

Documentatie:

^main.c

void encrypt

- citeste de la tastatura datele necesare pentru a apela functiile de criptare/printare a imaginii criptate

void decrypt

- citeste de la tastatura datele necesare pentru a apela functiile de decriptare/printare a imaginii decriptate

void t_matching

- citeste de la tastatura datele necesare pentru a apela functia de template matching

void chi

- citeste calea imaginii si apeleaza functia pentru testul chi

int main - functia main care contine un switch si in care se apeleaza functiile de mai sus

^bitmapIO.c - acest fisier sursa contine toate functiile necesare pentru a face citirea si afisarea unui bitmap

int bitmap_data_width

- returneaza lungimea imaginii

int bitmap_data_height

- returneaza inaltimea imaginii

int bitmap_data_size

- returneaza numarul de octetii pe care il ocupa imaginea

int pixel_to_pos

- primeste doua coordonate i si j si returneaza o valoare care corespunde cu indicele pixelului in imaginea liniarizata

unsigned char* bitmap_linearize

- liniarizeaza imaginea folosind functia pixel_to_pos si returneaza imaginea liniarizata

unsigned char* bitmap_unlinearize

primeste imaginea liniarizata si o deliniarizeaza pregatind imaginea pentru afisare

unsigned char* bitmap_load

- apeleaza functiile de mai sus si citeste o imagine o liniarizeaza si dupa returneaza imaginea liniarizata

void bitmap_unload

- folosind functiile de mai sus si primind un o imagine intr-un vector in forma liniarizata o deliniarizeaza si o afiseaza

^mathalgorithms.c - acest fisier sursa contine functii in care sunt implementati algoritmi care au legatura stricta cu

matematica

unsigned int* xorshift32

- functia implementeaza algoritmul xorshift32 si genereaza numere pseudoaleatoare pornind de la seed

int* random_permutation

- folosind numerele aleatoare generate de xorshift32 functia genereaza si returneaza un vector cu o permutare aleatoare

int* inverse_permutation

- primeste o permutare si returneaza inversa ei

^ bitmapalg.c - algoritmi care au leagatura cu prima parte a proiectului adica criptarea/decriptarea

unsigned char* bitmap_alg_permutation

- primeste o imagine si o permutare si aplica acea permutare asupra imaginii

unsigned char* bitmap_xor

- aplica algoritmul xor pentru criptarea imaginii

unsigned char* bitmap_inverse_xor

- aplica inversul algoritmului xor pentru decriptare

unsigned char* bitmap_alg_crypt

- primind o imagine si folosind functii de mai sus cripteaza o imagine si o returneaza

unsigned char* bitmap_alg_decrypt

- primind o imagine criptata si folosind functii de mai sus decripteaza imaginea si o returneaza

float* chi_test

- returneaza si afiseaza testul chi aplicat imaginii cu calea data

^ TemplateMalg.c - acest fisier sursa contine functii pentru aplicarea de Template Matching

int qsort_comparator

- functia specifica care compara elemente din struct pentru qsort

void TM_grayscale

- primeste o imagine si intoarce imaginea grayscale

int d2

- o formula matematica care din doua coordonate intoarce un indice.

- cu aceasta formula evitam folosirea matricei pentru retinerea imaginii

unsigned char* cut_template

- taie un dreptunghi in forma templateului din imagine

- aceasta functie primeste coltul din stanga sus si lungimea/inaltimea bucatii pe care o vrem taiata

float average_template_gray_intensity

- calculeaza media intensitatii din imagine

- utila pentru formula corelatiei

float standard_deriv_template

- calculeaza derivata dupa formula data

- utila in calculul corelatiei

float TM_corelation

- folosind cele doua functii de mai sus calculeaza corelatia si o returneaza

void template_matching

- parcurge aproape toata imaginea si gliseaza sablonul pe imagine
- aplica formula corelatiei si pentru corelatiile mai mare decat pragul dat prin parametru le introduce in un vector
- acel vector va fi sortat cu qsort si pe el va fi aplicat algoritmul eliminarii non-maximelor
- dupa va fi apelata functia draw

void draw_rect

- functia deseneaza pe imagine un dreptunghi incepand de la coordonatele coltului din stanga sus
- functia primeste culoarea dreptunghiului prin parametru

void draw

- functia parcurge vectorul dat si deseneaza corelitiile reusite din el
- cu culoarea corespunzatoare

void non_max

- functia aplica algoritmul eliminarii non maximelor utilizand functia area_sup

void area_sup

- functia calculeaza daca avem o intersectie intre cele 2 dreptunghiuri