

## Balance des blancs non linéaire

Gilles Fernandez, mise à jour du 9 juin 2014

---

### Introduction

Après étude d'un principe et réalisation d'un algorithme inspirés d'une publication de E. Y. LAM (référence), il s'avère que les résultats obtenus sont parfois intéressants mais assez aléatoires et quelquefois très décevants.

Ce document explore une autre approche reposant sur une transformation non linéaire moins contrainte. L'idée est de procéder en deux temps : d'abord, effectuer une transformation affine étirant l'histogramme de chaque canal de couleur pour avoir une dynamique maximale dans l'image ; puis effectuer une transformation non linéaire pour rendre les moyennes d'intensité de chaque canal égales à une valeur prescrite, en appliquant ainsi l'hypothèse du monde gris, mais en s'autorisant à choisir cette valeur prescrite pour pouvoir régler la luminosité globale de l'image.

### Définitions générales

L'objet de départ est une image de X pixels de largeur (numérotés de 0 à X-1) et Y pixels de hauteur (de 0 à Y-1). Le nombre total de pixels est alors  $N = X.Y$ .

Chaque pixel (x,y) est codé en RVB sur trois octets représentant le triplet d'intensités primaires

$$(I_R^0(x, y), I_V^0(x, y), I_B^0(x, y)) \quad ,$$

chacune appartenant à  $[0, 255]$ .

Si l'image comporte des pixels chauds perturbateurs, l'image de travail pourra être le résultat d'un seuillage éliminant les pixels les plus brillants. Après ce seuillage éventuel non considéré ici, l'image de travail est représentée par :

$$(I_R(x, y), I_V(x, y), I_B(x, y)) \quad .$$

Après la première transformation (affine) étirant le spectre, pour appliquer la balance des blancs, elle deviendra :

$$(I_R^t(x, y), I_V^t(x, y), I_B^t(x, y)) \quad ,$$

puis, après la seconde transformation (non linéaire) :

$$(I_R^T(x, y), I_V^T(x, y), I_B^T(x, y))$$

Chaque canal peut être traité indépendamment des autres. Dans le développement mathématique qui suit, on omet l'indice de couleur. La première transformation affine est évidente. Elle a la forme :

$$I^t(x, y) = C_1 \cdot I(x, y) + C_0$$

et doit vérifier, en posant  $I_{\max} = 255$  :

$$I_{\max} = C_1 \cdot \underset{x,y}{\text{Max}}(I(x, y)) + C_0 \quad \text{et} \quad 0 = C_1 \cdot \underset{x,y}{\text{Min}}(I(x, y)) + C_0 \quad .$$

d'où

$$C_1 = \frac{I_{\max}}{\left[ \underset{x,y}{\text{Max}}(I(x, y)) - \underset{x,y}{\text{Min}}(I(x, y)) \right]} \quad \text{et} \quad C_0 = -C_1 \cdot \underset{x,y}{\text{Min}}(I(x, y)) \quad .$$

Cette transformation ne s'applique donc pas à une image formée d'une seule teinte uniforme.

Ensuite, la transformation non linéaire doit être bien choisie. Elle doit conserver les deux

extrémités de l'histogramme. Une transformation quadratique est inadaptée car elle pourrait engendrer des couleurs négatives ou supérieures à  $I_{max}$  pour des forts déséquilibres de couleurs... On recherche une transformation qui permette d'atteindre tout le carré (0,0), (1,1) sans en déborder.

La fonction  $f(x) = x^n$  semble un candidat idéal du point de vue mathématique cependant elle implique une détermination numérique itérative de la transformation. En effet, l'inconnue du problème est dans ce cas l'exposant de la transformation, qui ne peut pas être obtenu par une résolution analytique. Le cas de cette fonction est étudié au chapitre II.

Pour bénéficier d'une résolution analytique, il est possible de choisir une cubique ; cette approche est étudiée en premier. La solution peut dans ce cas être explicite et directe. Elle est développée au chapitre I.

## CHAPITRE I – Transformation cubique

La formule choisie est donc une cubique, qui serait nécessairement de la forme suivante pour conserver les extrémités :

$$I^T(x, y) = I^t(x, y) + C_3 \cdot I^t(x, y) \cdot [I^t(x, y) - I_{max}] \cdot [I^t(x, y) - C_2]$$

et comporte deux coefficients à déterminer. Une équation provient de la moyenne imposée :

$$\sum_{x,y} I^T(x, y) = N I_{moy}$$

Une condition supplémentaire est que la transformation ne doit pas sortir de  $[0, I_{max}]$ . Ceci peut s'obtenir en travaillant sur les tangentes aux extrémités. La forme développée de la transformation est :

$$I^T = C_3 \cdot I^t{}^3 - C_3 \cdot (I_{max} + C_2) I^t{}^2 + [1 + C_3 \cdot C_2 I_{max}] I^t$$

Sa dérivée est donc :

$$I^T{}' = 3 C_3 \cdot I^t{}^2 - 2 C_3 \cdot (I_{max} + C_2) I^t + [1 + C_3 \cdot C_2 I_{max}]$$

A l'origine et à l'extrémité claire, elle vaut respectivement :

$$I^T{}' = 1 + C_3 \cdot C_2 I_{max} \quad \text{et} \quad I^T{}' = C_3 \cdot I_{max}^2 - C_3 \cdot C_2 I_{max} + 1$$

et ces deux valeurs doivent être positives ou nulles.

Au passage, la dérivée seconde de la transformation est :

$$I^T{}'' = 6 C_3 \cdot I^t - 2 C_3 \cdot (I_{max} + C_2)$$

Elle s'annule pour :  $I^t = \frac{I_{max} + C_2}{3}$  et cette valeur ne doit pas être dans l'intervalle  $[0, I_{max}]$

pour que la cubique n'y ait pas de point d'inflexion. Ceci pourrait se produire pour des grandes modifications de l'image et aurait pour conséquence de faire sortir des couleurs transformées de l'intervalle  $[0, 255]$ , impliquant ainsi un seuillage. Cette condition doit donc être évitée.

La moyenne imposée implique :

$$C_3 \cdot \sum_{x,y} I^t{}^3 - C_3 \cdot (I_{max} + C_2) \sum_{x,y} I^t{}^2 + [1 + C_3 \cdot C_2 I_{max}] \sum_{x,y} I^t = N I_{moy}$$

$$\text{soit : } C_3 \cdot \left[ \sum_{x,y} I^t{}^3 - I_{max} \sum_{x,y} I^t{}^2 \right] + \sum_{x,y} I^t - N I_{moy} = C_3 \cdot C_2 \left[ \sum_{x,y} I^t{}^2 - I_{max} \sum_{x,y} I^t \right]$$

$$\text{soit encore } C_3 \cdot C_2 = C_3 \cdot \left[ \frac{\sum_{x,y} I^t{}^3 - I_{max} \sum_{x,y} I^t{}^2}{\sum_{x,y} I^t{}^2 - I_{max} \sum_{x,y} I^t} \right] + \frac{\sum_{x,y} I^t - N I_{moy}}{\sum_{x,y} I^t{}^2 - I_{max} \sum_{x,y} I^t} = C_3 K_3 + K$$

en introduisant les deux constantes connues suivantes, toujours définies :

$$K_3 = \left[ \frac{\sum_{x,y} I^{t^3} - I_{max} \sum_{x,y} I^{t^2}}{\sum_{x,y} I^{t^2} - I_{max} \sum_{x,y} I^t} \right]$$

$$K = \frac{\sum_{x,y} I^t - N I_{moy}}{\sum_{x,y} I^{t^2} - I_{max} \sum_{x,y} I^t}$$

$K_3$  est positive (numérateur et dénominateur négatifs). Le signe de  $K$  dépend du choix de  $I_{moy}$ .

Les pentes initiale et finale valent donc :

- à l'origine :  $I^T' = 1 + I_{max} [C_3 K_3 + K] = 1 + K \cdot I_{max} + I_{max} K_3 C_3$
  - à l'extrémité claire :  $I^T' = C_3 \cdot I_{max}^2 - [C_3 K_3 + K] I_{max} + 1 = C_3 \cdot [I_{max}^2 - K_3 I_{max}] - K I_{max} + 1$
- et il faut écrire une condition sur elles. Il est possible de choisir d'avoir une symétrie géométrique approximative par rapport à la deuxième diagonale, en imposant que les deux pentes soient inverses :

$$[C_3 I_{max} \cdot (I_{max} - K_3) + (1 - K I_{max})][K_3 I_{max} C_3 + (1 + K \cdot I_{max})] = 1$$

et ceci conduit à une équation du second degré pour la variable  $x = I_{max} C_3$  :

$$[x \cdot (I_{max} - K_3) + (1 - K I_{max})][K_3 x + (1 + K \cdot I_{max})] = 1$$

soit

$$(I_{max} - K_3) K_3 x^2 + I_{max} [1 + K (I_{max} - 2 K_3)] x - K^2 I_{max}^2 = 0$$

La différence

$$K_3 - I_{max} = \left[ \frac{(I_{max} \sum_{x,y} I^{t^2} - \sum_{x,y} I^{t^3}) - I_{max} (I_{max} \sum_{x,y} I^t - \sum_{x,y} I^{t^2})}{I_{max} \sum_{x,y} I^t - \sum_{x,y} I^{t^2}} \right]$$

a le signe du numérateur de la fraction au second membre :

$$(I_{max} \sum_{x,y} I^{t^2} - \sum_{x,y} I^{t^3}) - I_{max} (I_{max} \sum_{x,y} I^t - \sum_{x,y} I^{t^2}) = I_{max} \sum_{x,y} I^{t^2} - \sum_{x,y} I^{t^3} - I_{max}^2 \sum_{x,y} I^t + I_{max} \sum_{x,y} I^{t^2}$$

$$\dots = - \sum_{x,y} I^t (I_{max} - I^t)^2 < 0$$

qui est strictement négatif si l'image n'est pas totalement noire ni totalement blanche. En conséquence, le coefficient de  $x^2$  dans l'équation est strictement positif et le produit des deux racines est négatif ou nul, si elles existent. Le discriminant de cette équation est :

$$\Delta = I_{max}^2 \left[ (1 + K (I_{max} - 2 K_3))^2 + 4 (I_{max} - K_3) K_3 K^2 \right]$$

soit

$$\Delta = I_{max}^2 \left( 1 + K^2 I_{max}^2 + 2 K (I_{max} - 2 K_3) \right) = I_{max}^2 \left( (1 + K I_{max})^2 - 4 K K_3 \right)$$

Considéré comme une fonction de  $K$ , ce discriminant a pour dérivée :

$$\frac{d\Delta}{dK} = 2 I_{max}^2 \left( K I_{max}^2 + (I_{max} - 2 K_3) \right)$$

qui est une fonction croissante de  $K$  donc le discriminant atteint un minimum pour :

$$K = K_{min} = \frac{2 K_3 - I_{max}}{I_{max}^2}$$

et ce minimum vaut :  $\Delta_{min} = 4 K_3 (I_{max} - K_3) > 0$

Ce discriminant est donc toujours positif. Les racines sont :

$$\{x_1, x_2\} = \frac{-I_{max}[1 + K(I_{max} - 2K_3)] \pm \sqrt{\Delta}}{2(I_{max} - K_3)K_3}$$

La pente à l'origine est alors :

$$I^{T'} = 1 + I_{max} \left( K - \frac{[1 + K(I_{max} - 2K_3)] \mp \sqrt{(1 + KI_{max})^2 - 4KK_3}}{2(I_{max} - K_3)} \right)$$

Aux deux valeurs de  $C_3$  correspondent deux valeurs de :  $C_2 = K_3 + \frac{K}{C_3}$ , la pente à l'origine s'exprimant aussi par  $I^{T'} = 1 + I_{max} C_2 C_3$ , avec deux valeurs possibles.

En tirant parti de :

$$x_1 + x_2 = -\frac{I_{max}[1 + K(I_{max} - 2K_3)]}{(I_{max} - K_3)K_3}, x_1 x_2 = \frac{-K^2 I_{max}^2}{(I_{max} - K_3)K_3}$$

il est facile de montrer que le produit des deux valeurs possibles pour la pente à l'origine est :

$$(I^{T'})_1 (I^{T'})_2 = -\frac{K_3}{I_{max} - K_3}$$

Ce produit étant toujours négatif, une seule des deux pentes est positive et peut correspondre à une solution viable pour la transformation.

Si cette valeur positive a une pente inférieure à 1, elle correspond à une intensité moyenne inférieure à l'image de départ, sinon elle est associée à la solution qui donne une intensité moyenne supérieure à celle de départ.

Les éléments qui précèdent permettent de déterminer la transformation.

### Récapitulation

Quelle que soit l'image de départ (ne possédant pas un canal primaire réduit à une teinte uniforme), il est possible d'opérer dessus une balance des blancs selon l'approche qui précède. Il est en outre possible de régler simultanément la luminosité moyenne de l'image résultante, à condition de ne pas demander une modification trop forte qui ferait sortir la transformation non linéaire de l'intervalle [0, 255].

Il faut commencer par effectuer l'étirement de l'histogramme par la transformation affine :

$$I'(x, y) = C_1 \cdot I(x, y) + C_0$$

avec

$$C_1 = \frac{I_{max}}{\left[ \underset{x,y}{\text{Max}}(I(x, y)) - \underset{x,y}{\text{Min}}(I(x, y)) \right]} \quad \text{et} \quad C_0 = -C_1 \cdot \underset{x,y}{\text{Min}}(I(x, y))$$

Il faut noter que, opérée canal par canal, cette transformation ne garantit pas la présence d'un pixel blanc et d'un pixel noir dans l'image transformée car ce n'est pas forcément le même pixel qui porte les trois maxima R, V et B. En pratique, cependant, les pixels les plus clairs sont transformés dans des couleurs très proches du blanc et les plus sombres dans des couleurs très proches du noir.

Ensuite, il faut calculer les quantités suivantes :

$$K_3 = \frac{I_{max} \sum_{x,y} I^{t^2} - \sum_{x,y} I^{t^3}}{I_{max} \sum_{x,y} I^t - \sum_{x,y} I^{t^2}}$$

$$K = \frac{N I_{moy} - \sum_{x,y} I^t}{I_{max} \sum_{x,y} I^t - \sum_{x,y} I^{t^2}}$$

$$\delta = \sqrt{\Delta} = I_{max} \sqrt{\left( (1 + K I_{max})^2 - 4 K K_3 \right)}$$

$$x_1 = \frac{-I_{max} [1 + K (I_{max} - 2 K_3)] + \delta}{2 (I_{max} - K_3) K_3}; x_2 = \frac{-I_{max} [1 + K (I_{max} - 2 K_3)] - \delta}{2 (I_{max} - K_3) K_3}$$

$$C_{31} = \frac{-I_{max} [1 + K (I_{max} - 2 K_3)] + \delta}{2 (I_{max} - K_3) K_3 I_{max}}; C_{32} = \frac{-I_{max} [1 + K (I_{max} - 2 K_3)] - \delta}{2 (I_{max} - K_3) K_3 I_{max}}$$

$$C_{21} = K_3 + \frac{K}{C_{31}}; C_{22} = K_3 + \frac{K}{C_{32}} .$$

De ces coefficients se déduit la pente à l'origine, qui peut mathématiquement prendre deux valeurs :

$$I_1^T = 1 + I_{max} C_{21} C_{31}; I_2^T = 1 + I_{max} C_{22} C_{32} .$$

Ces deux pentes sont de signes contraires, seule la solution à pente positive est exploitable.

Les coefficients  $C_2$  et  $C_3$  ainsi déterminés, on applique la transformation.

### **Implémentation informatique**

Cette approche a donné lieu à la réalisation d'un module additionnel en python (« python-fu ») pour le GIMP, `balance-des-blancs-cubique.py`. Il met en œuvre les résultats exposés ci-dessus. L'intensité moyenne de l'image transformée est choisie par l'utilisateur au moyen d'un coefficient qui est un pourcentage positif ou négatif nommé  $p_{lum}$ .

Si ce coefficient est nul, la luminosité globale de l'image transformée est réglée à la même valeur que celle de l'image initiale. S'il est positif, la luminosité transformée est définie par :

$$I_{moy} = I_{initiale} + p_{lum} \cdot \alpha \cdot (I_{max} - I_{initiale}) \quad \text{et, s'il est négatif, de manière dissymétrique, par :}$$

$$I_{moy} = I_{initiale} + p_{lum} \cdot \alpha \cdot (I_{initiale} - 0) .$$

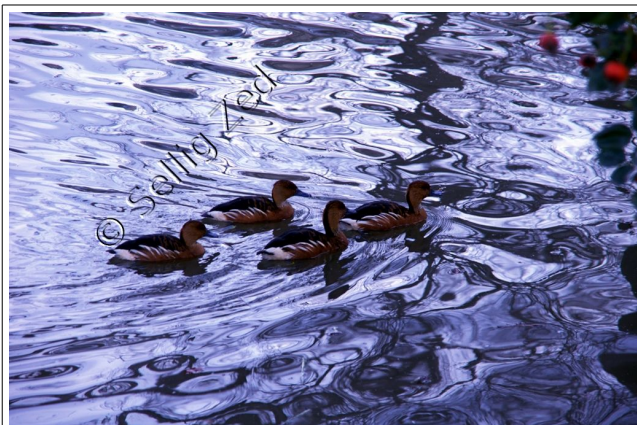
Dans ces égalités, le coefficient  $\alpha$  est un paramètre (variable *ampli-clair* du programme) inférieur à 1 qui contraint la variation exigible par l'utilisateur pour  $I_{moy}$ , afin que transformation ne soit pas exagérée. Si, malgré cette contrainte, des couleurs transformées sortent de l'intervalle  $[0, I_{max}]$ , elles sont écrêtées. Le seuillage est signalé à l'utilisateur et une nouvelle valeur de coefficient lui est suggérée ; en raison du caractère non linéaire de la transformation, il n'est cependant pas garanti que cette suggestion évite complètement le seuillage.

Par ailleurs, les coefficients des deux transformations affine et cubique sont identifiés en une seule passe sur l'image grâce aux identités suivantes :

$$I^t = C_1 \cdot I + C_0 \quad , \quad I^{t^2} = C_1^2 \cdot I^2 + 2 C_0 C_1 \cdot I + C_0^2 \quad , \quad I^{t^3} = C_1^3 \cdot I^3 + 3 C_0 C_1^2 \cdot I^2 + 3 C_0^2 C_1 \cdot I + C_0^3$$

utilisées pour calculer les sommes des premières transformées après la première passe, où seules sont calculées les sommes des puissances des intensités initiales. Le temps de calcul pouvant devenir notable pour de grosses images, il est recommandé de procéder aux essais sur des images réduites. La dernière version de la liste informatique est diffusée sous licence Cecill-C et téléchargeable sur <http://sellig.zed.myriapyle.net/Couleur/balablan.htm>.

Planche 1 : exemples de résultats obtenus avec la balance des blancs cubique



Avant



Après, avec coefficient de luminosité = 1.0



Avant



Après, avec coefficient de luminosité = 0.0



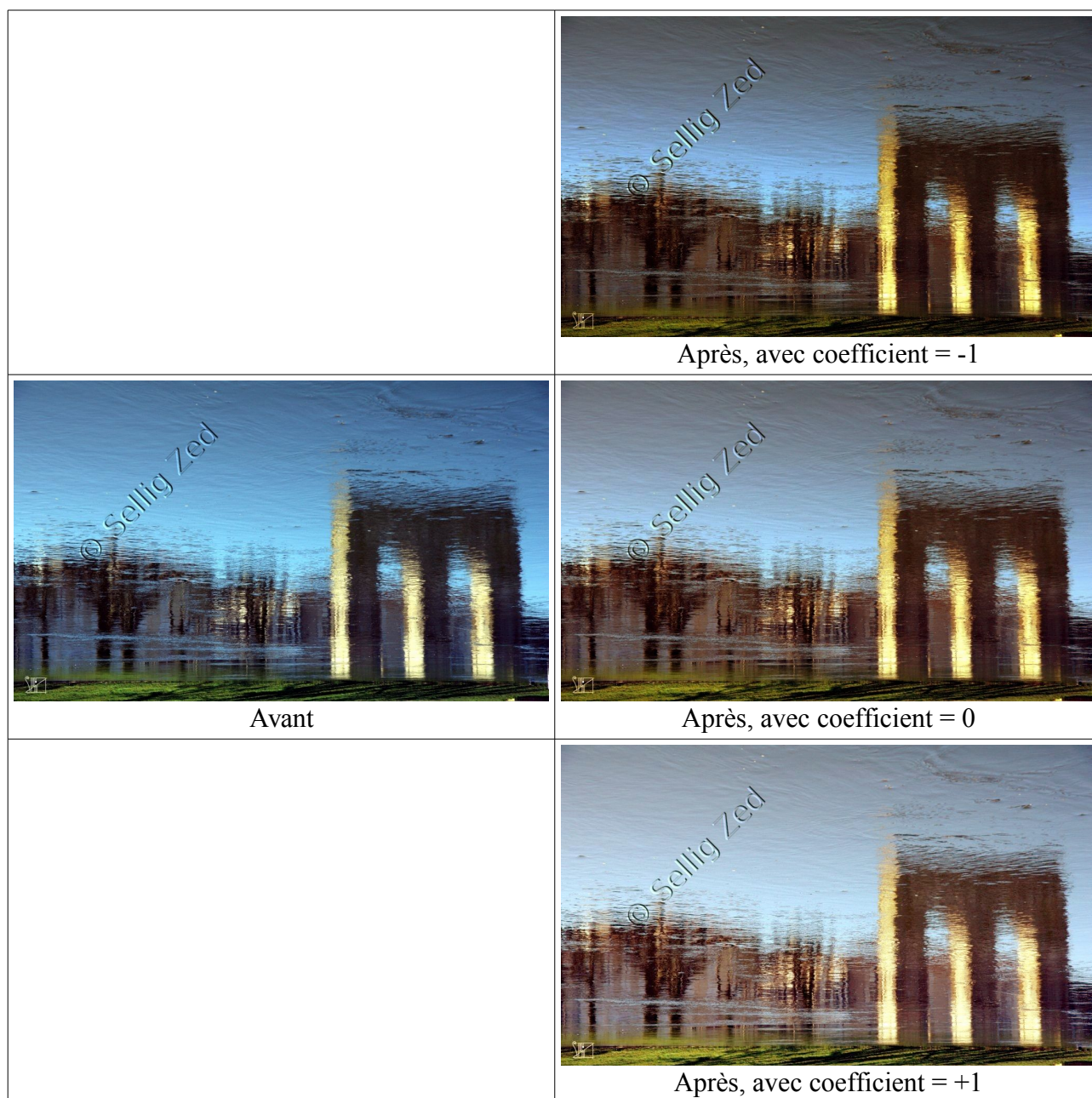
Avant



Après, avec coefficient de luminosité = - 1.0

L'élimination de dominante est effectuée de manière convaincante.

## Planche 2 : Influence du coefficient de luminosité avec la balance des blancs cubique



La latitude de réglage dépend de l'image de départ mais elle permet dans tous les cas d'obtenir une variation bien visible du rendu.

### **Conclusion**

La balance des blancs cubique présentée ici est applicable sans retouche à presque toutes les images et donne des résultats souvent satisfaisants. Dans quelques rares cas extrêmes et si l'utilisateur demande une forte modification de la luminosité, un seuillage peut s'avérer nécessaire et l'utilisateur en est avisé.

Quelques essais sont présentés ci-après par la planche 1 (page 6). Les coefficients de luminosité n'y sont pas sélectionnés d'un point de vue esthétique mais simplement choisis aux fins de test informatique. La planche 2 (page 7) illustre l'influence du paramètre de luminosité pour cette transformation.

## CHAPITRE II – Transformation monôme en loi de puissance n

Pour avoir une transformation qui marche à coup sûr, pour toutes les images, sans seuillage et avec une plus grande amplitude de réglage de la luminosité, il faut utiliser la loi  $x^n$  sur  $[0,1]$  (avec  $n > 0$ ) :

$$\frac{I^T}{I_{max}} = \left( \frac{I}{I_{max}} \right)^n \quad \text{soit} \quad I^T = I_{max}^{1-n} \cdot I^n .$$

La moyenne des pixels transformés est alors :

$$\bar{I} = \frac{I_{max}}{N} \sum_{x,y} \left( \frac{I}{I_{max}} \right)^n$$

or, sur tous les pixels,  $I$  ne prend que les valeurs entières de 0 à  $I_{max}$ . Par conséquent, si  $h(I)$  est le nombre de pixels portant la valeur  $I$ , c'est-à-dire le nombre de fois où la valeur  $I$  est rencontrée, la moyenne est aussi donnée par :

$$\bar{I} = \frac{I_{max}}{N} \sum_{I=0}^{I_{max}} h(I) \left( \frac{I}{I_{max}} \right)^n$$

ou bien encore, en introduisant le pourcentage  $p(I)$  de pixels portant la valeur  $I$  dans l'image non transformée, la moyenne transformée est une fonction décroissante de  $n$  donnée par :

$$p(I) = \frac{h(I)}{N} , \quad \bar{I} = I_{max} \sum_{I=0}^{I_{max}} p(I) \left( \frac{I}{I_{max}} \right)^n .$$

En notant  $\hat{I} = c I_{max}$  la moyenne cible, en choisissant  $c < 1$ , il s'agit alors de trouver l'inconnue  $x$  (réelle) telle que

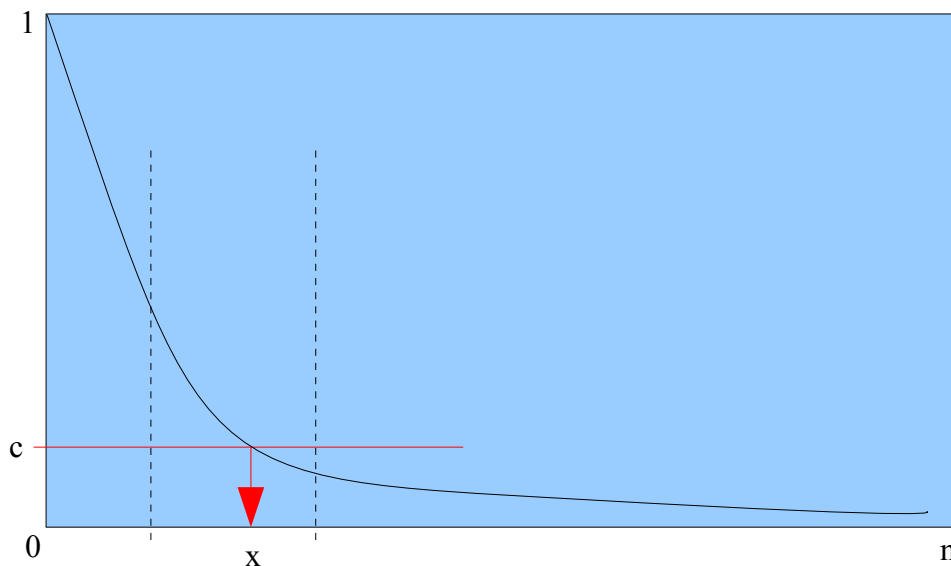
$$f(x) = c \quad \text{avec} \quad f(x) = \sum_{I=0}^{I_{max}} p(I) \left( \frac{I}{I_{max}} \right)^x .$$

Il est facile d'observer que

$$f(0) = \sum_{I=0}^{I_{max}} p(I) = 1 \quad \text{et} \quad f(1) = \sum_{I=0}^{I_{max}} p(I) \cdot (I/I_{max}) = I_0/I_{max} < 1 , \quad \text{où } I_0 \text{ est la moyenne initiale,}$$

et que  $f(x)$  a la décroissance monotone et rapide vers l'asymptote 0 d'une somme d'exponentielles.

L'équation a donc une solution unique dont une approximation peut être obtenue par dichotomie.





Le principe de la résolution par dichotomie est facile. En notant  $f(x)$  la fonction au premier membre de l'équation, on travaille classiquement avec trois valeurs :  $x$ ,  $x_{\text{inf}}$  et  $x_{\text{sup}}$  :

Au départ,  $x_{\text{inf}}$  et  $x_{\text{sup}}$  sont initialisées à 0 et  $x$  à 1.

a) Dans un premier temps, on effectue une étape d'encadrement, jusqu'à ce que les valeurs  $x_{\text{sup}}$  et  $x_{\text{inf}}$  soient différentes, avec la règle suivante :

si  $f(x) > c$ , on fait  $x_{\text{inf}} = x$ , puis on double  $x$  et on recommence ;

si  $f(x) < c$ , on fait  $x_{\text{sup}} = x$  sans changer  $x_{\text{inf}}$  car on a encadré et on fait  $x = (x_{\text{inf}} + x_{\text{sup}})/2$ .

b) Dans un second temps, on effectue l'étape de résolution par dichotomie, jusqu'à obtenir un écart suffisamment petit entre  $x_{\text{inf}}$  et  $x_{\text{sup}}$  :

si  $f(x) > c$ , on fait  $x_{\text{inf}} = x$ , puis  $x = (x_{\text{inf}} + x_{\text{sup}})/2$

sinon on fait  $x_{\text{sup}} = x$ , puis  $x = (x_{\text{inf}} + x_{\text{sup}})/2$

et on recommence.

Si le critère est atteint, on prend pour solution  $x = (x_{\text{inf}} + x_{\text{sup}})/2$ .

Ensuite on applique la transformation à l'image. L'utilisateur peut demander un compte rendu d'exécution qui s'enregistre à côté de l'image ; un exemple en est donné en fin de texte.

Avec cette transformation, le réglage de luminosité est beaucoup plus souple puisqu'il existe toujours une valeur de l'exposant qui permet de l'atteindre. Comme dans le cas cubique, l'utilisateur peut définir le paramètre de clarté par rapport à l'intensité initiale. Ainsi, si  $I_{\text{initiale}}$  est l'intensité initiale et  $\alpha$  (variable *amplclair* du programme) l'amplitude permise d'un côté ou de l'autre,

$$\hat{I} = I_{\text{moy}} + p_{\text{lum}} \cdot \alpha \cdot [(I_{\text{max}} - I_{\text{moy}}) \text{ ou bien } (I_{\text{moy}} - 0)] \text{ suivant le signe de } p_{\text{lum}}.$$

soit :

$$c = \frac{\hat{I}}{I_{\text{max}}} = \frac{I_{\text{moy}}}{I_{\text{max}}} + p_{\text{lum}} \cdot \alpha \cdot [(1 - \frac{I_{\text{moy}}}{I_{\text{max}}}) \text{ ou bien } (\frac{I_{\text{moy}}}{I_{\text{max}}} - 0)] .$$

## Conclusion

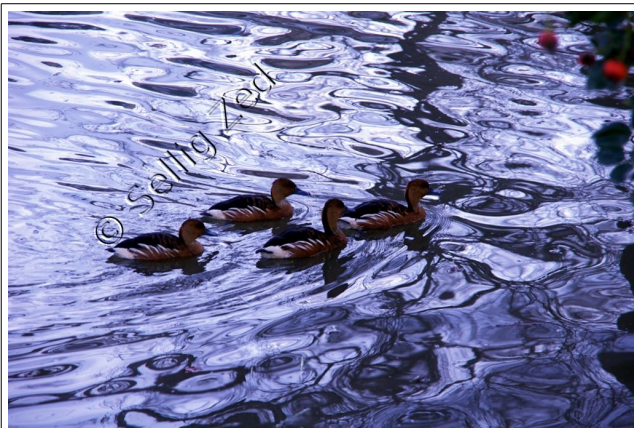
Plus simple analytiquement, cette approche reporte la difficulté sur la résolution numérique en obligeant à déterminer la transformation par itération. Elle demande donc un temps de calcul un peu supérieur à celui de la transformation cubique. La différence n'est cependant pas rédhibitoire et semble justifiée par la plus grande universalité du programme, par une amélioration de la qualité des résultats dans certains cas et par la plus grande amplitude de luminosité offerte à l'utilisateur. Les planches 3 (page 10) et 4 (page 11) présentent des résultats du programme. Ce dernier se présente sous la forme d'un module en python *balance-des-blancs-puissance-n.py* qui est téléchargeable sur <http://sellig.zed.myriapyle.net/Couleur/balablan.htm>.

## Références

Edmund Y. Lam, "Combining gray world and retinex theory for automatic white balance in digital photography," in *International Symposium on Consumer Electronics*, pp. 134–139, June 2005.

Sellig Zed, "Balance des blancs ?", <http://sellig.zed.myriapyle.net/Couleur/balablan.htm>.

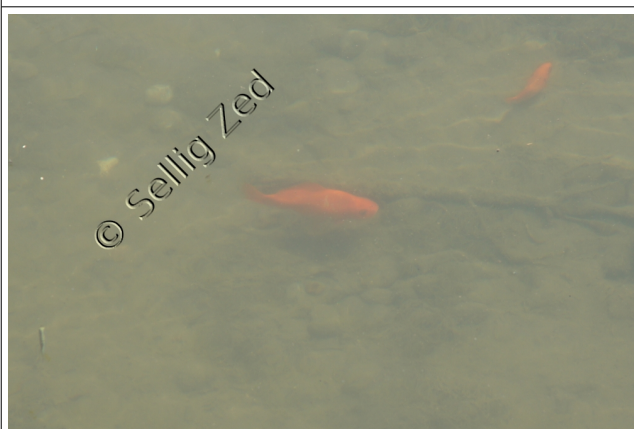
Planche 3 : exemples de résultats obtenus avec la balance des blancs en  $x^a$



Avant



Après, avec coefficient = +0,3



Avant (photo rejetée par la méthode « cubique »)



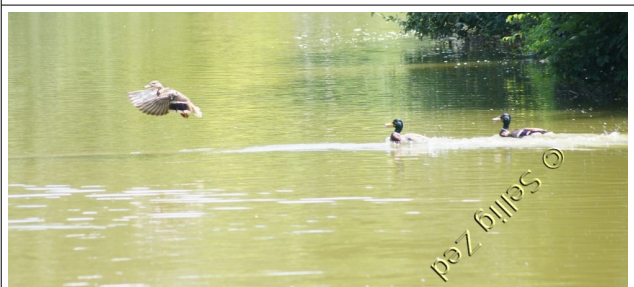
Après , avec coefficient = 0



Avant



Après, avec coefficient = 0









Avant



Après, avec coefficient = - 0.2

Planche 4 : influence du paramètre de luminosité avec la balance des blancs en  $x^u$

	 <p>Après, avec coefficient = -1.0</p>
<p>Transformation appliquée toujours à la même image (ci-dessous), en changeant seulement la valeur du paramètre de luminosité.</p>	 <p>Après, avec coefficient = -0.2</p>
 <p>Avant</p>	 <p>Après, avec coefficient = 0.0</p>
<p>Une valeur extrême (-1 ou +1) donne une image presque noire ou presque blanche. Le plus souvent, des valeurs comprises entre -0,3 et +0,3 conviennent.</p>	 <p>Après, avec coefficient = + 0.2</p>
	 <p>Après, avec coefficient = +1.0</p>

## Exemple de compte rendu d'exécution du module de transformation en x<sup>n</sup>

Fichier log de balance-puissance-n-SZ.py

=====

Image soumise au traitement :

/home/gilles/Documents/Culture(aussi dans doc)/Couleur/balablan/tremiere-ori.jpg

Dimensions image : largeur Xpix = 850, hauteur Ypix = 400, nombre de pixels Npix = 340000

Initialisation faite...

=====

=====

Phase 1 : calcul des histogrammes %

Calcul des histogrammes fini !

Intensité moyenne initiale = 113.178147059

=====

Les maxima des composantes sont :

max Rouge = 255, max Vert = 235, max Bleu = 233

Les minima des composantes sont :

min Rouge = 38, min Vert = 25, min Bleu = 0

Transformation affine OK :

R1= 1.17511520737, R0= -44.6543778802, V1= 1.21428571429, V0= -30.3571428571,

B1= 1.09442060086, B0= -0.0

=====

=====

Phase 2 : calcul des exposants ...

Coefficient de luminosité choisi = -0.2, d'où coefficient cible = 0.372822131488

... à comparer au coefficient de moyenne initial = 0.443835870819

Les exposants sont pour le rouge : 1.375, pour le vert : 1.625, pour le bleu : 0.875

=====

=====

Phase 3 - 1 : tabulation des transformations

Tables construites.

=====

Phase 3 - 2 : application des transformations

Transformations appliquées.

=====

=====

Au total, la transformation a pris 3 secondes.

Fin du programme.