

Convertirea imaginilor color în imagini monocrome utilizând metoda Luminosity

-proiect Aplicații Web cu suport Java-

Nume: Grama Alexandru

Grupa: 332AA

Introducere

Fotografia monocromă datează încă de la jumătatea secolului al XIX-lea. În ciuda apariției fotografiilor color, fotografiile alb-negru nu și-au pierdut popularitatea, fiind folosite în lumea contemporană în special în scop artistic sau în aplicațiile de imagistică tehnică. Cu atât mai mult, revoluția digitală, care a adus camerele de fotografiat digitale, a sporit popularitatea fotografiei monocrome, deoarece toate aceste dispozitive sunt capabile să realizeze fotografii alb-negru într-un mod mult mai simplu pentru utilizator (camerele de tip analogic necesitau un film special, monocromatic).

Deși popular cunoscute sub denumirea de *alb-negru*, fotografiile monocrome pot conține sute de mii de nuanțe de gri, care variază de la alb la negru. Există numeroase motive pentru care se preferă utilizarea unei imagini în tonuri de gri în defavoarea uneia color. Unul dintre acestea este complexitatea mai redusă a imaginilor monocrome din punctul de vedere al prelucrării computaționale. Deoarece în majoritatea situațiilor se dorește să păstrăm proprietățile imaginilor intacte, dar și să reducem pe cât posibil complexitatea și timpii de rulare ai algoritmilor cu care lucrăm, convertirea imaginilor color în imagini alb-negru este exact ceea ce avem nevoie în acest caz.

Descrierea aplicației

Aplicația realizată transformă o imagine color într-una alb-negru, prin metoda ponderată (weighted method), cunoscută și sub numele de metoda luminozității (luminosity method).

Proiectul este implementat în limbajul de programare *Java*, un limbaj orientat-obiect și cu o structură bazată pe clase. Acest limbaj a fost creat având în minte principiul “write once, run anywhere” (WORA), care le oferă fiabilitate și flexibilitate tuturor dezvoltatorilor de aplicații cu suport Java.

Fotografia convertită există în sistem în formatul BMP (Bitmap Picture). Acest tip de fișier este folosit în principal la stocarea de imagini, caracteristic fiind faptul că nu le comprimă, ceea ce

înseamnă că imaginile nu își pierd calitatea. Uneori, formatul *.bmp* se utilizează și pentru stocarea de date, întrucât poate înmagazina cantități mari de date fără a le comprima.

Partea teoretică

Există numeroși algoritmi concepuți pentru a converti o imagine color într-una monocrom. Deși fiecare astfel de algoritm urmează aceiași trei pași simpli, fotografiile rezultate după aplicarea diverselor metode de convertire diferă vizibil, deoarece logica matematică care stă la baza algoritmilor este diferită. Cele trei etape de bază menționate anterior sunt:

1. Obținerea valorilor roșu, verde și albastru ale pixelului.
2. Utilizarea unei formule matematice pentru obținerea, pe baza celor 3 valori, a unui singur număr care reprezintă valoarea griului rezultat.
3. Înlocuirea valorii pixelului original RGB cu cea rezultată pentru gri.

În continuare, sunt descrise cel mai des utilizate metode de conversie a imaginilor color în imagini monocrome.

Metoda *lightness* tinde să reducă contrastul. Calculează valoarea de gri a pixelului, făcând o medie între cele mai proeminente și cele mai puțin proeminente culori, după formula:

$$(\max(R, G, B) + \min(R, G, B)) / 2$$

Rezultatele generate depind de fotografia originală și sunt, în general, satisfăcătoare.

Metoda *average* (medie) este cel mai des întâlnită în programele și aplicațiile de editare foto, fiind și cel mai simplu de implementat, precum și cea mai rapidă. Formula $(R + G + B) / 3$ generează rezultate rezonabile, simplitatea ei aducând un plus atunci când vine vorba de optimizarea algoritmului. Cu toate acestea, metoda *average* nu este cea mai bună atunci când vine

vorba de reprezentarea nuanțelor de gri în raport cu felul în care oamenii percep luminozitatea. Pentru aceasta, este nevoie de ceva puțin mai complex, cum ar fi **metoda luminosity**.

Metoda luminosity, cunoscută și sub numele de **metoda weighted (ponderată)**, este o versiune mai sofisticată a metodei average. Deși folosește media, utilizează metoda ponderată de calcul a mediei luminozității pixelului gri, luând în calcul percepția umană.

Deoarece oamenii nu percep la fel de bine toate culorile, metoda mediei nu este întotdeauna conformă cu realitatea. În loc să tratăm toate cele trei culori ca fiind la fel, putem obține o mai bună conversie alb-negru dacă măsurăm, pentru fiecare dintre cele trei culori, modul și cantitatea în care sunt percepute de ochi. O formulă întâlnită în algoritmi de conversie (cu metoda ponderată) ai programelor de editare foto Photoshop sau GIMP este:

$$\text{Gray} = (\text{Red} * 0.3 + \text{Green} * 0.59 + \text{Blue} * 0.11)$$

Pentru simplificarea procedurilor și a timpilor de calcul, numerele de mai sus au fost approximate la o zecimală. Numerele exacte folosite de Y-Greyscale (YIQ/NTSC) sunt:

- Red: 0.299
- Green: 0.587
- Blue: 0.114

Aceasta este doar una dintre ponderile folosite în conversia alb-negru bazată pe metoda luminosity. Alte ponderi utilizate sunt:

- BT709 Greyscale: Red: 0.2126 Green: 0.7154 Blue: 0.0722
- RMY Greyscale: Red: 0.5 Green: 0.419 Blue: 0.081

În toate cazurile, se observă o diferență semnificativă între coeficienții celor trei culori. Nu se poate spune că unele ponderi sunt mai bune decât celelalte, diferențele dintre imaginile rezultate fiind ne semnificative și greu de perceput de către utilizatori.

Deși necesită ceva mai multe resurse de procesare, rezultatul este o imagine alb-negru mult mai dinamică.

Descriere structurală – architecturală și funcțională

În cadrul proiectului am ales să lucrez, în cazul ponderilor, cu recomandarea originală, anume BT.709 (Red: 0.2126 Green: 0.7154 Blue: 0.0722). Formula, denumită și *Luma*, este următoarea:

$$\text{Gray} = (\text{Red} * 0.2126 + \text{Green} * 0.7152 + \text{Blue} * 0.0722)$$

PSEUDOCOD:

```
for i=0 to i<w
```

```
    for j=0 to j<h
```

```
        red = value of red color at image[j][i]
```

```
        green = value of green color at image[j][i]
```

```
        blue = value of blue color at image[j][i]
```

```
        value = 0.299 * red + 0.587 * green + 0.114 * blue
```

```
        set color of pixel at image[j][i] as red=value, green=value  
        and blue = value
```

Evaluare performanțe

Din punct de vedere al rezultatelor obținute în materie de imagini monocrome, algoritmul bazat pe metoda ponderată depășește cu mult performanțele algoritmilor ce folosesc metoda mediei sau metoda “lightness”. Cu toate acestea, privind din unghiul resurselor utilizate de calculator la execuția acestui algoritm, putem spune că, deși oferă rezultate mai bune, este un consumator mai mare de resurse și timp, întrucât calculele pe care le efectuează sunt mai complicate.

Concluzii

Aceste trei metode sunt cele mai comune metode de conversie la scară gri. Metoda medie tratează fiecare valoare de culoare a pixelului ca fiind aceeași, iar efectul fiecărei culori asupra conversiei scalei gri este același. În caz real, acest lucru nu este necesar. O metodă bună este metoda de luminozitate, în care efectul culorii cele mai intense și culoarea cea mai puțin intensă sunt medii pentru a obține o singură valoare. Aceasta va avea mai multe detalii în imaginea rezultată decât metoda medie. Și, în final, putem observa că metoda luminozității este mult mai bună decât cele de mai sus, deoarece folosește media ponderată a culorii bazată pe percepția umană.

Bibliografie

1. Grayscale to RGB Conversion - https://www.tutorialspoint.com/dip/grayscale_to_rgb_conversion.htm
 2. Three algorithms for converting color to grayscale, *John D. Cook* - <https://www.johndcook.com/blog/2009/08/24/algorithms-convert-color-grayscale/>
 3. Monochrome photography - https://en.wikipedia.org/wiki/Monochrome_photography
 4. Seven grayscale conversion algorithms, *Tanner Helland* - <http://www.tannerhelland.com/3643/grayscale-image-algorithm-vb6/>
 5. Class GreyscaleFilter - <http://jscience.org/experimental/javadoc/org/jspace/computing/ai/vision/GreyscaleFilter.html#Y>
 6. Setting up colour depth - <http://www.verypdf.com/wordpress/201111/how-to-set-the-color-depth-when-converting-emf-to-bmp-13953.html>
-