

# Compte rendu du TP n° 1

## Changement d'espace couleur

Thibaut Castanié  
*M2 IMAGINA*

27 septembre 2015

# 1 Image originale choisie



*L'image originale utilisée pour le TP*

## 2 Compression dans l'espace RGB

J'ai choisi de réduire la taille des composantes Verte et Bleue. Pour diminuer leur taille, je fait la moyenne de la valeur des pixels dans chaque région de 4x4 de l'image, et je la sauvegarde, dans l'ordre, dans une image deux fois plus petite.



*La composante Verte et Bleue a été réduite*

Ensuite on agrandit de nouveau les deux composantes précédentes, en utilisant le même algorithme précédent, mais dans l'autre sens. Pour chaque pixel, on applique sa valeur quatre fois dans la nouvelle image : sur le pixel de gauche, celui du bas et celui en bas à droite.



*La composante Verte et Bleue agrandie*

Enfin, nous pouvons reconstruire l'image originale en fusionnant les deux composantes ré-échantillonnées avec la composante rouge non modifiée.



*L'image reconstruite*

Visuellement, on constate une perte de qualité de l'image, et les objets semblent être entourés d'un léger contour rouge. Nous avons calculé le PSNR (Peak Signal to Noise Ratio) qui permet de calculer la qualité de l'image compressée par rapport à l'image originale.

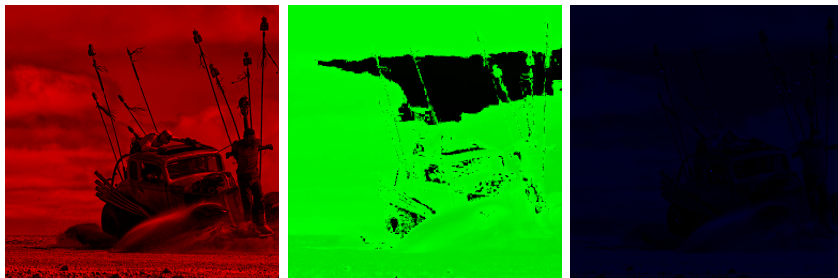
> **EQM : 1.20079e+08 | PSNR : 32.6639**

### 3 Compression dans l'espace YCbCr

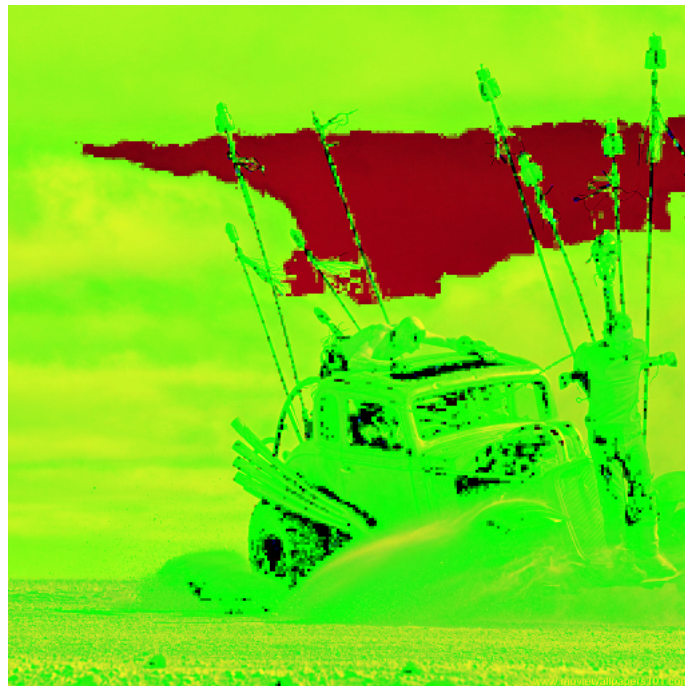
Les mêmes opérations ont été appliquées dans l'espace YCbCr. Cependant, afin d'avoir un aperçu du résultat, j'ai affiché l'image convertie de RGB vers YCbCr, dans l'espace RGB. J'obtiens ainsi les résultats suivants.



*L'image YCbCr affichée dans l'espace RGB*



*Les composantes G et B ont été ré-échantillonnées, réduites puis agrandies*



*L'image YCbCr reconstruite*

**EQM : 1.35187e+08 | PSNR : 33.1785**

## 4 Conclusion

Bien que le PSNR obtenu en comparant l'image originale et l'image compressée soit acceptable, le rendu visuel de l'image compressée est médiocre. En effet, on obtient un rendu très pixelisé de l'image, en plus d'avoir un léger contour rouge (la composante non compressée) autour des objets de l'image.

Une autre approche pour compresser une image serait de prendre les pixels dont la valeur est impaire, et de diminuer leur valeur de 1 pour qu'elle devienne paire. Ainsi, l'image comporterait deux fois moins de couleurs, et cela ne se verrait presque pas à l'œil nu.

## Annexes - Code

Code appliquant les opérations du TP : `compressionRGP.cpp`

Code convertissant une image RGB en YCbCr : `convertRGBtoYCbCr.cpp`