

VISA – TP Couleurs

Arnaud Cojez

mercredi 9 novembre 2016

Table des matières

1	Introduction	3
2	Manipulation de la luminance	4
2.1	Explication	4
2.1.1	Cas 1	4
2.1.2	Cas 2	5
2.1.3	Cas 3	6
2.1.4	Cas 4	7
2.2	Résultats	8
2.2.1	Cas 1	8
2.2.2	Cas 2	9
2.2.3	Cas 3	10
2.2.4	Cas 4	11
3	Rétablissement de la saturation	12
3.1	Explication	12
3.1.1	Cas 2	12
3.1.2	Cas 1	13
3.2	Résultats	14
3.2.1	Cas 2	14
3.2.2	Cas 1	15
4	Transformation de la teinte	16
4.1	Explication	16
4.1.1	Cas 3	16
4.2	Résultats	17
4.2.1	Cas 3	17
5	Analyse dans des espaces couleur adaptés	18
5.1	Explication	18
5.1.1	Cas 3	18
5.2	Analyse	19
6	Modification de la luminance adaptée	20
6.1	Explication	20
6.1.1	Cas 1	20
6.2	Analyse	21
7	Conclusion	22
8	Annexe	23
8.1	Manipulation de la luminance	23
8.2	Rétablissement de la saturation	24
8.3	Transformation de la teinte	25

1 Introduction

Une fois la capture de la lumière maîtrisée, il nous faut définir la façon dont sera stockée l'image formée.

L'information des pixels d'une image peut être stockée selon différentes représentations. Par exemple, nous pouvons considérer différentes couleurs comme les composantes de l'image.

Ainsi, chaque opération effectuée sur une image aura un effet lié à la représentation de celle-ci.

Le long de ce TP, nous utiliserons l'espace de couleurs HSB (Hue-Saturation-Brightness). Celui-ci nous permet de modifier la luminance, la teinte et la saturation de l'image.

Les fichiers utilisés sont des images composant le test d'Ishihara, qui permet de détecter les déficiences dichromatiques telles que le daltonisme.

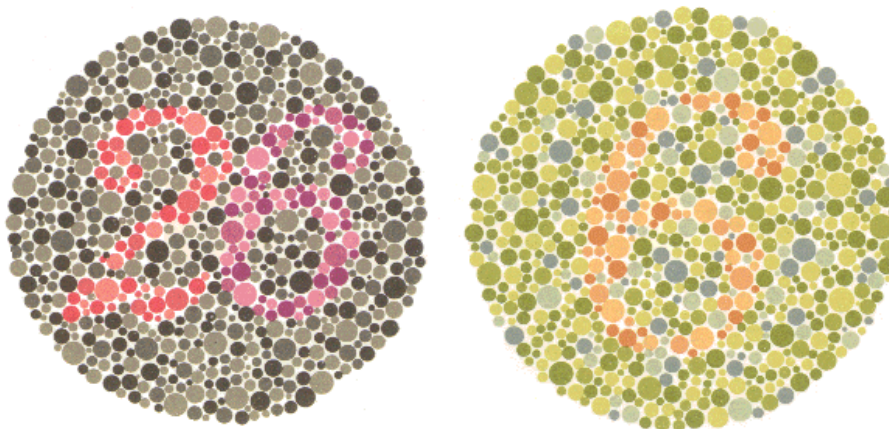


FIGURE 1 – Planches issues du Test d'Ishihara

2 Manipulation de la luminance

2.1 Explication

TODO expliquer principe

2.1.1 Cas 1

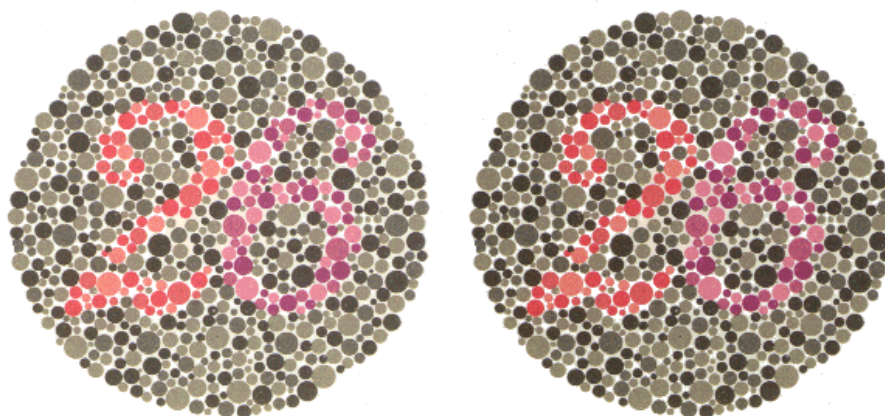


FIGURE 2 – À gauche, image originale. À droite, image modifiée

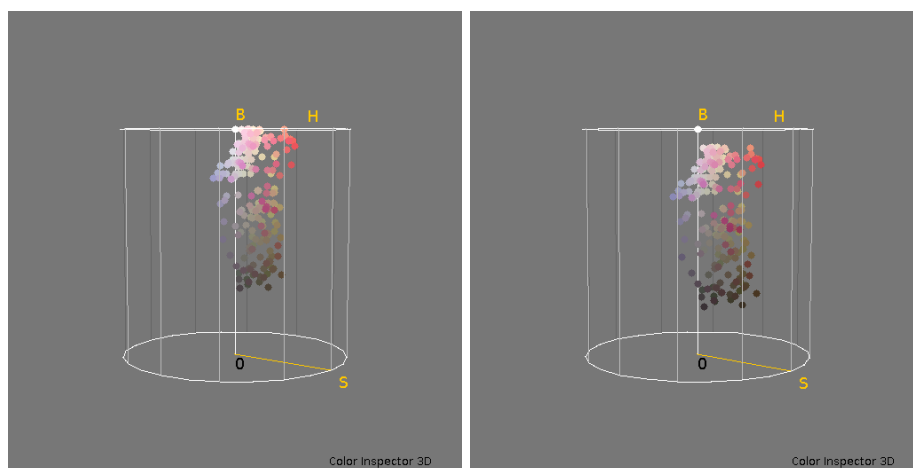


FIGURE 3 – Visualisation des 2 images dans l'espace HSB

TODO expliquer différence estimée

2.1.2 Cas 2

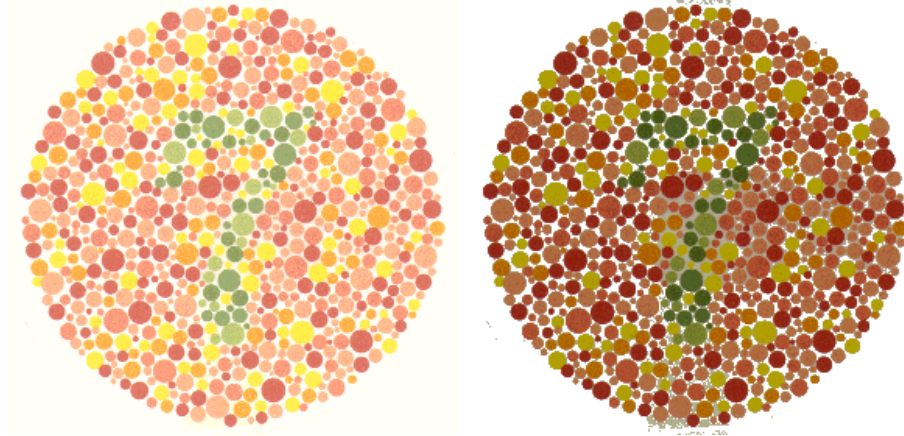


FIGURE 4 – À gauche, image originale. À droite, image modifiée

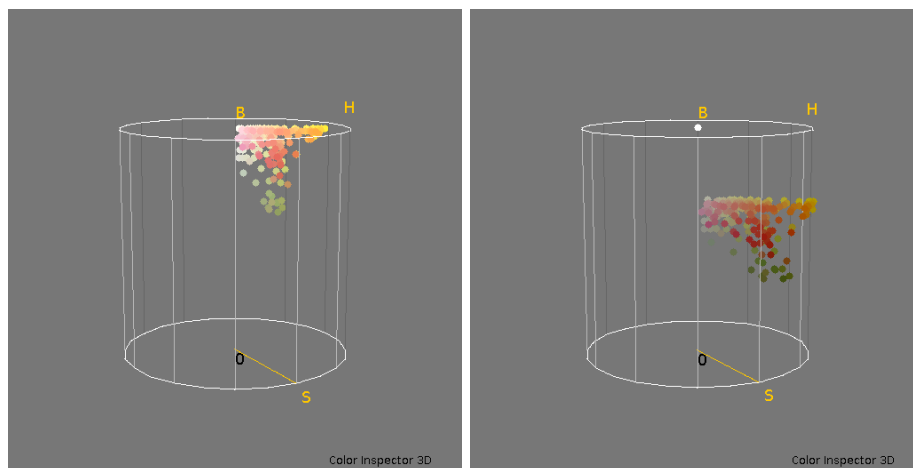


FIGURE 5 – Visualisation des 2 images dans l'espace HSB

TODO expliquer différence estimée

2.1.3 Cas 3

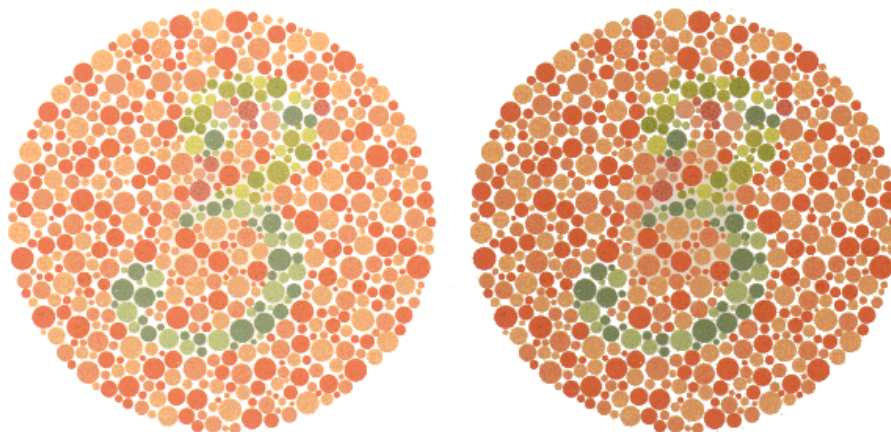


FIGURE 6 – À gauche, image originale. À droite, image modifiée

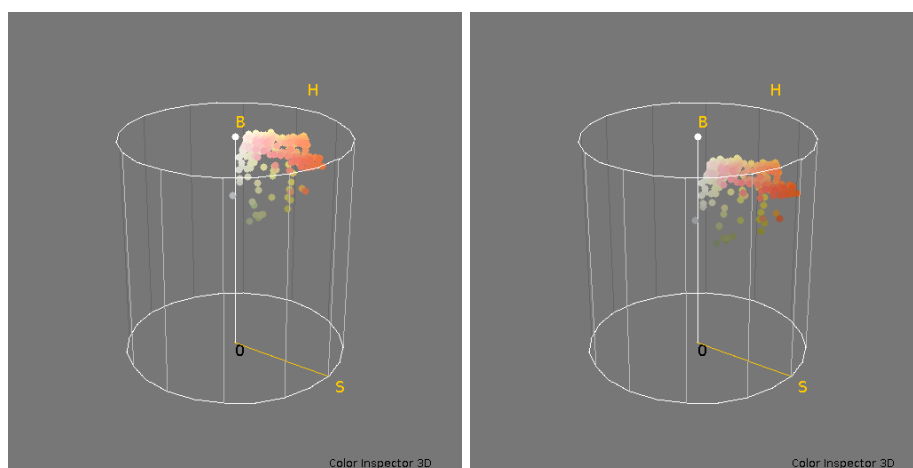


FIGURE 7 – Visualisation des 2 images dans l'espace HSB

TODO expliquer différence estimée

2.1.4 Cas 4

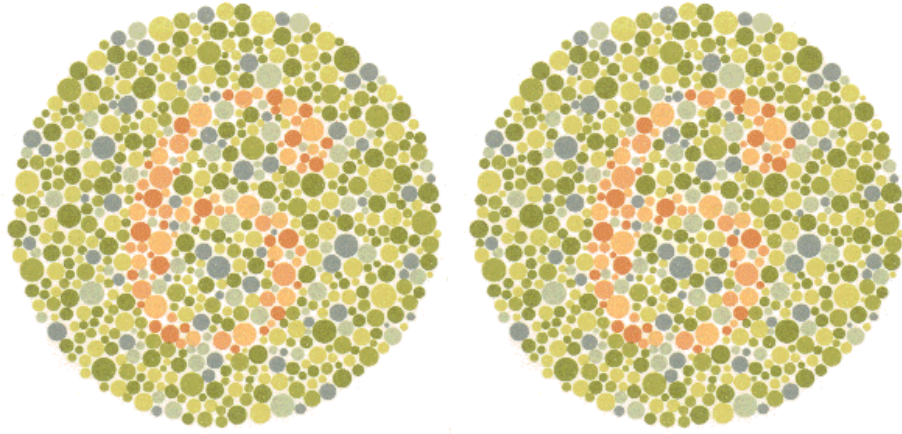


FIGURE 8 – À gauche, image originale. À droite, image modifiée

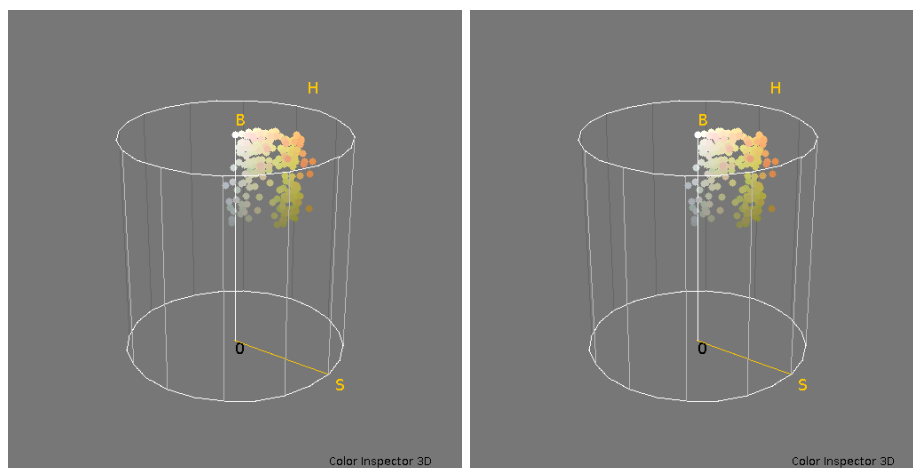


FIGURE 9 – Visualisation des 2 images dans l'espace HSB

TODO expliquer différence estimée

2.2 Résultats

2.2.1 Cas 1

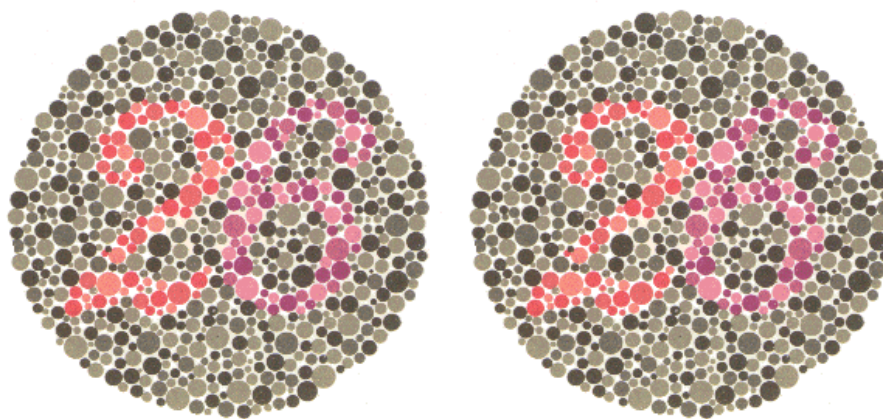


FIGURE 10 – À gauche, image originale. À droite, image retouchée

TODO analyse des images

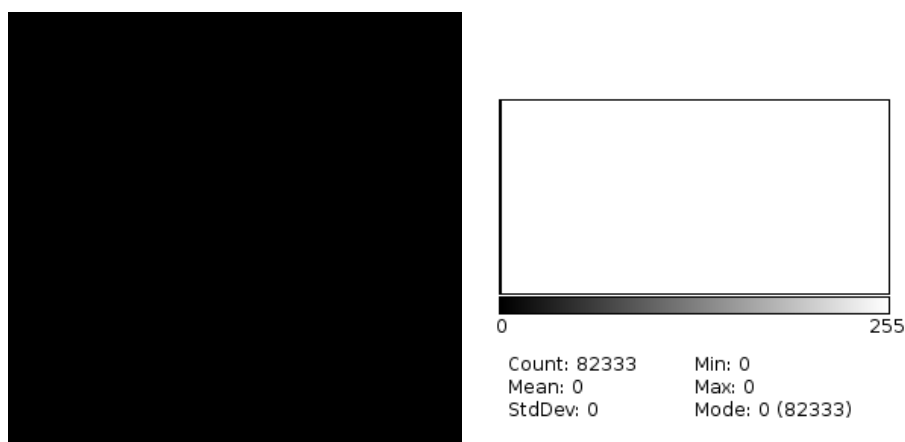


FIGURE 11 – Carte des différences et son histogramme

2.2.2 Cas 2

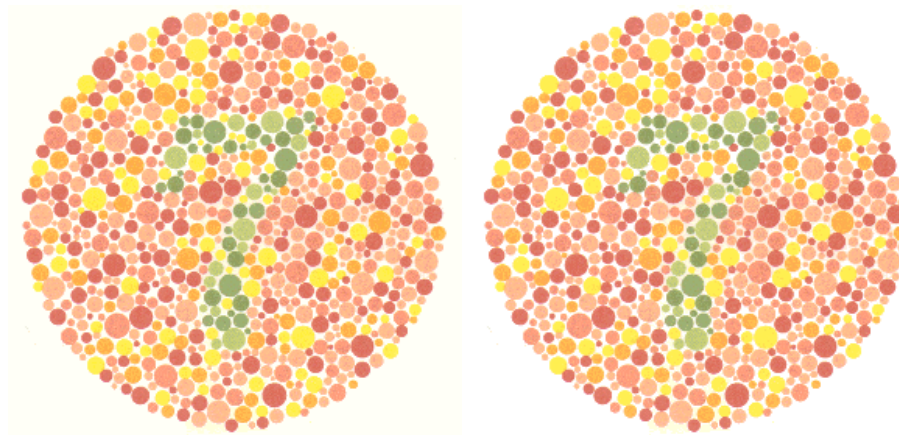


FIGURE 12 – À gauche, image originale. À droite, image retouchée

TODO analyse des images

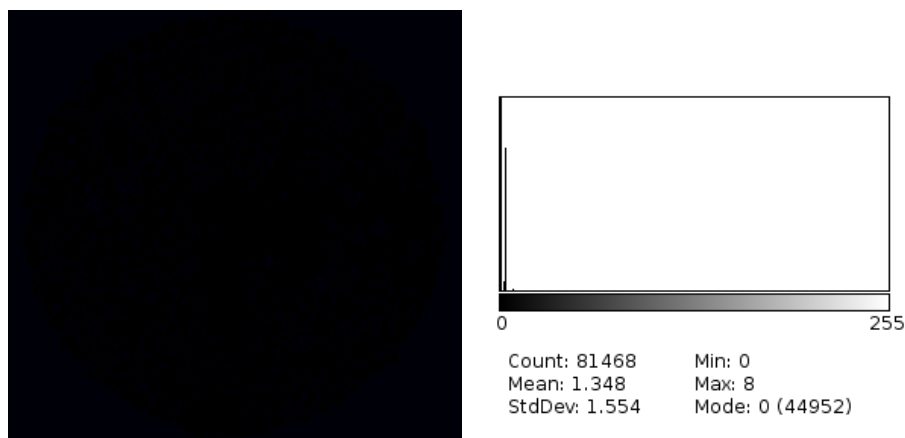


FIGURE 13 – Carte des différences et son histogramme

2.2.3 Cas 3

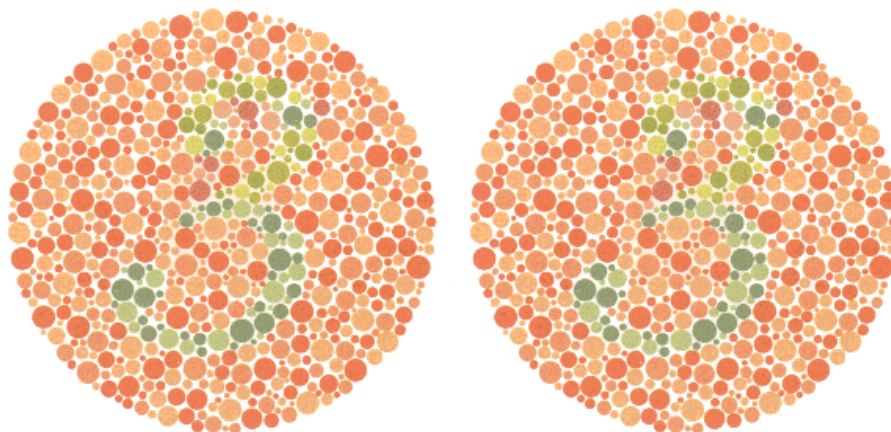


FIGURE 14 – À gauche, image originale. À droite, image retouchée

TODO analyse des images



FIGURE 15 – Carte des différences et son histogramme

2.2.4 Cas 4

TODO Pas de modification à faire

3 Rétablissement de la saturation

3.1 Explication

TODO expliquer principe

3.1.1 Cas 2

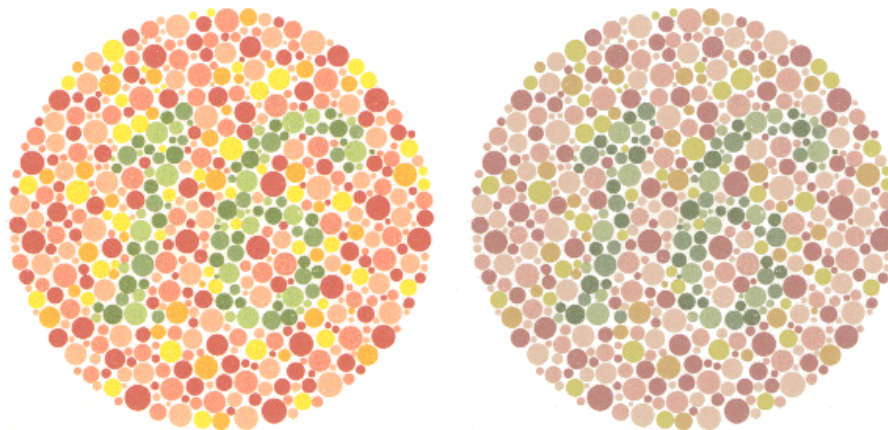


FIGURE 16 – Visualisation des 2 images dans l'espace HSB

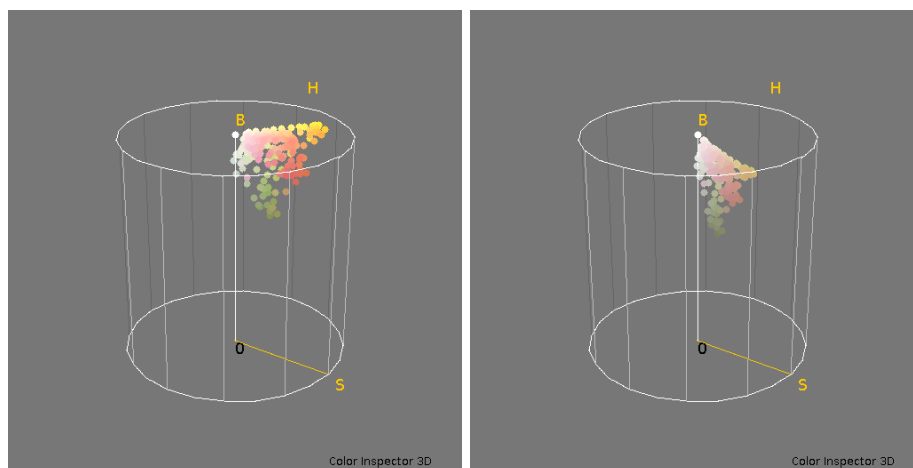


FIGURE 17 – Visualisation des 2 images dans l'espace HSB

TODO expliquer différence estimée

3.1.2 Cas 1

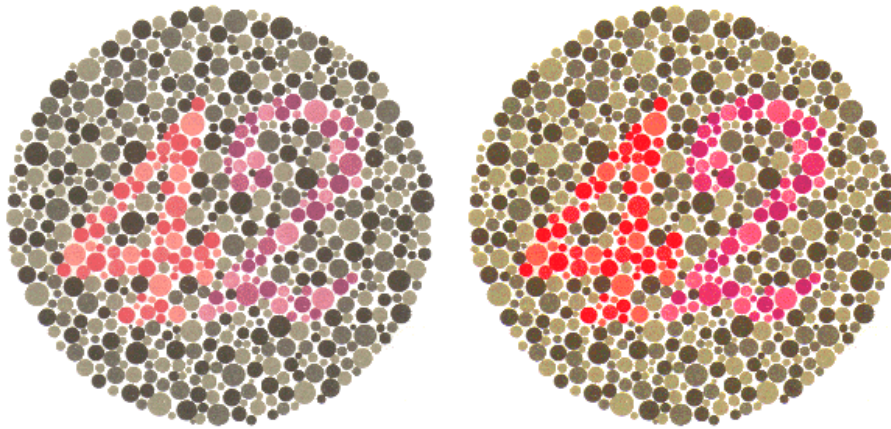


FIGURE 18 – Visualisation des 2 images dans l'espace HSB

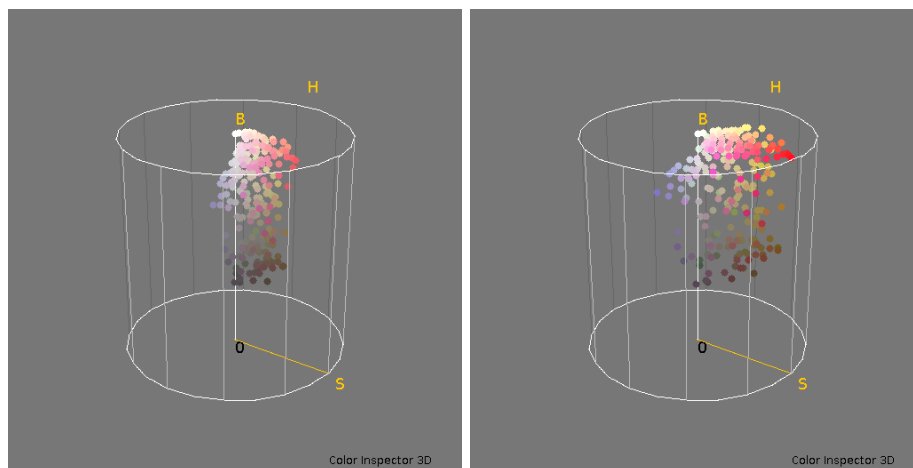


FIGURE 19 – Visualisation des 2 images dans l'espace HSB

TODO expliquer différence estimée

3.2 Résultats

3.2.1 Cas 2

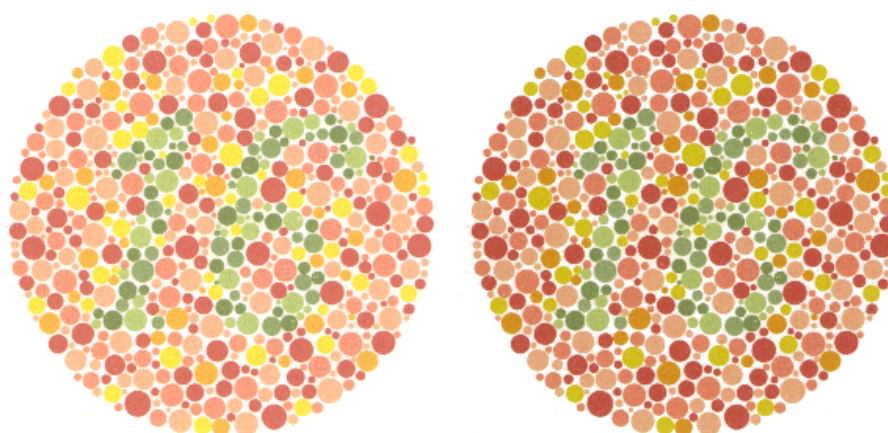


FIGURE 20 – À gauche, image originale. À droite, image retouchée

TODO analyse résultats

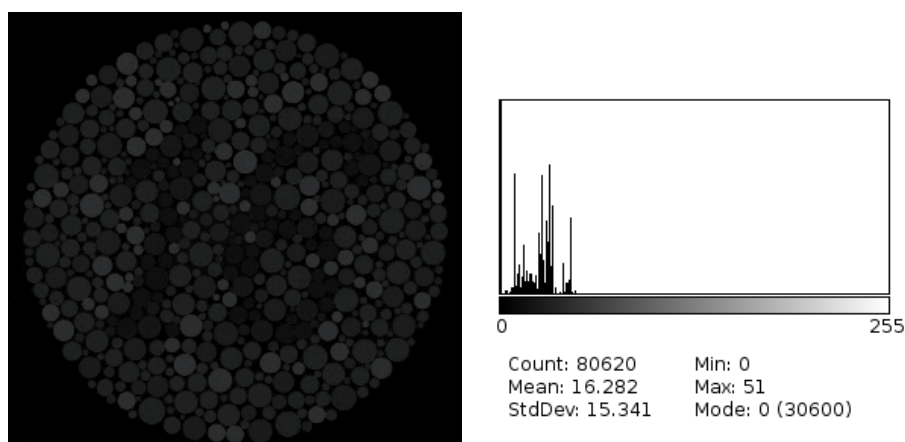


FIGURE 21 – Carte des différences et son histogramme

3.2.2 Cas 1

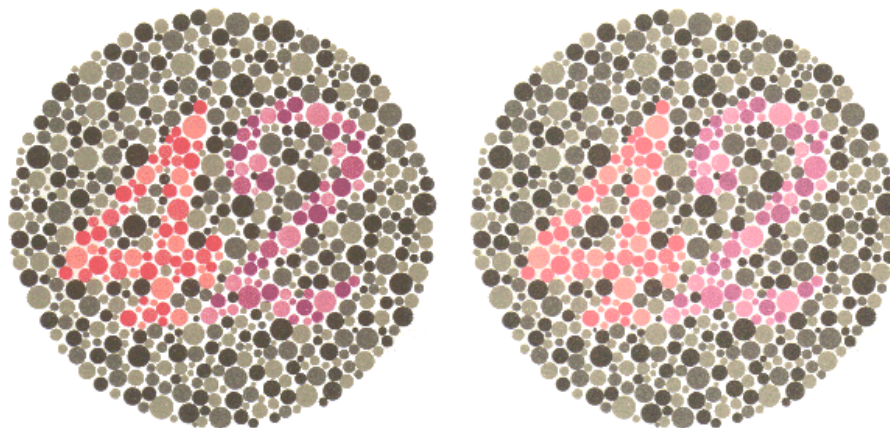


FIGURE 22 – À gauche, image originale. À droite, image retouchée

TODO analyse résultats

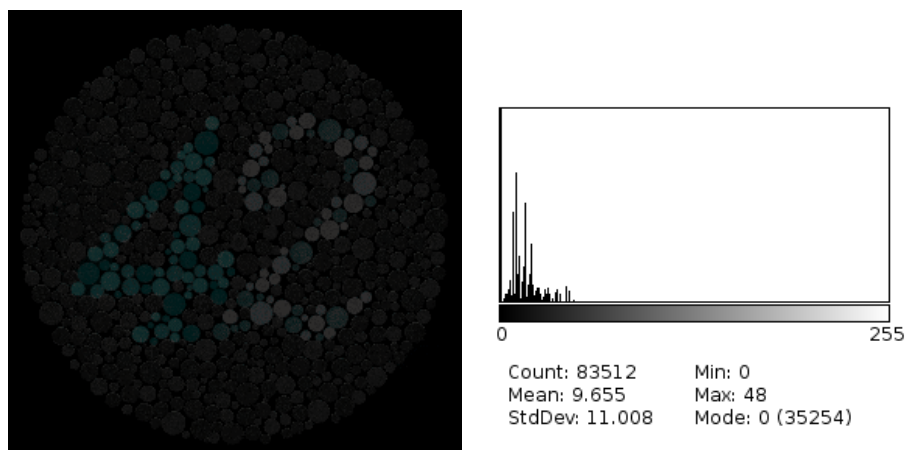


FIGURE 23 – Carte des différences et son histogramme

4 Transformation de la teinte

4.1 Explication

TODO expliquer principe

4.1.1 Cas 3

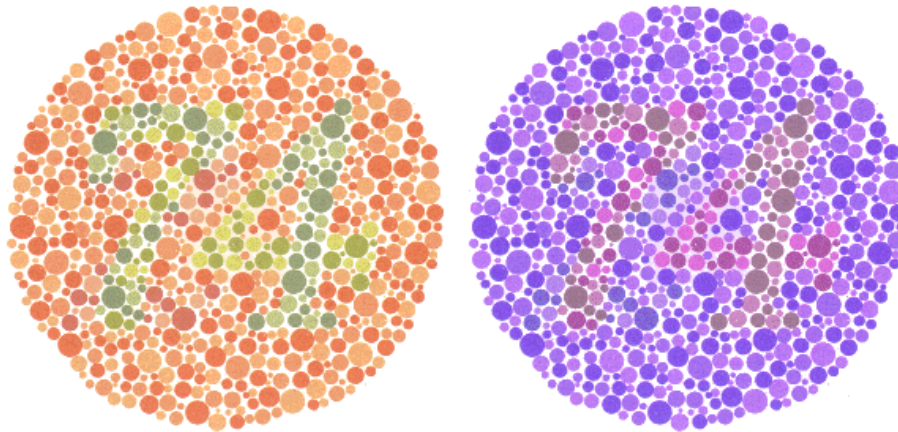


FIGURE 24 – À gauche, image originale. À droite, image modifiée

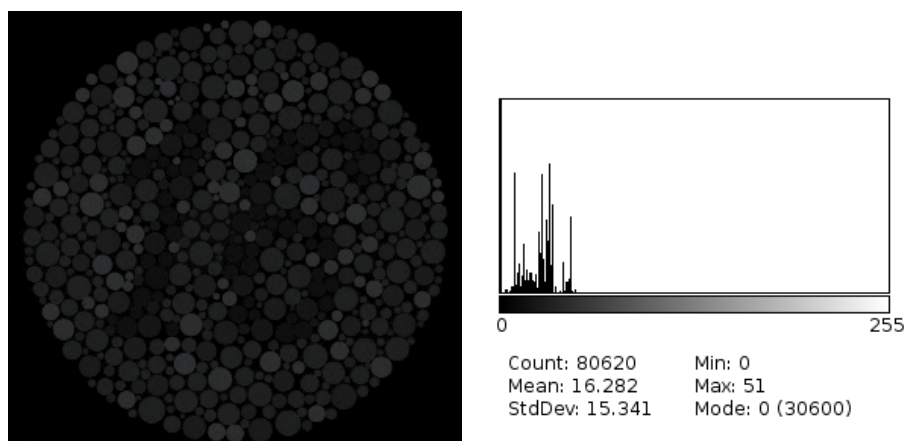


FIGURE 25 – Visualisation des 2 images dans l'espace HSB

TODO expliquer différence estimée

4.2 Résultats

4.2.1 Cas 3

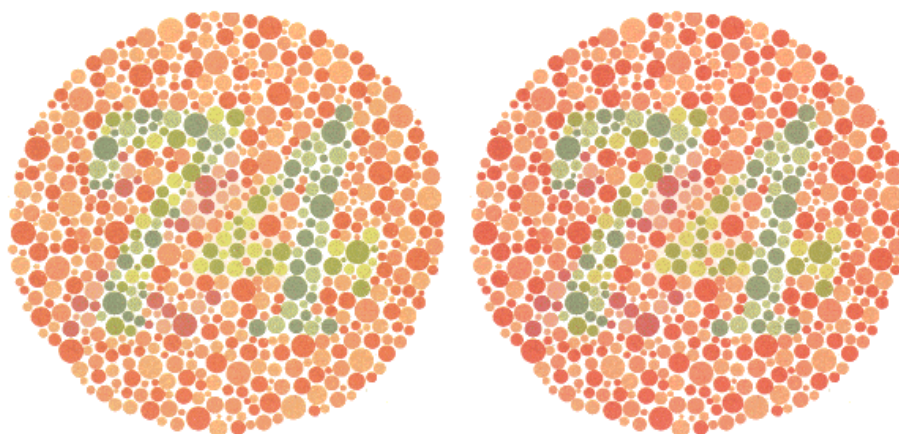


FIGURE 26 – À gauche, image originale. À droite, image retouchée

TODO analyse résultats

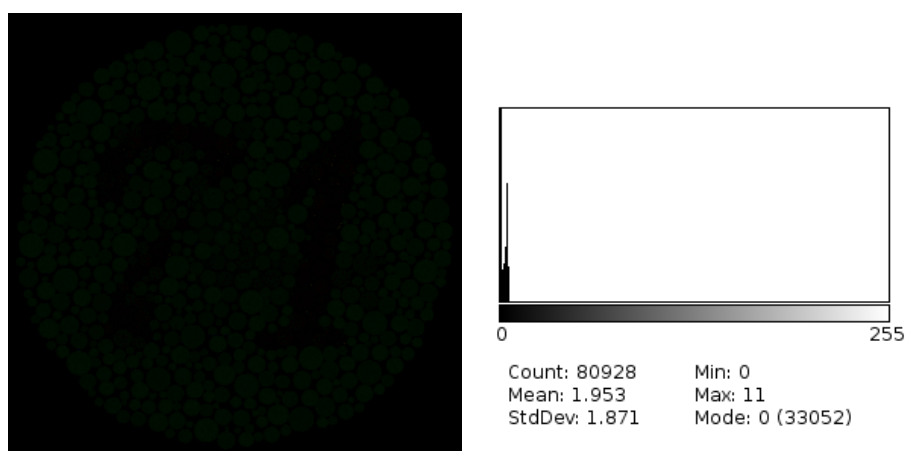


FIGURE 27 – Carte des différences et son histogramme

5 Analyse dans des espaces couleur adaptés

5.1 Explication

TODO expliquer question

5.1.1 Cas 3

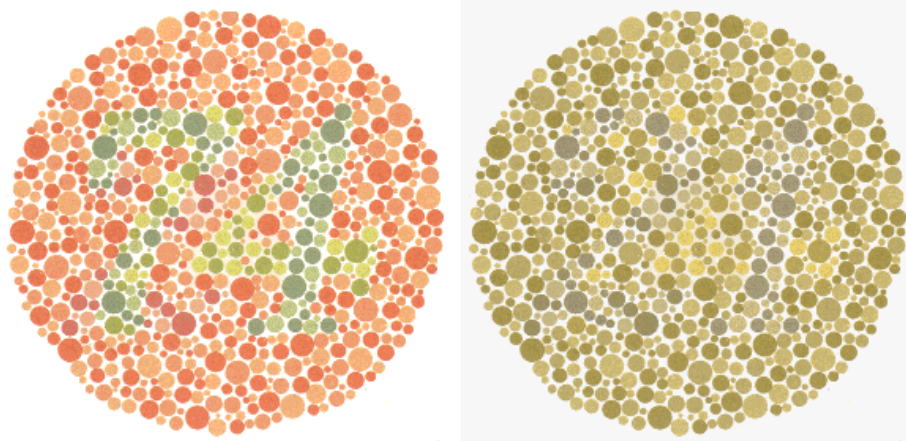


FIGURE 28 – À gauche, image originale. À droite, image modifiée

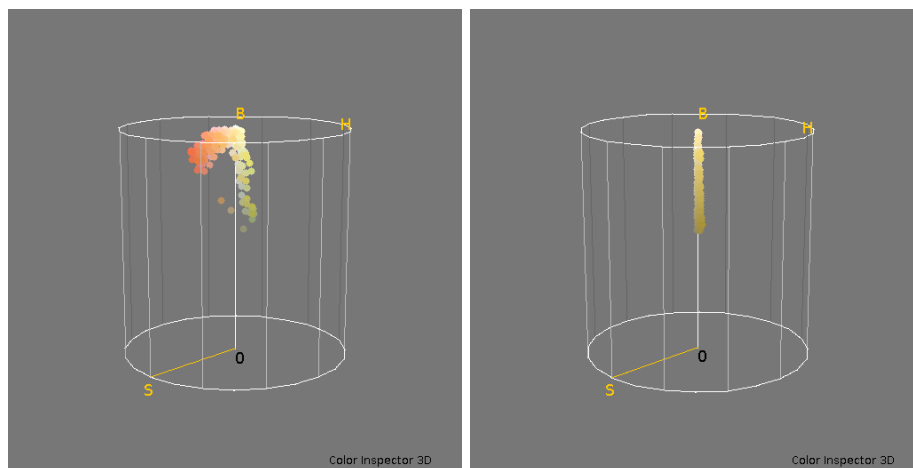


FIGURE 29 – Visualisation des 2 images dans l'espace HSB

5.2 Analyse

TODO analyse

6 Modification de la luminance adaptée

6.1 Explication

TODO expliquer question

6.1.1 Cas 1

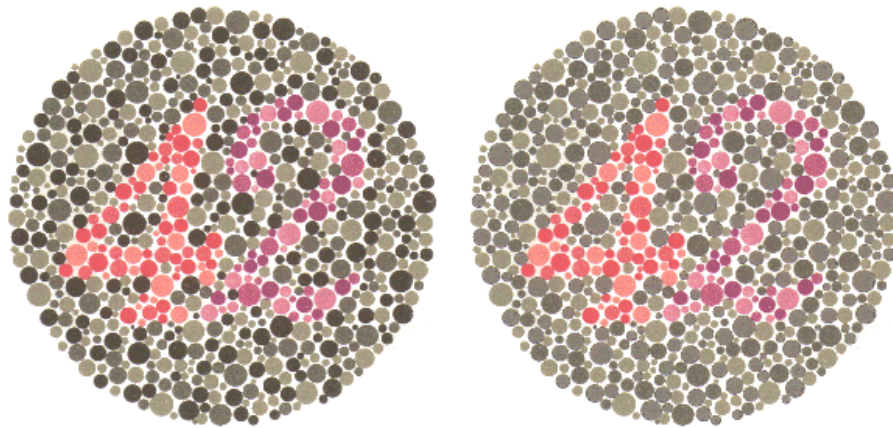


FIGURE 30 – À gauche, image originale. À droite, image modifiée

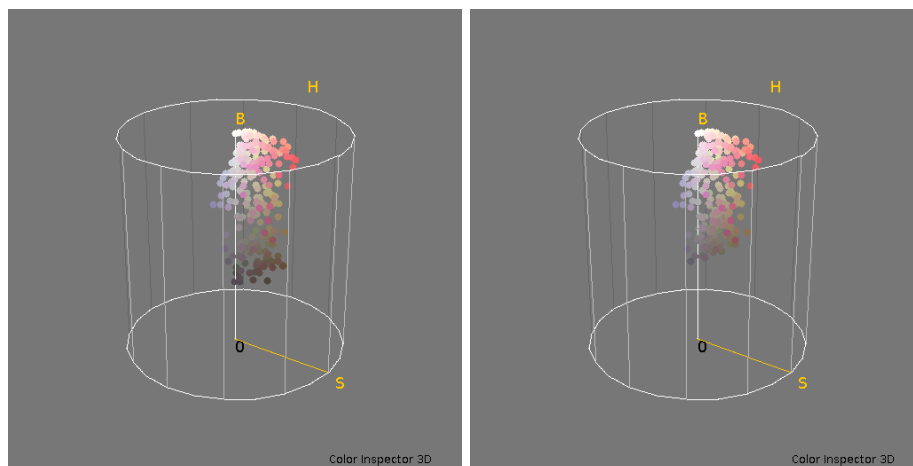


FIGURE 31 – Visualisation des 2 images dans l'espace HSB

6.2 Analyse

TODO analyse

7 Conclusion

Nous avons pu constater qu'une image possède plusieurs composantes, mais que celles-ci peuvent différer en fonction de la représentation de l'image, ou espace colorimétrique.

Chaque opération sur un espace colorimétrique aura un effet propre à l'espace. De plus, les jeux de composantes pourront mettre en évidence différents phénomènes apparaissant sur l'image.

Il existe cependant une limite aux opérations sur les espaces colorimétriques. Lorsqu'une image a été modifiée, il est possible de perdre de l'information à cause d'un effet de plancher/plafond.

Il peut donc être utile de convertir un espace colorimétrique en un autre afin de pouvoir profiter d'opérations spécifiques, ou pour analyser la composition de l'image sous un autre angle.

8 Annexe

8.1 Manipulation de la luminance

```
1 // Une macro-squelette pour la couleur.
2 // Version: 0.1
3 // Date: 14/09/2011
4 // Author: L. Macaire
5
6 macro "augmentation_luminance" {
7
8 // recuperation du ID de l'image
9 image = getImageID();
10
11 valeur = getNumber ("quelle augmentation (absolue) de
12     luminance",valeur);
13
14 Dialog.create("Debut");
15 Dialog.addMessage(" Cliquer sur OK pour commencer le
16     traitement ");
17 Dialog.show();
18
19 setBatchMode(true);
20
21
22 // recuperation de la taille W x H de l'image
23 W = getWidth();
24 H = getHeight();
25
26 run("Duplicate...", "title=luminance modifiee");
27 image_luminance_aug = getImageID();
28
29 for (j=0; j<H; j++) {
30     for (i=0; i<W; i++)
31     {
32         selectImage (image);
33         couleur_avant = getPixel(i,j);
34         R_avant = (couleur_avant & 0xff0000) >> 16;
35         G_avant = (couleur_avant & 0x00ff00) >> 8;
36         B_avant = (couleur_avant & 0x0000ff) ;
37
38         R_apres = minOf(R_avant + valeur, 255);
39         G_apres = minOf(G_avant + valeur, 255);
40         B_apres = minOf(B_avant + valeur, 255);
41
42         couleur_apres = ((R_apres & 0xff ) << 16) + ((G_apres & 0
43             xff) << 8) + B_apres & 0xff;
44
45         selectImage (image_luminance_aug);
46         setPixel(i,j,couleur_apres);
47     }
48 }
```

```

48     }
49
50     setBatchMode(false);
51
52     Dialog.create("Fin");
53     Dialog.addMessage(" Cliquer sur OK pour terminer le
        traitement");
54     Dialog.show();
55
56
57 }

```

TODO reproprer code

8.2 Rétablissement de la saturation

```

1  // Macro permettant de multiplier la saturation d'une image
2  // Version: 0.1
3  // Date: 26/10/2016
4  // Author: L. Macaire & A. Cojez
5
6  macro "multiplication saturation" {
7
8  // recuperation du ID de l'image
9  image = getImageID();
10
11  valeur = getNumber ("quelle multiplication (absolue) de
        saturation",valeur);
12
13  Dialog.create("Debut");
14  Dialog.addMessage(" Cliquer sur OK pour commencer le
        traitement ");
15  Dialog.show();
16
17
18  setBatchMode(true);
19
20  //operation start
21  titre=getTitle();
22  //print (titre);
23  run("Color Space Converter", "from=RGB to=HSB white=D65");
24  run("Split Channels");
25  command = titre+" (HSB) (green)";
26  selectWindow(command);
27  //selectWindow("it3_72pp_saturation_faible.bmp (HSB) (green)
        ");
28
29  run("Multiply...", "value=" + valeur);
30
31  command = "c1=["+titre+" (HSB) (red)] c2=["+titre+" (HSB) (
        green)] c3=["+titre+" (HSB) (blue)] ignore"
32  //run("Merge Channels...", "c1=[it3_72pp_saturation_faible.
        bmp (HSB) (red)] c2=[it3_72pp_saturation_faible.bmp (HSB)

```



```

        (green)] c3=[it3_72pp_saturation_faible.bmp (HSB) (blue)
        ] ignore");
33 run("Merge Channels...", command);
34 run("Color Space Converter", "from=HSB to=RGB white=D65");
35 //operation end
36
37
38 setBatchMode(false);
39
40 Dialog.create("Fin");
41 Dialog.addMessage(" Cliquer sur OK pour terminer le
    traitement");
42 Dialog.show();
43
44
45 }

```

TODO reproprer code

8.3 Transformation de la teinte

```

1 // Macro permettant de multiplier la saturation d'une image
2 // Version: 0.1
3 // Date: 26/10/2016
4 // Author: L. Macaire & A. Cojez
5
6 macro "Changement de Teinte" {
7
8 // recuperation du ID de l'image
9 image = getImageID();
10
11 valeur = getNumber ("quel degre de rotation de teinte",
    valeur);
12
13 Dialog.create("Debut");
14 Dialog.addMessage(" Cliquer sur OK pour commencer le
    traitement ");
15 Dialog.show();
16
17
18 setBatchMode(true);
19
20 //operation start
21 titre=getTitle();
22 //print (titre);
23 run("Color Space Converter", "from=RGB to=HSB white=D65");
24 run("Split Channels");
25 command = titre+" (HSB) (red)";
26 selectWindow(command);
27 //selectWindow("it3_72pp_saturation_faible.bmp (HSB) (green)
    ");
28
29 run("Add...", "value=" + (valeur * 256) / 360);
30

```

```

31 command = "c1=["+titre+" (HSB) (red)] c2=["+titre+" (HSB) (
    green)] c3=["+titre+" (HSB) (blue)] ignore"
32 //run("Merge Channels...", "c1=[it3_72pp_saturation_faible.
    bmp (HSB) (red)] c2=[it3_72pp_saturation_faible.bmp (HSB)
    (green)] c3=[it3_72pp_saturation_faible.bmp (HSB) (blue)
    ] ignore");
33 run("Merge Channels...", command);
34 run("Color Space Converter", "from=HSB to=RGB white=D65");
35 //operation end
36
37
38 setBatchMode(false);
39
40 Dialog.create("Fin");
41 Dialog.addMessage(" Cliquer sur OK pour terminer le
    traitement");
42 Dialog.show();
43
44
45 }

```

TODO repropser code