

Facultatea de matematica si informatica
Bucuresti

Proiect
Programare Procedurala
An I
Semestrul I

Iosif Gabriel, 143

Functia de citire – ReadBmpImg

- Primeste ca parametru calea unei imagini bmp ce urmeaza sa fie citite, si o liniarizeaza intr-un vector.
- Deoarece imaginile bmp pot contine padding, daca latimea acesteia nu este multiplu de 4, in liniarizare nu apar si octetii de 0, din padding, ci doar pixelii.
- Salveaza in primele 54 pozitii din vector, headerul imaginii.
- Returneaza un pointer catre un tablou de memorie din memoria interna ce contine vectorul liniarizat.

Functia de scriere – WriteBmpImg

- Primeste ca parametru un vector liniarizat si calea imaginii unde trebuie salvata.
- Deoarece in header se afla lungimea si latimea, acestea se acceseaza si daca latimea nu este multiplu de 4, la scriere se va adauga padding-ul necesar pentru o formatare corecta.

Main()

Pentru criptare:

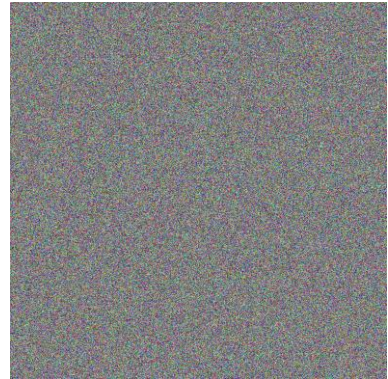
- In interiorul programului **main** se citeste calea imaginii initiale, cea care urmeaza sa fie criptata si se va salva in memoria externa, calea fisierului ce contine cheia secreta si calea imaginii decriptate, care de asemenea, se va salva in memoria externa din fisierul **paths.txt** .
- Se apeleaza functia de criptare folosind calea imaginii initiale, cea criptata si cheia secreta.
- Apoi, decriptarea prin parametrii de calea imaginii criptate, calea imaginii decriptate si cheia secreta.

Pentru template matching:

- Se citesc, pe rand caile celor 10 sabloane cu cifre de calculator si a imaginii initiale, care contine cifre scrise de mana.
- Culoarele pentru fiecare sablon sunt atribuite in program.
- Se declara static o structura ce contine date despre ferestrele detectate de functia template matching, si in functie de numarul acestora gasite, se declara dinamic o alta structura, iar prima se elibereaza.
- Se aplica **qsort** pentru a sorta ferestrele in functie de gradul de corelatie (cat de bine se pliaza un sablon peste o cifra de mana)
- Se apeleaza **ElimNonMaxime** pentru a pastra doar ferestrele cu corelatia cea mai mare, si care nu se suprapun.
- Se coloreaza cu **colorare** ferestrele ramase dupa eliminarea non maximelor.

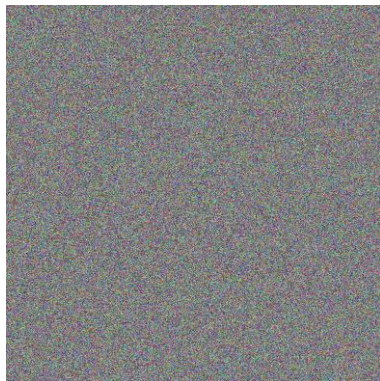
Criptarea unei imagini BMP

- S-a liniarizat imaginea intr-un vector unidimensional, cu observatia ca liniile intr-un format BMP sunt stocate incepand cu ultima, spre prima, folosind functia `ReadBmpImg` ce returneaza un pointer catre zona de memorie a liniarizarii.
- S-au generat un numar de $2 * \text{Latime} * \text{Inaltime}(\text{imagine}) - 1$ elemente, pornind de la o cheie secreta, conform algoritmului XORSHIFT32, propus de George Marsaglia, prin functia cu acelasi nume ce returneaza un pointer catre tabloul cu elemente.
- Se creeaza o permutare de $\text{Latime} * \text{Inaltime}(\text{imagine}) - 1$ elemente, iar indicele este un numar pseudoaleator.
- Se permute pixelii imaginii conform permutarii generate.
- Se aplica substitutii la nivel de octet, incepand cu SV, al doilea numar din fisierul ce contine cheia secreta, urmatoare valori se vor substitui cu valorile anteroare. Se xoreaza doar culorile asemenea, iar numerele intregi sunt “sparte” in bytes, cel mai putin semnificativ folosindu-se pentru albastru, al doilea pentru verde iar al treilea pentru canalul rosu.
- La final se salveaza in memoria externa imaginea criptata conform parametrului al doilea al functiei si se elibereaza toata memoria alocata dinamic.



Decriptarea unei imagini BMP

- Din memoria externa si preia imaginea criptata anterior, prin intermediul functiei ReadBmpImg.
- Deoarece decriptarea este complementara criptarii, se va genera, de asemenea, un numar de $2 * \text{Latime} * \text{Inaltime}(\text{imagine}) - 1$ numere pseudoaleatoare, o permutare, si inversa ei, allocate dinamic.
- Se va citi din fisierul text, seed-ul si SV pentru generarea acelorasi numere de la criptare.
- Mai intai se va xora intr-o matrice liniarizata auxiliara MatLinCryptPrim, conform canalelor cu SV, matricea liniarizata criptata si numarul generat prin XORSHIFT32, “spart” in bytes.
- Imaginea decriptata va fi o alta matrice linianara, care primeste pixelii permutati conform permutarii inverse generate mai devreme.
- Se elibereaza toata memoria folosita in functie si se va inchide fisierul ce contine seed-ul si SV.
- Se va scrie cu WriteBmpImg in memoria externa, matricea decriptata.



Testul Chi-Squared

- ❖ Primește ca parametru o imagine liniarizată
- ❖ Se inițializează o medie a tuturor culorilor, numită **ficubara**
- ❖ Se va calcula pentru fiecare culoare de la 0 la 255, de câte ori apare în întreaga imagine și se va stoca în frecvrosu, frecvverde, frecvalbastru.
- ❖ Afisează pe ecran câte o histogramă pentru fiecare canal de culoare

Pattern Matching

- In **main** se citeste calea fiecarui sablon si este declarata o structura de dimensiune maxima ce va retine coordonatele de inceput ale unei ferestre (x si y) , culoarea (distincta pentru fiecare sablon) si ok ce va lua valoarea 0(nu afisez) sau 1(afisez) in functie de suprapuneri.
- Tot in **main** am alocat memorie pentru a transforma imaginea initiala in matrice, pentru functia de colorare, care mai apoi se va rescrie peste vectorul liniarizat si salvat in memoria externa.
- Se apeleaza functia de template matching pentru fiecare sablon.
- Atat sablonul cat si imaginea initiala, sunt transformate in imagini gri, prin functia grayscale care transforma imaginea color in una alb-negru.
- Tablourile unidimensionale sunt transformate in matrice, pentru a lucra mai usor.
- Se afla corelatia fiecărei ferestre pe cate un sablon si se salveaza in vectorul de structuri Fereastră pe pozitia de contor, care se incrementeaza cu fiecare salvare, daca trece de pragul 0.5 (o potrivire cat de cat buna).
- Se sorteaza cu qsort in ordine descrescatoare a corelatiei toate detectiile obtinute
- Se elimina toate suprapunerile in functia de **ElimNonMaxime**, dand valoarea Fereastră[j].ok=0 fiecărei ferestre care se suprapune si are corelatie mai mica.
- Dupa eliminarea nonmaximelor, creez un vector auxiliar de structuri cu dimensiunea de contor si eliberez memoria inefficient utilizata prin vectorul Fereastră.

- Incep colorarea tuturor ferestrelor ramase, in matricea imaginii alocata in main, pe care apoi o transform in tablou unidimensional si o scriu in memoria externa conforma cu calea salvata.

3	9	3	3	7	3	4	5	5	4	8	3	2	2	0	7	3	5	9	0	6	3	6	0	4
5	4	3	3	7	1	0	9	9	6	4	2	5	3	8	1	2	5	9	0	7	3	0	8	2
2	2	3	2	6	1	1	3	0	2	2	5	3	8	1	6	1	2	3	4	0	6	2	9	1
2	2	8	0	3	8	1	3	3	7	8	1	6	2	5	6	7	0	6	5	0	4	1	6	0
9	3	9	0	8	4	7	6	6	9	7	3	8	1	5	6	0	1	8	0	7	8	3	0	6
5	3	6	4	6	0	6	9	3	8	1	5	0	7	7	6	0	1	6	9	1	8	3	0	6
4	2	6	5	7	2	1	4	1	7	1	2	1	1	0	7	8	4	2	1	1	8	7	6	1
3	8	1	2	0	4	4	8	0	9	0	3	7	6	1	6	4	1	5	1	8	9	4	0	2
1	4	2	7	4	6	3	0	4	2	0	4	9	7	7	2	1	0	4	5	1	2	0	8	0
4	8	8	2	1	7	7	3	6	6	7	7	4	7	3	4	7	3	9	1	8	5	3	1	0
0	0	7	2	6	1	9	2	7	4	1	2	3	6	7	3	1	0	3	7	5	1	0	4	3
3	6	4	0	6	0	5	5	8	0	6	7	5	8	3	7	3	9	1	8	5	4	9	2	9
0	7	5	4	2	9	8	5	8	0	6	2	6	8	3	8	7	3	9	9	6	4	9	7	7
5	9	7	9	4	5	1	6	7	2	6	2	6	8	3	1	8	7	3	9	9	7	9	7	7
9	9	8	4	4	8	2	2	7	4	8	8	8	1	2	5	5	7	8	7	1	2	0	4	8
7	6	9	3	5	7	5	7	6	3	4	0	2	1	6	9	4	2	8	9	3	7	1	5	5
8	3	3	8	9	6	2	0	3	8	6	1	6	9	9	3	1	6	2	5	3	7	1	5	5
7	5	7	0	4	3	0	9	9	3	5	2	4	6	5	0	4	2	3	1	4	5	0	5	0
1	2	2	3	9	1	7	7	8	5	4	9	0	7	4	0	3	5	0	7	2	7	1	2	0
8	8	6	1	1	6	6	4	4	0	8	5	8	2	2	4	3	3	9	4	2	6	2	9	2

Fara eliminarea
nonmaximelor



5	9	3	3	7	1	0	9	9	6	4	2	5	3	8	1	2	5	9	0	6	3	6	0	4
5	4	3	3	7	1	0	9	9	6	4	2	5	3	8	1	2	5	9	0	6	3	6	0	4
2	2	3	2	6	1	1	3	0	2	2	5	3	8	1	6	1	2	3	4	0	6	2	9	1
2	2	8	0	3	8	1	3	3	7	8	1	6	2	5	6	7	0	6	5	0	4	1	6	0
9	3	9	0	8	4	7	6	6	9	7	3	8	1	5	6	0	1	8	0	7	8	3	0	6
5	3	6	4	6	0	6	9	3	8	1	5	0	7	7	6	0	1	6	9	1	8	3	0	6
4	2	6	5	7	2	1	4	1	7	1	2	1	1	0	7	8	4	2	1	1	8	7	6	1
3	8	1	2	0	4	4	8	0	9	0	3	7	6	1	6	4	1	5	1	8	9	4	0	2
1	4	2	7	4	6	3	0	4	2	0	4	9	7	7	2	1	0	4	5	1	2	0	8	0
4	8	8	2	1	7	7	3	6	6	7	7	4	7	3	4	7	3	9	1	8	5	3	1	0
0	0	7	2	6	1	9	2	7	4	1	2	3	6	7	3	1	0	3	7	5	1	0	4	3
3	6	4	0	6	0	5	5	8	0	6	7	5	8	3	7	3	9	1	8	5	4	9	2	9
0	7	5	4	2	9	8	5	8	0	6	2	6	8	3	1	8	7	3	9	9	6	4	9	7
5	9	7	9	4	5	1	6	7	2	6	2	6	8	3	1	8	7	3	9	9	7	9	7	7
9	9	8	4	4	8	2	2	7	4	8	8	8	1	2	5	5	7	8	7	1	2	0	4	8
7	6	9	3	5	7	5	7	6	3	4	0	2	1	6	9	4	2	8	9	3	7	1	5	5
8	3	3	8	9	6	2	0	3	8	6	1	6	9	9	3	1	6	2	5	3	7	1	5	5
7	5	7	0	4	3	0	9	9	3	5	2	4	6	5	0	4	2	3	1	4	5	0	5	0
1	2	2	3	9	1	7	7	8	5	4	9	0	7	4	0	3	5	0	7	2	7	1	2	0
8	8	6	1	1	6	6	4	4	0	8	5	8	2	2	4	3	3	9	4	2	6	2	9	2

Cu eliminarea
nonmaximelor



5	9	3	3	7	1	0	9	9	6	4	2	5	3	8	1	2	5	9	0	6	3	6	0	4
5	4	3	3	7	1	0	9	9	6	4	2	5	3	8	1	2	5	9	0	6	3	6	0	4
2	2	3	2	6	1	1	3	0	2	2	5	3	8	1	6	1	2	3	4	0	6	2	9	1
2	2	8	0	3	8	1	3	3	7	8	1	6	2	5	6	7	0	6	5	0	4	1	6	0
9	3	9	0	8	4	7	6	6	9	7	3	8	1	5	6	0	1	8	0	7	8	3	0	6
5	3	6	4	6	0	6	9	3	8	1	5	0	7	7	6	0	1	6	9	1	8	3	0	6
4	2	6	5	7	2	1	4	1	7	1	2	1	1	0	7	8	4	2	1	1	8	7	6	1
3	8	1	2	0	4	4	8	0	9	0	3	7	6	1	6	4	1	5	1	8	9	4	0	2
1	4	2	7	4	6	3	0	4	2	0	4	9	7	7	2	1	0	4	5	1	2	0	8	0
4	8	8	2	1	7	7	3	6	6	7	7	4	7	3	4	7	3	9	1	8	5	3	1	0
0	0	7	2	6	1	9	2	7	4	1	2	3	6	7	3	1	0	3	7	5	1	0	4	3
3	6	4	0	6	0	5	5	8	0	6	7	5	8	3	7	3	9	1	8	5	4	9	2	9
0	7	5	4	2	9	8	5	8	0	6	2	6	8	3	1	8	7	3	9	9	6	4	9	7
5	9	7	9	4	5	1	6	7	2	6	2	6	8	3	1	8	7	3	9	9	7	9	7	7
9	9	8	4	4	8	2	2	7	4	8	8	8	1	2	5	5	7	8	7	1	2	0	4	8
7	6	9	3	5	7	5	7	6	3	4	0	2	1	6	9	4	2	8	9	3	7	1	5	5
8	3	3	8	9	6	2	0	3	8	6	1	6	9	9	3	1	6	2	5	3	7	1	5	5
7	5	7	0	4	3	0	9	9	3	5	2	4	6	5	0	4	2	3	1	4	5	0	5	0
1	2	2	3	9	1	7	7	8	5	4	9	0	7	4	0	3	5	0	7	2	7	1	2	0
8	8	6	1	1	6	6	4	4	0	8	5	8	2	2	4	3	3	9	4	2	6	2	9	2