

UNIVERSITATEA "BABEȘ-BOLYAI"  
FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ

# Recunoașterea obiectelor în imagini

*Referat de doctorat*

**Doctorand**  
Radu-Lucian Lupșa

**Conducător științific**  
Prof. dr. Leon Țâmbulea

2003

# Cuprins

<b>1</b>	<b>Introducere</b>	<b>2</b>
<b>2</b>	<b>Detectia muchiilor și conturilor</b>	<b>4</b>
2.1	Problema detectiei conturilor	4
2.2	Metode bazate pe histerezis	5
2.3	Alte metode	6
2.3.1	Metoda lui Nalwa	6
2.3.2	Metoda lui Bergholm	6
2.4	Masca pentru completarea conturului	7
2.5	Contururi deformabile (Șerpi)	8
2.5.1	Interpretare fizică. Metode de simulare	9
2.6	Extensii ale modelului conturilor active	10
2.6.1	Gaussiana multirezoluție	10
2.6.2	Presiunea	10
2.6.3	Funcția distanță	11
2.6.4	Forțe introduse interactiv	11
2.7	Contururi active definite implicit	11
<b>3</b>	<b>Recunoașterea și sinteza texturilor</b>	<b>13</b>
3.1	Texturi	13
3.2	Modelele inițiale în recunoașterea texturilor	14
3.2.1	K-goane	14
3.3	Modele Markov	14
3.4	Filtrare spațială	15
3.5	Modele bazate pe transformate wavelets	16
3.5.1	Descompuneri wavelets utilizate	17
3.5.2	Semnăturile energetice	17
3.5.3	Semnăturile histogramelor	18
3.5.4	Matrici de corelație	18
<b>4</b>	<b>Concluzii și cercetări viitoare</b>	<b>20</b>

# Capitolul 1

## Introducere

Problema recunoașterii obiectelor în imagini este probabil cea mai mare și complexă problemă în analiza și prelucrarea imaginilor, și are legături strânse cu alte domenii ale matematicii și informaticii, o mare parte intrând în domeniul inteligenței artificiale.

Deși problema recunoașterii obiectelor în imagini este veche, domeniul este foarte probabil la început de drum, departe de-a se fi maturizat. Există în momentul de față trei categorii de cercetări în domeniu:

1. Găsirea unei soluții simple la o problemă simplă și concretă. E vorba aici de soluții simple, lipsite de eleganță, și mai ales de posibilități de generalizare, dar care rezolvă simplu și rapid o problemă concretă. De exemplu, numărarea unor boabe de cafea dintr-o imagine (boabele putându-se suprapune parțial): rezolvarea simplă este să se calculeze distanța din fiecare punct la cea mai apropiată margine a unui bob, și să se numere apoi punctele de maxim ale acestei distanțe.
2. Rezolvarea unor subprobleme. Cele mai importante subprobleme sunt: detecția muchiilor, segmentarea imaginilor, recunoașterea texturilor.
3. Rezolvarea problemelor dificile. În acest domeniu sunt câteva soluții mulțumitoare, sau chiar bune, la câteva probleme. Probabil problema cea mai studiată din această categorie este recunoașterea optică a caracterelor. Soluțiile cele mai performante existente în acest moment combină elemente de prelucrarea și analiza imaginilor cu elemente de prelucrarea limbajului natural.

Între problemele din ultima categorie, se numără câteva care beneficiază de finanțări semnificative, întrucât implică mize importante din punct de vedere economic sau militar. Câteva dintre această sunt analiza imaginilor

provenite de la avioanele de recunoaștere sau de la sateliți, sau recunoașterea fețelor unor oameni din imaginile preluate de videocamerele de supraveghere. Evident, rezultatele acestor cercetări sunt strict secrete, și foarte puține dintre ele sunt publicate, de regulă la nivel de reviste de popularizare.

# Capitolul 2

## Detecția muchiilor și contururilor

### 2.1 Problema detecției contururilor

Problema detecției contururilor este următoarea: dându-se o imagine, să se găsească contururile obiectelor ce apar în imagine. De notat că imaginea în cauză poate fi:

- o fotografie 2D a unei scene 3D,
- o fotografie aeriană sau din satelit (imaginea 2D este o simplă reducere la scară, scena originala fiind în esență tot 2D),
- o secțiune obținută prin tomografie,
- o ecograma sau o secțiune generată de un sonar.

Se consideră că detecția contururilor trebuie să fie primul pas în vederea recunoașterii obiectelor dintr-o imagine. Din această cauză, problema detecției contururilor s-a bucurat de o atenție deosebită și numărul metodelor de rezolvare propuse este imens.

Punctul de plecare în detecția contururilor îl constituie observația că punctele de pe un contur trebuie să fie puncte în care luminozitatea imaginii variază brusc. Din acest motiv, multe din metodele de detecție a contururilor caută acele puncte din imagine în care norma gradientului este cât mai mare.

Cum operatorii de derivare, precum gradientul, sunt foarte sensibili la zgomot, o schemă de reducere a acestuia este aproape întotdeauna necesară ca etapă premergătoare în procesul de determinare a muchiilor. Tradițional, filtrarea se face prin medierea ponderată a fiecărui pixel cu pixelii vecini.

Ponderile alese în schemele de filtrare sunt cel mai adesea aproximări ale clopotului lui Gauss bidimensional. Se poate arăta de altfel că aplicarea repetată a oricărui filtru liniar converge către rezultatul aplicării unui filtru gaussian.

Din păcate, metodele de reducere a zgomotului au aproape întotdeauna dezavantajul de-a face imaginea neclară sau de-a modifica semnificativ imaginea; în cazuri extreme, poate fi mai dificil pentru om să perceapă imaginea “curățată” de zgomot decât imaginea originală, zgomotoasă.

Cea de-a doua problemă în detectarea contururilor prin selectarea punctelor cu valori mari ale normei gradientului este aceea că, mai ales în urma aplicării filtrelor pentru micșorarea zgomotului, liniile pe care gradientul este mare — și pe care le vom detecta ca fiind contururi — au tendința de-a se “îngroșa”. Din acest motiv, se cere ca punctul declarat ca fiind punct de contur să fie punct de maxim local al modulului gradientului, pe direcția gradientului.

Aceste observații conduc la următoarea metodă de detecție a contururilor: se declară ca fiind puncte de contur acele puncte din imagine care satisfac simultan condițiile:

1. modulul gradientului este mai mare decât un anumit prag
2. laplacianul este zero (în practică, se va pune de fapt condiția ca valoarea absolută a laplacianului să fie mai mică decât o valoare de prag)

Metoda de mai sus este așa-numitul *filtrul laplacian*.

O astfel de metodă nu este de fapt un *detector de contur* ci un *detector de muchii*, deoarece nu garantează detectarea unui contur închis, ci detectează doar muchii disperate.

## 2.2 Metode bazate pe histerezis

Filtrul laplacian are cusurul că este foarte sensibil la valorile alese pentru praguri. Mai mult, aceeași valoare de prag va duce la “detecția” unui număr mare de muchii secundare în unele zone ale imaginii, și în același timp la omiterea unor muchii importante în alte zone ale imaginii. Problema principală este faptul că pragul pentru gradient este același pentru toate zonele imaginilor, indiferent de condițiile locale de iluminare.

Un algoritm de referință în detecția contururilor este algoritmul lui Canny [5], datând din 1986, și care constă în următorii pași:

1. se aplică imaginii originale o convoluție cu o gaussiană

2. pentru pixelii imaginii rezultate, se calculeaza modulul și direcția gradientului (se aproximează de fapt prin diferențe finite).
3. pentru fazele următoare se rețin doar acele puncte în care norma gradientului este mai mare decât norma gradientului în pixelii vecini pe direcția gradientului
4. pixelii, trecuți de etapa anterioară, cu modulul gradientului mai mare de o valoare prag (numita prag superior) sunt declarați ca fiind puncte de contur
5. pixelii, trecuți de pasul 3 și având modulul gradientului mai mic decât pragul superior dar mai mare decât un al doilea prag (numit prag inferior) sunt declarați puncte de contur doar dacă sunt vecini cu alte puncte de contur (selectate la pasul anterior sau în cadrul pasului curent).

Spunem că este un algoritm bazat pe histerezis deoarece “urmărirea” unui contur este declanșată la întâlnirea unei valori ce depășește pragul superior, și, odată începută, continuă până ce valoarea cade sub pragul inferior.

## 2.3 Alte metode

### 2.3.1 Metoda lui Nalwa

Metoda lui Nalwa consta în a încerca potrivirea, în sensul minimizării erorii pătratice, a unei funcții sinus hiperbolic pe o fereastră de dimensiune  $5 \times 5$  în jurul punctului considerat. Dacă potrivirea funcției sinus hiperbolic este mai bună decât potrivirea unei funcții quadratice atunci punctul este declarat punct de contur.

### 2.3.2 Metoda lui Bergholm

Metoda lui Bergholm poate fi considerată o metodă multirezoluție. Metoda pleacă de la imaginea filtrată cu o gaussiană cu coeficientul de dispersie ridicat. Această imagine va conține deci numai trăsăturile majore. Asupra ei se aplică filtrul laplacian pentru a detecta muchiile. Muchiile detectate în imaginea “de rezoluție mică” vor fi urmărite apoi, de la o imagine la alta, în imagini filtrate cu gaussiene cu dispersie din ce în ce mai mică.



Figura 2.1: Masca pentru convoluție; valorile mari sunt figurate cu negru, valorile mici cu alb

## 2.4 Masca pentru completarea conturului

Ideea de bază a algoritmului constă în a detecta liniile pe baza maximului normei gradientului și a prelungirii muchiilor prin linii (imaginare) trecând prin linii detectate mai înainte [11].

În acest scop, algoritmul de detectare a muchiilor conține următoarele etape:

1. Calculează, în fiecare pixel, gradientul și norma gradientului; în cele ce urmează vom numi norma gradientului *tăria muchiei* în acel punct.
2. Filtrează tăria muchiei printr-o convoluție cu ajutorul unei măști.
3. Aplică o scheletizare rezultatului precedent și declară ca muchii “crestele” tăriei rezultate în urma filtrării

O versiune mai rapidă a algoritmului calculează, în fiecare punct, integrala produsului dintre tăria muchiei și valoarea măștii, cu masca rotită după direcția gradientului în acel punct. În felul acesta, continuarea muchiei este căutată numai pe direcția perpendiculară pe gradient. O metodă mai precisă, dar mult mai costisitoare în privința timpului de calcul, constă în găsirea orientării măștii pentru care integrala are valoare maximă. Aceasta din urmă este capabilă să refacă continuitatea unei linii punctate, unde direcția calculată a gradientului în fiecare punct este complet aleatoare.

Masca este creată cu următoarea funcție:

$$f(x, y) = \begin{cases} \frac{ax^2 - y}{a^2 x^4} & , |y| \leq x^2 \\ 0 & , |y| > x^2 \end{cases} \quad (2.1)$$

Reprezentarea grafică a măștii, calculate pentru  $a = 1$ , este prezentată în figura 2.1.

Algoritmul sus-menționat privilegiază muchiile lungi, chiar dacă contrastul este mic între zonele între care se găsește muchia, și chiar în condiții de zgomot local sau texturi destul de pronunțate.



O metodă promițătoare pentru îmbunătățirea algoritmului este găsirea unei scheme adaptive pentru dimensiunea măștii și pentru parametrul  $a$ .

## 2.5 Contururi deformabile (Șerpi)

Metodele de detecție a muchiilor au dezavantajul că nu construiesc, în general, un contur continuu și închis în jurul elementelor din imagine. O abordare diferită a problemei detecției conturilor este următoarea: În fiecare punct al imaginii se estimează o “măsură a discontinuității”; aceasta va evalua posibilitatea ca prin acel punct să treacă un contur.

Se ia apoi o curbă închisă, de exemplu un cerc, plasată într-un loc relativ arbitrar al imaginii. Această curbă va evolua, conform unui algoritm, având predilecție în a se stabiliza în zone în care măsura posibilității existenței unui contur este mare. Această curbă este numită *contur activ* (*active contour*) sau “șarpe” (*snake*).

Modelul original al conturului activ a fost introdus de Terzopoulos et. al. [2] la sfârșitul anilor 1980. Modelul este construit, mai exact, în modul următor:

Se consideră o curbă în scriere parametrică  $f : [0, 1] \rightarrow \mathbb{R}^2$ . Se consideră acum funcționala

$$\mathcal{E}(f) = \mathcal{S}(f) + \mathcal{P}(f), \quad (2.2)$$

unde

$$\mathcal{S}(f) = \frac{1}{2} \int_0^1 \alpha(s) \|f'(s)\|^2 ds + \frac{1}{2} \int_0^1 \beta(s) \|f''(s)\|^2 ds \quad (2.3)$$

și

$$\mathcal{P}(f) = \int P(f(s)) ds \quad (2.4)$$

Cautarea conturului se face acum prin minimizarea funcționalei  $\mathcal{E}(f)$ .

Componenta  $\mathcal{S}(f)$  reprezintă “energia internă” de deformare a conturului activ. Primul termen se opune întinderii conturului activ; conturul se comportă astfel ca un elastic care tinde să se scurteze. Al doilea termen se opune curburii, și face conturul să se comporte ca o bară elastică, ce se opune deformării. În principiu,  $\alpha(s)$  și  $\beta(s)$  pot fi constante.

Componenta  $\mathcal{P}(f)$  reprezintă “energia potențială”. Câmpul de potențial  $P$  are rolul de-a dirija conturul către punctele de gradient maxim din imagine. Varianta cea mai simplă este să definim

$$P = -w |\nabla(G_\sigma \star I)|^2 \quad (2.5)$$

unde  $w \in \mathbb{R}, w > 0$ , este un coeficient pozitiv,  $G_\sigma$  este gaussiană bidimensională de dispersie  $\sigma^2$ , iar  $I$  reprezintă imaginea în care căutăm contururi. Acest câmp de potențial va dirija conturul activ către punctele de variație rapidă a intensității.

Alte posibilități de construcție ale câmpului de potențial  $P$  conduc la căutarea altor trăsături în imagine. De exemplu, dacă ceea ce se caută sunt linii, putem defini

$$P = w(G_\sigma \star I) \quad (2.6)$$

Pentru  $w > 0$  conturul activ va tinde să găsească liniile negre (de intensitate minimă), iar pentru  $w < 0$  conturul va cauta liniile albe (de intensitate maximă).

Trebuie subliniat faptul că alegerea coeficienților  $\alpha$  și  $\beta$  este critică: coeficienți prea mici conduc la găsirea unor soluții pe neregularități ale câmpului de potențial, iar valori prea mari conduc la imposibilitatea detectării unor contururi mai neregulate.

O problemă critică similară apare în privința coeficientului de dispersie  $\sigma$  al filtrului gaussian cu care se produce câmpul de potențial.

### 2.5.1 Interpretare fizică. Metode de simulare

Minimizarea expresiei (2.2) este o problemă de calcul variational; soluția va verifica ecuația cu derivate parțiale

$$\frac{\partial}{\partial s} \left( \alpha \frac{\partial f(s)}{\partial s} \right) - \frac{\partial^2}{\partial s^2} \left( \beta \frac{\partial^2 f(s)}{\partial s^2} \right) - (\nabla P)(f(s)) = 0 \quad (2.7)$$

Formula anterioară poate fi interpretată, prin analogie cu fenomenele din fizică, cu un echilibru de forțe. Cu această modelare, relația (2.7) se scrie ca

$$F_{int} + F_{pot} = 0 \quad (2.8)$$

unde

$$F_{int} = \frac{\partial}{\partial s} \left( \alpha \frac{\partial f(s)}{\partial s} \right) - \frac{\partial^2}{\partial s^2} \left( \beta \frac{\partial^2 f(s)}{\partial s^2} \right) \quad (2.9)$$

și

$$F_{pot} = -(\nabla P)(f(s)) \quad (2.10)$$

Păstrând analogia fizică, se poate scrie ecuația evoluției poziției unei bare elastice. Aceasta înseamnă că poziția “șarpelui” este dependentă de timp, așadar se scrie parametric  $f = f(s, t)$ , unde  $s$  este parametrul,  $s \in [0, 1]$ , iar  $t$  reprezintă timpul. Ecuația ce descrie mișcarea “șarpelui” este atunci:

$$\mu \frac{\partial^2 f}{\partial t^2} = -\gamma \frac{\partial f}{\partial t} + F_{int}(f) + F_{pot}(f) \quad (2.11)$$

unde  $\mu$  reprezintă masa unitară a “șarpelui”, iar  $\gamma$  reprezintă coeficientul de viscozitate.

Evoluția conturului activ conform ecuației (2.11) se poate simula prin diferențe finite. Conturul activ pleacă de la o configurație oarecare, și, după un număr de iterații conform scrierii cu diferențe finite a ecuației de evoluție (2.11), se stabilizează pe un contur închis.

Masa unitară  $\mu$  se ia adesea 0 pentru a evita ca “șarpele” să “sară” peste conturul de detectat.

## 2.6 Extensii ale modelului conturilor active

De-a lungul timpului au fost propuse diverse îmbunătățiri ale modelului, și diverse formule de calcul pentru câmpul de potențial.

### 2.6.1 Gaussiană multirezoluție

Alegerea coeficientului de dispersie  $\sigma$  al gaussianei cu care se face filtrarea imaginii este critică — așa cum este de altfel și în alte metode de detecție a conturilor care au ca etapă preliminară filtrarea cu o gaussiană. În cazul modelului conturului activ, alegerea unei valori prea mici pentru  $\sigma$  face ca “raza de atracție” a punctelor de contur să fie mică, și ca urmare conturul de detectat nu va fi găsit dacă conturul activ inițial este prea departe. Pe de altă parte, o valoare mare a coeficientului de dispersie  $\sigma$  dăunează preciziei găsirii conturului. O soluție de compromis este execuția algoritmului pentru valori din ce în ce mai mici pentru  $\sigma$ , de fiecare dată pornind de la configurația pe care s-a stabilizat conturul activ la valoarea  $\sigma$  anterioară.

### 2.6.2 Presiunea

O altă îmbunătățire a modelului clasic este introducerea unei presiuni în interiorul curbei, presiune care tinde să “umfle” “șarpele” cât timp nu i se opune faptul de a se fi stabilizat pe un contur foarte bine evidențiat.

Presiunea se introduce ca un termen

$$F_p(f) = w_p N(f) \tag{2.12}$$

unde  $N(f)$  este normala interioară unitară a direcției curbei  $f$ .

Contururile active, în varianta tridimensională, ce folosesc presiune, se mai numesc “baloane”.

Semnul coeficientului  $w_p$  determină sensul de acțiune al presiunii: dacă  $w_p < 0$  atunci presiunea tinde să umfle conturul activ, iar dacă  $w_p > 0$  atunci ea tinde să-l restrângă.

Avantajul introducerii presiunii este că nu mai necesită inițializarea conturului activ în vecinătatea conturului de detectat. În prezența presiunii, conturul activ va baleia imaginea până la întâlnirea unui contur suficient de bine evidențiat.

Alegerea valorii coeficientului  $w_p$  este problematică: o valoare prea mică determină oprirea “șarpelui” pe un contur fals; o valoare prea mare determină “șarpele” să sară peste conturul de detectat.

### 2.6.3 Funcția distanță

Metoda conturilor active poate fi aplicată după un algoritm de detectare a conturilor, sau asupra unei imagini în care, pe o cale sau alta, au fost deja evidențiate anumite puncte de contur. Se presupune însă că au fost detectate doar fragmente de contur, fragmente ce trebuie reunite într-un contur închis.

În această situație, conturul activ poate fi folosit pentru a reuni fragmentele de contur existente. Pentru aceasta, se introduce ca și câmp de potențial funcția distanță, care asociază fiecărui punct distanța până la cel mai apropiat punct de pe conturul detectat.

### 2.6.4 Forțe introduse interactiv

Există aplicații în care detecția conturilor se face semiautomat; în această situație este necesar un mecanism prin care utilizatorul să poată influența conturul activ, trăgându-l spre anumite puncte, sau îndepărtându-l de anumite puncte.

Pentru a atrage conturul activ spre un anumit punct, se adaugă forța

$$F_a(x) = w_a(p - x) \quad (2.13)$$

unde  $w_a$  este ponderea ( $w_a > 0$ ), iar  $p$  este punctul spre care se dorește atragerea conturului.

Pentru respingere, expresia forței este

$$F_r(x) = w_r \frac{x - p}{\|x - p\|^3} \quad (2.14)$$

## 2.7 Contururi active definite implicit

Exprimarea parametrică a conturului activ are câteva neajunsuri, dintre care cele mai importante sunt:

- dificultăți de modelare în cazul necesității schimbării topologiei conturului activ
- dependența de parametrizarea aleasă

Pentru a înlătura neajunsurile conturilor parametrice, se pot folosi contururi definite implicit, ca mulțimea zerourilor unei funcții  $\Phi : \mathbb{R}^2 \rightarrow \mathbb{R}$ .

Funcția implicită a unui contur activ este legată de reprezentarea parametrică a aceluiași contur prin relația

$$\Phi(f(s)) = 0 \quad (2.15)$$

unde  $s \in \mathbb{R}$  este parametrul reprezentării parametrice, sau, dacă introducem timpul  $t$  în vederea urmăririi evoluției în timp,

$$\Phi(f(s, t), t) = 0 \quad (2.16)$$

Ecuția de evoluție, analoagă relației (2.11), și considerând  $\mu = 0$ , se deduce derivând ecuația anterioară în raport cu  $t$ :

$$\frac{\partial \Phi}{\partial t} + (\nabla \Phi) \frac{\partial f}{\partial t} = 0 \quad (2.17)$$

unde, înlocuind din (2.11),

$$\frac{\partial \Phi}{\partial t} = -(\nabla \Phi) \frac{1}{\gamma} (F_{int} + F_{pot}) \quad (2.18)$$

# Capitolul 3

## Recunoașterea și sinteza texturilor

### 3.1 Texturi

Problema recunoașterii texturilor a fost pusă de Julesz încă din 1962; cu toate acestea, progrese spectaculoase în domeniul recunoașterii și chiar sintezei texturilor au fost realizate abia după 1990.

O textură este un model ce acoperă o suprafață (chiar dacă noțiunea se poate generaliza în  $n$  dimensiuni), și care este perceput unitar de către un om, la o privire superficială. O textură conține de obicei un număr extrem de mare de elemente (muchii, obiecte identificabile), iar un om nu percepe fiecare element în parte, ci ansamblul lor.

Se pune firesc întrebarea (pusă de Julesz în 1962): care sunt elementele caracteristice ale unei texturi, altfel spus, ce deosebește o textură de alta.

Problema recunoașterii texturilor este extrem de complexă. În primul rând, evaluarea este prin definiție subiectivă, deoarece căutăm elementele relevante pentru privitorul uman.

În al doilea rând, texturile pot fi regulate (de exemplu, cărămizile unui zid, firele unui material textil) sau neregulate (de exemplu, pietrișul sau firele de iarbă).

În al treilea rând, așa cum se argumentează în [10], elaborarea unei teorii corecte pentru caracterizarea texturilor nu poate fi făcută independent de sinteza texturilor — strict necesară verificării oricărei ipoteze asupra trăsăturilor definitorii ale unei texturi.

## 3.2 Modelele inițiale în recunoașterea texturilor

### 3.2.1 K-goane

Una din primele metode de abordare a recunoașterii texturilor a fost metoda k-goanelor. Modelul k-goanelor este următorul:

Considerăm imaginea discretă  $I : D \rightarrow C$ , unde  $D \subseteq \mathbb{Z}^2$ , iar  $C \subseteq \mathbb{R}$  poate fi un interval sau o multime finită (discretă) de valori posibile.

Un k-gon este un tuplu  $\alpha = (u_1, u_2, \dots, u_k)$ , unde  $u_i = (\Delta x_i, \Delta y_i)$  este deplasamentul vârfului  $i$  față de originea sistemului de coordonate; originea va fi "centrul" k-gonului. Deplasând centrul k-gonului pe domeniul imaginii, și în condiții de frontieră potrivit alese, putem construi imaginea tuplelor

$$v \mapsto (I(v + u_1), I(v + u_2), \dots, I(v + u_k)), v \in D \quad (3.1)$$

Asociem acum k-gonului  $\alpha$  histograma imaginii definite în (3.1), adică

$$h^{(\alpha)}(b_1, b_2, \dots, b_k) = \frac{1}{|D|} \sum_{v \in D} \prod_{i=1}^k \delta_{b_i, I(v+u_i)} \quad (3.2)$$

unde  $\delta$  este simbolul lui Kronecker ( $\delta_{i,i} = 1, \forall i$ , și  $\delta_{i,j} = 0, \forall i \neq j$ ).

Primele încercări în recunoașterea texturilor au constat în caracterizarea texturilor prin histogramele asociate unui număr de k-goane. Din păcate, abordarea aceasta suferă de o creștere explozivă cu numărul de vârfuri ale k-gonului, și, de îndată ce numărul de poziții în histogramă devine comparabil cu numărul de pixeli din imagine, statistica încetează să mai fie relevantă.

## 3.3 Modele Markov

În modelarea Markov, imaginii  $i$  se asociază un câmp aleator, adică o familie de variabile aleatoare  $X_{i,j}$ , cu  $(i, j) \in D$ .

Se face presupunerea că aceste variabile aleatoare alcătuiesc un camp Markov. Asta înseamnă că distribuția valorii unui pixel nu depinde decât de valorile pixelilor dintr-o anumită vecinătate.

Mai exact,

$$\begin{aligned} P(X_{i_0, j_0} = x_{i_0, j_0} | X_{i, j} = x_{i, j} \forall (i, j) \in D) = \\ = P(X_{i_0, j_0} = x_{i_0, j_0} | X_{i_0+i, j_0+j} = x_{i_0+i, j_0+j} \forall (i, j) \in N) \end{aligned} \quad (3.3)$$

unde  $N$  este mulțimea vecinilor în  $\mathbb{Z}^2$  ai punctului  $(0, 0)$ .

De exemplu, se poate lua  $N = \{(-1, 0), (1, 0), (0, -1), (0, 1)\}$  adică mulțimea vecinilor direcți pe orizontală sau verticală. Modelul care rezultă este modelul Gauss-Markov de ordinul 1; adăugând vecinii de pe diagonală se obține modelul de ordin 2 etc.

Modelul se poate considera cunoscut, altfel spus, mulțimea vecinilor  $N$  se consideră dată, și se încearcă determinarea parametrilor; după cum se poate încerca însăși determinarea vecinătății relevante  $N$ .

Determinarea parametrilor este o problemă dificilă datorită exploziei combinatoriale.

Se vede imediat că determinarea parametrilor câmpului Markov este echivalentă cu construcția histogramelor asociate tuturor k-goanelor conținute în vecinătatea  $N$ .

Recunoașterea texturilor într-o imagine în care apar mai multe texturi este însă o problemă mai complicată, deoarece pentru determinarea parametrilor modelului Markov al unei texturi este necesară segmentarea prealabilă a imaginii. În acest scop se construiește un al doilea model Markov, în care pe lângă valoarea intensității în fiecare pixel apare și etichetarea texturii asociate pixelului.

Modelul Markov oferă posibilități imediate de generare a texturilor. Un algoritm simplu prezentat în [10] pleacă de la o imagine aleatoare, de exemplu zgomot alb, și apoi modifică pe rând valoarea fiecărui pixel folosind distribuția aleatoare dată de modelul Markov.

### 3.4 Filtrare spațială

Metodele de filtrare spațială utilizează variațiile intensității ca metodă de clasificare a texturilor. Printre primele încercări în domeniu a fost aplicarea unor filtre spațiale, urmate de o operație neliniară, cum ar fi calcularea unei energii sau aplicarea unui prag. Filtrele conduc la detectarea unor trăsături (muchii, linii, unghiuri).

Laws [6] are una din primele încercări de abordare a problemei în acest mod, și folosește metode ad-hoc la proiectarea filtrelor.

Încercările mai recente se bazează pe filtrele Gabor.

Filtrul Gabor bidimensional are expresia

$$g(x, y) = \frac{1}{2\pi\sigma_x\sigma_y} \exp\left(-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2}\right) + 2\pi i(u_0x + v_0y)\right) \quad (3.4)$$

Folosirea filtrului Gabor este, se pare, sprijinită pe considerente biologice, legate de percepția texturilor de către oameni.



Ca fapt divers, filtrul Gabor este optim din punctul de vedere al minimizării incertitudinii corelate spațiu-frecvență.

Ca și în cazul k-goanelor, folosirea rezultatului filtrării se poate face în mai multe moduri:

- momentele unui singur filtru, adică media și dispersia,
- histograma completă a rezultatului aplicării filtrului,
- coeficientul de corelație între două filtre sau între rezultatele aplicării unui filtru cu doi parametri,
- matricea de corelație (histograma corelată) a două filtre.

Așa cum am argumentat mai înainte, problema recunoașterii texturilor este strâns legată de problema generării automate a unei texturi. În cazul folosirii filtrelor spațiale, se caută o imagine pe care un set de filtre să dea același răspuns ca și textura dată ca model.

De principiu, problema se poate aborda plecând de la o imagine oarecare și actualizând pe rând câte un pixel, încercând minimizarea diferenței între valoarea dorită și valoarea obținută de la fiecare filtru. Din păcate, dacă suportul filtrului este destul de mare, metoda de mai sus e dificil de aplicat, deoarece variația ieșirii unui filtru în cazul variației intensității unui singur pixel este foarte mică.

O soluție prezentată în [10] se bazează pe inversarea filtrelor, într-un algoritm randomizat: se alege direcția de îmbunătățire a ieșirii unui filtru, și se găsește imaginea care optimizează acel criteriu, astfel se modifică mulți pixeli într-un singur pas.

### 3.5 Modele bazate pe transformate wavelets

Multe din abordările actuale în recunoașterea texturilor sunt bazate pe descompunerea wavelets. Descompunerea wavelets are avantajul de-a furniza concomitent informații multirezoluție despre o textură.

În general, dintr-un pas de descompunere wavelets a unei funcții (imagini) bidimensionale rezultă trei seturi de coeficienți wavelets și o imagine de rezoluție înjumătățită pe ambele direcții (numărul total de pixeli fiind deci un sfert).

Descompunerea wavelets cu filtre separabile se scrie:

$$\begin{aligned}
 I_{n+1}(x, y) &= (L_x \star (L_y \star I_n))(2x, 2y) \\
 W_{n+1,1}(x, y) &= (L_x \star (H_y \star I_n))(2x, 2y) \\
 W_{n+1,2}(x, y) &= (H_x \star (L_y \star I_n))(2x, 2y) \\
 W_{n+1,3}(x, y) &= (H_x \star (H_y \star I_n))(2x, 2y)
 \end{aligned} \tag{3.5}$$

unde  $I_n$  reprezintă imaginea la nivelul de rezoluție  $n$ , imaginea originală fiind  $I_0$ ;  $L$  și  $H$  reprezintă filtrele trece-jos și respectiv trece-sus, iar  $W$  reprezintă subimaginele wavelets. De notat că după filtrare se aplică subeșantionarea atât pentru imaginea de rezoluție redusă cât și pentru subimaginele wavelets, astfel că, dacă imaginea inițială este  $x_0$  pe  $y_0$  pixeli, în (3.5) avem  $x = 0, 2^{-n}x_0 - 1$  și  $y = 0, 2^{-n}y_0 - 1$ .

Deoarece în analiza texturilor se caută elemente repetitive, descompunerea se oprește la un anumit nivel de rezoluție  $d$ , imaginea de rezoluție redusă rămasă,  $L_d$ , nemaifiind folosită. Procesarea ulterioară extrage informații din subimaginele wavelets  $W_{n,i}$ , cu  $n = \overline{1, d}$  și  $i = \overline{1, 3}$ .

### 3.5.1 Descompuneri wavelets utilizate

Una dintre cele mai populare funcții de bază folosite în descompunerea wavelets este construită pe baza filtrului Gabor și are expresia:

$$g(x, y) = \frac{1}{2\pi\sigma_x\sigma_y} \exp\left(-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2}\right) + 2\pi i W x\right) \quad (3.6)$$

O altă descompunere wavelets utilizată în recunoașterea texturilor este *steerable pyramids* [16]. Descompunerea lucrează în coordonate polare.

### 3.5.2 Semnăturile energetice

Din fiecare subimagine wavelets  $W_{n,i}$  se extrage *energia subimaginii*

$$E_{n,i} = \frac{2^{2n}}{x_0 y_0} \sum_{x,y} (W_{n,i}(x, y))^2 \quad (3.7)$$

Energia coincide cu dispersia pătratică a valorilor pixelilor imaginii, întrucât media valorilor este nulă datorită faptului că rezultă din aplicarea unor filtre trece-sus asupra imaginii:

$$\sum_{x,y} W_{n,i}(x, y) = 0, \forall n, i$$

Altă posibilitate este să se folosească în locul energiei subimaginii mărimea

$$MD_{n,i} = \frac{2^{2n}}{x_0 y_0} \sum_{x,y} |W_{n,i}(x, y)| \quad (3.8)$$

### 3.5.3 Semnăturile histogramelor

Fie  $h_{n,i}(u)$  histograma imaginii wavelets  $W_{n,i}(x, y)$ .

S-a constatat experimental [15] că, în cazul texturilor ce apar în mod natural, histograma  $h_{n,i}(u)$ , notată în continuare  $h(u)$  pentru simplitate, poate fi aproximată destul de bine de o exponențială de forma

$$h(u) \approx K e^{-(|u|/\alpha)^\beta} \quad (3.9)$$

În relația de mai sus,  $\beta$  arată cât de “ascuțit” est vârful histogramei,  $\alpha$  arată cât de lat este, iar  $K$  este un coeficient de normalizare, care asigură că  $\int h(u)du = 0$  :

$$K = \frac{\beta}{2\alpha\Gamma(1/\beta)} \quad (3.10)$$

unde  $\Gamma$  este funcția gamma a lui Riemann

$$\Gamma(x) = \int_0^\infty e^{-t} t^{x-1} dt$$

Coeficienții  $\alpha$  și  $\beta$  ai celei mai bune aproximări sunt determinați de momentele de ordinul 1 și 2 ale histogramei

$$\begin{aligned} m_1 &= \int |u| h(u) du \\ m_2 &= \int |u|^2 h(u) du \end{aligned} \quad (3.11)$$

după relațiile

$$\alpha = m_1 \frac{\Gamma(1/\beta)}{\Gamma(2/\beta)} \quad (3.12)$$

$$\beta = F^{-1} \left( \frac{m_1^2}{m_2} \right) \text{ unde } F(x) = \frac{\Gamma^2(2/x)}{\Gamma(3/x)\Gamma(1/x)} \quad (3.13)$$

Semnăturile histogramelor sunt deci familiile  $(\alpha_{n,i})$  și  $(\beta_{n,i})$  asociate histogramelor imaginilor wavelets.

Aproximația (3.9) de mai sus are, între altele, aplicații în compresia imaginilor.

### 3.5.4 Matrici de corelație

Matricile de corelație oferă informații complementare față de histograme sau momentele histogramelor.

Se investighează mai multe matrici de corelație

- matrici de corelație în cadrul unei singure imagini wavelets [14]

$$C_{n,i}^{(\Delta x, \Delta y)}(a, b) = \sum_{x,y} \delta_{W_{n,i}(x,y),a} \delta_{W_{n,i}(x+\Delta x, y+\Delta y),b} \quad (3.14)$$

calculate doar pentru valori mici ale parametrilor  $\Delta_x$  și  $\Delta_y$ .

- matrici de corelație între imaginile wavelets de pe nivele de rezoluție consecutive [12], [13]

# Capitolul 4

## Concluzii și cercetări viitoare

În lucrare am trecut în revistă principalele direcții de lucru în detecția conturilor și în recunoașterea texturilor. Se poate spune că, deși primele cercetări în domeniu datează de cel puțin 20 de ani, aceste direcții de cercetare sunt încă active.

Personal am fost interesat de probleme legate de detecția conturilor și de segmentarea imaginilor, atât în cazul clasic, bazat pe variații de luminositate, cât și în cazul imaginilor cu texturi. În privința detecției conturilor în cazul clasic, preocupările personale s-au materializat într-un algoritm de detecție a conturilor.

Cercetări viitoare intenționate:

- studierea posibilității de-a folosi algoritmul de detecție a conturilor sus-menționat la detectarea conturilor parțial obstrucționate
- studierea aplicativității în imagini cu texturi
- combinarea algoritmului de detecție a conturilor cu algoritmi de interpolare a formelor, în vederea îmbunătățirii performanțelor

# Bibliografie

- [1] LARRY S. DAVID ET. AL *Foundations of Image Understanding* Kluwer Academic Publishers, 2001
- [2] M. KASS, A. WIKTIN, D. TERZOPOULOS *Snakes: Active contour models* International Journal of Computer Vision, 1, 1988, p. 321–331
- [3] CHENYANG XU, DZUNG PHAN, JERRY L. PRINCE *Image Segmentation Using Deformable Models*
- [4] MICHAEL D. HEATH *A Robust Visual Method for Assessing the Relative Performance of Edge Detection Algorithms* Master thesis, Univ. of South Florida, 1996
- [5] J. CANNY *A computational approach to Edge Detection* Pattern Analysis And Machine Intelligence, 8(6) p. 679–698, November 1986
- [6] K. LAWS *Textured image segmentation* PhD. Dissertation, University of Southern California, January, 1980
- [7] J. MALIK, S. BELONGIE, TH. LEUNG, J. SHI *Contour and Texture Analysis for Image Segmentation* Kluwer Academic Publishers, 2000
- [8] J. MALIK, S. BELONGIE, TH. LEUNG, J. SHI *Textons, Contours and Regions: Cue Integration in Image Segmentation* IEEE International Conference on Computer Vision, September 1999
- [9] MICHAEL D. HEATH *A robust visual method for assesing the relative performance of edge detection algorithms* PhD thesis, Univ. of South Florida, December 1996
- [10] SONG CHUN ZHU, XIU WEN LIU, YING NIAN WU, *Exploring Texture Ensembles by Efficient Markov Chain Monte Carlo — Toward a “Trichromacy” Theory of Texture*, IEEE Transaction on Pattern Recognition and Machine Intelligence, Vol. 22, No. 6, June 2000

- [11] RADU LUPȘA, *A Method for Contour Continuity Reconstruction* Proc. of the Symposium “Colocviul Academic Clujean de Informatică”, p. 79–82, September 2003
- [12] EERO P. SIMONCELI, JAVIER PORTILLA *Texture Characterization via Joint Statistics of Wavelet Coefficient Magnitudes* Proc. of 5th International Conference on Image Processing, Vol 1, 4–7 October 1998
- [13] NICU SEBE, MICHAEL S. LEW *Wavelet Based Texture Classification*
- [14] G. VAN DE WOUWER, P. SCHEUNDERS, D. VAN DYCK *Statistical texture characterization from discrete wavelet representations*
- [15] STEFANE G. MALLAT *A Theory for Multiresolution Signal Decomposition: The Wavelet Representation* IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 11, No. 7, July 1989
- [16] E. P. SIMONCELLI, W. T. FREEMAN *The Steerable Pyramid: A flexible architecture for multi-scale derivative computation* Second International Conf. on Image Processing, vol. 3, p. 444–447, October 1995
- [17] MINH N. DO, MARTIN VETTERLI *Rotation Invariant Texture Characterization and Retrieval using Steerable Wavelet-domain Hidden Markov Models* submitted to IEEE Transaction on Multimedia, April 2001