

Modification d'images couleurs

25/10/2017

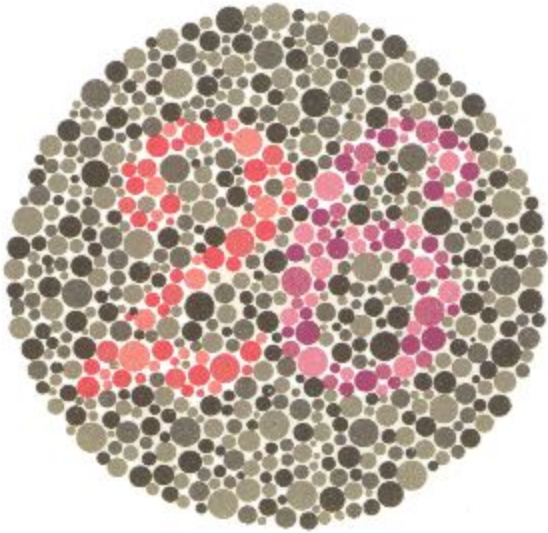
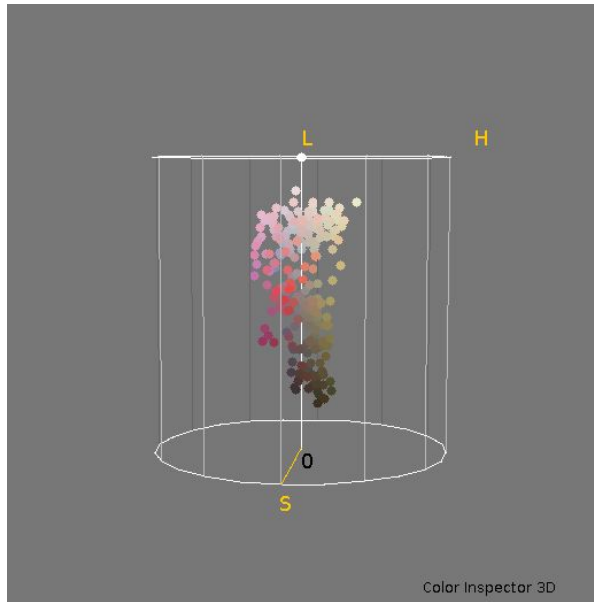
Introduction

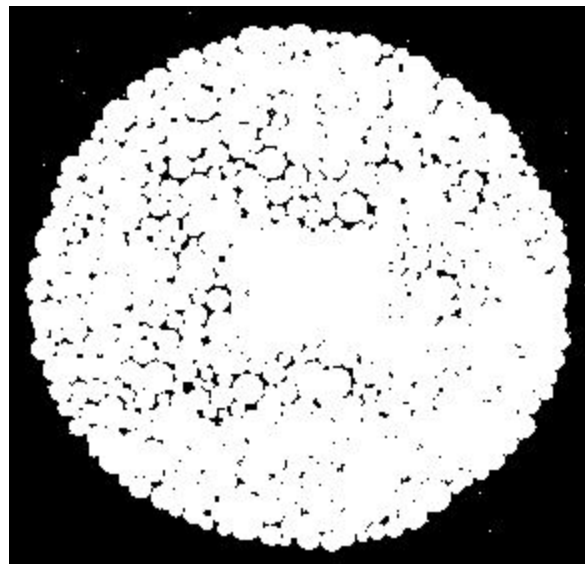
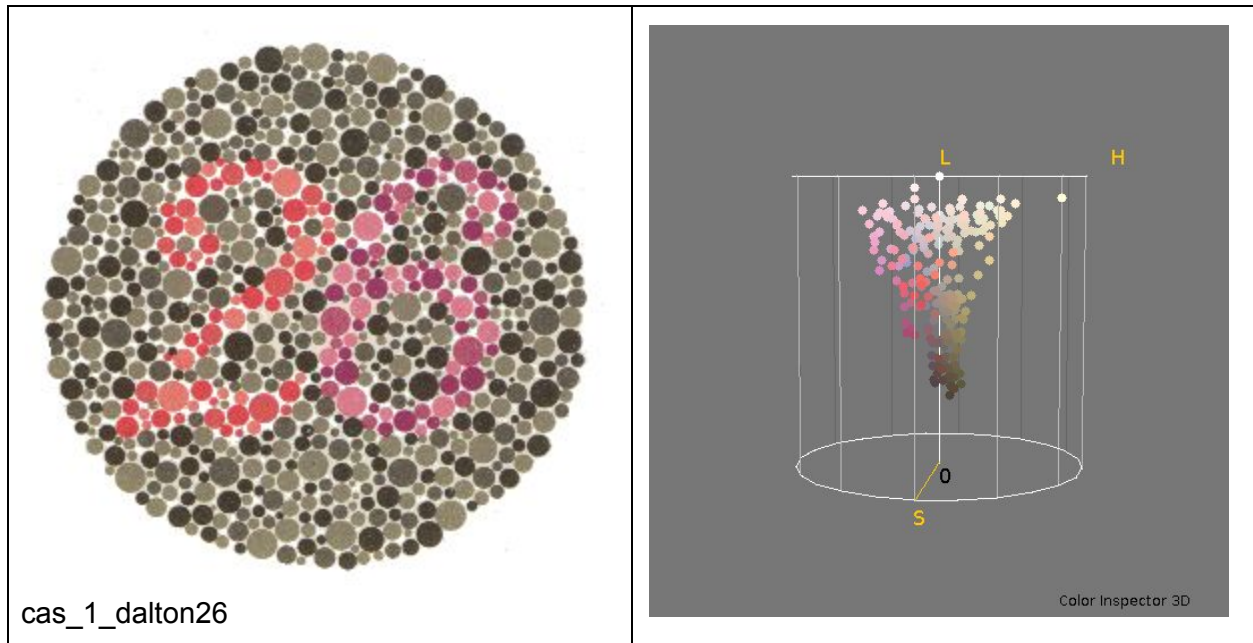
Pour représenter une image couleur il existe plusieurs types de représentations comme RGB (Red, Green, Blue) ou encore HSL (Hue, Saturation, Lightness).

Dans ce TP nous allons utiliser ces formats afin à la fois d'appliquer des changements aux différentes images mais aussi de détecter les changements qui ont été effectués en amont. Tout ceci se fera avec l'aide du plugin Color_Inspector_3D et Color_Space_Converter.

Manipulation de la luminance

Nous avons commencé par manipuler la luminance de l'image en utilisant le plugin "Color Inspector 3D". Nous avons donc inspecté les deux images "cas_1_luminance" et "cas_1_dalton26" dans leur domaine de couleur HSL.

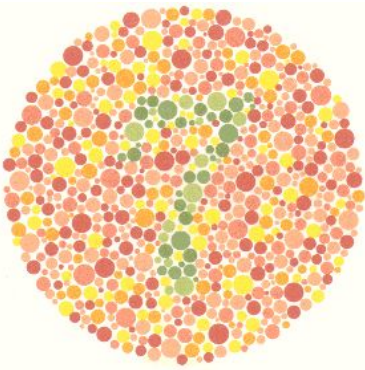
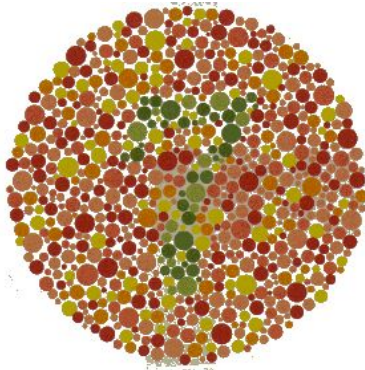
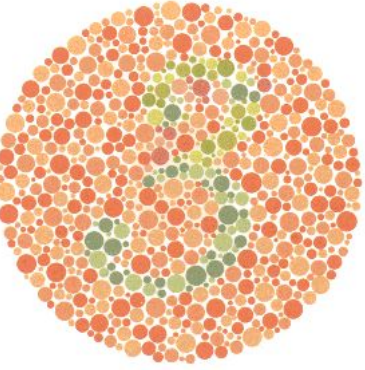
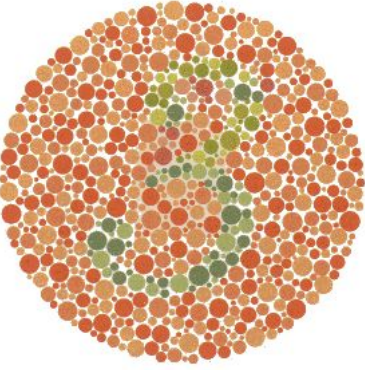
Image	Visualisation
 cas_1_luminance	 Color Inspector 3D



Différence de luminance entre les deux images

Lorsque l'on fait la différence entre les deux images on obtient des différences de 20. Ce qui nous montre que la différence de luminance est de 20. Nous avons donc appliqué une diminution de la luminance de 20 avec les sliders de l'inspecteur 3D. Nous sommes bien retombé sur une image très similaire, voire identique à l'image dont la luminance n'avait pas été modifié.

Nous avons fait de même pour les cas 2 et 3 (le cas 4 ne fonctionnant pas).

Image	Image Luminance diminuée	Différence de luminance
		80
		30

Nous avons ensuite modifié la macro donnée pour lui permettre de changer la luminance $Y=(R+G+B)/3$ en entrant une valeur de luminance étant ajoutée à chaque composante:

$$R_{modif} = R_{orig} + \phi$$

$$G_{modif} = G_{orig} + \phi$$

$$B_{modif} = B_{orig} + \phi$$

Ceci permet à partir de l'image dont la luminance a été diminuée de retrouver l'image d'origine par tâtonnement.

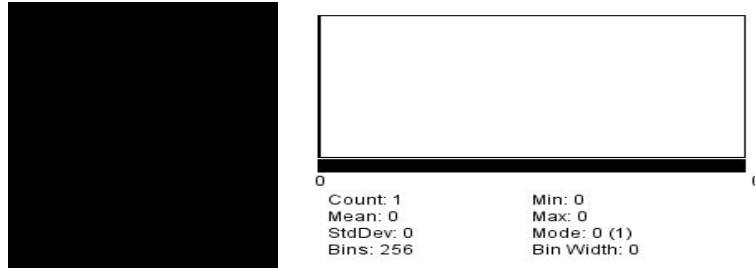
Modification dans la macro:

```
for (j=0; j<H; j++) {  
    for (i=0; i<W; i++)  
    {  
        selectImage (image);  
        couleur_avant = getPixel(i,j);  
        R_avant = (couleur_avant & 0xff0000) >> 16;  
        G_avant = (couleur_avant & 0x00ff00) >> 8;  
        B_avant = (couleur_avant & 0x0000ff) ;  
  
        R_apres = R_avant + valeur; // fonction de R_avant ;  
        G_apres = G_avant + valeur; // fonction de G_avant ;  
        B_apres = B_avant + valeur; // fonction de B_avant ;  
  
        if(R_apres >255) R_apres = 255;  
        if(G_apres >255) G_apres = 255;  
        if(B_apres >255) B_apres = 255;  
  
        couleur_apres = ((R_apres & 0xff) << 16) + ((G_apres & 0xff) << 8) + B_apres & 0xff;  
  
        selectImage (image_luminance_aug);  
        setPixel(i,j,couleur_apres);  
    }  
}
```

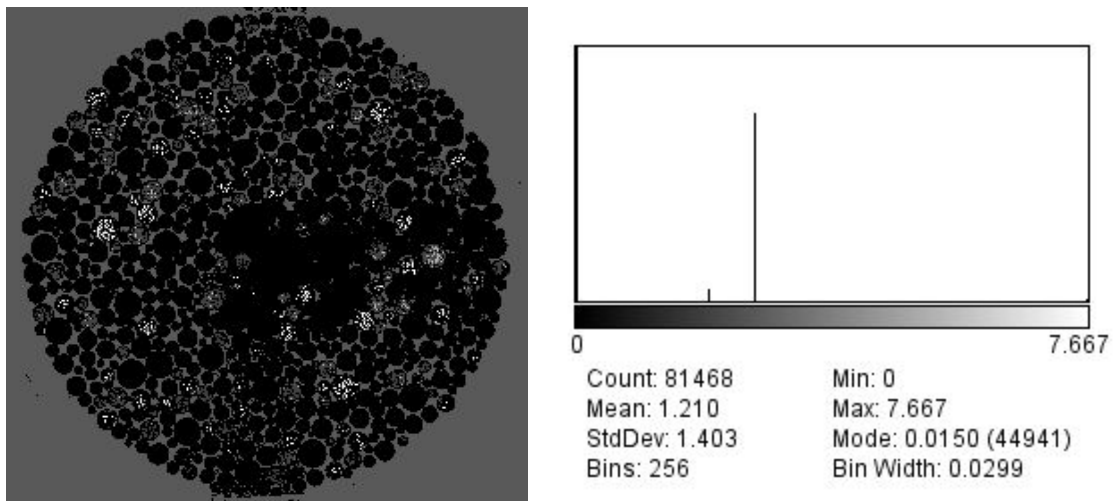
En tâtonnant avec cette macro nous obtenons des résultats similaires aux résultats obtenues avec l'inspecteur 3D à savoir 20 pour le cas 1, 80 pour le 2 et 30 pour le 3.

Pour vérifier les résultats nous avons effectué une différence entre le résultat et l'image d'origine puis nous avons analysé l'histogramme de cette image.

Pour les cas 1 et 3 nous obtenons des images complètement noirs pour la différence entre l'image dont la luminance a été modifiée et l'image d'origine. Ce qui veut dire que les images calculées et l'image d'origine sont identiques.



Cependant pour le cas 2, le résultat est sensiblement différent alors qu'à l'oeil les images semblent identiques. Pour le cas 2 nous obtenons les résultats suivants:



Ce résultat nous montre que la modification de la luminance n'a pas donné la même image que celle d'origine, effectivement lorsque l'on regarde le fond de l'image dalton nous pouvons observer qu'il est légèrement rosé contrairement à celui de l'image dont la luminance a été modifiée. On aperçoit aussi l'apparition d'artefact blanc dans la différence ce qui nous montre que la luminance n'a pas été modifiée avec le même facteur partout dans l'image "cas_2_luminance". Tout ceci nous montre que la luminance ne transforme pas parfaitement l'image.

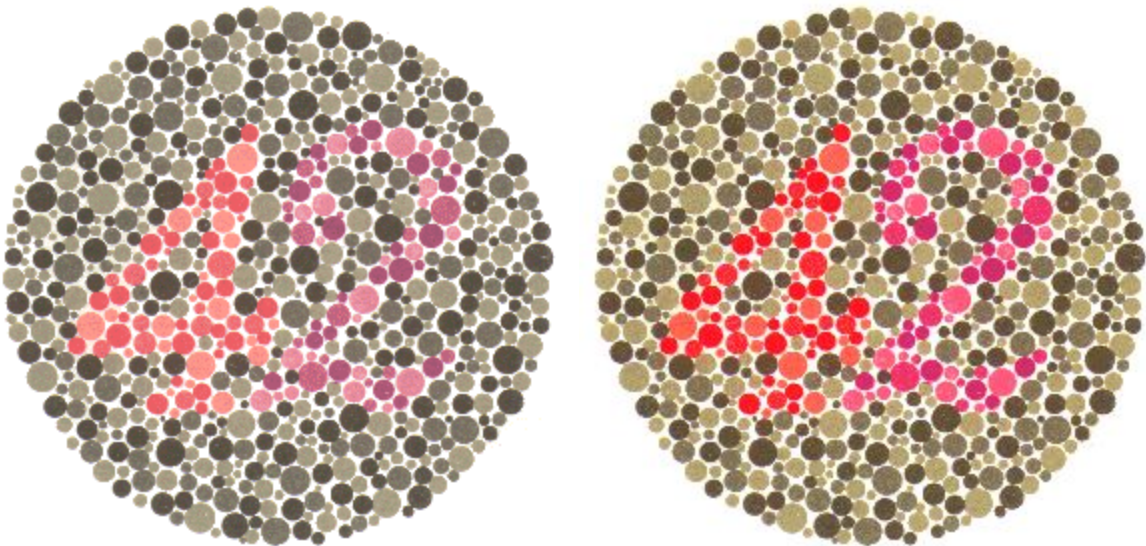
Nous allons maintenant nous intéresser à la saturation.

Rétablissement de la saturation

Afin d'étudier la saturation nous allons utiliser les images originales des cas 1 et 2 ainsi que les images modifiées suivantes:

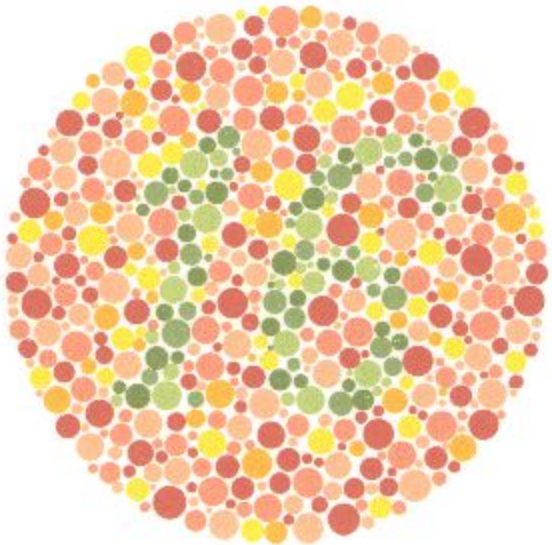
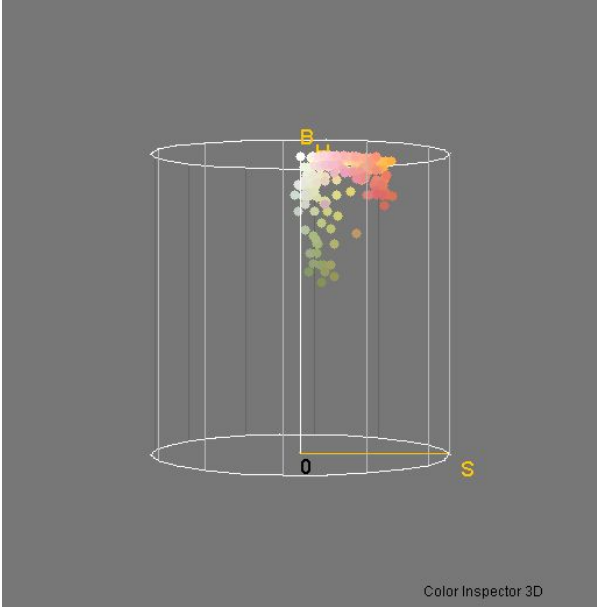
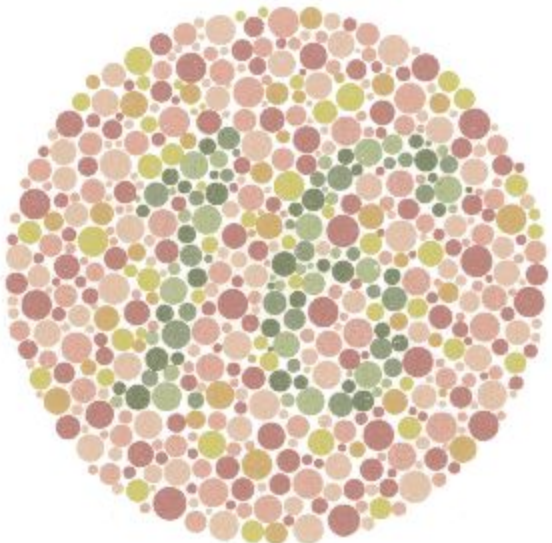
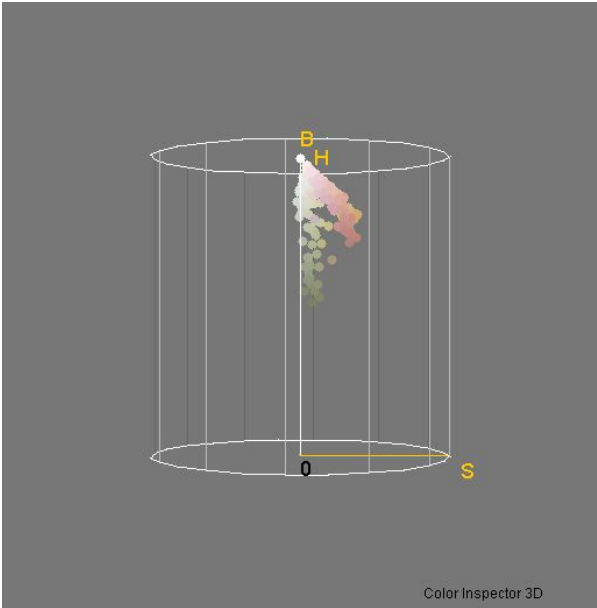


Cas 2 Image originale(gauche), Modifiée (droite)

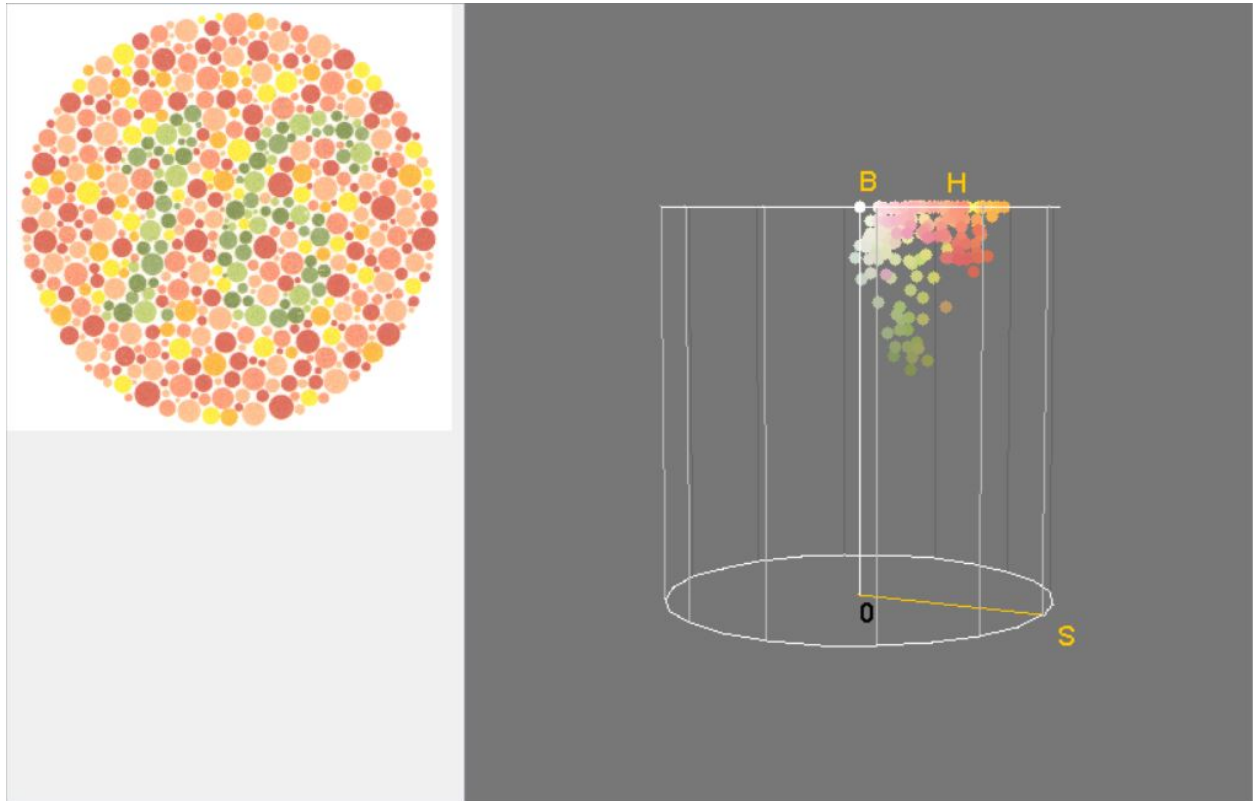


Cas 1 Image originale (gauche), Modifiée (droite)

Nous avons utilisé l'inspecteur 3D pour observer la différence entre les images modifiés et les images d'origine.

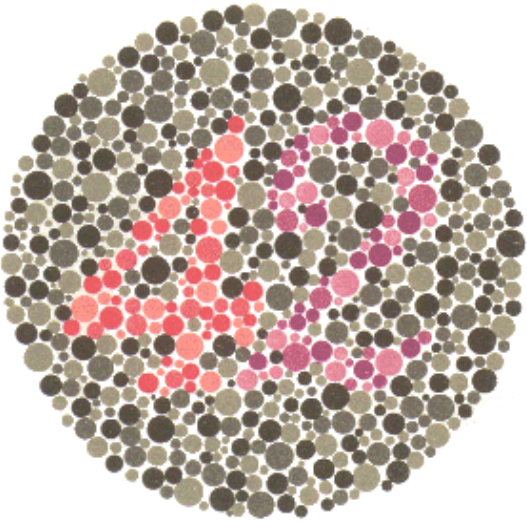
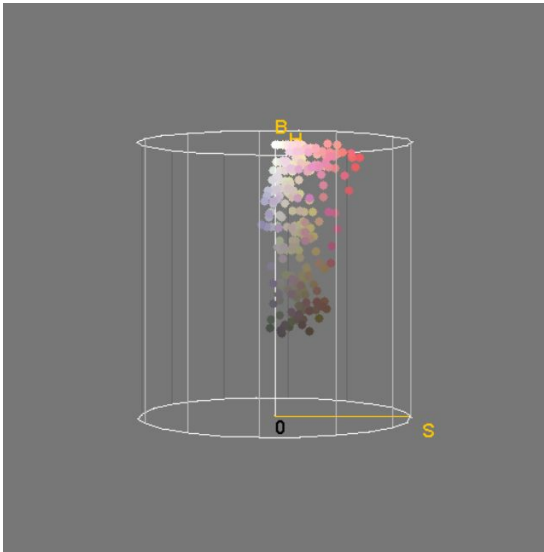
Image	Visualisation (HSB)
 <p data-bbox="399 877 602 909">Cas 2 Originale</p>	 <p data-bbox="1247 898 1369 909">Color Inspector 3D</p>
 <p data-bbox="399 1522 602 1554">Cas 2 Modifiée</p>	 <p data-bbox="1247 1543 1369 1554">Color Inspector 3D</p>

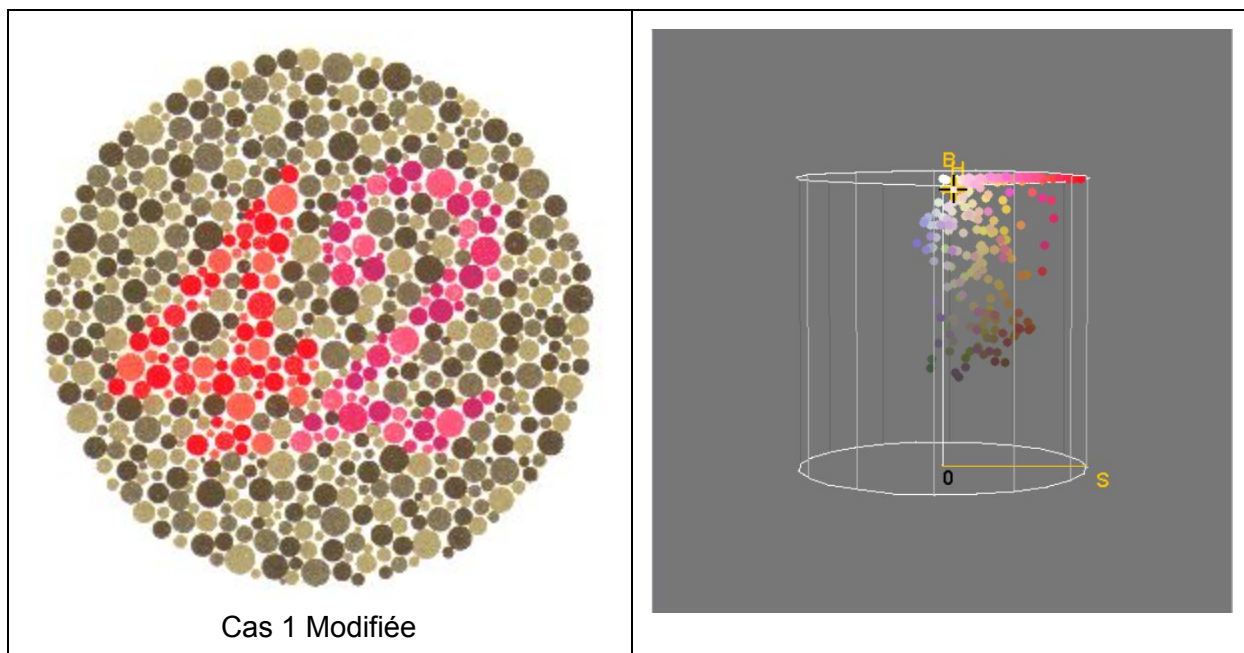
En observant les deux images on remarque que l'image modifiée a subi une diminution de saturation. Nous pouvons ainsi multiplier la saturation de l'image modifier jusqu'à avoir un resultat satisfaisant ressemblant fortement à l'image originale. Nous arrivons ainsi à un coefficient multiplicateur de 2.04 pour le résultat suivant:



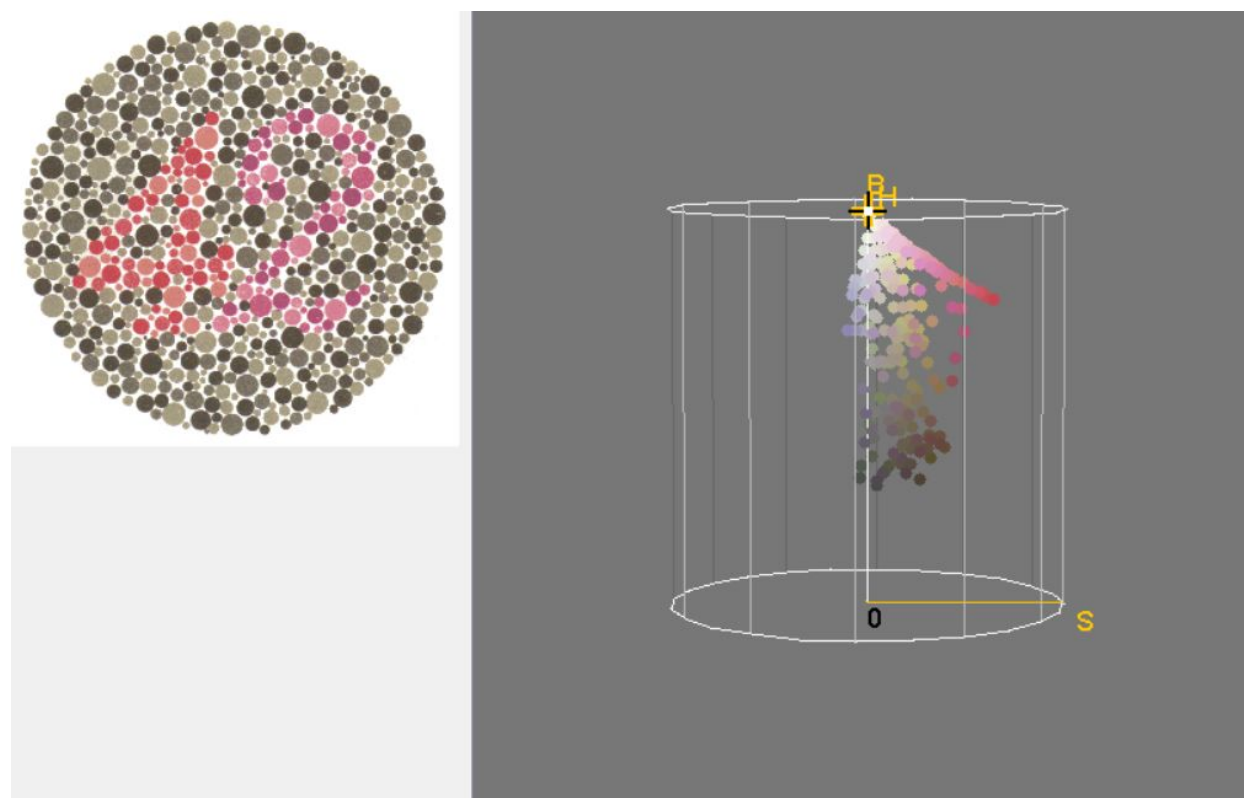
Nous allons maintenant nous intéresser au cas 1.

Nous avons dans ce cas les images et les répartitions dans l'espace HSB suivants:

Image	Visualisation
 <p data-bbox="397 1837 604 1871">Cas 1 Originale</p>	



Dans ce cas on s'aperçoit que l'image modifiée a subi une augmentation de saturation par rapport à l'image originale. Nous pouvons ainsi diviser la saturation de l'image modifier jusqu'à avoir un résultat satisfaisant ressemblant fortement à l'image originale. Nous arrivons ainsi à un coefficient multiplicateur de 0.5 pour le résultat suivant:



On peut quand même voir à l'oeil qu'en corrigeant la saturation, l'image modifiée n'est pas parfaitement égale à l'image d'origine. Mais cela semble quand même satisfaisant.

Nous avons ensuite écrit une macro permettant de convertir une image en HSB afin de d'appliquer un coefficient multiplicateur sur la saturation et ensuite de la convertir en RGB pour visualiser le résultat.

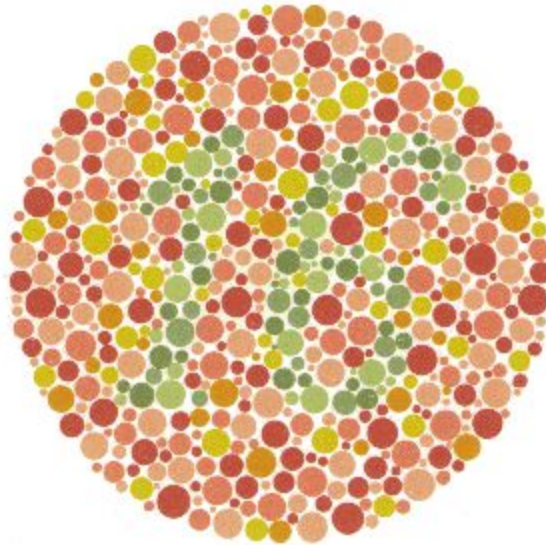
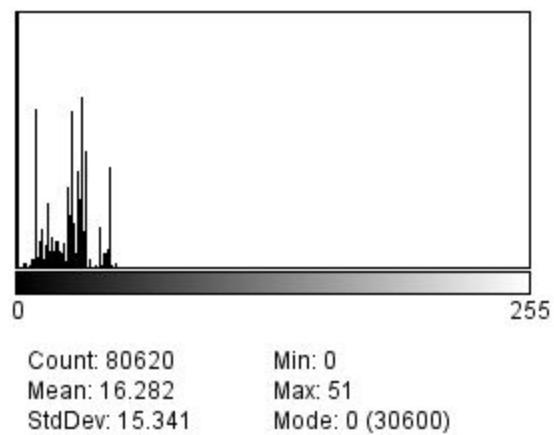
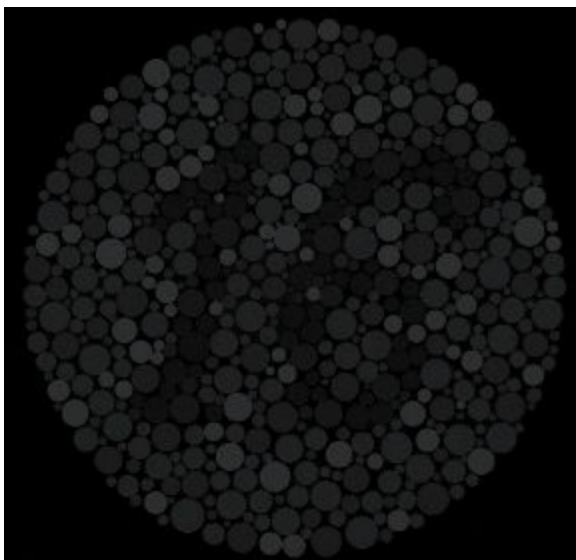


Image Reconstituée



Analyse de la différence entre la reconstruction et l'originale

On peut s'apercevoir qu'à l'oeil et même sur l'histogramme les images ne sont pas identiques cela est expliqué par le fait que nous divisons l'image en 3 channels Rouge/Vert et Bleu et pour altérer la saturation nous alterons le canal vert et nous perdons ainsi une information de

chrominance de l'image ce qui explique les différences que nous n'avons pas avec l'inspecteur 3D. Cependant on peut voir que l'on s'est rapproché tout de même de l'originale car l'histogramme est "serré" sur la gauche vers les valeurs de différences nulles.

Nous avons fait la même chose pour le cas 1 et nous obtenons les résultats suivants:

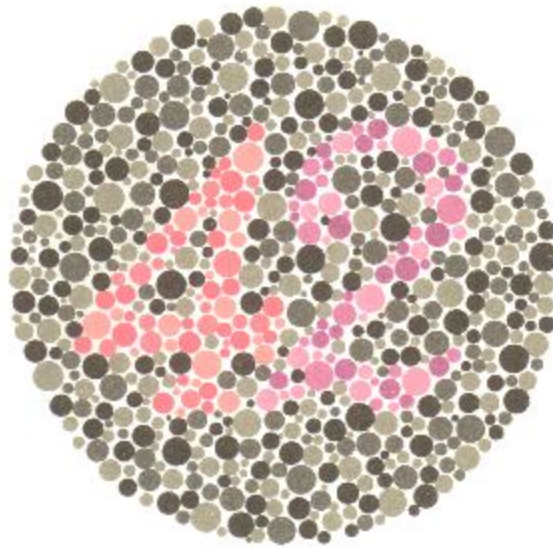
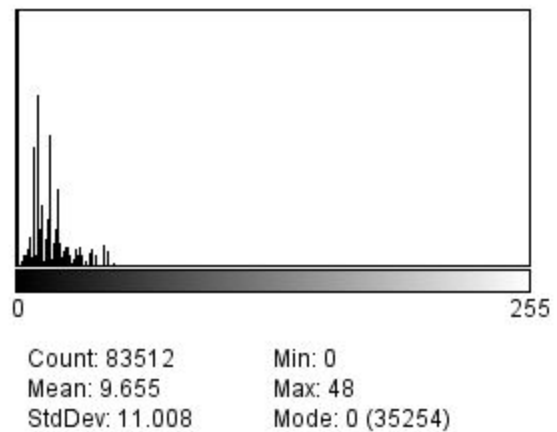
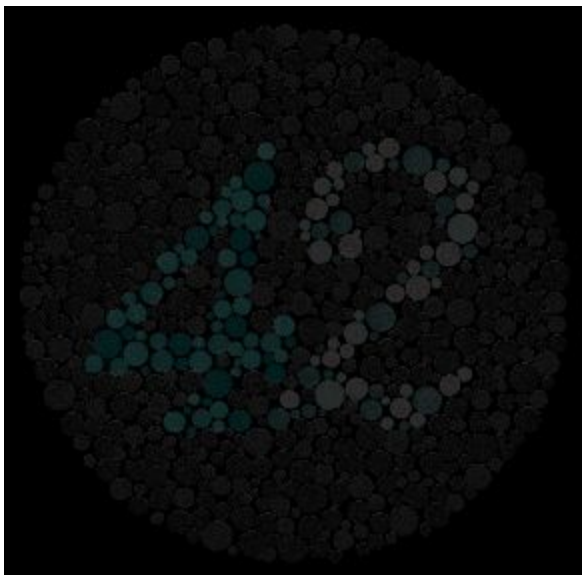


Image reconstruite



Analyse de la différence entre l'image reconstruite et l'originale

On peut voir grâce à l'histogramme que le résultat est tout de même plus probant que pour le cas 2 car la moyenne des différences est plus satisfaisante, 9.65 contre 16.28 pour le cas 2.

Nous allons maintenant nous intéresser à la teinte (Hue) de l'image.

Macro:

```
image = getImageID();
valeur = getNumber ("coefficient multiplicateur", valeur);

Dialog.create("Debut");
Dialog.addMessage("Cliquer sur Ok pour commencer le traitement");
Dialog.show();

setBatchMode(true);

titre=getTitle();
print (titre);
run("Color Space Converter", "from=RGB to=HSB white=D65");
run("Split Channels");
command = titre+" (HSB) (green)";
selectWindow(command);
run("Multiply...", "value=" + valeur);
command = "c1=["+titre+" (HSB) (red)] c2=["+titre+" (HSB) (green)] c3=["+titre+" (HSB) (blue)] ignore"
run("Merge Channels...", command);
run("Color Space Converter", "from=HSB to=RGB white=D65");

setBatchMode(false);

Dialog.create("Fin");
Dialog.addMessage("Cliquez sur OK pour terminer le traitement");
Dialog.show();
```


Transformation de la teinte

Dans cette partie nous allons altérer la teinte d'une image dont la teinte a préalablement été modifiée afin de retrouver l'image originale.

Nous travaillerons sur le couple d'image suivant:

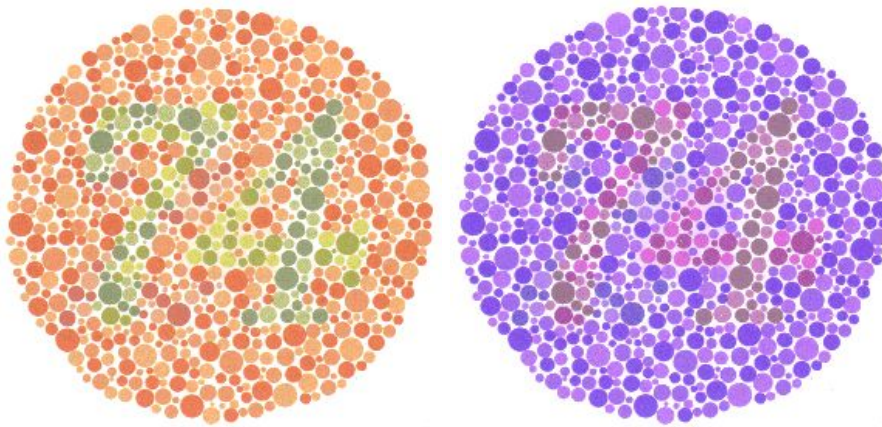
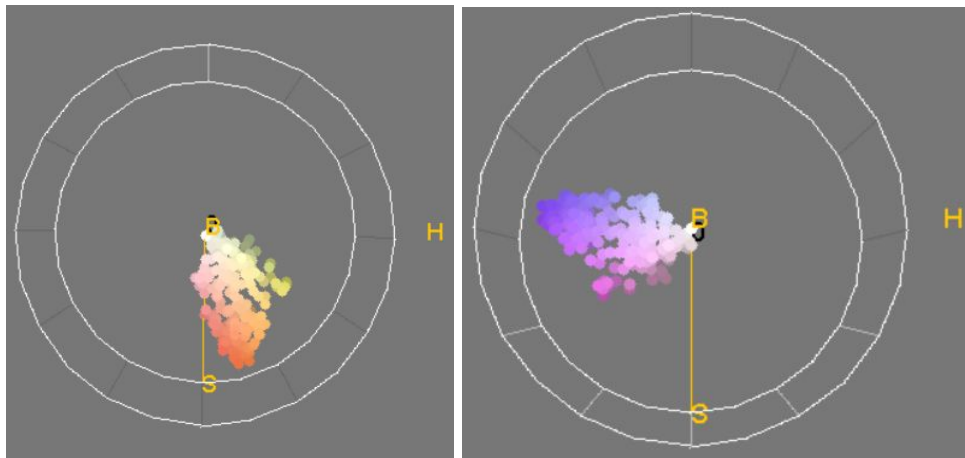


Image originale (gauche), Modifiée (droite)



Pour ce faire nous allons convertir l'image en HSB, séparant les canaux RGB, appliquer la transformation de la teinte, reformer les canaux RGB et convertir l'image de HSB vers RGB.

Pour plus de confort, la valeur à entrer est en degrés.

Ceci est fait par la macro suivante:

```

image = getImageID();
valeur = getNumber ("coefficient de teinte", valeur);

Dialog.create("Debut");
Dialog.addMessage("Cliquer sur Ok pour commencer le traitement");
Dialog.show();

setBatchMode(true);

titre=getTitle();
print (titre);
run("Color Space Converter", "from=RGB to=HSB white=D65");
run("Split Channels");
command = titre+" (HSB) (red)";
selectWindow(command);
valeur = valeur * 256 / 360; // conversion en degres
run("Add...", "value=" + valeur);
command = "c1=["+titre+" (HSB) (red)] c2=["+titre+" (HSB) (green)] c3=["+titre+" (HSB) (blue)] ignore"
run("Merge Channels...", command);
run("Color Space Converter", "from=HSB to=RGB white=D65");

setBatchMode(false);

Dialog.create("Fin");
Dialog.addMessage("Cliquez sur OK pour terminer le traitement");
Dialog.show();

```

Après tâtonnement, nous avons trouver que la valeur de rotation de la couleur la plus satisfaisante et donnant un très bon résultat est -234 degrés.

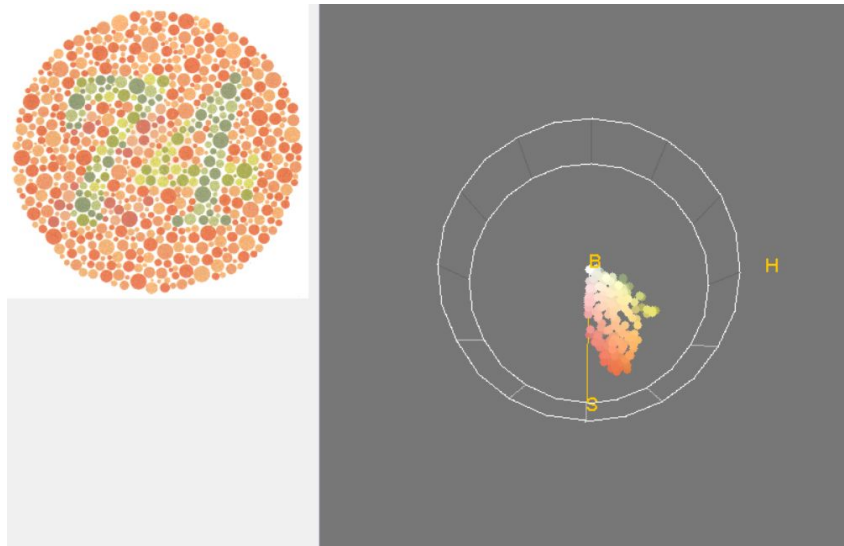
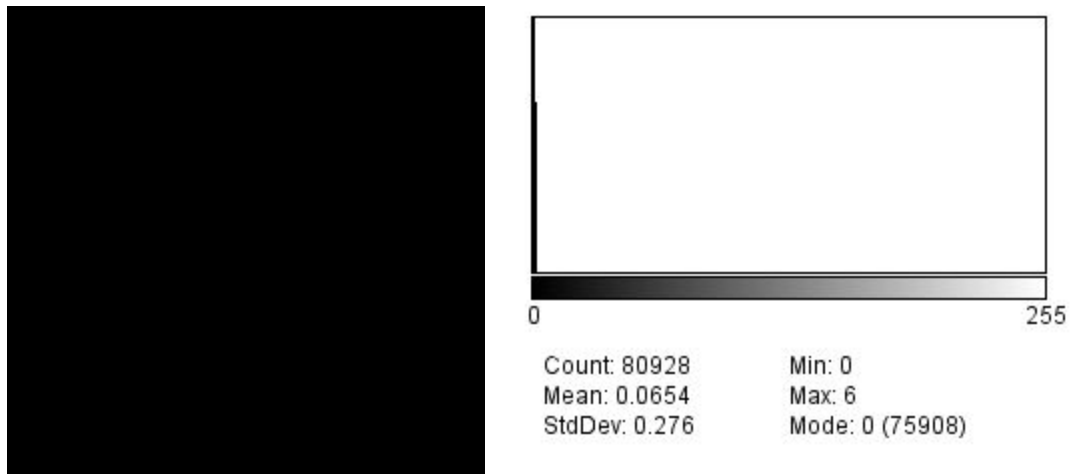


Image reconstruite

Analysons maintenant la différence entre cette image et l'image d'origine.

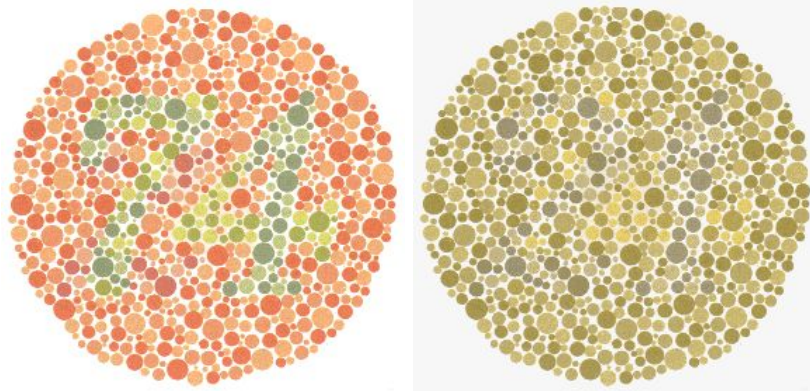


Différence entre l'image reconstruite et l'originale

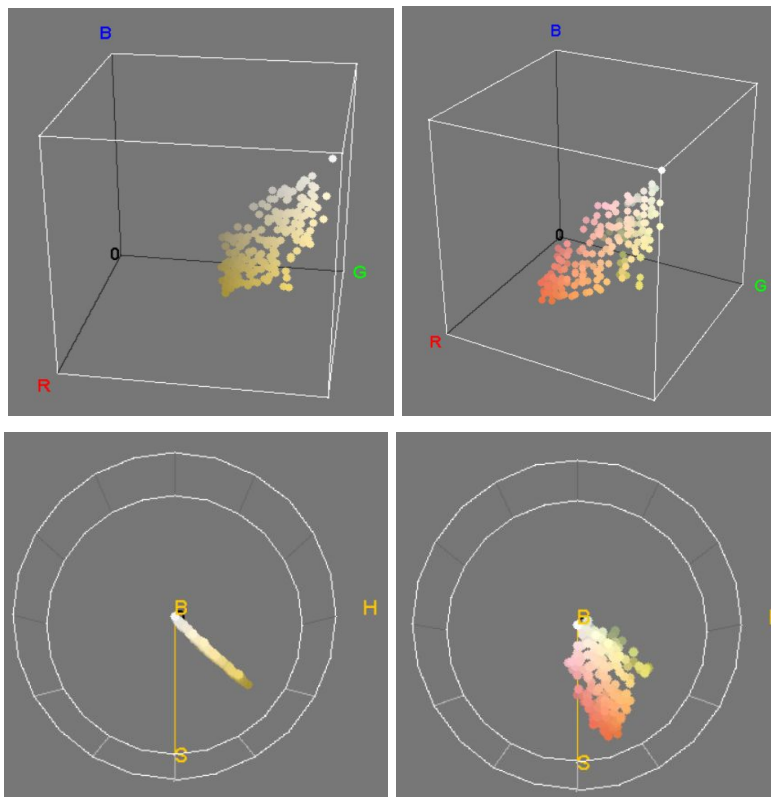
On voit sur l'histogramme que le résultat est très satisfaisant en regardant la moyenne de la différence qui est seulement de 0.065

Analyse dans des espaces couleurs adaptés

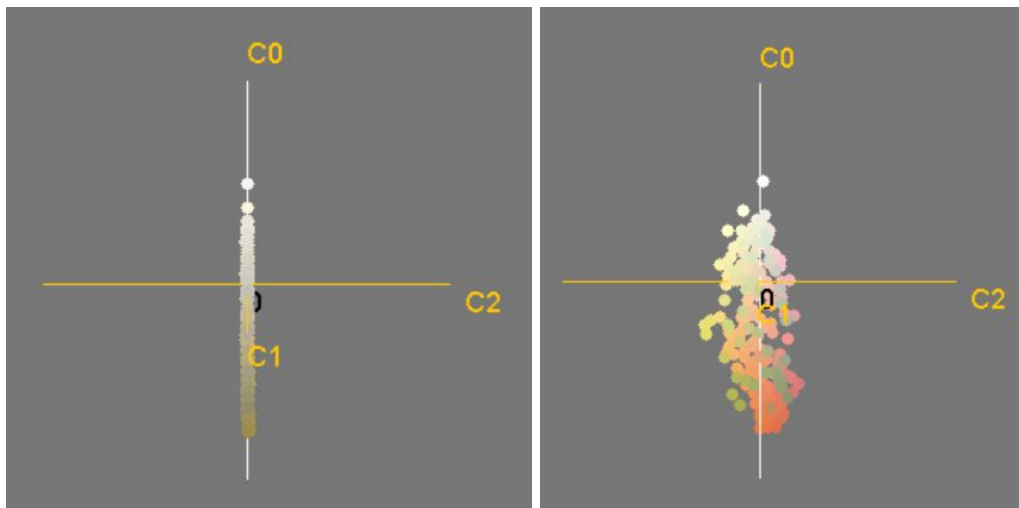
Nous allons étudier la différence entre ces deux images



En analysant les deux images dans l'espace de couleur RGB on se rend compte que la deuxième image Les valeurs sont projetées sur le plan suivant l'axe achromatique du cube représentant l'espace RGB. Ainsi, dans la deuxième image on a l'impression de voir le chiffre 21 et non 74. Il est aussi plus difficile de décèler le chiffre du fait la conséquence de ceci est que l'ensemble des pixels sont regroupés dans la même teinte de l'espace HSB.



On peut aussi voir la différence en regardant l'espace KLT/PCA dans lequel on voit très clairement que la deuxième image est le résultat de la multiplication de la valeur sur l'axe C2 par 0. On projette ainsi les valeurs sur le plan (C0,C1).



Modification de la luminance adaptée

Nous allons étudier la différence entre ces deux images

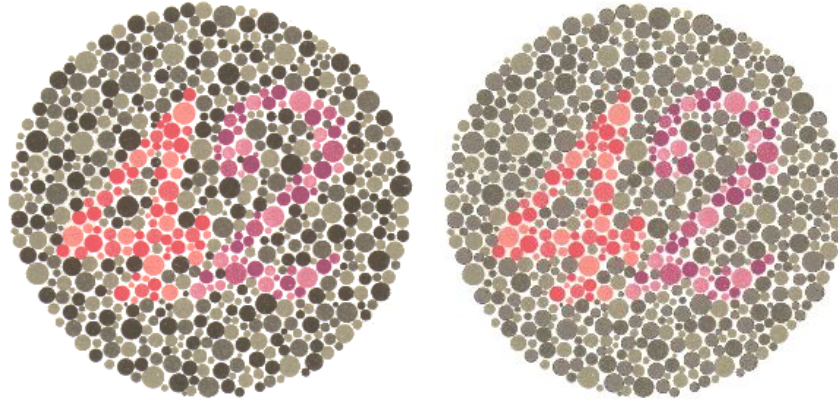
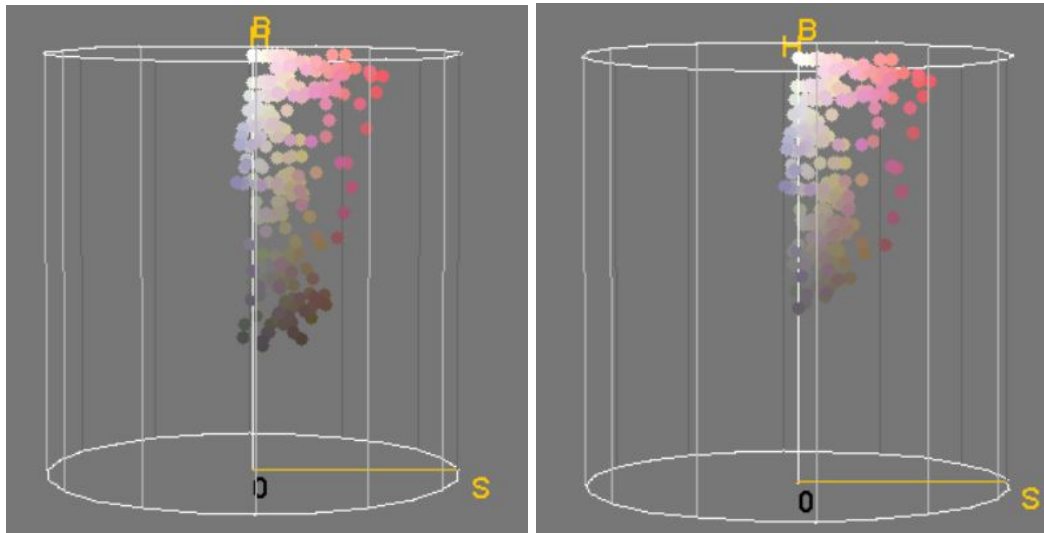


Image originale (gauche), modifiée (droite)

Dans l'image modifiée on peut voir que la luminance est plus faible que dans l'image originale mais ne résulte pas d'une addition simple d'une valeur à la luminance. En effet on peut s'apercevoir sur la représentation HSB que le nuage de pixels n'est ni juste monté, ni juste descendu, au lieu de ça il s'est plutôt "tassé". On peut donc en déduire que la luminance a été multipliée par une constante au lieu d'être additionné à celle-ci comme dans la première partie de ce TP.



Visualisation HSB: originale (gauche), modifiée (droite)

Ainsi on obtient la luminance en appliquant les équations suivantes:

$$R_{modif} = R_{orig} * \phi$$

$$G_{modif} = G_{orig} * \phi$$

$$B_{modif} = B_{orig} * \phi$$

Conclusion

Nous avons vu et utilisé plusieurs moyen permettant de transformer les informations d'une image couleur, notamment HSB (Hue, Saturation, Brightness). Nous avons pu constater que l'on peut faire beaucoup de choses en manipulant seulement 3 paramètres, mais aussi qu'il y a parfois des pertes d'informations irréversibles ne permettant pas de retrouver l'image originale.

De plus nous avons pu nous familiariser avec des espaces couleurs différents à travers l'outil "Color Inspector 3D" nous permettant de modifier en temps réel les valeurs des différents espaces couleur.