

UNIVERSITATEA "POLITEHNICA"
TIMIȘOARA

BIBLIOTECA CENTRALĂ

EHNICĂ "POLITEHNICA" TIMIȘOARA
AUTOMATICĂ ȘI CALCULATOARE

Nr. inv. 45.477

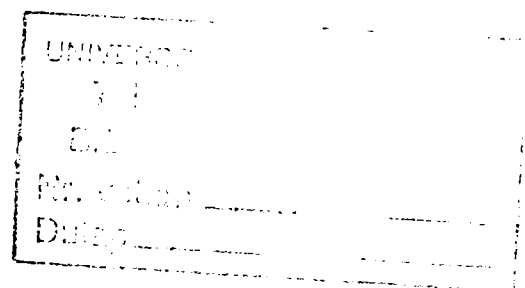
Dulap Lit. C

CONTRIBUȚII LA SELECTAREA CARACTERISTICILOR ÎN PROCESUL DE RECUNOAȘTERE ȘI INTERPRETARE A IMAGINILOR

- TEZĂ DE DOCTORAT -

BIBLIOTECA CENTRALĂ
UNIVERSITATEA "POLITEHNICA"
TIMIȘOARA

Conducător științific
Prof. Dr. Ing. Ștefan Holban



Doctorand
Ing. Robert Ștefan Györödi

CUPRINS

1. Introducere	6
1.1. Domeniul abordat	6
1.2. Aspecte ale recunoașterii feței	6
1.3. Scopul lucrării	7
1.4. Conținutul tezei	7
2. Concepte fundamentale ale recunoașterii și interpretării imaginilor	9
2.1. Introducere	9
2.2. Etape ale prelucrării de imagini	9
2.3. Sisteme bidimensionale și noțiuni matematice de bază	11
2.3.1.1. Distribuția gaussiană și normală	12
2.4. Operații de bază folosite în prelucrarea imaginilor	12
2.4.1. Histograma nivelelor de gri	12
2.4.2. Histograma bi-dimensională	13
2.4.3. Proprietățile histogramelor	13
2.4.4. Moduri de utilizare a histogramelor	14
2.4.4.1. Parametri de digitizare	14
2.4.4.2. Selecția pragului pentru delimitare	14
2.4.4.3. Densitatea optică integrată	14
2.4.5. Relația dintre histogramă și imagine	15
2.4.5.1. Cazul unidimensional	15
2.4.5.2. Cazul bidimensional	16
2.4.6. Operații punctuale	18
2.4.6.1. Aplicații ale operațiilor punctuale	18
2.5. Teoria sistemelor liniare	20
2.5.1. Generalități	20
2.5.2. Analiza semnalelor armonice și a semnalelor complexe	22
2.5.2.1. Semnale armonice	22
2.5.2.2. Răspunsul la o intrare armonică	22
2.5.2.3. Semnale armonice și sinusoidale	24
2.5.2.4. Funcția de transfer	24
2.5.3. Operația de convoluție	25
2.5.3.1. Convoluția unidimensională	26
2.5.4. Convoluția unidimensională discretă	28
2.5.5. Convoluția bidimensională	29
2.5.6. Aplicații ale convoluției	29
2.6. Concluzii	30
3. Metode de selectare a caracteristicilor în procesul de recunoaștere și interpretare a imaginilor	32
3.1. Introducere	32
3.2. Metode de selectare a caracteristicilor	34
3.2.1. Transformări diagonale	34

3.2.2.	Transformări rotaționale	36
3.2.3.	Transformarea Karhunen - Loève	37
3.2.4.	Transformata Fourier	40
3.2.5.	Transformarea Cosinus discretă	42
3.2.6.	Transformarea Sinus discretă	43
3.2.7.	Transformarea Hartley	44
3.2.8.	Transformarea Haar	45
3.2.9.	Transformata Wavelet	47
3.2.9.1.	Introducere	47
3.2.9.2.	Analiza timp – frecvență	48
3.2.10.	Tipuri de transformări Wavelet	49
3.2.10.1.	Transformarea wavelet continuă	50
3.2.10.2.	Transformarea Wavelet continuă bidimensională	51
3.2.10.3.	Interpretarea cu ajutorul bancurilor de filtre	52
3.2.11.	Implementarea cu bancuri de filtre (cazul bidimensional)	55
3.2.12.	Dezvoltarea în serie wavelet	55
3.2.12.1.	Wavelet diadic	55
3.2.12.2.	Funcții Wavelet diadice compacte	57
3.2.13.	Transformarea wavelet discretă (DWT)	58
3.2.13.1.	Teoria bancurilor de filtre	58
3.2.13.2.	Filtre ideale trece-bandă	59
3.2.13.3.	Filtre trece-bandă netede	60
3.2.14.	Analiza multirezoluție	61
3.2.14.1.	Algoritmul piramidal	62
3.2.14.2.	Codarea piramidală laplaciană	63
3.3.	Concluzii	64
4.	Detectarea și extragerea caracteristicilor feței umane	65
4.1.	Introducere	65
4.2.	Proiectarea filtrelor	66
4.2.1.	Mediere și filtrare trece-jos	66
4.2.1.1.	Accentuarea conturilor	68
4.2.2.	Filtrare trece-bandă și oprește-bandă	68
4.2.2.1.	Filtru trece-bandă	68
4.2.2.2.	Filtru oprește-bandă	69
4.2.3.	Filtrare liniară generalizată	70
4.2.4.	Filtrare neliniară	70
4.3.	Recunoașterea feței pe bază de caracteristici	71
4.4.	Segmentarea imaginilor	72
4.4.1.	Segmentarea prin delimitarea cu prag (fereastră)	72
4.4.2.	Segmentarea prin extragerea conturului	73
4.4.2.1.	Detectia de contur	73
4.4.2.2.	Operatori Laplace și metoda trecerilor prin zero	75
4.4.2.3.	Extragerea conturilor	76
4.4.3.	Segmentarea prin extragerea regiunilor	77
4.4.4.	Segmentarea pe baza informației de culoare	79
4.4.4.1.	Analiza componentelor conectate	81
4.4.4.2.	Evaluarea informației de formă	81
4.5.	Extragerea caracteristicilor feței	84
4.5.1.	Accentuarea caracteristicilor feței	84

4.5.2.	Extragerea caracteristicilor feței prin analiza min-max	85
4.5.3.	Evaluarea extracției caracteristicilor feței prin analiza min-max	91
5.	CTAG - O metodă propusă de abordare a texturilor color în procesul de detectare a feței umane	93
5.1.	Introducere	93
5.2.	Calculul graficelor $N_s(t)$	94
5.3.	Proprietățile reprezentării MRCG pentru texturi.	95
5.3.1.	MRCG supus unor transformări monotone ale nivelului de gri.	95
5.3.2.	Efectul netezirii/întăririi imaginii asupra graficelor $N_s(t)$.	96
5.3.3.	Influența unor caracteristici de bază ale texturilor asupra MRCG.	97
5.4.	Aspecte de implementare	100
6.	Metode de detectare și recunoaștere a feței umane	102
6.1.	Introducere	102
6.2.	Metode bazate pe caracteristici geometrice	102
6.3.	Metode bazate pe șabloane	103
6.3.1.	Metode bazate pe corelare	103
6.3.2.	Metode bazate pe analiza componentelor principale	104
6.3.2.1.	Metoda fețelor proprii (eigenfaces)	104
6.3.2.2.	Metoda caracteristicilor proprii (eigenfeatures)	104
6.3.3.	Metoda bazată pe rețele neuronale	105
6.4.	Metode bazate pe modele	105
6.4.1.	Metode bazate pe modele Markov ascunse	105
6.5.	Concluzii	107
7.	Detectarea și recunoașterea feței folosind modelul Markov ascuns încapsulat	108
7.1.	Introducere	108
7.2.	HMM încapsulat	109
7.2.1.	Vectorii de observație	111
7.2.2.	Antrenarea modelelor faciale	112
7.2.3.	Detectarea feței	114
7.2.4.	Recunoașterea feței	117
7.3.	Rezultate experimentale și concluzii	118
8.	Concluzii	120
8.1.	Contribuții	120
8.2.	Direcții viitoare de cercetare	121
8.3.	Concluzii finale	121
9.	Bibliografie	124

Glosar

HMM	- Model Markov ascuns (Hidden Markov Model)
eHMM	- Model Markov ascuns încapsulat (Embedded Hidden Markov Model)
DCT	- Transformata cosinus discretă
KLT	- Transformata Karhunen Loeve
FFT	- Transformata Fourier rapidă
MRCG	- Multi Resolution Clusters Graph
CTAG	- Graful de Adiacență al Texturilor Color (Color Texture Adjacency Graph)
DWT	- Transformarea Wavelet discretă

Prefață

De-a lungul ultimilor ani s-a observat un interes crescând pentru prelucrarea, compresia, recunoașterea și analiza de imagini.

Un domeniu major de aplicație al tehnicilor de prelucrare a imaginilor este analiza automată a acestora. Scopul, în acest caz, este extragerea informației într-o formă cât mai potrivită pentru prelucrarea pe calculator.

În orice sistem automat, înaintea extragerii efective a informației, sunt aplicate diverse operații de corectare, care încearcă aducerea imaginilor într-o formă cât de cât unitară, în lucrare prezentându-se câteva dintre cele mai utilizate operații în acest sens.

Ținând cont pe de o parte de domeniul vast al extragerii de caracteristici în prelucrarea și interpretarea automată a imaginilor, precum și de posibilele aplicații practice de interes pe care le are detectarea și recunoașterea fețelor umane, în această lucrare s-au luat în considerare tehnicile de extragere a caracteristicilor care au un impact major asupra acestui din urmă subdomeniu, și anume extragerea de caracteristici în vederea detectării și recunoașterii fețelor umane.

Ca metodă de reprezentare a caracteristicilor și recunoaștere este prezentată o abordare folosind un HMM încapsulat.

Comparat cu metodele bazate pe șablon sau cele HMM standard pentru detectarea feței această abordare este mai flexibilă în raport cu variații de scară și deformări naturale, atât pe plan vertical cât și orizontal.

În modelul HMM încapsulat, caracteristicile extrase formează vectori de observație, care sunt folosiți în procesul de detecție și recunoaștere a feței. Spre deosebire de Samaria [68] și Nefian [70] care au folosit culori de pixeli respectiv transformata cosinus discretă (DCT) pentru a calcula elementele vectorilor de observație, în această lucrare am propus folosirea metodei CTAG pentru calcularea elementelor vectorilor de observație.

Îmbunătățirea propusă în această lucrare este luarea în considerație nu numai a culorii ci și a texturii obiectelor (regiunilor), reprezentării propuse fiindu-i dată denumirea de Graful de Adiacență al Texturilor Color (CTAG) [30]. În această reprezentare fiecare nod conține descrierea de culoare și textură a regiunii respective. Pentru această reprezentare au existat anumite cerințe și anume nu trebuie să fie consumatoare de timp și nu trebuie să fie consumatoare de memorie. Față de [30] unde am folosit această metodă de extragere de caracteristici în conjuncție cu o structură propusă de Mattas [9], aici am folosit-o ca și metodă de extragere de caracteristici de bază.

Din experimentele efectuate un HMM încapsulat are performanțe la fel de bune sau mai bune decât alte metode având totodată și complexitate redusă de calcul și asta pentru o gamă largă de metode de extragere de caracteristici (CTAG, DCT, KLT, valori de pixeli), prin aceasta confirmând rezultatele obținute atât de Samaria [68] cât și de Nefian [70].

1. Introducere

1.1. Domeniul abordat

Recunoașterea formelor are ca obiect clasificarea unei mulțimi de obiecte, procese sau evenimente, indiferent care ar fi natura lor. Procesul de recunoaștere nu se limitează doar la clasificarea formei respective ci el include și metode capabile să descrie acele aspecte ale formelor care le fac să nu poată fi atribuite altor clase. Un exemplu tipic pentru astfel de probleme de recunoaștere a formelor îl constituie recunoașterea imaginilor. Aici formele care fac obiectul procesului de clasificare sunt extrem de complexe, numărul de caracteristici fiind foarte mare.

Deoarece domeniul de recunoaștere și interpretare a imaginilor este foarte mare, în această lucrare am încercat tratarea selectării de caracteristici în cazul unui sistem de recunoaștere/confirmare a identității bazate pe caracteristicile feței umane.

Identificarea automată a fețelor umane este o arie activă de cercetare de peste 25 de ani. Deși recunoașterea fețelor nu prezintă dificultăți pentru cei mai mulți dintre oameni, sistemele automate de recunoaștere a fețelor nu ating performanțe satisfăcătoare datorită unui număr mare de variații posibile în aparițiile faciale și a similarității dintre diverse fețe. Acest obiectiv este considerat ca și una dintre problemele fundamentale ale prelucrării și interpretării imaginilor.

În afara importanței recunoașterii feței din punctul de vedere al cercetării, aceasta are un număr mare de aplicații comerciale. De exemplu, identificarea persoanelor pe baza feței, operație efectuată pe baza unor măsurători indirecte ceea ce are numeroase avantaje asupra tehnicilor folosind măsurători directe (amprente) sau apropiate (de exemplu textura irisului).

Aplicațiile de recunoașterea feței care sunt momentan implementate includ potrivirea statică a unor fotografii realizate într-un mediu controlat. O mărire semnificativă a numărului de servicii disponibile prin internet a ridicat problema unor tranzacții monetare securizate. De exemplu, pentru a realiza identificarea unei persoane deținătoare a unui card în magazine, un număr valid de carte de credit trebuie să fie însoțit de semnătura deținătorului. Corecta identificare a persoanei prin internet nu poate fi realizată prin mijloace convenționale. O posibilă soluție ar fi utilizarea recunoașterii feței folosind o cameră de luat vederi.

Sistemele de recunoaștere a feței pot fi utilizate la verificarea accesului persoanelor într-o zonă restrictivă sau la recunoașterea lor într-o zonă specifică, cum ar fi bănci sau sedii de firme etc.

1.2. Aspecte ale recunoașterii feței

O caracterizare generală a operației de recunoaștere a feței poate fi descrisă ca identificarea uneia sau mai multor persoane din imagini statice sau video ale unei scene folosind o bază de date de fețe. Soluția problemei implică segmentarea fețelor din scene complexe, extragerea de caracteristici din regiunea facială, identificarea și recunoașterea.

Un prim pas către recunoașterea feței este localizarea feței ca și regiuni și extragerea caracteristicilor faciale ca ochi, gură și nas. Cu toate că multe lucruri s-au făcut deja în această direcție de cercetare, recunoașterea feței umane în cadrul unor scene cu fundal difuz este încă o problemă care necesită mai departe investigare.

Există multe tipuri de aplicații pentru sistemele de localizarea și recunoaștere, de exemplu modele video bazate pe codare, sisteme de securitate, potrivire de personaje, etc. În contextul unor variații de lumină, fundale, unghiuri vizuale și expresii faciale problema devine complexă. Pentru detecția regiunilor faciale și a caracteristicilor faciale au fost

publicate până acum câteva abordări folosind intensități, forme și informații a culorii sau combinații ale acestora. De exemplu, extragerea regiunilor faciale din fundale complexe este făcută pe baza informației de culoare și de textură.

Culoarea are un mare avantaj asupra altor metode, deoarece este în mare măsură independentă de punctul de vedere și rezoluție, dar pentru o recunoaștere corectă trebuie să mai folosim și alte caracteristici.

1.3. Scopul lucrării

Îmbunătățirea propusă în această lucrare este luarea în considerație nu numai a culorii ci și a texturii obiectelor (regiunilor). În acest fel, reprezentarea propusă am numit-o Graful de Adiacență al Texturilor Color (CTAG) [30], în care fiecare nod conține descrierea de culoare și textură a regiunii respective. Pentru această reprezentare au existat anumite cerințe și anume nu trebuie să fie consumatoare de timp și nu trebuie să fie consumatoare de memorie.

Față de [30] unde am folosit această metodă de extragere de caracteristici în conjuncție cu o structură propusă de Mattas [9], aici am folosit-o ca și metodă de extragere de caracteristici de bază.

Din experimentele efectuate, un HMM încapsulat are performanțe la fel de bune sau mai bune decât alte metode având totodată și complexitate redusă de calcul și asta pentru o gamă largă de metode de extragere de caracteristici (CTAG, DCT, KLT, valori de pixeli), prin aceasta confirmând rezultatele obținute atât de Samaria [60] cât și de Nefian [70].

Scopurile acestei lucrări pot fi sumarizate astfel:

- Introducerea unei noi metode de extragere de caracteristici, bazată pe o caracterizare topologică a texturilor – metodă pe care am denumit-o CTAG.
- Prezentarea modelului Markov ascuns încapsulat (eHMM), model folosit în recunoașterea fețelor.
- Obținerea elementelor vectorului de observație a HMM prin metoda propusă (CTAG).
- Demonstrarea caracteristicilor abordării propuse (HMM +CTAG) printr-o serie de experimentări.
- Studiarea modului în care este afectată performanța de recunoaștere prin modificarea parametrilor modelului propus.
- Compararea modelului propus cu modelele propuse de Samaria și Nefian aplicându-le pe același set de date, și anume baza de date ORL.
- Prezentarea principalelor metode de extragere a caracteristicilor folosite în prelucrarea imaginilor și implementarea unora dintre acestea în vederea incorporării într-un sistem de recunoaștere/confirmare a identității.
- Prezentarea unora dintre metodele de detectare a feței umane.

1.4. Conținutul tezei

Lucrarea prezentată are ca scop detectarea și recunoașterea feței folosind o abordare bazată pe modelul HMM încapsulat. Lucrarea este structurată pe opt capitole.

Capitolul doi face o introducere în prelucrarea imaginilor, evidențiind etapele fundamentale în prelucrarea digitală de imagini și prezentând câteva noțiuni matematice de bază care vor fi utilizate în această lucrare. Totodată în acest capitol sunt descrise operațiile de bază folosite în prelucrarea imaginilor, cum ar fi histogramele, operațiile punctuale și teoria sistemelor liniare.

Capitolul trei descrie câteva dintre metodele de selectare a caracteristicilor în procesul de recunoaștere și interpretare a imaginilor. Sunt tratate pe scurt transformările des întâlnite în prelucrarea de imagini și se prezintă efectul acestor transformări asupra câtorva imagini de test.

Capitolul patru prezintă câteva metode de detectare și extragere a caracteristicilor feței umane, cum ar fi detectarea și extragerea conturilor, detectarea regiunilor, detectarea folosind informația de culoare.

Capitolul cinci descrie metoda propusă CTAG de descriere a texturilor color folosită în procesul de detectare și recunoaștere. Deasemenea se prezintă aspectele de implementare și câteva caracteristici ale modelului.

Capitolul șase prezintă principalele tipuri de metode de detectare și recunoaștere întâlnite în literatura de specialitate și prezintă concluziile referitoare la acestea.

Capitolul șapte descrie modelul Markov ascuns încapsulat și prezintă sistemul de detectare și recunoaștere a feței folosind acest model așa cum este prezentat de Samaria și Nefian, precum și modificările propuse acestui model prin folosirea metodei CTAG pentru calcularea vectorilor de observație. Totodată sunt prezentate și rezultatele comparative ale modelelor prezentate vizavi de modelul propus, precum și influența modificării parametrilor în cazul metodei CTAG.

Concluziile, rezultatele experimentale și recomandările pentru viitoarele cercetări sunt prezentate în capitolul opt.

2. Concepte fundamentale ale recunoașterii și interpretării imaginilor

2.1. Introducere

Prelucrarea imaginilor este un domeniu care în ultimi ani a cunoscut o rapidă dezvoltare. De-a lungul ultimilor ani s-a observat un interes crescând pentru prelucrarea imaginilor color, compresia, recunoașterea și analiza de imagini.

În medicină, spre exemplu, metodele computerizate de prelucrare fac posibilă îmbunătățirea contrastului sau codarea intensităților (nivelelor de gri) imaginilor monocrome în culori, pentru ușurarea interpretării radiografiilor sau altor tipuri de imagini biomedicale.

Un alt domeniu major de aplicație al tehnicilor de prelucrare a imaginilor este analiza lor cu calculatorul. Scopul, în acest caz, este extragerea informației într-o formă cât mai potrivită pentru prelucrarea pe calculator.

2.2. Etape ale prelucrării de imagini

Termenul de imagine monocromă (imagine alb-negru) sau, simplu imagine, se referă la funcția de intensitate bidimensională $f(x,y)$, unde x și y reprezintă coordonatele spațiale iar f este funcția proporțională în fiecare punct (x,y) cu strălucirea sau cu nivelul de gri al imaginii în acel punct.

O imagine digitală este o imagine $f(x,y)$ care a fost discretizată atât în coordonate spațiale cât și ca strălucire. Ea poate fi considerată ca o matrice la care indicii rândurilor și coloanelor identifică un punct din imagine, iar elementul corespunzător al matricii reprezintă nivelul de gri în punctul respectiv. Elementele unei asemenea rețele digitale sunt numite elemente de imagine sau pixeli.

Prelucrarea digitală a imaginilor presupune o succesiune de etape de procesare hard și soft, precum și de implementare a unor metode teoretice. Etapele fundamentale în prelucrarea digitală de imagini sunt prezentate în figura de mai jos Fig. 2.2-1 :

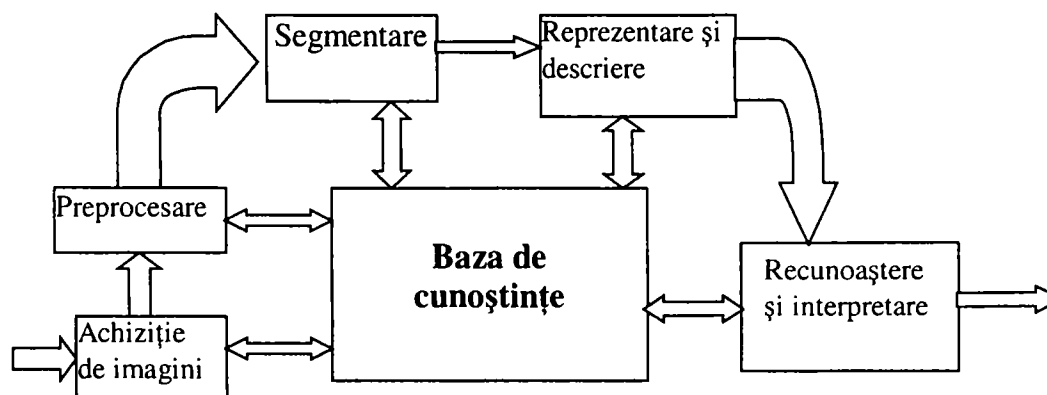


Fig. 2.2-1 Etape fundamentale în prelucrarea digitală de imagini

Prima etapă în acest proces este achiziția de imagine. Aceasta necesită un senzor de imagine precum și posibilitatea de a digitaliza semnalul de la ieșirea acestuia. Senzorul poate fi o cameră de televiziune alb-negru sau color, care produce o imagine completă sau o cameră cu scanare liniară, producând la un moment dat o singură linie dintr-o imagine; în acest caz, mișcarea cu viteză uniformă a obiectului prin fața scannerului duce la obținerea unei imagini

bidimensionale. Dacă la ieșirea camerei rezultă un semnal analogic, este necesară digitizarea lui cu ajutorul unui convertor analog-numeric.

După obținerea imaginii digitale, următoarea etapă este preprocesarea acesteia. Scopul principal al procesării este îmbunătățirea imaginii în sensul măririi șanselor de succes ale prelucrărilor ulterioare. De exemplu, procesarea poate consta în aplicarea unor algoritmi de îmbunătățire a contrastului, curățare de zgomot și izolare a regiunilor a căror textură indică o asemănare cu informația alfanumerică avută în vedere.

Etapă următoare constă în segmentarea imaginii, respectiv împărțirea acesteia în părți constituente sau în obiecte distincte. În general, segmentarea este una dintre cele mai dificile etape în prelucrarea digitală de imagini. Pe de o parte, algoritmi sofisticati de segmentare, cu șanse mari de selectare a regiunilor de interes, necesită calcule complicate și, deci un timp îndelungat. Pe de altă parte, algoritmi simpli sau insuficient elaborați nu se pot aplica cu șanse mari de succes.

Rezultatul segmentării este, de obicei un șir de intensități ale pixelilor, reprezentând fie conturul regiunii de interes, fie întreaga regiune. În fiecare caz este necesară conversia datelor într-o formă mai potrivită pentru prelucrarea pe calculator. Prima decizie ce trebuie luată este dacă datele trebuie să fie reprezentate ca un contur sau ca o regiune completă: reprezentarea ca și contur este necesară în cazul în care prelucrarea urmărește evidențierea caracteristicilor de formă (colțuri, inflexiuni, etc), pe când reprezentarea ca regiuni este mai potrivită când se urmărește studiul caracteristicilor interne ale regiunii, cum ar fi textura sau structura internă. În unele aplicații cele două tipuri de reprezentare coexistă.

Alegerea modalității de reprezentare a datelor este urmată de specificarea metodei de descriere a acestora, astfel încât caracteristicile de interes să fie scoase în evidență cu prioritate.

Descrierea sau selecția caracteristicilor reprezintă procesul de selectare a caracteristicilor, având ca rezultat o informație cantitativă sau caracteristici ce diferențiază o clasă de obiecte de altele.

Ultima etapă este constituită de recunoașterea și interpretarea datelor. Recunoașterea reprezintă procesul de clasificare a unui obiect într-o anumită categorie, pe baza informației rezultate în urma imaginii segmentate. Interpretarea presupune găsirea unei anumite semnificații pentru un ansamblu de obiecte recunoscute.

Cunoștințele necesare pentru rezolvarea unei anumite probleme prin prelucrarea digitală de imagini sunt codificate în sistem sub forma unei baze de date. Cunoștințele pot fi de tip simplu, de exemplu detalierea unei regiuni dintr-o imagine, unde se presupune că este localizată informația de interes, limitându-se în acest fel procedura de căutare a respectivei informații. Cunoștințele pot avea însă și un caracter mult mai complex, ca de exemplu o listă a defectelor interne mai importante ale unor produse, în aplicații de tipul unei inspecției automate sau o bază de date de imagini conținând imagini de înaltă rezoluție. Pe lângă faptul că dirijează operațiile din fiecare modul de prelucrare, baza de cunoștințe are și rolul de a controla interacțiunea dintre diferitele module. Distincția este făcută în figura Fig. 2.2-1 prin săgețile cu vârf dublu dintre baza de cunoștințe și modulele de prelucrare, spre deosebire de săgețile cu un singur vârf ce fac legătura între modulele de prelucrare propriu-zisă.

Baza de cunoștințe permite și asigurarea unei reacții între module, pe lângă dirijarea operațiilor în modulele de prelucrare. O cerere de reacție, survenită prin intermediul bazei de cunoștințe spre modulul de segmentare, poate solicita o nouă segmentare, eventual cu alți parametri, a imaginii propuse spre analiză. Vizualizarea imaginii rezultate în urma prelucrării se poate face după fiecare etapă a acesteia.

Este de remarcat că majoritatea aplicațiilor prelucrărilor de imagini au o structură mult mai simplă și nu necesită interacțiuni de complexitatea celor prezentate în figura Fig. 2.2-1. De fapt, în unele cazuri nici nu sunt necesare toate modulele prezentate în figură. În general, prelucrările care includ recunoaștere și interpretare sunt asociate cu aplicații de analiză a

imaginilor, unde obiectivul este extragerea automată sau semiautomată de informații din imagini.

2.3. Sisteme bidimensionale și noțiuni matematice de bază

În acest paragraf se vor prezenta câteva noțiuni matematice de bază care vor fi utilizate în această lucrare. Deoarece imaginile sunt sisteme bidimensionale, conceptele matematice vor respecta caracteristicile acestora.

Un semnal unidimensional continuu poate fi reprezentat ca o funcție de o singură variabilă: $f(x)$, $u(x)$, $s(t)$, iar un semnal unidimensional eșantionat se scrie: u_n sau $u(n)$.

O imagine continuă poate fi reprezentată ca o funcție de două variabile independente: $u(x,y)$, $v(x,y)$, $f(x,y)$, etc. O imagine eșantionată poate fi considerată ca o secvență bidimensională (sau multidimensională) de numere reale: $u_{m,n}$, $v(m,n)$, $u(i,j)$, etc. Simbolurile $i, j, k, l, m, n \dots$ pot fi folosite pentru a specifica indicii întregi ai tablourilor și ai vectorilor. Complex conjugata unei variabile complexe z , va fi notată cu z^* .

Transformările bidimensionale cum este, de exemplu transformata Fourier sunt fundamentale pentru prelucrările digitale de imagini.

În domeniul unidimensional, transformata Fourier a unei funcții complexe $f(x)$ este definită ca:

$$F(\xi) = F[f(x)] = \int_{-\infty}^{\infty} f(x) \exp(-j2\pi \cdot \xi \cdot x) dx \quad (2.1)$$

Transformata inversă a lui $F(\xi)$ este:

$$f(x) = F^{-1}[F(\xi)] = \int_{-\infty}^{\infty} F(\xi) \exp(j2\pi \cdot \xi \cdot x) dx \quad (2.2)$$

Transformata bidimensională directă și inversă sunt definite în mod asemănător:

$$F(\xi_1, \xi_2) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \exp(-j2\pi \cdot (x\xi_1 + y\xi_2)) dx dy \quad (2.3)$$

$$f(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F(\xi_1, \xi_2) \exp[j2 \cdot \pi(x\xi_1 + y\xi_2)] dx dy \quad (2.4)$$

Dacă $f(x,y)$ reprezintă luminanța și x, y coordonatele spațiale, atunci ξ_1, ξ_2 sunt frecvențele spațiale care reflectă schimbările de luminanță în funcție de distanțele spațiale.

Semnale aleatoare

Un semnal aleator discret sau un proces aleator discret este o secvență de variabile aleatoare $u(n)$. Pentru secvențe aleatoare discrete putem defini, următoarele caracteristici:

Media:

$$\mu_n(n) = \mu(n) = E[u(n)] \quad (2.5)$$

Varianța:

$$\sigma_u^2(n) = \sigma^2(n) = E[|u(n) - \mu(n)|^2] \quad (2.6)$$

Covarianța:

$$\text{Cov}[u(n), u(n')] = r_u(n, n') = r(n, n') = E\{[u(n) - \mu(n)][u^*(n') - \mu^*(n')]\} \quad (2.7)$$

Covarianța a doi vectori:

$$\text{Cov}[u(n), v(n')] = r_{uv}(n, n') = E\{[u(n) - \mu_u(n)][v^*(n') - \mu_v^*(n')]\} \quad (2.8)$$

Autocorelația:

$$a_{uu}(n, n') = a(n, n') = E\{[u(n)u^*(n')]\} = r(n, n') + \mu(n)\mu^*(n') \quad (2.9)$$

Corelația a doi vectori:

$$a_{uv}(n, n') = E[u(n)v^*(n')] = r_{uv}(n, n') + \mu_u(n)\mu_v^*(n') \quad (2.10)$$

Simbolul E are semnificația operatorului matematic de estimare. Pentru un vector u de dimensiune $N \times 1$, covarianța și celelalte proprietăți pot fi definite astfel:

$$E[u] = \mu = \{\mu(n)\} \text{ este un vector de dimensiune } N \times 1 \quad (2.11)$$

$$Cov[u] = E(u - \mu)(u^* - \mu^*)^T = R_u = R = \{r(n, n')\} \text{ este o matrice } N \times N \quad (2.12)$$

$$Cov[u, v] = E(u - \mu_u)(v^* - \mu_v^*)^T = R_{uv} = \{r_{uv}(n, n')\} \text{ este o matrice } N \times N \quad (2.13)$$

În acest caz, μ și R reprezintă vectorul medie, respectiv matricea de covarianță a vectorului u .

2.3.1.1. Distribuția gaussiană și normală

Funcția densitate de probabilitate a unei variabile aleatoare u se notează $p_u(u)$. Pentru o variabilă aleatoare de tip gaussian avem:

$$p_u(u) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{|u - \mu|^2}{2\sigma^2}\right\} \quad (2.14)$$

unde μ și σ^2 reprezintă valoarea medie și varianța, iar u este valoarea luată de variabila aleatoare. Dacă $\mu = 0$ și $\sigma^2 = 1$, atunci distribuția se numește normală (standard).

O secvență, posibil infinită, se numește **proces aleator gaussian** (sau normal) dacă densitatea de probabilitate a oricărei subsecvențe este o distribuție gaussiană. De exemplu, pentru o secvență $\{u(n), 1 \leq n \leq N\}$, densitatea de probabilitate va fi:

$$p_u(u) = p_u(u_1, u_2, \dots, u_N) = [(2\pi)^{N/2} |R|^{1/2}]^{-1} \exp\{-1/2(u - \mu)^* R^{-1} (u - \mu)\} \quad (2.15)$$

unde R este matricea de covarianță a vectorului u și se presupune a fi nesingulară.

2.4. Operații de bază folosite în prelucrarea imaginilor

2.4.1. Histograma nivelelor de gri

Una dintre cele mai simple și de folos unelte în procesarea imaginilor este histograma nivelurilor de gri. Această funcție sumarizează conținutul nivelelor de gri dintr-o imagine.

Histograma nivelelor de gri este o funcție care arată, pentru fiecare nivel de gri, numărul de pixeli din imagine care au acel nivel de gri. Pe abscisă este reprezentat nivelul de gri iar pe ordonată frecvența de apariție.

Există și un alt mod de a defini histograma nivelelor de gri. Presupunând că avem o imagine continuă, definită de funcția $D(x, y)$, care variază lin de la nivel de gri ridicat la centru spre nivel de gri scăzut la margini. Putem selecta un nivel de gri D_1 și defini un set de linii de contur conectând toate punctele din imagine care au valoarea D_1 . Rezultatul, adică liniile de contur, formează curbe închise care înconjoară regiuni în care nivelul de gri este mai mare sau egal cu D_1 .

Funcția de arie cu un prag $A(D)$ a unei imagini continue este aria mărginită de toate liniile de contur având nivelul gri D . Astfel, histograma poate fi definită în următorul mod:

$$H(D) = \lim_{\Delta D \rightarrow 0} \frac{A(D) - A(D + \Delta D)}{\Delta D} = -\frac{d}{dD} A(D) \quad (2.16)$$

Astfel histograma unei imagini continue este negativul derivatei funcției sale de arie. Semnul minus rezultă din faptul că $A(D)$ descrește cu creșterea lui D . Dacă considerăm imaginea ca și o variabilă aleatoare având două dimensiuni, funcția de arie este proporțională cu funcția sa de distribuție cumulativă, iar histograma nivelelor de gri cu funcția sa de densitate a probabilității.

În cazul funcțiilor discrete, fixăm ΔD la unitate, astfel ecuația (2.16) devine:

$$H(D) = A(D) - A(D+1) \quad (2.17)$$

Funcția de arie a unei imagini digitale este aproximativ egală cu nivelul de pixeli având nivelul de gri mai mare sau egal cu D pentru orice nivel de gri D .

2.4.2. Histograma bi-dimensională

În mod frecvent, este folosită construcția unor histograme de dimensiuni mai mari decât una singură. Acestea sunt folosite mai ales pentru imagini color.

Histograma bi-dimensională este o funcție de două variabile: nivele de gri în imaginea cu filtru roșu și cel cu filtru albastru (de exemplu). Valoarea sa în punctul de coordonate (D_R , D_B) este numărul de perechi de pixeli având nivelul de gri D_R în imaginea "roșie" și respectiv D_B în imaginea "albastră".

2.4.3. Proprietățile histogramelor

Când o imagine este condensată într-o histogramă, toate informațiile spațiale sunt ignorate. Histograma specifică numărul de pixeli pentru fiecare nivel de gri, dar nu dă nici o informație despre locul unde acei pixeli sunt localizați în imagine. Astfel histograma este unică unei imagini, dar reversul nu este valabil, adică imagini foarte diferite pot avea histograme identice.

Dacă schimbăm variabilele, în ecuația (2.16) și integrăm ambele părți de la D la infinit, obținem:

$$\int_0^{\infty} H(P)dP = -[A(P)]_0^{\infty} = A(0) \quad (2.18)$$

adică funcția de arie. Dacă setăm $D = 0$, presupunând nivelele de gri nenegative, obținem:

$$\int_0^{\infty} H(P)dP = \text{aria imaginii} \quad (2.19)$$

sau în cazul discret:

$$\sum_{D=0}^{255} H(D) = NL \times NS \quad (2.20)$$

unde NL și NS sunt numărul de linii și respectiv coloane din imagine.

Dacă o imagine conține un singur obiect având un nivel de gri uniform și un fundal contrastant și specificăm faptul că granița aceluși obiect este conturul definit de nivelul D_1 , atunci:

$$\int_{D_1}^{\infty} H(D)dD = \text{aria obiectului} \quad (2.21)$$

Dacă imaginea conține mai multe obiecte, ale căror granițe sunt date prin contururi de nivel D_1 , atunci ecuația (2.21)

dă suma ariilor tuturor acestor obiecte.

Normalizarea histogramei prin împărțirea cu aria imaginii produce funcția de probabilitate a densității imaginii. O normalizare similară a funcției arie produce funcția de distribuție cumulativă a imaginii. Aceste funcții sunt folosite în prelucrarea statistică a imaginilor.

Histograma mai are o proprietate folosită, care derivă chiar din definiția sa. Dacă o imagine constă din două regiuni disjuncte, iar histogramele ambelor regiuni sunt cunoscute, atunci histograma întregii imagini este suma celor două histograme regionale. Această proprietate poate fi extinsă la orice număr de regiuni disjuncte.

2.4.4. Moduri de utilizare a histogramelor

2.4.4.1. Parametri de digitizare

Histograma dă o indicație vizuală simplă a faptului dacă o imagine este sau nu bine scalată în intervalul disponibil de nivele de gri. În principal, o imagine digitală ar trebui să folosească toate sau aproape toate nivelele de gri disponibile. Dacă nu se reușește, acest lucru duce la mărirea intervalului efectiv de cuantizare. Odată ce imaginea a fost digitizată folosind mai puțin decât 256 nivele de gri, informația pierdută nu mai poate fi refăcută fără o redigitizare.

La fel, dacă imaginea are un interval mai mare de intensități decât cel pe care îl poate procesa digitizorul, atunci nivelele de gri vor fi tăiate la 0 și/sau 255, producând vârfuri la una sau ambele capete ale histogramei.

Astfel o simplă vizualizare a histogramei, înainte de digitizare poate duce la o parametrizare corectă a operației.

2.4.4.2. Selecția pragului pentru delimitare

Liniile de contur oferă un mod de a stabili limitele unui obiect simplu dintr-o imagine. Presupunând că o imagine conține un obiect întunecat pe un fundal deschis, atunci pixelii întunecați din interiorul obiectului vor produce vârful cel mai din dreapta din histogramă. Vârful din stânga este datorat numărului mare de nivele de gri din fundal. Numărul relativ mic de pixeli având nivel mediu de gri din jurul marginii obiectului produc o vale (un minim) dintre cele două vârfuri. Un prag având nivelul de gri în zona văii va produce niște limite rezonabile pentru obiect (**Fig. 2.4-1**).

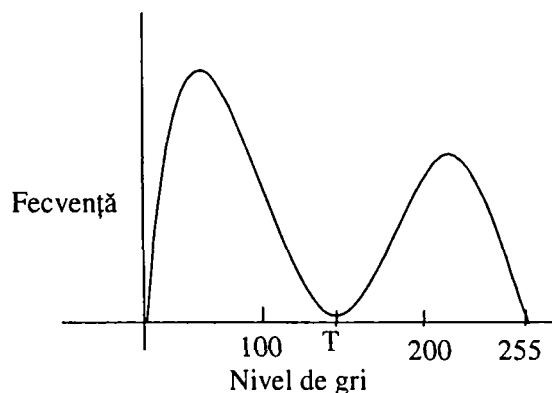


Fig. 2.4-1 Selecția pragului de delimitare.

Într-un anumit sens, nivelul de gri corespunzător minimului dintre două vârfuri este optimal pentru definirea limitelor.

Deoarece histograma este derivata funcției de arie, în vecinătatea minimului histograma va avea valori relativ scăzute, implicând faptul că funcția de arie se schimbă încet cu nivelul de gri. Deci dacă plasăm pragul în vale, atunci minimizăm efectul său asupra limitelor obiectului.

2.4.4.3. Densitatea optică integrată

O altă mărime care poate fi calculată direct din histograma unor imagini simple este densitatea optică integrată (IOD).

$$IOD = \int_a^b \int_a^b D(x, y) dx dy \quad (2.22)$$

unde a și b delimitează regiunea din imagine. Când o imagine constă dintr-un obiect întunecat situat pe un fundal cu un nivel 0, IOD-ul reflectă o combinație dintre aria și densitatea aceluia obiect.

Pentru o imagine digitală:

$$IOD = \sum_{i=1}^{NL} \sum_{j=1}^{NS} D(i, j) \quad (2.23)$$

unde $D(i, j)$ este nivelul de gri al pixelului de pe linia i și coloana j .

Fie N_k numărul de pixeli din imagine cu nivelul de gri egal cu k . Atunci ecuația (2.23) poate fi scrisă ca:

$$IOD = \sum_{k=0}^{255} kN_k = \sum_{k=0}^{255} kH(k) \quad (2.24), (2.25)$$

care este o sumă ponderată cu nivelul de gri al histogramei.

Egalând ecuațiile (2.23) și (2.25) și considerând că incrementul dintre nivelele de gri tinde spre zero, putem obține o expresie similară pentru imaginile continue:

$$IOD = \int DH(D)dD \quad (2.26)$$

$$\int \int D(x, y)dxdy = \int DH(D)dD \quad (2.27)$$

Dacă un obiect din imagine este delimitat printr-un prag cu nivelul T , IOD-ul din interiorul limitelor obiectului este dat de:

$$IOD(T) = \int DH(D)dD \quad (2.28)$$

Nivelul de gri mediu din interior este raportul dintre IOD și arie:

$$MGL = \frac{IOD(T)}{A(T)} = \frac{\int DH(D)dD}{\int H(D)dD} \quad (2.29)$$

2.4.5. Relația dintre histogramă și imagine

Deoarece histograma unei imagini este unică, este posibilă obținerea histogramei unei imagini simple a cărei formă funcțională este cunoscută. Presupunând că avem o imagine dată de o formă funcțională și dorim să calculăm histograma sa, știm că aceasta este negativul derivatei în funcție de nivelul de gri al funcției de arie (2.16). Deci putem obține histograma dacă mai întâi obținem funcția de arie din expresia imaginii. În anumite cazuri aceasta poate fi făcută printr-o simplă observație.

2.4.5.1. Cazul unidimensional

Pentru simplificare, mai întâi considerăm cazul unidimensional. În acest caz, aria este de fapt o lungime, dar demonstrează relația dintre histogramă și imagine.

Considerăm pulsul Gaussian unidimensional dat de (Fig. 2.4-2):

$$D(x) = e^{-x^2} \quad -\infty \leq x \leq \infty \quad (2.30)$$

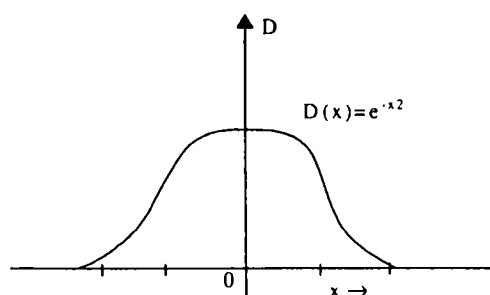


Fig. 2.4-2 Pulsul Gaussian

Pentru x pozitiv se observă că funcția este monotonă. Mai mult, aria este aproape inversa funcției imagine. Astfel, pentru valori nenegative ale lui x se poate rezolva ecuația pentru x ca și funcție de nivel de gri:

$$x(D) = \sqrt{-\ln(D)} \quad x \geq 0 \quad (2.31)$$

care este funcția de arie pentru jumătatea din dreapta a imaginii. Deoarece cele două jumătăți ale imaginii sunt simetrice, funcția globală de arie este dublul aceleia din ecuația (2.31).

Histograma este dată de:

$$H(D) = -\frac{d}{dD} [2\sqrt{-\ln(D)}] = \frac{1}{D\sqrt{-\ln(D)}} \quad (2.32)$$

Histograma prezintă un vârf pentru $D=0$ din cauza unei arii largi alcătuită din nivele de gri joase la valori mari pozitive și negative ale lui x . Vârful mai redus la $D=1$ rezultă din faptul că imaginea are o curbă pentru $x=0$, curba Gaussiană prezentând un platou local în vârful ei (Fig. 2.4-3).

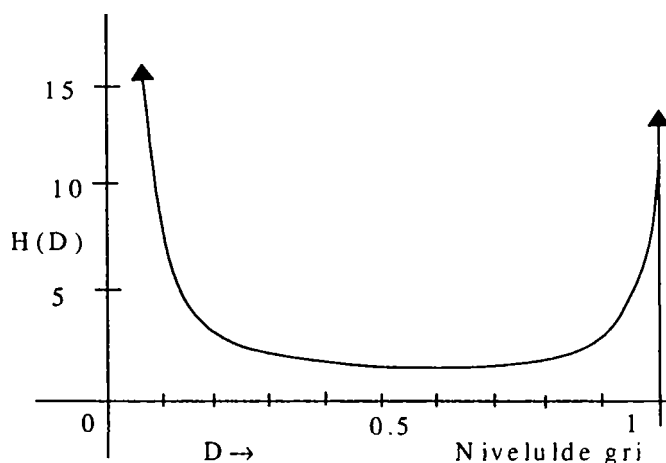


Fig. 2.4-3 Histograma pulsului Gaussian.

2.4.5.2. Cazul bidimensional

Aceeași procedură poate fi extinsă la imagini bidimensionale printr-o folosire judicioasă a simetriei din imagine. De exemplu, dacă pulsul Gaussian unidimensional este de fapt o linie dintr-o imagine bidimensională, atunci dacă toate liniile sunt identice, histograma va avea aceeași formă ca cea pentru cea unidimensională, diferind doar în scara ordinatei.

S-ar putea considera avantajul simetriei circulare în următorul fel: presupunând că imaginea este un puls Gaussian circular simetric centrat în origine (Fig. 2.4-4), funcția imaginii în coordonate polare este dată de:

$$D(r, \theta) = e^{-r^2} \quad 0 \leq \theta \leq 2\pi, \quad 0 \leq r \leq \infty \quad (2.33)$$

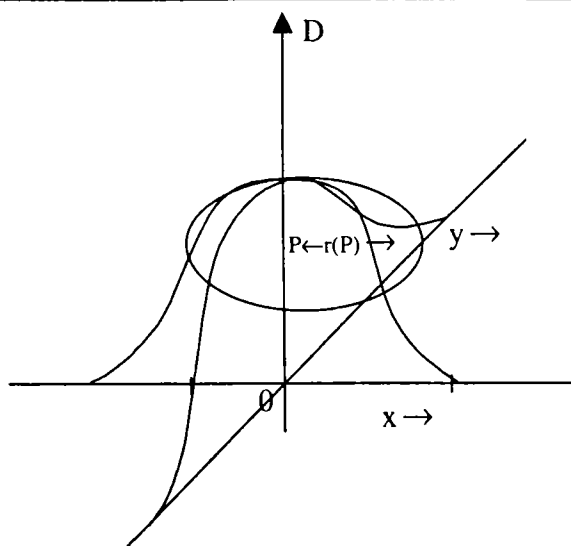


Fig. 2.4-4 Spotul circular Gaussian

Un contur P de nivel de gri constant este un cerc de rază:

$$r(P) = \sqrt{-\ln(P)} \quad (2.34)$$

Un astfel de contur cuprinde o arie:

$$A(P) = \pi[r(P)]^2 = -\pi \ln(P) \quad (2.35)$$

Funcția de arie poate fi acum diferențiată pentru a obține histograma (Fig. 2.4-5):

$$H(P) = \frac{d}{dP} A(P) = \frac{\pi}{P} \quad (2.36)$$

Este de notat faptul că punctul de curbură zero din origine nu este destul de puternic pentru a produce un vârf la $D=1$, așa cum a făcut-o în cazul unidimensional.

Pentru imagini mai complexe, histograma poate fi obținută prin partiționarea în primă fază a imaginii în regiuni disjuncte pentru care funcțiile de arie pot fi determinate. Histograma imaginii complete în acest caz este dată de suma histogramelor tuturor regiunilor disjuncte.

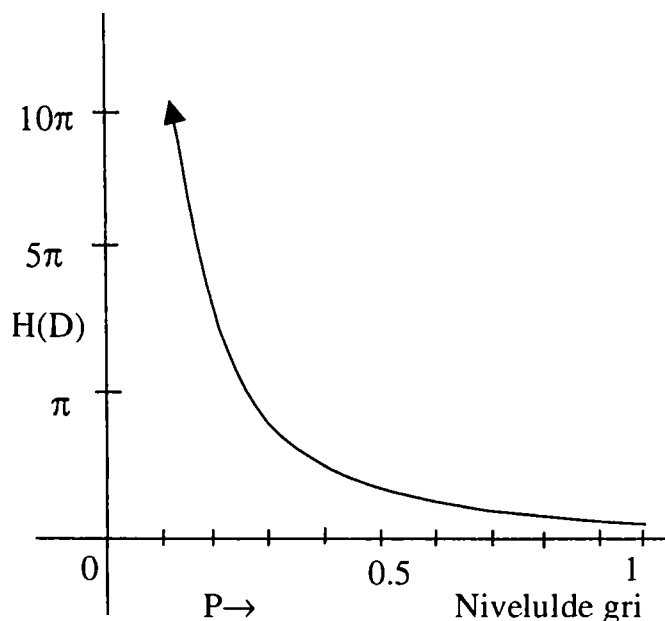


Fig. 2.4-5 Histograma spotului circular Gaussian

633.077
 0000 3015
 Bibliotecă Centrală

2.4.6. Operații punctuale

Operațiile punctuale constituie o clasă simplă dar importantă de tehnici de procesare a imaginii. Acestea permit utilizatorului să modifice felul în care datele imaginii umplu intervalul disponibil de nivele de gri, care în mod particular afectează felul în care imaginea va apărea când va fi afișată.

O operație punctuală trece o singură imagine de intrare într-o singură imagine de ieșire într-un mod în care nivelul fiecărui pixel de ieșire depinde doar de nivelul pixelului de intrare corespunzător. Aceasta este în contrast cu operațiile locale, în care o vecinătate a pixelilor de intrare determină nivelul de gri al fiecărui pixel de ieșire. Mai mult, într-o operație punctuală, fiecare pixel de ieșire corespunde în mod direct pixelului de intrare, având aceleași coordonate. Astfel o operație punctuală nu poate modifica relațiile spațiale din imagine.

Operațiile punctuale mai sunt cunoscute și sub alte denumiri incluzând: mărirea contrastului, extinderea contrastului și transformări ale scării de gri. Acestea sunt de multe ori înglobate în produse software de digitizare sau afișare.

Operațiile punctuale modifică histograma nivelelor de gri într-un mod predictibil. Acestea pot fi văzute ca și operații de copiere pixel cu pixel, cu deosebirea că nivelul de gri este modificat în funcție de, funcția de transformare a scării de gri. O operație punctuală care primește o imagine de intrare $A(x,y)$ și produce o imagine de ieșire $B(x,y)$ poate fi exprimată astfel:

$$B(x, y) = f[A(x, y)] \quad (2.37)$$

Operația punctuală este specificată complet prin funcția de transformare a scării de gri (GST), $f(D)$ specifică maparea nivelului de gri în nivele de ieșire.

2.4.6.1. Aplicații ale operațiilor punctuale

Operațiile punctuale sunt uneori folosite pentru a depăși limitările digitizoarelor, înainte ca procesarea propriu-zisă să aibă loc. La fel de importante sunt și operațiile punctuale folosite pentru îmbunătățirea procesului de afișare a imaginilor.

Calibrarea fotometrică

De multe ori este de dorit ca nivelele de gri ale unei imagini digitizate să reflecte o anumită proprietate fizică, ca de exemplu intensitatea luminii sau densitatea optică. Operațiile punctuale pot realiza aceasta prin eliminarea efectelor neliniarității senzorilor. Ca și un exemplu, presupunând că o imagine a fost digitizată cu un instrument care are un răspuns neliniar la intensitatea luminii. O operație punctuală poate transforma scara nivelurilor de gri astfel încât nivelele de gri să reprezinte incrementele egale în intensitatea luminii. Aceasta este un exemplu de calibrare fotometrică.

O altă utilizare a operațiilor punctuale este de a transforma unitățile scării nivelelor de gri. Presupunând că o imagine luată printr-un microscop a fost digitizată de un instrument care produce nivele de gri care sunt liniare cu transmitanța specimenului, atunci o operație punctuală poate fi folosită pentru a crea o imagine în care nivelele de gri reprezintă pași egali în densitatea optică. Putem considera calibrarea fotometrică ca și partea software a digitizării imaginilor.

Mărirea contrastului

În unele imagini digitale, trăsăturile de interes ocupă doar un interval relativ îngust din scara nivelelor de gri. Pentru a lărgi contrastul acestor trăsături se pot folosi operații punctuale astfel încât acestea să ocupe o porțiune mai largă din intervalul nivelelor de gri afișate.

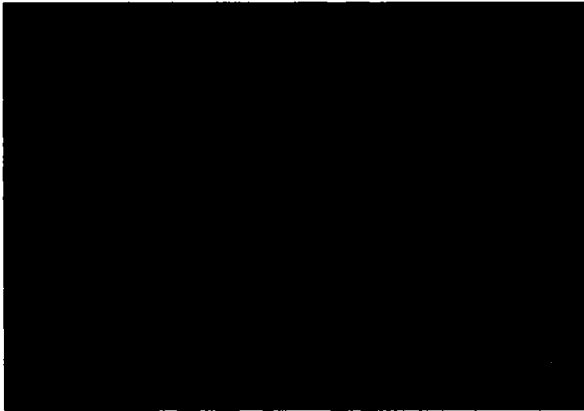


Fig. 2.4-6 Imaginea

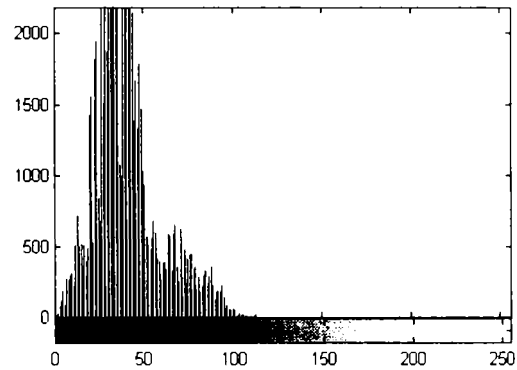


Fig. 2.4-7 Histograma

De exemplu imaginea din figura Fig. 2.4-6 a fost digitizată folosind doar aproximativ 100 de nivele (Fig. 2.4-7) în loc de 256, ceea ce a dus la o imagine neclară. În acest caz calitatea imaginii s-ar putea îmbunătăți printr-o operație de „întindere” a histogrammei, așa cum este arătat în figura rezultată în urma operației (Fig. 2.4-8, Fig. 2.4-9).



Fig. 2.4-8 Imagine corectată

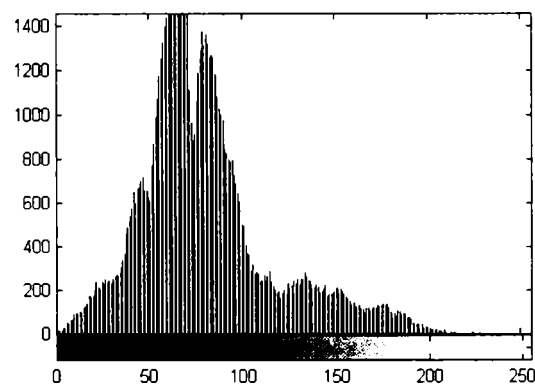


Fig. 2.4-9 Noua histogramă

Calibrarea display-ului

Anumite display-uri au un interval de nivele de gri preferat peste care acestea accentuează trăsăturile imaginilor. În acest fel trăsături mai întunecate și cele mai luminoase, având același contrast în imaginea digitală, dar nu vor apărea la fel pe un astfel de display. În astfel de cazuri, utilizatorul ar putea utiliza operații punctuale pentru a se asigura ca trăsăturile de interes să cadă în intervalul de vizibilitate maximă a display-ului.

Multe display-uri nu mențin o relație liniară dintre nivelul de gri al unui pixel din imaginea digitală și luminozitatea punctului corespunzător de pe display. În mod similar, multe recordere de film nu sunt în stare să transforme într-un mod liniar nivelele de gri în densitate optică. Aceste neajunsuri pot fi eliminate prin unele operații punctuale bine proiectate, aplicate înainte de afișarea imaginii. Luate împreună, operația punctuală și neliniaritatea display-ului se anulează reciproc, iar aceasta păstrează liniaritatea din imaginea afișată. Această procedură este denumită „calibrarea display-ului”.

În anumite cazuri, este dorită o relație neliniară în afișare, pentru o prezentare potrivită a imaginii. Această neliniaritate deliberată este specificată de parametrul γ al monitoarelor de televiziune și CRT. Operațiile punctuale pot corecta sau ajusta γ -ul display-urilor.

Liniile de contur

O operație punctuală poate adăuga linii de contur unei imagini. Deasemenea se poate efectua o divizare a imaginii în imagini disjuncte, prin aplicarea unui prag, pe baza nivelului de gri. Aceasta este folositoare pentru definirea granițelor sau pentru construirea unor măști pentru operații ulterioare.

Clipping-ul

Deoarece imaginile digitale sunt stocate în principal utilizând numere întregi, intervalul de nivele de gri este limitat. De exemplu pentru imagini pe 8 biți, nivelul de gri trebuie redus la intervalul 0-255 înainte ca orice valoare de pixel să fie memorată.

2.5. Teoria sistemelor liniare

2.5.1. Generalități

Teoria sistemelor liniare este un domeniu bine dezvoltat, folosit mai mult pentru a descrie comportarea circuitelor electrice și a sistemelor optice. Aceasta oferă o bază matematică solidă pe care se pot examina efectele cuantizării, filtrării și a rezoluției spațiale. Teoria sistemelor liniare este de asemenea de ajutor în multe alte aplicații.

Considerăm un sistem ca fiind orice care acceptă o intrare și produce în răspuns o ieșire. Deoarece suntem interesați doar de relația dintre intrare și ieșire, avem un interes minor în ceea ce este în interiorul sistemului. Intrarea și ieșirea poate fi unidimensională, bidimensională sau multidimensională.

Inițial exemplele se vor restrânge doar la două cazuri: funcții unidimensionale de timp și funcții bidimensionale de două variabile spațiale. Acestea păstrează modalitățile simple, fac analiza mai puțin abstractă și evită generalizarea excesivă. Analiza poate fi ușor generalizată la mai multe dimensiuni dacă este cazul.

Figurile Fig. 2.5-1 și Fig. 2.5-2 arată notarea convențională pentru sistemele liniare uni și bidimensionale. În fiecare caz, intrarea în sistem este o funcție de una sau două variabile și produce un răspuns al sistemului care este o altă funcție de aceeași variabilă.

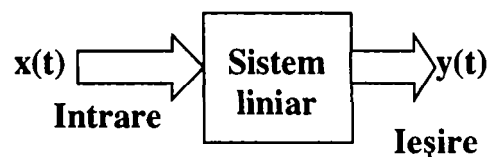


Fig. 2.5-1 Notăția pentru un sistem liniar

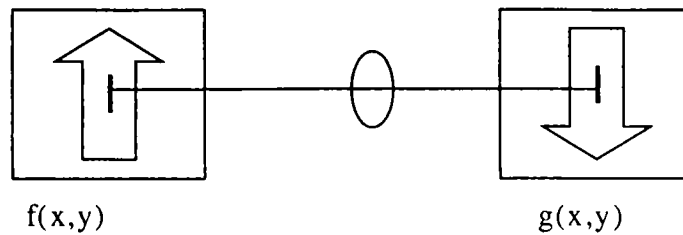


Fig. 2.5-2 Un sistem liniar bi-dimensional

Liniaritatea. Sistemele liniare au o proprietate de la care le derivă și denumirea. Presupunând că pentru un anumit sistem particular, o intrare $x_1(t)$ dă naștere la o ieșire $y_1(t)$:

$$x_1(t) \rightarrow y_1(t) \quad (2.38)$$

Presupunând de asemenea că o a doua intrare $x_2(t)$ dă naștere la o ieșire $y_2(t)$:

$$x_2(t) \rightarrow y_2(t) \quad (2.39)$$

Sistemul este liniar dacă și numai dacă are proprietatea:

$$x_1(t) + x_2(t) \rightarrow y_1(t) + y_2(t) \quad (2.40)$$

Adică, un al treilea semnal de intrare, care este suma celor două anterioare, produce un semnal de ieșire care este suma celor două semnale de ieșire originale. Orice sistem care nu respectă această constrângere nu este liniar.

Definiția unui sistem liniar afirmă că o intrare care este suma a două semnale produce o ieșire care este suma ieșirilor produse de fiecare semnal de intrare dacă ar acționa singur. Din aceasta rezultă că dacă un semnal de intrare este multiplicat cu un număr rațional, ieșirea este crescută sau scăzută cu același factor, adică:

$$ax_1(t) = ay_1(t) \quad (2.41)$$

Ca și o axiomă, considerăm că ecuația (2.41) este valabilă și pentru numere iraționale.

Proprietatea definită de ecuațiile (2.38), (2.39) și (2.40) și corolarul său din (2.41) servește pentru definirea unui sistem liniar. Când se folosește teoria sistemelor liniare pentru a analiza un proces, este imperativ ca procesul modelat să fie, măcar aproximativ, liniar. Dacă sistemul aflat în studiu nu satisface criteriul pentru liniaritate, atunci este neliniar și teoria sistemelor liniare va produce rezultate incorecte și chiar unele care pot induce în eroare. Dacă sistemul este doar puțin neliniar, se poate considera liniar pentru analiză, dar rezultatele analizei vor fi pe atât de bune pe cât au fost presupunerile.

Invarianța la deplasare. O proprietate folositoare pe care anumite sisteme o posedă este denumită invarianță la deplasare. Presupunând că pentru un anumit sistem avem:

$$x(t) \rightarrow y(t) \quad (2.42)$$

și presupunând că deplasăm semnalul de intrare în timp cu o cantitate T , semnalul este invariant la deplasare dacă

$$x(t - T) \rightarrow y(t - T) \quad (2.43)$$

adică, ieșirea este deplasată cu aceeași cantitate ca și intrarea, dar altfel rămâne neschimbată. Deci, pentru un sistem invariant la deplasări, deplasarea intrării doar deplasează ieșirea cu aceeași cantitate. Important este faptul că natura ieșirii nu este schimbată de o deplasare a

semnalului de intrare. Invarianța la deplasarea spațială este analogul bidimensional al invarianței la deplasarea în timp. Dacă imaginea de intrare este deplasată relativ la originea sa, imaginea de ieșire rămâne la fel ca înainte, doar că este deplasată identic.

2.5.2. Analiza semnalelor armonice și a semnalelor complexe

În mod obișnuit, semnalele și imaginile pot fi reprezentate prin funcții reale de una sau respectiv două variabile. Valoarea funcției reprezintă magnitudinea unui anumit parametru fizic, ca de exemplu tensiunea, ca o funcție de timp sau intensitatea luminoasă ca și o funcție de două coordonate spațiale. Deoarece funcțiile cu valori reale pot fi considerate cazuri speciale ale funcțiilor cu valori complexe, putem generaliza prezentarea pentru funcții cu valori complexe.

2.5.2.1. Semnale armonice

Considerăm un semnal complex de forma:

$$x(t) = e^{j\omega t} = \cos(\omega t) + j \sin(\omega t) \quad (2.44)$$

unde $j^2 = -1$. Aceasta este denumită semnal armonic. Este o funcție complexă de timp care poate fi văzută ca și vectorul unitate care se rotește în planul complex cu o amplitudine unghiulară ω figura **Fig. 2.5-3**. Frecvența unghiulară ω , în radiani pe secundă, este legată de f , frecvența în rotații sau cicluri pe secundă (Hertz) prin $\omega = 2\pi f$.

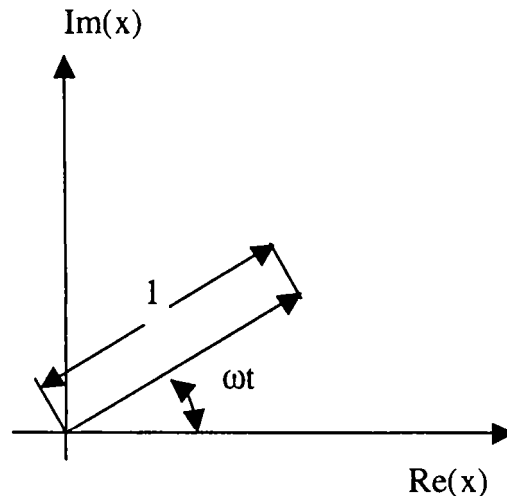


Fig. 2.5-3 Semnal armonic

2.5.2.2. Răspunsul la o intrare armonică

Presupunând că unui sistem liniar invariant la deplasări îi este prezentat un semnal de intrare armonic

$$x_1(t) = e^{j\omega t} \quad (2.45)$$

Putem exprima răspunsul sistemului ca:

$$y_1(t) = K(\omega, t)e^{j\omega t} \quad (2.46)$$

unde:

$$K(\omega, t) = \frac{y_1(t)}{e^{j\omega t}} \quad (2.47)$$

este o funcție complexă de ω și t selectați astfel încât, când este înmulțit cu $e^{j\omega t}$ să producă $y_1(t)$. Astfel va exista totdeauna un $K(\omega, t)$ care va corespunde.

Presupunem că generăm o a doua intrare prin deplasarea în timp a lui $x_1(t)$. Astfel avem:

$$x_2(t) = e^{-j\omega(t-T)} = e^{-j\omega T} e^{j\omega t} = e^{-j\omega T} x_1(t) \quad (2.48)$$

De notat că $x_2(t)$ este aproape $x_1(t)$, înmulțită cu o constantă complexă. Aceasta deoarece $x_1(t)$ este un semnal armonic. Răspunsul sistemului liniar la $x_2(t)$ este:

$$y_2(t) = K(\omega, t-T) e^{j\omega(t-T)} \quad (2.49)$$

adică:

$$y_2(t) = K(\omega, t-T) e^{-j\omega T} e^{j\omega t} \quad (2.50)$$

sau

$$y_2(t) = K(\omega, t-T) e^{-j\omega T} x_1(t)$$

Aplicând ecuația (2.41), putem scrie:

$$x_2(t) = e^{-j\omega T} x_1(t) \rightarrow e^{-j\omega T} y_1(t) = e^{-j\omega T} K(\omega, t) e^{j\omega t} \quad (2.51)$$

Din ecuația (2.45), recunoaștem factorul cu exponentul din dreapta ca fiind $x_1(t)$. Deasemenea, știm că partea cu răspunsul al ecuației (2.51) trebuie să fie $y_2(t)$, deoarece este răspunsul sistemului la $x_2(t)$. Așadar putem scrie:

$$y_2(t) = e^{-j\omega T} K(\omega, t) x_1(t) \quad (2.52)$$

care este o a doua expresie pentru răspunsul sistemului la intrarea armonică deplasată.

Ecuația (2.50) a fost obținută prin inserarea unui deplasament de timp în ecuația (2.46). Ecuația (2.52) a rezultat din proprietatea liniară (ecuația (2.41)). Ambele ecuații sunt expresia răspunsului sistemului liniar la intrarea armonică deplasată în timp, astfel ele trebuie să fie egale. Combinând ecuațiile (2.50) și (2.52) rezultă:

$$K(\omega, t-T) e^{-j\omega T} x_1(t) = K(\omega, t) e^{-j\omega T} x_1(t) \quad (2.53)$$

de unde rezultă:

$$K(\omega, t-T) = K(\omega, t) \quad (2.54)$$

trebuie să fie adevărată pentru orice valoare a deplasamentului T . Ecuația (2.54) poate să fie adevărată doar dacă $K(\omega, t)$ este independentă de t . Astfel, ecuația (2.46) poate fi rescrisă în forma generală:

$$y(t) = K(\omega) x(t) \quad (2.55)$$

Funcția generală a cărei formă a fost presupusă în ecuația (2.47) a reieșit a fi o funcție care depinde doar de variabila frecvență ω . Ecuația (2.55) afirmă o proprietate importantă și anume că răspunsul unui sistem liniar invariant la deplasări la o intrare armonică este acea intrare înmulțită cu un număr complex dependent de frecvență. De observat că o intrare armonică întotdeauna produce o ieșire armonică la aceeași frecvență.

2.5.2.3. Semnale armonice și sinusoidale

Când folosim un sistem liniar pentru modelarea comportamentului unui sistem fizic, intrările și ieșirile sunt în mod convenabil reprezentate ca și funcții reale. Astfel putem adăuga o altă restricție asupra sistemelor liniare invariante la deplasări și anume aceea că ele păstrează caracterul real. Prin definiție, aceasta înseamnă că o intrare reală poate produce doar o ieșire reală. Din aceasta, poate fi arătat că un astfel de sistem păstrează de asemenea și caracterul imaginar și că înlăturând partea imaginară a unei intrări complexe se înlătură partea imaginară a ieșirii corespondente, adică:

$$x(t) \rightarrow y(t) \Rightarrow R_e\{x(t)\} \rightarrow R_e\{y(t)\} \quad (2.56)$$

Cu alte cuvinte, părțile reale și cele imaginare ale unei intrări armonice trec prin sistem independent unele de altele.

Restricția de păstrare a caracterului real pe sistemele liniare ne permite simplificarea analizei. De exemplu, dacă intrarea este un cosinus, putem adăuga o componentă imaginară sinusoidală pentru a forma un semnal armonic, pentru a determina răspunsul sistemului la acest semnal armonic, după care putem înlătura partea imaginară din ieșirea complexă. Această abordare indirectă este justificată printr-o semnificativă simplificare a analizei.

Orice semnal sinusoidal poate fi privit ca și partea reală a unui semnal armonic. Această abordare ne permite să derivăm răspunsul unui sistem liniar prin:

- (1) reprezentarea sinusoidei de intrare printr-un semnal armonic;
- (2) derivarea răspunsului sistemului liniar la intrarea armonică;
- (3) separarea părții reale a ieșirii armonice pentru a obține ieșirea corectă.

Pentru a efectua această operație, folosim o metodă de transformare a soluției, adică transformăm semnalul sinusoidal în armonic, rezolvăm problema în termeni armonici și transformăm ieșirea armonică înapoi într-o sinusoidă.

Transformarea în semnale armonice simplifică analiza sistemelor liniare într-un mod considerabil.

2.5.2.4. Funcția de transfer

Funcția $K(\omega)$ este denumită funcție de transfer a sistemului liniar și este suficientă pentru a specifica complet sistemul. Pentru un sistem liniar invariant la deplasări, funcția de transfer conține toate informațiile existente despre sistem.

Putem converti $K(\omega)$ la forma polară pentru a obține:

$$K(\omega) = A(\omega)e^{j\phi(\omega)} \quad (2.57)$$

unde $A(\omega)$ este o funcție reală de frecvență, iar exponentul complex este un vector unitate în planul complex, adică un număr complex având mărimea unitate. Presupunând că intrarea este un cosinus, luată ca partea reală a unui semnal armonic:

$$x(t) = \cos(\omega t) = R_e\{e^{j\omega t}\} \quad (2.58)$$

Răspunsul sistemului la intrarea armonică este:

$$K(\omega)e^{j\omega t} = A(\omega)e^{j\phi}e^{j\omega t} = A(\omega)e^{j(\omega t + \phi)} \quad (2.59)$$

În final, semnalul de ieșire este:

$$y(t) = R_e \{ A(\omega) e^{j(\omega t + \phi)} \} = R_e \{ A(\omega) [\cos(\omega t + \phi) + j \sin(\omega t + \phi)] \} = A(\omega) \cos(\omega t + \phi) \quad (2.60)$$

$A(\omega)$ este factorul de câștig multiplicativ și reprezintă gradul până la care sistemul amplifică sau atenuează semnalul de intrare. $\phi(\omega)$ este unghiul deplasării fazei. Singurul său efect este de a deplasa originea de timp a funcției armonice de intrare.

În concluzie, s-au arătat trei proprietăți importante ale sistemelor liniare invariante la deplasări:

- (1) O intrare armonică produce totdeauna o ieșire armonică de aceeași frecvență;
- (2) Sistemul este complet specificat prin funcția sa de transfer, care este o funcție cu valoare complexă, dependentă doar de frecvență;
- (3) Funcția de transfer produce doar două efecte asupra intrării armonice: o schimbare în amplitudine și o deplasare a fazei (o deplasare a originii de timp).

2.5.3. Operația de convoluție

Considerăm sistemul liniar din figura Fig. 2.5-1 . Ar fi util să avem o expresie generală care să facă legătura între semnalul de ieșire $y(t)$ și semnalul de intrare $x(t)$. Putem obține o astfel de relație în următorul mod. Expresia funcțională liniară:

$$y(t) = \int_{-\infty}^{\infty} f(t, \tau) x(\tau) d\tau \quad (2.61)$$

este suficient de generală pentru a exprima relația dintre $x(t)$ și $y(t)$ pentru orice sistem liniar. O funcție $f(t, \tau)$ de două variabile poate fi aleasă, pentru ca ecuația (2.61) să fie îndeplinită, pentru orice sistem liniar.

Impunem constrângere de invarianță la deplasare într-o încercare de a simplifica ecuația (2.61). Substituind ecuația (2.43) în (2.61) rezultă:

$$y(t - T) = \int_{-\infty}^{\infty} f(t, \tau) x(\tau - T) d\tau \quad (2.62)$$

Facem o schimbare de variabilă prin adunarea lui T atât la t cât și la τ . Aceasta produce:

$$y(t) = \int_{-\infty}^{\infty} f(t + T, \tau + T) x(\tau) d\tau \quad (2.63)$$

Comparând ecuațiile (2.63) și (2.61) se observă că:

$$f(t, \tau) = f(t + T, \tau + T) \quad (2.64)$$

trebuie să fie adevărat pentru toate valorile lui T . Aceasta înseamnă că $f(t, \tau)$ nu se schimbă dacă adăugăm aceeași constantă ambelor argumente. Cu alte cuvinte, $f(t, \tau)$ este constantă atâta timp cât diferența dintre t și τ este constantă. Astfel putem defini o nouă funcție, și anume:

$$g(t - \tau) = f(t, \tau) \quad (2.65)$$

iar ecuația (2.61) devine:

$$y(t) = \int_{-\infty}^{\infty} g(t - \tau)x(\tau)d\tau \quad (2.66)$$

Aceasta este familiara integrală de convoluție. Aceasta afirmă că ieșirea unui sistem liniar invariant la deplasări este dată de convoluția semnalului de intrare cu o funcție $g(t)$ care este caracteristică acestui sistem (Fig. 2.5-4). Această funcție caracteristică este denumită răspuns la impuls a sistemului. De notat că sistemul păstrează caracterul real dacă și numai dacă $g(t)$ este reală.

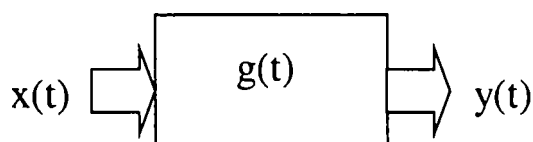


Fig. 2.5-4. Un sistem liniar

Avem acum două moduri de specificare a relației dintre intrarea și ieșirea unui sistem liniar invariant la deplasări:

- (1) Orice sistem liniar invariant la deplasări are o funcție de transfer complexă, care atunci când este multiplicat cu o intrare armonică, produce ieșire armonică;
- (2) Astfel de sistem are un răspuns la impuls real, care, când este convolvat cu semnalul de intrare, produce semnalul de ieșire.

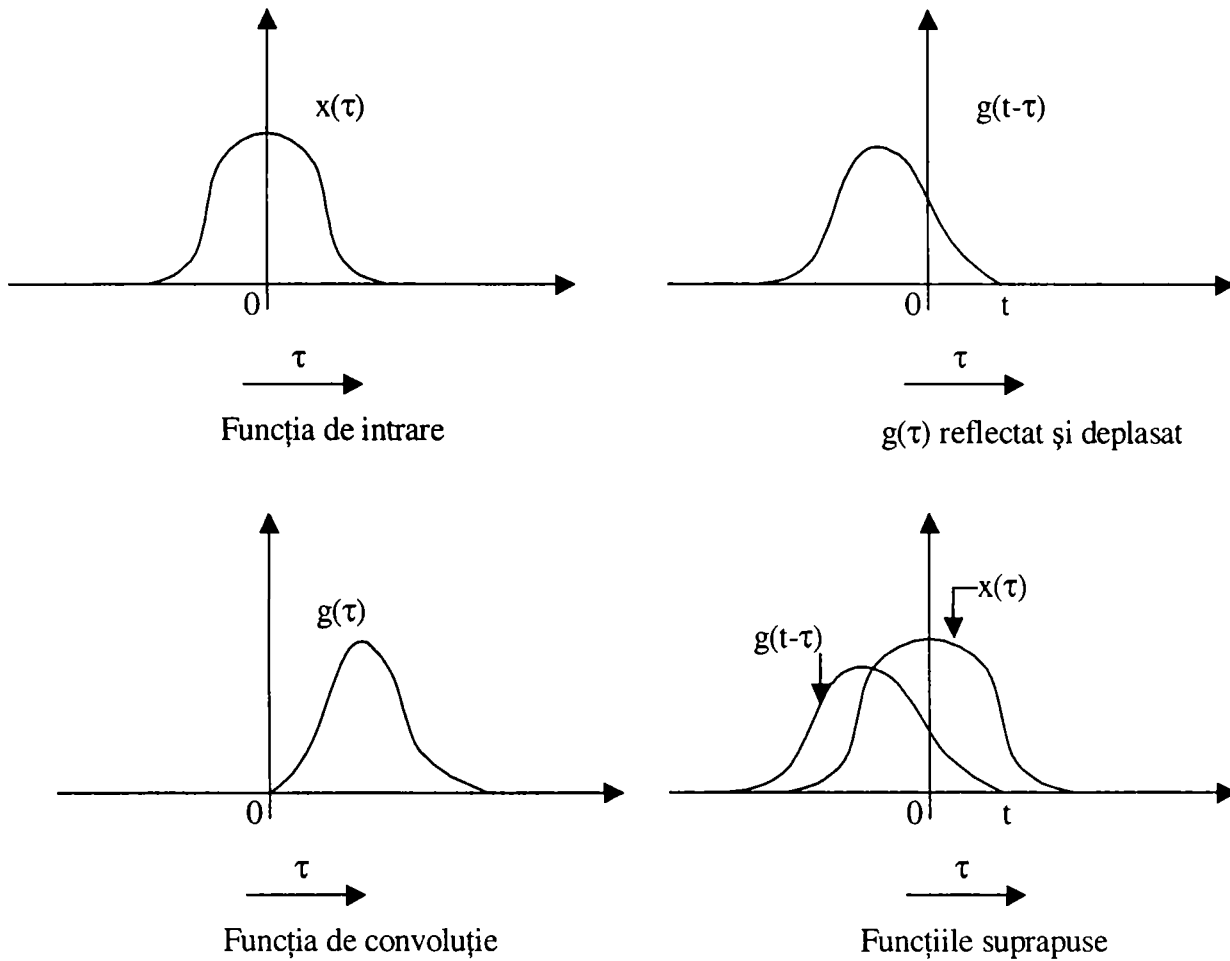
Deoarece funcția de transfer și răspunsul la impuls al unui sistem liniar invariant la deplasări sunt fiecare unice și adecvate pentru specificarea completă a sistemului, presupunem că cele două funcții sunt legate una de alta.

2.5.3.1. Convoluția unidimensională

Integrala de convoluție din ecuația (2.66) poate fi prescurtată prin notația:

$$y = g * x \quad (2.67)$$

unde $*$ este folosit pentru indicarea convoluției a două funcții. Figura Fig. 2.5-5 prezintă o ilustrare grafică a operației de convoluție. Un punct de pe curba $y(t)$ este obținut în următorul mod: o funcție g este reflectată față de originea sa și deplasată cu o cantitate t la dreapta. Produsul punct cu punct al lui x și al lui g reflectat și deplasat este format și integrat pentru a produce valoarea de ieșire la t . Acest proces este repetat pentru toate valorile lui t pentru a produce alte puncte pe curba de ieșire. Pe măsură ce t este modificat, funcția reflectată este deplasată după funcția de staționare și valoarea lui $y(t)$ depinde de cantitatea suprapusă a celor două funcții.


Fig. 2.5-5 Convoluția

$$y(t) = \int_{-\infty}^{\infty} x(\tau)g(t - \tau)d\tau = A$$

Convoluția are câteva proprietăți folositoare. În primul rând este comutativă, adică:

$$f * g = g * f \quad (2.68)$$

și putem reflecta orice funcție și vom obține același rezultat. Aceasta poate fi arătată prin scrierea:

$$f * g = \int_{-\infty}^{\infty} f(\tau)g(t - \tau)d\tau \quad (2.69)$$

făcând schimbarea de variabilă:

$$x = t - \tau \quad \tau = t - x \quad dx = -d\tau \quad (2.70)$$

și rearanjând pentru a produce:

$$f * g = \int_{-\infty}^{\infty} f(t - x)g(x)dx = g * f \quad (2.71)$$

În ecuația (2.71) limitele au trebuit interschimbate și aceasta compensează minusul de la $d\tau$.

Convoluția este de asemenea distributivă față de adunare, adică:

$$f * (g + h) = f * g + f * h \quad (2.72)$$

Aceasta poate fi arătată prin:

$$f * (g + h) = \int_{-\infty}^{\infty} f(t - \tau)[g(\tau) + h(\tau)]d\tau \quad (2.73)$$

rearanjând produce:

$$f * (g + h) = \int_{-\infty}^{\infty} f(t - \tau)g(\tau)d\tau + \int_{-\infty}^{\infty} f(t - \tau)h(\tau)d\tau = f * g + f * h \quad (2.74)$$

Convoluția este de asemenea și asociativă, adică:

$$f * (g * h) = (f * g) * h \quad (2.75)$$

La derivare:

$$\frac{d}{dt}[f * g] = f' * g = f * g' \quad (2.76)$$

2.5.4. Convoluția unidimensională discretă

Secvențele discrete pot fi convolute în aceeași manieră ca și funcțiile continue. Variabila independentă devine un index, iar integrala este înlocuită cu o sumă. Astfel, pentru două secvențe $f(i)$ și $g(i)$ de lungimi m și respectiv n :

$$h(i) = f(i) * g(i) = \sum_i f(j)g(i - j) \quad (2.77)$$

care produce o secvență de ieșire de lungime $N=m+n-1$.

Deși convoluția discretă și cea continuă sunt operații destul de diferite, ele au în comun multe proprietăți. Convoluția discretă pe care o putem implementa pentru imaginile digitale urmărește îndeaproape convoluția continuă care descrie multe dintre lucrurile care se petrec cu imaginile înainte și după ce ele sunt aduse în formă digitală. Aceasta este exploatată în restaurarea imaginilor, ceea ce încearcă să refacă efectele de degradare care au acționat deja pe imagini.

Este convenabil de reprezentat secvențele discrete ca și vectori și de a beneficia de notația compactă și de proprietățile bine definite de algebra liniară. Deși ecuația (2.77) este o sumă de produse, convoluția a două secvențe nu poate fi efectuată de o simplă înmulțire vectorială. Poate fi descrisă de o înmulțire de matrici dacă situația este bine pusă la punct.

În primul rând presupunem că $f(i)$ este o porțiune a unei secvențe de lungime infinită, care este periodică cu o perioadă de cel puțin N , lungimea unei secvențe de ieșire produse de convoluția conform ecuației (2.77) . Deoarece $f(i)$ este mai scurt decât N , este necesar să completăm elementele adiționale cu zero.

O perioadă din secvența infinită care rezultă după această procedură este dată de:

$$f_p(i) = \begin{cases} f(i) & 1 \leq i \leq m \\ 0 & m < i \leq N \end{cases} \quad (2.78)$$

Repetăm aceeași procedură cu $g(i)$ și $h(i)$. Acum toate cele trei secvențe au aceeași lungime.

După aceasta, fie f un vector coloană $N \times 1$, a cărui elemente sunt $f_p(i)$, o perioadă din secvența infinită formată din $f(i)$. Totodată, fie G o matrice a cărei primă linie este secvența completată cu zerouri $g_p(i)$ în ordine inversă. Liniile următoare ale lui G sunt formate printr-o deplasare circulară la dreapta a elementelor rândului precedent. Acum putem scrie:

$$h = G \cdot f = \begin{bmatrix} g_p(1) & g_p(N) & \dots & g_p(2) \\ g_p(2) & g_p(1) & \dots & g_p(3) \\ \dots & \dots & \dots & \dots \\ g_p(N) & g_p(N-1) & \dots & g_p(1) \end{bmatrix} \cdot \begin{bmatrix} f_p(1) \\ f_p(2) \\ \dots \\ f_p(N) \end{bmatrix} \quad (2.79)$$

unde h este un vector $N \times 1$ conținând secvența de ieșire. Convoluția discretă este astfel exprimată ca și un produs dintre o matrice $N \times N$ și un vector $N \times 1$. De reamintit că N este lungimea secvenței de ieșire rezultată din ecuația (2.77) .

Matricea G din ecuația (2.79) este denumită o matrice circulantă, deoarece fiecare rând este o versiune deplasată la dreapta în mod circular a rândului precedent. Aceasta este structura care ne permite scrierea convoluției ca și un produs de matrici. Fiecare rând din G servește pentru a putea genera un element din secvența de ieșire.

2.5.5. Convoluția bidimensională

Convoluția funcțiilor continue de două variabile este similară convoluției unidimensionale. De notat că pentru situația bidimensională se vor utiliza variabilele independente x și y . Ecuația pentru convoluția bidimensională este:

$$h(x, y) = f * g = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(u, v) g(x-u, y-v) du dv \quad (2.80)$$

De notat că $g(0-u, 0-v)$ este chiar $g(u, v)$ rotit cu 180° față de origine și că $g(x-u, y-v)$ este traslatat astfel că originea g -ului rotit se mută în x, y . Funcțiile sunt înmulțite punct cu punct, iar produsul funcțiilor este integrat peste două dimensiuni. Ca și exemplu presupunem că:

$$f(x, y) = A e^{-\frac{(x^2+y^2)}{2\sigma^2}} \quad (2.81)$$

și

$$g(x, y) = \begin{cases} 1, & -1 \leq x \leq 1, -1 \leq y \leq 1 \\ 0, & \text{altfel} \end{cases} \quad (2.82)$$

În acest caz, un puls rectangular bidimensional este convolvat cu un Gaussian bidimensional mai mare. Deoarece $g(x, y)$ este simetric față de origine, rotația de 180° nu are nici un efect. Valoarea lui $h(x, y)$ este chiar volumul funcției produs, când pulsul rectangular este deplasat în x, y .

2.5.6. Aplicații ale convoluției

Filtrarea liniară, implementată digital este folositoare pentru trei clase majore de aplicații de procesare a imaginilor.

1. Deconvoluția, adică eliminarea efectelor produse nedorit dar aplicate anterior ale unor sisteme liniare care au operat pe imagine. Un exemplu este folosirea convoluției pentru a restaura detaliile pierdute printr-un sistem de lentile sau de bruiatul datorat de mișcare, ambele putând fi asimilate cu operațiile liniare.
2. Eliminarea zgomotului, adică reducerea efectelor unor semnale nedorite contaminatoare, care au fost adăugate liniar imaginii (Fig. 2.5-6, Fig. 2.5-7, Fig. 2.5-8).



Fig. 2.5-6 Imaginea originală



Fig. 2.5-7 Imagine cu zgomot



Fig. 2.5-8 Imagine convolvată cu un nucleu 5x5

3. Întărirea caracteristicilor, adică creșterea contrastului unor caracteristici specifice (muchii, spoturi, etc) pe cheltuiala altor obiecte din scenă (Fig. 2.5-9).



Fig. 2.5-9 Accentuarea conturilor

2.6. Concluzii

În acest capitol s-a făcut o prezentare a unui sistem de prelucrare a imaginilor, evidențiind etapele fundamentale în prelucrarea digitală de imagini și prezentând câteva noțiuni matematice de bază care vor fi utilizate în această lucrare. Totodată în acest capitol au fost descrise operațiile de bază folosite în prelucrarea imaginilor, cum ar fi histogramele, operațiile punctuale și teoria sistemelor liniare, prezentându-se efectul acestora asupra unor imagini de test.

Operațiile punctuale mai sunt cunoscute și sub alte denumiri incluzând: mărirea contrastului, extinderea contrastului și transformări ale scării de gri. Acestea sunt de multe ori înglobate în produse software de digitizare sau afișare.

Operațiile punctuale modifică histograma nivelelor de gri într-un mod predictibil. Acestea pot fi văzute ca și operații de copiere pixel cu pixel, cu deosebirea că nivelul de gri este modificat, în funcție de funcția de transformare a scării de gri.

Convoluția este una dintre cele mai folosite operații în prelucrarea de imagini, având cea mai mare aplicabilitate în construcția de filtre liniare. Clasele majore de aplicații ale acestora sunt: deconvoluția, eliminarea zgomotului și creșterea contrastului unor caracteristici specifice.

3. Metode de selectare a caracteristicilor în procesul de recunoaștere și interpretare a imaginilor

3.1. Introducere

Cel mai important aspect al recunoașterii formelor este cel al selectării (extragerii) caracteristicilor. Dacă selectorul de caracteristici este adecvat și eficient atunci orice algoritm de clasificare, fie simplu, fie extrem de sofisticat poate fi implementat deosebit de ușor, în principal datorită reducerii dimensionalității și extragerii din mulțimea caracteristicilor a acelor care sunt esențiale procesului de clasificare. În felul acesta formele se vor grupa mai compact într-un spațiu al formelor redus ca dimensionalitate, formând domenii care permit adoptarea unor suprafețe de decizie mai simple (liniare în loc de funcții de discriminant de ordin mai mare). Procedeele prin care se realizează această funcție în cadrul unui sistem de recunoașterea formelor poartă denumirea de selector de caracteristici sau preprocesor.

Obiectele sau fenomenele aparținând unei baze de date sunt caracterizate printr-o mulțime de măsurători (caracteristici). Din punctul de vedere al recunoașterii formelor este important ca numărul de caracteristici să fie cât mai mic și aceasta cel puțin din următoarele două motive.

- (1) Caracteristicile neesențiale problemei de clasificare trebuie eliminate deoarece, chiar dacă nu determină clasificări incorecte, măresc nejustificat durata procesului de clasificare din cauza calculelor suplimentare care trebuie efectuate.
- (2) Numărul de caracteristici n trebuie să fie cât mai mic decât numărul de forme m pentru a fi siguri că clasificatorul nu separă în mod aleator clasele din care fac parte formele din setul de formare. În acest caz un raport $m/n > 3$ este acceptabil deși este de dorit ca acest raport să fie mult mai mare.

În esență, problema selectării caracteristicilor se reduce la găsirea unei combinații optime a caracteristicilor. Multe din metodele care se aplică încearcă să demonstreze care este cea mai bună caracteristică individuală, capabilă să discrimineze formele care aparțin unor clase diferite. Testarea exhaustivă a tuturor caracteristicilor cu scopul de a găsi pe cea cu puterea discriminatorie cea mai mare este aproape întotdeauna imposibil de realizat practic, acest mod de abordare devenind eficient doar în cazul unor seturi mici de date [17].

O altă strategie, care este foarte utilizată constă în generarea unui grup de caracteristici noi, rezultat al transformării caracteristicilor inițiale. Acele caracteristici care sunt esențiale pentru procesul de clasificare sunt uneori denumite *caracteristici intrinseci* [16],[17]. Noile caracteristici intrinseci pot fi generate prin combinarea liniară sau neliniară a unora sau a tuturor caracteristicilor inițiale. Selecția celor mai importanți termeni se face aplicând metodele uzuale de selectare a caracteristicilor.

Generarea unor noi caracteristici ortogonale se poate face cu ajutorul metodei de analiză a componentelor principale aplicând transformarea Karhunen-Loève cu ajutorul căreia se obțin noi componente ale vectorilor de formă, necorelate însă, sau folosind diferite alte transformări, ca de exemplu Fourier, Hadamard, Haar, DCT, DST, etc .

Metodele folosite pentru selectarea caracteristicilor modifică structura aranjamentului punctelor în spațiul formelor. Dacă caracteristicile au diferite mărimi ele trebuie aduse pe aceeași scară deoarece cele cu valori absolute mai mari vor avea un efect mai pronunțat asupra rezultatelor clasificării. În cadrul acestui proces de raportare echivalentă a caracteristicilor măsurătorile sunt ajustate de asemenea manieră încât să releveze o proprietate comună oarecare. Acest procedeu devine foarte important atunci când caracteristicile sunt extrase din mai multe surse diferite, ele fiind combinate într-un vector unic. În acest sens, cel mai simplu procedeu este cel de *ajustare a domeniului*, în care noua caracteristică ajustată $x_{i,nou}$ este dată de următoarea relație:

$$x_{i,nou} = (x_{i,vechi} - x_{i,min}) / (x_{i,max} - x_{i,min}) \quad (3.1)$$

unde $x_{i,vechi}$ reprezintă componenta inițială a vectorului de formă, iar $x_{i,min}$, $x_{i,max}$ reprezintă cea mai mică valoare, respectiv cea mai mare valoare, a caracteristicii $x_{i,vechi}$ din întregul set de date. Metoda ajustării prezintă dezavantajul că dacă există o valoare foarte diferită de celelalte, extremă, atunci toate celelalte valori rezultate $x_{i,nou}$ vor avea valori foarte apropiate.

Un alt procedeu similar, foarte folosit, este cel cunoscut sub numele de *autoajustare*. În acest caz caracteristicile sunt transformate astfel încât media lor să fie zero iar varianța unitară, astfel:

$$x_{i,nou} = (x_{i,vechi} - m_{i,vechi}) / v_{i,vechi} \quad (3.2)$$

unde $m_{i,vechi}$ reprezintă media componentelor inițiale $x_{i,vechi}$, iar $v_{i,vechi}$ varianța lor. După autoajustare toate componentele vor avea aceeași pondere și în consecință, efectul lor asupra clasificării va fi egal.

O altă categorie de metode utilizate în cadrul preprocesorului sunt *metodele de ponderare* cu ajutorul cărora se evaluează importanța caracteristicilor ajustate pentru o anumită problemă concretă de clasificare. Aceste tehnici pot fi utilizate numai în cazul învățării controlate, atunci când formarea clasificatorului se face pe un set de formare compus din vectorii prototip a căror clasificare este complet cunoscută. Factorii de ponderare sunt calculați sau estimați pe baza analizei statistice a datelor, în urma evaluării clasificatorului format, sau cu ajutorul informațiilor apriorice legate de problema concretă de clasificare. Metodele de ponderare selectează caracteristicile esențiale procesului de clasificare îmbunătățind astfel separarea dintre clase.

Un alt tip de metode folosite la selectarea caracteristicilor constă în transformarea acestora de o anumită manieră, sau în combinarea lor. Astfel, în locul caracteristicilor se poate folosi o funcție matematică, de exemplu \sqrt{x} , $\log x$, $\log(x + const)$ etc. O simplificare severă a datelor se realizează dacă se aplică următoarea codificare binară:

$$\begin{aligned} x_{nou} &= 1, \text{ dacă } x_{vechi} \geq \text{valoarea de prag} \\ x_{nou} &= 0, \text{ dacă } x_{vechi} < \text{valoarea de prag} \end{aligned} \quad (3.3)$$

Se pot genera noi caracteristici prin combinarea liniară sau neliniară a unora sau a tuturor caracteristicilor originale. În acest caz pot fi luate în considerare corelările dintre caracteristici prin calcularea termenilor produsului vectorial care sunt fie adăugați vectorului de formă, fie substituie cele mai importante caracteristici. Pentru o formă cu n componente vor exista $n(n+1)/2$ astfel de termeni. Selecția celor mai importanți termeni poate fi efectuată cu ajutorul aceluiași metode folosite pentru selecția caracteristicilor.

Combinarea euristică a caracteristicilor originale a fost de asemenea folosită cu scopul de a genera noi caracteristici. Totuși o asemenea manieră de abordare este dependentă exclusiv de problema concretă de clasificare.

Metodele matematice dezvoltate în scopul selecției caracteristicilor se referă la un spațiu vectorial R -dimensional, unde R este definit de numărul maxim de caracteristici dintr-un vector de formă dat. Se presupune că cele R dimensiuni ale spațiului formelor au fost astfel alese încât să permită comparații semnificative din punctul de vedere al procesului de clasificare. Obiectivul selectorului de caracteristici constă în reținerea celor mai remarcabile caracteristici necesare procesului de recunoaștere și reducerea dimensionalității astfel încât clasificatorul să poată fi implementat într-un spațiu al formelor cu o dimensionalitate cât mai redusă. În felul acesta dimensionalitatea spațiului formelor va fi n , unde n este mult mai mic decât R .

Esența oricărei metode de selectare a caracteristicilor constă în reținerea informației cu capacitate de discriminare între clase și reducerea informației care prezintă aspecte comune pentru clasele în discuție. În felul acesta, punerea în evidență a acelor proprietăți ale formelor dintr-o clasă care se deosebesc de formele care fac parte dintr-o altă clasă permite menținerea

capacității de clasificare într-un spațiu al formelor de o dimensionalitate redusă. În consecință, selecția caracteristicilor poate fi privită ca un proces care constă în reținerea optimă a unui număr minim de dimensiuni (variabile) astfel încât să se mențină și/sau să se maximizeze probabilitatea de clasificare corectă. Din păcate, un asemenea criteriu de fidelitate este dificil de evaluat și de aceea adeseori se apelează la alte criterii care permit o selecție a caracteristicilor optimă dar ușor de mânuit analitic.

În continuare se va prezenta o serie de metode de selectare a caracteristicilor folosite curent în cadrul sistemelor de recunoaștere.

3.2. Metode de selectare a caracteristicilor

3.2.1. Transformări diagonale

Una din tehnicile folosite pentru accentuarea caracteristicilor comune și îmbunătățirea grupării formelor care fac parte din aceeași clasă, constă în minimizarea unei metrici definite pentru punctele care constituie clasa respectivă. O astfel de măsură intra-clasă se bazează pe o metrică euclidiană pătratică [17]. Această metodă poate fi privită ca o tehnică de grupare deoarece cu ajutorul ei vectorii-prototip care definesc o clasă oarecare sunt grupați în spațiul formelor, delimitând astfel mult mai net clasa din care fac parte.

Fie o clasă oarecare ω_k pentru care avem un număr total de M_k vectori-prototip (forme care fac parte din clasă) $X_m^{(k)}$, indexați astfel $m = 1, 2, \dots, M_k$. Fiecare vector-prototip (vector coloană) constituie un punct în spațiu R-dimensional și fie o metrică în acest spațiu definită astfel:

$$d^2(X, X_m^{(k)}) = \sum_{i=1}^R (x_i - x_{im}^{(k)})^2 \quad (3.4)$$

sau

$$d^2(X, X_m^{(k)}) = (X - X_m^{(k)})^t (X - X_m^{(k)}) \quad (3.5)$$

În acest caz, pătratul mediei distanțelor intraclasă pentru ω_k , presupunând o generare egal probabilă a fiecărui vector prototip, va fi dat de relația:

$$D_k^2 = \frac{1}{M_k(M_k - 1)} \sum_{j=1}^{M_k} \sum_{m=1}^{M_k} d^2(X_j^{(k)}, X_m^{(k)}) \quad (3.6)$$

În relația (3.6) sunt sumate $M_k(M_k - 1)$ perechi diferite de zero pentru toate punctele care definesc clasa ω_k . Pentru a minimiza pe D_k^2 se introduce un factor de ponderare variabil, w_i , pentru fiecare dimensiune din spațiul formelor. În acest fel, pătratul mediei distanțelor intraclasă, care urmează a fi minimizat, este de forma:

$$D_k^2 = \frac{1}{M_k(M_k - 1)} \sum_{j=1}^{M_k} \sum_{m=1}^{M_k} \sum_{i=1}^R w_i^2 (x_{ij}^{(k)} - x_{im}^{(k)})^2 \quad (3.7)$$

Se observă că procedeul descris mai sus este echivalent cu înmulțirea spațiului formelor pentru clasa ω_k , cu o matrice diagonală $[W_k]$. De fapt, minimizarea lui D_k^2 din relația (3.7) implică determinarea ponderilor, w_i care trebuie să îndeplinească următoarele condiții standard:

$$w_i > 0, \quad \forall i = 1, \dots, R \quad (3.8)$$

$$\sum_{i=1}^R w_i = 1 \quad (3.9)$$

sau

$$\prod_{i=1}^R w_i = 1 \quad (3.9')$$

Condiția (3.8) este necesară pentru ca D_k^2 din relația (3.7) să respecte condițiile impuse unei metrici. Condiția exprimată prin relația (3.9) implică ponderarea dimensiunii i în proporție aritmetică față de celelalte caracteristici (dimensiuni), conform unei sume constante, iar cealaltă condiție (3.9'), se referă la o ponderare de volum constant, fiind legată de norma spațiului formelor.

Utilizând metoda *multiplicatorilor lui Lagrange* în cazul minimizării lui D_k^2 , pentru suma ponderilor constantă (relația (3.9)), rezultă:

$$w_i = \frac{\left(\sum_{n=1}^R \frac{1}{\sigma_n^2} \right)^{-1}}{\sigma_i^2} \quad (3.10)$$

unde σ_i^2 reprezintă varianța corespunzătoare dimensiunii i .

Din relația (3.10) rezultă că ponderea corespunzătoare dimensiunii i este direct proporțională cu o constantă $\left(\sum_{n=1}^R \frac{1}{\sigma_n^2} \right)^{-1}$ și invers proporțională cu variația dimensiunii i .

Acest rezultat poate fi interpretat relativ ușor, în sensul că acele coordonate a căror varianță este mare vor fi slab ponderate, cu factori de pondere mici, deoarece ele nu sunt semnificative pentru clasa ω_k , existând puține elemente comune în formele care aparțin acestei clase. Trebuie remarcat faptul că acele dimensiuni din spațiul formelor care prezintă valori relativ constante în toate formele care fac parte din clasa ω_k vor avea varianțe mici și în consecință, factorii de pondere corespunzători vor fi mari iar caracteristicile corespunzătoare vor fi extrem de importante.

Pentru cazul în care pentru factorii de pondere ai coordonatelor, w_i , se impune condiția de volum constant (relația (3.9')), se obține un rezultat relativ similar, de forma:

$$w_i = \frac{\left(\prod_{n=1}^R \sigma_n \right)^{\frac{1}{R}}}{\sigma_i} \quad (3.11)$$

Deoarece și în acest caz factorul de pondere corespunzător coordonatei i este direct proporțional cu o constantă, de astă dată $\left(\prod_{n=1}^R \sigma_n \right)^{\frac{1}{R}}$, și invers proporțional cu deviația standard a datelor corespunzătoare dimensiunii respective i , se pot trage aceleași concluzii intuitive ca înainte, cu observația că de astă dată deviația standard este parametrul de proporționalitate.

Un dezavantaj evident pe care îl prezintă soluțiile de tipul (3.10) și (3.11) este legat de marea sensibilitate pe care o manifestă acestea față de acele dimensiuni care în mod sigur au varianța extrem de mică sau chiar neglijabilă.

După minimizarea pătratului mediei distanțelor intraclasă, D_k^2 și determinarea factorilor de pondere se pune problema selecției caracteristicilor. Aceasta se realizează prin ordonarea ponderilor w_i , în ordine monoton descrescătoare și observând că matricea diagonală $[W_k]$ va fi definită de ponderile astfel ordonate, selecția caracteristicilor pentru clasa ω_k se va

realiza printr-o operație de permutare matriceală ordonată pe spațiul formelor R -dimensional urmată de o transformare matriceală diagonală. În felul acesta va rezulta un nou spațiu al formelor, cu o dimensionalitate redusă, care va fi definit prin cele mai mari n ponderi, cu $n < R$. Deci, în felul acesta pot fi reținute cele mai importante caracteristici care definesc cel mai bine clasa ω_k . Factorul de reducere a dimensionalității este exprimat prin raportul R/n . De asemenea, se poate exprima gradul de încredere γ în determinarea lui n pentru condițiile (3.9), respectiv (3.9') prin relațiile:

$$\gamma = \sum_{i=1}^n w_i \quad (3.12)$$

$$\gamma = \frac{\sum_{i=1}^n w_i}{\sum_{i=1}^R w_i} \quad (3.12')$$

Un alt dezavantaj al acestei metode de selectare a caracteristicilor se datorează faptului că ea trebuie să fie repetată pentru fiecare din cele K clase, rezultând astfel un număr de K matrice diagonale $[W_k]$, $k = 1, \dots, K$. Astfel, dacă numărul de clase K este mare, procesul de calcul devine prea lung și deci neeconomic. Pe de altă parte, faptul că alegerea celor mai semnificative caracteristici se face exclusiv pe baza unei analize intraclasă, neglijându-se orice interacțiune interclase, poate constitui un impediment serios pentru rezolvarea unor probleme curente de clasificare.

Din punct de vedere statistic rezultatele obținute cu ajutorul tehnicii prezentate anterior au la bază numai proprietăți de ordinul doi, adică varianțe și derivații standard. În acest sens, ar fi de dorit dezvoltarea unor metode care să utilizeze statistici de ordin mai înalt. Acest fapt implică cunoașterea funcțiilor de distribuție de probabilitate, situație care din punct de vedere practic nu este întotdeauna posibilă.

3.2.2. Transformări rotaționale

Pentru a extinde aria tehnicilor de selectare a caracteristicilor prin cuprinderea unor metode care să nu se limiteze doar la schimbări de scală, un mod de abordare care se impune constă în aplicarea unei transformări liniare descrise printr-o matrice ortogonală [17]. O astfel de transformare este echivalentă cu o rotație a spațiului original, rezultând astfel un nou set de coordonate ortogonale care pot să definească un nou domeniu (spațiu) în care identificarea caracteristicilor esențiale și reducerea dimensionalității se fac mult mai ușor și în felul acesta întregul proces de clasificare este optimizat. Dintre metodele de selectare a caracteristicilor care utilizează transformări ortogonale, majoritatea se bazează pe utilizarea matricei de covarianță.

În cazul vectorilor prototip, care deci aparțin aceleiași clase, matricea de covarianță se poate defini astfel:

$$[\Phi_k] = A_p \left\{ \mathbf{Y}_m^{(k)} (\mathbf{Y}_m^{(k)})^T \right\} \quad (3.13)$$

unde

$$\mathbf{Y}_m^{(k)} = \mathbf{X}_m^{(k)} - A_p \left\{ \mathbf{X}_m^{(k)} \right\} \quad (3.14)$$

reprezintă vectorul de medie zero, iar operatorul A_p se aplică tuturor formelor din clasa k . Dacă se cunoaște o densitate de probabilitate p care descrie setul de vectori prototip, atunci operatorul A_p se definește în raport cu aceea densitate de probabilitate. Astfel, matricea de covarianță devine:

$$[\Phi_k] = \left(\mathbf{X}_m^{(k)} - \overline{\mathbf{X}_m^{(k)}} \right) \cdot \left(\mathbf{X}_m^{(k)} - \overline{\mathbf{X}_m^{(k)}} \right)^T \quad (3.15)$$

unde linia continuă de deasupra membrului drept al relației de mai sus indică o valoare medie așteptată, iar elementele de pe diagonală sunt egale cu σ_i^2 , care reprezintă varianța dimensiunii i .

3.2.3. Transformarea Karhunen - Loève

Dacă vectorii prototip din clasa ω_k sunt reali, atunci $[\Phi_k]$ este definită reală și pozitivă și va avea toate valorile proprii mai mari decât zero.

În acest caz vor fi selectate acele caracteristici ("cele mai bune") care, formând un spațiu cu o dimensionalitate mai redusă, vor aproxima formele originale cel mai bine astfel încât eroarea medie pătratică să fie minimă. Pentru aceasta vom considera caracteristicile dezvoltate sub forma unui set de vectori $\{\mathbf{e}_p : p = 1, 2, \dots, n < R\}$ care sunt ortonormați și constituie elementele unei familii de baze ortonormate complete în spațiul formelor. Fie $\mathbf{Y}_m^{(k)}$ vectorul prototip de medie zero, definit conform ecuației (3.14), pe care îl vom dezvolta în funcție de cei n vectori de bază \mathbf{e}_p , astfel:

$$\mathbf{Y}_m^{(k)} = a_{m1}\mathbf{e}_1 + a_{m2}\mathbf{e}_2 + \dots + a_{mn}\mathbf{e}_n \quad (3.16)$$

Prin minimizarea erorii datorate acestei dezvoltări vom obține caracteristicile (coordonatele) "cele mai bune", \mathbf{e}_n . Această eroare poate fi exprimată în felul următor:

$$E_n^{(k)} = A_p \left\{ \left| \mathbf{Y}_m^{(k)} - \sum_{p=1}^n a_{mp}\mathbf{e}_p \right|^2 \right\} \quad (3.17)$$

unde operatorul A_p se aplică tuturor vectorilor prototip conform distribuției lor p , dacă aceasta este cunoscută. După cum se știe $\mathbf{Y}_m^{(k)}$ este de forma:

$$\mathbf{Y}_m^{(k)} = \sum_{r=1}^R a_{mr}\mathbf{e}_r \quad (3.18)$$

pentru orice set complet de vectori de bază ortonormați. Înmulțind ambii membri ai ecuației (3.18) cu \mathbf{e}_i' vom obține:

$$\mathbf{e}_i' \cdot \mathbf{Y}_m^{(k)} = \sum_{r=1}^R a_{mr} \cdot \mathbf{e}_i' \cdot \mathbf{e}_r \quad (3.19)$$

Prin aplicarea condițiilor de ortonormare rezultă:

$$\mathbf{e}_i' \cdot \mathbf{Y}_m^{(k)} = a_{mi} \quad (3.20)$$

sau

$$a_{mi} = (\mathbf{Y}_m^{(k)})' \cdot \mathbf{e}_i \quad (3.20')$$

În felul acesta eroarea se va exprima astfel:

$$E_n^{(k)} = A_p \left\{ \left| \sum_{r=n+1}^R a_{mr}\mathbf{e}_r \right|^2 \right\} \quad (3.21)$$

sau utilizând ecuația (3.20'),

$$E_n^{(k)} = A_p \left\{ \left| \sum_{r=n+1}^R ((\mathbf{Y}_m^{(k)})' \cdot \mathbf{e}_r) \mathbf{e}_r \right|^2 \right\} \quad (3.21')$$

Dezvoltând pătratul din (3.21') avem:

$$E_n^{(k)} = A_p \left\{ \sum_{r=n+1}^R \mathbf{e}_r' \cdot \mathbf{Y}_m^{(k)} \cdot (\mathbf{Y}_m^{(k)})' \cdot \mathbf{e}_r \right\} \quad (3.22)$$

și ținând cont de ecuația (3.13) vom obține, în final, relația următoare:

$$E_n^{(k)} = \sum_{r=n+1}^R \mathbf{e}_r' \cdot [\Phi_k] \cdot \mathbf{e}_r \quad (3.23)$$

Minimizarea erorii $E_n^{(k)}$ se poate face minimizând fiecare termen al rezultatului, $\mathbf{e}_r' \cdot [\Phi_k] \cdot \mathbf{e}_r$. Astfel, prin aplicarea metodei multiplicatorilor lui Lagrange și ținând cont de condiția de ortonormare rezultă:

$$E_n^{(k)} = \sum_{r=n+1}^R \mathbf{e}_r' \cdot [\Phi_k] \cdot \mathbf{e}_r - \lambda_r \cdot \mathbf{e}_r' \cdot \mathbf{e}_r + \lambda_r = \text{minim} \quad (3.24)$$

ceea ce revine la

$$\frac{\partial E_n^{(k)}}{\partial \mathbf{e}_r'} = [\Phi_k] \cdot \mathbf{e}_r - \lambda_r \cdot \mathbf{e}_r = 0 \quad (3.25)$$

adică

$$[\Phi_k] \cdot \mathbf{e}_r = \lambda_r \cdot \mathbf{e}_r \quad (3.26)$$

Din ecuația (3.26) rezultă că \mathbf{e}_r este un vector propriu al matricei de covarianță a datelor $[\Phi_k]$, iar λ_r reprezintă valoarea proprie corespunzătoare. În felul acesta eroarea minimă estimată devine:

$$E_n^{(k)} = \sum_{r=n+1}^R \mathbf{e}_r' \cdot \lambda_r \cdot \mathbf{e}_r \quad (3.27)$$

adică

$$E_n^{(k)} = \sum_{r=n+1}^R \lambda_r \quad (3.27')$$

unde λ_r , $r = n + 1, n + 2, \dots, R$, reprezintă valorile proprii corespunzătoare acelor vectori proprii care nu au fost incluși în dezvoltarea ecuației (3.18). $E_n^{(k)}$ va fi minim dacă primii n vectori proprii ai caracteristicilor sunt astfel aleși încât să corespundă celor mai mari valori proprii. În felul acesta problema minimizării lui $E_n^{(k)}$ reprezintă o problemă de analiză a factorilor sau a componentelor principale.

Transformarea optimă pentru o reducere a dimensionalității dată R/n se va obține prin selectarea acelor n caracteristici reprezentate prin vectorii proprii corespunzători celor mai mari valori proprii ale matricei de covarianță definită de formele din clasa ω_k , astfel încât să se asigure reproductibilitatea vectorilor originali. Prin aceasta se definește $[\mathbf{T}_k]$. Deși această selecție a "celor mai bune caracteristici" s-a făcut pe baza erorii medii pătratice, trebuie menționat faptul că nu întotdeauna aceasta constituie cel mai bun criteriu. Se observă că transformarea de mai sus constituie, în fond, o rotație a spațiului formelor R -dimensionale în urma căreia "liniile de forță" datorate vectorilor de formă se grupează în spațiu pe mai puține dimensiuni ale acestuia, astfel încât $(R-n)$ coordonate pot fi neglijate.

Transformarea rotațională de mai sus, bazată pe minimizarea erorilor de estimare, nu ia în considerare nici un fel de relații posibile interclase [17] Din această cauză setul de K transformări rotaționale, câte una pentru fiecare clasă, poate să fie ineficient pentru procesul de clasificare tocmai din cauza inexistenței vreunei relații între cele K transformări. În consecință, este de preferat o unică rotație interclase care poate să țină cont atât de relații intraclasă (similaritățile vectorilor prototip din cadrul unei clase), cât și de relații interclase (disimilaritățile dintre clase). Una din tehnicile folosite în acest scop și care se bazează pe informații interclase este transformarea Karhunen-Loève [16] care a fost generalizată pentru probleme de clasificare multicategorială. În acest caz se definește o matrice de covarianță globală $[\Phi]$, pentru cazuri discrete, matrice care nu se limitează la o anumită clasă de forme ci ea sistematizează de o anumită manieră toate informațiile cu privire la problema de

recunoaștere a formelor. Acest mod de abordare are la bază ideea existenței a K procese stohastice staționare, câte unul pentru fiecare clasă ω_k , $k = 1, \dots, K$, fiecare proces fiind cunoscut cel puțin la nivelul statisticii de ordinul doi $[\Phi_k]$. Distribuția celor K procese este dată de probabilitățile apriori ale claselor, $P(\omega_1), P(\omega_2), \dots, P(\omega_k), \dots, P(\omega_K)$. În acest fel matricea de covarianță totală $[\Phi]$ reprezintă o medie a matricelor de covarianță ale claselor, $[\Phi_k]$, adică:

$$[\Phi] = \sum_{k=1}^K P(\omega_k) \cdot [\Phi_k] \quad (3.28)$$

Cei n vectori proprii $\{e_p : p = 1, \dots, n\}$ formează un set de vectori ortonormat. În cazul în care ei au fost astfel selectați încât corespund celor mai mari n valori proprii atunci transformarea rotațională particulară realizată constituie cea mai bună aproximație, în sensul minimizării erorii medii pătratice a datelor (forme) din spațiul inițial al formelor.

Ca măsură a fidelității coordonatelor rămase în urma transformării rotaționale (fidelitate în raport cu coordonatele inițiale) se utilizează o măsură γ , exprimată cu ajutorul relației următoare:

$$\gamma = \frac{\sum_{i=1}^n \lambda_i}{\sum_{i=1}^R \lambda_i} \quad (3.29)$$

De remarcat că $0 \leq \gamma \leq 1$ măsoară raportul dintre "energia" reținută de noul spațiu față de "energia" totală.

Există mai multe moduri de aplicare a acestor tehnici de transformare rotațională prezentate mai sus, deoarece prin rotația coordonatelor nu se separă întotdeauna suficient de bine clasele. Matricea de transformare $[T]$, de dimensiune $n \times R$, constă din vectorii proprii ordonați e_1, \dots, e_n . Ea poate fi definită astfel:

$$[T] = \begin{bmatrix} e'_1 \\ e'_2 \\ \vdots \\ M \\ e'_n \end{bmatrix} \quad (3.30)$$

Transformarea completă $R \times R$ poate fi definită prin atașarea vectorilor proprii eliminați în cursul transformării adică:

$$[T] = \begin{bmatrix} e'_1 \\ e'_2 \\ \vdots \\ M \\ e'_R \end{bmatrix} \quad (3.31)$$

Aceasta definește o transformare ortogonală sau o rotație a spațiului formelor iar selecția caracteristicilor se realizează prin eliminarea unor coordonate (în număr de $(R-n)$) din spațiul transformat.

Capacitatea acestei metode de a reduce dimensiunile vectorilor transformați o face extrem de utilă în compresia de imagini.

Din cauza dificultăților care apar la calcularea matricei de covarianță și la implementarea rotației spațiului formelor aplicarea acestei metode necesită un volum de calcul foarte mare. Transformarea K-L depinde și de statistica imaginii și în general, se poate spune că vectorii de bază nu au o reprezentare analitică, ca în cazul altor transformări. S-a

demonstrat, totuși, că anumite modele statistice de forme permit implementarea unui algoritm rapid pentru transformarea K – L. Acesta se bazează pe o descompunere stohastică a formei de intrare (în general a imaginilor) într-o sumă de două secvențe aleatoare. Prima se alege în așa fel încât transformarea K – L să fie o transformare rapidă iar a doua, numită *răspunsul conturului*, depinde numai de informația conținută în conturile conținute în formă.

Elementele caracteristice ale principalelor transformări rotaționale multidimensionale utilizate frecvent în recunoașterea formelor cu scopul selectării caracteristicilor, transformări care se pretează la o implementare mai avantajoasă din punct de vedere al calculului care urmează a fi efectuate, sunt: transformarea Fourier rapidă (Fast Fourier) și transformarea Hadamard sau Walsh. Trebuie menționat că în plus față de acestea a fost dezvoltată o nouă clasă de transformări rapide care realizează rotații ortogonale ale spațiului formelor cu scopul selectării celor mai eficiente caracteristici. Astfel, transformările Fourier, Hadamard (sau Walsh), Walsh generalizată, Kronecker, Haar și Haar generalizată fac parte din clasa transformărilor rotaționale rapide, avantajoase din punct de vedere al implementării și de aceea utilizate mai mult pentru selecția caracteristicilor.

O transformare liniară se realizează cu ajutorul unui operator liniar $[A]$ care va transforma spațiul formelor inițial într-un spațiu rotit, vectorii de bază ai acestui nou spațiu fiind chiar coloanele ortogonale ale matricei $[A]$. În noul spațiu caracteristicile reprezintă combinații liniare ale axelor inițiale, efectuate conform structurii lui $[A]$. În felul acesta, matricea de covarianță în spațiul rotit este:

$$[\Phi_A] = [A]^t \cdot [\Phi_X] \cdot [A] \quad (3.32)$$

unde $[\Phi_X]$ reprezintă matricea de covarianță generalizată inițială dată de:

$$[\Phi_X] = \sum_{k=1}^K P(\omega_k) [\Phi_X^{(k)}] \quad (3.33)$$

iar $P(\omega_k)$ reprezintă distribuțiile a priori ale claselor, $[\Phi_X^{(k)}]$ fiind matricele de covarianță estimate sau modelate pentru fiecare clasă în parte.

3.2.4. Transformata Fourier

Există trei tipuri diferite de tehnici de transformare Fourier: transformata Fourier continuă, transformata prin dezvoltare în serie Fourier și transformata Fourier discretă.

Transformata Fourier continuă asociază două funcții continue (semnalul și spectrul său). Transformata directă și inversă pentru cazul unidimensional sunt date prin relațiile:

$$F(s) = \int_{-\infty}^{\infty} f(x) * e^{-j2\pi(xs)} dx \quad \text{și} \quad f(x) = \int_{-\infty}^{\infty} F(s) * e^{j2\pi(xs)} ds \quad (3.34)$$

Dezvoltarea în serie Fourier reprezintă o funcție periodică (sau, eventual, o funcție tranzitorie considerată ca fiind o perioadă a unei funcții periodice) printr-o secvență (finită sau infinită) de coeficienți Fourier.

Transformata și inversa sa se obțin prin înlocuirea lui s cu $n\Delta s$, care este o variabilă discretă:

$$F_n = F(n\Delta s) = \int_0^L f(x) * e^{-j2\pi(xn\Delta s)} dx \quad \text{și} \quad f(x) = \Delta s \sum_{n=0}^{\infty} F_n e^{j2\pi(xn\Delta s)} \quad (3.35)$$

unde L este perioada și $\Delta s = 1/L$.

Transformata Fourier discretă reprezintă o funcție eșantionată, având și spectrul eșantionat, numărul de eșantioane independente fiind același în ambele domenii. Ea se obține înlocuind x cu $i\Delta x$, (x fiind, de asemenea, o variabilă discretă). Dacă se consideră $g(x)$ o funcție de bandă limitată, eșantionată conform cerințelor teoremei eșantionării, atunci $g_i = g(i\Delta x)$ și:

$$G_k = \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} g_i * e^{-j2\pi k \frac{i}{N}} \quad \text{și} \quad g_i = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} G_k * e^{j2\pi \frac{k}{N} i} \quad (3.36)$$

Un exemplu este prezentat în figurile: Fig. 3.2-1, Fig. 3.2-2, Fig. 3.2-3

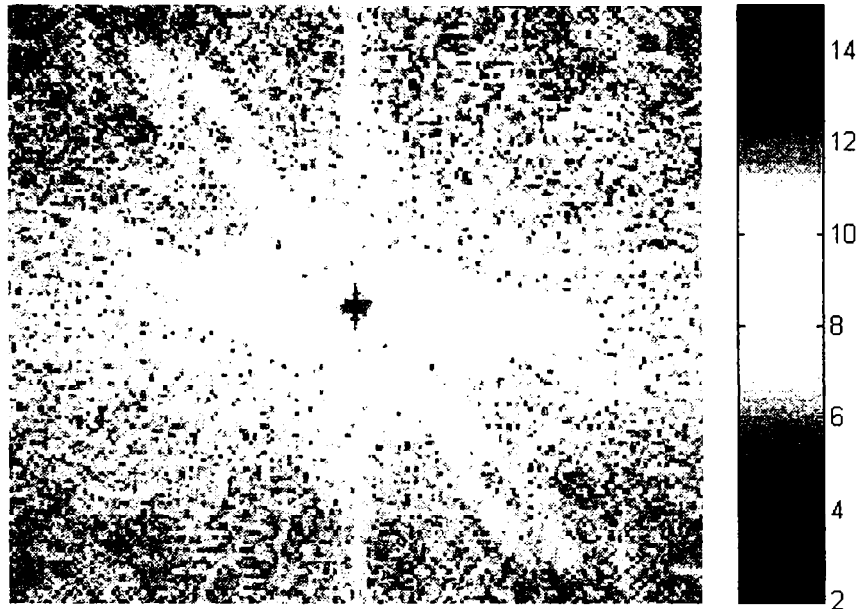


Fig. 3.2-1 Transformata Fourier pentru Fig. 3.2-2



Fig. 3.2-2 Imagine originală



Fig. 3.2-3 Imagine reconstruită (16x16 coef FFT)

În anul 1946, Gabor prezintă transformarea Fourier de scurtă durată - STFT („*Short Time Fourier Transform*”), care deplasează o fereastră de durată fixă peste funcția variabilă în timp și extrage conținutul de frecvență din interval. Acest tip de transformare poate fi util, spre exemplu, pentru analiza semnalului vocal, semnal care este local staționar, dar global nestaționar. Energia semnalului pe intervale trebuie să fie finită:

$$\int_{-\infty}^{\infty} |f(t)|^2 dt < \infty \quad (3.37)$$

STFT poziționează o fereastră $g(t)$ în anumite puncte τ pe axa timpului și calculează transformata Fourier în interiorul acelei ferestre:

$$F(\Omega, \tau) = \int_{-\infty}^{\infty} f(t)g(t - \tau)e^{-j\Omega t} dt \quad (3.38)$$

Dacă fereastra $g(t)$ este de tip Gauss, atunci STFT se numește **transformare Gabor**. Funcțiile de bază ale transformării sunt generate prin *modularea și translația* funcției de fereastră $g(t)$, unde Ω și τ sunt parametri de modulare și respectiv translație.

3.2.5. Transformarea Cosinus discretă

Transformarea cosinus discretă se definește, în spațiul bidimensional ca :

$$v(k, l) = \alpha(k) \cdot \alpha(l) \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} u(m, n) \cdot \cos\left[\frac{(2m+1)k\pi}{2N}\right] \cos\left[\frac{(2n+1)l\pi}{2N}\right] \quad (3.39)$$

unde $k, l = 0, 1, \dots, N-1$, iar inversa :

$$u(m, n) = \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} \alpha(k) \cdot \alpha(l) \cdot v(k, l) \cdot \cos\left[\frac{(2m+1)k\pi}{2N}\right] \cos\left[\frac{(2n+1)l\pi}{2N}\right] \quad (3.40)$$

unde $m, n = 0, \dots, N-1$, iar coeficienții sunt :

$$\alpha(0) = \sqrt{\frac{1}{N}} \quad \text{și} \quad \alpha(k) = \sqrt{\frac{2}{N}} \quad \text{pentru } 1 \leq k \leq N \quad (3.41)$$

La fel ca și transformarea Fourier discretă, transformarea cosinus discretă poate fi exprimată ca o operație cu matrici unitare :

$$\mathbf{V} = \mathbf{C} \mathbf{u} \mathbf{C}^T \quad (3.42)$$

unde coeficienții transformării \mathbf{C} sunt :

$$c_{k,m} = \alpha(k) \cdot \cos\left[\frac{(2m+1)k\pi}{2N}\right] \quad (3.43)$$

Un exemplu este ilustrat în figurile Fig. 3.2-4 și Fig. 3.2-5.

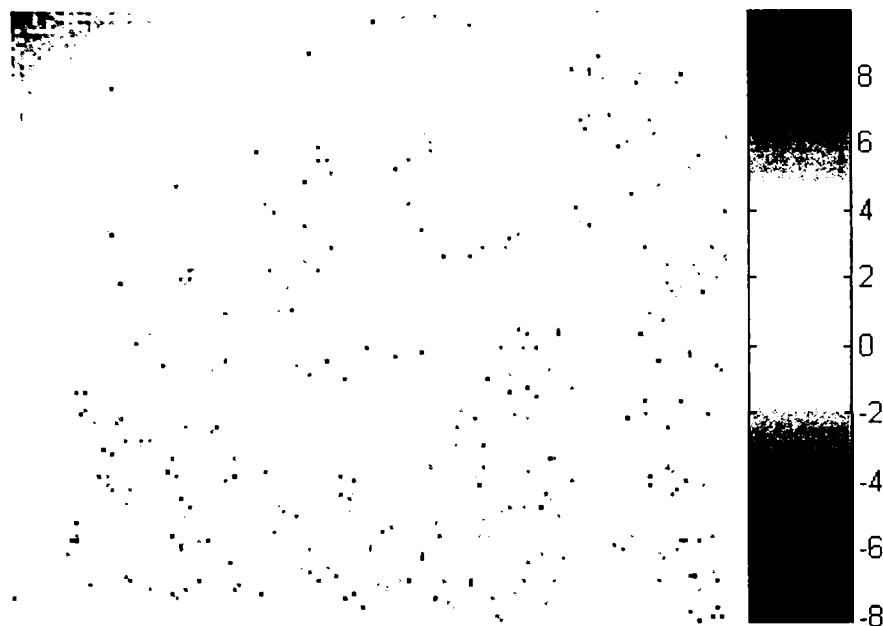


Fig. 3.2-4. Transformata DCT pentru Fig. 3.2-2

Transformarea cosinus discretă are, de asemenea, un algoritm rapid de implementare dar, spre deosebire de transformarea Fourier discretă, are numai *valori reale*. Ca domenii de aplicabilitate, este cea mai folosită transformare în compresia de imagini.



Fig. 3.2-5 Imagine din Fig. 3.2-2 reconstruită (16x16 coef DCT)

3.2.6. Transformarea Sinus discretă

Transformarea sinus discretă bidimensională, introdusă de A. Jain, este definită de relația:

$$v(k, l) = \frac{2}{N+1} \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} u(m, n) \cdot \sin \left[\frac{\pi(m+1)(k+1)}{N+1} \right] \sin \left[\frac{\pi(n+1)(l+1)}{N+1} \right] \quad (3.44)$$

iar transformarea inversă:

$$u(m, n) = \frac{2}{N+1} \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} v(k, l) \cdot \sin \left[\frac{\pi(m+1)(k+1)}{N+1} \right] \sin \left[\frac{\pi(n+1)(l+1)}{N+1} \right] \quad (3.45)$$

Transformarea sinus discretă are elementele matricii unitare de forma:

$$S_{m,k} = \sqrt{\frac{2}{N+1}} \sin\left[\frac{\pi(m+1)(k+1)}{N+1}\right] \quad (3.46)$$

Spre deosebire de alte transformări de tip sinusoidal, transformarea *DST* este mai ușor de implementat dacă $N = 2^p - 1$, cu p număr întreg. În acest caz, poate fi considerată ca partea imaginară a unei transformări Fourier rapide de dimensiune $(2N + 2)$. Are, totodată un algoritm rapid de implementare și proprietățile lui impun pentru aplicații de tip compresie de imagini.

3.2.7. Transformarea Hartley

Transformarea Hartley este o transformare integrală continuă, propusă de matematicianul cu același nume. Ca alternativă la transformarea Fourier. Varianta discretă a acestei transformări (*DHT*) este dată de relațiile:

$$v(k,l) = \frac{1}{N} \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} u(m,n) \cdot \text{cas}\left[\frac{2\pi}{N}(mk + nl)\right] \quad (3.47)$$

și

$$u(m,n) = \frac{1}{N} \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} v(k,l) \cdot \text{cas}\left[\frac{2\pi}{N}(mk + nl)\right] \quad (3.48)$$

folosind funcția de bază:

$$\text{cas}(\theta) = \cos(\theta) + \sin(\theta) = \sqrt{2} \cos(\theta - \pi/4) \quad (3.49)$$

Elementele matricii unitare de transformare Hartley sunt de forma:

$$h_{m,k} = \frac{1}{\sqrt{N}} \left[\text{cas}\left(2\pi \frac{mk}{N}\right) \right] \quad (3.50)$$

Spre deosebire de transformarea Fourier discretă, care transformă N numere reale în N numere complexe cu simetrie conjugată, transformarea Hartley discretă produce N numere reale. Ea este strâns legată de transformarea Fourier discretă fiind pur și simplu diferența dintre partea reală și partea imaginară a transformării Fourier corespondente. Analog, transformarea Fourier este partea pară minus j ori partea impară a transformării Hartley.

Din acest motiv *DHT* se constituie o alternativă din punctul de vedere al implementării pe calculator, a transformării *DFT*, având și ea un algoritm rapid de implementare. Anumite aplicații de filtrare liniară de imagini, în special în cazul unor imagini simetrice, impun folosirea *DHT* datorită reducerii semnificative a volumului de calcule (se evită lucrul cu numere complexe).

Transformarea Walsh/Hadamard

Matricea acestei transformări poate fi descrisă astfel:

$$[A_W] = \left[(-1)^{\sum_{i=0}^{u-1} u_i x_i} \right] \quad (3.51)$$

unde u_i și x_i sunt biți diadici reprezentând indicii liniei și coloanei.

Această transformare este utilizată din mai multe motive, din care menționăm în special faptul că odată implementată ea necesită $R \cdot \log_2 R$ adunări (sau scăderi) și nu înmulțiri. În plus ea este binară prin însăși natura ei. Ea oferă rotații multidimensionale de 45° , fiind utilizată pentru selecția caracteristicilor în recunoașterea caracterelor.

3.2.8. Transformarea Haar

Transformarea Haar prezintă atât o sensibilitate locală cât și una globală. Astfel, în cazurile prezentate anterior, Fourier și Hadamard, fiecare coeficient al transformării este o funcție de toate coordonatele din spațiul inițial al formelor, în timp ce în cazul transformatei Haar aceasta rămâne valabil doar pentru primii doi coeficienți, restul coeficienților măsurând corelarea adiacentă a coordonatelor. Astfel, $R/2$ coeficienți din transformarea Haar exprimă corelarea coordonatelor din spațiul inițial luate două câte două deodată, $R/4$ - câte patru în același timp etc., iar R/R - cele R coordonate din spațiul formelor. Transformarea este implementată sub forma a $2(R-1)$ adunări (sau scăderi). Această transformare a fost propusă ca un posibil selector de caracteristici în recunoașterea vocii și a vorbirii deoarece ea conține atât informații locale cât și globale.

Funcțiile Haar fiind variabile de doi parametri (scală și poziție) indexarea lor trebuie făcută după o schemă duală după cum urmează :

- se consideră k un întreg , $0 \leq k \leq N-1$ unic determinat de alți doi întregi p și q astfel :

$$k = 2^p + q - 1 \quad (3.52)$$

- funcțiile Haar se definesc astfel:

$$h_0(x) = \frac{1}{\sqrt{N}} \quad (3.53)$$

$$h_k(x) = \frac{1}{\sqrt{N}} \begin{cases} 2^{p/2}, & \text{dacă } \frac{q-1}{2^p} \leq x < \frac{q-1/2}{2^p} \\ -2^{p/2}, & \text{dacă } \frac{q-1/2}{2^p} \leq x < \frac{q}{2^p} \\ 0, & \text{altfel} \end{cases}$$

În continuare prezentăm cele trei transformări Fourier , Walsh/Hadamard și Haar, pentru un spațiu al formelor inițial cu $R=8$:

transformata Fourier:

$$[A_F] = \frac{1}{2\sqrt{2}} \begin{bmatrix} w^0 & w^0 & w^0 & w^0 & w^0 & w^0 & w^0 & w^0 \\ w^0 & w^1 & w^2 & w^3 & w^4 & w^5 & w^6 & w^7 \\ w^0 & w^2 & w^4 & w^6 & w^0 & w^2 & w^4 & w^6 \\ w^0 & w^8 & w^6 & w^1 & w^4 & w^7 & w^2 & w^5 \\ w^0 & w^4 & w^0 & w^4 & w^0 & w^4 & w^0 & w^4 \\ w^0 & w^5 & w^2 & w^7 & w^4 & w^1 & w^6 & w^3 \\ w^0 & w^6 & w^4 & w^2 & w^0 & w^6 & w^4 & w^2 \\ w^0 & w^7 & w^6 & w^5 & w^4 & w^3 & w^2 & w^1 \end{bmatrix} \quad w = \exp\left\{\frac{2\pi i}{8}\right\} \quad (3.54)$$

transformata Haar (Fig. 3.2-6):

$$[A_H] = \frac{1}{\sqrt{8}} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} \\ 2 & -2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & -2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & -2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & -2 \end{bmatrix} \quad (3.55)$$



Fig. 3.2-6 Transformata Haar aplicată Fig. 3.2-2

transformata Walsh/Hadamard:

$$[A_w] = \frac{1}{2\sqrt{2}} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix} \quad (3.56)$$

Este evident că într-un spațiu finit din punct de vedere dimensional orice transformare poate fi luată în considerare pentru obținerea unui domeniu de selectare a caracteristicilor. Desigur, spațiul transformărilor ortogonale este infinit dar o serie de transformări de rotație se dovedesc a fi extrem de utile, oferind un domeniu semnificativ pentru selectarea caracteristicilor. Două dintre transformările menționate mai sus, Fourier și Walsh, se dovedesc extrem de utile deoarece oferă algoritmi implementabili foarte rapizi. Ambele tehnici permit factorizarea transformării într-un produs de matrice, fiecare de dimensiune $R \times R$ și cu un număr mic de elemente diferite de zero, astfel încât rotația spațiului formelor poate fi realizată printr-un număr de $R \cdot \log_2 R$ operații, față de R^2 .

3.2.9. Transformata Wavelet

3.2.9.1. Introducere

Un interes considerabil a apărut în ultimii ani referitor la noi tehnici de transformare care se adresează în special *compresiei de imagine, detecției și extragerii conturului și analizei texturilor*. Aceste tehnici au apărut în domeniul analizei multirezoluție, analizei timp-frecvență, algoritmilor de tip piramidal și *transformatei wavelet*.

În acest subcapitol se revăd câteva limitări ale transformatei Fourier tradiționale și a altor transformări similare și se definesc trei tipuri de transformate wavelet care permit îmbunătățirea performanțelor pentru anumite aplicații. Vor fi prezentate câteva direcții de dezvoltare care aparțin analizei Wavelet, cu notarea atributelor comune care subsumează aceste dezvoltări sub denumirea de transformate wavelet.

Prezentarea teoretică se va limita doar la transformarea funcțiilor de valori reale, măsurabile, uni- și bidimensionale, deoarece pentru transformarea semnalelor și imaginilor doar acestea prezintă importanță.

Transformata Fourier utilizează ca funcții de bază ortonormale unde sinusoidale. Pentru transformările integrale, aceste funcții se extind la infinit în ambele direcții. Vectorii de bază ai transformatei discrete Fourier sunt, de asemenea, diferiți de zero pe întreg domeniul și din această cauză nu au un suport compact.

Prin contrast, semnalele tranzitorii sunt diferite de zero doar pe un interval scurt de timp. Analog acestora, multe atribute importante din imagine (contururile, de exemplu) sunt localizate spațial. Asemenea componente nu se aseamănă cu nici una din funcțiile de bază Fourier și nu sunt reprezentate compact în coeficienții transformării. Aceste aspecte fac transformarea Fourier, precum și alte transformări ce utilizează unde ca funcții de bază, să nu fie optime pentru compresia și analiza imaginilor care conțin componente tranzitorii sau localizate.

Pentru corectitudine, trebuie notat faptul că transformata Fourier poate reprezenta orice funcție analitică (chiar și un semnal de bandă îngustă tranzitoriu) ca o sumă de sinusoidale. Acest lucru se face prin aranjarea destul de complicată a coeficienților sinus,

pentru a crea o funcție care este zero aproape pe tot intervalul. Această comportare este normală pentru o transformare reversibilă, dar spectrul va conține o multitudine de componente spectrale.

Pentru a combate această deficiență, s-au făcut studii în direcția utilizării unor transformări având funcții de bază de durată limitată. Aceste funcții de bază variază atât în poziție cât și în frecvență, sunt unde de durată limitată și sunt referite ca **wavelets** (**undișoare**). Transformările bazate pe wavelet se numesc *transformări wavelet*.

Figura Fig. 3.2-7 ilustrează diferența dintre unde și wavelets. Primele două grafice de sus reprezintă unde cosinus care diferă în frecvență, dar nu și în durată. Ultimele două curbe din figură sunt wavelets, care diferă atât prin frecvență cât și prin poziția de-a lungul axei.

Transformata Haar este un prim exemplu de ceea ce se numește transformata wavelet. Ea diferă de alte transformări prin aceea că vectorii de bază sunt generați în totalitate prin translația și scalarea unei singure funcții.

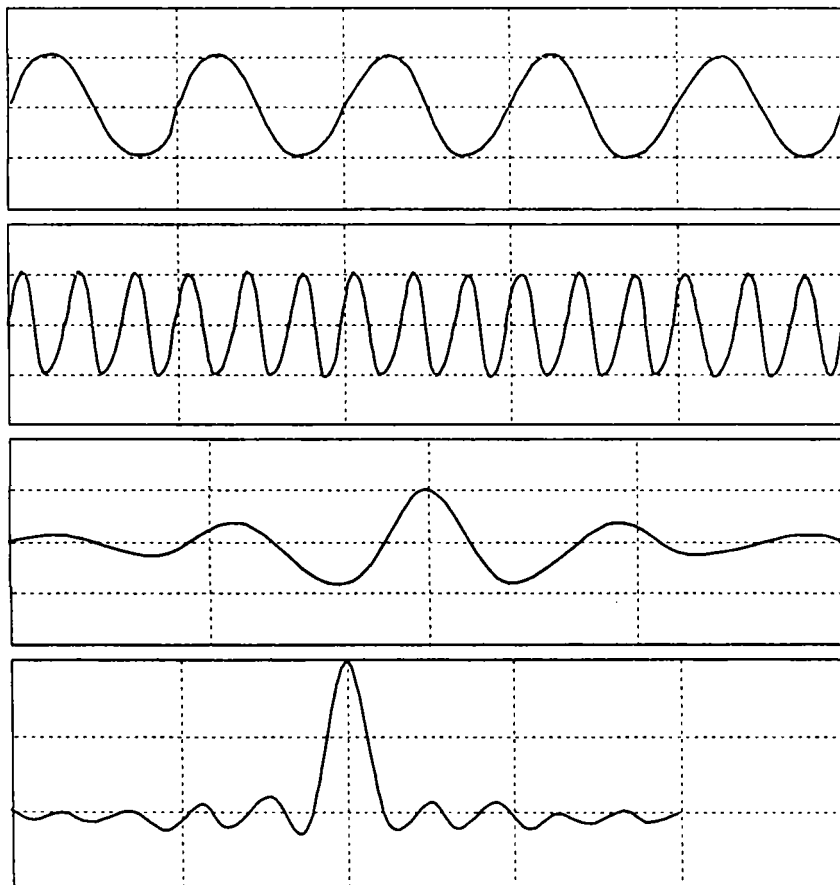


Fig. 3.2-7 Unde și undișoare ("wavelets")

3.2.9.2. Analiza timp – frecvență

Literatura în domeniul procesării de semnal include multe titluri referitoare la analiza de semnal în spațiul bidimensional timp – frecvență. În concordanță cu această modalitate de a privi un semnal, fiecare componentă tranzitorie a acestuia este poziționată în planul timp – frecvență într-un punct ce corespunde componentei de frecvență predominante și timpul de apariție.

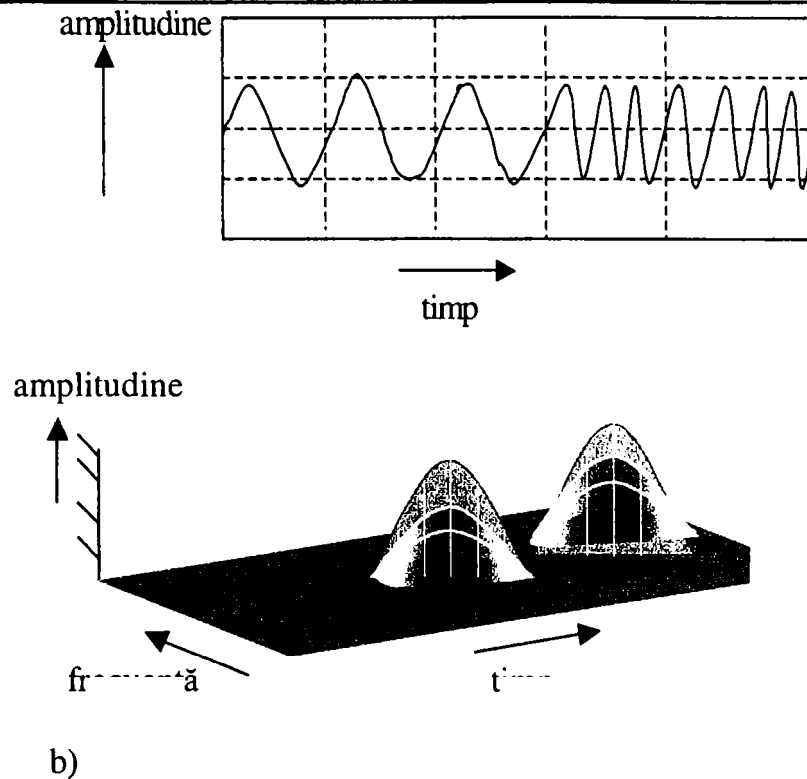


Fig. 3.2-8 Spațiul timp-frecvență
a) semnalul b) reprezentarea în acest spațiu

În analiza de imagine, spațiul este tridimensional și poate fi văzut ca o stivă de imagini. O componentă localizată va apărea la acel nivel în stivă corespunzător frecvenței componentei predominante. Figura Fig. 3.2-8 prezintă o imagine având două componente localizate, care este trecută prin două filtre trece bandă. În acest caz cele două filtre izolează aproape complet cele două componente.

3.2.10. Tipuri de transformări Wavelet

Reamintim faptul că fiecare coeficient dintr-o transformare este determinat prin produsul intern dintre funcția de intrare și una din funcțiile de bază. Această valoare reprezintă, într-o oarecare măsură, gradul de similaritate dintre funcția de intrare și acea funcție de bază particulară. Dacă funcțiile de bază sunt ortogonale (sau ortonormale), atunci produsul intern dintre două funcții de bază este zero, indicând faptul că ele sunt complet diferite. De aici, dacă semnul sau imaginea este formată din componente care sunt similare cu una sau mai multe funcții de bază, atunci toți coeficienții, în afară de unul sau câțiva, vor fi mici.

În mod asemănător, transformarea inversă poate fi văzută ca o reconstituire a semnalului sau imaginii originale, prin însumarea funcțiilor de bază, ponderate în amplitudine de coeficienții transformării. Deci, dacă semnalul sau imaginea sunt construite din componente care sunt similare cu una sau cu câteva dintre funcțiile de bază, atunci însumarea respectivă va necesita doar câțiva termeni de amplitudine semnificativă. Mulți dintre termeni pot fi ignorați, iar semnalul sau imaginea pot fi reprezentate compact prin doar câțiva coeficienți ai transformării.

Pe de altă parte, în cazul similarității componentelor de interes dintr-un semnal sau imagine cu una sau câteva dintre funcțiile de bază, atunci acele componente vor produce coeficienți de valoare mare corespunzător pentru acele (și doar acele) funcții bază, fiind deci ușor de găsit între coeficienții transformării. În final, dacă o componentă indezirabilă (zgomot, de exemplu) este similară cu una sau câteva dintre funcțiile de bază, ea va fi ușor de localizat

și, implicit de anulat prin reducerea (sau setarea la zero) a valorilor coeficienților corespunzători acelor funcții de bază.

În concluzie se remarcă valoarea potențială a utilizării transformărilor având funcții de bază similare cu componentele posibil să apară în imaginile ce vor fi transformate. Este de asemenea de notat că acele componente care sunt tranzitorii nu pot fi similare cu funcții de bază Fourier ori cu alte transformări care folosesc ca vectori de bază unde.

Există trei tipuri de transformări wavelet: *transformata wavelet continuă (CWT)*, *dezvoltarea în serie wavelet și transformata wavelet discretă (DWT)*. Situația este puțin mai complicată însă, deoarece funcțiile de bază wavelet pot fi ortonormale sau neortonormale.

Un set de funcții wavelet de bază pot genera o transformare chiar dacă funcțiile nu sunt ortonormale. Aceasta înseamnă că o dezvoltare în serie wavelet poate reprezenta o funcție de bandă limitată printr-o infinitate de coeficienți. Dacă această secvență se trunchiază la o lungime finită, se va putea reconstitui doar o aproximare a funcției originale. Din această cauză, transformata wavelet poate cere mai mulți coeficienți decât numărul de eșantioane ale funcției originale, pentru a o reconstitui exact sau cu aproximare acceptabilă.

Clasele de funcții care pot fi reprezentate prin transformări wavelet sunt acelea care sunt de pătrat integrabile în spațiul real. Aceste clase se notează cu $L^2(\mathbb{R})$. Astfel notația $f(x) \in L^2(\mathbb{R})$ înseamnă

$$\int_{-\infty}^{\infty} |f(x)|^2 dx < \infty \quad (3.57)$$

În analiza wavelet se generează un set de funcții de bază prin dilatarea și translația unei singure funcții prototip, $\Psi(x)$, care se numește *funcție de bază wavelet*. Această funcție, de obicei centrată în origine, scade rapid spre zero când $|x| \rightarrow \infty$, deci $\Psi(x) \in L^2(\mathbb{R})$.

3.2.10.1. Transformarea wavelet continuă

Dacă $\Psi(x)$ este o funcție reală a cărei transformată Fourier, $\Psi(s)$ satisface criteriul de admisibilitate:

$$C_{\varphi} = \int_{-\infty}^{\infty} \frac{|\varphi(s)|^2}{|s|} ds < \infty \quad (3.58)$$

atunci $\Psi(x)$ se numește funcție de bază wavelet. De notat că, datorită apariției lui s la numitor, este necesar ca:

$$\varphi(0) = 0, \text{ adică } \int_{-\infty}^{\infty} \varphi(x) dx = 0 \quad (3.59)$$

Mai mult, deoarece $\Psi(\infty) = 0$, amplitudinea spectrului pentru o funcție wavelet este asemănătoare cu funcția de transfer a unui filtru trece bandă. De fapt, orice răspuns la impuls al unui filtru trece bandă cu medie zero (3.59), care cade în zero destul de repede cu creșterea frecvenței (3.58), poate servi ca funcție de bază wavelet pentru această transformare.

Un set de funcții de bază $\{\Psi_{a,b}(x)\}$ poate fi generat prin translatarea și scalarea funcției wavelet $\Psi(x)$, astfel:

$$\varphi_{a,b}(x) = \frac{1}{\sqrt{a}} \varphi\left(\frac{x-b}{a}\right) \quad (3.60)$$

unde $a > 0$ și b este un număr real. Variabila a reprezintă scara (lărgimea) unei funcții bază particulare, în timp ce b specifică translația de-a lungul axei x .

De obicei, funcția de bază wavelet este centrată în origine, deci $\Psi_{a,b}(x)$ este centrată în $x = b$. Figura Fig. 3.2-9 reprezintă un exemplu de funcție wavelet. Aceasta este dată prin:

$$\varphi(x) = \frac{2}{\sqrt{3\sqrt{\pi}}} (1 - x^2) e^{-x^2/2} \quad (3.61)$$

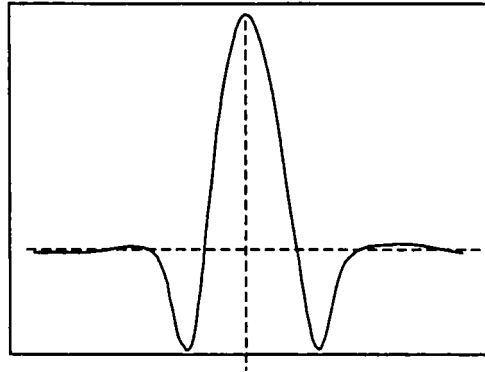


Fig. 3.2-9 Exemplu de funcție wavelet

Transformata wavelet continuă a funcției $f(x)$, având ca funcție de bază $\Psi(x)$ este:

$$W_f(a,b) = \langle f, \varphi_{a,b} \rangle = \int_{-\infty}^{\infty} f(x) \varphi_{a,b}(x) dx \quad (3.62)$$

Coefficienții transformatei wavelet sunt dați de produsul intern al funcției de transfer cu fiecare dintre funcțiile bază.

Se poate arăta că transformarea continuă inversă wavelet este dată de relația:

$$f(x) = \frac{1}{C_\varphi} \int_0^\infty \int_{-\infty}^\infty W_f(a,b) \varphi_{a,b}(x) db \frac{da}{a^2} \quad (3.63)$$

Factorul de scalare din fața lui $\varphi\left(\frac{x-b}{a}\right)$ în ecuația (3.60) asigură faptul că norma funcțiilor de bază este egală, deoarece:

$$\left\| \varphi\left(\frac{x-b}{a}\right) \right\| = \sqrt{\int_{-\infty}^\infty \left| \varphi\left(\frac{x-b}{a}\right) \right|^2 dx} = \sqrt{a} \|\varphi(x)\| \quad (3.64)$$

Deoarece funcția de bază wavelet are media zero (ecuația (3.59)), toate funcțiile scalate și translatate vor avea media zero.

3.2.10.2. Transformarea Wavelet continuă bidimensională

Transformarea Wavelet continuă $W(a,b)$ a unei funcții unidimensionale $f(x)$ este o funcție de două variabile. Pentru funcții de mai mult de o variabilă, această transformare are, de asemenea, dimensiunea mai mare cu o unitate.

Dacă $f(x,y)$ este o funcție bidimensională, transformarea wavelet continuă este:

$$W_f(a,b_x,b_y) = \int_{-\infty}^\infty \int_{-\infty}^\infty f(x,y) \varphi_{a,b_x,b_y}(x,y) dx dy \quad (3.65)$$

unde b_x și b_y specifică translația pe cele două direcții. Transformarea wavelet continuă inversă este:

$$f(x, y) = \frac{1}{C\varphi} \int_0^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} W_f(a, b_x, b_y) \varphi_{a, b_x, b_y}(x, y) db_x db_y \frac{da}{a^3} \quad (3.66)$$

unde:

$$\varphi_{a, b_x, b_y}(x, y) = \frac{1}{|a|} \varphi\left(\frac{x - b_x}{a}, \frac{y - b_y}{a}\right) \quad (3.67)$$

și $\Psi(x, y)$ este o funcție bidimensională.

3.2.10.3. Interpretarea cu ajutorul bancurilor de filtre

Următorul exemplu ilustrează un mod în care poate fi privită transformarea wavelet continuă. Mai întâi se definește funcția generală de bază la scara a :

$$\varphi_a(x) = \frac{1}{\sqrt{a}} \varphi\left(\frac{x}{a}\right) \quad (3.68)$$

Ecuția definește un set de funcții care devin mai largi, odată cu creșterea lui a . Se definește de asemenea:

$$\bar{\varphi}_a(x) = \varphi_a^*(-x) = \frac{1}{\sqrt{a}} \varphi^*\left(-\frac{x}{a}\right) \quad (3.69)$$

care este componenta reflectată la scara a a conjugatei funcției complex wavelet. Dacă $\Psi(x)$ este reală și pară (cum se întâmplă în majoritatea cazurilor), reflectarea și conjugarea nu au nici un efect.

În aceste condiții, transformarea wavelet continuă se poate scrie ca:

$$W_f(a, b) = \int_{-\infty}^{\infty} f(x) \bar{\varphi}_a(b - x) dx = f \otimes \bar{\varphi}_a \quad (3.70)$$

Pentru a fix, $W_f(a, b)$ se obține prin convoluția lui $f(x)$ cu funcția de bază reflectată și conjugată la scara a .

Figura Fig. 3.2-10 prezintă transformarea continuă wavelet sub forma unui banc de filtre liniare (convoluție) având la intrare funcția $f(x)$. Fiecare valoare a lui a definește un alt filtru trece bandă, iar ieșirile tuturor filtrelor, luate împreună, alcătuiesc transformarea wavelet.

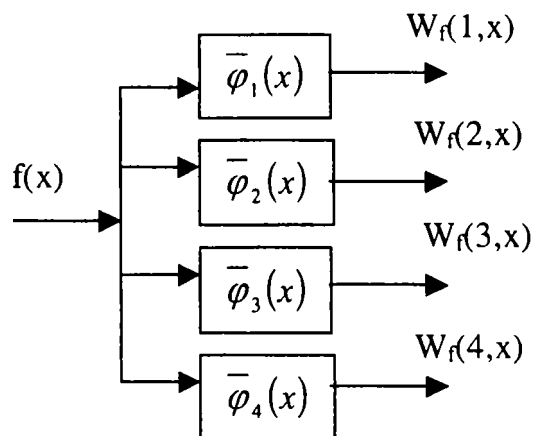


Fig. 3.2-10 Analogia transformatei wavelet cu bancuri de filtre

Egalitatea (3.63) devine:

$$f(x) = \frac{1}{C_\varphi} \int_0^\infty \int_{-\infty}^\infty [f \otimes \bar{\varphi}_a](b) \varphi_a(x-b) db \frac{da}{a^2} = \frac{1}{C_\varphi} \int_0^\infty [f \otimes \bar{\varphi}_a \otimes \varphi_a](x) \frac{da}{a^2} \quad (3.71)$$

cea ce implică faptul că ieșirile filtrelor, fiecare dintre ele filtrată din nou prin $\Psi_a(x)$ și scalată corespunzător, conduc la semnalul reconstituit $f(x)$.

De notat că, din teorema de similaritate, avem:

$$\mathfrak{S}\{f(ax)\} = \frac{1}{|a|} F\left(\frac{s}{a}\right) \quad (3.72)$$

De unde rezultă că:

$$\varphi_a(s) = \mathfrak{S}\{\varphi_a(x)\} = \sqrt{a} \varphi(as) \quad (3.73)$$

cea ce înseamnă că centrul benzii de frecvență al filtrelor trece bandă descrește odată cu îngustarea funcției de transfer datorată creșterii valorii lui a .

Câteva funcții wavelet tipice și transformatele asociate sunt prezentate în figura Fig. 3.2-11 .

Din ecuația (3.73) se poate trage concluzia că o contracție într-un domeniu este însoțită de o expansiune în celălalt, dar în mod neuniform în planul timp – frecvență; acest fapt corespunde și principiului de incertitudine. În figura următoare Fig. 3.2-12 este prezentat planul timp-frecvență pentru transformata wavelet.

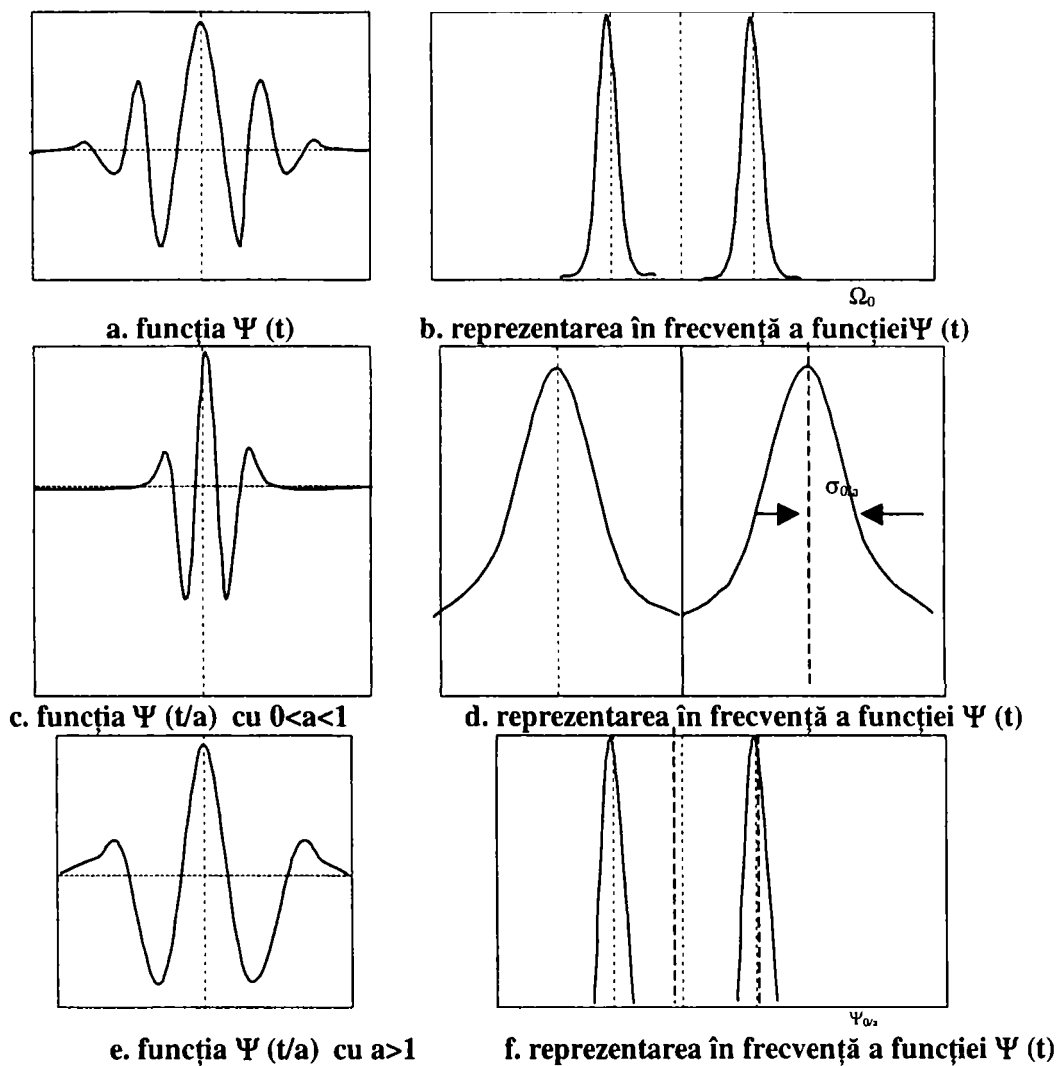


Fig. 3.2-11 Exemplu de scalare a unei funcții wavelet

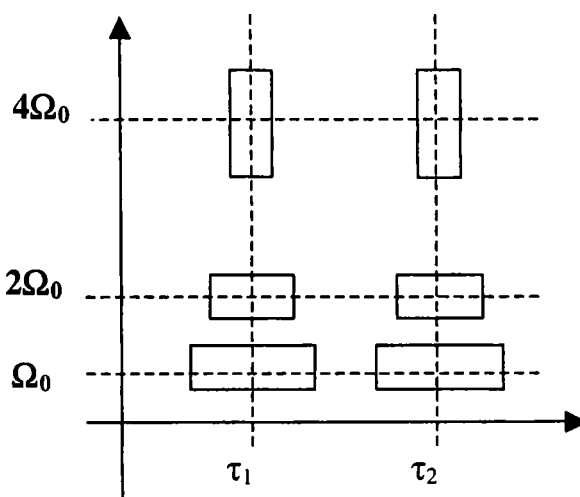


Fig. 3.2-12 Planul timp frecvență și celulele de rezoluție pentru transformata Wavelet.

3.2.11. Implementarea cu bancuri de filtre (cazul bidimensional)

Figura Fig. 3.2-13 ilustrează implementarea cu ajutorul bancurilor de filtre pentru cazul bidimensional. Aici, fiecare filtru $\Psi_a(x,y)$ este un răspuns la impuls bidimensional, iar ieșirea sa este imaginea filtrată trece-bandă. Șirul de imagini filtrate alcătuiesc transformata wavelet. Din nou redundanța este considerabilă. De fapt, teoretic, dacă $\Psi(u,v)$ care este funcția de transfer a lui $\Psi(x,y)$ este diferită de zero în orice punct, cu excepția originii, putem teoretic reconstitui imaginea originală din oricare dintre ieșirile filtrului prin filtrare inversă (prin deconvoluție).

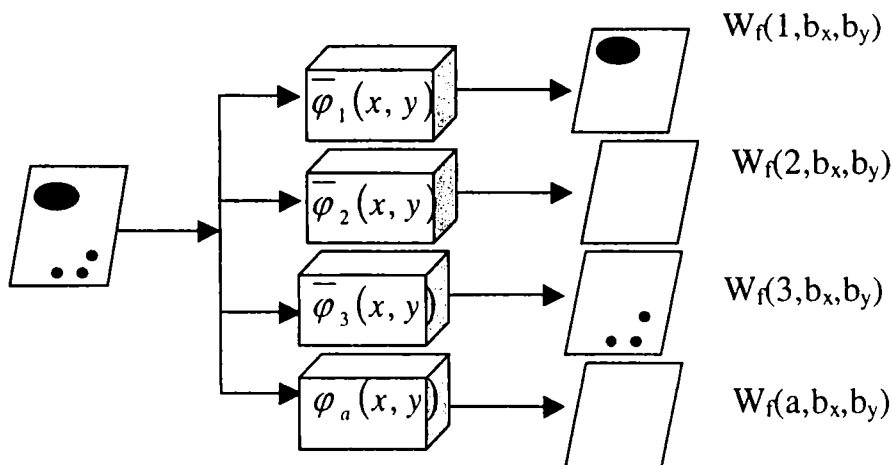


Fig. 3.2-13 Analogia cu bancuri de filtre pentru transformata wavelet continuă bidimensională.

O alternativă, dacă imaginea este de bandă limitată și măcar o funcție $\Psi_a(u,v)$ este diferită de zero, constă în refacerea lui $f(x,y)$ doar din ieșirile aceluia filtru. *Concluzia este că valoarea transformatei wavelet continue nu constă în reprezentarea compactă, ci în decompoziția și analiza semnalelor și imaginilor.*

Pentru a ilustra aceasta, să presupunem că imaginea din figura Fig. 3.2-13 conține, de exemplu, obiecte circulare de diferite dimensiuni și funcția bază răspunde numai la obiecte circulare de rază unitate. Atunci fiecare obiect va apărea numai în acea imagine de ieșire care corespunde cu dimensiunea sa.

3.2.12. Dezvoltarea în serie wavelet

3.2.12.1. Wavelet diadic

Al doilea tip de transformare wavelet este mai restrictivă decât prima. Din nou, o funcție bază este scalată și traslatată pentru a forma un set de funcții de bază. Aici, însă, scalarea și translația sunt specificate prin întregi și nu prin numere reale.

În această modalitate de definire ne vom limita doar la formarea funcțiilor de bază prin scalare binară (cu factor 2) și translație diadică a funcției de bază $\Psi(x)$. O translație diadică reprezintă o deplasare („shift”) cu $k/2^j$, care este un multiplu întreg al factorului de scară și deci a lățimii funcției wavelet. Scalarea binară și translația diadică sunt ilustrate în figura Fig. 3.2-14.

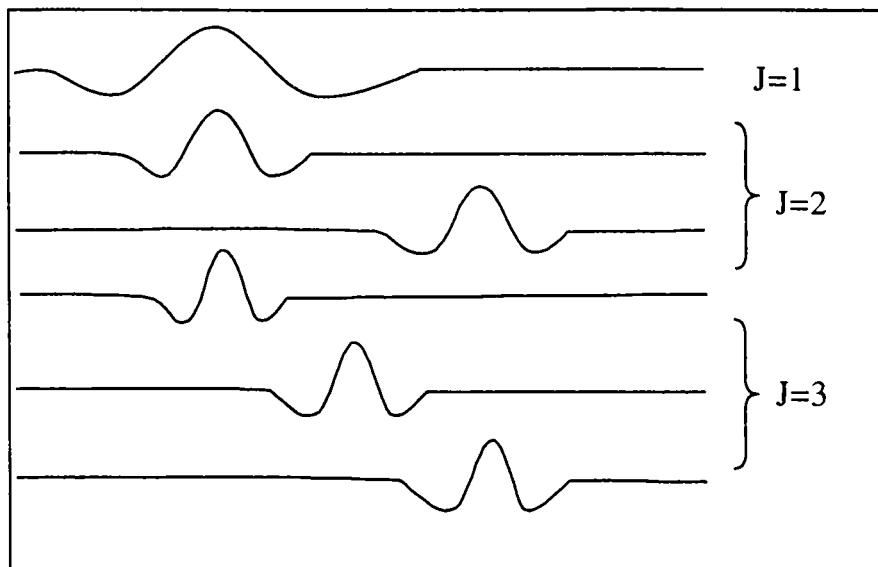


Fig. 3.2-14 Scalarea binară și translatarea diadică a unei funcții wavelet

O funcție $\Psi(x)$ este o funcție wavelet ortogonală dacă setul $\{\Psi_{j,k}(x)\}$ de funcții definite de:

$$\Psi_{j,k}(x) = 2^{j/2} \Psi(2^j x - k) \quad (3.74)$$

Unde $-\infty < j, k < \infty$, sunt întregi, formează o bază ortonormală în $L^2(\mathcal{R})$. Parametrul j determină dilatarea, în timp ce k specifică translația.

Precedentul set de funcții wavelet formează o bază ortonormală dacă:

$$\langle \Psi_{j,k}, \Psi_{l,m} \rangle = \delta_{j,k} \delta_{l,m} \quad (3.75)$$

unde l și m sunt întregi, $\delta_{j,k}$ este funcția delta Kronecker și $\langle \bullet, \bullet \rangle$ indică produsul intern. O a doua condiție este ca orice funcție $f(x) \in L^2(\mathcal{R})$ să poată fi scrisă ca:

$$f(x) = \sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} c_{j,k} \varphi_{j,k}(x) \quad (3.76)$$

unde coeficienții transformării sunt dați din nou de produsul intern, care este:

$$c_{j,k} = \langle f(x), \varphi_{j,k} \rangle = 2^{j/2} \int_{-\infty}^{\infty} f(x) \varphi(2^j x - k) dx \quad (3.77)$$

Ecuatiile (3.76) și (3.77) definesc dezvoltarea în serie wavelet a funcției $f(x)$ relativ la funcția wavelet $\Psi(x)$.

Dat fiind faptul că funcțiile de bază se extind, de obicei, la infinit în ambele direcții, o reconstrucție completă ar trebui să includă toți termenii. Totuși, dacă $\Psi(x)$ este aleasă corespunzător, se poate trunchia seria fără erori serioase de aproximare. Dacă $f(x)$ este de durată finită, iar funcția bază wavelet este bine localizată (se apropie rapid de zero departe de origine), atunci mulți coeficienți cu $|k|$ mare vor fi neglijați. Totodată, coeficienți cu $|j|$ mare vor avea, de asemenea, valori mici.

3.2.12.2. Funcții Wavelet diadice compacte

Dacă se restrâng și mai mult $f(x)$ și funcțiile de bază wavelet, la funcții care sunt zero în afara intervalului $[0,1]$ atunci familia de funcții de bază ortonormale poate fi specificată printr-un singur indice n :

$$\Psi_n(x) = 2^{j/2} \Psi(2^j x - k) \quad (3.78)$$

unde j și k sunt acum funcție de n , după cum urmează :

$$n = 2^j + k \text{ pentru } j = 0, 1, \dots \quad k = 0, 1, \dots, 2^j - 1 \quad (3.79)$$

Pentru oricare n , j este cel mai mare întreg, astfel încât $2^j \leq n$ și $k = n - 2^j$. În aceste condiții, transformata wavelet inversă este:

$$f(x) = \sum_{n=0}^{\infty} c_n \varphi_n(x) \quad (3.80)$$

unde se presupune că $\Psi_0(x) = 1$. Coeficienții transformării sunt dați de produsul intern:

$$c_n = \langle f(x), \varphi_n \rangle = 2^{j/2} \int_{-\infty}^{\infty} f(x) \varphi(2^j x - k) dx \quad (3.81)$$

Aici o funcție continuă este reprezentată printr-o singură secvență infinită, la fel ca la reprezentarea în serie Fourier. Redundanța nedorită a transformării continue wavelet nu mai apare. De fapt, dacă una sau câteva dintre $\Psi_n(x)$ sunt similare cu $f(x)$ (sau cu componentele sale majore), atunci este posibil să trunchiem seria la un număr redus de termeni, fără o eroare apreciabilă de aproximare.

Rezultă, astfel, vectorii de bază ai transformatei discrete wavelet. Dacă $f(i\Delta t)$ este o funcție discretă eșantionată pe N puncte (unde N este o putere a lui 2), iar dacă $\Psi(x)$ este o funcție wavelet diadică compactă, atunci putem calcula transformata wavelet discretă utilizând variantele discrete ale ecuațiilor (3.80) și (3.81) ambele ecuații devin sume având N termeni.

Transformata Haar este un prim exemplu de ceea ce numim transformare wavelet ortonormală, diadică, compactă. Ea diferă de celelalte transformări prin aceea că funcțiile de bază sunt toate generate prin dilatarea unei funcții de bază wavelet. Funcția Haar, care este un impuls dreptunghiular impar, este cel mai simplu și vechi exemplu de funcție wavelet ortonormală cu suport compact (Fig. 3.2-15). Funcția de bază wavelet se îngustează progresiv (reducere la scară) printr-un pas egal cu lățimea sa, în așa fel încât setul complet wavelet este scalat în jos (micșorat) cu o putere a lui 2, amplitudinea este scalată în sus (mărită) prin multiplicare cu o putere a lui 2, pentru a menține ortonormalitatea.

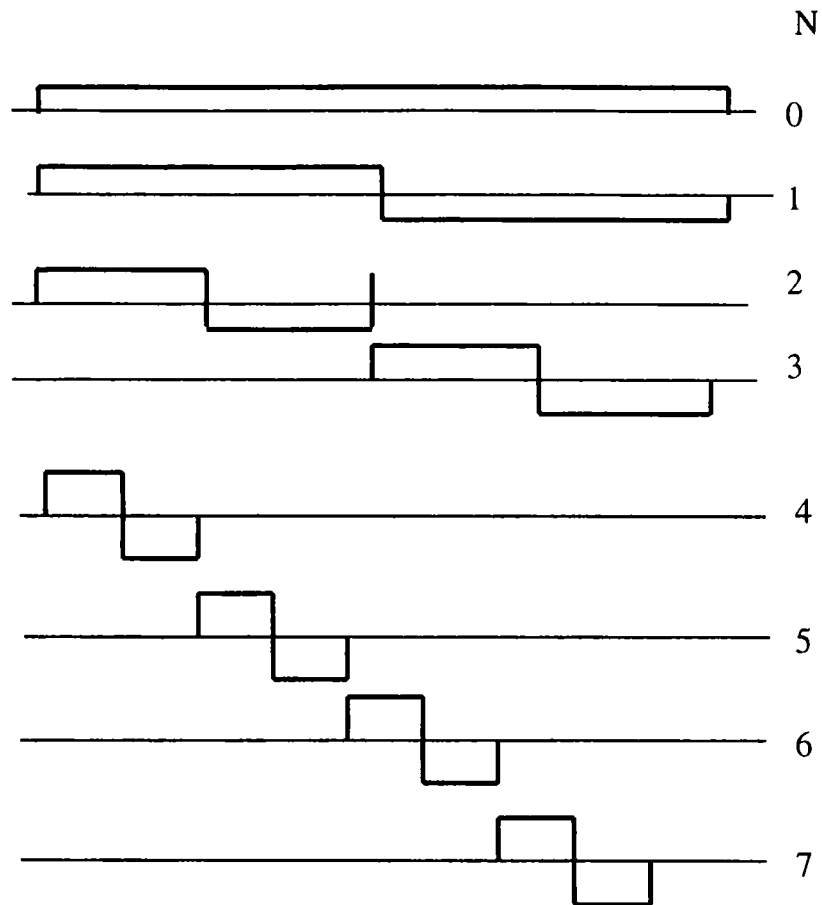


Fig. 3.2-15 Funcții de bază ale transformării Haar.

3.2.13. Transformarea wavelet discretă (DWT)

Transformarea discretă wavelet este utilă în compresia de imagini, procesare și analiză. Dându-se un set ortonormal de funcții de bază, putem calcula DWT asemănător cu orice transformare unitară (de exemplu transformarea Haar). Obținerea unui set de funcții de bază potrivit necesită, totuși, cunoștințe suplimentare. În această secțiune vom discuta două tehnici care au contribuit la dezvoltarea transformării discrete wavelet: teoria bancurilor de filtre și multirezoluția sau analiza timp-scară, utilizând reprezentarea piramidală.

3.2.13.1. Teoria bancurilor de filtre

Cercetările din domeniul analizei de voce și prelucrării semnalelor acustice au utilizat mult timp conceptul de **bancuri de filtre** trece bandă pentru descompunerea semnalelor în componente de diferite frecvențe. Într-adevăr, metoda este precursora *analizei timp-frecvență*, în care componentele semnalului sunt afișate într-un spațiu bidimensional ale cărui coordonate sunt timpul de apariție și frecvența de oscilație.

Presupunem un semnal compus din două semnale de tip „burst” (trenuri de sinusoidă de scurtă durată) înecate în zgomot. Presupunem, de asemenea, că dorim să analizăm semnalul pentru a detecta numărul, frecvența și poziția acestor tonuri „burst”.

Transformata Fourier va reflecta, desigur, întregul conținut al semnalului, dar nu într-un mod ușor de interpretat. Informația de poziție, de exemplu, este codată în faza spectrului într-o manieră complicată. Deși amplitudinea spectrului prezintă vârfuri distincte corespunzătoare fiecărei componente tranzitorii, ea poate fi utilizată doar pentru semnale a căror componente au amplitudini și durate suficient de mari pentru a domina spectrul. Spectrul de fază, însă, oferă relativ puține informații referitoare la localizarea acestor

componente în timp. Adesea, componente inutile ale semnalului, ca de exemplu zgomotul, complică spectrul în așa măsură, încât o simplă analiză în frecvență este insuficientă pentru a determina complet semnalele.

3.2.13.2. Filtre ideale trece-bandă

Să presupunem că partiționăm axa frecvenței într-un set disjunct de intervale (regiuni adiacente, nesuprapuse) și utilizăm partiționarea pentru a defini un set de funcții de transfer trece bandă ideale, așa cum se arată în figura Fig. 3.2-16.b. Răspunsul la impuls corespunzător apare în figura Fig. 3.2-16.a.

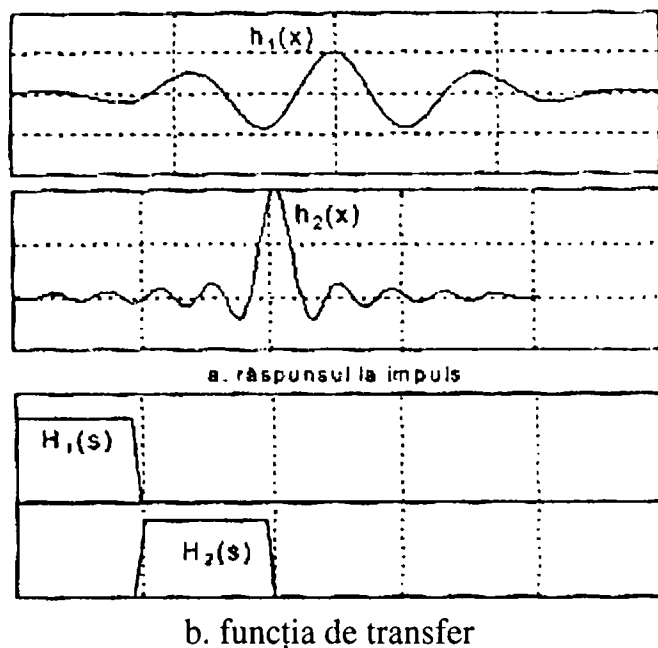


Fig. 3.2-16 Generarea unei serii de filtre trece-bandă prin partiționarea axei frecvenței

Figura Fig. 3.2-17 prezintă implementarea cu ajutorul bancurilor de filtre. Semnalul de intrare este aplicat, în paralel, la intrarea fiecărui filtru trece bandă. Ieșirile corespunzătoare sunt $g_i(x)$.

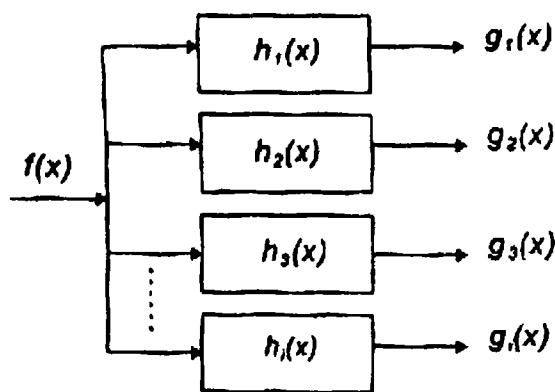


Fig. 3.2-17 Implementarea cu bancuri de filtre

Funcțiile $H_i(x)$ sunt astfel construite, încât suma lor pentru toate frecvențele este egală cu unitatea și, de aceea, $g_i(x)$ se vor însuma pentru a forma $f(x)$.

$$\sum_{i=1}^{\infty} H_i(s) = 1 \Rightarrow \sum_{i=1}^{\infty} g_i(x) = f(x) \quad (3.82)$$

Dacă avem două semnale de "burst" de frecvențe mult diferite, ele vor apărea la ieșirea unor filtre diferite și, dat fiind faptul că semnalul de la ieșirea filtrelor este în domeniul

timp, localizarea semnalelor de-a lungul axei timpului este evidentă. De aici rezultă o metodă de descompunere a semnalelor compuse, capabilă să și identifice componentele de interes.

Fiecare dintre ieșirile filtrelor trece bandă se datorează unei convoluții de forma:

$$g_i(x) = \int_{-\infty}^{\infty} f(t) \cdot h_i(x-t) dt \quad (3.83)$$

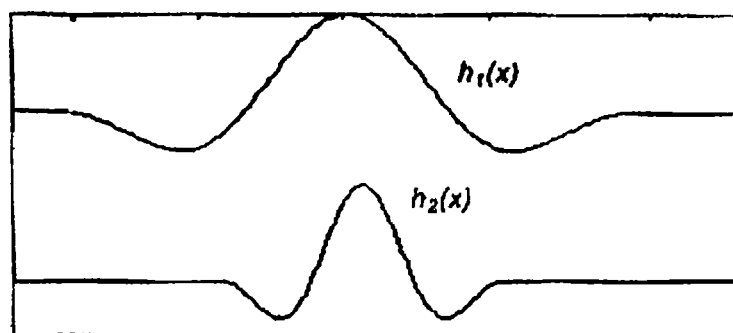
Deoarece $H_i(s)$ este real și par, $h_i(x)$ va fi de asemenea real și par, iar ieșirile filtrelor pot fi scrise ca:

$$g_i(x) = \int_{-\infty}^{\infty} f(t) \cdot h_i(t-x) dt = \langle f(t) \cdot h_i(t-x) \rangle \quad (3.84)$$

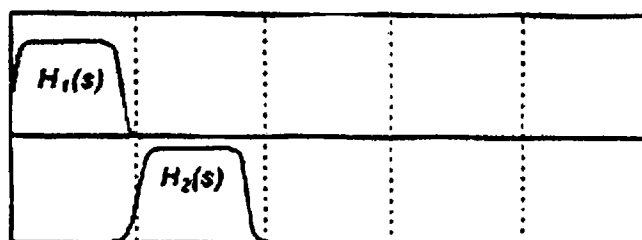
Din relația (3.84) rezultă că fiecare punct din $g_i(x)$ este produsul intern dintre $f(x)$ și o versiune a lui $h_i(t)$ care a fost deplasată la locația x . Putem considera, de asemenea, $\{g_i(x)\}$ ca un set (bidimensional) de coeficienți ai transformatei wavelet, unde $\{h_i(x)\}$ este setul de funcții de bază wavelet. Mai mult, $\{g_i(x)\}$ este suficient pentru a reconstrui exact $f(x)$, așa după cum arată ecuația (3.82).

3.2.13.3. Filtre trece-bandă netede

Funcțiile $h_i(x)$ din figura Fig. 3.2-16.a nu sunt foarte bine localizate (nu au o pantă de cădere spre zero suficient de mare în afara benzii lor de trecere-regiunea lor centrală) și deci nu aproximează suficient de bine funcțiile wavelet. Din această cauză, $h_i(x-x_0)$ va răspunde la componente puternice, localizate la distanță față de x_0 și acest lucru se datorează muchiilor ascuțite ale $H_i(s)$, care duc la lățirea nedorită a răspunsului $h_i(x)$.



a. răspunsul la impuls



b. funcția de transfer

Fig. 3.2-18 Filtre trece-bandă netede

Dacă se proiectează $H_i(s)$ în așa fel încât să aibă muchii rotunjite, atunci lărgimea lui $h_i(x)$ se va reduce. Deoarece suma tuturor $H_i(s)$ trebuie să fie egală cu unitatea pe întreg domeniul, funcțiile de transfer trece bandă rezultate se vor suprapune la muchii. Un asemenea

tip de filtre se prezintă în figura Fig. 3.2-18. Îngustarea răspunsului la impuls este evidentă. Prin utilizarea filtrelor cu caracteristică netedă se îmbunătățește localizarea componentelor tranzitorii ale semnalului atât în domeniul timp (poziție) cât și în frecvență.

3.2.14. Analiza multirezoluție

Multe dintre dezvoltările care au precedat analiza wavelet au apărut în domeniul numit, la modul general, *analiză multirezoluție* și au încercat să combată limitările transformatei Fourier. Teoria bancurilor de filtre oferă o metodă convenabilă de reprezentare a semnalelor compuse prin suprapunerea de sinusoides de diferite frecvențe (cum sunt, de exemplu, notele muzicale sau semnalele „burst”). Aceste componente includ, pe durata lor, mai multe componente. În analiza de imagine, însă, majoritatea componentelor localizate nu sunt de acest tip, deoarece pot include doar o perioadă a unui semnal, sau chiar o parte a acestuia (de exemplu liniile, contururile și punctele izolate).

Obiectele dintr-o imagine pot să apară la diferite scări. Un contur, de exemplu, poate fi o tranziție abruptă de la alb la negru sau o tranziție care apare gradual, pe o distanță considerabilă. În general, analiza multirezoluție a imaginilor caută să exploateze această idee. Cartografia ilustrează această tehnică. Hărțile sunt, de obicei, desenate la diferite scări. Scara hărții este raportul dintre mărimea unui teritoriu și reprezentarea lui pe hartă. La scară mare, ca cea a globului, obiecte de dimensiuni mari, cum sunt continentele și oceanele, sunt vizibile pe hartă, în timp ce detaliile, cum ar fi o stradă dintr-un oraș, nu mai sunt vizibile la rezoluția hărții. La scară mică, detaliile devin vizibile, în timp ce obiectele mari se pierd. Deci, pentru a fi capabili să ne orientăm într-o călătorie la mare distanță, vom avea nevoie de un set de hărți desenate la diferite scări.

Transformata wavelet s-a dezvoltat având la bază aceste idei ale analizei multirezoluție. La fel ca și în cazul analizei timp – frecvență, un semnal se reprezintă în spațiul bidimensional, dar aici verticală este scalarea funcției, și nu frecvența. *Scalarea* se obține prin dilatarea și contracția funcției wavelet de bază, pentru a forma setul funcțiilor de bază.

Funcția wavelet de bază, $\psi(x)$, este scalată ca $\psi_a(x)$ (care este o dilatare a funcției de bază, dacă $a > 1$ sau o îngustare a acesteia, dacă $a < 1$) pentru a forma setul de funcții de bază. Pentru valori mari ale lui a , funcțiile dilatate vor extrage obiectele mari, în timp ce pentru a mic, se vor scoate în evidență detaliile fine.

Teoria multirezoluției poate fi descrisă de teoria funcțiilor spațiale. Analiza multirezoluție este descrisă în spațiul $L^2(\mathbb{R})$, care adună subspațiile V_m având proprietățile:

$$\begin{aligned} \bigcap_{m \in \mathbb{Z}} V_m &= \{0\}; & \bigcup_{m \in \mathbb{Z}} V_m &= L^2(\mathbb{R}) \\ \dots \subset V_2 \subset V_1 \subset V_0 \subset V_{-1} \subset V_{-2} \subset \dots & & & \end{aligned} \quad (3.85)$$

Reprezentarea spațiului $L^2(\mathbb{R})$ (Fig. 3.2-19):

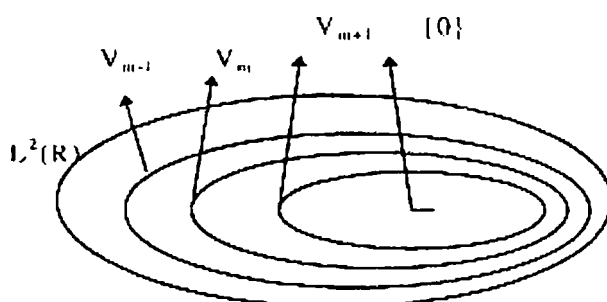


Fig. 3.2-19 Reprezentarea spațiului $L^2(\mathbb{R})$

Fiecare subspațiu V_m are proprietatea: $f(x) \in V_{m-1}$, atunci $f(2x) \in V_{m-1}$.

În final, în analiza multirezoluție, avem funcția $\Phi(x) \in V_0$, astfel că versiunile deplasate $\Phi_{0,n}(x) = \Phi(x-n)$, $n \in \mathbb{Z}$, constituie o bază ortonormală pentru V_0 , iar

$$\Phi_{m,n}(x) = \frac{1}{\sqrt{2}} \Phi(2^{-m} \cdot x - n) \quad (3.86)$$

constituie o bază ortonormală pentru V_m . Funcția $f(x)$ poate fi văzută ca o succesiune de funcții $f_m(x)$ din V_m adică:

$$f(x) = \lim_{m \rightarrow \infty} f_m(x) \quad (3.87)$$

Această proprietate permite examinarea funcției sau semnalului la rezoluții sau scări multiple. Variabila m indică scara. Dacă m crește, scara este mai largă, funcția V_m este o aproximare brută a funcției $f(x)$, iar detaliile sunt neglijate. Dacă m scade, $f(x)$ este aproximată în detaliu. Toate funcțiile din spațiul V_m pot fi reprezentate utilizând combinații liniare ale funcțiilor de scalare. De fapt, $f_m(x)$ este o proiecție ortogonală a lui $f(x)$ pe V_m .

$$f_m \sum_n C_{m,n} \cdot \Phi_{m,n}(x) \\ \Phi_{0,0}(x) = \Phi(x) \in V_0 \subset V_{-1} \quad (3.88)$$

$$\Phi_{0,0}(x) = \sqrt{2} \sum_n h_n \Phi(2x - n)$$

Prin trecerea în domeniul frecvență a formulei de mai sus obținem:

$$\Phi(\omega) = H\left(\frac{\omega}{2}\right) \cdot \Phi\left(\frac{\omega}{2}\right); \quad (3.89)$$

Prin substituirea în partea dreaptă a ecuației vom obține:

$$\Phi(\omega) = \prod_{m=1}^{\infty} H\left(\frac{\omega}{2^m}\right); \quad (3.90)$$

Datorită ortonormalității funcției de bază, h_n trebuie să îndeplinească următoarele condiții:

$$\sum_k |\Phi(\omega + 2k\pi)|^2 = 1 \Leftrightarrow |H(\omega)|^2 + |H(\omega + \pi)|^2 = 1 \quad (3.91)$$

și $H(0) = 1$, respectiv $H(\pi) = 0$

Acum se poate defini funcția $\Psi_{m,n}(x)$ ca fiind versiunea deplasată și scalată a funcției $\Psi(x)$:

$$\Psi_{0,0}(x) = \sqrt{2} \sum_n g_n \Phi_{-1,n}(x) \quad (7.41)$$

$$\Psi_{m,n}(x) = 2^{-m/2} \Psi(2^{-m} x - n)$$

La trecerea în domeniul frecvență se obține:

$$\Psi(\omega) = G\left(\frac{\omega}{2}\right) \cdot \Phi\left(\frac{\omega}{2}\right) \text{ respectiv } \Psi(\omega) = G\left(\frac{\omega}{2}\right) \cdot \prod_{m=2}^{\infty} H\left(\frac{\omega}{2^m}\right) \quad (3.92)$$

3.2.14.1. Algoritmul piramidal

Presupunem că se generează, dintr-o imagine de 1024x1024 pixeli, 10 imagini adiționale prin aplicarea succesivă a mediei pe blocuri de 2x2 pixeli, de fiecare dată eliminând a doua linie și a doua coloană. Imaginile rămase vor avea, respectiv, rezoluții de 512x512, 256x256, etc., până la imaginea de 1x1 pixel. Dacă apoi se aplică extragerea de contur pe fiecare imagine, utilizând o matrice de 3x3 elemente (vezi capitolul 8) vom obține contururile

fine în imaginea originală, apoi contururi ceva mai groase în imaginile de 512x512 și 256x256 pixeli și doar contururi foarte groase în imaginea de 16x16 pixeli și în imaginile de dimensiuni mai mici. Se poate face observația că toate contururile, mari și mici, apar și în imaginea originală, de 1024x1024 pixeli, și deci nu este necesară schimbarea rezoluției pentru a le localiza. Problema este că acele contururi largi, care se manifestă prin tranziții de nivele de gri pe o distanță considerabilă, sunt greu de detectat cu operatori convenționali de dimensiuni mici. S-ar putea scala operatorul în sus (dilata) pentru a detecta contururi largi, dar este mult mai eficient să scalăm imaginea în jos. Utilizarea operatorilor de dimensiuni mari pentru a căuta într-o imagine de mare rezoluție contururi grosiere necesită un număr enorm de operații.

3.2.14.2. Codarea piramidală laplaciană

Codarea piramidală se bazează pe funcții gaussiene. Imaginea este filtrată trece jos cu ajutorul unui filtru având un răspuns la impuls de tip gaussian, iar rezultatul este scăzut din imaginea inițială. Detaliile de frecvență înaltă din imagine sunt reținute în imaginea diferență. Imaginea filtrată trece jos poate apoi să fie subeșantionată, fără pierderea detaliilor. Algoritmul de codare piramidală este prezentat în cele ce urmează.

Să considerăm $f_0(i, j)$ imaginea originală și $g(i, j)$ răspunsul la impuls al filtrului trece jos gaussian. La fiecare pas de codare, imaginea se descompune într-o imagine de joasă frecvență, $f_1(i, j)$, având o rezoluție de două ori mai mică și o imagine de aceeași rezoluție, $h_1(i, j)$, dar care va conține componentele de înaltă frecvență.

$$f_1(i, j) = [f_0 \otimes g](2i, 2j) \text{ și } h_1(i, j) = f_0(i, j) - [f_0 \otimes g](i, j) \quad (3.93)$$

Procesul se iterează la fel pentru fiecare imagine subeșantionată. După n iterații ale unei imagini de dimensiuni $N \times N$, unde $N=2^n$, funcția $f_n(i, j)$ este corespunzătoare unui singur punct. Codarea piramidală a imaginii constă în codarea tuturor $h_k(i, j)$ și a imaginii care conține frecvențele joase, $f_n(i, j)$. Acest algoritm este ilustrat în figura Fig. 3.2-20.

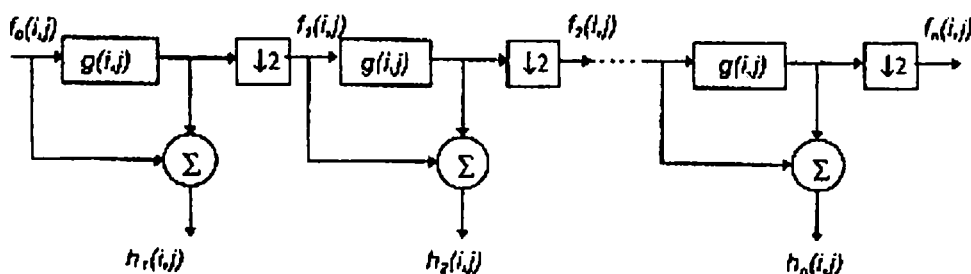


Fig. 3.2-20 Schema de codare piramidală Laplace

Decodarea imaginii este făcută în ordine inversă. Supraeșantionarea („upsampling”) este procesul de inserare de zerouri între eșantioane. Fiecare imagine subeșantionată $f_k(i, j)$, începând cu ultima, $f_n(i, j)$, este supraeșantionată și apoi interpolată prin convoluție cu $g(i, j)$. Rezultatul interpolării este apoi însumat cu imaginea $h_{k-1}(i, j)$ și algoritmul se continuă pe imaginea rezultată. Prin această metodă se poate reconstitui imaginea originală fără erori.

Fiecare imagine $h_k(i, j)$ este diferența a două imagini, obținute prin convoluția aceleiași imagini cu o funcție gaussiană de lărgime dată și, respectiv, de lărgime dublă. Acest lucru este echivalent cu convoluția imaginii cu diferența celor două filtre Gauss, care de fapt aproximează filtrul trece sus „laplacian-gaussian”, de unde și numele acestui algoritm piramidal.

Deși algoritmul de codare piramidală laplacian duce la creșterea numărului de pixeli necesari pentru reprezentarea imaginii cu 33%, se poate obține totuși o compresie

semnificativă, deoarece imaginile componente $h_k(i, j)$ au o corelație și o gamă dinamică mult mai mică și de aceea sunt propice pentru o cuantizare brută și pentru neglijarea unor coeficienți. Modelul piramidal laplacian (Fig. 3.2-21) a stat la baza transformării wavelet discrete.



Fig. 3.2-21 Un exemplu de piramidă laplaciană și reconstruire.

3.3. Concluzii

Esența oricărei metode de selectare a caracteristicilor constă în reținerea informației cu capacitate de discriminare între clase și reducerea informației care prezintă aspecte comune pentru clasele în discuție. În felul acesta, punerea în evidență a acelor proprietăți ale formelor dintr-o clasă care se deosebesc de formele care fac parte dintr-o altă clasă permite menținerea capacității de clasificare într-un spațiu al formelor de o dimensionalitate redusă. În consecință, selecția caracteristicilor poate fi privită ca un proces care constă în reținerea optimă a unui număr minim de dimensiuni (variabile) astfel încât să se mențină și/sau să se maximizeze probabilitatea de clasificare corectă. Din păcate, un asemenea criteriu de fidelitate este dificil de evaluat și de aceea adeseori se apelează la alte criterii care permit o selecție a caracteristicilor optimă dar ușor de mânuit analitic.

Dintre metodele de selectare a caracteristicilor sunt evidențiate metodele Karhunen-Loève, care este considerată transformarea optimă, reținând cea mai mare cantitate de energie, și DCT, care datorită caracteristicilor sale este folosită la compresia de imagini. Aceste metode sunt folosite în modelul HMM pentru obținerea vectorilor de observație.

4. Detectarea și extragerea caracteristicilor feței umane

4.1. Introducere

Un prim pas către recunoașterea feței este localizarea feței ca și regiuni și extragerea caracteristicilor faciale ca ochi, gură și nas. Cu toate că multe lucruri s-au făcut deja în această direcție de cercetare, recunoașterea feței umane în cadrul unor scene cu fundal difuz este încă o problemă care necesită mai departe investigare.

Există multe tipuri de aplicații pentru sistemele de localizarea și recunoaștere, de exemplu modele video bazate pe codare, sisteme de securitate, potrivire de personaje, etc. În contextul unor variații de lumină, fundale, unghiuri vizuale și expresii faciale problema devine complexă. Pentru detecția regiunilor faciale și a caracteristicilor faciale au fost publicate până acum câteva abordări folosind texturi, intensități, forme și informații a culorii sau combinații ale acestora. De exemplu, extragerea regiunilor faciale din fundale complexe este făcută pe baza informației de culoare și de textură. Imaginea de intrare este în primul rând intensificată folosind informația de culoare, apoi caracteristicile texturii sunt obținute pe baza modelului texturii pentru fețe, părțile faciale fiind astfel detectate. În primul pas imagini conținând vederi frontale sunt luate ca mostră de la secvența video de intrare. Punctul de corespondență pe o gamă largă de diferențe este determinat folosind un algoritm multirezoluție ierarhic de potrivire. În final nasul, ochii și gura sunt localizate pe baza intensității informației. Localizarea feței este făcută utilizând informația de formă. O elipsă este aleasă ca și model pentru forma facială și pe baza informației de contur sunt determinate componentele ce se aseamănă cu conturul capului. Înainte de a extrage caracteristicile faciale, imaginea de intrare este segmentată folosind caracteristicile de culoare, apoi sunt determinate punctele caracteristicii și în ultimul pas diferitele caracteristici faciale sunt approximate prin polinoame. Deseori informația de contur (muchii) este folosită pentru extragerea caracteristicilor faciale cu toate că, caracteristicile faciale nu sunt separate în interiorul feței prin contururi proeminente. Combinația caracteristicilor permite o detecție a feței, deoarece fețele sunt caracterizate de culoarea obrajilor și de forma lor ovală. Extragerea caracteristicilor feței este făcută folosind informația nivelului de gri din interiorul regiunilor feței. Pe baza acestei observații putem spune că ochii și gura diferă de restului feței din cauza luminozității lor scăzute.

Scopul final al aplicațiilor de analiză de imagine este extragerea unor caracteristici importante din imagine, care să ajute calculatorul să înțeleagă, să interpreteze o scenă.

Analiza de imagine diferă esențial de alte operații de prelucrare de imagini, cum ar fi refacerea, îmbunătățirea sau codarea imaginii, operații din care în final rezultă tot o imagine. Structura unui sistem general de analiză și prelucrare a imaginii pe calculator este prezentată în figura Fig. 4.1-1.

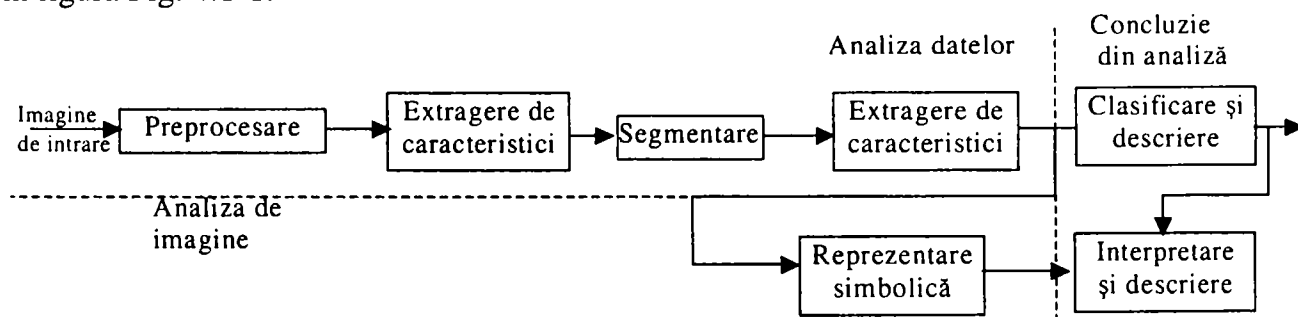


Fig. 4.1-1. Sistem de analiză a imaginilor

Preprocesarea poate să însemne refacere, îmbunătățire sau reprezentare corespunzătoare a imaginii. Extragerea de caracteristici servește pentru segmentarea imaginii,

de exemplu separarea unor obiecte prin recunoașterea conturului. Imaginea segmentată poate fi introdusă într-un sistem de clasificare sau într-un sistem de înțelegere.

Un sistem de înțelegere a imaginii determină relația dintre diferitele obiecte din scenă, în scopul descrierii lor și interpretării globale a scenei.

Extragerea caracteristicilor imaginii presupune separarea caracteristicilor spațiale, a caracteristicilor de tip transformare, separarea muchiilor și conturilor, a caracteristicilor de formă, de textură și a caracteristicilor statistice ale imaginii.

De exemplu, cele mai simple și utile caracteristici spațiale sunt caracteristicile de amplitudine. În imaginea medicală realizată cu raze X, diversele intensități ale nivelului de gri ne informează asupra locului oaselor în țesutul moale și permite identificarea zonelor bolnave și sănătoase.

Caracteristicile histogramei se bazează pe histograma unei regiuni din imagine. Dacă u este o variabilă aleatoare care reprezintă nivelul de gri într-o anumită zonă a imaginii, atunci se definește probabilitatea: $p_u(x) = p[u = x]$, respectiv numărul pixelilor cu nivel de gri "x" raportat la numărul total al pixelilor din zonă, pentru $x = 0, 1, \dots, L-1$. Pentru această variabilă statistică se definesc caracteristici ca entropia, dispersia, abaterea medie pătratică. Acestea se definesc de obicei pe o fereastră mișcătoare de dimensiuni mici, fereastră ce parcurge în timp toată imaginea.

Segmentarea imaginii presupune identificarea modelului, detecția de contur, clasificarea și gruparea obiectelor, etc.

Clasificarea și descrierea obiectelor dintr-o imagine presupune gruparea acestora, statistica și realizarea de arbori decizionali, măsurarea similarității cu imaginea originală, etc.

4.2. Proiectarea filtrelor

În etapa de preprocesare a unui sistem de analiză și prelucrare de imagini o importanță accentuată o au filtrele, care ajută la îmbunătățirea imaginilor. În continuare vom trece în revistă câteva dintre cele mai utilizate tipuri de filtre.

4.2.1. Mediere și filtrare trece-jos

La frecvențe înalte informațiile de interes sunt deseori afectate de zgomot. Astfel, un filtru care reduce amplitudinea componentelor de frecvență înaltă poate reduce vizibil efectele de zgomot. O cale simplă de reducere a zgomotului de frecvență înaltă este prin medierea locală. În acest caz, fiecare pixel este înlocuit cu o medie ponderată a pixelilor din vecinătate:

$$v(m,n) = \sum_{(k,l) \in W} a(k,l) y(m-k, n-l) \quad (4.1)$$

unde $y(m,n)$ și $v(m,n)$ sunt imaginile de intrare, respectiv ieșire, w este o fereastră aleasă corespunzător, iar $a(k,l)$ reprezintă ponderarea filtrării. O clasă comună de filtre spațiale mediane este cea la care ponderile sunt aceleași.

$$v(m,n) = \frac{1}{N_w} \sum_{(k,l) \in W} y(m-k, n-l) \quad (4.2)$$

unde $a(k,l) = \frac{1}{N_w}$, iar N_w reprezintă numărul pixelilor din fereastra W . O altă filtrare spațială mediană, utilizată frecvent, este dată de transformarea:

$$v(m,n) = \frac{1}{2} \left[y(m,n) + \frac{1}{4} \{ y(m-1,n) + y(m+1,n) + y(m,n-1) + y(m,n+1) \} \right] \quad (4.3)$$

Se observă că, în acest caz, fiecare pixel este înlocuit cu media sumei dintre valoarea lui și media a patru pixeli din vecinătatea sa. Ferestrele utilizate pot fi de dimensiuni mai mici sau mai mari, în funcție de aplicație și de performanțele necesare. Este evident că folosirea

unor ferestre de dimensiuni mai mari duce la creșterea performanțelor, în detrimentul, însă, al timpului de calcul și al capacității de memorie necesare.

Medierea spațială este utilizată pentru atenuarea zgomotului, filtrarea trece-jos și subeșantionarea imaginilor. Presupunem o imagine dată de funcția:

$$v(m,n) = u(m,n) + \eta(m,n) \quad (4.4)$$

unde $\eta(m,n)$ este zgomotul alb, cu valoarea medie nulă și varianța σ^2_η . Media spațială a acestei imagini va fi:

$$v(m,n) = \frac{1}{N_w} \sum_{(k,l) \in W} u(m-k, n-l) + \bar{\eta}(m,n) \quad (4.5)$$

unde $\bar{\eta}(m,n)$ este media spațială a lui $\eta(m,n)$. Se poate demonstra relativ simplu că

$\bar{h}(m,n)$ este nulă și varianța $\overline{\sigma^2_\eta} = \frac{\sigma^2_\eta}{N_w}$, ceea ce înseamnă că puterea de zgomot se reduce cu

un factor egal cu numărul n de pixeli ai ferestrei W . Medierea spațială aplicată unei imagini fără zgomot $u(m,n)$, constantă într-o fereastră W , va determina o îmbunătățire a raportului semnal/zgomot de N_w ori. În practică, mărimea ferestrei W este limitată, datorită faptului că $u(m,n)$ nu este de obicei constantă, astfel că medierea spațială introduce distorsiuni de genul micșorării clarității imaginii.

Filtrarea spațială direcțională: Metoda de filtrare spațială direcțională se folosește în vederea protejării conturilor care se atenuează în procesul de mediere spațială. Mediile spațiale $v(m,n:\theta)$ se calculează în diferite direcții cu formula:

$$v(m,n:\theta) = \frac{1}{N_\theta} \sum_{(k,l) \in W_\theta} y(m-k, n-l) \quad (4.6)$$

Se poate găsi o direcție, de-a lungul căreia valoarea $|y(m,n) - v(m,n:\theta^*)|$ să fie minimă. În acest caz, valoarea $v(m,n) = v(m,n:\theta_0)$ dă rezultatul dorit. Metoda este exemplificată în figura de jos (Fig. 4.2-1).

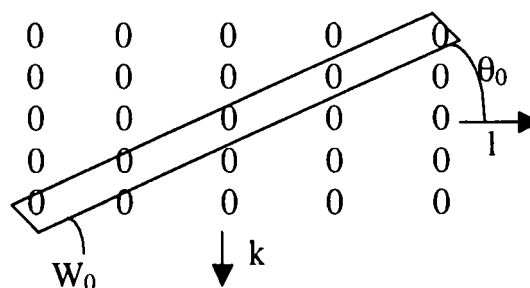


Fig. 4.2-1 Filtrare spațială direcțională

Filtrarea mediană: pixelul din imaginea de intrare se înlocuiește cu media pixelilor conținuți într-o fereastră din jurul pixelului:

$$v(m,n) = \text{media}\{y(m-k, n-l), (k,l) \in W\} \quad (4.7)$$

unde W este o fereastră aleasă corespunzător. Algoritmul de filtrare mediană necesită așezarea valorilor pixelilor din fereastra W în ordine crescătoare sau descrescătoare și determinarea valorii medii. În general, fereastra se alege astfel ca N_w să fie impar. În cazul când N_w este par, media se determină ca media celor două valori din mijlocul șirului ordonat. Ferestrele tipice utilizate au dimensiuni de 3×3 , 5×5 , 7×7 .

Filtrul median are următoarele proprietăți:

- Este un filtru neliniar : pentru două secvențe $x(m)$ și $y(m)$, media $\{x(m,n) + y(m,n)\} \neq \text{media}\{x(m,n)\} + \text{media}\{y(m,n)\}$

- Este util pentru înlăturarea liniilor sau pixelilor izolați, cu păstrarea rezoluției spațiale
- Are performanțe slabe când numărul pixelilor afectați de zgomot este mai mare sau egal cu jumătate din numărul total al pixelilor din fereastră.

Alternative pentru filtrarea mediană constau în înlocuirea zgomotului izolat prin găsirea unei medii spațiale după formula (4.6) și înlocuirea pixelului cu această valoare medie dacă valoarea zgomotului este mai mare, adică dacă " $v(m,n) - y(m,n)$ " este mai mare decât o valoare prestabilită.

Pentru zgomot de tip gaussian aditiv se folosesc algoritmi ce utilizează proprietățile statistice ale imaginii și zgomotului. Există, de asemenea, algoritmi adaptivi ce ajustează răspunsul filtrului în concordanță cu variațiile locale ale proprietăților statistice

4.2.1.1. Accentuarea conturilor

Operația de accentuare de contur constă în extragerea din imaginea originală a unui semnal proporțional cu imaginea filtrată trece-jos. Aceasta este echivalent cu adunarea unui semnal trecut printr-un filtru trece-sus la imagine. În general, accentuarea de contur poate fi exprimată matematic printr-o relație de forma:

$$v(m,n) = u(m,n) + \lambda g(m,n) \quad (4.8)$$

unde $\lambda \geq 0$ și $g(m,n)$ este gradientul definit în mod convenabil al funcției $u(m,n)$. O funcție gradient uzuală este laplacianul discret:

$$g(m,n) = u(m,n) - \frac{1}{4} [u(m-1,n) + u(m+1,n) + u(m,n-1) + u(m,n+1)] \quad (4.9)$$

Din punct de vedere al prelucrărilor pe care le suferă semnalul, figura Fig. 4.2-2 prezintă succesiunea lor și efectul final asupra imaginii.

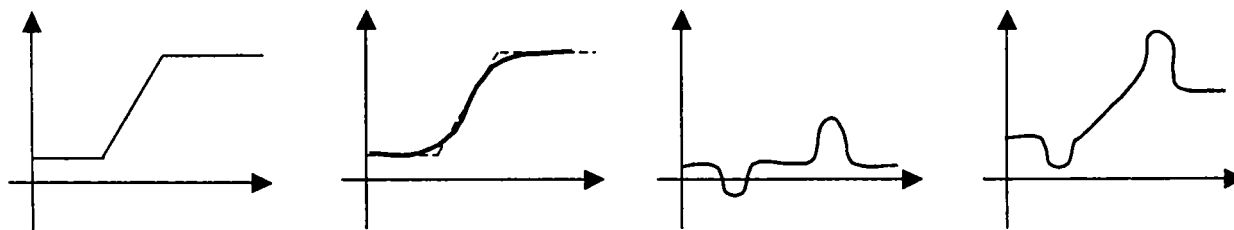


Fig. 4.2-2. Algoritm de accentuare de contur

Filtrarea de tip trece-jos se utilizează pentru atenuarea zgomotului și pentru interpolare, iar filtrarea de tip trece-sus se folosește pentru extragerea sau accentuarea conturilor.

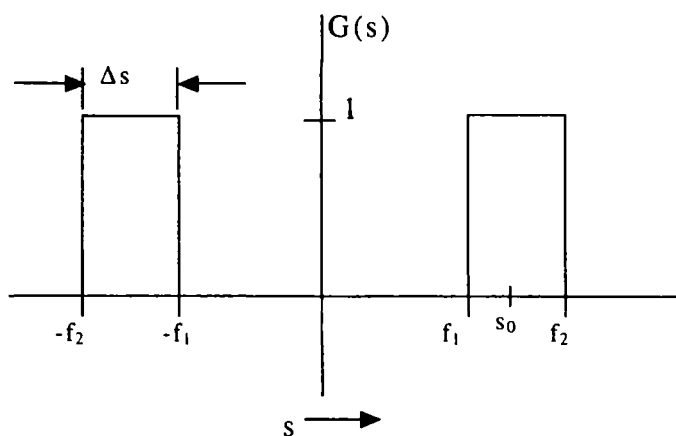
4.2.2. Filtrare trece-bandă și oprește-bandă

4.2.2.1. Filtru trece-bandă

Presupunem că dorim să implementăm, prin convoluție un filtru care trece energia numai la frecvențe cuprinse între f_1 și f_2 , unde $f_2 > f_1$. Funcția de transfer este dată de:

$$G(s) = \begin{cases} 1, & f_1 \leq |s| \leq f_2 \\ 0, & \text{altfel} \end{cases} \quad (4.10)$$

Deoarece $G(s)$ este o pereche de pulsuri dreptunghiulare ca în figura Fig. 4.2-3, el poate fi considerat ca și un puls convolvat cu un impuls par regulat.


Fig. 4.2-3. Funcția de transfer trece-bandă

Dacă considerăm:

$$s_0 = \frac{1}{2}(f_1 + f_2) \text{ și } \Delta s = f_2 - f_1 \quad (4.11)$$

putem scrie funcția de transfer al filtrului oprește-bandă:

$$G(s) = \prod \left(\frac{s}{\Delta s} \right) * [\delta(s - s_0) + \delta(s + s_0)] \quad (4.12)$$

Cu expresia funcției de transfer în această formă, noi putem scrie ușor răspunsul impulsului astfel:

$$g(t) = \Delta s \frac{\sin(\pi \Delta s t)}{\pi \Delta s t} 2 \cos(2\pi s_0 t) = 2 \Delta s \frac{\sin(\pi \Delta s t)}{\pi \Delta s t} \cos(2\pi s_0 t) \quad (4.13)$$

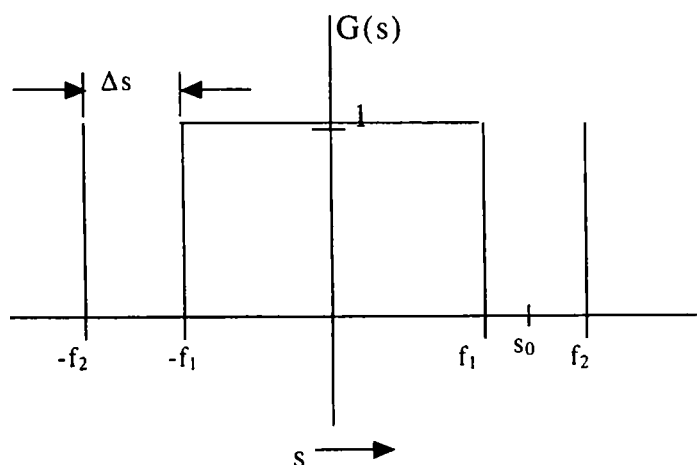
Deoarece $\Delta s < s_0$, ecuația descrie un cosinus de frecvență s_0 închis într-o anvelopă $\sin(x)/x$ având frecvența $\Delta s / 2$. Numărul de cicluri cosinus depinde de relația dintre s_0 și Δs .

4.2.2.2. Filtru oprește-bandă

Funcția de transfer care trece energia la toate frecvențele exceptând pentru o bandă dintre f_1 și f_2 este dată de:

$$G(s) = \begin{cases} 0, & f_1 \leq |s| \leq f_2 \\ 1, & \text{altfel} \end{cases} \quad (4.14)$$

Graficul funcției este reprezentat în figura Fig. 4.2-4.


Fig. 4.2-4. Funcția de transfer oprește-bandă

Considerăm s_0 frecvența de centru și Δs lățimea benzii date de ecuația (4.11). Putem scrie astfel funcția de transfer ca unu minus filtru trece-jos:

$$G(s) = 1 - \prod \left(\frac{s}{\Delta s} \right) * [\delta(s - s_0) + \delta(s + s_0)] \quad (4.15)$$

Răspunsul impulsului pentru funcția de transfer este:

$$g(t) = \delta(t) - 2\Delta s \frac{\sin(\pi\Delta s t)}{\pi\Delta s t} \cos(2\pi s_0 t) \quad (4.16)$$

Filtrarea de tip trece-bandă este utilă pentru îmbunătățirea contururilor sau altor caracteristici de tip trece-sus ale unei imagini, în prezența zgomotului.

4.2.3. Filtrare liniară generalizată

Operațiunile punctuale pe imaginea transformată sunt produse de forma:

$$v'(k, l) = g(k, l) * v(k, l) \quad (4.17)$$

unde $g(k, l)$ se numește *mască zonală* (este zero în afara regiunii respective). Figura Fig. 4.2-5.a prezintă măștile zonale corespunzătoare filtrării trece-jos, trece-sus și trece-bandă pentru o transformare Fourier digitală, iar figura Fig. 4.2-5.b prezintă aceleași măști zonale pentru alte tipuri de transformări ortogonale.

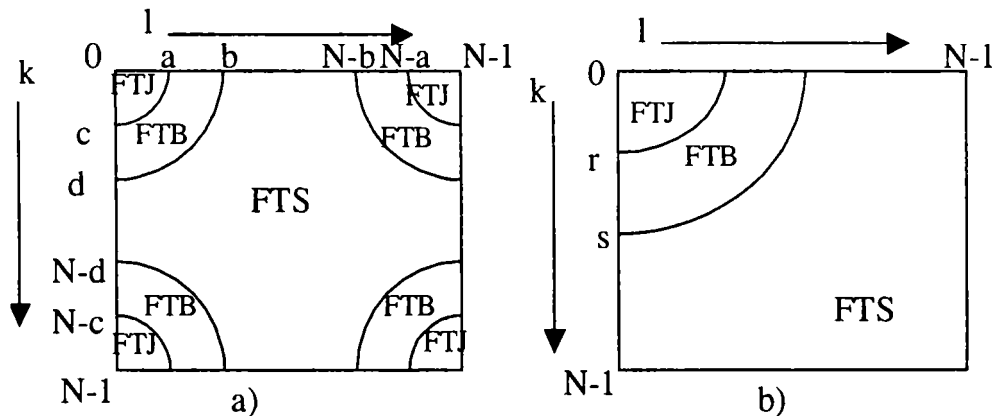


Fig. 4.2-5. Măști zonale pentru filtrarea liniară generalizată

Un filtru de interes particular este filtrul gaussian invers, a cărui mască zonală pentru o imagine A de dimensiuni $N \times N$ este definită ca:

$$g(k, l) = \begin{cases} \exp \frac{k^2 + l^2}{2\sigma^2}, & a \leq k, l \leq N/2 \\ g(N - k, N - l), & \text{in rest} \end{cases} \quad (4.18)$$

unde imaginea A este transformată prin transformare Fourier digitală. Pentru alte transformări ortogonale:

$$g(k, l) = \exp \frac{(k^2 + l^2)}{2\sigma^2}, \quad 0 \leq k, l \leq N - 1 \quad (4.19)$$

Aceste filtrări sunt de tip trece-sus, și sunt utilizate pentru refacerea imaginilor încetoșate, (de turbulențe atmosferice, spre exemplu) sau de alte fenomene ce pot fi modelate analog.

4.2.4. Filtrare neliniară

În acest caz, coeficienții transformării $v(k, l)$ pot fi scriși ca:

$$v(k, l) = |v(k, l)| e^{j\theta(k, l)} \quad (4.20)$$

Pentru acest tip de filtrare se ia rădăcina de ordinul alfa a mărimii componentei $v(k, l)$, în timp ce faza se reține ca atare:

$$v'(k, l) = |v(k, l)|^a e^{j\theta(k, l)} \quad 0 \leq a \leq 1 \quad (4.21)$$

Pentru imagini obișnuite, dat fiind faptul că mărimea lui $v(k, l)$ este mai mică la frecvențe spațiale mari, efectul este de îmbunătățire a frecvențelor spațiale înalte, comparativ cu frecvențele spațiale joase.

4.3. Recunoașterea feței pe bază de caracteristici

În general recunoașterea feței pe bază de caracteristici poate fi descrisă după cum urmează: ca intrare sunt disponibile imagini sau secvențe de imagini și o bază de date cu fețe. În primul pas al procesului de recunoaștere a feței, regiunile feței sunt segmentate afară din scenă. Caracteristicile feței sunt apoi extrase din regiunile feței, urmând apoi o identificare prin potrivirea caracteristicilor extrase cu caracteristicile din baza de date. Ca rezultat al identificării sunt obținute una sau mai multe persoane (Fig. 4.3-1).

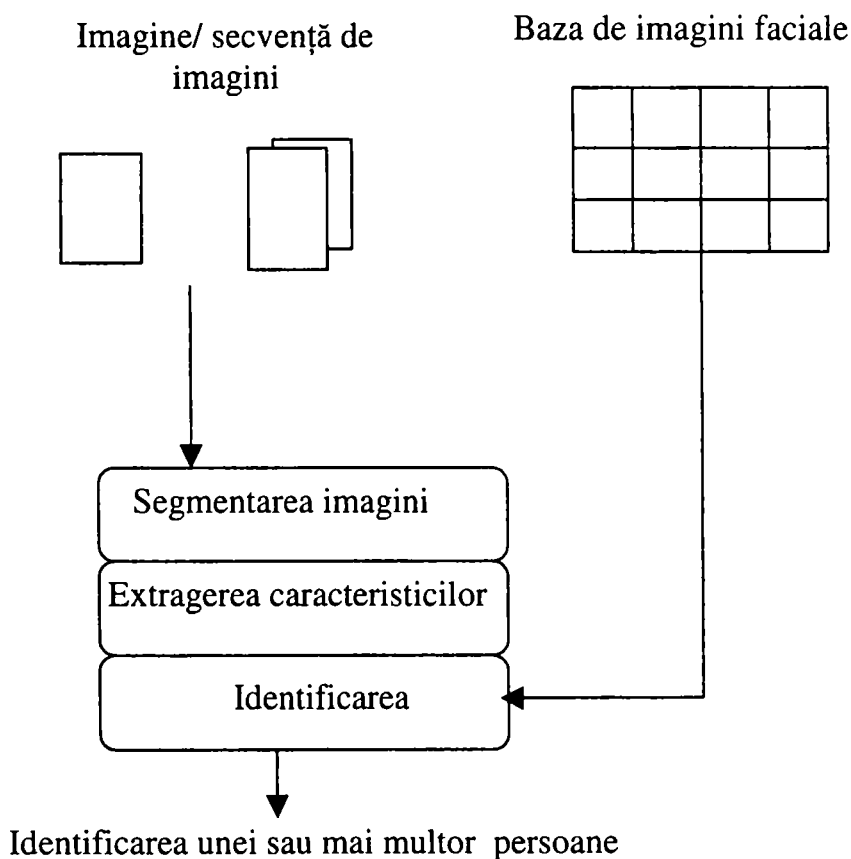


Fig. 4.3-1 Problema recunoașterii feței

Desigur ordinea celor doi primi pași poate fi interschimbată. În general prima dată se segmentează regiunile feței și apoi sunt extrase caracteristicile din aceste regiuni. În cazul imaginilor color această ordine de procesare permite o analiză solidă deoarece fețele diferă mult de fundal prin culoarea și forma lor.

4.4. Segmentarea imaginilor

Putem defini procesul de segmentare a imaginii ca unul care împarte o imagine digitală în regiuni disjuncte (despărțite). În acest scop, o regiune este un set de pixeli conectați - ceea ce înseamnă că este un set în care toți pixeli sunt adiacenți sau se ating. Definiția formală a conexiunii este următoarea: între oricare doi pixeli aflați într-un set conectat, există o cale de conexiune aflată în întregime în interiorul setului, unde o cale de conexiune este o cale care se mișcă întotdeauna între pixelii aflați în vecinătate (sau vecini). Deci într-un set de conexiuni se poate trasa o cale de conexiune între oricare doi pixeli fără a fi nevoiți să păstrăm setul.

Există două reguli de conectivitate și oricare din cele două poate fi aplicată. Dacă doar pixeli adiacenți lateral (sus, jos, stânga, dreapta) sunt considerați a fi conectați, este vorba de o cvartă-conectivitate și în acest caz obiectele sunt cvarto-conectate. În consecință, fiecare pixel are doar patru vecini la care poate fi conectat. Dacă, în completare, pixeli adiacenți pe diagonală (la 45 de grade) sunt și ei considerați a fi conectați, este vorba de o octo-conectivitate, iar obiectele sunt octo-conectate. Oricare regulă de conectivitate poate fi folosită, atâta timp cât este consistentă. Deseori rezultatele octo-conectivității se apropie mai mult de intuiția noastră.

Când un observator uman urmărește o scenă, procesul care are loc în sistemul vizual segmentează în mod esențial scena ei sau a lui. Acest lucru are loc în așa fel încât efectiv acesta nu urmărește o scenă complexă, ci mai degrabă ceva ce pare a fi o colecție de obiecte. Totuși în cazul procesului digital, trebuie să izolăm laborios obiectele dintr-o imagine, complementând imaginea respectivă într-un set de pixeli, în care fiecare din aceștia reprezintă un obiect. În timp ce sarcina de a segmenta o imagine nu prea are o pondere mare în experiența vizuală umană, este o sarcină netrivială în analiza digitală a imaginii.

Segmentarea imaginii poate fi privită din trei perspective filosofice diferite. În cazul în care este vorba de aproximarea regiunilor, fiecare pixel este asociat unui obiect sau unei regiuni particulare. În cazul aproximării marginale, se încearcă localizarea legăturilor care există între regiuni. În aproximarea limită, se caută identificarea și legarea pixelilor limită pentru a forma marginea cerută. Toate cele trei aproximări sunt utile în vizualizarea problemei.

În acest subcapitol vom trece în revistă cele mai importante tehnici de segmentare.

4.4.1. Segmentarea prin delimitarea cu prag (fereastră)

Delimitarea cu prag este utilă atunci când caracteristicile de amplitudine sunt suficiente pentru a individualiza un obiect sau o regiune din imagine. Valorile caracteristicilor de amplitudine alese se calibrează astfel încât un interval de amplitudini să fie reprezentativ pentru un singur obiect din imagine. Tehnicile de segmentare prin delimitare se folosesc în cazul imaginilor binare (documente tipărite, desene și grafice) sau în cazul imaginilor color și multispectrale.

Alegerea pragului sau pragurilor de delimitare este cel mai important pas din această metodă și se realizează în felul următor:

- Se examinează histograma imaginii, pentru localizarea vârfurilor și adânciturilor acesteia, sau
- Se face delimitarea adaptivă, prin examinarea histogramelor corespunzătoare vecinătăților locale.

4.4.2. Segmentarea prin extragerea conturului

Tehnicile de extragere a conturului realizează o segmentare a obiectelor pe baza profilului acestora. Astfel, prin urmărirea conturului, conectivitate și alte tehnici se poate face o segmentare a obiectelor dintr-o imagine.

Dificultățile apar în cazul în care obiectele se ating sau se suprapun parțial, sau în cazul în care conturul nu este complet închis, datorită zgomotului sau distorsiunilor imaginii. Figura Fig. 4.4-1 prezintă un sistem de recunoaștere a formei (obiectelor) folosind informația de contur pentru segmentarea imaginii.

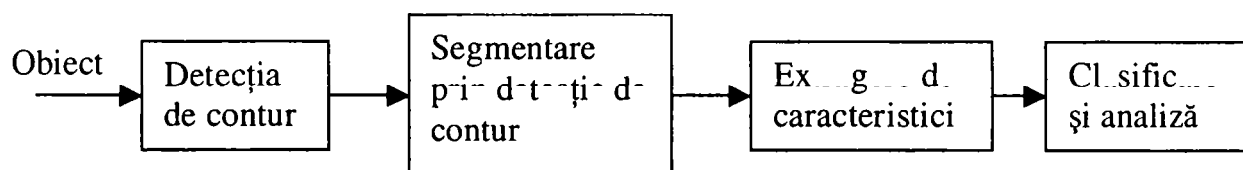


Fig. 4.4-1 Segmentare bazată pe extragerea conturului

4.4.2.1. Detecția de contur

Muchiile caracterizează contururile obiectelor și sunt, din această cauză utile pentru segmentarea, recunoașterea sau clasificarea obiectelor dintr-o imagine. Muchiile pot fi imaginate ca locații de pixeli cu salturi mari ale nivelului de gri. În imagini binare (pixeli albi și negri) muchiile se definesc ca pixeli negri cu cel puțin un pixel alb în vecinătate:

$$g(m,n) = [u(m,n) \oplus u(m \pm 1, n) \text{ OR } [u(m,n) \oplus u(m, n \pm 1)]] \quad (4.22)$$

unde prin \oplus s-a notat operatorul "SAU-EXCLUSIV".

Pentru o imagine continuă $f(x,y)$, muchiile sunt sesizabile prin maximul derivatei (gradientului funcției) de-a lungul direcției muchiei. De aceea, o tehnică de detecție a conturilor constă în măsurarea gradientului lui f de-a lungul unei direcții θ . Pe baza acestui concept, se pot introduce două tipuri de operatori de detecție: operatori gradient și operatori unghiulari.

Pentru imagini digitale, acești operatori, numiți și "măști", reprezintă aproximări cu diferențe finite ale gradientelor ortogonali f_x , f_y sau a unui gradient după o direcție oarecare r , ∂_f / ∂_r . Dacă notăm cu H o mască de dimensiuni $p \times p$ și definim pentru o imagine arbitrară u produsul intern pentru locația (m, n) ca fiind corelația:

$$(u, H)_{m,n} = \sum_i \sum_j h(i, j) u(i + m, j + n) = u(m, n) \oplus h(-m, -n) \quad (4.23)$$

Calculul se efectuează pe două direcții ortogonale. Definind gradientii pe cele două direcții ca fiind $g_1(m,n) = (u, H_1)_{m,n}$ și $g_2(m,n) = (u, H_2)_{m,n}$ atunci amplitudinea vectorului gradient și direcția acestuia sunt date de:

$$g(m,n) = \sqrt{g_1^2(m,n) + g_2^2(m,n)} \quad (4.24)$$

$$\theta_g(m,n) = \arctg \frac{g_2(m,n)}{g_1(m,n)} \quad (4.25)$$

și semnifică direcția și amplitudinea relativă a muchiei față de imaginea din jur.

Modul de detectare a muchiiilor cu ajutorul operatorilor gradient este prezentat în figura Fig. 4.4-2

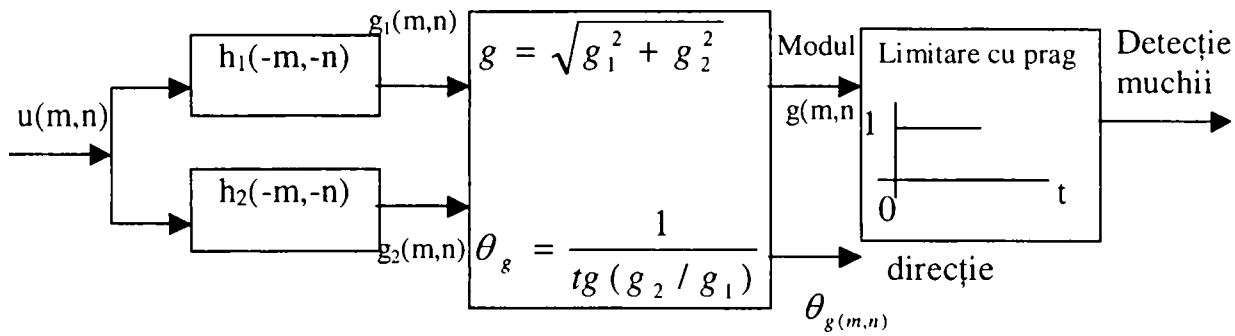


Fig. 4.4-2 Deteția muchiilor cu operatori gradient

Un exemplu de operator gradient, utilizat în deteția de contur, este operatorul Roberts:

$$H_1 = \begin{bmatrix} \underline{0} & 1 \\ -1 & \underline{0} \end{bmatrix}, \text{ respectiv } H_2 = \begin{bmatrix} 1 & \underline{0} \\ \underline{0} & -1 \end{bmatrix} \quad (4.26)$$

Operatorul Sobel are structura dată de relația :

$$H_1 = \begin{bmatrix} -1 & \underline{0} & 1 \\ -1 & \underline{0} & 1 \\ -1 & \underline{0} & 1 \end{bmatrix}, \text{ respectiv } H_2 = \begin{bmatrix} -1 & -2 & -1 \\ 0 & \underline{0} & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad (4.27)$$

Elementul subliniat în matricile H_1 și H_2 este considerat originea operatorului.

Un exemplu fiind dat în figura Fig. 4.4-3



Fig. 4.4-3 Detețarea de contur folosind operatorul Sobel

În afara operatorilor gradient se folosesc și așa-numiții operatori unghiulari, care măsoară gradientii într-un număr selectat de direcții. Câteva exemple de operatori unghiulari mai des utilizați în deteția de contur sunt prezentați în figura Fig. 4.4-4 (operatorii sunt ilustrați pentru direcția nord, ceilalți operatori corespunzători altor direcții putând fi obținuți din aceștia, prin rotirea elementelor din matrice, în sensul acelor de ceasornic, cu 90, 180, și respectiv 270 grade).

$$\begin{matrix} \begin{bmatrix} 1 & 1 & 1 \\ 1 & -2 & 1 \\ -1 & -1 & -1 \end{bmatrix} & \begin{bmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix} & \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} \\ \text{a} & \text{b} & \text{c} \end{matrix}$$

Fig. 4.4-4 Operatori unghiulari

Când zona de muchie devine mai lată (muchie mai puțin pronunțată) este avantajoasă folosirea derivatei de ordinul doi. Pentru aceasta se folosesc operatori Laplace sau de trecere prin zero. În prezența zgomotului se folosesc gradienti statistici, care echivalează cu o filtrare cu caracteristică de tip trece-jos.

4.4.2.2. Operatori Laplace și metoda trecerilor prin zero

Metodele prezentate mai sus sunt utile mai ales pentru tranziții bruște ale nivelelor de gri, ca în cazul funcțiilor – treaptă (salturi alb-negru sau invers). Dacă regiunea de tranziție este mai largă (saltul este mai lin), este mai avantajos să se folosească derivate de ordinul doi în locul gradientilor. Un operator des întâlnit este operatorul Laplace, definit ca:

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \quad (4.28)$$

Operatorul Laplace poate fi aproximat, în variantă discretă, sub forma unor matrici centrate în jurul elementului asupra căruia se aplică:

$$(1) \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \quad (2) \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \quad (3) \begin{bmatrix} 1 & -2 & 1 \\ -1 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix}$$

Un exemplu fiind dat în figura următoare (Fig. 4.4-5):



Fig. 4.4-5 Operator Laplacian 5x5

Datorită derivatelor de ordinul doi, acest operator este mai sensibil la zgomot decât cei definiți anterior. De asemenea, la extragerea conturului prin limitare cu prag se vor obține contururi duble. Din aceste motive, și totodată, deoarece nu este capabil să detecteze direcția conturilor, operatorul Laplace ca atare nu este foarte des utilizat. O modalitate mai bună o constituie folosirea trecerilor prin zero pentru detecția locațiilor conturului sau muchiei. Un operator laplacian generalizat, care aproximează laplacianul funcțiilor gaussiene (LOG) este foarte potrivit pentru detecția trecerilor prin zero. El se definește ca:

$$h(m,n) = c \left[1 - \frac{(m^2 + n^2)}{\sigma^2} \right] \exp\left(-\frac{m^2 + n^2}{2\sigma^2} \right) \quad (4.29)$$

unde σ controlează lățimea funcției gaussiene iar c realizează o normalizare a sumei elementelor măștii date la valoarea unitară. Trecerile prin zero ale unei imagini, convoluționate cu $h(m,n)$, vor da locațiile muchiilor.

Operatorul $h(m,n)$ reprezintă răspunsul eșantionat la impuls al unui filtru analogic trece-bandă, a cărui răspuns în frecvență este proporțional cu:

$$(\xi_1^2 + \xi_2^2) \exp[-2\sigma^2(\xi_1^2 + \xi_2^2)] \quad (4.30)$$

Din această cauză, detectorul de treceri prin zero este echivalent cu un filtru trece-jos având un răspuns la impuls de tip gaussian, urmat de un operator Laplace. Filtru trece-jos este

util pentru atenuarea sensibilității la zgomot a operatorului Laplacian, iar parametrul σ controlează răspunsul în amplitudine al filtrului, dar nu afectează locația trecerii prin zero.

O informație utilă referitoare la direcția muchiilor se poate obține căutând trecerile prin zero ale derivatei de ordinul doi de-a lungul unei drepte r , pentru fiecare direcție posibilă:

$$\frac{\partial^2 f}{\partial r^2} = \frac{\partial f_x}{\partial r} \cdot \cos \theta + \frac{\partial f_y}{\partial r} \cdot \sin \theta = \frac{\partial^2 f}{\partial x^2} \cdot \cos^2 \theta + 2 \frac{\partial^2 f}{\partial x \partial y} \cdot \sin \theta \cos \theta + \frac{\partial^2 f}{\partial y^2} \cdot \sin^2 \theta \quad (4.31)$$

4.4.2.3. Extragerea conturilor

Conturile sunt muchii legate împreună și caracterizează forma unui obiect. Ele sunt utile în prelucrarea unor caracteristici geometrice, cum sunt mărimea și orientarea obiectului. O noțiune utilizată în acest context este cea de **conectivitate**. Despre un pixel se spune că are patru sau opt conexiuni dacă în vecinătatea sa se găsesc patru sau opt pixeli cu aceleași proprietăți (Fig. 4.4-6.). Această modalitate de definire a conectivității determină apariția unor dificultăți de implementare, precum și apariția unor situații paradoxale. Spre exemplu, în figura Fig. 4.4-6. c, în cazul conexiunii cu patru pixeli, segmentele 1, 2, 3 și 4 ar fi clasificate ca fiind disjunctie, cu toate că ele sunt percepute ca formând un inel. Dacă se folosește conectivitatea cu 8 pixeli, segmentele sunt legate între ele, dar “**gaura**” din interior (spre exemplu pixelul notat cu B) este, de asemenea, conectată cu exteriorul (pixelului C).

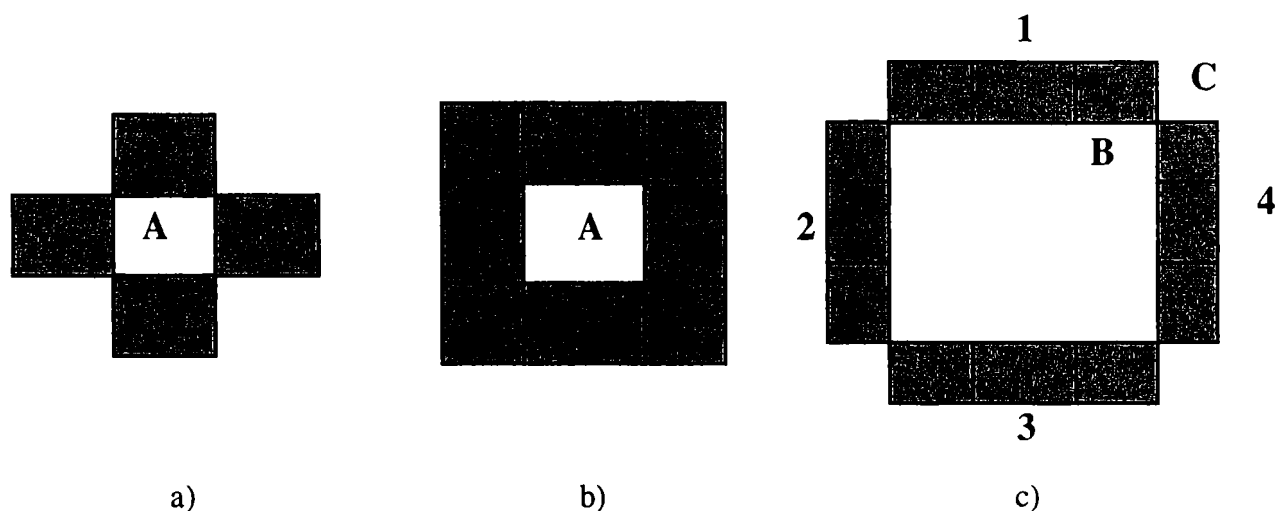


Fig. 4.4-6. Conectivitate cu 4 sau 8 conexiuni

Pentru evitarea situațiilor paradoxale pot fi utilizate grile cu elemente triunghiulare sau hexagonale în locul celor pătrate, grile pe care pot fi definiți pixeli cu trei, respectiv șase conexiuni.

Algoritmul de **urmărire a conturului** trasează conturul prin ordonarea și unirea punctelor succesive de pe o muchie. Testarea apartenenței la muchie poate fi făcută prin testul numărului de legături, de exemplu prin selectarea pixelilor cu opt legături. Alte metode de urmărire a conturului pot fi parcurgerea grafului format din elemente de muchie prin **transformata Hough** care convertește linii drepte aflate la o distanță s de origine și având o orientare dată de ϕ ca:

$$s = x \cdot \sin \phi + y \cos \phi \quad (4.32)$$

într-un punct de coordonate (s, ϕ) și care permite detectarea colțurilor unor obiecte cu contur format din linii drepte.

4.4.3. Segmentarea prin extragerea regiunilor

Ideea de bază în acest tip de segmentare este identificarea diferitelor regiuni din imagine având caracteristici similare. Metoda, numită "clustering" constă în identificarea grupurilor naturale în spațiul caracteristicilor ("cluster") Un "cluster" este un set de puncte din spațiul caracteristicilor, având densitatea locală mare (un maximum relativ) în comparație cu densitatea caracteristicii respective în regiunile înconjurătoare.

O categorie de tehnici de segmentare prin extragerea regiunilor se bazează pe contopirea regiunilor. Imaginea este divizată în regiuni foarte mici având nivele de gri constante. Regiunile adiacente similare se contopesc secvențial, până când regiunile adiacente devin suficient de diferite între ele. Secretul constă în selectarea criteriului de contopire. Un posibil algoritm de contopire conține următorii pași:

- Se contopesc două regiuni R_1 și R_2 , dacă $w/P_m > \theta_1$, unde $P_m = \min(P_i, P_j)$ iar P_i și P_j sunt perimetrele lui R_1 și R_2 , iar w este numărul de locații de contur "slabe" (pixelii de pe fiecare parte a conturului au diferențe de nivele de gri mai mici decât o valoare aleasă de prag σ). Parametrul θ_1 controlează mărimea regiunii ce va fi contopită. De exemplu, pentru $\theta_1=1$, două regiuni se vor contopi numai dacă una dintre regiuni o înconjoară pe cealaltă aproape în totalitate. O valoare tipică este $\theta_1=0,5$.
- Se contopesc R_1 și R_2 dacă $w/I > \theta_2$, unde I este lungimea conturului comun celor două regiuni; tipic $\theta_2=0,75$. Două regiuni se vor contopi dacă conturul comun este suficient de „slab”. Adeseori acest pas se aplică după ce primul pas a redus semnificativ numărul de regiuni.
- Se contopesc R_1 și R_2 numai dacă există puncte de contur „puternice” între ele. Se impune observația că metoda conectivității curselor în cazul imaginilor binare poate fi interpretată ca un astfel de algoritm.

Figura Fig. 4.4-7 prezintă un exemplu de contopire a regiunilor.

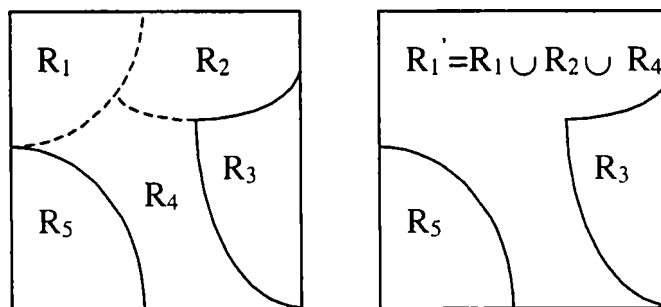


Fig. 4.4-7. Contopirea regiunilor

În locul metodei de contopire a regiunilor se poate face o segmentare prin divizarea unei regiuni date. De exemplu, o imagine poate fi împărțită în regiuni prin metoda arborilor, după care regiunile se contopesc (Fig. 4.4-8, Fig. 4.4-9, Fig. 4.4-10, Fig. 4.4-11, Fig. 4.4-12).

Metodele de segmentare prin extragerea regiunilor sunt, în general, mai puțin sensibile la zgomot decât cele bazate pe extragerea de contur, dar complexitatea algoritmilor este, în general mai mare.

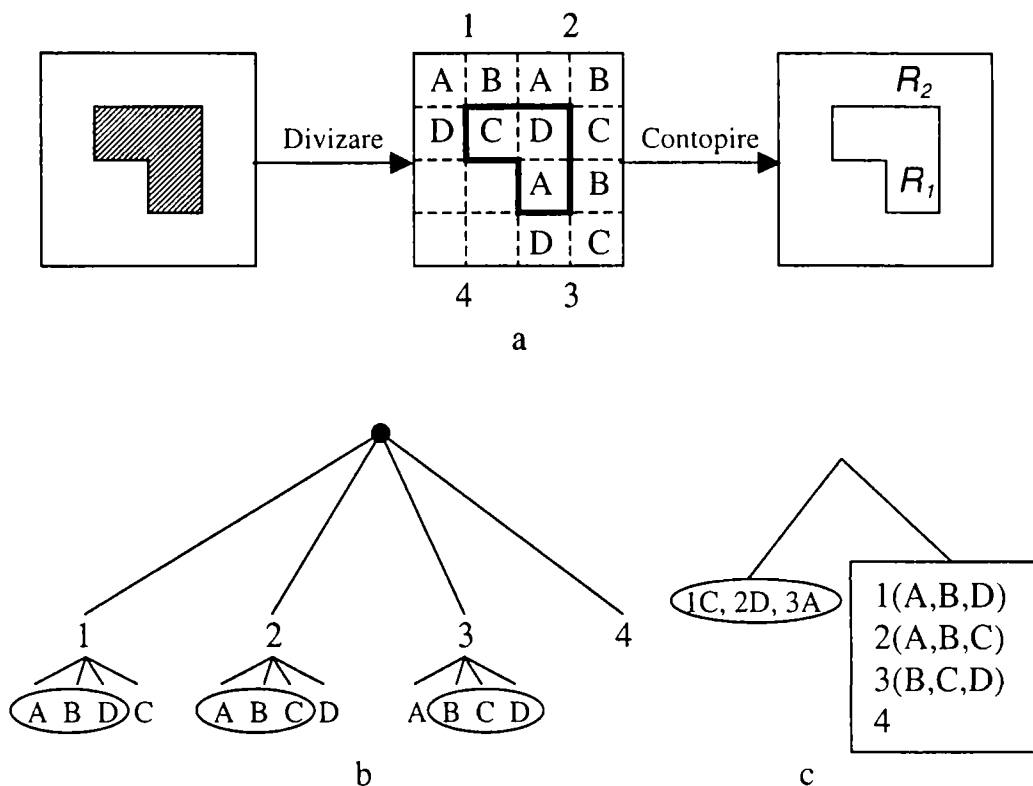


Fig. 4.4-8 Segmentare prin divizare și contopire:

a. intrare b. divizare cu arbori c. regiunile segmentate



Fig. 4.4-9 Imaginea de intrare

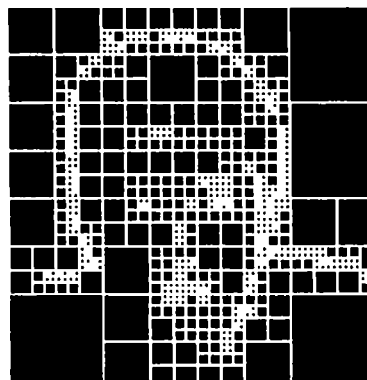


Fig. 4.4-11 Divizare cu arbori



Fig. 4.4-10 Regiunile segmentate

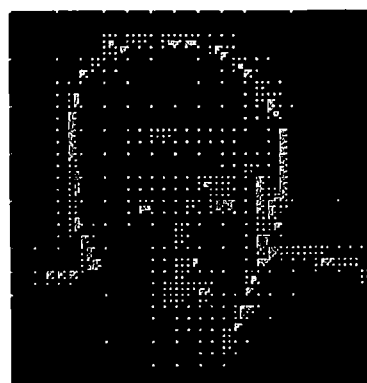


Fig. 4.4-12 Poziții inițiale

4.4.4. Segmentarea pe baza informației de culoare

Fețele sunt semnificativ caracterizate de forma lor ovală și de o culoare specifică a pielii. Pentru acest motiv se segmentează regiunile feței pe bază de formă și informație de culoare.

În special componentele primare Roșu, Verde și Albastru sunt utilizate pentru segmentare. Se consideră Hue-Saturation-Value (HSV) spațiul de culoare pentru extragerea regiunilor de culoare ale obrazului, deoarece acesta este compatibil cu percepția umană de culoare. Alternativ spațiul de culoare Hue-Saturation-Intensity (HSI) poate fi folosit la fel de bine deoarece este foarte similar cu HSV.

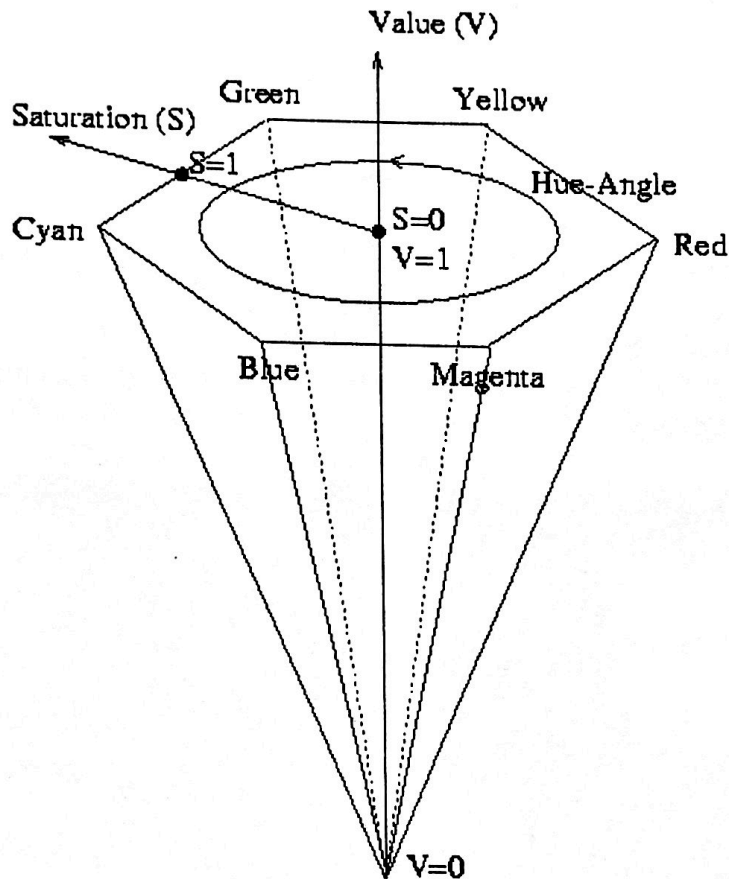


Fig. 4.4-13. Spațiul de culoare Hue-Saturation-Value

Spațiul de culoare HSV are formă de hexagon și este ilustrat în figura Fig. 4.4-13. Verticalele din nivelul de sus reprezintă Rosu, Galben, Verde, Turcoaz, Albastru și Indigo. Verticalele învecinate sunt separate printr-un unghi de 60 de grade. Culoarele complementare sunt situate opus una față de altele. Saturația (S) definește puritatea culorilor și variază de la 0 la 1. Ea crește cu unghiul hexagonului în interiorul nivelului hexagonului. Nuanța închisă a culorii este specificată de valoarea componentei (V), care variază de la 0 la 1. Baza hexagonului este definită de valoarea $V=0$ și la vârful de valoarea $V=1$. În caz că $S=0$, valoarea componentei descrie valorile nivelului de gri, cu negru la bază și cu alb în vârful (nivelul de sus). Pentru segmentarea obrazului în regiuni este suficient să considerăm nuanța de culoare și saturația ca informație de culoare pentru distincție. Segmentarea obrazului în spațiul HSV poate fi efectuată folosind domenii apropiate a nuanței de culoare și a saturației care descrie culoarea pielii umane. Așa cum este arătat în figura Fig. 4.4-14, acesta este echivalent cu decuparea unui sector din hexagon.

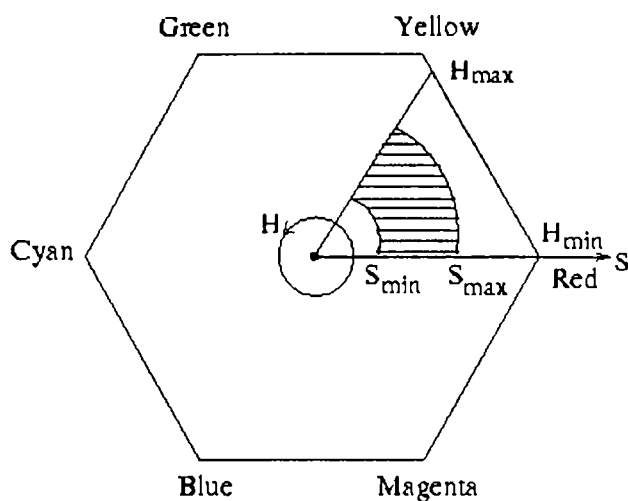


Fig. 4.4-14. Pragul de trecere în spațiul HSV

Domeniile de nuanță și saturatie pot fi definite sau estimate apriori și folosite succesiv ca referință pentru orice culoare a pielii. În acest caz am ales următorii parametrii : $S_{\min}=0.23$, $S_{\max}=0.68$, $H_{\min}=0^{\circ}$ și $H_{\max}=50^{\circ}$. Așa cum se poate observa în figura Fig. 4.4-15 acești parametrii sunt apropiați segmentării pielii de culoare albă în aceeași măsură cu segmentarea pielii de culoare galbenă. Pielea mâinilor și feței sunt corect localizate. Exista câțiva pixeli incorect detectați pe cămașă, pe geacă și pe fundal.



a) imagini color

b) imagini segmentate

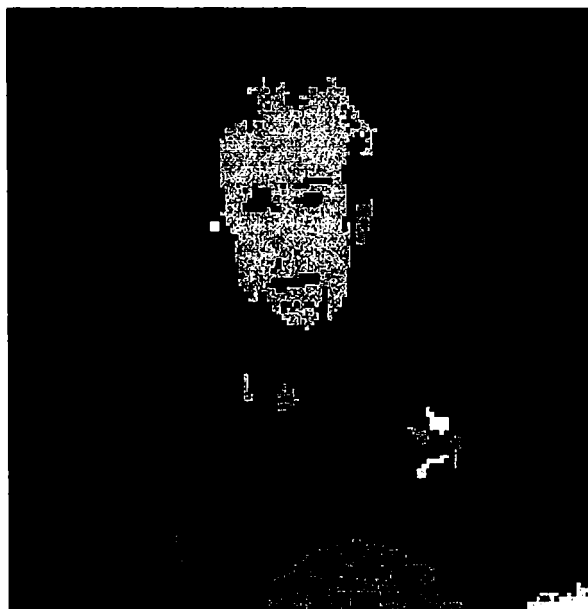
Fig. 4.4-15. Rezultatele segmentării bazate pe informația de culoare

4.4.4.1. Analiza componentelor conectate

Deoarece fața este o regiune mai mult sau mai puțin conectată prin culoarea pielii, se va efectua analiza componentelor conectate pe imaginea segmentată. Determinăm componentele conectate prin aplicarea unui algoritm de creștere a regiunii la o rezoluție slabă a imaginii segmentate. Conectivitatea este determinată pe baza a patru pixeli învecinați.



a) imagine segmentată



b) componente conectate

Fig. 4.4-16. Rezultatele analizei componentelor conectate

După cum se observă în figura Fig. 4.4-16, scăpăm de pixeli izolați, incorect detectați în fundal prin acest pas. Obținem o arie de conectare mare pentru față și pentru mâini. Câteva componente conectate de dimensiuni mici se găsesc în fundal. Pe baza informației de formă se va efectua o reducere a regiunilor candidate.

4.4.4.2. Evaluarea informației de formă

Forma ovală a unei fețe poate fi aproximată cu o elipsă. Prin urmare căutarea fețelor în imagini poate fi realizată prin detectarea obiectelor cu forma eliptică apropiată. Aceasta poate fi realizată pe baza atât a muchiilor (marginilor) cât și a regiunilor. Avantajul considerării regiunilor este că ele sunt mai robuste la zgomot și la schimbările de lumină. Astfel prima dată calculăm pentru fiecare componentă conectată C cea mai potrivită elipsă E pe baza momentelor. Apoi stabilim cât de bine componenta conectată este aproximată după cea mai potrivită elipsă. O elipsă este definită prin centrul ei (\bar{x}, \bar{y}) , orientarea θ și lungimea a și b a axei mici și mari (Fig. 4.4-17).

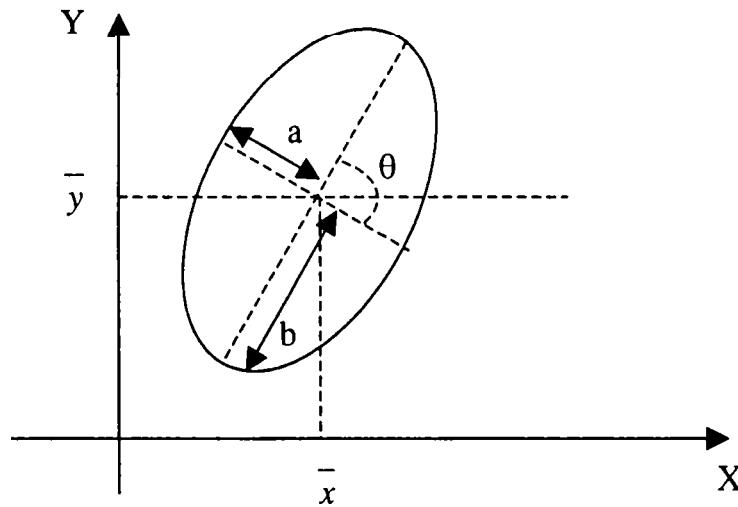


Fig. 4.4-17 Parametrii unei elipse

Centrul (\bar{x}, \bar{y}) al elipsei este dat de centrul de gravitate a componentelor conectate:

$$\bar{x} = \frac{1}{N} \sum_{(x,y) \in C} x \quad \bar{y} = \frac{1}{N} \sum_{(x,y) \in C} y \quad (4.33)$$

În ecuația de mai sus N reprezintă numărul de pixeli a componentei conectate C . Orientarea θ a elipsei poate fi calculată folosind componentele centrale $\mu_{i,j}$ a componentelor conectate:

$$\theta = \frac{1}{2} \cdot \arctan\left(\frac{2\mu_{1,1}}{\mu_{2,0} - \mu_{0,2}}\right) \quad (4.34)$$

Lungimea axei mari și axei mici a celei mai potrivite elipse poate fi determinată evaluând momentele inerției. Cu I_{min} cel mai puțin important (ultimul) și I_{max} cel mai important moment de inerției al elipsei cu orientarea θ ,

$$I_{min} = \sum_{(x,y) \in C} [(x - \bar{x})\cos\theta - (y - \bar{y})\sin\theta]^2$$

$$I_{max} = \sum_{(x,y) \in C} [(x - \bar{x})\sin\theta - (y - \bar{y})\cos\theta]^2 \quad (4.35)$$

lungimile a respectiv b rezultă din:

$$a = \left(\frac{4}{\pi}\right)^{1/4} \left[\frac{(I_{max})^3}{I_{min}}\right]^{1/8}$$

$$b = \left(\frac{4}{\pi}\right)^{1/4} \left[\frac{(I_{min})^3}{I_{max}}\right]^{1/8} \quad (4.36)$$

Apoi stabilim cât de bine componenta conectată C este aproximată cu cea mai potrivită elipsă E . În acest scop considerăm următoarele:

$$V = \frac{\sum_{(x,y) \notin E} (1-b(x,y)) + \sum_{(x,y) \in C \setminus E} b(x,y)}{\sum_{(x,y) \in E} 1} \quad (4.37)$$

cu $b(x,y)$ fiind dat de formula:

$$b(x,y) = \begin{cases} 1 & \text{dacă } (x,y) \in C \\ 0 & \text{altfel} \end{cases}$$

V determină distanța dintre componenta conectată și cea mai potrivită elipsă prin calcularea „găurilor” din interiorul elipsei și a punctelor componentei conectate care sunt în afara elipsei (Fig. 4.4-18).

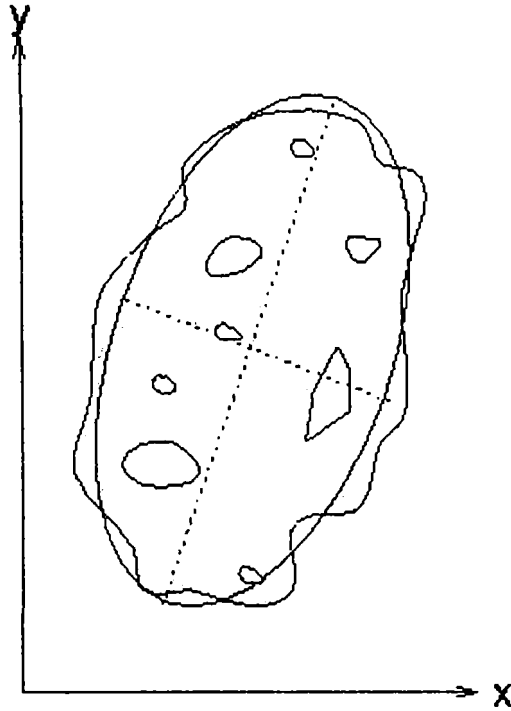


Fig. 4.4-18 Aproximarea unei componente conectate printr-o elipsă

Pe baza unui prag și a raportului V , componentele conectate care sunt bine approximate prin cea mai potrivită elipsă sunt selectate ca și “candidate” ale feței. Pe baza parametrilor elipsei deja calculați, este posibilă o reducere a numărului de posibile candidate ale feței. Acest lucru este făcut prin aplicarea pentru fiecare elipsă a criteriului cu privire la orientare și la relația dintre axa mică și axa mare. În cele ce urmează candidatele feței selectate sunt verificate pentru caracteristicile feței din interiorul componentelor conectate (Fig. 4.4-19).

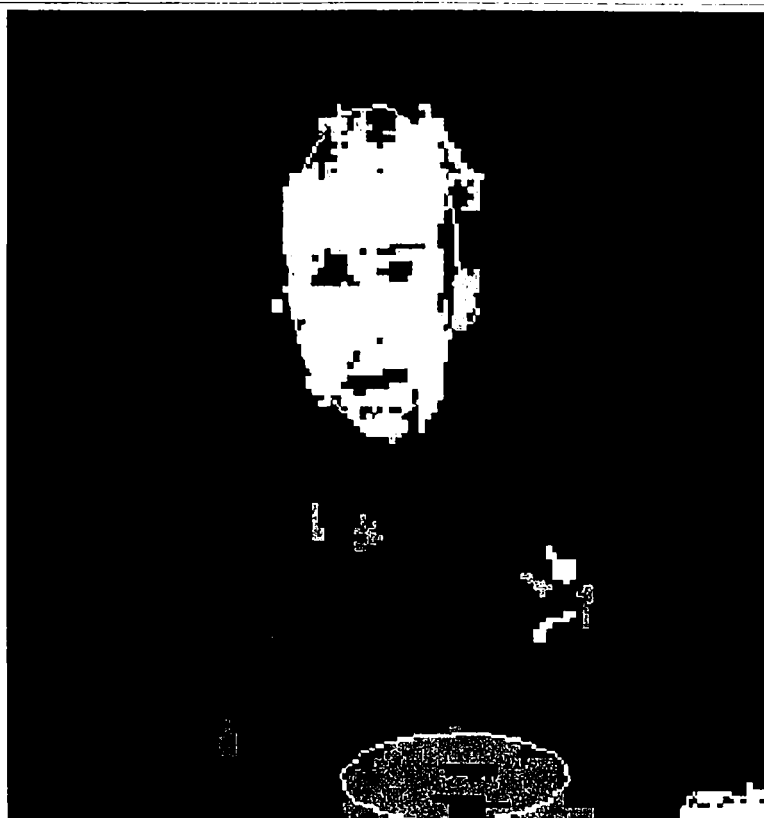


Fig. 4.4-19 Aproximarea componentelor conectate prin elipse

4.5. Extragerea caracteristicilor feței

Problema extragerii caracteristicilor feței se bazează pe observația că în intensitatea imaginilor, ochii și gura diferă de restul feței din cauza luminozității lor scăzute. În cazul ochilor, motivul este culoarea pupilelor și scobitura orbitelor ochilor. Chiar și dacă ochii sunt închiși, culoarea închisă a orbitelor oculare sunt suficiente pentru extragerea regiunilor ochilor. Lumina culorii roșii a buzelor scoate în relief gura în contrast cu regiunea înconjurătoare și reduce luminozitatea locală a imaginii în intensitate. Acest lucru este mai evident când gura este parțial deschisă.

În continuare vom lua în considerare informația de intensitate din interiorul componentelor conectate. În pasul de preprocesare vom intensifica regiunile întunecate în interiorul componentelor conectate. Pentru extragerea caracteristicilor feței vom prezenta o abordare care se bazează pe analiza min-max și evaluează direct proiecțiile lui x și y în relieful nivelului de gri.

4.5.1. Accentuarea caracteristicilor feței

Accentuarea regiunilor întunecate este făcută prin folosirea unor operații morfologice. Prima dată se aplică o eroziune la scara gri. Ca elemente structurale vom folosi un dreptunghi de 5×3 , care este alungit în direcție orizontală în ambele părți, ochii și gura sunt alungiți orizontal. Apoi îmbunătățim contrastul din interiorul componentelor conectate prin următoarea operație extremă de accentuare:

$$g(x,y) = \begin{cases} \min & \text{dacă } f(x,y) - \min < \max - f(x,y) \\ \max & \text{altfel} \end{cases}$$

În relația de mai sus min și max indică minimumul și maximumul valorii nivelului de gri în vecinătatea lui $f(x,y)$. Rezultatele acestei preprocesări sunt ilustrate în figura Fig. 4.5-1. Ochii, gura, părți din păr și din barbă sunt scoase în evidență.

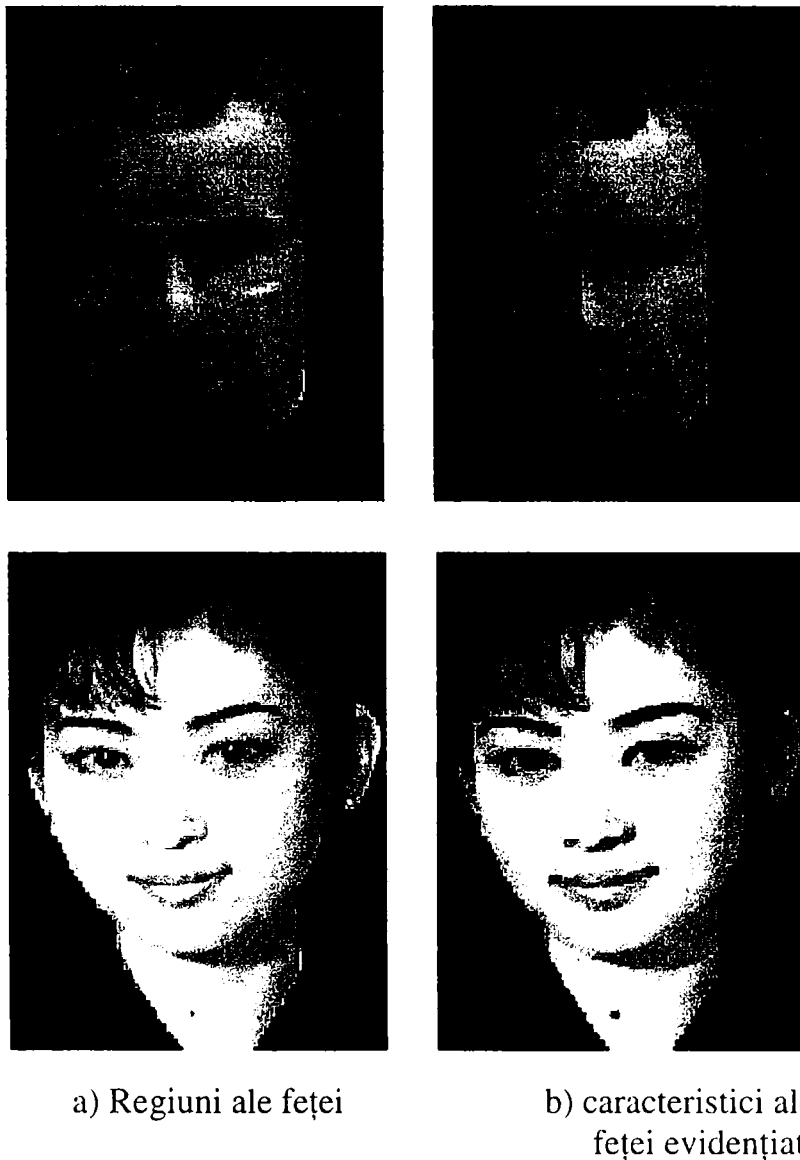


Fig. 4.5-1. Accentuarea caracteristicilor feței

4.5.2. Extragerea caracteristicilor feței prin analiza min-max

În continuare voi încerca extragerea caracteristicile feței printr-o abordare care se bazează pe analiza min-max. În acest caz ochii și gura sunt căutate direct prin evaluarea proiecției y și a proiecției x din relieful topografic al nivelului de gri. Din nou vom considera informația preprocesată a nivelului de gri din interiorul componentelor conectate.

Normalizarea

Deoarece ochii și gura sunt orientați orizontal, este necesară o normalizare de orientare a feței candidat. Avantajul normalizării este acela că, după acest pas putem căuta ușor după ochi și gură de-a lungul direcției orizontale. Pentru a obține această normalizare, este necesară o rotație a interiorului componentei conectate. Unghiul de rotație este acceptat ca fiind egal cu

orientarea θ a celei mai potrivite elipse a componenteii conectate. Coordonatele transformării sunt definite de:

$$\begin{pmatrix} xr \\ yr \end{pmatrix} = \begin{pmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

Aici prin xr și yr se specifică coordonatele de rotație a lui x și y

Calcularea lui x-relief și y-relief

După normalizarea componenteii conectate proiecția lui y este determinată. În acest scop calculăm media nivelului de gri pentru fiecare rând al componenteii conectate. Rezultatul y -relief este netezit printr-un filtru de înmulțire de lărgime 3. Apoi minimul și maximul este căutat în y -relief netezit. Prin căutarea gradientilor pentru fiecare minim în vecinătatea maximului său, minimul cel mai semnificativ este selectat.

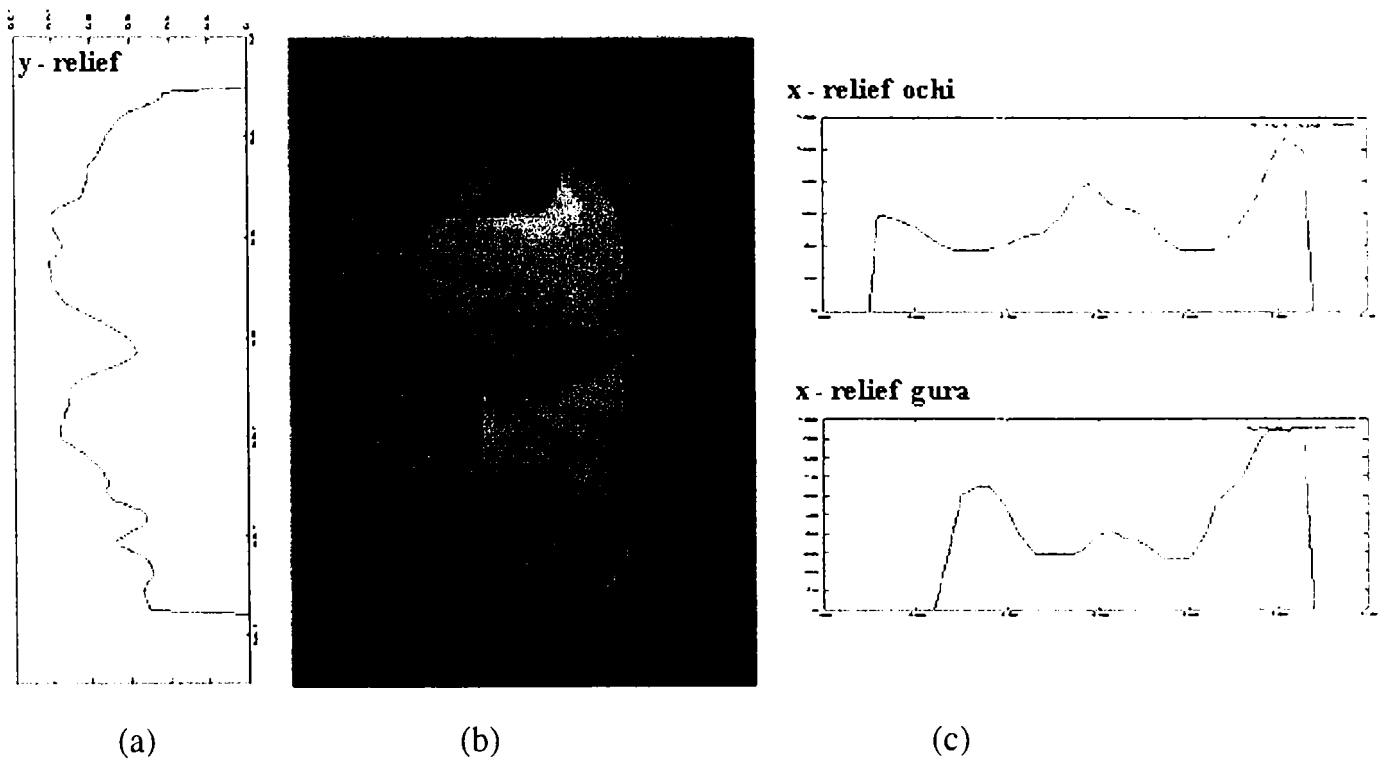


Fig. 4.5-2. Analiza min-max

Pentru cel mai semnificativ minim al lui y -relief, x -relief este calculat. Pentru a obține mai multă robustețe, determinăm x -relief pentru mai multe rânduri din jurul minimului y -relief. Fiecare x -relief este calculat prin media nivelelor de gri a 3 rânduri vecine pentru fiecare coloană. După aceea x -relief rezultat este netezit în direcția x printr-un filtru mediu de lărgime 3. Astfel minimul și maximul sunt determinate.

Ca rezultat, se obține un y -relief netezit (Fig. 4.5-2.a) pentru regiunea feței candidat cu o listă atașată de minime și de maxime, și pentru minimul semnificativ al lui y -relief se obțin x -relief netezite (Fig. 4.5-2 c) cu o listă atașată de minime și maxime ale lor.

Căutarea candidatelor pentru ochi

Începând cu minimumul cel mai de sus al y-reliefului, vom căuta prin lista de minime a x-relief pentru a găsi două minime care satisfac cerințele pentru pozițiile ochilor.

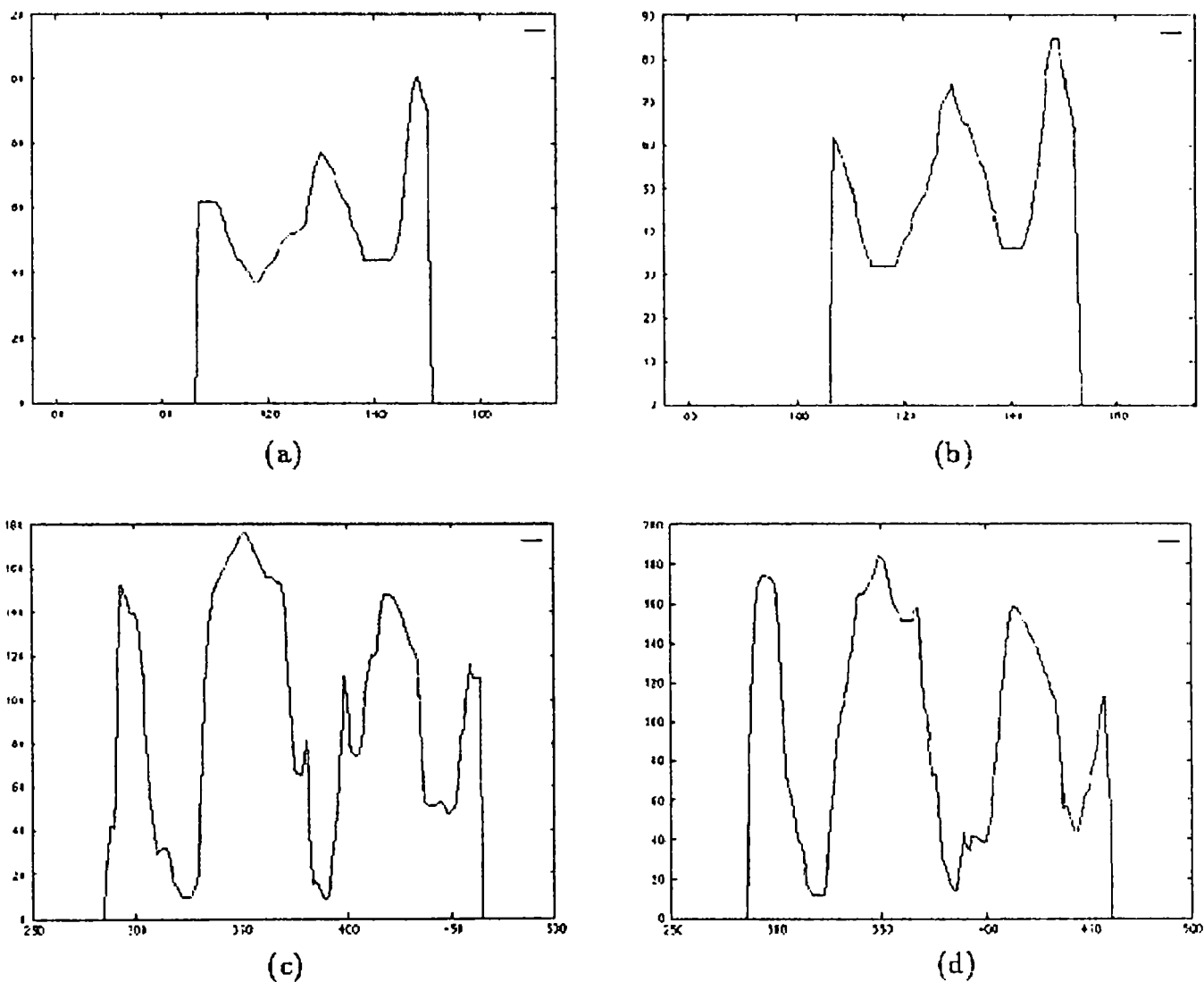


Fig. 4.5-3. Exemple de x-relief pentru regiunea feței

Exemple tipice pentru x-relieful regiunii ochilor sunt arătate în figura Fig. 4.5-3. În Fig. 4.5-3.a, Fig. 4.5-3.b sunt prezentate exemple de x-relief care conțin doar minime corespunzătoare ochiului stâng și drept, iar exemplele din Fig. 4.5-3.c respectiv Fig. 4.5-3.d includ de asemenea și minime corespunzătoare părului.

Cerințele pentru ochi sunt specificate în cele ce urmează :

- Ochii sunt situați în partea de sus sau de mijloc a capului.
- Acolo sunt minime semnificative pentru ochiul stâng și ochiul drept.
- Minima are valori de nivel gri similare.
- Un maximum semnificativ este între cele două minime ale ochilor.
- Raportul dintre distanța ochilor și lățimea capului este într-un anumit interval.

Evaluarea cât de bine o pereche de minime îndeplinesc cerințele este făcută pe baza teoriei fuzzy. Astfel definim o funcție de apartenență pentru fiecare cerință. Funcția de apartenență pentru evaluarea raportului dintre distanța ochilor și lățimea capului este ilustrată în exemplul din Fig. 4.5-4.a. Parametrii $P1$, $P2$, $P3$ și $P4$ sunt definiți în funcție de lățimea

componentei conectate care este evaluată în conformitate cu lățimea capului. În cazul în care raportul măsurat r este între $P2$ și $P3$, cerințele sunt complet îndeplinite și factorul de certitudine (CF) are valoarea 1.0. Dacă $P1 < r < P2$ sau $P3 < r < P4$, cerințele sunt parțial îndeplinite și $0.0 < CF < 1.0$. În alte cazuri cerințele nu sunt îndeplinite și $CF = 0.0$. Un alt exemplu este funcția de apartenență pentru evaluarea similarității valorilor de nivel gri, ilustrat în figura Fig. 4.5-4.b. Odată cu creșterea diferenței nivelului de gri a ochiului stâng și a ochiului drept, CF scade. Pe baza evaluării acestor criterii, sunt selectate candidatele pentru ochi.

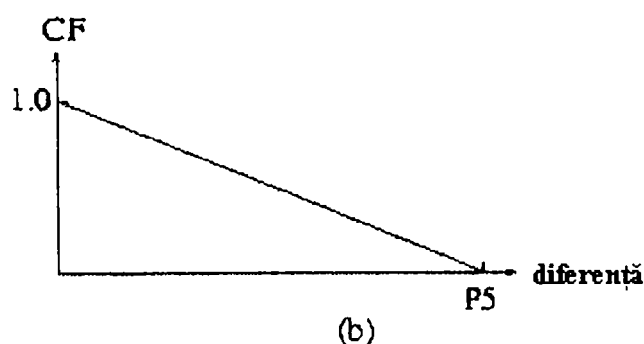
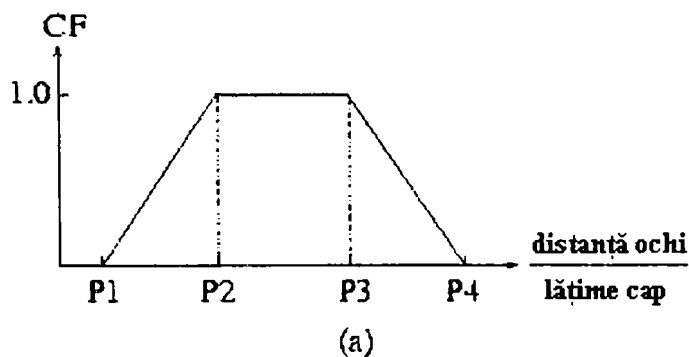


Fig. 4.5-4. Evaluarea criteriului: (a) funcția de apartenență pentru raportul dintre distanță ochi și lățime cap. (b) funcția de apartenență pentru similaritatea valorilor nivelului de gri

Fețele candidate trebuie să îndeplinească un minim de evaluare pentru fiecare criteriu la fel ca și un minim de evaluare pentru suma tuturor evaluărilor. Exemple de fețe candidate selectate pentru ochi, reprezentate prin încrucișări sunt arătate în figura Fig. 4.5-5 de mai jos.

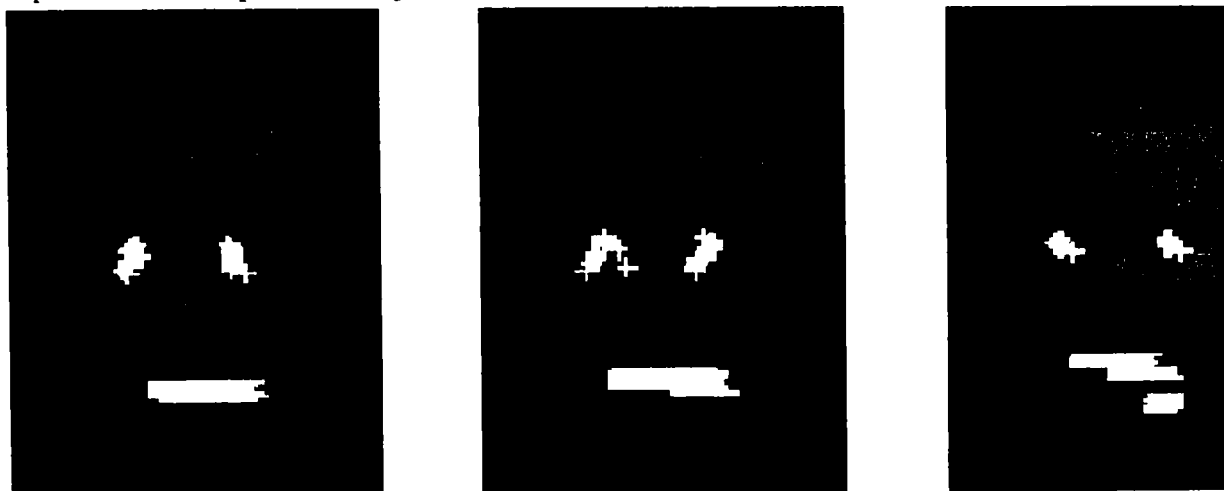


Fig. 4.5-5. Fețe candidat ale ochilor și gurii

Căutarea candidatelor pentru gură

O căutare completă pentru gură, printre toate x-reliefele determinate se face la fel ca și în cazul ochilor. După cum se vede în figura Fig. 4.5-6, gura este caracterizată în x-reliefe printr-un bazin semnificativ cu una sau două minime în interior. În special colțurile gurii pot fi văzute foarte bine în x-reliefe (Fig. 4.5-6.c,d).

Cerințele pentru gură sunt definite după cum urmează:

- Gura este situată în partea de jos sau de mijloc a capului.
- Acolo sunt două maxime semnificante cu nici o maximă mai mare între ele.
- Un minim semnificativ este între maxime
- Raportul lățimii gurii la lățimea capului este într-un anumit interval.

În analogie cu căutarea candidatelor pentru ochi, pentru a vedea cât de bine se îndeplinesc cerințele pentru gură, fiecare pereche de maxime este evaluată. Ca rezultat obținem pentru regiunea feței un set de candidate pentru gură. Exemple de candidate ale gurii, reprezentate ca segmente de linie orizontale sunt arătate în figura Fig. 4.5-5 .

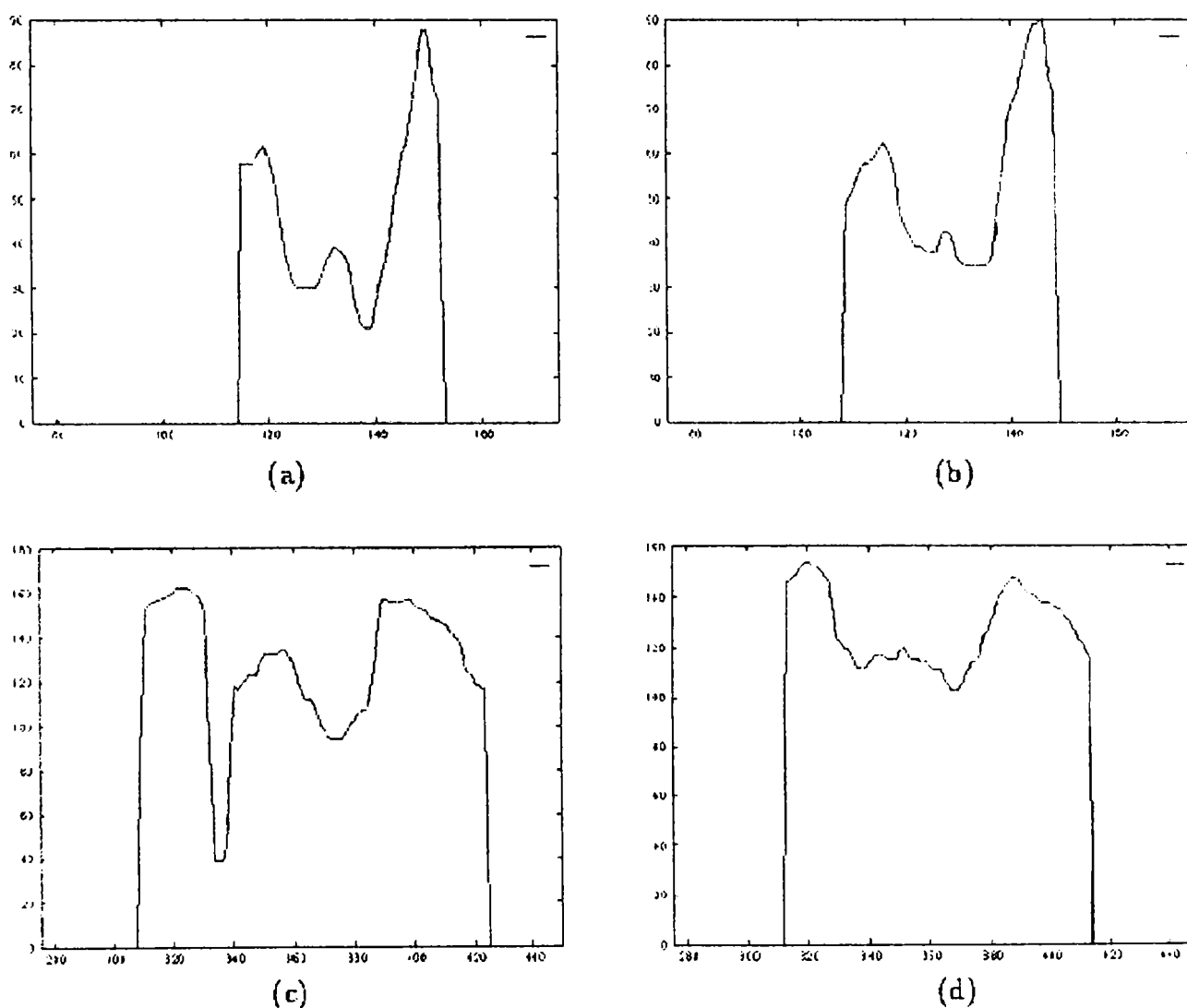


Fig. 4.5-6. Exemple de x-reliefe pentru regiunea gurii

Selecția caracteristicilor feței

După determinarea candidatelor pentru ochi și gură, cea mai bună combinație a caracteristicilor feței este selectată. Pentru aceasta trebuie să examinăm cazurile din tabelul următor.

Cazuri	Candidate ochi	Candidate gură	Selecție
1	Nu	Nu	Fără caracteristici faciale
2	Da	Nu	Cel mai bun candidat pentru ochi
3	Nu	Da	Cel mai bun candidat pentru gură
4	Da	Da	Cea mai bună combinație ochi-gură

Tabel 1. Cazurile pentru selecția caracteristicilor feței.

Dacă nu găsim candidate pentru ochi și gura (cazul 1). Atunci nu avem caracteristici faciale ca și rezultat. În cazul în care detectăm doar candidate pentru ochi (cazul 2) sau doar candidate pentru gura (cazul 3) alegem cea mai bună candidată. Pentru aceasta grupăm candidatele pentru ochi și gură în conformitate cu coordonatele lor pe x, din stânga și dreapta ochilor sau din stânga și dreapta gurii. Această grupare este făcută folosind un algoritm Min-Max necontrolat. Începând cu o pereche de ochi sau colțuri ale gurii ca primul centru de grupare, distanțele la toate celelalte candidate sunt calculate. Candidatele cu distanța maximă sunt alese ca și al doilea centru. Apoi următoarea procedură iterativă este executată: Pentru fiecare candidată sunt calculate toate distanțele la toate centrele de grupare și se face o atașare a candidatei la gruparea cu distanța maximă. Candidata cu distanță maximă față centrele de grupare este aleasă ca un nou centru de grupare. Procesul iterativ se oprește când această distanță maximă este mai mică decât jumătate din distanța medie dintre centrele de grupare. Ca rezultat se obține un set de centre de grupare. Fiecare din ele este o pereche, constând din centrul stâng și drept pentru ochi sau colțurile gurii. Pe baza numărului de voturi pentru un centru și media evaluării candidatelor atașate la centru, cea mai bună candidată este aleasă pentru gură sau ochi.

Cea mai bună combinație ochi-gură este determinată, dacă candidatele ochilor la fel ca și candidatele gurii sunt detectate (cazul 4). Pentru aceasta la început candidatele ochilor și gurii sunt grupate așa cum s-a descris în paragraful anterior. Apoi se construiesc combinațiile de caracteristici prin combinarea fiecărui centru de grupare a ochilor cu fiecare centru de grupare a gurii. Cea mai bună combinație este selectată după următorul criteriu:

- Candidata ochilor la fel ca și candidata gurii sunt bine stabilite.
- Raportul distanței verticale ochi-gură față de înălțimea capului este într-un anumit interval.
- Gura acoperă parțial pe orizontală regiunea dintre ochiul stâng și drept.

Un exemplu este ilustrat în figura Fig. 4.5-7 de mai jos:

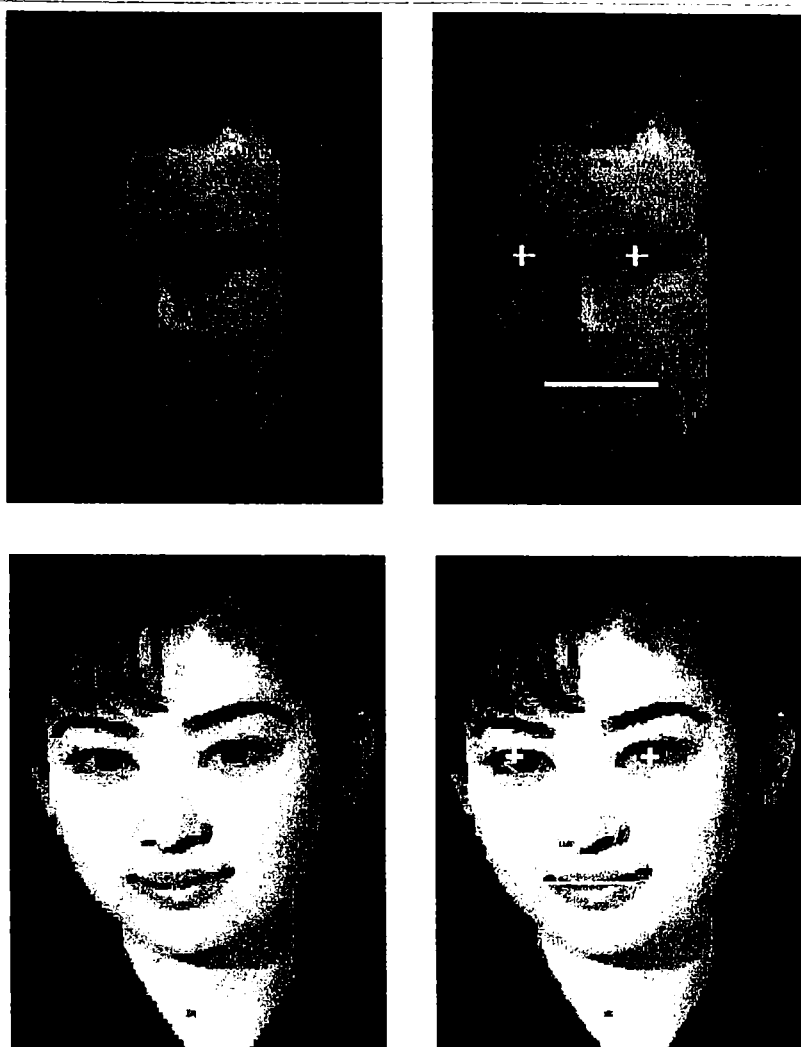


Fig. 4.5-7. Rezultatele extragerii de caracteristici prin metoda min-max

4.5.3. Evaluarea extracției caracteristicilor feței prin analiza min-max

Pentru a evalua robustețea detecției de caracteristici ale feței prin analiza min-max, se utilizează metoda pe o secvență de imagini constând din 150 de cadre. Exemple de cadre sunt arătate în Fig. 4.5-8 a, b și c. Deoarece fața conține caracteristici cum ar fi: barbă și ochelari, acest lucru determină schimbarea expresiei feței. Rezultatele evaluării sunt ilustrate în Tabelul 2. Caracteristicile feței, care sunt ochii sau gura sunt detectate în proporție de 86% din cadre.

Rezultatele extragerii de caracteristici prin metoda min-max	%
Caracteristicile feței	86
Ochi	96
Gura	87
Ochi corect detectați	98
Gura corect detectată	86

Tabel 2

În 96% din cadre ochii sunt extrași, dar doar în 98% din aceste cazuri corect. Gura este detectată în proporție de 87% din cazuri, dar corect doar în proporție de 86%. Aceste rezultate sunt foarte satisfăcătoare. Pentru unele cadre apar erori. Exemple de astfel de erori sunt arătate în Fig. 4.5-8. De exemplu în unele cazuri uneori barba este confundată cu gura (Fig. 4.5-8.a). Rareori ochelarii sunt confundați cu ochii (Fig. 4.5-8.b). În cazul secvenței

imaginii din Fig. 4.5-8.c, uneori apar probleme cu sprâncenele, în special când ochii sunt închiși.



(a)



(b)



(c)

Fig. 4.5-8. Probleme întâlnite la extracția caracteristicilor feței prin analiza min-max

5. CTAG - O metodă propusă de abordare a texturilor color în procesul de detectare a feței umane

5.1. Introducere

Metoda propusă încercă să contribuie la îmbunătățirea sistemului de recunoaștere descris în [1]. Acest sistem a fost dezvoltat ca și un proiect de doctorat și este orientat pe monitorizarea automată a mediului folosind imaginile animalelor în mediul lor natural și a fost aplicat în recunoașterea acestor animale în scene naturale, în principal, provenite din pădurea Amazoniană.

J. Camapum folosește culoarea ca una dintre principalele caracteristici în procesul de recunoaștere. La început sistemul a fost bazat pe compararea grafurilor regiunilor adiacente atributate, metodă prezentată de Fathima în [8], dar care a fost înlocuită printr-o altă reprezentare, propusă de Mattas în [9], și anume Graful de Adiacență al Culoarelor (CAG), care combină atât histogramele cât și reprezentarea graf al regiunilor adiacente.

Culoarea are un mare avantaj asupra altor metode, deoarece este în mare măsură independentă de punctul de vedere și rezoluție, dar pentru o recunoaștere corectă trebuie să mai folosim și alte caracteristici.

Îmbunătățirea propusă în această lucrare este luarea în considerație nu numai a culorii ci și a texturii obiectelor (regiunilor). În acest fel, ca și o consecință naturală, reprezentarea propusă am denumit-o Graful de Adiacență al Texturilor Color (CTAG) [30], în care fiecare nod va conține descrierea de culoare și textură a regiunii respective. Descrierea texturii a fost propusă de A. Pikaz și A. Averbuch în [3]. Pentru a face această reprezentare cumva, independentă de scară, luăm în considerare posibilitatea generalizării ei considerând-o la scări diferite (teoretic am putea folosi un parametru scară continuu, dar în practică vom folosi un parametru scară discret). Această abordare este bazată pe teoria **scale-space**, prezentată de Lindberg în [2]. Motivul pentru care incorporăm un factor scară explicit este acela că la diferite scări, pentru aceeași textură corespund diferiți vectori de descriere a texturilor, dar la scări suficient de grosolane vectorii de descriere a texturilor tind spre componenta originală de culoare a CTAG, aceasta prezervând toate caracteristicile reprezentării originale CAG.

Pentru această reprezentare au existat anumite cerințe și anume ar trebui să nu fie consumatoare de timp și nu ar trebui să fie consumatoare de memorie.

În următoarele rânduri aș vrea să vă prezint pe scurt structura de bază, dezvoltată de A. Pikaz și A. Averbuch, pe care se bazează descrierea propusă. Ei au propus o caracterizare topologică a texturilor în nuanțe de gri, folosind o reprezentare multirezoluție. Ei au folosit o structură de date numită **MRCG MultiResolution Clusters Graphs**, compusă dintr-o secvență de grafice notate prin $\{N_s(t)\} s \in I$. Pentru un s și t dat $N_s(t)$ este definit ca numărul de obiecte conectate care sunt compuse din cel puțin s pixeli dacă imaginii i s-a aplicat pragul t . Această structură este o reprezentare multinivel a imaginii unde parametrul s controlează rezoluția. În [3] este arătat cum această reprezentare păstrează multe proprietăți relevante și dorite ale texturilor.

În continuare voi prezenta structura MRCG $\{N_s(t)\} s \in I$. Notăm prin I_t imaginea binară rezultată după aplicarea asupra imaginii I a pragului t . Pentru un întreg dat s , funcția $N_s(t)$ este definită ca numărul de componente, din imaginea I_t , 4-conectate (componente conectate din 4 direcții), având cel puțin s pixeli. Valorile pentru t aparțin unui set de nivele de gri, $T = \{0, 1, \dots, 255\}$. Valoarea pentru s provine dintr-un set R de valori "rezoluție". Secvența de grafice, $\{N_s(t)\}_{s \in R}$, pentru fiecare $t \in T$, $s \in R$, poate fi calculată în cel mai rău caz într-un timp de

complexitate $O(\alpha(n,n) \cdot |T| \cdot |R| \cdot n)$, unde n este numărul de pixeli din imagine, iar $\alpha(n,n)$ inversul funcției lui Ackerman. Deoarece $|T|$ este independent de dimensiunea imaginii, iar $|R|$ este de asemenea practic o constantă și fiindcă inversa funcției lui Ackerman este aproape constantă, complexitatea de mai sus este aproape liniară.

5.2. Calculul graficelor $N_s(t)$

Pentru fiecare dimensiune s , trebuie construit graficul funcției $N_s(t)$. Astfel pentru fiecare valoare de prag t , trebuie calculat numărul obiectelor cu dimensiunea s . Dacă notăm cu I_t imaginea obținută prin aplicarea pragului t imaginii de intrare, fiecare pixel al lui I_t va fi negru (0) sau alb (255), depinzând de valoarea pixelului corespunzător din imaginea de intrare (dacă este $\leq t$ atunci este negru, altfel este alb). Este clar că pixelii negri ai lui I_{t+1} sunt formați din pixelii negri ai lui I_t și poate câțiva pixeli în plus. Astfel calculul lui $N_s(t+1)$ se poate baza pe calculul lui $N_s(t)$.

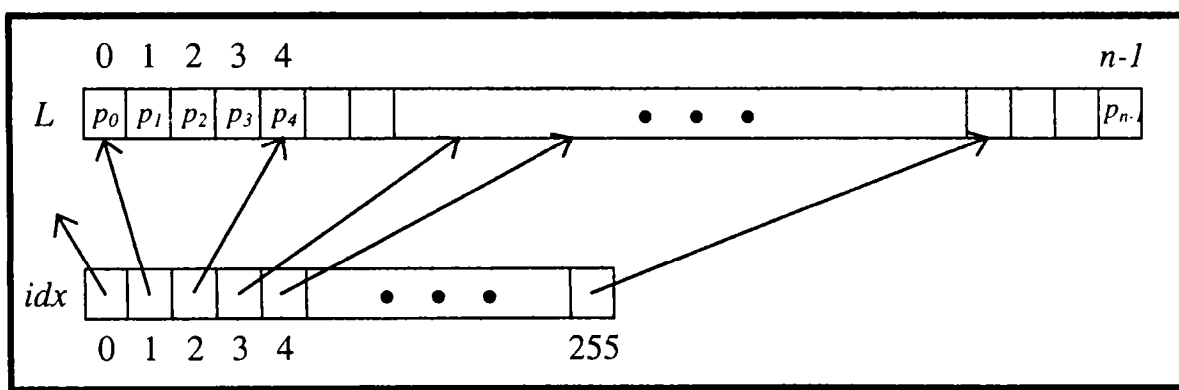


Fig. 5.2-1. Structura listei de indecși L .

Fie m numărul de nivele de gri (în acest caz 256), și fie n numărul de pixeli din imaginea de intrare. Construcția graficului $N_s(t)$ este realizată în felul următor :

1. Construim o listă L de n pixeli (puncte bi-dimensionale (x,y)), și un vector idx de dimensiunea m , astfel încât indexul primului pixel din L cu nivelul gri egal cu i (Fig. 5.2-1). Construcția acestor structuri se face într-un timp direct proporțional cu n , în felul următor :

inițializează idx cu zerouri;
pentru fiecare pixel p din imaginea de intrare
incrementează $idx(\text{gray_level}(p))$;
pentru $i \leftarrow 1$ până la $m-1$
$idx(i) \leftarrow idx(i-1) + idx(i)$;
pentru fiecare pixel p
$L(\text{idx}(\text{gray_level}(p))) \leftarrow p$;
incrementează $idx(\text{gray_level}(p))$;

2. Definim o structură *Union-Find*, unde fiecare pixel este un element, și inițial, fiecare pixel este un set disjunct.
3. Fie $count$ numărul curent de obiecte. Inițializăm $count$ pe 0. Parcurgem elementele listei L , de la început spre sfârșit. Fie t nivelul de gri al pixelului curent, p . Unim elementul p pe rând cu fiecare dintre cei 4 vecini, dacă nivelul de gri al vecinului

respectiv nu este mai mare de t . Dacă s-a făcut cel puțin o uniune, atunci $N_s(t)$ este actualizat în următorul mod. Fie s_1 și s_2 dimensiunile celor două seturi care urmează a fi reunite. Dacă atât s_1 cât și s_2 sunt mai mici decât s , dar $s_1 + s_2 \geq s$, atunci este adăugat un nou obiect și astfel $count$ este incrementat. Dacă atât s_1 cât și s_2 nu sunt mai mici decât s , atunci un obiect va fi eliminat și astfel $count$ este decrementat. Valoarea lui $N_s(t)$ este numărul actualizat de obiecte, obținut după procesarea și a ultimului pixel având nivelul de gri egal cu t (Fig. 5.2-2).

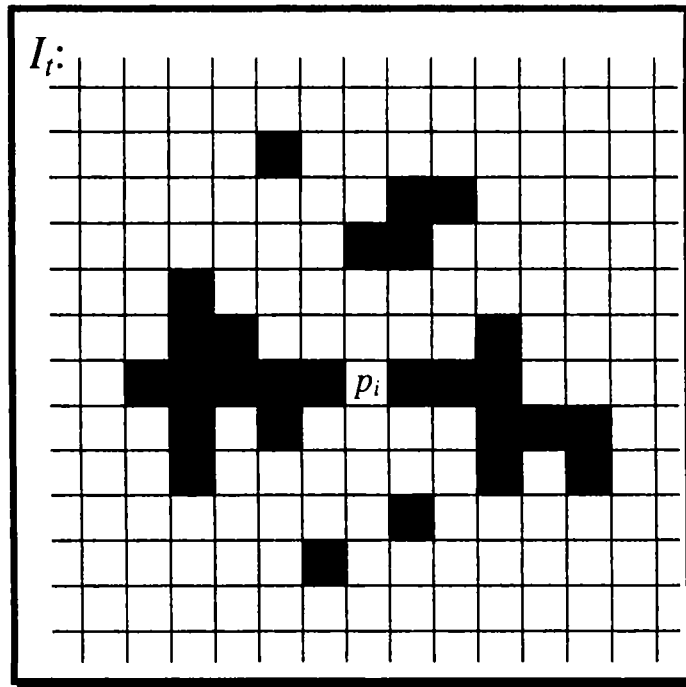


Fig. 5.2-2. Ilustrarea construirii graficului $N_s(t)$.

5.3. Proprietățile reprezentării MRCG pentru texturi.

Graficul $N_s(t)$ are o formă generică de clopot distorsionat atâta timp cât textura este compusă dintr-o repetiție regulată a unui element de bază. Are o formă generică a unei sume de clopote distorsionate, atunci când textura este compusă dintr-o repetiție regulată a câtorva elemente de bază diferite.

5.3.1. MRCG supus unor transformări monotone ale nivelului de gri.

Comportamentul MRCG supus unor transformări monotone ale nivelului de gri este sumarizat prin următoarea propoziție.

Fie I o imagine de intrare și fie \hat{I} imaginea obținută după aplicarea unei transformări monotone a nivelului de gri asupra lui I , transformare notată prin $a(t)$. Fie $N_s(t)$ reprezentarea MRCG corespunzătoare lui I , atunci reprezentarea MRCG a lui \hat{I} este $N_s(a(t))$.

Demonstrația acestei propoziții derivă direct din faptul că pentru orice t , $I_t = \hat{I}_{a(t)}$.

Există două cazuri particulare ale proprietății enunțate în propoziția de mai sus:

- Notând prin $I + c$ imaginea obținută prin adăugarea unei valori constante c nivelelor de gri din intervalul $0, \dots, 255 - c$ ale imaginii I și deasemenea presupunând că $N_s(t) = 1$ pentru $t \geq 255 - c$, atunci graficul $N_s(t)$ al imaginii $I + c$ este graficul $N_s(t + c)$.
- Întărirea contrastului imaginii originale la o scară neuniformă a axei- t a graficului $N_s(t)$. Deoarece egalizarea histogramei este o transformare monotonă nedescrescătoare a nivelelor de gri, un corolar direct a acestei propoziții este următoarea :

Fie I imaginea de intrare și fie \tilde{I} imaginea obținută după o transformare de egalizare a histogramei. Atunci reprezentarea MRCG a lui \tilde{I} și I diferă cel mult cu o transformare monotonă a parametrului t .

Acest corolar furnizează o unealtă pentru normalizarea texturilor, asta ca un pas de preprocesare înaintea comparării acestora.

5.3.2. Efectul netezirii/întării imaginii asupra graficelor $N_s(t)$.

Efectul unei neteziri asupra graficului $N_s(t)$ este similară mării parametrului s . Un cluster corespunde unei văi în suprafața imaginii, care este sub planul pragului. Dimensiunea clusterului este aria de pe planul de prag, care este tăiat de suprafața imaginii. Văile care taie planul de prag dar care au o arie mai mică decât s nu sunt numărate pentru $N_s(t)$. Astfel tăind văile sub primul vârf care creează un cluster de dimensiune $\geq s$, nu afectează graficul $N_s(t)$. Figura Fig. 5.3-1 ilustrează ideea unor imagini 1D.

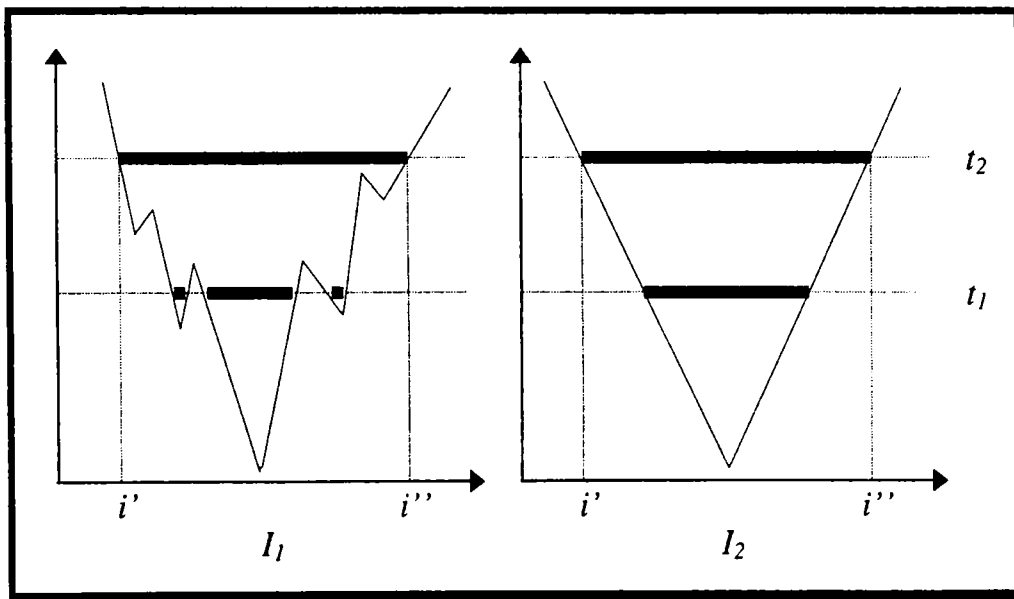


Fig. 5.3-1. Ilustrarea graficului $N_s(t)$ pentru un semnal 1D.

Graficul $N_s(t)$ al unei versiuni netezite a imaginii de intrare este astfel micșorat pe ambele axe. Deoarece limita unei filtrări trece-jos ("low-pass") este o imagine cu o valoare gri constantă, graficul $N_s(t)$ a unei imagini la care s-a aplicat o filtrare trece-jos iterativă infinită este un vârf de înălțime 1 la $t =$ cu media imaginii. În mod similar, graficul $N_s(t)$ a unei versiuni întărite a imaginii de intrare este extins pe ambele axe. Deoarece limita unei filtrări trece-sus ("high-pass") este o imagine binară, graficul $N_s(t)$ a unei imagini la care s-a aplicat o filtrare trece-sus iterativă infinită este o constantă.

5.3.3. Influența unor caracteristici de bază ale texturilor asupra MRCG.

În această secțiune vom examina influența unor caracteristici “clasice” ale texturilor asupra reprezentării MRCG. În acest context ar trebui menționat faptul că reprezentarea MRCG este invariantă la orientare. Dacă este necesar, atunci caracteristica orientare poate fi calculată separat.

Regularitatea. Ne referim la o textură ca fiind regulată dacă este compusă din repetiții uniforme ale elementelor de bază. Vom arăta că, cu cât o textură este mai regulată, cu atât mai neted este graficul $N_s(t)$ corespunzător. Deasemenea, pentru fiecare populație de elemente de textură, acestora le corespunde câte o componentă în formă de clopot distorsionat din graficul $N_s(t)$. Într-adevăr, fie $I_{s,t}(i,j)$ o variabilă aleatoare index care ia valoarea 1 dacă pixelul (i,j) este pixelul din dreapta-jos a unei componente conectate cu cel puțin s pixeli în I_t . Cele două variabile aleatoare $I_{s,t}(i,j)$ și $I_{s,t}(i',j')$ sunt independente pentru pixeli suficient de distanți. Astfel varianța variabilei aleatoare $N_s(t)$, care este suma variabilelor index, este mai mică decât dacă textura ar fi neregulată. Aceasta înseamnă că graficul $N_s(t)$, care este numărul mediu de clustere, este așteptat să fie mai neted.

Liniaritatea. Considerăm două imagini, I_1 și I_2 . Imaginea I_1 compusă din segmente de linii drepte având câte s' pixeli, iar imaginea I_2 compusă din pătrate având câte s' pixeli. Deoarece avem de a face cu imagini digitale, nivelurile de gri ale pixelilor aceluiași obiect nu sunt constante. Aceasta fiind privită ca efectul unor zgomote. Astfel există valori ale pragului t astfel ca imaginea I_t să conțină doar porțiuni ale obiectelor originale. Deoarece o linie este mai ușor spartă prin zgomot decât un pătrat, graficul său $N_s(t)$ va avea valori mai mari pentru s mici și valori mai mici pentru s mari.

Densitatea. Factorul de densitate are un efect trivial asupra MRCG, deoarece texturi mai dense înseamnă valori mai mari ale funcției $N_s(t)$ pentru valori destul de mari ale lui s . Un alt efect este acela că obiectele mai dense sunt unite mai repede. Deasemenea este de observat că graficul $N_s(t)$ unor texturi mai dense este mai neted. Așa cum am menționat mai devreme $N_s(t)$ este o sumă a unor variabile index aleatoare și astfel pentru texturile mai dense rezultă că funcția $N_s(t)$ este o sumă de mai multe variabile index aleatoare, ceea ce implică o varianță mai mică.

Asprimea. Aceasta are o influență trivială asupra graficului $N_s(t)$, deoarece asprimea înseamnă clustere mai mari în imagine.

În figura următoare prezint modificările propuse pentru sistemul vizual descris în [1] (Fig. 5.3-2).

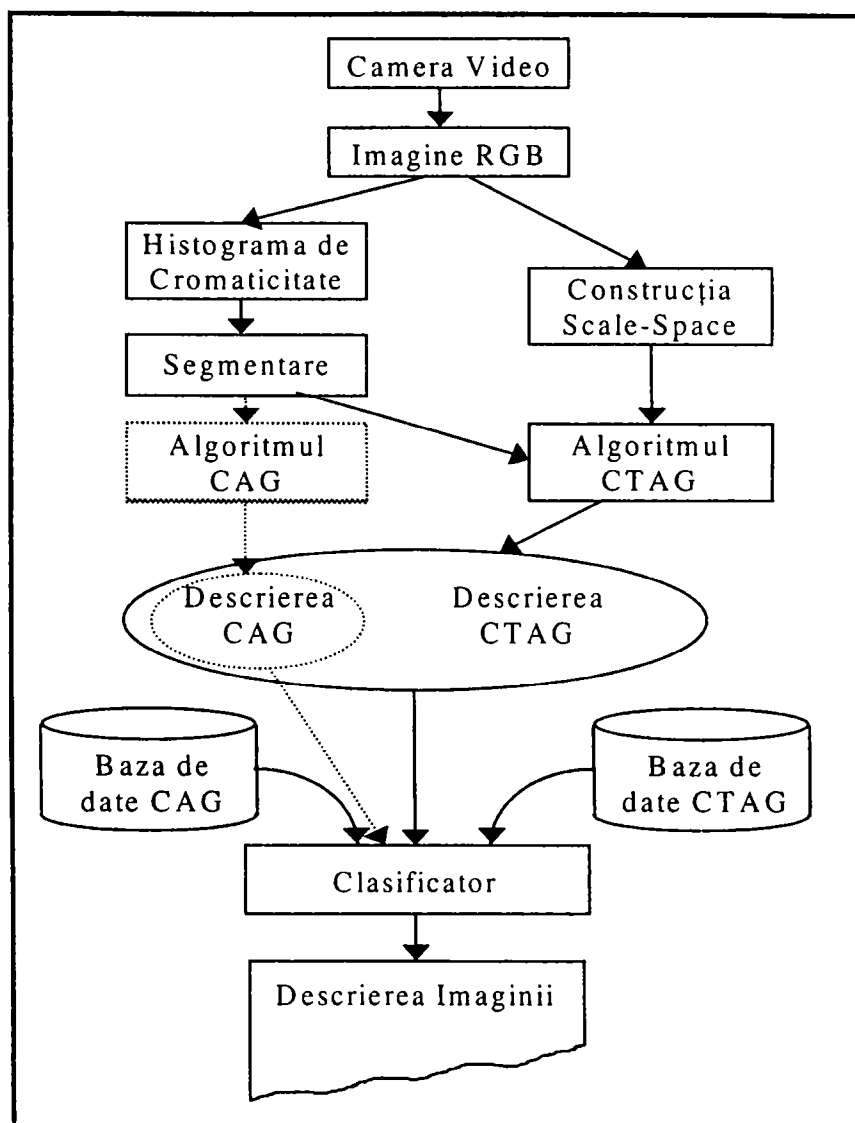


Fig. 5.3-2 Modificarea propusă pentru sistemul vizual.

Metoda originală de descriere a texturii, descrisă de Pikaz și Averbuch, a fost proiectată pentru imaginile în nuanțe de gri, astfel a trebuit să dezvoltăm o extensie a acestui model, care a trebuit să fie cât mai simplă. Astfel am identificat trei metode de extensie[30]:

- O posibilitate este utilizarea a trei vectori în locul unui singur vector. Acești 3 vectori ar descrie cele trei componente de culoare ale imaginii.
- A doua posibilitate este folosirea a doar doi vectori, care descriu cromaticitatea imaginii (chr și chg).
- A treia opțiune este reducerea numărului de culori sub un anumit număr rezonabil de valori diferite (de exemplu 4096). Aceasta fiind necesară din cauza cerinței minimizării timpului de calcul – metodă neaplicabilă în mod direct.

O altă categorie de extensii care a trebuit să fie dezvoltată a fost în legătură cu faptul că descrierea originală a fost aplicată doar pe anumite mostre de texturi și nu pe regiuni rezultate după procesul de segmentare.

Acest proces poate fi aplicat doar regiunilor care au o suprafață compactă cu o dimensiune minimă $minA$. Pentru celelalte regiuni vom considera doar informația de culoare a regiunii respective.

Pentru a folosi această reprezentare într-o fază de comparare trebuie ca vectorii să fie normalizați astfel încât pentru două regiuni cu aceeași textură dar cu arii diferite $regA1$ și $regA2$ cei doi vectori corespunzători să fie la o distanță foarte mică. Aceasta poate fi realizată prin calcularea unui nou vector scalat cu un factor egal cu maximul vectorului original.

O altă problemă pe care am întâlnit-o a fost aceea că această reprezentare nu este invariantă la modificarea scării, fapt pentru care am considerat scări multiple pentru regiunile detectate de procesul de segmentare. Pentru fiecare scară și pentru fiecare regiune calculăm vectorii corespunzători (funcțiile $N_s(t)$). În acest fel obținem o reprezentare care include și factorul scară.

$$N'_{kl}(s,t) = \frac{N_{kl}(s,t)}{\max(N_{kl}(s,t))} \quad (5.1)$$

unde :

$$\begin{aligned} N_{kl}(s,t) &= N_s(t) && \text{pentru regiunea } k \text{ la scala } l \\ N_{kl}(s,t) &&& \text{vectorul de caracterizare normalizat al regiunii} \\ k=1\dots N_{reg} &&& N_{reg} - \text{numărul total de regiuni} \\ l=1\dots N_{Sc} &&& N_{Sc} - \text{numărul total de scări} \end{aligned}$$

Probleme nu apar la construcția acestor reprezentări ci în utilizarea acestora în procesul de recunoaștere. Pentru fiecare regiune trebuie să stabilim mai întâi scara corectă înainte de a încerca compararea cu vectorii corespunzători. Tehnica folosită pentru aceasta este de a compara primele două maxime ale vectorului model cu valorile corespundente pentru regiunea care trebuie clasificată la diferite scări, scara aleasă fiind aceea pentru care distanța dintre cei doi vectori este cea mai mică.

Regula de decizie poate avea următoarea expresie:

$$\begin{aligned} (d(\text{Max}_1(N_{kl}(s,t)), \text{Max}_1(N_{mq}(s,t))) < d_{thres}) \\ \text{și} \\ (d(\text{Max}_2(N_{kl}(s,t)), \text{Max}_2(N_{mq}(s,t))) < d_{thres}) \end{aligned} \quad (5.2)$$

unde:

$$\begin{aligned} d() & - \text{o funcție distanță (de exemplu modulul diferențelor)} \\ \text{Max}_1 & - \text{primul maxim al unui vector} \\ \text{Max}_2 & - \text{al doilea maxim al unui vector} \\ m & - \text{identificatorul modelului comparat} \\ q & - \text{scara modelului} \end{aligned}$$

Deci pentru fiecare regiune putem avea trei, doi, sau un singur vector pe scară, decizia dacă textura unei regiuni este asemănătoare cu aceea a unei alte regiuni (model) va fi făcută după calcularea unei distanțe D_{rm} care implică distanțele pentru cei trei, doi, sau un vector :

$$D_{rm} = \sum d_i^2(k,m) ; i = 1..N ; N \in \{3,2,1\} \quad (5.3)$$

unde :

$$\begin{aligned} d(k,m) & - \text{distanța euclidiană dintre } N'_{kl}(s,t) \text{ și } N'_{mq}(s,t) ; \\ & \text{unde } l = q \text{ (aceeași scară) și a fost determinată de ecuația (5.2)} \end{aligned}$$

Dacă această distanță este mai mică decât o distanță maximă admisă atunci cele două regiuni sunt considerate că au aceeași textură color.

$$\text{AceeșiTextură}(K,m) = D_{rm} < D_{prag} \quad (5.4)$$

5.4. Aspecte de implementare

Așa cum putem vedea în figura Fig. 5.3-2, sistemul poate fi implementat astfel încât procesarea necesară pentru acest algoritm să se facă în paralel cu algoritmi existenți. În acest fel spațiul scării (scale space) ar putea fi construit în același timp cu calcularea histogrammei de cromaticitate și a procesului de segmentare a imaginii originale.

Spațiul scării este construit prin operații de convoluție efectuate asupra imaginii originale cu nuclee gaussiene cu dimensiuni crescătoare (1,2,4...) , rezultând în acest fel un număr de imagini netezite.

Pentru fiecare scară aplicăm algoritmul specificat în [3] pentru a calcula pentru fiecare regiune determinată, după procesul de segmentare a imaginii originale, vectorul corespunzător de caracterizare $N_s(t)$. Algoritmul a fost modificat astfel încât în locul nivelului de gri considerăm o funcție care produce un index în intervalul $[0, MaxId]$, unde $MaxId$ depinde de metoda de extensie aleasă (de exemplu pentru 3 vectori $MaxId=256$, pentru reducerea numărului de culori $MaxId=4096$, iar pentru metoda cromaticității $MaxId=1$, caz în care indexul este un număr real și nu un întreg).

După procesul de segmentare rezultă informațiile de adiacență a regiunilor, iar după aplicarea algoritmului precedent putem construi relativ simplu structura CTAG. Având construită această structură o putem folosi în locul structurii CAG în algoritmul de clasificare prezentat în [1].

În acest algoritm pentru fiecare regiune ca și un prim pas determinăm scara care va fi folosită în procesul de clasificare folosind ecuația (5.2). După ce am calculat scara putem calcula distanța dintre cei doi vectori normalizați (vectorul regiunii și vectorul model), și putem decide dacă regiunea are sau nu aceeași textură ca și modelul. Pentru rezultate de clasificare mai bune vom folosi mai multe modele în procesul de comparare și vom efectua decizia după calcularea distanțelor folosind ecuația (5.2). Procesul de clasificare a regiunilor va considera distanța minimă până la un model de textură dacă acea distanță satisface ecuația (5.4).

Un exemplu de aplicare a algoritmului pentru figura de mai jos (Fig. 5.4-1):



Fig. 5.4-1. Imagine „bandă” de test.

Această bandă de 95 x 25 pixeli a fost împărțită în blocuri de câte 25 x 25 pixeli, pentru care au fost calculate graficele $N_s(t)$, acestea fiind reprezentate în graficul de mai jos (Fig. 5.4-2).

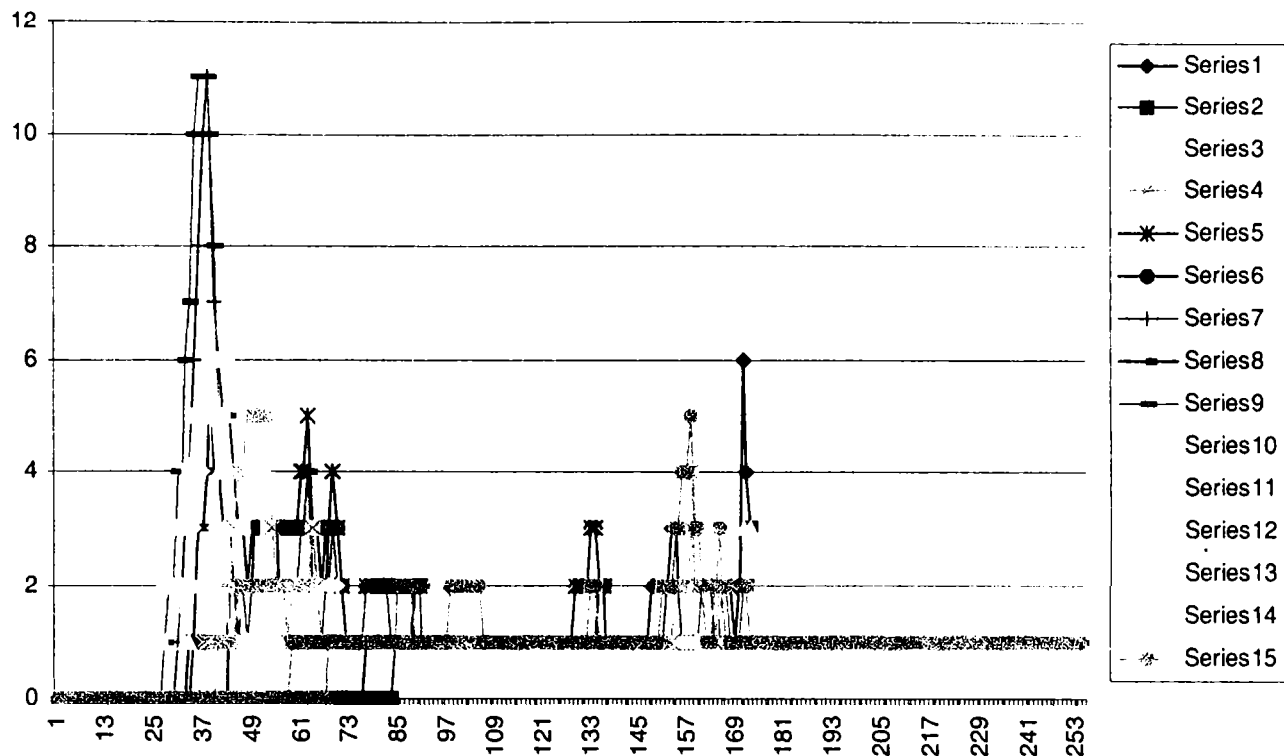


Fig. 5.4-2 Graficele $N_s(t)$ calculate pentru banda de imagine de mai sus pe 256 nivele.

Pentru a minimiza calculele se pot reduce nivelele de culori pentru care se calculează graficele $N_s(t)$. De exemplu următoarea figură (Fig. 5.4-3) reprezintă pentru aceeași bandă de imagine, același grafic ca mai sus, dar utilizând doar 16 nivele de culori.

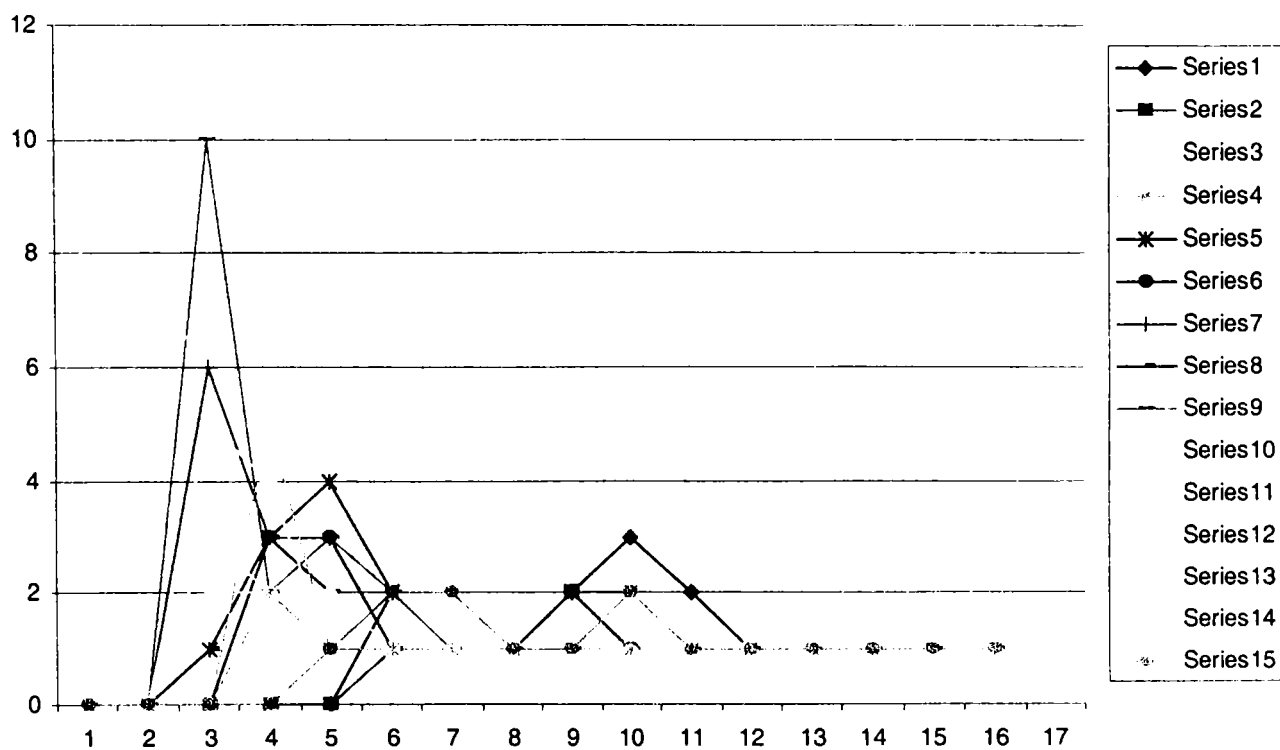


Fig. 5.4-3 Graficele $N_s(t)$ calculate pentru banda de imagine de mai sus pe 16 nivele.

6. Metode de detectare și recunoaștere a feței umane

6.1. Introducere

Recunoașterea feței umane joacă un rol important în interacțiunile noastre sociale. Oamenii sunt capabili să identifice un număr mare de fețe într-un mod aproape sigur, și psihologii sunt interesați în înțelegerea mecanismelor cognitive și perceptuale care stau la baza recunoașterii fețelor.

Fețele umane constă din aceleași elemente (nas, gură, ochi, etc) și recunoașterea unui individ se întâmplă, când noi distingem între aceleași configurații de bază. Fiecare individ trebuie deci să poată fi distins dintre alții datorită variațiilor elementelor de configurație de bază așa cum este arătat de Young și Bruce. Diamond și Carey au accentuat și mai mult studiul acestor elemente de configurare introducând conceptele de caracteristici izolate și interdependente (relaționale). Caracteristicile izolate sunt acele caracteristici care pot fi descrise prin ele însele, de exemplu culoarea ochilor, părul, etc. Trăsăturile interdependente sunt acele trăsături care descriu aspecte ale formei și în ce relație se află diferitele elemente faciale unele față de altele, de exemplu poziția nasului relativ la ochi, etc.

Descrierea trăsăturilor interdependente este utilă pentru deosebirea diferitelor persoane și de asemenea la determinarea diferitelor tipuri de comportamente sau expresii faciale ale aceleași persoane. Au fost efectuate mai multe cercetări cu scopul de a dezvolta un sistem capabil să distingă și să modeleze diferite tipuri de expresii faciale. Un astfel de sistem a fost Facial Action Coding System (FACS) propus de Ekman și Friesen și mai târziu folosit de Rydfalk la crearea unui set de date pentru descrierea unor fețe parametrizate. Sistemul a fost bazat pe faptul că fiecare expresie facială este rezultatul unei acțiuni musculare.

De-a lungul ultimilor 25 de ani au fost propuse diferite tehnici pentru detectarea și recunoașterea feței. Aceste tehnici pot fi clasificate în linii mari ca metode bazate pe caracteristici geometrice, metode bazate pe șabloane și mai recent metode bazate pe modele, așa cum este arătat de către Chellappa, Willson și Sirohey [34].

Multe dintre aceste abordări sunt constrânse de un număr de presupuneri asupra datelor de antrenare și de test, câteva dintre acestea fiind prezentate de Samal și Iyengar [35]. Cele mai întâlnite presupuneri sunt:

- Imaginile feței sunt fie vederi frontale fie din profil;
- Fața este într-o poziție verticală;
- Nu este tolerată nici o înclinare sau cel mult o înclinare mică;
- Nu se permite acoperirea parțială a feței, barbă, ochelari sau cicatrici;
- Luminozitatea și fundalul sunt controlate;
- Dimensiunea setului de test este limitat la cel mult câteva sute;
- Cele mai multe dintre cazurile de test sunt bărbați albi.

6.2. Metode bazate pe caracteristici geometrice

Abordările bazate pe caracteristicile geometrice [36] sunt cele mai timpurii abordări ale detecției și recunoașterii feței. În aceste sisteme caracteristicile faciale semnificative sunt detectate și distanțele dintre ele precum și alte caracteristici geometrice sunt combinate într-un vector de caracteristici care este folosit pentru reprezentarea unei fețe. Pentru a recunoaște o față mai întâi sunt obținuți vectorii de caracteristici a imaginilor de test și a imaginilor din baza

de date, după care este folosită o măsură de similaritate dintre acești vectori, cel mai adesea un criteriu de distanță minimă pentru a determina identitatea feței.

Ideea de comparare a măsurătorilor introdusă de Galton a fost în cele mai recente lucrări bazate pe calcularea unui set de caracteristici distincte din imaginea unei fețe. Aceste caracteristici sunt în general obținute sau din vederi frontale sau din profil.

Harmon și alți [36] au propus o abordare bazată pe caracteristici geometrice ale vederilor din profil ale fețelor umane. Caracteristicile erau calculate din câteva marcaje esențiale plasate automat de-a lungul profilului.

S-a început cu o bază de 112 subiecți cu trei imagini de antrenare și cu una de test pentru fiecare subiect, ei au reprezentat o față folosind un vector de caracteristici având 17 elemente. Rata de succes obținută a fost de 96%. Najman și alții [37] au utilizat de asemenea caracteristicile geometrice ale unei imagini din profil. Schița unui profil a fost construită utilizând între 8 și 100 de puncte de control. Testele au fost efectuate pe o bază de date de 10 subiecți, cu 31 de imagini de antrenare și 10 imagini de test pentru fiecare subiect. Trei metode de clasificare au fost încercate: analiza componentelor principale urmată de o discriminare quadratică, cel mai apropiat vecin de ordinul k și propagarea înapoi a gradientului obținându-se o rată de succes de aproximativ 90%.

În una din primele încercări de recunoaștere automată a feței, Kanade [38] a dezvoltat un sistem care extrage 16 caracteristici geometrice ale unei vederi frontale. El a folosit o bază de date de 20 de subiecți cu o imagine de antrenare și cu una de test pentru fiecare subiect, raportând o rată de recunoaștere de 75%.

Brunelli și Poggio [39] au implementat un sistem de recunoaștere bazat pe caracteristici geometrice folosind conceptele din lucrările lui Kanade. Sistemul a fost testat pe o bază de date de 47 de subiecți ajungându-se la o rată de recunoaștere de aproximativ 90%. O altă abordare bazată pe caracteristici ale vederilor frontale a fost implementată de Wong și alți [40] folosind diferite distanțe (ochi la ochi, ochi din dreapta și din stânga la nas, nas la marginea din stânga respectiv dreapta) ca și caracteristici.

În lucrările lor Goldstein și alții [41] au folosit un set de 34 de caracteristici mixte ale vederilor frontale și din profil (reduse la 22). Caracteristicile cuprindeau printre altele lungimea părului, textura părului, lungimea nasului, lățimea gurii și profilul bărbiei. Caracteristicile au fost notate pe o scară de la 1 la 5 localizarea făcându-se manual. Modelul a sugerat că pentru o populație de 25 de subiecți pentru identificare erau suficiente doar 6 caracteristici.

Așa cum este arătat de Brunelli și Poggio [39] abordările bazate pe șabloane au performanțe mai bune decât cele bazate pe caracteristicile geometrice.

6.3. Metode bazate pe șabloane

Abordările bazate pe șabloane reprezintă cea mai populară tehnică folosită pentru recunoașterea și detectarea fețelor. Spre deosebire de abordările bazate pe caracteristicile geometrice abordările bazate pe șabloane folosesc un vector de caracteristici care reprezintă întregul șablon facial spre deosebire de cele mai semnificative caracteristici faciale.

6.3.1. Metode bazate pe corelare

Metodele de corelare pentru detectarea și recunoașterea fețelor [42] sunt bazate pe calcularea coeficientului de intercorelare normalizat C_N [43] definită de :

$$C_N = \frac{E\{I_T T\} - E\{I_T\}E\{T\}}{\sigma(I_T)\sigma(T)} \quad (6.1)$$

unde I_T este imaginea care trebuie comparată cu șablonul T , $I_T T$ reprezintă produsul pixel cu pixel, E este operatorul de probabilitate, iar σ este devierea standard a zonei comparate. În

[39], [44], [42] autorii descriu o metodă bazată pe corelare atât pentru detectarea cât și recunoașterea fețelor din vederi frontale. Primul pas din aceste metode este determinarea poziției unor caracteristici semnificative, ca de exemplu ochii, nasul sau gura. Importanța detectării cu precizie a caracteristicilor faciale a condus la dezvoltarea unei varietăți de algoritmi de detectare a caracteristicilor faciale [45], [46], [47], [48], [49]. Metoda de detectare a caracteristicilor faciale propusă de Brunelli și Poggio [39], [44] folosește un set de șabloane pentru detectarea poziției ochilor, prin căutarea maximului valorilor absolute ale coeficientului de corelare normalizat al acestor șabloane în fiecare punct din imaginea de test. Pentru a trata problema variației de scară, au fost folosite un set de șabloane la diferite scări. Aceste probleme pot fi reduse în mod semnificativ folosind o corelare ierarhică, așa cum este propus de Burt în [50]. Pentru recunoașterea fețelor, șabloanele corespunzătoare caracteristicilor faciale semnificative ale imaginii de test au fost comparate, pe rând, cu șabloanele corespunzătoare ale tuturor imaginilor din baza de date, calculându-se un vector de grad de asemănări (una pentru fiecare caracteristică) folosind intercorelarea normalizată. Gradele de asemănare ale diferitelor caracteristici sunt integrate pentru a obține un scor global care este folosit pentru recunoaștere. Alte metode similare care folosesc corelația [42] sau statistici avansate au arătat acuratețea acestor metode dar și complexitatea lor.

6.3.2. Metode bazate pe analiza componentelor principale

6.3.2.1. Metoda fețelor proprii (*eigenfaces*)

Cea mai simplă versiune a metodelor bazate pe șabloane este obținută când întreaga imagine facială este folosită ca un singur șablon. O imagine de test este recunoscută prin calcularea distanței sale în termeni euclidiani față de șabloanele generate din imaginile din setul de antrenare selectând în final distanța cea mai mică. Procedul Karhunen-Loève dezvoltat de Kirby și Sirovich [51] și abordarea bazată pe analiza componentelor principale ale lui Turk și Pentland [52] sunt bazate pe această metodă simplă. Deosebirea este că șabloanele nu corespund imaginilor faciale originale ci proiecțiilor acestora într-un sistem de coordonate optimal. Setul de vectori de bază care alcătuiesc acest sistem de coordonate sunt vectorii proprii ai matricei de covarianță ai setului de fețe de antrenare, acești vectorii proprii fiind denumiți fețe proprii (*eigenfaces*). Spațiul alcătuit de vectorii proprii v_k unde $k = 1..K$, corespunzător celor mai mari K valori proprii ai matricei de covarianță, este denumit spațiul facial. O nouă imagine facială este transformată în componentele sale proprii prin proiecția asupra spațiului facial. Proiecțiile formează vectorul de caracteristici care descriu contribuția fiecărei fețe proprii în reprezentarea imaginii de intrare. Folosind această metodă Turk și Pentland [52] au raportat rate de recunoaștere de până la 96% pe o bază de date de 16 subiecți.

Metoda fețelor proprii a fost de asemenea utilizată pentru detectarea fețelor [53], [54] prin măsurarea distanței dintre fiecare tipar local dintr-o imagine de test și spațiul facial definit de fețele proprii.

6.3.2.2. Metoda caracteristicilor proprii (*eigenfeatures*)

În timp ce metoda clasică a fețelor proprii folosește coeficienții transformatei Karhunen-Loève, ai șabloanelor corespunzând întregii imagini a feței, în [55] Pentland și alți au introdus un sistem de detectare și recunoaștere care utilizează coeficienții transformatei Karhunen-Loève ai șabloanelor corespunzând caracteristicilor faciale semnificative, cum ar fi ochi, nas și gură. Pentru fiecare caracteristică facială se construiește un spațiu al caracteristicilor prin selectarea celor mai semnificative caracteristici proprii "eigenfeatures", care sunt vectorii proprii corespunzând celor mai mari valori proprii a matricei de corelație a caracteristicilor. Caracteristicile faciale semnificative au fost detectate utilizând distanța până la spațiul de caracteristici și selectând cea mai apropiată potrivire. Rezultatele asemănări

dintre șabloanele imagini de test și șabloanele imagini de antrenare sunt integrate într-un rezultat cumulativ care măsoară distanța dintre imaginea de test și imaginile setului de antrenare. Metoda a fost extinsă pentru detectarea caracteristicilor sub diferite vederi geometrice prin utilizarea fie a unui spațiu propriu bazat pe o vedere fie a unui spațiu propriu parametrizat.

6.3.3. Metoda bazată pe rețele neuronale

Șabloanele au fost folosite ca intrări ale sistemelor bazate pe rețele neuronale. Allinson și alți [60] au folosit un șablon de 32X32 pentru întreaga imaginea și două șabloane de 64X32 pentru regiunea ochilor respectiv gurii. Aceste șabloane au fost folosite ca și intrări în hărți de caracteristici cu reorganizare automată a lui Kohonen. Aceste hărți produceau o topologie care păstra structura șabloanelor de intrare. Hărțile erau folosite ca și intrări ale unui perceptron multistrat care realiza clasificarea.

Cotrell și Fleming [56] au folosit în sistemul lor de recunoaștere două rețele neuronale cu propagare înapoi (backpropagation). Prima fiind într-un mod de auto asociere rețea iar cea de-a doua într-un mod de clasificare. Rețeaua neuronală de auto asociere extrage automat caracteristici care sunt utilizate de rețeaua de clasificare. Vectorul de caracteristici rezultat este la fel ca cel produs de metoda fețelor proprii dacă rețeaua neuronală auto asociativă este liniară. Mai recent, Lawrence și alți [57] au propus o abordare bazată pe rețele neuronale hibride care combină mostre de imagini locale, o hartă cu reorganizare automată (self organising map –SOP) și o rețea neuronală convoluțională. SOP oferă un set de caracteristici care alcătuiesc o reprezentare mai compactă și robustă a mostrelor de imagini. Aceste caracteristici sunt preluate apoi de rețeaua neuronală convoluțională. Această arhitectură oferă parțial invarianță la translația, rotația, scalarea și deformarea feței. În [58], [59] autorii au introdus o rețea neuronală bazată pe decizie probabilistică (PDBNN) pentru detectarea și recunoașterea feței. Vectorul de caracteristici folosit consta din valorile de intensitate și de muchie obținute dintr-o regiune facială a imaginilor cu rezoluție mică din setul de antrenare. Regiunea facială conține ochii și nasul, dar exclude părul și gura. Două rețele PDBNN au fost antrenate cu acești vectori de caracteristici și apoi utilizate una pentru detectarea feței și cealaltă pentru recunoașterea feței.

6.4. Metode bazate pe modele

Recent cercetătorii din domeniul vederi artificiale au început să investigheze metodele bazate pe modele pentru recunoașterea și detectarea feței obținând câteva rezultate promițătoare. Spre deosebire de metodele bazate pe șabloane, abordările bazate pe modele permit o mai mare flexibilitate în sensul deformărilor faciale naturale și a condițiilor de iluminare. Această flexibilitate este rezultatul utilizării unui model matematic pentru incorporarea informațiilor din diferitele instanțe ale fețelor la diferite scări și orientări. În abordarea bazată pe modele în locul comparării vectorilor de caracteristici care reprezintă șabloane faciale pentru a determina identitatea unei fețe pentru recunoaștere sunt folosiți parametri modelului facial.

6.4.1. Metode bazate pe modele Markov ascunse

Modelele Markov ascunse (Hidden Markov Models – HMM) sunt un set de modele statistice folosite pentru a caracteriza proprietățile statistice ale unui semnal [61], [62]. HMM-urile au fost utilizate extensiv pentru recunoașterea vorbirii, unde datele sunt în mod natural unidimensionale, de-a lungul axei timpului. Modelul HMM bidimensional total conectat

echivalent, ar crea o problemă computațională foarte mare. Au fost realizate diverse încercări pentru a utiliza reprezentări multi-model care au condus la HMM-uri pseudo-bi-dimensionale [63]. Aceste modele sunt folosite mai ales în recunoașterea de caractere.

În [64] Samaria și alții au propus folosirea HMM-urilor unidimensionale continue pentru recunoașterea fețelor. Presupunând că fiecare față este în poziție frontal-verticală, caracteristicile trebuie să apară într-o ordine predictibilă, de ex. Frunte, ochi, nas etc. Această ordonare sugerează folosirea unui model de sus-în-jos, unde sunt permise doar tranzițiile dintre stări adiacente și numai într-o manieră de sus-în-jos [65]. Stările modelului corespund caracteristicilor faciale semnificative, ca de ex. Fruntea, ochii, nasul, gura și bărbia [66]. Secvența de observare O este generată dintr-o imagine $X \times Y$ folosind o fereastră de eșantionare de $X \times L$ cu o suprapunere de $X \times M$ pixeli. Fiecare vector de observație este un bloc de L linii. Există o suprapunere de M linii între observațiile succesive [67].

Fiind date c imagini pentru fiecare subiect din setul de antrenare, scopul etapei de antrenare este de a optimiza parametri HMM-ului pentru a descrie cel mai bine observațiile, în sensul maximizării probabilității observațiilor fiind dat un anumit model. Recunoașterea este efectuată prin compararea imaginii de test cu fiecare dintre modelele antrenate. Pentru a realiza acest lucru, imaginea este convertită la o secvență de observații, după care sunt calculate asemănările cu fiecare model de față. Modelul cu cea mai mare asemănare identifică fața necunoscută.

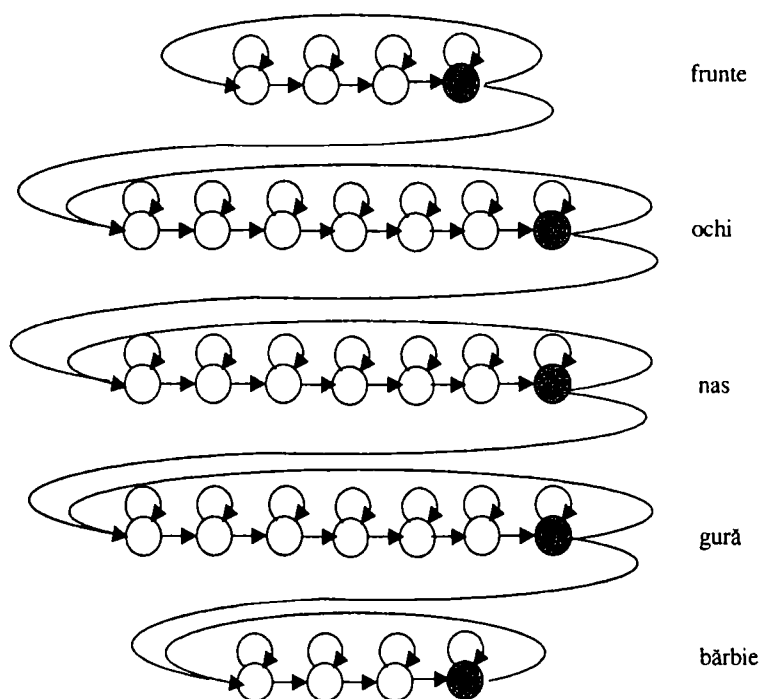


Fig. 6.4-1 Modelul HMM cu stări de sfârșit de linie

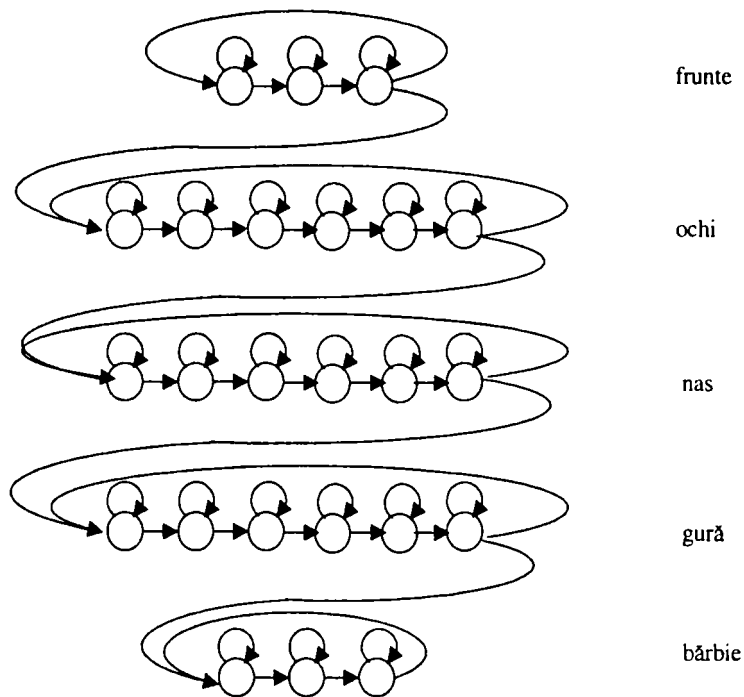


Fig. 6.4-2 Modelul HMM fără constrângeri

În [68] Samaria a crescut numărul de stări folosite pentru caracterizarea fiecăruia dintre caracteristicile faciale semnificative. Secvența de observare folosită cu acest model este obținută prin deplasarea unei ferestre dreptunghiulare de la stânga la dreapta, de sus în josul imaginii folosind intensitățile pixelilor extrași din fiecare fereastră ca și vectori de observație. Pentru a păstra structura bi-dimensională a datelor, a fost adăugat un bloc de marcaj la sfârșitul fiecărei linii din imagine și în plus a fost adăugată o stare de sfârșit de linie la structura fiecărui HMM orizontal Fig. 6.4-1. Stărilor sfârșit de linie le sunt permise două tranziții: una spre aceeași linie de stări și alta spre următoarea linie de stări. A fost găsit [68] că prin setarea deviației standard inițiale pentru stările de sfârșit de linie pe o valoare mică și a mediilor apropiate intensității blocului marcaj de sfârșit de linie, topologia stărilor a fost păstrată iar parametri stărilor sfârșit de linie au rămas neafecți după re-estimare. În aceeași lucrare, a fost arătat că rezultate similare au fost obținute pentru structura P2D-HMM fără constrângeri Fig. 6.4-2. Această structură permite tranziții la o stare corespunzând unei alte caracteristici faciale dintr-un bloc care nu este la sfârșitul unui rând, și astfel nu păstrează structura bi-dimensională a datelor. Rezultatele preliminare au arătat că pentru această structură, recunoașterea ajunge până la 95%, dar datorită dimensiunilor mari ale vectorilor de observație folosiți, sistemul necesită timpuri destul de mari pentru realizarea efectivă a recunoașterii.

6.5. Concluzii

În acest capitol au fost prezentate câteva dintre metodele de succes întâlnite în detectarea și recunoașterea feței. Nu se poate realiza o comparație efectivă între aceste metode deoarece ele au fost aplicate pe baze de imagini diferite. În general metodele bazate pe șabloane au arătat o acuratețe mare pentru medii controlate, gen imagine de dimensiune fixă. Deși aceste dificultăți ar putea fi înlăturate cu costul creșterii complexității sistemului abordările bazate pe modele mai precise cele bazate pe HMM reprezintă o metodă mai potrivită datorită invarianței la dimensiune pentru recunoașterea și detectarea fețelor. Unul din dezavantajele lor fiind însă timpul mare de calcul, dar care așa cum este arătat de Nefian [70] [70] poate fi înlăturat.

7. Detectarea și recunoașterea feței folosind modelul Markov ascuns încapsulat

7.1. Introducere

Hidden Markov Models (HMM) a fost utilizat cu succes pentru recunoașterea vocii și a mișcării acolo unde datele ce trebuiau modelate au fost unidimensionale (1D). Au existat încercări de a folosi acest model unidimensional HMM și pentru recunoașterea feței însă reușitele au fost parțial încununate de succes din cauza faptului că imaginile sunt bidimensionale (2-D). În acest capitol este prezentată o nouă abordare pentru recunoașterea feței folosind un HMM încapsulat și se face o comparație între această nouă abordare cu metoda valorilor proprii pentru recunoașterea feței, și cu alte metode bazate pe HMM. În mod sigur, un HMM încapsulat are performanțe la fel de bune sau mai bune decât alte metode, având totodată și o complexitate redusă de calcul.

Cele mai semnificative trăsături faciale ale unei imagini frontale includ părul, fruntea, ochii, nasul, gura și bărbia.

Mai mult, aceste trăsături survin într-o ordine naturală, de sus în jos, chiar dacă imaginea suferă mici rotații în planul imaginii, și/sau rotații în planul perpendicular al imaginii. Prin urmare, imaginea feței poate fi modelată folosind modelul unidimensional HMM prin atribuirea fiecărei regiuni a unei stări Fig. 7.1-1. În acest model, stările însăși nu sunt direct observabile.

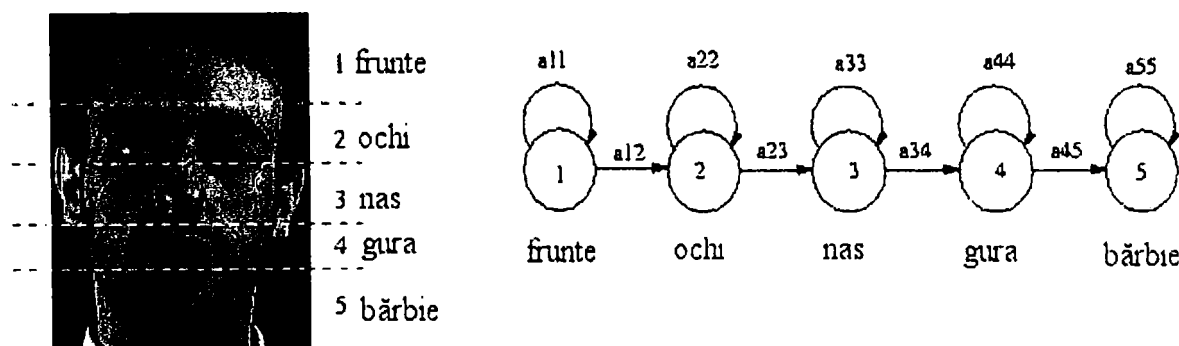


Fig. 7.1-1. HMM pentru recunoașterea feței

Ce este remarcabil sunt vectorii de observare care sunt statistic dependenți de starea modelului HMM. Acești vectori sunt obținuți în cele L rânduri care sunt extrase secvențial dinspre partea sus a imaginii spre partea de jos. Datorită faptului că lungimea fiecărui rând este fixă, și înălțimea imaginii feței este proporțională cu lățimea, acest model HMM este limitat pentru imagini fixe ale feței. Acest model unidimensional HMM realizează o rată de recunoaștere de până la 85%.

Acest model unidimensional a fost extins de Samaria [68] la o structură care este referită ca și 2D-HMM, prin adăugarea unui marker (*marker block*) la sfârșitul fiecărei linii a imaginii, și introducând în plus o stare de sfârșit de linie (*end-of-line-state*) la sfârșitul fiecărei linii orizontale a HMM așa cum se arată în figura Fig. 6.4-1. Starea de sfârșit de linie permite două tranziții posibile: una înapoi la începutul aceluiași rând, și cealaltă la următorul rând de stări. Prin setarea standardului inițial de deviație al stării de sfârșit de linie să fie închis la zero, și media aproape de intensitatea blocului marcator de sfârșit de linie, topologia stării a fost conservată, și parametrii stărilor de sfârșit de linie rămân nealterați după reestimare. Samaria [68] a considerat un model 2D-HMM în care au fost îndepărtate blocurile de sfârșit de linie așa cum este ilustrat în figura Fig. 6.4-2. Spre deosebire de modelul

anterior, această topologie permite tranziții la o nouă superstare de la un bloc care nu este la sfârșitul rândului și prin urmare nu păstrează structura bidimensională a datelor. Cu toate acestea, Samaria a relatat rezultate similare de recunoaștere pentru ambele modele, care erau între 90% și 95%.

În acest paragraf se descrie o nouă abordare a recunoașterii feței folosind un HMM încapsulat așa cum a fost acesta introdus de către Kuo și Agazzi pentru recunoașterea caracterelor. Spre deosebire de abordările anterioare ale HMM pentru recunoașterea feței, care folosesc pixeli de intensitate pentru a forma vectorii de observare, un HMM încapsulat folosește vectori de observare care pot fi formați de exemplu din coeficienții bidimensionali ai transformatei cosinus discretă (Discret Cosine Transform 2D-DCT). Comparate cele două metode bazate pe modele, HMM unidimensional și HMM încapsulat, metoda care folosește un sistem HMM încapsulat oferă mai multă flexibilitate în cazul recunoașterii feței, și poate fi folosită mai eficient în scalarea unor sisteme invariante.

7.2. HMM încapsulat

Un model HMM unidimensional este un lanț Markov cu un număr finit de stări neobservabile. Cu toate că stările Markov nu sunt direct observabile, fiecare stare are o distribuție de probabilitate asociată cu seturile de observații posibile. Astfel, când HMM este în starea i , ieșirea observabilă este determinată în conformitate cu o funcție de densitate a probabilității, condiționată, deseori Gaussiană sau o compusă Gaussiană. Ce este necesar statistic să caracterizeze un HMM este o matrice cu probabilitățile tranzițiilor de stări, distribuția probabilității stării inițiale, și un set de funcții de densitate a probabilității asociate cu observațiile pentru fiecare stare.

Un model unidimensional HMM poate fi generalizat, pentru ai da aparența unei structuri bidimensionale, permițând fiecărei stări din modelul unidimensional HMM să fie un HMM. În acest fel, modelul HMM constă dintr-un set de super stări (*super state*), împreună cu un set de stări imbricate. Super stările pot fi folosite pentru modelarea datelor bidimensionale de-a lungul unei direcții, iar cu HMM încapsulat modelând datele de-a lungul celeilalte direcții. Acest model diferă față de un adevărat model HMM bidimensional deoarece tranziții dintre stările din diferite super stări nu sunt permise. Prin urmare, acesta este denumit un sistem HMM încapsulat Fig. 7.2-1. Elementele unui sistem HMM încapsulat sunt:

- Numărul de super stări, N_0 , și setul de super stări, $S_0 = \{S_{0,i} \mid 1 \leq i \leq N_0\}$.
- Distribuția inițială a super stărilor, $\Pi_0 = \{\pi_{0,i}\}$, unde $\pi_{0,i}$ sunt probabilitățile de a fi în super starea i la timpul zero.
- Matricea probabilității tranziției dintre super stări:

$$A_0 = \{a_{0,ij}\}$$

unde $a_{0,ij}$ este probabilitatea de tranziție dintr-o super stare i într-o super stare j .

- Parametrii HMM-urilor încapsulate, care includ:
 - Numărul de stări încapsulate din super starea k , $N_1^{(k)}$, și setul de stări încapsulate, $S_1^{(k)} = \{S_{1,i}^{(k)}\}$.
 - Distribuția stării inițiale $\Pi_1^{(k)} = \{\pi_{1,i}^{(k)}\}$, unde $\pi_{1,i}^{(k)}$ reprezintă probabilitatea de a fi în starea i din super starea k la timpul zero.
 - Matricea probabilității tranziției dintre stări:

$$A_1^{(k)} = \{a_{1,jk}^{(k)}\}$$

care specifică probabilitatea de tranziție din starea k în starea j .

- În final există o matrice a probabilității stărilor,

$$B^{(k)} = \{ b_i^{(k)}(O_{t_0,t_1}) \}$$

pentru setul de observații, unde O_{t_0,t_1} reprezintă vectorul de observație din rândul t_0 și coloana t_1 . Într-un HMM cu *densitate continuă*, stările sunt caracterizate de funcții de densitate continue. Funcția de densitate a probabilității care este de obicei folosită este o combinație finită de forma:

$$b_i^{(k)}(O_{t_0,t_1}) = \sum_{m=1}^M c_{im}^{(k)} N(O_{t_0,t_1}, \mu_{im}^{(k)}, U_{im}^{(k)})$$

unde $1 \leq i \leq N_1^{(k)}$, $c_{im}^{(k)}$ este coeficientul de combinație pentru a m -a combinație din starea i a super stării k .

$N(O_{t_0,t_1}, \mu_{im}^{(k)}, U_{im}^{(k)})$ este o funcție de densitate a probabilității de tip Gaussian cu vectorul medie $\mu_{im}^{(k)}$ și matricea de covarianță $U_{im}^{(k)}$.

Fie $\Lambda^{(k)} = \{ \Pi_1^{(k)}, A_0, \Lambda \}$ un set de parametri care definesc super starea k . Utilizând o notație, un HMM încapsulat a fost definit ca un triplet:

$$\lambda = (\Pi_0, A_0, \Lambda)$$

unde $\Lambda = \{ \Lambda^{(1)}, \Lambda^{(2)}, \dots, \Lambda^{(N_0)} \}$.

Cu toate că este mult mai complex decât HMM unidimensional, un HMM încapsulat este mult mai potrivit pentru date bidimensionale, și are o complexitate proporțională cu suma pătratelor numărului de stări $\sum_{k=1}^{N_0} (N_1^{(k)})^2$. Structura stărilor modelului facial și probabilitățile de tranziție nenule ale HMM încapsulat sunt arătate în figura Fig. 7.2-1.

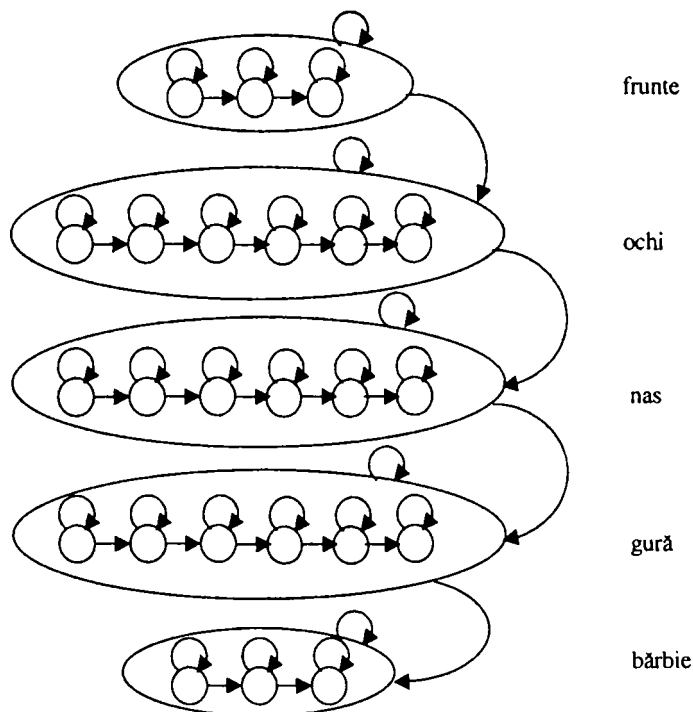


Fig. 7.2-1. Un HMM încapsulat pentru recunoașterea feței

Fiecare stare din HMM general de sus în jos este asignată unui HMM de la stânga la dreapta. Acest model este potrivit pentru imagini faciale deoarece ține cont de o caracteristică facială importantă: fețele frontale păstrează aceeași structură de super stări de sus în jos, și de asemenea aceeași structură de stări de la stânga la dreapta în interiorul fiecăreia dintre aceste super stări. Comparat cu alte structuri, un HMM încapsulat are următoarele avantaje:

1. Complexitatea este redusă atât în termeni de antrenare cât și de recunoaștere.
2. Pot fi obținute estimări inițiale mai bune ale parametrilor modelului.
3. Structura bidimensională a datelor este conservată într-un mod natural fără folosirea unor cadre suplimentare sau stări de sfârșit de linie care ar crește complexitatea modelului.

Stările unui HMM încapsulat sunt descrise printr-o singură funcție de densitate a probabilității de tip Gaussian

$$b_i^{(k)}(O_{t_0:t_1}) = N(O_{t_0:t_1}, \mu_i^{(k)}, U_i^{(k)})$$

și de matricea de covarianță care se presupune a fi diagonală.

7.2.1. Vectorii de observație

Secvența de observare este generată folosind tehnica arătată în figura Fig. 7.2-2, unde o fereastră $P \times L$ scanează imaginea de la stânga la dreapta și de sus în jos. Suprapunerea dintre ferestrele adiacente este M linii pe verticală și Q coloane pe orizontală.

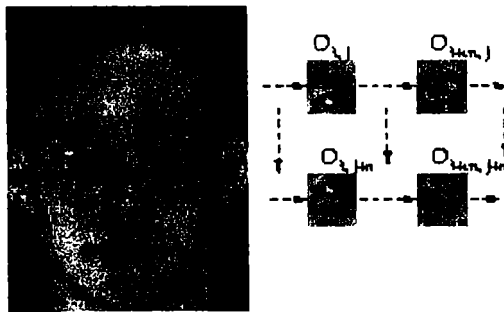


Fig. 7.2-2. Parametrizarea imaginii faciale și extragerea blocurilor

În [68] Samaria a considerat că vectorii de observație constau din valorile tuturor pixelilor din fiecare bloc în parte, astfel rezultând dimensiunea vectorului de observație de $L \times P$. Folosirea valorilor pixelilor ca vectori de observație are două dezavantaje importante. Primul, valorile pixelilor nu reprezintă trăsături robuste deoarece acestea tind să fie sensibile atât la zgomotul cât și la rotațiile sau deplasările imaginii, și schimbările în iluminatie. Al doilea, dimensiunea mare a vectorilor de observare conduce la o complexitate de calcul ridicată a antrenării și a etapelor de recunoaștere. Acesta poate fi critic pentru un sistem de recunoaștere a feței care operează pe o baza de date mare, sau când sistemul de recunoaștere este folosit pentru aplicații în timp real.

În [70] Nefian a format vectorii de observare ai HMM-ului încapsulat din coeficienții 2D-DCT ai fiecărui bloc al imaginii. Proprietățile de comprimare și decorelare ale parametrilor 2D-DCT pentru imagini naturale îi fac adecvați pentru folosirea lor ca vectori de observație. În special coeficienții din interiorul unei ferestre dreptunghiulare definite pe cele mai joase frecvențe din domeniul 2D-DCT, unde se află majoritatea energiei imaginii, au fost folosiți ca vectori de observație. Folosind coeficienții 2D-DCT în locul valorilor pixelilor se reduce considerabil dimensiunea vectorilor de observație, scăzând astfel complexitatea sistemului de recunoaștere. Nefian în experimente a folosit valorile, $L=8$, $P=10$ iar pentru

vectorii de observație șase coeficienți ai 2D-DCT. Astfel, dimensiunea vectorilor de observație a fost redusă de peste 13 ori în comparație cu metoda care folosește intensitățile pixelilor ca vectori de observație.

7.2.2. Antrenarea modelelor faciale

Fiecare individ din baza de date este reprezentat printr-un HMM încapsulat. Un set de imagini reprezentând imagini diferite ale aceleași fețe au fost folosite pentru antrenare. Vectorii de observație extrași din fiecare bloc au fost folosiți pentru a antrena modelele după cum se arată în figura următoare (Fig. 7.2-3).

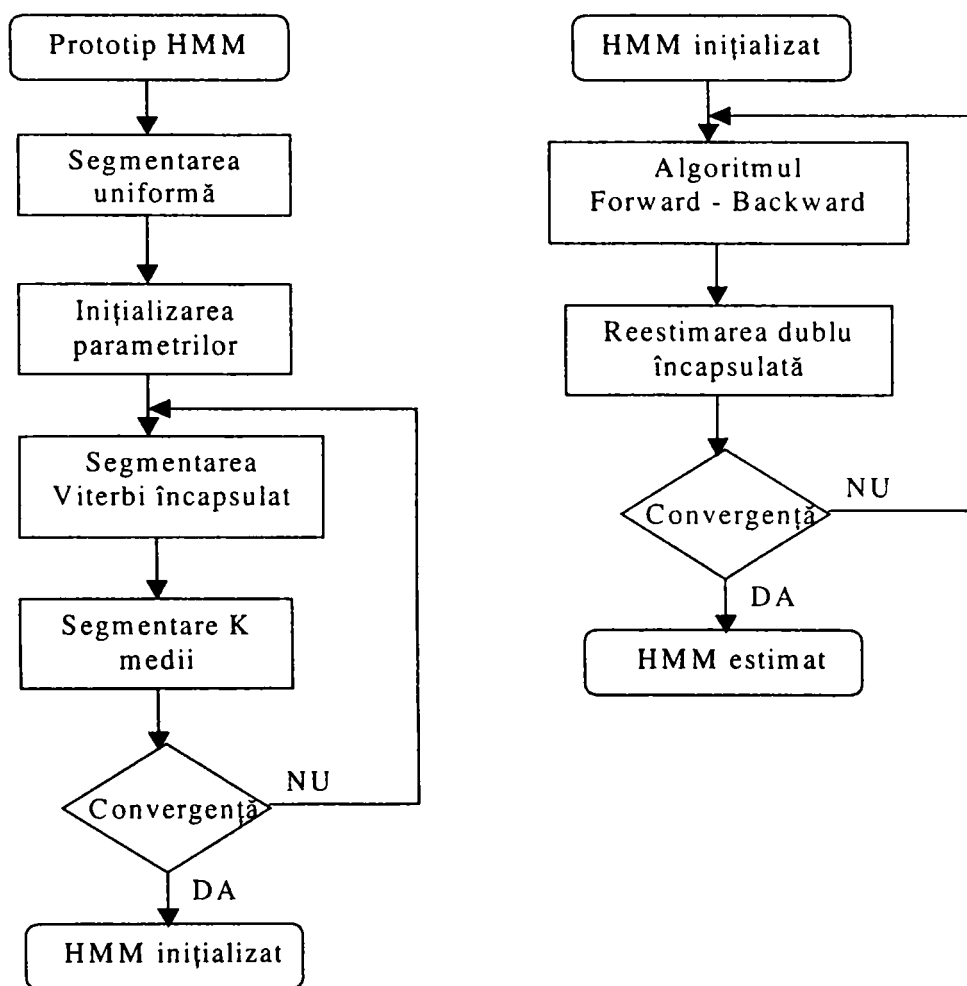


Fig. 7.2-3 Schema de antrenare

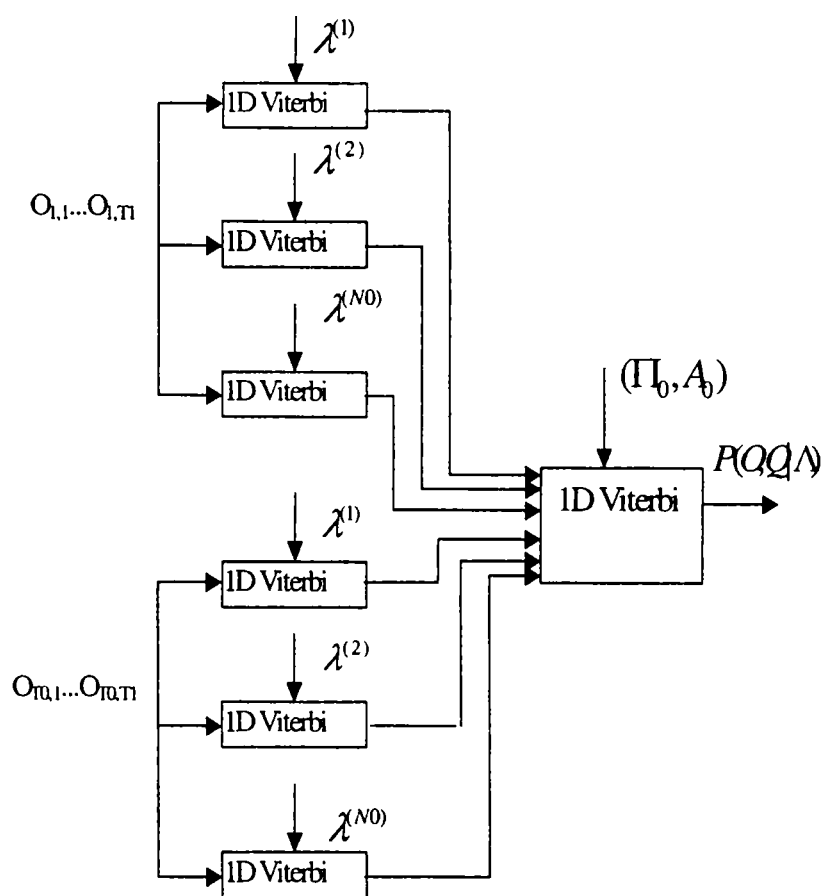


Fig. 7.2-4. Algoritmul Viterbi dublu încapsulat

1. În conformitate cu numărul de super stări, numărul de stări din fiecare super stare, și structura de sus în jos și de la stânga la dreapta a unui prototip HMM încapsulat, datele sunt uniform segmentate pentru a obține estimările inițiale a parametrilor modelului. În primul rând observațiile de sus în jos ale întregului HMM sunt segmentate în N_0 super stări verticale, apoi datele corespunzătoare fiecăreia dintre aceste super stări sunt uniform segmentate de la stânga la dreapta în $N_1^{(k)}$ stări.
2. La următoarea iterație, segmentarea uniformă este înlocuită de un algoritm de segmentare Viterbi dublu încapsulat Fig. 7.2-4.

Algoritmul de segmentare Viterbi dublu încapsulat, ilustrat în figura Fig. 7.2-4, constă din următorii pași. Primul, segmentarea Viterbi este aplicată fiecărui rând a imaginii, și probabilitățile $P(O_{t_0,1} \dots O_{t_0,T_1}, q_{1,1}^{(t_0)} \dots q_{1,T_1}^{(t_0)} | \lambda^{(k)})$, $1 \leq k \leq N_0$ sunt calculate, unde $q_{1,t_1}^{(t_0)}$, $1 \leq t_1 \leq T_1$ reprezintă starea unei super stări asignată observației O_{t_0,t_1} . Probabilitățile stărilor și observațiile dintr-un rând, fiind dat modelul de super stare, obținut din segmentarea Viterbi, reprezintă probabilitățile super stării. Probabilitățile super stărilor împreună cu probabilitățile tranziției a super stărilor A_0 și a probabilităților super stării inițiale Π_0 sunt folosite pentru a realiza segmentarea Viterbi de sus în jos și la determinarea:

$$P(O_{1,1} \dots O_{1,T_1}, \dots, O_{T_0,1} \dots O_{T_0,T_1}, q_{0,1} \dots q_{0,T_0} | \lambda)$$

Sau folosind o notație prescurtată $P(O, Q | \lambda)$. q_{0,t_0} , $1 \leq t_0 \leq T_0$ sunt super stările corespunzătoare rândului t_0 .

3. Parametrii modelului sunt estimați folosind o extensie a algoritmului de segmentare k -medii [61] la două dimensiuni. Așadar parametrii modelului sunt obținuți conform relației:

$$a_{1,ij}^{(k)} = \frac{\text{numarul de tranzitii de la } S_{1,i}^{(k)} \text{ la } S_{1,j}^{(k)}}{\text{numarul de tranzitii din } S_{1,i}^{(k)}}$$

$\mu_i^{(k)}$ = modelul mediei vectorului i din super starea k

$U_i^{(k)}$ = modelul matricii de covarianță a vectorilor din starea i a super stării k

$$a_{0,ij} = \frac{\text{numarul de tranzitii de la } S_{0,i} \text{ la } S_{0,j}}{\text{numarul de tranzitii din } S_{0,i}}$$

4. Iterația se oprește, și HMM-ul este inițializat, când asemănarea dintre segmentările Viterbi în iterații consecutive este mai mică decât un prag.
5. Parametrii HMM încapsulat sunt în continuare reestimați folosind procedura „forward backward” și o procedură de reestimare pentru a maximiza probabilitatea observațiilor dat fiind un model facial. Procedura forward și backward împreună cu ecuațiile de reestimare sunt prezentate de către Nefian în [70].
6. Iterațiile se opresc și HMM-ul încapsulat este antrenat când probabilitatea observațiilor, fiind dat modelul, în iterații consecutive este mai mică decât un prag specificat.

7.2.3. Detectarea feței

Procedura de antrenare descrisă mai sus este folosită atât pentru detectarea feței cât și pentru recunoașterea ei. Singura diferență dintre antrenarea pentru recunoaștere și cea pentru detecție constă în imaginile care sunt folosite în setul de antrenare. Pentru detectarea feței imaginile de antrenare reprezintă fețe a diferite persoane luate sub diferite condiții de iluminare și mici devieri față de o vedere frontală. Imaginile de asemenea constau din diferite expresii faciale, coafuri și reprezintă atât bărbați cât și femei de diferite vârste și rase. Toate aceste imagini din setul de antrenare sunt folosite pentru a antrena un singur model facial.

Structura HMM încapsulată permite o implementare eficientă a unui sistem de detecție facială folosind algoritmul Viterbi dublu încapsulat.

Un sistem de detecție a feței ar trebui să poată localiza toate fețele dintr-o imagine, atât a bărbaților cât și a femeilor independent de apariția, rasa, vârsta, scara, orientarea, fundalul sau iluminarea lor. Așa cum este arătat de Nefian în [70] un HMM încapsulat permite o mai mare flexibilitate în cazul deformărilor spațiale atât în plan orizontal cât și vertical. Sistemul de detecție a feței este ilustrat în figura Fig. 7.2-5.

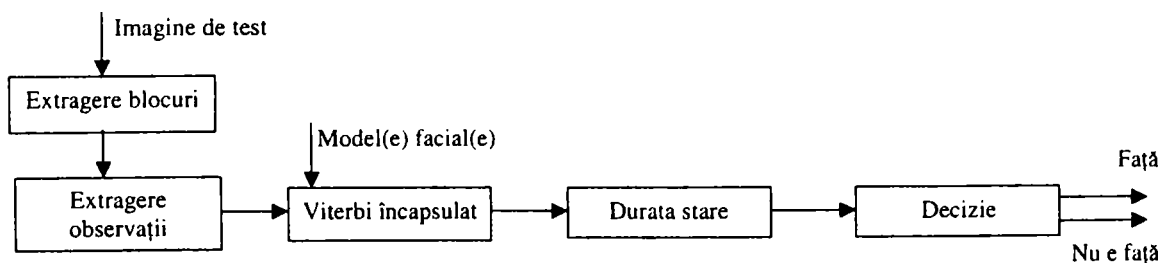


Fig. 7.2-5. Detectarea feței folosind HMM încapsulat.

Acest sistem calculează probabilitatea existenței unei fețe pentru toate tiparele dreptunghiulare dintr-o imagine. Asemănarea cu o față a fiecărui tipar dreptunghiular este dată de asemănarea Viterbi dublu încapsulată calculată pentru vectorii de observație din cadrul tiparului dreptunghiular, ponderat cu un factor de corecție durată a stării. Tiparele dreptunghiulare pentru care asemănarea cu o față depășește un prag stabilit sunt considerate ca și fețe valide. Complexitatea calculului poate fi redusă semnificativ așa cum este arătat în [70] ținând cont de suprapunerea tiparelor care rezultă într-un număr mare de vectori de observație comuni și care pot fi calculați printr-o singură trecere a unei ferestre dreptunghiulare $L_x \times L_y$ având o suprapunere de P_x respectiv P_y , exact ca și blocurile folosite pentru antrenare (Fig. 7.2-6). Fiind dată o imagine de dimensiune $W \times H$ perechile $(W_m \times H_m)$ și $(W_M \times H_M)$ reprezintă lățimea și înălțimea a celei mai mici respectiv a celei mai mari fețe pe care dorim să o găsim într-o imagine, Fig. 7.2-7.

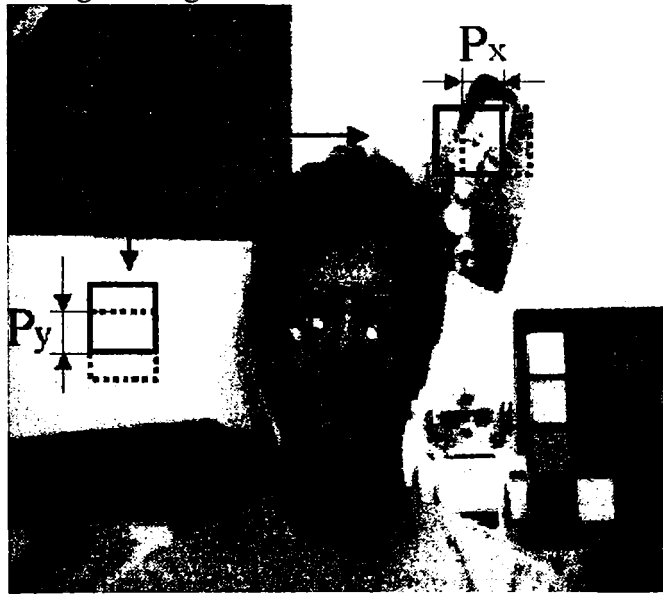


Fig. 7.2-6. Extragerea de blocuri pentru detectarea feței folosind HMM încapsulat

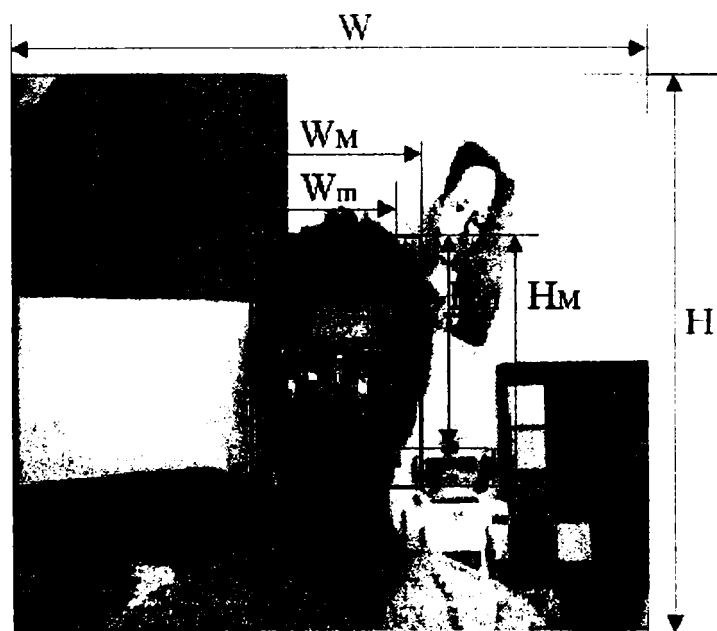


Fig. 7.2-7. Parametrizarea imaginii pentru detectarea feței folosind HMM încapsulat

Pentru a detecta o față de lățime W_f și înălțime H_f , unde $W_m \leq W_f \leq W_M$ și $H_m \leq H_f \leq H_M$, numărul de blocuri care sunt extrase pe orizontală, T_x și pe verticală T_y este:

$$T_x = \frac{W - L_x}{L_x - P_x} + 1 \quad (7.1)$$

$$T_y = \frac{W - L_y}{L_y - P_y} + 1 \quad (7.2)$$

Pentru această metodă numărul de blocuri extrase este independentă de dimensiunea feței de detectat. Vectorii de observație sunt obținuți fie din coeficienți 2D-DCT fie din coeficienți KLT extrași din blocurile de imagine folosind aceeași tehnică ca și pentru antrenare. Pe lângă aceste metode, în lucrarea de față, propun folosirea coeficienților CTAG în obținerea vectorilor de observație. O idee de dezvoltare a modelului ar fi combinarea acestor coeficienți cu coeficienții DCT, dând astfel vectori de observație care țin cont și de caracteristicile texturii blocurilor extrase, crescând astfel eficiența detecției.

Următorul pas al algoritmului de detecție a feței este acela de a calcula asemănarea cu o față a fiecărui tipar dreptunghiular folosind algoritmul Viterbi dublu încapsulat. Complexitatea calculelor pentru asemănare poate fi redusă semnificativ așa cum este arătat de Nefian în [70] ținând cont de suprapunerea verticală mare a tiparelor dreptunghiulare. În momentul calculării scorului Viterbi dublu încapsulat pentru un tipar dreptunghiular de lățime W_f înălțime H_M și având colțul din stânga sus la (x, y) , rezultatele de asemănare, fiind date modele faciale, sunt calculate implicit pentru toate tiparele dreptunghiulare având același colț stânga sus lățime și orice înălțime H_f , unde $1 \leq H_f \leq H_M$. Calcularea gradului de asemănare cu o față pentru imagini la scări diferite folosind același model facial este un avantaj important al acestei abordări față de abordările bazate pe șabloane sau abordările HMM standard.

Timul de detecție fost calculat de către Nefian [70], și poate fi redus și mai mult printr-o implementare paralelă a algoritmului Viterbi dublu încapsulat.

Acuratețea detecției poate fi mărită incluzând durata stării în modelul HMM de sus în jos. Durata d_i a super stării i este modelată folosind o distribuție Poisson așa cum este arătat de către Nefian în [70], neafectând semnificativ complexitatea sistemului.

Gradele de asemănare obținute pentru secvențele de observare corespunzând fiecărui tipar dreptunghiular sunt normalizate prin medierea gradului de asemănare cu dimensiunea matricii secvenței de observare. Mai departe gradele de asemănare obținute pentru fiecare tipar dreptunghiular din imagine sunt comparate cu un prag iar tiparele care au un grad de asemănare care depășește acest prag sunt posibile fețe candidat. Pentru a elimina „alarmele false” o față posibilă candidat reprezintă o localizare validă a unei fețe dacă gradul său de asemănare este mai mare decât gradele de asemănare ale tuturor candidaților din vecinătate.

Comparat cu metodele bazate pe șablon sau cele HMM standard pentru detectarea feței această abordare are următoarele avantaje:

- Este mai flexibilă în raport cu variații de scară și deformări naturale, atât pe plan vertical cât și orizontal.
- Permite o implementare eficientă a algoritmului de detecție a feței datorită împărțirii șablonului sau benzilor orizontale în blocuri de imagine care sunt procesate pentru a obține vectori de observație.
- Permite o implementare mai rapidă atât pentru detecție cât și pentru recunoaștere datorită complexității reduse a algoritmului Viterbi dublu încapsulat.
- Permite includerea caracteristicilor texturii blocurilor folosite pentru extragerea vectorilor de observație, măbind acuratețea detecției.
- Poate fi implementat folosind o arhitectură paralelă reducând astfel considerabil complexitatea atât a detecției cât și a recunoașterii.
- Păstrează structura bidimensională a datelor fără a folosi structuri speciale.

7.2.4. Recunoașterea feței

Spre deosebire de antrenarea pentru detecție, la antrenarea pentru recunoaștere, fiecărui individ din baza de imagini îi este asociat un model HMM încapsulat. Pentru antrenarea fiecăruia din aceste modele sunt folosite câte un set de imagini reprezentând diferite instanțe ale aceleiași persoane având diferite expresii, coafuri, cu sau fără ochelari.

După extragerea vectorilor de observație corespunzători imaginilor de test a feței, fiind dat un model facial HMM încapsulat, este calculată probabilitatea secvenței de observare printr-un algoritm Viterbi dublu încapsulat. Modelul cu cea mai mare asemănare este selectat și acest model dezvăluie identitatea feței necunoscute (Fig. 7.2-8).

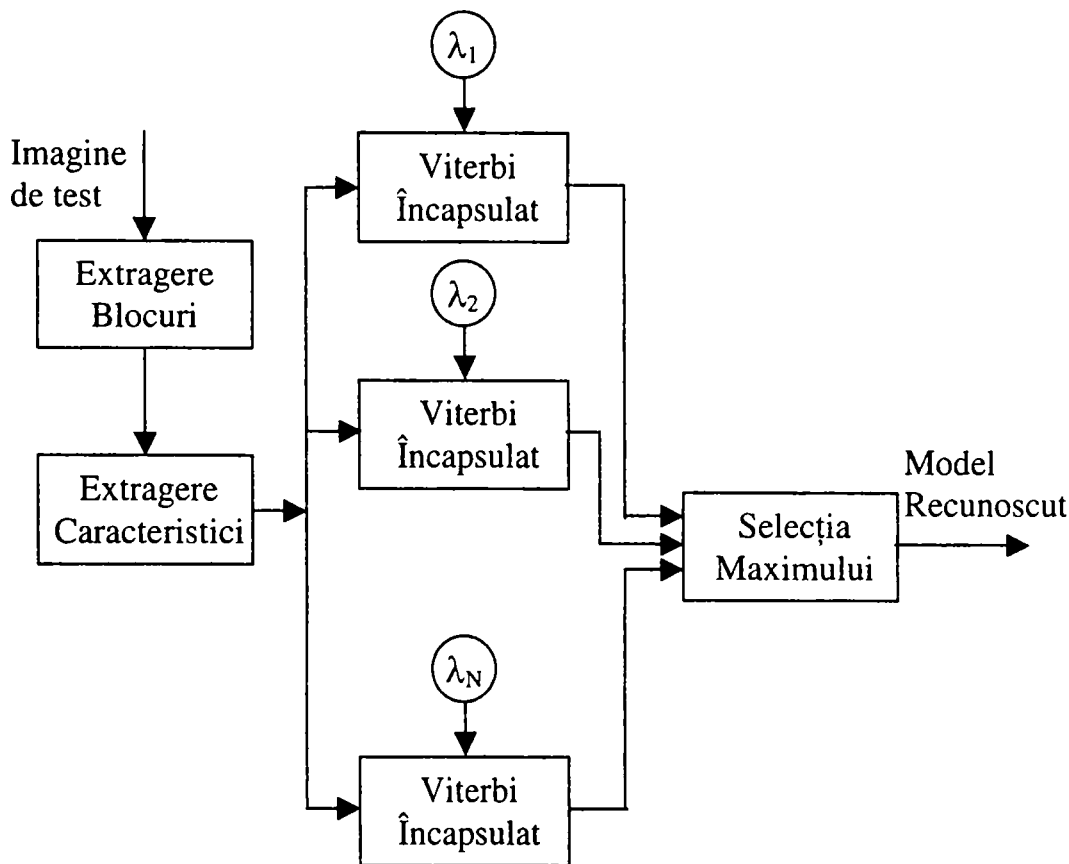


Fig. 7.2-8 Schema pentru recunoaștere de fețe folosind HMM încapsulat

Sistemul de recunoaștere al feței a fost testat pe bazele de date din laboratoarele firmei Olivetti Research Ltd (400 de imagini a 40 de indivizi la o rezoluție de 92x112 pixeli). Jumătate din imagini au fost folosite în antrenare și cealaltă jumătate au fost folosite pentru testare. Baza de date conține imaginile feței unor oameni de diferite vârste, atât bărbați cât și femei, arătând diferite expresii faciale, coafuri și purtători de ochelari sau nu. În aceeași bază de date performanța de recunoaștere cu un HMM unidimensional a fost în jur de 85%. Rata de recunoaștere a metodei "feței proprii" care depinde de numărul de fețe proprii folosite, variază de la 73% cu mai puțin de 5 fețe proprii până la aproximativ 90% când sunt folosite 200 fețe proprii. Performanța de recunoaștere pentru 2D-HMM al lui Samaria [68] depinde de structura și mostrele folosite de la 90% la 95%. Cu toate acestea, datorită dimensiuni mari a vectorilor de observație sistemul necesită un timp destul de mare pentru a identifica o anumită persoană.

Acuratețea sistemului prezentat de Nefian [70] crește la 98% folosind vectori de observare bazați pe DCT sau KLT, în timp ce timpul de recunoaștere necesar pentru identificarea unei fețe scade în mod considerabil față de sistemul lui Samaria. Eficiența acestui sistem se datorează atât alegerii unui vector de observare mai eficient, cât și folosirii

unei structuri eficiente a HMM. În timp ce este evident că o dimensiune mică a vectorului de observație reduce complexitatea atât a antrenării cât și a recunoașterii, eficiența structurii HMM încapsulat este demonstrată în [70]. În plus HMM încapsulat permite o implementare paralelă a algoritmului de decodare (algoritmul Viterbi dublu încapsulat) ceea ce duce la o reducere considerabilă a timpului de recunoaștere.

Metoda propusă în această lucrare, și anume calcularea elementelor vectorilor de observație pe baza coeficienților CTAG, conduce la performanțe mai mici decât cele raportate de Samaria [68] și Nefian [70], ajungând până la 83%, depinzând de parametri aleși.

Timpul de recunoaștere este unul intermediar între timpii obținuți aplicând metodele lui Samaria și respectiv Nefian. Acest lucru se poate explica pe baza faptului că dimensiunea vectorilor de observație rezultați pe baza acestei metode este mai mică decât a celor corespunzătorii metodei abordate de Samaria, dar mai mare decât cele rezultate prin folosirea coeficienților DCT (Nefian). O comparație a acestor timpi în funcție de metodă și dimensiunile vectorilor de observație se găsește în următorul tabel (Fig. 7.2-9):

	Dimensiune	Timp recunoaștere (relativ)	Procentaj de recunoaștere
Samaria (pixeli)	400 (20 x 20)	575%	93%
Nefian (DCT)	9 (3 x 3)	100%	98%
CTAG	28 (14 x 2)	175%	83%
CTAG	14 (14 x 1)	88%	83%

Fig. 7.2-9 Comparație metode/dimensiuni vect obs/timp rec/procentaj rec.

7.3. Rezultate experimentale și concluzii

Așa cum am specificat și mai sus, experimentele au fost realizate utilizând baza de date ORL de la Olivetti Research Laboratory. Motivul folosirii acestei baze de date a fost acela de a putea compara rezultatele obținute cu cele raportate de către Samaria [68] și Nefian [70].

Pentru experimentare am utilizat suportul oferit de către pachetul Open Computer Vision Library, dezvoltat de către cercetătorii de la Intel, printre care se numără și Nefian. Ca și suport hardware am folosit un PC AMD K6-2 la 450MHz, 128MB RAM, Windows 2000, având ca și placă de achiziție video un tuner TV care împreună cu o cameră video au fost utilizate pentru implementarea și respectiv testarea metodelor abordate în această lucrare. Folosindu-se acest suport am implementat metoda propusă în această lucrare și anume CTAG, iar pentru a putea realiza compararea și cu metoda propusă de Samaria, am implementat-o și pe aceasta din urmă.

Pentru metoda propusă, CTAG, am identificat următorii parametri ai modelului:

1. Dimensiunea blocului – o dimensiune prea mică nu reușește să capteze suficiente informații despre structura texturii, iar una prea mare conduce la un număr insuficient de vectori de observație pentru stările încuibate.
2. Plaja de culori asupra căreia se aplică metoda CTAG – aceasta ar putea fi o direcție de continuare a cercetării, o idee în acest sens ar putea fi o plajă auto-ajustabilă, dependentă de regiunea asupra căreia se aplică metoda (de exemplu pentru piele poate fi definită o anumită plajă).
3. Nivelele de rezoluție peste care se calculează coeficienții CTAG – s-a constatat că nu există o îmbunătățire semnificativă măbind numărul de nivele de rezoluție.

4. Suprapunerile pe x și y . În cazul acestora s-a constatat că micșorând ușor aceste valori, ceea ce duce la accelerarea procesului, nu are un impact negativ, astfel s-a ajuns de la o deplasare de 2×2 la DCT la o deplasare de 5×5 la CTAG.

Rezultatele experimentale, folosind eHMM + CTAG aplicat pe baza de date ORL, sunt date, în funcție de parametri modelului, în următorul tabel (Fig. 7.3-1):

Dimensiune bloc	Plajă culori *	Nivele **	Deplasări	Factor de corecție ***	Procentaj recunoaștere
25 × 25	(16,0,16)	(2,2,2)	5 × 5	10	63.3%
25 × 25	(16,0,16)	(2,2,2)	5 × 5	10	73.3%
25 × 25	(32,0,8)	(3,2,2)	5 × 5	10	76.7%
25 × 25	(32,0,8)	(3,2,2)	2 × 2	10	76.7%
25 × 25	(16,0,16)	(2,2,2)	2 × 2	10	66.7%
30 × 30	(16,0,16)	(2,2,2)	2 × 2	10	73.3%
30 × 30	(14,16,16)	(2,2,2)	5 × 5	10	76.7%
30 × 40	(14,16,16)	(2,2,2)	5 × 5	10	63.3%
30 × 30	(14,16,16)	(2,2,2)	5 × 5	10	76.7%
30 × 30	(14,16,16)	(2,2,2)	5 × 5	40	83.3%
30 × 30	(14,16,16)	(2,2,2)	5 × 5	100	80.0%
30 × 30	(10,128,8)	(2,2,2)	5 × 5	40	36.7%
30 × 30	(12,16,16)	(3,2,2)	5 × 5	40	70.0%
30 × 30	(14,16,16)	(1,2,2)	5 × 5	40	83.3%
30 × 30	(14,16,16)	(1,2,2)	2 × 2	40	83.3%
35 × 35	(14,16,16)	(1,2,2)	5 × 5	40	80.0%

Fig. 7.3-1 Rezultate experimentale eHMM + CTAG

* (Număr de valori, Valoare inițială, Valoare increment)

** (Număr de nivele, Valoare inițială, Valoare increment)

*** Factor de corecție a valorilor pentru o mai bună discriminare între vectori (valoare cu care sunt înmulțite valorile vectorului de observație)

8. Concluzii

Acest capitol prezintă contribuțiile, concluziile finale și dă câteva recomandări pentru viitoarele cercetări.

În orice sistem automat, înaintea extragerii efective a informației, sunt aplicate diverse operații de corectare, care încearcă aducerea imaginilor într-o formă cât de cât unitară, în lucrare prezentându-se câteva dintre cele mai utilizate operații în acest sens.

Ținând cont pe de o parte de domeniul vast al extragerii de caracteristici în prelucrarea și interpretarea automată a imaginilor, precum și de posibilele aplicații practice de interes pe care le are detectarea și recunoașterea fețelor umane, în această lucrare s-au luat în considerare tehnicile de extragere a caracteristicilor care au un impact major asupra acestui din urmă subdomeniu, și anume extragerea de caracteristici în vederea detectării și recunoașterii fețelor umane.

8.1. Contribuții

Contribuțiile la domeniul abordat prin această lucrare pot fi sumarizate astfel:

- Introducerea unei noi metode de extragere de caracteristici, bazată pe o caracterizare topologică a texturilor – metodă pe care am denumit-o CTAG.
- Prezentarea modelului Markov ascuns încapsulat (eHMM), model folosit în recunoașterea fețelor.
- Obținerea elementelor vectorului de observație a HMM prin metoda propusă (CTAG).
- Demonstrarea caracteristicilor abordării propuse (HMM + CTAG) printr-o serie de experimentări.
- Studiarea modului în care este afectată performanța de recunoaștere prin modificarea parametrilor modelului propus.
- Compararea modelului propus cu modelele propuse de Samaria și Nefian aplicându-le pe același set de date, și anume baza de date ORL.
- Prezentarea principalelor metode de extragere a caracteristicilor folosite în prelucrarea imaginilor și implementarea unora dintre acestea în vederea incorporării într-un sistem de recunoaștere/confirmare a identității.
- Prezentarea unora dintre metodele de detectare a feței umane, evidențiind totodată problemele care apar în localizarea și detectarea feței folosind unele dintre aceste metode.

8.2. Direcții viitoare de cercetare

Bazat pe experimentele și rezultatele prezentate în această lucrare, încerc să sugerez câteva direcții de continuare a cercetării în acest domeniu:

- O idee de dezvoltare a modelului propus, CTAG, ar fi combinarea acestor coeficienți cu coeficienții DCT, dând astfel vectori de observație care țin cont și de caracteristicile texturii blocurilor extrase, crescând astfel eficiența detecției.
- Cu toate că multe lucruri s-au făcut deja în această direcție de cercetare, recunoașterea feței umane în cadrul unor scene cu fundal difuz este încă o problemă care necesită mai departe investigare.
- Creșterea numărului de imagini folosite în antrenarea și testarea modelului va permite o mai bună estimare a parametrilor acestuia, ceea ce ar putea conduce la creșterea ratei de recunoaștere a modelului.
- Construcția unor vectori de observație cu o putere de discriminare mai mare, ar conduce la o reprezentare mai bună a blocurilor imaginilor și mai ales la o discriminare mai bună între stările modelului.
- Îmbunătățirea algoritmilor de calcul a coeficienților CTAG, astfel încât sistemul să fie capabil de o recunoaștere în timp real.
- Elaborarea unui model de grupare a persoanelor astfel încât la o creștere semnificativă a numărului de persoane din baza de date, timpul necesar regăsirii unei persoane asemănătoare să fie minimizat.

8.3. Concluzii finale

Dintre metodele de selectare a caracteristicilor prezentate în capitolul trei sunt evidențiate metodele Karhunen-Loève, care este considerată transformarea optimă, reținând cea mai mare cantitate de energie, și DCT, care datorită caracteristicilor sale este folosită la compresia de imagini. Aceste metode sunt folosite în modelul HMM pentru obținerea vectorilor de observație.

În capitolul cinci am încercat să folosesc reprezentarea topologică multirezoluție a texturilor cu nivele de gri propusă de Pikaz și Averbuch, să o extind la texturile color și să combin reprezentarea rezultată cu reprezentarea multiscară bazată pe teoria **scale space** descrisă în [2]. Rezultatul acestei combinații este o nouă reprezentare pe care am denumit-o CTAG (Graful de Adiacență al Texturilor Color)[30], în care fiecare nod va conține descrierea de culoare și textură a regiunii respective. Pentru această reprezentare au existat anumite cerințe și anume nu trebuie să fie consumatoare de timp și nu trebuie să fie consumatoare de memorie.

Principalul avantaj al unei asemenea reprezentări este conservarea tuturor avantajelor reprezentării CAG [9], îmbunătățirea recunoașterii prin posibilitatea folosirii mai multor caracteristici în principal descrierea multiscară, multirezoluție a texturilor color, a regiunilor precum și oferirea pașilor necesari comparării acestor descrieri.

Ca metodă de extragere de caracteristici și recunoaștere este prezentată o abordare folosind un HMM încapsulat. Utilizarea metodei HMM încapsulat pentru modelarea feței este justificată atât din cauza flexibilității cât și a structurii.

Comparată cu metodele bazate pe șablon sau cu cele HMM standard pentru detectarea feței această abordare bazată pe metoda HMM încapsulat are următoarele avantaje:

- Este mai flexibilă în raport cu variații de scară și deformări naturale, atât pe plan vertical cât și orizontal.
- Permite o implementare eficientă a algoritmului de detecție a feței datorită împărțirii șablonului sau benzilor orizontale în blocuri de imagine care sunt procesate pentru a obține vectori de observație.
- Permite o implementare mai rapidă atât pentru detecție cât și pentru recunoaștere datorită complexității reduse a algoritmului Viterbi dublu încapsulat.
- Permite includerea caracteristicilor texturii blocurilor folosite pentru extragerea vectorilor de observație, măbind acuratețea detecției.
- Poate fi implementat folosind o arhitectură paralelă reducând astfel considerabil complexitatea atât a detecției cât și a recunoașterii.
- Păstrează structura bidimensională a datelor fără a folosi structuri speciale.

Comparat cu metodele bazate pe șablon sau cele HMM standard pentru detectarea feței această abordare este mai flexibilă în raport cu variații de scară și deformări naturale, atât pe plan vertical cât și orizontal.

În modelul HMM încapsulat, caracteristicile extrase formează vectori de observație, care sunt folosiți în procesul de detecție și recunoaștere a feței. Spre deosebire de Samaria [68] și Nefian [70] care au folosit culori de pixeli respectiv transformata cosinus discretă (DCT) pentru a calcula elementele vectorilor de observație, în această lucrare am propus folosirea metodei CTAG pentru calcularea elementelor vectorilor de observație.

Pentru metoda propusă, CTAG, am identificat următorii parametri ai modelului:

1. Dimensiunea blocului – o dimensiune prea mică nu reușește să capteze suficiente informații despre structura texturii, iar una prea mare conduce la un număr insuficient de vectori de observație pentru stările încuibate.
2. Plaja de culori asupra căreia se aplică metoda CTAG – aceasta ar putea fi o direcție de continuare a cercetării, o idee în acest sens ar putea fi o plajă auto-ajustabilă, dependentă de regiunea asupra căreia se aplică metoda (de exemplu pentru piele poate fi definită o anumită plajă).
3. Nivelele de rezoluție peste care se calculează coeficienții CTAG – s-a constatat că nu există o îmbunătățire semnificativă măbind numărul de nivele de rezoluție.
4. Suprapunerile pe x și y . În cazul acestora s-a constatat că micșorând ușor aceste valori, ceea ce duce la accelerarea procesului, nu are un impact negativ, astfel s-a ajuns de la o deplasare de 2×2 la DCT la o deplasare de 5×5 la CTAG.

Îmbunătățirea propusă în această lucrare, pentru procesul de recunoaștere, este luarea în considerație nu numai a culorii ci și a texturii obiectelor (regiunilor). Față de [30] unde am folosit această metodă de extragere de caracteristici în conjuncție cu o structură propusă de Mattas [9], aici am folosit-o ca și metodă de extragere de caracteristici de bază.

Din experimentele efectuate un HMM încapsulat are performanțe la fel de bune sau mai bune decât alte metode având totodată și complexitate redusă de calcul și asta pentru o gamă largă de metode de extragere de caracteristici (CTAG, DCT, KLT, valori de pixeli), prin aceasta confirmând rezultatele obținute atât de Samaria [60] cât și de Nefian [70].

Pentru o recunoaștere performantă este esențială etapa de localizare și detecție a feței într-o imagine. În acest sens s-a combinat metoda CTAG cu metoda de extragere a caracteristicilor min-max rezultând o îmbunătățire a localizării feței.

Pe partea de recunoaștere efectivă a feței s-au comparat metodele propuse de Samaria și Nefian cu o metodă de extragere de caracteristici folosind metoda CTAG (Fig. 7.2-9, Fig. 7.3-1) pentru detecția și localizarea feței ajungându-se la concluzia că datorită proprietăților transformării cosinus discretă (DCT), algoritmul propus de Nefian atinge cele mai bune performanțe.

La variații în structura fundalului nici una dintre metodele testate nu a dat rezultate satisfăcătoare. S-a observat totuși că metoda CTAG, care ține cont mai mult de textura blocurilor care formează imaginile, prezintă o rată de recunoaștere mai mare, indicând că direcția de cercetare abordată este o direcție promițătoare.

Metoda	Procentaj de recunoaștere
Samaria	16.7%
Nefian	20.0%
CTAG	26.7%

Fig. 8.3-1. Comparație metode la variații de fundal.

În concluzie se poate spune că prelucrarea și interpretarea imaginilor și în cadrul acestora recunoașterea feței rămâne în continuare un domeniu de cercetare care prezintă un real interes pentru lumea de azi, în contextul în care interconectarea pe cale electronică devine o problemă tot mai necesară și tot mai actuală.

9. Bibliografie

- [1] Camapum, J. F., Color-based Recognition for Remote Sensing in Environmental Systems, *Progress Report 1995-1996*, School of Engineering, Coventry University, UK.
- [2] Lindeberg, T., Scale-Space Theory in Computer Vision, *Kluwer Academic Publishers, Boston, USA, 1994*.
- [3] Pikaz, A. and Averbuch, A., An Efficient Topological Characterization of Gray-Levels Textures, Using a Multiresolution Representation, *Graphical Models and Image Processing*, Vol. 59, No. 1, January, 1-17, 1997.
- [4] Khotanzad, A., Bouarfa, A., Image Segmentation by a Parallel, Non-Parametric Histogram based Clustering Algorithm, *Pattern Recognition*, Vol. 23, No.9, 961-973, 1990.
- [5] Chakravarthy, S. V., Gosh, J., Scale-based Clustering using the Radial Basis Function Network. *NSF - ECS-9307632 && ONR Contract N00014-92C-0232*.
- [6] Mayer, H., Steger, C., A New Approach For Line Extraction and its Integration in a Multi-Scale, Multi-Abstraction-Level Road Extraction System. *DF Eb 74/8-2*.
- [7] Fosgate, C. H., Krim, H., Irving, W. W., Karl, W. C. and Willsky, A. S., Multiscale Segmentation and Anomaly Enhancement of SAR Imagery, *IEEE Transactions on Image Processing*, Vol. 6, No. 1, January 1997.
- [8] Fathima, S. T., Data and model-driven selection using color regions, *Image Understanding Workshop (San Diego, CA, Jan 26-29, 1992)*, 705-716. Morgan Kaufmann, 1992.
- [9] Mattas, J., Color-based Object Recognition, *Submitted for the Degree of Doctor of Philosophy* from the University of Surrey, September 1995.
- [10] Sonka, M., Hlavac, V., Boyle, R., *Image Processing, Analysis and Machine Vision*, Chapman & Hall Computing, London. 1994.
- [11] Tomita, F., Tsuji, S., *Computer Analysis of Visual Textures*, Kluwer Academic Publishers, Norwell, Ma. 1990.
- [12] Burt, P. J., Adelson, E. H., The Laplacian pyramid as a compact image code, *IEEE Transaction Communications*, 9, 532-540, 1983.
- [13] Klinger, A., Pattern and search statistics, *Optimizing Methods in Statistics*, New York, Academic Press, 1971.

- [14] Mallat, S. G., A theory for multiresolution signal decomposition: the wavelet representation, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11, 674-694, 1989.
- [15] Koenderink, J. J., The structure of images, *Biological Cybernetics*, 50, 363-370, 1984.
- [16] Ripley, B. D., Pattern Recognition and Neural Networks, *Cambridge University Press*, 1996.
- [17] Vancea, R., Holban, S., Ciubotariu, D., Recunoașterea Formelor - Aplicații, *Editura Academiei R.S.R.*, București 1989.
- [18] Duda, R. O., Hart, P. E., Pattern Classification and Scene Analysis, *John Wiley & Sons, Inc.*, 1973
- [19] Bergholm, F., Edge focusing, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9, pp. 726-741, 1987
- [20] Florack, L. M. J., Ter Haar Romeny, B. M., Koenderink, J. J., Viergever, M. A., Cartesian differential invariants in scale-space, *Journal of Mathematical Imaging and Vision*, 3, pp. 327-348, 1993
- [21] Crowley, J. L., Sanderson, A. C., Multiple resolution representation and probabilistic matching of 2-D gray-scale shape, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9, pp. 113-121, 1987
- [22] Crowley, J. L., Parker, A. C., A representation for shape based on peaks and ridges in the difference of low-pass transform, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6, pp. 156-170, 1984
- [23] Lindeberg, T., Scale-space for discrete signals, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12, pp. 234-254, 1990
- [24] Lindeberg, T., Discrete scale space theory and the scale space primal sketch, *PhD Thesis*, Royal Institute of Technology, Stockholm, Sweden, 1991
- [25] Lindeberg, T., Gårding, J., Shape from texture from a multi-scale perspective, *Proceedings of 4th International Conference on Computer Vision*, Berlin, pp. 683-691, 1993
- [26] Tanimoto, S., Pavlidis, T., A hierarchical structure for picture processing, *Computer Vision, Graphics, and Image Processing*, 4, pp. 104-119, 1975
- [27] Witkin, A. P., Scale-space filtering, *Proceedings of 8th International Conference on Artificial Intelligence*, Karlsruhe, pp. 1019-1022, 1983
- [28] Kenneth R. Castleman, "Digital Image Processing", Editura Prentice Hall, New Jersey, 1996.

- [29] A. Vlaicu , “Prelucrarea digitală a imaginilor”, Editura Albastră, Cluj-Napoca, 1997.
- [30] Robert Györödi, Dr. Mark H. Fisher, Juliana Camapum - “Multi-scale color texture description” - 12th International Conference on Systems Engineering, Vol1, Coventry University, 9-11 September 1997, UK, pag. 284-289.
- [31] Cornelia Györödi, Robert Györödi, Alexandru Toth - “Basics of Image Processing and Recognition.” - A & Q '98 International Conference an Automation and Quality Control 28.05-29.05.1998, Cluj-Napoca, Romania, pag. 336-342.
- [32] Robert Györödi, Cornelia Györödi - “Mathematical Modeling In Pattern Recognition.” - 113th Pannonian Applied Mathematical Meting, October 1995, Dep. S. Mathematics, Transp. Fac. of T.U. Budapest, pag 69-74.
- [33] Robert Györödi, Cornelia Györödi - “Identity Confirmation System Based on the Features of the Human Face.” - Analele Universității din Oradea, Fascicola Electrotehnică, Oradea, Romania, 30.05-01.06, 1997, pag. 29-34.
- [34] R. Chellappa, C. Wilson, and S. Sirohey – “Human and Machine Recognition of Faces: A survey”, Proceedings of IEEE, vol. 83, Mai, 1995.
- [35] A. Samal and P. A. Iyengar – “Automatic Recognition and analysis of human faces and facial expressions: A survey”, Pattern Recognition, 25(1) pag. 65-77, 1992.
- [36] L. D. Harmon, M.K. Khan, R. Lasch și P. F. Ramig – ”Machine Identification of Human Faces”, Pattern Recognition, 13(2), pag. 97-110, 1981.
- [37] L. Najman, R. Vaillan și E. Pernot - ”Face from sideview to identification”, G. Vernazza, A.N. Venetsanopoulos și C. Braccini, editori, Image Processing: Theory and Applications, Elsevier Science Publishers, 1993.
- [38] T. Kanade - “Computer Recognition of Human Faces”, Interdisciplinary Systems Research, Birkhäuser Verlag, 1977.
- [39] R. Prunelli și T. Poggio - ”Faces Recognition: Features versus templates”, IEEE Transactions on Pattern Analysis and Machine Intelligence, 15(10), pag. 1042-1052, 1993.
- [40] K. H. Wong, H. H. M. Law și P. W. M. Tsang – ”A system for recognising human faces”, Proceedings of the International Conference on Acoustics, Speech and Signal Processing, pag 1638-1642, 1989.
- [41] A. J. Goldstein, L. D. Harmon și A. B. Lesk – ”Identification of Human Faces”, Proceedings of the IEEE 59(5), pag 748-760, 1971.
- [42] R. Brunelli și S. Messelodi - ”Robust Estimation of Corelation with Applications to Computer Vision”, Pattern Recognition, vol 28, nr. 6, pag 833-841, 1995.

- [43] M. Bichsel și A. Pentland, - "Human Faces Recognition and Faces Image Set's Topology", CVGIP: Image Understanding, vol 59 pag 254-261, Martie, 1994.
- [44] R. Brunelli și D. Falavigna - "Person Identification Using Multiple Cues", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 17, pag 955-966, Octombrie 1995.
- [45] L. Yuille, P. Hallinan și D. Cohen – "Feature Extraction from Faces Using Deformable Templates", International Journal of Computer Vision, vol 8, no 2, pag 99+111, 1992.
- [46] B. Manjunath, R. Chellappa și C.D. Malsburg – "A Feature Based Approach to Face Recognition", Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pag 373-378, 1992.
- [47] G. Chow și X. Li – "Towards a system for automatic facial feature detection", Pattern Recognition, vol 26, no 12, pag 1739-1755, 1993.
- [48] L. Stringa – "Eyes Detection for Face Recognition", Applied Artificial Intelligence, vol 7, pag 365-382, Oct-Dec 1993.
- [49] M. Michaelis, R. Herpes, L. Witta și G. Sommer – "Hierarchical Filtering Scheme for the Detection of Facial Keypoints", International Conference on Acoustics, Speech and Signal Processing, vol 4, pag 2541-2544, 1997.
- [50] P. Burt – "Smart Sensing Within a Pyramid Vision Machine", Proceedings of the IEEE, vol 76, pag 1006-1015, August 1988.
- [51] M. Kirby, L. Sirovich – "Application of the Karhunen-Loève procedure for the characterization of human faces", IEEE Transactions on Pattern Analysis and Machine Intelligence 12 (1), pag. 103-108, 1990.
- [52] M. Turk și A. Pentland – "Eigenfaces for Recognition", Journal of Cognitive Neuroscience, 3(1), pag.71-86, 1991.
- [53] B. Moghaddam și A. Pentland – "Probabilistic visual learning for object Representation", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol 19, pag 676-710, iulie 1997.
- [54] B. Moghaddam și A. Pentland – "Probabilistic visual learning for object representation", in The 5th International conference on Computer Vision, Cambridge MA, pag. 786-793, iunie 1995.
- [55] A. Pentland, B. Moghaddam și T. Starner – "View based and modular eigenspaces for face recognition", in Proceedings on IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pag. 84-91, 1994.
- [56] G.Cotrell și M. Fleming – "Face recognition using unsupervised feature extraction", Proceedings International Neural Network Conference, pag. 322-325, 1990.

- [57] A. Lawrence, C. Giles, A. Tsoi și A. Back – “Face recognition : A convolution neural network approach”, IEEE Transactions on Neural Networks, vol. 8, nr 1, pag. 98-113, 1997.
- [58] S. H. Lin, Y. Chan și S. Kung – “ A probabilistic decision-based neural network for locationg deformable objects and its applications to surveillance system and video browsing” International Conference on Acoustics, Speech and Signal Processing, vol. 6, pag. 3553-3556, 1996.
- [59] S. H. Lin, S. Kung și L. J. Lin - “ Face recognition/detection by probabilistic decision-based neural network”, IEEE Transactions on Neural Network, vol. 8, pag. 114-132, ianuarie 1997.
- [60] N. M. Allinson, A. W. Ellis , B. M. Flude și A. J. Luckman - “ A connectionist model of familiar face recognition ”, IEE Colloquium on Machine Storage and Recognition of Faces, 1992.
- [61] L. Rabiner – “A tutorial on Hidden Markov Models and selected applications in speech recognition”, Proceedings of IEEE, vol 77, pag 257-286, Februarie 1989.
- [62] L. Rabiner și B. Huang – “Fundamentals of Speech Recognition” Englewood Cliffs, NJ: Prentice Hall, 1993.
- [63] O. Agazzi, S. Kuo, E. Levin și R. Pieraccini – ”Connected and degraded text recognition using planar HMM”, International Conference on Acoustics, Speech and Signal Processing '93, vol 5, pag 113-116, 1993.
- [64] F. Samaria – ”Face segmentation for identification using hidden Markov models”, British Machine Vision Conference, 1993.
- [65] F. Samaria și F. Fallside – ”Face identification and feature extraction using hidden Markov models”, Image Processing: Theory and Applications, 1993.
- [66] F. Samaria și F. Fallside – ”Automated face identification using hidden Markov models”, Proceedings of the International Conference on Advanced Mechatronics, pag 1-9, 1993.
- [67] F. Samaria și A. Harter – ”Parametrisation of stohastic model for human face identification”, Proceedings of the Second IEEE Workshop on Application of Computer Vision, pag 138-142, 1994.
- [68] F. Samaria – ”Face Recognition Using Hidden Markov Models”, PhD thesis, University of Cambridge, 1994.
- [69] A.V. Nefian și M.H. Hayes – ”Face Recognition Using an Embedded HMM”, Proceedings of the IEEE Conference on Audio and Video-based Biometric Person Authentication, pag 19-24, Martie 1999.
- [70] A.V. Nefian – ”A Hidden Markov Model-Based Approach for Faces Detection and Recognition”, PhD thesis, Georgia Institute of Technology , 1999.

[71] Robert Györödi, Cornelia Györödi - "Identity Confirmation System Based on CTAG and HMM" – International Symposium, Universitatea de Vest, Timisoara, Romania, 2001, în curs de publicare.

[72] Robert Györödi, "A comparative Study of Classical Feature Extraction Techniques Used in Pattern Recognition", Proceedings of Oradea EMES, May 27-30, 2000, Oradea, Romania.

[73] Cornelia Györödi, Robert Györödi - "Utilisation of Neural Networks in Contur Finding and Following." - Analele Universității din Oradea, Fascicola Electrotehnică, Oradea, Romania, 30.05-01.06, 1997, pag. 23-28.

[74] Robert Györödi, Cornelia Györödi - "Applications of Pattern Recognition", Scientific Conference on Electronic Computers & Informatics, Kosice - Herlany, Slovakia, September 26-27, 1996, pag. 78-83.

Lista lucrărilor publicate

- Robert Györödi, Cornelia Györödi - "Identity Confirmation System Based on CTAG and HMM" – International Symposium, Universitatea de Vest, Timisoara, Romania, 2001, în curs de publicare.
- Robert Györödi, "A comparative Study of Classical Feature Extraction Techniques Used in Pattern Recognition", Proceedings of Oradea EMES, May 27-30, 2000, Oradea, Romania.
- Robert Györödi, Dr. Mark H. Fisher, Juliana Camapum - "Multi-scale color texture description" - 12th International Conference on Systems Engineering, Vol1, Coventry University, 9-11 September 1997, UK, pag. 284-289.
- Robert Györödi, Cornelia Györödi - "Identity Confirmation System Based on the Features of the Human Face." - Analele Universității din Oradea, Fascicola Electrotehnică, Oradea, Romania, 30.05-01.06, 1997, pag. 29-34.
- Cornelia Györödi, Robert Györödi - "Utilisation of Neural Networks in Contur Finding and Following." - Analele Universității din Oradea, Fascicola Electrotehnică, Oradea, Romania, 30.05-01.06, 1997, pag. 23-28.
- Cornelia Györödi, Robert Györödi, Alexandru Toth - "Basics of Image Processing and Recognition." - A & Q '98 International Conference an Automation and Quality Control 28.05-29.05.1998, Cluj-Napoca, Romania, pag. 336-342.
- Robert Györödi, Cornelia Györödi - "Applications of Pattern Recognition", Scientific Conference on Electronic Computers & Informatics, Kosice - Herlany, Slovakia, September 26-27, 1996, pag. 78-83.
- Robert Györödi, Cornelia Györödi - "Mathematical Modeling In Pattern Recognition." - 113th Pannonian Applied Mathematical Meeting, October 1995, Dep. S. Mathematics, Transp. Fac. of T.U. Budapest, pag 69-74.