Proiect la Programare Procedurală

Seria 13, grupa 133, Panţiru Luana-Cătălina

În realizarea proiectului am folosit mai multe funcţii ajutătoare pe lângă cele cerute în enunţul proiectului. În cele ce urmează o să explic fiecare funcţie în parte.

Modulul de criptare/decriptare

* Funcţia header: în această funcţie se realizează aflarea înălţimii şi lăţimii unei imagini, cât şi memorarea într-un vector a primilor 54 de octeţi ai imaginii.
* Funcţia padding: în această funcţie se calculează padding-ul care trebuie luat în considerare la citirea şi afişarea imaginii în cazul în care lăţimea unei imagini nu este multiplu de 4.
* Funcţia xorshift32: în această funcţie se realizează determinarea unui număr random pornind de la un seed. Această funcţie o apelez în alta funcţie pentru a forma un vector de mai multe numere random, seed-ul pentru un nou număr random fiind elementul anterior din vector.
* Funcţia liniarizare: în această funcţie se citeşte o imagine şi se pune conţinutul ei într-un vector. Vectorul meu (L) este un vector de structuri, fiecare poziţie a vectorului memorând 3 valori, cele 3 canale de culoare, structura folosită fiind RGB. Am folosit două for-uri,unul pentru înălţime şi unul pentru lăţime pentru a îmi da seama mai uşor dacă parcurg imaginea corect şi să ştiu când să sar peste padding dacă îl întâlnesc. Primul element din vector reprezintă pixelul din stânga jos al imaginii.
* Funcţia afisare: în această funcţie are loc afişarea imaginii. Această imagine este sub formă liniarizată, memorată într-un vector. La afişarea imaginii se scrie header-ul, conţinutul vectorului şi padding-ul de fiecare dacă când se termină de afişat o linie a imaginii.
* Funcţia swap: funcţie folosită pentru a inversa valorile a două variabile.
* Funcţia criptare: în această funcţie are loc criptarea imaginii. Întai se realizează un vector de numere random folosind funcţia xorshift32 explicată mai sus. Seed-ul de la care se pleacă se citeşte din fişierul text dat ca parametru, fiind memorat în primul element din vector, în cazul meu, în R[0]. Apoi se realizează modul de permutare ,aici apelându-se la funcţia swap prezentată anterior, în cazul meu fiind vectorul v. După se permută pixelii după modul de permutare aflat, realizându-se alstfel imaginea intermediară (L1). În final se realizează imaginea criptată folosind formulele explicate în enunţul problemei (C).
* Funcţia decriptare: în această funcţie are loc decriptarea unei imaginii. Decriptarea se realizează folosind aceeaşi paşi ca la criptare, diferă doar modul în care aplicăm formulele,acest lucru fiind explicat şi în enunţul proiectului. Se calculează numerele random, se face vectorulu de permutări, se realizează imaginea intermediară (C1) după formulele date, iar apoi se fac permutările în funcţie de modul de permutare rezultat anterior, determinându-se decriptarea (D).
* Funcţia chi\_patrat: în această funcţie se calculează uniformitatea distribuţiei valorilor pixelilor. Se află întâi frecvenţa fiecărui canal de culoare folosind un vector de apariţii, în cazul meu fR pentru roşu, fG pentru verde şi fB pentru albastru. Apoi se calculează x2 pentru fiecare canal de culoare după formula dată (x2R e chi-pătrat pentru roşu, x2G chi-pătrat pentru verde şi x2B chi-pătrat pentru albastru).

Modulul de recunoaştere de pattern-uri într-o imagine

* Funcţia matrice: în această funcţie se citeşte imaginea şi se memorează într-o matrice , iar padding-ul nu se reţine în matrice. Am folosit de această dată imaginea memorată sub formă de matrice deoarece glisarea şabloanelor, memorate şi ele într-o matrice, poate fi vizualizată mai uşor.
* Funcţia afisare\_matrice: această funcţie se apelează pentru a se afişa imaginea memorată într-o matrice. Aici se afişează şi header-ul şi padding-ul.
* Funcţia grayscale: în această funcţie se realizează tranformarea în tonuri de gri a unei imagini color, imaginea fiind transmisă sub forma unei matrici.
* Funcţia med\_intensitatii: această funcţie este folosită pentru a se calcula media valorilor intensităţilor grayscale a pixelilor într-o fereastră sau într-un şablon. Se adună toate valorile uneia dintre culori , roşu, verde şi albastru având aceeaşi intensitate într-o imagine gri, apoi se împarte la produsul dintre înalţimea şi lăţimea ferestrei/şablonului.
* Funcţia deviatia\_stan: aici se realizează deviaţia standard a valorilor intensităţilor grayscale a unei imagini, trimisă ca parametru sub forma unei matrici alături de dimensiunile acesteia, lăţimea şi înălţimea imaginii. Aceasta se realizează folosind formula dată în enunţ şi apelându-se şi la funcţia med\_intensitatii.
* Funcţia colorare: în această funcţie se realizează desenarea marginilor unei ferestre. Fereastra se transmite prin coordonatele colţului din stânga sus şi dimensiunile acesteia şi matricea imaginii mari pentru a putea fi modificată, dar şi cele trei canale de culoare pentru a se putea forma culoarea cu care se trasează. Întâi se realizează desenarea marginilor de sus şi jos, apoi cele laterale.
* Structura detectie: această structură este realizată pentru ca vectorul de detecţii să reţină coordonatele colţului din stânga sus al ferestrei respective, corelaţia acesteia, dar şi valorile canalelor de culoare pentru a şti cum să fie desenate marginile.
* Funcţia template\_matching: în această funcţie se realizează corelaţiile dinte imagine şi un anumit şablon. Atât imaginea cât şi şablonul se transformă în tonuri de gri. Apoi se alocă memorie unui vector, D, acesta reprezentând vectorul de detecţii pentru o anumită cifră. Acest vector este un vector de structuri de tip detecţie. Se apelează funcţia deviatia\_stan pentru a se calcula deviaţia standart a şablonui (v\_s), apoi se apelează funcţia med\_intensitatii pentru a se calcula media intensităţilor pixelilor din şablon (v\_s1). După se glisează şablonul pe întreaga imagine şi în fiecare poziţie se calculează corelaţia şi se reţine în vectorul D fereastra cu corelaţia mai mare sau egală decât pragul dat. În momentul glisării şablonului se verifică dacă nu se depăşesc dimensiunile imaginii. Această funcţie este apelată pentru fiecare şablon în parte, furnizând de fiecare dată un nou vector.
* Funcţia culoare: această funcţie este apelată după formarea unui vector de detecţie pentru a şti cu ce culoare trebuie desenată marginea ferestrei.
* Funcţia copie: această funcţie realizează copierea unui vector în alt vector. Apelarea acestei funcţii este folosită pentru a alipi toate detecţiile găsite pentru fiecare şablon într-un singur vector de tip detecţie.
* Funcţia cmp: această funcţie este folosită pentru a transmite modul în care se realizează sortarea vectorului la apelarea funcţiei qsort. În acest caz se vor ordona elementele în ordine descrescătoare.
* Funcţia arie: această funcţie este folosită pentru calcularea ariei unei ferestre, a unui şablon sau a intersecţiei dintre şablon şi fereastră.
* Funcţia nonmaxime: această funcţie are rolul de a elimina detecţiile care au o suprapunere spaţială mai mare decât 0,2. Se parcurge vectorul şi se iau câte două detecţii şi se verifică întâi dacă există intersecţie între ferestrele pe care le definesc. Apoi se calculează suprapunerea spaţială dintre acestea facând apel la funcţia arie de câte ori e nevoie. Primul caz reprezintă cel dat în enunţ ca exemplu, al doilea caz e asemănător ca primul, dar dj se află în partea de sus şi di în partea de jos. Al treilea caz reprezintă momentul în care intersecţia este în partea din stânga, iar cazul al patrulea reprezintă aceeaşi intersecţie dar dj e în partea de sus şi di în partea de jos. Mai jos v-am desenat fiecare caz în parte pentru o întelegere mai bună.

Cazul 1 Cazul 2

dj

di

diꓵdj

diꓵdj

di

dj

Cazul 3 Cazul 4

dj

di

diꓵdj

diꓵdj

dj

di