Universitatea Tehnică "Gh. Asachi" Iași
Facultatea de Automatică și Calculatoare
Specializarea Tehnologia Informației

Lucrare de licență

Identificarea obiectelor bazată pe puncte de interes, în secvențe video

Absolvent

Carata Lucian

Coordonator științific
Prof. Dr. Ing. Vasile Manta

Iași, 2009

Programs must be written for people to read, and only incidentally for machines to execute.

Programele trebuie scrise mai întâi pentru a fi citite de oameni și doar apoi pentru a fi executate de mașini.

— Abelson and Sussman

Cuprins

Capitolul 1

Introducere

1.1 Recunoașterea automată a obiectelor

Identificarea şi recunoaşterea automată a unor obiecte, în imagini statice sau secvenţe video, este îndelung studiată în grafica pe calculator, prezentând interes din perspectiva dificultăţilor întâmpinate în rezolvarea problemei de sistemele de calcul în comparaţie cu sistemele biologice, dar şi datorită aplicabilităţii în domenii din cele mai diverse.

Astfel, primele aplicaţii s-au conturat în mediul industrial, pentru inspectarea automată a produselor de pe liniile de fabricaţie (de exemplu, identificarea defectelor unor plăci integrate - lipituri incorecte, plasari incorecte de componente etc). Totuşi, mediul de recunoaştere în aceste cazuri este unul controlat, existând limite stricte între care recunoaşterea se realizează cu succes. Mai mult, obiectele pentru care se realizează identificarea au caracteristici bine cunoscute. Pornind de aici, s-au dezvoltat metode care încearcă să elimine cât mai multe dintre restricţii, şi să permită recunoaşterea în cazul general. Aceste abordări lărgesc gama de aplicaţii şi în zona utilizatorilor obişnuiţi, pentru îmbunătăţirea următoarelor generaţii de motoare de căutare, programe de chat sau de supraveghere a locuinţelor. Desigur, domenii precum medicina (recunoaşterea sau numărarea celulelor de un anumit tip), robotica (dezvoltarea unor roboţi care să interacţioneze cu mediul înconjurător folosind "vederea artificială") utilizează şi ele recunoaşterea obiectelor ca subproblemă. Aplicaţii similare există în domeniul militar (recunoaşterea unor dispozitive suspecte în aeroporturi, identificarea persoanelor pe baza înfăţişării).

Problemele cele mai mari în identificarea şi recunoaşterea obiectelor apar datorită variaţiilor din mediul în care obiectul este plasat (culoare, lumină, umbre, reflexii, ocluzionarea obiectului ţintă de către alte obiecte). De asemenea, apar dificultăţi şi datorită diferenţelor de poziţionare şi perspectivă între reprezenta-

rea iniţială a obiectului care se doreşte a fi identificat (de cele mai multe ori, o fotografie a respectivului obiect) şi situaţia reală în care se încearcă identificarea acestuia (când el poate fi "privit" la o altă scală, rotit sau dintr-un punct de vedere diferit).

Au fost găsite mai multe abordări pentru rezolvarea acestor probleme, majoritatea detectând în fiecare imagine anumite zone caracteristice, invariante la modificări ale parametrilor de mediu/perspectivă. Se realizează apoi o potrivire între ele şi o bază de date în care au fost anterior reţinute caracteristicile obiectelor căutate. Diferenţele între metode se referă la modalitatea de detecţie a zonelor, la forma lor (puncte, arii din imagine) şi la informaţiile reţinute pentru fiecare zonă în parte astfel încât ea să poată fi regăsită într-o nouă imagine şi atribuită ca aparţinând obiectului căutat.

La momentul actual, tehnicile de recunoaştere nededicate permit o detecţie cu un procentaj de reuşită şi repetabilitate a rezultatelor suficient de mare (tipic peste 80%) pentru a fi considerate aplicabile cu succes în aplicaţii de orice tip. Totuşi, se pune problema selectării unor metode cât mai eficiente, care să poată fi aplicate "in timp real".

1.2 Formularea și abordarea temei

Recunoaşterea unor obiecte (furnizate ca imagini, drept date de intrare), într-o secvenţă video live sau filmată anterior (video salvat pe hdd), presupune detectarea existenţei obiectelor şi identificarea poziţiei acestora în fiecare frame, urmată de "adnotarea" frame-ului în zona obiectului recunoscut. Atât iniţial, pentru imaginile ce definesc obiectele, cât şi pentru secvenţa video, se aplică aceeaşi algoritmi de determinare a zonelor caracteristice. Apoi, se realizează o potrivire între rezultatele obţinute pentru frame-ul curent şi modelul determinat pentru fiecare dintre obiecte. În măsura în care există corespondenţe (în frame-ul curent există zone similare cu cele ale obiectului), se stabileşte prezenţa obiectului în frame, precum şi poziţia acestuia. Se doreşte o variaţie cât mai mică a rezultatelor la schimbări de scală, rotaţii, modificări ale perspectivei 3D şi a luminozităţii ambientale, urmărind în acelaşi timp o repetabilitate crescută a experimentelor. De asemenea, este de preferat să nu se impună restricţii legate de modul în care sunt capturate imaginile sau de calibrarea anterioară a dispozitivelor (camere video, aparate foto digitale).

Dacă drept zone caracteristice se folosesc puncte de interes, există 2 paşi generali în analiza fiecărei imagini prelucrate:

1. *Localizarea*: determinarea automată a poziţiei punctelor de interes (în două imagini ale aceluiaşi obiect, se doreşte ca punctele de interes să fie localizate în aceeaşi zonă relativ la obiect - Figura 1.1)

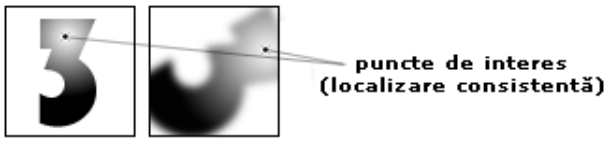

Figura 1.1: *Localizarea punctelor de interes:* rulând în mod independent algoritmul pe două imagini ale aceluiaşi obiect în situaţii diferite, se doreşte ca punctele de interes să fie identificate în aceleaşi poziţii relativ la obiect

2. *Descrierea*: fiecărui punct de interes determinat anterior îi sunt asociate o mulţime de date rezultate din analiza imaginii, astfel încât el să poată fi identificat cu un grad ridicat de individualitate în comparaţie cu restul punctelor de interes (Figura 1.2). În imagini diferite ale aceluiaşi obiect, vectorul obţinut trebuie să fie invariant la modificări ale mediului (luminozitate) sau la transformări afine asupra obiectului (translaţii, rotaţii, scalări), pentru a asigura o recunoaştere adecvată (căutarea se realizează pe baza descriptorilor).

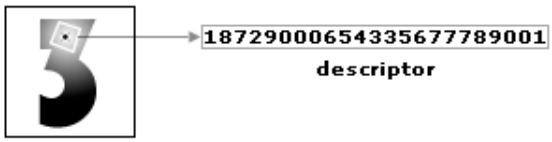

Figura 1.2: *Descrierea punctelor de interes:* Asocierea de informaţii pentru identificare, considerând vecinătatea punctului de interes.

Pentru fiecare dintre aceşti paşi, există diferiţi algoritmi, unii asigurând o ”acoperire” mai bună a obiectelor cu puncte de interes, alţii concentrându-se pe stabilitatea trăsăturilor determinate sau pe eficienţa computaţională. Desigur, trebuie realizat un compromis astfel încât să se ajungă la o soluţie acceptabilă pentru cât mai multe aplicaţii. De menţionat că algoritmii de localizare şi cei de descriere pot fi aleşi în mod independent, dar o analiză a performanţelor nu poate fi realizată decât la nivelul efectului aplicării celor 2 paşi, în mod secvenţial.

Metodele ce au la bază trasături identificate prin puncte de interes pot fi folosite şi în alte aplicaţii, nu doar cea a recunoaşterii. Astfel de algoritmi pot fi aplicaţi, de exemplu, ca prim pas în reconstruirea unor modele 3D ale obiectelor din imagini sau video, pentru calibrarea aparatelor foto digitale sau pentru crearea de panorame din secvenţe de imagini. Prin urmare, sunt de utilitate mare implementările cât mai generale, flexibile, care să poată fi utilizate într-o gamă largă de aplicaţii sau teste comparative. Lucrarea de faţă se referă la detaliile unei astfel de implementări.

Capitolul 2 face o scurtă trecere în revistă a cercetărilor realizate în domeniu, punând accentul pe descrierea noţiunilor teoretice şi a algoritmilor folosiţi.

Capitolul 3 prezintă detaliile legate de proiectarea aplicaţiei propuse, descriind structura detaliată a modulelor unei platforme software pentru prelucrarea fluxurilor de imagini şi a submodulelor ce implementează algoritmii pentru detecţia de obiecte.

Capitolul 2

Noțiuni teoretice

2.1 Abordări ale temei în literatura de specialitate

În funcţie de restricţiile aplicaţiei practice în care este utilizată, recunoaşterea obiectelor poate lua mai multe forme, de la simpla împărţire a imaginii în zone ce pot reprezenta obiecte (segmentare pe bază de culoare), la o recunoaştere completă, ce implică determinarea locaţiei (x, y) a unui obiect, reconstituirea poziţionării sale în spaţiu (sau 2D în planul imaginii) şi recunoaşterea denumirii obiectului respectiv pe baza unor cunoştiinţe anterioare ale sistemului.

Oricare ar fi gradul de complexitate, la modul general se pune problema ca pornind de la o matrice de pixeli (imaginea), să fie determinată o submulţime a acestora care reprezintă un obiect. Fără a scădea din generalitate, considerăm că obiectul este dat de o regiune contiguă de pixeli din imaginea originală.

O abordare directă a problemei, presupunând că deţinem o imagine a obiectului, ar fi căutarea tuturor pixelilor săi într-o altă imagine dată. Îmbunătăţiri ale acestei metode, caracterizată de potrivirea unor "tipare" reprezentând obiectul în scene care îl conţin, au reprezentat începutul cercetării în domeniu (Figura 2.1a). Soluţia (template matching), în forma ei iniţială, este ineficientă computaţional şi sensibilă atât la modificări ale mediului în care dorim să realizăm recunoaşterea (luminozitate, reflexii) cât şi la ocluzionări parţiale ale obiectelor. Pentru obţinerea unei oarecare invarianţe, a fost propusă corelarea nivelurilor de gri din diverse zone ale imaginii reprezentând obiectul, cu zone din imagini care se presupune că îl conţin. [Ballard and Brown, 1982, Goshtasby et al., 1984] Aceste studii sunt făcute în contextul sistemelor de stereo-vizualizare, unde scena este fotografiată simultan din perspective diferite şi se doreşte determinarea unor corespondenţe între imagini, evitând o calibrare anterioară sau cunoaşterea geometriei epipolare a sistemului. Mai recent, există variante care propun modificări ale metodei pentru a o putea rula în timp real [Cole et al., 2004].

template matching

Pentru a depăşi o parte din problemele metodei anterioare, se pleacă de la observaţia că pentru recunoaşterea unui obiect nu este nevoie de toţi pixelii săi, ci doar de o parte din aceştia, ce definesc forma specifică a obiectului sau caracteristici importante ale acestuia. Se realizează o sintetizare a informaţiei din imaginea originală, făcându-se primul pas înspre reprezentarea respectivului obiect într-un mod abstract. Abordarea recunoaşterii obiectelor prin potrivirea unor astfel de trăsături abstracte (feature matching) este cea de-a doua direcţie de cercetare în domeniu. (Figura 2.1b)

feature matching

Aplicarea algoritmilor de acest tip presupune macarea trăsăturilor din imagine ca puncte de interes, având ca informaţie minimală locaţia, (x_t, y_t). Există desigur şi posibilitatea stocării unor date suplimentare precum orientarea sau scala caracteristicii determinate.

În mod tradiţional, trăsăturile alese pentru identificare şi potrivire sunt muchii, colţuri sau contururi [Cheng and Huang, 1984, Ullman, 1979]. În momentul de faţă, sunt propuşi algoritmi care realizează şi identificarea unor alte structuri, precum petele luminoase sau întunecate (eng. blob) [Lowe, 2003, Bay et al., 2006].

(a) (b)

Figura 2.1: (a): Potrivire bazată pe tipare (template matching), (b): Potrivire bazată pe trăsături (feature matching)

Avantajul acestei metode este că necesită mai puţină putere de calcul (operând asupra unui număr relativ restrâns de pixeli) şi poate fi prin urmare aplicată cu uşurinţă în timp real. În plus, datorită faptului că se lucrează cu o reprezentare intermediară a obiectului (trăsături), metodele pot fi proiectate pentru a obţine un grad ridicat de invarianţă la anumiţi parametrii de mediu sau la transformări afine aplicate obiectului. Abordarea eşuează însă dacă nu se reuşeşte o determinare repetabilă şi consistentă a trăsăturilor unui obiect în imagini diferite. În acest caz, potrivirea nu are loc şi obiectul nu este detectat.

Datorită flexibilităţii crescute şi a rezultatelor foarte bune obţinute în practică de către abordarea potrivirii bazate pe trăsături, lucrarea de faţă utilizează

această metodă pentru recunoaşterea obiectelor.

2.2 Identificarea trăsăturilor

În identificarea trăsăturilor, se disting 2 metode mai importante, utilizate, cu unele adaptări, în cele mai multe dintre aplicaţiile practice curente: Detectorul Harris, şi SIFT (Scale Invariant Feature Transform). Aplicaţia propusă în lucrare utilizează o variantă îmbunătăţită a algoritmului SIFT, adaptată pentru procesarea fluxurilor de imagini, în timp real. O parte a ideilor propuse iniţial de Harris si Stephens pentru detectorul Harris sunt reluate în algoritmul SIFT, prin urmare considerăm utilă prezentarea ambelor metode.

2.2.1 Detectorul Harris

Prima abordare, propusă de Harris şi Stephens, identifică în imagine colţurile şi muchiile [Harris and Stephens, 1988]. Cei doi pornesc de la o observaţie anterioară a lui Moravec, care defineşte un colţ ca fiind un pixel care nu se aseamănă cu pixelii din vecinătatea sa. Astfel, pe o suprafaţă uniformă, un pixel va avea valori apropiate de cele ale vecinilor săi; pe o muchie, în vecinătatea pixelului se vor identifica modificări mari relativ la valorile vecinilor perpendiculari pe direcţia muchiei, dar modificări mici în direcţia muchiei. Însă dacă pixelul aparţine unei trăsături cu variaţii în toate direcţiile (un colţ), atunci nici una dintre vecinătăţi nu va fi similară pixelului. În [Harris and Stephens, 1988], formalizând matematic aceste observaţii, se defineşte noţiunea de autocorelaţie. Funcţia de autocorelaţie măsoară modificările locale ale semnalului 2D (imaginea), folosind ferestre deplasate pe distanţe mici în vecinătatea punctului considerat. Fiind dată o deplasare $(\Delta x, \Delta y)$ şi un punct (x, y), funcţia de autocorelaţie este

$$E(\Delta x, \Delta y) = \sum_{x,y} w(x, y)[I(x + \Delta x, y + \Delta y) - I(x, y)]^2 \tag{2.1}$$

unde $w(x, y)$ reprezintă funcţia fereastră (şi poate fi aleasă ca fiind o funcţie nucelu rectangulară sau, pentru a reduce influenţa zgomotului, un nucleu Gaussian) iar $I(\cdot, \cdot)$ este funcţia imagine.

Imaginea din fereastra deplasată este aproximată prin dezvoltarea în serie Taylor, trunchiată la primii termeni,

$$I(x + \Delta x, y + \Delta y) = I(x, y) + \begin{bmatrix} I_x & I_y \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \tag{2.2}$$

I_x şi I_y fiind derivatele parţiale pe direcţia x, respectiv y.

Înlocuind 2.2 în 2.1 şi considerând Δx şi Δy suficient de mici, obţinem o ecuaţie de forma:

$$E(\Delta x, \Delta y) \cong \begin{bmatrix} \Delta x & \Delta y \end{bmatrix} M \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}$$

M fiind o matrice 2×2 calculată din derivatele locale parţiale ale imaginii,

$$M = \sum_{x,y} w(x,y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

Matricea M descrie structura locală a imaginii în vecinătatea pixelului considerat. Fie λ_1, λ_2 valorile proprii ale acestei matrici. Există 3 cazuri care trebuie considerate:

1. Dacă atât λ_1 cât şi λ_2 au valori mici, astfel încât funcţia de autocorelaţie este plată (schimbări mici ale lui $E(\Delta x, \Delta y)$ în orice direcţie), zona din fereastra considerată este aproximativ uniformă.

2. Dacă o valoare proprie este mare iar cealaltă este mică, astfel încât funcţia de autocorelaţie are forma unei trepte, atunci deplasările ferestrei într-o direcţie (de-a lungul treptei) produc modificări mici ale lui E, iar deplasările pe o direcţie ortogonală primeia produc modificări mari. Acest lucru indică o muchie.

3. Dacă valorile proprii sunt ambele mari, deplasările în orice direcţie vor produce modificări mari ale lui E, indicând un colţ.

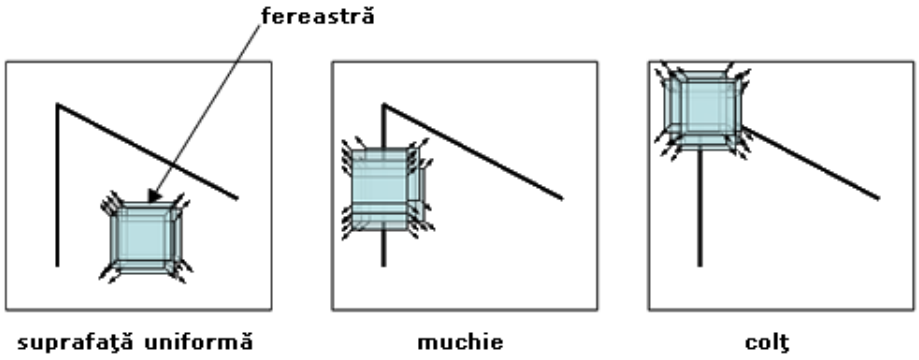

Figura 2.2: Detectorul Harris (muchii şi colţuri)

Intuitiv, modul de operare al detectorului Harris este prezentat în Figura 2.2. Performanţele sale au fost analizate în detaliu [C.Schmid et al., 2000]. Concluzia studiului este că detectorul Harris este unul robust, putând fi aplicat cu succes

chiar şi pe imagini zgomotoase şi fiind invariant la rotaţii sau schimbări ale luminozităţii ambientale. Totuşi, repetabilitatea rezultatelor sale scade drastic la schimbări ale perspectivei. O altă problemă a detectorului este că nu este invariant la modificările de scală ale obiectelor considerate. Acest poate fi observat cu uşurinţă în Figura 2.3.

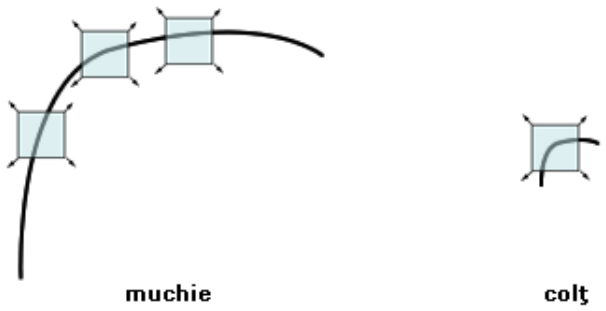

Figura 2.3: Modificarea scalei imaginii poate duce la rezultate diferite în cazul detectorului Harris

Au fost propuse şi variante care să fie invariante la scalări (Harris-Laplacian), acestea fiind similare ca abordare cu cel de-al doilea detector important, SIFT.

2.2.2 Detectorul SIFT

SIFT (Scale Invariant Feature Transform) este un algoritm propus de Lowe [Lowe, 2003], care include ca pas intermediar detecţia unor puncte de interes asimilate unor trăsături de tip ”zonă luminoasă” sau ”zonă întunecată”. Prin construcţia algoritmului, aceste zone sunt determinate pentru a fi invariante la scalări, rotaţii şi parţial invariante la modificări ale luminozităţii şi la transformări afine. Metoda aplică o filtrare în cascadă, pentru a asigura calitatea punctelor de interes determinate, dar şi pentru a aplica operaţiile intensive computaţional doar acelor zone care trec unele teste iniţiale. Pe lângă determinarea locaţiei punctelor de interes, algoritmul SIFT propune şi metode de descriere a acestora în mod individual, astfel încât să poată fi identificate cu probabilitate mare în imagini noi. Practic, fiecărui punct de interes îi este asociat un descriptor (vector caracteristic), calculat pe baza informaţiilor imaginii în vecinătatea punctului de interes.

Aceste caracteristici fac SIFT ideal pentru aplicarea în zona recunoaşterii obiectelor. Pentru aceasta, mai intâi se extrag trăsăturile SIFT pentru un set de imagini de referinţă ce reprezintă obiectele, stocând descriptorii rezultaţi într-o bază de date. Unei imagini noi, în care se doreşte identificarea unuia dintre obiectele existente în baza de date, i se aplică acelaşi algoritm, iar descriptorii

punctelor de interes rezultate sunt comparaţi individual cu descriptorii din baza de date. Potrivirile între descriptori se fac pe baza distanţei Euclidiene între vectori (nu se caută doar potriviri exacte). Totuşi, într-o imagine aglomerată, multe trăsături din fundal nu vor avea corespondenţi în baza de date, dând potriviri false, pe lângă cele corecte. Potrivirile corecte pot fi însă filtrate prin identificarea unor submulţimi de puncte de interes care sunt consistente cu aceeaşi localizare, scală şi orientare a obiectului în noua imagine. Determinarea acestor clustere poate fi realizată eficient folosind transformata Hough.

Localizarea punctelor de interes

Primul pas în determinarea punctelor de interes SIFT îl reprezintă detectarea locaţiilor din imagine care sunt invariante la scalări, prin căutarea trăsăturilor stabile, folosind o funcţie de scală cunoscută sub denumirea de spaţiu al scalărilor (eng. scale space). Spaţiul scalărilor pentru o imagine este definit de funcţia $L(x, y, \sigma)$, obţinută prin convoluţia unui nucleu Gaussian $G(x, y, \sigma)$ cu imaginea, $I(x, y)$. Pentru a obţine scalări diferite, se variază σ:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y),$$

unde $*$ reprezintă operaţia de convoluţie, iar nucleul Gaussian G este dat de formula:

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Pentru a detecta punctele de interes stabile în spaţiul scalărilor, Lowe propune determinarea extremelor locale ale funcţiei "diferenţă de nuclee Gauss cu scalări diferite", în convoluţie cu imaginea, $D(x, y, \sigma)$. Aceasta poate fi calculată din diferenţa a două scalări separate de un factor k:

$$\begin{aligned} D(x, y, \sigma) &= (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \\ &= L(x, y, k\sigma) - L(x, y, \sigma) \end{aligned}$$

Există mai multe motive pentru care a fost aleasă această funcţie în mod particular. În primul rând, imaginile pentru care se aplică filtrul Gaussian (convoluţie), trebuie oricum calculate în procesul de creare al spaţiului scalărilor, D calculându-se în mod eficient prin scăderea imaginilor din două scale adiacente. În al doilea rând, diferenţa nucleelor Gauss (Difference of Gaussian, DOG) aproximează foarte bine Laplacianul Gaussian-ului, $\sigma^2\nabla^2 G$. S-a demonstrat că extremele acestei funcţii reprezintă trăsături foarte stabile ale imaginii, în comparaţie cu trăsăturile determinate cu alte funcţii precum gradientul, Hessian-ul sau detectorul Harris.

Pentru a detecta extremele locale ale lui D, se realizează o eşantionare a funcţiei atât spaţial (x, y), cât şi pentru parametrul de scală (σ). Frecvanţa aleasă pentru eşantionare reprezintă un compromis între precizia localizării extremelor

şi puterea de calcul necesară pentru determinarea lor. Astfel, o eşantionare cu frecvenţă mare duce la costuri mari din punct de vedere computaţional, iar o frecvenţă mică duce la o precizie scăzută a algoritmului.

Fiecare punct eşantionat este comparat cu cei 8 vecini ai săi din imaginea curentă, şi cei 9 vecini din scalările adiacente celei curente (Figura 2.4). Punctul este selectat doar dacă este mai mare sau mai mic comparativ cu toţi vecinii săi. Această abordare se dovedeşte eficientă pentru că majoritatea punctelor sunt eliminate după doar câteva comparaţii.

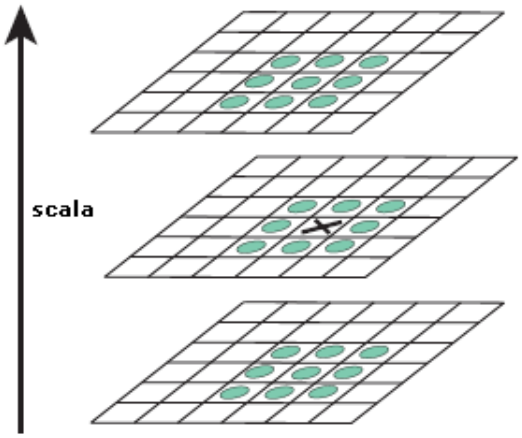

Figura 2.4: *SIFT: Detectarea minimelor şi maximelor locale;* punctul central este comparat cu toţi vecinii marcaţi ([Lowe, 2003])

O precizie crescută a localizării punctelor de interes se poate obţine folosind o metodă de aproximare a poziţionării maximului, prin interpolare. Astfel, se încearcă aproximarea punctelor eşantionate cu o funcţie cuadrică, 3D. Practic, se realizează o dezvoltare în serie Taylor până la termenii de grad 2, a funcţiei $D(x, y, \sigma)$, translată astfel încât punctul eşantionat să fie în origine:

$$D(\mathbf{x}) = D + \frac{\partial D}{\partial \mathbf{x}}^T \mathbf{x} + \frac{1}{2}\mathbf{x}^T \frac{\partial^2 D}{\partial \mathbf{x}^2}\mathbf{x} \tag{2.3}$$

unde D şi derivatele sale sunt evaluate în punctul de eşantionare iar $\mathbf{x} = (x, y, \sigma)^T$ este deplasarea faţă de acest punct. Localizarea precisă a extremului, $\hat{\mathbf{x}}$ este determinată prin derivarea ecuaţiei 2.3 în raport cu $\mathbf{x}$ şi egalarea cu zero, rezultând

$$\hat{\mathbf{x}} = -\frac{\partial^2 D}{\partial \mathbf{x}^2}^{-1} \frac{\partial D}{\partial \mathbf{x}} \tag{2.4}$$

Pentru a elimina punctele care sunt maxime locale dar se află într-o regiune cu un contrast slab (fiind deci instabile), se vor reţine doar acelea pentru care $D(\hat{\mathbf{x}})$ este mai mare decât o valoare prag (Lowe alege valoarea de prag 0.03 pentru experimentele sale).

Totuşi, pentru o stabilitate crescută, nu e suficientă îndepărtarea trăsăturilor cu un contrast slab. Funcţia "diferenţă de nuclee Gauss" va avea un răspuns puternic de-a lungul muchiilor, chiar dacă locaţia respectivă este determinată imprecis, sensibilă la zgomotele din imagine. Pentru eliminarea acestor răspunsuri, se foloseşte o abordare bazată pe o matrice Hessiană 2×2, calculată în poziţia şi pentru scala punctului de interes:

$$\mathbf{H} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix} \tag{2.5}$$

Derivatele se estimează prin diferenţele faţă de punctele eşantionate din vecinătate. Pentru eliminarea răspunsurilor de-a lungul muchiilor, se impune ca raportul valorilor proprii ale acestei matrici să fie sub o valoare prag (Lowe alege valoarea 10). Pentru că eliminarea se face în funcţie de raportul valorilor proprii, nu este necesară calcularea individuală a acestora. În loc, se folosesc determinantul şi urma matricii $\mathbf{H}$. Dacă notăm valorile proprii cu λ_1 şi λ_2, atunci:

$$\begin{aligned} Tr(\mathbf{H}) &= D_{xx} + D_{yy} = \lambda_1 + \lambda_2 \\ Det(\mathbf{H}) &= D_{xx}D_{yy} - (D_{xy})^2 = \lambda_1\lambda_2 \end{aligned}$$

Considerăm arbitrar $\lambda_1 > \lambda_2$ şi notăm raportul valorilor proprii cu r, astfel încât $\lambda_1 = r\lambda_2$. Atunci, avem:

$$\frac{Tr(\mathbf{H})^2}{Det(\mathbf{H})} = \frac{(\lambda_1 + \lambda_2)^2}{\lambda_1\lambda_2} = \frac{(r\lambda_2 + \lambda_2)^2}{r\lambda_2^2} = \frac{(r+1)^2}{r} \tag{2.6}$$

Prin urmare, pentru a impune r ca valoare prag, trebuie verificată doar condiţia:

$$\frac{Tr(\mathbf{H})^2}{Det(\mathbf{H})} < \frac{(r+1)^2}{r} \tag{2.7}$$

Descrierea punctelor de interes

După stabilirea precisă a locaţiei unei trăsături, se doreşte asocierea unui vector caracteristic (descriptor), astfel încât ea să poată fi identificată şi în alte imagini.

Primul pas constă în atribuirea unei orientări fiecărui punct de interes, astfel încât descriptorul să poată fi reprezentat relativ la orientarea sa locală. Pentru operaţiile care urmează, se alege imaginea filtrată cu nucleu Gaussian având scala cât mai apropiată de cea determinată prin interpolare pentru punctul de interes. Folosind această imagine, se calculează norma şi orientarea gradientului într-un număr de puncte din vecinătatea punctului de interes. Valorile obţinute sunt organizate într-o histogramă a orientărilor, cu 36 de intervale. Fiecare vector gradient este adăugat în intervalul corespunzător orientării sale şi ponderat cu valoarea normei. Vârfurile din histogramă corespund orientărilor dominante ale gradienţilor locali. Cel mai mare vârf este ales ca orientare a punctului de interes.

Dacă cel de-al doilea vârf al histogramei este comparabil ca mărime, atunci în aceeaşi poziţie din imagine se va crea un al doilea punct de interes, care să aibă orientarea acestui al doilea vârf.

Parametrii de poziţie, scală şi orientare determinaţi până acum stabilesc un sistem de coordonate 2D, local punctului de interes, faţă de care se realizează descrierea acestuia.

În vecinătatea determinată de sistemul local de coordonate al punctului de interes se realizează o eşantionare, iar în punctele alese se calculează norma şi orientarea gradientului, relativ la orientarea punctului de interes (Figura 2.5). Normele sunt ponderate de o funcţie Gaussiană, (cercul din figură) cu σ de 1.5 ori mai mare decât dimensiunea vecinătăţii considerate (în experimente 16×16 pixeli). Vecinătatea este împărţită apoi într-un număr de subregiuni care nu se suprapun (16 regiuni de 4×4 pixeli). Pentru fiecare subregiune, valorile gradienţilor sunt acumulate într-o histogramă, similară celei folosite anterior. Dacă o histogramă discretizează unghiurile de orientare în 8 valori posibile, descriptorul punctului de interes va conţine $4 \times 4 \times 8 = 128$ elemente, obţinute prin concatenarea valorilor din toate histogramele. (Figura 2.5).

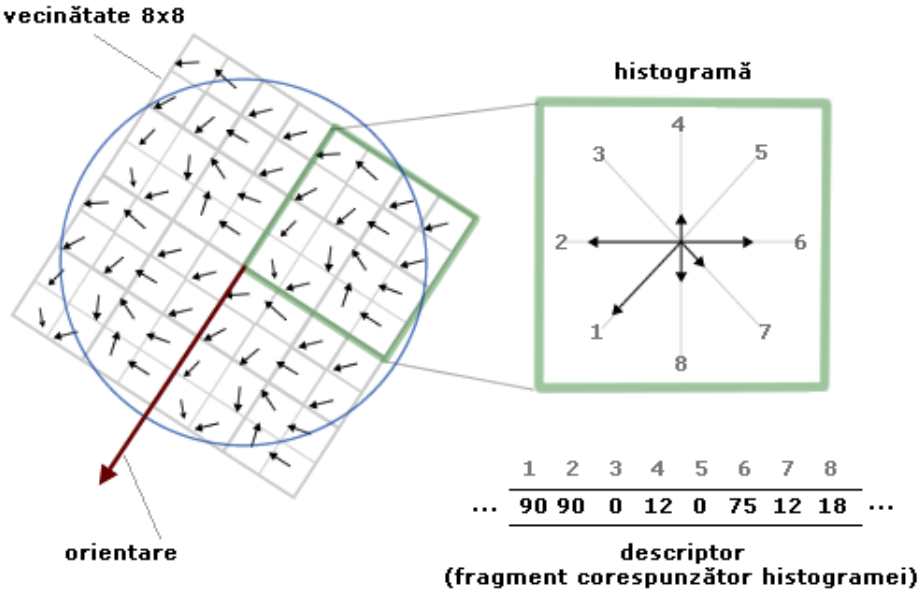

Figura 2.5: *SIFT: Procesul de determinare al descriptorului;* pentru claritatea reprezentării, a fost aleasă o vecinătate 8x8 a punctului de interes. Algoritmul foloseşte vecinătăţi 16x16.

2.3 Identificarea trăsăturilor în timp real

Dintre cele 2 metode prezentate, SIFT se remarcă datorită invarianţei la un număr mare de parametrii precum şi datorită stabilităţii punctelor de interes determinate. Totuşi, este evident că aplicarea algoritmului SIFT implică un număr mult mai mare de operaţii în comparaţie cu alţi detectori (Harris). Deoarece majoritatea aplicaţiilor îl vor utiliza doar ca pas intermediar, se pune problema unei post-procesări a punctelor de interes (de exemplu, pentru a identifica obiecte) şi se doreşte ca ansamblul algoritmilor de procesare să ruleze în timp real. În forma prezentată, SIFT poate prelucra în jur de 5 frame-uri (de dimensiune 650×315) pe secundă. Prin urmare, se justifică o căutare a unor îmbunătăţiri care să determine o scădere a timpului de prelucrare, fără a afecta calitatea rezultatelor finale.

SURF (Speeded-Up Robust Features) este una dintre soluţiile propuse în acest sens, fiind şi metoda utilizată de aplicaţia descrisă în această lucrare. Deoarece majoritatea paşilor urmaţi sunt identici cu cei ai algoritmului SIFT, vom prezenta în continuare doar elementele noi pe care le aduce în comparaţie cu acesta.

Capitolul 3

Chapter's title

... some text ...
Some reference

Bibliografie

[Ballard and Brown, 1982] Ballard, D. and Brown, C. (1982). *Computer Vision*. Prentice-Hall, Englewood Cliffs. [cited at p. 5]

[Bay et al., 2006] Bay, H., Tuytelaars, T., and Gool, L. V. (2006). Surf: Speeded-up robust features. In *ECCV*, pages 404–417. [cited at p. 6]

[Cheng and Huang, 1984] Cheng, J. and Huang, T. (1984). Image registration by matching relational structures. *Pattern Recognition*, 17(1):149–159. [cited at p. 6]

[Cole et al., 2004] Cole, L., Austin, D., and Cole, L. (2004). Visual object recognition using template matching. In *Proceedings of Australian Conference on Robotics and Automation*. [cited at p. 5]

[C.Schmid et al., 2000] C.Schmid, Mohr, R., and C.Bauckhage (2000). Evaluation of interest point detectors. *International Journal of Computer Vision*, 37(2):151–172. [cited at p. 8]

[Goshtasby et al., 1984] Goshtasby, A., Gage, S., and Bartholic, J. (1984). A two-stage cross correlation approach to template matching. *IEEE Transactions, Pattern Analysis & Machine Intelligence*, 6(3):374–378. [cited at p. 5]

[Harris and Stephens, 1988] Harris, C. and Stephens, M. (1988). A combined corner and edge detection. In *4th Alvey Vision Conference*, pages 147–151. [cited at p. 7]

[Lowe, 2003] Lowe, D. G. (2003). Distinctive image features from scale-invariant keypoints. [cited at p. 6, 9, 11, 24]

[Ullman, 1979] Ullman, S. (1979). *The Interpretation of Visual Motion*. MIT Press, Cambridge, MA. [cited at p. 6]

Anexe

Anexa A

Anexa 1

... some text ...

Lista de simboluri și prescurtări

Prescurtare	Descriere	Definiţie
SIFT	Scale Invariant Feature Transform	page 9
DOG	Difference of Gaussian	page 10
SURF	Speeded-Up Robust Features	page 14

Listă de figuri

Listă de tabele

Glosar