



ALGORITHMIQUE

IMAGE-saturation

21.01.2020

Mr Jean-Christophe TAROT

Ecole Publique

CCIMP Marseille

SOMMAIRE

- I. Rappel de la problématique page 2
 - II. Partie A
 - 1. fonctionnement de la fonction MYSTERE().... page 3
 - 2. algorithme qui va accentuer le contraste.... page 4
 - III. Partie B
 - 3. implementation de la fonction MYSTERE() et CONTRASTE_PLUS().... page 6
- non fait :
- implementation de la fonction MOYENNE().... page 9
- implementation de la fonction CONTRASTE_MOINS().... page 9
- IV. Jeux de Test.... page 13
 - V. Conclusion

I. Rappel de la Problématique

On souhaite écrire quelques algorithmes modifiant la saturation d'une image matricielle à deux dimensions et composé d'entier compris entre 0 et 100.

Les valeurs : 0 correspondant a une case NOIR

100 correspondant a une case BLANCHE

entre ces deux valeurs : NUANCE DE GRIS

Nous avons une base en pseudo-code de la fonction MYSTERE

II. PARTIE A

1. fonctionnement de la fonction MYSTERE()

FONCTION MYSTERE (tableau de ENTIER:Matrice[][]):tableau de ENTIER

// Matrice : tableau d'entiers de 5 lignes et 6 colonnes

VARIABLE

ENTIER i,j

DEBUT

// lecture du tableau

POUR i **ALLANT_DE** 0 à 4 **FAIRE**

POUR j **ALLANT_DE** 0 à 4 **FAIRE**

Matrice[i][j]<-100 - Matrice[i][j]

FIN_POUR

FIN_POUR

RETOURNE Matrice

FIN

// Matrice initialisée du tableau d'entiers de 5 lignes et 6 colonnes

	25	28	100	100	100	100
	44	45	50	100	100	100
M =	65	65	50	20	100	100
	90	62	100	5	0	0
	60	87	100	0	100	100

// Résultat de la fonction MYSTERE

	75	72	0	0	0	0
	56	55	50	0	0	0
MYSTERE (M) =	35	35	50	80	0	0
	10	38	0	95	100	100
	40	13	0	100	0	0

// On obtient la matrice inverse qui produit une image négative.

2. algorithme qui va accentuer le contraste

```
FONCTION CONTRAST_PLUS (tableau de ENTIER:Matrice[][]):tableau de ENTIER
// Matrice : tableau d'entiers de 5 lignes et 6 colonnes
VARIABLE
    ENTIER i,j
    // cp pour contrast plus
    INT cp
DEBUT
    POUR i ALLANT_DE 0 à 4 FAIRE
        POUR j ALLANT_DE 0 à 5 FAIRE
            SI Matrice[i][j] > 75 FAIRE
                // saturation strictement supérieur à 75
                cp <- 100
            SINON_SI Matrice[i][j] >= 50 FAIRE
                // saturation supérieur à 50
                cp <- 75
            SINON Matrice[i][j]
                // saturation divisé par 2
                cp <- Matrice[i][j] / 2
            FIN_SI
            Matrice[i][j] <- cp
        FIN_POUR
    FIN_POUR

    RETOURNE Matrice
FIN
```

----corrigé

```
POUR i ALLANT_DE 0 à 4 FAIRE
  POUR j ALLANT_DE 0 à 5 FAIRE
    SI Matrice[i][j] > 75 FAIRE
      Matrice[i][j] <- 100
    SINON_SI Matrice[i][j] >= 50 FAIRE
      Matrice[i][j] <- 75
    SINON Matrice[i][j]
      Matrice[i][j] <- Matrice[i][j] / 2
    FIN_SI
  FIN_POUR
FIN_POUR
```

III. PARTIE B

3. implementation de la fonction MYSTERE() et CONTRASTE_PLUS()

```

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package image;

/**
 *
 * @author Jean-Christophe Tarot formation_ep
 */
public class Image {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // CONSTANTES
        final int NL=5,NC=6;
        // VARIABLES

        // tableau test
        int[][] T = new int[][] {
            {5,50},
            {75,40}
        };

        // initialisation de la matrice image
        int[][] M = new int[][] {
            {25,    28,   100,   100,   100,   100},
            {44,    45,    50,   100,   100,   100},
            {65,    65,    50,   20,   100,   100 },
            {90,    62,   100,    5,    0,    0},
            {60,    87,   100,    0,   100,   100}
        };
    }
}

```

```

        };

    AfficheTableau(M);
    System.out.println("Matrice Image");
    // matrice image inversé pour image négative
    Mystere(M);
    AfficheTableau(M);
    System.out.println("Matrice Négative");

    ContrastPlus(M);
    AfficheTableau(M);
    System.out.println("Matrice Image Contrast Plus");
}

public static int[][] Mystere(int[][] tab) {
    // Matrice : tableau d'entiers de 5 lignes et 6 colonnes
    // CONSTANTES
    final int NL=5,NC=6;
    for (int i = 0; i < NL; i++) {
        for (int j = 0; j < NC; j++) {
            tab[i][j] = 100 - tab[i][j];
        }
    }
    return tab;
}

public static int[][] ContrastPlus(int[][] tab) {
    // Matrice : tableau d'entiers de 5 lignes et 6 colonnes
    // CONSTANTES
    final int NL=5,NC=6;
    // VARIABLES
    int cp;

    for (int i = 0; i < NL; i++) {
        for (int j = 0; j < NC; j++) {
            if ( tab[i][j] > 75 ) {
                cp = 100;
            } else if ( tab[i][j] > 50 ) {
                cp = 75;
            } else {
                cp = tab[i][j]/2;
            }
            tab[i][j] = cp;
        }
    }
}

```



```
    }  
    }  
    return tab;  
}  
  
public static void AfficheTableau(int[][] tab) {  
    // Affiche Tableau Dimension 2 ordre 5  
    // CONSTANTES  
    final int NL=5,NC=6;  
    for (int i = 0; i < NL; i++) {  
        System.out.println();  
        for (int j = 0; j < NC; j++) {  
            System.out.print(tab[i][j] + " \t");  
        }  
    }  
}
```

Implementation de Moyenne et Contraste Moins

```

1.  /*
2.  * To change this license header, choose License Headers in Project Properties.
3.  * To change this template file, choose Tools | Templates
4.  * and open the template in the editor.
5.  */
6.  package image;
7.
8.  /**
9.   *
10.  * @author Jean-Christophe Tarot formation_ep
11.  */
12. public class Image {
13.
14.  /**
15.   * @param args the command line arguments
16.   */
17.  public static void main(String[] args) {
18.      // CONSTANTES
19.      final int NL=5,NC=6;
20.      // VARIABLES
21.
22.      // initialisation de la matrice image
23.      int[][] M = new int[][] {
24.          {25,      28,    100,   100,   100,   100},
25.          {44,      45,    50,    100,   100,   100},
26.          {65,      65,    50,    20,    100,   100 },
27.          {90,      62,    100,    5,     0,     0},
28.          {60,      87,    100,    0,     100,   100}
29.      };
30.
31.      AfficheTableau(M);
32.      System.out.println("Matrice Image");
33.      // matrice image inversé pour image négative
34.      Mystere(M);
35.      AfficheTableau(M);
36.      System.out.println("Matrice Négative");

```

```
37.    // matrice image contrasté
38.    ContrastPlus(M);
39.    AfficheTableau(M);
40.    System.out.println("Matrice Image Contrast Plus");
41.    // moyenne des valeurs de la matrice image
42.    System.out.println();
43.    System.out.printf("\rMatrice Image Moyenne : %d", Moyenne(M));
44.    System.out.println();
45.    // matrice image désaturé
46.    ContrastMoins(M);
47.    AfficheTableau(M);
48.    System.out.println("Matrice Image Contrast Moins");
49.    // moyenne des valeurs de la matrice image
50.    System.out.println();
51.    System.out.printf("\rMatrice Image Moyenne : %d", Moyenne(M));
52.    System.out.println();
53. }
54.
55. public static int[][] Mystere(int[][] matrice) {
56.     // Matrice : tableau d'entiers de 5 lignes et 6 colonnes
57.     // CONSTANTES
58.     final int NL=5,NC=6;
59.     for (int i = 0; i < NL; i++) {
60.         for (int j = 0; j < NC; j++) {
61.             matrice[i][j] = 100 - matrice[i][j];
62.         }
63.     }
64.     return matrice;
65. }
66.
67. public static int[][] ContrastPlus(int[][] matrice) {
68.     // Matrice : tableau d'entiers de 5 lignes et 6 colonnes
69.     // CONSTANTES
70.     final int NL=5,NC=6;
71.     // VARIABLES
72.     int cp;
73.
74.     for (int i = 0; i < NL; i++) {
75.         for (int j = 0; j < NC; j++) {
76.             if ( matrice[i][j] > 75 ) {
77.                 cp = 100;
```

```
78.         } else if ( matrice[i][j] > 50 ) {
79.             cp = 75;
80.         } else {
81.             cp = matrice[i][j]/2;
82.         }
83.         matrice[i][j] = cp;
84.     }
85. }
86. return matrice;
87. }
88.
89. public static int Moyenne(int[][] matrice) {
90.     // Moyenne du Tableau Dimension 2 ordre 5
91.     // CONSTANTES
92.     final int NL=5,NC=6;
93.     int somme = 0;
94.     for (int i = 0; i < NL; i++) {
95.         for (int j = 0; j < NC; j++) {
96.             somme = matrice[i][j] + somme;
97.         }
98.     }
99.     return somme / (NL*NC);
100. }
101.
102. public static int[][] ContrastMoins(int[][] matrice) {
103.     // Matrice : tableau d'entiers de 5 lignes et 6 colonnes
104.     // CONSTANTES
105.     final int NL=5,NC=6;
106.     // VARIABLES
107.     int seuil = Moyenne(matrice);
108.
109.     int cm;
110.
111.     for (int i = 0; i < NL; i++) {
112.         for (int j = 0; j < NC; j++) {
113.             cm = ( matrice[i][j] - seuil ) / 2;
114.             matrice[i][j] = matrice[i][j] - cm;
115.         }
116.     }
117.     return matrice;
118. }
```

```
119.  
120.    public static void AfficheTableau(int[][] matrice) {  
121.        // Affiche Tableau Dimension 2 ordre 5  
122.        // CONSTANTES  
123.        final int NL=5,NC=6;  
124.        for (int i = 0; i < NL; i++) {  
125.            System.out.println();  
126.            for (int j = 0; j < NC; j++) {  
127.                System.out.print(matrice[i][j] + " \t");  
128.            }  
129.        }  
130.    }  
131.    // fin main  
132. }
```

IV Jeux de test

run:

25	28	100	100	100	100	
44	45	50	100	100	100	
65	65	50	20	100	100	
90	62	100	5	0	0	
60	87	100	0	100	100	Matrice Image
75	72	0	0	0	0	
56	55	50	0	0	0	
35	35	50	80	0	0	
10	38	0	95	100	100	
40	13	0	100	0	0	Matrice Négative
75	75	0	0	0	0	
75	75	25	0	0	0	
17	17	25	100	0	0	
5	19	0	100	100	100	
20	6	0	100	0	0	Matrice Image Contrast Plus

Matrice Image Moyenne : 31

53	53	15	15	15	15	
53	53	28	15	15	15	
24	24	28	66	15	15	
18	25	15	66	66	66	
25	18	15	66	15	15	Matrice Image Contrast Moins

Matrice Image Moyenne : 30

BUILD SUCCESSFUL (total time: 0 seconds)

CONCLUSION

utilité de ce code ?

on peut voir une fluctuation de la fonction Image Moyenne (passe de 31 a 30) avant et après l'appel de la fonction ImageContrasteMoins .