
TP - CAV

Transfert de couleur

1 Introduction

L'une des tâches les plus courantes en traitement d'images consiste à modifier la couleur d'une image. Souvent, cela signifie supprimer une couleur dominante et indésirable, comme le jaune sur les photos prises sous éclairage à incandescence. Ce rendu décrit une méthode de correction des couleurs qui emprunte les caractéristiques de couleur d'une image à une autre. La figure 1 montre un exemple de ce processus, où nous appliquons les couleurs d'une photographie de coucher de soleil à une image de nuages en journée.



FIGURE 1 – Figure 1 : Exemple de transfert de couleur

2 Transfert de couleur classique

Le transfert de couleur est une méthode pratique pour modifier l'apparence d'une image source en fonction du motif de couleur d'une image référence. Pour commencer, nous avons donc réalisé un transfert de couleur pour les images RGB classiques. Pour cela, nous avons implementé une méthode simple, l'algorithme de Reinhard [1].

2.1 Reinhard

L'algorithme de Reinhard est l'approche la plus simple pour faire du transfert de couleur. Cet algorithme a pour objectif de transférer la répartition des coefficients de luminance et de chrominance d'une image source, en se basant sur la répartition donnée par une image cible. On convertit préalablement au transfert les images source et cible dans l'espace couleur $L \alpha \beta$. Cet espace est un espace couleur décorrélé, ce qui signifie que chaque canal couleur est quasi-indépendant des autres. Cette propriété permet de garantir que les répartitions des valeurs au sein

des trois canaux peuvent être traitées indépendamment des autres, ce qui permet d'améliorer la qualité du transfert. Dans cet algorithme, on suppose que la répartition des coefficients au sein de chaque canal suit une loi normale. Pour faire le transfert, il suffit alors de centrer et normaliser les trois répartitions sources, puis multiplier par l'écart type des répartitions cibles et enfin y ajouter leurs espérance. Voici un résumé des étapes importantes avec une image source et une image référence :

Étape 1 : Convertir l'image source et l'image référence dans l'espace colorimétrique $L \alpha \beta$. Cet espace modélise l'uniformité perceptuelle, où une petite variation d'une valeur de couleur devrait également produire un changement relativement important de l'importance de la couleur.

Étape 2 : Calculer la moyenne et l'écart-type de chacun des canaux $L \alpha \beta$ pour les images source et référence.

Étape 3 : Normaliser la source avec la moyenne et l'écart-type. On applique ensuite la moyenne et l'écart-type de la référence sur la source.

Étape 4 : Convertir à l'espace colorimétrique RGB à partir de l'espace $L \alpha \beta$.

Voici comment convertir une image de RGB vers $L \alpha \beta$:

$$\begin{pmatrix} L \\ M \\ S \end{pmatrix} = \begin{pmatrix} 0.3811 & 0.5783 & 0.0402 \\ 0.1967 & 0.7244 & 0.0782 \\ 0.0241 & 0.1288 & 0.8444 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

$$\begin{pmatrix} L \\ M \\ S \end{pmatrix} = \begin{pmatrix} \log(L) \\ \log(M) \\ \log(S) \end{pmatrix}$$

$$\begin{pmatrix} L \\ \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} 1/\sqrt{3} & 0 & 0 \\ 1 & 1/\sqrt{6} & 0 \\ 0 & 0 & 1/\sqrt{2} \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & -2 \\ 1 & -1 & 0 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

Voici quelques exemples que nous avons obtenu avec notre implémentation :



FIGURE 2 – Image source



FIGURE 3 – Image référence



FIGURE 4 – Résultat



FIGURE 5 – Image source 2



FIGURE 6 – Image référence 2

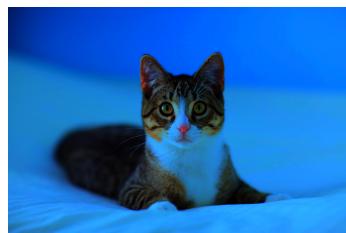


FIGURE 7 – Résultat 2

Nous remarquons que sur nos exemples, le transfert de couleur s'applique bien, mais les résultats sont assez sombres. Nous pourrions arranger cela en appliquant une correction γ afin de faire varier l'espace dans lequel on réalise le transfert de couleur de manière légère. Cependant, une erreur nous a empêché d'effectuer cette correction. Nous affichons les résultats avec un mappage tonale.

2.2 Tranfert de couleur via MGD

Il existe plusieurs travaux visant à améliorer le résultat obtenu avec la solution de Reinhard. L'un de ces travaux a pour nom Distribution Gaussienne Multivariée, et a été proposé par F.Pitie et A. Kokaram [2].

Leur réflexion se base sur le constat que l'espace couleur $L^{\alpha\beta}$ n'est pas un espace parfaitement décorrélé, ce qui signifie que les canaux ne sont pas entièrement indépendants. Considérer alors des transformations indépendantes sur chaque canal constitue une approximation, et peut amener à des pertes de précision. La solution proposée est donc de ne plus considérer trois distributions gaussiennes 1D (1 par canal), mais bien une seule distribution gaussienne 3D. Il faut alors trouver une transformation t permettant de passer de la distribution de l'image source à celle de l'image cible.

Si u l'image source suit une distribution f , et v l'image cible suit une distribution g , il s'agit de trouver t tel que $t(u)$ suit une distribution g .

Il s'agit d'un problème ayant une infinité de solutions, c'est pourquoi F.Pitie et A. Kokaram ne considèrent que les solutions linéaires. Considérer des distributions gaussiennes multivariées permet de garantir l'existence d'une solution dans le cas linéaire, et correspond à la répartition classiquement obtenue pour une scène naturelle.

Une distribution gaussienne multivariée est définie par une espérance et une matrice de covariance. Si les données ne suivent pas ou pas parfaitement une distribution gaussienne multivariée, une telle distribution peut toutefois être estimée à partir de l'espérance μ et de la matrice de covariance Σ . Le problème revient alors à calculer t , tel que :

$$t(u) = T(u - \mu_u) + \mu_v$$

avec : $T\Sigma_u T^T = \Sigma_v$

Il y a plusieurs solutions pour calculer la matrice T , nous utiliserons celle donnée par Monge-Kantorovitch. Il s'agit de la solution la plus contrainte, qui ajoute une notion de minimisation de déplacement pour garder une cohérence vis à vis des couleurs.

La matrice T est alors donnée par :

$$T = \Sigma_u^{-1/2} (\Sigma_u^{1/2} \Sigma_v \Sigma_u^{1/2}) \Sigma_u^{-1/2}$$



FIGURE 8 – Image source



FIGURE 9 – Image référence



FIGURE 10 – Résultat


FIGURE 11 – Image source

FIGURE 12 – Image référence

FIGURE 13 – Résultat

On peut constater que les résultats diffèrent un peu de ceux obtenus avec l'algorithme de Reinhard. La colorimétrie de l'image obtenue avec la MGD semble plus fidèle à l'image référence, ce qui est cohérent car l'estimation des distributions est plus fidèle et donc le transfert est plus précis.

Une des améliorations possibles de la MGD est de traiter différemment la luminance et la chrominance, comme présenté dans le papier de Lee et al [3].

Dans ce papier eset présenté une méthode permettant de faire du transfert de style automatique à partir d'un ensemble d'images sélectionnées dans un dataset. La partie qui nous intéresse est celle relative au transfert de couleur, dans lequel les auteurs utilisent la MGD pour le transfert de chrominance et une méthode faisant appel à une fonction de transfert pour la luminance.

Soit x une image, et l_x sa luminance.

$$l_o = g(l_x) = \frac{\arctan(\frac{m}{\delta}) + \arctan(\frac{l_x - m}{\delta})}{\arctan(\frac{m}{\delta}) + \arctan(\frac{1-m}{\delta})}$$

Avec m et δ deux paramètres déterminés en minimisant la fonction $\|g(L_I) - L_*\|$

$$\text{où } L_* = L_I + (L_S - L_I) \frac{\rho}{\min(\rho, |L_S - L_I|_\infty)}$$

ρ a pour valeur 0,4, L_I et L_S correspondent respectivement aux luminosités de l'image source et de référence exprimées sous la forme d'un vecteur contenant les n pourcentiles de l'histogramme.

Dans l'implémentation que nous avons fait, et qui se base sur une implémentation déjà existante [4], on utilise l'outil `scipy.optimize.minimize` pour trouver les m et δ optimaux. On a juste ensuite à calculer $g(l_x)$ avec les m et δ calculés. La minimisation échoue cependant de temps en temps.


FIGURE 14 – Reinhard

FIGURE 15 – MGD

FIGURE 16 – MGD avec traitement différent de la luminance

On peut constater que les résultats obtenus avec la méthode décrite dans le papier de Lee et al sont plus proches de la teinte de l'image de référence. Le transfert s'effectue de façon plus marquée sur la luminance.

3 Transfert de couleur HDR

Le transfert de couleur est facilement extensible aux images HDR car ses concepts restent quasiment identiques. Nous avons donc étendu l'algorithme de Reinhard à celles-ci. Il s'agit du même algorithme utilisé dans la section II, mais en changeant les données d'entrées.



FIGURE 17 – Image source



FIGURE 18 – Image référence



FIGURE 19 – Résultat



FIGURE 20 – Résultat avec Gamma ajusté

On peut remarquer que sur les résultats, les couleurs ressortent plus que sur images LDR. On peut remarquer que notre premier résultat est un peu sombre, on utilise donc un gamma pour le rendre plus lumineux.

4 Biliographie

- [1] Reinhard, E., Ashikhmin, M., Gooch, B., Shirley, P. (2001). Color Transfer between Images [Ebook]. IEEE Computer Graphics and Applications. Retrieved from <http://www.cs.tau.ac.il/turkel/imagepapers/ColorTransfer.pdf>
- [2] Pitié, F., Kokaram, A. (2007). The Linear Monge-Kantorovitch Linear Colour Mapping for Example-Based Colour Transfer [Ebook]. Dublin. Retrieved from <http://www.mee.tcd.ie/sig-media/pmwiki/uploads/Main.Publications/fpitie07b.pdf>
- [3] Lee, J., Sunkavalli, K., Lin, Z., Shen, X., Kweon, I. (2015). Automatic Content-Aware Color and Tone Stylization [Ebook]. Retrieved from <https://arxiv.org/pdf/1511.03748.pdf>
- [4] jinyu121/ACACTS. (2016). Retrieved 5 January 2021, from https://github.com/jinyu121/ACACTS/blob/master/transfer/transfer_style.py

Nous attestons que ce travail est original, qu'il indique de façon appropriée tous les emprunts, et qu'il fait référence de façon appropriée à chaque source utilisée.