LABORATOR 3

1. Calculați histograma (într-un vector de întregi de dimensiune 256) pentru o imagine

grayscale (cu 8 biți/pixel).

//initializam valorile vectorului din histograma la 0

for (int i = 0; i < 255; i++) {

hist[i] = 0;

}

//parcurgem matricea de pixeli din imagine

for (int i = 0; i < height; i++){

for (int j = 0; j < width; j++){

//pentru fiecare valoare gasita

//se va incrementa valoarea elementului din vector

hist[src.at<uchar>(i, j)]++;

}

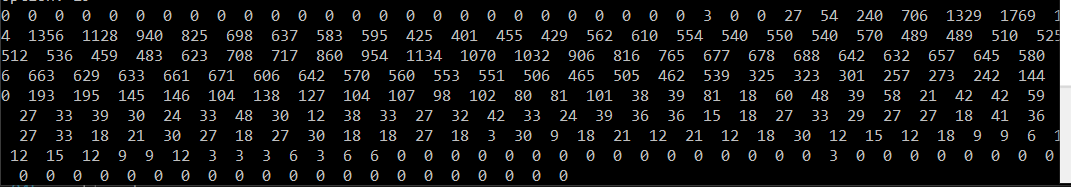
}

//afisam sirul de elemente

for (int i = 0; i < 255; i++) {

printf("%d ", hist[i]);

}



2. Calculați FDP (într-un vector de tip float de dimensiune 256).

3. Afișați histograma calculată utilizând funcția din laborator.

4. Calculați histograma folosind un număr redus de m ≤ 256 acumulatoare.

5. Implementați algoritmul de reducere a nivelurilor de gri (praguri multiple) de la 3.3.

6. Îmbunătățiți algoritmul de reducere a nivelurilor de gri (praguri multiple) utilizând

distribuirea erorii cu algoritmul Floyd-Steinberg de la 3.4.

7. Realizați algoritmul de reducere a nivelurilor de gri pe canalul Hue din spațiul de culoare

HSV al unei imagini color. Modificați doar valorile din canalul H, păstrând canalele S și V

neschimbate. O altă opțiune este setarea lor (S și V) la valoarea maximă permisă. Pentru

vizualizare transformați înapoi în spațiul RGB.