



UNIVERSITÉ
DE MONTPELLIER



FACULTÉ DES SCIENCES
DE MONTPELLIER

Projet Images - Compte Rendu 2

Harmonie des couleurs

Andrew Mansour , Victor Onic , Clément Saperes

Faculté des Sciences de Montpellier

2024/2025
M1 Imagine

Table des matières

1	Préface	1
2	Début d'implémentation	1
2.1	Histogramme	1
2.2	Algorithme de Cohen-Or	2
3	Visualisation	3
4	Prochaines étapes	3
	Références	4

1 Préface

La première partie du travail était d'établir une division des tâches pour que les trois membres du groupe aient un travail équivalent à faire pour la semaine. La première tâche était de comprendre le format HSV et d'extraire un histogramme d'une image en ce format. La deuxième est de lire et expliquer au groupe l'article choisi avec un début d'implémentation, et la dernière est de commencer à implémenter un outil de visualisation des histogrammes et des couleurs.

Le premier algorithme que nous avons choisi d'implémenter est l'algorithme d'harmonisation des couleurs décrit par Cohen-Or et Al. [1]. Nous avons choisi ce document car la plupart des documents que nous avons consultés le mentionnent, il semble être à l'origine de la notion d'harmonisation des couleurs en traitement d'image, et ainsi, les bases de l'harmonie des couleurs et des algorithmes plutôt intuitifs sont décrits en détail.

L'implémentation du projet est faite en python, car c'est un langage couramment utilisé en traitement d'images, et qui nous laissera une plus facile implémentation éventuelle d'un algorithme par réseau de neurones convolutifs.

2 Début d'implémentation

2.1 Histogramme

Pour harmoniser une image, nous avons donc besoin de manipuler son histogramme, mais sous un format indiquant sa couleur. Il nous faut donc un histogramme de la composante H du format HSV.

C'est pourquoi nous avons implémenté une classe d'histogramme en python que nous ne manquerons pas de réutiliser, que ce soit durant le déroulement de l'algorithme ou encore pour afficher nos résultats.

Cependant, grâce aux premiers tests effectués les semaines précédentes, nous avons pu remarquer que le nombre de valeurs que pouvait prendre la composante H n'était pas le même que pour les composantes R, G, et B. On a pu observer, pour des composantes RGB sur 256 valeurs, plusieurs dizaines de milliers de valeurs possibles pour la composante H après conversion. Certaines dû à l'arrondi machine ($1/3$ est parfois arrondie 0.333333 ou à 0.33334) mais pas que. En effet, la composante H représente un angle sur le cercle chromatique, mais n'est pas forcément un degré entier. Il est donc nécessaire de faire des arrondis, et pour cela, l'histogramme prend à son initialisation le nombre de valeurs différentes qu'il peut prendre, puis chaque valeur est arrondie à la plus proche permise. Notez que, ces valeurs représentant un angle sur le cercle, la valeur 1 est congrüe à 0 qui est le rouge.

Ensuite, on n'a plus qu'à remplir l'histogramme. On peut notamment remarquer qu'en faisant tendre la taille de notre échantillon et le nombre de valeurs possibles vers l'infini, notre histogramme tendra vers une densité.



FIGURE 1 – Petit test effectué sur une image, avec les quantités de l’histogramme qui ont été converties en pourcentage afin de tester la classe

Comme on peut le voir, les couleurs dominantes sont le jaune, l’orange et une teinte de bleu, ce qui correspond bien à l’image.

2.2 Algorithme de Cohen-Or

Cet algorithme décrit une méthode pour faire converger un histogramme vers un modèle de couleurs donné. Ces modèles sont appelés les "Harmonic Templates" , ou modèles harmoniques. Pour compléter l’algorithme, il fallait donc définir ces modèles en python, en utilisant une structure qui, pour chaque modèle, définit une liste de sections décrites par un angle de début et un angle de fin.

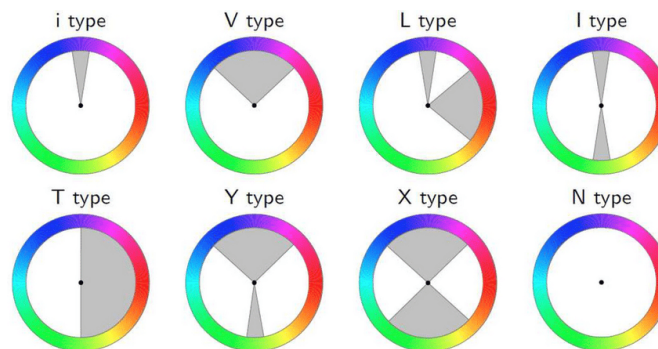


FIGURE 2 – Les 8 modèles harmoniques décrits

L’avantage de ces modèles est leur capacité à s’orienter librement. Par exemple sur la Figure 2, le type i pointe surtout vers le violet, mais si l’on a une image verte, on peut orienter cette zone pour qu’elle soit dans le vert. Ce qui se fait simplement en déplaçant le début et la fin de la zone par l’angle choisi modulo 2π , pour le tour du cercle.

Prenant en entrée une image et extrayant son histogramme, plusieurs étapes sont nécessaires. Voici alors la démarche à suivre : tout d’abord, il faut choisir le modèle et l’angle, pour simplifier le travail au début, le modèle et l’angle seront donnés par l’utilisateur, plus tard nous pourrions implémenter un choix automatique de modèle et d’angle via un calcul d’harmonie de l’image. Après ce choix, il faut assigner à chaque pixel son bord de section le plus proche, ce qui se fait avec un calcul de

distance minimale entre la position dans le cercle de la couleur du pixel, et les bords des différentes sections harmoniques. C'est uniquement cette dernière étape qui a été faite lors de cette semaine. Plus tard, une optimisation sera nécessaire afin de s'assurer que les pixels voisins de couleur assez proches seront envoyés à la même section.

Ensuite, lorsque tous les pixels seront assignés à leur bon bord de section, nous implémenterons le décalage de la couleur. Ce décalage ne peut pas être fait linéairement afin d'essayer de conserver la densité des nuances (hue) des pixels déjà présents dans la zone harmonique du modèle.

3 Visualisation

En s'inspirant du papier de Cohen-Or et al., nous visualisons l'histogramme d'une image autour d'une roue de couleurs. Cette façon est pertinente car l'espace de couleur HSV est conçu de manière circulaire. De plus, elle permet de mieux voir les rapprochements entre les valeurs obtenues et les modèles mentionnés précédemment. Nous pouvons observer les couleurs principales d'une image de manière claire et précise, avant et après traitement.

Cette visualisation se construit facilement à partir des données d'un histogramme classique. Il suffit de transformer les intervalles de teintes en valeurs angulaires. Les proportions de couleurs sont représentées comme des pics allant de l'anneau couleur vers le centre. Ce choix a été fait pour ressembler aux schémas donnés dans l'article. Cependant, d'autres choix de visualisation pourront être faits au fur et à mesure de notre avancée dans le projet, afin d'apporter de la variation.



FIGURE 3 – Image avec son histogramme (teinte) représenté sous forme de roue

4 Prochaines étapes

Maintenant, nos objectifs à court terme sont de terminer d'implémenter l'algorithme de Cohen-Or. Ensuite, de trouver une méthode d'évaluation nous permettant de déterminer quels sont les modèles, couleurs et paramètres, rendant les images les plus harmonieuses. Nous aurons ainsi de premiers résultats satisfaisants. Pour cela, nous essayerons probablement d'organiser des sondages auprès d'autres étudiants.

Références

- [1] D. Cohen-Or, O. Sorkine, R. Gal, T. Leyvand, and Y.-Q. Xu, “Color harmonization,” in *ACM SIGGRAPH 2006 Papers*, SIGGRAPH ’06, (New York, NY, USA), p. 624–630, Association for Computing Machinery, 2006.