

Documentatie proiect

Programare Procedurala

Nume: Voinea Andrei

Grupa: 141

1. Functia XORSHIFT32 genereaza “lungime”-1 numere pseudo-aleatoare conform algoritmului cu acelasi nume. Seed-ul va fi considerat primul element al vectorului.
2. Functia liniarizare primeste o imagine si baga pixelii care constituie imaginea intr-un vector de $\text{inaltime_imagine} \times \text{latime_imagine} \times 3$, datorita faptului ca interschimb si canalele de culoare R cu B, astfel vectorul meu avand pe pozitia $v[i], v[i+1], v[i+2]$, valorile R, G, respectiv B ale pixelului i.
3. Functia MemorieExternaLiniarizare transforma vectorul primit ca parametru in imaginea “destinatie”, copiind in primul rand header-ul din imaginea “sursa”.
4. Functia criptare_poza primeste ca parametri imaginea care trebuie criptata, numele fisierului unde se va salva imaginea criptata, cat si .txt-ul unde se afla cheia secreta. Functia cripteaza poza in modul prezentat in PDF.
5. Functia decriptare_poza primeste ca parametri imaginea care trebuie decriptata, numele fisierului unde se va salva imaginea decriptata, cat si .txt-ul unde se afla cheia secreta. Functia decripteaza poza in modul prezentat in PDF.

6. Functia chitest primeste ca parametru numele unei imagini si va calcula CHI SQUARED pentru aceasta.
7. Structura Punct contine corr (corelatia), x,y (coordonate), height,width si culoare (inaltimea, latimea sablonului, cat si culoarea care va fi folosita pentru acesta).
8. Functia comparator este realizata in asa fel incat qsort-ul va ordona vectorul in ordinea descrescatoare a valorilor de corelatie.
9. Functia liniarizare_Matrice face aceeasi chestie ca si functia liniarizare, doar ca pune pixelii intr-o matrice.
10. Analog MemorieExternaLiniarizare_Matrice cu MemorieExternaLiniarizare.
11. Functia gray_img primeste ca parametru o imagine si va pune intr-o matrice (ca la liniarizare_Matrice) octetii R,G,B dupa aplicarea calcului de grayscale. (Octetii vor fi egali).
12. Functia template_matching primeste ca parametrii numele imaginii mare, pe care se vor glisa sabloanele, numele unui sablon, pragul pentru care corelatia trebuie retinuta, un vector de tip Punct (structura) ce se va modifica in interiorul functiei, lungimea vectorului de tip Punct care se va modifica si ea in interiorul functiei, cat si ce culori se folosesc pentru colorarea sablonului.
13. Functia colorare primeste numele imaginii mare, o variabila de tip Punct (structura) si matricea imaginii (se foloseste liniarizare_Matrice in main). Functia coloreaza un chenar plecand de la Punct.x si Punct.y mergand pana la capat folosindu-se de Punct.height si de Punct.width. Culoarea folosita se afla in Punct.culoare.
14. Functia eliminare_non_maximale primeste ca parametri vectorul de tip Punct si lungimea acestuia, variabile modificate si la template_matching. Functia sorteaza cu un qsort la inceput vectorul dupa corr (corelatie) si

folosind doua for-uri, se elimina detectiile ale caror suprapunere cu (*D)[i] (vectorul de tip Punct) este mai mare de 0.2.

15. MAIN. In MAIN, se apealeaza `criptare_poza`, `decriptare_poza` si `chitest` pentru imaginile care vor fi citite de la tastatura, cat si `.txt`-ul cu cheia secreta. Pentru partea de template matching se ia vectorul `*D` care este de tip Punct (structura) si se face NULL si lungimea acestuia care se initializeaza cu 0. Se citeste numele imaginii mare, pe care se doreste sa se sa faca `template_matching` (ex: `test.bmp`), dupa care se introduce de la tastatura numarul sabloanelor folosite. Dupa aceasta, verificatorul este pus dintre a alege sa se foloseasca culorile corespunzatoare fiecărei cifre folosite in exemplul din PDF (ex: cifra0 avea 255,0,0 si tot asa) sau sa aleaga sa le introduca de la tastatura (in cazul in care se foloseste un alt exemplu). Dupa decizia luata, se vor introduce de la tastatura cele n sabloane. Dupa introducerea celor n sabloane, se vor elimina non-maximalele si se va utiliza functia de colorare. La final, verificatorul este pus sa aleaga unde va salva poza modificata.