

ELABORAREA APLICAȚIEI PENTRU SELECTAREA FILTRELOR DE PRELUCRARE A IMAGINILOR RECEPȚIONATE DE PE SATELITUL SATUM

Aurelia BALMUȘ, Denis MALENDĂ

Universitatea Tehnică a Moldovei

Abstract: În lucrarea dată s-a realizat aplicația pentru filtrarea imaginilor pe baza matricei de convoluție în sistemul Delphi Builder. Această aplicație, permite alegerea filtrului pentru prelucrarea imaginilor transmise la stația terestră a satelitul SATUM. Pe viitor metoda respectivă v-a fi implementată hardware pe FPGA.

Cuvinte cheie: convoluție, prelucrarea imaginilor, SATUM, masca/filtru.

1. Introducere

Imaginile primite de la satelitul SATUM sunt de o calitate foarte joasă, iar pe stația terestră ele trebuie prelucrate cu scopul de a diminua contrastul redus, pentru obținerea imaginilor de o calitate mai înaltă.

Prelucrarea volumului mare de informație cere mult timp, iar pentru înlăturarea acestor neajunsuri, pe stația terestră se va utiliza FPGA (field-programmable gate array – rețea programabilă de porți logice), pe care calculele se fac în paralel. Elaborarea aplicației s-a realizat pentru a putea alege filtrele pentru realizarea pe FPGA. Înaintea realizării pe FPGA este necesară realizarea unei aplicații, folosind metoda de convoluție care permite procesarea imaginilor utilizând orice tip de filtru pentru îmbunătățirea caracteristicilor de calitate. Această aplicație permite să alegem filtrul optimal pentru cazul nostru, care apoi o să fie realizat pe FPGA. S-a ales metoda spațială, deoarece timpul de prelucrare a unei imagini este de aproximativ de zece ori mai mic în comparație cu metodele spectrale.

2. Filtrarea imaginilor prin metoda de convoluție

Metoda de convoluție stă la baza aplicării operațiilor liniare de filtrare a imaginilor aplicate în domeniul spațial (în planul imagine prin manipularea directă a pixelilor de imagine). Exemple de astfel de filtre sunt: filtre trece jos (de netezire a imaginilor, de eliminare, de eliminare a zgomotelor), filtre trece sus (de evidențiere a muchiiilor).

Operația de convoluție implică folosirea unei măști/nucleu de convoluție H (de obicei de formă simetrică de dimensiune $w \times w$, cu $w=2k+1$) care se aplică peste imaginea sursă în conformitate cu (1).

$$I_D = H * I_S \quad (1)$$

$$I_D(x, y) = \sum_{i=-k}^k \sum_{j=-k}^k H(i, j) * I_S(x + i, y + j), \quad x = 0 \dots \text{înălțime} - 1, y = 0 \dots \text{lățime} \quad (2)$$

Aceasta implică parcurgerea imaginii sursă I_S , pixel cu pixel, ignorând primele și ultimele k linii și coloane (Fig.1.) și calcularea valorii intensității de la locația curentă (x, y) a imaginii de ieșire I_D în conformitate cu (2). Nucleul de convoluție se poziționează cu elementul central peste poziția curentă (x, y) .

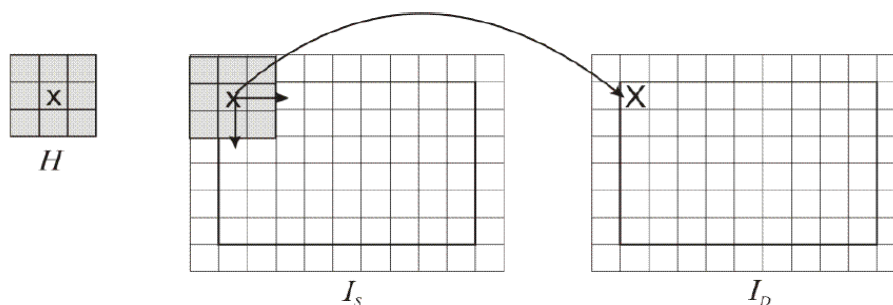


Figura 1 Ilustrarea operației de convoluție

Nucleele de convoluție pot avea și forme ne-simetrice (elemental central / de referință) nu mai este poziționat în centrul de simetrie). Modul de aplicare a operației de convoluție cu astfel de nuclee este similar, dar astfel de exemple nu vor fi prezentate în lucrarea de față.

3. Prelucrarea imaginii cu ajutorul filtrelor

Prelucrarea imaginilor are loc conform algoritmului prezentat în (fig.2). În primul rând se efectuează alegerea imaginii prelucrate (blocul 1) și citirea măștii introduse de către utilizator (blocul 2). După această se începe citirea și prelucrarea fiecărui pixeli a imaginii, care constă din ciclul după lățimea imaginii (blocul 3), în care se află ciclul după lungimea imaginii (blocul 4). Din imagine se citește pixelul curent, care o să fie prelucrat și 8 pixeli din prejur (blocul 5), care apoi sunt multiplcate după formula:

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} * \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} = (1 * a) + (2 * b) + (3 * c) + (4 * d) + (5 * e) + (6 * f) + (7 * g) + (8 * h) + (9 * i) \quad (3)$$

și înscrise în loc de pixelul curent (blocul 13). Acest algoritm este efectuat de 3 ori pentru fiecare culoare separat. Înainte de a înscrie valoarea nouă a pixelului este necesar de a-l verifica, dacă rezultatul formulei (3) nu a depășit valoarea maximală pentru format „bmp” – 255 (blocul 10) sau nu este mai mic ca zero (blocul 8).

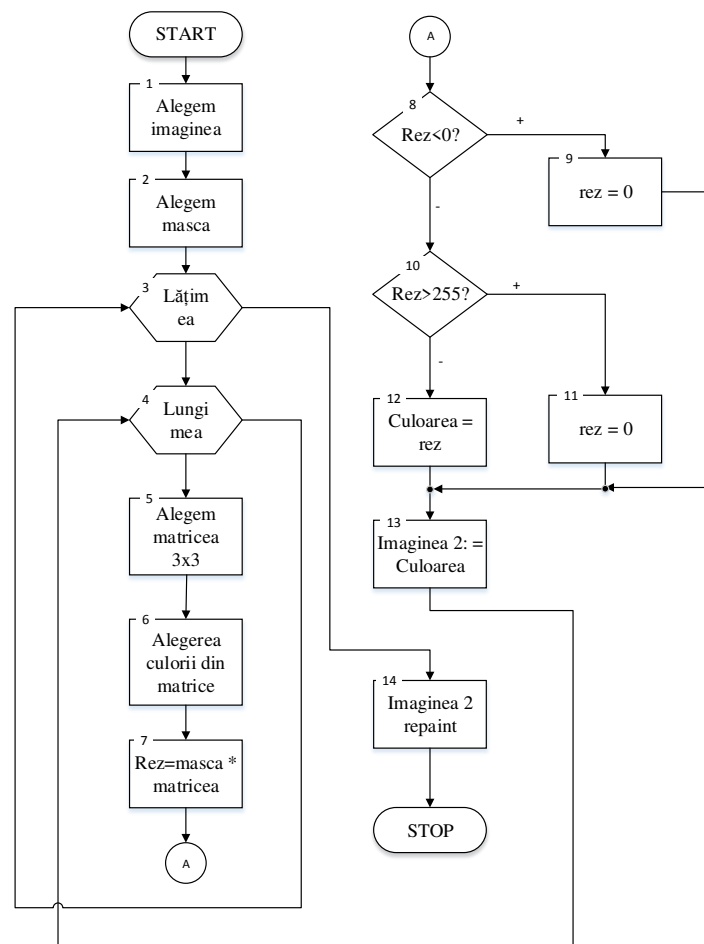


Figura 2 Algoritm de filtrare a imaginii

Pentru punerea în aplicare a metodei de prelucrare a imaginilor s-a realizat programul în sistemul Delphi Builder care este prezentat mai jos.

```

for u := 0 to Image1.Picture.Width - 3 do //Ciclu după lățimea a imaginii prelucrabile
begin
  for v := 0 to Image1.Picture.Height - 3 do // Ciclu după lungimea a imaginii prelucrabile
  begin
    pixel:= Image1.Canvas.Pixels[u,v]; //Alegerea pixelul care va fi prelucrat
    r:=GetRValue (pixel); //Alegerea culorii din pixel care va fi prelucrat
    temp[0,0]:=r; //Atribuirea matricei prelucrabile valorile culorilor din pixel curent și cei din prejur
    cash:=(mask[0,0]*temp[0,0])+(mask[1,0]*temp[1,0])+(mask[2,0]*temp[2,0])+(mask[0,1]*temp[0,1])+(
    mask[1,1]*temp[1,1])+(mask[2,1]*temp[2,1])+(mask[0,2]*temp[0,2])+(mask[1,2]*temp[1,2])+(mask[2,2]*t
    emp[2,2]); //Înmulțim măca cu matricea prelucrabilă temp și atribuim rezultatul primit variabilei cash
    if cash<0 then r:=0 //Dacă valoarea cash e mai mică decât 0 atunci valoarea culorii va fi minimă
    else if cash>255 then r:=255 //Dacă valoarea cash e mai mare decât 255 atunci valoarea culorii va fi maximă
    else r:=cash; //În caz contrar valoarea culorii va fi egală cu valoarea cash
    Image2.Canvas.Pixels[u+1,v+1]:= RGB(r,g,b); //Efectuând toți acești pași cu toate trei
    culori atribuim rezultatele primite a r,g,b imaginii a doua și așa până se va prelucra toată imaginea
    Image2.Repaint; //Afișăm imaginea
  end
end

```

În urma efectuării programului este necesar punerea în aplicare a metodei de prelucrare a imaginii, pentru a putea vedea rezultatele prezentate mai jos.

În (fig.3) este prezentat high pass filter - filtru trece sus, care este destinat creșterii contrastului și accentuării detaliilor. Operația de convoluție cu nuclee de acest tip are ca efect punerea în evidență a zonelor din imagine în care există variații bruște ale intensității pixelor (cum sunt de exemplu muchiile). Ele realizează o filtrare de tip “trece -sus”, care permit doar trecerea frecvențelor înalte. Nucleele folosite pentru detecția punctelor de muchii au suma elementelor component nulă.

Matricea nucleu pentru filtru trece sus:

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

(4)



a)

b)

Figura 3 Filtru trece sus: a) imaginea originală, b) rezultatul obținut în urma filtrării de tip filtru trece sus cu nucleul (3x3)

Filtru de tip “trece - jos”, prezentat în (fig.4), folosește pentru operații de netezire și/sau filtrare a zgomotelor care permit trecerea doar a frecvențelor joase. Efectul lor este o mediere a pixelului curent cu valorile vecinilor săi. Aceste nuclee au doar elemente pozitive. Din acest motiv, o practică curentă este împărțirea rezultatului convoluției cu suma elementelor nucleului de convoluție cu scopul de a scala rezultatul în domeniul de valori al intensității pixelor din imaginea de ieșire.

Matricea nucleu pentru filtru medie aritmetică:

$$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

(5)



a)

b)

Figura 4 Filtru trece jos: a) imaginea originală, b) rezultatul obținut în urma filtrării de tip filtru medie aritmetică cu nucleul (3x3)

4. Concluzii

În lucrarea dată a fost realizată aplicația pe baza metodei de convoluție care reduce costurile hardware și timpul de procesare în comparație cu aplicațiile bazate pe metodele din domeniu de frecvență. Un alt avantaj al aplicației respective, este aceea că permite utilizatorului să introducă orice tip de filtru necesar situației. Aceste caracteristici permit programei dezvoltate în viitor, să pună în aplicare algoritmul dezvoltat pe baza FPGA. În cazul dat s-a utilizat filtru trece sus și filtru trece jos cu matricea nucleu 3x3, care permit evidențierea muchiilor sau netezirea, în baza cărora am efectuat prelucrarea imaginilor și am obținut o calitate mai bună pentru cazul nostru de lucru cu satelitul.

Bibliografie

1. Mircea V. Popa, Romana Oancea, Ștefan Demeter, Alina Hangan, "Prelucrarea numerică a imaginilor".
2. Rafael C. Gonzalez, Richard E. Woods, Steven L. Eddins, "Digital Image Processing Using MATLAB, 2nd edition," Gatesmark Publishing, 2009, 827 p. - ISBN: 9780982085400.
3. P.A. Шовенгердт, "Дистанционное зондирование. Модели и методы обработки изображений. Часть 1," М.: Техносфера, 2010. - 560 с. - ISBN: 978-5-94836-244-1.
4. A. Cosolapov, "Prelucrarea imaginilor captate de microsatelitul SATUM cu aplicarea Hard-urilor în baza FPGA," Lucrarea de master, UTM, FIMET, 2012.