Explicarea functiilor folosite in cadrul proiectului

Encriptare/Decriptare:

1.void xorshift(unsigned int seed,unsigned int n,unsigned int *randomArray)

Functia xorshift creeaza un vector de “n” elemente pseudo-random,pe baza unui seed.

2. void perm(unsigned int n,unsigned int *p,unsigned int *randomArray)

Functia perm foloseste algoritmul lui Durstenfeld pentru a genera o permutare aleatoare de lungime “n” in vectorul “p”,pentru alegerea unui numar aleator intre 0 si k se foloseste un vector de elemente aleatoare generate de xorshift si “%(k+1)” pentru a asigura faptul ca numarul este mai mic sau egal cu k.

3. void bmpTOarray(const char *numeIMG,unsigned char **header,RGB **V)

Functia bmpTOarray construieste prin referinta forma liniarizata a imaginii si header-ul,valorile sunt puse in vector in asa fel incat pixelii imaginii sa fie reprezentati de la stanga la dreapta,de sus in jos din cauza modului in care functioneaza formatul bitmap.Padding-ul este calculat,iar octetii de padding sunt omisi la punerea pixelilor in vector.

4. void arrayTObmp(char *numeIMG,unsigned char **header,RGB **V)

Functia arrayTObmp primeste prin referinta forma liniarizata a imaginii si header-ul,calculeaza padding-ul si construieste imaginea bitmap.

5. unsigned int RGBtoINT(RGB a)

Functia RGBtoINT primeste o valoare de tip RGB “a” si o reprezinta drept un numar de tip int,in care cel mai semnificativ octet are valoarea 0 iar ceilalti 3 octeti reprezinta cate o valoare a fiecarui pixel,acest numar este returnat.

6. RGB INTtoRGB(unsigned int a)

Functia INTtoRGB primeste o valoarea de tip int “a” si o transforma intr-o valoare de tip RGB pe baza a celor mai nesimnificativi 3 octeti.

7. void criptare(unsigned char **header,RGB **V,const char *cheie)

Functia criptare primeste ca parametru header-ul,forma liniarizata a unei imagini si numele fisierului text care contine cheia secreta.Se genereaza un vector cu 2*latime*inaltime numere pseudo-random folosind xorshift,un vector cu permutarea aleatoare folosind functia perm si un vector identic cu forma liniarizata a imaginii pentru a fi folosit la permutarea pixelilor.Dupa permutarea pixelilor,pentru fiecare pixel se foloseste operatia xor intre pixelul anterior asupra caruia s-a aplicat aceasta operatie si un numar aleator generat de xorshift.Pentru primul pixel se va folosi valoarea SV drept pixel anterior.

8. void decriptare(unsigned char **header,RGB **V,const char *cheie)

Functia decriptare primeste ca parametru header-ul,forma liniarizata a unei imagini si numele fisierului text care contine cheia secreta.Se genereaza un vector cu 2*latime*inaltime numere pseudo-random folosind xorshift,un vector cu permutarea aleatoare folosind functia perm si un vector identic cu forma liniarizata a imaginii pentru a fi folosit la permutarea pixelilor.Se aplica operatia xor asupra pixelilor,se genereaza permutarea inversa,iar pixelii sunt permutati conform acestei permutari inverse.

9. void chiTest (unsigned char **header,RGB **V)

Functia chiTest primeste ca parametru header-ul si forma liniarizata a unei imagini.Este calculat prima data frecventa estimata teoretic a oricarei valori i,dupa care sunt generati vectorii de frecventa pentru fiecare canal de culoare.Pentru fiecare canal de culoare este aplicata formula matematica si este afisat rezultatul.

Template matching:

10. void bmpTOmatrix(const char *numeIMG,unsigned char **header,RGB ***V)

Functia bmpTOmatrix construieste prin referinta forma liniarizata bidimensional a imaginii si header-ul,valorile sunt puse in matrice in asa fel incat coordonatele pixelilor din imagine sa fie echivalente cu coordonatele din matrice.Padding-ul este calculat,iar octetii de padding sunt omisi la punerea pixelilor in matrice.

11. void matrixTObmp(char *numeIMG,unsigned char **header,RGB ***V)

Functia matrixTObmp primeste prin referinta forma liniarizata bidimensional a imaginii si header-ul,calculeaza padding-ul si construieste imaginea bitmap.

12. void contur(RGB ***M,fereastra F,RGB C)

Functia contur primeste ca parametru forma liniarizata bidimensional a imaginii,o variabila de tip “fereastra” F care contine coordonatele coltului din stanga sus al ferestreii al carui contur trebuie colorat si culoarea “C”.

13. double xBar(RGB **S,int x,int y)

Functia xBar calculeaza media intensitatilor grayscale a pixelilor din fereastra cu coltul din stanga sus situat la coordonatele (x,y).

14. double xSig(RGB **S,int x,int y,double sBar)

Functia xSig calculeaza deviatia standard a valorilor intensitatilor grayscale a pixelilor din fereastra cu coltul din stanga sus situat la coordonatele (x,y).

15. double xDif(RGB **S,RGB **F,int x,int y,double fBar,double sBar)

Functia xDif calculeaza suma (f I (i,j) – f I barat) *(S(i,j) – S barat) pentru fiecare i si j care apartine sablonului S.

16. int cmp(const void *a,const void *b)

Functia cmp compara 2 variabile de tip “fereastra” si returneaza valoarea -1 daca corelatia primei ferestre este mai mare decat corelatia celei de-a doua,si valoarea 1 in caz contrar.

17. fereastra * templateMatching(unsigned char **header,RGB ***M,RGB ***S,double ps)

Functia templateMatching primeste ca parametru header-ul, forma liniarizata bidimensional a imaginii si un prag ps si returneaza un vector de elemente de tip “fereastra”.Prima valoarea a vectorului ce urmeaza sa fie returnat retine marimea acestui vector,iar valoarea corelatiei sale este 2 pentru a asigura ca va ramane pe pozitia 0 dupa sortare.Se aplica functiile xBar,xSig si xDif pentru a calcula corelatia fiecarei ferestre cu sablonul ales,iar daca valoarea corelatiei este mai mare sau egala cu ps,coordonatele si corelatia acestei detectii este retinuta in vectorul “D”.

18. int suprapunere(fereastra A,fereastra B)

Functia suprapunere calculeaza numarul de pixeli pe care 2 ferestre il au in comun si calculeaza suprapunerea spatiala dintre aceste 2 ferestre,daca aceasta valoare este peste 0.2 functia returneaza 1,in caz contrar 0.

19. void elimNONmax(fereastra **D)

Functia elimNONmax proceseaza un tablou D sortat in ordine descrescatoare a corelatiei de detectii de tip “fereastra”.Daca 2 detectii nenule se suprapun,detectia din dreapta va primi valoarea corelatiei 0.Tabloul se modifica astfel incat contine doar detectile cu o valoare a corelatiei nenula,iar valoarea contorului de pe prima pozitie se modifica.

20. void deallocate_mem(int*** arr, int n)

Functia deallocate_mem elibereaza memoria ocupata de un tablou bidimensional,cu un numar de “n” linii, alocat dinamic prin referinta.