

Documentatie proiect Programare

Procedurala

Dobre Mihaela-Beatrice – Grupa 142

1. In headerul UTILE am creat o structura in care sunt memorate principalele informatii ale unei imagini de care voi avea nevoie pe parcursul realizarii programului.
In structura IMG sunt salvate inaltimea(H), latimea(W), header-ul si vectorul liniarizat (L) al imaginii ce trebuie criptate.
In structura IMG2 sunt salvate inaltimea(H), latimea(W), header-ul si matricea (M) corespunzatoare pixelilor unei imagini.
In structura POINT sunt memorate coordonatele centrului unei ferestre (x si y), corelatia (corr) corespunzatoare, iar digit reprezentand cifra asociata ferestrei.
2. Functia XORSHIFT genereaza numere pseudo-aleatoare conform algoritmului XORSHIFT32 cu seed-ul preluat dintr-un fisier text, memorandu-le intr-un tabloul unidimensional(R);
3. In functia LINIARIZARE citesc matricea corespunzatoare unei imagini BMP si o memorez intr-un vector liniarizat incepand cu coltul din stanga jos
4. Functia SALV_EXT salveaza in memoria externa o imagine BMP memorata in forma liniarizata;
5. Functia PERM_ALEATOARE construiește o permutare aleatoare folosind algoritmul lui Durstenfeld;
6. Functia PERMUTA permuta pixelii folosind permutarea aleatoare generata anterior(pepper.bmp);
7. Functia CRIPTARE realizeaza criptarea propriu-zisa, aplicand substitutia imaginii permutate anterior(criptat.bmp);

8. Functia PERMUTARE_INVERSA construiește o permutare aleatoare folosind algoritmul lui Durstenfeld, apoi îi calculează și returnează inversa;
9. Functia DECRYPTARE1 aplică asupra imaginii criptate inversa relației de substituție, obținându-se o imagine intermediară(semi_decr.bmp);
10. Functia PERMUTA_DECRYPTARE realizează permutarea pixelilor din imaginea intermediară, obținându-se astfel imaginea decriptată(decriptat.bmp);
11. Functia CHI realizează testul CHI SQUARED pentru verificarea corectitudinii criptării;
12. Functia GRAYSCALE transformă o imagine color în una alb negru;
13. Functia CITIRE extrage din imagine și adaugă în structura IMG2 informațiile necesare pentru prelucrarea unei imagini;
14. Functia TEMPLATE_MATCHING implementează algoritmul de template matching între o imagine I și un șablon S;

În cadrul acesteia, se creează o matrice(a) care reprezintă matricea corespunzătoare imaginii la care se adaugă pixeli de culoare neagră în afara imaginii I. După aceea, se calculează corelația dintre șablonul S și fereastra f1 cu centrul în punctele x și y.

Corelațiile cu valoare mai mare decât pragul, sunt memorate într-un tablou unidimensional(D) de tip POINT .
15. Functia CITIRE_COMPLETA creează un tablou tridimensional ce memorează matricea corespunzătoare pixelilor unei imagini, dar permite și accesul la fiecare din cei trei octeți ai fiecărui pixel;
16. Functia COLORARE realizează colorarea unei ferestre cu o anumită culoare;

17. Functia SALVARE salveaza extern noua imagine dupa colorare;

18. Functia QSORT_DETECTII realizeaza sortarea descrescatoare a corelatiilor din tabloul D cu ajutorul functiei qsort din biblioteca stdlib.h;

19. In MAIN am creat:

- Functia SUPRAPUNERE care calculeaza suprapunerea spatiala dintre doua detectii a si b dupa formula data.
- Functia CRIPTARE_IMG care combina functiile necesare si realizeaza modulul de criptare si decriptare al unei imagini.
- Functia RECUNOASTERE_PATTERN in care se elimina non-maximele si se realizeaza modulul de recunoastere a pattern-urilor intr-o imagine.

In final, sunt apelate cele 2 functii (CRIPTARE_IMG si RECUNOASTERE_PATTERN) pentru finalizarea proiectului.

In fisierele text NUME_FISIERE_CRIPTARE si NUME_FISIERE_TEMPLATE se gasesc caile fisiereilor.