

Proiect PP

1.Criptare/Decriptare

void xorshift(unsigned int w,unsigned int h,unsigned int **R,unsigned int seed)

-functia care imi genereaza numerele random pe care le retin in vectorul transmis prin referinta R

Parametrii:

w-latimea imaginii pe care trebuie sa o criptez

h-inaltimea imaginii pe care trebuie sa o criptez

R-vectorul care retine valorile generate

seed-seed-ul citit din fisierul secret_key.txt

void permutare(unsigned int **p,unsigned int w,unsigned int h,unsigned int *R)

-functie care genereaza permutarea sigma pe care o retin in vectorul p

Parametrii:

p-vectorul in care salvez permutarea generata

w,h-latimea si inaltimea imaginii de criptat

R-vectorul in care am valorile generate random din functia precedenta

void liniarizare(char* nume_imagine,unsigned int **v,unsigned int *latime_img,unsigned int *inaltime_img,unsigned char **header,int *padding)

-functie care imi liniarizeaza o imagine sub forma unui vector

nume_imagine-calea fisierului ce urmeaza a fi liniarizat

v-vectorul transmis prin referinta in care voi avea pixelii din imagine

latime_img,inaltime_img-latimea si inaltimea imaginii transmise prin referinta deoarece le citesc in cadrul functiei si vreau sa imi ramana vizibile si in afara acestuia

header-un vector in care retin headerul imaginii ca sa il scriu si in imaginile de iesire,pt ca au acelasi header

padding-variabila in care retin paddingul imaginii calculat in cadrul functiei si transmis prin referinta deoarece am nevoie de el atunci cand scriu imaginea

void deliniarizare(char* nume_fisier,unsigned int **v,unsigned int latime_img,unsigned int inaltime_img,unsigned char **header,int padding)

-functia pentru crearea si scrierea unei imagini conform valorilor din vectorul v

nume_fisier=numele fisierului pe care urmeaza sa il creeze functia si sa scrie in el ,obtinand o imagine

v-vectorul care contine pixelii ce vor fi scrisi in fisier

latime_img,inaltime_img-latimea si inaltimea imaginii ce urmeaza a fi scrise

header-vector care contine header ul salvat anterior si pe care il voi scrie si in imaginea curenta

padding-paddingul cu care completez imaginea in caz ca e nevoie

void criptare(char* imagine_sursa,char* imagine_criptata,char* fsecret)

-functia care realizeaza criptarea unei imagini

imagine_sursa-calea imaginii de criptat

imagine_criptata-calea imaginii criptate

fsecret-calea fisierului ce contine R0 si SV

void invers(unsigned int **p,unsigned int n,unsigned int **sigma)

-functia care afla permutarea inversa necesara la decriptare

p-vector in care am permutarea initiala

sigma-vector in care voi retine inversa permutarii

void decriptare(char* imagine_criptata,char* imagine_decriptata,char* fsecret)

-functia care realizeaza decriptarea unei imagini

Imagine_criptata-calea imaginii criptate

Imagine_decriptata-calea in care urmeaza sa creez si sa am imaginea decriptata

Fsecret-calea fisierului care contine R0 si SV

void chi(char* imagine)

-functie care realizeaza testul chi patrat si afiseaza valorile rezultate

Imagine-calea imaginii supuse testului

Template matching

void grayscale_image(char* nume_fisier_sursa,char* nume_fisier_destinatie)

-functie care realizeaza transformarea unei imagini in grayscale

Nume_fisier_sursa-calea imaginii originale

Nume_fisier_destinatie-calea in care voi crea si voi avea fisierul grayscale rezultat

int MAX(int a,int b)

-functie care returneaza maximul dintre a si b;voi avea nevoie de ea la calculul ariei pentru suprapunere

int MIN(int a,int b)

-functie care returneaza minimul dintre a si b

int compare (const void * a, const void * b)

-functie pe care o folosesc la qsort pt sortarea numerelor descrescator

unsigned int liniarizare_mat(char* nume_imagine,unsigned int *latime_img,unsigned int *inaltime_img,unsigned char **header,unsigned int *padding)**

-functie pe care o folosesc ca sa pun pixelii unei imagini intr-o matrice ca sa nu ma complic cu vector;returneaza adresa de inceput a matricei create

Nume_imagine-calea imaginii pe care vreau sa o trec in matrice

Latime_img,inaltime_img-sunt transmise prin parametru deoarece voi avea nevoie de ele mai tarziu si sa ramana salvate

Header-vector care retine headerul imaginii;voi avea nevoie de el cand transform matricea in imagine

Padding-paddingul imaginii ; voi avea nevoie tot cand transform din matrice in imagine

void deliniarizare_mat(char* nume_fisier,unsigned int **v,unsigned int latime_img,unsigned int inaltime_img,unsigned char **header,int padding)

-functie de transformare a unei matrici in imagine

Nume_fisier-numele fisierului care va fi creat si care contine imaginea rezultata

v-matricea pe care vreau sa o transform in imagine

latime_img,inaltime_img –latimea si inaltimea imaginii pe care o creez

header-vector ce contine headerul salvat anterior,am nevoie la creare noii imagini

padding-paddingul de adaugat in caz ca este nevoie

void colorare(char* nume_imagine,int j,int i,int latime_sablon,int inaltime_sablon,unsigned char r,unsigned char g,unsigned char b)

-functie care imi realizeaza colorarea marginilor ferestrei corespunzatoare si o si scrie in memoria interna dupa fiecare colorare

Nume_imagine-imaginea initiala in care voi realiza colorarea;in cazul asta este test.bmp, imaginea color cu cifrele scrise de mana

J,i-indicii care marcheaza coltul din stanga sus al ferestrei

Latime_sablon,inaltime_sablon-am nevoie de ele ca sa aflu celelalte puncte ale ferestrei si sa o pot parcurge

R,g,b-valorile care compun culoarea

void template_matching(char* nume_imagine,char* nume_sablon,int *s,struct detectie **d,unsigned char r,unsigned char g,unsigned char b,double ps)

-functie care gliseaza un sablon pe imaginea initiala si imi calculeaza corelatiile si imi pune ferestrele intr-un vector d de detectii

Nume_imagine – imaginea initiala, cea mare pe care voi glisa sablonul

Nume_sablon – calea sablonului

s-nr de elemente al vectorului de detectii (numarul de detectii)

d-vectorul in care salvez ferestrele

r,g,b-culoarea corespunzatoare fiecarei ferestre

ps-pragul pt corelatie(0,5 in cazul asta)

void sortare(struct detectie **d,int s)

-functie care imi realizeaza sortarea descrescatoare a vectorului d cu functia qsort

void eliminare(struct detectie **d,int *s,char*nume_sablon)

-functie care realizeaza eliminarea suprapunerilor

Nume_sablon-calea unui sablon ca sa citesc inaltimea si latimea acestuia deoarece am nevoie de ele la calculul ariei