

# Utilisation du transport optimal pour le transfert de couleurs

Maël COLIN

## Intro :

Ce document explique comment réaliser un transfert de couleurs entre deux images à l'aide du transport optimal. De plus, ce document est mon troisième document portant sur le transport optimal, je recommande donc fortement de lire les précédents documents pour comprendre le fil de ce document.

## Sommaire :

- 1 – Lien entre l'imagerie numérique et le transport optimal
- 2 – Exemple de transfert de couleur dans un cas simple avec la méthode « naïve »
- 3 – Implémentation d'un algorithme de transfert de couleur et observations
- 4 – Conclusion

## 1 – Lien entre l'imagerie numérique et le transport optimal

Petite contextualisation : L'idée d'utiliser le transport optimal pour le transfert de couleurs est apparu aux alentours de 2010 pour l'industrie du cinéma. En effet, certains films nécessitaient de fortes colorations (désert que

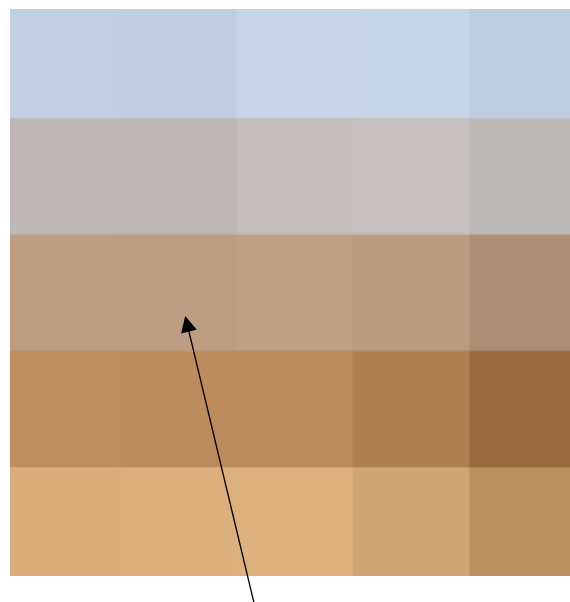
l'on rend plus orange, jungle que l'on assombrie...). Ces colorations obligatoirement faites à la main était longue et peu pratiques. Il a fallu alors trouver un moyen numérique pouvant réaliser de telle colorations, tout en gardant les contrastes et formes des images. Le transfert de couleur à l'aide du transport optimal a donc été la solution la plus simple à ce besoin.

Dans la suite du document on se ramènera toujours, quitte à redimensionner les deux images, à des images carrées de même dimension. De plus, par soucis de compréhension, on appellera dans la suite du document « image support » l'image dont l'on veut changer la couleur et « image palette » l'image dont la couleur est transféré sur l'image support.

Pour définir un problème de transport optimal, trois éléments sont nécessaire :

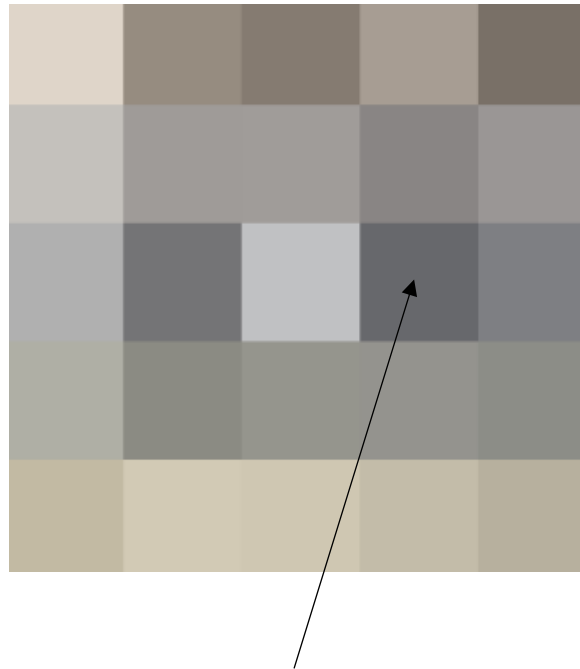
- Définir la matière à transporter
- Définir des sites de production/utilisation de matière
- Définir une distance entre un site de production de matière et un site d'utilisation de matière

Dans le cas d'un transfert de couleur entre deux images, les sites de production et d'utilisation de matière sont respectivement les pixels de l'image support et de l'image palette. Pour cela on, représente chaque pixel des deux images par un vecteur dont les coordonnées sont la valeur de rouge, la valeur de vert et la valeur de bleu du pixel.



Par exemple, le pixel pointé par la flèche possède 188 comme valeur rouge , 157 comme valeur de vert et 129 comme valeur de bleu, son vecteur est donc (188,157,129).

On définit alors la distance entre deux pixels des deux images comme la distance euclidienne entre les vecteurs des deux pixels. Par exemple :



Le vecteur du pixel de la deuxième image pointé par la flèche est (103,104,108). La distance entre les deux pixels vaut donc :

$$\text{distance} = \sqrt{(188 - 103)^2 + (157 - 104)^2 + (129 - 108)^2} \approx 102,3$$

## 2 – Exemple de transfert de couleur dans un cas simple avec la méthode naïve

Pour bien comprendre comment faire un transfert de couleur à l'aide du transport optimal, réalisons un exemple dans un cas simple en utilisant la méthode de calcul dite « naïve ».

Pour cela on prend les deux images suivantes :

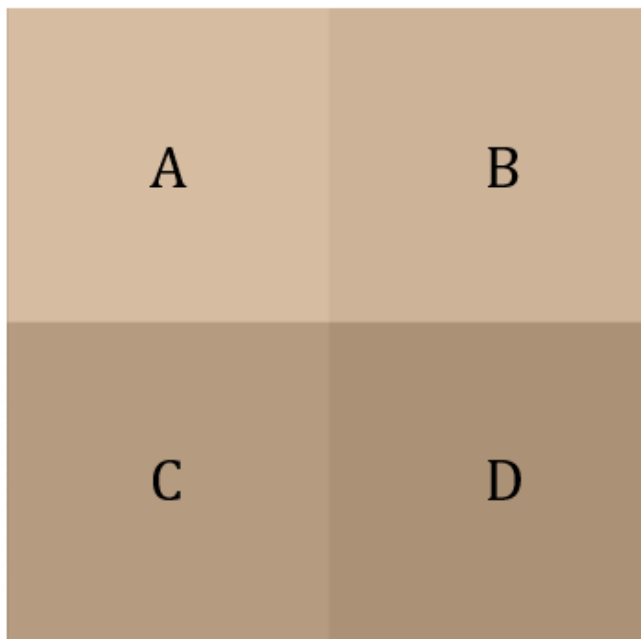


Image support

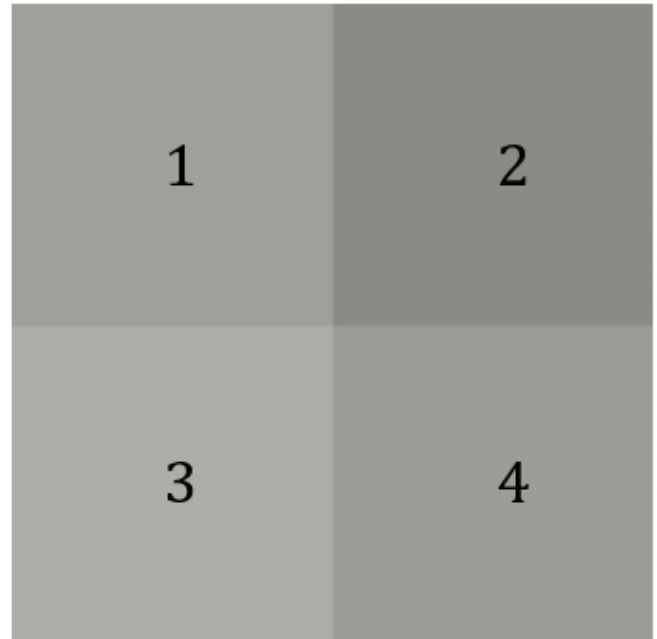


Image palette

De plus, par soucis de compréhension, on note de A à D les pixels de l'image support, de 1 à 4 les pixels de l'image palette.

On note alors les vecteurs de tous les pixels des deux images dans ces deux tableaux :

A : (214, 188, 161)	B : (205, 179, 152)
C : (181, 155, 128)	D : (171, 145, 118)

1 : (160, 159, 154)	2 : (139, 138, 133)
3 : (174, 173, 168)	4 : (157, 156, 151)

Les calculs de toutes les distances entre les pixels des deux images sont laissés au lecteur. On obtient toutefois après calcul les tableaux suivant :

Distance entre A et 1 : 61,69278726	Distance entre A et 2 : 94,38749917	Distance entre B et 1 : 49,28488612	Distance entre B et 2 : 79,98749902
Distance entre A et 3 : 43,28972164	Distance entre A et 4 : 66,12866247	Distance entre B et 3 : 35,39774004	Distance entre B et 4 : 53,23532662

Distance entre C et 1 : 33,66006536	Distance entre C et 2 : 45,58508528	Distance entre D et 1 : 40,16217126	Distance entre D et 2 : 36,02776707
Distance entre C et 3 : 44,41846463	Distance entre C et 4 : 33,25657830	Distance entre D et 3 : 57,38466694	Distance entre D et 4 : 37,49666652

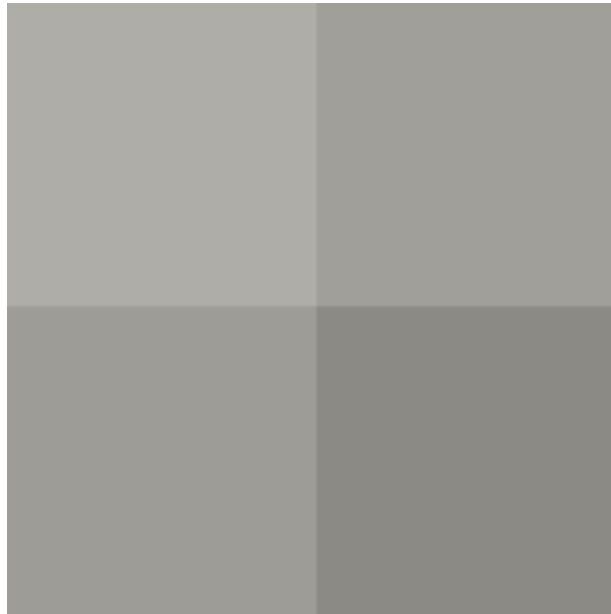
Je rappelle qu'optimiser le transport consiste à minimiser la somme des distances. On calcul donc toutes les sommes de distances en faisant attention au fait qu'un pixel de l'image palette ne peut être associé qu' à un pixel de l'image support. Par soucis de place, les calculs de l'ensemble des sommes sont présent à la fin du document. On trouve ainsi que la somme minimal est obtenue pour les distances en rouge dans les tableaux et sa valeur vaut :

$$\min d_{tot} = 43,28972164 + 53,23532662 + 33,66006536 + 36,02776707 = 161,21288069$$

Remarque : Le programme utilisé pour les prochains transferts de couleurs est codé sur python. Python arrondissant les calculs à partir du 9ème chiffre significatif, j'ai bien fais attention de prendre 9 chiffres significatifs lors des calculs de l'exemple. Cette explication est importante puisqu'en fonction des arrondies, les pixels minimisant la somme des distances peuvent changer, le transfert de couleur est alors complètement différent.

Pour finir, on change la couleur de chaque pixel de l'image support en la couleur du pixel associé de l'image palette. Ainsi le pixel A prend la couleur du pixel 3, le pixel B prend la couleur du pixel 4, le pixel C prend la couleur du

pixel 1 et le pixel D prend la couleur du pixel 2. On obtient donc l'image suivante :



### 3 – Implémentation d'un algorithme de transfert de couleurs et observations

Maintenant que le concept de transfert de couleurs à l'aide du transport optimal, on peut passer à l'implémentation d'un programme réalisant des transferts de couleur.

Principe du programme :

- 1 - Transformer chaque pixel des images support et palette en vecteur de 3 coordonnées R ,G,B.
- 2 - Calculer les distances entre chaque pixel des deux images.
- 3 - A l'aide du transport optimal, minimiser la somme des distances entre les pixels des deux images.

4 – Appliquer à chaque pixel de l'image support, la couleur du pixel de l'image palette associée.

5 - Renvoyer l'image support avec les couleurs de l'image palette.

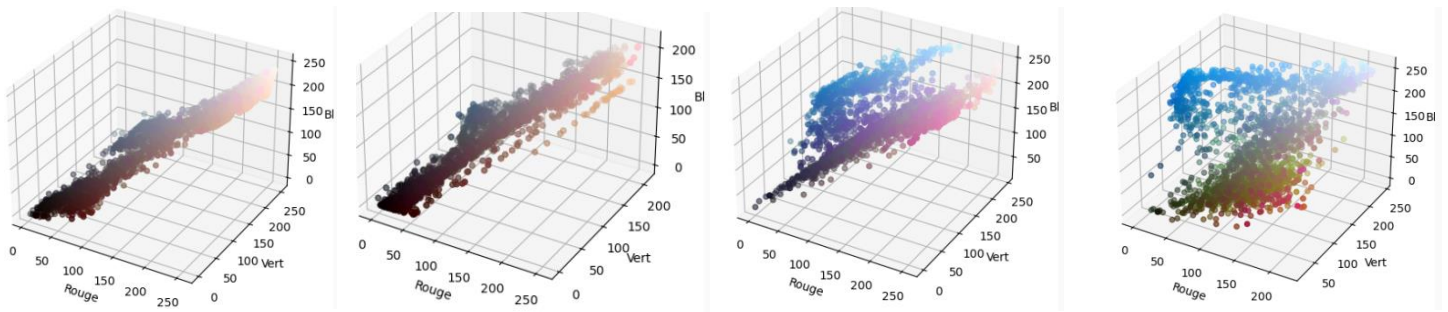
Maintenons que nous avons un algorithme de transfert de couleur, regardons comment bien réaliser un transfert de couleur. Dans la suite du document, étant donné que mon ordinateur n'est pas puissant, les transferts de couleur seront réalisés avec des photos de dimension 50 sur 50. Avec un ordinateur plus puissant on peut évidemment réaliser un transfert avec des photos moins pixélisées.

L'image support choisie est l'image numéro 86 de la banque d'image libre de droit de Laurent Condat. Voici le lien pour télécharger la banque : <https://lcondat.github.io/imagebase.html>

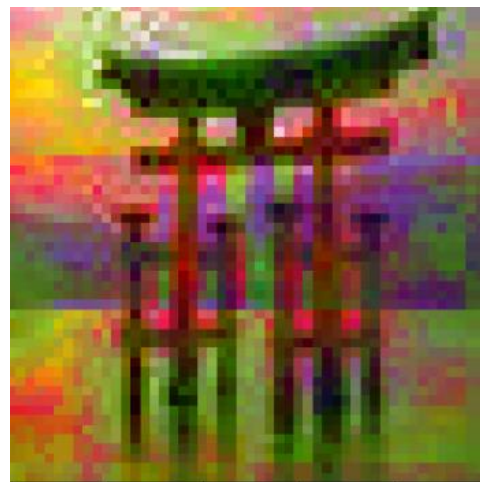
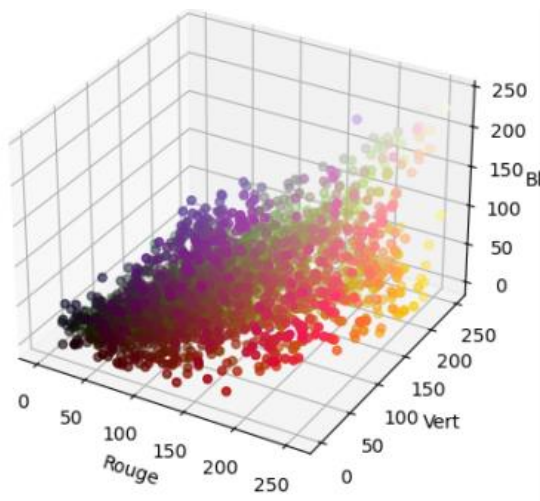


Image support

Tout d'abord on peut réaliser des transferts de couleurs pour des images palettes dont les couleurs sont de plus différentes de celle de l'image support.



On remarque alors que plus les images possèdent des couleurs qui sont différentes, moins le transfert de couleurs est satisfaisant. On ne reconnaît même plus le paysage de l'image lorsque les couleurs sont complètement différentes :

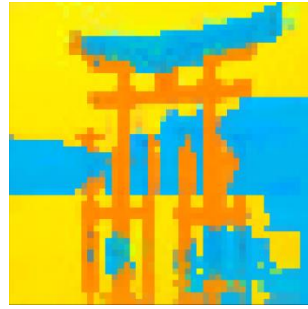


On peut donc étudier l'influence sur le transfert de couleurs du nombre de couleurs différentes entre les deux images. On réalise donc des transferts pour des nombres de couleurs différentes croissants.

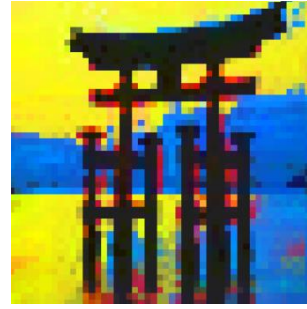




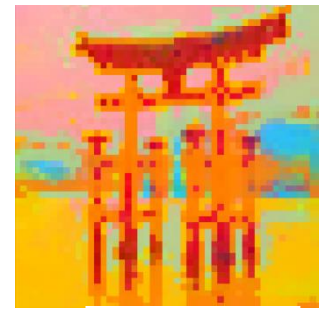
2 couleurs



3 couleurs



5 couleurs



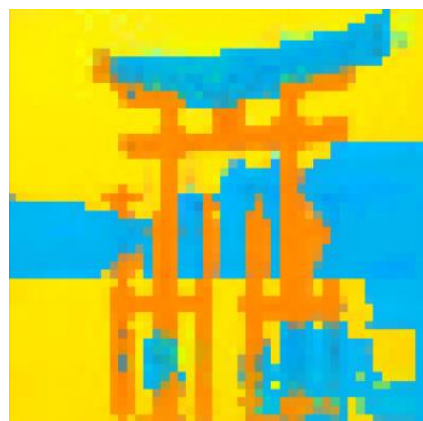
6 couleurs

Contrairement à ce que l'on avait supposé précédemment, on observe que le transfert est de plus en plus satisfaisant lorsqu'on augmente le nombre de couleurs différentes jusqu'à 5 couleurs. Toutefois, le transfert est de moins en moins satisfaisant lorsque le nombre de couleurs différentes dépasse 5 couleurs. Ce palier de 5 couleurs correspond au nombre de couleurs de l'image support. Le transfert est donc plus satisfaisant lorsque l'image support et l'image palette ont le même nombre de couleurs.

Cependant, le transfert de couleurs pour 4 couleurs différentes ne suit pas cette règle. En effet, le transfert de couleur pour 4 couleurs est moins satisfaisants que pour 3 couleurs :



Image support



3 couleurs

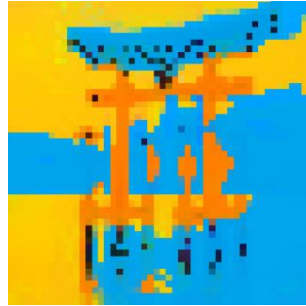


4 couleurs

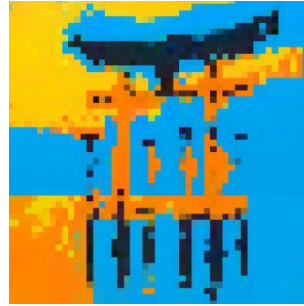
Pour comprendre pourquoi le transfert de couleurs pour 4 couleurs est moins satisfaisants que pour 3 couleurs, on peut réaliser des transferts de couleurs avec l'image palette de 4 couleurs mais cette fois appauvri à chaque fois en 1 couleur :



Appauvri en orange



Appauvri en noir



Appauvri en jaune



Appauvri en bleu

On remarque alors, le transfert de couleur est plus satisfaisant lorsque l'on appauvri l'image palette en bleu, la couleur en surplus. En appauvrissant l'image palette avec 5 couleurs on obtient ce transfert de couleur :



Les couleurs de l'image palette et les couleurs associés de l'image support possèdent alors un nombre de pixel similaire. Pour réaliser un transfert de couleur satisfaisant, il faut donc que les couleurs associées entre les deux images possèdent des nombres de pixels similaires.

## 4 – Conclusion

Je propose comme conclusion de « corriger » le transfert de couleurs de la section précédente où les couleurs entre les deux images sont complètement différentes :

Je rappelle que la première image obtenue après le transfert de couleur est celle-ci :



Le transfert de couleur est plus satisfaisant lorsque les deux images ont des nombres de couleurs similaires .On réduit donc le nombre de couleurs de l'image palette pour que ce nombre soit similaire au nombre de couleur de l'image support. Par exemple, on obtient ainsi l'image :



Le transfert de couleur est plus satisfaisant lorsque les couleurs associées entre les deux images ont un nombre similaire de pixel. On réduit donc le nombre de pixels de couleur violet et on augmente le nombre de couleurs de pixels de couleur rose de l'image palette. On obtient ainsi l'image :




Pour comparer entre le premier transfert de couleur et le dernier transfert de couleur :



Le résultat reste grossier, pour un résultat parfait il faudrait compter tous les pixels de chaque couleur de l'image support. Cependant, ce n'est pas le but du transfert de couleurs avec le transport optimal qui, justement, est utilisé pour éviter de faire le transfert manuellement.

Si vous voyez une coquille, merci de me prévenir au mail :  
mcolin.maths@gmail.com

Calcul de toutes les sommes de distances :

0	61.69278726	1	79.98749902	2	44.41846463	3	37.4966652	223.5954
0	61.69278726	1	79.98749902	3	33.2565783	2	57.38466694	232.3215
0	61.69278726	2	35.39774004	1	45.58508528	3	37.4966652	180.1723
0	61.69278726	2	35.39774004	3	33.2565783	1	57.38466694	187.7318
0	61.69278726	3	53.23532662	1	45.58508528	2	57.38466694	217.8979
0	61.69278726	3	53.23532662	2	33.66006536	1	37.4966652	186.0848
1	94.38749917	0	49.28488612	2	44.41846463	3	37.4966652	225.5875
1	94.38749917	0	49.28488612	3	33.2565783	2	57.38466694	234.3136
1	94.38749917	2	35.39774004	0	33.66006536	3	37.4966652	200.9419
1	94.38749917	2	35.39774004	3	33.2565783	0	40.16217126	203.2039
1	94.38749917	3	53.23532662	0	33.66006536	2	57.38466694	238.6676
1	94.38749917	3	53.23532662	2	44.41846463	0	40.16217126	232.2035
2	43.28972164	0	49.28488612	1	45.58508528	3	37.4966652	175.6564
2	43.28972164	0	49.28488612	3	33.2565783	1	57.38466694	183.2159
2	43.28972164	1	79.98749902	0	33.66006536	3	37.4966652	194.4339
2	43.28972164	1	79.98749902	3	33.2565783	0	40.16217126	196.6959
2	43.28972164	3	53.23532662	0	33.66006536	1	36.02776707	166.213  Minimum
2	43.28972164	3	53.23532662	1	45.58508528	0	40.16217126	182.2723
3	66.12866247	0	49.28488612	1	45.58508528	2	57.38466694	218.3833
3	66.12866247	0	49.28488612	2	33.66006536	1	37.4966652	186.5703
3	66.12866247	1	79.98749902	0	33.66006536	2	57.38466694	237.1609
3	66.12866247	1	79.98749902	2	44.41846463	0	40.16217126	230.6968
3	66.12866247	2	35.39774004	0	33.66006536	1	36.02776707	171.2142
3	66.12866247	2	35.39774004	1	45.58508528	0	40.16217126	187.2737