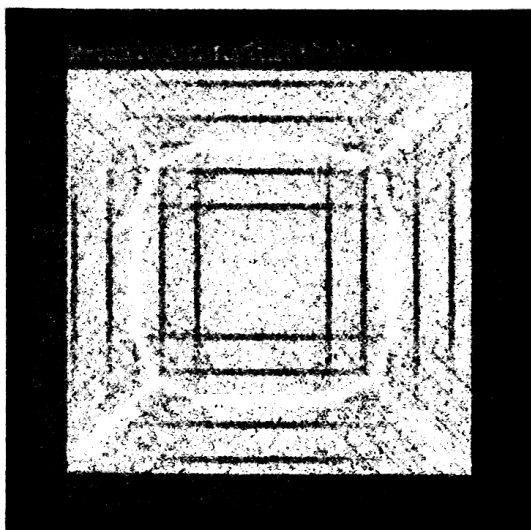
Pentru imaginea de test PĂTRAT, am ilustrat extragerea conturilor și scheletul minimal pentru planul de biți 6.



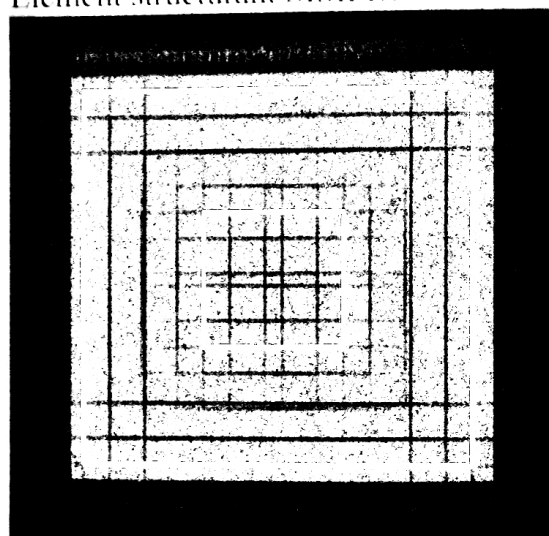
Contur morfologic  
Element structural ROMB



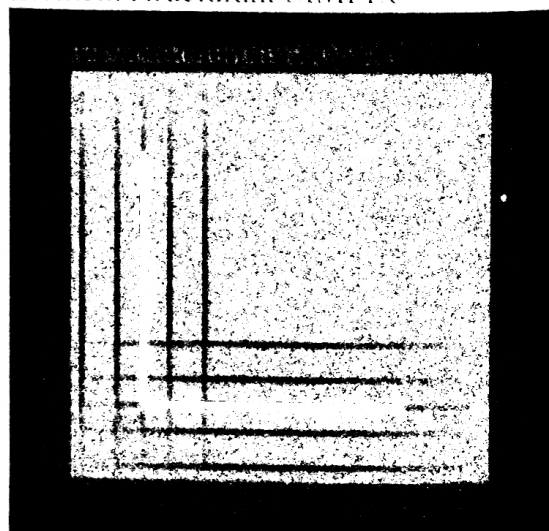
Schelet morfologic  
Element structural ROMB



Element structural SIMPLU



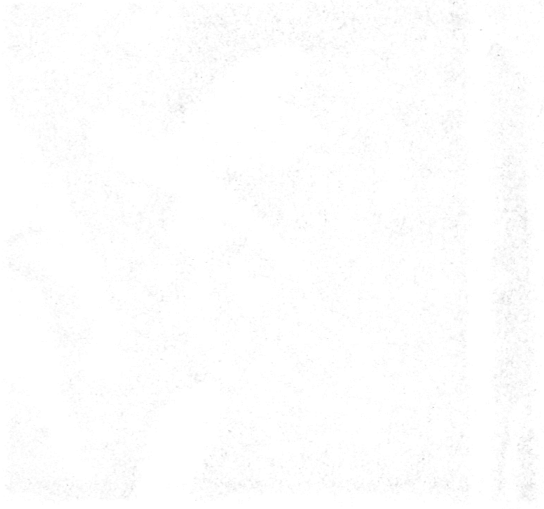
Element structural SIMPLU



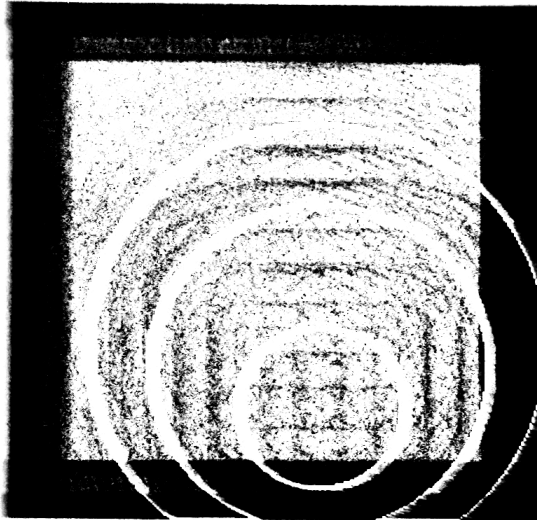
Pentru imaginea de test CERCURI, am ilustrat extragerea contururilor și scheletul minimal pentru planul de biți 7.



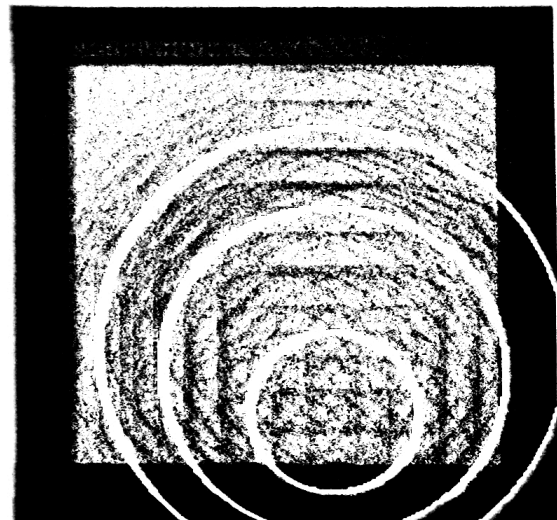
Contur morfologic  
Elemente structurale NCMIS



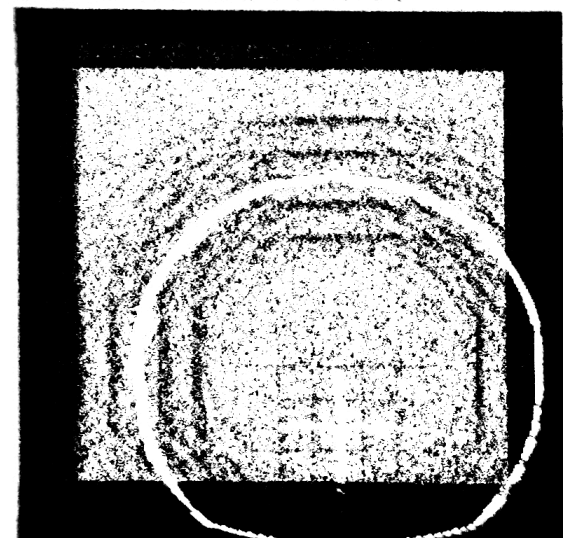
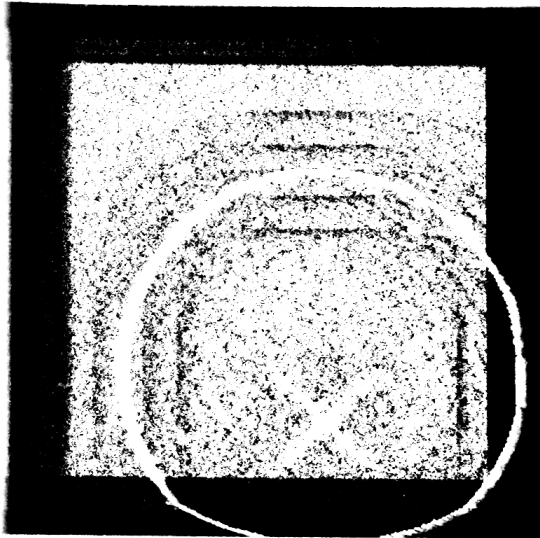
Elemente structurale SIMPLU



Schelet morfologic  
Elemente structurale NCMIS



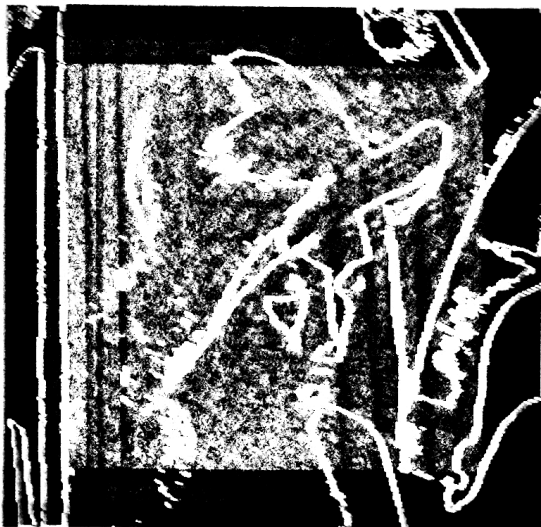
Elemente structurale SIMPLU



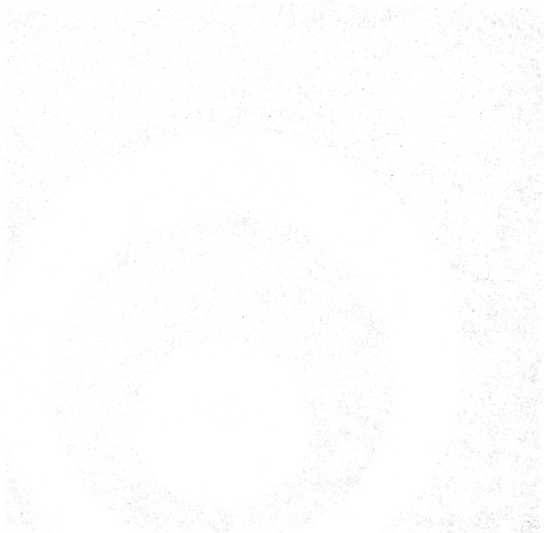
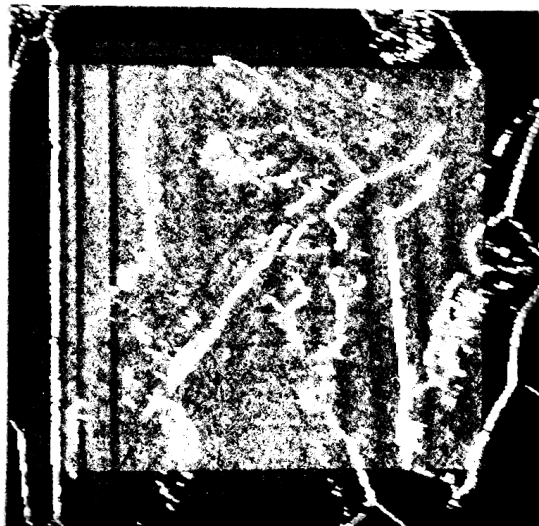
Pentru imaginea de test LENA, am ilustrat extragerea contururilor și scheletului minimal pentru planul de biți 7.



Contururile logice  
Elemente structurale ROMB



Scheletul minim  
Elemente structurale ROMB



Elemente structurale SIMPL



Elemente structurale SIMPL



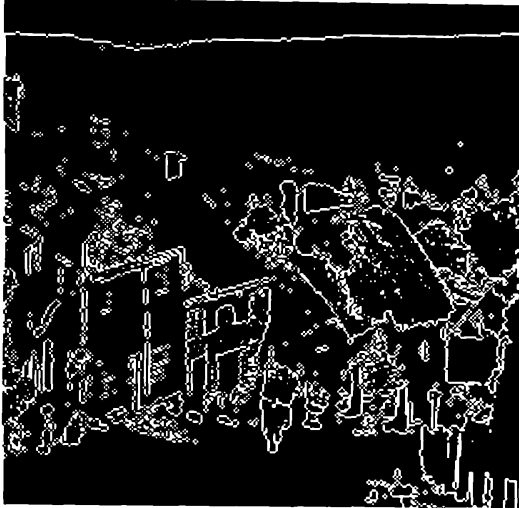


Pentru imaginea de test GOLDHILL, am ilustrat extragerea conturilor și scheletul minimal pentru planul de biți 6.



Contur morfologic:

Element structurant ROMB



Element structurant SIMPLU



Schelet minimal:

Element structurant ROMB



Element structurant SIMPLU

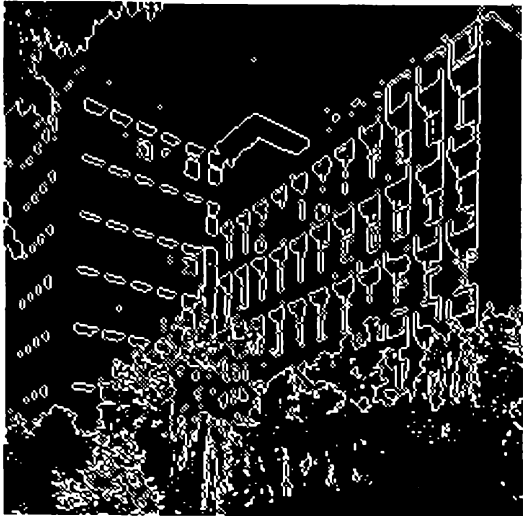


Pentru imaginea de test ELECTRO, am ilustrat extragerea conturilor și scheletul minimal pentru planul de biți 7.



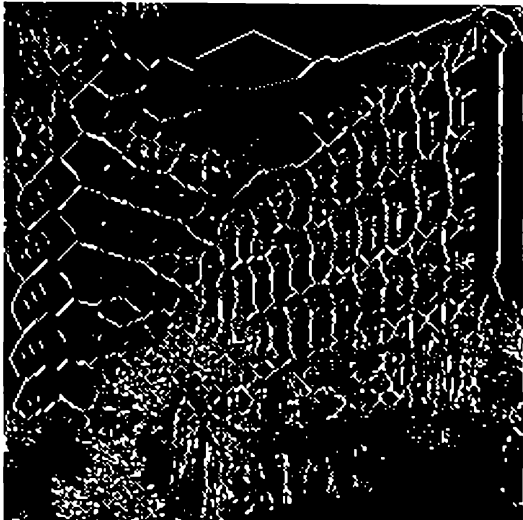
Contur morfologic:

Element structurant ROMB



Schelet minimal:

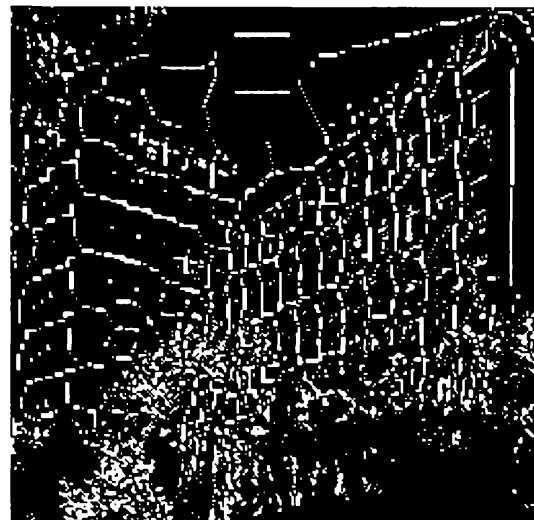
Element structurant ROMB



Element structurant SIMPLU



Element structurant SIMPLU



### 7.3. Rezultatele compresiei

Voi prezenta rezultatele comparative în compresia imaginilor, exprimate prin rata de transmisie biți/pixel, obținute prin implementarea algoritmului pe care l-am propus și a metodei Maragos și Schafer, care este prezentată în literatură ca având cea mai mare eficiență în această clasă de metode. Reamintesc că metoda pe care am propus-o utilizează creșterea exponențială a dimensiunii elementului structurant și reprezentarea schelelelor condiționată de contururi.

Tabelele 7.1 până la 7.10 prezintă numărul de seturi de schelete obținute, în comparație cu algoritmul inițial al lui Maragos și Schafer, precum și rata bit/pixel obținută. Seturile de schelete au fost codate run-length și cu cod Huffman cu lungime variabilă.

Tabelul 7.1.

Imagine LENA, elem. structurant ROMB	Număr schelete algoritm Maragos - Schafer	Număr schelete algoritm propus	Număr biți	Rata biți/pixel
plan 7	20	5	8722	0,13
plan 6	39	6	20298	0,3
plan 5	24	5	14991	0,22
plan 4	10	4	7993	0,12
Total	93	20	52004	0,77

Tabelul 7.2.

Imagine LENA, elem. structurant SIMPLU	Număr schelete algoritm Maragos - Schafer	Număr schelete algoritm propus	Număr biți	Rata biți/pixel
plan 7	18	5	9462	0,14
plan 6	32	5	17669	0,26
plan 5	22	5	15398	0,23
plan 4	9	4	7858	0,11
Total	81	19	50387	0,74

Tabelul 7.3.

Imagine GOLDHILL, elem. structurant ROMB	Număr schelete algoritm Maragos - Schafer	Număr schelete algoritm propus	Număr biți	Rata biți/pixel
plan 7	20	5	13589	0,16
plan 6	3	6	18332	0,27
plan 5	24	5	15854	0,23
plan 4	10	4	12887	0,18
Total	93	20	60662	0,84

Tabelul 7.4.

Imagine GOLDHILL, elem. structurant SIMPLU	Număr schelete algoritm Maragos - Schafer	Număr schelete algoritm propus	Număr biți	Rata biți/pixel
plan 7	18	5	9445	0,14
plan 6	33	6	15854	0,24
plan 5	22	5	15239	0,20
plan 4	9	4	10857	0,16
Total	82	20	51395	0,74

Tabelul 7.5.

Imagine ELECTRO, elem. structurant ROMB	Număr schelete algoritm Maragos - Schafer	Număr schelete algoritm propus	Număr biți	Rata biți/pixel
plan 7	32	5	15179	0,23
plan 6	35	6	15877	0,24
plan 5	14	4	10438	0,12
plan 4	19	5	15187	0,23
Total	100	20	56681	0,86

Tabelul 7.6.

Imagine ELECTRO, elem. structurant SIMPLU	Număr schelete algoritm Maragos - Schafer	Număr schelete algoritm propus	Număr biți	Rata biți/pixel
plan 7	27	5	12649	0,19
plan 6	25	5	12426	0,18
plan 5	18	5	12598	0,19
plan 4	15	4	13097	0,19
Total	85	19	50770	0,75

Tabelul 7.7.

Imagine CERCURI, elem. structurant ROMB	Număr schelete algoritm Maragos - Schafer	Număr schelete algoritm propus	Număr biți	Rata biți/pixel
plan 7	39	6	13461	0,2
plan 6	78	7	23644	0,36
plan 5	36	6	11186	0,17
plan 4	73	7	24971	0,38
Total	216	26	73262	1,11

Tabelul 7.8.

Imagine CERCURI, elem. structurant SIMPLU	Număr schelete algoritm Maragos - Schafer	Număr schelete algoritm propus	Număr biți	Rata biți/pixel
plan 7	29	5	9435	0,14
plan 6	57	6	16894	0,25
plan 5	28	5	13089	0,19
plan 4	56	6	25130	0,38
Total	170	22	63548	0,96

Tabelul 7.9.

Imagine PĂTRAT, elem. structurant ROMB	Număr schelete algoritm Maragos - Schafer	Număr schelete algoritm propus	Număr biți	Rata biți/pixel
plan 7	56	6	16928	0,25
plan 6	40	6	12624	0,19
plan 5	17	5	17751	0,27
plan 4	1	1	18061	0,26
Total	114	18	65364	0,97

Tabelul 7.10.

Imagine PĂTRAT, elem. structurant SIMPLU	Număr schelete algoritm Maragos - Schafer	Număr schelete algoritm propus	Număr biți	Rata biți/pixel
plan 7	56	6	17588	0,26
plan 6	30	5	9116	0,13
plan 5	13	4	18268	0,27
plan 4	1	1	17592	0,26
Total	100	16	62564	0,92

Tabelele 7.11 până la 7.13 prezintă rata biți/pixel obținută pentru unele dintre imaginile de test prezentate, utilizând reprezentarea condiționată a scheletului minimal. Scheletul minimal și contururile au fost codate run-length și Huffman cu lungime variabilă.

Tabelul 7.11.

Imagine LENA	Număr biți de contur	Biți/pixel pt. contur	Număr biți de schelet	Biți/pixel pt. schelet	Număr total de biți	Biți/pixel
ROMB	27469	0,41	18055	0,27	45524	0,68
SIMPLU	49007	0,74	18207	0,27	67214	1,01

Tabelul 7.12.

Imagine GOLDHILL	Număr biți de contur	Biți/pixel pt. contur	Număr biți de schelet	Biți/pixel pt. schelet	Număr total de biți	Biți/pixel
ROMB	25190	0,38	24024	0,36	49214	0,74
SIMPLU	50550	0,77	23938	0,36	74488	1,06

Tabelul 7.13.

Imagine ELECTRO	Număr biți de contur	Biți/pixel pt. contur	Număr biți de schelet	Biți/pixel pt. schelet	Număr total de biți	Biți/pixel
ROMB	20369	0,31	17983	0,27	38352	0,58
SIMPLU	41480	0,63	19086	0,29	60566	0,92

La aceste valori obținute pentru rata de bit trebuie adăugată contribuția celor patru plane de bit cel mai puțin semnificative (a se vedea paragraful 6.7). Această contribuție nu depinde de imaginea de test utilizată, fiind după cum urmează:

- planul de biți 0: 2 bytes = 16 biți
- planul de biți 1: 2 bytes  $\times$  4 blocuri = 64 biți
- planul de biți 2: 2 bytes  $\times$  16 blocuri = 256 biți
- planul de biți 3: 2 bytes  $\times$  64 blocuri = 1024 biți (pentru imagini de  $256 \times 256$  pixeli)

Contribuția totală pentru reprezentarea acestor plane de biți este de 1360 biți, sau  $1360/65536 = 0,02$  biți/pixel. Această valoare trebuie adunată la toate valorile prezentate în tabelele de mai sus.

#### 7.4. Aprecierea calității

În cele ce urmează voi face evaluarea calității imaginilor reconstruite, prin intermediul rapoartului semnal de vârf / zgomot (PSNR - peak-signal/noise ratio) și a erorii medii pătratice (RMSE - root mean square error). Pentru fiecare imagine am utilizat două elemente structurante, calculând pentru fiecare din ele PSNR și RMSE. Rezultatele obținute sunt prezentate în tabelele 7.14 până la 7.21.

O comparație mai sugestivă este realizată prin reprezentările grafice prezentate în figurile 7.1 până la 7.4.

Tabelul 7.14.

Imagine LENA, elem. structurant ROMB	Codare doar plan bit 7	Codare doar plane bit 7 și 6	Codare doar plane bit 7,6 și 5	Codare doar plane bit 7, 6, 5 și 4	Codare toate planele de biți
RMSE	123,45	67,23	36,54	12,62	7,14
PSNR	6,30	11,57	16,87	26,10	31,05
Rată bit/pixel	0,13	0,43	0,65	0,77	0,79

Tabelul 7.15.

Imagine LENA, elem. structurant SIMPLU	Codare doar plan bit 7	Codare doar plane bit 7 și 6	Codare doar plane bit 7,6 și 5	Codare doar plane bit 7, 6, 5 și 4	Codare toate planele de biți
RMSE	123,45	67,23	36,54	12,62	7,14
PSNR	6,30	11,57	16,87	26,10	31,05
Rată bit/pixel	0,14	0,40	0,63	0,74	0,76

Tabelul 7.16.

Imagine ELECTRO, elem. structurant ROMB	Codare doar plan bit 7	Codare doar plane bit 7 și 6	Codare doar plane bit 7,6 și 5	Codare doar plane bit 7, 6, 5 și 4	Codare toate planele de biți
RMSE	128,75	66,43	29,25	17,12	6,23
PSNR	5,93	11,68	18,80	23,46	32,24
Rată bit/pixel	0,23	0,47	0,59	0,86	0,88

Tabelul 7.17.

Imagine ELECTRO, elem. structurant SIMPLU	Codare doar plan bit 7	Codare doar plane bit 7 și 6	Codare doar plane bit 7,6 și 5	Codare doar plane bit 7, 6, 5 și 4	Codare toate planele de biți
RMSE	128,75	66,43	29,25	17,12	6,23
PSNR	5,93	11,68	18,80	23,46	32,24
Rată bit/pixel	0,19	0,37	0,56	0,75	0,77

Tabelul 7.18.

Imagine PĂTRAT, elem. structurant ROMB	Codare doar plan bit 7	Codare doar plane bit 7 și 6	Codare doar plane bit 7,6 și 5	Codare doar plane bit 7, 6, 5 și 4	Codare toate planele de biți
RMSE	111,16	45,13	24,63	11,74	2,24
PSNR	7,21	15,04	20,30	26,73	41,12
Rată bit/pixel	0,25	0,44	0,71	0,97	0,99

Tabelul 7.19.

Imagine PĂTRAT, elem. structurant SIMPLU	Codare doar plan bit 7	Codare doar plane bit 7 și 6	Codare doar plane bit 7,6 și 5	Codare doar plane bit 7, 6, 5 și 4	Codare toate planele de biți
RMSE	111,16	45,13	24,63	11,74	2,24
PSNR	7,21	15,04	20,30	26,73	41,12
Rată bit/pixel	0,26	0,39	0,66	0,92	0,94

Tabelul 7.20.

Imagine CERCURI, elem. structurant ROMB	Codare doar plan bit 7	Codare doar plane bit 7 și 6	Codare doar plane bit 7,6 și 5	Codare doar plane bit 7, 6, 5 și 4	Codare toate planele de biți
RMSE	116,09	53,12	28,17	13,85	3,16
PSNR	6,83	13,62	19,13	25,30	38,13
Rată bit/pixel	0,2	0,56	0,73	1,11	1,13

Tabelul 7.21.

Imagine CERCURI, elem. structurant SIMPLU	Codare doar plan bit 7	Codare doar plane bit 7 și 6	Codare doar plane bit 7,6 și 5	Codare doar plane bit 7, 6, 5 și 4	Codare toate planele de biți
RMSE	116,09	53,12	28,17	13,85	3,16
PSNR	6,83	13,62	19,13	25,30	38,13
Rată bit/pixel	0,14	0,39	0,58	0,96	0,98

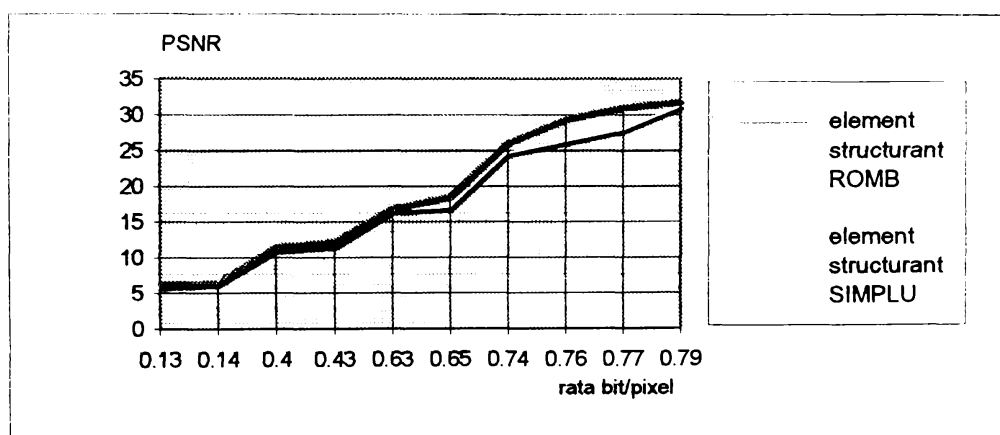


Fig. 7.1. Reprezentarea grafică a Tabelor 7.14 și 7.15 (imagine LENA)

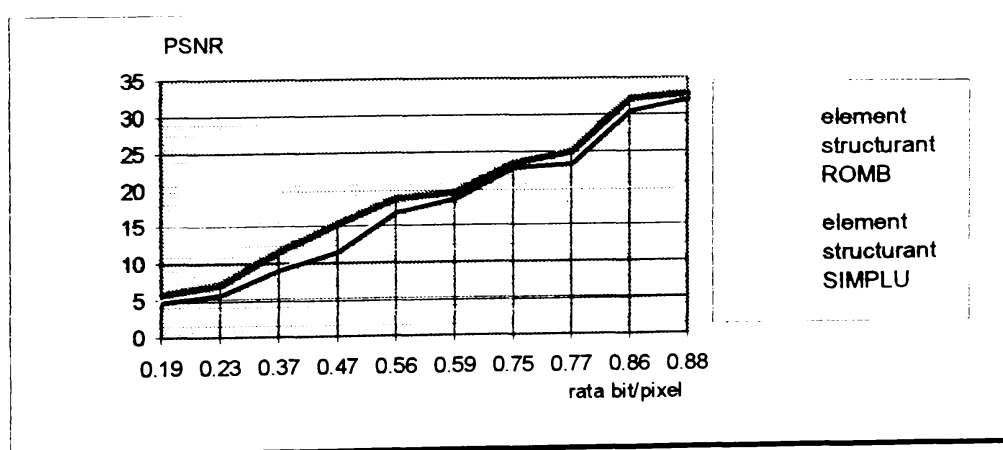


Fig. 7.2. Reprezentarea grafică a Tabelor 7.16 și 7.17 (imagine ELECTRO)

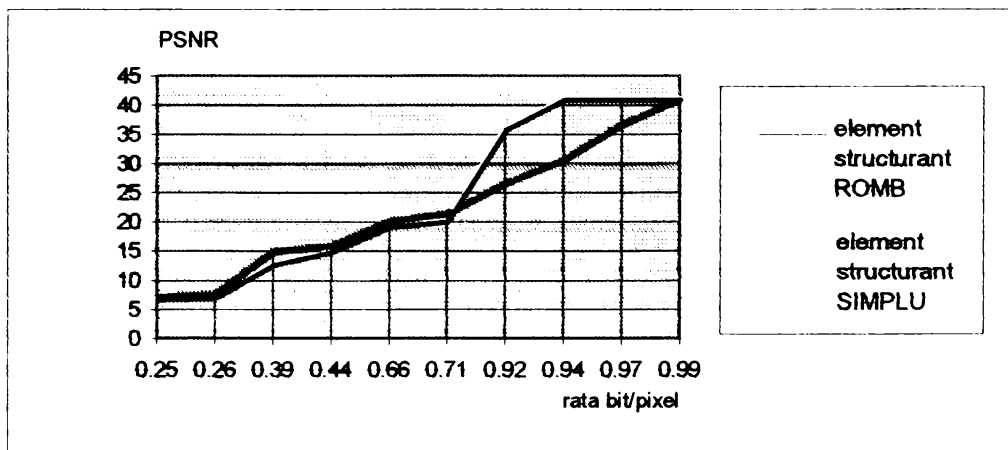


Fig. 7.3. Représentarea grafică a Tabelor 7.18 și 7.19 (imagine PĂTRAT)

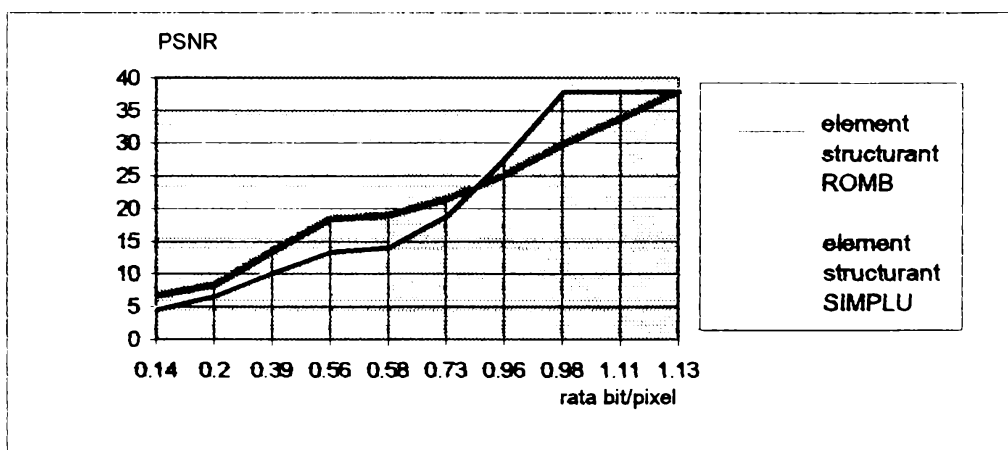


Fig. 7.4. Représentarea grafică a Tabelor 7.20 și 7.21 (imagine CERCURI)

Prin compararea rezultatelor obținute, am putut obține anumite concluzii referitoare la compresia imaginilor cu algoritmul propus:

- raportul semnal/zgomot obținut este dependent de conținutul imaginii. Pentru imagini similare ca și conținut (fețe umane, peisaje fotografice, forme regulate) am obținut aproximativ aceleași valori ale PSNR.
- rapoartele semnal/zgomot obținute depind mult de acuratețea cu care sunt approximate cele 4 plane de biți mai puțin semnificative. Pentru imagini care reprezintă forme regulate, am obținut un raport mai bun datorită proprietăților statistice constante ale fiecărui plan de biți.
- în toate cazurile, calitatea imaginii percepută vizual (conform paragrafului 5.3) este mai bună decât cea așteptată pe baza raportului semnal/zgomot calculat, datorită caracteristicilor geometrice ale metodelor morfologiei matematice, care sunt foarte apropiate de caracteristicile percepției vizuale umane.



### 7.5. Imaginile de test decodate

În cele ce urmează voi prezenta imaginile de test decodate pentru a facilita aprecierea subiectivă a calității, pe baza criteriului scorului mediu al opiniilor, invocat în paragraful 5.3. Nu pot fi detectate diferențe esențiale între imaginile decodate și cele originale, tocmai datorită faptului că metodele morfologiei matematice sunt foarte apropiate de caracteristicile percepției vizuale umane.

Imagine LENA:



Original



Reconstruită

Imagine GOLDHILL:



Original

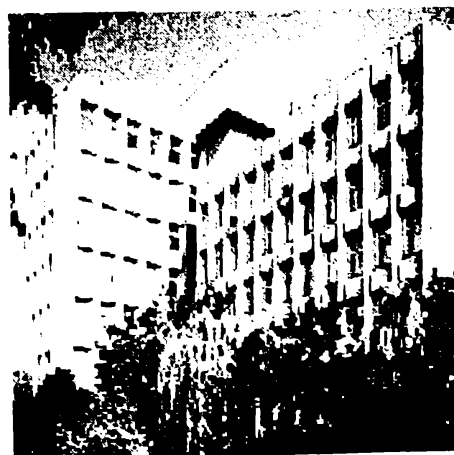


Reconstruită

Imagine ELECTRO:



Original



Reconstruită

### 7.6. Concluzii și perspective

În concluzie, pot spune că metodele morfologiei matematice sunt foarte potrivite pentru tehnicile de codare orientate pe formă. Aceste metode conduc la rezultate mai bune în comparație cu tehnicile tradiționale de codare, bazate pe teoria reprezentării. Pe de altă parte, deoarece morfologia matematică a fost dezvoltată ca o abordare geometrică a prelucrării semnalelor, tehnicile care înglobează metode morfologice sunt mai potrivite pentru codarea imaginilor în conformitate cu caracteristicile percepției vizuale umane. Din acest motiv, pentru toate imaginile de test utilizate, rezultatele pe care le-am obținut sunt mai bune din punct de vedere al percepției vizuale decât s-ar aștepta în conformitate cu raportul semnal/zgomot calculat.

Rezultatele pe care le-am obținut, nu numai că sunt comparabile cu cele raportate în literatura de specialitate, dar sunt chiar mai bune. Acest lucru mă încurajează să cred că acest domeniu de cercetare relativ nou în compresia imaginilor este unul foarte promițător.

După cum am menționat deja în paragraful 6.8, algoritmul pe care l-am propus are avantajul de a fi adecvat transmisiei și refacerii progresive a imaginilor, care sunt de mare interes pentru un număr tot mai mare de aplicații.

Consider că rezultatele obținute pot constitui baza unor cercetări în domeniul codării imaginilor și secvențelor video la rate foarte mici de bit. În acest sens consider de interes combinarea rezultatelor obținute cu tehnicile de codare din generația a doua (după clasificarea lui Murat Kunt [108]), care implică segmentarea multirezoluție a imaginilor, compensarea mișcării și reprezentarea imaginii prin contur-textură.

## Concluzii

Compresia digitală a imaginilor este un domeniu de mare actualitate, care a cunoscut o dezvoltare impresionantă în cursul ultimelor decenii, ea constituind elementul cheie al unor noi tehnologii informatice în diverse aplicații: multimedia, efecte speciale în televiziune și cinematografie, grafică pe calculator, sisteme de videoconferință, teledetecție, medicină etc.

Teza prezentată de autor se înscrie în această problematică, constituindu-se într-o contribuție la înțelegerea aspectelor importante ale tehnologiilor și metodelor matematice specifice compresiei imaginilor utilizate în prelucrarea și transmiterea numerică a imaginilor. Teza urmărește realizarea unei abordări sistematice a domeniului, atât din punct de vedere teoretic cât și al implementării practice a algoritmilor de compresie a imaginilor. Pe tot parcursul ei, teza urmărește prezentarea contribuțiilor autorului în ceea ce privește abordarea și evaluarea unor metode cunoscute din literatura de specialitate, propunerea unor noi algoritmi de compresie a imaginilor și evaluarea acestora din punct de vedere al eficienței compresiei, comparativ cu metodele clasice. În centrul preocupărilor autorului, referitoare la elaborarea de noi algoritmi, au stat metodele moderne ale morfologiei matematice, mai puțin utilizate până în prezent pentru abordarea aplicațiilor de codare a imaginilor. Teza se constituie într-o dovadă a eficienței acestor metode, dovedind potențialul ridicat al morfologiei matematice și deschizând perspective noi de cercetare în domeniu.

Una dintre sarcinile cele mai dificile a fost aceea de a detecta și sistematiza direcțiile de dezvoltare ale domeniului compresiei imaginilor și cele posibile de urmat în viitor. Această sarcină a implicat analiza unui material bibliografic deosebit de bogat (aproape 300 de titluri). Printre lucrările citate în bibliografie figurează și articole ale autorului, fie deja publicate, fie acceptate spre publicare, care reflectă preocupările autorului pe durata pregătirii tezei și contribuțiile acestuia la studiul domeniului abordat.

În cele ce urmează voi face o sumară recapitulare a contribuțiilor aduse pe parcursul tezei.

În capitolul 2 autorul examinează pe scurt aspectele teoretice și practice ale procesului de compresie a imaginilor. Se propune un model general pentru procesul de compresie-decompresie, examinând totodată câteva noțiuni de teoria informației și rolul lor în stabilirea limitelor fundamentale în reprezentarea imaginilor.

În continuare sunt prezentate în mod sistematic principalele tehnici utilizate în compresia imaginilor, care sunt împărțite în două mari categorii: metode reversibile și metode nereversibile. Totodată se face o discuție privitoare la domeniile de aplicație pentru care sunt adecvate fiecare dintre aceste metode. Metodele din prima categorie sunt utile în principal în arhivarea imaginilor, ele permițând ca o imagine să fie compresată și decompresată fără pierdere de informație. Metodele din cea de-a doua categorie asigură un nivel mai înalt de compresie, dar implică o anumită pierdere de informație la reproducerea imaginii, fiind din acest motiv utilizate în aplicații ca: televiziune radiodifuzată, transmisii de videoconferințe și facsimil.

Deoarece principalul domeniu de cercetare al autorului a fost constituit de utilizarea unor metode ale morfologiei matematice în codarea imaginilor, metode care

fac apel în mod intensiv la proprietățile geometrice ale imaginilor, s-a impus prezentarea în capitolul 3 a tehnicilor de reprezentare a formei și conturilor. Deși specifice, în general, aplicațiilor de analiză a imaginilor, aceste tehnici pot sta la baza unor metode avansate de codare a imaginilor. În acest capitol contribuțiile originale ale autorului se concretizează într-o abordare sistematică a diferitelor tehnici cunoscute pentru reprezentarea regiunilor, cu sublinierea celor care își pot găsi aplicabilitatea în domeniul compresiei imaginilor. În contextul segmentării imaginilor, autorul prezintă o clasificare originală a codării partițiilor, pornind de la necesitatea determinării acelor descriptori care pot conduce la o codare reversibilă a imaginilor. O altă contribuție originală constă în definirea așa-numitelor *coduri lanț run-length*, o versiune modificată a codurilor lanț propuse de Freeman, care ține cont în notația sa formală și de lungimea fiecărei muchii. Deși utilizate, ca principiu, și în alte articole publicate în reviste de specialitate, nu a fost găsită o prezentare teoretică a acestor coduri, motiv pentru care autorul și-a definit propria reprezentare pe care o utilizează în capitolul 4 pentru deducerea unor proprietăți teoretice.

În capitolul 4 este realizată o prezentare teoretică a principalelor concepte ale morfologiei matematice. Autorul definește operatorii morfologici binari de bază: dilatarea, erodarea, deschiderea, închiderea, transformarea "hit-or-miss", prezentând totodată și proprietățile generale ale acestora. Din această prezentare sunt apoi derivați algoritmi morfologici de bază, care sunt considerați potențial utili pentru abordarea aplicațiilor de codare a imaginilor: extragerea conturilor, umplerea regiunilor, dilatarea condiționată, extragerea componentelor conectate, înfășurătoarea convexă, obținerea scheletului, subțierea, îngroșarea și curățarea, punându-se accentul pe elementele practice de implementare a acestora. În continuare, sunt introduse noțiunile de morfologie pe scară de gri, necesare pentru realizarea compresiei imaginilor, împreună cu principalele aplicații dezvoltate în literatura de specialitate: netezirea morfologică, gradientul morfologic, segmentarea texturilor, transformarea top-hat și distribuția obiectelor după mărime. Este abordată în continuare și problema morfologiei color.

În acest capitol autorul aduce o serie de contribuții teoretice originale, legate de utilizarea morfologiei matematice în compresia imaginilor. După ce sunt evidențiate proprietățile morfologiei matematice care fac posibilă utilizarea sa în compresia imaginilor și se face o trecere în revistă a stadiului actual în domeniu, autorul scoate în evidență caracterul primordial al alegerii elementului structurant pentru succesul strategiei utilizate. Pornind de la această constatare, autorul inițiază o discuție asupra formei și dimensiunilor elementului structurant care trebuie utilizat. Sunt determinate limitările impuse de structura hardware a mașinilor care realizează prelucrările morfologice, care conduc la necesitatea utilizării unor elemente structurante cu dimensiunile maxime de  $3 \times 3$  pixeli, subliniindu-se totodată faptul că dimensiuni mai mici pot conduce la un câștig în viteza de execuție. Din acest motiv, în paragraful 4.5 autorul abordează problema descompunerii elementului structurant, problemă înrudită cu cea a descompunerii domeniilor conexe, deja cunoscută din literatura de specialitate. Pe parcursul prezentării stadiului actual în descompunere, autorul demonstrează că rezultatele raportate de Park și Chin [168] sunt aplicabile de fapt doar pentru formele concave, nu și pentru cele convexe. Ca strategie generală de descompunere, autorul propune realizarea descompunerii unui element concav prin exprimarea sa ca reuniune a unui număr de poligoane convexe, furnizând totodată și formulele de calcul pentru implementarea dilatării și erodării unor forme arbitrare în acest mod.

În continuare, autorul propune utilizarea codurilor lanț run-length pentru realizarea descompunerii. În acest scop el deduce un set de 6 proprietăți interesante ale reprezentării domeniilor convexe prin aceste coduri, care sunt deosebit de utile pentru calculul dilatării și erodării. Pentru evitarea unor posibile ambiguități de reprezentare este introdusă o convenție în ceea ce privește alegerea punctului inițial și a direcției de parcurgere a conturului, fără ca acest lucru să micșoreze gradul de generalitate al metodei propuse.

În continuare este prezentată o interpretare grafică a dilatării și erodării domeniilor convexe, care permite determinarea rapidă a coordonatelor originii domeniului dilatat (erodat) în funcție de coordonatele originii domeniului inițial și cele ale elementului structurant, precum și determinarea rapidă a lungimii muchiilor după fiecare direcție în funcție de cele ale muchiilor domeniului original și ale elementului structurant.

În paragraful 4.5.3.3 autorul dă o interpretare problemei descompunerii, prin intermediul reprezentării domeniilor convexe prin coduri lanț de tip run-length. Se introduce o soluție constructivă pentru descompunere și se demonstrează că este necesar un set de doar 12 elemente de bază pentru realizarea descompunerii, și nu de 13 elemente așa cum se prezintă în bibliografia studiată. Este propus un set redus de 12 elemente structurante de bază care pot fi utilizate pentru descompunerea oricărui domeniu convex și se demonstrează că acest set redus este capabil să dea rezultate similare cu cele furnizate de setul tradițional de 13 elemente prezentat în literatură ([168], [169], [285], [292]). În timpul demonstrației, este propusă și o metodă grafică simplă pentru realizarea descompunerii.

În capitolul 5 se face o analiză a modalităților de apreciere a calității imaginilor și performanțelor tehnicilor de compresie. Autorul sintetizează cerințele de calitate pentru informația video și măsurile cantitative și subiective care pot fi utilizate pentru caracterizarea acestei calități. Este prezentată de asemenea și teoria rată-distorsiune deoarece ea furnizează o informație teoretică excelentă relativă la calitatea imaginilor codate și la reducerea redundanței.

În capitolul 6 este prezentată implementarea algoritmilor propuși de autor pentru compresia imaginilor prin utilizarea metodelor morfologiei matematice.

Autorul a implementat mai întâi modulele de prelucrare morfologică de bază: operatorii de dilatare, erodare, deschidere și închidere, atât în variantă binară cât și în variantă pe scară de gri. Toți operatorii morfologici de bază au fost implementați sub forma unor module software realizate în limbaj C, prezentate în paragraful 6.2.

Începând cu paragraful 6.3 este descris algoritmul propus de autor pentru compresia imaginilor pe scară de gri și modul său de implementare. Algoritmul original utilizează o descompunere pe plane de biți, care sunt apoi codate prin metode diferite, pentru a obține o eficiență mai mare. Cele mai semnificative 4 plane de biți sunt codate fără pierderi, deoarece ele conțin cea mai importantă informație cuprinsă în imaginea respectivă. Autorul a testat două posibilități de codare a acestor plane și a efectuat o comparație care este prezentată în paragraful 6.5. Prima posibilitate constă în utilizarea codării conturilor cu ajutorul codurilor lanț, iar cea de-a doua constă în utilizarea reprezentării unei imagini prin scheletul său. S-a constatat că utilizarea scheletelor poate conduce la rezultate mai bune pe o clasă mai largă de imagini. Metoda utilizată este similară cu cea propusă de Maragos și Schafer pentru aplicații binare [130], dar utilizează un element structurant cu creștere exponențială a dimensiunii. Pentru minimizarea setului de schelete rezultate s-a utilizat aceeași metodă pe care au propus-o Maragos și Schafer. Reuniunea seturilor de schelete obținute a fost minimizată în

continuare prin așa numita minimizare a scheletelor condiționată de contururi. Pentru codarea în vederea transmiterii setului de 4 schelete rezultate în urma minimizării, s-a utilizat o codare cu lungime variabilă.

Pentru ultimele 4 plane, mai puțin semnificative, autorul a propus un algoritm de codare diferit. Ținând cont de faptul că aceste plane poartă puțină informație semnificativă, s-a dorit obținerea unui foarte bun raport de compresie pentru aceste plane. Pentru planul 0 s-a determinat o densitate de probabilitate pentru apariția pixelilor albi/negri pe suprafața întregului plan, urmând ca acesta să fie singurul parametru transmis pentru caracterizarea planului. În cursul procesului de decompresie a imaginii, se generează local o imagine aleatoare cu aceeași densitate de probabilitate, care este apoi adăugată imaginii reconstruite. Pentru planele 1, 2 și 3, nefiind de așteptat o densitate de probabilitate quasi-constantă pe toată suprafața planului, s-a efectuat o segmentare a acestor plane în blocuri de dimensiuni  $1/4$ ,  $1/8$ , respectiv  $1/16$ . S-a calculat apoi același parametru ca și în cazul planului 0 pentru fiecare bloc, acești parametri fiind transmiși pentru caracterizarea planului.

Autorul a dezvoltat două programe în C pentru descompunerea unei imagini de intrare pe scară de gri în planele sale de biți (utilizând reprezentarea în cod Gray), respectiv pentru reconstrucția imaginii inițiale din reprezentarea lor prin planele de biți corespunzătoare. Pentru a avea posibilitatea efectuării unor comparații, au fost dezvoltate și variante ale acestor programe care utilizează reprezentarea binară, demonstrându-se eficiența mai mare a reprezentării în cod Gray. În acest scop a fost necesară inclusiv implementarea unor algoritmi de conversie din cod binar în cod Gray și invers.

Pentru codarea celor mai semnificative 4 plane de biți a fost implementat algoritmul original propus de autor și descris mai sus. Pentru a putea avea o bază de comparație pentru metoda propusă de autor în această teză, a fost implementat și algoritmul propus de Maragos și Schafer. Implementarea acestor algoritmi s-a realizat prin două programe C, care au fost denumite **skel.c** și **skelref.c**. Primul program calculează  $N$  subseturi de schelete ale unei imagini binare de intrare, subseturile rezultate fiind înscrise în  $N$  fișiere separate cu numele **skel0**, **skel1**, ..., **skelN**. Cel de-al doilea program realizează refacerea imaginii inițiale din cele  $N$  subseturi de schelete.

Un avantaj al algoritmului propus de autor constă în posibilitatea de a fi utilizat în sisteme de transmisie și stocare progresivă a imaginii, lucru de mare importanță, în special pentru aplicațiile de arhivare. Modul de manifestare este similar cu cel al aplicațiilor Internet. În timpul în care se decodează doar partea cea mai relevantă a informației din planele de biți, utilizatorul poate localiza imaginea recepționată și poate decide dacă procesul de decodare trebuie continuat sau se poate trece la căutarea unei noi imagini. Dacă se dorește acest lucru, atunci rezoluția imaginii obținute poate fi crescută gradat prin decodarea succesivă a tuturor celor patru plane de biți mai semnificative și a unu până la patru plane de biți mai puțin semnificative.

Rezultatele obținute după implementare pentru diferite clase de imagini sunt prezentate în capitolul 7. Imaginile de test utilizate pentru evaluarea performanțelor algoritmilor propuși de autor sunt imagini standard, pentru a asigura posibilitatea comparării rezultatelor cu cele furnizate de algoritmi similari descriși în literatura de specialitate.

Autorul prezintă rezultatele obținute în extragerea conturilor și reprezentarea prin metodele morfologiei matematice, precum și calcularea seturilor de schelete, comparativ, prin diferite metode: metoda lui Maragos și Schafer [130], metoda lui Maragos și Schafer modificată cu creșterea exponențială a dimensiunii elementului

structurant și reprezentarea scheletului condiționată de contur. Aceste metode sunt comparate din punctul de vedere al ratei bit/pixel.

Toți algoritmi au fost implementați prin utilizarea de elemente structurante diferite, comparând între ele rezultatele obținute. S-a confirmat concluzia că cele mai bune rezultate se obțin într-adevăr cu elementele structurante cel mai des menționate în literatura de specialitate: elementul SIMPLU (pătrat de  $3 \times 3$  pixeli) și elementul ROMB.

Rezultatele obținute prin aplicarea algoritmului propus de autor sunt apreciate prin prisma ratei de transmisie biți/pixel. S-au obținut valori cuprinse între 0,60 și 1,08 biți/pixel, ceea ce înseamnă o compresie de circa 1:8 până la 1:14, în condițiile unei foarte bune calități a imaginii reconstruite. Comparativ cu algoritmi similari (de tipul celor propuși de Maragos și Schafer), se constată un câștig de aproximativ 25% în ceea ce privește rata de compresie, la o calitate similară a reconstrucției, demonstrată prin aprecierea subiectivă a calității (prin intermediul scorului mediu al opiniilor) și prin prezentarea imaginilor decodate.

Este prezentată de asemenea și o apreciere a calității prin intermediul raportului semnal/zgomot. Totuși, această apreciere nu este foarte relevantă, datorită caracterului aleator al reconstrucției ultimelor patru plane de biți. Totuși, rezultatele testelor subiective sunt mai bune decât cele indicate prin aceste măsuri relative la zgomot, acest lucru datorându-se faptului că metodele morfologiei matematice sunt foarte apropiate de caracteristicile percepției vizuale umane.

La toate aceste avantaje prezentate de metodele propuse și implementate de autor, trebuie adăugat faptul că ele sunt metode progresive care permit codarea multirezoluție a imaginilor, element crucial în abordarea unor aplicații de prelucrare și transmitere numerică a imaginilor.

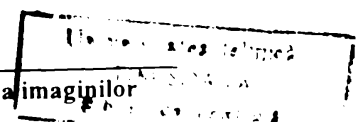
Rezultatele obținute în cadrul acestei teze de doctorat îl îndreptătesc pe autor să considere că morfologia matematică are un potențial ridicat în abordarea problemei compresiei imaginilor. Din acest motiv, prezenta teză este considerată doar o etapă în preocupările mai generale ale autorului în domeniul prelucrării imaginilor și a aplicațiilor multimedia. Direcțiile de perspectivă sunt considerate cele legate de abordarea compresiei secvențelor de imagini, cu compensarea mișcării, precum și cele legate de creșterea eficienței cu care se înglobează secvențele video în aplicațiile multimedia mai complexe.

## Bibliografie

1. A. J. Abrantes, J. S. Marques: "A Class of Constrained Clustering Algorithms for Object Boundary Extraction", *IEEE Transactions on Image Processing*, vol. 5, no. 11, Nov. 1996, pp. 1507-1521
2. K. Aizawa, T. S. Huang: "Model-Based Image Coding: Advanced Video Coding Techniques for Very Low Bit-Rate Applications", *Proceedings of the IEEE*, vol. 83, no. 2, Feb. 1995, pp. 259-271
3. D. Anastassiou: "Digital Television", *Proceedings of the IEEE*, vol. 82, nr.4, Apr. 1994, pp. 510-519
4. K. E. Batcher, "Bit-serial Parallel Processing Systems", *IEEE Transactions on Computers*, vol. C-31, 1982, pp. 377-384
5. F. Bellifemine, R. Picco: "Video Signal Coding with DCT and Vector Quantization", *IEEE Transactions on Communications*, vol. 42, no. 2/3/4, Febr./March/April 1994, pp. 200-207
6. S. Benedetto, E. Biglieri, V. Castellani: *Digital Transmission Theory*, Prentice Hall, Englewood Cliffs, New Jersey, 1987
7. K. B. Benson, D. G. Fink: *HDTV. Advanced Television for the 1990s*, McGraw-Hill, New York, 1991
8. R. Bernardini: "An Efficient Algorithm to Find Lattice Chains", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, no. 5, Oct. 1996, pp. 496-505
9. P. Bhattacharya, K. Qian, X. Lu: "An Algebraic Approach for Morphological Operations on 2D and 3D Images", *Pattern Recognition*, vol. 26, no. 12, 1993, pp. 1785-1796
10. S. S. Bisnas and A. K. Ray, "Region Merging in 3-D Images Using Morphological Operators", *Pattern Recognition Letters*, vol. 14, no. 1, January 1993, pp. 23-31
11. H. Blum: "A Transformation for Extracting New Descriptors of Shape", *Models for the Perception of Speech and Visual Forms*, Ed. Cambridge, 1967
12. Z. Bojković: "Upper Bound on Signal/Quantizing Noise Ratio for Television DPCM Systems", *Electronics Letters*, Jan. 1975, vol.11, no.1, pp. 29-30
13. Z. Bojković: "A Bound on Signal-to-Quantizing Noise Ratio in Linear Transform Picture Coding Systems", *Archiv für Elektronik und Übertragungstechnik, Electronics and Communications*, Band 32, 1978, pp. 430-433
14. Z. Bojković: "Optimization of Entropy Coded Uniform Quantizer in High Bit Rate Region", *Signal Processing III: Theories and Applications*, EURASIP, 1986, pp. 57-60
15. Z. Bojković: "A Method for Incorporating the human Visual System Model in Rate Adaptive Form", *IEEE-SP International Symposium on Time-Frequency and Time-Scale Analysis*, Victoria, Canada, Oct. 4-6, 1992
16. Z. Bojković: "Digital HDTV System Compression", *Journal on Communications*, vol. XLV, May-June 1994, pp. 2-10



17. Z. Bojković: "Image Decomposition and Compression in Digital Multimedia Systems", *LXth International Conference on Signal Processing Applications & Technology ICSPAT'95*, Boston, Massachusetts, Oct. 24-26, 1995, pp. 940-944
18. Z. Bojković: "Image Quality Estimation in Subband Coding Techniques Based on Human Visual System", *Proceedings of the 1996 International Conference on Communication Technology ICCT'96*, Beijing, China, May 5-7, 1996, pp. 651-653
19. Z. Bojković, D. Milanović, A. Samčović: "On a Bound on Signal-to-Noise Ratio in Subband Coding of Gaussian Image Process", *Proceedings of the 5<sup>th</sup> International Conference CAIP'93*, Budapest, Hungary, Sept. 1993, pp. 113-117
20. Z. Bojković, A. Samčović, B. Reljin: "Subjective Image Quality Estimation in Some Subband Coding Techniques", *Proceedings of the Third IEEE International Conference on Electronics, Circuits and Systems*, Rodos, Greece, Oct. 13-16, 1996, pp. 582-585
21. Z. Bojković, C. I. Toma, V. Gui, R. Vasii: *Advanced Topics in Image Compression Techniques*, Editura Politehnica, Timișoara, 1997
22. A. Boukelif: *Techniques Numeriques en Television*, Ed. Masson, Paris, 1993
23. R. C. Brainard, A. N. Netravali, D. E. Pearson: "Composite Television Coding: Subsampling and Interpolation", *Journal of the SMPTE*, Aug. 1982, pp. 717-724
24. R. C. Brainard, A. N. Netravali: "Digital Television at 45 Mbit/sec", *Journal of the SMPTE*, Jan. 1987, pp. 4-10
25. R. W. Brockett, P. Maragos: "Evolution Equations for Continuous-Scale Morphological Filtering", *IEEE Transactions on Signal Processing*, vol. SP-42, no. 12, Dec. 1994, pp. 3377-3385
26. J. Brown, A. Hoger: "A Morphological Point Thinning Algorithm". *Pattern Recognition Letters*, vol. 17, 1996, pp. 197-207
27. A. C. Bovik, "Analysis of Multichannel Narrow-Band Filters for Image Texture Segmentation", *IEEE Transactions on Signal Processing*, vol. 39, no. 9, Sept. 1991, pp. 2025-2043
28. R. Burkhardt, J. Wasser: "Digital Television Transmission with 34 Mbit/s", *Journal of the SMPTE*, April 1980, pp. 244-248
29. L. Calabi: "A Study of the Skeleton of Plane Figures", *Parke Mathematical Labs.* June 1965
30. B. Carpentieri, J. A. Storer: "Split-Merge Video Displacement Estimation". *Proceedings of the IEEE*, vol. 82, no. 6, June 1994, pp. 940-947
31. M. D. Carr: "Video Codec Hardware to Realise a New World Standard", *British Telecom Journal*, vol. 8, nr. 3, July 1990, pp. 28-35
32. M. D. Carr: "New Video Coding Standard for the 1990s", *Electronics & Communication Engineering Journal*, June 1990, pp. 119-124
33. B. Chanda, "Applications of binary Morphology to Separate Overlapped Objects". *Pattern Recognition Letters*, vol. 13, no. 9, Sept. 1992, pp. 639-647
34. L. W. Chang, K. L. Leu: "A New Method to Obtain Chain Codes from Y-axis Representation of a Region in a Binary Image", *Multidimensional Systems and Signal Processing*, no. 3, 1992, pp. 79-87
35. L. W. Chang, K. L. Leu: "A Fast Algorithm for the Restoration of Images Based on Chain Codes Description and Its Applications", *Computer Vision, Graphics and Image Processing*, vol. 50, 1990, pp. 296-307
36. M. Charif-Chefchaoui, D. Schonfeld: "On the Invertibility of the Morphological Representation of Binary Images", *IEEE Transactions on Image Processing*, vol. IP-3, no. 6, Nov. 1994, pp. 847-849



37. S. Chen, R. M. Haralick: "Recursive Erosion, Dilation, Opening and Closing Transforms", *IEEE Transactions on Image Processing*, vol. IP-4, no. 3, March 1995, pp. 335-345
38. L. Chiariglione: *Signal Processing of HDTV*, Elsevier Science Publishers. Amsterdam, 1988
39. R. J. Clarke: *Digital Compression of Still Images and Video*, Academic Press, 1995
40. R. J. Clarke: "On Transform Coding of Motion-Compensated Difference Images", *IEE Proceedings-I: Communications, Speech and Vision*, vol. 139, nr.3, June 1992, pp. 372-376
41. D. Coltuc, I. Pitas: "Morphological Residual Representations of Signals", *IEEE Transactions on Image Processing*, vol. IP-5, no. 11, Nov. 1996, pp. 1569-1572
42. M. Coster, J. L. Chermant: *Precis d'analyse d'images*, Presses du CNRS, 1989
43. G. Cullman: *Coduri detectoare si corectoare de erori*, Ed. Tehnică, Bucuresti, 1972
44. S. Cucchi, F. Molo: "DCT-Based Television Codec for DS3 Digital Transmission", *SMPTE Journal*, vol.98, nr.9, Sept. 1989, pp. 640-646
45. P. J. Czerepinski and D. R. Bull, "Morphological Video Coder with Conditional Smoothing", *Electronic Letters*, vol. 32, no. 7, 28 March 1996, pp. 645-647
46. S. D'Agostino, J. David, R. Heiss, W. Verbiest: "Universal ATM Video Coding Architecture", *Electrical Communication, Alcatel*, vol. 64, nr.2/3, 1990, pp. 186-193
47. A. Datta, S. K. Parui: "A Robust Parallel Thinning Algorithm for Binary Images", *Pattern Recognition*, vol. 27, no. 9, 1994, pp. 1181-1192
48. D. G. Daut, D. Zhao: "A Flaw Detection Method Based on Morphological Image Processing", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. CSVT-3, no. 6, Dec. 1993, pp. 389-398
49. G. S. Di Baja, E. Thiel: "(3,4)-Weighted Skeleton Decomposition for Pattern Representation and Description", *Pattern Recognition*, vol. 27, no. 8, 1994, pp. 1039-1049
50. A. R. Dill, M. D. Levine, P.B. Noble: "Multiple Resolution Skeletons", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-9, no. 4, July 1987, pp. 495-504
51. D. L. Duttweiler, C. Chamzas: "Probability Estimation in Arithmetic and Adaptive-Huffman Entropy Coders", *IEEE Transactions on Image Processing*, vol. IP-4, no. 3, March 1995, pp. 237-246
52. V. Dzhakonia: *Television*, Mir Publishers, Moscow, 1989
53. T. Ebrahimi, E. Reusens, W. Li: "New Trends in Very Low Bitrate Video Coding", *Proceedings of the IEEE*, vol. 83, no. 6, June 1995, pp. 877-891
54. O. Egger, W. Li, M. Kunt: "High Compression Image Coding Using an Adaptive Morphological Subband Decomposition", *Proceedings of the IEEE*, vol. 83, no. 2, Febr. 1995, pp. 272-287
55. J. A. Elliott, C. Cubiss, P. M. Grant, J. T. E. McDonnell: "Real-Time Simulation of Videophone Image Coding Algorithms on Reconfigurable Multicomputers", *IEE Proceedings-E : Computer and Digital Techniques*, vol.139, no.3, May 1992, pp. 269-279
56. W. H. Equiz: "A New Vector Quantization Clustering Algorithm", *IEEE Transactions on Acoustic, Speech and Signal Processing*, vol. ASSP-37, no. 10, Oct. 1989, pp. 1568-15759
57. A. M. Eskicioglu, P. Fisher: "Image Quality Measures and Their Performance", *IEEE Transactions on Communications*, vol. COMM-43, no. 12, Dec. 1995, pp. 2959-2965

58. B. Evans: *Understanding Digital TV. The Route to HDTV*, IEEE Press, 1995
59. K. Feher: "Comunicatii digitale avansate", vol. I, Ed. Tehnică, Bucuresti, 1993
60. A. Ferreira, S. Ubeda: "Ultra-Fast Parallel Contour Tracking with Applications to Thinning", *Pattern Recognition*, vol. 27, no. 7, 1994, pp. 867-878
61. D. S. Fritsch, S. M. Pizer, B. S. Morse, D. H. Eberly, A. Liu: "The Multiscale Medial Axis and Its Applications in Image Registration", *Pattern Recognition Letters*, vol. 15, May 1994, pp. 445-452
62. Y. Ge, J. M. Fitzpatrick: "On the Generation of Skeletons from Discrete Euclidean Distance Maps", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-18, no. 11, Nov. 1996, pp. 1055-1066
63. C. R. Giardina, E. R. Dougherty: *Morphological Methods in Image and Signal Processing*, Prentice Hall, 1988
64. J. Golston, B. Gove, F. Laczko, J. Niehaus, C. Read, D. Shaver, I. F. Wang: "VLSI Implementations of Image and Video Compression Algorithms", *Texas Instruments Technical Journal*, pp. 42-55, sept.-oct. 1991
65. R. C. Gonzalez, P. Wintz: *Digital Image Processing*, Addison Wesley Publishing, New York, 1987
66. R. C. Gonzalez, R. E. Woods: *Digital Image Processing*, Addison Wesley Publishing, New York, 1992
67. J. Goutsias, D. Schonfeld: "Morphological Representation of Discrete and Binary Images", *IEEE Transactions on Signal Processing*, vol. SP-39, no. 6, June 1991, pp. 1369-1379
68. J. Goutsias, H. J. A. M. Heijmans, J. Sivakumar: "Morphological Operators for Image Sequences", *Computer Vision Image Understanding*, nov. 1995, pp. 326-346
69. E. L. Hall: *Computer Image Processing and Recognition*, Academic Press, New York, 1979
70. C. C. Han, K. C. Fan: "A Greedy and Branch and Bound Searching Algorithm for Finding the Optimal Morphological Erosion Filter on Binary Images", *IEEE Signal Processing Letters*, vol. 1, no. 2, Febr. 1994, pp. 41-44
71. C. C. Han, K. C. Fan: "Skeleton Generation of Engineering Drawings Via Contour Matching", *Pattern Recognition*, vol. 27, no. 2, 1994, pp. 261-275
72. R. M. Haralick, S. R. Sternberg, K. Zhuang: "Image Analysis Using Mathematical Morphology", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-9, no. 4, July, 1987, pp. 532-550
73. R. M. Haralick, P. L. Katz, E. R. Dougherty: "Model-Based Morphology: The Opening Spectrum", *Graphical Models and Image Processing*, vol. 57, no. 1, Jan. 1995, pp. 1-12
74. H. Harrer: "Multiple Layer Discrete-Time Cellular Neural Networks Using Time-Variant Templates", *IEEE Transactions on Circuits and Systems-II: Analog and Digital Signal Processing*, vol. CIS-40, no. 3, March 1993, pp. 191-199
75. P. W. Hawkes: "Manipulation of Multivalued Images in Image Algebra", *Journal of Mathematical Imaging and Vision*, vol. 2, no. 1, Oct. 1992, pp. 83-85
76. J. P. Hayes, T. Mudge: "Hypercube Supercomputers", *IEEE Proceedings*, vol. 77 (12), Dec. 1989, pp. 1829-1841
77. H. J. A. M. Heijmans, C. Ronse: "The Algebraic Basis of Mathematical Morphology; I. Dilations and Erosions", *Computer Vision, Graphics and Image Processing*, vol. 50, 1990, pp. 245-295

78. H. J. A. M. Heijmans: "Theoretical Aspects of Gray-Level Morphology", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-13, no. 6, June 1991, pp. 568-582
79. H. J. A. M. Heijmans: "*Morphological Image Operators*", Academic Press, 1994
80. H. Hofmann: "A Codec for the Distribution of Digital HDTV Signals via Broadcasting Satellites and Digital Networks", *Fourth International Workshop on HDTV and Beyond*, Torino, 4-6 sept. 1991
81. C. L. Huang: "Contour Image Sequence Compression Through Motion Analysis and Hybrid Coding Method", *Multidimensional Systems and Signal Processing*, vol. 3, no. 2/3, May 1992, pp. 267-296
82. H. C. Huang, J. L. Wu: "Windowed Huffman Coding Algorithm with Size Adaptation", *IEE Proceedings-I*, vol. I-40, no. 2, April 1993, pp. 109-113
83. Y. Huh, K. Panusopone, K. R. Rao: "Variable Block Size Coding of Images with Hybrid Quantization", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, no. 6, Dec. 1996, pp. 679-685
84. Y. Itoh: "An Edge-Oriented Progressive Image Coding", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, no. 2, April 1996, pp. 135-142
85. P. T. Jackway, "Gradient Watersheds in Morphological Scale-Space", *IEEE Transactions on Image Processing*, vol. 5, no. 6, June 1996, pp. 913-921
86. P. T. Jackway and M. Deriche, "Scale-Space Properties of the Multiscale Morphological Dilatation-Erosion", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-18, no. 1, Jan. 1996, pp. 38-52
87. B. Jahne: *Digital Image Processing. Concepts, Algorithms and Scientific Applications*, Springer-Verlag, Berlin, 1995
88. B. K. Jang, R. T. Chin: "Analysis of Thinning Algorithms Using Mathematical Morphology", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-12, no. 6, June 1990, pp. 541-551
89. B. K. Jang, R. T. Chin: "Comments on the Invertibility of Morphological Representation of Binary Images", *IEEE Transactions on Image Processing*, vol. IP-5, no. 3, March 1996, pp. 529-532
90. N. Jayant, J. Johnston, R. Safranek: "Signal Compression Based on Models of Human Perception", *Proceedings of the IEEE*, vol. 81, no. 10, Oct. 1993, pp. 1385-1422
91. N. S. Jayant, P. Noll: "*Digital Coding of Waveforms. Principles and Applications to Speech and Video*", Prentice Hall, Englewood Cliffs, New Jersey, 1984
92. L. Ji, J. Piper: "Fast Homotopy-Preserving Skeletons Using Mathematical Morphology", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-14, no. 6, June 1992, pp. 653-664
93. Y. Jiguni: "Hierarchical Image Coding via Cerebellar Model Arithmetic Computers", *IEEE Transactions on Image Processing*, vol. 5, no. 10, Oct. 1996, pp. 1393-1401
94. X. C. Jin, S. H. Ong, Jayasooriah: "A Domain Operator for Binary Morphological Processing", *IEEE Transactions on Image Processing*, vol. IP-4, no. 7, July 1995, pp. 1042-1046
95. R. Jones, I. Svaibe: "Morphological Filtering as Template Matching", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-16, no. 4, April 1994, pp. 439-443

96. R. Jones, I. Svaibe: "Algorithms for the Decomposition of Gray-Scale Morphological Operations", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-16, no. 6, June 1994, pp. 581-588
97. R. L. Joshi, V. J. Crump, T. R. Fischer: "Image Subband Coding Using Arithmetic Coded Trellis Coded Quantization", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 5, no. 6, Dec. 1995, pp. 515-523
98. H. Kai, T. Ping-Sheng, K. Dongseung, "An Orthogonal Multiprocessor for Parallel Scientific Computations", *IEEE Transactions on Computers*, vol. 38(1), Jan. 1989, pp. 47-61
99. T. Kanungo, R. Haralick: "Vector-Space Solution for a Morphological Shape-Decomposition Problem", *Journal of Mathematical Imaging and Vision*, vol. 2, no. 1, Oct. 1992, pp. 51-82
100. M. Khoroshavi and R.W. Schaffer, "Implementation of Linear Digital Filters Based on Morphological Representation Theory", *IEEE Transactions on Signal Processing*, vol. 49, no. 9, Sept. 1994, pp. 2264-2276
101. M. Khoroshavi and R. W. Schaffer, "Template Matching Based on a Grayscale Hit or Miss Transform", *IEEE Transactions on Image Processing*, vol. 5, no. 6, June 1996, pp. 1060-1066
102. J. T. Kim, H. J. Lee, J. S. Choi: "Subband Coding Using Human Visual Characteristics for Image Signals", *IEEE Journal on Selected Areas in Communications*, vol. 11, no. 1, Jan. 1993, pp. 59-64
103. R. Klette: *A Simplified Approach to Image Processing. Classical and Modern Techniques in C*, Prentice Hall, 1997
104. M. J. Knee: "Adaptive prediction for High-Quality Television Transmission Coding Based on the LMS Algorithm", *SMPTTE Journal*, vol.98, no.8, Aug. 1989, pp.580-589
105. S. J. Ko, A. Morales, K. H. Lee: "A Fast Implementation Algorithm and a Bit-Serial Realization Method for Grayscale Morphological Opening and Closing". *IEEE Transactions on Signal Processing*, vol. 43, no. 12, Dec. 1995, pp. 3058-3061
106. J. C. Kotelly: "A Mathematical Model of Blum's Theory of Pattern Recognition". *Air Force Cambridge Research Labs.*, Apr. 1963
107. E. R. Kretzmer: "Statistics of Television Signals", *Bell System Technical Journal*, vol. 31, pg. 751-763
108. M. Kunt, A. Ikonomopoulos, M. Kocher: "Second-Generation Image-Coding Techniques", *Proceedings of the IEEE*, vol. 73, no. 4, April 1985, pp. 549-574
109. G. Lakhani, V. Ayyagari: "Improved Huffman Code Tables for JPEG's Encoder". *IEEE Transactions on Circuits and Systems for Video Technology*, vol. CSVT-5, no. 6, Dec. 1995, pp. 562-564
110. L. Lam, S. W. Lee, C. Y. Suen: "Thinning Methodologies-A Comprehensive Survey", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-14, no. 9, Sept. 1992, pp. 869-885
111. C. Lantuejoul: "Skeletonization in Quantitative Metallography", in *Issues of Digital Image Compression*, R. M. Haralick, J. C. Simon, eds., Groningen, 1980
112. L. Leboucher, T. Irinopoulou, S. Hazout: "Grey-Tone Skeletons of Elongated Objects Using the Concept of Morphological Automaton. Application to Images of DNA Molecules", *Pattern Recognition Letters*, vol. 15, 1994, pp. 309-315
113. C. Lee, M. Nadler: "Interframe Image Coding with 3-Dimensional Edge Detection", *IEE Third International Conference of Image Processing Applications*, 18-20 July 1989, University of Warwick, UK, pp. 541-545

114. C. K. Lee and S. P. Wong, "A Mathematical Morphology Approach for Segmenting Heavily noise-corrupted Images", *Pattern Recognition*, vol. 29, no. 9, Aug. 1996
115. J. H. Lee, H. T. Liu: "Digital Image Coding with High Compression Ratio", *IEE Third International Conference on Image Processing and Applications*, 18-20 July 1989, University of Warwick, UK, pp. 595-598
116. G. H. Lee, J. S. Kim, R. H. Park: "Video Coding Using Variable Block-Size Segmentation by Motion Vectors", *Journal of Visual Communications and Image Representation*, vol. 5, no. 4, Dec. 1994, pp. 342-355
117. M. Leonard: "Scanning the Options for Image Compression", *Electronic Design*, July 1992, pp. 40-46
118. X. Li, A. Basu: "Variable-Resolution Character Thinning", *Pattern Recognition Letters*, vol. 12, 1991, pp. 241-248
119. H. Li, A. Lundmark, R. Forchheimer: "Image Sequence Coding at Very Low Bitrates: A Review", *IEEE Transactions on Image Processing*, vol. IP-3, no. 5, September 1994, pp. 589-609
120. E. H. Liang, E. K. Wong: "Hierarchical Algorithms for Morphological Image Processing", *Pattern Recognition*, vol. 26, no. 4, 1993, pp. 511-529
121. E. H. Liang, E. K. Wong: "An Efficient Method for Obtaining Morphological Skeletons", *Pattern Recognition Letters*, vol. 14, Aug. 1993, pp. 689-693
122. M. Lighthstone, S. K. Mitra: "Optimal Variable-Rate Mean-Gain-Shape Vector Quantization for Image Coding", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, no. 6, Dec. 1996, pp. 660-668
123. C. A. Lindley: *Practical Image Processing in C*, John Wiley & Sons, New York, 1991
124. A. C. P. Loni, A. N. Venetsanopoulos and K. C. Smith, "Flexible Architectures for Morphological Image Processing and Analysis", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 2, no.1, March 1992, pp. 72-83
125. G. Lu: *Communication and Computing for Distributed Multimedia Systems*, Artech House, 1996
126. A. C. Luther: *Digital Video in the PC Environment*, McGraw-Hill Book Company, New York, 1991
127. B. M. Macq, J. J. Quisquater: "Cryptology for Digital TV Broadcasting", *Proceedings of the IEEE*, vol. 83, no. 6, June 1995, pp. 944-957
128. P. Maragos: "A Representation Theory for Morphological Image and Signal Processing", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 6, June 1989, pp. 586-599
129. P. Maragos: "Differential Morphology and Image Processing", *IEEE Transactions on Image Processing*, vol. IP-5, no. 6, June 1996, pp. 922-937
130. P. A. Maragos, R. W. Schafer: "Morphological Skeleton Representation and Coding of Binary Images", *IEEE Transactions on Acoustic, Speech and Signal Processing*, vol. ASSP-34, no. 5, Oct. 1986, pp. 1228-1244
131. P. Maragos, R. Schafer: "Morphological Filters-Part I: Their Set-Theoretic Analysis and Relations to Linear Shift-Invariant Filters", *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. ASSP-35, no. 8, Aug. 1987, pp. 1153-1169
132. P. Maragos, R. Schafer: "Morphological Filters-Part II: Their Relations to Median, Order-Statistic and Stack Filters", *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. ASSP-35, no. 8, Aug. 1987, pp. 1170-1184

133. P. Maragos, R. W. Schafer: "Morphological Systems for Multidimensional Signal Processing", *Proceedings of the IEEE*, vol. 78, no. 4, April 1990, pp. 690-710
134. P. Maragos, R. D. Ziff: "Threshold Superposition in Morphological Image Analysis Systems", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-12, no. 5, May 1990, pp. 498-504
135. P. Maragos, R. W. Schafer, M. A. Butt: *Mathematical Morphology and Its Applications to Image and Signal Processing*, Kluwer Academic Publishers, 1996
136. A. Marion: *An Introduction to Image Processing*, Chapman and Hall, London, 1991
137. M. Marković, D. Dimitrijević, Z. Bojković: "Interactive Voice Communications Over Internet Based on CELP 4800 b/s Speech Coding", *Proceedings of the 4<sup>th</sup> International Workshop on Systems, Signals and Image Processing*, Poznan, Poland, May 28-30, 1997, pp. 243-246
138. S. Marshall: "Application of Image Contours to Three Aspects of Image Processing: Compression, Shape Recognition and Stereopsis", *IEE Third International Conference on Image Processing and Applications*, 18-20 July 1989, University of Warwick, UK, pp. 604-608
139. A. Mateescu, I. Băncilă, S. Popescu, E. Borcoci: *Manualul inginerului electronist. vol. II. Transmisiuni de Date*, Ed. Tehnică, Bucuresti, 1984
140. P. Meer, C. A. Sher, A. Rosenfeld: "The Chain Pyramid: Hierarchical Contour Processing", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-12, no. 4, April 1990, pp. 363-375
141. D. Milovanović, A. Samčović, Z. Bojković: "A Comparative Study on Some Subband Transforms for Still Image Compression", *Journal on Communications*, vol. XLV, May-June 1994, pp. 23-24
142. D. Milovanović, Z. Bojković, A. Samčović, B. Reljin: "Upper Bound on Redundancy Reduction in Predictive-Entropy Subband Image Coding", *China Journal of Image and Graphics*, vol. 2, no. 4, Apr. 1997, pp. 239-249
143. M. Mirmehdi, R. Marik, M. Petron ad J. Kittler, "Iterative Morphology for Fault Detection in Stochastic Textures", *Electronics Letters*, vol. 32, no. 5, 29 Feb. 1996, pp. 443-444
144. Ghe. I. Mitrofan: *Televiziunea Digitală*, Ed. Academiei, Bucuresti, 1986
145. Ghe. I. Mitrofan: *Televiziune*, Ed. Teora, București, 1993
146. M. A. Mohamed, J. Saniie: "Statistical Evaluation of Sequential Morphological Operations", *IEEE Transactions on Signal Processing*, vol. SP-43, no. 7, July 1995, pp. 1703-1709
147. G. Mongatti, L. Alparone, G. Benelli, S. Baronti, F. Lotti, A. Casini: "Progressive Image Transmission by Content Driven Laplacian Pyramid Encoding", *IEE Proceedings-I*, vol. 139, no. 5, Oct. 1992, pp. 495-500
148. U. Montanari: "A Method for Obtaining Skeletons Using a Quasi-Euclidean Distance", *Journal Assoc. Computer*, vol. 15, Oct. 1968, pp. 600-624
149. M. E. Montiel, A. S. Aguando, M. A. Garza and J. Alarcon, "Image Manipulation Using M-Filters in a Pyramidal Computer Model", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 11, Nov. 1995, pp. 1110-1117
150. A. Morales and R. Acharya, "Statistical Analysis of Morphological Openings", *IEEE Transactions on Signal Processing*, vol. 41, no. 10, Oct. 1993, pp. 3052-3056
151. A. Morales, R. Acharya and C. Steele, "Morphological Pyramid with Alternating Sequential Filters", *IEEE Transactions on Image Processing*, vol. 4, no. 7, July 1995, pp. 978-989

152. D. G. Morrison: "Variable Bit-Rate Video Coding for Asynchronous Transfer Mode Networks", *British Telecom Technology Journal*, vol. 8, no. 3, July 1990, pp. 70-80
153. J. C. Mott-Smith: "Medial Axis Transformation", in *Picture Processing and Psychopictorics*, B. S. Lipkin, A. Rosenfeld, eds., New York, Academic Press, 1970
154. J. D. Murray, W. VanRyper: *Encyclopedia of Graphics File Formats*, O'Reilly & Associates, 1996
155. S. Na and T. Y. Choi, "Second Order Statistics of Morphological Dilation and Erosion of a Memoryless Source", *IEEE Transactions on Signal Processing*, vol. 43, no. 10, Oct. 1995, pp. 2418-2421
156. A. N. Netravali, B. G. Haskell, *Digital Pictures. Representation and Compression*, Plenum Press, New York, 1991
157. A. N. Netravali, B. Prasada: *Visual Communications Systems*, IEEE Press, New York, 1989
158. NTL: *MPEG-Digital Television for All*, Swift Television Publications, Cricklade, England, 1995
159. R. L. Ogniewicz, O. Kubler: "Hierarchic Voronoi Skeletons", *Pattern Recognition*, vol. 28, no. 3, 1995, pp. 343-359
160. J. B. O'Neal: "Predictive Quantizing Systems (DCPM) for the Transmission of Television Signals", *Bell Systems Technical Journal*, May-June 1966 pp. 689-719
161. T. P. O'Rourke, R. L. Stevenson: "Improved Image Decompression for Reduced Transform Coding Artifacts", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 5, no. 6, Dec. 1995, pp. 490-499
162. L. A. Overturf, M. L. Comer, E. J. Delp: "Color Image Coding Using Morphological Pyramid Decomposition", *IEEE Transactions on Image Processing*, vol. IP-4, no. 2, Febr. 1995, pp. 177-185
163. J. Ozer: *Video Compression for Multimedia*, Academic Press, London, 1995
164. J. Ozer: *Publishing Digital Video*, Academic Press, London, 1997
165. T. W. Pai, J. H. L. Hansen: "Boundary-Constrained Morphological Skeleton Minimization and Skeleton Reconstruction", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-16, no. 2, Febr. 1994, pp. 201-208
166. S. Panchanathan, M. Goldberg: "Mini-max Algorithm for Image Adaptive Vector Quantisation", *IEE Proceedings-I*, vol. 138, no. 1, Febr. 1991, pp. 53-60
167. K. R. Park and C. N. Lee, "Scale-Space Using Mathematical Morphology", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 11, Nov. 1996, pp. 1121-1127
168. H. Park, R. T. Chin: "Optimal Decomposition of Convex Morphological Structuring Elements for 4-Connected Parallel Array Processors", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-16, no. 3, March 1994, pp. 304-313
169. H. Park, R. T. Chin: "Decomposition of Arbitrarily Shaped Morphological Structuring Elements", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-17, no. 1, Jan. 1995, pp. 2-15
170. C. Pei, F. C. Chen: "Hierarchical Image Representation by Mathematical Morphology Subband Decomposition", *Pattern Recognition Letters*, vol. 16, Febr. 1995, pp. 183-192
171. D. E. Pearson: "Developments in Model-Based Video Coding", *Proceedings of the IEEE*, vol. 83, no. 6, June 1995, pp. 892-906



172. W. B. Pennebaker, J. L. Mitchell: *JPEG Still Image Data Compression Standard*, Van Nostrand Reinhold, New York, 1993
173. R. A. Peters: "A New Algorithm for Image Noise Reduction Using Mathematical Morphology", *IEEE Transactions on Image Processing*, vol. IP-4, no. 5, May 1995, pp. 554-568
174. P. Pirsch, N. Demassieux, W. Gehrke: "VLSI Architectures for Video Compression-A Survey", *Proceedings of the IEEE*, vol. 83, no. 2, Febr. 1995, pp. 220-246
175. I. Pitas: *Parallel Algorithms for Digital Image Processing, Computer Vision and Neural Networks*, John Wiley & Sons, New York, 1993
176. I. Pitas, A. N. Venetsanopoulos: "Morphological Shape Decomposition", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-12, no. 1, Jan. 1990, pp. 38-45
177. R. H. J. M. Plompen, *Motion Video Coding for Visual Telephony*, PTT Research Neher Laboratories, Delft, 1989
178. J. L. Potter, "Image Processing on the Massively Parallel Processor", *Computer*, vol. 16, 1983, pp. 62-67
179. W. K. Pratt: *Developing Visual Applications*, Sun Microsystems Press, Prentice Hall, 1997
180. R. J. Qian, T. S. Huang: "Optimal Edge Detection in Two-Dimensional Images". *IEEE Transactions on Image Processing*, vol. 5, no. 7, July 1996, pp. 1215-1220
181. M. Rabbani, P. W. Jones, *Digital Image Compression Techniques*, SPIE Optical Engineering Press, Washington, 1991
182. I. Ragnemalm: "Fast Erosion and Dilation by Contour Processing and Thresholding of Distance Maps", *Pattern Recognition Letters*, vol. 13, March 1992, pp. 161-166
183. P. Rainger, P. A. Ratliff: "Low Bit-Rate System for Digital Coding of the Television Picture", *Journal of the SMPTE*, August 1979, pp. 550-554
184. X. Ran, N. Farvardin: "A Perceptually Motivated Three-Component Image Model - Part I: Description of the Model", *IEEE Transactions on Image Processing*, vol. IP-4, no. 4, April 1995, pp. 401-415
185. X. Ran, N. Farvardin: "A Perceptually Motivated Three-Component Image Model - Part II: Applications to Image Compression", *IEEE Transactions on Image Processing*, vol. IP-4, no. 4, April 1995, pp. 430-447
186. N. Ranganathan, S. G. Romaniuk, K. R. Namuduri: "A Lossless Image Compression Algorithm Using Variable Block Size Segmentation", *IEEE Transactions on Image Processing*, vol. 4, no. 10, Oct. 1995, pp. 1396-1405
187. D. F. Read: "*Digital Video: Some Bit-rate Reduction Methods Which Preserve Information in Broadcast-quality Digital Video Signals*" Report 1974/37, British Broadcasting Corporation, Research Department
188. J. M. Reinhardt, W. E. Higgins: "Efficient Morphological Shape Representation", *IEEE Transactions on Image Processing*, vol. IP-5, no. 1, Jan. 1996, pp. 89-101
189. J. M. Reinhardt, W. E. Higgins: "Comparison Between the Morphological Skeleton and Morphological Shape Decomposition", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-18, no. 9, Sept. 1996, pp. 951-957
190. M. J. Riley, I. E. G. Richardson: *Digital Video Communications*, Artech House, London, 1997
191. A. Rosenfeld, A. C. Kak: *Digital Picture Processing*, Academic Press, New York, 1976

192. A. Rosenfeld, A. C. Kak: *Digital Picture Processing*, Academic Press, New York, 1982
193. A. Rosenfeld, J. L. Pfaltz: "Sequential Operations in Digital Picture Processing", *Journal Assoc. Computer*, vol. 13, Oct. 1966, pp. 471-494
194. T. S. Rzeszewski, R. L. Pawelski: "Efficient Transmission of Digital Component Video", *Journal of the SMPTE*, no. 9, Sept. 1986, pp. 889-898
195. S. Sabri, B. Prasada: "Video Conferencing Systems", *Proceedings of the IEEE*, April 1985, pp. 671-688
196. J. A. Saghri, A. G. Tescher: "Feature-Based Image Bandwidth Compression", *Optical Engineering*, vol. 27, no. 10, Oct. 1988, pp. 854-860
197. A. Said, W. A. Pearlman: "An Image Multiresolution Representation for Lossless and Lossy Compression", *IEEE Transactions on Image Processing*, vol. 5, no. 9, Sept. 1996, pp. 1303-1310
198. P. Salambier, P. Brigger, J. R. Casas, M. Pardas: "Morphological Operators for Image and Video Compression", *IEEE Transactions on Image Processing*, vol. IP-5, no. 6, June 1996, pp. 881-897
199. P. Salambier, M. Pardas: "Hierarchical Morphological Segmentation for Image Sequence Coding", *IEEE Transactions on Image Processing*, vol. 3, no. 5, Sept. 1994, pp. 639-651
200. P. Salambier, J. Serra: "Flat Zones Filtering, Connected Operators and Filters by Reconstruction", *IEEE Transactions on Image Processing*, vol. IP-4, no. 8, Aug. 1995, pp. 1153-1160
201. P. Salambier, L. Torres, F. Meyer, C. Gu: "Region-Based Video Coding Using Mathematical Morphology", *Proceedings of the IEEE*, vol. 83, no. 6, June 1995, pp. 843-857
202. A. Samčović, Z. Bojković, D. Milovanović: "Image Quality in LOT Subband Decomposition and VQ Coding", *Proceedings of the 4<sup>th</sup> International Workshop on Systems, Signals and Image Processing*, Poznan, Poland, May 28-30, 1997, pp. 153-156
203. C. P. Sandbank: *Digital Television*, John Wiley & Sons, London, 1990
204. G. Sapiro, D. Malah: "Morphological Image Coding Based on a Geometric Sampling Theorem and a Modified Skeleton Representation", *Journal of Visual Communications and Image Representation*, vol. 5, no. 1, March 1994, pp. 29-40
205. D. Sarkar: "Boolean Function-Based Approach for Encoding of Binary Images", *Pattern Recognition Letters*, vol. 17, 1996, pp. 839-848
206. K. Sauer, G. Jones: "Bayesian Block-Wise Segmentation of Interframe Differences in Video Sequences", *CVGIP: Graphic Models and Image Processing*, vol. 55, no. 2, March 1993, pp. 129-139
207. R. Schaphorst: *Videoconferencing and Videotelephony. Technology and Standards*, Artech House, 1996
208. D. Schonfeld: "Optimal Structuring Elements for the Morphological Pattern Restoration of Binary Images", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-16, no. 6, June 1994, pp. 589-601
209. G. M. Schuster, A. K. Katsaggelos: *Rate Distortion Based Video Compression*, Kluwer Academic Publishers, 1997
210. I. Sebestyen, C. J. Starkey, S. J. Urban, Y. Yamazaki and T. Yoshida, "Technical Features of the JBIG Standard for Progressive Bi-level Image Compression", *Image Communication*, vol. 4, no. 2, April 1992, pp. 103-113

211. J. Serra: *Image Analysis and Mathematical Morphology*, vol. I, Academic Press, 1982
212. J. Serra: *Image Analysis and Mathematical Morphology. Vol. II, Theoretical Advances*, Academic Press, 1988
213. J. Serra: "Introduction to Mathematical Morphology", *Computer Vision, Graphics and Image Processing*, vol. 35, 1986, pp. 283-305
214. Y. M. Sharaiha, P. Garat: "Digital Straightness and the Skeleton Property", *Pattern Recognition Letters*, vol. 16, April 1995, pp. 417-423
215. D. J. Sheely, C. G. Armstrong and D. J. Robinson, "Shape Description by Medial Surface Construction", *IEEE Transactions on Visualization and Computer Graphics*, vol. 2, no. 1, March 1996, pp. 62-72
216. D. Sheinwald: "On the Ziv-Lempel Proof and Related Topics", *Proceedings of the IEEE*, vol. 82, no. 6, June 1994, pp. 866-871
217. F. Y. Shih, O. R. Mitchell: "Threshold Decomposition of Gray-Scale Morphology into Binary Morphology", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 1, Jan. 1989, pp. 31-42
218. F. Y. Shih, C. C. Pu: "A Skeletonization Algorithm by Maxima Tracking on Euclidean Distance Transform", *Pattern Recognition*, vol. 28, no. 3, 1995, pp. 331-341
219. F. Y. Shih and C. C. Pu, "Analysis of the Properties of Soft Morphological Filtering Using Threshold Decomposition", *IEEE Transactions on Signal Processing*, vol. 43, no. 2, Feb. 1995, pp. 539-543
220. N. D. Sidiropoulos, D. Meleas and T. Stragas, "MAP Signal Estimation in noisy Sequencies of Morphological Smooth Images", *IEEE Transactions on Image Processing*, vol. 5, no. 6, June 1996, pp. 1088-1093
221. E. P. Simoncelli, E. H. Adelson: "Non-Separable Extensions of Quadrature Mirror Filters to Multiple Dimensions", *Proceedings of the IEEE*, vol. 78, no. 4, April 1990, pp. 652-664
222. B. Singh, M. U. Siddiqi: "A Simplified Algorithm for Approximate Separable Decomposition of Morphological Templates", *Pattern Recognition*, vol. 29, no. 9, 1996, pp. 1519-1522
223. S. Singhal, D. Le Gal, C. T. Chen: "Source Coding of Speech and Video Signals", *Proceedings of the IEEE*, vol. 78, no. 7, July 1990, pp. 1233-1249
224. D. Sinha, C.R. Giardina: "Discrete Black and White Object Recognition Via Morphological Functions", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-12, no. 3, March 1990, pp. 275-293
225. R. M. K. Sinha: "A Width-Independent Algorithm for Character Skeleton Estimation", *Computer Vision, Graphics and Image Processing*, vol. 40, 1987, pp. 388-397
226. A. Sirjani, G. R. Cross: "On Representation of a Shape's Skeleton", *Pattern Recognition Letters*, vol. 12, March 1991, pp. 149-154
227. K. Sivakumar, J. Goutsias: "Binary Random Fields, Random Closed Sets and Morphological Sampling", *IEEE Transactions on Image Processing*, vol. IP-5, no. 6, June 1996, pp. 899-911
228. J. Slater: *Modern Television Systems: to HDTV and Beyond*, Pittman Publishing, London, 1991
229. B. Smith: "Instantaneous Companding of Quantized Signals", *Bell Systems Technical Journal*, no. 36, 1957, pp. 653-709

230. P. Soille, E. J. Breen, R. Jones: "Recursive Implementation of Erosions and Dilations Along Discrete Lines at Arbitrary Angles", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-18, no. 5, May 1996, pp. 562-567
231. J. Song, E. J. Delp: "The Analysis of Morphological Filters with Multiple Structuring Elements", *Computer Vision, Graphics and Image Processing*, vol. 50, 1990, pp. 308-328
232. M. Sonka, V. Hlavac, R. Boyle: *Image Processing, Analysis and Machine Vision*, Chapman and Hall Computing, 1993
233. Al. Spătaru: *Teoria transiterii informației*, vol.I, II, Ed. Tehnică, București, 1965, 1971
234. R. C. Staunton: "An Analysis of Hexagonal Thinning Algorithms and Skeletal Shape Representation", *Pattern Recognition*, vol. 29, no. 7, 1996, pp. 1131-1146
235. S. R. Sternberg: "Grayscale Morphology", *Computer Vision, Graphics and Image Processing*, vol. 35, 1986, pp. 333-355
236. A. Stewart: "A One-Pass Thinning Algorithm with Interference Guards", *Pattern Recognition Letters*, vol. 15, 1994, pp. 825-832
237. J. Storer: *Data Compression. Methods and Theory*, Computer Science Press, 1988
238. J. Storer: *Image and Text Compression*, Kluwer Academic Publishers, 1992
239. I. D. Svalbe: "The Geometry of Basis Sets for Morphologic Closing", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-13, no. 12, Dec. 1991, pp. 1214-1224
240. S. Takriti, P. Gader: "Local Decomposition of Gray-Scale Morphological Templates", *Journal of Mathematical Imaging and Vision*, vol. 2, no. 1, Oct. 1992, pp. 39-50
241. S. H. Tan, K. K. Pang, K. N. Ngan: "Classified Perceptual Coding with Adaptive Quantization", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. CSVT-6, no. 4, Aug. 1996, pp. 375-388
242. D. Taubman and A. Zchor, "A Common Framework for Rate and Distortion Based Scalling of Highly Scalable Compressed Video", *IEEE Transactions on Circuits and Systems for Video Tehnology*, vol. 6, no. 4, Aug. 1996, pp. 329-354
243. W. Tawbi, F. Horn, E. Horlait, J. B. Stefani: "Video Compression Standards and Quality of Service", *Computer Journal*, vol 36, nr.1, Jan. 1993, pp. 43-54
244. I. Tchoukanov, R. Safaee-Rad, K. C. Smith, B. Benhabib: "The Angle-of-Sight Signature for Two-Dimensional Shape Analysis of Manufactured Objects", *Pattern Recognition*, vol. 25, no. 11, Nov. 1992, pp. 1289-1305
245. P. J. Toivanen: "Image Compression by Selecting Control Points Using Distance Function on Curved Space", *Pattern Recognition Letters*, vol. 14, June 1993, pp. 475-482
246. L. Torres, J. Huguet: "An Improvement on Codebook Search for Vector Quantization", *IEEE Transactions on Communications*, vol. 42, no. 2/3/4, Febr./March/April 1994, pp. 208-210
247. O. D. Trier, A. K. Jain: "Goal-Directed Evaluation of Binarization Methods", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-17, no. 12, Dec. 1995, pp. 1191-1201
248. O. D. Trier, T. Taxt: "Evaluation of Binarization Methods for Document Images", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-17, no. 3, March 1995, pp. 312-315

249. M. J. Tsai, J. D. Villasenor, F. Chen: "Stack-Run Image Coding", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, no. 5, Oct. 1996, pp. 519-522
250. M. E. Underwood: "Broadcast-Quality Television 45 Mbit/sec (DS3) Encoding Algorithm", *Journal of the SMPTE*, Sept. 1988, pp. 678-686
251. R. Van Den Boomgaard, A. Smeulders: "The Morphological Structure of Images: The Differential Equations of Morphological Scale-Space", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-16, no. 11, Nov. 1994, pp. 1101-1113
252. M. Van Droogenbroeck, H. Talbot: "Fast Computation of Morphological Operations with Arbitrary Structuring Elements", *Pattern Recognition Letters*, vol. 17, 1996, pp. 1451-1460
253. R. Vasiu: *Digital and High Definition Television*, University of Central Lancashire, Preston, England, 1993
254. R. Vasiu: "On the Decomposition of Convex Domains using Mathematical Morphology", *Proceedings of the 6th International Conference Optimization of Electrical and Electronic Equipment*, Braşov, România, Mai 14-15, 1998 (acceptată spre publicare)
255. R. Vasiu: "A Coding Algorithm for Gray-Scale Images using Mathematical Morphology", *Proceedings of the 6th International Conference Optimization of Electrical and Electronic Equipment*, Braşov, România, Mai 14-15, 1998 (acceptată spre publicare)
256. R. Vasiu: "On Using the Morphological Skeleton for Image Coding", *Proceedings of the 6th International Conference Optimization of Electrical and Electronic Equipment*, Braşov, România, Mai 14-15, 1998 (acceptată spre publicare)
257. R. Vasiu: "Mathematical Morphology Tools for Image Compression", *Proceedings of the conference Development and Application Systems*, Suceava, România, Mai 21-22, 1998 (acceptată spre publicare)
258. R. Vasiu, V. Gui: "A New Algorithm for Region-Based Image Compression", *Proceedings of the 6th International Conference Optimization of Electrical and Electronic Equipment*, Braşov, România, Mai 14-15, 1998 (acceptată spre publicare)
259. R. Vasiu, I. Z. Kovacs, S. Ivaşcu: "On Contours Coding for Efficient Image Compression", *Proceedings of the Symposium on Electronics and Telecommunications*, Timişoara, Sept. 26-27, 1996, vol. II, pp. 214-218
260. R. Vasiu, S. Ivaşcu, M. Mioc: "On Quality Assessment for JPEG Image Compression", *Proceedings of the Symposium on Electronics and Telecommunications*, Timişoara, Sept. 26-27, 1996, vol. II, pp. 249-252
261. P. L. Venetianer, F. Werblin, T. Roska, L. O. Chua: "Analogic CNN Algorithms for Some Image Compression and Restoration Tasks", *IEEE Transactions on Circuits and Systems-I: Fundamental Theory and Applications*, vol. CS-42, no. 5, May 1995, pp. 278-284
262. J. G. Verly, R. L. Delanoy: "Adaptive Mathematical Morphology for Range Imagery", *IEEE Transactions on Image Processing*, vol. IP-2, no. 2, April 1993, pp. 272-275
263. M. Vetterli, K. Metin-Uz: "Multiresolution Coding Techniques for Digital Television: A Review", *Journal of Multidimensional Systems and Signal Processing*, Kluwer Academic Publishers, vol. 3, no. 2/3, May 1992, pp. 53-79

264. L. Vincent: "Morphological Grayscale Reconstruction in Image Analysis: Applications and Efficient Algorithms", *IEEE Transactions on Image Processing*, vol. IP-2, no. 2, April 1993, pp. 176-201
265. J. G. Wade: *Codage et traitement du signal*, Ed. Masson, Paris, 1991
266. D. Wang, V. Haese-Coat, A. Bruno, J. Ronsin: "Some Statistical Properties of Mathematical Morphology", *IEEE Transactions on Signal Processing*, vol. SP-43, no. 8, Aug. 1995, pp. 1955-1965
267. D. Wang, J. Ronsin: "Bounded Gray-Level Morphology and Its Applications to Image Representation", *IEEE Transactions on Image Processing*, vol. IP-5, no. 6, June 1996, pp. 1067-1077
268. L. Wang, M. Goldberg: "Pyramid Transform Coding", *IEE Proceedings-I: Communications, Speech and Vision*, vol. 139, nr.3, June 1992, pp.364-371
269. M. Wang, J. Evans, L. Hassebrook, C. Knapp: "A Multistage, Optimal Active Contour Model", *IEEE Transactions on Image Processing*, vol. 5, no. 11, Nov. 1996, pp. 1586-1591
270. X. Wang, G. Bertrand: "Some Sequential Algorithms for a Generalized Distance Transformation Based on Minkowski Operations", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-14, no. 11, Nov. 1992, pp. 1114-1121
271. C. Watkins, A. Sadun, S. Marenka: *Modern Image Processing. Warping, Morphing and Classical Techniques*, Academic Press, London, 1993
272. J. Watkinson: *The Art of Digital Video*, Focal Press, London, 1994
273. J. Watkinson: *An Introduction to Digital Video*, Focal Press, London, 1994
274. J. Watkinson: *Compression in Video and Audio*, Focal Press, London, 1995
275. M. J. Weinberger, J. J. Rissanen, R. B. Arps: "Applications of Universal Context Modeling to Lossless Compression of Gray-Scale Images", *IEEE Transactions on Image Processing*, vol. 5, no. 4, April 1996, pp. 575-586
276. W. J. Welsh, S. Searby, J. B. Waite: "Model-Based Image Coding", *British Telecom Technology Journal*, vol. 8, no. 3, July 1990, pp. 94-106
277. W. J. Welsh: "Model-Based Coding of Videophone Images", *Electronics & Communication Engineering Journal*, 1991, pp. 29-36
278. P. Willemin, T.R. Reed, M. Kunt: "Image Sequence Coding by Split and Merge", *IEEE Transactions on Communications*, vol. 39, no. 12, Dec. 1991, pp. 1845-1855
279. S. S. Wilson: "Theory of Matrix Morphology", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-14, no. 6, June 1992, pp. 636-652
280. I. H. Witten, T. C. Bell, H. Emberson, S. Inglis, A. Moffat: "Textual Image Compression: Two-Stage Lossy/Lossless Encoding of Textual Images", *Proceedings of the IEEE*, vol. 82, no. 6, June 1994, pp. 878-888
281. J. W. Woods, S. D. O'Neil: "Subband Coding of Images", *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. ASSP-34, no. 5, Oct. 1986, pp. 1278-1288
282. X. Wu, Y. Fang: "A Segmentation-Based Predictive Multiresolution Image Coder". *IEEE Transactions on Image Processing*, vol. 4, no. 1, Jan. 1995, pp. 34-47
283. X. Wu, L. Guan: "Acceleration of the LBG Algorithm", *IEEE Transactions on Communications*, vol. 42, no. 2/3/4, Febr./March/April 1994, pp. 1518-1523
284. J. Xu: "Decomposition of Convex Polygonal Morphological Structuring Elements into Neighborhood Subsets", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-13, no. 2, Febr. 1991, pp. 153-162

- 
285. P. Yu, A. N. Venetsanopoulos: "Hierarchical Finite-State Vector Quantization for Image Coding", *IEEE Transactions on Communications*, vol. 42, no. 11, Nov. 1994, pp. 3020-3026.
  286. S. S. Yu, N. P. Galatsanos: "Binary Decompositions for High-Order Entropy Coding of Grayscale Images", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. CSVT-6, no. 1, Febr. 1996, pp. 21-30
  287. P. C. Yuen, Y. Y. Wong and C. S. Tong, "Contour Detection Using Enhanced Snakes Algorithm", *Electronic Letters*, vol. 32, no. 3, 1 Febr. 1996, pp. 202-204
  288. W. J. Zeng, Y. F. Huang, S. C. Huang: "Two Greedy Tree Growing Algorithms for Designing Variable Rate Vector Quantizers", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 5, no. 3, June 1995, pp. 236-242
  289. Y. Q. Zhang, W. W. Wu, K. S. Kim, R. L. Pickholtz, J. Ramasastry: "Variable Bit-Rate Video Transmission in the Broadband ISDN Environment", *Proceedings of the IEEE*, vol. 79, no. 2, febr. 1991, pp. 214-221
  290. R. W. Zhou, C. Quek, G. S. Ng: "A Novel Single-Pass Thinning Algorithm and an Effective Set of Performance Criteria", *Pattern Recognition Letters*, vol. 16, 1995, pp. 1267-1275
  291. X. Zhuang, R. M. Haralick: "Morphological Structuring Element Decomposition". *Computer Vision, Graphics and Image Processing*, col. 35, 1986, pp. 370-382
  292. M. A. Zmuda, L. A. Tamburino: "Efficient Algorithms for the Soft Morphological Operators", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-18, no. 11, Nov. 1996, pp. 1142-1147