

Tema 1 - Prelucrarea imaginilor

Nume: Mireanu Cosmin

Grupa: 331AB

In fisierul process.v este reprezentat un state machine, cu 21 de stari, care proceseaza imagini de 64x64 dupa cele 3 cerinte, mirror, grayscale si filtru de sharpen, in ordinea ceruta.

Codul este impartit in 2 parti:

- Partea secventiala

Aici se actualizeaza state, row si col, care ajuta la parcurgerea imaginii. Out_we devine LOW, de fiecare data pe acest front. Aceasta parte din cod se ruleaza pe frontul pozitiv al ciclului de ceas.

- Partea combinationala

In aceasta parte a codului, se executa pe toate fronturile de ceas, case-urile, care sunt parcurse pe rand, toate.

Algoritm mirror:

Am declarat 2 variabile auxiliare, aux1_pix si aux2_pix, de aceeasi marime ca in_pix/out_pix, pentru a retine pixelul curent si cel oglindit, de pe aceeasi coloana, dar pe randul: 63- randul curent.

Dupa ce retinem pixelul curent, verificam daca nu am ajuns la jumatatea imaginii ($row \geq 32$), si trecem la pixelul oglindit, in caz opus, se considera ca oglindirea a fost efectuata cu succes (se activeaza flag-ul mirror_done) si se trece mai departe la greyscale.

Retinem pixelul oglindit, si scriem in out_pix, prima valoare, a pixelului curent. Ne intoarcem la pixelul original, si scriem in out_pix, a doua valoare, a pixelului opus.

In ultima stare, verificam daca am ajuns la ultima coloana, in caz afirmativ trecem la urmatorul rand si resetam coloana la 0.

Algoritm Grayscale:

Am declarat 3 variabile R,G,B pentru a retine valorile din pixel, max_value, min_value si gray_value, unde vom calcula media.

Retinem din pixelul curent valorile pentru cele 3 canale de culori si aflam care este maximul si minimul.

Calculam media dintre cele 2, si scriem in out_pix, pe canalul R valoarea calculata, iar pe canalele G si B le reinitializam cu 0.

Verificam in ultima stare daca am ajuns la capat de coloana, daca nu, trecem la urmatorul element ($\text{next_col} = \text{col} + 1$), altfel trecem la urmatorul rand si resetam coloana la 0. Dupa ce se trece prin toata matricea, flag-ul gray_done este activat si se trece la filtru de sharpness.

Algoritm Filter:

Chiar daca nu am reusit sa termin codul, voi prezenta ideea mea de a rezolva acest subpunct al temei.

Am declarat 2 variabile auxiliare care retin la inceput valoarea curenta a randului si a coloanei, a pixelului unde ne aflam. Sum va fi actualizat cu elementele din jurul pixelului, inmultite cu elementele corespondente din matricea de convolutie (pixelul curent * 9, restul din jur, daca exista, cu (-1)).

Retinem valorile curente ale indicilor, si initializam pe cele 3 canale sum = 0.

Pentru cazul general (singurul scris in cod), cand nu ne aflam pe margini sau colturi si putem forma o matrice 3x3, adaugam valorile din pixelul curent inmultite cu 9.

Trecem prin toate cele 8 pixeli vecini si inmultim cu (-1), si le adaugam in sum. La final, revenim la randul si coloana initiala, cu ajutorul celor 2 auxiliare, si

notam in out_pix valorile calculate. Continuum, trecand prin restul de elemente ale matricei.

In cazul in care ne aflam intr-un colt (neimplementat), ar trebui sa calculam doar valoarea curenta, si cei 3 pixeli posibili vecini, restul fiind elemente nule care nu exista in matrice.

In cazul unei margini, avem 5 posibili vecini, restul de 3 fiind elemente nule.

Exemplu colt stanga sus:

0 0 0

0 X y

0 y y

Exemplu margine sus:

0 0 0

y X y

y y y

Unde X este pixelul curent, iar y vecinii.

Update:

In urma unei analize mai amanuntite a cerintei, am descoperit de ce nu imi functioneaza codul meu.

Abordarea era gresita deoarece, eu nu am retinut valorile originale din matrice, si le modificam pe restul cu valorile obtinute dupa aplicarea matricei de convolutie.

O solutie ar fi stocarea in o variabila auxiliara 2 randuri a cate 64 elemente, 2x64, initializata la inceput cu 0.

Dupa ce este aplicata matricea de convolutie pe primul element, se pune valoarea in out_pix, iar in matricea 2x64, pe prima pozitie, se pune valoarea initiala. La calcularea urmatorului element, se folosesc valorile din in_pix, iar pentru elementul din stanga, valoarea initiala, cea din matricea 2x64.

Se va aplica matricea de convolutie si pe restul elementelor, iar dupa ce se ajunge la ultimul element de pe coloana, la primul element de pe randul 2, se vor folosi toate elementele de sus din matrice, iar restul din in_pix.

Ulterior, cu cat avansam in matrice, shiftam al doilea rand mai sus, sa putem memora randul pe care suntem curent.

Acest lucru se va face pana se ajunge la ultimul rand.

Cazurile cu colturi si margini vor fi luate individual, ca mai sus.