



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 5
- non fait :
- implementation de la fonction MOYENNE().... page 00
- implementation de la fonction CONTRASTE_MOINS().... page 00
- IV. Jeux de Test.... page 9
 - 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) =	45	45	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 4 **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

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");  
        }  
    }  
}  
}
```




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

BUILD SUCCESSFUL (total time: 0 seconds)

CONCLUSION

utilité de ce code ?

.