-DOCUMENTATIE PROIECT-

BULMACI RALUCA MARINELA

Criptarea/decriptarea unei imagini în format BMP

Funcții și structuri folosite

typedef struct {unsigned char r,g,b;} pixel;

* Definește tipul de date PIXEL, ce va fi folosit pe tot parcursul programului pentru a reține pixelii din imaginile în format BMP utilizate.

typedef struct{ unsigned int W, H, dim\_img, dim\_padding; unsigned char header[54]; pixel \*vector; } imagine;

* Defineste tipul de date imagine, aceasta retine informatii legate de o imagine precum: latime(W) , inaltime (H), dimensiunea imaginii (dim\_img), headerul care are 54 de pixeli (header[54]) , un vector care va contine pixelii imaginii (\*vector) acesta urmand sa fie alocat dinamic.

imagine citire\_imagine(char \* nume\_fisier)

* Funcția încarcă în memoria internă imaginea de la adresa specificată.
* Se deschide fișierul aflat la adresa nume\_fisier prentru citire; în cazul unei erori la deschidere se afișează un mesaj sugestiv și se returnează valoarea 1.
* Se obțin dimensiunile (width, height) imaginii pentru a se calcula padding-ul imaginii și respectiv numărul de pixeli
* Se aloca memorie pentru tabloul “vector”
* Liniarizarea matricei, se parcurge imaginea pixel cu pixel și se copiază pixelul curent în tabloul \*vector, pe poziția corespunzătoare - se ține cont de ordinea inversată a liniilor.

unsigned int xorshift ( unsigned int numar\_generat\_anterior)

* Functia returneaza un numar aleator folosind algoritmul XORSHIFT32

pixel pixel\_xor\_constanta (pixel pixel1, unsigned int x)

* Se obtin cei 4 octeti ai intregului .
* Se returneaza rezultatul obtinut prin xor-area celor 3 canale de culoare cu octetul intregului corespunzator.
* Rezultatul este de tip pixel

pixel pixeli\_xor\_pixeli (pixel pixel1, pixel pixel2)

* Primeste ca parametri 2 pixeli
* Se xoreaza canalele de culoare ale celor 2 pixeli

pixel \* criptare (imagine img, char \* fisier\_cheie)

* Se citestesc din fisierul text fisier\_cheie cele doua valori, a- reprezentand numarul de la care se va porni pentru generarea numerelor aleatorii, SV – numarul intreg folosit la formula pentru xorare
* Se aloca memorie pentru vectorul cu numere aleatori
* Se genereaza o permutare conform algoritmului lui Durstenfeld
* Se permuta pixelii imaginii conform permutarii generate anterior
* Se xoreaza fiecare pixel respectand formula din pdf
* Returneaza vectorul cu pixelii permutati si xorati

imagine decriptare (imagine img, char \* fisier\_cheie)

* Se citestesc din fisierul text fisier\_cheie cele doua valori, a- reprezentand numarul de la care se va porni pentru generarea numerelor aleatorii, SV – numarul intreg folosit la formula pentru xorare
* Se aloca memorie pentru vectorul cu numere aleatori
* Se genereaza o permutare conform algoritmului lui Durstenfeld
* Se calculeaza permutarea inversa
* Se xoreaza pixelii conform formulei din pdf
* Se permuta pixelii imaginii conform permutarii inverse
* Se returneaza imaginea decriptata

void chi\_patrat (imagine img, char \*nume\_fisier)

* Cu ajutorul functiei fseek deplasam cursorul dupa header
* Citim fiecare canal de culoare urmand sa facem frecventa
* Calculez frecventa teoretica iar apoi conform formulei valoarea 𝜒2

Pattern-matching într-o imagine în format BMP

Functii si structure folosite

typedef struct { unsigned int x,y,W,H; double indice\_corelare; unsignedsablon\_folosit;} fereastra

* Defineste tipul de date fereastra, retine detalii legate de o fereastra: inaltimea(H), latimea(W), si coordonatele punctului din coltul stang, din partea de jos.

typedef struct{fereastra \*v\_ferestre; unsigned int numar\_ferestre;} vector\_cu\_ferestre;

* Defineste tipul de date vector\_cu\_ferestre, care retine numarul ferestrelor dar si un vector de tip fereastra.

typedef struct{unsigned int W, H, dim\_img, dim\_padding, numar\_pixeli; unsigned char header[54];pixel\* vector;double deviatie\_standard;double medie;} info\_sablon;

* Defineste tipul de date info\_sablon, retine detalii legate de un sablon: inaltimea(H), latimea(W), dimensiunea imaginii (dim\_img), deviatia standard (deviatie\_standard) calculata ulterior pentru aflarea corelatiei, medie calculate ulterior pentru aflarea corelatiei.

imagine grayscale ( imagine img)

* Modifica valoarea pixelilor transformand imaginea intr-una grayscale

info\_sablon grayscale\_sablon ( info\_sablon img)

* Funcția încarcă în memoria internă imaginea de la adresa specificată.
* Se deschide fișierul aflat la adresa nume\_fisier prentru citire; în cazul unei erori la deschidere se afișează un mesaj sugestiv și se returnează valoarea 1.
* Se obțin dimensiunile (width, height) imaginii pentru a se calcula padding-ul imaginii și respectiv numărul de pixeli
* Se aloca memorie pentru tabloul “vector”
* Liniarizarea matricei, se parcurge imaginea pixel cu pixel și se copiază pixelul curent în tabloul \*vector, pe poziția corespunzătoare - se ține cont de ordinea inversată a liniilor.
* Se calculeaza deviatia si media sablonului

double medie\_fereastra (imagine frst, info\_sablon sablon)

* Calculeaza si returneaza media valorilor pixelilor din fereastra

double deviatie\_fereastra ( imagine img, info\_sablon sablon)

* Calculeaza deviatia ferestrei conform formulei din pdf

double corr(info\_sablon sablon, imagine frst)

* Calculeaza corelatia dintre sablon si imaginea conform formulei din pdf

vector\_cu\_ferestre repartizare(imagine img\_principala, info\_sablon sablon)

* Se calculeaza corelatia pentru fiecare fereastra si se salveaza detectiile in tabloul vector de tip vector\_cu\_ferestre
* Se returneaza vectorul cu detectii

pixel get\_culoare(int k)

* Functia get\_culaore are ca paramentru un numar intreg care reprezinta numarul sablonului
* In functie de numarul sablonului in switch se returneaza culoarea reprezentativa

imagine colorare\_img (imagine imagine\_principala, info\_sablon sablon, vector\_cu\_ferestre \*vector)

* Parcurge imaginea principala si coloreaza conturul ferestrei in functie de numarul sablonulu
* Returneaza imaginea colorata

int cmp\_detectii (const void \*a, const void \*b)

* Functia de compararea specifica qsort-ului care ordoneaza vectorul cu ferestre descrescaor in functie de valoarea corelatiei

double max (double a, double b)

* Calculeaza maximul dintre doua numere

double min (double a, double b)

* Calculeaza minimul dintre doua numere

int suprapunere (fereastra f1, fereastra f2)

* Calculeaza aria celor doua ferestre primate ca parametru
* Calculeaza aria intersectiei dintre cele doua ferestre
* Calculeaza ratia conform formulei din pdf
* Daca ratia > 0.2 returneaza valoarea 1, altfel 0

void eliminare\_suprapuneri ( vector\_cu\_ferestre \*vector )

* Ordoneaza cu ajutorul functiei qsort tabloul vector primit ca parametru
* Parcurge vectorul cu ferestre si verifica daca functia de suprapunere returneaza 1 sau 0
* Daca returneaza 1 se sterge din vector fereastra care are aria mai mica