



UNIVERSITATEA “POLITEHNICA” din BUCUREȘTI

Facultatea de Automatică și Calculatoare

Departamentul de Calculatoare

TEZĂ DE DOCTORAT

Clasificarea în Imagini Digitale

Autor: ing. Octavian Georgescu

Conducător de doctorat: Prof. dr. ing. Florica Moldoveanu

BUCUREȘTI, 2012

Prezentarea capitolelor lucrării

În cadrul analizei imaginilor, ne-am pus problema Clasificării în Imagini (CI) digitale. Cu toate că ne referim numai la aceste imagini, cele discutate de noi se pot generaliza în cadrul unei analize a semnalelor, imaginea putând fi văzută ca un semnal bidimensional. La modul general, clasificarea în (interiorul lui) X se poate face după anumite forme sau după anumite caracteristici ale primitivelor lui X . În cazul CI, aceste primitive sunt pixelii imaginii digitale. Așadar, CI se poate face prin metode de clasificare bazate pe identificarea de regiuni cu anumite geometrii (RB = ‘region based methods’) și / sau prin metode de clasificare bazate pe pixeli (PB = ‘pixel based methods’) cu anumite proprietăți. Putem vedea metodele RB ca metode de clasificare de nivel înalt comparativ cu metodele PB. Lucrarea noastră tratează în principal metodele PB. Vom menționa metodele RB, care mai poartă și denumirea generică de metode de Recunoaștere a Formelor, în special pentru a face trimitere la metodele PB.

Din punct de vedere al teoriei mulțimilor, CI înseamnă partiționarea imaginii (digitale) în mulțimi (clase de echivalență) de pixeli cu trăsături relativ comune. Partiționarea, în cele mai multe cazuri reale, nu se poate face exact, ci aproximativ, într-un anumit grad. În plus, partiționarea depinde de rezoluția / precizia / granularitatea conform căreia se face clasificarea.

Ca redactare, lucrarea e organizată conform ierarhiei: capitol – subcapitol – secțiune – subsecțiune. Din punct de vedere al contribuțiilor, lucrarea e orientată pragmatic și prezintă într-un tot unitar – poate fi și o carte folositoare pentru cei care, de la începători la avansați, se ocupă de analiza și, mai ales, de clasificarea în imagini – câteva contribuții (sinteze / observații) teoretice și contribuții practice. Întinderea lucrării a rezultat din scopul ei de a fi conținută în sine (totuși, s-a trecut repede peste teoria elementară). Prezentăm pe scurt conținutul lucrării, pe capitole, făcând precizări sumare în dreptul capitolelor de contribuție teoretică.

IȘ Încadrarea problemei CI și descrierea ei în Domeniul Spațial și în Domeniul Probabilistic

- precizăm problemele practice ale CI și facem o scurtă incursiune teoretică a abordării lor în cel mai ușor de înțeles domeniu de reprezentare a imaginilor: cel spațial;

- plecăm de la principii de bază de filtrare în scopul găsirii pixelilor de muchie (frontieră), a conturilor, pentru ca în subcapitolul “Reprezentarea formelor” să menționăm câteva dintre metodele RB tipice;

- subcapitolele “Tehnici de segmentare”, “Recunoașterea imaginilor prin potrivire” și “Tehnici de clasificare” duc discuția în aria efectivă a CI (metodele PB);

- subcapitolul “Domeniul Probabilistic” e prezent în acest capitol din două motive principale: 1) vedem Domeniul Probabilistic ca “istorie” a Domeniului Spațial și 2) acest domeniu e aproape la fel de ușor de abordat precum cel spațial.

II§ “Separabilitatea claselor”

- în primul subcapitol, “Descompunere ortogonală cu Unde și Undine”, tratăm acest subiect din perspectiva analizei semnalelor în Domeniul Frecvență și în Domeniul Frecvență-Timp (folosim denumirea “Frecvență-Timp” pentru că așa se regăsește în literatura de specialitate, deși, în cazul imaginilor, ar fi mai corect să spunem “Frecvență-Spațiu”, frecvențele de care vorbim fiind spațiale). Avem în vedere decelarea anumitor proprietăți în imagini după ce le trecem în domeniul frecvențelor prin aplicarea transformatei Fourier. Abordăm aici chestiunea dependenței rezoluției din domeniul Timp de cea din domeniul Frecvență, atât la nivelul teoremei eșantionării cât și la nivelul principiului incertitudinii (al lui Heisenberg). Ultima secțiune, II.1.5. “Relații fundamentale în jurul conceptului de corelare, între principalele domenii de reprezentare a semnalelor”, propune o sinteză originală în jurul problemei clasificării, problemă care poate fi văzută ca una de corelare sau de descompunere.

- în al doilea subcapitol, “Granularitate și gradație în Domeniul Fuzzy”, ne referim la clase separabile într-un anumit grad, clasa corespunzând unei “partiții fuzzy”. În literatura de specialitate “partițiile fuzzy” au diverse înțelesuri. Plecăm de la așa-numitele partiții / clase fuzzy, care se pot suprapune, putând fi proiectate ca seturi fuzzy ([Shakhatreh & Hayajneh 2010]), pentru a propune un termen care ni se pare mai semnificativ decât acela de partiție fuzzy, granulă ș.a.: PROTOTIPUL (vezi ultima secțiune: II.2.5. Clasa fuzzy = PROTOTIPUL). Generalizând conceptul de prototip, în domeniul frecvență, prototipul corespunde unei / undinei, iar în domeniul fuzzy, prototipul corespunde clasei. Cu precizarea că, în general (conform teoriei clasice), dacă în primul caz, separarea semnalului în unde / undine se face printr-o descompunere ortogonală, în cazul al doilea, separarea în clase admite o suprapunere.

Domeniul fuzzy permite generalizarea / integrarea conceptuală a mai multor abordări ale clasificării. Mă refer în special la acele abordări bazate pe noțiunea de probabilitate sau posibilitate (acesta e un concept fuzzy mai larg decât probabilitatea – vezi [Zadeh, 1978]). În secțiunea II.2.2., “Domeniul posibilului: o fuziune fuzzy a diverselor tipuri de cunoștințe”, scoatem în evidență această versatilitate a conceptului de fuzzy. Vom continua în această direcție în II.2.4., “Fuzzy logic = calculul cu cuvinte” [Zadeh, 1996], arătând că prin termenul lexical se poate îngloba în raționamentul ingineresc intuiția, flerul, calități care, desigur, sunt favorizate de o experiență îndelungată, experiență pe care o putem vedea și ca pe o statistică implicită, mai greu de formalizat (e.g., față de teoria probabilităților). Domeniul fuzzy îmi permite să vorbesc într-un cadru unitar despre două din cele trei contribuții practice (redate în capitolul IV)

III§ Domeniul multispectral al imaginilor color – pregătiri teoretice pentru capitolul IV.1.

- în subcapitolul III.3., “Caracterul perceptual al culorii”, evidențiem calitatea domeniului fuzzy de a se preta procesării multi-canal – în cazul contribuțiilor noastre practice, clasificării în imagini color

IV§ Contribuții practice – vezi Cuprinsul de mai jos

V§ Tabel sinoptic al contribuțiilor originale. Studiul viitor

Mulțumiri

Am onoarea să-i mulțumesc Doamnei prof. dr. ing. Florica Moldoveanu pentru generosul sprijin, pentru substanțiala îndrumare în întocmirea acestei lucrări, pentru răbdarea și încrederea acordate chiar și când timpul părea că “nu mai are răbdare”. Mulțumesc de asemenea familiei și firmei Siveco pentru înțelegerea de care au dat dovadă în toată această perioadă.

CUPRINS

<i>IS Încadrarea problemei CI și descrierea ei în Domeniul Spațial și în Domeniul Probabilistic</i>	9
I.1 Introducere	9
I.1.1 Scopul sistemelor Computer Vision și al CI	9
I.1.2 Organizarea unui sistem Computer Vision (CV / CI)	14
I.2. Detecția muchiilor (edge detection).....	17
I.2.1. Folosirea derivatei de ordinul 1.....	18
I.2.1.1. Principiul	18
I.2.2. Folosirea derivatei de ordinul 2	20
I.2.2.1. Operatori Laplace.....	20
I.3. Extragerea contururilor	21
I.3.1. Conectivitatea	21
I.3.2. Tehnici de extragere a conturului	21
I.4. Reprezentarea formelor	22
I.4.1. Simplificarea formelor.....	24
I.4.1.1. Anvelopa convexă	24
I.4.1.2. Skeletoane	25
I.4.2. Funcții asociate formelor.....	28
I.4.2.1. Semnătura formei	28
I.4.2.2. Curba granulometrică.....	29
I.4.3. Parametrii de formă scalari.....	31
I.4.3.1. Parametri geometrici (rapoarte de aspect).....	31
I.4.3.2. Momente statistice și invarianți.....	33
I.4.3.3. Descriptori Fourier	35
I.4.4. Concluzii	37
I.5. Tehnici de segmentare	38
I.5.1. Limitarea intensității / Prăguirea ('Thresholding')	39
I.5.1.1. Metoda Otsu (de prăguire globală optimă).....	40
I.5.2. Etichetarea componentelor	42
I.6. Clasificarea imaginilor prin potrivire ('Image Matching').....	44
I.6.1. Potrivirea pe baza corelării ('Template Matching and Area Correlation')	45
I.6.2. Cazuri particulare	47
I.6.2.1. Căutarea logaritmică.....	47
I.6.2.2. Potrivirea imaginilor binare	47
I.7. Tehnici de clasificare.....	48

I.7.1. Clasificarea supervizată.....	52
I.7.1.1. Clasificarea fără distribuții.....	52
I.7.1.2. Rețele neurale	59
I.7.2. Clasificarea nesupervizată (clustering).....	62
I.8. Domeniul probabilistic	64
I.8.1. Statistici de ordinul 1 / Trăsăturile de histogramă.....	64
I.8.1.1. Momente.....	64
I.8.1.2. Exemplu practic: caracterizarea unei texturi	66
I.8.1.3. Covarianță și corelație.....	67
I.8.1.4. Folosirea componentelor principale în cadrul Transformării Hotelling (sau KLT).....	68
I.8.2. Statistici de ordinul 2. Studiu de caz: texturi	72
I.8.2.1. Matricea de coocurență.....	72
I.8.3. Clasificarea statistică	74
I.8.3.1. Clasificatori statistici optimi.....	74
II.8 Separabilitatea claselor.....	78
II.1. Descompunere ortogonală cu Unde și Undine.....	78
II.1.1. Cazul tipic: Transformarea Fourier (TF) sau descompunerea în frecvență (în unde)	79
II.1.2. Transformări de imagini. Cazul special al transformărilor unitare.....	80
II.1.2.1. Transformări de imagini	80
II.1.2.2. Transformări unitare.....	81
II.1.3. Eșantionarea semnalului: precizia necesară și intercondiționarea reprezentărilor în Domeniul Frecvență și în Domeniul Frecvență-Timp	84
II.1.3.1. Precizia necesară reconstruirii semnalului	84
II.1.3.2. Intercondiționarea rezoluției reprezentărilor în Frecvență și în Frecvență-Timp: principiul nedeterminării.....	86
II.1.4. Concluzii și aplicații.....	90
II.1.4.1. Filtrarea în domeniul frecvență.....	90
II.1.4.2. Aplicații ale undinelor.....	91
II.1.5 O viziune abstractă asupra recunoașterii formelor, în jurul conceptului de corelare	96
II.2. Granularitate și gradație în Domeniul Fuzzy.....	99
II.2.1. Logica fuzzy și "oficializarea" unei alte paradigme de cunoaștere.....	99
II.2.1.1. Promisiunile începturilor erei informatice și problemele actuale ale inteligenței artificiale	99
II.2.1.2. Logica fuzzy, utilitatea și legătura ei cu problema clasificării	100
II.2.2. Domeniul posibilului: o fuziune fuzzy a diverselor tipuri de cunoștințe.....	106
II.2.2.1. Teoria fuzzy și teoria posibilităților.....	106
II.2.2.2. Teoria fuzzy și bazele de cunoștințe (sistemele expert) cu informații incerte / "granulate"	108
II.2.2.3. Concluzii	110
II.2.3. Conceptele de bază ale logicii fuzzy.....	112
II.2.3.1. Matematica fuzzy.....	112
II.2.3.2. Mulțimi și partiții fuzzy	114
II.2.3.3. Calificatori fuzzy ('hedges').....	127
II.2.3.4. Reguli fuzzy.....	128

II.2.2.5. Sisteme expert / de inferență de tip fuzzy	131
II.2.4. "Fuzzy logic = calculul cu cuvinte" [Zadeh, 1996].....	136
II.2.4.1. Termenul lexical și abilitatea de a opera cu informații bazate pe percepții sau intuiție	136
II.2.4.2. Scale fuzzy.....	139
II.2.5. PROTOTIPUL = clasa fuzzy	143
II.2.5.1. Prototipul ca sursă de potențial: metoda funcțiilor de potențial	144
II.2.5.2. Vedere sintetică a capitolului prin prisma termenului de prototip.....	146
IIIS Domeniul multispectral al imaginilor color	147
III.1. Introducere în colorimetrie.....	147
III.1.1. Principiul tricromaticității	148
III.1.2. Diagrama de cromaticitate CIE	150
III.2. O trecere în revistă a spațiilor de culoare.....	153
III.2.1. Sisteme de reprezentare primare	153
III.2.2. Sisteme de reprezentare derivate.....	154
III.2.3. Sisteme de reprezentare perceptuale	154
III.2.4. Sisteme de reprezentare perceptual uniforme.....	156
III.3. Caracterul perceptual al culorii	159
III.3.1. Metamerismul sau granularitatea spectrului vizibil	159
III.3.2. Constanța culorii și vederea prin 'retinex'	160
III.3.3. Teoria culorilor opuse.....	160
III.3.4. Fuzitatea culorii și procesarea multi-canal.....	161
IVS Contribuții practice	163
IV.1 Transformări eficiente între spații de culoare	163
IV.1.1. Conversia între spații de culoare pentru echipamente.....	164
IV.1.1.1. Principiul	165
IV.1.1.2. De la spațiul de culoare echipament la PCS	166
IV.1.1.3. De la colorimetria relativă a PCS la colorimetria absolută.....	167
IV.1.2 Implementarea conversiei.....	167
IV.1.2.1. Modulele aplicației.....	167
IV.1.2.2. Testarea acurateței și vitezei de calcul. Surse de erori (puncte slabe).....	168
IV.1.2.3. Tratarea erorilor. Optimizări.....	170
IV.1.3 Testarea conversiei.....	172
IV.1.4 Concluzii	174
IV.2 Definierea formei spațiului unei probleme de recunoaștere nedeterminate și modelarea acestuia. Cazul 'skin detection'.....	176
IV.2.1. Stadiul actual și problematica metodelor de detecție a pielii	177
IV.2.2. Alegerea spațiului de culoare și colectarea pixelilor de piele.....	179
IV.2.2.1 Alegerea spațiului de culoare	179
IV.2.2.2. Colectarea pixelilor de piele	180
IV.2.3. Reprezentarea Skin Locus-ului în vederea eficienței Skin Detection.....	181
IV.2.4. Modelarea distribuției SL	182
IV.2.4.1 Informații despre forma și probabilitate pentru un SL. Ideea algoritmului nostru	182

IV.2.4.2 Scurt breviar teoretic al metodelor de modelare a SL și comparația acestora cu metoda noastră.....	183
IV.2.4.3. Conectarea SL.....	187
IV.2.4.4. Codificarea SL.....	188
IV.2.5 Concluzii.....	189
IV.3 Folosirea subclasării și a marcajelor pentru reducerea fals pozitivelor. Cazul 'skin detection'.....	190
IV.3.1 Abordare.....	190
IV.3.2 Clasificarea skin / non skin.....	192
IV.3.2.1 Alegerea spațiului de culoare.....	192
IV.3.2.2 Stabilirea prototipurilor.....	194
IV.3.3 Condiții suplimentare clasificării skin / non skin.....	196
IV.3.3.1 Algoritm bazat pe variația specifică de luminanță pentru fiecare prototip al pielii.....	197
IV.3.3.2 Eliminarea zgomotelor și alinierea imaginii obținută fără 'flash' cu imaginea obținută cu 'flash'.....	199
IV.3.4. Discuție în urma experimentelor.....	204
IV.3.4.1. Precauții asupra metodei.....	204
IV.3.4.2. Câteva rezultate promițătoare.....	205
IV.3.5. Concluzii.....	207
<i>VȘ Sinopsis al contribuțiilor originale. Studiul viitor.....</i>	212
V.1. Vedere sinoptică a contribuțiilor noastre.....	212
V.1.1. Contribuții teoretice (per capitole).....	212
V.1.2. Contribuții practice.....	215
V.2. Studiul viitor.....	216
<i>BIBLIOGRAFIE.....</i>	218

IȘ Încadrarea problemei CI și descrierea ei în Domeniul Spațial și în Domeniul Probabilistic

I.1 Introducere

I.1.1 Scopul sistemelor Computer Vision și al CI

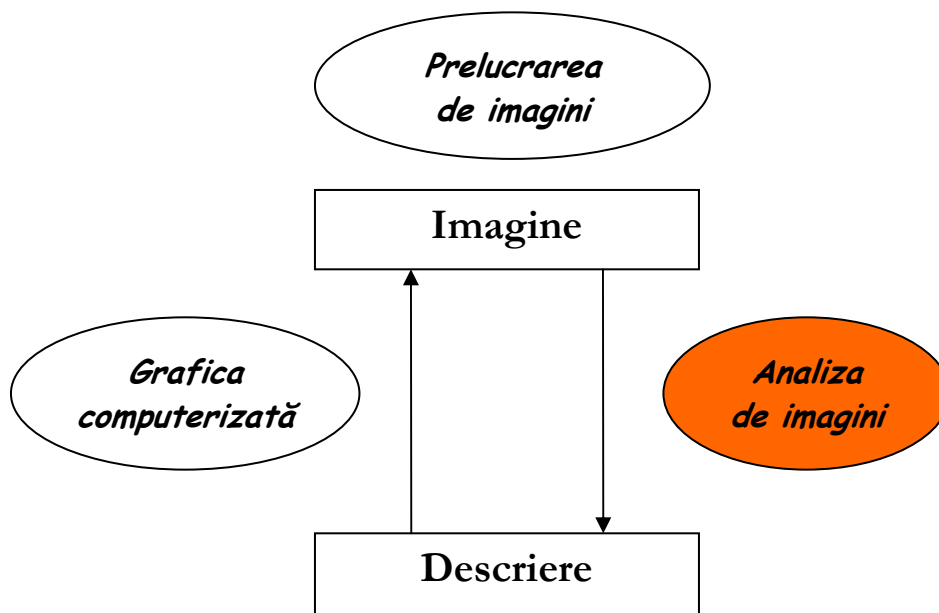


Fig. I.1.1_3 Schemă sintetică [Pavlidis 1982] a imagisticii computerizate

Imagistica computerizată ('Computer Image'), adică utilizarea calculatorului pentru operațiuni legate de domeniul vizualului (2D sau 3D) sau folosirea imaginilor prin intermediul calculatorului, ocupă o parte dominantă a 'Computer Culture', tendința fiind în creștere, mai ales după răspândirea în public a unor echipamente digitale ca televizorul, camera video, scanner-ul. În literatura de specialitate s-au consacrat ca 3 mari domenii:

1. Sinteza imaginilor / Grafica computerizată ('Image Synthesis' / 'Computer Graphics')

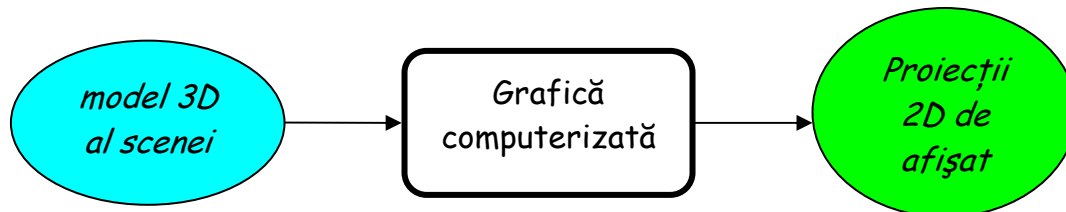


Fig. I.1.1_1 Intrarea și ieșirea graficii computerizate

Prescurtat, se folosește termenul de Grafică. Nu e aria noastră de interes în această lucrare. Aria mare în care ne încadrăm cu lucrarea de față o numim prescurtat: Imagistică. Să avem în vedere abuzul de limbaj ca efect al prescurtării: "Imagistica" nu e totuna cu "Imagistica computerizată". În continuare menționăm cele două subarii (vom vedea ca nu și partiții strict delimitate) ale imagisticii:

2. Prelucrarea imaginilor ('(Digital) Image Processing') :

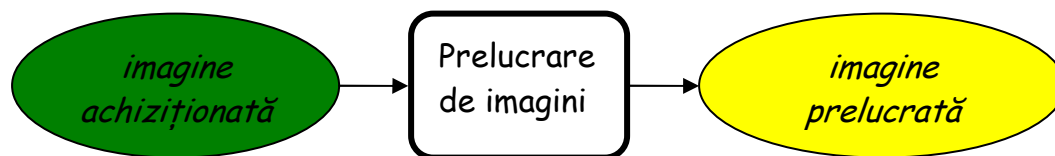


Fig. I.1.1_2 Intrarea și ieșirea prelucrării de imagini. Imaginea e prelucrată pentru a fi adusă într-o reprezentare cât mai adecvată unei analize de imagini, analiză presupusă de un sistem cu viziune.

Aici suntem într-un domeniu preliminar al problemei CI. Acest domeniu, cu "preproblemele" restaurării și îmbunătățirii imaginilor nu e aprofundat în lucrarea noastră, ci doar trecut în revistă.

3. *Computer Vision*, care s-a definit ca "preluarea de date-imagini brute și, pe baza acestor date, producerea unei acțiuni corespunzătoare categoriilor (claselor) datelor" [wikipedia] Sistemele cu viziune ('Computer Vision') se pot clasifica ca fiind de nivel:

- scăzut: sunt sisteme de prelucrare de imagini - intrarea este o imagine digitală, iar ieșirea o imagine prelucrată prin tehnici de restaurare și îmbunătățire
- intermediar: intrarea este o imagine prelucrată, iar ieșirea o reprezentare simbolică de nivel coborât (e.g., se extrag linii drepte, contururi)
- înalt: intrarea este o reprezentare simbolică de nivel coborât, iar ieșirea o reprezentare simbolică de nivel înalt (de ex., se extrag forme de obiecte).

În general, CI se încadrează la 3., presupunând și tehnici din 2..

CI își propune să clasifice informațiile obținute din imaginile de intrare pe baza unor cunoștințe apriorice sau pe baza unor informații statistice extrase din date.

Limitări ale CI

Deși calculatorul e indispensabil pentru analiza unei cantități mari de date și pentru calcule complexe, cum se întâmplă în general în cazul CI, pentru multe aplicații *critice*, cum ar fi cele din domeniul medicinei sau securității, analistul uman nu poate fi înlocuit de computer, chiar dacă s-au proiectat aparate inspirate din modelele de percepție vizuală umane. Cortexul vizual uman e, indiscutabil, un inegalabil aparat de analiză de imagini, în special pentru extragerea informațiilor de nivel înalt.

Problema fundamentală în CI e de a determina automat dacă o imagine conține sau nu un obiect, o caracteristică. Aceasta problemă, care poate fi soluționată natural de un om, pune de multe ori în impas un computer, mai ales în cazul general: obiecte arbitrare, în situații arbitrare. Metodele existente pot rezolva problema, în cazul cel mai bun, pentru obiecte *specifice* - e.g., obiecte geometrice simple (poliedre), caractere scrise de mână sau imprimare, vehicule, problema devenind nedeterminată și dificilă pentru fețe umane. Chiar și pentru aceste situațiile relativ simple se impun condiții: iluminare bine definită, mediu etc.

Denumiri

Taxonomiile nu sunt fixe. Oficial se spune că acest domeniu a fost dezvoltat în anii 1950 în S.U.A. pornind de la robotică și inteligență artificială.

‘Computer vision’, termen care stă în multe cazuri pentru CI, e acel domeniul larg, atât științific cât și tehnologic, al mașinilor care “văd”. Ca disciplină științifică, computer vision se preocupă de teoria necesară construcției de sisteme (artificiale) care să obțină informație semnificativă din imagini. Imaginile pot veni în multe forme: secvență video, proiecții de la mai multe camere, tomograf etc. CI e văzută și ca un câmp al inteligenței artificiale sau al științei calculatoarelor în general. Se înrudește cu prelucrarea de imagini, analiza de imagini, termenii suprapunându-se în multe cazuri. Tehnicile de bază care se folosesc în diversele domenii citate se aseamănă cel puțin ca principiu, ceea ce justifică interpretarea că, de fapt, e un singur domeniu cu mai multe nume, “afluenți”. E și din cauza grupurilor de cercetători pentru care e necesar și normal să-și încadreze activitatea unui câmp specific.

Se poate spune că analiza de imagini - termen sinonim cu CI - se încadrează în domeniul larg ‘computer / machine vision’, că e folosită din plin în industrie, că folosește geometria digitală și procesarea de semnal. Chiar și procese interne CI, pot avea o parte CI, e.g., restaurarea de imagine – analizând inițial imaginea în termeni de structură locală (linii, muchii, regiuni) putem obține o metoda mai bună de diminuare a zgomotului.

Probleme concrete de CI. Aplicații tipice



O aplicație software de CI care, în multe cazuri, ne vine odată cu aparatul de fotografiat achiziționat:

Fig. I.1.1_4 Recunoașterea/detecția feței

Au devenit de uz comun aplicațiile OCR (‘Optical Character Recognition’) și, mai nou, OMR (‘Optical Markup Recognition’ – e.g., corectarea testelor grilă prin recunoașterea marcajelor)

CI, după scop general, prezintă următoarele categorii:

- Recunoaștere: unul sau câteva (clase de) obiecte “știute” se recunosc, împreună cu poziția lor în imaginea 2D/scena 3D
- Identificare: o instanță individuală a unui obiect e recunoscută - e.g., identificarea amprentei unei persoane
- Detecție: imaginea e scanată pentru o condiție specifică - e.g., detecția unor celule anormale

Tehnicile CI se implementează intensiv în

- Robotică (roboți industriali, vehicule autonome etc.) sau în alte automatizări, prospectări:
- Inspecția vizuală a articolelor
- Recunoașterea și interpretarea obiectelor într-o scenă, urmate de acțiuni corespunzătoare - estimarea poziției/orientării : e.g un braț de robot poate fi ghidat de informații vizuale (robotul poate și citi de pe etichete);
- Evaluarea mișcării cu aspectele ‘*Egomotion*’ - determinarea mișcării 3D a camerei foto/video sau ‘*Tracking*’ - urmărirea traiectoriilor obiectelor
- Controlul proceselor și detectarea evenimentelor (supravegherea vizuală) – e.g., Interacțiunea om ↔ mașină: pornirea robinetului când trecem mâinile pe sub el
- Analiza de imagini medicale (imagistica medicală) – detecție și măsurători ale tumorilor, organelor, ‘*Computer Added Diagnosis*’
- Procesarea imaginilor achiziționate de la distanță (‘*Remote Processing*’): imagistica radar – detecția și identificarea țintei; cartografieri și prognoze meteo pe baza imaginilor satelitare, analize de trafic
- Expertiză (judiciară sau nu): potrivirea amprentelor, recunoașterea scrisului de mână, a vorbirii
- Organizarea informației vizuale (pentru baze de date de imagini) - Recuperarea imaginii după conținut (‘*Content-based image retrieval*’): găsirea tuturor imaginilor cu un conținut specific într-un set larg de imagini. Conținutul poate fi specificat în mai multe moduri, e.g., în termeni de similaritate relativ la o imagine reper sau în termeni de nivel înalt (“toate imaginile care conțin multe case”)
- Modelarea obiectelor și Reconstrucția scenelor: dându-se mai multe imagini ale unei scene se reconstruiește scena 3D; în cazul tomografiei computerizate, imaginile achiziționate sunt secțiuni interne ale corpurilor/obiectelor.

Organizarea unui sistem CV e foarte dependentă de aplicație. Anvergura problemei concrete dictează lungimea lanțului de subprobleme componente. Implementarea unui sistem CV depinde, de asemenea de posibilitatea învățării/modificării în timp real a anumitor funcționalități. Totuși, oricum s-ar prezenta datele problemei, există o succesiune de etape/ funcționalități tipice unui sistem CV. Le redăm în continuare în ordinea cronologică în lanțul procesării.

I.1.2 Organizarea unui sistem Computer Vision (CV / CI)

La modul general, obiectivul oricărui sistem de CV / CI se poate defini ca: analiză de imagini spre a extrage mai întâi informații cu semnificație și, apoi, producerea de *descrieri*, *interpretări* sau *înțelegeri* ale imaginilor de intrare.

Prima etapă e achiziția de imagini. Aceasta necesită un senzor de imagine și capacitatea de a digitiza semnalul. Deși acest prim pas are un rol determinant pentru o CI adecvată, în afară de a menționa cazul tomografiei computerizate - unde achiziția imaginii prezintă provocări și variabile interesante având la bază un aparat matematic și fizic remarcabil - în lucrare nu vom intra în detaliile acestei etape, de altfel, mai puțin “flexibile” a procesului de CI.

A doua etapă e preprocesarea imaginii: putem mări șansele de succes ale prelucrărilor ulterioare prin restaurarea (reducerea zgomotelor), îmbunătățirea imaginilor (e.g., îmbunătățirea contrastului sau accentuarea conturilor) sau izolarea zonelor de interes. Această etapă de prelucrare a imaginii va fi doar sumar detaliată în lucrare, întrucât ea e tratată până aproape de epuizare în literatura de specialitate.

Așadar, când nu vom explicita alte detalii, vom presupune că avem la intrarea sistemului de CI imagini digitale fără defecte care să afecteze major procesarea lor ulterioară. Următoarele etape/module definesc în principal sistemul de CI:

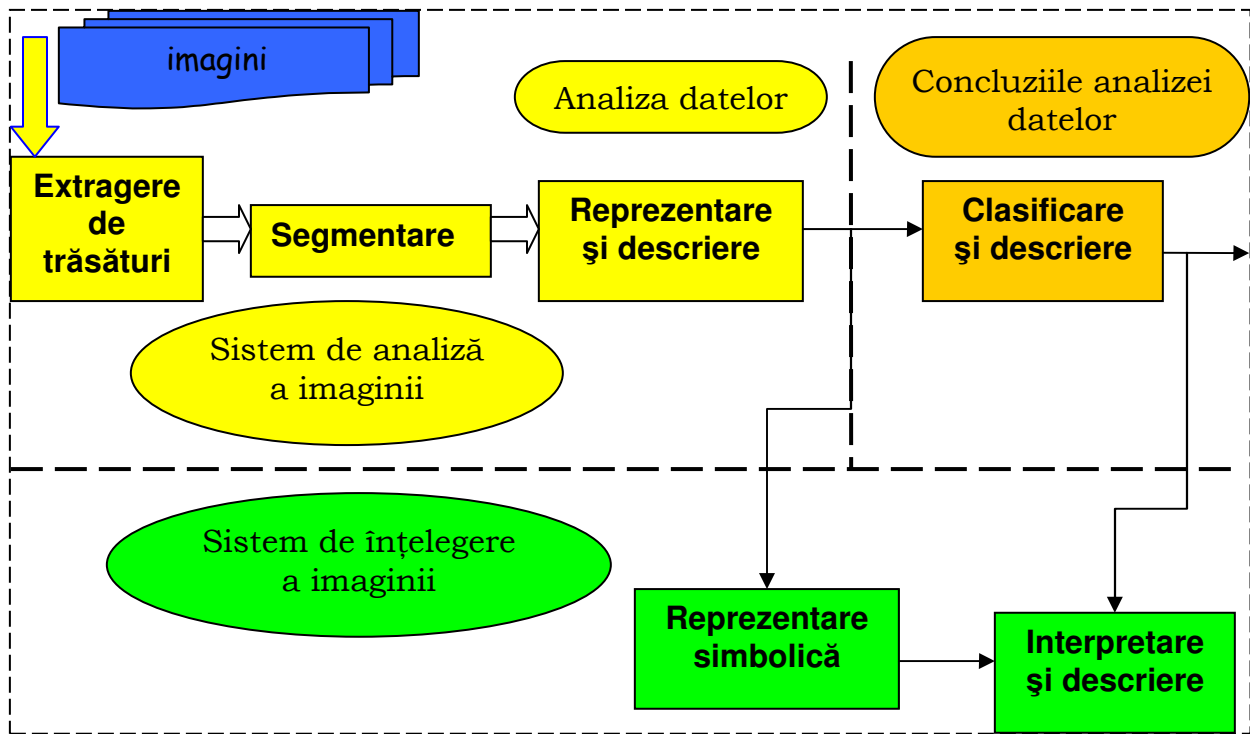


Fig. I.1.2_1 Schema unui sistem de CI

Segmentarea înseamnă partiționarea imaginii în zone relativ omogene. Ea presupune extragerea unor anumite trăsături în vederea acestei împărțiri, e.g., extragerea conturilor. Se poate ca problema de CI să se termine odată cu segmentarea - e cazul unor probleme concrete de detecție, selecție a unei zone de interes. Imaginea segmentată e reprezentată și descrisă într-o formă potrivită pentru a fi dată clasificatorului. Acesta trebuie să mapeze diverse regiuni (în cazul unei reprezentări interne) sau contururi (în cazul unei reprezentări externe) în clase de obiecte. Să reținem că, odată ieșiți din zona analizei datelor (cea marcată cu galben și galben-oranj), intrăm în zona concluziilor analizei datelor (cea marcată cu verde), unde pe baza unor reguli predefinite putem genera descrieri și interpretări utile calculatorului sau omului. De exemplu, în cazul unei probleme de sortare a merelor de pere, obiectele de formă mai rotundă vor fi clasificate ca mere, cele de formă mai alungită, ca pere. Sistemele de înțelegere a imaginii preiau o clasificare sau o reprezentare simbolică și determină relațiile dintre obiecte în scena 'input'. De exemplu, un asemenea sistem poate raporta: <<În scenă se văd 3 pere înconjurate de 7 mere.>>

Sistemele de înțelegere nu vor fi abordate în această lucrare – ele s-au consacrat mai degrabă în domeniul inteligenței artificiale, în timp ce noi ne propunem să abordăm CI dinspre partea analizei imaginilor.

Probleme specifice analizei de imagini

Analiza de imagini, înseamnă în principal operațiunile de: extragere de trăsături, segmentare și clasificare.

Extragerea de trăsături se referă la:

- trăsături spațiale
- trăsături ale transformatei Fourier, Hough
- muchii, colțuri și contururi
- trăsături de formă
- texturi

Segmentarea se poate baza pe:

- praguri (thresholding)
- acumulări (clustering)
- arbori cuaternari
- detecție de contururi (boundary detection)
- potrivire de șabloane (pattern matching)
- potrivire de texturi (texture matching)

Clasificarea se poate baza pe:

- statistici
- acumulări (clustering)
- arbori de decizie
- arbori de acoperire minimă
- măsuri de similaritate

În acest capitol trecem în revistă aceste operațiuni în domeniul spațial al imaginii pe care o reprezentăm matematic cu o matrice 2-D. Cea mai simplă și probabil cea mai folosită trăsătură spațială e *amplitudinea*. Amplitudinea se regăsește ca proprietate fizică ca atare sau ca nivel de

gri/luminozitate (când nu facem alte precizări explicite, presupunem că lucrăm cu imagini în scară de gri – cu imagini color vom lucra începând cu capitolul III). Ca proprietăți fizice ce pot da o amplitudine exemplificăm: reflectivitatea, refractivitatea, valorile tristimulus care dau culoarea. În cazul nivelului de gri vorbim despre trăsăturile de histogramă.

1.2. Detecția muchiilor (edge detection)

Aceasta e o problemă fundamentală în analiza de imagini, deoarece muchiile caracterizează contururile obiectelor, iar identificarea obiectelor, segmentarea și, eventual, alinierea obiectelor sunt scopurile analizei de imagini.

Numim pixelii care aparțin unei muchii, *pixeli de muchie (sau de contur, 'edge pixel')*. Acești pixeli sunt caracterizați de schimbări “mari” ale nivelelor de gri în vecinătatea lor spațială. Se pot detecta pixeli de muchie ca:

- pixeli în care nivelul de gri se schimbă brusc
- în imaginile binare: pixeli negri cu cel puțin un vecin direct (= vecin lateral, nu pe diagonală) alb.

Ceea ce e important pentru detecția muchiilor e rata de variație a nivelelor de gri, deci derivata funcției imagine – o variație lentă nu poate corespunde unei muchii.

În detecția de muchii putem folosi operatori *gradient* sau operatori *busolă (compass)*, care sunt operatori de diferențiere de ordinul 1. Putem folosi și operatori de diferențiere de ordinul 2, ca operatorul Laplace. Se mai folosesc și gradienti stocastici, care iau în considerare prezența zgomotului și se prezintă ca o alternativă mai bună față de mediere.

!! Condiția fundamentală a oricărui filtru folosit la detecția muchiilor e ca suma elementelor / ponderilor filtrului să fie 0 (e condiția duală celei de la filtrele de netezire (“blurring”): suma să fie 1) – se asigură “ieșirea” 0 pentru zone uniforme. Pe baza acestei condiții se pot proiecta filtre.

De exemplu, ținând cont de dezvoltarea binomului $(1 - \rho)^4$ se definește filtrul de accentuare a maximelor:

$$\frac{1}{(1+\rho^2)^2} \begin{bmatrix} \rho^2 & -\rho(1+\rho^2) & \rho^2 \\ -\rho(1+\rho^2) & (1+\rho^2)^2 & -\rho(1+\rho^2) \\ \rho^2 & -\rho(1+\rho^2) & \rho^2 \end{bmatrix}$$

, unde ρ e o măsură a corelației existente între pixelii adiacenți. Conform [Bulea 1, 2003] această metodă permite accentuarea semnificativă a maximelor funcției de intercorelație, pentru valori apropiate de 1 ale parametrului ρ (vezi și I.6.1.).

I.2.1. Folosirea derivatei de ordinul 1

I.2.1.1. Principiul

Pentru o imagine continuă $f(x,y)$, pixelul de contur e caracterizat de un maxim al amplitudinii vectorului gradient pe direcția perpendiculară pe aceea a direcției muchiei. Pentru o direcție oarecare, dată de versorul \vec{r} , de orientare θ față de orizontală, gradientul lui f în (x,y) e:

$$f_{\vec{r}} = \frac{\partial f(x,y)}{\partial \vec{r}} = \frac{\partial f(x,y)}{\partial x} \frac{\partial x}{\partial \vec{r}} + \frac{\partial f(x,y)}{\partial y} \frac{\partial y}{\partial \vec{r}} = f_x(x,y) \cos \theta + f_y(x,y) \sin \theta$$

Maximizarea gradientului revine la determinarea orientării θ ce anulează derivata expresiei de mai sus:

$$\frac{\partial}{\partial \theta} \left(\frac{\partial f(x,y)}{\partial \vec{r}} \right) = 0 \Rightarrow \theta_M(x,y) = \arctg \frac{f_y(x,y)}{f_x(x,y)},$$

pe această direcție modulul gradientului fiind dat de:

$$\left(\frac{\partial f(x, y)}{\partial r} \right)_{\max} = \sqrt{f_x^2(x, y) + f_y^2(x, y)}$$

Ultimele două relații constituie elementele pe baza cărora se pot determina muchiile în imagini.

Operatori gradient

Acești operatori măsoară diferențele pe direcții ortogonale.

Un exemplu este operatorul Sobel, definit prin următoarea mască:

$$\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

Operatorul “măsoară” pixelii de muchie verticală.

O pereche de măști (H1,H2) măsoară muchiile verticale, respectiv orizontale, rezultând prin convoluție cu aceste măști imaginile G1 și G2. Se obține:

- matricea gradient G cu elementele $g = \sqrt{g_1^2 + g_2^2}$

- matricea de direcții (maximele locale în această matrice corespund direcției muchiei)

$$\theta_g = \arctg(g_2 / g_1)$$

Din G se poate obține “harta muchiilor” (edge map), punând un prag stabilit pe baza histogramei cumulative de ex., impunem ca 5-10% din numărul total de pixeli cu cel mai mare gradient să fie declarați pixeli de muchie.

Observatie: sumele locale scad “eficiența” zgomotului.

Operatori busolă (compass)

Aceștia sunt în esență tot operatori gradient: măsoară diferențele într-un număr selectat de direcții ce nu sunt neapărat ortogonale.

Un exemplu de mască/filtru e $\begin{bmatrix} -1 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$: măsoară pixelii de muchie la 45 de grade. Pot

crește rezoluția unghilară, crescând dimensiunea măștii.

Fie un operator busolă în direcția $\theta_k = \frac{\pi}{2} + k \frac{\pi}{4}$ (din 45 în 45 de grade). Se calculează gradientul

g ca $g = \max\{|g_k|\}$. În acest caz (45 de grade) numai 4 operatori sunt liniar independenți.

I.2.2. Folosirea derivatei de ordinul 2

I.2.2.1. Operatori Laplace

Acești operatori sunt mai fini, mai afectați de zgomot, și mai potriviți când tranziția nu e foarte abruptă. Dacă $f(x,y)$ e imaginea *laplacianul* se definește:

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

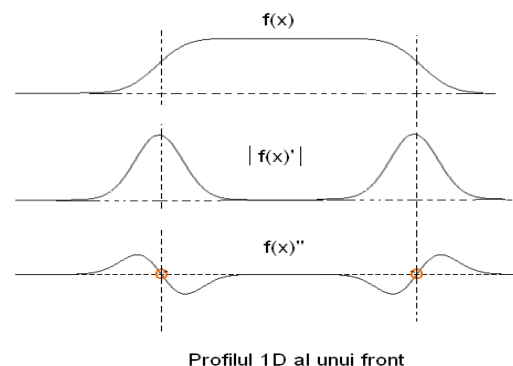
Se observă că laplacianul “pierde” informația de orientare.

Determinarea punctelor de muchie ca puncte de extrem ale laplacianului produce muchii duble, deoarece aceste puncte de minim/maxim sunt situate de o parte și de alta a regiunii de tranziție.

Ilustrăm pentru cazul 1D:

Fig. I.2.2_1 (conform Florica Moldoveanu, note de curs, Sisteme de prelucrare grafică). În cazul laplacianului, *trecurile prin 0* sunt candidate ca puncte de muchie.

Dată fiind “sensibilitatea” mai mare a laplacianului, în practică se folosește operatorul LoG = Laplaceian of Gaussian: laplacianul se aplică pe



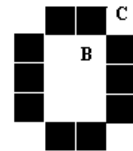
imaginea “blurată” cu un filtru trece jos (de netezire) de tip gaussian. Laplacianul poate fi preferat gradientului și datorită calității sale isotropice.

I.3. Extragerea conturilor

Conturul poate fi văzut ca un set de *muchii înlănțuite (linked edges)*, ce caracterizează forma obiectului.

I.3.1. Conectivitatea

Un pixel poate fi conectat numai cu vecini laterali și atunci el este 4-conectat, sau poate fi conectat și cu vecini de pe diagonală, caz în care este 8-conectat. E.g. pentru cazul din figura, pixelii negri formează un contur conectat-8.



I.3.2. Tehnici de extragere a conturului

Iată un sumar al acestor tehnici:

- *Urmărirea ('following') conturului* – are loc ordonarea pixelilor de muchie succesivi
- *Legarea muchiilor ('edge linking') și căutarea euristică în graful muchiilor ('heuristic graph searching')*: legarea muchiilor se face conform unor reguli prestabilite, trebuind maximizată alegerea următoarei muchii în graf
- *Programarea dinamică*: metoda de găsimă a optimului global în cazul unui proces în mai multe faze; *principiul optimalității* (Bellman): calea optimă între două puncte date e de asemenea optimă între oricare două puncte de pe cale.

- **Transformarea Hough** – în forma elementară găsește liniile drepte, în formă generalizată poate fi folosită, de exemplu, pentru găsirea conturilor. În fapt, **această metodă e reprezentativă pentru tehnicile de vot majoritar (tehnici de care vom face uz și noi în capitolul IV), pretându-se la o abordare prin prisma probabilităților, cu frecvența relativă pe post de probabilitate.**

Operatorul optimal de extragere a conturilor

Conform [Vertan & Ciuc, 2007, p.129], problema detecției conturilor pe baza prăguirii unei hărți de variații, harta fiind obținută prin filtrare liniară, poate fi văzută ca o “problemă de proiectare a unui filtru liniar, al cărui răspuns să marcheze cu maxime bine definite pozițiile tranzițiilor importante din imaginea-intrare. Filtrul dorit va fi optimal în sensul de a oferi cea mai bună performanță medie de detecție a muchiilor (pentru orice poziție și magnitudine a variației ce definește muchia și orice putere a zgomotului suprapus acestuia).” Pornind de la cazul unui semnal treaptă 1D, degradat cu zgomot alb, gaussian și aditiv, Canny a sintetizat un filtru FIR (‘Finite Impulse Response’ cu suport finit al răspunsului la impuls), caracterizat de un produs detecție x localizare cât mai apropiat de valoarea optimă: filtrul DoG = derivata (de ordinul întâi) de gaussiană.

1.4. Reprezentarea formelor

În general, termenul de *formă* se referă la aspectul (sau înfățișarea, conturul, silueta) unei figuri, neținând seama de dimensiuni. În analiza imaginilor, prin formă înțelegem o mulțime de pixeli din suportul imaginii, reprezentând o singură componentă conexă. Deci forma (plană) poate fi văzută ca o funcție binară de două variabile – coordonatele spațiale, cu suport compact și conex: $f(x, y) : A \subset \mathbb{N}^2 \rightarrow \{0,1\}$. Forma fiind o singură componentă conexă, problema reprezentării

formeii, în vederea recunoașterii ei, poate fi văzută și ca problemă a reprezentării conturilor, dar și a *regiunilor*.

Reprezentări mai simple ale conturilor sunt:

- *Codurile înlănțuite* ('*chain codes*'): se pleacă cu un pixel de start, indicându-se în continuare direcția pixelului următor

- *Aproximarea poligonală* ('*fitting line segments*'): se "segmentează" curbele, astfel încât abaterea să fie sub un prag ales conform preciziei cerute

- Curbe B-spline

Dacă vorbim de regiuni, acestea se pot reprezenta ca:

1. 'Secvențe de coduri' ('*run length codes*')

- avantaje: se obțin la o singură baleiere a imaginii

- dezavantaje: nu dau contururi

2. 'Arbori cuaternari' ('*quad trees*')

- avantaje: compresie

- dezavantaje: nu dau măsurători ale formeii ca perimetru, arie

3. Proiecții

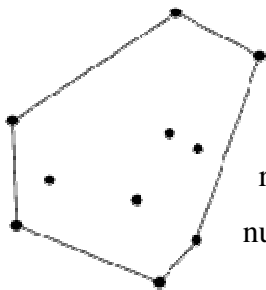
- e, mai ales, cazul specific al tomografiei computerizate

Vom trece în revistă și alte posibile reprezentări ale unei forme, cu adăugări și observații la [Vertan & Ciuc, 2007, p. 157-171]. Menționăm și că, în multe aplicații 'Computer Vision', obiectele pot fi caracterizate ca structuri compuse, e.g., din linii și arcuri, în general vorbind, din *elemente structurale* (*primitive*). Dând o *sintaxă* pentru elementele primitive, sintaxă bazată pe anumite *reguli de conectivitate*, putem obține o reprezentare unică a sceneii într-o *înșirare de simboluri* corespunzătoare primitivelor. Stabilirea regulilor de conectivitate, a unei *gramatici*, vine din *înțelegerea / interpretarea sceneii* ('*image understanding*') și poate fi dificilă, chiar imposibilă fără cunoștințe *a priori*.

Reprezentarea formelor aduce ca protagonist și prelucrarea morfologică a imaginilor. De așteptat: într-o reprezentare a formeii încercăm să "surprindem" structura ei.

Parametrii de formă sunt definiți astfel încât să se respecte invarianța la transformările geometrice afine uzuale (translație, rotație, scalare) și să fie asigurată o toleranță la zgomot (în caz de mici variații ale formei). În general, parametrii de formă sunt scalari sau funcții, cu observația ca există reprezentări ale formei obținute prin simplificări ale structurii acesteia (un exemplu de simplificare e chiar ‘aproximarea poligonală’ pomenită mai sus).

I.4.1. Simplificarea formelor



În contextul simplificării formelor trebuie menționată și proprietatea de *simetrie* a unei forme. Simetria poate fi de mai multe feluri: în oglindă, rotațională (aici avem și simetria rotațională de ordinul n , ‘ n -fold’ = obiectul nu se schimbă la rotația cu un unghi de $360^\circ / n$), elicoidală etc.

I.4.1.1. Anvelopa convexă

Pentru o formă plană A , *anvelopa convexă* (*convex hull*), $CH(A)$, e cea mai mică formă plană convexă ce include forma dată: $A \subseteq CH(A)$, cu $CH(A)$ mulțime convexă. Putem spune că anvelopa convexă e forma cea mai asemănătoare formei inițiale A , în care concavitățile conturului au fost “umplute.” Anvelopa convexă a oricărei mulțimi discrete e un poligon ale cărui vârfuri sunt o parte din punctele situate pe frontiera formei. Metodele clasice de determinare a acestor puncte sunt marșul Jarvis (*Jarvis march*) și baleierea Graham (*Graham scan*).

Pentru un set oarecare de puncte, care nu alcătuiesc neaparat o componentă conexă, metodele amintite se bazează pe găsirea următorului punct de pe CH , ordonând toate punctele rămase în mulțime după unghiul dintre segmentul ce le unește cu punctul curent de pe CH și direcția dreptei determinate de ultimele două puncte adăugate CH . Următorul punct de pe CH va avea

unghiul minim față de această direcție de referință. Algoritmul pleacă de la un punct extrem al mulțimii, ce aparține cu siguranță CH – de obicei se pleacă din primul punct întâlnit la baleiajul stânga-dreapta, sus-jos al mulțimii. În cazul unei forme plane, mulțimea de puncte pentru care se calculează CH se poate reduce la punctele de pe conturul forme.

O variantă aproximativă de construire a CH se bazează pe închiderea morfologică, știindu-se că aceasta operație are proprietatea de a umple concavitățile forme, dacă aceste concavități sunt mai mici decât elementul structurant:

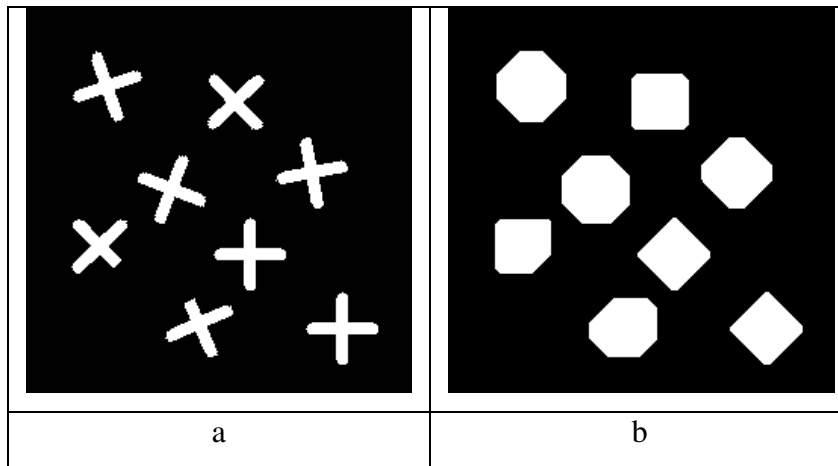


Fig.I.4.1._1. Închiderea morfologică (b), elementul structurant fiind un disc mai mare decât concavitățile din imaginea inițială (a)

I.4.1.2. Skeletoane

Skeletonul (sau *axa mediană*) e o prin linii, *regenerativă* a unei forme plane. S-a propus un model intuitiv de descriere a skeletonului, model care oferă și o sugestie de implementare a acestuia: modelul “focului de preerie.” Să ne imaginăm o suprafață de preerie acoperită omogen și izotrop cu iarba uscată. Într-o zi fără vânt, focul izbucnește în fiecare punct de pe marginea suprafeței de preerie. Cum materialul inflamabil – iarba uscată – se găsește numai în interiorul suprafeței, frontul de flacără se propagă numai în interiorul regiunii, consumând iarba uscată. La un moment dat, flăcările venind din direcții opuse, cu aceeași viteză, se întâlnesc și focul se stinge, nemaivând material de combustie. Punctele de întâlnire ale fronturilor de flacără (și respectiv punctele de stingere a focului) se găsesc pe axa mediană a regiunii.

Construirea skeletonului unei forme plane implică deci generarea setului de puncte aflate la distanță egală față de punctele cele mai apropiate de pe frontiera forme. Definierea matematică a skeletonului necesită definierea noțiunii de disc maximal într-o formă. Pantru o formă oarecare A,

discul maximal se definește ca discul de centru x și raza r , $B_x(r)$ ce e inclus în formă, fără a fi inclus în nici un alt disc inclus formei.

$$B_x(r) \subseteq B_{x'}(r') \subseteq A \Leftrightarrow \begin{cases} r = r' \\ x = x' \end{cases}$$

O formă poate avea mai multe discuri maximale, de dimensiuni diferite. *Skeletonul se definește ca mulțimea centrelor acestor discuri*. Ca exemple: skeletonul unui disc e centrul său, cel al unui pătrat e reuniunea diagonalelor sale.

Skeletonul morfologic

Calculul skeletonului unei forme în spațiul discret se poate face prin formula Lantujoul:

$$SK(A) = \bigcup_{n=0}^{N_{\max}} S_n(A), \quad S_n(A) \text{ fiind seturi skeleton:}$$

$$S_n(A) = (A \ominus nB) - (A \ominus nB) \circ B$$

Dimensiunea N_{\max} corespunde momentului în care seturile skeleton successive devin nule, moment marcat de $A \ominus N_{\max} B = \emptyset$; elementul structurant nB fiind iterarea de n ori a elementului structurant B , $nB = B \oplus \dots \oplus B$, iar \circ fiind operația de deschidere morfologică. B e în general un disc unitar. Seturile skeleton sunt disjuncte două câte două: $S_i(A) \cap S_j(A) = \emptyset, \forall i \neq j$.

Transformarea skeleton e *idempotentă* ($SK(SK(A)) = SK(A)$) și *antiextensivă* ($SK(A) \subseteq A$).

Reconstrucția (regenerarea) formei din skeleton se face după formula: $A = \bigcup_{n=0}^{N_{\max}} (S_n(A) \oplus nB)$

Se pot realiza și reconstrucții parțiale (aproximări ale mulțimii A) prin neglijarea seturilor skeleton ce reprezintă detaliile – a seturilor skeleton de ordin mic; aproximarea de ordin k e:

$$A = \bigcup_{n=k}^{N_{\max}} (S_n(A) \oplus nB)$$

Folosirea skeletonului morfologic pentru recunoașterea formelor e restricționată de puternica sa sensibilitate la zgomot: o mică schimbare a formei duce la o modificare mare a skeletonului – e.g., un disc cu centru decupat își va modifica skeletonul din centru într-o coroană circulară situată la jumătatea razei. În același timp, o problemă e reprezentarea în spațiul discret. Avem aici în vedere sensibilitatea skeletonului la rotații și scalări, dar și folosirea metricii euclidiene în definirea elementului structurant de tip disc, aproximat la pătrat sau cruce de pixeli. S-a propus folosirea unei clase de elemente structurante care să nu depindă de vreo metrică: *elemente structurante generalizate*.

Skeletonul generalizat

Fie $\{G_i\}$ un set de mulțimi, numit *set generator*. Elementele structurante generalizate sunt

definite recurent prin:

$$B_i = B_{i-1} \oplus G_i, i = 1, 2, \dots$$

$$B_0 = O_n$$

În general, se folosesc seturi generatoare de perioadă T ($G_{i+kT} = G_i, i, k \in Z$). Exemplu pentru T

= 1, deci pentru un singur element generator : $G = \begin{bmatrix} + & + \\ + & \oplus \end{bmatrix}$, obținem elementele structurante:

$$B_0 = \oplus, B_1 = \begin{bmatrix} \oplus & + \\ + & + \end{bmatrix}, B_2 = \begin{bmatrix} \oplus & + & + \\ + & + & + \\ + & + & + \end{bmatrix} \dots$$

Fiecărui pixel al formei A i se va asocia indicele +1 al elementului structurant generalizat maximal cu originea în pixelul respectiv, construind astfel o “hartă de distanțe”:

$$D(x) = \begin{cases} 0, x \notin A \\ n, (B_{n-1})_x \subseteq A \text{ \& } (B_n)_x \text{ nu } \subseteq A \end{cases}$$

Exemplificăm:

(în locurile neocupate sunt valori nule și am încadrat în “[]” pixelii de skeleton

$\begin{bmatrix} + & + & + & + \\ + & + & + & + \\ + & + & + & + \\ + & + & + & + \\ + & + & + & + & + & + \\ + & + & + & + & + & + \\ + & + & + & + & + & + \end{bmatrix}$	\xrightarrow{D}	$\begin{bmatrix} [4] & 3 & 2 & 1 \\ [4] & 3 & 2 & 1 \\ [4] & 3 & 2 & 1 \\ [4] & 3 & 2 & 1 \\ [3] & [3] & 3 & 3 & [3] & [3] & 2 & 1 \\ 2 & 2 & 2 & 2 & 2 & 2 & 2 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$
---	-------------------	---

determinați ca origini ale elementelor structurante generalizate maximale – cu proprietatea:

$$GSK(A) = \{x \in A \mid (B_{D(x)-1})_x \text{ nu } \subseteq (B_{D(y)-1})_y, \forall x \neq y\}$$

I.4.2. Funcții asociate formelor

I.4.2.1. Semnătura formei

E o funcție scalară de o variabilă, definită de distanța de la un punct de referință fixat x_0 (în general centrul de greutate al formei) la fiecare punct de pe conturul formei. Variabila e unghiul la centru θ realizat de punctul curent de pe contur cu axa orizontală de referință, $d_{x_0}(\theta)$, sau variabila poate fi abscisa curbilinie $\rho =$ lungimea conturului cuprins între punctul curent de pe contur și punctul în care axa de referință intersectează conturul. Semnătura formei e o reprezentare regenerativă.

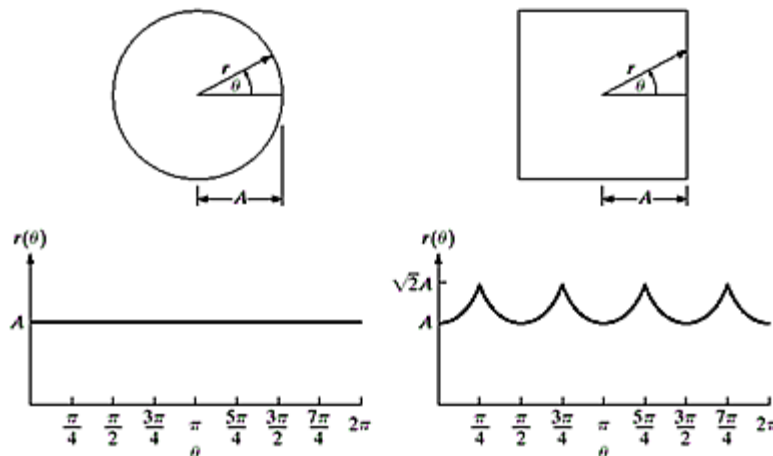


Fig.I.4.2._1 Exemplu de semnătură pentru discuri cu defecte la margine – defectul fiind reprezentat ca abaterea de la “linie” [Gonzalez & Woods, 2008, p. 863]

Varianta a doua de definire a variabilei funcție semnătură - ρ - e mai generală, întrucât se poate aplica și la formele concave, unde intersecția dreptei de direcție θ cu conturul poate să fie formată din mai multe puncte. În cazul semnăturii bazate pe unghi se impune restricția ca punctul de referință x_0 să aparțină *nucleului* formei. Nucleul formei se definește ca:

$$\text{Ker}(A) = \{x \mid \forall y \in A, \text{segmentul } [xy] \subseteq A\}$$

Se observă că în cazul formelor concave, nucleul formeii e mulțimea vidă.

Pentru compararea semnăturilor provenind de la forme diferite, domeniul și codomeniul funcției semnătură trebuie normalizate.

Folosind semnătura formeii putem defini parametrii de forma scalari de tip geometric. Un raport de simetrie, pentru formeii convexe, poate fi definit ca:

$$\text{Simetrie}(A) = \sup_{x \in A} \inf_{\theta \in [0, \pi)} \frac{d_x(\theta)}{d_x(\theta + \pi)}, \text{ iar un raport de circularitate ca}$$

$$\text{Circularitate}(a) = \frac{\sup_{\theta} h_A(\theta)}{\inf_{\theta} h_A(\theta)}, \text{ unde } h_A(\theta) = d_{x_0}(\theta) + d_{x_0}(\theta + \pi) \text{ se numește suportul formeii,}$$

fiind diametrul formeii pe direcția θ .

De asemenea, e posibilă marcarea unor puncte de extremitate de pe frontiera formeii (care aparțin aproximării poligonale a acesteia) prin identificarea punctelor de inflexiune ale semnăturii unghiulare – e.g., pentru un pătrat, vârfurile prezintă maxime de forma



I.4.2.2. Curba granulometrică

Reprezintă o funcție scalară de o variabilă reală pozitivă, definită de măsurarea ariei formeii prin “umplerea” formeii printr-o operație morfologică convenabil aleasă. Variabila reală ce indexează funcția granulometrică e dimensiunea elementului structurant folosit în umplerea morfologică.

Formal, granulometria asociată unei formeii A e secvența de mulțimi $\Phi_A(\lambda)$, λ fiind variabila, iar $\Phi_A(\lambda)$ reprezintă rezultatul prelucrării formeii A cu un element structurant de dimensiune λ .

Transformarea morfologică se alege astfel încât să fie îndeplinite condițiile:

Transformarea să fie crescătoare în raport cu forma pe care se aplică – pentru $A_1 \subseteq A_2$ și λ fixat : $\Phi_{A_1}(\lambda) \subseteq \Phi_{A_2}(\lambda)$;

transformarea să fie antiextensivă : $\forall \lambda \in R^+, \Phi_A(\lambda) \subseteq A$ și

să verifice relația : $\forall \lambda, \mu \in R^+, \Phi_{\Phi_A(\lambda)}(\mu) = \Phi_{\Phi_A(\mu)}(\lambda) = \Phi_A[\max(\lambda, \mu)]$

Ultima condiție implică idempotența transformării – pentru $\lambda = \mu$.

Cea mai simplă transformare ce respectă condițiile de mai sus e deschiderea, ceea ce duce la definirea granulometriei ca:

$$\Phi_A(\lambda) = A \circ \lambda B = (A \oplus \lambda B) \oplus \lambda B^s$$

De obicei, elementul structurant folosit – B – e un disc unitar.

Și curbele granulometrice, pentru a putea fi comparate, trebuie normalizate, atât în ceea ce privește domeniul, cât și codomeniul. Normalizarea codomeniului se face prin împărțirea ariei rezultatului transformării – presupunem cazul deschiderii - la cea mai mare închidere morfologică. Astfel, pe ordonata curbei granulometrice vom avea:

$$\Omega_A(\lambda) = \frac{\text{aria}[\Phi_A(\lambda)]}{\text{aria}\left[\lim_{n \rightarrow \infty} (A \bullet \lambda B)\right]}$$

După cum rezultă din definiția curbei granulometrice, aceasta e descrescătoare și invariantă la: translație, rotație (proprietăți ale deschiderii și închiderii morfologice), scalare (prin normare).

Să precizăm că termenul de “curbă granulometrică” e uzitat și în alte domenii, cum ar fi compoziția materialelor. Deși curba granulometrică se “instanțiază” altfel în acest domeniu, considerăm ilustrativ următorul citat: ”*curba granulometrică a pământului* e reprezentarea grafică a cantității de pământ (în procente din greutate) dintr-o anumită probă, funcție de diametrul particulelor care alcătuiesc proba - figura de mai jos. Ea se determină prin cernerea probei succesiv prin site cu dimensiunile ochiurilor din ce în ce mai mici și cântărindu-se greutatea materialului care nu a trecut prin ochiurile sitei.

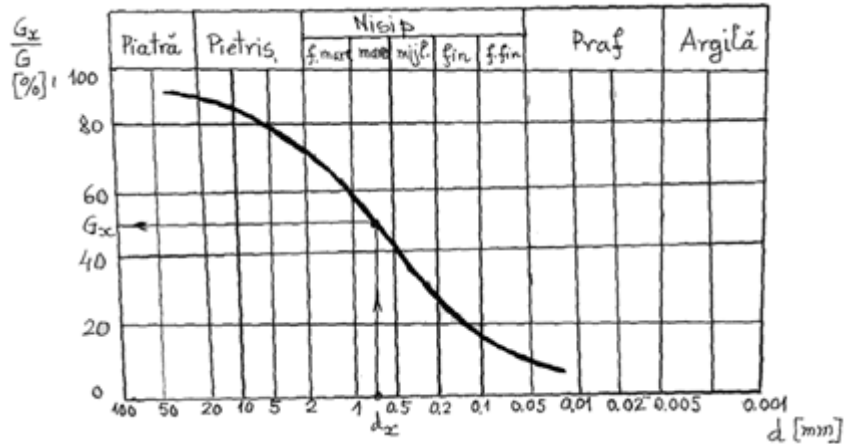


Fig. I.4.2_2 Curbă granulometrică [crb grn] Alura curbei granulometrice ne dă indicații asupra gradului de uniformitate a materialului.

Materialele trebuie să conțină fracțiuni din toate diametrele: prea mult material fin nu e indicat, deoarece e ușor spălat de apă, iar o cantitate prea mare de material grosier mărește gradul de permeabilitate. Se definește chiar un *indice de uniformitate a materialului*:

$u = d_{60\%} / d_{10\%}$, în funcție de ale căruia valori materiale pot fi:

- uniforme dacă $u < 5$;
- cu uniformitate mijlocie pentru $5 < u < 15$;
- neuniforme pentru $u > 15$.”

I.4.3. Parametrii de formă scalari

I.4.3.1. Parametri geometrici (rapoarte de aspect)

Această categorie de parametri se bazează pe măsurarea unor atribute geometrice simple ca: aria S , perimetrul P , numărul de găuri, numărul lui Euler (diferența dintre numărul de componente conexe și numărul de găuri), dimensiunile maxime ale formei pe direcție orizontală (D_h) sau verticală (D_v), dimensiunile cercurilor înscrise (r) și circumscrise (R) formei. Evident, atribute ca

S, P, r, R nu pot fi parametri de formă invariante la scalare, dar pe baza lor se pot construi rapoarte invariante la scalare, în intervalul [0,1].

Raportul de compactitate (numit uneori și factor de formă) e definit ca:

$$k = \frac{4\pi S}{P^2}$$

Pentru disc, raportul e 1; cu cât k e mai apropiat de acesta valoare, cu atât mai mult seamănă a disc (pătratul are raportul de compactitate 0.785).

Excentricitatea sau *circularitatea* formei (măsura în care se deosebește de disc) e definită ca raportul razelor

$$e = \frac{r}{R}$$

Ne amintim aici și rapoartele de simetrie și circularitate definite pe baza semnăturii formei.

Raportul de alungire e definit ca raportul dimensiunilor maxime ale formei pe verticală și orizontală $a = \min\left\{\frac{D_v}{D_o}, \frac{D_o}{D_v}\right\}$.

Observăm că acest raport e dependent de orientarea formei, nefiind invariant la rotația plană.

Raportul de convexitate e definit de raportul dintre aria formei și a anvelopei conexe a acesteia; raportul e 1 pentru formele conexe și scade cu cât forma e “mai concavă” :

$$c = \frac{\text{aria}(A)}{\text{arie}[CH(A)]}$$

În condițiile în care anvelopa conexă poate fi aproximată prin închiderea maximă a formei (după cum s-a discutat în secțiunea dedicată anvelopei convexe), c e valoare în origine a curbei granulometrice.

I.4.3.2. Momente statistice și invarianți

Interpretând funcția caracteristică a formei $(f(x,y):R \rightarrow \{0..L-1\})$ normalizat la $[0,1]$, R fiind regiunea care definește suportul formei) ca pe o funcție de densitate de probabilitate 2D, putem defini *momentele statistice de ordin $p+q$ sau p,q* pentru cele două variabile aleatoare corespunzătoare coordonatelor punctelor formei:

$$m_{pq} = \iint_R f(x, y) x^p y^q dx dy$$

Scalarul m_{pq} e proiecția funcției $f(x,y)$ pe polinoamele x^p și y^q ale bazei complete de polinoame.

Teorema reprezentării cu momente afirmă că mulțimea infinită de momente m_{pq} determină unic $f(x,y)$ și reciproc.

În cazul imaginilor binare, formula momentelor devine:

$$m_{pq} = \sum \sum_{f(x,y) \neq 0} x^p y^q$$

Caracterizarea formei printr-o serie infinită de numere – conform teoremei reprezentării cu momente, fiind imposibilă, în practică se folosesc pentru aproximare serii de momente până la un anumit ordin : $p + q \leq N$. Acestea însă caracterizează o altă funcție $g(x,y)$, o aproximare a lui $f(x,y)$, dată de o combinație liniară a polinoamelor bazei, ponderate de necunoscuții scalari g_{pq} :

$$g(x, y) = \sum \sum_{p+q \leq N} g_{pq} x^p y^q .$$

Găsirea acestor scalari se face prin egalarea momentelor cunoscute ale lui $f(x,y)$ cu momentele lui $g(x,y)$. Folosind o bază ortogonală de poligoane, cum sunt polinoamele Legendre, sistemul de ecuații devine mai simplu de rezolvat.

Trebuie însă remarcat că, folosirea momentelor statistice definite ca mai sus pentru caracterizarea unei forme nu asigură îndeplinirea nici unuia dintre principiile de invarianță căutate; de aceea, au fost introduse momente statistice invariante. Momentele statistice invariante la translație sunt *momentele statistice centrate*:

$$\mu_{pq} = \sum \sum_{f(x,y) \neq 0} (x - \bar{x})^p (y - \bar{y})^q ,$$

unde \bar{x} și \bar{y} sunt mediile celor două coordonate sau coordonatele *centrului de greutate* – date de

$$\frac{m_{10}}{m_{00}} \text{ și } \frac{m_{01}}{m_{00}}, m_{00} \text{ fiind numărul de pixeli sau aria formei.}$$

Momentele statistice invariante la translație, dar și la scalare, denumite *momente statistice normalizate*, sunt date de:

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^\gamma}, \gamma = 1 + \frac{p+q}{2}$$

Pe baza momentelor centrate normalizate s-au găsit șapte momente invariante la translație, scalare și rotație, iar în condițiile folosirii unor momente de ordin maxim trei. Invarianțele bazate pe momente pot fi exprimate mai convenabil folosind momente Zernike ce sunt proiecții ale lui $f(x,y)$ pe clasa polinoamelor Zernike (separabile în coordonate polare și ortogonale pe cercul unitate).

Inițial, în anii '60, acești invarianți au fost folosiți pentru recunoașterea caracterelor mari de tipar, cu rezultate modeste. Eficiența lor constă însă în modul rapid de calcul și posibilitatea de a le utiliza cu succes pentru recunoașterea formelor geometrice convexe.

Folosind momentele, se mai pot deduce alte atribute: excentricitatea suprafeței, care măsoară gradul de uniformitate al distribuției punctelor formei în jurul centrului de greutate:

$$\varepsilon = \frac{(\mu_{20} - \mu_{02})^2 + \mu_{11}}{\text{arie}} .$$

Un parametru geometric interesant e și *orientarea suprafeței*, caracterizată de unghiul θ față de orizontală pentru care momentul de inerție al formei e minim: $\theta = \frac{1}{2} \arctg\left(\frac{2\mu_{11}}{\mu_{20} - \mu_{02}}\right)$. Pe baza

orientării, calculând coordonatele extreme în urma rotirii cu unghiul θ , se determină *dreptunghiul de încadrare*.

I.4.3.3. Descriptori Fourier

Aceștia pleacă de la ideea că un contur închis poate fi văzut ca o funcție complexă periodică, de perioadă N ce reprezintă lungimea conturului. Deci putem scrie:

$$u(n) = x(n) + iy(n), n = 0..N - 1$$

În termeni Fourier, conturul devine:

$$u(n) = \frac{1}{N} \sum_{k=0}^{N-1} a(k) e^{j2\pi kn / N} : \text{TF inversă, coeficienții } a(k), \text{ numiți } \textit{descriptori Fourier (FD)} \textit{ ai}$$

$$\textit{conturului, fiind dați de TF directă: } a(n) = \frac{1}{N} \sum_{k=0}^{N-1} u(k) e^{-j2\pi kn / N} .$$

DF sunt o reprezentare a conturilor, care s-a dovedit a fi foarte utilă în problema recunoașterii caracterelor.

Transformări geometrice ale DF

În cazul transformărilor geometrice ale conturului, DF se modifică după cum urmează:

- translație: DF rămân aceiași cu excepția lui $a(0)$:

$$u_T(n) = u(n) + \Delta_{xy} \Rightarrow a_T(k) = a(n) + \Delta_{xy} \delta(n)$$

- scalare: DF se scalează cu același factor (pentru α real, faza rămâne invariantă)

- rotație: DF se rotesc cu același unghi θ_0

schimbarea punctului de start: DF își schimbă faza = se modulează:

$$u_p(n) = u(n - n_0) \Rightarrow a_p(n) = a(n) e^{-j2\pi n_0 n / N}$$

Observații:

- amplitudinile sunt invariante la transformările de rotație și schimbare a punctului de start

- faza e invariantă la scalare

- DF sunt caracteristici de formă regenerative, care se folosesc pentru recunoașterea formelor, mergând cu calculul DF până la precizia dorită – precizia e dată de rangul ultimului DF calculat.

DF se pot folosi și pentru:

- analiza curbelor / liniilor – le putem considera contururile unor obiecte cu grosimea de un pixel

- calculul ariei unei suprafețe

- skeletizare

- ‘potrivirea conturilor’ (*‘boundary matching’*); putem folosi DF pentru a potrivi forme similare, chiar dacă au dimensiuni și orientări diferite. Se poate stabili un prag de eroare între potrivirea coeficienților DF ai celor două contururi ale formelor de comparat, prag care să minimizeze efectele transformărilor geometrice descrise mai sus:

$$d(u_0, \alpha, \theta_0, n_0) \stackrel{def}{=} \min \left\{ \sum_{n=0}^{N-1} \left| u_n - \alpha v(n + n_0) e^{j\theta_0} - u_0 \right|^2 \right\}$$

Variante ale DF

În loc să folosim două funcții ($x(t)$ și $y(t)$), putem folosi o singură funcție cu parametrul $t =$ lungimea arcului de-a lungul curbei:

$$\theta(t) = \text{tg}^{-1} \left[\frac{dy(t)/dt}{dx(t)/dt} \right]$$

Curba se poate reconstrui dacă se cunosc $x(0)$, $y(0)$ și $\theta(t)$. De exemplu coordonata x e dată de:

$$x(t) = x(0) + \int_0^t \cos \theta(\tau) d\tau$$

Pentru a avea invarianță la translație, putem folosi ca funcție ce reprezintă conturul, *curbura* conturului:

$$k(t) = \frac{d\theta(t)}{dt}$$

Pentru curbele închise cu colțuri s-a propus folosirea în colțuri a funcției:

$$\hat{\theta}(t) = \int_0^s k(\tau) d\tau - 2\pi / T, \text{ unde } T \text{ e perioada conturului (} N \text{ în cazul discret).}$$

Să menționăm și că la parametrii de formă geometrici se pot adauga *numărul de colțuri* – unde curbura depășește un prag, și *energia curbării (bending energy)* $E = \frac{1}{T} \int_0^T |k(t)|^2 dt$, care în

termeni de DF devine $E = \sum_{k=0}^{N-1} a(k)^2 \left(\frac{2k\pi}{T} \right)^4$.

I.4.4. Concluzii

Reprezentarea formelor poate fi folosită la:

- Recunoașterea formelor
- Potrivirea formelor
- Măsurarea formelor

Măsurarea formelor poate fi văzută ca extragerea de caracteristici bazate pe geometria sau momentele formei. Pentru potrivirea și recunoașterea formelor, reprezentările cu adevărat “sigure”, care garantează identificarea formei, admițând transformare inversă (de la reprezentare la formă), sunt reprezentările regenerative, cum ar fi conturul, regiunea, reprezentarea structural sintactică, skeletonul, semnătura formei, descriptorii Fourier, aproximările cu momente. Ceea ce

decide metoda optimă de reprezentare folosită e impactul asupra acurateții, vitezei și costului aplicației.

1.5. Tehnici de segmentare

Pe scurt, segmentarea e procesul care partiționează imaginea în regiuni relativ omogene. Îi putem spune și recunoaștere a regiunilor, fiind în fond o clasificare de nivel jos, care pregătește clasificarea de nivel înalt ce va avea loc la finalul întregului proces de CI.

Vom trece în revistă tehnicile cele mai cunoscute de segmentare, ca pregătire pentru tehnicile de clasificare tratate ulterior, pentru că, așa cum am spus, segmentarea poate fi privită ca o clasificare.

Enumerăm tipuri de segmentare :

- Limitarea intensității ('amplitude thresholding/window slicing')
- Etichetarea componentelor ('component labeling')
- bazată pe contururi ('boundary based approaches')
- bazată pe regiuni ('region based approaches & clustering')
- bazată pe potrivirea de șabloane

Precizăm câteva aspecte din ultima categorie, segmentarea bazată pe potrivirea de șabloane, mai specială în felul ei, și, mai jos, trecem în revistă celelate tehnici, oprindu-ne la câteva aspecte pe care le considerăm interesante.

De segmentare cu șabloane putem vorbi în cazul separării textului de grafice într-o pagină, fiecare necesitând în continuare tehnici specifice de prelucrare. La această categorie de

segmentare includem și *segmentarea bazată pe texturi*, abordare care își poate dovedi valoarea în cazul imaginilor cu o *densitate mare de muchii*, caz în care abordarea bazată pe contur eșuează. Ca procedee de prelucrare, menționăm prelucrările morfologice, ca închiderea și deschiderea – putem intra aici în domeniul *granulometriei*, al cărui principiu general ar fi că “au efect” prelucrările morfologice cu elemente structurante de dimensiuni similare cu obiectele. Cazul mai rafinat al acestui segmentării cu șabloane e potrivirea imaginilor (‘image matching’), despre care vorbim în subcapitolul următor.

I.5.1. Limitarea intensității / Prăguirea (‘Thresholding’)

Datorită proprietăților sale intuitive și simplității implementării, având și atuu unei viteze de calcul deosebite, prăguirea ocupă o poziție centrală în segmentarea imaginilor.

Prăguirea e efectivă când intensitățile pixelilor caracterizează suficient obiectul, la modul ideal aceasta însemnând că un interval de intensități identifică unic obiectul. Zgomotul și iluminarea / reflectanța neuniformă pot defavoriza major prăguirea.

Iată un simplu algoritm iterativ de prăguire, care “merge” (eficient) când obiectele din imagine sunt suficient de distincte față de fundal, astfel încât putem aplica un singur prag global:

Alege T = un prag global (dacă nu avem vreo recomandare specială, alegem chiar media imaginii)

Partiționează imaginea în două (clase) pe baza lui T : $C1$ și $C2$

Atribuie $T_{\text{nou}} = \text{media} [\text{media}(C1) + \text{media}(C2)]$

Repetă pasul 2. cât timp $(T - T_{\text{nou}}) < \text{un parametru } DT \text{ dat}$ (cu cât DT e mai mare, cu atât vor fi mai puține iterații)

Abordările mai avansate se bazează în general pe histograma imaginii:

- histograma e examinată pentru a localiza “piscuri și văi” – care pot da valorile pragurilor de delimitare a obiectelor
- se poate aplica un prag procentual pe ordonata histogramei cumulative e.g., am fețe umane încadrate manual cu poligoane și vreau să determin pixelii “de piele” din poligon
- praguri adaptive în funcție de histogramele vecinătăților
- delimitare pe baza proiecțiilor: analiza histogramele proiecțiilor pe fiecare direcție (corespunzătoare fiecărei coordonate, deci x și y) sau histogramele pixelilor care satisfac un anumit criteriu
- determinarea pragurilor astfel încât să minimizăm eroarea de segmentare

În continuare, vom prezenta o metodă generală de prăguire care se încadrează la ultimul punct. Atenția specială acordată acestei metode provine din faptul că metoda își găsește “natural” un corespondent în clasificatorul de tip Bayes și că apar aspecte posibil relevante pentru o problemă CI abordată probabilistic.

I.5.1.1. Metoda Otsu (de prăguire globală optimă)

Această metodă e optimă în ce privește maximizarea varianței între clase. Un alt avantaj: se bazează doar pe histograma imaginii.

Presupunem L nivele de intensitate ale imaginii: de la 0 la $L-1$. Notăm p_i frecvența relativă a nivelului $i = \frac{\text{nr. de pixeli cu intensitatea } i}{\text{nr. total de pixeli}} \rightarrow \sum_{i=0}^{L-1} p_i = 1$

Fie pragul $k \in (0, L - 1)$ și $C1$ și $C2$ clasele (regiunile) rezultate în urma aplicării acestui prag. Probabilitatea ca un pixel să fie în $C1$ e $P1(k) = \sum_{i=0}^k p_i$, iar $P2(k) = 1 - P1(k)$.

Intensitatea medie a pixelilor asigurați lui $C1$: $m1(k) = \sum_{i=0}^k i P(i/C1)$

Aplicând legea lui Bayes:

$$m1(k) = \frac{1}{P1(k)} \sum_{i=0}^k i p_i$$

Analog:

$$m2(k) = \frac{1}{P2(k)} \sum_{i=k+1}^{L-1} i p_i$$

Media globală, intensității pe întreaga imagine nu depinde de k:

$$mG = \sum_{i=0}^{L-1} i p_i$$

Din cele trei relații de mai sus reiese: $P1(k) m1(k) + P2(k) m2(k) = mG$.

Evaluarea pragului se face cu:

$$\eta(k) = \frac{\sigma_B^2(k)}{\sigma_G^2}$$

, unde $\sigma_B^2(k) = P1(k)[m1(k) - mG]^2 + P2(k)[m2(k) - mG]^2$ e varianța între ('between') cele două clase, iar $\sigma_G^2 = \sum_{i=0}^{L-1} (i - mG)^2 p_i$ e varianța globală.

Se mai poate scrie – ignorăm k prezent la toți termenii:

$$\sigma_B^2 = P1P2(m1 - m2)^2$$

Din această relație se vede cum cu cât "distanța" între mediile celor două clase e mai mare, cu atât $\sigma_B^2(k)$ e mai mare.

Dacă notăm $m(k) \stackrel{\text{def}}{=} \sum_{i=0}^k i p_i$, obținem o formulă mai eficientă pentru calculul σ_B^2 :

$$\sigma_B^2(k) = \frac{[mG P1(k) - m(k)]^2}{P1(k)[1 - P1(k)]}$$

Deci algoritmul Otsu alege k^* care să maximizeze $\sigma_B^2(k)$. Se demonstrează că acest maxim există – pot exista mai multe valori ale lui k pentru care se maximizează $\sigma_B^2(k)$.

Alte informații interesante (afară de k^*) care se obțin rulând algoritmul Otsu:

- $P1(k^*)$ și $P2(k^*)$: ariile ocupate de cele două clase în reprezentarea histogramei
- $m1(k^*)$ și $m2(k^*)$: media intensităților în interiorul celor două clase

$\eta(k) \in [0; 1]$: $\eta(k) = 0$ ⇔ imaginea e de intensitate constantă și $\eta(k)$ atinge maximul 1 ⇔ imaginea are intensități 0 și $L-1$.

Observații:

- o prăguire satisfăcătoare are loc în condițiile în care lobiile histograme sunt suficient de: înalți, înguști, simetrici și separați de “văi adânci.”
- în general, o prăguire globală afectată de zgomot poate fi îmbunătățită prin atenuarea detaliilor (‘smoothing’), chiar dacă pierdem precizia conturilor sau prin folosirea pixelilor de tranziție / muchie când avem un lob prea lat (folosind pixeli de muchie, segmentarea nu depinde de dimensiuni, astfel încât se poate lua în calcul și o regiune a cărei mică contribuție la histogramă ar fi altfel neglijată).

I.5.2. Etichetarea componentelor

Să luăm cazul elementar al algoritmului de *segmentare prin etichetarea pixelului* unei imagini binare. Presupunem că în imaginea baleiată în mod normal (stanga-dreapta, sus-jos), ajungem la pixelul X :

E	F	G
Q	X	\rightarrow
x	x	x

Am marcat vecinii necunoscuți încă, ai lui X, cu x. Dacă valoarea în X e 1, atunci e asignat la obiectul la care e conectat. Sunt patru variante posibile pentru o 8-conectivitate: E, F, G și Q. Dacă nici unul dintre vecinii parcurși ai lui X nu a fost asignat la vreun obiect, adică apare o tranziție din 0 către 1 odată cu întâlnirea lui X; tot astfel va apărea și o nouă etichetă de obiect. Odată ce pixelul a fost etichetat, caracteristicile obiectului (centru de greutate, arie) căruiua îi aparține sunt actualizate.

Desigur, se poate lucra și cu imagini în scară de gri și cu imagini color, dacă definim criteriul de apartenență al unui pixel la un obiect. Se poate lucra și pe o reprezentare de tip “secvență” (‘run length’), în algoritmi corespunzători, obiectelor fiindu-le asociate nivele.

Avantajul major al acestei abordări e că obținem o *hartă de segmentare* într-un singur pas, la o singură baleiere.

Aspecte legate de segmentarea bazată pe regiuni

Segmentarea bazată pe regiuni e, în general, tipul de segmentare cel mai versatil. Față de segmentarea bazată pe contururi, care se bazează de fapt pe detecția muchiilor și urmărirea conectivității, are avantajul că e mai robustă la zgomote (care alterează detecția muchiilor) și alte artefacte, la rupturi în contururi sau în situații de întretăiere a obiectelor (care alterează conectivitatea conturilor). Principiul segmentării bazată pe regiuni e divizarea inițială în regiuni *atomice* și *unificarea* acestora (‘*region growing*’). Regiuni adiacente similare se unifică secvențial până obținem regiuni adiacente suficient de diferite. Provocarea e alegerea criteriului de unificare. Iată câteva sugestii de alegere:

$w/P_m > T_1$, unde $P_m = \min(P_i, P_j)$ e perimetrul minim între cele două regiuni, iar w e numărul de *locații de contur slab* (‘*weak boundary locations*’) = pixeli de o parte și de alta a graniței între cele două regiuni, a căror diferență de amplitudine e sub un anumit prag.

Pragul T_1 controlează dimensiunea regiunii de unificat, e.g., pentru $T_1 = 1$ unificarea se face dacă o regiune aproape o include pe cealaltă.

$w/L > T_2$, unde L e lungimea graniței comune celor două regiuni.

În general, acest pas se aplică după ce o primă euristică a redus numărul de regiuni. O valoare tipică pentru T_2 e 0.75.

Unificăm dacă avem numai locații de contur slab pe graniță.

Dezavantajul segmentării bazate pe regiuni e că implementarea poate avea complexitate mare.

1.6. Clasificarea imaginilor prin potrivire ('Image Matching')

Detecția prezenței sau schimbării poziției unui obiect într-o scenă dată e o problemă de mare importanță în analiza de imagini. Domenii concrete de interes sunt robotica, prognozele meteo conform imaginilor obținute din sateliți, diagnosticele medicale conform imaginilor obținute prin radiologie sau tomografie computerizată etc. Detecția schimbării e utilă și în *alinierea* ('*alignement or spatial registration*') imaginilor "luate" la moment / din poziții sau cu aparate foto diferite.

Cea mai simplă metodă de detecție a schimbărilor e *scăderea imaginilor*. Deși rudimentară, această metodă dă rezultate bune în cazul unor condiții controlate de formare a imaginilor e.g., făcând diferența între imaginile medicale ale unui organ înainte și după injectarea unei substanțe "lizibile" la radiologie, în cadrul sistemelor de securitate bazate pe monitorizare, în segmentarea unei părți dintr-un ansamblu complex.

În esență, "potrivirea" unui model într-o imagine constă în căutarea minimului unei măsuri a diferenței între imagine și model (șablon). Prezentăm separat, în acest capitol, o metodă generală de potrivire, importantă și prin felul în care pune în lumină conceptul de **corelare**, concept asupra căruia vom reveni pentru a face o sinteză din punct de vedere matematic asupra

recunoașterii formelor ('Pattern Recognition') în secțiunea II.1.5, "O viziune abstractă asupra recunoașterii formelor, în jurul conceptului de corelare".

I.6.1. Potrivirea pe baza corelării ('Template Matching and Area Correlation')

Presupunem că vrem să găsim / potrivim într-o imagine $f(x,y)$ un șablon w de dimensiune semnificativ mai mică. Definim coeficientul normalizat de corelație:

$$\gamma(x, y) = \frac{\sum_s \sum_t [w(s, t) - \bar{w}][f(x + s, y + t) - \bar{f}]}{\sqrt{\left(\sum_s \sum_t [w(s, t) - \bar{w}]^2 \right) \left(\sum_s \sum_t [f(x + s, y + t) - \bar{f}]^2 \right)}}$$

unde \bar{w} e media intensităților pixelilor șablonului, iar \bar{f} se calculează ca medie a intensităților pixelilor imaginii, în dreptunghiul de suprapunere a acesteia cu w – procesul e analog corelării imaginii f cu masca w .

Conform inegalității lui Cauchy, $\gamma(x, y) \in [-1, 1]$ (dacă luăm $w(s,t)$ și $f(x+s,y+t)$ drept capetele a doi vectori într-un hiperspațiu – celălalt capăt e originea -, coeficientul normalizat de corelație e cosinusul unghiului dintre cei doi vectori). Ne asigurăm că avem o măsură normalizată a potrivirii a potrivirii șablonului în imagine, măsură independentă de diferența de scală între cei doi termeni de comparat. Numărătorul ar putea fi calculat ca produs, trecând în domeniul Fourier, pe baza teoremei convoluției (în cazul de față a corelării – ceea ce presupune doar o conjugare față de formula corespunzătoare convoluției). Dar coeficientul de corelație nu poate fi calculat în totalitate cu aceasta teoremă, din cauza termenilor neliniari (ridicare la pătrat și împărțire).

Conform aceleiași inegalități a lui Cauchy, γ atinge valoarea extremă (-1 sau 1) când w centrat e o scalare a lui f centrat: $w(s,t) - \bar{w} = k[f(x+s, y+t) - \bar{f}]$

Corelația extremă (γ maxim în modul) înseamnă cea mai bună potrivire posibilă. Rezultă imediat algoritmul de potrivire: parcurgerea incrementală a imaginii pentru corelația ei cu șablonul și reținerea poziției maximului (sau maximelor) ca poziție centrală de potrivire a șablonului în imagine. În acest context se poate folosi filtru definit la sfârșitul secțiunii I.2, “Detectia muchiilor”.

Conform [Bulea 1, 2003, p. 165]: “Volumul mare de calcule necesare pentru obținerea funcției de intercorelație (coeficientului normalizat de corelație) poate fi diminuat aplicând unele strategii. Rosenfeld și Vandenburg au propus reducerea într-o primă fază a rezoluției imaginilor ce se compară, fie prin subeșantionare, fie prin simpla selecție a unui subset de pixeli. Determinarea, folosind aceste imagini, a maximumului funcției de intercorelație, duce la o localizare grosieră a lui. Aceasta permite restrângerea semnificativă a dimensiunilor ferestrei de corelație. Barnea și Silverman propun calculul cumulativ al funcției de eroare pentru cele două imagini potrivite, oprindu-se calculul corelației dacă eroarea calculată, la un anumit pas, depășește un anumit prag.”

Mai sus am arătat că e posibilă normalizarea corelației pentru schimbarea intensității imaginilor potrivite. Normalizarea față de redimensionarea imaginilor și față de rotația lor sunt probleme mai complicate. Redimensionarea implică cunoașterea dimensiunii comune la care să fie aduse imaginile. Dacă luăm exemplul unei aplicații controlate la distanță, această dimensiune se poate deduce știind geometria senzorilor și poziția lor față de obiectele “fotografiate.” În mod asemănător, s-ar putea rezolva și normalizarea față de rotație, care presupune cunoașterea unghiului după care trebuie rotite imaginile. Dar, în general, potrivirea în spațiul parametrilor

$\left(\frac{x-s}{\gamma_1}, \frac{y-t}{\gamma_2}, \theta \right)$ unde numitorii și al treilea parametru corespund normalizării față de

redimensionare și, respectiv, rotire, devine imposibil dacă nu avem estimări rezonabile și dacă detecția trăsăturilor imagistice nu ne ajută îndeajuns. Generalizarea metodei de potrivire bazată pe calculul corelației constă în înlocuirea funcțiilor de imagine cu trăsături extrase din imagini, cum ar fi momentele invariante, coeficienții unor transformări ortogonale, etc. În acest sens,

putem încerca potrivirea pe baza unor puncte cheie ('control points'), detectate automat sau alese manual.

I.6.2. Cazuri particulare

I.6.2.1. Căutarea logaritmică

Dacă nu există informații *a priori* despre variația abaterii potrivirii șablonului în diferitele locații ale imaginii, suntem nevoiți să căutăm în toată imaginea, pixel cu pixel. Dacă dimpotrivă, știm, de exemplu, că abaterea crește monoton, pe măsură ce ne îndepărtăm de locația potrivirii, putem reduce costul căutării. În exemplul presupus, căutarea ajunge la complexitate $O(\log n)$ de la $O(n^2)$ dacă rulăm un algoritm care constă în:

Inițial) căutarea în cinci locații = centrul imaginii + mijloacele segmentelor definite de centru și mijloacele laturilor imaginii;

Apoi) căutare în cinci locații în jurul locației cu cea mai mică abatere și tot așa până putem ajunge la o căutare secvențială într-o zonă de 3x3 pixeli dacă nu găsim potrivire pe parcurs. Algoritmul presupune definirea unei funcții de abatere în fiecare pixel, a cărei valoare trebuie

minimizată :
$$A(i, j) = \frac{1}{MN} \sum_{m=1}^M \sum_{n=1}^N p[f(i-m, j-n) - w(m, n)],$$
 $p(x)$ fiind o funcție crescătoare,

pozitivă – se obișnuiește folosirea lui $|x|$ sau a lui x^2 .

I.6.2.2. Potrivirea imaginilor binare

Pentru potrivirea acestor imagini e suficient sa calculăm, ca masură de potrivire, diferența minimă:

$$\sum_m \sum_n f(i-m, j-n) \oplus w(m,n)$$

În practică, acest caz se întâlnește în multe situații e.g., recunoașterea caracterelor / obiectelor tipărite, a schemelor / circuitelor, etc.

1.7. Tehnici de clasificare

După extragerea de trăsături, segmentare, reprezentare și descriere, CI își propune să obțină o descriere simbolică de nivel înalt a conținutului imaginii. Să observăm că gradul de complexitate al problemei clasificării poate fi micșorat împărțind în subprobleme – vom proiecta un clasificator care să separe în grupe de clase.

Un sistem complet de recunoaștere a formelor constă într-un senzor care adună datele brute, un mecanism de extragere a caracteristicilor care calculează numeric sau simbolic informația din aceste date și o schemă de clasificare sau descriere a datelor pe baza caracteristicilor extrase. Reprezentarea formelor, pasul imediat anterior clasificării, e foarte importantă pentru clasificare. Ea poate fi cheia rezolvării unei probleme de clasificare. Așa cum am arătat în subcapitolul “Reprezentarea formelor”, e de preferat o reprezentare simplificată (dar nu simplistă !) pentru o clasificare performantă.

În general, schema de clasificare folosește două abordări: *statistică* (sau *decizional teoretică*) – bazată pe caracteristicile statistice ale datelor – și *sintactică* (sau *structurală*) – bazată pe relațiile structurale între *caracteristici* (*‘features’*). Prima categorie lucrează cu *descriptori cantitativi*, e.g., lungime, arie, textură. A doua categorie lucrează cu *descriptori calitativi (relații)*, e.g., șiruri pentru conectivități simple (cum ar fi în cazul unei scări) între primitive (o scară se poate reprezenta ca un șir (ab)+ unde a semnifică ‘_’ și b semnifică ‘|’). Descriptorii

calitativi pot însemna gramatici cu reguli de producție - o producție nouă poate însemna adunarea / multiplicarea de vectori sau de structuri compuse (rezultate în urma adunării / multiplicării de vectori).

Datele de clasificat sunt de obicei măsurători și / sau observații, ce se pot asocia cu puncte dintr-un spațiu multidimensional definit adecvat. O varietate largă de algoritmi poate fi aplicată pentru recunoașterea de forme, de la *clasificarea prin potrivire* la mult mai complexe *rețele neurale*.

Tema centrală a clasificării e *învățarea delimitării claselor de forme ('pattern classes')*. Clasa e o mulțime de forme cu proprietăți comune, e.g., în cadrul clasei fructelor putem distinge (sub)clasa merelor de (sub)clasa perelor. Înțelegem prin *prototipul unei clase* o formă de referință a unei clase, o reprezentare a clasei. Prototipul se calculează pe baza unor (forme) mostre (forme a căror apartenență la clasă e cunoscută) e.g., poate fi centrul de masă pentru reprezentarea formelor prin puncte în spațiul N-dimensional al caracteristicilor / trăsăturilor formelor.

Clasificarea poate fi:

- *supervizată* - prin *învățare*: se folosește un set de forme mostră sau prototipuri, numit *setul de învățare*; pentru a învăța (antrena) clasificatorul să recunoască clasa formelor pe care le primește ca intrări.

- *nesupervizată sau 'clustering'* - se încearcă identificarea de grupări.

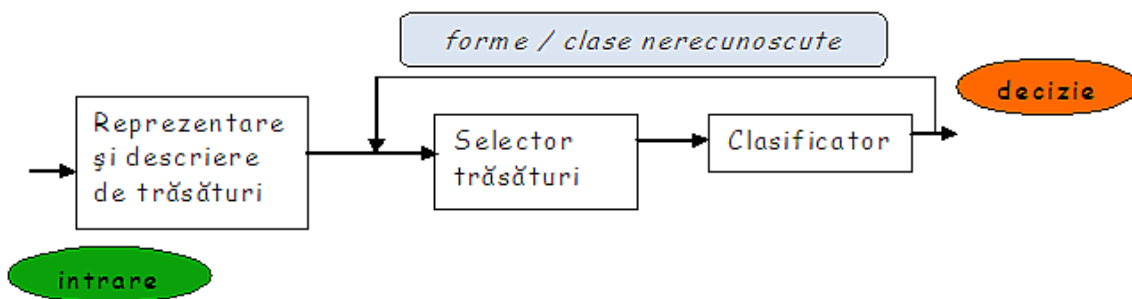


Fig. I.7_1 Schema bloc a unui sistem de recunoaștere a formelor cu învățare. Săgeata înapoi din figura de mai sus (fără de care avem schema generală a unui clasificator, nu neapărat cu învățare a formelor), poate însemna că:

- pentru formele nerecunoscute se mai face o încercare în ipoteza că ele reprezintă forme încă “neînvățate”, se reantrenează clasificatorul, aceste forme fiind rePLICATE la intrarea lui

- se pot înlătura din setul de învățare forme care nu au avut impact semnificativ asupra învățării și adăuga forme de intrare noi.

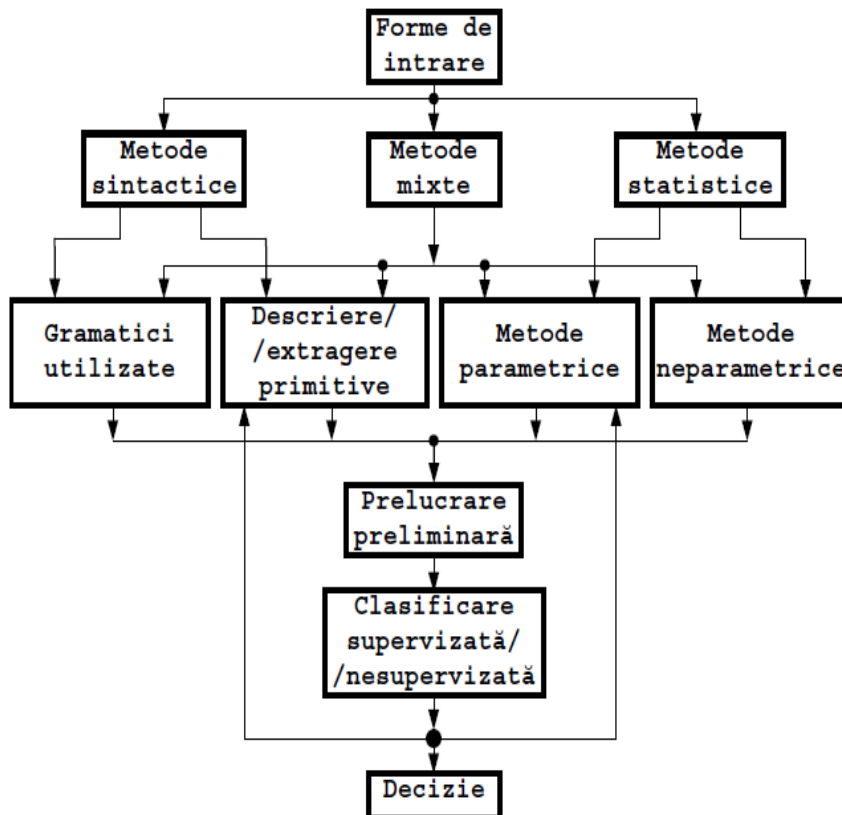


Fig. I.7_2 Moduri posibile de abordare a unei probleme de clasificare conform [Bulea 2003, p. 20]

Proprietăți care permit evaluarea unui clasificator

(Rata de) *Recunoaștere* = cât la sută din setul de antrenare (în vederea recunoașterii) e recunoscut corect.

Convergență = viteza de răspuns; a se face un compromis inteligent între recunoaștere și convergență.

Siguranța = capacitatea de a clasifica corect formele distorsionate aplicate la intrare.

Predicția = capacitatea de a clasifica corect forme care nu aparțin setului de antrenare.

E locul să facem o sumară paranteză despre:

Metode conexioniste în prelucrarea de imagini și recunoașterea de forme

Acestea se bazează pe utilizarea unei rețele de “procesoare” conectate intensiv unele cu altele și care, de regulă, execută aceleași instrucțiuni. Spre deosebire de mașinile von Neuman, se încearcă exploatarea cât mai eficientă a paralelismului de obicei inerent prelucrării imaginilor și recunoașterii formelor.

“Rețeaua de procesoare” se poate concretiza, după ordinea crescătoare a complexității în *structuri sistolice, procesoare matriceale și rețele neurale artificiale*. Orice rețea neurală e caracterizată de: modelul neuronului, arhitectura rețelei și algoritmul de *antrenare* – în cazul clasificării supervizate. Cel mai folosit model de *neuron* e cel fără memorie (nu apare timpul în relație), numit și *perceptron*, caracterizat de relația ieșire(Y)-intrare(X), cu ponderile W și pragurile θ :

$$y_j = f\left(\sum_{i=1}^N w_{ij} x_i - \theta_j\right), \text{ funcția } f \text{ fiind neliniară, e.g., de tip prag, sigmoidă, fiind în general ușor}$$

de implementat hardware. Modelele mai sofisticate de neuroni introduc variabila timp.

Între arhitecturile de rețele neurale, *rețelele cu propagare înainte* (*‘feed-forward’*) caracterizate prin conexiuni unidirecționale, fără bucle de reacție, au fost studiate cel mai mult. Într-o asemenea rețea, neuronii sunt dispuși în straturi successive, ieșirile neuronilor de pe straturile inferioare aplicându-se la intrările neuronilor de pe stratul următor. Binecunoscutul perceptron multistrat are o asemenea structură, gama largă a aplicațiilor în care a fost folosit fiind o dovadă a posibilităților unui asemenea sistem.

După [Bulea 2003, p.24], remarcăm interdependența strânsă între cele două domenii de studiu: “pe de o parte, paralelismul extins oferit de rețelele neurale oferă perspective noi teoriei recunoașterii formelor, iar pe de altă parte, fundamentul matematic solid al teoriei recunoașterii formelor direcționează și dinamizează în permanență cercetări asupra rețelelor neurale”.

I.7.1. Clasificarea supervizată

Clasificarea supervizată poate fi:

- *statistică*: o abordare probabilistică a CI – clasele sunt văzute ca variabile aleatoare N-dimensionale, care se bazează pe modele de distribuție parametrice sau neparametrice, e.g., clasificarea de risc minimizat tip Bayes (vezi secțiunea “Clasificarea statistică” din capitolul 3);
- *fără distribuții*: nu necesită cunoștințe *a priori* despre probabilitățile prototipurilor / claselor - clasificare euristică (nu ne vom referi în acest subcapitol la clasificarea bazată pe gramatici).

I.7.1.1. Clasificarea fără distribuții

Mulțimea de N trăsături cu valori comune clasei poate fi caracterizată de prototipul clasei - poate fi văzut ca un vector (coloană, de dimensiune Nx1) reprezentativ al clasei. Elementele vectorului de trăsături sunt măsurători de a căror alegere inspirată depinde succesul clasificării - e.g., pentru a separa flori pot alege dimensiunile petalelor, pentru a separa discuri cu defecte la margine, pot alege o funcție $r(\theta)$ care e o funcție semnătură (vezi secțiunea “Reprezentarea formelor | Funcții asociate formelor”).

Problema centrală e învățarea recunoașterii clasei formei de intrare, pornind de la un set de prototipuri (forme exemplu sau *mostre*) a căror apartenență la clase se știe.

Fie k clase de obiecte (‘pattern classes’). Se definește funcția discriminant pentru clasa k :

$$d_k(X) > d_i(X), \forall k \neq i \Leftrightarrow X \in C_k, X \text{ fiind vectorul de trăsături al formei de intrare.}$$

Funcția separator între clasele i și j se obține ca $s_{ij}(X) = [d_i(X) - d_j(X) = 0]$.

Pentru o problemă cu K clase trebuie K-1 funcții separator: trebuie divizat spațiul N-dimensional al caracteristicilor în K regiuni diferite. Aceste regiuni sunt hiperplane dacă funcția discriminant / separator e liniară : $d_k(X) = A_k^T X + b_k$, A_k fiind un vector (coloană, de dimensiune $N \times 1$), iar b_k un scalar. În acest caz, vorbim de *clase liniar separabile*.

Clasificatori liniari

Ilustrăm cazul plan pentru două clase C1 și C2 – știm că în acest caz C1 și C2 sunt liniar separabile dacă putem separa printr-o dreaptă formele C1 de formele C2, având nevoie de o singură funcție discriminant $D(X) \begin{cases} \geq 0, X \in C1 \\ < 0, X \in C2 \end{cases}$

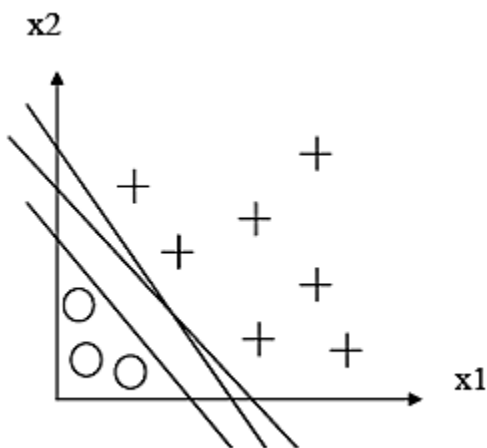


Fig. I.7.1_1 Mai multe variante de a separa liniar între două clase

Observație: Pentru clasificarea de mai sus, conform figurii, se pot utiliza mai mulți separatori ai claselor. Se urmărește să se asigure maximizarea distanței între cele mai apropiate două puncte din clase diferite, adică se dorește să se asigure distanța maximă între clase. În cazul ilustrat, separatorul între cele două clase e mediatoarea segmentului ce unește cele mai apropiate două puncte de clase diferite.

Între algoritmi de clasificare liniară, cei mai cunoscuți sunt:

* Clasificatorul de distanță minimă:

forma X e decisă ca aparținând clasei cu centroidul (prototipul) cel mai aproape de X. Separatorul între două clase e mediatoarea segmentului ce unește prototipurile claselor.

Vom face și o scurtă descriere formalizată a acestui clasificator. Astfel, dacă avem K clase C_k , pentru fiecare clasă, care are N_k forme cunoscute, stabilim prototipul

$$R_k = \frac{1}{N_k} \sum_{X \in C_k} X$$

Decizia se ia conform regulii:

$$X \in C_k \leftrightarrow d(X, R_k) \text{ minim}, \quad k = 1 \dots K$$

Unde $d(X, R_k)$ e o distanță (euclidiană, generalizată (Minkowski), Manhattan, Hamming (pentru forme binare) etc.). Dacă lucrăm cu distanța euclidiană, condiția de minim de mai sus devine condiție de maxim pt.

$$D_k(X) = 2R_k^T X - \|R_k\|^2$$

* Clasificatorul cu cei mai mulți k-vecini:

forma X e decisă ca aparținând clasei cu cei mai mulți vecini între cei k-vecini ai lui X (aici trebuie precizat termenul în funcție de problema specifică e.g., se stabilește o k-rază în jurul careia se numără vecinii)

În cazul *funcțiilor discriminant liniare pe porțiuni*, fiecare clasă are mai mulți vectori de referință (prototip). Distanța de la forma de clasificat la clasă e distanța minimă între distanțele de la formă la prototipuri. Acest clasificator poate fi văzut ca fiind obținut prin divizarea fiecărei clase în subclase, fiecare fiind caracterizată de câte un vector prototip și construind câte un clasificator multicategorial de distanță minimă pentru fiecare clasă.

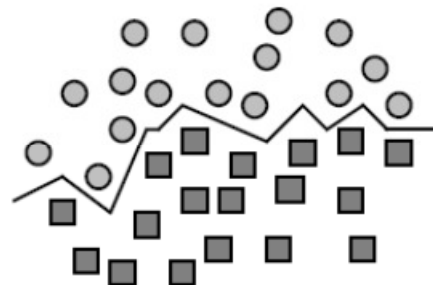


Fig. I.7.1_2 Clasificator liniar pe porțiuni conform [Bulea 2003, p.208]

Generalizând, o funcție discriminant liniară în spațiul (N+1)-dimensional al formelor e dată de:

$$D_i(X) = \sum_{j=1}^{N+1} w_{ij}x_j, \text{ cu } x_{N+1} = 1$$

Prin $D_i(X)$ se definește ecuația unui hiperplan. Matriceal, relația de mai sus se scrie $\mathbf{D} = \mathbf{W} \times \mathbf{X}$.

Limitele de decizie între două clase sunt date de:

$$D_i(X) - D_j(X) = \sum_{m=1}^{N+1} (w_{im} - w_{jm})x_m = 0$$

Dacă notăm $w_m^{ij} \stackrel{\text{def}}{=} w_{im} - w_{jm}$, ecuația limitei de decizie între două clase i și j e de forma $\sum_{m=1}^{N+1} w_m^{ij}x_m = 0$. Această ultimă relație reprezintă ecuația unui hiperplan care trece prin originea spațiului extins al trăsăturilor, spațiu denumit și al ponderilor, a cărui utilizare simplifică mult studiul clasificatorilor liniari.

Cazul a mai multe clase se poate reduce la clasificarea cu două clase, dacă se adoptă una din următoarele strategii de construcție a clasificatorilor multicategoriali, strategii valabile pentru orice tip de clasificator (nu numai pentru cei liniari):

Repetând pentru fiecare clasă, separarea unei clase de restul, obținem zone corespunzătoare claselor. Dar, astfel, obținem și zone de nedeterminare – în zona din centru nu avem nicio clasă, în celelalte trei zone marcate de la margine nu putem decide între două clase:

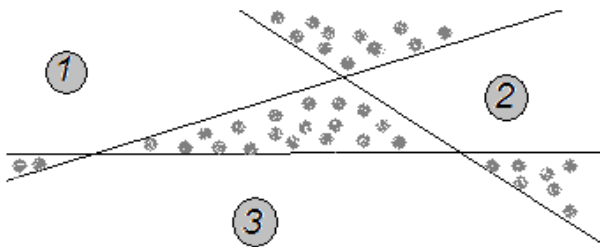


Fig. I.7.1_3 Zone de nedeterminare ivite în urma clasificării liniare cu strategia 1.

Se construiește, corespunzător fiecărei perechi de clase, un clasificator care să împartă spațiul trăsăturilor în două regiuni, astfel încât una să conțină clasele respective, iar cealaltă restul

claselor. Față de strategia 1., zonele corespunzătoare claselor se măresc, micșorându-se zonele de nedeterminare. Cu D_{ij} funcțiile discriminator astfel construite, se ia decizia

$$X \in C_i \leftrightarrow D_{ij}(X) > 0, \quad \forall j \neq i, j = 1 \dots K(\text{numărul de clase})$$

Fiecare funcție discriminant separă doar două clase între ele, caz în care zona de nedeterminare se reduce foarte mult. În acest caz se construiesc funcții de decizie de forma $D_{ij}(X) = D_i(X) - D_j(X)$, regula de decizie fiind ca la strategia a doua. Această strategie e cea mai puternică, reușind să separe liniar clase care nu pot fi separate folosind primele două metode – să remarcăm separarea perfectă în cazul de la mijloc din figura de mai jos (s-au putut proiecta funcțiile discriminant astfel încât dreptele separatoare să fie toate concurente în același punct)

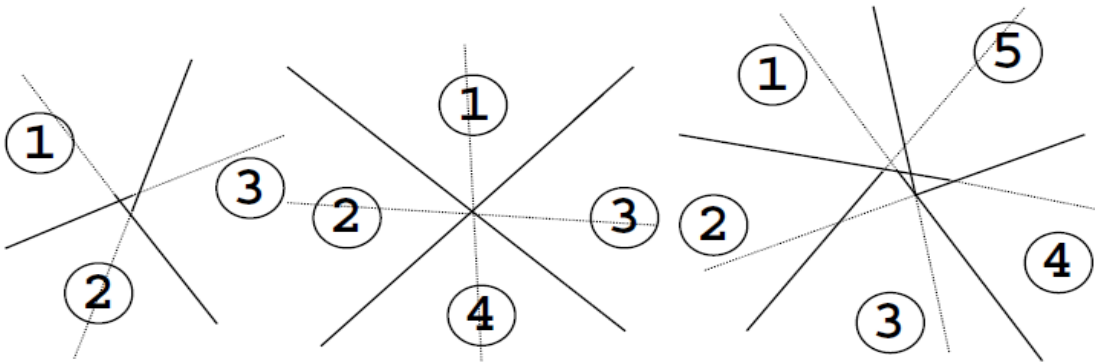


Fig. I.7.1_4 Zone de nedeterminare ivite în urma clasificării liniare cu strategia a treia conform [Bulea 2003, p.202]

Dacă algoritmi liniari, ca în cazul algoritmului perceptronului, sunt cel mai puțin costisitori și mai populari, ei nu sunt “acoperitori” – există multe situații când clasele nu sunt liniar separabile:

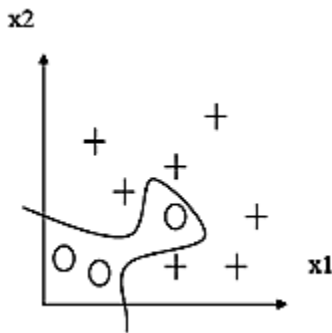


Fig. I.7.1_5 Clasele neseperabile liniar

Rezolvarea problemei de clasificare când spațiul intrărilor nu e liniar separabil are mai multe variante:

- Utilizarea unui clasificator liniar, acceptând unele erori de clasificare - “optimizarea erorilor”
- Utilizarea unui clasificator neliniar – abordările pot fi deterministe sau nu (de exemplu o *rețea backpropagation*).
- Utilizarea unui clasificator liniar, într-un spațiu de dimensiune mai mare în care au fost transferate (transformate) datele de instruire (de intrare)

Desigur, cea de-a treia variantă e de preferat, dacă găsim o transformare adecvată - e.g:

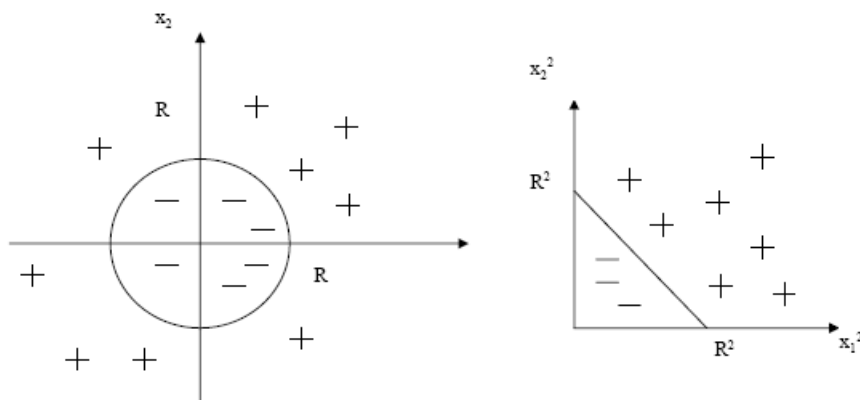


Fig. I.7.1_6 O transformare care dă posibilitatea clasificării liniare

În continuare, prezentăm pe scurt un clasificator neliniar foarte eficient.

* Clasificarea cu arbore de decizie ('decision tree classification')

Acest tip de clasificare are două avantaje importante:

1. converge chiar dacă spațiul trăsăturilor nu e liniar separabil
2. nu trebuie testată fiecare clasă – avantaj mare când avem multe clase

Fie $\mu_k(i)$ și $\sigma_k(i)$ media și deviația standard, calculate pentru o serie de K observații independente, i corespunzând trăsăturii observate. Definim caracteristicile prototipurilor mediate și normalizate ale claselor ca $z_k(i) = \frac{\mu_k(i)}{\sigma_k(i)}$ și matricea corespunzătoare $Z =$

$$\begin{bmatrix} z_1(1) & z_2(1) & \cdot & \cdot & z_k(1) \\ z_1(2) & z_2(2) & \cdot & \cdot & z_k(2) \\ \vdots & \vdots & & & \vdots \\ z_1(N) & z_2(N) & \cdot & \cdot & z_k(N) \end{bmatrix}$$

Algoritm:

- 1) Din Z obținem Z' aranjând crescător pe linie
- 2) Găsim distanța maximă D între elementele unei linii. Fie l linia pe care s-a găsit această distanță.
- 3) Obținem matricea Z'' din Z' , tăind virtual matricea Z' la mijlocul distanței D . "Tăind virtual", pentru că în Z'' linia l rămâne aceeași ca în Z' , dar coloanele se modifică astfel încât în Z'' să am pe coloană numai prototipuri de forma z_k
- 4) Repetă 1) și 2) pentru cele două matrici care se obțin din Z'' prin tăierea, de data aceasta, "nevirtuală", în locul însemnat la 2). Se repetă atâta timp cât matricile rezultate din Z'' au mai mult de o coloană.

Exemplificăm cu următorul Z :

$$\begin{bmatrix} z(1) = \left(\frac{\mu}{\sigma}\right)_{arie} & 6 & 12 & 20 & 16 & 7 \\ z(2) = \left(\frac{\mu}{\sigma}\right)_{perimetru} & 56 & 28 & 42 & 35 & 48 \end{bmatrix} \Rightarrow Z' = \begin{bmatrix} 6 & 7 & 12 & 16 & 20 \\ 8 & 35 & 42 & 48 & 56 \end{bmatrix}$$

Aplicând 2), distanța maximă D se obține pe lina a doua, $D=8$, distanța între 48 și 56. Vom tăia între aceste elemente, mijlocul fiind media lor, $T1 = 52$. Refăcând coloanele astfel încât să reprezinte prototipuri obținem $Z'' =$

$$\begin{bmatrix} 12 & 16 & 20 & 7 & | & 6 \\ 28 & 35 & 42 & 48 & | & 56 \end{bmatrix}$$

Deja am făcut o clasificare după pragul $T1$ al lui $z(2)$, urmând să rafinăm. Întâmplător, nu mai e nevoie să continuăm algoritmul decât cu una din cele două matrici rezultate, cu matricea-coloană (6,56) am terminat. Arborele de decizie va cuprinde ca primă condiție 'if ($z(2) > T1$), urmând ca prin repetarea pașilor 1) și 2) să determinăm celelate condiții suplimentare.

I.7.1.2. Rețele neurale

În general, când procesul de învățare nu e atât de simplu ca în cazul tehnicilor de potrivire sau de clasificare statistică optimă, vorbim de antrenament ('training'), formele din care se învață constituind un 'set de antrenare'. Clasificatorul de tip rețeaua neurală nu impune estimări inițiale ale funcțiilor de decizie – în realitate, de multe ori, acestea pot fi riscante –, funcțiile de decizie fiind obținute în urma unui antrenament laborios, folosind în paralel mai multe *elemente nonliniare de procesare: neuronii*.

Interesul pentru rețele neurale a început în anii '40, odată cu modelarea echipamentelor cu praguri binare și a algoritmilor stochastici (Pitts & Mc Culloch, în 1943). În anii '50 și '60, 'learning machines' introduse de Rosenblatt au "sedus" multă lume din domeniul teoriei și practicii CI. S-a dovedit matematic că aceste mașini, numite și perceptroni, antrenate cu seturi liniar separabile, adică separabile printr-un hiperplan, converg către soluție după un număr finit de pași. Valabilitatea teoremei învățării perceptronilor, a lui Rosenblatt (1962), demonstrează că perceptronii pot învăța orice poate fi reprezentat.

Observație: E important de făcut distincția între reprezentare și învățare. *Reprezentarea* se referă la abilitatea perceptronului (sau a altor rețele) de a simula o funcție specifică. *Învățarea* reclamă

existența unei proceduri sistematice de ajustare a ponderilor din rețea, pentru a produce acea funcție.

A urmat o dezamăgire în urma nepotrivirii perceptronului cu majoritatea problemelor practice de clasificare. În 1984 Simon a publicat o lucrare despre perceptron care se intitula “Nașterea și moartea unui mit”. Dar în 1986, Rumelhart, Hinton și Williams au “resuscitat” perceptronul cu mai multe straturi (‘multilayer’): deși algoritmul de antrenare pentru acest tip de rețea neurală nu converge, ca în cazul perceptronului cu un singur strat, o metodă generală, numită ‘*generalized delta rule for learning by backpropagation*’, a fost folosită cu succes în multe probleme practice.

Neuronul

În cazul neuronului fără memorie, fiecare intrare e multiplicată cu o pondere corespunzătoare, și, apoi, toate rezultatele sunt însumate pentru a determina nivelul de activare al neuronului (a):

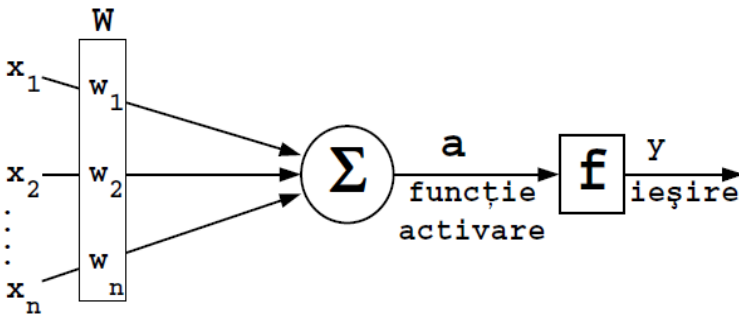


Fig. I.7.1_7 Structura neuronului fără memorie

Funcția $Y = f(a)$ e o funcție liniară care poate fi de tipul limitare hardware, prag logic (folosită inițial de Pitts & Mc Culloch), sigmoidă, simetrică sau nu, sau chiar funcția identică.

Pentru recunoașterea a numai două clase poate fi suficient numai un neuron – perceptronul inițial (cea mai simplă rețea neurală) se reducea la schema din Fig. I.7.1_7 funcția de activare fiind de tip prag. Îmbogățind forma de intrare cu o trăsătură $x_{n+1} = 1$ (‘augmented pattern vector’) și notând cu w_{n+1} pragul corespunzător acestei trăsături, putem scrie vectorial funcția de decizie:

$$D(X) = \sum_{i=1}^n w_i x_i + w_{n+1} = WX$$

Ceea ce reprezintă o clasificare liniară, primele n ponderi stabilind orientarea hiperplanului de separare între cele două clase (pentru $w_j = 0$ acest hiperplan e paralel cu axa O_j), w_{n+1} fiind proporțională cu distanța de la originea O la hiperplan. Semnul produsului scalar WX stabilește clasa: $WX > 0 \leftrightarrow X \in C1$, altfel $X \in C2$.

* Algoritm tipic de *antrenare* (determinare a ponderilor) pentru clasele liniar separabile $C1$ și $C2$:

Presupunem că știm: apartenența formelor de antrenare X la $C1$ sau $C2$

- inițializare (pas 1): $W(1) =$ valori inițiale eventual nearbitrare, $X(1) =$ prima formă din set

- pas k :

Fie c o constantă reală (o vom folosi de mai multe ori în cele ce urmează).

> calculează $D12 = W^T(k) X(k)$

> dacă ($D12 \leq 0$ și $Y(k) \in C1$): $W(k+1) = W(k) + c X(k)$ ($D12$ va crește cu $c \|X(k)\|$)

> dacă ($D12 > 0$ și $Y(k) \in C2$): $W(k+1) = W(k) - c X(k)$ ($D12$ va scăde cu $c \|X(k)\|$)

altfel $W(k+1) = W(k)$

Observații:

- Următorul $X(k)$ va fi următoarea formă de intrare – aceste forme nu se schimbă - algoritmul le ia ciclic (după ultima formă va urma prima) la rând până nu mai trebuie modificat W .

- *constantă c* trebuie să fie pozitivă

* Algoritm tipic de antrenare cu (Least Mean Square/Widrow-Hoff) ‘delta rule’ pentru clase care nu sunt liniar separabile (cazul majoritar întâlnit în practică):

Considerăm funcția criteriu:

$$J(W) = \frac{1}{2} (r - W^T X)^2$$

, r fiind răspunsul dorit $\begin{cases} 1, & X \in C1 \\ -1, & X \in C2 \end{cases}$

$J(W)$, ca diferență între răspunsul dorit și cel actual, trebuie minimizată la fiecare pas de antrenare.

Impunând ca algoritmul să convergă ($J(W)$ să își atingă minimumul și W să nu mai varieze), folosind și ideea de minimizare a variației de la algoritmul anterior, ajungem prin deducții succesive la

$$\boxed{\Delta W = c e(k) X(k)} \quad (\text{'delta rule'}),$$

unde am notat $\Delta W \stackrel{\text{def}}{=} W(k+1) - W(k)$ și eroarea de clasificare $e(k) \stackrel{\text{def}}{=} r(k) - W^T(k)X(k)$.

Acesta e '*delta correction algorithm*', constanta pozitivă c fiind '*magnitudinea corecției*'. Obținem stabilitatea algoritmului pentru $c \in (0; 2)$; în general se ia $c \in (0,1; 1)$.

Dacă clasele sunt separabile, algoritmul 'delta rule' nu garantează un plan separator între clase. Această incertitudine e prețul folosirii unui algoritm care converge atât în cazurile separabile cât și în cazurile neseparabile.

Metodele de clasificare cu rețele neurale sunt relativ limitate, în ciuda faptului că încearcă să imite rețele neurale prezente la om. Practica a dovedit că succesul major al inteligenței artificiale se leagă de 'soft computing', de teoria fuzzy - vezi subcapitolul II.2, aceasta fiind o "imitare" mai realistă a gândirii umane. Chiar folosind funcții de activare neliniare între straturi, vezi principiul algoritmului de antrenare back-propagation rețelele neurale artificiale s-au dovedit departe de a concretiza "speranțele" cu care erau investite la apariția lor.

I.7.2. Clasificarea nesupervizată (clustering)

Această abordare încearcă să identifice clustere, altfel spus, grupări de forme în spațiul trăsăturilor. Prin *cluster* înțelegem un set / nor de puncte în spațiul trăsăturilor în interiorul căruia densitatea de puncte=forme e mare comparativ cu punctele vecine clusterului. Tehnicile de grupare sunt generale, folosindu-se de exemplu și la comprimare sau segmentare.

Gruparea formelor în clase pleacă de la definirea în spațiul N-dimensional al formelor a unei *măsuri de similaritate*. Măsuri de similaritate uzuale sunt produsul scalar, distanța euclidiană (!

în sensul minimizării ei) și corelația normalizată (vezi secțiunea “Potrivirea și detecția formelor|Potrivirea pe baza corelării”).

Inițial, se formează (*a priori*) partiții în spațiul formelor (clustere) suficient de distincte. Apoi, în funcție de similaritatea fiecărei forme cu clusterul se unesc clusterul sau se adaugă forma la cluster. Procedura se repetă până când clusterul nu se mai modifică sau e satisfăcut alt criteriu de convergență.

Metoda lanț ('chain')

Algoritm: prima formă mostră e desemnată ca reprezentând clusterul. Măsura de similaritate a următoarei forme se va calcula față de acest reprezentant. Dacă această distanță e sub un anumit prag, forma se include în cluster, altfel ea va reprezenta al doilea cluster. Procesul continuă până când au fost epuizate toate formele, în încercarea de a le atribui vreunui din clusterul. Avantajul e clasificarea într-un singur pas, dezavantajul e că trebuie știut *a priori* pragul ales pentru a clasifica toate formele.

Metoda iterativă ('c-means')

Algoritm: presupunem că știm numărul K de clase / clusterul / partiții. Partiționarea se face astfel încât împrăștierea medie sau varianța în partiție sunt minimize. Fie $\mu_k(n)$ centru cluster-ului k la a n -a iterație.

Inițial, $\mu_k(0)$ primește valori arbitrare. La a n -a iterație forma X_i e asimilată în clusterul de al cărui centru e cea mai apropiată – în sensul maximizării măsurii de similaritate, e.g., a minimizării distanței euclidiene d :

$$x_i \in C_k \Leftrightarrow d(x_i, \mu_k(n)) = \min_{j=1..K} [d(x_i, \mu_{kj}(n))]$$

Se recalculează centrul clusterului care a primit x_i alegându-se punctul (forma) care minimizează suma distanțelor de la el la celelate puncte/forme din cluster.

Procedura se repetă până când clusterelor rămân neschimbate.

Observație: Dacă știm numărul de clase / cluster, pornim cu o valoare mare pentru K și apoi putem unifica după o măsură de similaritate între cluster.

- Avantaje: simplitate, viteză, se minimizează varianța intracluster
- Dezavantaje: rezultatele depind de partiționarea inițială, NU se garantează că se minimizează varianța globală

I.8. Domeniul probabilistic

I.8.1. Statistici de ordinul 1 / Trăsăturile de histogramă

I.8.1.1. Momente

Histograma e statistica de ordinul 1 a imaginii, cu alte cuvinte densitatea de probabilitate a variabilei aleatoare u reprezentând nivelul de gri (presupunem imagini în scară de gri).

$$p_u(x) \stackrel{\text{def}}{=} \text{probabilitatea}(u = x) = \frac{\text{nrPixeliCuNivelulDeGri} = x}{\text{nrTotalPixeli}}$$

Pe baza lui $p_u(x)$ se definesc *momentele de grad i* :

$m_i = E[u^i] = \sum_{x=0}^{L-1} x^i p_u(x)$, unde L e numărul de nivele de gri. m_1 e media (m), m_2 e valoarea medie patrativă ('Root Mean Square') sau energia medie.

Momentele centrate (în jurul mediei) de grad i:

$$\mu_i = E[(u - m_1)^i] = \sum_{x=0}^{L-1} (x - m_1)^i p_u(x)$$

Evident $\mu_1 = 0$. μ_2 se numește varianța și se mai notează cu σ^2 , σ fiind *deviația standard*.

$\frac{\sigma}{m}$ se numește *coeficient de variație* și poate fi folosit pentru separarea semnalelor cu medii mult diferite între ele, dar și de 0. Coeficientul de variație poate fi văzut ca inversul marimii SNR (Signal to Noise Ratio) definită ca raportul dintre puterea semnalului (informație cu semnificație) și cea a zgomotului (semnal nedorit); SNR în sensul de senzitivitate a unui sistem de formare a imaginii se definește ca raportul dintre semnal și fundal conform formulei $SNR = \frac{\mu_{semnal} - \mu_{fundal}}{RMS_{fundal}}$ pentru un o fotografie relevantă, e.g., a Lunei (semnalul) pe timp de noapte (fundalul).

μ_3 e oblicitatea ('skewness'), μ_4 e o măsură a platitudinii (kurtosis).

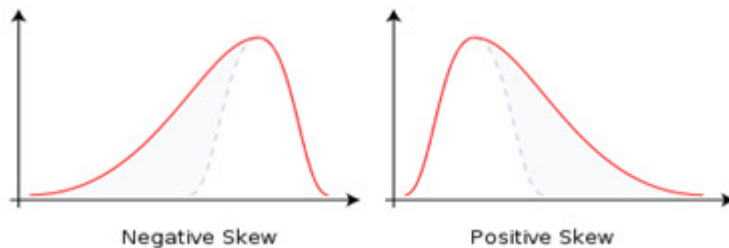


Fig. I.8.1_1 Oblicitatea ca măsură a asimetriei distribuției de probabilitate a unei variabile aleatoare

Observație: Mai sunt și alte măsuri ale oblicității. Karl Pearson a propus măsurarea acesteia pe baza celei mai frecvente realizări / valori a variabilei aleatoare ('mode'), a mediei și deviației standard: $(m - mode)/\sigma$. Mai eficient ca effort de calcul, s-a definit oblicitatea ca $(nrRealizări-mai-mari-ca-m - nrRealizări-mai-mici-ca-m) / nrTotalRealizări$. Aceste simplificări, ca și aplicarea formulei teoretice, pot fi foarte utile pentru probleme concrete.

Momentele absolute se definesc considerând modulul termenilor de însumat ($|x - m_1|$).

Se pot defini și momentele standardizate de ordin k , prin normalizarea față de deviației standard: de forma μ_k/σ^k . acestea sunt scalari (nu au dimensiune, așa cum au momentele prezentate mai sus), μ_k fiind polinoame omogene de ordin k , cu $\mu_k(cx) = c^k\mu_k(x)$. Primele două momente standardizate sunt constante (0 și 1), al 3-lea și al 4-lea au aceeași semnificație ca în cazul momentelor centrate.

Proeminența (negatul platitudinii) e un fel de varianță extremă. Dacă în lucrările mai vechi se folosește momentul standardizat de ordin 4, mai nou se scade 3 din acest moment pentru a asigura o proeminență aproape nulă pentru o distribuție normală de probabilitate.

Alternativ momentelor se pot folosi cumulanți (Kn de ordin n) definiți printr-o funcție generatoare – aceștia au proprietatea interesantă că, pt două variabile aleatoare independente X și Y , $\text{Kn}(X + Y) = \text{Kn}(X) + \text{Kn}(Y)$.

Entropia se definește ca:

$$H = E[-\log_2 p_u] = -\sum_{x=0}^{L-1} p_u(x) \log_2 p_u(x)$$

În cazul 2D, se practică măsurarea acestor trăsături într-o fereastră. Termenul de fereastră are aceeași semnificație cu fereastra folosită în cazul filtrării.

I.8.1.2. Exemplu practic: caracterizarea unei texturi

Uneori se poate caracteriza o textură prin momente – varianța σ^2 ar corespunde unei măsuri a fineții relative a texturii; sau putem folosi $R = 1 - \frac{1}{1+\sigma^2} \in [0,1)$ care se anulează în cazul unei regiuni uniforme și tinde la 1 pentru o varianță foarte mare (putem normaliza și σ^2 prin $1/(I_{max} - I_{min})^2$, I fiind nivelul de gri.

Caracterizarea statistică a texturilor prin momente nu e suficient de precisă – conform [JP SC] s-au găsit “texturi distincte perceptual, dar cu momente identice până la ordinul 5.” Momentele de ordin cel puțin 5 nu sunt legate vădit de forma histogramei, dar pot prezenta informație adițională pentru discriminare între texturi.

Au fost definite două măsuri duale ale texturii bazate pe histograma $h(i)$: *Uniformitatea* $U = \sum h(i)^2 \in [\frac{1}{L}, 1]$

Entropia $E = - \sum h(i) \lg h(i) \in [0, \lg L]$

Conform definiției, statistica de ordinul 1 nu dă informații referitoare la poziția relativă a pixelilor de diferite intensități – pot avea aceeași histogramă pentru imagini net diferite [CV MC1].

I.8.1.3. Covarianță și corelație

Presupunem clare noțiunile de medie și varianță a unei variabile aleatoare.

Varianța unei variabile aleatoare e nulă \leftrightarrow variabila aleatoare e constantă.

Covarianța generalizează varianța, luând în considerare două variabile aleatoare: varianța e cazul special al covarianței când cele două variabile aleatoare coincid.

Expresia matematică a covarianței pentru două variabile aleatoare x și y cu valori (numere) reale, cu varianța finită e:

$$\text{cov}(x,y) = E[(x - Mx)(y - My)] = E(xy) - E(x) E(y),$$

$Mx = E(x)$ și $My = E(y)$ fiind mediile (‘expectation’) celor două variabile aleatoare.

Presupunem cazul multidimensional al variabilei aleatoare $X \in \mathbb{R}^N$, adică X e un vector $N \times 1$:

$$X = (x_1, x_2, \dots, x_N)^T$$

Atunci matricea de covarianță a lui X e dată de $S_{ij} = \text{cov}(x_i, x_j)$

Ilustrăm mai jos trecerea de la 1-D la N-D pentru o distribuție (densitate) normală de probabilitate.

Distribuția normală de probabilitate a unei variabile aleatoare unidimensionale:

$$\varphi(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\left[\frac{(x-m)^2}{2\sigma^2}\right]}$$

unde m și σ sunt media și respectiv dispersia variabilei aleatoare.

Pentru cazul multidimensional $X \in \mathbb{R}^N$, mediei 1-D M_x , definită ca mai sus, îi corespunde vectorul (N x 1) medie $M_X = (M_{x1}, M_{x2}, \dots, M_{xN})^T$, iar dispersiei:

matricea (N x N) de covarianță $S_X = E\{(X - M_X)(X - M_X)^T\}$.

Densitatea de repartiție normală multidimensională e:

$$\Phi(X) = \frac{1}{(2\pi)^{N/2} [\det(S_X)]^{1/2}} \exp \left[-\frac{1}{2} (X - M_X)^T S_X^{-1} (X - M_X) \right]$$

Observație: Dacă luăm E ca operator medie $\rightarrow S_X = \frac{1}{N_X} \sum_X X X^T - M_X M_X^T$,

N_X fiind numărul de realizări ale lui X .

Matricea S_X e reală și simetrică pentru variabilele aleatoare cu care lucrăm noi (evenimentelor/măsurătorilor/realizărilor le sunt asociate numere reale). Conform teoriei algebrei liniare, orice asemenea matrice e diagonalizabilă și (deci) inversabilă.

Distribuția gaussiană N-D a lui X se reduce la produsul distribuțiilor gaussiene 1-D pentru fiecare componentă a lui X ($x_i, i = 1 \dots N$) când matricea de covarianță a lui X e diagonală, adică componentele lui X sunt decorelate (ceea ce se poate face aplicând transformarea Hotelling).

I.8.1.4. Folosirea componentelor principale în cadrul Transformării Hotelling (sau KLT)

Există mai multe descompuneri ortonormale ale unui proces stocastic: dacă procesul e indexat pe $[a,b]$, orice bază ortonormală din $L^2[a,b]$ realizează o asemenea descompunere. Transformarea

Karhunen–Loève (KLT, căreia i se mai spune și Transformarea Hotelling, sau ‘Principal Component Analysis’ (PCA)), furnizează cea mai bună bază în sensul minimizării erorii pătratice medii. Dacă în seriile Fourier coeficienții descompunerii sunt numere reale și funcțiile de bază sunt sinusoidale, în KLT ei sunt variabile aleatoare și depind de proces - funcțiile de bază se determină calculând funcția / matricea de covarianță.

KLT: Un proces stochastic $\{X_t\}_{t \in [a, b]}$ cu media nulă (dacă nu îndeplinește această condiție, considerăm procesul $(X_t - E[X_t])$), satisfăcând unele condiții de continuitate tehnică (ușor de îndeplinit în realitate), admite descompunerea:

$$X_t = \sum_{k=1}^{\infty} Z_k e_k(t) \quad (*)$$

, unde Z_k sunt perechi de variabile aleatoare necorelate și funcțiile e_k sunt funcții continue cu valori reale pe $[a, b]$, ortogonale două câte două pe $L^2[a, b]$. Putem spune că (*) dă o descompunere bi-ortogonală deoarece coeficienții Z_k sunt ortogonali în spațiul probabilităților, adică decorelați, în timp ce funcțiile de bază e_k sunt ortogonale în domeniul temporal / spațial. Mai mult, dacă procesul e gaussian, variabilele aleatoare Z_k sunt gaussiane și independente stochastice.

Se pune problema decorelării componentelor/trăsăturilor unei descrieri. De exemplu, atunci când lucrăm cu pixeli reprezentați prin componentele / canalele (R,G,B), pentru relevanța acestor canale ne interesează ca informația conținută de ele să fie cât mai decorelată.

Presupunem că avem o descriere dată printr-un vector X ($N \times 1$) al trăsăturilor – o variabilă aleatoare multidimensională, e.g., culoarea (R,G,B) a unui pixel. Știm că matricea de covarianță a lui X , S_X , e reală și simetrică, diagonalizabilă la:

$$S_X = Q^T S_Y Q$$

, Q fiind o matrice ($N \times N$) formată din vectorii proprii ortonormali ai lui S_X .

$S_Y = \text{diag}(v_1, v_2, \dots, v_N)$ e o matrice cu toate elementele nule, afară de cele de pe diagonală, care sunt chiar valorile proprii ale lui S_X . S_Y are aceleași valori proprii ca S_X .

Transformarea Hotelling: $Y = Q(X - M_X)$, M_X fiind vectorul medie corespunzător variabilei aleatoare N-D X și Q fiind matricea de mai sus, cu precizarea că vectorii proprii (ortonormali) ai lui S_X sunt ordonați în sensul descrescător al valorii proprii corespunzătoare și puși de la prima linie la ultima linie în matricea Q (pe prima linie se află vectorul propriu corespunzător celei mai mari valori proprii).

Această transformare creează o variabilă aleatoare N-D, Y , cu proprietățile:

- media nulă: $M_Y = 0$
- componentele decorelate: conform teoriei algebrei liniare matricea de covarianță a lui Y e chiar S_Y de mai sus
- reconstrucție: $X = Q^T Y + M_X$

Presupunem că nu folosim toți vectorii proprii ai lui S_X pentru a forma matricea transformării Hotelling: din cei N folosim K . \rightarrow *matricea transformării Hotelling* $\begin{matrix} Q_K \\ K \times N \end{matrix} \rightarrow$

reconstrucția nu va mai fi exactă: $\boxed{\hat{X} = Q_K^T Y + M_X}$

Se poate arăta că eroarea medie pătratică ('mean square') între X și \hat{X} e:

$$e_{ms} = \sum_{i=1}^N v_i - \sum_{i=1}^K v_i = \sum_{i=K+1}^N v_i$$

De aceea am ordonat descrescător valorile proprii v_i .

Prin KLT, imaginile de bază, care sunt vectorii proprii ('eigen') ai matricei de covarianță a procesului stochastic, se numesc *imagini proprii* ('eigenimages'). Dacă concentrarea energiei lucrează bine, e nevoie doar de un număr limitat de imagini proprii pentru a aproxima satisfăcător imaginile de intrare.

KLT se folosește rareori în practică în forma sa originală, deoarece nu e o transformare separabilă, având o complexitate $O(N^4)$. În practică se folosesc aproximări suboptimale pentru KLT, cu implementare eficientă – cea mai folosită e Transform Cosinus Discretă care lucrează

bine pe semnale foarte corelate. Pentru a recunoaște forme complexe, e.g., fețe, trebuie considerate porțiuni mari dintr-o imagine ceea ce înseamnă un efort computațional prea mare. Soluția e să reducem dimensiunea adaptând KLT la un set *specific* de imagini, astfel încât să păstrăm caracteristicile discriminatoare în vederea recunoașterii formelor.



Fig. I.8.1_2 ‘eigenfaces’ (‘eigenimages’) obținute de MIT Media Lab în urma reducerii la o dimensiune $J = 8$, dintr-un set de antrenare cu

500 de vederi frontale conform [eig img]

Metoda cu imagini proprii maximizează dispersia în subspațiul liniar corespunzător întregului set de intrare, indiferent de task-ul de clasificare. Analiza cu Discriminanți Liniari (LDA) și Discriminarea Liniară Fisher (FLD) sunt metode care maximizează dispersia între clase, minimizând dispersia în clase (ceea ce poate fi un avantaj, e.g., în cazul când diferențele datorate iluminărilor variate sunt mai mari decât diferențele între fețe), pe baza unor *vectori proprii generalizați*, adică corespunzători celor mai mari valori proprii.

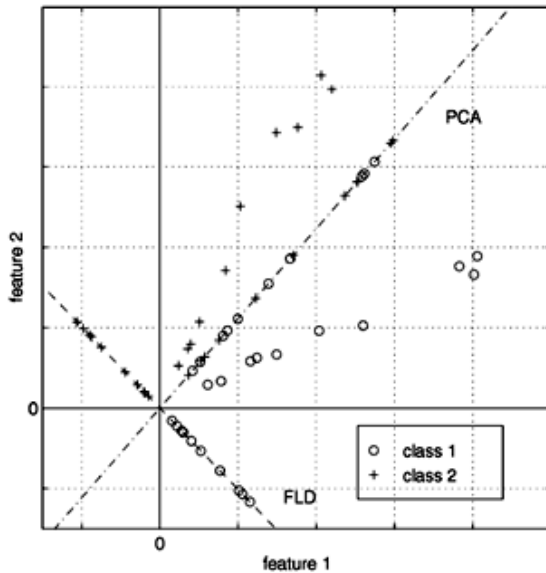


Fig. I.8.1_3 Exemplu 2D conform [eig img]: Elementele a două clase proiectate pe un subspațiu 1D folosind KLT și FLD (LDA). PCA păstrează energia maximă, dar cele două clase se disting mai greu. FLD separă clasele alegând un subspațiu 1D mai bun

I.8.2. Statistici de ordinul 2. Studiu de caz: texturi

Atributul de textură poate fi asociat multor categorii de obiecte: materiale textile, aliaje, roci, vegetație, piele etc. Textura presupune o configurație “mixată” – în afară de cazurile banale, e.g., o tablă de șah, ea nu poate fi defintă ca o combinație de regiuni uniforme, precizând culori sau contururi. Abordarea deterministă care se referă la repetarea spațială a unui motiv de bază, alcătuit din elemente de textură (‘texeli’), o menționăm doar, referindu-ne la abordarea mai generală – cea statistică. De cele mai multe ori, la o scară dată, textura are același aspect, indiferent de subregiunea avută în vedere -> textura poate fi defintă prin uniformitatea spațială a unor invarianțe relative, calculate statistic. Astfel, putem considera textura ca realizare particulară a unui proces aleator staționar. O altă ipoteză pe care o considerăm, e aceea a ergodicității: procesul aleator poate fi caracterizat de un număr finit de stări.

Statistica de ordin cel puțin 2 ia în calcul și poziția relativă a pixelilor. Statistica de ordin 2 are în vedere probabilitatea de apariție a unei perechi de valori ale intensității pixelilor.

I.8.2.1. Matricea de coocurență

Probabilitățile conjugate de ordinul 2 (second order joint probabilities) se definesc astfel:

$$P_{u_1, u_2}(x_1, x_2) = \text{probabilitatea}(u_1 = x_1, u_2 = x_2) = \frac{\text{nrPerechiPixeliCu } u_1 = x_1, u_2 = x_2}{\text{nrTotalPerechiPixeli}}$$

Pe baza acestor probabilități se construiește matricea de coocurență, măsură a probabilității de apariție a unei perechi de valori (intensități). Pentru un vector de translație spațială t , și două nivele de gri a și b , f fiind imaginea și r vectorul de poziție al unui pixel, această matrice e definită în [Vertan & Ciuc, 2007, p. 142]

$$M_t[a, b] = \frac{\text{Card}\{(r, r+t) \in R \times R \mid f(r)=a \wedge f(r+t)=b\}}{\text{Card}\{(r, r+t) \in R \times R\}}$$

Observații:

- matricea de coocurență verifică condiția de normare $\sum_a \sum_b M_t[a, b] = 1$
- în practică nivelul de gri se poate reduce, e.g., de la 16 biți la 8 biți → se modifică și M_t
- având în vedere influența lui t , s-a propus folosirea unei matrici de coocurență care să facă o medie a influențelor [Vertan & Ciuc, 2007, p. 143]:

$$M[a, b] = \frac{1}{4} \{M_{(0,1)}[a, b] + M_{(1,0)}[a, b] + M_{(0,-1)}[a, b] + M_{(-1,0)}[a, b]\}$$

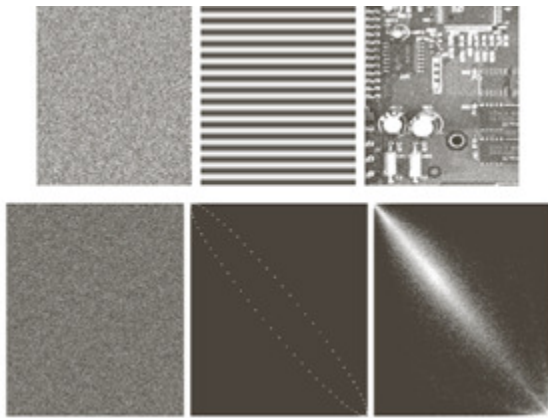


Fig. I.8.2_1 Reprezentarea matricii de coocurență pentru 3 tipuri de imagini [Gonzalez & Woods, 2008, p. 833-834]– se observă că matricea (randul de jos) prinde gradul de “dezordine” din imagini

Asemănător statisticii de ordinul 1, au fost definiți mai mulți parametri statistici pe baza matricii de coocurență:

Uniformitatea (energia) $U_t = \sum_a \sum_b M_t^2[a, b] \in [0,1]$

Entropia $H_t = -\sum_a \sum_b M_t[a, b] \lg M_t[a, b] \in [0, 2 \lg L]$, L fiind numărul nivelelor posibile de gri

Contrastul $C_t = \sum_a \sum_b (a - b)^2 M_t^2[a, b] \in [0, (L - 1)^2]$;

Omogenitatea $O_t = \sum_a \sum_b \frac{1}{1+(i-j)^2} M_t[a, b] \in \left[\frac{1}{1+(L-1)^2}, 1 \right]$, favorizându-se contribuțiile valorilor apropiate de valorile de pe diagonala principală a matricii de coocurență.

Se poate folosi și o măsură a repetării intensității pe direcția t : $D_t = \sum_a M_t[a, a]$

Putem vorbi și de o corelație pe baza matricei de coocurență. Pentru comoditatea notațiilor în loc de $M_t[a, b]$ vom scrie P_{ij} . Mai întâi se definesc mediile:

- pe linii $m_l = \sum_i i \sum_j P_{ij}$

- pe coloane $m_c = \sum_j j \sum_i P_{ij}$

Apoi varianțele:

$$\sigma_l = \sum_i (i - m_l) \sum_j P_{ij}$$

și

$$\sigma_c = \sum_j (j - m_c) \sum_i P_{ij}$$

Corelația, adică covarianța normalizată, corespunzătoare este:

$$\rho = \frac{1}{\sigma_l \sigma_c} \sum_i \sum_j (i - m_l)(j - m_c) P_{ij}$$

I.8.3. Clasificarea statistică

I.8.3.1. Clasificatori statistici optimi

Acești clasificatori estimează astfel încât să avem cele mai mici erori de clasificare

Notăm:

$P(C_k)$ = probabilitatea apriorică a clasei C_k

$P(X)$ = probabilitatea apariției formei X

$P(C_k/X)$ = probabilitatea ca dată fiind forma X să fie în clasa C_k

$P(X/C_k)$ = probabilitatea ca dată fiind clasa C_k , forma X să îi aparțină

L_{ij} = pierderea ('loss') ce afectează procesul de clasificare când o formă din C_i e clasificată ca fiind din C_j ($L_{jj} = 0$)

Pierdere medie ('conditional average risk') pentru $X \in C_j$, X putând fi în orice clasă, e:

$$l_j(X) = \sum_{k=1}^K L_{kj} P(C_k/X)$$

Cu formula lui Bayes ("formula cauzei") : $P(E \cap F) = P(E/F)P(F) = P(F/E)P(E) \rightarrow$

$$l_j(X) = \frac{1}{P(X)} \sum_{k=1}^K L_{kj} P(X/C_k)P(C_k)$$

Decizia de clasificare se ia pe baza riscului minim de eroare, deci conform regulii:

$$X \in C_k \leftrightarrow l_j(X) \text{ minim}, j = 1 \dots K$$

Probabilitatea $P(X)$ e aceeași pentru toate funcțiile de risc condiționat \rightarrow decizia:

$$X \in C_k \leftrightarrow \sum_{k=1}^K L_{kj} P(X/C_k)P(C_k) \text{ minim}, \quad j = 1 \dots K$$

(Acest clasificator se numește Bayes)

Cu o funcție de pierdere L_{ij} binară (simetrică), adică $L_{ij} = 1 - \delta_{ij}$, \rightarrow decizia:

$X \in C_k \leftrightarrow$ funcția discriminant $D_k(X) = P(X/C_k)P(C_k)$ maxim, $j = 1 \dots K$ (FDB)

Altfel scris: $X \in C_k \leftrightarrow \gamma = \frac{P(X/C_k)}{P(X/C_j)} > \frac{P(C_j)}{P(C_k)}$, $\forall j = 1 \dots K, j \neq k$, γ numindu-se raportul de verosimilitate.

Metode parametrice de clasificare statistică

În cazul metodelor parametrice se presupun cunoscute caracteristicile statistice ale formelor de intrare, adică densitatea de probabilitate pentru fiecare clasă și probabilitățile apriorice ale claselor. Cel mai des, sistemele reale sunt descrise de legea normală (gaussiană). Pentru cazul unidimensional avem distribuția (densitatea) de probabilitate gaussiană: $\varphi(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\left[\frac{(x-m)^2}{2\sigma^2}\right]}$

, unde m și σ sunt media și respectiv dispersia variabilei aleatoare cu distribuție normală.

Presupunem că vrem să clasificăm forme 1D în condițiile existenței a doar două clase echiprobabile, adică $P(C_1) = P(C_2) = 1/2$ în interiorul cărora distribuția formelor e normală:

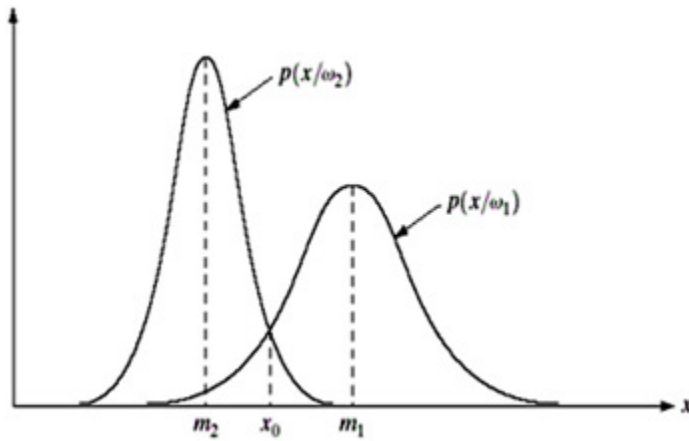


Fig.7. Clasificator Bayes pentru forme 1D, distribuite normal în interiorul a două clase echiprobabile [Gonzalez & Woods, 2008, p. 875]

Separarea în acest caz se poate face prin verticala care trece prin x_0 . Dacă clasele nu sunt echiprobabile – să presupunem că $P(C_1) > P(C_2)$, abscisa separatoare x_0 se va “muta” către clasa mai puțin probabilă. La limită, dacă C_2 e “imposibilă”, $x_0 \rightarrow -\infty$.

Funcția discriminant $D_k(X) = P(X/C_k)P(C_k)$ (conform relației (FDB) de mai sus) se poate logaritma, întrucât aplicarea logaritmului nu schimbă monotonia

$$\boxed{D_k(X) = \lg[P(X/C_k)] + \lg[P(C_k)]} \quad (\text{FDBL})$$

Pentru cazul multidimensional, $X \in \mathbb{R}^N$, mediei unei clase k îi corespunde vectorul medie ($N \times 1$)

1) $M_k = E_k\{X\}$, iar dispersiei matricea de covarianță ($N \times N$) $S_k = E_k\{(X - M_k)(X - M_k)^T\}$.

Densitatea de repartiție normală multidimensională este:

$$\boxed{P\left(\frac{X}{C_k}\right) = \frac{1}{(2\pi)^{\frac{N}{2}} [\det(S_k)]^{\frac{1}{2}}} \exp\left[-\frac{1}{2} (X - M_k)^T S_k^{-1} (X - M_k)\right]}$$

Renunțând la termenii independenți de clasă, cu relația (FDBL) de mai sus →

$$D_k(X) = \lg[P(C_k)] - \frac{1}{2} (X - M_k)^T S_k^{-1} (X - M_k) - \frac{1}{2} \lg[\det(S_k)]$$

Aceasta e funcția de decizie la clasa C_k a formei X , în cazul unui clasificator Bayes cu o funcție de pierdere binară (0-1) și a unei distribuții normale a formelor în interiorul oricărei clase.

$D_k(X)$ e o hiperquadrică – →între fiecare două clase se va plasa o hipersuprafață de separare (ca diferență între funcțiile de decizie corespunzătoare claselor).

În practică, aplicarea formulei de mai sus se face în condițiile unor presupuneri simplificatoare.

Dacă matricea de covarianță e aceeași pentru fiecare clasă, adică $S_k = S$, eliminând termenii independenți de clasa (k): $D_k(X) = \lg[P(C_k)] - \frac{1}{2} M_k^T S^{-1} M_k + X^T S^{-1} M_k$,

ceea ce înseamnă că separatorul va fi un hiperplan.

Dacă, în plus, clasele sunt echiprobabile și $S = I_N$: $D_k(X) = X^T M_k - \frac{1}{2} M_k^T M_k$

am “picat” peste decizia pentru clasificatorul de distanță minimă. Să mai observăm că în acest din urmă caz clasele sunt ca niște nori sferici de aceeași dimensiune, separarea lor, doi câte doi, făcându-se prin hiperplane mediatore segmentelor ce unesc centrelor norilor.

Una dintre cele mai de succes aplicații practice ale clasificatorului Bayes e percepția la distanță (‘remote sensing’) prin captarea de imagini multispectrale din nave/stații/sateți spațiali. Imaginile pot fi “fotografii” ale unor terenuri, păduri, zone de trafic, calității apei, evoluției unor maladii etc. Cantitatea imensă de date astfel culese recomandă analiza/clasificarea automată.

Să nu uităm însă că gradul de optimalitate al clasificării e proporțional cu adecvarea la realitate a probabilităților presupuse: probabilitatea fiecărei clase și a formelor în fiecare clasă. **Domeniul probabilistic, deși poate “veni” cu informații în plus față de cel spațial, nu încurajează folosirea intuițiilor, a adaptărilor pentru o rezolvare cât mai adecvată a unor probleme mai complexe de clasificare.**

IIȘ Separabilitatea claselor

II.1. Descompunere ortogonală cu Unde și Undine

Putem vedea descompunerea ortogonală a semnalelor în Domeniul Frecvență ca pe o clasificare conform asemănării / compatibilității semnalului cu fiecare din undele în care descompunem semnalul – cu cât coeficientul din descompunerea Fourier este mai mare, cu atât semnalul este mai compatibil cu “clasa” undei corespunzătoare. Aceasta e prima din observațiile, credem interesante, pe care le facem în acest subcapitol. Descompunerea ortogonală în unde poate fi văzută, abstractizând, ca o partiționare / clasificare fuzzy a semnalului printr-o metodă de ‘clustering’. A doua observație a acestui subcapitol e legătura intrinsecă dintre **corelație** și ortogonalitate: acestea sunt antonime perfecte – vezi [Georgescu 2, 2012]. Dacă vorbim de clasificare, vorbim de (gradul de) corelație și, deci, (de gradul) de descompunere ortogonală. A treia observație se referă la semnificația principiului incertitudinii (al lui Heisenberg) privind invers proporționalitatea **rezoluției** din domeniul Timp de cea din domeniul Frecvență; vom menționa teorema eșantionării și ‘aliasing’-ul ca efect nedorit ce afectează o partiționare în unde a semnalului astfel încât să îl putem reconstrui cu acuratețe.

II.1.1. Cazul tipic: Transformarea Fourier (TF) sau descompunerea în frecvență (în unde)

Mai întâi vom preciza că discuția noastră are loc în cadrul *sistemelor liniare invariante în timp*. Rezultatul fundamental al teoriei sistemelor LTI spune că orice astfel de sistem poate fi caracterizat complet prin *răspunsul la impuls*: ieșirea sistemului e dată de convoluția intrării cu răspunsul la impuls. Echivalent, un sistem LTI poate fi caracterizat în domeniul frecvență prin *funcția sa de transfer*, care e transformata Laplace a răspunsului la impuls (sau transformata Y în cazul sistemelor discrete). Prin **teorema convoluției** se afirmă că multiplicarea prin produs scalar cu funcția de transfer în domeniul frecvență e echivalentă convoluției în domeniul timp răspunsul la impuls.

Sistemele LTI sunt potrivite descrierii multor sisteme importante – orice sistem care poate fi modelat de o ecuație diferențială liniară omogenă. În +domeniul prelucrării semnalelor, operațiile de filtrare pot fi văzute ca sisteme LTI, masca filtrării fiind răspunsul la impuls. Prin exponențiale complexe putem reprezenta spectrul de frecvență al semnalului

Pt sistemele LTI, funcțiile proprii, și funcțiile de bază ale transformării sunt exponențiale complexe. TF poate fi văzută ca un caz special al transformării Laplace - când exponențiala are un argument pur imaginar -, obținându-se valorile proprii pentru unde (sinusoide complexe) pure. TF pentru un semnal real $f(x)$ cu energie finită (L_2):

$$F(u) = \int_{-\infty}^{\infty} f(x)e^{-2\pi i x u} dx \quad (1)$$

Comparativ cu transformarea Laplace, TF are avantajul major că e inversabilă, în unele condiții (nu foarte greu de îndeplinit), f poate fi reconstruit din F prin TF inversă:

$$f(x) = \int_{-\infty}^{\infty} F(u)e^{2\pi i x u} dx \quad (2)$$

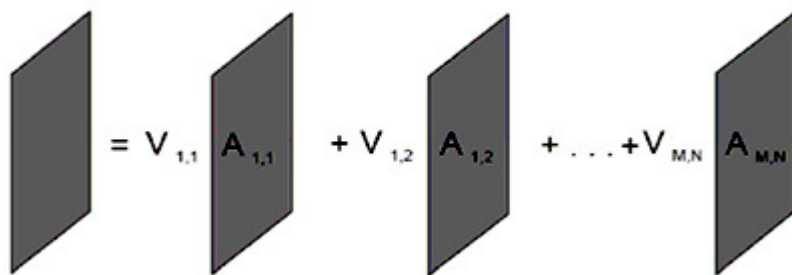
$F(u)$ sunt coeficienții Fourier, adică ai reprezentării semnalului în domeniul Fourier (frecvență), FT mai are marele avantaj că e o transformare integrală cu nucleu separabil și simetric: operează pe întreaga imagine și, pentru o intrare 2D, putem reduce timpul de calcul aplicând numai transformări 1D: întâi pe linii și apoi pe coloane (sau invers). Cu un astfel de nucleu convoluția e identică corelației.

Putem vedea TF ca o descompunere ortogonală unică în funcții simetrice. Aceste funcții sunt explicite (sinusoide) și sunt asociate frecvent conceptelor fizice de frecvență și energie.

II.1.2. Transformări de imagini. Cazul special al transformărilor unitare

II.1.2.1. Transformări de imagini

Putem spune că a aplica o transformare unui obiect e echivalent cu a reprezenta acel obiect în spațiul transformatei. De exemplu, într-un spațiu vectorial, scrierea unui vector sub forma unei combinații liniare a vectorilor unei baze a spațiului, o putem vedea ca pe o trecere într-un spațiu al transformatei în care reprezentarea e data de coeficienții vectorilor bazei. În cazul 2D, această descompunere se poate ilustra ca în figura de mai jos:



II.1.2._1 Descompunerea unei imagini în $M \times N$ imagini de bază

Exemplificăm numeric descompunerea de mai sus:

$$\begin{bmatrix} 6 & 0 \\ 4 & 6 \end{bmatrix} = 4 \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} + 1 \begin{bmatrix} 1 & -1 \\ 1 & -1 \end{bmatrix} - 1 \begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix} + 2 \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

V_{11} V_{12} V_{21} V_{22}

II.1.2._2 Principiul general al descompunerii unui obiect sau de trecere din reprezentarea sa (cuantificată) într-un spațiu (A_{ij}) în alt spațiu – al coeficienților descompunerii / transformării (V_{ij}) Relația transformării se poate scrie în forma matricială $Img = V \times A$.

II.1.2.2. Transformări unitare

Def) A *matrice unitară* : A pătratică NxN, cu elemente complexe cu

$$A^{-1} = A^{*T}, \text{ * fiind operatorul de conjugare complexă}$$

Vom lucra cu matrice cu elemente reale, fără a restrânge prin aceasta generalitatea rezultatelor.

Observație: Matricea se numește *ortogonală* dacă în relația de mai sus nu apare conjugarea complexă -> Matricele unitare cu elemente reale sunt ortogonale (cazul nostru).

Pe baza matricelor unitare/ortogonale se definesc transformatele unitare/ortogonale ale vectorilor Nx1, care nu sunt altceva decat înmulțirea la stânga cu aceste matrice. Rezumăm în continuare, cazurile 1D și 2D din [Vertan & Ciuc, 2007, p. 71-76]

Cazul 1D

Semnalul u(n) formează vectorul coloană $u = [u(0) \ u(1) \ \dots \ u(N-1)]^T$

$A = a(k,n)$, $k,n = 0.. N-1$ unitară

Transformata lui u e: $v = A u$ (3)

Transformarea inversă $u = A^{*T} v$ se poate scrie: $u(n) = \sum_{k=0}^{N-1} a^*(k, n)v(k)$ (4)

Semnificația unei transformate unitare poate fi înțeleasă punând în evidență coloanele matricei

$$A^{*T} = [a_0 \ a_1 \ \dots \ a_{N-1}]$$

$$\Rightarrow (1) : v(k) = \langle a_k, u \rangle, k = 0.. N-1 \quad (3')$$

, unde cu $\langle \cdot, \cdot \rangle$ e notat produsul scalar, iar (3) devine $u = \sum_{k=0}^{N-1} v(k)a_k$ (4')

, deci se descompune u în baza formată de a_k , coeficienții dezvoltării lui u în această bază fiind chiar componentele $v(k)$ ale vectorului transformat; acești coeficienți se obțin ca produse scalare $\langle a_k, u \rangle$. Noua bază e ortonormală, deoarece $A^{*T} A = I^N$.

Cazul 2D

Analog cazului 1D, transformata unitară a unei imagini reprezintă scrierea acestei imagini într-o altă bază, ca o combinație liniară de imagini de bază.

Din cazul 1D relația (3) devine în 2D:

$$V(k,l) = \langle A_{kl}, U \rangle = \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} A_{kl}^*(m,n)U(m,n), k,l = 0.. N-1 \quad (3'')$$

A_{kl} sunt N^2 matrici $N \times N$, formând o bază ortonormală.

$$(4) \text{ devine } U = \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} V(k,l) A_{kl} \quad (4'')$$

Scris pe elemente:

$$U(m,n) = \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} v(k,l) A_{kl}(m,n), \quad m,n = 0.. N-1$$

Transformări unitare bidimensionale separabile

Sunt date de: Δ) cele N^4 elemente care definesc matricele $A_{kl}(m,n)$ ce pot fi calculate pornind de la o singură matrice A $N \times N$:

$$A_{kl}(m,n) = A(k,m) A(l,n), \quad k,l,m,n = 0.. N-1$$

Se poate arăta ca pentru ca ortonormalitatea bazei formate de matricele A_{kl} să fie respectată și în cazul separabil, e necesar și suficient ca matricea A sa fie unitară.

Relațiile transformării 2D separabile se pot scrie:

$$V(k,l) = \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} A(k,m) U(m,n) A(l,n), \quad k,l = 0.. N-1 \quad \text{și}$$

$$U(m,n) = \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} A(k,m)^* V(k,l) A(l,n)^*, \quad m,n = 0.. N-1$$

Scris matriceal:

$$V = A U A^T \quad \text{și}$$

$$U = A^{*T} V A^*$$

Proprietăți ale transformărilor unitare

Mai întâi precizăm că **o transformare unitară conservă energia semnalului:**

$$E(v) = \|v\|^2 = v^{*T} v = (Au)^{*T} Au = u^{*T} A^{*T} A u = u^{*T} u = \|u\|^2 = E(u)$$

Energia, deși se conservă, poate fi “distribuită inegal între coeficienții $v(k)$ ai spațiului transformat. Ea poate fi concentrată într-un număr redus de coeficienți ai transformării, restul putând fi neglijați. Se poate arăta că, odată cu compactarea energiei, se reduce gradul de corelație între coeficienții din spațiul transformatei.” [Vertan & Ciuc, 2007, p. 77]

Transformata unitară care compactează optim (la maxim) energia și care, în același timp, decorează complet coeficienții, e transformata Karhunen-Loeve (pe care am tratat-o în cadrul subcapitolului “Domeniul probabilităților”, la subsecțiunea I.8.1.4, “Folosirea componentelor principale în cadrul Transformării Hotelling(sau KLT)”, Hotelling fiind un alt nume pentru transformarea Karhunen-Loeve).

II.1.3. Eșantionarea semnalului: precizia necesară și intercondiționarea reprezentărilor în Domeniul Frecvență și în Domeniul Frecvență-Timp

II.1.3.1. Precizia necesară reconstruirii semnalului

Semnalul dreptunghiular (‘box’) și fenomenul Gibbs

Definim un semnal box ca $\text{rec}_{\mathbb{R}}(A, D) = \begin{cases} A, & x \in [-\frac{D}{2}, \frac{D}{2}] \\ 0, & \text{în } \mathbb{R} \end{cases}$, A și D fiind constante reale pozitive.

Calculând Transformatei Fourier obținem:

$$TF(\text{rec}_{\mathbb{R}}(A, D)) = A D \text{sinc}(D v)$$

Deci TF a unui semnal dreptunghiular e semnalul sinus cardinal. Ilustrăm corespondența pentru $A=D=1$:



Fig.II.1.3_1 Semnalul dreptunghiular și TF; echidistanța între abscisele de anulare a TF – e invers proporțională cu suportul semnalului dreptunghiular.

Fenomenul Gibbs a primit această denumire după fizicianul american Gibbs care l-a enunțat “pentru prima dată” - la jumătate de secol după “obscurul” [wikipedia] matematician englez Henry Wilbraham. Enunțul: dacă resintetizăm prin serie F o funcție periodică cu salturi de discontinuitate, sumele F parțiale au mari oscilații în jurul punctelor de discontinuitate. Aceste oscilații nu dispar dacă adăugăm sinusoidă cu frecvențe de ordin superior, dar sunt totuși mărginite de o limită finită (știută). Când atât funcția f , cât și TF sunt integrabile în modul (L^1), are loc reconstrucția semnalului, pentru *aproape orice* punct; dacă f e un semnal continuu, egalitatea ține pentru orice punct.. Avem o exemplificare în figura de mai sus: funcția sinc nu e L^1 , de unde apariția fenomenului Gibbs. În procesarea de semnal fenomenul Gibbs se traduce prin *artefacte de ecou* (*ringing artifacts*). Ilustrăm:

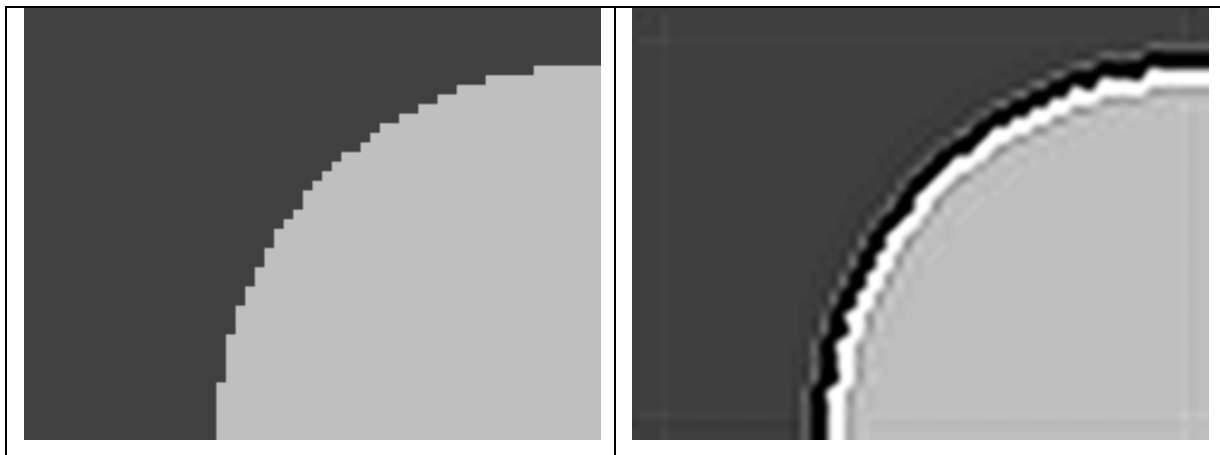


Fig.II.1.3_2 Reconstrucția imaginii prin TFI: observăm “unde” în zona tranzițiilor bruște de intensitate

Cauza principală a acestor artefacte e *limitarea în bandă*, adică *tăierea frecvențelor înalte*, ceea ce se face prin aplicarea unui *filtru trece jos* - o funcție box: aceasta e descrierea în domeniul frecvență. În domeniul inițial, ondulara se vede în graficul funcției sinc conform figurii

Fig.II.1.3_1. Ce putem face pentru a avea o filtrare cât mai precisă, dar fără ‘ringing’? E momentul să trecem în revistă TF unei funcții gauss-iene (cu distribuție normală):

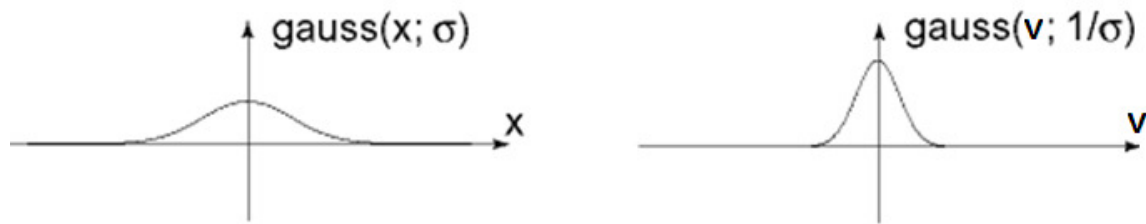


Fig.II.1.3_3 Semnalul gaussian și TF; observăm o altă reprezentare a principiului incertitudinii (vezi subsecțiunea următoare)

Față de transformările anterioare, aceasta e cea mai puțin spectaculoasă: practic se modifică numai varianța. Pentru ‘No ringing’, filtrul gaussian e ideal. Dar în ce privește precizia filtrului trece jos? Nu putem avea la cotele cele mai înalte și-și. Aici ar fi gradul de libertate al unui compromis inteligent: alegem între filtrul ideal și cel gaussian în funcție de datele concrete ale problemei.

Teorema eșantionării

Problema eșantionării a fost rezolvată de H. Nyquist și C. Shannon prin așa-numita teoremă a eșantionării, teoremă fundamentală pentru teoria informației, în particular pentru telecomunicații și procesarea de semnal. Într-o formă cât mai simplă, teorema eșantionării se enunță:

Dacă un semnal temporal nu conține frecvențe mari mari decât B hertzi, el e complet determinat dacă se dau eșantioanele sale la o perioadă a eșantionării de cel mult $1/(2B)$ secunde.

II.1.3.2. Interconținerea rezoluției reprezentărilor în Frecvență și în Frecvență-Timp: principiul nedeterminării

TF discretă

Scriind TF pentru un semnal $f(t)$ eșantionat la o perioadă T , avem

$$F_s(\nu) = \sum_{n=-\infty}^{\infty} f_n e^{-2\pi i \nu n T}$$

Această funcție e continuă și de perioadă $1/T$ - ne trebuie, deci, doar pe perioada $1/T$ pentru a caracteriza funcția. Discretizăm funcția, luând M eșantioane, atât pentru semnalul original (f), cât și pentru cel eșantionat și apoi transformat: $\nu = \frac{m}{MT}$. Se obține astfel o relație care nu depinde de T . Iată perechea TF directă și TF inversă, transformări care furnizează m numere finite pornind de la oricare alte m numere finite:

$$F_m = \sum_{n=0}^{M-1} f_n e^{-\frac{2\pi i n m}{M}} \quad \text{și} \quad f_n = \frac{1}{M} \sum_{m=0}^{M-1} F_m e^{\frac{2\pi i n m}{M}}$$

Observații:

- luând intervalul $[0, 1/T)$, conform Fig.II.1.3_6, F_m nu sunt ordonate de la cea mai mică la cea mai mare frecvență: perioada $1/T$ acoperă două semiperioade “spate în spate.” De acest lucru se va ține cont la reprezentarea grafică a TF.

- factorul $1/M$ din afara integralei se poate distribui la cele două relații. Putem trece acest factor la expresia lui F_m , eliminându-l de la expresia lui f_n ; contează ca **produsul** factorilor din afara celor două integrale să fie $1/M$.

- F și f sunt de perioadă M

! rezoluția în frecvență e invers proporțională cu durata în timp a semnalului eșantionat și reciproc: aceasta e baza principiului nedeterminării – vezi și [Georgescu 1, 2012].

Din punct de vedere matematic principiul nedeterminării spune că, pentru $f(x)$ absolut continuă și $x \cdot f(x)$ și $f'(x)$ din L^2 (funcții cu pătratul integrabil), produsul între dispersia (normalizată) în jurul lui zero în domeniul timp și dispersia (normalizată) în jurul lui zero în domeniul frecvență e limitată inferior de o constantă:

$$f(t) \leftrightarrow F(\omega) \Rightarrow \sigma_t \sigma_\omega \geq const, \quad \text{cu:}$$

$$\sigma_t^2 = \frac{\int t^2 |f(t)|^2 dt}{\int |f(t)|^2 dt} \quad \text{și} \quad \sigma_\omega^2 = \frac{\int \omega^2 |F(\omega)|^2 d\omega}{\int |F(\omega)|^2 d\omega}$$

Egalitate are loc numai pentru $f(t)$ gaussiană. Conform [Stein & R. Shakarchi, 2003], în mecanica cuantică, momentul și poziția funcțiilor undă sunt ca în perechea TF: inegalitatea de mai sus fiind principiul nedeterminării al lui Heisenberg.

Pentru a ilustra principiului nedeterminării vom adapta în continuare un experiment găsit pe wikipedia la descrierea TF localizată ('Short Time Fourier Transform'). Amintim că această variantă a TF localizează semnalul cu o funcție de limitare înainte de a calcula transformata [Pesquet-Popescu and Popescu, 2010]. Un dezavantaj al STFT e rezoluția fixă data de funcția fereastră. O fereastră mai mare furnizează o mai bună rezoluție în frecvență, dar o mai slabă rezoluție în timp. Să vedem cum arată spectrogramele unui semnal constând în două sinusoide – prima parte a semnalului e o undă de frecvență 10 Hz, a doua parte a semnalului e o undă de frecvență 25 Hz – pentru diferite lățimi ale funcției fereastră.

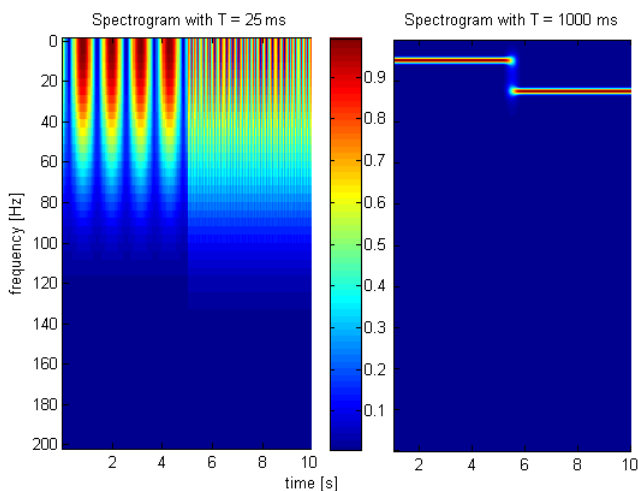


Fig.II.1.3_4 Spectrograme pentru ferestre diferite ale Transformatei Fourier localizate

Compromisul inteligent între acuratețea în domeniul temporal (sau spațial) și acuratețea în domeniul frecvență e Transformarea Undină (Wavelet) : TW. Sau, cu alte cuvinte, analiza multirezoluție. Principiul nedeterminării a fost ilustrat grafic în [Gonzalez & Woods, 2008, p.

501], atribuind fiecărei funcții din baza de reprezentare o *cutie Heisenberg* ('*Heisenberg box*') în planul timp-frecvență:

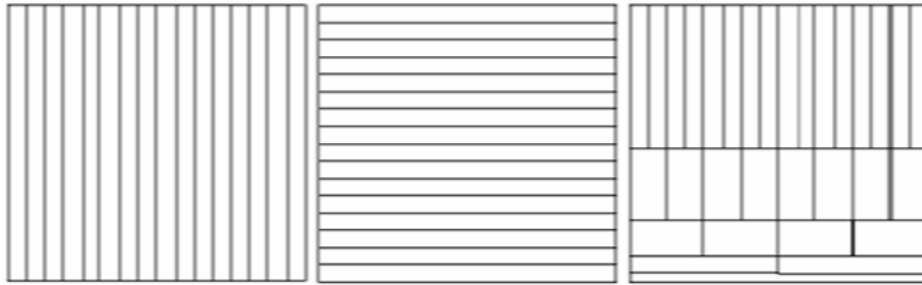


Fig.II.1.3_11 “Acoperirea” prin 3 tipuri de funcții de bază: impulsuri (domeniu temporal), sinusoidale (domeniu frecvență) și TW (aproximare / scalare + detalii în domeniu frecvență-timp)

Aria fiecărei cutii Heisenberg e aceeași. TW e o bună alegere în cele mai multe situații practice: semnale nestaționare, adică cu frecvențe variind în timp. În plus, e o reprezentare mai flexibilă, putând “jongla” în interiorul principiului nedeterminării. Putem reprezenta conceptual figura de mai sus

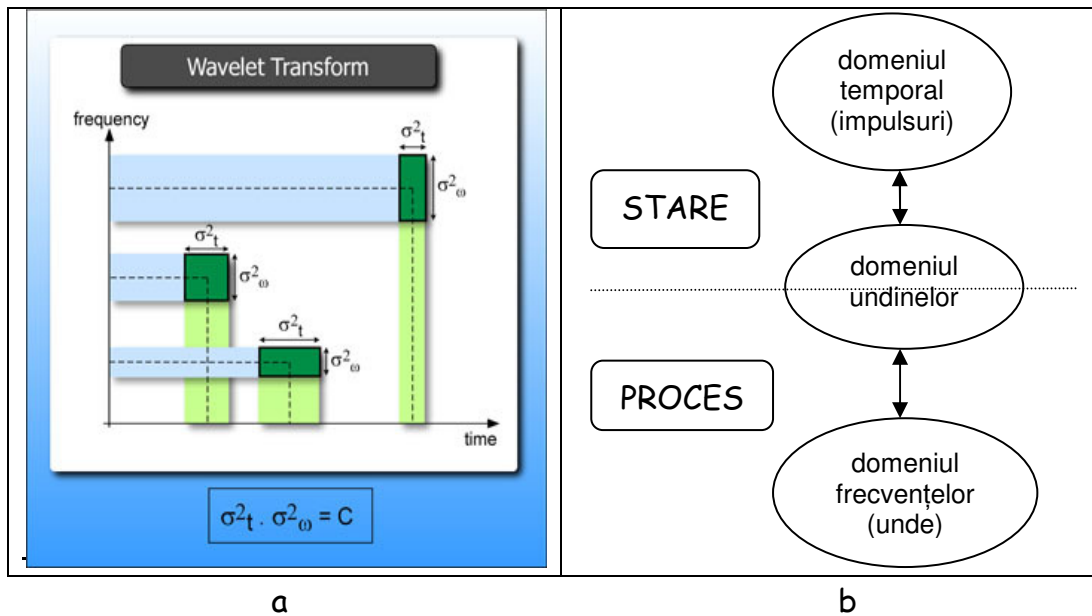


Fig.II.1.3_12 Posibilitatea de a “jongla” în interiorul principiului nedeterminării (am folosit termenii din engleză) [wv hsn] și o reprezentare conceptuală originală a relațiilor dintre domeniile: temporal, al undinelor și al undelor.

II.1.4. Concluzii și aplicații

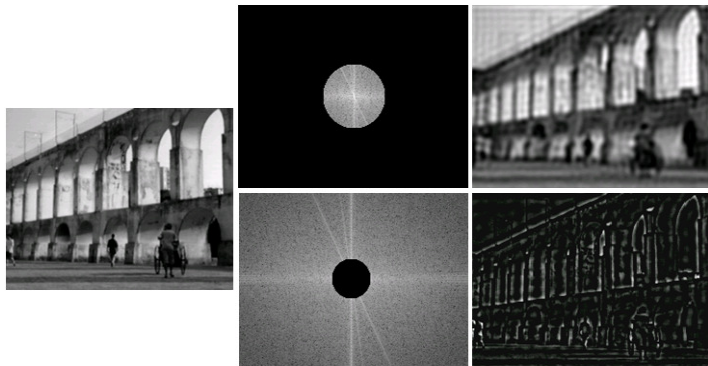
II.1.4.1. Filtrarea în domeniul frecvență

Informație relevantă din domeniul frecvență

$F(u,v)$ conține toate valorile imaginii $f(x,y)$ ponderate de termeni exponențiali ce țin cont de toate pozițiile pixelilor imaginii. Astfel, cu excepția unor cazuri banale, nu putem deduce din $F(u,v)$ forme localizate în interiorul imaginii. Deși faza TF ne dă informații despre aceste forme, în cele mai multe cazuri, aceeași informație se deduce “mai complet” și mai ușor dacă apelăm direct la imagine. Dacă, în general, nu putem face asociații directe între spectrul TF și componente ale imaginii, putem însă obține informații despre frecvența caracteristicilor spațiale ale imaginii.

Se știe că centrul spectrului TF (STF), adică $F(0,0)$ e proporțional cu intensitatea medie a imaginii (intensitate = nivel de gri). Conform **distribuției radiale a STF frecvențele joase sunt reprezentate mai aproape de origine**, iar cele înalte mai departe. Să ne imaginăm o textură. Dacă textura e “grosolană” STF se va concentra în jurul originii; pentru o textură “fină”, STF se va “împrăștia.” O aplicație imediată a acestei observații e proiectarea filtrelor trece jos, respectiv trece sus în domeniul frecvență:

Fig.II.1.4_1 Imaginea inițială e, în ambele cazuri, în prima coloană. Coloana a treia conține imaginile trecute înapoi în domeniul spațial, după filtrările trece jos, respectiv trece sus în domeniul frecvență, filtrări reprezentate în coloana din mijloc. Remarcăm complementaritatea celor două filtre.



Distribuția unghiulară a STF e sensibilă la direcția texturii imaginii - pentru o textură orientată în unghiul θ față de abscisă, STF va fi concentrat într-o direcție orientată în unghiul θ față de ordonată:

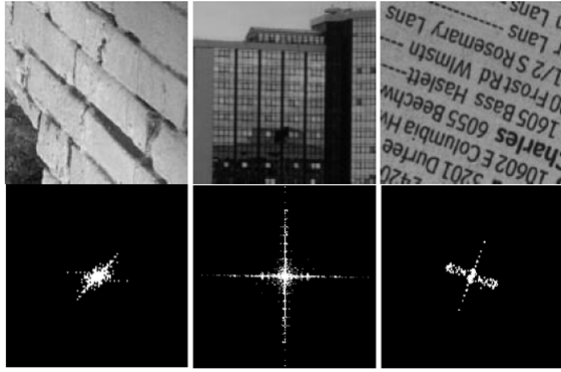


Fig.II.1.4_2 Concentrarea direcțională a STF funcție de direcțiile dominante în imagine. Observăm dependența “ $+\pi/2$.” Mai observăm și intermitența punctelor din ultimul STF, pe direcția corespunzătoare scrisului $+\pi/2$. Explicația e imediată, putând fi și vizualizată în STF: dacă era de așteptat continuitatea axelor în STF corespunzător imaginii cu clădirea, la fel de așteptat era discontinuitatea axelor în STF

corespunzător imaginii cu scrisul – se vede aici efectul spațiilor dintre litere.

Prin caracterul ei general, transformarea TF e un instrument puternic de tratare a multor probleme, e.g. :

- comprimarea și scalarea rapidă – vezi abordarea multirezoluție
- înregistrarea / alinierea imaginilor (‘registration image’) pe baza corelării fazei (tehnica ‘phase correlation’) în domeniul frecvență – această tehnică o folosim și noi la IV.3.
- restaurarea imaginilor și eliminarea zgomotelor (prin eliminarea coeficienților de detaliu cei mai mari)
- reconstrucția în cazul imaginilor secțiune rezultate prin tomografie computerizată

Se știe, ca principiu general, că domeniul frecvență oferă “viziuni mai intuitive” în ce privește conceperea filtrelor, a funcțiilor de scalare și undinelor din analiza multirezoluție. Implementarea se face însă în domeniul spațial, folosindu-se pentru optimizarea vitezei măști mici. Concepția și implementarea, aparent opuse, ca în principiul incertitudinii al lui Heisenberg, ambele, au aportul lor, fiind o dualitate de luat în seamă în abordarea problemelor practice.

II.1.4.2. Aplicații ale undinelor

TW se aplică în domeniile de aplicare a transformării Fourier, față de care am dovedit că transformarea Wavelet e chiar mai versatilă prin furnizarea, pe lângă informațiile de frecvență, și a celor de timp. În general, pentru implementarea de filtre trece jos se pleacă de la funcțiile de scalare DWT iar pentru filtrele trece-sus se pleacă de la undine.

În multe cazuri mai eficientă, ca acuratețe și comprimare, decât celelalte transformări ortogonale, transformata W reprezintă semnalul atât în timp, cât și în frecvență. De exemplu, să considerăm un semnal în timp care conține numai o formă îngustă și izolată la o locație $t = t_0$. Dacă în domeniul Fourier acest semnal are un spectru larg, adică sunt necesare multe componente pentru TF a semnalului, e foarte probabil ca transformata W să modeleze această “denivelare” cu un număr mic de componente/coeficienți. Pentru că modelarea denivelării nu are un efect global – componentele transformării W sunt localizate în timp, spre deosebire de transformarea F . Datorită acestor avantaje specifice, transformata W a fost folosită pentru a înlocui transformata Cosinus discretă din standardul de compresie JPEG (1992) în noua versiune JPEG 2000. S-a proiectat chiar o metodă de codare pe baza transformatei W : EZW. EZW se bazează pe o codare / comprimare *progresivă*: cu cât mai mulți biți se adaugă, cu atât codarea câștigă în acuratețe. Rezultatul e o comprimare remarcabilă, cu proprietatea că poate avea *orice* rată. “Orice”, desigur dacă acceptăm o comprimare cu pierderi. În varianta comprimării fără pierderi rezultatele nu mai sunt atât de spectaculoase.

Există adaptări ale undinelor, într-o adevărată zoo de ‘objectlets’:

- *Multiundinele*: avem mai mult de o funcție de scalare / undină → o aproximare mai bună a unor semnale;
- *‘undine difuzie* : permit analiza multiscală pe structuri generale, ca structuri “variate” ce se pot modela în spațiul euclidian (=‘*manifolds*’), grafuri și nori de puncte în spațiul euclidian;
- *‘Brushlets*’: [Meyer1 & Coifman, 1997] au introdus *brushlets* pentru a obține o rezoluție unghiulară mai bună decât cu pachetele undină;
- *‘Wedgelets*’ [Donoho, 1999] e o colecție organizată diadic de funcții indicator cu o varietate de locații, scale și orientări. Ele sunt folosite pentru a estima locația unei muchii fine într-o imagine cu zgomot;
- *Dicționarul de ‘beamlets*’ [Donoho et al., 2001] e o colecție organizată diadic de segmente de linie, similar *wedgelets*’-urilor. Transformarea beamlet a unei imagini $f(x,y)$ e o colecție de integrale ale lui f pe fiecare segment din dicționarul beamlet; rezultatele sunt stocate în piramida beamlet. Graful beamlet are ca vârfuri pixeli, iar ca muchii beamlets: o cale în acest graf înseamnă un poligon în imaginea originală. Autorii formulează algoritmi bazați pe beamlet-uri, capabili să identifice și să extragă beamlet-uri și lanțuri de beamlet-uri cu proprietăți speciale. Acești algoritmi s-au dovedit în practică a avea performanțe fără precedent, de exemplu, în detecția curbilor fine în imagini cu mult zgomot;

- ‘*Ridgelets*’ sunt o bază ortonormală pentru L_2 , proiectată pentru a reprezenta eficient funcții care au singularități/discontinuități de-a lungul “culmilor.”;

- în [Starck et al., 2002] s-a construit un cadru strâns de ‘*curvelets*’ care furnizează o reprezentare stabilă, eficientă și aproape optimală a obiectelor cu discontinuități de-a lungul curbilor. Punând un prag “naiv” transformării curvelet, se obțin rate ale aproximării comparabile cu cele obținute prin scheme adaptive complexe care încercă să înregistreze setul de discontinuități;

- ‘*Countourlets*’ [Do & Vetterli 2005] presupun o transformată direcțională multirezoluție, asemenea curvelets, dar direct în domeniul discret. Ele formează un cadru strâns cu redundanța 4/3. Transformata counterlet are o implementare rapidă $O(n^2 \lg(n))$, cel puțin pentru counterlet-urile cu suport compact;

- ‘*Bandelets*’ [Peyré & Mallat, 2005] de Pennec și Mallat e încă o abordare în dezvoltarea de reprezentări eficiente mixând idei venind din domeniul analizei multirezoluție și al geometriei

... și cu siguranță problemele specifice vor aduce multe alte adaptări undinelor clasice. E interesant de semnalat că proprietățile benefice ale undinelor, ca adaptivitatea și comportamentul multirezoluție, le-au recomandat chiar și în domeniul ecuațiilor diferențiale.

Undine de generația a doua

Undinele de generația a doua (‘Second Generation Wavelet Transform’ = SGWT) nu sunt undine în sensul clasic (undine de generația întâi). Ele nu sunt proiectate explicit, ci în funcție de datele de reprezentat/comprimat, ele înseamnă, mai degrabă, o “schemă de proiectare” (‘*Lifting Scheme*’ – vezi [Sweldens, 1997]), transformată rezultând în urma aplicării acestei scheme. De fapt, secvența pașilor aplicării schemei poate fi convertită într-o transformată W .

Nevoia de îmbunătățire a undinelor clasice vine din neajunsurile inerente acestor undine prin chiar construcția lor. Principala limitare e că ele nu sunt eficiente mai ales în ce privește domeniile limitate, neperiodice, ceea ce e cazul multor aplicații. Mai mult, chiar semnalele 1D sunt adesea eșantionate neregulat. În multe probleme, metrica nu e “plată”: avem de-a face cu suprafețe și spații variate(‘manifolds’). Întrebarea e cum să păstrăm proprietățile benefice ale undinelor, adică localizarea în timp-frecvență și algoritmi rapizi, ocupându-ne și de geometrii mai puțin simple. Răspunsul pe care l-a dat Wim Sweldens a fost să renunțăm la translații și dilatări, să renunțăm la analiza Fourier și să folosim în loc noua abordare dată de ‘The Lifting Scheme’. Ideea de bază e să împărțim semnalul în eșantioane pare și eșantioane impare, adică să împărțim semnalul original x_k în seturile $\{x_{2k}\}$ și $\{x_{2k+1}\}$. Să ne reamintim că semnalele digitale sunt, în general, puternic corelate, această corelare fiind locală. De aici, ideea de *a prezice partea impară din partea pară*. Ce e pierdut în urma pezierii se numește *detaliu*. Apoi partea pară e ajustată pentru a servi ca aproximare a semnalului original. Algoritm:

- 1) Împarte semnal de intrare: par și impar
- 2) Predicție impar din par: $\text{detaliu} = \text{impar} - P(\text{par})$
- 3) Actualizează par folosind detaliu: $\text{aproximare} = \text{par} + A(\text{detaliu})$ astfel încât aproximarea să nu schimbe media semnalului original

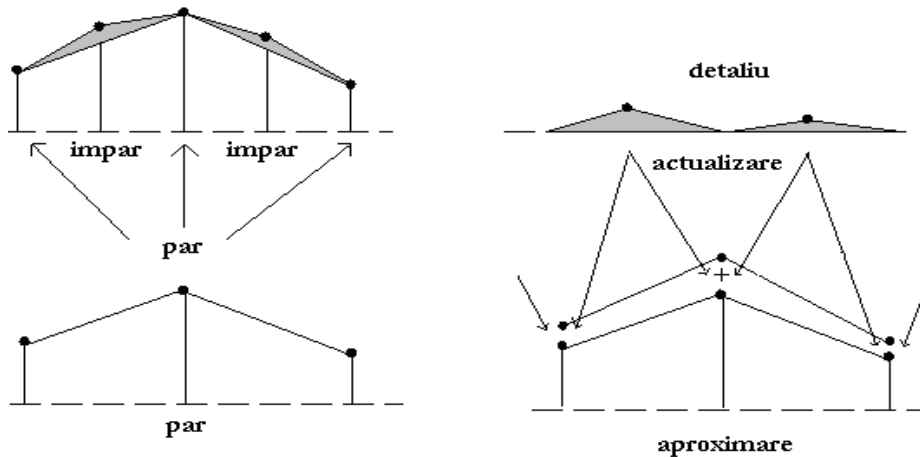


Fig.II.1.4_3 ‘The Lifting Scheme’ pentru undinele de generația a doua

Partea impară e precisă liniar din partea pară; în pasul 3, detaliul calculat la pasul 2 e folosit pentru a actualiza, tot liniar, partea pară astfel încât să se mențină media semnalului original. Procedul se repetă până când ajungem la o aproximare formată dintr-un singur număr reprezentând media semnalului original.

Observație: Pentru o eșantionare neregulată, operatorii de predicție și actualizare vor varia corespunzător de la o locație la alta. Acești pași încorporează elementele cheie ale transformatei undină clasice. Se pot calcula ușor coeficienții “Fourier”, din aproximații și detalii.

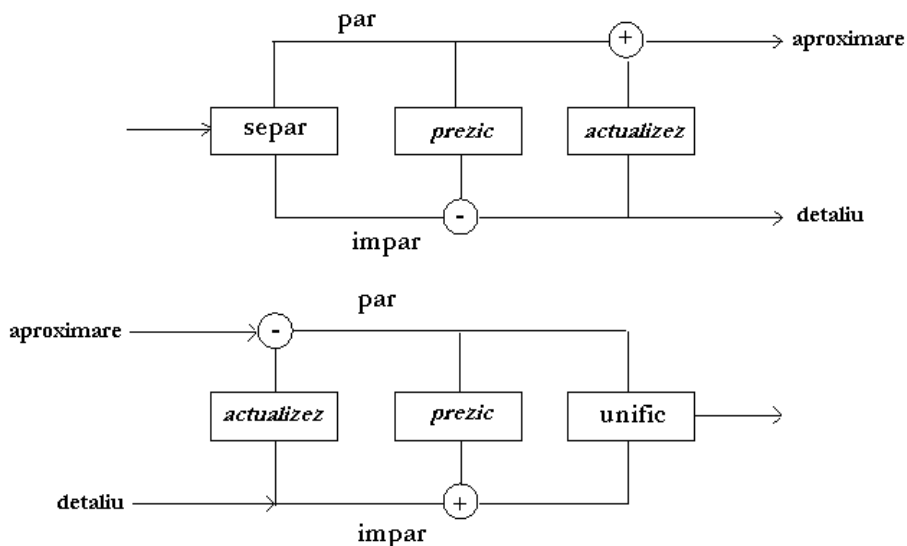


Fig.II.1.4_4 ‘The Lifting Scheme’ / TW de generația a doua directă și inversă

Pentru domeniul 2D se poate extrapola acest mecanism. Posibilitatea de a generaliza pentru mai multe dimensiuni și pentru ‘manifolds’, depinde de posibilitatea “rețetizării” (‘meshing’) acestor domenii. O caracteristică nouă a undinelor de a doua generație, care poate fi observată din schemele de mai sus, e că transformarea directă și cea inversă sunt întotdeauna procese inversabile, indiferent de alegerea operatorilor de predicție și actualizare.

O strategie de compresie simplă (cu pierderi), bazată pe schema de proiectare descrisă, e să “debarcăm” detaliile înainte de stocare. O aplicație nouă a undinelor e analiza datelor geografice. Ne putem gândi la topografia Pământului ca o funcție definită pe o sferă. Datorită flexibilității schemei de proiectare, putem crea undine care se află pe sferă. Astfel datele topografice pot fi comprimate și manipulate asemenea unui semnal 1D. Desigur că această abordare poate fi exploatată și în grafica computerizată. Un obiect 3D triangularizat poate fi analizat folosind o schemă multirezoluție dacă ne gândim că triangularizarea e domeniul de definiție al unei funcții, iar coordonatele vertex-urilor sunt valori ale funcției. De asemenea, se studiază posibilitatea folosirii analizei multirezoluție pentru a rezolva mai eficient problema luminozității în grafica computerizată.

Am văzut că analiza multirezoluție e un puternic concept general, care se pretează la multe adaptări practice și adaptări în ideea optimizării atât a acurateței, cât și a vitezei de execuție. **Am derivat întregul arsenal de analiză multirezoluție, pornind de la conceptul central de ortogonalitate.** Am văzut că ultimele tipuri de undine nu mai operează în domeniul analizei în frecvență (domeniul Fourier), pentru a oferi o adaptabilitate și viteză maximă. Remarcăm că unele tipuri de undine nu mai operează în domeniul analizei în frecvență (domeniul Fourier), pentru a oferi o adaptabilitate și viteză maximă. Înainte de a intra în domeniul fuzzy, pentru a introduce acest domeniu, vom face o sinteză.

II.1.5 O viziune abstractă asupra recunoașterii formelor, în jurul conceptului de corelare

Ne propunem să captăm într-o viziune unitară, într-o singură imagine, esența (matematică) a problematicii recunoașterii formelor ('pattern recognition'), în principalele domenii de reprezentare a semnalelor: domeniul spațial (sau temporal, sau de altă natură), domeniul frecvență (frecvențele pot fi spațiale, sau temporale, sau de altă natură) și domeniul probabilităților. Se spune că oamenii dobândesc informații de cea mai înaltă calitate din imagini și concepte. Iată o **schiță originală care grupează cele 3 domenii în jurul conceptului de corelare**:

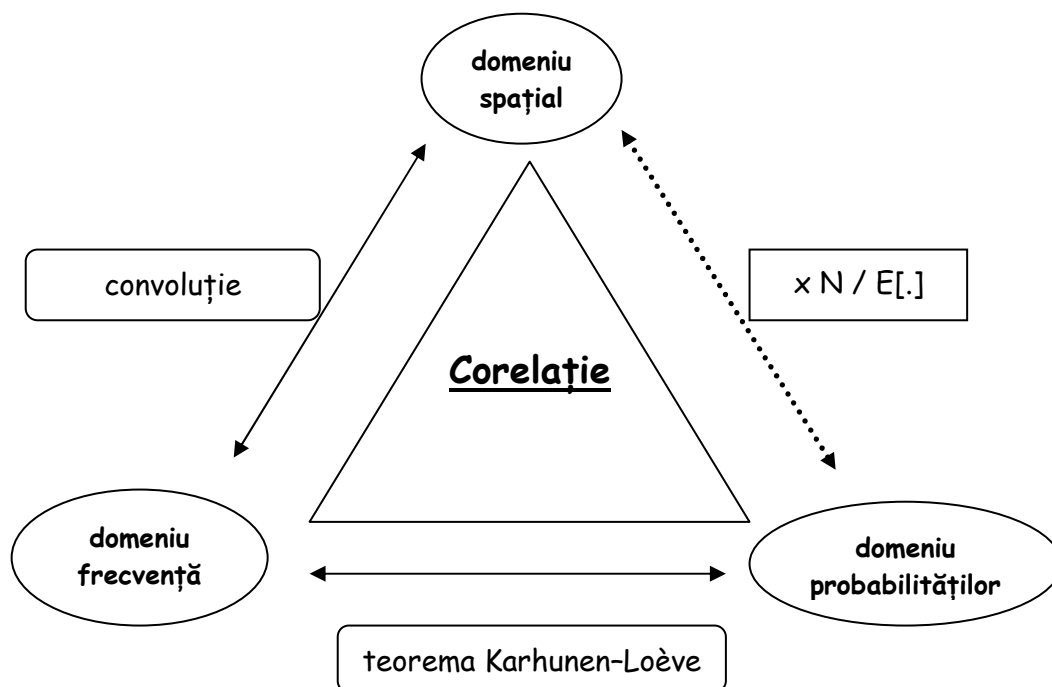


Fig. II.1.5_1 Imagine de ansamblu a relațiilor fundamentale între domeniile: spațial / temporal, frecvență și al probabilităților

Relații fundamentale între cele 3 domenii:

- Teorema de convoluție (sau de corelație, conform [Gonzalez & Woods, 2008, p. 255]) stabilește o relație între domeniul spațial și domeniul frecvență.

- Prin Teorema Karhunen–Loève (vezi wikipedia) un process stochastic poate fi descompus ortogonal, similar descompunerii Fourier. Importanța acestei teoreme e că descompunerea e optimă în ce privește minimizarea erorii pătratice medii. Sau, cu alte cuvinte, în cazul acestei transformări în domeniul frecvență, compactarea energiei e maximă → putem reconstrui cu o bună aproximație semnalul inițial luând în considerare numai primii coeficienți ai transformatei.

- Relația între domeniul spațial și domeniul probabilistic e asemenea celei dintre ceea ce vedem la un moment dat pe o scenă și istoria evenimentelor petrecute pe acea scenă. Dacă ‘x N’ înseamnă acumularea de «imagini» spațiale care ar putea genera o statistică suficientă, E[...] înseamnă așteptarea (‘expectation’) sau media, operator care estimează cea mai probabilă realizare conform cu datele probabilistice.

În esență, problema recunoașterii formelor (‘pattern recognition’) sau clasificarea bazată pe forme / regiuni / șabloane e o problemă de corelare. O observație generală, ce se degajă din rezultatele experimentale ale unei largi categorii de cercetători, e aceea că performanța unei metode vine odată cu adaptabilitatea metodei la spațiul concret al problemei. Vezi cercetarea adaptivă (‘adaptive research’) din sectorul clinic - conform [Rosenberg, 2010],[Kruschke, 2011] ș.a. - sau planificarea cultivării terenurilor (‘land use planning’) [N. van Lier, 2002]. În capitoul IV vom numi aceasta “principiul specializării și al calibrării”. KLT e un prototip din acest punct de vedere: interpretând geometric o transformare unitară ca rotație a axelor de coordonate, rotație caracterizată de o matrice de rotație. Putem vedea KLT ca transformarea unitară care reprezintă forma de intrare / semnalul în sistemul de coordonate al vectorilor săi proprii – lucrând în acest sistem poate fi foarte folositor pentru

recunoașterea formelor: putem aproxima optimal.

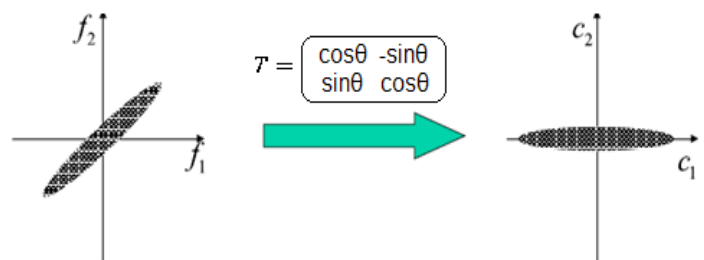


Fig. II.1.5_2 KLT ca transformare în sistemul de coordonate al vectorilor proprii ai semnalului de intrare: de la variabile aleatoare puternic corelate (energii egale) la variabile aleatoare decorelate, cu ajoritatea energiei în primii coeficienți.

În marea lor majoritate, problemele de clasificare contemporane sunt probleme complexe, de clasificare neliniară, cu granițe imprecise între clase (clasele se pot suprapune). Abordarea lor riguroasă se face în cadrul teoriei fuzzy. La Fig. II.1.5_1 am mai putea adăuga punctat:

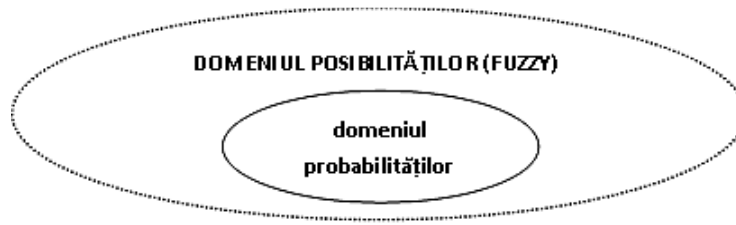


Fig. II.1.5_2 Relația între domeniul probabilităților și cel al posibilităților (fuzzy) vezi (mai jos) subsecțiunea “II.2.2.1. Teoria fuzzy și teoria posibilităților”

La finalul acestei secțiuni putem da câteva grupuri de concepte sinonime sau, cel puțin, “înrudite”, din diferite puncte de vedere:

- * algebra lineară: produs intern peste un spațiu vectorial, nucleu simetric și separabil, bază ortogonală, proiecție și decompoziție, factorizare, valori și vectori proprii, transformări unitare;
- * principiul nedeterminării al lui Heisenberg: stare-proces, poziție (sau timp)-frecvență, principiul incompatibilității (vezi mai jos subsecțiunea II.2.1.2., “Logica fuzzy, utilitatea și legătura ei cu problema clasificării”) în teoria fuzzy;
- * teoria probabilităților: varianță, covarianță, corelație (=covarianță normalizată),
- * procesarea și analiza imaginilor: corelație spațială (= convoluție “în oglindă”), produs scalar în domeniul frecvență, potrivirea șabloanelor (‘template matching’), ‘pattern recognition’, analiza componentelor principale

În continuare ne vom ocupa cu precădere de domeniul fuzzy. Sumar spus, dacă domeniul probabilistic exprimă o luare în calcul a statisticilor, domeniul «fuzzy» exprimă o luare în calcul a intuițiilor.

II.2. Granularitate și gradație în Domeniul Fuzzy

II.2.1. Logica fuzzy și “oficializarea” unei alte paradigme de cunoaștere

II.2.1.1. Promisiunile începuturilor erei informatice și problemele actuale ale inteligenței artificiale

Trecem peste perioada anterioară apariției calculatorului, apariție care, se spune, a inaugurat a “doua revoluție științifică”, pentru a observa că la începuturile “erei informatice”, în bună tradiție a spiritului iluminist, optimismul era nemăsurat. În [fz lz], L.A. Zadeh, inițiatorul logicii fuzzy, spune că “încă din anii '40, când singurele calculatoare existente erau cele cu relee, se iveau titluri răsunătoare ca: "Electric Brain Capable of Translating Foreign Languages is Being Built." Nici în zilele noastre, 60 de ani mai târziu, calculatoarele nu s-au apropiat măcar într-un grad considerabil de nivelul general de inteligență umană.” După [fz lz], iată o tematică reprezentativă a problemelor (multe dintre ele probleme de CI) provocatoare pentru construcția de mașini / calculatoare cât mai “inteligente”:

- recunoașterea scrisului de mână
- recunoașterea vorbirii
- înțelegerea limbajului natural (NL)
- rezumarea
- eliminarea ambiguității
- înțelegerea imaginilor și recunoașterea formelor
- diagnoza
- recuperarea și stocarea informației nestructurate

- execuția instrucțiunilor de nivel înalt (exprimate în NL)
- învățarea
- raționamentul
- planificarea
- luarea deciziilor

În [Balaș & Balaș, 2007] se asociază apariția Inteligenței Artificiale ('AI') cu apariția Electronicii, dar și a Postmodernismului. Astfel, Electronica ar fi prima tehnologie care permite controlul energiei și informației. În acest articol se mai spune că, deși "moderniștii" contestă abordarea "fuzzy" a "postmoderniștilor", singurul avans real în AI a avut loc în domeniul "*soft computing, i.e. clustering fuzzy logic, neural networks, genetic algorithms and evolutionary computing*". Se mai spune că, dacă modernii puneau accentul pe materie și energie, postmodernii pun accentul pe formă și model. De unde și **euristica sistemelor expert de tip fuzzy**: "în loc să rezolvăm ecuații diferențiale, integrăm numeric prin simulare pe calculator" (vezi și metoda Monte Carlo). Zadeh crede că această evoluție a fost condiționată în principal de evoluția hardware și software a calculatoarelor și senzorilor, paradigma computațională de prin care s-a avansat fiind, de asemenea, aceea de 'soft computing' – calcul care se poate adapta domeniului impreciziei problemei. Concluzia principală pe care o trage Zadeh [fz lz] e că pentru a merge în direcția obținerii unei inteligențe a mașinii cât mai apropiată de cea umană, e necesar să avem o capacitate de a lucra cu informații bazate pe percepție.

II.2.1.2. Logica fuzzy, utilitatea și legătura ei cu problema clasificării

În 1973, Zadeh a enunțat un adevăr pe cât de simplu, pe atât de general și fertil: "Pe măsură ce complexitatea unui sistem crește, capacitatea noastră de a face afirmații precise și semnificative despre comportarea sistemului respectiv se diminuează până e atins un prag, dincolo de care precizia și semnificația (sau relevanța) devin caracteristici mutual exclusive" (**Principiul Incompatibilității**).

Intuitiv, putem traduce acest principiu așa: omul are abilitatea ca, în fața complexității, să neglijeze detaliile ne semnificative pentru a ajunge la o idee mai clară. C.V.Negoită, (co)autor al uneia din primele monografii dedicate domeniului fuzzy [Negoita & Ralescu, 1974], a postulat în cibernetică principiul ‘**pull-back**’, care, în esență, spune că pentru a obține o agregare informațională, o creștere a complexității percepției, trebuie să ne retragem. Pull-back poate explica nevoia de abstractizare ca nevoie de stabilitate structurală. În [pb VN] Negoită afirmă: “Pentru a înțelege ansamblul, trebuie să te repliezi spre a dobândi perspectiva și a reduce lumea la proporții din sfera realizabilului. Detaliile imensului tablou se pierd, dar subiectul în întregime lui e cuprins dintr-un punct. Pullback-ul nu e marginal. În afara pullback-ului nu putem opera cu lumea. Pullback-ul, paradoxul, antinomia reunesc lucruri care înainte se aflau izolate. Trebuie să învățăm cum să le identificăm în diversele lor manifestări”.

În general, sisteme reale sunt complexe. Orice enunț asupra unui astfel de sistem, pentru a cuprinde cât mai multă semnificație, va fi cu atât mai imprecis, nedeterminat, vag. Noțiunea de **nedeterminare** e pusă astfel în legătură directă cu semnificația; pare că redescoperim principiul incertitudinii, al lui Heisenberg.

Când a introdus pentru prima dată termenul de ‘fuzzy’, în 1965, profesorul Zadeh l-a folosit pentru a descrie mulțimile de elemente multivalente. Mai târziu el va spune: “Articolul meu din 1965 despre seturile Fuzzy reflecta concluzia mea că era nevoie de o teorie a impreciziei și incertitudinii care să nu se bazeze pe teoria probabilităților”). În lucrarea sa, [Zadeh, 1965], pleacă de la o logică cu mai multe valori, concept uzitat în anii ’20 în legătură cu Principiul Nedeterminării formulat de Heisenberg în mecanica cuantică. Logica cu mai multe valori a avut un pionier în Jan Łukasiewicz, care a formulat logica trivalentă – logica de la care a plecat și Zadeh. În ce privește alegerea cuvântului ‘fuzzy’ – în engleză: vag, cețos, rufos – Zadeh spune: “Motivul pentru care am ales cuvântul fuzzy e că ceea ce am avut în minte sunt **clasele care nu au granițe precis delimitate**”. Conceptul de mulțime fuzzy, a apărut ca o urmare firească a necesității de a modela un sistem imposibil de definit exact (‘ill-defined’), adică cu ajutorul unor instrumente matematice precise. Imperfecția definiției unor astfel de sisteme – majoritare în lumea reală – se poate vedea sub forma falsității sau impreciziei datelor. În lucrarea noastră avem în vedere mai ales imprecizia datelor.

Dăm câteva exemple ale utilității utilă logicii fuzzy. Chiar dacă nu există (pentru marele public) încă mașini de calcul care construite (la nivel microelectronic) pe baza unei logici trivalente, multe aplicații deja utilizează această logică pe computerele binare actuale. Astfel, limbajul SQL (Structural Query Language), folosit pentru lucrul cu baze de date, folosește o a treia valoare numită NULL, cu scopul de a arăta absența (sau necunoașterea) unei valori 0 sau 1. Alt exemplu:

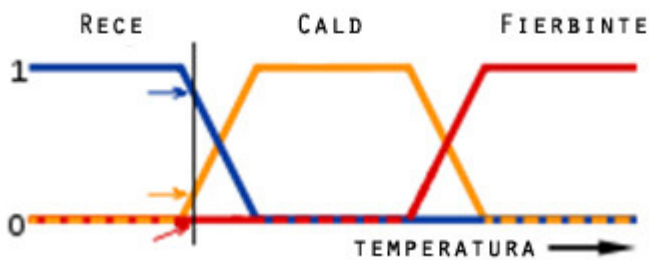


Fig. II.2.1_3 Reprezentarea grafică a unor aprecieri calitative [in fz]. Mărimea fizică “temperatură” poate fi descrisă prin cuvintele “rece”, “cald” și “fierbinte”, cărora li se atribuie valori cuprinse între 0 și 1, valori procesate într-un sistem inteligent, care poate să comande un sistem de încălzire-răcire, e.g., un termostat (în acest caz ar fi nevoie de numai două cuvinte pentru descrierea schemei de funcționare: “rece” și “cald”).

În general, informațiile pe care le avem - despre rezultatele posibile, despre valoarea unei noi informații, despre felul în care condițiile se modifică în timp (*dinamicitate*), chiar și despre preferințele noastre pentru diverse opțiuni – e destul de vagă, ambiguă. Anumite informații ar putea fi aleatorii și deci le-am putea modela cu ajutorul teoriei probabilităților. Prin “timp probabil” înțelegem, pornind de la termenul de “probabil”, în sensul teoriei probabilităților, mai degrabă o prognoză pe baza unei statistici care a “numărat” evenimentele meteo de-a lungul multor ani – o logică a frecvenței relative a fenomenelor meteo într-o anumită perioadă temporală. Pe când, fără să dăm la o parte “experiența” în studiul acestor fenomene, care a validat / ajustat diverse modelări teoretice, prognoza se face în principal pe baza unor modelări și măsurători recente. Modelările, chiar dacă încorporează multă matematică, nu pot fi obiective. Deci, plauzibilitatea are o inerentă natură subiectivă, fie ea de consens colectiv și validată îndelung. În general, chestiuni cum sunt preferințele personale, obiective / criterii de apreciere multiple, **evaluări subiective și consens colectiv, s-ar preta ca, în loc să fie abordate (numai) probabilistic, să fie abordate (și) fuzzy.**

Rezumăm câteva situații care se pretează la folosirea logicii fuzzy:

- logica fuzzy devine necesară în situațiile când **percepția umană**, judecata umană și luarea de decizii sunt implicate inextricabil.

- întrebare: cum inserăm **intuiția** în rezolvarea unor probleme? Teoria fuzzy își propune să răspundă și la această întrebare.

- modelarea prin **expresii lingvistice**, dată fiind creșterea interesului pentru modelarea prin cuvinte în diferite arii științifice (în probleme de decizie, științe sociale, teoria controlului etc.). Acesta e un atu foarte important al logicii fuzzy. Prin natura sa, limbajul e vag și imprecis. Totuși prin limbaj se schimbă cele mai multe informații. În pofida impreciziei limbajului natural, oamenii înțeleg conceptele și ideile celorlalți. Acest fel de înțelegere nu e posibil în comunicarea cu un calculator. Instrucțiunile oferite unui calculator trebuie să fie (extrem de) precise. Să luăm înțelesul sintagmei "persoană înaltă". Omul nu "vizualizează" un prag fix când se gândește la o "persoană înaltă." Mai mult, aprecierile pot diferi de la om la om. Și, totuși, e surprinzător faptul că în ciuda impreciziei și a "relativismului" persoane diferite se gândesc la aproximativ aceeași înălțime (indiferent de înălțimea lor), comunicarea și înțelegerea corectă și eficientă fiind posibile între ei, fără să fie nevoie să știe o aceeași definiție exactă a cuvântului "înalt."

- **reducerea costurilor**. Indubitabil, "abilitatea" fuzzy nu poate rezolva probleme ce necesită o precizie mare – probleme cum ar fi focalizarea unei raze de electroni microscopice pe un specimen de mărimea unui nanometru. Dar, în general, multe problemele nu necesită o precizie prea mare; probleme precum parcarea unor mașini, susținerea unei remorci, conducerea unei mașini printre altele pe o autostradă, spălarea rufelor, controlarea traficului într-o intersecție ș.a.m.d. Modelele sau produsele ingineresti clasice necesită precizie, ceea ce se repercutează în costuri înalte și timpi lungi de producție și dezvoltare. Când se ia în considerare utilizarea logicii fuzzy într-o problemă dată, trebuie evaluată și toleranța la imprecizie. ***Precizia ridicată nu implică doar costuri mari, ci și maleabilitate scăzută în rezolvarea multor probleme.*** Există probleme care necesită exploatarea impreciziei. De exemplu "problema comisului-voiajor." În această problemă clasică de optimizare (*euristică*), un comis voiajor își propune să minimizeze distanța totală parcursă considerând itinerarii și variante între o serie de orașe într-o călătorie anume. Pe măsură ce numărul orașelor crește, problema atinge cu rapiditate o explozie combinatorică: dacă aplicăm "metoda forței brute" complexitatea e $O(n!)$, varianta optimizată – algoritmul Held–Karp – obținând $O(n^2 2^n)$ ceea ce e, de asemenea impractic. Există probleme reale, practice asemănătoare cu problema comisului-voiajor. De exemplu în procesul de fabricare a bordurilor de circuite, lasere precise sapă sute de mii de găuri în bord: stabilirea ordinii în care să se efectueze găurile precum și minimizarea timpului de fabricație e o problema analoagă problemei comisului voiajor [Kolata, 1991]. Au fost implementați algoritmi de rezolvare a problemei comisului-voiajor optimi într-un anume sens: dacă răspunsul exact în timp real nu e garantat, în schimb putem obține un răspuns mai puțin exact în timp real - un răspuns cu mult mai puțin efort de calcul.

- Logica fuzzy poate **integra / fuziona informații imprecise**. Cunoașterea fuzzy poate fi folosită în combinație și cu cea probabilistică, e.g., s-a proiectat luarea deciziilor în sistem Bayesian, prin introducerea informațiilor fuzzy. Vezi și subsecțiunea următoare.

În general sistemele simple liniare sau procesele automatizate conform unei scheme “rigide” nu se pretează la a fi îmbunătățite apelând la o logică fuzzy. Totuși aplicabilitatea acestei logici a fost demonstrată în multe procese / produse care interesează un număr mare de consumatori. Conform [fz ap] “în videografie firme ca Fisher, Sanyo și alții construiesc aparate de filmat ('videocamera recorder'/camcoder') bazate pe o logică fuzzy de focalizare și stabilizare a imaginii. Mitsubishi comercializează aparate de aer condiționat care controlează schimbările de temperatură în funcție de confortul omului. S-au construit și mașini de spălat fuzzy: o mulțime de senzorii detectează culoarea, felul materialului de spălat și gradul de murdărie și un microprocesor fuzzy selectează cea mai potrivită combinație (din 600 disponibile) între temperatura apei, cantitatea de detergent necesară, durată și numărul de rotații. Orașul japonez Sendai are o rețea de linii de metrou cu 16 stații care e controlată de un computer programat pe sistem fuzzy. Călătoria e atât de confortabilă și de liniștită încât călătorii nu sunt nevoiți să se țină de bare, iar conducătorul-computerizat face în medie cu 70% mai puține erori când accelerează și frânează, decât fac în medie operatorii umani. În Japonia de la prăjitoare de pâine, aspiratoare și până la procedee de control aplicate în industrie, sunt multe sisteme construite pe principiile logicii fuzzy.” Se poate pune și întrebarea: “De ce găsește aplicabilitate în numeroase produse de consum în Japonia, o paradigmă inventată în SUA pentru a modela imprecizia în sisteme complexe? O explicație ar fi că spiritul japonez e unul antreprenorial și că japonezii sunt educați să valorifice orice idee nouă din cercetare, să-i găsească aplicabilitate mai repede decât oricine altcineva. Culturile vestice par fixate pe mentalități gen: alb ori negru, da ori nu - caracteristice sistemului binar al logicii aristotelice. În plus, termenul “fuzzy” are conotații negative în engleză, sugerând dezorganizare sau imprecizie. Nu și în alte limbi, de exemplu în cele asiatice. Să considerăm sintagma oximoronică “focalizare fuzzy (vagă)” pentru noile camere de filmat cu stabilizarea imaginii. Ar cumpăra americanul de rând o asemenea cameră de filmat? O altă explicație a faptului ca logica fuzzy a fost acceptată cu mai mare promptitudine în Asia ține de *limbaj*. În japoneză simbolurile kanji, adoptate din chineză, sunt descrieri vagi, conceptuale ale obiectelor. De asemenea, logica și filosofia Orientului nu au insistat pe precizie, separare netă, fiind mai degrabă integratoare.”

Și în România, logica fuzzy a avut o timpurie, aparte și interesantă prezență. Grigore Moisil, unul din cei mai importanți promotori ai ei, a avut ideea de a construi un calculator “în baza 3”, conceput astfel încât dialogul om-calculator să poată fi îmbogățit prin includerea unor valori precum "îndoielnic" sau "problematic", intermediare între adevărat și fals. Tot Gr. Moisil a reformulat logica fuzzy într-o axiomatică algebrică și a extins logica la n-valori (așa numitele logici n-valente sau poli-valente). Sistemul algebric creat de pe fundamentul lui Jan Łukasiewicz

e cunoscut ca algebra Łukasiewicz-Moisil. Cu referire la logica multivalentă investigată la un nivel algebric de Moisil, Solomon Marcus a apreciat că originalitatea ei constă în faptul că “reține din structura conceptelor vagi nu atât aspectul impreciziei, cât pe acela al nuanțării” - vezi [Moisil, 1975]. Deschiderea de orizont dată de conceptul de set fuzzy, a fost remarcată de Moisil, care, în prefața acestei cărți (una din primele apărute în România, despre logica fuzzy), spunea: “această idee putea fi înțeleasă ca extensia unui predicat în logica cu o infinitate de valori.” Ca și Zadeh, el dădea o semnificație logicii cu o infinitate de valori și afirma că această semnificație putea interesa un domeniu al tehnicii: teoria automatelor și teoria sistemelor. Moisil mai explică în prefața cărții: ”Logicile cu un număr finit de valori puteau fi concepute ca aproximări ale unei logici cu o infinitate de valori. Totuși, dezvoltând studiul logicii predicatelor și relațiilor în cazul polivalent, ne-am lovit de o dificultate care ne-a părut foarte importantă. Propozițiile de identitate $a = b$ nu puteau avea decât două valori. [...] *Numai studiind teoria clasificării automate am înțeles care e calea de urmat pentru construirea unei teorii a predicatelor în logica cu mai multe valori, teorie care să cuprindă și identitatea.* Prezentele lecții au fost elaborate pentru înțelegerea acestei direcții: **identitatea, în logica cu mai multe valori, nu e decât o clasificare.** Or, aceasta ne-a condus la înțelegerea dificultăților întâlnite de aceia care ar fi vrut să întrebuițeze matematica, dar care aveau de a face cu concepte prost definite: *fuzzy*, nuanțat, precum și dificultățile filosofice de care se loveau cei care întrebuițau statistica în studiul fenomenelor necantitative.” Să remarcăm și distincția care se face între statistica fenomenelor cantitative și logica fuzzy, care se pretează fenomenelor necantitative, distincția între “*randomness*” și “*fuzziness*.”

Acceptarea principiului incompatibilității s-a tradus în realismul abordării unor probleme nedeterminate - majoritatea acestor probleme fiind de clasificare. S-a vorbit de o schimbare oficială de paradigmă epistemologică, în sensul acceptării datelor imprecise și / sau de natură subiectivă, date, în fapt, inerente multor probleme reale, sisteme complexe etc., și care, până mai înainte, nu puteau intra în corpusul “riguros” al științei. În câteva decenii, lumea științifică, chiar și cea a științelor “exacte”, a devenit conștientă de faptul că, practic, întotdeauna operăm cu informații imprecise, cu aproximații într-un anumit GRAD, cu informații subiective (conform perspectivei din care privim) În logica fuzzy, în loc de valori de adevăr, există “*grade de adevăr*”. Folosind conceptul creat de Virgil Negoită, am putea spune că atitudinea omului de

știință a “suferit” un proces de “pullback”, de desituare, sau, mai precis, de resituare față de “imperfecția” datelor.

II.2.2. Domeniul posibilului: o fuziune fuzzy a diverselor tipuri de cunoștințe

II.2.2.1. Teoria fuzzy și teoria posibilităților

În 2002, apărea o culegere de texte “Fuzzy logic: a framework for the new millennium” [Dimitrov, 2002], ale cărei părți se intitulau “Understanding Society, Mathematics, Modelling and Control Systems”. Prima parte pune în discuție diferite aspecte ale *fuziologiei*, o disciplină care se ocupă cu “fuziness inherent in human knowledge”, a doua parte explora fundațiile matematice ale “soft computing, iar a treia parte se referea la aplicațiile ingineresti. S-a enunțat și un Principiu al Conectivității în Dinamică: “Niciun lucru sau ființă nu pot exista doar în sine sau pentru sine însuși, ci numai în relație dinamică cu alte lucruri și ființe. Într-un cadru al dinamicității proceselor, cadrul propriu lumii vii, Fuzitatea apare și ca imagine și concept de aproximare (măsură) a influenței / prezenței unei multitudini de factori, multitudine imposibil de precizat”. Să ne amintim și că Zadeh propunea să ne inspirăm din capacitatea omului de a raționa, a rezuma și a lua decizii într-un mediu al impreciziei, incertitudinii și cunoștințelor parțiale, pentru a apropia nivelul inteligenței artificiale de nivelul de inteligență uman. Tot Zadeh, cel care a definit teoria fuzzy ca “**precizie a impreciziei (n.n.: precizie maximală în condiții de imprecizie)**”, avertizează asupra riscurilor pe care le poate implica această teorie, apelând, de exemplu, la ‘sorites paradox’. Traducem acest paradox în lumea GIS(‘Geographical Information System’). Să presupunem că vrem să ajungem din reședința județului A în reședința județului vecin, B - pentru acesta, plecăm din A și facem pași înspre B. Cum pasul e foarte mic comparativ cu distanța de parcurs, putem raționa după regula: printr-un singur pas nu trecem din A în B; totuși e cert că după un timp ajungem în B. Paradoxul constă în întrebarea: la al câtelea pas ajungem în B? În mod natural nu ne interesează să precizăm această informație – această

precizare ar fi și foarte costisitoare, presupunând niște condiții practic imposibil de atins. Să observăm că în *raționamentul aproximativ* iterarea unei reguli se face într-o anumită măsură (grad). **Să reținem că logica fuzzy, deși utilă în rezolvarea multor probleme de clasificare complexe, cu date imprecise, trebuie aplicată cu precauții.**

Spre deosebire de teoria probabilităților, teoria fuzzy își propune captarea informațiilor mai degrabă calitative, decât cantitative. Așa cum am menționat, conceptul de fuzzy are o legătură privilegiată mai degrabă cu domeniul plauzibilului / posibilului decât cu cel al probabilului, probabilul fiind văzut ca o restrângere / probă a posibilului. Ceea ce a fost evidențiat și în [Zadeh, 1978]. *Teoria posibilităților*, pe care o introduce Zadeh, e văzută ca o generalizare a teoriei probabilității, diferențele majore față de aceasta fiind că:

1. $p(A \cup B) = \max[p(A), p(B)]$ față de $P(A \cup B) = P(A) + P(B)$,

pentru A și B disjuncte (ceea ce nu e o condiție necesară pentru prima egalitate), p(A) fiind posibilitatea asociată evenimentului A.

2. Posibilitatea unui eveniment nu condiționează posibilitatea complementului său:

$p(\bar{A}) + p(A) \geq 1$. Aceasta e o relație care, în domeniul clasificării se traduce într-o **eventuală suprapunere a claselor**. Conceptul de posibilitate îl induce pe cel de *necesitate*: $n(A) \stackrel{\text{def}}{=} 1 - p(\bar{A})$, inegalitatea de mai sus devenind $n(A) \leq p(A)$

Și Bart Kosko [Kosko, 1991] a arătat că probabilitatea e un caz special al posibilității. El a folosit o formulare ușor diferită pentru seturile fuzzy, în care funcția caracteristică a unui set fuzzy era înlocuită cu gradul în care un set poate fi considerat subset al altui set – putem formula imediat acest lucru în termeni de clasificare!

II.2.2.2. Teoria fuzzy și bazele de cunoștințe (sistemele expert) cu informații incerte / “granulate”

De multe ori avem și vrem să valorificăm informații imprecise, vagi, imposibil de exprimat cu exactitate, e.g., ce se pot cere imaginiile cu persoanele de rasă caucaziană (ne aflăm în domeniul sistemelor expert de tip fuzzy).

Să presupunem afirmațiile:

A) Există viață pe Marte.

B) Nu există viață pe Marte.

Nu știm să decidem între A) și B). Într-o stare completă de ignoranță toate lumile sunt posibile → avem presupunerile $p(A) = p(B) = 0.5$. Totuși, 0.5 e o valoare prea mare pentru prea puține cunoștințe, adică evidențe. Teoria Dempster-Shafer a evidenței, operează în domeniul probabilităților. Plecând de la regula bayesiană, *regula de combinare Dempster-Shafer*, schimbă presupunerile în lumina unor noi evidențe. Această teorie diferențiază între *credibilitate*, care corespunde numărului de evidențe în favoarea unei propoziții, care ar corespunde probabilității, și *plauzibilitate*, care ar corespunde posibilității. Analog modului de definiție a necesității din teoria posibilității (vezi subsecțiunea anterioară), credibilitatea e definită prin relația $credibilitate(A) \stackrel{\text{def}}{=} 1 - plauzibilitate(\bar{A})$ (prin antonimie, diferența între credibilitate și plauzibilitate e aceea dintre lipsă de încredere și respectiv neîncredere). În [Zadeh, 1986] se “luminează” simplu și clar teoria Dempster-Shafer (TDS) din perspectiva bazelor de date relaționale, ca o inferență asupra relațiilor de ordinul 2, adică a relațiilor ale căror intrări sunt relații de ordinul 1. În aceste relații atributele au o *distribuție de posibilitate*, putând lua valori într-un interval - putem spune ca informația ni se prezintă granular (incertitudinea ia forma granularității). Interogarea Q asupra unor baze de date cu astfel de informații presupune un răspuns format din două părți, necesitatea și posibilitatea: $R(Q) = (N(Q), \Pi(Q))$, care corespund credibilității și respectiv plauzibilității din TDS. Zadeh arată că acestea pot fi interpretate ca prag inferior și respectiv superior ale probabilităților lui Q. $N(Q)$ și $\Pi(Q)$ reprezintă rezultatul unor medieri – într-o “ball box analogy” o bilă corespunde unei valori a atributului pe care îl avem în

vedere și o cutie corespunde unei granule, granulele putându-se suprapune, poziția bilei în cutie nefiind precizată. E important să diferențiem între o bilă **pusă** în cutie și o bilă **aflată** în cutie (pentru $C_1 \subset C_2$ putem avea mai multe bile puse în C_1 , decât în C_2). Prin această analogie **necesitatea și posibilitatea corespund operațiilor morfologice de eroziune și respectiv dilatare, cu același tip de dualitate.**

Problema normalizării

Întrebare: cum tratăm atributele “nedefinite”, adică care pot lua valoarea NULL, și informațiile contradictorii dintr-o bază de date? Necesitatea soluționării acestei probleme apare în multe situații practice. Un exemplu de atribut nedefinit e “vârsta” unei mașini (vm) aparținând lui x: dacă x nu are mașină, $vm(x) = \text{NULL}$. Într-o stare de ignoranță completă, când toate lumile sunt posibile, NULL corespunde valorii “orice”. În teoria DS a evidenței nu se ține cont de valorile NULL - *lumi închise*: nu știu P \rightarrow ignor P, spre deosebire de *lumi deschise*: nu știu să nu fie P \rightarrow “culeg” informația “P e posibil” -, ceea ce, s-a arătat în multe rânduri, e.g. în [Zadeh, 1979] sau [Zadeh, 1984], duce la rezultate contrare intuiției. Există și abordări mai complexe, e.g. *introspecția*: X nu știe P \rightarrow X știe că nu știe P, dar și ‘default reasoning’: presupun ceea ce nu pot verifica că nu e adevărat, e.g. presupun că toate bufnițele polare sunt albe. De asemenea se pune problema **consistenței** informației: cum tratăm o informație nouă în contradicție cu cea existentă. Informație care poate veni de la o **altă sursă**. Prin prisma bazelor de date relaționale, în [Zadeh, 1986] se stabilește o concluzie importantă privind regula de combinare Dempster-Shafer: în cazul unor atribute definite, intersecția atributelor asociate cu o intrare (n.n.: un rând într-o tabela care ține o relație de ordin 2, conform cu două surse diferite) nu poate fi vidă, adică relația trebuie să fie ‘*conflict free*’. În limbajul distribuțiilor granulare, *două distribuții D_1 și D_2 sunt combinabile conform regulii DS numai dacă există o relație ‘conflict free’ care să le cuprindă.*

Roboții dispun de senzori (bazați pe radar, raze X, video-camere, sunete, etc.), primind multe informații ce trebuie procesate, dar adesea aceste informații sunt contradictorii într-un grad mai mic sau mai mare. Robotul trebuie s-o prelucreze în mod automat aceste informații “paradoxiste” și să ia singur o decizie. Fuziunea acestor informații conflictuale, paradoxiste, în știință e o problemă veche nerezolvată în întregime de teoriile existente Dempster-Shafer,

Dubois-Prade, TBM a lui Smets, ori regula de combinare a lui Yager etc. Conform wikipedia: “Ca generalizare a Teoriei Dempster-Shafer a evidenței (TDS), *Teoria Dezert-Smarandache (TDSm) a raționamentului plauzibil și paradoxist* permite combinarea formală a oricărui fel de informații: certe, incerte, paradoxale. Teoria DSm poate furniza informații complexe și rezolva probleme în care TDS nu se aplică, în special când conflictele (paradoxurile) între surse devin mari și când rafinamentul cadrului de discernământ (n.n.: vezi definiția de mai jos a “seturilor grosiere”) e imposibil din cauza naturii imprecise, vagi, relative a elementelor” – vezi și [DSmT].

Aceste teorii se dovedesc utile mai ales în cadrul sistemelor expert. *Logica clasică e explozivă*: putem deduce orice dacă admitem o contradicție. Se impune să cooperăm cu informația contradictorie, fie și numai păstrând-o pentru a o clarifica ulterior. În cadrul unor *logici paraconsistente* – putem rezolva inconsistența prin retragerea, selectarea, reviziunea sau rafinarea presupunerilor, principiul general fiind acela al *schimbării minime*. Dacă Kurt Gödel a demonstrat matematic, prin teoremele sale, incompletitudinea sistemelor formale și imposibilitatea demonstrării necontradicției sistemelor formale cu mijloacele sistemului însuși, în limbajul bazelor de date acesta se traduce în imposibilitatea unei *baze de date complete*, adică a unei baze conform căreia orice propoziție adevărată se poate deduce, și respectiv a unei *baze de date verificate*, adică a unei baze conform căreia orice propoziție dedusă e adevărată.

În [Zemankova & Kandel, 1984] se propune conceptul și implementarea unei ‘Fuzzy Relational Database’ – o abordare realistă a naturii imprecise a informațiilor cuprinse în multe baze de date. Dacă în cazul bazelor de cunoștințe (sistemelor expert) clasice, cunoștința e o presupunere dovedită a fi adevărată, în cazul bazelor de cunoștințe, cunoștința e **parțial** adevărată. O altă extensie e aceea a combinării a surse de informație diferite, cu grade de credibilitate diferite.

II.2.2.3. Concluzii

Atuurile majore ale logicii fuzzy sunt date de capacitatea de a opera, într-un cadru unitar, cu mai multe tipuri de informație (de neînlocuit în cunoașterea umană): **probabilistică / posibilistică, gradată și granulată.**

În ce privește informația de tip probabilistic / posibilistic, probabilitatea poate corespunde frecvenței relative, cum e cazul tehnicilor de decizie pe bază de vot (tehnici de care vom face uz și noi în capitolul IV), e.g. transformarea Hough. Informația de tip posibilistic corespunde mai degrabă unei informații calitative, decât cantitative.

Gradația și granularitatea sunt două concepte înrudite, corespunzând modului în care se generalizează noțiunea de egalitate în cazul mulțimilor fuzzy: două obiecte sunt considerate similare, într-un anumit grad, dacă aparțin aceleiași clase (granule). **În logica fuzzy orice element / propoziție poate avea un grad de apartenență /compatibilitate /credibilitate la o clasă / concept / variabilă lingvistică sau aserțiune.** Variabila fuzzy are un domeniu granulat - prin granulă înțelegem o grupare de valori în interiorul căreia nu putem discerne aceste valori: gruparea se face după criteriile de indiscernabilitate, similaritate sau proximitate.

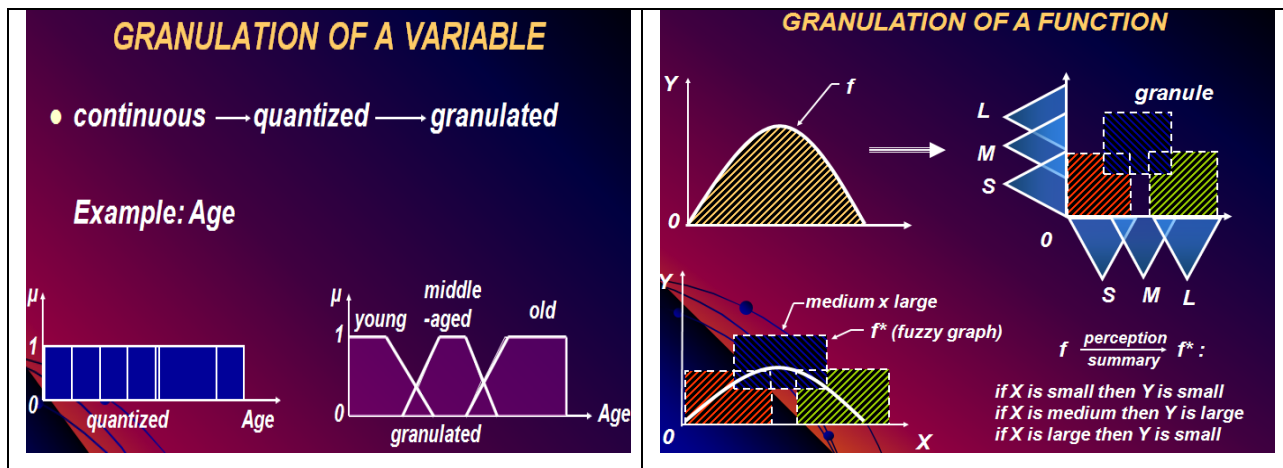


Fig. II.2.2_1 [fz lz] de granulație: variabila lingvistică “Vârsta” poate lua valorile granulare: “tânăr”, “între două vârste” și “bătrân”. Figura următoare: se poate granula și o funcție, care prin generalizare poate fi văzută ca o schemă de inferență.

Conform [fz lz] "if in bivalent logic, one writes and draws with a ballpoint pen, in fuzzy logic, one writes and draws with a spray pen which has an adjustable and precisely defined spray pattern". O alternativă la variabilele / seturile fuzzy în ce privește raționamentul în condițiile granularității și indiscernabilității sunt *seturile grosiere* ('rough sets'). Pentru o mulțime A se definesc:

1. aproximația superioară a lui A: $\bar{A} = \{g \in X/\rho \mid g \cap A \neq \emptyset\}$
2. aproximația inferioară a lui A: $\underline{A} = \{g \in X/\rho \mid g \subseteq A\}$

, g fiind o granulă corespunzătoare definirii unei relații de similaritate ρ peste X (X/ρ se numește *cadrul de discernământ*). Seturile grosiere se folosesc în ‘data mining’, pentru deducerea de dependențe funcționale. Să remarcăm asemănarea definițiilor de mai sus cu definițiile pentru operațiile morfologice de eroziune și dilatare, operații evidențiate și la II.2.2.2.

Mulțimile fuzzy adaugă o nouă dimensiune evaluării noastre prin ordonare, gradare, mulțimea fuzzy putând fi văzută ca o mulțime de mulțimi ‘crisp’ (vezi și teorema de prezentare Negoită-Ralescu). Logica booleană clasică este privită ca un caz limită (particular) al logicii fuzzy, iar probabilitatea ca un caz particular al posibilității.

Teoria fuzzy oferă un instrument matematic pentru reprezentare cunoștințelor umane de forma regulilor, pentru formularea lingvistică a informației, informație de multe ori imposibil de cuantificat numeric. Cunoștințele sunt interpretate ca o mulțime de constrângeri elastice - în cazul bazelor de date relaționale, menționat mai sus, acestea înseamnă relații de ordin 2 (intervale) - asupra unei mulțimi de variabile, *inferența fiind un proces de propagare a constrângerilor elastice*.

II.2.3. Conceptele de bază ale logicii fuzzy

II.2.3.1. Matematica fuzzy

Se poate spune ca matematica fuzzy a ajuns la o maturitate care a deschis și poate deschide părții largii aplicațiilor ingineresti, practica fuzzy consacrand această matematică. E interesant de observat că această practică provine și vine mai ales din domeniul disciplinelor “computer science”. Dacă, exagerând sensul unor tendințe dominante ale epocilor, începând cu “prima revoluție științifică”, aceea a Renașterii, și până în urmă cu câteva decenii, doar “științele exacte” “garantau adevărul”, acestea fiind considerate în fapt singurele “cu adevărat” științe, în contemporaneitate, la “a doua revoluție științifică”, aceea a noii “ere informatice”, informatica pare să aibă cea mai mare greutate în a garanta validitatea unei noi teorii. Oricum e clar ca, odata cu teoria fuzzy putem vorbi de o nouă paradigmă de cunoaștere științifică. Precum am arătat

anterior, această teorie plecând de la cautarea soluționării unor probleme de clasificare, probleme dintre care cele mai multe intră în aria inteligenței artificiale, problema noastră de CI, legată intrinsec teoriei fuzzy , devine, în contextul precizat în rezumat-ul lucrării, o problemă cu o multitudine de forme, în cele mai (aparent) diverse domenii. Ca ingineri ne interesează noua instrumentație a logicii fuzzy din punct de vedere al procesării informației, al deciziei și controlului proceselor.

În [Flondor, 2003] se remarcă: “ceea ce algebra booleană e pentru logica clasică, e algebra MV (*many-valued*) pentru logica Łukasiewicz, raționamentul aproximativ putând fi modelat în algebrele MV [...] Prin funcțiile Mc. Naughton algebrele MV pot fi aplicate în studiul rețelelor neurale”. În articol se găsește o conexiune între algebrele MV libere și haosul deterministic, văzut ca dinamică prin substituții între stări modelate printr-o algebră MV, și se evidențiază **caracterul topologic** al raționamentului aproximativ și **caracterul trunchiat al operației de adunare** în algebrele MV, caracter prin care se pot. descrie mai compact automatele finite deterministe și rețelele neurale.

În [Neumaier, 2003] se vorbește de o abordare a modelării fuzzy pornind de la conceptul de surpriză, abordare care “are o interpretare semantică consistentă - mai puțin ambiguă decât în cazul conceptului de posibilitate, permite combinarea informațiilor cantitative cu cele calitative, e o extensie directă a metodelor celor mai mici pătrate de reconciliere a conflictelor generate de date numerice aroximative, se pretează la optimizarea prin intervale și permite o abordare directă a construirii familiilor de funcții consistente cu memorii asociative de tip fuzzy precum cele utilizate în controlul fuzzy”.

Pe baza măsurii fuzzy s-a definit, într-o manieră asemănătoare integralei Lebesgue, integrala fuzzy care permite o abordare generală a problemelelor de decizie multicriterială [if fl] și de clasificare supervizată [Grabisch et al., 2000], fiind folosită ca pas de agregare peste mai multe atribute. Apartenența la o clasă se definește ca tipicalitate a valorilor, sau drept compatibilitate a elementului cu clasa.

În teoria măsurării, dar și în practica senzorilor fuzzy s-au încetățenit scările fuzzy (‘fuzzy scales’) – prin care măsurarea asociază unei stări un set fuzzy de simboluri (cuvinte), spre

deosebire de scările nominale prin care măsurarea asociază unei stări un simbol, fiind convertite numai relațiile de echivalență (în relații de egalitate).

Clasele fuzzy

Din punct de vedere matematic, apartenența la o clasă corespunde definiției unei relații de echivalență, fiind o relație reflexivă, simetrică și tranzitivă. Aceste relații se pot relaxa prin eliminarea condiției de tranzitivitate – raționamentul aproximativ (inferența fuzzy) nu poate fi aplicat iterativ (vezi ‘Sorites paradox’). Obținem o **relație de similaritate**. **Acest tip de relație corespunde clasificării fuzzy, care permite suprapuneri parțiale ale claselor.**

II.2.3.2. Mulțimi și partiții fuzzy

Teoria clasică a mulțimilor a fost fundamentată de Georg Cantor (1845 - 1918). În această teorie, în cadrul unui *univers al discursului*, U , care conține toate elementele posibile, o mulțime de elemente e formată din elemente cu aceleași caracteristici. Apartenența unui element la o mulțime e totală (100%) sau deloc (0%).

Din perspectiva teoriei mulțimilor există doar două predicate fundamentale: egalitatea și apartenența. În cazul mulțimilor generalizate ambele pot fi generalizate. În teoria fuzzy doar apartenența poate fi generalizată. În cazul teoriei clasice a mulțimilor, funcția de apartenență (f.a.) se mai numește funcție caracteristică a mulțimii, codomeniul ei fiind $\{0,1\}$. Funcții de apartenență tipice pentru reprezentarea unui set fuzzy, având codomeniul $[0,1]$ sunt sigmoida, gaussiană ș.a.m.d. Însă din considerente de efort de calcul, în practică, cele mai folosite sunt funcțiile liniare.

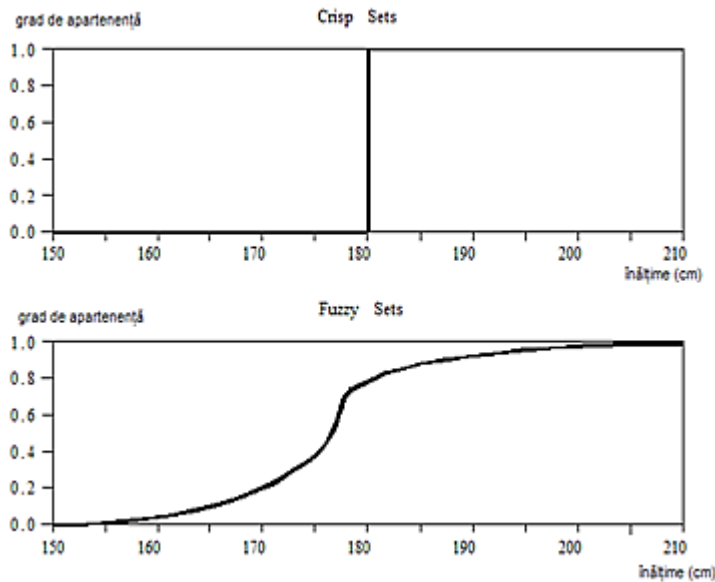
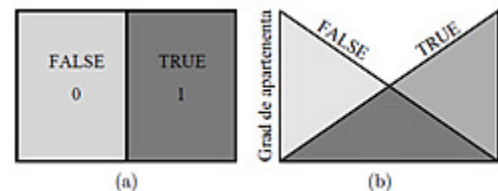


Fig. II.2.3_2 Un exemplu clasic de reprezentare fuzzy: reprezentarea conceptului de “om înalt”

În logica booleană / convențională avem doar două valori de adevăr: 0 (FALSE) și 1 (TRUE). În logica fuzzy avem un interval pentru valorile de adevăr posibile: $[0,1]$ În Fig.1. e ilustrată diferența între logica booleană și cea fuzzy: dacă în primul caz există o delimitare clară între cele două valori de adevăr, în al doilea caz există o zonă de nedeterminare, în care valoarea de adevăr poate fi 0 și 1 în același timp, într-o anumită măsură dată de gradul de apartenență.

Fig. II.2 3_1 Ilustrarea mulțimilor “TRUE” și “FALSE” în (a) logica booleană și (b) cea fuzzy



Teoria mulțimilor clasice se bazează pe logica bivalentă a terțului exclus: un obiect nu poate aparține simultan unei mulțimi și complementului sau. În teoria mulțimilor fuzzy un obiect poate fi membru al mai multor mulțimi, cu grade de apartenență distincte la fiecare mulțime. Putem spune că gradul de apartenență al unui obiect la o mulțime fuzzy exprimă compatibilitatea obiectului cu termenul lingvistic reprezentat de mulțimea fuzzy.

Def.: O *variabilă lingvistică* are ca valori termeni lingvistici într-un limbaj natural sau artificial. De exemplu, dimensiunea unui obiect e o variabilă lingvistică, care poate lua valorile “mică”, “medie” și “mare”.

Întreaga reușită practică a folosirii logicii fuzzy depinde de alegerea corectă a funcțiilor de apartenență pentru fiecare situație concretă analizată. Nu există o metodologie generală pentru stabilirea formei f.a..

Def.: Pentru o variabilă lingvistică, căreia îi corespunde o mulțime fuzzy A , se poate partiționa această mulțime în mai multe submulțimi fuzzy A_i , adică $\mu_{A_i} \leq \mu_A$. Un element $a \in A$ poate avea grade diferite de apartenență la A_i . Dacă notăm aceste grade cu u_i , adică f.a. pentru A_i , u_i poate fi interpretat drept probabilitatea ca $a \in A_i$ și dacă impunem condiția

$$\sum_i u_i = 1$$

Aceasta se numește *partiție fuzzy probabilistică*. Să observăm și că probabilitatea și fuzitatea pot colabora. Aceasta se întâmplă mulțumită relaxării pe care o induce conceptul de fuzzy.

Deși conceptul de fuzzy și cel de probabilitate sunt similare, ambele aparținând domeniului impreciziei, putându-se combina cunoștințele fuzzy cu cele probabilistice, există diferențe de categorie între ele. Probabilitatea ține mai mult o numărătoare, schimbându-se odată ce achiziționăm noi date, în timp ce fuzitatea ține mai mult de simțul comun, de intuiția formată în timp. Într-un fel, fuzitatea încercă să se ocupe de adevăr, fie el parțial (într-un anumit GRAD), în timp ce probabilitatea încercă să se ocupe de asemănare. Uneori pot echivala din punct de vedere cantitativ cele două măsuri ale impreciziei. Dacă, de exemplu, 60% din populație crede că ceva e adevărat, atunci acel ceva poate fi considerat ca adevărat într-un grad de 0.6. O proiectare fuzzy are în vedere o judecată umană - Zadeh a evidențiat că, în general, oamenii au o remarcabilă însușire de a cuantifica o funcție de apartenență, adică a alege pe o scară de la 0 la 1 gradul de adevăr al unei propoziții - și un context.

La început teoria fuzzy a lui Zadeh a fost întâmpinată defavorabil de multă lume. Argumentul cel mai tare era că funcțiile de apartenență au o inerentă componentă subiectivă. William Kahan susținea cum că logica fuzzy întreține "gândurile ilogice, neverificabile logic", numind logica fuzzy "cocaine of science" [McNeill & Freiburger, 1993]. Într-adevăr, conform paradigmei moderne de cunoaștere, a așa-ziselor "științe exacte", logica fuzzy părea prea permisivă. Profesorul Rudolph Kalman, fost student al profesorului Zadeh, afirmă: "lucrurile apar "fuzzy" (n.n.: în ceață) până când le înțelegem. Kalman mai spune: "Aș dori să comentez pe scurt

prezentarea profesorului Zadeh. propunerile sale ar putea fi criticate sever, și chiar feroce, din punct de vedere tehnic. [...] Și totuși o întrebare rămâne: profesorul Zadeh prezintă idei importante sau se complace în propuneri nerealiste?” [fz lz] Cel puțin sumedenia de aplicații practice au răspuns clar la această întrebare.

Formalizat, mulțimea fuzzy A definită pe mulțimea univers al discursului X e o mulțime de perechi ordonate:

$$A = \{\mu_A(x)/x\},$$

unde $\mu_A(x)$ reprezintă funcția caracteristică sau de apartenență a lui x la mulțimea A. Semnul ‘/’ e folosit doar pentru a delimita valoarea reală x de valoarea funcției de apartenență $\mu_A(x)$. Funcția de apartenență (f.a.) se definește pe intervalul de valori posibile ale lui x cu valori în intervalul [0,1]: $\mu_A(x) \rightarrow [0,1]$

Observație: În teoria mulțimilor există doar două predicte fundamentale: apartenența și egalitatea. În *teoria mulțimilor generalizate*, ambele ! pot fi într-un anumit grad, adică fuzzy (în cazul seturilor fuzzy doar apartenența poate fi într-un anumit grad).

Dacă funcția de apartenență e discretă, atunci mulțimea fuzzy A se scrie:

$$A = \sum_{i=1}^N \mu_A(x_i) / x_i$$

Observație: Pe lângă faptul că e compactă, această notație facilitează scrierea matriceală a *relațiilor fuzzy*, care nu sunt altceva decât produse carteziane între mulțimi fuzzy.

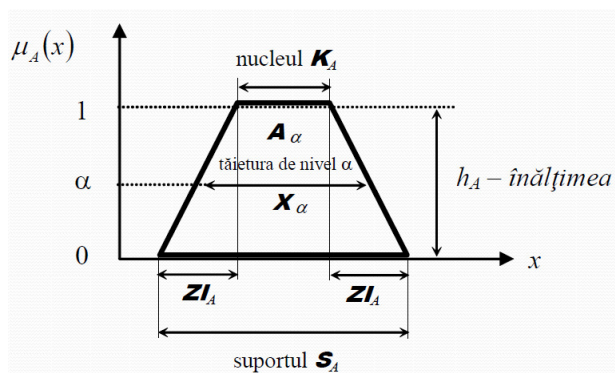


Fig. II.2.3_4 Câteva caracteristici ale unei mulțimi fuzzy (de formă trepezoidală). În cazul mulțimilor crisp, pentru intervalele ZI, f.a. s-ar fi anulat.

Tăieturile (' α -cut') / nivelele (închise) ale unui set fuzzy sunt fundamentale în teoria seturilor fuzzy. Caracterizarea seturilor prin tăieturi, adică $A_T = \{x/\mu_A(x) \geq T\}$, o dă **teorema de reprezentare Negoită-Ralescu**. Conform acestei teoreme o aplicație descrescătoare de la $[0,1]$ la $P(X)$ care satisface convergența: $A_t = \bigcap_{s < t} A_s$ corespunde unui ! unic set fuzzy în X . Și reciproc: nivelele $A_{(t)}$ ale unui set fuzzy sunt o aplicație descrescătoare de la $[0,1]$ la $P(X)$ care satisface convergența de mai sus. Observații:

1) Putem spune că **teorema de reprezentare de mai sus leagă matematica clasică / mulțimile clasice de cea / cele fuzzy**. Mulțimea fuzzy poate fi văzută ca o mulțime de mulțimi 'crisp'.

2) **Pe baza α -cut** se definesc operații între numere fuzzy.

3) α din α -cut poate fi văzut ca un nivel de aproximare al setului (sau numărului) fuzzy.

Proprietăți ale mulțimilor fuzzy:

- Normalitatea

E o mulțime fuzzy A ce e normală dacă valoarea maximă a funcției sale de apartenență e 1:

$$\bigvee_x \mu_A(x) = 1$$

În caz contrar, mulțimea se zice subnormală.

- Convexitatea

E o mulțime fuzzy A se zice convexă dacă mulțimile

$$A_T = \{x/\mu_A(x) \geq T\} \text{ sunt convexe } \forall T \in [0,1],$$

ceea ce se traduce în: orice tăietură de nivel $\geq T$ e un singur interval. Sau în:

$\lambda\mu_A(x_1) + (1 - \lambda)\mu_A(x_2) \geq \mu_A(\lambda x_1) \wedge \mu_A[(1 - \lambda)x_2]$, $\forall x_1, x_2 \in X$ și $\lambda \in [0,1]$; dacă această inegalitate e strictă, ceea ce corespunde mulțimilor A_T strict convexe (adică mijlocul unui segment cu capetele în mulțime e în interiorul mulțimii), A se zice puternic ('strong') convexă.

Observație: A convexă $\nRightarrow \mu_A$ convexă

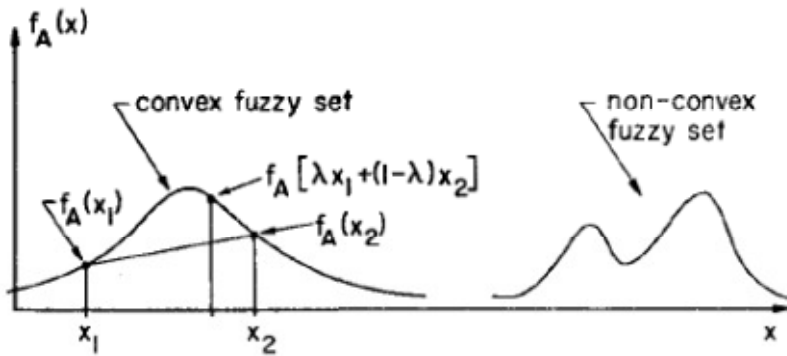


Fig. II.2.3_5. mulțime fuzzy convexă, respectiv mulțime fuzzy neconvexă [Zadeh, 1965]

Def.: Un număr fuzzy e o mulțime fuzzy definită pe \mathbb{R} , normală, convexă și cu f.a. continuă pe un suport compact. Acest număr se reprezintă de obicei ca o familie de T-tăieturi, sau ca o f.a. discretizată:

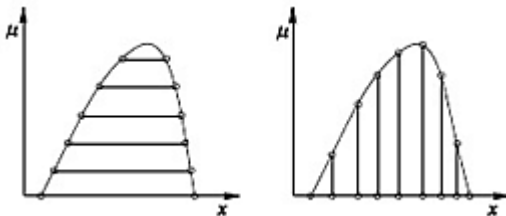


Fig. II.2.3_6 Un număr fuzzy. În Fig. II.2.2_4 am reprezentat un număr fuzzy trapezoidal

- Valoarea singulară ('singleton')

E o mulțime fuzzy reprezentată de o singură valoare. Exemplu: presupunem X e universul intensităților gri și avem 3 mulțimi fuzzy care acoperă X : "închis" (la culoare), "deschis" și gri. Fiecare mulțime poate fi reprezentată de către o valoare singulară: "închis" de o valoare apropiată de negru (absolut cu intensitatea 0), "deschis" de o valoare apropiată de alb (alb absolut cu intensitatea 255 în cadrul unei reprezentări a intensității în $[0,255]$) și "gri" de o valoare "la mijloc" (128).

Operații cu mulțimi fuzzy

- Intersecția

E analogă cu operația ȘI logic, luându-se minimumul: $A(x) \cap B(x) = \sum_x \mu_A(x) \wedge \mu_B(x)$

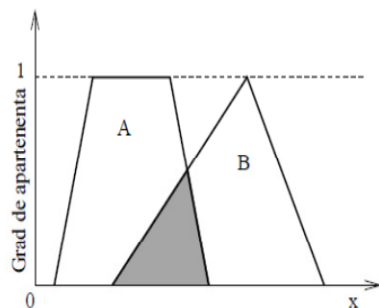


Fig. II.2.3_7 Intersecția a două mulțimi fuzzy

Observații:

- Dacă în cazul seturilor ‘crisp’ întrebarea pentru intersecția a două seturi e: “Ce elemente aparțin ambelor seturi?”, în cazul seturilor fuzzy întrebarea e: “Ce elemente aparțin într-un anumit grad ambelor seturi?” (în cazul seturilor fuzzy, un element poate aparține mai multor seturi în grade diferite)

- Intersecția se poate defini prin orice normă triunghiulară (T). A și B convexe $\rightarrow A \cap B$ convexă

Reuniunea

E analogă cu operația SAU logic, luându-se maximumul:

$$A(x) \cup B(x) = \sum_x \mu_A(x) \vee \mu_B(x)$$

Observații:

1) Reuniunea se poate defini prin orice conormă T.

2) Operațiile fuzzy de reuniune și intersecție se pot generaliza la o structură de latică.

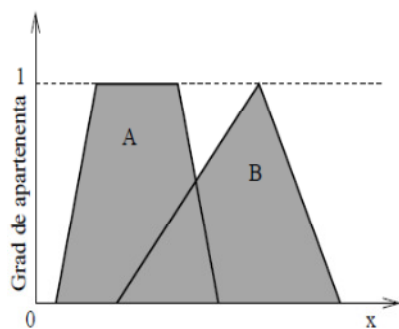


Fig. II.2.3_8 Reuniunea a două mulțimi fuzzy

- Complementul

E analoag cu operația NU logic:

$$\bar{A}(x) = \sum_x 1 - \mu_A(x)$$

- Combinația convexă

E un operator liniar care combină mai multe mulțimi fuzzy într-una singură, folosind ponderi asociate fiecărei mulțimi. Conform definiției combinației convexe, aceste ponderi sunt pozitive și au suma 1.

Generalizare conform [Zadeh, 1965]: Să luăm cazul combinației convexe a două mulțimi A și B. Putem combina convex aceste mulțimi prin intermediul unei a treia mulțimi Λ :

$$\{A, B; \Lambda\} = \Lambda A + C_{\Lambda} B, \quad C_{\Lambda} \text{ fiind complementul lui } \Lambda$$

În termeni de f.a. avem:

$$\mu_{(A,B;\Lambda)}(x) = \mu_{\Lambda}(x)\mu_A(x) + [1 - \mu_{\Lambda}(x)]\mu_B(x), \quad \forall x \in X \quad (*)$$

O proprietate de bază a unei combinații convexe spune că:

$$A \cap B \subset \{A, B; \Lambda\} \subset A \cup B, \quad \forall \Lambda$$

Această proprietate e o consecință imediată a inegalităților:

$$\begin{aligned} \min[\mu_A(x), \mu_B(x)] &\leq \lambda \mu_A(x) + (1 - \lambda)\mu_B(x) \\ &\leq \max[\mu_A(x), \mu_B(x)], \quad \forall \lambda \in [0,1] \end{aligned}$$

E de remarcat că $\forall C$ un set fuzzy cu $A \cap B \subset C \subset A \cup B$,

C se poate scrie ca o combinație convexă de A și B, luând $\mu_{\Lambda}(x) = \frac{\mu_C(x) - \mu_B(x)}{\mu_A(x) - \mu_B(x)}$ în relația (*).

- Concentrarea fuzzy

E o reducere a gradului de apartenență, prin ridicare la o putere $n > 1$ a valorilor f.a.. Dual, prin ridicare la o putere $1/n$ a valorilor f.a putem obține o 6

are (vezi și calificatorii fuzzy menționați mai jos).

După [Zadeh, 1965], dăm câteva teoreme legate de mulțimi fuzzy convexe pentru că, **în general, clasificarea se face în clase / mulțimi convexe** (vezi, pentru exemplificare și secțiunea “Definirea formei spațiului unei probleme de recunoaștere nedeterminate și modelarea acestuia. Cazul recunoașterii pielii”):

Nucleul unui set fuzzy convex e convex – putem aproxima clasa la nucleul ei.

Fie A un set fuzzy cu suport mărginit și fie $M_A = \sup_x \mu_A(x)$. M_A se numește *gradul maximal* în A . Atunci $\exists x_0$ în care M_A e *atins esențial*, în sensul că, $\forall \varepsilon > 0$, orice vecinătate sferică a lui x_0 conține puncte în setul $Q(\varepsilon) = \{x/\mu_A(x) \geq M - \varepsilon\}$; în cazul 1-D, dacă setul A e puternic convex, atunci punctul x_0 e unic.

Teorema de separare pentru seturi fuzzy convexe [Zadeh, 1965]: Fie A și B seturi fuzzy n-D, cu suport mărginit și fie M_A și M_B gradele lor maxime, M fiind gradul maximal al setului $A \cap B$. Atunci gradul maxim de separare între A și B e $D = 1 - M$.

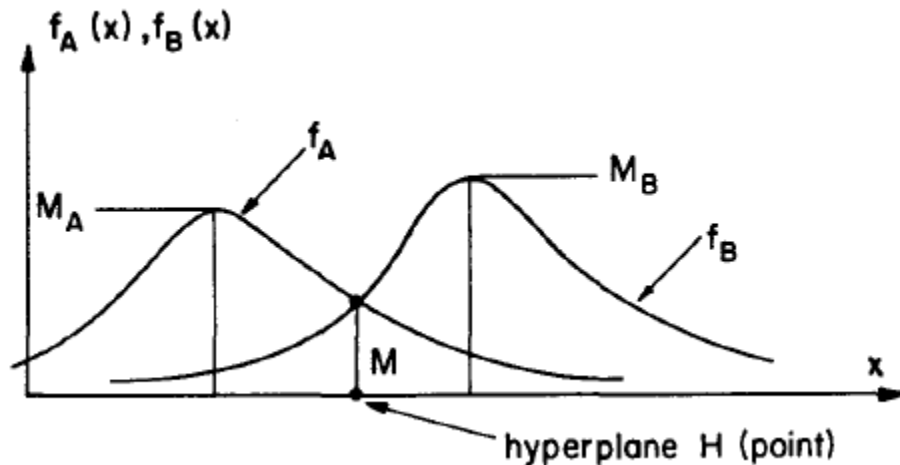


Fig. II.2.3_9 Un grad mai mare de separare între A și B nu se poate realiza. Exemplificare pentru cazul 1-D [Zadeh, 1965]

Această teoremă are o importanță specială: se poate demonstra astfel utopia unor probleme de clasificare – clasificare – vom exemplifica în finalul secțiunii IV.3.5.

Dacă f.a. pentru un element x la un concept / clasă / descriptor lingvistic C e $\mu_C(x) = 0,5$ (punctul 'crossover'), înseamnă că nu suntem deloc siguri dacă elementul aparține sau nu conceptului: fuzitatea e la maxim. Legat de acest punct se dau definițiile:

Def.: Două submulțimi ale unei mulțimi fuzzy se numesc slab separate dacă $\mu_{A \cap B} < 0,5$

Observație: Mai sus am definit partiția fuzzy probabilistică. Definiția mulțimilor fuzzy slab separate se poate folosi, conform [Shakhatreh & Hayajneh 2010], la definirea unei **partiții fuzzy** mai generale: se impune submulțimilor care partiționează mulțimea să fie slab separate și să acopere această mulțime.

Două exemple de aplicații concrete ale logicii fuzzy:

1. Detecția pixelilor de contur

Datorită capacității abordărilor fuzzy de a reprezenta informațiile imprecise sau neclare, acestea sunt mai puțin sensibile la alegerea parametrilor, cum ar fi valoarea de prag pentru un detector de contur. Putem stabili reguli de tip 'If-Then' pentru a atribui fiecărui pixel o valoare reprezentând gradul de apartenență a acestuia la mulțimea pixelilor de contur, formând astfel o mulțime a potențialilor pixeli de contur. Funcțiile de apartenență (f.a.) caracterizează diferențele dintre nivelul de gri al pixelului-argument și nivelele de gri ale pixelilor vecini.

Pentru aceste diferențe (în valoare absolută) se definesc două mulțimi fuzzy cărora le pot aparține: mulțimea fuzzy a diferențelor "mici" și cea a diferențelor "mari". F.a., alese liniar pentru simplitate, μ_{mic} și μ_{mare} sunt definite complementare:

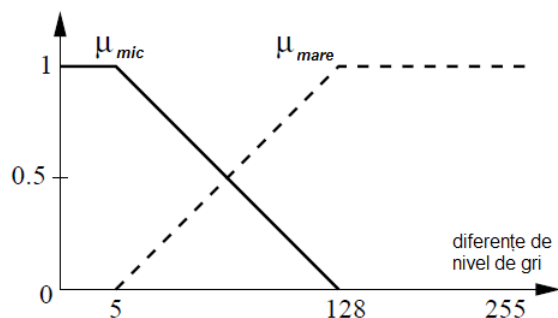


Fig. II.2.3_10 Funcțiile de apartenență ale diferențelor de nivele de gri.

Pragul inferior, 5, e ales diferit de zero pentru a evita o prea mare sensibilitate la zgomot. Pentru aplicarea regulilor fuzzy ‘If-Then’ se folosesc 8 măști pătrate de dimensiune 3x3, astfel:

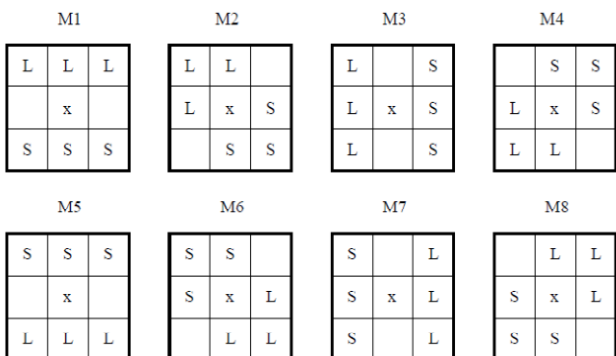


Fig. II.2.3_11 Măști prin care cercetăm dacă pixelul din mijloc e pixel de contur

Calculăm diferențele de intensitate (în modul) dintre pixelul central și pixelii notați cu S, DS, și L, DL. Regula fuzzy e:

DACĂ toate diferențele DS sunt mici ȘI toate diferențele DL sunt mari

ATUNCI pixelul central al măștii e considerat un posibil pixel de contur (‘Potential Edge Pixel’).

Pentru fiecare mască M_i se calculează gradul de apartenență a pixelului

central la mulțimea PEP: $\mu_{PEP}^{M_i} = \wedge [toate \mu_{mic}(D_S), toate \mu_{mare}(D_L)]$

După ce au fost calculate cele 8 valori reprezentând gradele de apartenență la mulțimea fuzzy PEP, pentru fiecare mască în parte, se calculează pentru pixelul central gradul final de apartenență la mulțimea fuzzy PEP: $\mu_{PEP} = \vee_i (\mu_{PEP}^{M_i})$

Pentru fiecare pixel din imagine se calculează gradul μ_{PEP} de apartenență la mulțimea PEP și, pentru o paletă a nivelelor de gri reprezentată pe 8 biți, i se atribuie o valoare de nivel de gri = $255 \times \mu_{PEP}$: această reprezentare poate fi privită ca valoare în modul unui operator de tip gradient; direcția acestui gradient e dată de masca ce corespunde valorii maxime dintre cele 8 valori $\mu_{PEP}^{M_i}$.

2. Algoritmul de clustering c-means fuzzy

În practică există multe cazuri în care partițiile nu sunt (complet) disjuncte, vectorii putând fi clasificați ca aparținând mai multor partiții. Conform wikipedia: “In hard clustering, data is

divided into distinct clusters, where each data element belongs to exactly one cluster. In fuzzy clustering (also referred to as soft clustering), data elements can belong to more than one cluster, and associated with each element is a set of membership levels. These indicate the strength of the association between that data element and a particular cluster. Fuzzy clustering is a process of assigning these membership levels, and then using them to assign data elements to one or more clusters”. Gradele de apartenență ale unui element la un cluster pot fi privite ca grad de compatibilitate al elementului la cluster.

Observație: Putem pleca de la varianta clasică a algoritmului c-means – vezi spre sfârșitul capitolului 1 – pentru a construi grade de compatibilitate / apartenență. Având centrele și varianțele valorilor din fiecare partiție, putem genera funcții de apartenență pentru clasele / partițiile respective. Obținem o împărțire a spațiului, pe baza căreia putem construi f.a. pe care să le folosim apoi pentru o anumită aplicație ce folosește logica fuzzy.

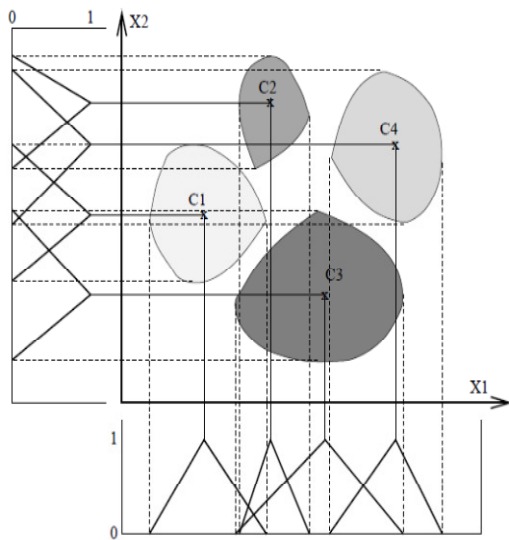


Fig. II.2.3_11 Construcția f.a. pe baza varianței și centrelor partițiilor pentru cazul unor vectori bidimensionali.

Algoritmul fuzzy c-means (FCM, Bezdek 1981) e cel mai cunoscut și mai utilizat algoritm fuzzy de clustering. FCM încearcă partiționarea unei colecții finite $X = \{x_1, \dots, x_n\}$. Algoritmul întoarce o listă de c clustere $C = \{c_1, \dots, c_c\}$ sub forma centrelor v_i ale acestor clustere și a unei matrice

$[u_{ik}]$ de dimensiune $c \times n$, u_{ik} fiind gradul de apartenență a vectorului x_k la partiția i . $u_{ik} \in [0,1]$, cu

$$\sum_{i=1}^c u_{ik} = 1, \quad k = 1 \dots n$$

(un element poate “ține de” mai multe clustere, dar suma gradelor de apartenență e limitată la 1).

Ca și algoritmul c-means clasic se încearcă iterativ minimizarea unei funcții-criteriu -

$$J(U, V) = \sum_{i=1}^c \sum_{k=1}^n u_{ik}^m (x_k - v_i)^2$$

, m fiind un factor de ponderare de tip exponent. Cu cât m e mai mare, cu atât se reduce influența vectorilor cu grade de apartenență mici, deci fuzitatea scade.

$J(U; V)$ reprezintă suma pătratelor distanțelor euclidiene dintre fiecare vector de intrare și fiecare centru de partiție, ponderate de gradele de apartenență.

Algoritmul fuzzy c-means:

Se inițializează U (pas = 1) cu valori aleatorii, sau, dacă se poate, pe baza unei aproximări.

Se ia următoarea formă de intrare x_{pas} și se (re)calculează centroizii partițiilor pe baza lui U (pas) - formula de calcul e asemănătoare cu cea a centrului de greutate:

$$v_i = \frac{1}{\sum_{k=1}^n u_{ik}^m} \sum_{k=1}^n u_{ik}^m x_{ik}, \quad i = 1, 2, \dots, c$$

Se calculează U (pas+1) conform v_i .

Se oprește iterația dacă:

$$\max |u_{ik}^{(\alpha)} - u_{ik}^{(\alpha-1)}| \leq \varepsilon$$

, altfel se trece la iterația următoare, pas++, și se reia algoritmul de la pasul 2. ε e un prag minim de convergență a gradelor $[u_{ik}]$.

Se trasează astfel o *hartă fuzzy a datelor*.

II.2.3.3. Calificatori fuzzy ('hedges')

Conform [fz ext] Zadeh și lingvistul George Lakoff au lucrat la Universitatea din Berkeley pentru a descrie delimitatorii de tip 'hedge' (se poate traduce cu "gard viu"). 'Hedges' se folosesc pentru a modifica seturile fuzzy – iată câteva categorii ale lor:

- modificatori generali: foarte, extrem de
- valori de adevăr: mai mult fals, nu chiar adevărat
- probabilități: puțin probabil, asemănător
- cuantificatori: majoritatea, câțiva, puțini
- posibilități: aproape imposibil, foarte probabil / posibil

Unii calificatori dilată un set fuzzy – e.g.: comparativul "mai." Alții, ca "foarte" concentrează setul. Hedges sunt vagi pentru că nu au o definiție precisă, dar reflectă gândurile omului. Mai multe experimente au confirmat faptul că oamenii "ordonează" aceste cuvinte la fel.

Dacă setului fuzzy îi corespunde o variabilă lingvistică, calificatorului fuzzy îi corespunde funcția de adverb pe lângă un adjectiv.

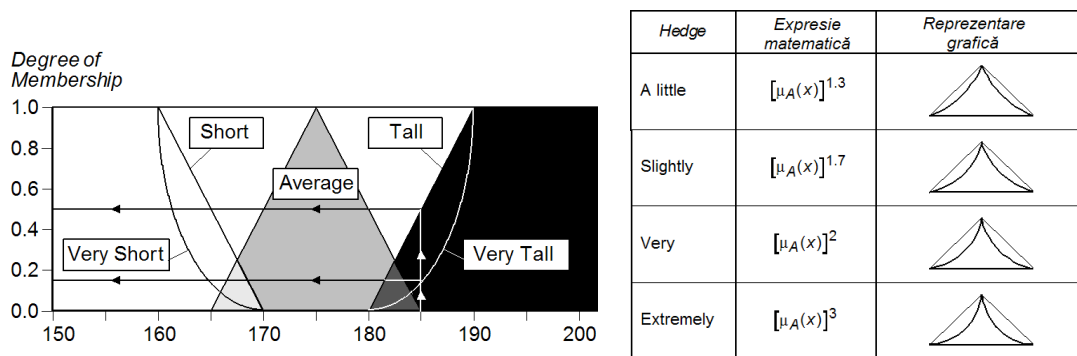


Fig. II.2.3_12 Calificatorii fuzzy modifică f.a. a unui set fuzzy [fl ppt]

II.2.3.4. Reguli fuzzy

Chiar dacă nu credem în computaționalism, acea teorie conform căreia mintea omenească poate fi văzută ca un sistem de procesare a informației, gândurile fiind “deductibile” prin calcul, ne propunem să putem exploata informația incompletă. Informația care poate fi descrisă printr-un concept / clasă sau alt descriptor lingvistic. În aplicațiile teoriei fuzzy, informația dată prin descriptori lingvistici, informația incompletă, imprecisă sau calitativă, e.g., informații de expertiză sau reguli / cerințe de proiectare, sisteme “laxe”, e procesată conform unor reguli de inferență ale sistemului expert / fuzzy. Aceste reguli de inferență sunt de forma: DACĂ x_1 ȘI x_2 ATUNCI y , e.g., DACĂ (clientul vrea foarte repede dezvoltarea modulului de administrare) ȘI (echipa de dezvoltare e mică) ATUNCI (urmează planul b). Remarcăm în cazul acestui sistem expert:

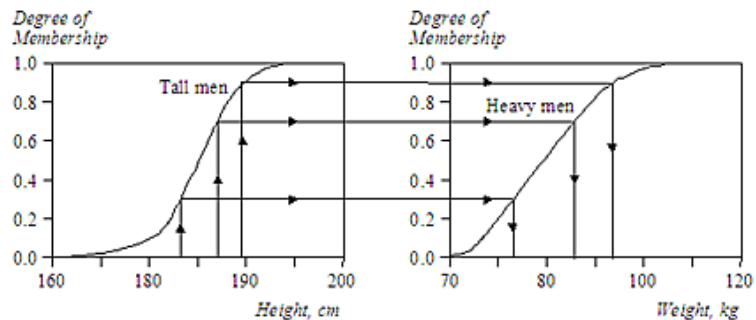
- Folosirea variabilelor lingvistice, atât ca intrări cât și ca ieșiri, variabilelor care încorporează cunoaștere imprecisă; acestor variabile le corespund seturi de valori și nu valori unic determinate (ca în cazul cunoașterii precise / complete)
- Conexiuni logice clasice ale variabilelor lingvistice: ȘI, SAU ș.a.m.d.

Variabilele lingvistice, sau, mai general spus, cuvintele, nu sunt numere. Când spunem “mașină” subînțelegem un SET de mașini. Mai mult, cuvintele se “suprapun” chiar dacă sunt antonime – între “intervalul” corespunzător lui “scund” și cel corespunzător lui “înalt” poate exista o mică zonă comună cu o, teoretic, infinitate de valori). După V. Negoită, “paradigma fuzzy reprezintă o distanțare radicală de atitudinea științifică occidentală postrenascentistă, bazată pe legea terțului exclus, deși ea e intens folosită ca practică poetică (și teologică)” – se folosesc metafore “pentru a comunica imagini care ar trebui să facă manifestă unitatea unei lumi absolute; prin metafore se comunică ceea ce altfel nu s-ar putea comunica, nefăcând parte din experiența de rutină”. Negoită [Negoită, 2006] mai spune că, pe lângă calitatea sa revelatorie, când nu e “doar” o “metaforă plasticizantă” [Blaga, 1937], “metafora ține de capacitatea de rezumare, o proprietate

fascinantă a omului, obiect de studiu în cercetarile privind simularea inteligenței cu ajutorul calculatorului”

După ce în 1965 Lotfi Zadeh publicase celebra lucrare ce introducea în limbajul științific mulțimile fuzzy, în 1973, el va publica a doua lucrare de mare influență. De data aceasta se pune în lumină o metodologie nouă de a trata sistemele complexe, prin captarea cunoștințelor umane în reguli ('rules') fuzzy. Într-un sistem fuzzy regulile (fuzzy) se îndeplinesc într-un anumit grad. Dacă premiza e adevărată într-un anumit grad, atunci consecința e de asemenea adevărată într-un grad corespunzător.

Fig. Fig. II.2.3_14 Exemplu de regulă fuzzy: DACĂ înalt ATUNCI greu.



În figura de mai sus am

exemplificat o 'Monotonic selection': gradul de adevăr al consecinței regulii fuzzy poate fi calculat direct din gradul de adevăr al premizei (mai general, numim *logică monotonică* o logică conform căreia creșterea numărului de premize atrage creșterea numărului de consecințe. Se mai spune că regulile fuzzy se aplică într-un anumit grad: dacă premiza e adevărată într-un grad μ , consecința e adevărată în același grad μ .

O regulă fuzzy poate avea multiple premize conectate logic (DACĂ x_1 SAU x_2 ATUNCI y_1). În ce privește cazul consecințelor multiple, e.g., DACĂ x_1 ATUNCI y_1, y_2 , aceasta corespunde unei conexiuni ȘI între consecințe – pentru a nu confunda semnificația acestui ȘI cu ȘI-ul ce poate conecta premisele, folosim virgula între consecințe. Mai mult, o regulă cu m consecințe se poate traduce în m reguli cu o consecință, e.g., regula de mai sus devine: regula DACĂ x_1 ATUNCI y_1 și regula DACĂ x_1 ATUNCI y_2 . Deci, fără a pierde din generalitate, putem considera regula fuzzy ca având o singură consecință.

Principiul extensiei

Fie funcția $f : X \rightarrow Y$ și un set fuzzy

$$A = \sum_i \mu_A(x_i) / x_i$$

Principiul extensiei spune că imaginea lui A prin f poate fi exprimată ca un set fuzzy

$$B = f(A) = \sum_i \mu_A(x_i) / y_i, \text{ unde } y_i = f(x_i).$$

Dacă $f(\cdot)$ nu e injectivă, atunci există $x_1, x_2 \in X, x_1 \neq x_2$ cu $f(x_1) = f(x_2) = y^*, y^* \in Y$.

Gradul de apartenență pentru $y=y^*$ va fi maximul gradelor de apartenență în x_1 și x_2 .

$$\text{Mai general, } \mu_B(y) = \bigvee_{x=f^{-1}(y)} \mu_A(x)$$

Exemplu pentru $y = f(x) = x^2 - 6x + 11$. Intrare: număr fuzzy number – în jurul lui 4:

$$n4 = 0.3/2 + 0.6/3 + 1/4 + 0.6/5 + 0.3/6 \rightarrow$$

$$y = 0.3/f(2) + 0.6/f(3) + 1/f(4) + 0.6/f(5) + 0.3/f(6)$$

$$y = 0.6/2 + (0.3 \vee 1)/3 + 0.6/6 + 0.3/1$$

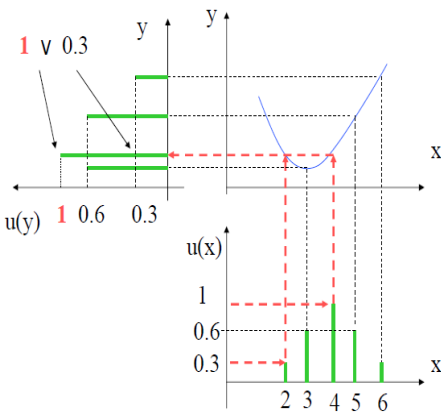


Fig. II.2.3_14 Principiul extensiei cu o funcție care nu e injectivă.

Putem defini operații aritmetice fuzzy. Fie A și B seturi fuzzy corespunzătoare operanzilor și x și y, “parametri” acestor seturi. Conform principiului extensiei, putem defini:

$$\mu_{A+B} = \bigvee_{x+y=z} [\mu_A(x) \wedge \mu_B(y)]$$

II.2.2.5. Sisteme expert / de inferență de tip fuzzy

În continuare, redăm logica tipică a unui asemenea sistem conform [fs ppt].

Presupunem o problemă cu două intrări, *project_funding* (x) și *project_staffing* (y) și trei reguli – în dreapta avem codificarea regulilor din stânga folosind seturi fuzzy:

IF <i>project_funding</i> is adequate (A3) OR <i>project_staffing</i> is small (B1) THEN risk is low (C1)	IF x is A3 OR y is B1 THEN z is C1
IF <i>project_funding</i> is marginal (A2) AND <i>project_staffing</i> is large (B2) THEN risk is normal (C2)	IF x is A2 AND y is B2 THEN z is C2
IF <i>project_funding</i> is inadequate (A1) THEN risk is high (C3)	IF x is A1 THEN z is C3

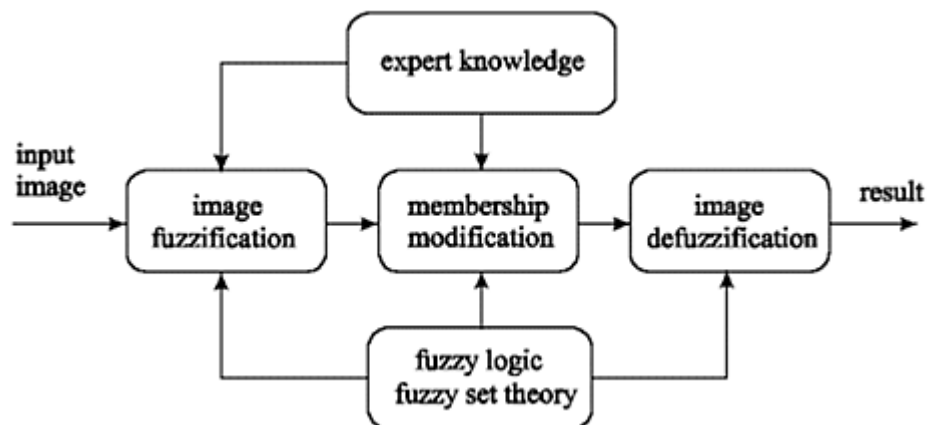


Fig. II.2.3_15
Schema generală a unui sistem expert de tip fuzzy pentru procesarea imaginilor [fl ppt]

Schema de inferență Mamdani e o secvență de 4 pași:

- 1) fuzificarea intrărilor
- 2) evaluarea regulilor
- 3) agregarea rezultatelor regulilor
- 4) defuzificarea

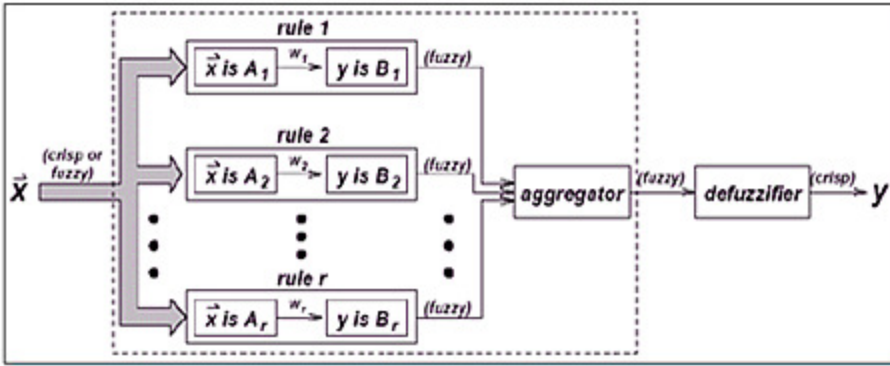


Fig. II.2.3_16 Schema de inferență Mamdani a unui sistem expert de tip fuzzy [fl ppt]

În continuare redăm logica tipică a unui asemenea sistem conform folosind un exemplu [fl ppt].

Primul pas constă în stabilirea seturilor fuzzy A_i și B_i , preluarea valorile crisp, x_1 și y_1 – ‘project funding’ și ‘project staffing’: valorile lingvistice ale variabilelor lingvistice corespunzătoare seturilor fuzzy A_i și B_i – , și determinarea apartenenței acestora la seturile fuzzy:

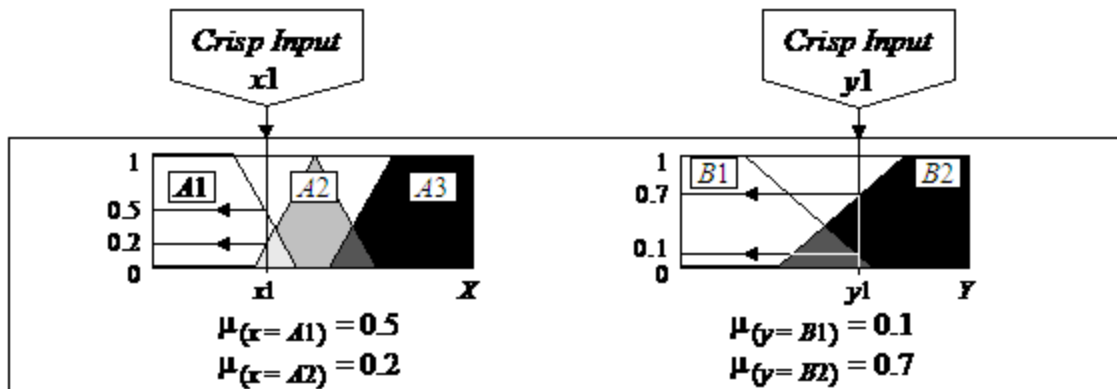


Fig. II.2.3_17 Primul pas conform schemei de inferență Mamdani

În al doilea pas se preiau intrările fuzzificate, $\mu(x=A_1) = 0.5$, $\mu(x=A_2) = 0.2$, $\mu(y=B_1) = 0.1$ și $\mu(y=B_2) = 0.7$, și se aplică premizelor regulilor. Operatorii fuzzy (AND sau OR) se folosesc pentru a obține un singur număr ca rezultat al evaluării tuturor premizelor dintr-o regulă. Acest nr., care reprezintă gradul de adevăr al premizei (premizelor) regulei fuzzy e “aplicat” f.a. a setului corespunzător consecinței regulei:

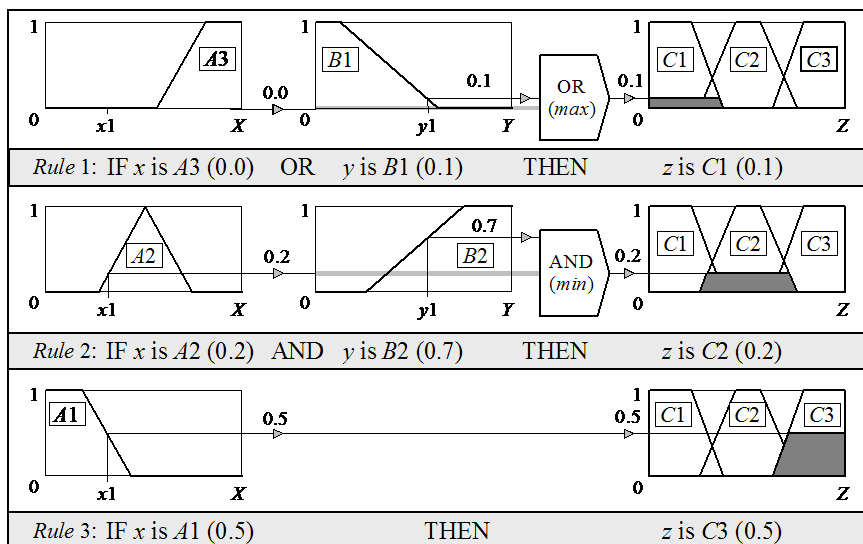


Fig. II.2.3_18 Evaluarea regulilor conform schemei de inferență Mamdani [fl ppt]

Cea mai populară metodă de a corela gradul de adevăr al premizei cu gradul de adevăr al consecinței e limitarea ('clipping') f.a. a consecinței la gradul de adevăr al premizei – vezi figura de mai sus. Această metodă e foarte rapidă (generând și o suprafață care e ușor de defuzificat), dar "pierde" din forma f.a. a consecinței. O metodă "mai exactă", care păstrează forma f.a. a consecinței, e scalarea f.a. a consecinței cu un factor = gradul de adevăr al premizei:

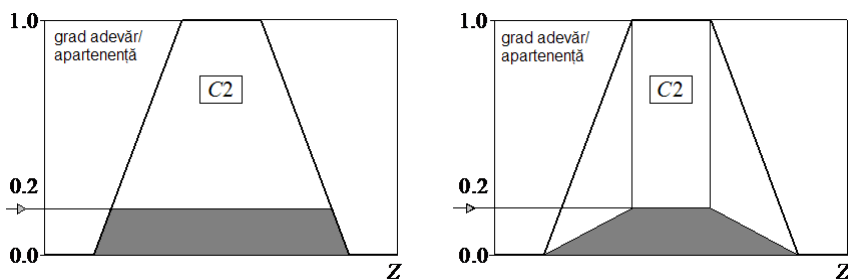


Fig. II.2.3_19 Corelarea premiză – consecință: limitare vs. scalare [fl ppt]

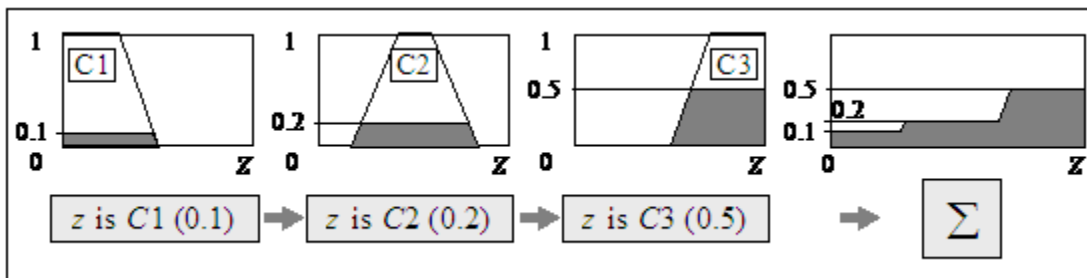


Fig. II.2.3_20 Pasul 3: agregarea gradelor de adevăr ale consecinței (metoda limitării) [fl ppt]

În pasul 4, ultimul, defuzificăm Σ , dacă ne-am propus să obținem ieșirea sub forma unui număr ('crisp'). Cea mai populară metodă de a face aceasta e aceea a centrului de greutate: stabilim abscisa care împarte setul agregat în două subseturi de arii egale:

$$CG = \frac{(0 + 10 + 20) \times 0.1 + (30 + 40 + 50 + 60) \times 0.2 + (70 + 80 + 90 + 100) \times 0.5}{0.1 + 0.1 + 0.1 + 0.2 + 0.2 + 0.2 + 0.2 + 0.5 + 0.5 + 0.5 + 0.5} = 67.4$$

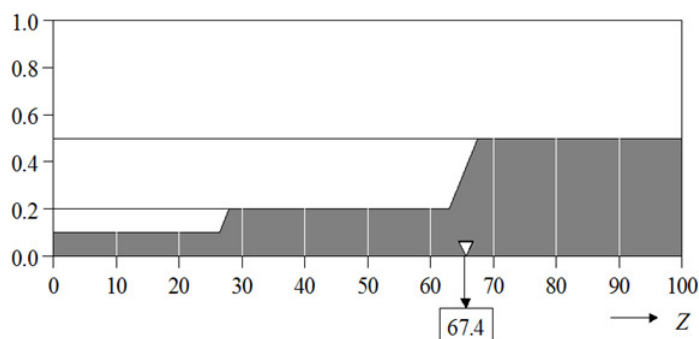


Fig. II.2.3_21. Defuzificarea prin metoda centrului de greutate [fl ppt]

Așa cum am văzut, inferență de tip Mamdani reclamă găsirea CG (sau ceva asemănător) unei forme 2-D, ceea ce impune o integrare “de-a lungul” suportului setului fuzzy rezultat, ceea ce e ineficient din punct de vedere computațional. Michio Sugeno propune un tip de inferență în care consecințele regulilor sunt reprezentate de valori singulare (‘singleton’). Formatul unei reguli de inferență de tip Sugeno e:

IF (premiza x) THEN z is f(x), f fiind o funcție matematică

Observație: Am văzut că, fără a pierde din generalitate, ne putem referi nu numai la consecințe, dar și la premise singulare. Mai mult, în practică se folosește des ‘zero-order Sugeno fuzzy model’ – funcția f devine o constantă, regula fiind:

IF (premiza x) THEN z = k.

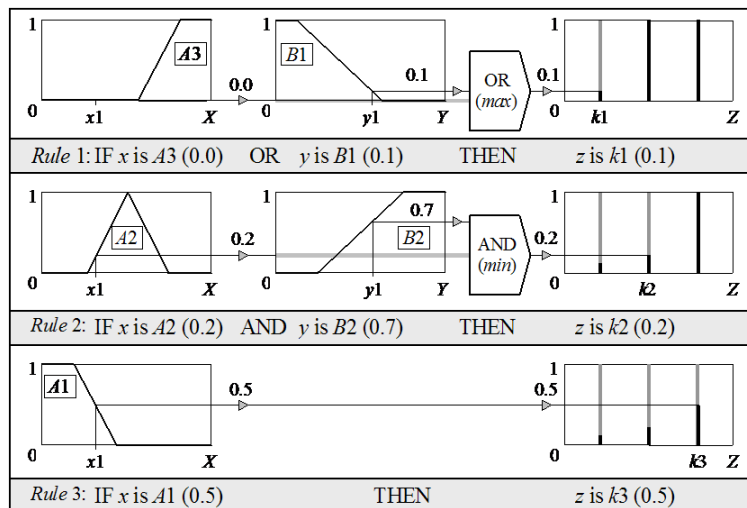


Fig. II.2.3_22 Evaluarea regulilor conform schemei de inferență Sugeno [fl ppt]

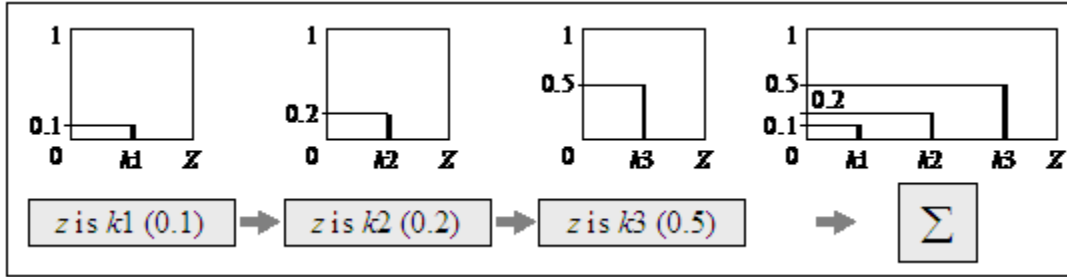


Fig.II.2.2_22 agregarea gradelor de adevăr ale conecinței [fl ppt]

Astfel, defuzificarea se obține prin calculul unei medii ponderate [fl ppt]:

$$\frac{\mu(k1) \times k1 + \mu(k2) \times k2 + \mu(k3) \times k3}{\mu(k1) + \mu(k2) + \mu(k3)} = \frac{0.1 \times 20 + 0.2 \times 50 + 0.5 \times 80}{0.1 + 0.2 + 0.5} = 65$$

Cam același rezultat ‘crisp (65 față de 67.4) se calculează cu o complexitate de calcul $O(1)$, față de $O(n)$, cât era în cazul schemei de inferență Mamdani.

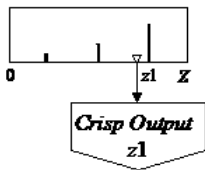


Fig. II.2.3_25 Defuzificarea în cazul schemei de inferență Sugeno [fl ppt]

Dacă metoda Mamdani e larg acceptată, întrucât permite captarea mai bună și într-o manieră mai intuitivă, a cunoștințelor furnizate de o expertiză, în schimb angajează mai mult efort de calcul. Metoda Sugeno poate fi privită ca o optimizare a metodei, ceea ce o face foarte atractivă în problemele de control, în particular pentru sistemele dinamice neliniare.

Pentru calibrarea(‘tunning’) sistemelor fuzzy putem aplica:

- Redefinirea seturilor fuzzy corespunzătoare intrărilor și ieșirilor.

Adăugarea de seturi fuzzy – folosirea seturilor “largi” atrage o comportare “grosieră” a sistemului.

Acoperirea suficientă între seturile fuzzy. În cazul învecinării seturilor triunghi / trapez - triunghi / trapez se recomandă o suprapunere între 25% și 50% a suporturilor.

Examinarea bazei de reguli în ce privește oportunitatea de a scrie reguli care să folosească calificatori fuzzy ('hedge rules') pentru a capta mai bine comportamentul sistemului.

Ordonarea după importanță regulilor prin aplicarea unor ponderi asupra lor.

Observație: Descrierea sistemului expert de mai sus se poate "mula" pe descrierea algoritmului dat mai sus pentru "Detectia pixelilor de contur", de la rubrica "Două exemple de aplicații concretă ale logicii fuzzy"

Putem formaliza și generaliza comportamentul sistemului expert descris mai sus pentru următorul sistem expert:

Avem M reguli R_i , N intrări z_i și o ieșire v . Regulele se pot scrie:

R1: IF $(z_1, A11)$ ȘI/SAU $(z_2, A12)$ ȘI/SAU... $(z_N, A1N)$ THEN $(v, B1)$...

RM: IF $(z_1, AM1)$ ȘI/SAU $(z_2, AM2)$ ȘI/SAU... (z_N, AMN) THEN (v, BM)

[ELSE (v, BD)]

unde $[A_{ij}]_{M \times N}$ și BD (de la 'default') sunt seturi fuzzy.

Evaluând premisele pentru R_i , obținem scalarul λ_i : gradul de aplicare 'strength level' pentru regula i . Ramura ELSE, în caz că există, corespunde unui grad de aplicare $\lambda_D = \bigwedge_{i=1...M} (1 - \lambda_i)$: nu se aplică nici regula 1 ȘI nici regula 2 ȘI nici regula M .

II.2.4. "Fuzzy logic = calculul cu cuvinte" [Zadeh, 1996]

II.2.4.1. Termenul lexical și abilitatea de a opera cu informații bazate pe percepții sau intuiție

"Atâta timp cât legile matematicii descriu realitatea, ele nu sunt exacte; dacă sunt exacte, atunci nu descriu fenomene reale" (Albert Einstein). Werner Heisenberg: "Orice noțiune sau concept,

oricât de clar ar fi, are numai un domeniu limitat de aplicabilitate”. Fizicienii, și numai ei, știu că metodele lor de analiză și raționamentul logic sunt insuficiente pentru a cuprinde fenomenele naturale în toată complexitatea lor și, de aceea, ei izolează un grup de fenomene și caută să construiască un model care să descrie acest grup. Astăzi știm că modelul mecanicii clasice (newtoniene) e valabil numai pentru obiecte constituite dintr-un număr foarte mare de atomi și pentru viteze de deplasare mici în comparație cu viteza luminii. Dacă prima condiție nu e îndeplinită, mecanica clasică trebuie înlocuită cu cea cuantică; dacă cea de-a doua condiție nu e îndeplinită, mecanica clasică trebuie înlocuită cu teoria relativității.

Alfred Korzybski a remarcat că “harta nu e totuna cu teritoriul”. Mai general, e vorba de relația dintre un obiect și reprezentarea lui, reprezentare care abstractizează obiectul dintr-o anumită perspectivă, oferind un fel de interfață a obiectului. Korzybski a prezentat această expresie la începutul anilor '30, într-o lucrare intitulată “A Non-Aristotelian System and its Necessity for Rigour in Mathematics and Physics” [Korzybski, 1933]. Făcând apel la analogia lui Korzybski dintre hartă și teritoriu, am putea spune, datorită impreciziei și ambiguității sale intrinseci, că limbajul comun e o hartă destul de flexibilă ca să poată urmări suprafața “curbă” a teritoriului. Cu cât limbajul devine mai riguros, cu atât gradul de “cuprindere” al limbajului scade. Astfel apare necesitatea de a alătura modelelor matematice interpretări verbale, făcând apel la concepte înțelese intuitiv, mai mult sau mai puțin ambigue.

Logica fuzzy avea să fie “luată în seamă” oficial de comunitatea științei abia în anii '60, odată cu lucrările profesorului Zadeh; peste încă alte decenii, în urma succeselor tehnologice va fi încadrată fără contestații majore în patrimoniul științific internațional. Printre succesele de anvergură ale teoriei fuzzy se numără valorificarea raționamentului aproximativ și a limbajului, ca procese creative [Teodorescu, 2011]. Renunțarea la logica binară a dat posibilitatea gânditorilor să exploateze fuzitatea sistemelor lingvistice, făcând posibile dezvoltări extraordinare în inginerie.

În [Zadeh, 2002], se pleacă de la ideea că, pentru a **înzestra conceptul de probabilitate cu abilitatea de a opera cu informații bazate pe percepții, trebuie să generalizăm însuși conceptul de probabilitate**. Generalizarea se poate face la trei niveluri:

1) PT+ (“Probability theory”) corespunde fuzificării (f) – la nivelul PT+ seturile ‘crisp’ se generalizează la seturi fuzzy, prin introducerea conceptului de apartenență într-un anumit grad

2) PT++ corespunde fuzificării granulate (fg) – prin introducerea conceptului de granulă (vezi figura de mai sus)

3) PTP corespunde adăugării nivelului PT++ a capacității de a descrie percepțiile prin propoziții într-un limbaj natural (NL = Natural Language). ! NL nu e orice limbaj natural – se impune condiția ca înțelesul unei propoziții exprimate în NL să poată fi reprezentat sub forma unei constrângeri generalizate (GC) a unei variabile. PTP evită excesul preciziei care poate însemna supra-idealizare și lipsă a robusteții.

Primul pas al “raționamentului probabilistic pe baza percepțiilor” e traducerea descrierii percepțiilor în Generalized Constraint Language (GCL) folosind semanticile centrate pe constrângeri ale limbajului natural (CSNL). În pasul al doilea, conform inferenței date de regulile fuzzy, constrângerile generalizate (GC) se propagă de la nivelul premizelor la cel al consecințelor – regulile fuzzy coincid cu regulile de propagare în GCL. Principala regulă de inferență e aceea a generalizării principiului extensiei – în general, prin folosirea acestui principiu, problema găsirii “probabilităților” dorite devine o problemă de calcul variațional cu constrângeri impuse. Am pus între ghilimele “probabilităților”, chiar dacă acest termen e folosit chiar în textul original, pentru a semnala că nu e vorba de sensul clasic al cuvântului, ci de sensul fuzzy, fiind vorba de un grad de apartenență. Prin semanticile centrate pe constrângeri ale GCL (CSGCL) și regulile care guvernează propagarea GC, o constrângere compusă în GCL se poate reprezenta ca o singură constrângere generalizată. GCL e expresiv maximal în sensul că se pot traduce în GCL orice constrângeri pe care le putem concepe. O propoziție dintr-un limbaj natural (NL), se zice că e admisibilă dacă se poate traduce în GCL. Conceptul de bază al GCL e constrângerea generalizată. În forma canonică, aceasta se scrie $X \text{ isr } R$, ceea ce formal se scrie $X \text{ isr } R$, X fiind variabila constrânsă și R relația de constrângere. ‘isr’ corespunde lui “este r”. Explicităm semnificația lui r pentru principalele tipuri de constrângeri:

- constrângerea de probabilitate: $r = p$, X fiind variabilă aleatoare, R fiind distribuția sa de probabilitate

- constrângerea de posibilitate: $r = (gol)$, R fiind distribuția de posibilitate a lui X

- constrângerea de adevăr: $r = v$, R fiind distribuția de adevăr a lui X

- constrângerea de reprezentare fuzzy (‘fuzzy graph’): $r = fg$, X fiind variabilă care ia valori într-un set fuzzy, R fiind distribuția de posibilitate a valorilor lui X.

În practică, există foarte multe percepții care pot fi prinse în concepte care precizează un NL, e.g., valoarea uzuală a unei variabile aleatoare, netezimea unei funcții, relevanța în căutarea într-o bază de date. În multe cazuri, în contextul analizei deciziilor, devine mai avantajos să definim scopurile, constrângerile, riscul în PNL (limbaj natural precizat), decât analitic și numeric: propozițiile admisibile din NL alcătuiesc NL precizat. **Conceptul de limbaj natural precizat deschide cale largă limbajelor naturale în procesarea informației, decizie și control.**

II.2.4.2. Scale fuzzy

Un proces de măsurare mapează o stare fizică într-o entitate informațională. Într-o definiție generală [Berka, 1983], un proces de măsurare e definit de o scală care stabilește o legătură între sistemul empiric al manifestărilor realității și un sistem abstract (de modelare a realității).

Se definește *simbolismul* $\langle E, S, R \rangle$ cu:

- E setul manifestărilor,
- S setul simbolurilor folosite pentru a reprezenta rezultatele măsurării și
- R o relație pe $E \times S$.

Numim descriere o mapare $D: E \rightarrow S$ conform R și 'meaning' (semnificație) o mapare $D: S \rightarrow E$ conform R . **Simbolismul se zice necontradictoriu și complet când $M(S)$ partiționează E** [Finkelstein, 1975].

Pentru a putea raționa plauzibil (fără a contrazice cel puțin flagrant realitatea) în lumea abstractă, relațiile între stări fizice (manifestări) trebuie mapate în relații între entitățile informaționale (componente ale teoriei). Scala procesului de măsurare se definește ca un *homomorfism de la structura relațională a manifestărilor la cea a valorilor proprietăților măsurate conform teoriei*.

O scală înseamnă $\langle E, S, R, RE, RS, F \rangle$ unde:

- RE e un set de relații pe $E \times E$,
- RS e un set de relații pe $S \times S$,
- $F: RE \rightarrow RS$ e o mapare 1:1.

S-a definit astfel homomorfismul $\langle R, F \rangle : \langle E, RE \rangle \rightarrow \langle S, RS \rangle$.

Conform [Benoit, 2010] o *transformare admisibilă* e o trecere de la o scală la alta ($: S \rightarrow S$) care păstrează homomorfismul. *Tipul unei scale* se definește printr-un subgrup de transformări admisibile considerate echivalente. Schimbând între două astfel de scale nu are niciun impact asupra raționamentului în lumea abstractă. De exemplu în cazul scalelor raționale, cu S setul numerelor reale pozitive, transformările admisibile L_a , cu $a > 0$, sunt $: L_a: S \rightarrow S, L_a(x) = a \cdot x$. Există un singur subgrup pentru aceste transformări, toate scalele raționale fiind echivalente în lumea abstractă. Un exemplu de trecere de la o scală rațională la alta e trecerea de la măsurarea centimetrilor la ‘inches’.

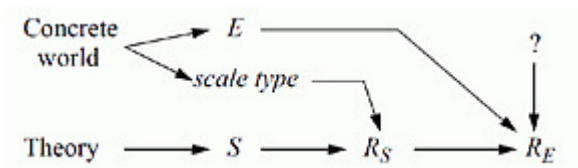


Fig. II.2.5_2 Relațiile între manifestări se pot stabili pornind de la lumea empirică și / sau cea abstractă [Benoit, 2010]

Pentru *scalele nominale* transformările admisibile sunt orice mapare 1:1 pe S , orice subgrup al acestor manifestări fiind definit de o partiție pe E .

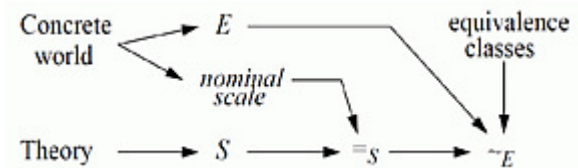


Fig. II.2.5_3 Alegerea unei scale nominale specifice e echivalentă cu definirea unei partiții pe E [Benoit, 2010]

Măsurătorile nominale transformă numai relațiile de echivalență [Benoit & Foulloy, 2002].

Formal, pentru o scală nominală, RS conține numai relațiile de egalitate ($=$) pe S , iar RE conține relații de echivalență (\sim definește o partiție pe E). F e definit de $F = \{(\sim, =)\}$ cu:

$$\forall(e_i \in E), \forall(e_j \in E), e_i \sim e_j \Leftrightarrow M(e_i) = M(e_j).$$

Symbolismul general pentru scalele nominale: $C = \langle E, S, M, \{\sim\}, \{=\}, \{(\sim, =)\} \rangle$

Situația e similară pentru *scalele ordinale*, numai că transformărilor admisibile li se mai cere să fie și monotone, măsurătorile corespunzătoare transformând și relațiile de ordine.

În [Coombs, 1951] s-a introdus conceptul de *scală compusă* în urma compunerii unei scale pe E și a unei scale metrice pe distanțele pe E și S . În această categorie intră și scalele ordinale metrice definite prin:

$$\forall (x, y, z, t) \in E^4, d_E(x, y) \circ d_E(z, t) \Leftrightarrow d_S(D(x), D(y)) = d_S(D(z), D(t)) \quad (*)$$

, cu $\circ = \{=, <\}$, cu $D: E \rightarrow S$ conform R .

Pentru a extinde gama posibilă de definiții ale distanțelor, [Benoit, 2010] a propus folosirea unei scale ordinale metrice bazate pe o *scală fuzzy nominală*.

Scala fuzzy nominală

Scalele fuzzy, introduse pe baza unui operator de similaritate [Benoit & Foulloy, 2003], reprezintă proprietățile manifestărilor prin simboluri numite '*lexical fuzzy subsets*' (*LFS*) sau *subseturi de termeni*. Se definește un φ -symbolism $\langle E, T, R \rangle$ cu R relație fuzzy pe ExT , T fiind setul de termeni. Notății:

- semnificația ('*fuzzy meaning*') $M: T \rightarrow FS(E)$ conform R , unde $FS(E)$ sunt subseturi fuzzy ale lui E
- descrierea ('*fuzzy description*') $D: E \rightarrow FS(T)$ conform R , unde $FS(T)$ sunt subseturi fuzzy ale lui T

Un φ -symbolism se zice necontradictoriu și complet când $M(S)$ e o φ -partiție a lui E [Foulloy & Benoit, 2006]. O φ -partiție a lui E corespunde unei relații de echivalență fuzzy, care se mai numește și relație de similaritate.

În general se preferă partițiile care satisfac condiția:

$$\forall (e \in E), \sum_{t \in T} \mu_{M(t)}(e) = 1$$

, ceea ce corespunde nevoii de excludere între termenii lingvistici folosiți pentru descrierea realității. Exemplu conform cu "x" pe post de "e" și "w" pe post de "t":

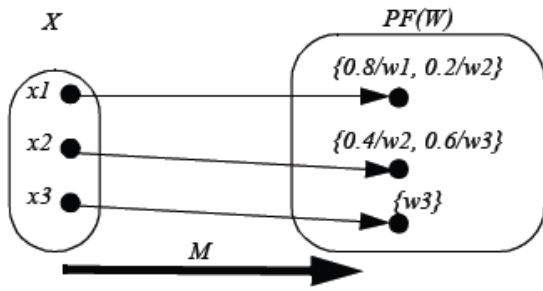


Fig. II.2.5_4 “Interpretare de scală” [Benoit & Foulloy, 2002]

Corespunzător acestei partiții, poate fi dată o relație de similaritate:

$$\mu_{\sim}(e, f) = \sum_{t \in T} \left[\min \left(\mu_{M(t)}(e), \mu_{M(t)}(f) \right) \right]$$

Această relație e reflexivă, simetrică și verifică o versiune de tranzitivitate slabă în sensul T-normei Łukasiewicz.

O scală fuzzy e definită prin $\langle E, T, R, RE, RT, F \rangle$ cu:

- RE e un set de relații fuzzy pe $E \times E$,
- RT e un set de relații pe $T \times T$,
- $F: RE \rightarrow RS$ e o mapare 1:1, cu RS un set de relații fuzzy pe $D(E) \times D(E)$ care respectă ‘singleton coincidence’ în relație cu RT :

$$\forall (a, b) \in T^2, r_S \in RS \Leftrightarrow \exists r_T \in RT, \{a\} r_S \{b\} \Leftrightarrow a r_T b.$$

Principala diferență față de scalele non-fuzzy constă în aceea că simbolurile (entitățile informaționale ale lumii abstracte) sunt subseturi fuzzy de termeni în loc de termeni pur și simpli. O altă diferență e aceea că relația fuzzy r_E on E e mapată pe o relație fuzzy intermediară r_S pe $D(E) \times D(E)$ astfel încât $x r_E y \Leftrightarrow D(x) r_S D(y)$.

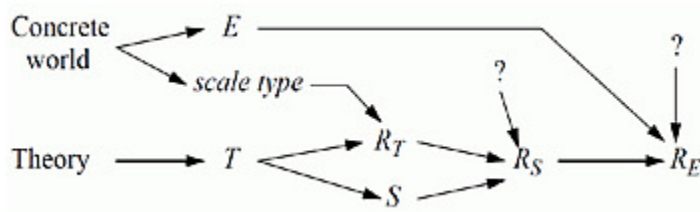


Fig. II.2.5_5 Sistemul abstract relațional e dedus din tipul de scală și T. Introducerea lui S, plecând de la setul de termeni T, furnizează noi posibilități pentru definirea scalei [Benoit, 2010]

În particular, în ideea unei scale metrice, putem avea $dS(\{a\}, \{b\}) = dT(a, b)$. [Benoit, 2010] arată cum de la o scală fuzzy nominală și de la această scală metrică putem construi o scală ordinală metrică $\langle E, T, R, \{\sim E, dE\}, \{=T, dT\}, \{(\sim E, =T), (dE, dT)\} \rangle$. Se propune ca, întrucât în scalele nominale nu se pot defini distanțe, acestea fiind “apanajul” scalelor (nominale) fuzzy, să folosim numai scale ordinale metrice construite peste acestea din urmă, când nu avem la dispoziție informație suplimentară care ar justifica folosirea unei scale metrice mai “puternice”.

Distanța dT , dacă e furnizată de teorie, poate fi dată de scopurile măsurării, desprinzându-se de lumea concretă. Pentru a impune aderența la lumea concretă dT poate fi măsurată empiric. Dificultatea pe care trebuie să o avem în vedere e că lumea concretă e măsurată în cadrul unei teorii. În finalul articolului [Benoit, 2010] se “avertizează” punându-se: întreabarea: “are some distances artefacts of the abstract world, with no equivalent entity in the concrete world?” Desigur, **folosirea unei scale metrice fuzzy e o îmbunătățire, dacă astfel se ia în considerare o relație de proximitate pe setul de reprezentare, care are acoperire într-o relație de proximitate pe setul reprezentat.** De exemplu, în spațiile de culoare perceptual uniforme putem vorbi de o distanță între culori, deci și între denumirile lor.

În cele mai multe aplicații, măsurătorile nominale reprezintă o stare fizică printr-un simbol. Într-o scală nominală fuzzy putem reprezenta o stare fizică printr-o mulțime fuzzy de simboluri (vezi și Fig. II.2.5_4). Acest mecanism, introdus de Zadeh în [Zadeh, 1971], la câțiva ani după introducerea mulțimilor fuzzy, și numit *descriere fuzzy*, e în mod obișnuit aplicat în proiectarea senzorilor fuzzy [Benoit et al., 1994].

II.2.5. PROTOTIPUL = clasa fuzzy

Psihologul Eleanor Rosch a arătat [Rosch, 2010] că anumite cuvinte – pe care le numește “prototipuri” – reprezintă mai bine o clasă decât altele, și că ordonarea acestor prototipuri e conformă cu înțelegerea noastră intuitivă. Rosch a mai afirmat că **necesitatea conceptului de clasă ține de randamentul informațional: clasele (n.n.: cuvintele / prototipurile) sunt**

necesare întrucât furnizează maximum de informație “with the least cognitive effort” [fz ext].

După cum am văzut “legile fuzzy” permit unui obiect să aparțină în același timp mai multor clase: clasele fuzzy se pot suprapune. Dacă pentru clasele / partițiile definite precis (clasic), adică prin relații de echivalență avem termenul de tipuri, pentru clasele / partițiile fuzzy definite imprecis, prin relații de similaritate, propunem termenul de PROTOTIPURI.

Prin PROTOTIP înțelegem un “sâmbure” în care se centrează clasa fuzzy, așa cum un set fuzzy, care poate fi văzut și el ca o clasă fuzzy, are un nucleu și o distribuție de posibilitate în jurul nucleului. Această semnificație a prototipului ni se pare mai potrivită pentru a descrie o clasă fuzzy decât cea de granulă: granula se construiește în jurul prototipului. Aceasta e ideea esențială de set fuzzy. În secțiunea anterioară am avut pe post de prototip termenul lexical.

II.2.5.1. Prototipul ca sursă de potențial: metoda funcțiilor de potențial

Clasificatorul celor mai apropiați "k" vecini devine o metodă laborioasă dacă numărul de forme prototip e mare. O variantă a acestei metode e clasificarea cu ajutorul funcțiilor de potențial. Această metodă constă în construirea unui "câmp potențial" în jurul fiecărui punct (formă) cunoscut din spațiul formelor. Apoi se calculează câmpul potențial global al fiecărei clase folosind (după metoda potențialelor electrice din electrotehnică) principiul superpoziției. O formă de intrare necunoscută va fi atribuită acelei clase care exercită asupra ei câmpul cel mai puternic.

Funcția de potențial poate fi diferită de la o formă la alta în interiorul aceleiași clase, după influența fiecărei forme asupra procesului de recunoaștere, dar și diferită de la o clasă la alta. Funcția de potențial scade monoton cu distanța de la centrul formei punctul de maxim al funcției:

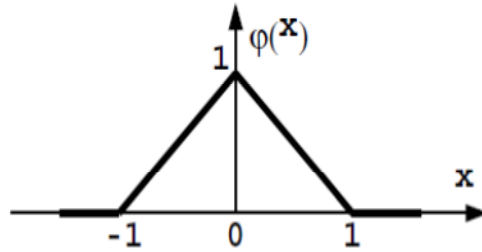


Fig. II.2.5_1 În cazul funcției de potențial triunghiulare, $\varphi(x) = \begin{cases} 0 & , |x| > 1 \\ 1 - |x|, & |x| \leq 1 \end{cases}$ scăderea cu distanța față de centrul funcției e liniară.

Se observă similitudinile care există, ca expresie, între funcția de potențial și densitatea de probabilitate. În practică, funcțiile de potențial se folosesc mult pentru estimarea densității de probabilitate. După definirea funcțiilor de potențial pentru fiecare clasă în parte, ele pot fi apoi modificate aditiv sau multiplicativ, pentru a accentua sau diminua influența unui anume câmp.

O altă variantă permite evaluarea funcțiilor discriminant după metoda celor "k" vecini:

- (1) se determină cei mai apropiați "k" vecini ai formei de intrare în clasificator.
- (2) se sortează pe clase acești "k" vecini.
- (3) se estimează potențialul global al fiecărei clase doar pe baza potențialului celor câțiva vecini.
- (4) se execută clasificarea după regula descrisă anterior.

Descrierea regiunilor (clusterilor) se poate face și printr-un număr de forme prototip atașate fiecărui cluster și o măsură a distanței între forma de intrare și aceste forme prototip. Teoria funcțiilor de potențial poate fi utilizată cu succes pentru deducerea măsurii acestei distanțe și, evident, pentru construcția unei funcții discriminant între clase.

II.2.5.2. Vedere sintetică a capitolului prin prisma termenului de prototip

În lumea reală, incertitudinea / imprecizia e prezentă în toate fazele intervenției ființei umane în încercarea ei de a modela / clasa realitatea. Din punct de vedere al modelării matematice imprecizia se poate datora unor specificații (faza de concepție), măsurători (faza de măsurare și reprezentare) sau transformări în vederea obținerii unor date de interes (faza de analiză). În ce privește faza de concepție, care e determinantă și pentru celelalte faze, s-a încercat captarea realității / semnalelor după un model liniar: acesta e domeniul frecvență (Fourier) de reprezentare a semnalelor. Pentru o mai mare diversitate a granulelor – în cazul de față undelor - care se pot folosi pentru a modela semnalul, atât în ceea ce privește rezoluția cât și forma, s-au introdus ca parametru și timpul: undinele, cu toate varietățile lor (vezi subsecțiunea II.1.4.2.: Aplicații ale undinelor), se definesc atât în scală, cât și în timp. Multe probleme reale, în special cele complexe, e.g. cele care țin de domeniul inteligenței artificiale, nu se pretează unui model liniar. Așa a apărut matematica fuzzy, cu *spațiile metrice probabilistice* și *logicile cu mai multe valori* peste care putem defini T (co)norme, prin care, pe baza noțiunii de grad (de apartenență), se definesc seturi fuzzy care pot modela cunoștințe imprecise. Putem spune că seturile fuzzy corespund unor clase, identitatea a două obiecte în domeniul fuzzy nefiind altceva decât apartenența la aceeași clasă (vezi și argumentația lui Grigore Moisil citată la subsecțiunea II.2.1.2.: Logica fuzzy, utilitatea și legătura ei cu problema clasificării). **Generalizând conceptul de prototip, în domeniul frecvență, prototipul corespunde undei / undinei, iar în domeniul fuzzy, prototipul corespunde clasei. Cu precizarea că, dacă în primul caz, separarea semnalului în unde / undine se face printr-o descompunere ortogonală, în cazul al doilea, separarea în clase admite o suprapunere a claselor.** Dacă în domeniul frecvență separabilitatea e controlată conform principiului incertitudinii (vezi II.1.3.2.: Intercondiționarea rezoluției reprezentărilor în Frecvență și în Frecvență-Timp: principiul nedeterminării), în domeniul fuzzy ea e controlată conform teoremei de separare pentru seturi fuzzy convexe (vezi II.2.2.2.: Mulțimile / partițiile fuzzy). Această teoremă, în “spirit” fuzzy, fără a stabili niște valori precise, vorbește despre un grad maxim de separare.

III§ Domeniul multispectral al imaginilor color

În acest capitol, deși ne vom referi numai la cazul domeniului multispectral al culorilor, adică la receptarea radiației electromagnetice pe 3 componente, discuția noastră se poate generaliza la reprezentări spectrale pentru mai multe componente. Practic, acest capitol prezentăm pe scurt noțiunile teoretice necesare capitolului de contribuții practice (IV§), aducând în același timp argumente în ce privește “fuzitatea” sau caracterul perceptual al culorii

III.1. Introducere în colorimetrie

Spectrul vizibil (numit uneori “spectrul optic”) reprezintă domeniul spectrului electromagnetic ce e vizibil și poate fi detectat de ochiul uman. Radiațiile electromagnetice din acest interval de lungimi de undă se numesc lumină (vizibilă). În condiții normale, ochiul uman percepe în aer lungimile de undă din domeniul 380 - 750 nm (nanometri).



Fig. III.1_1 Spectrul vizibil: de-a lungul $\lambda \in [380,750]$ nm [wikipedia]

Acest spectru corespunde descompunerii luminii ce trece printr-o prismă. În cadrul lui putem delimita fâșii verticale foarte înguste de lumină monocromatică, sau culori spectrale / pure (o idealizare pentru $\Delta\lambda \rightarrow 0$). Lumina monocromatică nu se descompune la trecerea printr-o prismă – e doar deviata.

Dacă măsurăm puterea fiecărei componente a spectrului vizibil obținem:

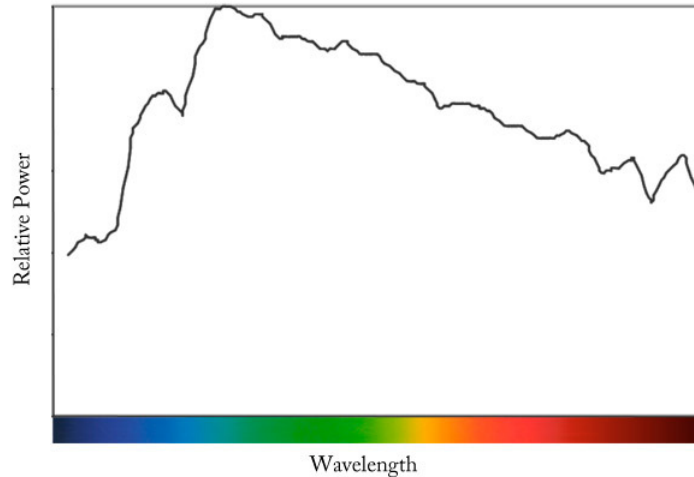


Fig. III.1_2 Spectrul de putere ('spectral power distribution' (SPD)) - o metoda folosită pentru a specifica surse de lumină, e.g., becuri, stele [wikipedia]

III.1.1. Principiul tricromaticității

SPD ar fi o metodă obiectivă de a măsura compoziția unei culori, nedepinzând de biologia și percepția umană (culoarea e pentru noi o senzație). Culoarea ar fi măsurată în watt-i și metri. Dar, lucru dificil și totodată interesant, prin SPD nu putem specifica în mod unic aparența unei culori. Putem vedea culori foarte diferite între ele, dar cu același SPD: oamenii nu văd culorile conform SPD al acestora. Ochiul uman conține trei tipuri de receptori de culoare (conuri) responsabili de formarea senzației de culoare. Pornind de la această constatare, ce ține de natura umană, s-a enunțat principiul tricromaticității: orice culoare monocromatică poate fi sintetizată prin amestecul aditiv a trei stimuli de culoare. Cele trei tipuri de conuri responsabile de formarea senzației de culoare conțin pigmenți fotosensibili ce au absorbții spectrale diferite. Sunt conuri sensibile la lungimi de undă: mici (B - blue), medii (G - green) și mari (R - red).

În 1931, 'International Commission on Illumination' (în limba franceză: 'Commission internationale de l'éclairage'), făcând o statistică a observațiilor culese prin experimente cu mai

mulți subiecți umani, a definit Observatorul Standard, de fapt o tabelă de date care ține în antet lungimile de undă din cadrul spectrului vizibil, iar pe coloane, puterile / coeficienții pentru trei culori spectrale primare, care erau mixate să producă culoarea spectrală (subiecții umani alegeau culoarea sintetizată din culorile primare) dată de lungimea de undă din antet. Această tabelă e foarte importantă întrucât face legătura între SPD și modul în care vedem și putem reproduce o culoare — o legătură între radiometrie și fotometrie. Tabela poate fi reprezentată grafic reprezentând pe abscisă lungimea de undă, iar pe ordonată coeficientul de culoare primară:

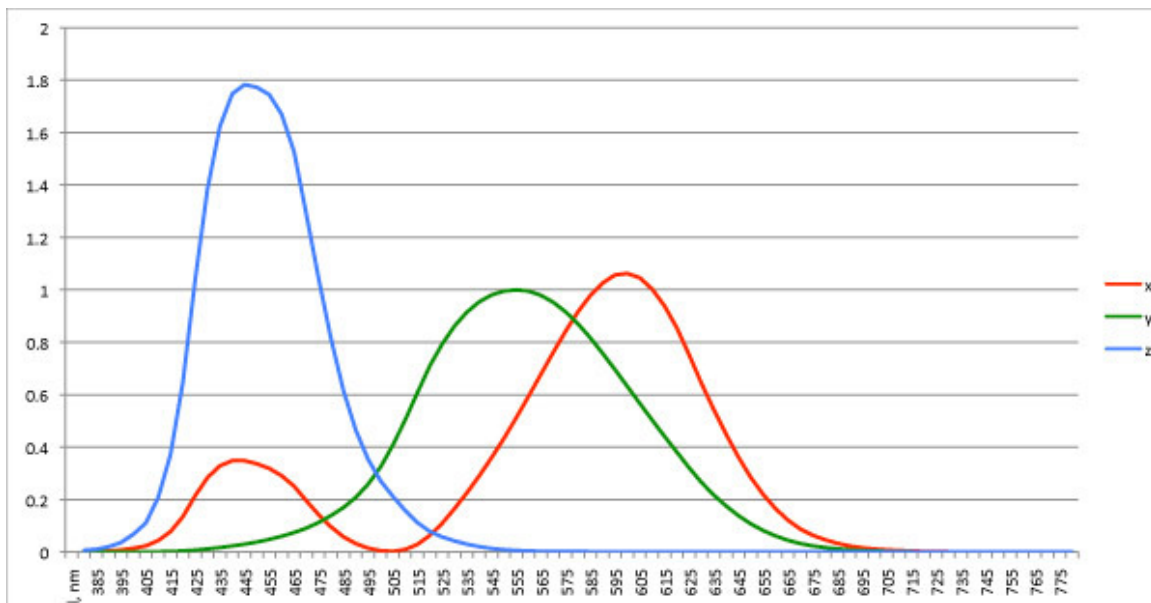


Fig. III.1.1_1 Observatorul Standard conform CIE. Graficul se poate interpreta și ca reprezentare a sensibilităților celor trei tipuri de conuri corespunzătoare celor trei culori primare [cl cr]

Principiul tricromaticității stă la baza colorimetriei și a implementării reprezentării culorilor pe echipamente de afisaj sau imprimare, prin mixajul a numai câteva culori primare.

Matematic, putem scrie răspunsul celor trei tipuri de conuri la o lumină incidentă cu descompunerea spectrală data de $f(\lambda)$, sub forma unui produs scalar:

$$C_{R/G/B} = \int_{\lambda_{min}}^{\lambda_{max}} R/G/B(\lambda) f(\lambda) d\lambda,$$

$R/G/B(\lambda)$ fiind sensibilitățile conurilor. Produsul are loc în spațiul Hilbert L2 al radiațiilor electromagnetice de energie finită. Răspunsul conurilor poate fi văzut și ca proiecția luminii

incidente pe spațiul generat de funcțiile de sensibilitate spectrală $R/G/B(\lambda)$, spațiu care se numește spațiul vizual uman ('Human Visual SubSpace'). Putem vedea această proiecție ca pe o transformare directă, transformarea inversă corespunzând reproducerii culorilor pe / cu baza funcțiilor de sensibilitate spectrală – numite în Fig. III.1.1_3 "culori primare".

În practică, e suficientă o eșantionare la un interval de 10 nm a spectrelor. Relația de mai sus devine: $c = [c_R, c_G, c_B] = S^T f$, unde c se numește vector tristimulus – pe baza lui se poate modela potrivirea, reproducerea și percepția culorilor.

Potrivirea culorilor înseamnă egalitatea răspunsurilor conurilor receptoare. Pentru două spectre (culori) eșantionate f și g , aceasta se scrie: $S^T f = S^T g$. Se demonstrează că folosind trei culori primare, adică, niciuna nu poate fi obținută combinându-le liniar pe celelalte, putem potrivi orice culoare f cu o combinație liniară a culorilor primare. Mai mult, se demonstrează că f și g se potrivesc dacă și numai dacă $A^T f = A^T g$, unde A e matricea a cărei coloană k ($k = 1, 2$ sau 3), corespunde ponderilor culorii primare P_k pentru reproducerea culorilor monocromatice, matrice care se determină experimental mai ușor ca S .

III.1.2. Diagrama de cromaticitate CIE

Conform Fig. III.1.1_3., în definitiv în tabela Observatorului Standard conform CIE, pentru fiecare lungime de undă, avem un rând al valorilor tristimulus:

lungime de undă (nm)	X	Y	Z
380	0.001368	0.000039	0.006450

Pentru a normaliza vectorii tristimulus, proiectăm spațiul 3-D al acestora, într-un spațiu 2-D definit de coeficienții de tricromaticitate:

$$\begin{cases} x = X / (X + Y + Z) \\ y = Y / (X + Y + Z) \\ z = 1 - x - y \end{cases}$$

Deși aceste coordonate au fost obținute pe baza unor experiențe cu subiecți umani, ele identifică cu suficientă acuratețe o culoare, pe baza lor putându-se atât sintetiza cât și analiza culorii. Ceea ce nu reușește reprezentarea SPD, deși e o reprezentare (mult mai) obiectivă.

Reprezentarea acestor coeficienți e diagrama de cromaticitate, o reprezentare care relevă grafic cele mai importante proprietăți ale culorilor:

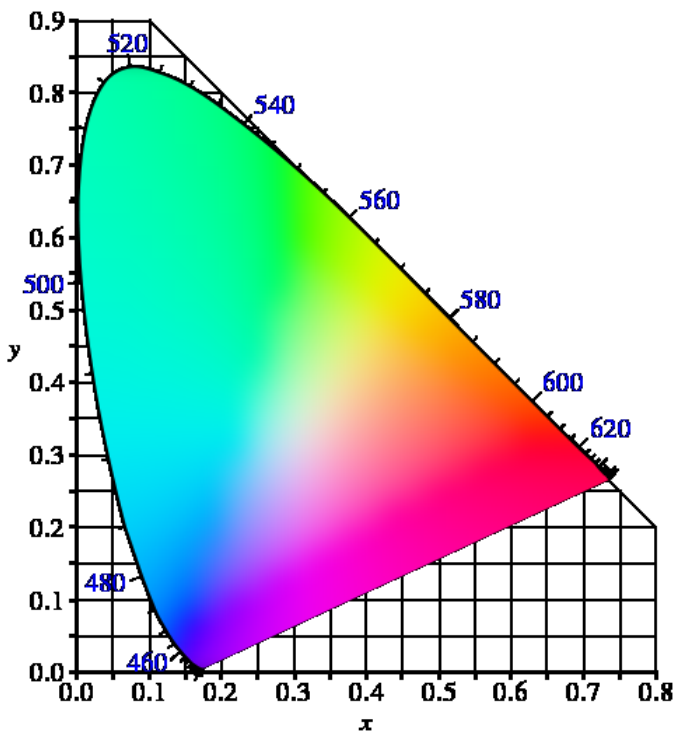


Fig. III.1.2_1 Diagrama de cromaticitate – o hartă plană a culorilor [cl cr]

Observăm că nu se poate defini un triunghi al cărui gamut să acopere toate culorile (reprezentate de diagramă), deci nu toate culorile pot fi obținute prin combinația liniară a trei culori primare. De reținut că această diagramă nu conține culorile, care se pot reprezenta adecvat într-un spațiu 3D, ci e o hartă (de vizualizare a lor). O culoare poate fi descompusă ortogonal în luminanță (o componentă) \oplus crominanță (două componente) - diagrama de cromaticitate corespunde crominanței. Una din cele mai importante proprietăți ale culorilor ilustrate prin diagramă, e aceea

a obținerii unei culori c_3 ca interpolare liniară între capetele c_1 și c_2 ale segmentului pe care se află c_3 (desigur, lucrurile se întâmplă, așa cum am atras atenția, în domeniul proiecțiilor culorilor: e important să știm că reprezentările pe diagrama de cromaticitate sunt orientative, întrucât se referă la proiecțiile acestora !). Presupunem trei puncte necoliniare în interiorul diagramei – ele definesc un triunghi al cărui interior ('gamut') corespunde tuturor culorilor care se pot obține prin interpolarea liniară a culorilor reprezentate de vârfurile triunghiului:

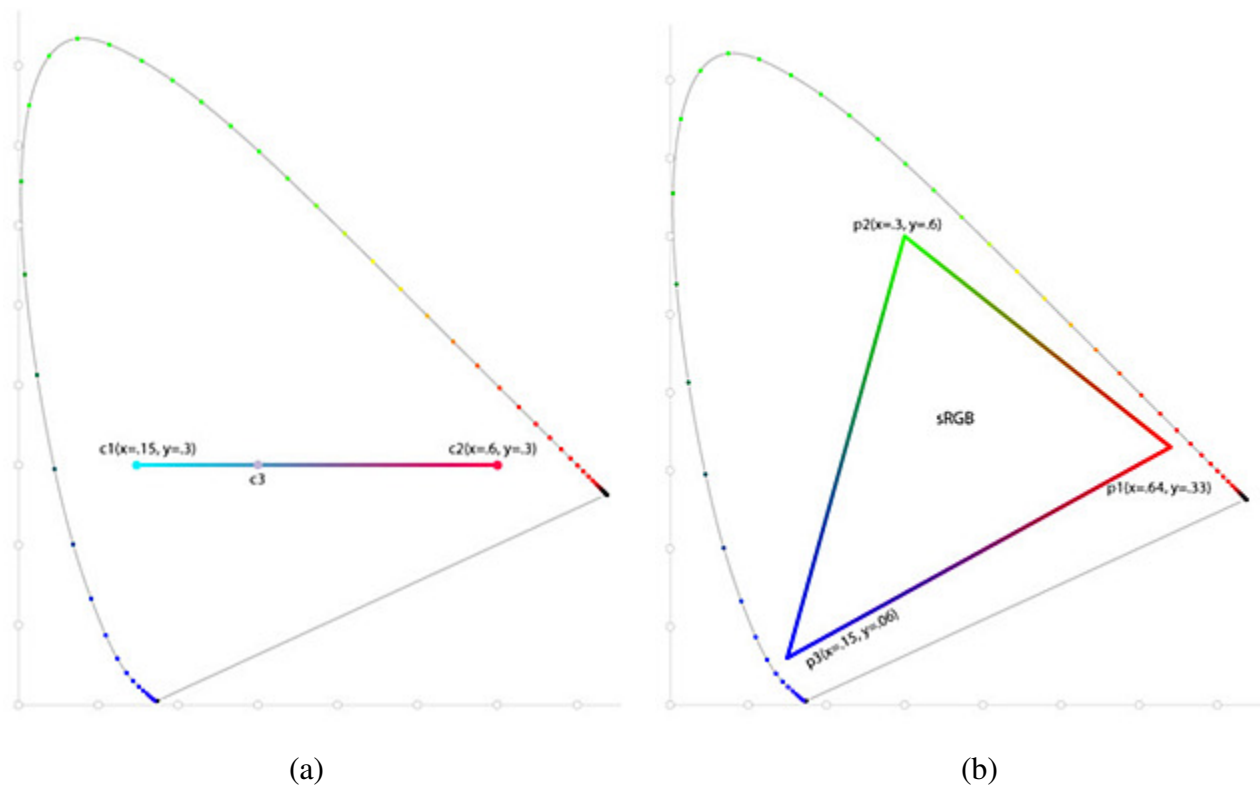


Fig. III.1.2_2 (a) Interpolarea liniară a culorilor: ! aceeași culoare poate fi obținută din oricare alte două culori care definesc un segment ce conține culoarea – acesta e un caz de metamerism – vezi subsecțiunea următoare. (b) Definirea spațiilor colorimetrice înseamnă definirea celor trei culori primare - iată cazul spațiului de culoare Standard RGB [$P_1(0.64, 0.33)$, $P_2(0.3, 0.6)$, $P_3(0.15, 0.06)$]

Observație: E important să știm că reprezentările 'full gamut' ale spațiilor de culoare pe diagrama de cromaticitate sunt orientative, întrucât se referă la proiecțiile acestora

III.2. O trecere în revistă a spațiilor de culoare

III.2.1. Sisteme de reprezentare primare

Am arătat cum culoarea oricărui spectru vizibil f poate fi specificată pe baza valorilor tristimulus $A^T f$, A fiind o matrice $N \times 3$. Coloanele lui A corespund eșantionării a ceea ce se numesc funcțiile de potrivire a culorilor. Definind un set standard de funcții de potrivire a culorilor, precizăm un spațiu de culoare în care măsurătorile valorilor tristimulus să fie consecvente.

Prima standardizare a terminologiei și măsurătorilor colorimetrice a fost realizată de CIE în 1931. CIE a recomandat atunci două seturi standard echivalente de funcții de potrivire:

CIE RGB

- funcțiile de potrivire $R/G/B(\lambda)$ sunt asociate culorilor primare monocromatice cu lungimile de undă: 400 / 546, 1 / 435, 8 nm, alese astfel încât valorile tristimulus pentru un spectru corespunzător luminii albe să fie egale.

CIE XYZ

- e o transformare liniară a spațiului CIE RGB venind cu două condiții în plus: toate valorile tristimulus să nu fie negative și componenta Y să corespundă luminanței.

Sisteme de culoare primare sunt aditive: culoarea e reprodusă prin suma ponderată a culorilor primare. Dual, dacă folosim culori complementare celor primare, bazându-ne pe proprietatea de absorbție a culorilor primare de către straturile corespunzătoare acestor culori complementare, avem sisteme de culoare ce sunt substructive, e.g., spațiul de culoare CMYK, folosit mai ales la imprimante.

III.2.2. Sisteme de reprezentare derivate

Aceste sisteme, în fapt transformări lineare ale sistemelor primare, au fost impuse mai ales de industria TV: dacă pentru afișarea imaginilor se folosesc culorile primare date de caracteristici hardware, pentru transmisia datelor se folosesc alte standarde. Actualmente există două standarde *de facto* pentru transmisia TV: PAL cu sistemul YUV și NTSC cu sistemul YIQ. S-a avut în vedere compatibilitatea cu TV alb / negru - nivelul de gri fiind dat de luminanță (Y) – și alegerea componentelor de cromaticitate a.î plaja lor să fie cât mai mică.

Spațiul de culoare Ohta a fost gândit astfel încât componentele de culoare să fie cât mai decorelate. Astfel, transformarea e o bună aproximare a transformării Karhunen–Loève.

III.2.3. Sisteme de reprezentare perceptuale

Se poate spune că aceste sisteme se inspiră din modul în care pictorii, pentru a obține culori noi, amestecă pigmenți relativ puternici – “saturați” – cu alb și / sau negru. Munsell a separat atributele unei culori în:

- nuanță ('hue' : H) – e.g., cât de “roșie” sau “verde” e culoarea
- cromaticitate (C) / saturație (S) – cât de amestecată e culoarea cu alb și/sau negru
- strălucire / intensitate / luminanță ('value': V)

E o reprezentare care se apropie și mai mult de percepția noastră asupra culorii. Spații de culoare bazate pe coordonate de tip H, S și V: HSV, HSL, HIS, HVC, HSL, LCH.

Cele 3 atribute se reprezintă de această dată într-un spațiu de coordonate polare:

- nuanței îi corespunde unghiul polar
- saturației îi corespunde distanța până la axa de simetrie rotațională
- intensității îi corespunde axa

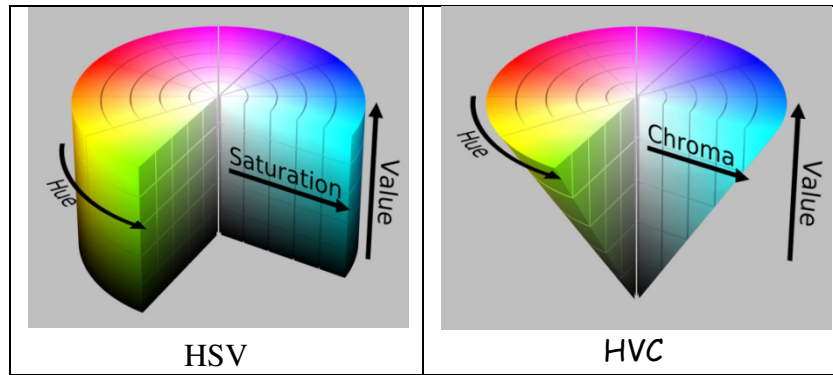


Fig. III.2.3_1 Spațiul de culoare HSV. Dacă reprezentăm cromaticitatea în locul saturației, în locul cilindrului obținem un con (spațiul de culoare HVC) [wikipedia]

Avantajele sistemelor de reprezentare perceptuală:

- sunt intuitive, chiar dacă modelează și cazuri excepționale care contrazic intuiția, care spune, de exemplu, că un galben saturat e întotdeauna mai luminos decât un albastru la fel de saturat
- separă luminanța de cromaticitate - acesta e un avantaj foarte important (folosit pe larg în prelucrarea și segmentarea de imagini) întrucât permite folosirea independentă a componentelor, e.g., H pentru segmentare, I pentru îmbunătățire contrastă

Dezavantaje:

- există singularități în transformările $RGB \leftrightarrow HSV$: nuanța nu e definită pentru gri, saturația nu e definită pentru alb sau negru pur; în jurul acestor singularități avem o sensibilitate care poate fi semnificativă
- nuanța, având valori unghiulare, nu suportă operații cum ar fi medierea
- depind de sistemele RGB, iar transformarea $RGB \rightarrow HSV$ e neliniară și cu sensibilități accentuate la extremități, corespunzând rotirii și proiecției cubului RGB într-un hexagon / cerc
- neuniformitate perceptuală: nu se pot defini grade de similaritate între (două) culori în funcție de distanța dintre aceste culori.

Observație: Calificarea spațiilor de culoare ca “independent de echipament” (‘Device Independent’) sau ‘dependent de echipament’ se folosește frecvent în problemele de înregistrare și reproducere a culorilor. Un spațiu de culoare e indep de echipament, dacă există o transformare nesingulară între el și spațiul de culoare CIE XYZ. Aceste spații se mai numesc “neredate”, deoarece valorile din spațiu nu sunt folosite pentru redarea pe un “output device”. DD sunt în general “legate” de un ‘input / output device’. Un exemplu de spațiu DD e DD RGB: valorile din spațiu sunt trimise pentru afișare direct la monitor sau imprimantă – acesta e avantajul spațiilor DD, chiar dacă imaginile afișate nu arată la fel cu cele originale (scanate).

Pentru a menține simplitatea spațiilor DD, dar și pentru a potrivi măcar într-un anumit grad culorile între un echipament de intrare și unul de ieșire, se folosesc spațiile DD standard – care se pot defini în funcție de spații DI.

III.2.4. Sisteme de reprezentare perceptual uniforme

Charles Poynton, un expert în prelucrarea imaginilor digitale, spunea: “HSL și HSV au fost concepute pentru a specifica numeric cele trei componente (H,S,L). La acea vreme culorile trebuiau specificate numeric. Dar acum utilizatorul poate alege culorile vizual, dintr-un Color Checker sau Pantone, sau se pot folosi sisteme perceptual uniforme ca $L^*u^*v^*$ sau $L^*a^*b^*$.”

În practică, e mai la îndemână evaluarea diferenței dintre culori asemănătoare decât dintre culori mult diferite. Astfel, ca unitate de măsură a diferenței dintre culori, apare “diferența abia perceptibilă” (‘Just Noticeable Difference’). JND în sistemele de reprezentare primare e foarte neregulată. În planul de cromaticitate xy, JND corespunde elipselor Mac Adams – în interiorul acestora culorile nu pot fi diferențiate vizual. Mărimea și excentricitatea acestor elipse variază puternic cu poziția centrului lor.

Construirea unui spațiu de culoare perceptual uniform se face stabilind o scară de iluminare uniformă și o diagramă de cromaticitate uniformă pentru culori cu aceeași luminanță. Cele două tipuri de informație – de luminanță și crominanță se combină prin scalarea coordonatelor, astfel încât unitățile să corespundă aceleiași proporții de JND.

În spațiul Uniform Chromaticity Scale uniformitatea perceptuală e aproape realizată – elipsele Mac Adams sunt de dimensiuni și forme aproximativ egale. Pentru o uniformitate perceptuală mai bună și pentru a suprapune originea spațiului *culorii albe de referință* (‘White Point’) standard, UCS a fost modificat în $U^*V^*W^*$, spațiu standardizat de CIE în 1960. În prezent, conform recomandărilor CIE 1976, se folosesc spațiile Luv și Lab. De data aceasta, deși sistemul de coordonate e cartezian (ca în cazul CIE RGB și XZY), componentele iau valori reale (nu neapărat întregi).

WP pentru un iluminant se referă doar la cromaticitate (nu și la luminozitate) - aceea a unui obiect alb “sub” acel iluminant. În general, nu se precizează WP, ci iluminantul (implicit e D50).

WP poate fi stabilit în funcție de aplicație (*'white balancing'*) – pentru imaginile dintr-o încăpere cu iluminare roșiatică, albul de referință va fi și el mai roșiatic, altfel, e posibil să obținem rezultate inacceptabile când încercăm să corectăm aceste imagini.

Dacă pentru majoritatea iluminanților, cum ar fi iluminanții standard (CIE) de tip D (*'daylight'*) corespondența iluminant – WP e 1 : 1, pentru alți iluminanții relația e 1 : n. De exemplu, în cazul CIE E (*'equal'* – SPD are o distribuție uniformă), coordonatele WP sunt de forma (k,k,k), k = constant, coordonatele cromatice fiind $[x,y] = [1/3, 1/3]$

Conversia între iluminanți diferiți: dacă înregistrăm culoarea unui obiect sub un iluminant, putem evalua culoarea aceluiași obiect sub un alt iluminant, dându-se doar WPs pentru cei 2 iluminanți. O metodă des întâlnită e transformarea de adaptare cromatică *'chromatic adaptation transform'* – o vom folosi și noi la conversia între spații de culoare.

Spațiul CIE Luv pleacă de la spațiul UVW și, spre deosebire de Lab, se potrivește situațiilor cu un singur iluminant, unde nu se face vreo transformare de adaptare cromatică. Spațiul CIE Lab e cel mai complet spațiu de culoare standardizat și a fost creat pentru a fi o referință, independent de vreun echipament.

Chiar dacă Lab (e vorba de spațiul de culoare CIE Lab, și nu de spațiul de culoare Lab definit de Hun în 1948) are un gamut mult mai cuprinzător decât al vreunui echipament - mai cuprinzător și decât poate cuprinde vederea umană -, trecerea la o reprezentare 16 biți / componenta asigură suficientă precizie pentru coordonatele Lab, uneori culorile imaginare obținute în Lab sunt pași intermediari către culori reale, Lab oferind un spațiu de manevră generos. Cromatica Lab e prinsă în planul ab: Oa corespunde poziționării pe axa verde-roșu (> 0), iar Ob corespunde poziționării pe axa albastru-galben:

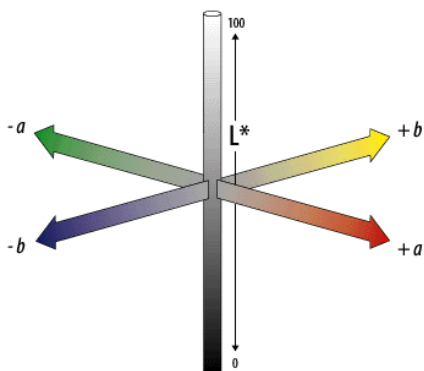


Fig. III.2.4_1 Spațiul de culoare Lab

După cum ne așteptăm, transformările între spațiile de culoare primare și cele perceptuale uniform sunt neliniare. Conform [Vertan & Ciuc, 2007, p. 190]

$$\left\{ \begin{array}{l} L = 116 f\left(\frac{Y}{Y_0}\right) - 16, \text{ unde } f(t) = \begin{cases} t^{1/3}, & t > (6/29)^3 \\ \frac{1}{3}(29/6)^3 t + \frac{16}{116}, & \text{în rest} \end{cases} \\ a = 500 \left[f\left(\frac{X}{X_0}\right) - f\left(\frac{Y}{Y_0}\right) \right] \\ b = 200 \left[f\left(\frac{Y}{Y_0}\right) - f\left(\frac{Z}{Z_0}\right) \right] \end{array} \right.$$

Lab (ca și Luv) poate fi trecut din coordonate carteziene în coordonate polare prin formulele:

- Nuanță: $H_{ab} = \arctg\left(\frac{b}{a}\right)$

- Saturație: $C_{ab} = \sqrt{a^2 + b^2}$

Distanța dintre două culori e:

$$\Delta E_{ab} = \sqrt{\Delta L^2 + \Delta a^2 + \Delta b^2}, \text{ în reprezentarea carteziană și}$$

$$\Delta E_{ab} = \sqrt{\Delta L^2 + \Delta H_{ab}^2 + \Delta C_{ab}^2}, \text{ în reprezentarea polară}$$

O JND corespunde lui $\Delta E_{ab} \cong 2.3$, în reprezentarea carteziană.

JND e un fapt care atestă fuzitatea conceptului de culoare – era de așteptat întrucât culoarea e o percepție, însăși standardizările CIE bazându-se pe experimente cu expertul uman. În [Vertan & Boujemaa, 2000], pe baza JND, se propun histograme fuzzy care să măsoare “tipicalitatea” fiecărei culori dintr-o imagine în raport cu toate culorile din imagine, pe baza definirii unor distanțe între culori care corespund unor T-norme. Astfel, s-au găsit aplicații în regăsirea

imaginilor pe baza conținutului (Content Based Image Retrieval). În [Vertan et al., 2000] să măsoară “credibilitatea” fiecărei culori în raport cu toate culorile din fereastra de filtrare, pe baza unei distanțe definite conform JND.

În subcapitolul următor vom dezvolta subiectul caracterului preceptual al culorii pentru a ajunge la “Fuzitatea culorii și procesarea multi-canal” (ultima secțiune a capitolului).

III.3. Caracterul perceptual al culorii

III.3.1. Metamerismul sau granularitatea spectrului vizibil

Potrivirea metamerică (a două culori) sau metamerismul e situația în care două spectre de puteri diferite apar ca aceeași culoare. Metamerismul survine faptului că există numai trei tipuri de conuri care trebuie să încarce cumulativ energie venind din partea unui interval larg de lungimi de undă - spectrul vizual -, astfel încât același răspuns al conurilor poate corespunde mai multor combinații de lumini monocromatice. Matematic, aceasta se datorează faptului că matricea S sau A nu e pătrată.

Două spectre eșantionate de culori diferite, dacă se potrivesc vizual se numesc *metameri*. În esență metamerismul e aliasing de culori și poate fi descris generalizând teorema de eșantionare a lui Shannon. Iar aliasing-ul, în general, e o formă de manifestare a nediscernabilității spectrelor de către om – ne aflăm într-un domeniu al impreciziei. Din perspectivă fuzzy metamerismul corespunde granulației care apare la trecerea culorii de la reprezentarea spectrală la cea dată de cele trei tipuri de conuri receptoare.

Potrivirea (metamerică) a culorilor are loc în special în cazul culorilor neutre (gri) sau închise. Metamerismul depinde de:

- iluminare, pentru că aparența culorii depinde de aceasta. Astfel, dacă schimbăm sursa de lumină, culorile pot să nu mai aibă aceeași aparență
- geometrie – schimbarea unghiului de vedere poate “strica” metamerismul
- observator – un observator poate sesiza metamerism acolo unde altul nu-l “vede”
- dimensiune

III.3.2. Constanța culorii și vederea prin ‘retinex’

Se poate vorbi de o inerție a percepției culorii: culorile aceluiași obiect apar ca neschimbate sub diferite condiții de vizualizare (mai ales de iluminare). Un măr verde ne apare ca verde atât la prânz, cât și la amiază. Constanța culorii e o calitate care ne ajută să recunoaștem obiectele. Conform wikipedia, în ‘computer vision’ au fost proiectați mulți algoritmi care să simuleze această calitate prin estimarea reflectanței. Constanța culorii e un concept înrudit cu memoria culorilor și actualizarea (‘discounting’) iluminantului (concept pe care îl vom folosi și noi în recunoașterea pixelilor de piele). În esență, e vorba de faptul că vedem “nu numai prin retină dar și prin cortex, adică prin ‘retinex’”, după cum inspirat a numit Edwin H. Land acest sistem.

Constanța culorii e antagonică aparenței culorii - calitate pe care modelele de aparență a culorii (ca Hunt sau RLAB) o estimează pentru diverse condiții de vedere.

III.3.3. Teoria culorilor opuse

Această teorie a fost creată de Hering în anii '20 pe baza similitudinii culorilor. Similitudinea implică un anumit grad de asemănare, ceea ce ne spune că suntem în domeniul fuzzy:

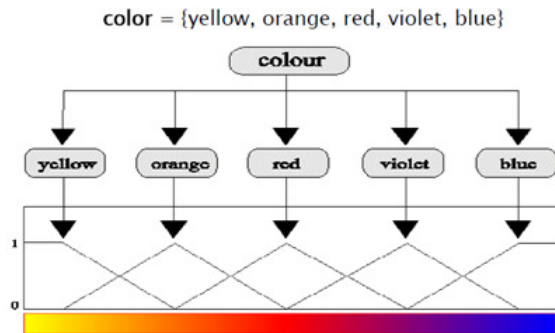


Fig. III.3.3_1 Reprezentarea fuzzy a culorilor, sau, mai exact, a variabilei lingvistice “culoare”

Teoria culorilor opuse stabilește o dihotomie - orice culoare poate fi descrisă ca:

- roșiatică SAU verzuie
- galbenă SAU albastră

O observație experimentală interesantă e aceea că o linie verde pe un fond roșu apare ca fiind gri. Explicația ține de capacitatea omului de a integra și însuma imediat informațiile pe care le primește. Dacă tricromaticitatea e un model al retinei, teoria culorilor opuse e un model al unui mecanism central din creier.

Culorile opuse pot fi văzute ca fiind complementare, culori care se definesc prin aceea că suma lor produce o culoare neutră (gri). Un experiment relevant în ce privește “echilibrul” culorilor opuse e acela al postimaginii: dacă fixăm cu privirea un obiect de o culoare C și apoi privim pe un perete alb, vom vedea aceeași imagine într-o culoare complementară lui C. Psihologia percepției vizuale atribuie această comportare oboselii unor părți ale sistemului vizual.

III.3.4. Fuzitatea culorii și procesarea multi-canal

În ce privește procesarea multi-canal a semnalelor în [Vertan & Buzuloiu, 2000] se observă că:

- 1) procesarea multi-canal a semnalelor a devenit o necesitate.

2) procesarea independentă a fiecărui canal s-a dovedit inadecvată deoarece neglijează corelația între canale; pe de altă parte tehnicile de decorelare, e.g., KLT, s-au dovedit ineficiente computațional și adecvate doar pentru clasa semnalelor cu proprietăți statistice stabile în timp.

În schimb, s-au exploatat cu succes câteva proprietăți interesante ale spațiilor de culoare perceptuale, cum ar fi:

- în imaginile naturale saturația e relativ scăzută și e proporțională într-un grad semnificativ cu nuanța

- zgomotul independent per canalele R / G / B e redus în componenta I (luminanță)

De exemplu, în [Carron & Lambert, 1996] s-au introdus câteva reguli fuzzy de decizie “soft” în ce privește importanța relativă a celor 3 componente:

1. if (Saturation is low) then (the Hue is irrelevant) AND (the Intensity is used for further processing)
2. if (Saturation is medium) then (the Hue is weakly relevant) AND (Hue and Intensity used for further processing)
3. if (Saturation is high) then (the Hue is very relevant) AND (the Hue is used for further processing)

Imaginile digitale sunt mapări 3D → 2D ale scenelor naturale, eșantionate, cuantizate și, astfel, înglobează multă incertitudine, atât în ceea ce privește valoarea cât și locația. Abordarea fuzzy a procesării multi-canal a venit natural – *filtrele fuzzy* sunt o categorie aparte de filtre, numite și “*granular adaptive*”, se bazează, în mare parte, pe experiența unui expert (uman) capabil să furnizeze reguli de clasificare.

IV§ Contribuții practice

IV.1 Transformări eficiente între spații de culoare

Utilitatea trecerii dintr-un spațiu de culoare în altul e evidentă din moment ce o problemă se poate rezolva mai bine (am construit deja un algoritm e.g., un algoritm de “reducere a ochilor roșii”, optim, calibrat) într-un spațiu de culoare, X , decât în altul, Y . Portarea rezolvării problemei din X în Y presupune etapele: aplicarea unei transformări directe $X \rightarrow Y$, rezolvarea problemei în Y și transformarea inversă $Y \rightarrow X$.

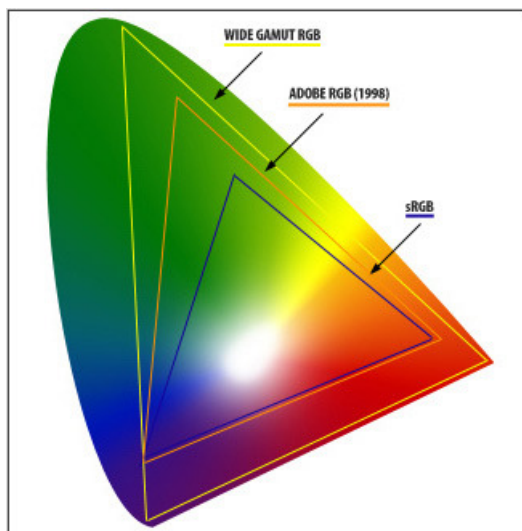
Unele spații de culoare decorează mai bine componenta de luminanță de cea de cromaticitate, ceea ce poate fi interesant, în particular, când ne propunem să extindem o prelucrare pe imagini (în scară de gri) la imagini color. Spațiile de culoare uniform perceptuale sunt mai adecvate când ne propunem să lucrăm cu o metrică în spațiul culorilor. Utilitatea apare și când, pur și simplu vrem să vedem o imagine în spațiul ei nativ, adică în spațiul de culoare în care a fost concepută. În ce privește problema generală pe care o tratăm în această lucrare, considerăm că alegerea unui spațiu de culoare potrivit poate fi decisivă pentru rezolvarea unei probleme de clasificare în imagini.

Contribuția noastră constă în faptul că introducem o metodă nouă prin care obținem performanțe mai bune în ce privește acuratețea și viteza de calcul a unei conversii oarecare între spații de culoare pentru echipamente – vezi și [Georgescu 1, 2009].

Necesitatea de spații de culoare diverse și de conversii între acestea a dus la necesitatea standardizării, lucru de care s-a ocupat ICC - International Color Consortium - , numele nou pentru CIE [ICC]. Să reamintim că CIEXYZ e “the master color space”, baza pentru toate *sistemele de management de culori*, conținând toate culorile perceptibile de către om. CIELAB derivă din CIEXYZ, fiind varianta sa uniform perceptuală. Astfel, aceste două spații de culoare sunt de referință.

IV.1.1. Conversia între spații de culoare pentru echipamente

În acest subcapitol vom folosi două spații de culoare echipament (pentru echipamente), aditive, care sunt un standard de facto în industrie: sRGB (standard RGB) și Adobe RGB (1998) - vezi [ICC]. sRGB e un standard promovat de corporațiile HP și Microsoft, fiind cel mai cunoscut spațiu de culoare pentru monitoarele PC-urilor și spațiul de culoare presupus de cele mai multe aplicații (inclusiv browser-ele web), deși gama sRGB e una relativ redusă. Adobe RGB (1998) e spațiul de culoare cel mai folosit în publicații de designeri, graficieni și, în general, lucrători profesioniști cu imagini digitale. Acest spațiu are o gama mult mai largă decât sRGB, putând captura nuanțe de culoare mai “adânci” sau cu o “strălucire” mai mare. Oricum, dacă aplicația nu “știe” că imaginea a fost înregistrată în Adobe RGB, ea va arăta la fel de “plată” pe un echipament cu spațiul de culoare implicit sRGB, conținând și conversii de culoare incorecte, e.g.



verdele va apărea prea galben etc. Să vedem poziționarea în interiorul diagramei de cromaticitate a celor două spații de culoare echipament între care vom implementa conversia:

Fig. IV.1.1_1 ‘Gamut’ sRGB și Adobe RGB (1998) conform wikipedia

IV.1.1.1. Principiul

Conform ICC o transformare, dintr-un spațiu de culoare echipament ('device color space') în altul, trece printr-un PCS ('Profile Connection Space'): Color Space 1 \rightarrow PCS (CIELAB sau CIEXYZ) \rightarrow Color Space 2. Fiecare echipament care capturează și afișează culori poate avea profilul său. Unele produse vin cu profilele furnizate de producător, dar putem genera și noi profilul pe care îl dorim folosind un colorimetru.

Ilustrăm procesul conversiei între spațiile de culoare echipament sRGB (iluminant D_{65}) și ProPhoto (iluminant D_{50}). Chiar dacă e vorba de aceste spații de culoare, procedura ilustrată e una generală, adică de conversie între oricare două spații de culoare (pentru echipamente). Principiul conversiei: transformarea trebuie să "treacă" prin stimulii absoluți ('absolute tristimulus' conform ICC):

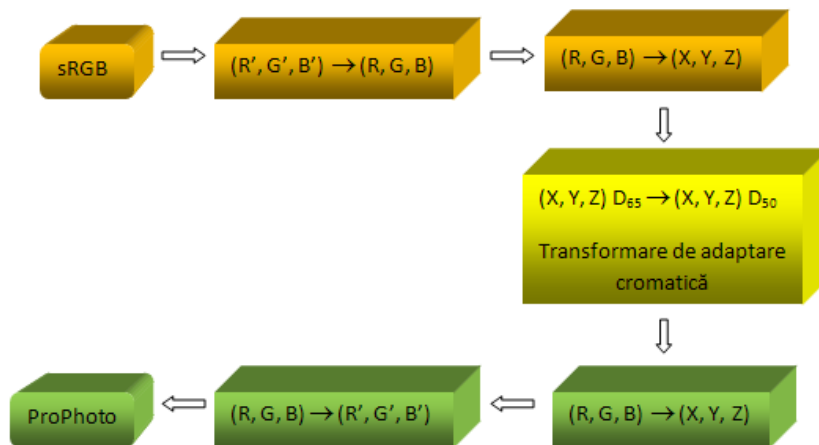


Fig. IV.1.1_2 Etapele procesului de conversie între două spații de culoare echipament

Precizăm aceste etape pentru, urmând să le detaliem în subsecțiunile următoare:

1. (R', G', B') echipament $\rightarrow (R, G, B)$ liniar după corecția gamma sRGB
2. (R, G, B) liniar $\rightarrow (X, Y, Z)$: tristimulus relativ sRGB
3. (X, Y, Z) , adică tristimulus relativ sRGB $\rightarrow (X, Y, Z)$ tristimulus absolut în cadrul conversiei între iluminanți diferiți ('chromatic adaptation transform')
4. și 5. reiau în sens invers 2. și 1., de data asta în spațiul de culoare destinație ProPhoto

IV.1.1.2. De la spațiul de culoare echipament la PCS

PCS e o prescurtare atât ‘Profile Connection Space’ cât și ‘Profile Color Space’, în fapt două expresii echivalente. Acest profil descrie transferul de la spațiul de culoare echipament la spațiul de culoare absolut, adică standardizat. Trecerea de la spațiul de culoare echipament la PCS cuprinde etapele 1. și 2. de mai sus.

Etapa 1. constă în trei corecții gamma, una per fiecare canal – e.g., pentru componeta R avem:

$$R_1 = \text{redTRC}[R_d] \quad (1)$$

: unde R_l e valoarea ‘linear’ și R_d e valoarea ‘device’. Similar, pentru celelalte componente, G și B. TRC (‘tone reproduction curves’) e o expresie echivalentă cu ‘gamma correction’, fiind legată de electronica echipamentului.

Transformarea gamma inversă (“RGB \rightarrow R'G'B”) e:

$$R_d = \begin{cases} \text{redTRC}^{-1}[R_1], & 0 \leq R_1 \leq 1 \\ \text{redTRC}^{-1}[0], & R_1 < 0 \\ \text{redTRC}^{-1}[1], & R_1 > 1 \end{cases} \quad (2)$$

Similar, celelalte componente, G și B. Observăm trunchierea la intervalul [0,1]. De reținut și că transformarea gamma (directă și inversă) e specificată prin puncte de control cu coordonatele în intervalul [0,1].

Etapa 2. Transformarea (R, G, B) liniar \rightarrow valorile de tristimulus relativ (X, Y, Z) sRGB e o transformare liniară, i.e. care poate fi descrisă printr-o matrice 3 x 3 (ea se găsește codificat în fișierul .icm). Etichetând cu ‘r’, ‘g’ și ‘b’ coloanele și cu ‘X’, ‘Y’ și ‘Z’ rândurile acestei matrici, expresia lui X_{relativ} e

$$X_r = (rX, gX, bX) \cdot (R, G, B)_l \quad (3)$$

Similar, Y și Z, vom folosi în formulă Y și, respectiv, Z. Transformarea inversă se face prin înmulțire cu inversa matricei.

IV.1.1.3. De la colorimetria relativă a PCS la colorimetria absolută

Etapa 2., adică transformarea $(X_r, Y_r, Z_r) \rightarrow (X_a, Y_a, Z_a)$ (tristimulus relativ sRGB \rightarrow tristimulus absolut), constă într-o scalare:

$$X_a = (X_{mw}/X_i) \cdot X_r \quad (4)$$

Similar, se calculează Y_a și Z_a , cu $(X, Y, Z)_{mw}$ media punctului de alb ('White Point'); $(X, Y, Z)_i$ reprezintă albul PCS-ului ('illuminant white', care, e.g., pentru spațiul de culoare echipament ProPhoto din Fig. IV.1.1_2, e D50, cu $X=0.9642$, $Y=1$, $Z=0.8249$).

IV.1.2 Implementarea conversiei

Vom implementa conversia din Fig. IV.1.1_2 cu Adobe RGB (1998) pe post de ProPhoto, în ambele sensuri: sRGB \leftrightarrow Adobe RGB.

Spațiul de culoare pentru un echipament se codifică conform ICC într-un fișier '.icm'. Vom folosi varianta Nikon a celor două fișiere. Deși implementarea noastră e pe 8 biți / canal (componentă), ea se poate adapta ușor pe 16 biți / canal.

IV.1.2.1. Modulele aplicației

Iată, în ordinea apelării, modulele aplicației pe care am proiectat-o și implementat-o în C++, în ideea portabilității codului, dar și a testării vitezei de calcul:

Citirea și decodarea informației dintr-un fișier .icm

- date fiind prin două .icm-uri cu profilele de culoare (PCS) cele două spații de culoare între care facem conversiile (directă și inversă), acest modul va furniza o structură cu informația necesară conversiilor

Implementarea efectivă a conversiilor - iată blocurile acestui modul:

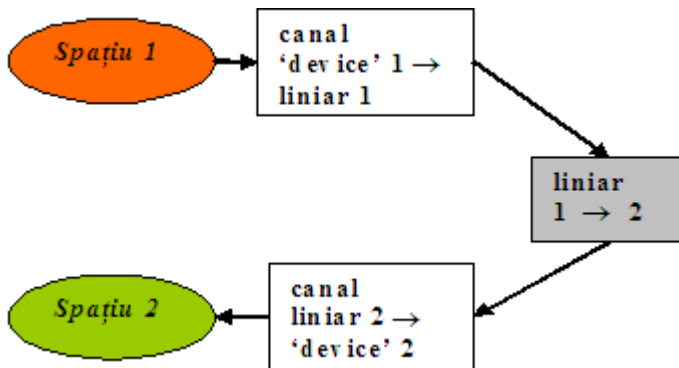


Fig. IV.1.2_1 Conversia de la spațiul de culoare 1 la spațiul de culoare 2

IV.1.2.2. Testarea acuratetei și vitezei de calcul. Surse de erori (puncte slabe)

Prin acuratețe înțelegem impresia perceptuală și reversibilitatea transformării, respectiv cât mai puține variații (erori) pe parcursul unei transformări dus-întors. e.g., dacă plecăm de la o imagine A în Adobe RGB și facem conversiile

$$A(\text{AdobeRGB}) \rightarrow S(\text{sRGB}) \rightarrow AA(\text{AdobeRGB})$$

AA trebuie să fie “cât mai” identică cu A.

Pentru eficiența și relevanța exprimării numim cele 3 blocuri din 2. IV.1.3_1: Cdev_lin, O1 (acest bloc operează cu numere din intervalul [0,1]) și Clin_dev. Mai mult, Cdev_lin și Clin_dev le pot numi gamma directă și inversă. Astfel, procesul unei conversii devine: gamma → O1 → gamma⁻¹. De asemenea, în loc de (R', G', B') echipament în spațiul 1, vom scrie RGBdev1, iar în loc de (R, G, B) liniar în spațiul 1, vom scrie RGBlin1.

În ce privește blocul 01, de conversie de la coordonatele liniare într-un spațiu de culoare la coordonatele liniare într-un alt spațiu de culoare, nu apar erori: într-o aplicație C optimizată erorile apar la a opta zecimal. Motivul e simplu: putem reduce toate operațiile care au loc în acest bloc la o înmulțire cu o matrice (3 x 3).

Erorile apar inerent pentru transformările gamma (directă și inversă), iar noi trebuie să interpolăm această transformare dată prin puncte, și nu printr-o formulă matematică, deși știm din teorie că transformarea (sau corecția) gamma e o funcție putere de forma $O = I^\gamma$, γ fiind coeficientul gamma, intrarea I și ieșirea O situându-se în intervalul [0, 1]. O valoare gamma $\gamma < 1$, micșorând prin transformarea gamma un interval de pe abscisă, e uneori numită ‘encoding gamma’. Decodarea corespunde unei transformări gamma inverse cu $\gamma > 1$. Codarea / compresia gamma se folosește pentru codarea luminanței liniare sau a celor RGB liniare în semnale video analogice sau digitale; decodarea / decompresia gamma apare “spontan” în tubul catodic (CRT) de care dispun monitoarele – putem spune că e o decodare hardware. Pe de altă parte, codarea gamma mapează datele (atât analogice cât și digitale) într-un domeniu mai uniform din punct de vedere perceptual.

Pentru un monitor CRT, se spune că γ e aproximativ 2.2. De fapt, γ variază între 1.8 și 2.5. Și, așa cum spuneam, un spațiu de culoare al unui echipament, γ se dă prin mai multe puncte de control, care pot diferi de la un producător la altul.

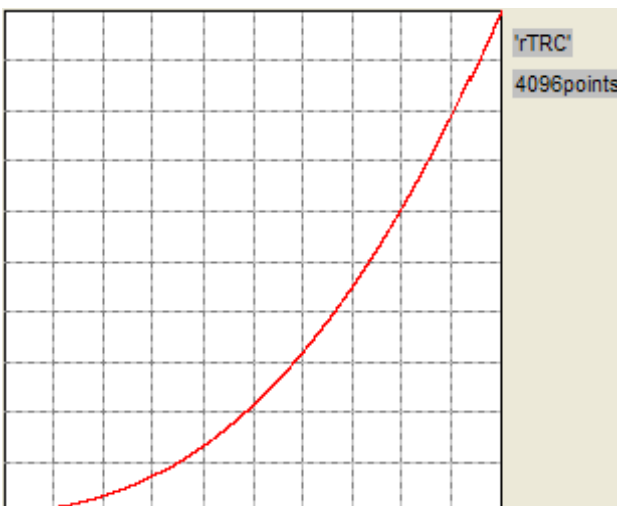


Fig. IV.1.2_2 Transformarea gamma pentru canalul R, în cazul spațiului de culoare sRGB de la Nikon[Nikon].

IV.1.2.3. Tratarea erorilor. Optimizări

Întrebarea principală care se pune în ce privește acuratețea conversiei e: cum să interpolăm între punctele de control care specifică transformarea gamma pentru a obține cea mai “exactă” transformare gamma? Conform teoriei, dar și graficului de mai sus, putem scrie:

$$y = x^{\text{exp}} \quad (5)$$

, unde x e valoarea unei componente R/G/B_{dispozitiv} normalizată la [0,1], y e R/G/B liniar și exp e un număr real. E o dependență nonlineară. Cum interpolăm? Interpolarea liniară ar fi cea mai bună metodă, prin simplitate și viteză de calcul. Ideea e să aplicăm logaritmul relației (5). Astfel, obținem o dependență liniară [Beu, 1999]:

$$\lg(y) = \text{exp} \cdot \lg(x) \quad (6)$$

Abordarea transformării gamma va fi:

1. Aplicăm logaritmul coordonatelor punctelor de control (la o intrare 0, ieșirea va fi 0: tratăm special acest caz pentru a ne asigura condițiile de existență pentru logaritmare):
2. Interpolează liniar pentru a calcula $\lg(\text{R/G/B liniar}) = \text{R/G/B} \cdot \text{exp}$
3. Calculează $\text{R/G/B liniar} = (\text{R/G/B} \cdot \text{exp})^2$ (folosim logaritmul în baza 2 pentru viteză de calcul – împărțirile la 2 se realizează ca deplasări la dreapta ale biților).

Toate acestea le facem într-o etapă de preprocesare - construim un LUT (‘lookup table’) pentru procesarea efectivă. Mai mult, să ne amintim că trecerea de la un RGB linear la altul (blocul 01) e, în fapt, înmulțire cu o matrice (3 x 3). Inserăm ȘI această informație (înmulțirea cu o constantă) în LUT. Astfel, pentru o acuratețe și o viteză de calcul mai mari avem

- pentru blocul gamma: lutTRC R/G/B : [0..255] \mapsto [0, 1] .

- pentru blocul 01: lutMatrix R/G/B : [0..255] \mapsto [0,1] . Acest LUT folosește lutTRC și precalculează termenii conform relației (3), astfel încât trecem un pixel din RGBdev1 în RGBlin2 prin trei operații de ‘lokup’ cu lutMatrix și două adunări.

- pentru blocul γ^{-1} folosim lutTRC^{inv} $[0,1] \mapsto [0..255]$.

Cum interpolăm pentru blocul γ^{-1} ? Avem de ales o singură valoare ieșirea corespunzătoare componentei R/G/B a unui pixel. Vom folosi această restricție și faptul că transformarea gamma (directă sau inversă) e monotonă. Astfel, calculăm la preprocesare conform coordonatelor punctelor de control (pentru acuratețe folosim aceeași abordare ca în cazul transformării gamma directe - logaritmare, interpolare liniară și ridicare la putere):

lutTRC^{inv} R/G/B: $[0, \frac{1}{2}, 1, \frac{3}{2}, 2, \dots, 255] \mapsto [0, 1]$, adică luăm și mijloacele intervalelor definite de

întregi în domeniul acestui LUT. Acest lucru ne va fi deosebit de util în a implementa o transformare gamma inversă eficientă. Fie x valoarea unei componente liniare R/G/B și, prin transformarea gamma inversă, vrem să aflăm valoarea componentei echipament $y \in [0..255]$. Folosim următoarea teoremă, valabilă (cu adaptarea inegalității) pentru orice f monotonă:

Teoremă: Fie o funcție $f : \mathfrak{X} \mapsto \mathfrak{X}$ monoton crescătoare. Pentru a decide dacă y e una din două valori consecutive pe ordonată (y_1 și $y_2 = y_1 + 1$), dându-se o valoare x pe abscisă, cea mai eficientă metodă e aceasta: $y > f(x_m) ? x_2 : x_1$, unde $x_m = x_1 + \frac{1}{2}$, i.e. alege x_2 dacă $y > f(x_m)$, altfel alege x_1 .

Funcția gamma e monoton crescătoare. Astfel, conform teoremei de mai sus, putem folosi un LUT (lutTRC⁻¹) și pentru blocul γ^{-1} . Operația de lookup, făcându-se pe un vector ordonat, are complexitatea $O(\lg N)$, aceasta fiind și complexitatea trecerii unui pixel dintr-un spațiu de culoare echipament într-altul.

Mai putem optimiza viteza de calcul folosind:

- operatori de deplasare pe biți pentru implementarea înmulțirilor și împărțirilor
- o abordare ‘run length’ – în general, într-o imagine sunt regiuni de mărime considerabilă în interiorul cărora pixelii au culori cel puțin similare (e o problemă foarte interesantă de clasificare) => într-un proces de conversie ar fi util să reținem valorile deja calculate. Există și CLUT (color look-up table): un mecanism folosit pentru a transforma un interval de culori în alt interval de culori. CLUT poate fi implementat hardware sau software.

IV.1.3 Testarea conversiei

Pentru testarea conversiei am folosit un ‘Color Checker’, la o rezoluție de 7 M px:



Fig. IV.1.3_1 și IV.1.3_2: prima imagine corespunde unei conversii Adobe RGB → sRGB cu logaritm (metoda noastră), următoarea corespunde aceleiași conversii cu metoda clasică (interpolare liniară)

Așa cum am spus la secțiunea IV.1.2.2 testăm acuratețea conversiei prin impresia perceptuală și reversibilitatea transformării, adică gradul în care apar cât mai puține variații (erori) pe parcursul unei transformări dus-întors între cele două spații de culoare. Impresie perceptuală pe care am încercat să o redăm în figura de mai sus, chiar dacă, restricționați de suportul acestei lucrări, prezentăm imagini nu numai comprimate de la formatul .bmp la formatul .jpg, dar și micșorate pentru a putea fi puse în pagină.

La o privire atentă, câteva diferențe între cele două imagini pot fi observate, în special, în regiunile unde negru e predominant, e.g. sub masă, la obiectele negre și în zonele de umbră. În conversia fără logaritm, pare că un praf foarte fin se întinde peste imagine. E predictibilă această aparență – iată un argument grafic:

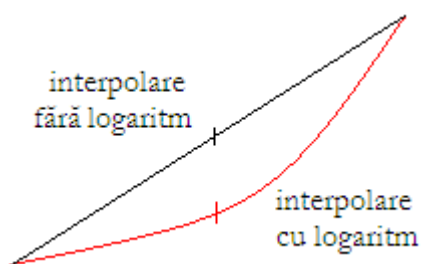


Fig. IV.1.3_3 Argument grafic al acurateții superioare a metodei noastre, dar și al aparenței imaginilor anterioare

Reversibilitatea transformării a fost măsurată în tabela de diferențe:

<i>DIFFs %</i>	<i>AAlog_A</i>	<i>AAdl_A</i>	<i>Sfoto_Slog</i>	<i>Sfoto_Sdll</i>	<i>Slog_Sdll</i>
0	0.2540932	73.9935169	77.5440351	77.6912546	64.4213062
1	0.0842800	0.0898927	24.5896225	36.5294016	0.4897142
2	0.0518853	0.0539477	2.8738575	5.5211811	0.0000000
3	0.0479098	0.0487633	1.2290542	1.1713816	0.0000000
4	0.0437978	0.048417	0.6488032	0.7836487	0.0000000
5	0.0167462	0.0228856	0.3099478	0.4948573	0.0000000
6	0.0004093	0.0014532	0.1009818	0.2982464	0.0000000
<i>Max difference</i>	34	36	52	59	8
<i>channel</i>	B	B	R	R	R
<i>(R,G,B)pixel</i>	(186,166,3)	(186,166,3)	(52,92,56)	(0,90,49)	(9,15,11)

Tabela arată diferențele între imaginile rezultate în urma procesării unei imagini inițiale: A.bmp = imaginea cu ColorChecker luată cu o cameră foto Nikon în spațiul de culoare Adobe RGB. Acesta e înțelesul lui ‘A’. ‘AA’ înseamnă imaginea trecută în sRGB (imaginea ‘S’) și înapoi în

Adobe RGB. ‘log’ se referă la conversia cu logaritm, ‘dll’ se referă la soft-ul Nikon pe care l-am avut la dispoziție pentru a avea un termen de comparație relevant, iar ‘foto’ se referă la o cameră Nikon performantă care putea achiziționa aceeași scena atât în sRGB, cât și în Adobe RGB.

Prima coloană indică:

- $\langle i \rangle$: = $i * 5$ prag pentru compararea imaginilor octet cu octet, e.g. ‘3’ se referă la diferențele peste 15. Diferențele sunt calculate ca procent din numărul total de octeți din imagine.

- ‘Max difference’: diferența maximă per octet între imagini

- ‘Channel’: componenta R/G/B pentru care am obținut ‘Max diff’

- ‘(R,G,B) pixel’: pixelul care conține ‘Channel’.

Cifrele din tabela arată nu numai că metoda noastră e comparabilă cu cea a lui Nikon – vezi coloana Sfoto_Sdll, coloană care mai arată că softul produs de Nikon nu interpolează în domeniul logaritm (și, știm, nici liniar) – ci și că, în unele privințe, chiar îi e superioară. Astfel, se confirmă impresia perceptuală că varianta ‘log’ e mai precisă, comparând coloanele ‘AAlog_A’ și ‘AAdll_A’ din tabelă, coloanele care numără erorile introduse de o conversie dus-întors. E interesant de observat că ‘(R,G,B) pixel’ e același pentru aceste coloane. Coloanele Sfoto_Slog și Sfoto_Sdll ne arată că, prin varianta noastră, ne apropiem mai bine decât însuși softul produs de Nikon de fotografia obținută cu camera Nikon, fotografie care beneficiază și de niște corecții suplimentare.

În ce privește viteza de calcul a conversiei, considerăm că am argumentat suficient. Mai afirmăm numai că, folosind LUT-uri, timpul de procesare se reduce de la minute la secunde.

IV.1.4 Concluzii

Termenul de comparație a metodei noastre a fost o metodă de conversie (între spații de culoare pentru echipamente) “de top” din industrie, anume cea folosită de firma Nikon, metodă a cărei implementare (cod executabil, dar și cod sursă) am avut-o la dispoziție. Am lucrat chiar cu

profilele de culoare sRGB și Adobe RGB pentru Nikon. Implementarea noastră a fost propusă firmei Nikon, nu știm ce s-a întâmplat mai departe, întrucât am schimbat serviciul. Avantajele majore ale metodei noastre sunt:

1. viteza de calcul superioară grație unei “lut-ificări” ingenioase, fără a folosi multă memorie suplimentară pentru aceasta. În fapt, construim în etapa de preprocesare următoarele 3 LUT:

$$\text{lutTRC } R/G/B : [0..255] \mapsto [0, 1]$$

$$\text{lutMatrix } R/G/B : [0..255] \mapsto [0, 1]$$

$$\text{lutTRC}^{-1} R/G/B : [0, \frac{1}{2}, 1, \frac{3}{2}, 2, \dots, 255] \mapsto [0, 1]$$

Prima și a treia sunt destinate transformării / corecției gamma, directe și, respectiv, inverse, într-un spațiu de culoare (pt.) echipament(e) – de fapt, transformarea care pune “la încercare” viteza de calcul și acuratețea metodei; cea de-a doua e destinată trecerii de la “coordonatele” liniare într-un spațiu de culoare echipament la “coordonatele” liniare într-un alt spațiu de culoare echipament.

“Lut-ificarea” e bazată atât pe faptul că reprezentarea componentelor unei culori se face într-un domeniu limitat, darși pe analiza pe componente a procesului conversiei între cele două spații de culoare. Pentru transformarea gamma inversă lut-ificarea” e bazată pe teorema pe care am enunțat-o la sfârșitul subsecțiunii IV.1.2.3.

2. acuratețe grație observării alurii graficului transformării gamma conform punctelor de control specificate de profilele de culoare și aplicării unei interpolări potrivite, adică a interpolării liniare în domeniul logaritmic – aceasta a accelerat ȘI viteza de calcul

3. generalitatea abordării: adresăm problema oricărei conversii între spații de culoare pentru echipamente. Așa cum am subliniat mai sus, zona slabă a unei conversii între două spații de culoare, transformarea gamma o tratăm într-o manieră care se pretează și la alte transformări specificate prin puncte de control, ideea fiind că interpolăm conform alurii curbei transformării, acest fapt dându-ne posibilitatea să avem performanțe atât în ceea ce privește acuratețea, cât și viteza de calcul.

Chiar și procesul de “lut-ificare” poate fi generalizat – de aceea am simțit nevoia să precizez la sfârșitul subsecțiunii IV.1.2.3. acea teoremă.

IV.2 Definirea formei spațiului unei probleme de recunoaștere nedeterminate și modelarea acestuia. Cazul ‘skin detection’

Chiar dacă vom trata problema specifică a determinării eficiente a unui “spectru” de culori conform căruia să putem detecta zonele de piele (umană) dintr-o imagine digitală, adică, ne propunem să determinăm un ‘skin locus’ (SL) în vederea detecției zonelor de piele (‘skin detection’: SD), metoda noastră de extragere maximală (în sensul eficienței menționate) a informației din date poate fi generalizată și pentru alte probleme nedeterminate de clasificare în imagini. Această “extragere maximală” presupune nu numai contorizarea pixelilor care aparțin unei clase (în cazul nostru, clasa pixelilor de skin (S)) și, mai departe, stabilirea unei măsuri probabilistice de a fi ‘skin pixel’ (SP) pentru fiecare pixel dintr-un anumit spațiu de culoare. Se rețin și condițiile de achiziție a imaginilor. Vom stabili o taxonomie a bazei de imagini de calibrare și după tipul și setările camerei foto. Astfel, vom furniza un cadru de lucru adaptiv și dinamic, capabil să îmbunătățească orice metodă de clasificare bazată pe clasificarea pixelilor. Această flexibilitate plus cele două ingrediente majore în industrie, acuratețea și viteza de calcul, vor configura față cu ‘the state of the art’-ul existent o metaeuristică mai eficientă de determinare a formei spațiului unei probleme de recunoaștere nedeterminate - vezi și [Georgescu, 2010]. Tot în acest subcapitol vom trata și problema modelării acestui spațiu - vezi și [Georgescu, 2011].

În particular SD joacă un rol important într-o categorie largă de aplicații de procesare a imaginilor, cum ar fi:

- detecția feței și / sau urmărirea feței (‘face tracking’) / persoanei
- analiza mimicii

- detecția mâinilor pt.: recunoașterea gesturilor, controlul robotic și diverse “interacțiuni” om-calculator
- filtrarea conținutului pornografic de pe internet
- găsirea imaginilor pe baza conținutului (‘content-based image retrieval systems’) etc.

IV.2.1. Stadiul actual și problematica metodelor de detecție a pielii

Așa cum e arătat într-un articol recapitulativ, [Kakumanu et al., 2006] “recent, metodele SD bazate pe informația culorii pielii au câștigat o mare atenție din punct de vedere al efortului computațional, al robusteții la rotații și scalări, și ocluzii parțiale”. Aceste metode s-au dovedit a fi optime în termeni de robustețe și, vom vedea, în cazul metodei noastre, modularitate.

Scopul nostru e de a găsi cel mai “generos” cadru de lucru pentru aceste metode - stabilim o metaheuristică care permite înglobarea eficientă (‘plug and play addition’) într-o metodă de SD a oricărui criteriu nou (e.g. un criteriu de mișcare, sau un criteriu de vecinătate, cum ar fi textura pielii, sau prototipuri de culori ale pielii, sau caracteristici de formă pentru fețe / ochi), în vederea creșterii selectivității metodei (reducerii fals pozitivelor). E o metodă dinamică de detecție (bazată pe pixeli) și în sensul că putem îngloba informații noi pentru detecție prin adăugarea la Skin Locus-ul existent de noi pixeli de piele, sau prin modificarea informației de probabilitate (vezi mai jos) pentru pixeli din SL-ul existent. Dinamicitatea metodei permite și calibrarea ei, e.g., putem calibra metoda pentru detecția pielii în imagini cu persoane de rasă caucaziană. Amintim că, în literatura de specialitate, o metodă de SD ‘pixel-based’ clasifică fiecare pixel ca fiind de piele sau nu, independent de vecinii săi; prin contrast, o metodă ‘region-based’ ar lua în considerare aranjamentul spațial al zonelor cu piele [Chang & Zhiqiang, 2007] – acesta e doar o taxonomie folosită pentru a clarifica principiile. Pentru a “profita” cât mai mult de pe urma oricărei informații, vom avea în vedere că SD trebuie să țină cont și de caracteristicile de conectivitate a pixelilor de piele – în faza de modelare a SL vom face prelucrări morfologice corespunzătoare asupra SL-ului obținut anterior prin culegerea informației pixelilor. Această

“intuiție” topologică, pusă în serviciul metodei noastre, împreună cu informația de probabilitate asociată fiecărui SP ne situează în domeniul tehnicilor fuzzy.

O cameră digitală (Charge-Coupled Device) color poate fi descrisă ca un filtru care trece semnale continue de culoare într-o reprezentare cu 3 componente (‘Red, Green and Blue’), fiecare componentă având o valoare într-un interval limitat. Această compresie a spectrului generează metamerism – vezi subsecțiunea III.1.2.1. (‘Metamerismul sau granularitatea spectrului vizibil’). În general, ne-linearitatea camerei (foto) e cauzată de conversia intrare-ieșire conversie care depinde ȘI de semnalul de intrare. Deși în formarea unei imagini color factorii principali sunt distribuția spectrului de putere al iluminantului, senzitivitățile spectrale ale camerei și reflectanța suprafețelor, de altfel, factori foarte greu de precizat, sunt mulți alți factori care pot avea un efect esențial: geometria (scenei și poziția camerei față de aceasta, umbre etc.), mediul înconjurător, tipul și setările camerei, electronica și alte non-idealități ale camerei.

Toate aceste aspecte reale ne conduc la o abordare centrată pe datele experimentale, “căzând” abordările care încearcă să deducă ecuații matematice, prin care am putea modela spațiul soluției problemei. De altfel, noi, ca și alți cercetători, am experimentat faptul că o asemenea abordare e mai eficientă decât a încerca să “prindem” SL în formule explicite.

SD are o istorie însemnată, și, în ultima decadă, au fost publicate unele articole recapitulative și comparative privind diferite tehnici [Gonzalez et al. 2010]. Metaeuristica pe care o propunem e un mixaj între câteva metode populare de SD, încercând să luăm ce e mai bun din acestea din punct de vedere al eficienței reclamate de industria procesării imaginilor. În același timp, în ideea modularității, dar și a programării orientate pe obiecte, pentru a ajunge la implementarea de SD, urmăm o succesiune de etape general acceptată – conform [Vezhnevets et al., 2003] apar 3 mari subprobleme în construcția unui sistem de SD:

1. ce spațiu de culoare să alegem
2. cum vom modela distribuția culorilor de piele (‘Skin Pixels’) în acest spațiu
3. algoritmul efectiv de SD (ce se întâmplă la preprocesarea unei imagini)

IV.2.2. Alegerea spațiului de culoare și colectarea pixelilor de piele

IV.2.2.1 Alegerea spațiului de culoare

Deși în literatura de specialitate există puncte de vedere conform cărora alegerea spațiului de culoare nu are legătură cu performanța unei SD, mai multe studii raportând o rată de 95% pentru detecția corectă și o rată între 15% și 30% pentru detecția incorectă, nu credem în generalitatea acestor evaluări. Suntem de acord însă cu faptul că, pentru orice SD, e mai puțin probabil să ratăm un pixel de piele(SP), decât să confundăm non-SP cu SP.

Simțul comun ne sugerează că apariția pielii în imagini poate fi mai degrabă controlată de cromaticitate decât de luminanță: luminanța poate varia într-un interval mult mai larg decât cromaticitatea. Multe lucrări de determinare a SL într-un spațiu de culoare au renunțat la componenta de luminanță. Această *reducere a dimensionalității* SL (de la 3D la 2D) a eficientizat foarte mult procesarea ulterioară. În plus, SL-ul 2D se poate “generaliza” la unul 3D, asigurându-ne astfel de reducerea fals negativelor – un deziderat principal, având în vedere că, de multe ori SD are loc ca etapă intermediară altei probleme, e.g. de reducere a ochilor roșii în imagini (vezi subcapitolul următor). O decorelare cât mai bună între cromaticitate și luminanță devine foarte utilă. E unul dintre motivele principale pentru care am ales spațiul de culoare YCbCr pentru metoda noastră de SD. YCbCr e un spațiu de culoare care se mapează foarte bine și pe cerințele industriei (avem în vedere și simplitatea transformării RGB ↔ YCbCr) procesării imaginilor, dar și pe cerințele industriei procesării semnalelor video

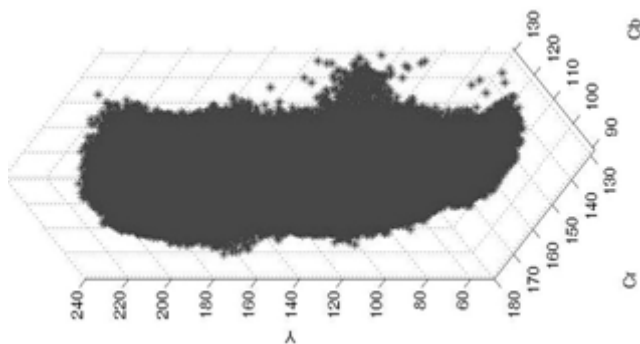
YCbCr e un spațiu de culoare foarte popular în ‘pixel based SD’ [Gonzales & al., 2010]. YCgCr, o variantă a lui YCbCr, a fost “raportat” că “produce” un SL mai compact [deDios & Garcia, 2003]. Acest spațiu de culoare folosește o diferență de culoare (G-Y) mai mică decât (B-Y) – cum se face în general pentru a minimiza erorile de codare-decodare – e, totuși, definit exclusiv pentru aplicațiile de analiză de imagini, mai ales pentru segmentarea fețelor. Oricum, alegerea ca spațiu de culoare YCgCr față de YCbCr nu ar schimba semnificativ considerațiile noastre.

IV.2.2.2. Colectarea pixelilor de piele

Pixelii de piele (SP) îi vom culege dintr-o bază cu fotografii, în care fețele au fost marcate (selectate) prin poligoane, iar benzile (dreptunghiurile) gurii și ochilor au fost îndepărtate. Marcarea nu trebuie să fie “foarte” exactă. Punând unele praguri pe ordonata graficului histogramei cumulative a luminanței putem elimina din selecție culorile minoritare, care nu pot fi luate drept culori de piele, e.g. fire de păr negre (prag inferior) sau albul ochilor (prag inferior). Aceste praguri le punem procentual – între 5 și 10%.

Reținem pentru fiecare SP: Y, Cb, Cr numărul de apariții (**informație de probabilitate**).

Informația de probabilitate, în fapt frecvența relativă a SP, ne dă posibilitatea construirii unui cadru “generos”, cu posibilități de extindere și înglobare a diverse informații, nu numai de natură probabilistică, dar și fuzzy, cu toate avantajele lor menționate la capitolul “Domeniul impreciziei”.



Să observăm forma Skin Locus-ului.

Fig. IV.2.2_1 Reprezentarea 3D of a SL - în spațiul YcbCr - conform [Georgescu, 2010]

E important de avut în vedere că lucrăm cu baze de date de imagini cât mai mari, mai complete, dar și mai “specializate”. Acesta credem că e bine să poarte numele de **principiul specializării și al calibrării**: vom lucra cu seturi de imagini care au fost achiziționate în diverse condiții, cu diverse tipuri și parametri de camere foto, cu persoane de diverse etnii și, deci, culori ale pielii ș.a.m.d.. De fapt, cu factori care ar putea fi știuți înaintea procesării efective a imaginii pentru SD, astfel încât să putem alege setul corespunzător. Dacă un om recunoaște foarte ușor o zonă de piele, e nu numai datorită capacităților încă neexplicate ale sistemului său nervos, dar și datorită faptului că această clasificare o face dispunând și de o informație 3D și de multe alte informații. Putem elabora un model de calcul satisfăcător al SD numai dacă ne delimităm cât mai bine

condițiile în care clasificăm. Clasificarea pielii la modul general, e nu numai o utopie tipică, dar și fără logică.

Astfel, într-un mod realist, în acest stadiu am obținut informații probabilistice despre subspațiul culorilor pielii în spațiul de culoare YCbCr, informații pe care le vom prelucra în continuare.

IV.2.3. Reprezentarea Skin Locus-ului în vederea eficienței Skin Detection

E momentul să tranșăm în vederea eficienței SD, mai exact a aplicabilității SD în timp real: observăm că reprezentarea 3D de la Fig. IV.2.2_1 seamănă destul de mult cu un cilindru. Se confirmă percepția de la subsecțiunea IV.2.2.1.: pentru componenta de luminanță putem reține doar două praguri globale: unul inferior și unul superior. Astfel, pentru un SP vom reține coordonata Cb, coordonata Cr și numărul de apariții. SL devine:

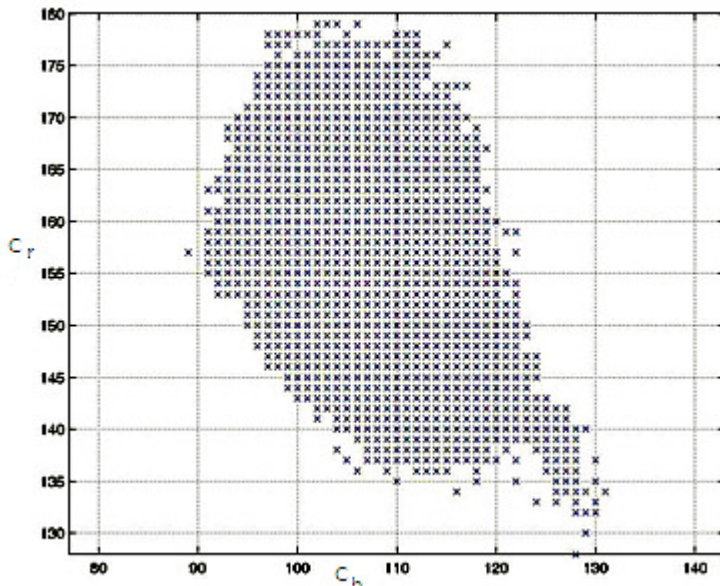


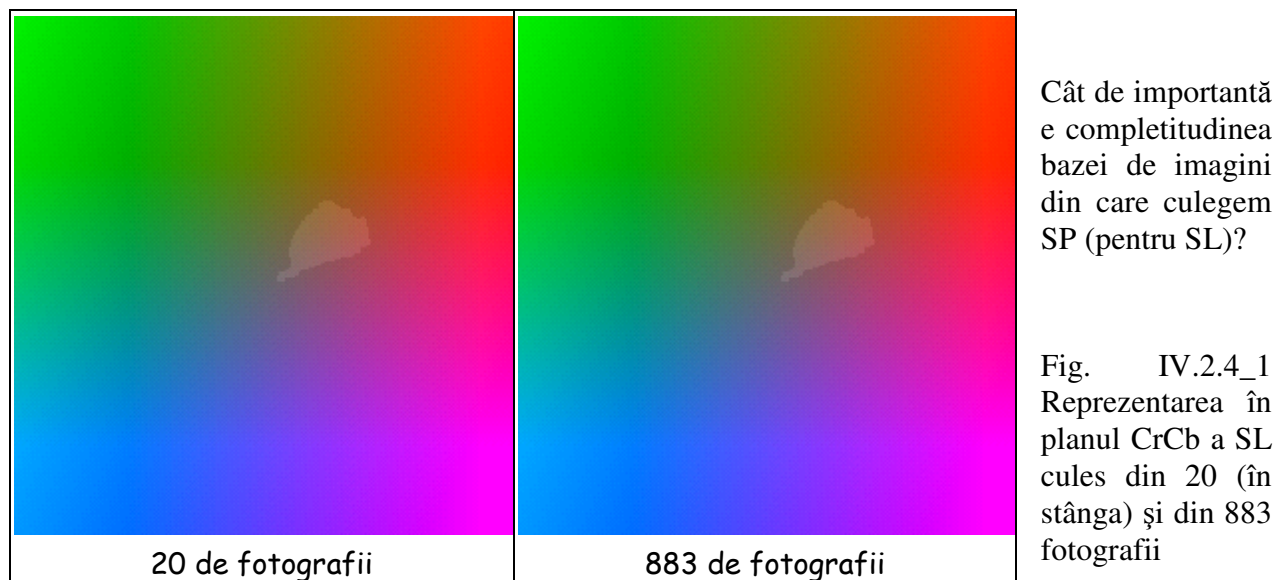
Fig. IV.2.3_1 Reprezentarea 2D a Skin Locus-ului - în planul CrCb conform [Georgescu, 2010]

Se confirmă o altă intuiție: pielea are o aparență de roșu și mai puțin de albastru (cu excepția unor cazuri de iluminare florescentă) - componenta Cr ar merita o “rezoluție” mai mare decât Cb într-o reprezentare a SL în planul CrCb.

Să avem în vedere că proiecția în planul CrCb a SL amestecă clasa pixelilor “de” piele SP, cu clasa non-SP. Cu atât mai importantă devine informația de probabilitate, care ne ajută să partiționăm cât se poate (fuzzy) în cele două clase.

IV.2.4. Modelarea distribuției SL

IV.2.4.1 Informații despre forma și probabilitate pentru un SL. Ideea algoritmului nostru



Cele două forme de SL de mai sus au câteva asemănări importante:

- sunt regiuni relativ compacte
- acoperă cam aceleași intervale Cb și Cr
- apar similare nu numai topologic, dar și ca formă

Aceste asemănări majore ne duc la două concluzii importante:

C1) nu avem nevoie de foarte multe imagini pentru a configura / culege un Skin Locus (mai degrabă avem nevoie de câteva scene fotografiate în diverse situații (câteva fețe, acestea

conținând mulți SP), ceea ce nu înseamnă abandonarea principiului specializării și al calibrării, ci doar că, între anumite limite putem interpola.

C2) SL e relativ **compact**

Să coroborăm aceste informații despre formă cu cele de probabilitate.

Să coroborăm aceste informații despre formă cu cele de probabilitate.

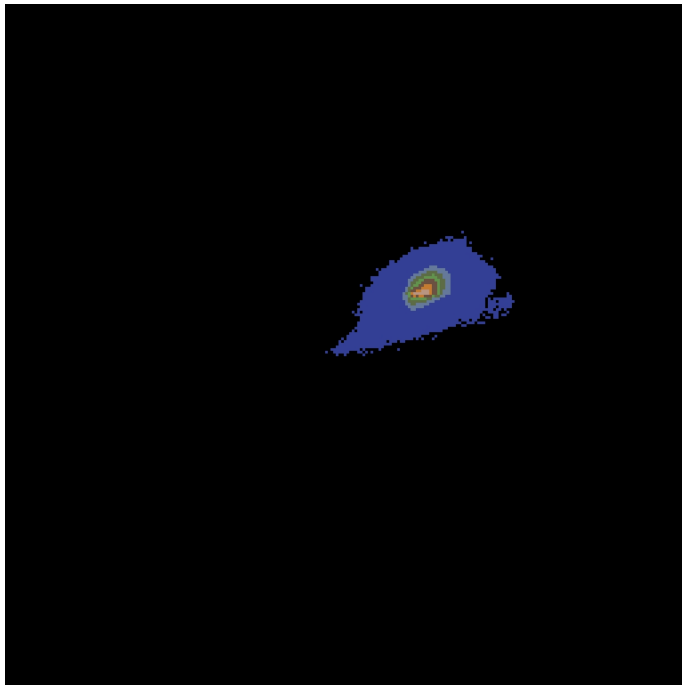


Fig. IV.2.4_2 Nivele de probabilitate ale SL (cules din 883 fotografii) în planul CrCb. Am reprezentat șapte nivele de probabilitate pseudocolorate (după o convenție asemănătoare cu cea pentru relief, adică albastru închis corespunde SP cu cea mai mică probabilitate, iar cea mai deschisă culoare corespunde SP cu cea mai mare probabilitate).

Observăm **dispunerea concentrică** a nivelelor de probabilitate ale SL, în centru fiind pixelii cu cea mai mare probabilitate. Această observație împreună cu C1) și C2) de mai sus ne face să ne gândim la un SL compact, care poate fi partiționat în componente convexe și conexe, foarte ușor de parcurs de un **algoritm de SD de tip ‘run-**

length’ (‘bin’) – cel mai rapid algoritm pentru problema noastră: testul dacă un pixel de coordonate [cr] [cb] e SP se reduce la două comparații (testăm dacă un punct aparține unui segment). Pentru a compacta adecvat SL, în ideea de a aplica un astfel de algoritm, putem aplica prelucrări morfologice, ținând cont și de informația de probabilitate.

IV.2.4.2 Scurt breviar teoretic al metodelor de modelare a SL și comparația acestora cu metoda noastră

Vom rezuma câteva din principiile relevante ale metodelor de SD pentru a preciza metoda noastră în contextul ‘state-of-art’-ului curent. Restrângem analiza la ‘pixel based methods’, numite, în literatura de specialitate și ‘pixel based methods’. Așa cum am menționat în secțiunea IV.2.1., aceste metode sunt apreciate. din punct de vedere al efortului computațional, al

robusteții la rotații, scalări, ocluzii parțiale, de fapt, la orice alte “deformări” geometrice, eliminând și nevoia urmăririi unor puncte / zone cheie ale formei. În ce privește ‘region based methods’, vom spune numai că pentru o mai bună selectivitate a pielii (mai puține fals pozitive), am putea adăuga un criteriu de vecinătate, de textură sau de formă geometrică pentru segmentarea obținută de SD.

Procesarea culorilor fiind mult mai rapidă decât orice altă caracteristică facială, ea poate fi folosită ca etapă preliminară pentru orice tehnică de detecție a feței, ca să luăm una din cele mai populare aplicații de analiză a imaginii [Hsu et al., 2002]. Uneori se poate proceda invers. [Zheng et al., 2004] au prezentat un SD adaptiv pentru detecția imaginilor oamenilor dezbrăcați. Într-o primă etapă tehnica lor folosea un detector de fețe – argumentația lor era că aceste culori depind într-un grad mare de iluminantul scenei și de rasa persoanei, astfel încât e mai potrivit să găsim întâi fețele și apoi, pe baza culorilor și texturilor “de” piele din fețe, să determinăm celelalte zone de piele.

Modelarea parametrică a distribuției SL

Aceste metode prezintă avantajul că pot interpola și generaliza datele de antrenament, adică semnalele / imaginile prin care sistemul de SD e învățat să recunoască zonele de piele. Un alt avantaj e că necesită un spațiu de stocare minimal. Dezavantajul major e că modelarea parametrică e greu de justificat. [Yang et al., 1999] arată că distribuția SL în spațiul de culoare RGB normalizat, pentru o singură persoană și într-o scenă cu iluminare fixă e o distribuție gaussiană. Dar foarte puțini autori răspund la asemenea chestiuni.

!! În general, modelarea parametrică e potrivită pentru construcția clasificatorilor când avem un set redus de date de antrenament.

Modelarea neparametrică a distribuției SL

Aceste metode nu presupun un model explicit pentru distribuția SL. Conform [Gomez, 2000] pe baza datelor de antrenament se poate defini o hartă probabilistică a pielii (SMP: ‘Skin Probability Map’), atribuind o probabilitate (de a fi SP) fiecărui pixel.

Variante SMP:

* ‘Lookup table’ (LUT) sau ‘Histogram Model’: în principiu e tipul de metodă care se pretează la o implementare cu ‘bin’-uri – algoritmul nostru se bazează pe acest principiu, fiind însă mai avansat prin prelucrările pe care le face pe baza informațiilor de probabilitate

* Clasificare Bayes: clasificarea SP / non-SP poate fi definită probabilistic astfel: dându-se un pixel cu culoarea c , să se afle probabilitatea ca acesta să fie un SP: $P(\text{skin}|c)$. Odată ce această probabilitate e calculată, pixelul e clasificat ca SP dacă ea e peste un prag, altfel pixelul e clasificat ca non-SP. Desigur, nu putem calcula probabilitatea pentru orice culoare posibilă (e.g., pentru o reprezentare pe 24 de biți am avea de calculat 2563 culori). Din fericire, folosind regula Bayes putem scrie:

$$P(\text{skin}|c) = P(c|\text{skin}) P(\text{skin}) / P(c)$$

$P(c)$ nu afectează clasificarea specifică de SP, iar $P(\text{skin})$ se poate estima. Conform regulii Bayes, rămâne să calculăm termenul de asemănare (‘likelihood’) $P(c|\text{skin})$ – ceea ce se face calculând densitatea de probabilitate conform unei baze de SP-i. Pentru calculul acestei densități se pot folosi histograme [Jones & Rehg, 2002], un model de estimare gaussian simplu sau mixat [Yang & Ahuja, 1999].

* ‘Self Organizing Map’ – proiectată de Kohonen în anii 80, a devenit una din cele mai populare tehnici de învățare nesupervizată.

SPM de tip Bayes a fost raportată în [Vezhnevets et al., 2003] ca având cele mai bune performanțe în sensul minimizării fals pozitivelor pentru o rată de detecție corect dată.

Pentru majoritatea metodelor neparametrice, mai mulți cercetători au remarcat un grad înalt de independență a performanțelor acestor metode față de alegerea spațiului de culoare. Mai mult, ca rezultat teoretic, [Albiol et al., 2001] au arătat că pentru o SD optimă $D(x)$ într-un spațiu de culoare C , și pentru o transformare ce se poate inversa între spații de culoare $T:C \rightarrow C1$, există un clasificator $D1(x)$ în $C1$, cu aceeași rată de fals pozitive și fals negative ca a lui $D(x)$ în C .

Fiind independente de forma SL, luând în considerare suprapunerea claselor SP și non-SP și construind automat reguli de clasificare - vezi [Gomez & Morales, 2002], unele din aceste

metode par foarte promițătoare pentru o metodă de SD antrenată conform unei baze de date mari.

Modelarea explicită a distribuției SL

De exemplu, putem defini SL ca un set de inegalități. Pornind de la culoarea “roșiatică” a pielii, într-o aplicație industrială, firma Nikon folosea, ca pas intermediar detecției feței, SD conform inegalităților:

$$R > G + k_1 \text{ and } R > B + k_2 \quad (1)$$

, k_1 și k_2 depinzând de saturație. Această SD a fost raportată ca fiind foarte rapidă – metoda noastră e superioară atât ca viteză de calcul cât și ca selectivitate. Pentru a compara selectivitatea am tradus relațiile în YCbCr:

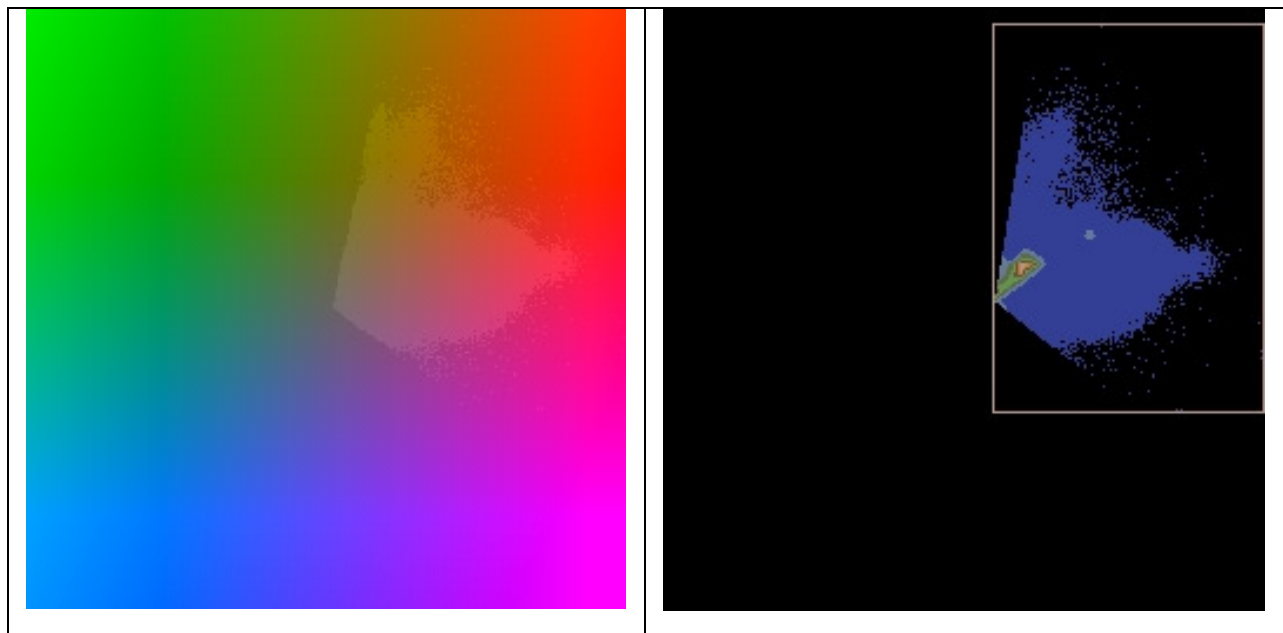


Fig. IV.2.4_3 SL definit de relațiile (1) e semnificativ mai mare decât cel definit de noi – vezi Fig. IV.2.4_1

Încercarea de a defini SL prin inegalități a dus la un rezultat interesant: SL ia forma unui dreptunghi în planul de culoare ($\log(R/G)$, $\log(RG/B^2)$), SD înseamnând comparațiile:

$$R > 0.97G \text{ and } R < 3G \text{ and } RG < 36B^2 \text{ and } RG > 0.7B^2 \text{ and } RG^2 > B^2(3.5R - 7G)$$

Pentru luminanță și în vederea eliminării culorilor gri s-au pus condițiile:

$$0.3R + 0.6G + 0.1B > 35 \text{ and } \max(R, G, B) - \min(R, G, B) > 15 \quad (2)$$

Această modelare a fost raportată ca fiind performantă[referinta], totuși, fiind inferior ca performanțe modelării noastre – vom reveni asupra modelului dat de (2) la subsecțiunea IV.3.2.2, “Stabilirea prototipurilor”.

Dezavantajul major al acestor modelări e selectivitatea redusă: să observăm în Fig. IV.2.4_3 că, după relația (1), sunt clasificați ca pixeli de piele și pixeli prea roșii sau roz. În Fig. IV.2.4_3 observăm că, până și în condițiile unui SL cu dispersie mare putem defini un *dreptunghi încadrator*, ceea ce arată gruparea SP-ilor. Mai observăm că probabilitatea unui SP nu crește spre centru – ceea ce e contrar percepției noastre.

După ce am “înconjurat” problema SD din perspectiva metodelor posibile / existente de SD, vom prezenta în detaliu modelarea noastră. Avem de compactat SL reprezentat în Fig. IV.2.4_2. După cum am spus în încheierea subsecțiunii IV.2.4.1, avem două obiective: conectivitatea și convexitatea. Mai întâi trebuie asigurată conectivitatea.

IV.2.4.3. Conectarea SL

SL reprezentat în Fig. IV.2.4_2 prezintă la margine pixeli izolați care ar putea fi “zgomot de piele”, adică pixeli care apar accidental ca fiind “de” piele. Putem înlătura acești pixeli prin câteva prelucrări morfologice simple, cum ar fi închiderea cu un mic element structurant. Putem prelucra și mai eficient după regulile:

- 1) doi pixeli de același tip (SP sau non-SP) “forțază” pixelul aflat direct între ei la tipul lor; probabilitatea în cazul unui SP creat astfel, fiind media aritmetică a vecinilor care l-au “forțat” să apară
- 2) un SP e făcut non-SP, dacă nu are cel puțin doi vecini

Aplicăm prelucrările 1) și 2) până obținem o singură regiune conexă pentru SL: CN. În cazul nostru, după doar două iterații ale acestor prelucrări, obținem un SL conex și aproape convex – ne punem acum problema obținerii convexității. Putem ajusta local SL (fie și manual). Sau putem aplica o transformare morfologică de tip *acoperire convexă* ('convex hull'). **Dacă obținem un CN cu concavități accentuate, îl partiționăm în componente convexe**; altfel, vom pierde mult din selectivitatea metodei de SD.

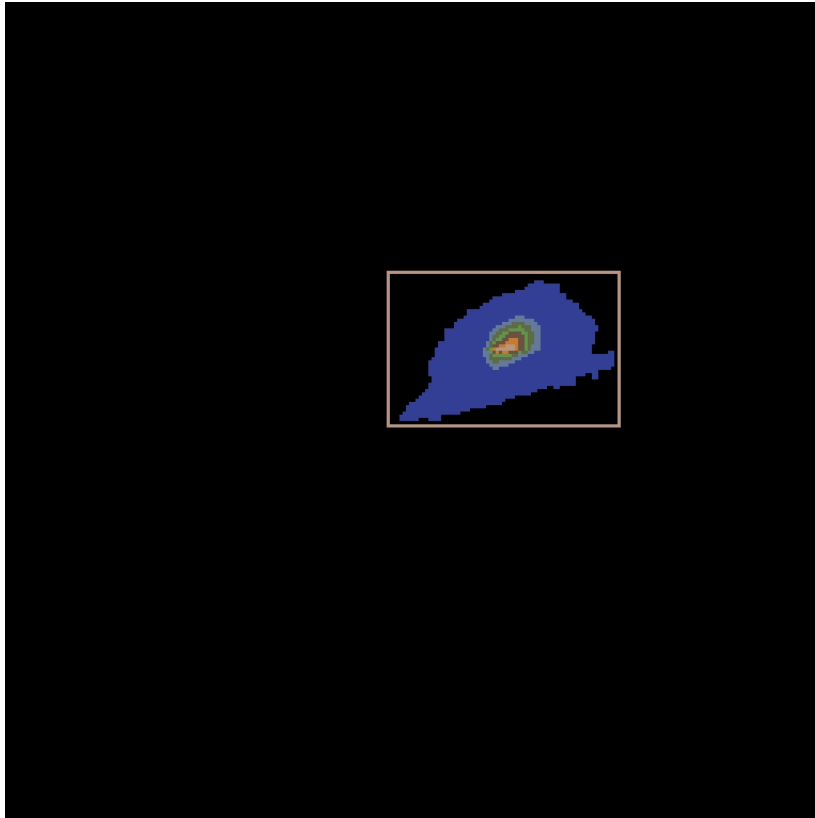


Fig. IV.2.4_4 Rezultatul aplicării pașilor 1) și 2) e un SL conex. În figură am reprezentat și dreptunghiul încadrator al SL înainte de a aplica 1) și 2)

IV.2.4.4. Codificarea SL

O componentă convexă și conexă (a SL) poate fi reprezentată într-o LUT, de fapt, într-o matrice cu două coloane și un număr de linii egal cu lungimea intervalului C_b pentru SL. (care e mai scurt decât intervalul C_r). În cele două coloane se pun capetele segmentelor orizontale care dau SL.

IV.2.5 Concluzii

Am abordat dintr-o perspectivă *experimentală, destinată să valorifice optimal informația care poate fi extrasă dintr-o bază de date*, proiectarea unei metaheuristici care să clasifice, în caz concret, pixelii “de” piele în imagini digitale.

Avantajele metodei noastre constau, așa cum am arătat pe parcursul acestui subcapitol, în:

- modularitate – etapa de calibrare înglobează dinamic noi informații și se desfășoară decuplată de etapa de procesare efectivă
- viteză de calcul – prin Lut-ificare, compactare astfel încât să rulăm un algoritm ‘run-length’ pe componente conexe și convexe
- acuratețe grație unei adaptabilități superioare conform a ceea ce am numit “principiul specializării și al calibrării”
- generalitate – am proiectat de fapt o metauristică
- deschidere – prin informația de probabilitate ne putem extinde metoda în domeniul “impreciziei”, adică putem îngloba cunoștințe probabilistice sau fuzzy (lingvistice, perceptuale).

IV.3 Folosirea subclasării și a marcajelor pentru reducerea fals pozitivelor. Cazul ‘skin detection’

IV.3.1 Abordare

Se pune problema creșterii selectivității unei metode de clasificare, adică a reducerii fals pozitivelor (FP). Ideea principală, pe care se bazează contribuția noastră, e aceea de a împărți / **partiționa clasele furnizate de metoda de clasificare dată (inițială, de intrare) în subclase găsiind condiții suplimentare de rafinare a clasificării în interiorul fiecărei clase.** Pentru a ne asigura de corectitudinea acestor condiții suplimentare vom folosi informații perceptuale, adică **marcaje de clasificare furnizate de expertul uman** – vezi și [Georgescu 1, 2012].

Dacă numim prototipuri ale unei clase, subclasele obținute în interiorul fiecărei clase prin partiționare, schema de selecție e dată de relația:

$$\mathit{zona_de_interes} = \mathit{zona_prototipurilor} \cap \mathit{zona_marcajelor}$$

Pentru punerea în lumină a acestei metode de creștere a selectivității unei clasificări, vom analiza cu date experimentale, cazul detecției zonelor de piele (Skin Detection) în fotografii – ceea ce corespunde faptului că pornim de la o clasificare cu două clase: ‘skin’ și ‘non-skin’.

Condițiile suplimentare le punem pentru a subclasa în interiorul clasei ‘skin’. Zona corespunzătoare clasei ‘non-skin’ nu ne mai interesează întrucât se consideră că, pentru metoda SD dată – acesta e cazul general, fals negativele nu creează probleme, fiind neglijabile. În general, problema majoră pentru SD e dată de faptul că se detectează prea multe fals pozitive, adică se detectează ca ‘skin’ și zone care în fapt nu sunt de ‘skin’.

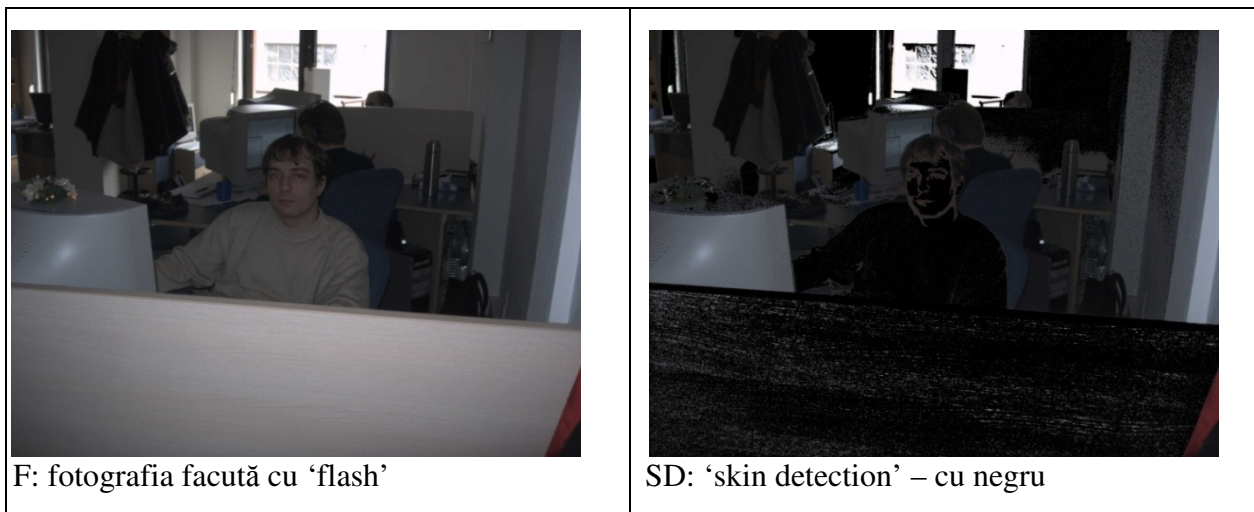


Fig. IV.3.1_1 O metodă tipică de detecție a zonelor de piele

Observație: (în legătură cu situația figurată mai sus): În mediul înconjurător, mai ales în interioare, există multe culori cu o cromaticitate ca a pielii – vezi biroul, puloverul, pereții etc.

Pentru luminanță s-a impus un anumit interval, un prag superior și unul inferior – de aceea nu e selectată zona de piele (S) din depărtare. Dacă ne interesează să selectăm numai S din plan apropiat, mai precis, cu luminanța peste un anumit prag, abordarea e corectă. E și cazul nostru, **întrucât am folosit SD ca pas intermediar pentru problema corecției ochilor roșii** ('Red Eyes Reduction') – e improbabil să obținem ochi roșii în depărtare, deoarece 'flash'-ul, cauza apariției ochilor roșii, are o rază de acțiune destul de mică.

A se vedea influența mare a direcției 'flash'-ului: dacă 'flash'-ul ar fi fost "pe" fața din prim plan, această față ar fi fost selectată mai compact – în situația figurată mai sus, pentru o selecție de zone S conexe și convexe, cum ne așteptăm să fie zonele de piele, ca și în cazul metodei de determinare a SL-ului (vezi subcapitolul anterior), pot fi utile anumite prelucrări morfologice. Remarcăm și că o **mulțime de alți factori apar odată cu 'flash'-ul**, fie și numai dacă menționăm numai artefacte ca umbre sau specularități.

Unul dintre scopurile principale ale proiectului la care am lucrat era acela de a crește selectivitatea unei metode de SD bazată pe clasificarea pixelilor (ca fiind de 'skin' sau nu), luând în calcul și informații legate de diferența de luminanță Dif între o fotografie făcută cu 'flash' și una fără 'flash', asupra aceleiași scene, ideea fiind că în zona feței (zona S pe care îmi propun să o selectez cu cât mai puține FP) Dif are o valoare specifică pentru reflectanța pielii (din acea zona a feței). Softul care permitea achiziția fotografiei fără 'flash' era un firmware Nikon. În

fapt, fotografia fără ‘flash’ e un ‘preview’ (P), adică acea imagine care se formează în timp real pe vizorul camerelor foto digitale de generație mai nouă. Pentru concizia exprimării vom folosi F în loc de “fotografia făcută cu ‘flash’”.

Ne vom raporta în continuarea subcapitolului la relația:

Dif (diferența pe canalul de luminanță) $\stackrel{\text{def}}{=} F$ (luminanța fotografiei cu ‘flash’) – **P** (luminanța fotografiei fără ‘flash’)

IV.3.2 Clasificarea skin / non skin

IV.3.2.1 Alegerea spațiului de culoare

Dacă în subcapitolul anterior am ales pentru modelarea culorii pielii, din considerentele menționate (inclusiv acela de efort computațional), spațiul de culoare YCbCr, de data aceasta vom alege, preferând acuratețea vitezei de calcul, spațiul de culoare Lab. În fapt, o comparație arată că aceste spații se comportă în mare măsură asemănător. Astfel, pentru simplitate, am considerat un exemplu semnificativ de pixeli “de piele” S pe baza căruia am definit un model gaussian mixat. Conform [Yang & Ahuja, 1999], am adoptat modelarea S prin două distribuții 2-D gaussiene, fiecare parametrizate de $(\mu_i, \sigma_i^2 I)$, ceea ce se justifică statistic. Mediile μ_i se calculează luând centrele celor două clustere de culori de piele, iar varianțele sunt dispersiile în interiorul acestor clustere.

Observație: Așa cum vom vedea în figura de mai jos, cromatica pixelilor de piele se grupează în două clustere, dintre care unul e mult mai mare decât primul. Dacă în subcapitolul anterior am

ales spațiul de culoare YCbCr și, sub imperativul vitezei de calcul, am aproximat cromatica pixelilor de piele ca fiind grupată într-un singur cluster (datele experimentale ne indicau prezența unui cluster mai mic – pe care l-am considerat neglijabil), de data aceasta, vom reprezenta cât mai exact SL-ul. În plus, pentru a testa rezultatele obținute anterior, am cules pixeli de piele din alte surse: am cules 4 milioane de pixeli din imagini cu opt persoane caucaziene, sub zece iluminări diferite. Am verificat astfel experimentele noastre anterioare, dar și pe acelea arătate în [Terrillon et.al. 2000]: segmentarea lucrează bine în spații unde luminanța e bine separată de cromaticitate și unde culorile pielii sunt reprezentate compact.

Știind că *spațiul de culoare Lab nu e numai cel mai “cuprinzător”, ci oferă și cea mai bună decorelare luminanță-cromaticitate*, am ales pentru acuratețea definiției SL-ului acest spațiu. În plus, culoarea pielii nu e o proprietate fizică a unui obiect, fiind, de fapt, rezultatul unei percepții, cum e și însăși culoarea - măsurătorile CIE, conform cărora s-a fundamentat colorimetria, sunt bazate, așa cum am arătat și subliniat în capitolul anterior, pe experiențe cu subiecți umani. Așadar, chiar dacă Lab nu e un spațiu de culoare prea popular în practica industrială, transformarea Lab ↔ RGB (spațiu aditiv, prezent sub diverse variante, cum e și aceea a spațiului de culoare YCbCr, ca spațiu al echipamentelor electronice de afișare sau scanare) fiind neliniară și deci “costisitoare”, spațiul CIE Lab e foarte important în activitatea de cercetare. Fără a pierde din generalitate, vom presupune că lucrăm numai în planul de cromaticitate – așa cum am arătat în subcapitolul anterior, pentru SD, luminanța e mult mai puțin relevantă decât cromaticitatea.

În figura următoare redăm grafic configurația Slin Locus-ului rezultată în urma culegerii a 8000000 de pixeli de piele din imagini cu persoane de rasă caucaziană:

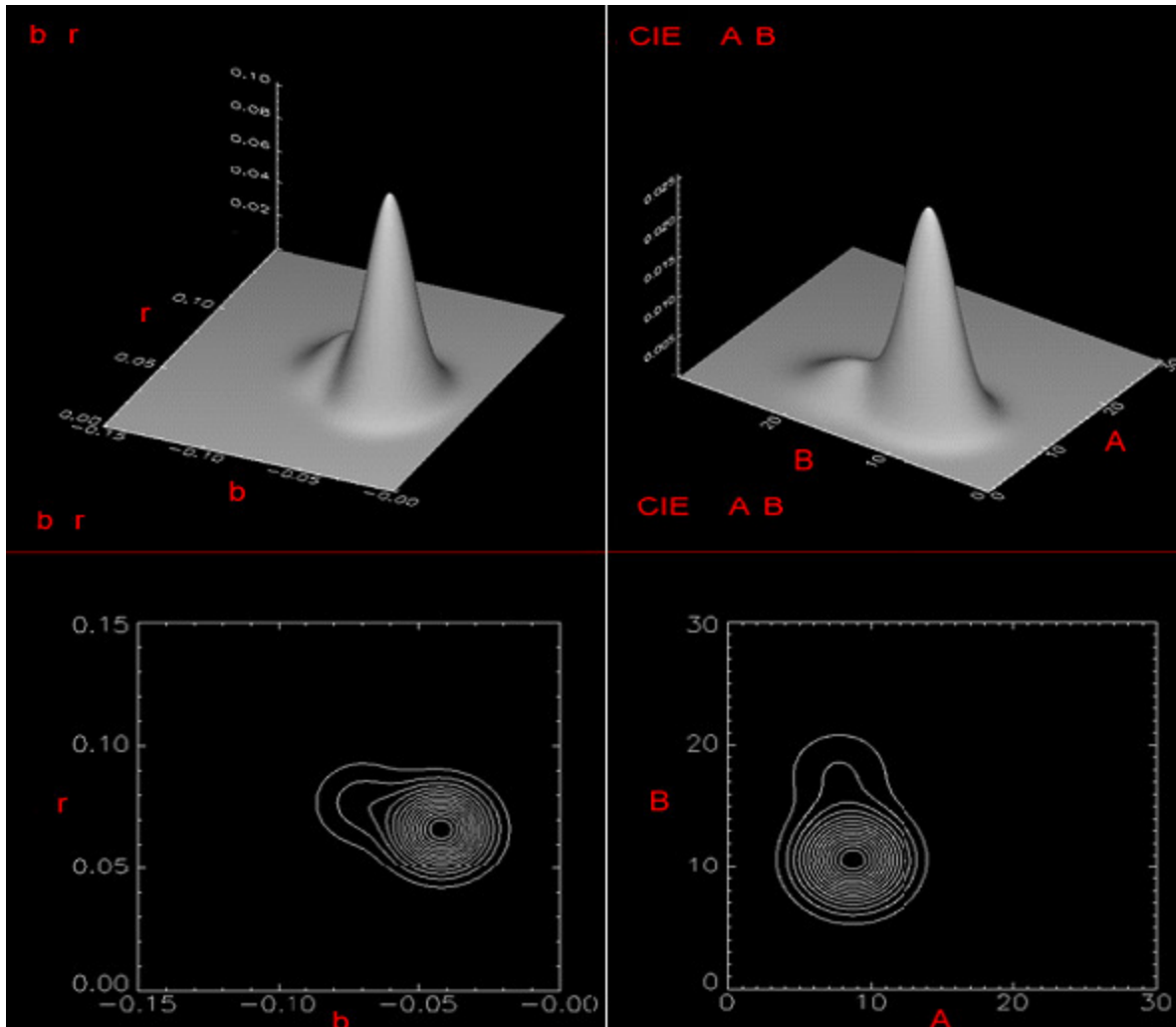


Fig. IV.3.2_1 SL și densitatea lui în spațiile de culoare YCbCr și CIE Lab

IV.3.2.2 Stabilirea prototipurilor

SD făcându-se într-un spațiu perceptual uniform, se justifică definirea de subclase (prototipuri) sub forma unor discuri D_i , $i = 1 \dots n$, de raze aproximativ egale (6) care acoperă un SL-ul.

În ciuda situației ilustrate în Fig.4.3.1.1. pentru a pune în evidență problemele majore ale SD, această metodă de definire de SD nu e una cu performanțe scăzute.



Fig. IV.3.2_2 Selecția zonelor de piele conform metodei noastre

Selecția noastră corespunde unui SL intermediar între cel definit conform [Hsu et al., 2002] și cel definit, precum am menționat în subcapitolul anterior, prin inegalități.

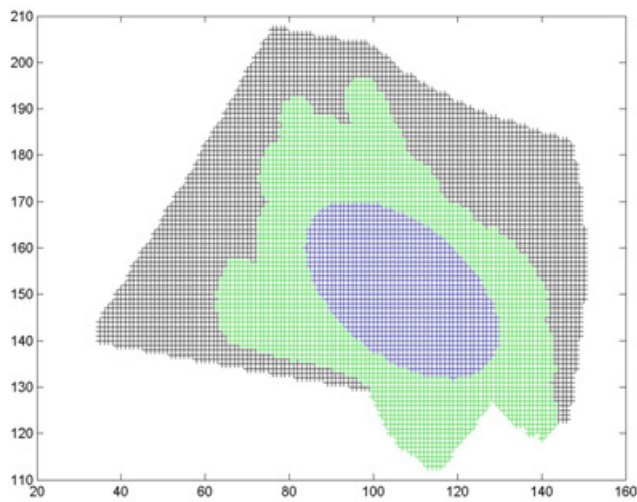


Fig. IV.3.2_3

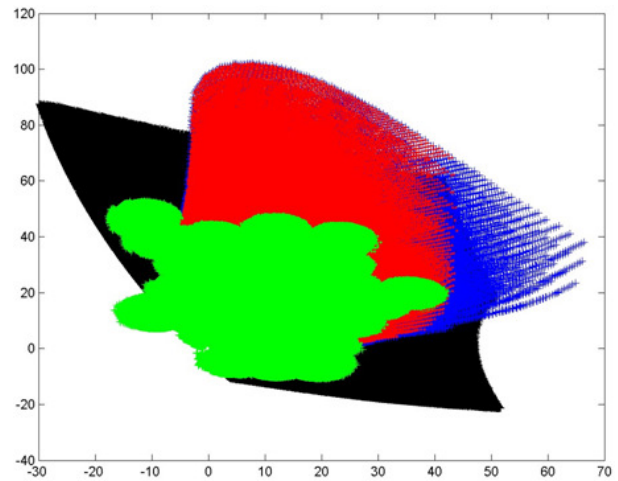


Fig. IV.3.2_4

Fig. IV.3.2_3 ‘Skin gamut’-uri în planul CrCb, definite prin trei metode: albastru – metoda conform [Hsu et al., 2002], verde – metoda noastră, negru – metoda cu inegalități; (b) aceleași ‘skin locus’-uri în planul ab (din Lab), roșu apare pentru o limitare inferioară globală a luminanței a modelului conform [Hsu et al., 2002].

Fig. IV.3.2_4 Observăm că în spațiul de culoare Lab, reprezentarea cu discuri e cea mai compactă. Vom compara cele trei metode și aplicându-le asupra unei imagini care conține mostrele de piele umane:

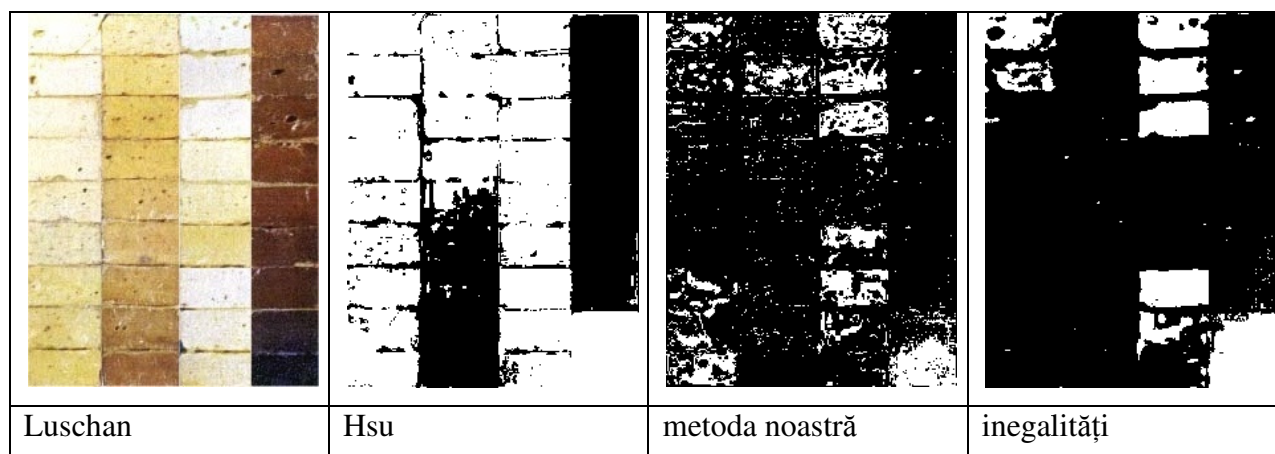


Fig. IV.3.2_5 De la stânga la dreapta și de sus în jos: harta antropologică a culorilor după von Luschan ('the skin color chart of Felix von Luschan'), SD conform [Hsu et al., 2002], conform metodei noastre și conform metodei care se bazează pe inegalități.

Se confirmă că metoda noastră pare să aibă cea mai bună "acoperire" – prin prelucrări morfologice cu mici elemente structurante putem acoperi aproape integral zonele de piele. Astfel, putem neglija problema fals negativelor.

IV.3.3 Condiții suplimentare clasificării skin / non skin

Conform celor menționate la secțiunea "Abordare", ne propunem să obținem condiții suplimentare pentru SD luând în calcul și Dif. Ca și în subcapitolul anterior, zonele de piele care ne interesează sunt cele corespunzătoare zonei feței – în majoritatea aplicațiilor din aceste zone se culeg pixeli "de piele" și acestea sunt zonele de interes. "Flash"-ul e inevitabil în condițiile fotografierii în medii în care iluminarea e redusă. Menționăm că problema reducerii ochilor roșii (RED: 'Red Eyes Reduction') se referă la eliminarea unui artefact "supărător", datorat folosirii 'flash'-ului: în spatele pupilei ochiului (*fundul ochiului*) se află o rețea sanguină, care, în momentul 'flash'-ului, dat fiind faptul că pupila ochiului nu a avut timp să se închidă (datorită vitezei mai mari de apariție a 'flash'-ului), reflectă 'flash'-ul, ceea ce se înregistrează de către

aparatur de fotografiat. RED se referă la detecția acestei pupile “înroșite” și “înnegrirea” ei. E remarcabil cum se poate produce efectul de “ochi roșii” - chiar și în condițiile în care fața e slab iluminată de ‘flash’. Putem restrânge mult aria în care trebuie căutați ochii roșii prin căutarea în interiorul zonelor de piele. Mai precis, ne interesează zona feței – aici e direcționat ‘flash’-ul, deci putem spune că detectăm zona de piele iluminată ȘI de ‘flash’. Sau, altfel spus, în faza de culegere a Skin Locus-ului, din aceste zone “culegem pixeli de piele”. Așa cum spuneam, condițiile suplimentare pentru acet tip de SD se bazează pe presupunerea unei anumite reflectanțe a pielii, adică se presupune că Dif are o variație specifică (de luminanță) pentru fiecare prototip al pielii.

Putem spune că subclasarea / prototipizarea în interiorul clasei ‘Skin Locus’ (SL) o facem prin partiționarea SL, cu observația că mulțimile care formează partiția nu sunt disjuncte – discurile D_i (discuri descrise în secțiunea anterioară, care definesc prototipurile) se suprapun într-o anumită măsură. În fapt, e vorba de o **partiție fuzzy** – în planul de cromaticitate ab; în interiorul SL putem acorda grade de apartenență la fiecare disc /prototip D_i pentru un pixel.

IV.3.3.1 Algoritm bazat pe variația specifică de luminanță pentru fiecare prototip al pielii

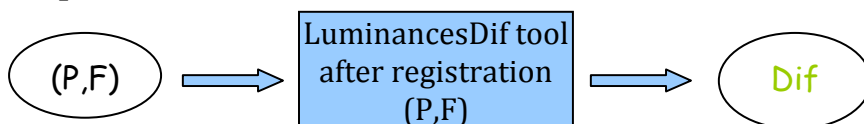
Redăm metoda folosită conform observațiilor de mai sus:

Etapa I. Detectarea prototipurilor de piele P_i în fotografia obținută cu ‘flash’ (F):

Notăm R_i = regiunea corespunzătoare prototipului P_i , fiecare pixel din F va fi etichetat cu “ P_i ”, adică prototipul / discul D_i cu centrul cel mai apropiat de pixel în planul cromaticității ab:



Etapa II. Calculul $Dif = F - P$. Pentru acest calcul trebuie să aliniem (‘registration’) F și P.



Etapa III. Determinarea pragurilor pentru fiecare prototip P_i

Vom descrie într-un pseudocod:

1) Inițializare algoritm de antrenare:

- pentru fiecare prototip de piele (P_i)

- inițializează pragurile de luminanță - inferior (t) și superior (T) -
la valorile maxim, respectiv minim posibil

2) Algoritm de antrenare pentru determinarea pragurilor de luminanță pentru fiecare prototip:

- pentru fiecare pereche $\langle P, F \rangle$ (corespondență 'preview' – fotografie cu
'flash')

- *interestRegion* (**vezi observația de după prezentarea acestui algoritm**) =
în interiorul (măștii) $Z1$ marchează manual zonele de piele

- pentru fiecare pixel \in *interestRegion* modifică pragurile astfel:

Fie P_i eticheta de prototip setată în cadrul etapei I și fie L = luminanța
aceluiași pixel în Dif.

if ($L < P_i.t$) $P_i.t = L$; // $P_i.t$ și $P_i.T$ se manifestă ca "praguri regionale",

if ($L > P_i.T$) $P_i.T = L$; // ca măsuri (praguri) ale prototipului P_i

După rularea acestui algoritm vom avea pragurile pentru fiecare prototip: $[P_i.t, P_i.T]$. Acesta e
algoritmul de antrenare. În cadrul fiecărui prototip P_i , metoda de clasificare va selecta pixelii cu:

Criteriu: pixel chromaticity \in skin (prototype P_i) AND Dif (pixel) $\in [P_i.t, P_i.T]$

Determinarea *interestRegion* nu trebuie să fie exactă ca să contribuie la creșterea selectivității
metodei (de SD în cazul nostru). Mai mult, prin faptul că folosim masca $Z1$, ne asigurăm că
selectivitatea metodei nu poate să scadă. Altfel spus, creșterea selectivității metodei e **tolerantă
la defecte**.

IV.3.3.2 Eliminarea zgomotelor și alinierea imaginii obținută fără ‘flash’ cu imaginea obținută cu ‘flash’

Având în vedere că imaginea fără ‘flash’ (P) e, mai ales în cazul nostru, de calitate inferioară (să nu uităm că asupra ‘preview’-ului nu se aplică corecțiile, cum ar fi ‘white balace’, care se aplică pentru o fotografie normală), mai întâi vom încerca să eliminăm cât mai mult din zgomotul ce vine cu această imagine. Urmărim să facem această imagine cât mai “netedă”, mai comparabilă cu imaginea obținută cu ‘flash’ (F):



Fig. IV.3.3_1 Înainte și după eliminarea zgomotelor pentru imaginea ‘preview’ în scară de gri (luminanță). Deși avem o dimensiune mică a imaginii, se observă eliminarea zgomotului.

Am testat pe sute de poze făcute în condiții cât mai diferite (orientare, distanță, iluminare, indoor etc). Zgomotul apare ca fiind mai degrabă uniform decât neuniform, fără a manifesta însă o regularitate clară. Apar frecvent în P, pentru obiectele cu o reflectivitate mai mare, ‘hot pixels’, adică pixeli cu exces de lumiananță. Chiar dacă după eliminarea zgomotelor nu putem spune că din P au fost eliminați toți “hot pixels”, și putem pierde unele detalii fine din P, am remarcat că prin eliminarea zgomotelor, în zonele cu piele, obținem o “netezime” relativ bună. Eliminarea zgomotelor: aducem P la o dimensiune comună cu F (pentru a putea face diferența lor) și aplicăm un filtru de rang pornind de la datele ce ne sunt oferite de aparatul foto: dintr-o secvență de patru

valori am ales-o pe a doua (am **observat vizual**) că astfel imaginea obținută e mai curată, decât dacă aș lua alt index - din cei patru - sau media între cele patru valori.

Desigur, există și alte metode de eliminare a zgomotelor. Mai general, se pot elimina și alte artefacte – e.g., în afara ochilor roșii, mai pot surveni ca artefacte umbre și specularități - care apar odată cu ‘flash’-ul, pornind de la perechea <P,F>. În esență, se poate spune că aceste metode fac o estimare a imaginii care ar fi trebuit achiziționată prin camera foto, în condițiile unor condiții bune pentru fotografiat. Estimarea se face pe baza imaginilor P și F ale aceleiași scene, ideea fiind că P ne dă informații despre ambient, iar F despre detalii. În această idee, se aliniază P cu F. Menționăm unele metode destul de populare: aceea a *filtrului bilateral* destinat să producă o imagine ‘large scale’ a imaginii pe care o prelucrează prin medierea pixelilor învecinați și cu luminanțe similare – vezi [Eisemann et al., 2004] și [Georg Petschnigg et al., 2004], aceea a *tehnicilor de gradient* – vezi [Agrawal et al., 2005], aceea a *estimării reflectanței* – vezi [J. M. DiCarlo et al., 2001]. Am încercat să ne inspirăm din aceste metode, dar ele prezintă un cost suplimentar prea mare pentru un firmware (presupunem că dorim să facem Red Eyes Reduction ‘in camera’) și, mai ales, nu oferă un suport satisfăcător pentru calibrare. Pentru a alinia P cu F, am încercat să aplicăm un model “factorial” pentru expunere: $E = k g T \text{ ISO}$, unde k e o constantă dependentă de cameră, g e câștigul (‘gain’) ‘CCD’, T e timpul de expunere și ISO caracterizează sensibilitatea camerei. Dar această formulă nu s-a dovedit realistă.

O concluzie generală asupra posibilității de a compara” P cu F ni s-a impus: comparația a două imagini achiziționate în condiții atât de diferite e dificilă. Transformarea intrare-ieșire pe care o efectuează camera foto e accentuat neliniară când trecem de la un mediu ambiant cu iluminare mică (insuficientă) la același mediu în prezența ‘flash’-ului

Am exclus problema care ar apărea datorită nealinierii spațiale a celor două imagini, deși e o problemă care poate apare și poate împiedica creșterea selectivității detecției zonelor de piele - exemplu:

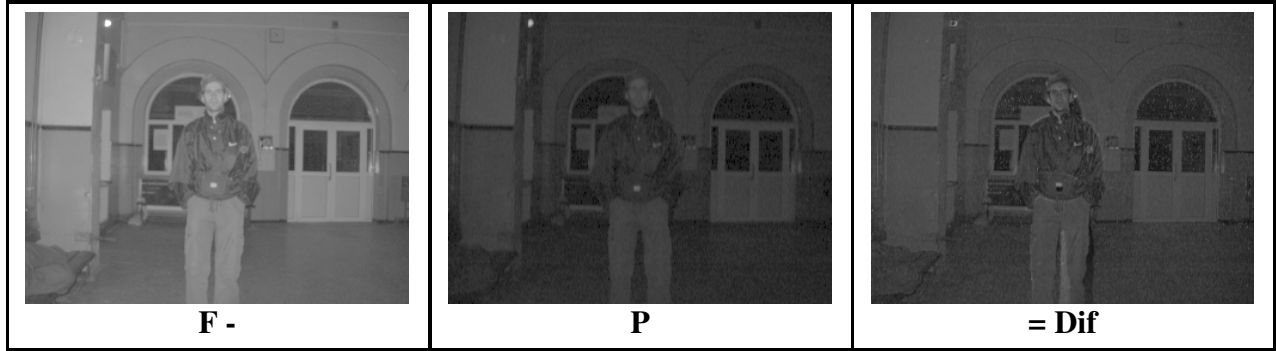


Fig. IV.3.3_2 Mișcarea persoanei fotografiate în momentul fotografierii cu ‘flash’ (F) față de momentul achiziției imaginii scenei fără ‘flash’ (P, care, astfel, apare “întunecată”) poate împiedica creșterea selectivității detecției zonelor de piele / față prin apariția unor neregularități suplimentare.

!! Vom presupune că problema mișcării scenei sau în scenă nu va împiedica alinierea spațială a lui P cu F, aliniere pe care o facem prin procedeul de *corelare a fazei* (*‘phase correlation’*) în domeniul frecvență. Dacă această diferență e prea mare, pur și simplu în algoritmul de la IV.3.3.1. nu mai punem și condiția $AND Dif (pixel) \in [Pi.t, Pi.T]$, nemaiputându-ne permite creșterea selectivității metodei de SD. Același lucru îl vom face și dacă P e o imagine “prea întunecată” în care nu distingem forme, contururi, adică luminozitatea globală a lui P e sub un anumit prag.

‘Flash’-ul introduce niște umbre diferite de cele ale scenei inițiale - vom exemplifica în figura de mai jos cu o scenă tipică, des întâlnită, în care lumina vine din lateral. Excludem și această problemă având posibilitatea să achiziționăm un ‘preview’ cu ‘preflash’. ‘*Preflash*’-ul e un mic ‘flash’, care, anunțând din timp ‘flash’-ul, dă mai multe șanse pupilei oculare de a se închide înainte de a se produce ‘flash’-ul, prevenind astfel, în mod hardware, apariția ochilor roșii. Cum ‘preflash’-ul iluminează scena din același “punct” din care o face și ‘flash’-ul, nu se mai pune problema umbrelor modificate între P și F. În plus, dacă e suficient de puternic, ‘preflash’-ul rezolvă P-urilor foarte întunecate și, deci, cu mult zgomot.

	<p>F – o imagine în care umbrele au fost înlăturate de ‘flash’</p>
	<p>P - o imagine de luminozitate redusă, care “păstrează” totuși umbrele din scena inițială. Observăm că, în mod oarecum neașteptat, pe partea dreaptă (cum privim la imagine) fețele prezintă zone cu luminozitate mai mare decât în F</p>
	<p>$Dif = F - P$: o imagine în care fețele sunt “tăiate” din partea dreaptă (cum privim la imagine) de zonele negre – zone corespunzătoare celor menționate mai sus.</p>

Fig. IV.3.3_3 Artefactul înlăturării umbrelor de către ‘flash’

Facem toate simplificările menționate mai sus pentru a avea șanse reale de a crește selectivitatea metodei noastre de SD. Chiar și așa, vom vedea cum, comparația lui P cu F se face în “condiții grele”, când pare că singura comparație care se poate face e aceea referitoare la luminanță, în condițiile pe care le-am specificat. Chiar și așa, comparația lui P cu F nu pare să aducă o condiție de discriminare foarte puternică – în figura următoare se observă dependența ΔI - a lungimii intervalului de variație a luminanței în Dif pentru zonele cu fețe - de mediul ambiant și direcția ‘flash’-ului, pentru fețe ale unor persoane de rase diferite, dar și asemănătoare:

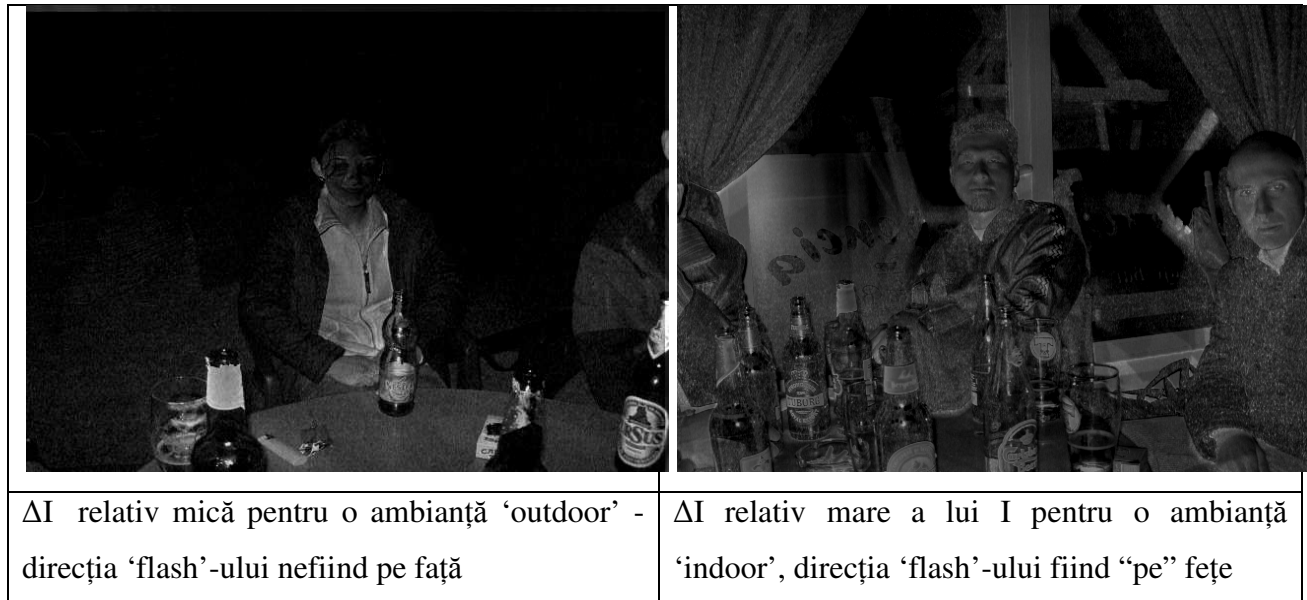


Fig. IV.3.3_4 Variația luminanței în Dif pentru zonele cu fețe

Am încercat să folosim un alt criteriu de comparație: în loc de diferența în luminanță Dif, am calculat rapoarte, am încercat să deducem diferențe adaptive pentru a elimina neregularități de genul obținerii unei valori mai mari în P decât în F, pentru același pixel. Nu am găsit o metodă care să acopere toate cazurile – aceste neregularități, inevitabile (am arătat anterior), pot fi detectate pentru a fi “sărite”. Totuși, mai important decât succesul metodei concrete de reducere a fals pozitivelor pentru SD e metaeuristica conform căreia instanțiem metoda concretă. Metaeuristica care ne garantează că putem reduce fals pozitivele dacă o aplicăm, eficiența implementării putând varia de la o aplicație la alta, de la o adaptare a metaeuristicii la altă adaptare a ei. Metoda de SD pe care am conceput-o e importantă și deoarece, în fapt, “ilustrează” metaeuristica noastră.

IV.3.4. Discuție în urma experimentelor

IV.3.4.1. Precauții asupra metodei

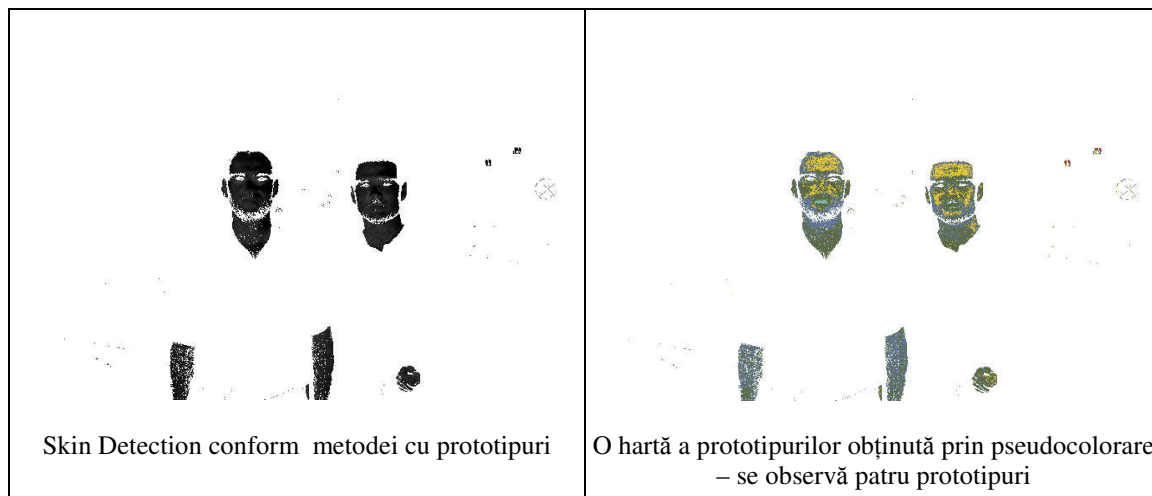


Fig. IV.3.4_1 Selecția zonelor de piele în poza cu ‘flash’ (culoarea albă a fost folosită pentru a marca pixelii care nu sunt “de” piele)

În etapa II a algoritmului obținem Dif. Voi prezenta imaginea la o dimensiune mai mare pentru a arăta slaba calitate a lui Dif – de fapt, Dif-ul următor e unul din cele de o calitate relativ bună, întrucât putem distinge ușor fețele. Așa cum am arătat în Fig. IV.3.3_3, fețele sunt “înnegrite” în partea dreaptă (cum privim la imagine) datorită sursei / surselor de lumină din partea dreaptă:



Fig. IV.3.4_2 Pentru acest Dif dacă, în etapa III a algoritmului, marcăm și zonele de față “înnegrite” în Dif vom afecta selectivitatea optimală (din punct de vedere al metodei noastre) a metodei, prin afectarea intervalelor de luminanță $[P_i.t, P_i.T]$, calculate de algoritm pentru fiecare prototip (“înnegriră” corespunde unui $P_i.t$ mai mic, iar “albirea” corespunde unui $P_i.T$ mai mare), ca și condiție suplimentară de selecție, aceste intervale ajungând să fie “prea” largi.

Putem trata această problemă înlăturând din marcarea (printr-o operație ȘI cu o mască) aceste zone de luminozitate sub un anumit prag global în Dif. Așadar, o marcarea nerestricționată a pielii (ca în figura de mai jos) poate afecta intervalul de variație a luminanței per prototipurile corespunzătoare zonei care nu ar trebui lăsată să fie luată în calcul de algoritmul nostru. Prin concepția metaheuristică noastră, deși acest lucru duce la o scădere mai mică a fals pozitivelor, totuși selectivitatea va crește. De fapt, așa cum am spus și mai înainte, selectivitatea va crește. Dar, precum am spus, metaheuristică noastră are un succes cu atât mai mare cu cât ne asigurăm că lucrăm cu un Dif care să nu prezinte neregularități prea mari – includem aici: zgomotele din P, umbrele (pe fețe) din P, deplasările prea mari ale camerei sau în scena fotografiată, între momentul achiziției lui F și momentul achiziției lui P etc. Succesul ar fi și mai mare dacă am reuși să obținem o partiționare cât mai clară (fie ea fuzzy) a SL și dacă am realiza o anumită taxonomie a scenei după: tipul de ambient în care se află scena (înăuntru sau afară), ce fel de iluminare e folosită și toată gama menționată sub numele de principiul specializării și al calibrării în subcapitolul anterior.

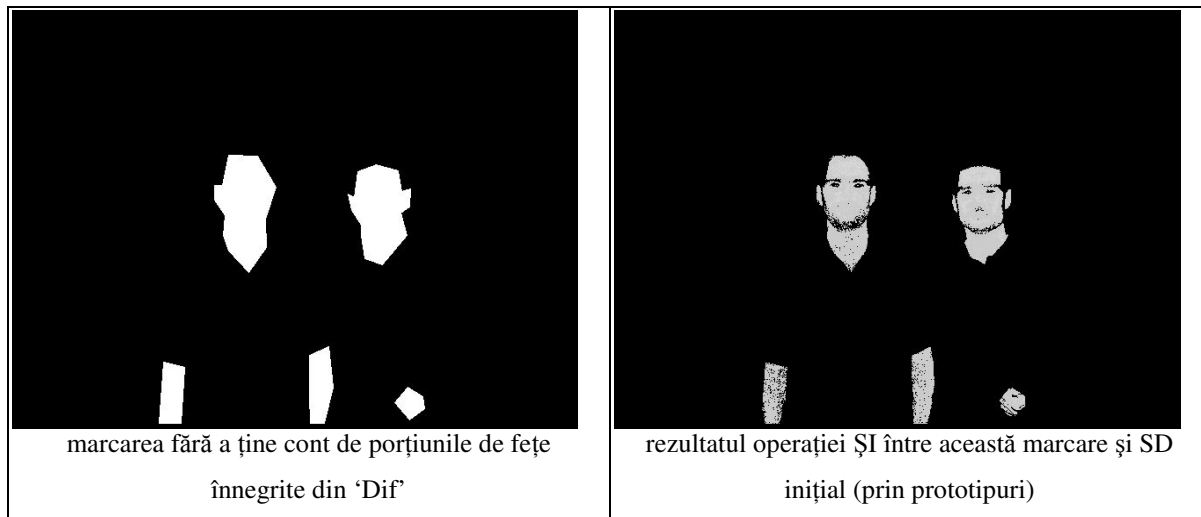


Fig. IV.3.4_3 În cazul unei marcări inadecvate, totuși, chiar dacă nu în măsura în care s-ar fi putut (printr-o marcarea adecvată, mai precis, care să nu ia în considerare zonele de față “înnegrite prea mult” în Dif), se vor reduce fals pozitivile.

IV.3.4.2. Câteva rezultate promițătoare

Ilustrăm un exemplu de reducere a fals pozitivelor, chiar în condițiile în care nu ne-am luat precauțiile menționate la subsecțiunea anterioară:

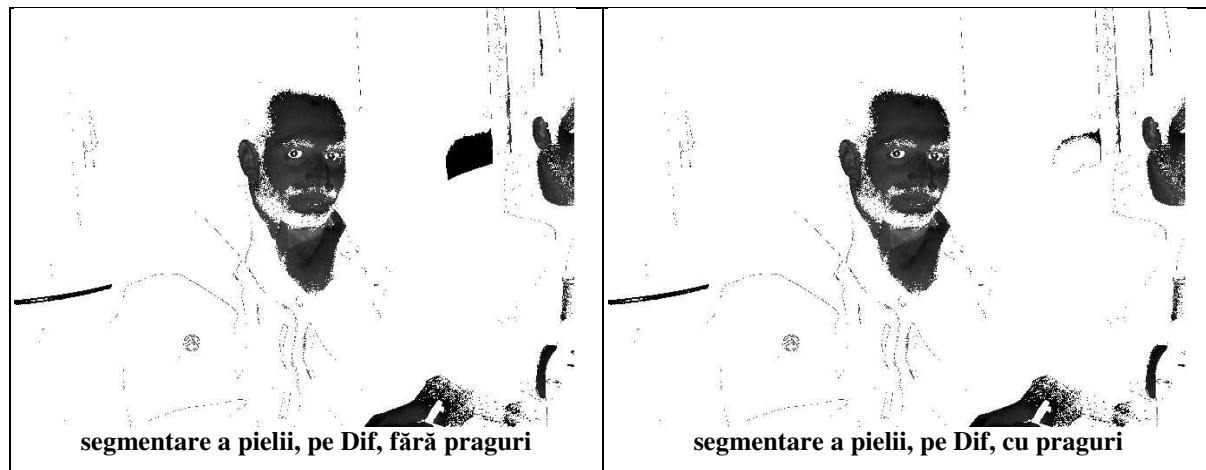


Fig. IV.3.4_4 O situație în care aplicarea pragurilor pentru luminanțele prototipurilor are ca efect restrângerea selecției zonelor de piele – observăm că porțiunea dintre cele două fețe a fost scoasă din selecție.



Fig. IV.3.4_5 Fotografia cu 'flash' ne confirmă că această porțiunea menționată la Fig. IV.3.4_4 putea fi luată relativ ușor drept zonă de piele

Alt exemplu:

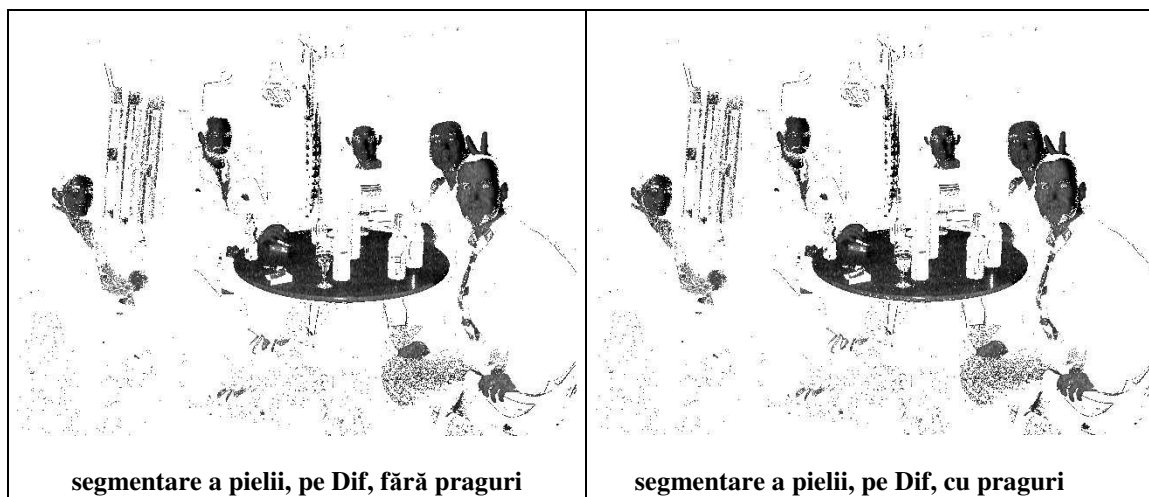


Fig. IV.3.4_6 4 O altă situație în care aplicarea pragurilor pentru luminanțele prototipurilor are ca efect restrângerea selecției zonelor de piele – observăm că zonele selectate corespunzătoare paharului și brichetei dispar în cazul segmentării pielii cu praguri.

În figura de mai sus, observăm și că am selectat zone de fețe în care luminozitatea e foarte mică. De ce sunt admise praguri inferioare atât de mici pentru prototipurile “de” piele corespunzătoare? Deoarece nu ne-am luat precauțiile menționate la subsecțiunea anterioară. Totuși, chiar și în aceste condiții, am redus fals pozitivele.

IV.3.5. Concluzii

Se presupune că, în majoritatea problemelor de clasificare, un fals negativ e mai “rău” ca un fals pozitiv – mai ales când clasificarea e folosită în cadrul unei etape de preselecție. Astfel, multe metodele de clasificare au ca prioritate să nu “rateze” zonele / clasele de interes. Reversul e că selectivitatea clasificării poate scădea. În această contribuție a noastră dăm o *metaheuristică* de reducere a fals pozitivelor (FP) prin folosirea subclasării și a marcajelor *perceptuale*, adică a marcajelor de clasificare furnizate de expertul uman (putem spune că metoda de Skin Detection e antrenată conform observațiilor expertului uman). Subclasarea presupune partiționarea în subclase a claselor furnizate de o metodă oarecare de clasificare inițială (care constituie intrarea metaheuristicii) și găsirea de condiții suplimentare de rafinare a clasificării în interiorul fiecărei

clase. Dacă numim *prototipuri* ale unei clase, subclasele obținute în interiorul fiecărei clase prin partiționare, creșterea selectivității unei clasificări, adică reducerea FP, e dată de relația:

$$\text{zona_de_interes} = \text{zona_prototipurilor} \cap \text{zona_marcajelor} \quad (*)$$

Problema concretă de care ne-am ocupat și cu ajutorul căreia punem în lumină eficiența metaheuristicii noastre e aceea a “Skin Detection” (SD), ceea ce corespunde faptului că pornim de la o clasificare cu două clase: ‘skin’ și ‘non-skin’. În fapt, ne propunem să subclasăm în interiorul clasei ‘skin’ (S): zona noastră de interes.

Abordarea e “pixel based” datorită avantajelor oferite de acest tip de metode (vezi secțiunea IV.2.1.). Căutăm să selectăm cât mai precis ‘Skin Pixels’. Presupunem că într-o etapă anterioară, de antrenare în vederea recunoașterii SP (vezi IV.2.), am colectat (dintr-o bază de date cu imagini) SP și avem **representare și, poate, chiar o partiționare a ‘Skin Locus’-ului.**

Subclasarea

Subclasarea corespunde partiționării SL. Pentru că ne propunem să partiționăm Skin Locus-ul, alegem ca spațiu de reprezentare a culorilor un spațiu de culoare perceptual uniform – desigur, am ales Lab și datorită generalității sale (capacității sale de reprezentare a culorilor) și decorelării în cel mai înalt grad a luminanței de cromaticitate. SD făcându-se într-un spațiu perceptual uniform, se justifică definirea de subclase (prototipuri) sub forma unor discuri D_i .

Partițiile / subclasele P_i , care corespund discurilor D_i , nu sunt disjuncte: se suprapun într-o anumită măsură. În fapt, e vorba de o *partiție fuzzy*.

Ne-am propus să obținem condiții suplimentare pentru SD luând în calcul Dif (diferența între luminanțe) = F (imaginea corespunzătoare unei fotografii făcută cu ‘flash’) – P (imaginea corespunzătoare unei fotografii făcută fără ‘flash’). De fapt, P corespunde unui preview, fiind, deci, o fotografie de calitate slabă – în aceste condiții (SD se face ca etapă de preselecție în vederea ‘Red Eyes Reduction’) pare că singura comparație care se poate face e aceea referitoare la luminanță.

În cadrul fiecărui prototip P_i , metoda de clasificare va selecta pixelii cu:

cromaticitate pixel \in prototip P_i ȘI $\text{Dif}(\text{pixel}) \in [P_i.t, P_i.T]$ (**)

, unde pragurile inferior și respectiv superior $P_i.t$ și $P_i.T$ se determină printr-un algoritm iterativ de antrenare care rulează peste imagini cât mai “apropiate” de cele pentru care ne propunem SD. Acest tip de **antrenare adaptivă** e foarte important pentru creșterea selectivității metodei. Portarea SD la altă categorie de imagini va însemna doar rerularea acestei preprocesări.

Folosirea marcajelor perceptuale

În ideea descrisă în secțiunea II.2.4., adică “Raționamentul probabilistic cu ajutorul percepțiilor”, folosim fotografiile în care expertul uman a marcat zone de piele ale feței pentru a face MAI selectivă clasificarea. Antrenarea menționată anterior are ca intrări aceste imagini marcate. “MAI” are atât sensul de antrenare adaptivă, cât și cel de a ne asigura de faptul că impunerea de condiții suplimentare (Dif) nu poate duce la o clasificare mai puțin selectivă – folosim o mască conform clasificării de intrare în acest scop. Mai mult, determinarea pentru zona_de_interes din (*) nu trebuie să fie exactă ca să contribuie la creșterea selectivității metodei (de SD în cazul nostru). Pe scurt, **creșterea selectivității metodei e garantată și tolerantă la defecte.**

Culegerea informațiilor utile

O concluzie generală asupra comparativității între P cu F ni s-a impus: comparația a două imagini achiziționate în condiții atât de diferite e dificilă. În principal, problema e că P e o imagine de slabă calitate, fiind achiziționată în condiții de întuneric (altfel nu ar mai fi necesar ‘flash’-ul) și, deci, cu mult zgomot. Așa am ajuns să folosim P cu ‘preflash’. Mai mult, cum ‘preflash’-ul iluminează scena din același “punct” din care o face și ‘flash’-ul, nu se mai pune problema umbrelor. În plus, dacă e suficient de puternic, ‘preflash’-ul rezolvă problema P-urilor foarte întunecate. Totuși, chiar și folosind P cu ‘preflash’, imaginile P și F pot prezenta caracteristici prea diferite pentru a fi relevantă o comparație a lor (Dif). Ne propunem să detectăm aceste situații pentru a nu le lua în calcul în etapa de antrenare, șansele reducerii FP (reducere, de altfel, garantată prin construcția metaheuristicii) putând fi grav afectate dacă

aplicăm întocmai algoritmul descris la IV.3.3.1. Exemple de **precauții** care pot fi luate (idei de îmbunătățire) la rularea algoritmului de determinare a pragurilor de luminanță per prototip:

- dacă numai câțiva pixeli impun lărgirea intervalului de luminanță per prototip, nu vom efectua această lărgire (prin **vot majoritar**)
- problema nealinierei P cu F: dacă această diferență e prea mare, nu vom folosi perechea <P,F> pentru a stabili praguri de luminanță per prototip

Pe lângă multitudinea factorilor ce influențează detecția pielii se adaugă neliniaritățile electronicii camerei foto, ceea ce configurează problema SD ca una de clasificare nedeterminată – imposibil de “prins” în formule, nu neapărat generale. SD nu poate fi aplicată cu eficiență dacă nu acționăm conform unui principiu al specializării și al calibrării – trebuie definite atât anumite condiții în care se face fotografierea scenei, cât și zonele de piele pe care suntem interesați să le selectăm (rase, piele sub incidența ‘flash’-ului etc). O concluzie generală s-a impus: culoarea pielii e o trăsătură discriminatoare a fețelor umane într-un mediu exterior controlat.

E important de avut în vedere să lucrăm cu baze de imagini cât mai complete (cu cât mai multe informații (marcaje) perceptuale), dar și mai “specializate”. E important să stabilim o taxonomie în baza de imagini – posibile criterii: orientare, distanță, iluminare, indoor / outdoor, tipuri și parametri de camere foto, etnii (și, deci, culori ale pielii) etc. Am numit aceasta *principiul specializării și al calibrării*. Se recomandă ca la procesare să lucrăm cu Skin Locus-ul potrivit situației (în care ce obține fotografia) pentru performanțe sporite.

Rezultate experimentale

În ce privește metoda inițială de SD, conform Fig. IV.3.2_5 metoda noastră are cea mai **bună acoperire**. Mai mult prin prelucrări morfologice cu mici elemente structurante putem acoperi aproape integral zonele de piele; astfel, se reduc și fals negativele.

Rulând pur și simplu algoritmul metaheuristicii (de la IV.3.3.1.), fără precauțiile menționate mai sus, conform subsecțiunii IV.3.4.2. (“Câteva rezultate promițătoare”), am redus FP-ele metodei inițiale de SD.

Viteza metodei la procesare e dată de simplitatea condiției de selecție (**) care se pune.

Observație: Un alt avantaj important al metaheuristicii noastre e că, practic, e o unealtă care poate fi foarte utilă pentru cercetare. **Se poate demonstra astfel, în mod practic și foarte benefic, utopia unor probleme de clasificare** – așa cum teorema de separare pentru seturi fuzzy convexe (vezi și II.2.3.2. “Mulțimi și partiții fuzzy”) vorbește de un grad maxim de separabilitate între două mulțimi fuzzy (putem proiecta prototipurile P_i ca mulțimi fuzzy).

Idei de îmbunătățire a metaheuristicii

- creșterea granularității prototipurilor
- creșterea “naturalității” / separabilității prototipurilor – evident, subclasarea pe care am aplicat-o (prin stabilirea discurilor D_i care partiționează SL) poate fi rafinată, “clusterizând”
- prin prelucrări morfologice putem elimina “accidentele” (situațiile necorespunzătoare) din faza de antrenare (în faza de procesare presupunem că nu e timp pentru asemenea precauții)
- proiectarea prototipurilor P_i ca mulțimi fuzzy – dacă partiționarea în prototipuri se face într-o manieră fuzzy (ele se suprapun), totuși ele nu au fost gândite ca mulțimi fuzzy. Aceasta ar fi dus la un control mai bun al prototipurilor și, deci, la o creștere suplimentară a selectivității clasificării

V§ Sinopsis al contribuțiilor originale. Studiul viitor

V.1. Vedere sinoptică a contribuțiilor noastre

În aceasta lucrare, teoria a fost abordată din punct de vedere pragmatic – chiar și în cazul explicațiilor care s-au dat pentru unele noțiuni, unele matematice, procedee folosite des în practică, dar mai rar “deconstruite” pentru a putea fi înțelese în profunzime, aplicate / adaptate creativ de către cei care caută rezolvări unor probleme nedeterminate concrete de Clasificare a Imaginilor (CI), de către cei care caută noi criterii de clasificare pentru rezolvarea unor asemenea probleme. Rezumăm în următoarele două secțiuni (V.1.1. și V.1.2.) contribuțiile noastre de sinteză / observație teoretică, contribuții la care am ajuns în urma investigațiilor cu ocazia acestor probleme, investigații care ne-au dus cercetarea teoretică spre domeniul fuzzy.

V.1.1. Contribuții teoretice (per capitole)

În această lucrare încadrăm în mod original problema Clasificării în Imagini într-o viziune unitară, per domenii: Domeniul Spațial, Domeniul Probabilistic, Domeniul Frecvență și Domeniul Fuzzy. Secțiunea II.1.5., “O viziune abstractă asupra recunoașterii formelor, în jurul conceptului de corelare” întărește această legătură între domenii, în jurul conceptului de corelare, concept care, din punct de vedere matematic, se bazează pe produsul intern într-un

spațiu vectorial. Multe alte informații sunt integrate în această viziune, având astfel o lucrare legată, coerentă.

Iș **Încadrarea problemei CI și descrierea ei în Domeniul Spațial și în Domeniul Probabilistic** – capitol introductiv, în care arătăm limitarea teoretică a unor tehnici clasice, e.g., a rețelelor neuronale, în ce privește rezolvarea unor probleme concrete de CI. Încheiem acest capitol remarcând că **domeniul probabilistic, deși poate “veni” cu informații în plus față de cel spațial, nu încurajează folosirea intuițiilor, a adaptărilor pentru o rezolvare cât mai adecvată a unor probleme mai complexe de clasificare.**

IIș “Separabilitatea claselor” – clasificarea fiind în esență o problemă de separare

- II.1.3. Eșantionarea semnalului: precizia necesară și intercondiționarea reprezentărilor în Domeniul Frecvență și în Domeniul Frecvență-Timp – imprecizia ca aliasing (II.1.3.1. Precizia necesară reconstruirii semnalului și efectul de aliasing) și ca principiu al nedeterminării (II.1.3.2. Intercondiționarea rezoluției reprezentărilor în Frecvență și în Frecvență-Timp: principiul nedeterminării). Fig.II.1.3_12 prezintă o imagine sintetică originală. Secțiunea II.1.5., “O viziune abstractă asupra recunoașterii formelor, în jurul conceptului de corelare”, propune o sinteză asupra problemei clasificării, în domeniile spațial, frecvență și probabilistic, problemă care poate fi văzută ca una de corelare / de-corelare sau de descompunere.

- secțiunea II.2.2., “Domeniul posibilului: o fuziune fuzzy a diverselor tipuri de cunoștințe” dă seama de capacitatea de integrare a informațiilor precise sau imprecise, cu diverse grade de credibilitate (probabilitate) de către domeniul fuzzy, un domeniul față de care domeniul probabilistic e un caz particular. Domeniu care poate fi văzut și ca “loc de trecere” între știința calculatoarelor și alte discipline de cunoaștere, cum ar fi cele sociale. Ceea ce dă **posibilitatea extinderii raționamentului fuzzy (aproximativ) în științele “inexacte”** e cuvântul / termenul lexical / variabila lingvistică. În secțiunea II.2.4., <<“Fuzzy logic = calculul cu cuvinte” [Zadeh, 1996]>>, sintetizăm concluziile pe care le-am desprins din literatura de specialitate pe această temă, propunând ca legătura între științele “exacte” (fizica a arătat până unde poate merge acest “exact” – vezi principiul incertitudinii al lui Heisenberg, enunțat încă din 1927) și cele **“inexacte” să se facă în jurul criteriului de proximitate, pe baza claselor fuzzy.** În esență facem următoarele observații:

- II.2.4.1. Termenul lexical și abilitatea de a opera cu informații bazate pe percepții sau fler – teoria fuzzy are o legătură profundă cu semantica

- II.2.4.2. Scale fuzzy – e interesant de observat că **scafele metrice ordinale** au fost introduse în contextul psihofizicii (deci, iată, putem vorbi de o contribuție la teoria fuzzy din partea psihologiei, o disciplină de care, acum un secol, științele exacte s-ar fi “ferit”) ca metodă de a măsura date multidimensionale (în capitolul III vom vorbi de procesarea multi-canal, cu aplicație concretă în procesarea imaginilor color, culoarea fiind, în fapt, o percepție psiho-fizică). Folosirea unei scale metrice fuzzy e o îmbunătățire dacă se ia în considerare o relație de proximitate pe setul de reprezentare, relație de proximitate care are acoperire într-o relație de proximitate pe setul reprezentat.

- în secțiunea II.2.5. **propunem conceptul de PROTOTIP pe post de clasă fuzzy**: “sâmbure” în jurul căruia se centreză clasa fuzzy conform unei distribuții de posibilitate. Acest concept, pe care l-am utilizat efectiv în contribuția practică IV.3, “Folosirea subclasării și a marcajelor pentru reducerea fals pozitivelor”, apare ca o generalizare și ca o tratare mai adecvată a claselor fuzzy.

III§ Domeniul multispectral al imaginilor color – deși ne vom referi numai la cazul domeniului multispectral al culorilor, discuția noastră se poate generaliza în ce privește procesarea informației care ne vine pe mai multe canale. Aici arătăm cum ceea ce poate integra - matematic, pentru descrierea sensibilităților de culoare, folosim produsul scalar - aceste informații multi-canal e percepția, percepție care stă la baza teoriei fuzzy, dar și a colorimetriei - spațiile de culoare uniform perceptuale au fost concepute plecând de la conceptul / percepția de proximitate (vezi Just Noticeable Difference).

În subcapitolul III.3., “Caracterul perceptual al culorii”, remarcăm că acolo unde filtrarea liniară nu a avut succes (e.g, pentru procesarea multi-canal), poate avea succes o abordare fuzzy prin caracterul său adaptiv, trunchiat (vezi operațiile pe mulțimi fuzzy), asemănător unei logici a percepțiilor.

CI, prin chiar obiectul ei, “ilustrează”, poate, cel mai bine multe din conceptele clasificării. Așa cum am spus și în rezumat-ul lucrării, discuția noastră se aplică și altor probleme de clasificare, mai ales în ce privește metodele bazate pe elemente componente – care corespund în cazul CI metodelor de clasificare bazate pe pixeli.

V.1.2. Contribuții practice

Rezumăm contribuțiile noastre practice, contribuții ivite, ca și cele teoretice, în urma confruntării cu probleme nedeterminate concrete de CI:

IV.1 Transformări eficiente între spații de culoare – am implementat o metodă de interpolare cu un grad ridicat de generalitate, superioară ca acuratețe și viteză de execuție.

IV.2 “Definirea formei spațiului unei probleme de recunoaștere nedeterminate și modelarea acestuia. Cazul ‘skin detection’” – proiectăm și implementăm o **metauristică adaptivă și dinamică în vederea rulării unor algoritmi ‘run-length’, superiori ca acuratețe și viteză de execuție**. Prin folosirea de **informație probabilistică** ne putem extinde în domeniul impreciziei, probabilistic / fuzzy, putem prelua și furniza cunoștințe probabilistice / fuzzy.

IV.3 “Folosirea subclasării și a marcajelor pentru reducerea fals pozitivelor. Cazul ‘skin detection’” – proiectăm și implementăm o metauristică adaptivă și dinamică a cărei noutate se bazează pe **folosirea subclasării și a marcajelor perceptuale**. Conform teoriei fuzzy **se poate demonstra în mod practic utopia unor probleme de clasificare, ceea ce poate scuti de foarte costisitoare investigații** – vezi și unele dintre concluziile din finalul secțiunii IV.3.5, între care și ideea de a găsi un criteriu de subclasificare care să crească (față de ‘Dif’ și partiționarea (fuzzy) a ‘Skin Locus’-ului în discuri de raze de valoare constantă) separabilitatea claselor. Din punct de vedere teoretic, teorema de separare pentru seturi fuzzy convexe, teoremă care încheie noul articol [Zadeh, 1965], este, și ea, grăitoare în acest sens.

V.2. Studiul viitor

Într-un articolul recapitulativ, [Vertan & Buzuloiu, 2000], s-a observat că pentru imagini gri au avut succes filtrările adaptive și, deci, neliniare. Pentru imagini color, extinderea acestor filtre prin prelucrarea individuală a componentelor de culoare s-a dovedit inadecvată în ce privește păstrarea topologiei. Ori păstrarea topologiei e un criteriu important de validare a reușitei multor procesări de imagini – de fapt, și noi am aplicat un raționament topologic când am configurat ‘Skin Locus’-ul, considerând că probabilitatea unui ‘Skin Pixel’ crește spre centrul SL – altfel, SL ar fi contrar percepției noastre. **Sustinem: criteriul păstrării topologiei e unul determinant în găsirea soluțiilor majorității problemelor de clasificare nedeterminate în imagini (putem avea și alte reprezentări decât imagini digitale). Raționamentul fuzzy, dovedindu-se până în zilele noastre cadrul teoretic cel mai “integrator” de abordare a acestora, remarcăm caracterul său intrinsec topologic (fundamentală fiind relația de proximitate).** Ne propunem să continuăm investigația CI pe partea de procesare neliniară a semnalelor (multi-canal), apelând la fundamentele teoriei fuzzy, în direcția prelucrărilor morfologice. Din punct de vedere morfologic, conceptul de PROTOTIP, pe care l-am propus, poate fi văzut ca un schelet (‘skeleton’) / nucleu de clasă (fuzzy) – noțiunea de ‘skeleton’ e prezentată în lucrare la subsecțiunea I.4.1.2., “Skeletoane”, în cadrul secțiunii I.4.1., “Simplificarea formelor”. Astfel, “fuzificarea” ar corespunde extrapolării de la un ‘skeleton’, iar “defuzificarea”, care, în general, ia forma mapării unui vector într-un scalar și e asociată transformării raționamentelor umane calitative în expresii numerice (cantitative), ar corespunde calculării ‘skeleton’-ului. În cazul GIS, operația de generalizare poate corespunde calculării ‘skeleton’-ului cu păstrarea topologiei [Gold & Shankar, 2001]. În [Georgescu 2, 2009] arătăm cum, în vederea reconstrucției 3D din secțiuni paralele, pentru o partiționare cât mai corectă în cazul ‘Double Branching Problem’ e nevoie să găsim un skeleton “cât mai adecvat” între cele două contururi în discuție - evident, problema având mai multe soluții, alegem soluția care se mulează cel mai bine criteriilor soluției (se știe că pentru o decizie multicriterială putem folosi o integrală fuzzy), dar și de acuratețe și viteză ale problemei concrete. În general, raționamentul

fuzzy, chiar dacă presupune îndeplinirea unor condiții topologice, se poate adapta problemei în cauză.

Pe de altă parte, în subsecțiunea II.2.2.2., <<Teoria fuzzy și bazele de date sau de cunoștințe (sistemele expert) cu informații incerte / “granulate”>>, menționăm exemplarul articol [Zadeh, 1986], în care se arată că dilatarea și eroziunea, sau închiderea și deschiderea, ca operațiuni morfologice de bază, corespund necesității și posibilității, având, în plus, același tip de dualitate. Necesitatea și posibilitatea fiind conceptele de bază ale teoriei posibilității, iar această teorie, după cum am evidențiat în subsecțiunea II.2.2.1., “Teoria fuzzy și teoria posibilităților”, corespunzând teoriei fuzzy.

Teorema de separare pentru seturi fuzzy convexe [Zadeh, 1965] e un alt indiciu în favoarea studiului CI în privința prelucrărilor morfologice care pot ajuta clasificarea. În general, clasificarea se face în clase / mulțimi **convexe** – vezi și reprezentarea Skin Locus-ului de la contribuția practică IV.2.

În subsecțiunea II.2.3.2., “Mulțimi și partiții fuzzy”, am accentuat importanța teoremei de reprezentare Negoită-Ralescu a seturilor fuzzy ca tăieturi α . E o teoremă care ne poate furniza o intuiție clară asupra unei posibile aproximări a unei mulțimi fuzzy: “ α din ‘ α -cut’ poate fi văzut ca un **nivel de apoximare** al setului sau numărului fuzzy” - “pe baza ‘ α -cut’ se definesc operații între numere fuzzy”.

Ordonarea cât mai “inspirată” a numerelor fuzzy se poate face în lumina unor intuiții geometrice [Rao & Shankar, 2011]. În [Vertan, 2009], se vorbește de “coordonate paralele” – fiecare culoare are asociat un poligon, calculul distanțelor între culori reducându-se la calculul intersecțiilor acestor poligoane; și aici α e pe post de grad de aproximare.

Toate acestea sunt, credem, suficiente motivații pentru studiul utilității prelucrărilor morfologice pentru problemele de CI, sau, mai general spus, pentru înțelegerea imaginilor. Mai credem că, și am susținut în lucrarea de față această aserțiune prin multiple argumente, cadrul teoretic mare care permite abordarea riguroasă a problemelor complexe de CI e ceea ce am numit “domeniul fuzzy”.

BIBLIOGRAFIE

Articole, cărți

[Agrawal et al., 2005] A. Agrawal, R. Raskar, S. K. Nayar and Y. Li, "Removing Photography Artifacts using Gradient Projection and Flash-Exposure Sampling", Proceedings of ACM SIGGRAPH 2005, Vol. 24, Issue 3, August, New York, USA.

[Albiol, Torres & Delp 2001] A. Albiol, L. Torres & E.J.Delp, "Optimum color spaces for skin detection, International Conference in Image Processing, vol.1, pp. 122-124.

[Balaș & Balaș, 2007] V. E. Balaș & M.M. Balaș, "Postmodernism and control engineering", EUSFLAT.

[Benoit & Foulloy, 2002] E. Benoit & L. Foulloy, "Fuzzy nominal scales", Proc. of IMEKO TC7 Symposium, Poland, Cracow, pp. 21-25.

[Benoit et al., 1994] E.Benoit, G. Mauris and L. Foulloy, "A fuzzy colour sensor", Proc. of XIII IMEKO World congress, Torino, pp. 1015-1020.

[Benoit & Foulloy, 2003] E. Benoit & L. Foulloy, "Towards fuzzy nominal scales Measurement", ed. Amsterdam: Elsevier, ISSN 0263-2241, vol. 34(1) (Fundamental of Measurement), pp. 49-55.

[Berka, 1983] Berka K., 1983, "Scales of measurement; a critical analysis of the concept of scales and of their function in the theory of measurement", Language, Logic and Method, ed. Dordrecht: D. Reidel Publishing Company, pp.1-73, apud. [Benoit, 2010].

[Benoit, 2010] E. Benoit , "An ordinal metrical scale built on a fuzzy nominal scale", 13th IMEKO TC1-TC7 Joint Symposium IOP Publishing, Journal of Physics: Conference Series 238.

[Beu, 1999] T.A.Beu , "Calcul numeric in C", capitolul "Aproximarea funcțiilor tabelate", p. 202-242, Ed. Microinformatica, Cluj-Napoca

[Blaga, 1937] L. Blaga, Trilogia culturii. III, Geneza metaforei și sensul culturii, p. 31-36, Ed. Humanitas, 1994

[Bulea 2003] M. Bulea, "Prelucrarea imaginilor și recunoașterea formelor (teorie și aplicații)", ed. Academiei Române, București

- [Chang & Zhiqiang 2007] Chang, F & Zhiqiang, T, “A region-based skin color detection algorithm”, Proceedings of the 11-th Pacific-Asia conference on Advances in knowledge discovery and data mining, Nanjing, China, pp. 417-424.
- [Coombs, 1951] C. H. Coombs, “Mathematical models in psychological scaling”, Journal of the American Statistical Association, Alexandria, American Statistical Association, vol. 46 (256) pp. 480-489, apud. [Benoit, 2010].
- [Carron & Lambert, 1996] T. Carron & P.Lambert, ”Symbolic Fusion of Hue-Chroma-Intensity Features for Region Segmentation”, proc. IEEE Conference on Image Processing, Lausanne, Switzerland, apud [Vertan & Buzuloiu, 2000].
- [Dimitrov, 2002] V. Dimitrov, “Introduction to fuzziology”, Fuzzy logic: a framework for the new millennium, Studies in fuzzyness and soft computing, vol. 81, Eds. Vladimir Dimitrov & Victor Korotkich, ed. Physica-Verlag Heidelberg New York.
- [de Dios & Garcia 2003] de Dios, J. and Garcia N., “Face detection based on a new color space YCgCr”, Proceedings of IEEE International Conference on Image Processing, Vol. 3, 14-17 Sept., IEEE Computer Society, Barcelona, Spain, pages 909-912.
- [Do & Vetterli 2005] M. N. Do & M. Vetterli, “The contourlet transform: an efficient directional multiresolution image representation”, proc. IEEE Transactions Image on Processing, vol. 14 (12), pp.2091-2106. [Donoho, 1999] D. L. Donoho, ”Wedgelets: nearly-minimax estimation of edges”, Ann. Statist, pp.859—897.
- [Donoho et al., 2001] D. L. Donoho, X. Huo, I. Jermyn, P. Jones, G. Lerman, O. Levi and F. Natterer, ”Beamlets and Multiscale Image Analysis”, proc. Multiscale and Multiresolution Methods, ed. Springer, pp.149—196 .
- [Eisemann & Durand 2004] Elmar Eisemann and Fredo Durand, “Flash Photography Enhancement via Intrinsic Relighting”, Proceedings of ACM SIGGRAPH 2004, Vol. 23, Issue 3, August, New York, USA.
- [Finkelstein, 1975] L. Finkelstein, “Representation by symbol systems as an extension of the concept of measurement”, Kybernetes , vol. 4, pp. 215-223, apud. [Benoit, 2010].
- [Flondor, 2003] Paul Flondor, “Dynamics and approximate reasoning”, IAFA, București.
- [Foulloy & Benoit, 2006] L. Foulloy & E. Benoit, “Building a class of fuzzy equivalence relations”, Fuzzy Sets and Systems, no. 157, pp. 1417- 1437.
- [Georgescu 1, 2009] O.F. Georgescu, “Accuracy and Speed for Device Color Spaces Conversions, CSCS17 at Politehnica University of Bucharest, May 26-29.
- [Georgescu 2, 2009] O.F. Georgescu, C.Tudose, C. Odubășteanu and C.Munteanu, “Reconstruction of Three-Dimensional Objects from Parallel Planar Sections - Analysis, Solutions and Implementation of Double Branching Problem”, CSCS17 at Politehnica University of Bucharest, May 26-29.

[Georgescu, 2010] O.F. Georgescu, “Determining the solution space form for an indeterminate pattern recognition problem. The ‘skin detection’ case”, *Annals of DAAM for 2010 & Proceedings*, pp. 1491-1492.

[Georgescu, 2011] O.F. Georgescu, “Modelling the solution space form for an indeterminate pattern recognition problem. The ‘skin detection’ case”, CSCS18 at Politehnica University of Bucharest, May 24-27.

[Georgescu 1, 2012] O.F. Georgescu, “Reducing false positives by marking and overclassifying”, *Scientific Bulletin of UPB, C series*.

[Georgescu 2, 2012] Octavian Georgescu, Florica Moldoveanu, “An abstract view on pattern recognition based on correlation”, accepted for publication in *Scientific Bulletin of UPB, C series*.

[Grabisch et al., 2000] M. Grabisch, “Fuzzy Integral for Classification and Feature Extraction”, eds. M. Grabisch, T. Murofushi, M. Sugeno, and J. Kacprzyk, *Fuzzy Measures and Integrals - Theory and Applications*, pp.415-434. Physica Verlag.

[Gomez, 2000] G. Gomez, “On selecting color components for skin detection”, *Proc of ICPR*, Volume no 2, pp. 961-964.

[Gomez & Morales 2002] G. Gomez & E. Morales, “Automatic feature construction and a simple rule induction algorithm for skin detection”, *ICML Workshop On Machine Learning in computer Vision*, pp. 31-38.

[Gonzalez et al. 2010] J. Gonzalez, M. Rodriguez, J. Pulido & J. Perez, “Detecting skin in face recognition systems: A colour spaces study”, *Digital Signal Processing*, Vol. 20, May 2010.

[Gonzalez, 2008] R.C. Gonzalez, “*Digital Image Processing (3rd ed.)*”, ed. Pearson Prentice Hall, New Jersey

[Gold & Shankar, 2001] C. Gold & D. Thibault, “Map generalization by skeleton retraction”, in *proc. 20th Int. Cartographic Conf. (ICC 2001)*, pp. 2072-2081.

[Hsu et al, 2002] R. Hsu, M. Abdek-Mottaleb, and A.K. Jain, “Face detection in color images”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24, Issue 5, May, pp.696–706.

[J. M. DiCarlo et al, 2001] J. M. DiCarlo, F. Xiao & B. A. Wandell, “Illuminating Illumination”, *Ninth Color Imaging Conference*, pp.27-34.

[Jones & Rehg, 2002] M.J. Jones and J.M. Rehg, “Statistical color models with application to skin detection”, in *International Journal of Computer Vision*, (IJCV) 46(1), pp. 81–96.

[Kakumanu et.al, 2006] P. Kakumanu, S. Makrogiannis and N. Bourbakis, “A survey of skin-color modeling and detection methods”, ITRI/Department of Computer Science and Engineering, Wright State University, Dayton OH 45435, USA.

[Kolata, 1991] G. Kolata, "Math Problem, Long Baffling, Slow Yields", *The New York Times*.

[Kosko, 1991] B. Kosko, "Neural Networks and Fuzzy Systems", ed. Prentice Hall, Englewood Cliffs, NJ.

[Kruschke, 2011] John K. Kruschke, "Doing Bayesian Data Analysis. A Tutorial with R and BUGS", ed. Academic Press / Elsevier, 2011

[N. van Lier, 2002] Hubert N. van Lier, "Land Use Planning: A Key to Sustainable Development", XXX International Symposium Actual Tasks on Agricultural Engineering, Croatia, 2002

[McNeill & Freiberger, 1993] D. McNeill & P. Freiberger, "Fuzzy Logic", ed. Simon & Schuster, New York, pp. 46-48.

[Meyer1 & Coifman, 1997] F.G. Meyer1 and R. R. Coifman, "Brushlets: a tool for directional image analysis and Image Compression", Applied and computational harmonic analysis, vol.4, pp.147–187.

[Moisil, 1975] Grigore Moisil, "Leccións despre logica ratiunamentului nuanat", ed. Științifică și enciclopedică.

[Negoiță, 2002] C. V. Negoită, "Postmodernism, Cybernetics and Fuzzy Set Theory", Proc. of IPMU, Annecy, France, pp. 681-686, apud [Balaș & Balaș, 2007].

[Negoiță, 2006] C. V. Negoită, "Concert la Carnegie Hall", p. 118, Ed. Paralela 45, 2006

[Negoiță & Ralescu, 1974] C. V. Negoită & D. A. Ralescu "Mulțimi vagi și aplicațiile lor", ed. Tehnică.

[Neumaier, 2003] A. Neumaier, "Fuzzy modeling in terms of surprise", Fuzzy Sets and Systems, Vol. 135, Issue 1, Elsevier North-Holland, Inc. Amsterdam, The Netherlands .

[Pavlidis, 1982] T. Pavlidis, "Algorithms for Graphics and Image Processing" , ed. Computer Science Press, apud M. Bulea, Prelucrarea imaginilor și recunoașterea formelor, ed. Academiei Române, p.15.

[Pesquet-Popescu and Popescu, 2010] Pesquet-Popescu B. and Popescu S., "Analiza cu undine (wavelets). Fractali – modele stocastice", ed. Politehnica Press

[Peyré & Mallat, 2005] G. Peyré & S. Mallat, "Surface Compression With Geometric Bandelets", ACM Transactions on Graphics, Vol. 24(3), (Proc. of SIGGRAPH'05), pp.601-608.

[Petschnigg et al, 2004] G. Petschnigg, M. Agrawala, H. Hoppe, R.Szeliski, M. Cohen and K. Toyama, "Flash Photography Enhancement via Intrinsic Relighting", Proceedings of ACM SIGGRAPH 2004, Vol. 23, Issue 3, August, New York, USA.

[Rao & Shankar, 2011] P.B. Rao & N. R. Shankar, "Ranking Fuzzy Numbers with a Distance Method using Circumcenter of Centroids and an Index of Modality", Advances in Fuzzy Systems, ed. Hindawi Publishing Corporation.

[Rosch, 2010] E. Rosch, "Principles of Categorization", Cognition and categorization pp. 27-48, University of California, Berkeley.

- [Rosenberg, 2010] Michael J. Rosenberg, “The Agile Approach to Adaptive Research: Optimizing Efficiency in Clinical Development”, ed. Wiley, 2010
- [Shakhatreh & Hayajneh 2010] M. A. Shakhatreh & M. A. Hayajneh, “S-H Fuzzy Partition and Fuzzy Equivalence Relation”, *Int. J. Contemp., Math. Sciences*, Vol. 5 (8), online ed., pp.381 – 392.
- [Starck et al., 2002] J. L. Starck, E. J. Candes and D. L. Donoho, “The curvelet transform for image denoising”, *proc. IEEE Trans. Image Process*, vol. 11 (6), pp.670-684.
- [Stein & R. Shakarchi, 2003] E. Stein & R. Shakarchi, *Fourier Analysis: An introduction*, ed. Princeton University Press.
- [Sweldens, 1997] W. Sweldens, “The lifting scheme: a construction of second generation wavelets”, *Siam J. Math. Anal.*, Vol. 29 (2).
- [Terrillon et.al. 2000] J.C. Terrillon, M.N. Shirazi, H. Fukamachi, and S. Akamatsu, “Comparative performance of different skin chrominance models and chrominance for the automatic detection of human faces in color images”, *Proceedings of the IEEE International conference of Face and Gesture Recognition*, pp.54–61.
- [Teodorescu, 2011] H.-N. L. Teodorescu, “On the meaning of approximate reasoning – An unassuming subsidiary to Lotfi Zadeh’s paper dedicated to the memory of Grigore Moisil –”, *Int. J. of Computers, Communications & Control*, Vol. VI, No. 3, pp. 577-580.
- [Vertan, 2009] C. Vertan, “Parallel Coordinates-based Fuzzy Color Distance and its Applications”, *IFSA-EUSFLAT*.
- [Vertan & Boujemaa, 2000] C. Vertan and N. Boujemaa, "Embedding fuzzy logic in content based image retrieval", *proc. AFIPS, Atlanta, Georgia*, pp. 85-90.
- [Vertan & Buzuloiu, 2000] C. Vertan & V. Buzuloiu, “Fuzzy Nonlinear Filtering of Color Images: A Survey”, *Fuzzy Technique in Image Processing, Studies in fuzzyness and soft computing*, vol. 81, Eds. Etienne E. Kerre & Mike Nachtgeael, ed. Physica-Verlag Heidelberg New York pp. 248-266.
- [Vertan et al., 2000] C. Vertan, N. Boujemaa and V. Buzuloiu, “A fuzzy color credibility approach to color image filtering”, *proc. International Conference on Image Processing*, vol. 2, pp. 808-811.
- [Vertan & Ciuc, 2007] Constantin Vertan & Mihai Ciuc, "Tehnici fundamentale de prelucrarea și analiza imaginilor", ed. Matrix Rom, București.
- [Vezhnevets et al. 2003] V. Vezhnevets, V. Sazonov & A. Andreeva, “A Survey on Pixel-Based Skin Color Detection Techniques”, *Graphics and Media Laboratory, Faculty of Computational Mathematics and Cybernetics, Moscow State University, Russia*.
- [Zadeh, 1965] L. A. Zadeh, “Fuzzy Sets”, *Department of Electrical Engineering and Electronics Research Laboratory, University of California, Berkeley*.

[Zadeh, 1971] L.A. Zadeh, "Quantitative fuzzy semantics", Information Sciences, Vol. 3, 1971, pp. 159-176

[Zadeh, 1978] L.A. Zadeh, "Fuzzy sets as a basis for a theory of possibility", Fuzzy Sets and Systems, Vol. 1, pp.3-28.

[Zadeh, 1979] L. Zadeh, "On the validity of Dempster's rule of combination", Memo M79/24, Univ. of California, Berkeley, USA.

[Zadeh, 1984] L. Zadeh, Book review: "A mathematical theory of evidence," The AI Magazine, Vol. 5(3), pp. 81-83.

[Zadeh, 1986] L. Zadeh, "A simple view of the Dempster-Shafer Theory of Evidence and its implication for the rule of combination", The AI Magazine, vol. 7 (2), pp. 85-90.

[Zadeh, 1996] L.A. Zadeh, "Fuzzy logic = computing with words", Fuzzy Systems, IEEE Transactions on, Vol. 4 Issue 2, pp. 103–111.

[Zadeh, 2002] L.A. Zadeh, "Toward a perception-based theory of probabilistic reasoning", Fuzzy logic: a framework for the new millenium, Studies in fuzzyness and soft computing, vol. 81, Eds. Vladimir Dimitrov & Victor Korotkich, ed. Physica Verlag New York.

[Zemankova & Kandel, 1984] M. Zemankova-Leech & A. Kandel , "Fuzzy Relational Databases—a Key to Expert Systems", Interdisciplinary Systems Research, Verlag TUV: Rheinland, Koln, p. 178.

[Zheng, Zhang & Wang, 2004] Q.F. Zheng, M.J. Zhang and W.Q. Wang, " A hybrid approach to detect adult web images", PCM (2), pp. 609–616.

[Yang & Ahuja, 1999] M. Yang & N. Ahuja, "Detecting human faces in color images", International Conference in Image Processing, Vol. 1, pp. 127-130.

[Yang & Ahuja, 1999] M. Yang and N. Ahuja, "Gaussian mixture model for human skin color and its applications in image and video databases", SPIE Proceedings Storage and Retrieval for Image and Video Databases VII, San Jose, USA, pp.458–466.

Link-uri

[AdobeRGB1998] <http://www.adobe.com/digitalimag/pdfs/AdobeRGB1998.pdf>

[ICC] <http://www.color.org>

[if fl] <http://egovbus.net/rdl/articole/No1Art50.pdf>

[cl cr] <http://www.exo.net/~pauld/workshops/Color2011.html>

[crb grn] http://www.hydrop.pub.ro/vn_cap361.pdf

[DSmT] <http://fs.gallup.unm.edu/DSmT.htm>

[eig img] http://www.stanford.edu/class/ee368/Handouts/10-EigenImagesMay25_27.pdf

[fz ap] <http://www.scribde.com/tehnica-mecanica/Fuzzy-sets102101833.php>

[fz hedge] <http://statpac.org/walonick/systems-theory.htm>.

[fz lz] www.nvc.cs.vt.edu/ictai06/Zadeh.ppt

[fz prot] <http://statpac.org/walonick/systems-theory.htm>

[fl ppt] www.comp.dit.ie/rlawlor/KE/notes/Fuzzy%20Logic.ppt

[fs ppt] www.cs.ru.ac.za/courses/Honours/ai/FuzzySystems/FuzzySystemsP2.ppt

[pb VN] <http://convorbiri-literare.dntis.ro/NEGOITamar6.htm>

[wv hsn] <http://lumenosys.com/wavelet-transforms.html>