

Rendu TP 1 : Couleurs et contraste

EL OUAFI, Moussa

30 janvier 2021

Consignes générales

- Copiez-Collez ce code **L^AT_EX**
- Répondez aux questions suivantes avec des phrases complètes.
- Ne dépassez pas 5 lignes par réponse.
- Si nécessaire, illustrez vos réponses avec des images insérées dans le document comme indiqué ci-dessous (figures ?? et ???).
- Toute figure doit être obligatoirement référencée depuis le texte avec \ref
- Les questions étoilées (*) sont plus difficiles. Laissez-les pour la fin.
- Le rapport en PDF doit être déposé sur le moodle pendant la journée.

Dans ce TP nous étudierons les articles suivants :

1. Simplest Color Balance
<http://www.ipol.im/pub/art/2011/l1mps-scb/>
2. Multiscale Retinex
<http://www.ipol.im/pub/art/2014/107/>
3. Image Color Cube Dimensional Filtering and Visualization
<http://www.ipol.im/pub/art/2011/blm-cdf/>

1 Simplest Color Balance (SCB)

1.1. Quel est le problème que la méthode cherche à résoudre ? La méthode SCP, consiste à effectuer une transformation affine $ax + b$ sur les trois chaînes de couleurs R (Red, Rouge), G(Green, Vert) et B(Blue, Bleu) pour qu'elles occupent tout l'intervalle $[0, 255]$ de sorte que la valeur maximale de la chaîne soit 255 et la valeur minimale soit 0. **Le problème** est au cas où les valeurs 0 et 255 sont déjà occupées par des pixels d'images, une transformation affine ne servira pas pour corriger l'image par la méthode souhaitée (SCB).

1.2. Quelle est la différence entre les deux méthodes (RGB et iRGB) ? La méthode RBG consiste à appliquer sur chaque chaîne R, B et G une transformation affine n'est pas nécessairement identique aux autres transformations appliquées sur les autres chaînes, ce qui modifie la couleur des pixels parce que la saturation n'est pas la même dans les différentes chaînes. Quant à la méthode iRGB, elle consiste à corriger l'intensité d'une image sans modifier le rapport des couleurs (R/B/G) des pixels.

La différence entre les deux est donc, RBG modifie la couleur d'image, et iRBG ne le modifie pas.

1.3. Décrire l'algorithme pour le cas d'images niveau de gris Pour des images niveau gris, les trois chaînes R, B et G sont égaux, donc on prend ($I = (R + G + B)/3$) l'intensité.

L'algorithme consiste donc à augmenter l'intensité de I jusqu'à une nouvelle valeur I' , sans changer la couleur. Donc les 3 chaînes subiront la même saturation et transformation affine, et en fin chaque chaîne de couleur sera multipliée par I'/I , et ceci en chaque pixel.

On note que parfois, les valeurs de ces chaînes peuvent dépasser la valeur maximale 255. Pour corriger ce problème on remplace ces pixels par les pixels qui forment l'intersection de la surface de cube de couleurs, et le segment passant par $(0,0,0)$ et ces pixels à remplacer. Dans ce cas d'image grise, ces pixels seront remplacés par le blanc(255,255,255).

Donc au cas d'images grises, on observe qu'une augmentation d'intensité avec une saturation à droite et à gauche.

1.4. Décrire le cas d'images couleurs l'Algorithme fonctionne de la même façon au cas gris, sauf que on aura un algorithme différent avec une paramétration différente dédié à chaque chaîne.

1.5. Donner un exemple de bon résultat, et un exemple de résultat insatisfaisant. Dans la figure 1, on observe un bon résultat dans la première ligne avec saturations $s = 2\%$ et $s = 3\%$, en effet on augmente l'intensité et modifie les couleurs, en particulier le couleur bleu inutile qui

se trouve dans la partie du chien, est corrigé et l'image semble plus naturelle.

Dans la deuxième ligne on trouve un résultat insatisfaisant Avec saturation $s = 2\%$ et $s = 3\%$. On observe la création d'un blue inaturelle, et la couleur orange est presque disparu.

1.6. Expliquer pourquoi l'algorithme donne un résultat insatisfaisant sur votre exemple.
Le problème vient de correction du gris-blue qui se reflecte sur la révière, et réside dans l'étape de projection des pixels sur la surface du cube de couleur lorsque la valeur des chaînes dépasse la valeur maximal 255.

En effet après l'augmentation d'intensité, la projection se fait au voisinage du blue sur le cube de couleur, ce que rend le blue très dominant d'une manière innaturelle.

Ce problème est commun dans les images ayant des pixels ambients, comme pour les images prises pendant la coucher du soleil.

1.7. Vous passez une vidéo 4 : 3 sur une télé 16 : 9 (ajout de bandes noires). Convient-il d'appliquer SCB avant ou après l'ajout de bande noire, et pourquoi ? Ca dépend de la nature d'image :

- Si on choisit une image pas sombre, appliquer SCP après l'ajout de bande noire rends des pixels brillant plus sombre donc on perd des informations utiles (en terme de qualité d'image et de précision) ce qui est une chose n'est pas toujours souhaitée.
- Si on souhaite que la bande ne forme pas un gradient dans les pixels obscures, il faut que on applique la bande en premier et puis appliquer une saturation avec un pourcentage bien précis.

En général, appliquer le SCB avant l'ajout de bande noire donne de bons résultats.

1.8. Justifier le choix des paramètres par défaut. les paramètres de saturation par défaut sont choisis $s_1 = 1,5\%$, $s_2 = 1,5\%$ petit pour saturer un minimum des pixels afin de ne pas créer des régions plates blanches ou noires.

La raison pour laquelle $s_1 = s_2$ par défaut revient peut être du traitement d'images au niveau gris. Pour ne pas avoir des zones plus sombres qu'autre dans une image su niveau gris, il faut que les pixels se saturent de la même manière(uniforme).

***1.9. Pouvez-vous envisager une méthode sans paramètres à régler ?** On peut utilser l'algorithme de multiscale rétinex pour faire sortir les détails des régions sombres.

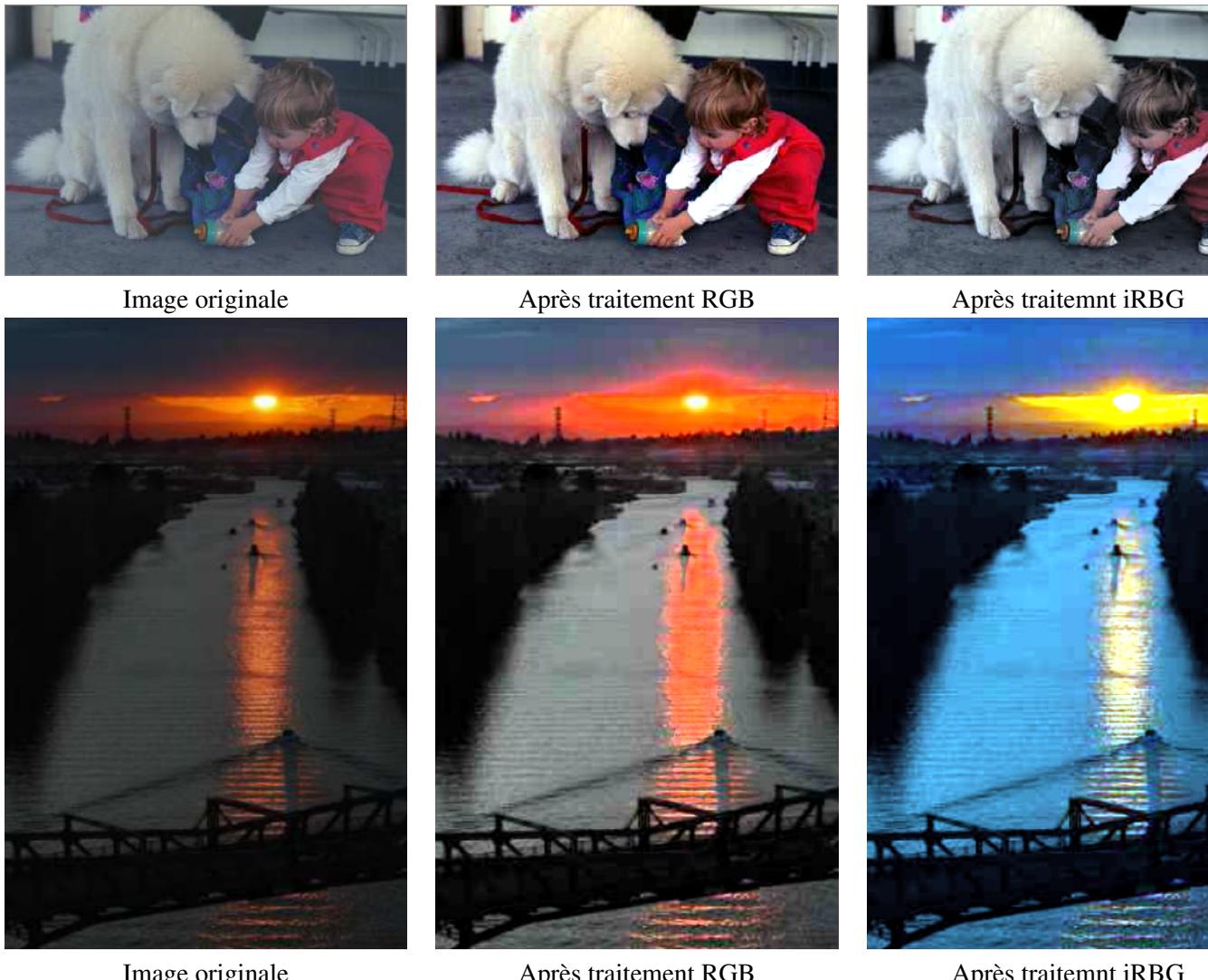


FIGURE 1 – Exemples : Premier ligne, bon résultat avec $s = 2\%$ et $s = 3\%$. Deuxième ligne : résultat insatisfaisante avec $s = 2\%$ et $s = 3\%$

2 Multiscale Retinex (MR)

2.1. Qu'est-ce que MR cherche à apporter en plus par rapport à SCB ? Par rapport a SCB qui cherche à rendre les images (prises à lumière artificielle, par exemple) plus équilibrées, Le MR cherche à restorer la couleur d'image est améliorer le contraste. Une différence très clair entre les deux méthodes et qui confirme la validité de MR est que contrairement au SCB qui sature un pourcentage des pixels brillants en pixels blancs et pixels sombres en noir, le MR fait sortir la plupart des détails des régions sombres.

Un exemple se trouve sans la figure 2.

2.2 Donner un exemple d'image où MR a un bon résultat tandis que SCB est insatisfaisant Dans la figure 3, on observe que après SCB le traitement RGB porte une amélioration à l'image mais ça reste très insuffisante. On observe aussi une perte de détails dans l'écriture au droit d'image après traitement iRGB à cause saturation des pixels brillants.

Par contre, on voit que après traitemet MR avec réstoration de couleur, L'image contient moins de bruit et la couleur de l'image est restorée assurant un très bon résultat en comparaison avec le SCB.

***2.3. Trouver un exemple où le résultat de MR est pire que celui de SCB. (Que ce soit lorsque appliqué sur les couleurs ou sur l'intensité - avec les paramètres pas défaut)** Dans la figure 4, on observe que le MR appliqué sur l'intensité avec les paramètres pas défaut, est pire que celui de SCB.

On remarque aussi que l'application de MR à toutes les chaîne indépendament rend l'image plus grise.

2.4. Si vous mettez les trois paramètres des gaussiennes à la même valeur σ_0 , cela est équivalent à utiliser une seule gaussienne de taille σ_0 . Construire trois images où ce σ_0 prend chacune des trois valeurs par défaut. Dans le cas ou on met les pramètres des gaussiennes a la même valeur, on remarque une différence entre les 3 images traitées de la figure 5 dans les couleurs, la contraste et les zones sombres (au bord d'images par exemple) connues par halo-artifacts.

Pour $\sigma_0 = 250$, on voit que le contraste au milieu (zonne blanche) semble plus flou et intense que celle de $\sigma_0 = 80$ et $\sigma_0 = 15$. Quant au zones sombres, on remarque que l'image traitée par un paramètre $\sigma_0 = 250$ contient plus des halo-artifacts que celle traitée par le paramètre $\sigma_0 = 15$. et pour le contraste, on remarque que celui d'image traitée par $\sigma_0 = 80$ semble être plus dynamique que les autres.

Donc pour avoir une bonne cohérence des couleurs et de teinte, et une compréssion dynamique il faut utiliser trois gaussiennes, pour obtenir des resultats comparable à la vision d'homme.



FIGURE 2 – la Méthode MR donne des résultats meilleurs que SCB.



FIGURE 3 – la Méthode MR donne des résultats meilleurs que SCB.

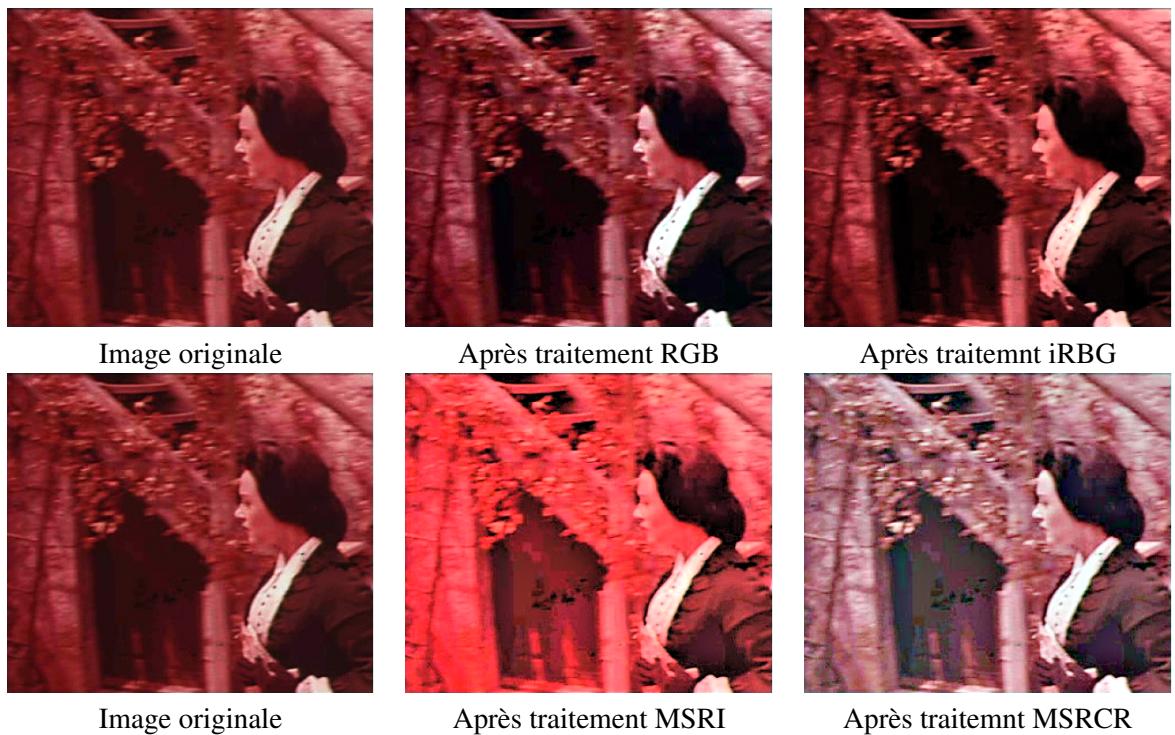


FIGURE 4 – Un exemple MR donne des résultats meilleurs que SCB.



Image originale



Image avec $\sigma_0 = 15$



Image avec $\sigma_0 = 80$



Image avec $\sigma_0 = 250$

FIGURE 5 – Multi-Scale Retinex on Intensity with Simplest color balance saturation parameters equal to 1% on each side.

2.6. Vous savez que l'ensemble de toutes les images est un espace vectoriel. L'application $I \mapsto MR(I)$ est-elle linéaire ? D'où vient la non-linéarité ? L'application $I \mapsto MR(I)$ n'est pas linéaire, car la fonction logarithme n'est pas linéaire.

***2.7. Imaginez une méthode “MR-linéarisée” où toutes les opérations non-linéaires sont remplacés par la fonction identité. Croyez-vous que cette méthode donnerait des résultats visuellement très différents ?** On effectue une régression linéaire, on approxime $\log(I)$ par une fonction affine $aI + b$, pour une image $I(x,y)$ prenant des valeurs dans l'intervalle $]\eta, 256[$, avec $\eta > 0$.

Pour avoir un erreur d'aprosimation minimale, on peut aussi choisir les valeurs a et b de la manière suivante :

$$(a, b) = \operatorname{Argmin} \int |\log(I(x,y)) - aI(x,y) - b|^2 dx dy$$

Cette méthode donne des résultats visuellemnet différents si l'image est très sombre sinon le résultat n'est pas si différent. En effet comme le montre le figure 6, On doit nécessairement saturer la zone du niveau gris $]0, \eta[$, donc si on choisi $\eta = 60$ par exemple, on n'observe pas grande différence pour image qui a pour une obscurité minimal 60.

3 Image Color Cube Dimensional Filtering and Visualization

3.1. Que représente le *color cube* d'une image couleur ? le *color cube* d'une image couleur est la représentation tridimensionnelle des points de couleur (R, G, B) de cette une image numérique en l'espace RGB de dimension trois.

3.2. À quoi devrait ressembler le *color cube* d'une image à niveau de gris ? Pou une image à niveau de gris, les trois couleur R, G, B ont la même valeur, donc l'image sera représentée sur l'axe gris (the gray axis) illustré dans la figure 7.

3.3. À quoi ressemble le *color cube* d'une image avec un tirage sépia ? Par définition la couleur sépia est une qualité de tirage qui ressemble au noir et blanc, mais avec des variations de brun, et non de gris.

Une représentation d'une image tirée en sépia se trouve dans la figure 8.
On observe que le *color cube* d'image est un morceau de surface.

3.4. À quoi ressemble le *color cube* d'une image (synthétique) du drapeau français ? Puisque l'image du drapeau est constituée des couleurs rouge, blanc et bleu, sa représentation dans *color cube* est donc les trois ponts de coordonnées ROUGE(255,0,0), BLANC(255,255,255) et BLEU(0,0,255).

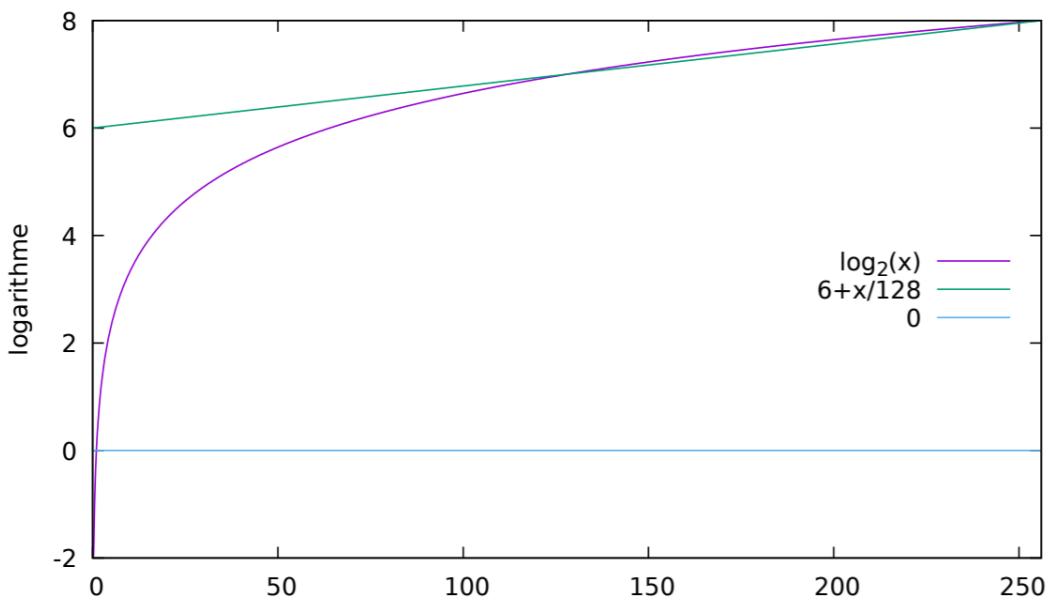
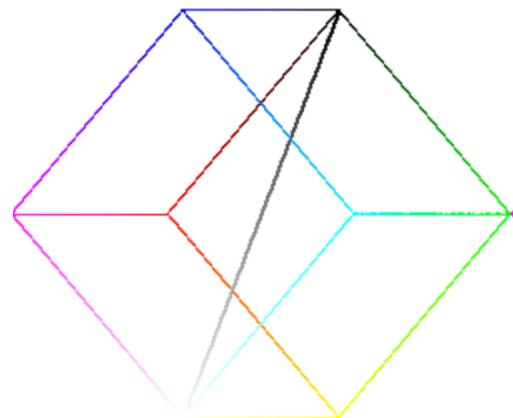


FIGURE 6 – Approximation du logarithme par une fonction affine.



Image à niveau de gris



son *color cube*

FIGURE 7 – le *color cube* d'une image à niveau de gris.

3.5. À quoi ressemble le *color cube* d'une photographie couleur du drapeau français ? La densité du *color cube* d'une photographie couleur du drapeau français 9 n'est pas quasiment nulle comme celle du *color cube* d'une image synthétique du drapeau. En effet, dans une photographie on peut trouver une variation de ces couleurs qui vient soit du fait qu'on a photographié une zone extérieur au drapeau soit par réflexion d'autres couleurs sur le drapeau, qui peuvent créer des nuances des couleurs.

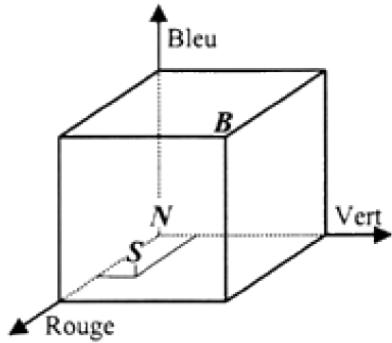
La figure 9 illustre ceci.

3.6. Pourquoi est-il raisonnable de supposer que le *color cube* d'une image en couleur générique sera formé de morceaux de surface ? C'est à cause du flou et la texture d'images. Une image non filtrée tend à avoir beaucoup de zones floues ce qui crée "une nuance de couleur" la représentation de ces nuances sur le *color cube* sera quasi continu donc la formation d'un morceau surfacique.

3.7. Quel est le problème que la méthode cherche à résoudre ? L'algorithme consiste à Débruiter les échantillons de nuages de couleurs et les merger afin de récupérer une image filtrée. Le problème réside dans rendre la distribution des couleurs lisse pour révéler la structure bidimensionnelle des couleurs.

***3.8. Généralement l'image filtrée est visuellement identique à l'image de départ. Donner un cas où l'image filtrée sera de meilleure qualité / de moins bonne qualité [Un exemple pour chaque].**

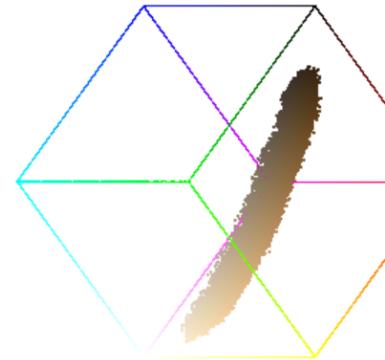
- cas où l'image filtrée sera de meilleure qualité : δ est très petit.
- cas où l'image filtrée sera de meilleure qualité : δ est assez grand ce qui rend la nuage non une surface connexe.



Sépia (94, 38, 18)



Image tirée en sépia

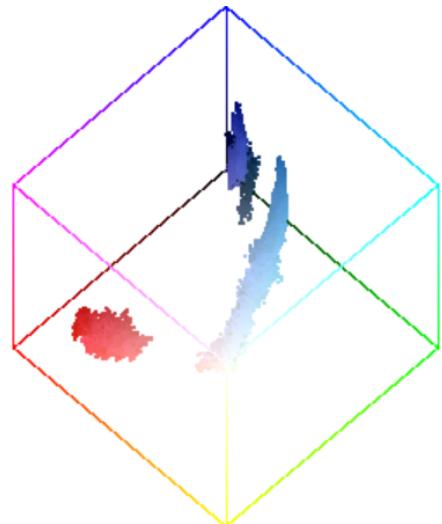


son *color cube*

FIGURE 8 – Réprésention du sépia dans le *color cube* RVB : S(94, 38, 18).



Photo du drapeau Français

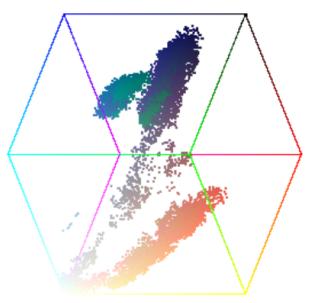


son *color cube*

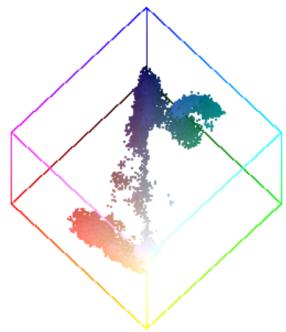
FIGURE 9 – le *color cube* d'une photographie couleur du drapeau français



image Générique



vue 16 de son *color cube*



vue 4 de son *color cube* son *color cube*

FIGURE 10 – le *color cube* d'une image générique.