

# TP 1 : Modification d'images couleur

Francois LEPAN

18 septembre 2013

## 1 Manipulation de la luminance

### 1.1 Décrire la différence entre les distributions dans l'espace adéquat des couleurs présentes au sein des images *it1\_72pp* et *it1\_72pp\_sombre*

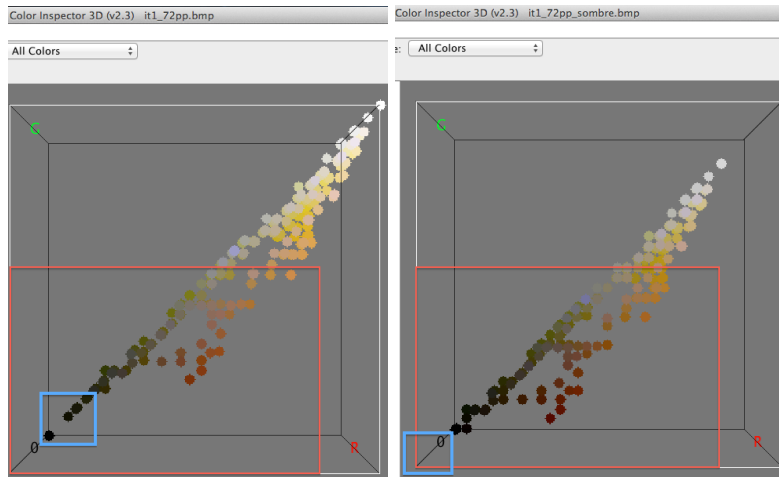


FIGURE 1 – Deux images inspecter via le plugin Color inspector 3D. On retrouve à gauche l'image *it1\_72\_pp.bmp* et à droite l'image *it1\_72pp\_sombre.bmp*

On remarque sur la Fig. 1 que la distribution est plus concentré vers le noir  $(0,0,0)$  sur l'axe achromatique pour l'image *it1\_72pp\_sombre.bmp*. Ceci est normale vu que cette image est plus sombre que l'image *it1\_72pp.bmp*. Plus précisément il y a eu une translation sur l'axe achromatique vers le noir  $(0,0,0)$  on voit bien qu'il y a eu une perte d'information entre l'image de gauche et l'image de droite (voir cadre bleu).

## 1.2 Macro modifiant la luminance

```
macro "augmentation_luminance" {
image = getImageID();

valeur = getNumber ("quelle augmentation (absolue) de luminance [0-255]",valeur);

while (valeur > 255 && valeur >= 0) {
    valeur = getNumber ("attention !! juste entre [0-255]",valeur);
}

setBatchMode(true);

W = getWidth();
H = getHeight();

run("Duplicate...", "title=luminance_augmente_de_"+valeur);
image_luminance_aug = getImageID();

max_1 = 0;
i_max_1 = 0;
j_max_1 = 0;

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 = minOf(R_avant + valeur, 255);
        G_apres = minOf(G_avant + valeur, 255);
        B_apres = minOf(B_avant + valeur, 255);

        couleur_apres = ((R_apres & 0xff) << 16) + ((G_apres & 0xff) << 8) + B_apres & 0xff;

        selectImage (image_luminance_aug);
        setPixel(i,j,couleur_apres);
    }
}

setBatchMode(false);
}
```

### 1.3 Valeur de $\phi$ qui donne le résultat le plus satisfaisant

Après plusieurs essais on trouve que pour une valeur de  $\phi = 40$  l'image résultante est la plus satisfaisante (*cf.* Fig. 2).

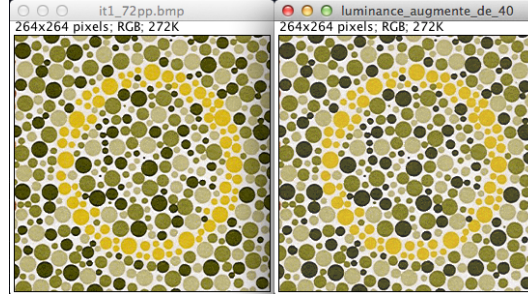


FIGURE 2 – À gauche l'image *it1\_72pp.bmp* et à droite l'image résultante pour  $\phi = 40$

### 1.4 Est-ce que l'image résultante est égale à l'image originale ?

Non car on fait une estimation. Comme expliqué à la question 1 on a perdu de l'information donc on ne fait qu'estimer les potentiels valeurs perdu.

## 2 Rétablissement de la saturation

### 2.1 Décrire la différence entre les distributions dans l'espace adéquat des couleurs présentes au sein des images *it2\_72pp* et *it2\_72pp-gris*

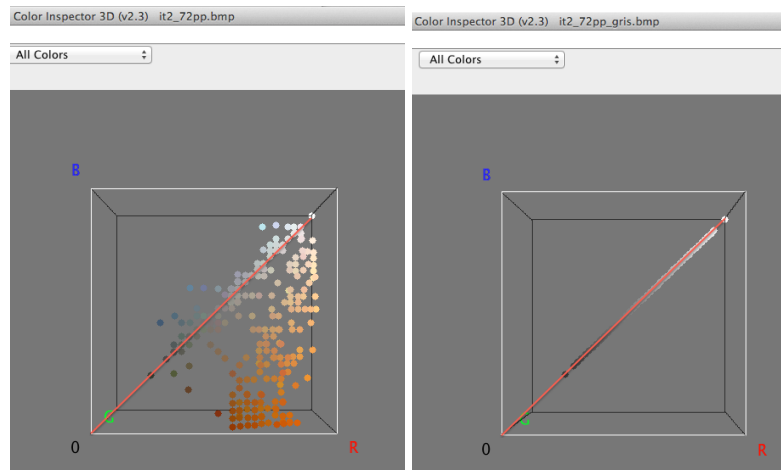


FIGURE 3 – Deux images inspecter via le plugin Color inspector 3D. On retrouve à gauche l'image *it2\_72pp.bmp* et à droite l'image *it2\_72pp-gris.bmp*

On remarque sur la Fig. 3 que la distribution des couleurs est exclusivement concentré sur l'axe achromatique pour l'image *it2\_72pp\_gris.bmp*.

## 2.2 Peut-on à partir de *it2\_72pp\_gris* retrouvé *it2\_72pp* ?

Non car on a perdu toutes les informations concernant les 3 couleurs RGB pour chaque pixels.

## 2.3 Décrire la différence entre les distributions dans l'espace adéquat des couleurs présentes au sein des images *it2\_72pp* et *it2\_72pp\_saturation*

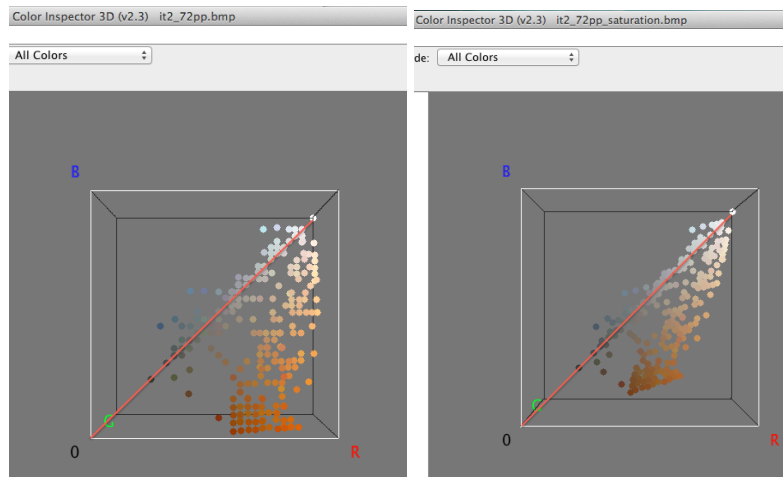


FIGURE 4 – Deux images inspecter via le plugin Color inspector 3D. On retrouve à gauche l'image *it2\_72\_pp.bmp* et à droite l'image *it2\_72pp\_saturation.bmp*

On remarque sur la Fig. 4 que la distribution est plus concentré sur l'axe achromatique pour l'image *it2\_72pp\_saturation.bmp*. Il y a juste une différence d'amplitude, l'image de droite saturé a une amplitude de "1" et celle de gauche en possède une supérieure. La macro suivante nous permettra donc de retrouver la même image *it2\_72\_pp.bmp* à partir de *it2\_72pp\_saturation.bmp*.

## 2.4 Macro

```
macro "changement_saturation" {

// recuperation du ID de l'image
image = getImageID();

alpha = getNumber ("quelle valeur pour alpha ? [0-1]",alpha);

while (alpha > 1 || alpha < 0) {
    alpha = getNumber ("attention !! juste entre [0-1]", alpha);
}

// on rajoute 1 car on veut augmenter et non abaisser "l'amplitude"
// des couleurs comme explique dans la question precedente
alpha += 1;

setBatchMode(true);

W = getWidth();
H = getHeight();

run("Duplicate...", "title= alpha = "+alpha);
image_luminance_aug = getImageID();

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) ;

        Y = (R_avant + G_avant + B_avant)/3;

        R_apres = Y + alpha*(R_avant - Y);
        G_apres = Y + alpha*(G_avant - Y);
        B_apres = Y + alpha*(B_avant - Y);

        couleur_apres = ((R_apres & 0xff) << 16) + ((G_apres & 0xff) << 8) + B_apres & 0xff;

        selectImage (image_luminance_aug);
        setPixel(i,j,couleur_apres);
    }
}
setBatchMode(false);
}
```

## 2.5 Valeur de $\alpha$ qui donne le résultat le plus satisfaisant

Après plusieurs essais on trouve que pour une valeur de  $\alpha = 1.7$ .

## 2.6 Est-ce que l'image résultante est égale à l'image originale ?

Oui car la seule différence entre l'image *it2\_72pp.bmp* et *it2\_72pp\_saturation.bmp* est "l'amplitude" des couleurs sur la représentation 3D. Tout ce que nous avons fait c'est augmenter cette "amplitude".

## 3 Décrire la différence entre les distributions dans l'espace adéquat des couleurs présentes au sein des images *it3\_72pp* et *it3\_72\_sans\_5*

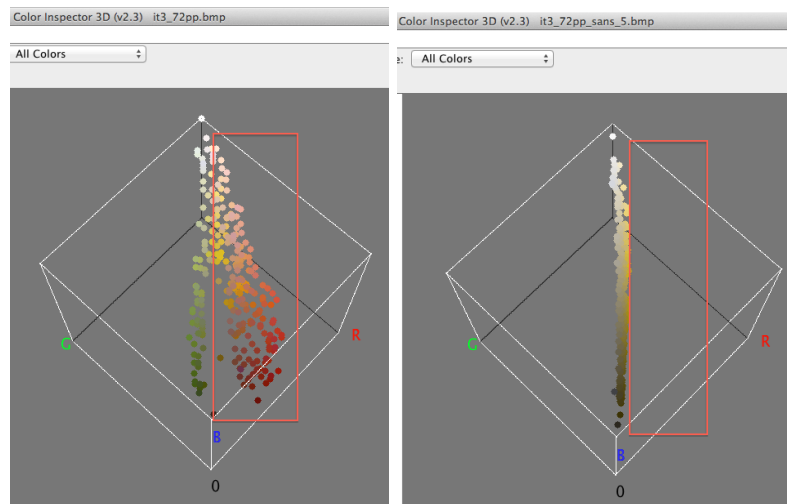


FIGURE 5 – Deux images inspecter via le plugin Color inspector 3D. On retrouve à gauche l'image *it3\_72pp.bmp* et à droite l'image *it3\_72pp\_sans\_5.bmp*

On remarque sur la Fig. 5 que pour l'image *it3\_72pp\_sans\_5.bmp* il n'y a pas la couleur rouge du 5 dans la distribution des couleurs.

4 Décrire la différence entre les distributions dans l'espace adéquat des couleurs présentes au sein des images *it1\_72pp* et *it1\_72pp\_sans\_cercle*

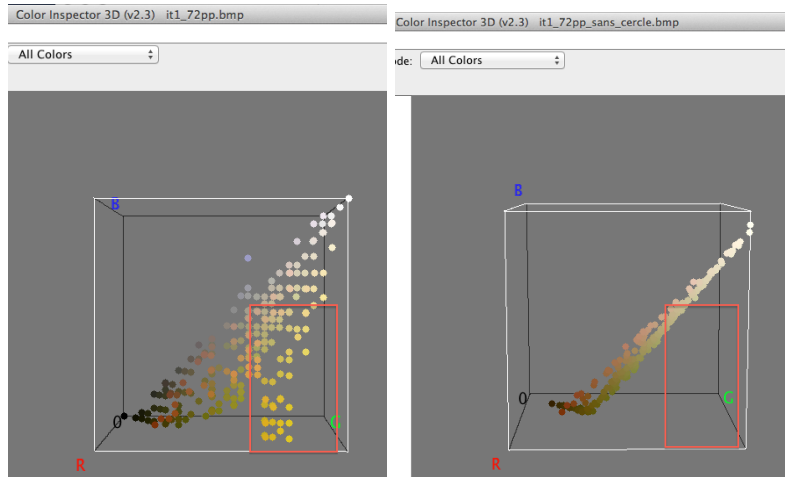


FIGURE 6 – Deux images inspecter via le plugin Color inspector 3D. On retrouve à gauche l'image *it1\_72pp.bmp* et à droite l'image *it1\_72pp\_sans\_cercle.bmp*

On remarque sur la Fig. 6 que pour l'image *it1\_72pp\_sans\_cercle.bmp* il n'y a pas la couleur jaune du cercle dans la distribution des couleurs.