

UNIVERSITATEA POLITEHNICĂ DIN BUCUREȘTI

FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE

DEPARTAMENTUL DE AUTOMATICĂ ȘI INGINERIA SISTEMELOR

PROIECT AWJ

Tema 12

Vîja Vlad-Alexandru

Grupa 331AB

Cerinta

Am avut de implementat funcționalitatea pentru tema 12, care consta in crearea unei imagini care sa conțină 11 niveluri de gri – toți pixelii care au nivelul de gri +/- 5 fata de nivelul de gri al maximului histogramei.

Pentru aceasta functionalitate a fost nevoie de obtinerea nivelurilor de luminotitate a pixelilor din imaginea greyscale, aflat valoarea de luminozitate cu aparitia maxima si modificat imaginea in asa fel incat sa ramana doar pixeli care raspund cerintei, ceilalti fiind facuti albi.

Clasele utilizate in implementarea acestei teme

Pentru implementarea cerintelor de procesare de imagine si de multi-threading am folosit urmatoarele clase:

1. TestPackage
 - 1.1. Test
2. MethodsPackage
 - 2.1. Buffer
 - 2.2. Producer
 - 2.3. Consumer
 - 2.4. IntensityInterface
 - 2.5. Intensity
 - 2.6. PictureToArray
 - 2.7. Histogram

Utilizarea claselor in implementare

1. **Testare** – clasa Test

In aceasta clasa preiau locatia imaginii ce urmeaza sa fie procesata si locatia unde trebuie salvata imaginea, pornesc thread-urile de productie si consum, iar apoi prelucrez imaginea care urmeaza sa fie salvata la locatia data.
2. **Multi-threading** – clasele Buffer, Producer si Consumer
3. **Interfata** – clasa IntensityInterface

Clasa interfata cu metoda interfata Intensity.
4. **Mostenirea pe 3 nivele** – clasele Intensity, PictureToArray, Histogram

Descrierea aplicatiei si a algoritmului

Aplicatia primeste locatia imaginii greyscale, care este un fisier de format bmp, caruia ii aplica algoritmul pentru obtinerea histogramei si are ca output tot un fisier bmp care contine imaginea doar cu nivelurile de gri corespunzatoare.

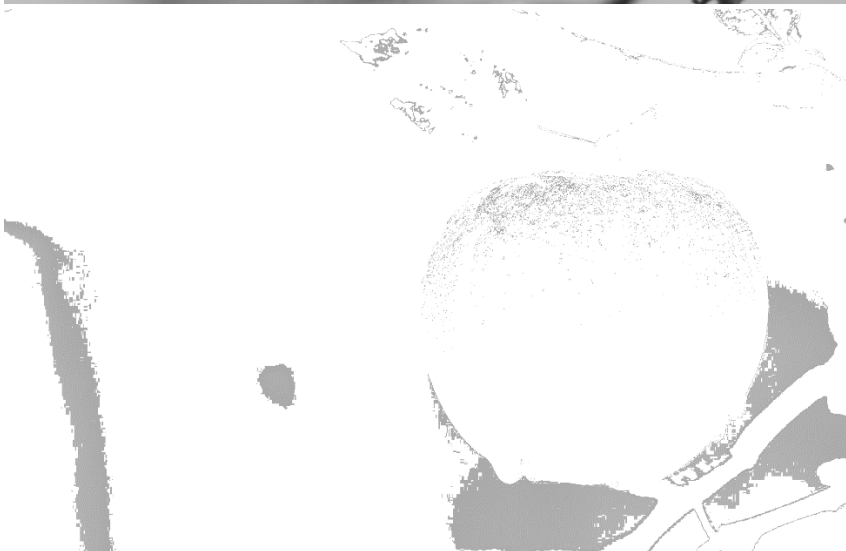
Algoritmul pentru obtinerea histogramei este nevoit mai intai sa obtina nivelul de luminozitate al fiecarui pixel. Acest lucru este efectuat prin aplicarea unei formule care se foloseste de valorile de pe canalele de rosu, verde si albastru ale imaginii($0.299*r + 0.587*g + 0.114*b$).

Folosindu-ma de aceasta formula creez o matrice de niveluri de gri corespunzatoare fiecarui pixel din matricea de imagine.

De aici aflu de cate ori a fost intalnit fiecare nivel de gri si si dupa aceea aflu care nivel de gri a fost cel mai intanit.

Partea finala a algoritmului este de a lasa in imagine doar pixelii care au nivelul de gri cu +/-5 fata nivelul cel mai intalnit de gri din imagine(nivelul maxim al histogramei), iar restul sa fie facuti albi.

Exemplificarea functionalitatii algoritmului



Valorile histogramei rezultate si valoarea maxima a acesteia

0 - 3	50 - 14745	100 - 24911	150 - 13836	200 - 672
1 - 2	51 - 7276	101 - 12517	151 - 13505	201 - 1218
2 - 0	52 - 0	102 - 0	152 - 12898	202 - 592
3 - 8	53 - 14794	103 - 26218	153 - 12475	203 - 526
4 - 0	54 - 0	104 - 0	154 - 12021	204 - 0
5 - 9	55 - 7119	105 - 13233	155 - 26236	205 - 527
6 - 8	56 - 15787	106 - 13290	156 - 12640	206 - 482
7 - 24	57 - 0	107 - 27547	157 - 0	207 - 917
8 - 0	58 - 14711	108 - 0	158 - 12173	208 - 421
9 - 20	59 - 0	109 - 15422	159 - 12396	209 - 0
10 - 78	60 - 14462	110 - 15542	160 - 10854	210 - 400
11 - 0	61 - 0	111 - 16694	161 - 22108	211 - 326
12 - 96	62 - 7374	112 - 16930	162 - 0	212 - 324
13 - 0	63 - 16044	113 - 37716	163 - 11153	213 - 323
14 - 63	64 - 0	114 - 18790	164 - 10177	214 - 265
15 - 230	65 - 9398	115 - 0	165 - 11152	215 - 469
16 - 0	66 - 10864	116 - 20271	166 - 12676	216 - 0
17 - 182	67 - 11575	117 - 37139	167 - 12862	217 - 232
18 - 203	68 - 12298	118 - 0	168 - 34309	218 - 190
19 - 298	69 - 16452	119 - 19932	169 - 0	219 - 159
20 - 445	70 - 14989	120 - 20068	170 - 15327	220 - 194
21 - 1526	71 - 18269	121 - 41994	171 - 20526	221 - 174
22 - 0	72 - 16737	122 - 0	172 - 23499	222 - 309
23 - 968	73 - 14361	123 - 22632	173 - 18383	223 - 0
24 - 1104	74 - 15142	124 - 24720	174 - 49009	224 - 301
25 - 2273	75 - 13479	125 - 23103	175 - 19208	225 - 0
26 - 1523	76 - 14652	126 - 43787	176 - 0	226 - 122
27 - 0	77 - 25042	127 - 20731	177 - 20587	227 - 203
28 - 1682	78 - 0	128 - 0	178 - 18252	228 - 0
29 - 1856	79 - 12659	129 - 22598	179 - 25357	229 - 177
30 - 2147	80 - 24017	130 - 21666	180 - 14169	230 - 0
31 - 5264	81 - 0	131 - 24821	181 - 24662	231 - 94
32 - 0	82 - 13476	132 - 25056	182 - 8548	232 - 84
33 - 3288	83 - 13511	133 - 24637	183 - 0	233 - 68
34 - 3881	84 - 14975	134 - 25789	184 - 9891	234 - 83
35 - 4359	85 - 12467	135 - 22344	185 - 7956	235 - 19
36 - 4982	86 - 13456	136 - 22489	186 - 13540	236 - 0
37 - 5429	87 - 24924	137 - 20311	187 - 15285	237 - 25
38 - 12526	88 - 0	138 - 18348	188 - 10099	238 - 36
39 - 0	89 - 12626	139 - 17540	189 - 15696	239 - 0
40 - 7037	90 - 26934	140 - 17305	190 - 0	240 - 19
41 - 7568	91 - 0	141 - 17730	191 - 7044	241 - 0
42 - 7869	92 - 15527	142 - 33030	192 - 6999	242 - 17
43 - 16112	93 - 13768	143 - 0	193 - 4416	243 - 17
44 - 0	94 - 25749	144 - 14614	194 - 8405	244 - 0
45 - 8114	95 - 0	145 - 14711	195 - 0	245 - 6
46 - 7816	96 - 12755	146 - 16014	196 - 4548	246 - 6
47 - 7553	97 - 12175	147 - 15915	197 - 4062	247 - 13
48 - 7645	98 - 12410	148 - 31008	198 - 3044	248 - 21
49 - 7303	99 - 12568	149 - 0	199 - 1235	249 - 0
50 - 14745	100 - 24911	150 - 13836	200 - 672	250 - 34
				251 - 0
				252 - 12
				253 - 24
				254 - 0
				255 - 27
				Gri maxim: 174
				Histograma a fost salvata.

Analiza performantelor

Timpul de executie al trasteritei datelor intre Producer si Consumer.

```
Quarter produced 1  
Quarter consumed 1  
Quarter produced 2  
Quarter consumed 2  
Quarter produced 3  
Quarter consumed 3  
Quarter produced 4  
Quarter consumed 4  
Timp transmisie imagine = 4407.0
```

Timpul de executie al prelucrării de imagine.

```
Histograma a fost salvata.  
Timp procesare imagine = 239.0
```

Bibliografie

[java - Turning a grayscale image into a histogram of the intensity of the pixels of that image - Code Review Stack Exchange](#)

[Software: Grey level / Grey value | STEMMER IMAGING \(stemmer-imaging.com\)](#)

[Bitmap/Histogram - Rosetta Code](#)

[\[Java Code Sample\] Combine multiple buffered images into a single image - RyiSnow's Programming Blog](#)