

Intégration d'une méthode de mélange de couleurs physiquement réaliste dans un logiciel de simulation de peinture

Nicolas Barroso Valentin Chevrier Hugo Lévêque Ilyes Naidji Moearii Teraitetia
David Vanderhaeghe
Université Paul Sabatier

Novembre 2016

1 Introduction

L'idée de cette étude est de calculer la couleur physiquement réaliste résultante d'un mélange de deux pigments de peinture. Le plus courant, dans le cadre d'un affichage via moniteur, est de représenter la couleur via le format RGB qui comprend trois composantes : rouge, vert et bleu. Il serait donc pratique pour l'utilisateur de définir les couleurs des pigments de façon naturelle en RGB.

Dès lors, un problème se pose lors du calcul de la couleur du mélange. Nous allons nous baser sur le modèle de Kubelka-Munk et dans ce dernier, nous avons besoin pour chaque pigment de ses spectres d'absorption K et de diffusion S . Ces valeurs permettent de calculer la réflectance du mélange, elles caractérisent un pigment et dépendent donc de sa nature (peinture à l'huile, gouache, aquarelle...). Elles ne peuvent pas être déduites d'un simple triplet RGB mais de mesures réelles faites avec un spectromètre.

C'est pourquoi dans un premier temps, nous fixons les spectres d'absorption et de diffusion des pigments à mélanger. Nous verrons dans le paragraphe suivant qu'il est tout de même possible d'approcher une reconstitution du spectre de réflectance d'une couleur RGB mais cette reconstitution ne donnera pas forcément de bons résultats.

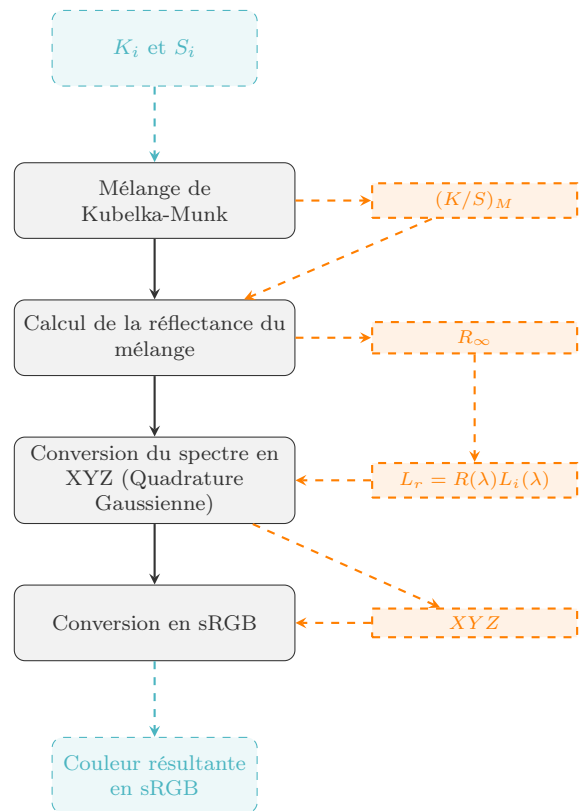


FIGURE 1 – Pipeline de rendu

2 Calcul du spectre à partir d'une couleur RGB

Brian Smits propose un algorithme capable de calculer un spectre physiquement plausible à partir d'un RGB pour les calculs de réflectance [Smits, 2000]. L'algorithme est rapide et efficace, conçu pour être utilisé dans un pipeline de rendu.

Data: Valeurs RGB
Result: Spectre lumineux
Spectre *spectreRet*;
if $R \leq G \wedge R \leq B$ **then**
 spectreRet+ = $R * spectreBlanc$;
 if $G \leq B$ **then**
 spectreRet+ = $(G - R) * spectreCyan$;
 spectreRet+ = $(B - G) * spectreBleu$;
 else
 spectreRet+ = $(B - R) * spectreCyan$;
 spectreRet+ = $(G - B) * spectreVert$;
else if $G \leq R \wedge G \leq B$ **then**
 spectreRet+ = $G * spectreBlanc$;
 if $R \leq B$ **then**
 spectreRet+ = $(R - G) * spectreMagenta$;
 spectreRet+ = $(B - R) * spectreBleu$;
 else
 spectreRet+ = $(B - G) * spectreMagenta$;
 spectreRet+ = $(R - B) * spectreRouge$;
else if $B \leq R \wedge B \leq G$ **then**
 spectreRet+ = $B * spectreBlanc$;
 if $R \leq G$ **then**
 spectreRet+ = $(R - B) * spectreJaune$;
 spectreRet+ = $(G - R) * spectreVert$;
 else
 spectreRet+ = $(G - B) * spectreJaune$;
 spectreRet+ = $(R - G) * spectreRouge$;

Algorithm 1: Algorithme de conversion RGB -> Spectre

3 Mélange de couleurs de Kubelka-Munk

Plusieurs modèles décrivent avec succès l'absorption et la diffusion de la lumière. Le plus répandu est celui de Kubelka-Munk [Kubelka and Munk, 1931]. Du fait de sa simplicité, ce modèle conduit à des résultats intéressants et il est toujours employé dans les systèmes d'aide à la formulation de peintures.

À partir d'un jeu de données de coefficients d'absorption K et de diffusion S de pigments, nous pouvons calculer les coefficients d'absorption et de diffusion pour n'importe quel mélange de pigments en utilisant la formule de Kubelka-Munk :

$$\left(\frac{K}{S}\right)_M = \frac{\sum_{i=1}^n K_i C_i}{\sum_{i=1}^n S_i C_i} \quad (1)$$

avec

n nombre de pigments dans le mélange

C_i concentration du i ème pigment

K_i coefficient d'absorption du i ème pigment

S_i coefficient de diffusion du i ème pigment

4 Calcul de la réflectance du mélange

Le résultat de l'équation 1 nous permet de calculer la réflectance du mélange :

$$R_\infty = 1 + \frac{K}{S} - \sqrt{\left(\frac{K}{S}\right)^2 + 2\frac{K}{S}} \quad (2)$$

Nous pouvons alors déduire la courbe spectrale du mélange.

5 Echantillonnage et conversion RGB

Dans cette partie, nous allons voir de quelle façon échantillonner les spectres pour rendre possible le calcul temps réel du mélange des couleurs dans notre pipeline graphique. En effet, nous souhaitons limiter le nombre d'échantillons à traiter dans l'optique de pouvoir afficher en temps réel les modifications apportées à une peinture de grande taille. Nous allons dans un premier temps considérer les spectres comme continus et la quadrature gaussienne nous permettra de résoudre les équations en échantillonnant ces spectres.

5.1 Spectre du mélange

Après l'application de la formule de Kubelka-Munk, nous obtenons la réflectance du mélange $R(\lambda)$. Pour obtenir le spectre de la lumière réfléchi $L_r(\lambda)$, il nous faut définir le spectre de la lumière incidente $L_i(\lambda)$ (par exemple, la CIE Standard Illuminant D65) :

$$L_r(\lambda) = R(\lambda)L_i(\lambda) \quad (3)$$

5.2 Conversion du spectre en coordonnées XYZ

Pour convertir le spectre de lumière réfléchi en notation XYZ, nous utilisons [Hughes et al., 2013] :

$$X = k \int L_r(\lambda) \bar{x}(\lambda) d\lambda$$

$$Y = k \int L_r(\lambda) \bar{y}(\lambda) d\lambda$$

$$Z = k \int L_r(\lambda) \bar{z}(\lambda) d\lambda$$

avec \bar{x} , \bar{y} , \bar{z} les fonctions de « color matching » et k une constante définie par le CIE.

5.3 Quadrature Gaussienne

Pour résoudre les intégrales X , Y , et Z , nous allons utiliser une méthode de quadrature gaussienne [Meyer, 1988] :

$$\int_a^b f(x)w(x)dx = \sum_{i=0}^n H_i f(x_i)$$

Pour trouver les abscisses x_0, x_1, \dots, x_n associés à la fonction de poids $w(x)$, nous devons trouver une famille de polynômes $P_n(x)$ orthogonaux à $w(x)$ sur l'intervalle (a, b) .

Si on définit le produit scalaire

$$\langle g, h \rangle = \int_a^b g(x)h(x)w(x)dx$$

Alors deux polynômes seront orthogonaux si

$$\langle P_n(x), P_m(x) \rangle = 0$$

Les polynômes sont définis récursivement par :

$$\begin{aligned} P_{k+1}(x) &= (x - A_k)P_k(x) - B_k P_{k-1}(x) \quad (k = 0, 1, \dots) \\ P_{-1}(x) &= 0 \\ P_x &= 1 \end{aligned}$$

Avec :

$$\begin{aligned} A_k &= \frac{\langle xP_k(x), P_k(x) \rangle}{\langle P_k(x), P_k(x) \rangle} \quad (k = 0, 1, \dots) \\ B_k &= \frac{\langle P_k(x), P_k(x) \rangle}{\langle P_{k-1}(x), P_{k-1}(x) \rangle} \quad (k = 0, 1, \dots), B_0 = 0 \end{aligned}$$

Les abscisses x_0, x_1, \dots, x_n sont les zéros du polynôme P_{n+1} . Une fois les abscisses obtenues, on peut calculer les poids H_i :

$$\begin{aligned} H_i &= \int_a^b l_i(x)w(x)dx \quad (i = 0, \dots, n) \\ l_i(x) &= \prod_{j=0, j \neq i}^n \frac{(x - x_j)}{(x_i - x_j)} \quad (i = 0, \dots, n) \end{aligned}$$

5.4 Quadrature Gaussienne appliquée au calcul des coordonnées XYZ

Dans le cadre de la quadrature gaussienne, nous pouvons formuler nos équations XYZ ainsi :

$$\begin{aligned} X &= k \int_a^b f(\lambda)w_X(\lambda)d\lambda = k \sum_{i=0}^n H_i^X f(\lambda_i^X) \\ Y &= k \int_a^b f(\lambda)w_Y(\lambda)d\lambda = k \sum_{i=0}^n H_i^Y f(\lambda_i^Y) \\ Z &= k \int_a^b f(\lambda)w_Z(\lambda)d\lambda = k \sum_{i=0}^n H_i^Z f(\lambda_i^Z) \end{aligned}$$

$$\begin{cases} \text{Avec } f(\lambda) = R(\lambda) \text{ et les fonctions de poids} \\ w_X(\lambda) = L_i(\lambda)\bar{x}(\lambda) \\ w_Y(\lambda) = L_i(\lambda)\bar{y}(\lambda) \\ w_Z(\lambda) = L_i(\lambda)\bar{z}(\lambda) \end{cases}$$

Les fonctions de poids dépendent du spectre de la lumière incidente. Les longueurs d'onde λ_i seront donc sélectionnées en fonction de la lumière incidente, ce qui est un comportement désiré : si la lumière tend vers le rouge par exemple, les longueurs d'onde sélectionnées pour échantillonner le spectre seront décalées vers le rouge apportant ainsi une meilleure précision pour l'évaluation de l'intégrale.

On peut choisir comme bornes d'intégration $a = 300nm$ et $b = 830nm$ ce qui correspond au spectre visible. Nous voulons retenir 8 échantillons, nous posons donc $n = 7$.

Pour résoudre ces intégrales, nous allons d'abord chercher les abscisses pour chaque équation comme indiqué dans 5.3. Nous obtenons $\lambda_i^X, \lambda_i^Y, \lambda_i^Z$ ($i = 0, 1, \dots, 7$). Nous avons donc $8 * 3$ longueurs d'onde, ce qui est un problème car nous voulons échantillonner notre spectre avec 8 échantillons seulement. Nous allons donc moyenner les longueurs d'onde obtenues pour chaque indice i :

$$\lambda_i = \frac{\lambda_{i,X} + \lambda_{i,Y} + \lambda_{i,Z}}{3} \quad (i = 0, 1, \dots, 7)$$

Nous calculons ensuite les poids H_i^X, H_i^Y, H_i^Z comme indiqué en 5.3 en utilisant les abscisses λ_i . Le calcul des intégrales XYZ se ramènera alors à :

$$\begin{aligned} X &= k \sum_{i=0}^n H_i^X R(\lambda_i) \\ Y &= k \sum_{i=0}^n H_i^Y R(\lambda_i) \\ Z &= k \sum_{i=0}^n H_i^Z R(\lambda_i) \end{aligned}$$

5.5 Conversion sRGB

On peut convertir directement les coordonnées XYZ en sRGB :

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 3.2410 & -1.5374 & -0.4986 \\ -0.9692 & 0.18760 & 0.0416 \\ 0.0556 & -0.2040 & 1.057 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

6 Connaissance de l'existant

Nous avons trouvé principalement du code existant sur internet que nous pourrions adapter ou reprendre. Le risque majeur est que le code ainsi récupéré engendre des erreurs suite à des cas non

prévus à notre application ou des erreurs algorithmiques. Pour réduire ce problème, nous effectuerons des tests afin de vérifier que le code correspond bien à notre besoin et qu'il vérifie nos contraintes.

- Mélange de couleur selon Kubelka-munk (java)
<https://github.com/benjhollla/ColorMixer>

- Conversion XYZ vers RGB avec correction gamma par John Walker (C)
<http://www.fourmilab.ch/>

Références

- [Baxter et al., 2004] Baxter, W., Wendt, J., and Lin, M. C. (2004). Impasto : a realistic, interactive model for paint. In *Proceedings of the 3rd international symposium on Non-photorealistic animation and rendering*, pages 45–148. ACM.
- [Haase and Meyer, 1992] Haase, C. S. and Meyer, G. W. (1992). Modeling pigmented materials for realistic image synthesis. *ACM Transactions on Graphics (TOG)*, 11(4) :305–335.
- [Hughes et al., 2013] Hughes, J. F., van Dam, A., McGuire, M., Sklar, D. F., Foley, J. D., Feiner, S. K., and Akeley, K. (2013). *Computer graphics : principles and practice (3rd ed.)*. Addison-Wesley Professional, Boston, MA, USA.
- [Kubelka and Munk, 1931] Kubelka, P. and Munk, F. (1931). An article on optics of paint layers.
- [Meyer, 1988] Meyer, G. W. (1988). Wavelength selection for synthetic image generation. *Computer Vision, Graphics, and Image Processing*, 41(1) :57–79.
- [Smits, 2000] Smits, B. (2000). An rgb to spectrum conversion for reflectances.