Documentatie proiect programare procedurala

noiembrie 2018

Biro Balan Antonia

Grupa 132

Introducere

Acest proiect cripteaza, decripteaza si detecteaza pattern-uri intr-o imagine. Mai jos sunt descrise pe scurt functiile si structurile folosite.

Structuri

- `pixel` contine cele trei canale de culori R G B
- `imagine` contine un vector de pixeli si dimensiunea imaginii
- `detectie` contine scorul, coordonatele coltului stanga sus a ferestrei, dimensiunea ferestrei si culoarea cu care trebuie
- `vect_detectii` contine un vector de detectii si lungimea sirului

Headere

Pentru criptare/decriptare:

`criptare.h`

- `criptare` care liniarizeaza imaginea primita in fisierul `imagine_init,` modifica vectorul de pixeli obtinut si rescrie imaginea modificata (criptata) in fisierul `imagine_fin,` iar in fisierul cheie returneaza seed-ul si sv-ul.
- `rescrie` construieste in memoria externa o imagine primita ca parametru in forma liniarizata. Header-ul se copiaza din fisierul inital `fisier_init` transmis ca parametru.
- `liniarizare` primeste ca parametru un fisier .bmp si transforma imaginea intr-un vector de pixeli. Returneaza adresa la care a fost memorat.
- `XORSHIFT32` genereaza un numar aleator in functie de seed-ul transmis ca parametru.
- `permutare` creeaza o permutare de n elemente cu ajutorul sirului pseudo-aleator r.
- `permutare_imagine` aplica o permutare asupra pixelilor unei imagini.
- `criptare_cu_XOR` primeste o imagine cu pixelii permutati si aplica criptarea cu XOR. Este folosita a doua jumatate a sirului pseudo-aleator r.
- `generare_sir_nr_pseudo_aleatoare` creeaza un sir pseudo-aleator de lungime n folosind algoritmul `XORSHIFT32` pornind de la valoarea initiala seed.

decriptare.h

- `permutare_inversa` generaza permutarea initiala si o cu ajutorul ei permuta invers imaginea.
- `decriptare_cu_XOR` primeste o imagine criptata si aplica decriptarea cu XOR. Este folosita a doua jumatate a sirului pseudo-aleator r.
- `decriptare` decripteaza o imagine primita ca parametru folosind o cheie secreta. Apeleaza functiile `permutare_inversa` si `decriptare_cu_XOR.`

test_chi_patrat.h

- `frecvente_pe_canale` creeaza trei vectori de frecventa pentru fiecare canal de culoare.
- `test_chi_patrat_pe_un_canal` calculeaza valoarea $\chi 2$ pentru un canal de culoare.
- `test_chi_patrat` afiseaza valoarea testului chi patrat pentru toate canalele de culoare.

Pentru Pattern-matching

grayscale.h

- `grayscale_image` transforma o imagine color intr-una gri.

cross_correlation.h

- `calculare_scor` calculeaza `intensitatea_medie` si `deviatie_standard` pentru un sablon si o fereastra date si le foloseste in determinarea scorului de corelatie.
- `template_matching` construieste un vector de detectii pentru un sablon si o fereastra date.

colorare.h

- `colorez` foloseste apelurile utile ale functiei `color_fereastra` pentru a colora toate detectiile din vector
- `aleg_culoare` decide culoarea corespunzatoare fiecarui sablon

eliminare_non-maxime.h

- `elimin_non_maxime` parcurge vectorul sortat de detectii si elimina detectiile non maxime marcandu-le cu un scor negativ, mai mic ca -1

identificare_patternuri.h

- `identificare_patternuri` ruleaza functia `template_matching` pentru fiecare sablon, reuneste detectiile intr-un vector final, il sorteaza, elimina non-maximele si apeleaza `colorez.`