

Prelucrarea imaginilor în JAVA

Conf. dr. Dan Laurențiu Lacrămă
Universitatea "Tibiscus" Timișoara,
Ing. Ionel Snep
NMS Communications Paris

ABSTRACT. This paper is focused on Image processing algorithms implementation in Java. Practical solutions are proposed in order to preserve application modularity, thus new algorithms can be implemented using the same basic frame.

1 Introducere

Prelucrarea imaginilor are ca obiectiv general facilitarea interpretării informațiilor conținute în imagini. În funcție de aplicația concretă, interpretarea poate fi realizată de către om sau un sistem de calcul.

În sens larg, prelucrarea imaginilor include în problematica abordată și realizarea unor sisteme inteligente de vedere artificială, capabile să recunoască forme și să ofere o interpretare a scenei captate de senzorii de imagine.

Principalele aplicații ale prelucrării imaginilor sunt:

- Tehnicile de ameliorare capabile să evidențieze selectiv în imagine aspecte de interes în raport cu obiectivele analizei vizuale. Exemple de aplicații: sistemele pentru inspecție vizuală folosesc metode de ameliorare pentru a detecta în timp real defectele apărute în diferite faze ale procesului de fabricație; îmbunătățirea imaginilor radiografice prin procedee de transformare a histogramei, pseudocromatizare, filtrare, mărire, substracție de fond, etc.; prelucrarea imaginilor multispectrale achiziționate și transmise de sateliți pentru a se

extrage informația de interes, imperceptibilă în imaginile aflate în forma brută.

- Tehnici de restaurare: eliminarea sau reducerea daunelor provocate imaginilor de perturbații prin compensarea efectului acestora. Algoritmii de restaurare sunt utilizați în domenii precum: sporirea acurateții unor imaginilor prin eliminarea perturbațiilor biologice (provenite din organismul pacientului) sau ale celor din lanțul de transmisie; eliminarea efectelor deformante provocate de sistemele optice și de atmosferă asupra imaginilor de teledetecție; eliminarea efectelor deformante provocate de sistemele de captare, de atmosferă și de interacțiunea luminii cu diferite corpuri cerești în imaginile astronomice.
- Tehnicile de analiză capabile să furnizeze automat evaluări cantitative ale unor mărimi de interes în imaginile procesate. Sunt utile pentru: controlul grosimii rostului la sudura automată, al grosimii tablei în procesul de laminare, sau al diametrului mediu al granulelor în industria pulberilor; analiza secvențelor de imagini ale inimii, captate sincron cu ciclul de contracție; estimarea unor parametri de interes cum ar fi producțiile agricole pe anumite suprafețe, concentrația agenților poluanți etc în imagistica provenită de la sateliții de teledetecție.
- Recunoașterea formelor, clasificarea automată a unui obiect necunoscut într-o mulțime finită de clase. Aplicații concrete sunt: recunoașterea subansamblelor de către roboții de montaj; recunoașterea caracterelor scrise (de tipar, de mână), interpretarea automată a formularelor și recunoașterea semnăturilor pentru operații bancare; recunoașterea monedelor pentru mașini automate de vânzare; recunoașterea tipului de avion în aplicațiile legate de aviația militară și civilă; ghidarea vehiculelor cu deplasare autonomă (avioane de spionaj fără pilot, rachete strategice); clasificarea automată a terenurilor în imagini de teledetecție; clasificarea amprentelor digitale și a fețelor în criminalistică.
- Tehnicile de compresie având ca obiectiv reducerea numărului de biți necesari redării imaginilor cu o fidelitate impusă. Între beneficiarii tehnicilor de compresie se numără utilizatorii calculatoarelor din generația multimedia, agențiile pentru tehnică spațială, astronomia, medicina, companiile de televiziune și nu în ultimul rând, armata.

Exemplele prezentate sunt departe de a fi exhaustive. Lista problemelor care își așteaptă rezolvarea pare să se extindă, o dată cu cristalizarea unor noi subdomenii de lucru.

Diversitatea aplicațiilor face dificilă încercarea de reprezentare unitară a etapelor de prelucrare și a relațiilor dintre acestea. Cu toate acestea în Figura 1 se propune o schemă-bloc generală a procesului de prelucrare numerică a imaginii.

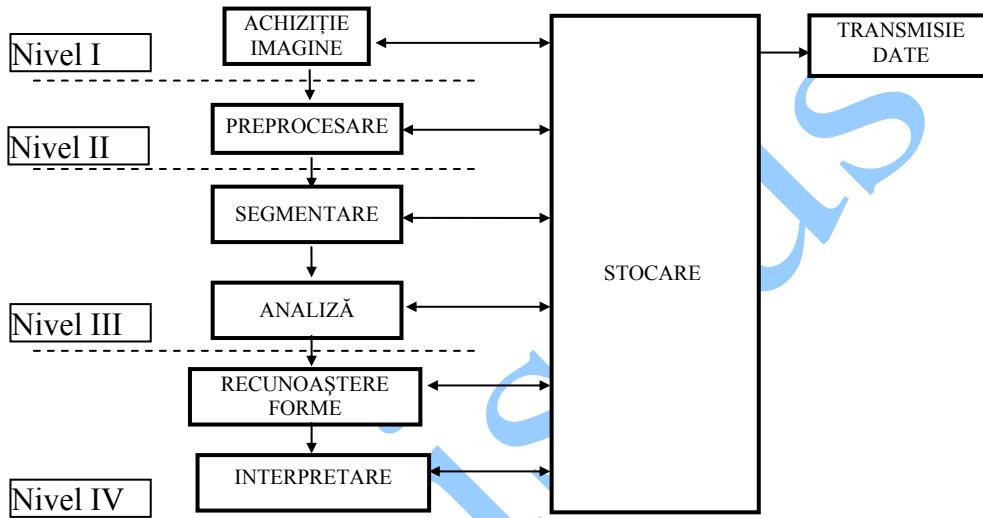


Figura 1. Schema-bloc generală a procesului de prelucrare digitală a imaginilor

Achiziția imaginii se poate considera în sens lărgit ca fiind preluarea imaginii direct de la sursă sau dintr-o memorie (bază de date) unde ea a fost anterior depozitată într-o formă comprimată. În aplicațiile de compresie a imaginilor (nivel de complexitate I) sunt implicate doar achiziția stocarea compresia și transmisia informației. La recepție prin procedura de decomprimare se obțin imagini identice sau foarte apropiate de cele inițiale.

Preprocesarea cuprinde etapele de eliminare a efectului perturbațiilor și de ameliorarea a imaginilor. Aceste prelucrări se execută de obicei prin aplicarea unui set de transformări sau filtrări asupra imaginii inițiale. Procedurile având nivelul de complexitate II (low level) se opresc la această etapă. Rolul lor este acela de a îmbunătăți imaginile în vederea unei interpretări efectuate ulterior de operatorul uman.

Segmentarea este o prelucrare ce permite extracția obiectului analizat din imagine. Acuratețea cu care se execută această separarea obiect-fond este esențială pentru obținerea unor rezultate corecte în etapele următoare.

Analiza imaginii presupune efectuarea unor măsurători ale parametrilor obiectului studiat și stabilirea unei descrieri matematice a acestuia. În situația în care procedura are nivelul de complexitate III rezultatele măsurătorilor se stochează sau se transmit ele fiind ulterior folosite de operatorul uman pentru interpretare.

În cazul procedurilor cu nivel de complexitate IV, parametrii obiectului analizat se assemblează într-un model matematic, care este apoi comparat cu modele tip ale unor clase de obiecte cunoscute. Funcție de gradul de asemănare se decide clasificarea obiectului studiat în una din clasele disponibile, adică se face recunoașterea acestuia. Recunoașterea formelor este în general urmată de o interpretare automată a întregii imagini. De exemplu, în cazul vederii artificiale la roboți, interpretarea înseamnă înțelegerea distribuției spațiale a obiectelor din câmpul vizual și adoptarea deciziilor în consecință.

Prelucrarea efectivă a imaginilor se face pe un calculator PC, o stație grafică sau un echipament specializat. În general, echipamente specializate se folosesc pentru aplicații în exploatare, la care metodele de prelucrare au fost dezvoltate și optimizate, astfel încât a fost posibilă stabilirea cerințelor hardware. În ultimul timp, au devenit disponibile comercial valoroase pachete software dedicate prelucrării imaginilor. Acestea conțin numeroși algoritmi de uz general și constituie un excelent punct de plecare în rezolvarea unor aplicații. Totuși, ca regulă generală, soluționarea unei probleme specifice necesită un efort de cercetare și dezvoltare proprie.

2 Proiectarea programului de prelucrare

Proiectarea aplicației de prelucrare a pornit de la stabilirea unor principii de implementare astfel încât să se asigure versatilitatea programului rezultat în final:

- a) Aplicația trebuie să funcționeze într-o fereastră din mediile (Windows, Linux sau Solaris) și să respecte criteriile acestora;
- b) Programul trebuie să asigure afișarea imaginii originale și ulterior a celei prelucrate;
- c) Aplicația trebuie să cuprindă procesările de bază și posibilitatea ca printr-o procedură simplă să-i fie atașate noi module;
- d) Fiecare procesare trebuie să poată fi anulată;
- e) Rezultatele procesărilor trebuie să poată fi salvate.

Prin urmare au rezultat un număr de specificații de proiectare a aplicației dintre care cele mai importante sunt:

- Implementarea se va face în limbajului Java datorită independenței de platformă;
- Fereastra programului va conține o bară de meniuri cu meniurile "File" (manipulare fișiere) și "Process" (algoritmii de prelucrare);
- Meniul "Process" va cuprinde o comandă de validare care controlează acceptarea sau nu a rezultatelor procesării;
- Se implementează comanda "Custom" care permite utilizarea unor filtre 3×3 definite de utilizator.
- Meniul "File" va cuprinde comanda "Save" pentru salvarea imaginilor procesate.

Analizând aceste cerințe s-a stabilit că trebuie structurată o familie de clase care să reprezinte o bază pentru oricare din filtrele ce se vor implementa ulterior [HC01], [CGI00]. Clasa executabilă P_im este fiică a clasei swing JFrame și cuprinde următoarele componente: definirea ferestrei, bara de meniuri cu meniurile File și Process, zona de afișare a imaginilor, toolkit-ul. Clasa Filtru subclasă a clasei ImageFilter reprezintă clasa de bază pentru clasele care execută procesările propriuzise (de exemplu F_bin, F_med, etc.). În această clasă sunt definite matricile pentru imaginea originală și finală și ciclul dublu for de scanare pixel cu pixel a imaginii de procesat. Clasa Dg moștenitoare a clasei Dialog utilizată ca superclasă din care se derivează toate clasele de definiție a ferestrelor de dialog proprii unor procesări de tipul binarizare, fereastră de gri, filtru Custom, etc.

Salvarea imaginilor finale a fost implementată sub forma unei clase SaveJpg care cuprinde liniile de cod necesare pentru crearea unui flux de ieșire a datelor, transmiterea imaginii într-un bufer și inscrierea ei într-un fișier [LS01].

3 Descrierea funcționării aplicației

Aplicația se prezintă, în conformitate cu specificațiile de proiectare anterioare, sub forma unei ferestre ce cuprinde o bară de meniu și un spațiu pentru afișarea imaginilor. Bara de meniu conține meniul "File" cu comenzile Open, Save și Exit și meniul "Process" unde sunt plasate comenzile Binarizare, Fereastră, Histograma, Netezire, Median, Laplace, Hough, Custom și Validate. Figura 2.a. prezintă fereastra cu o imagine afișată iar 2.b. prezintă execuția unei comenzi în meniul Process.

Comanda Binarizare produce apariția pe ecran a unei ferestre de dialog ce afișază histograma și permite utilizatorului să selecteze valoare pragului

P de segmentare. După scrierea valorii pragului și apăsarea butonului Try, P este marcat pe histogramă, iar utilizatorul poate să execute sau nu procesarea prin comanda butoanelor OK respectiv Exit.



a.

b.

Figura 2. Fereastra programului de procesare P_im

Comanda Fereastră este similară cu Binarizare dar în acest caz există două praguri P1 și P2. Utilizatorul are și de această dată posibilitatea să testeze valorile alese pentru ca apoi să decidă dacă aceste praguri sunt corecte. Figura 3. reprezintă fereastra de dialog proprie acestei comenzi în momentul în care P1 și P2 au fost selectate și afișate pe histogramă.

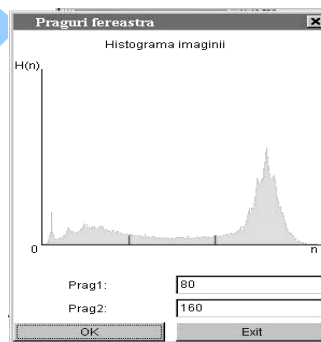


Figura 3. Fereastra de Dialog pentru selecția pragurilor

Comanda Histograma prezintă în mod informativ Histograma imaginii și reprezintă o posibilitate de verificare a efectului procesărilor asupra imaginilor prelucrate. Implementări ulterioare vor cuprinde pornind de la această comandă algoritmul de egalizare a histogramei.

Comenzile Netezire și Median sunt corespunzătoare celor două filtre standard pentru reducerea ygomotului în imaginile supuse procesării. Rularea aplicației în două ferestre separate permite efectuarea de comparații între rezultatele acestor tehnici. Filtarea Laplace este cea mai utilizată metodă de detecție a conturilor și este implementată aici în varianta clasică:

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

Transformata Hough reprezintă un caz mai special deoarece datele de ieșire sunt prezentate separat sub forma unui grafic într-o fereastră de dialog. Comanda Custom permite utilizatorului să experimenteze alte tipuri de filtre definite în bibliografia de specialitate. Dacă procesarea încercată se dovedește utilă, implementarea noului filtru este imediată prin introducerea unei noi clase moștenitoare a clasei filtru și similară cu cele deja existente. Comanda Validate este introdusă pentru a permite utilizatorului să accepte sau nu rezultatul ultimei procesări. În cazul în care ea nu este apăsată următoarea procesare porneste de la rezultatele anterioare etapei nevalidate.

Bibliografie

- [CGI00] **Chan M.C., Griffith S.W., Iasi A.F.**, *Java 1001 secrete pentru programatori*, Teora București, 2000
- [HC01] **Horstmann C. S., Cornell G.**, *Core Java*, Sun Microsystems Press, San Antonio, California USA, 2001
- [LS01] **Lacrămă D.L., Snep I.**, *JPG Image Manipulation in Java*, Proceedings of the AEG Symposium, 15-17 Nov. 2001, Prague, Czech Rep. pp. 110-113