Traitement d'images Couleur

par

Luc Brun

— 2013 —

Table des matières

Chapitre 1

Introduction

L'étude de la couleur est très différente de celle de la lumière. Cet état de fait est résumé par Kandinsky lorsqu'il déclare : *"La couleur est la touche. L'oeil est le marteau. L'âme est le piano aux cordes nombreuses. ...Il est donc clair que l'harmonie des couleurs doit reposer uniquement sur le principe de l'entrée en contact efficace avec l'âme humaine"*. En effet, si l'impression de couleur est bien provoquée par la lumière, phénomène physique mesurable, ce signal est traité et transformé par notre oeil et notre cerveau pour aboutir à la sensation finale de couleur. L'ensemble des transformations subies par le signal initial est si complexe que les mesures de l'énergie lumineuse rentrant dans l'oeil ne rendent que très imparfaitement compte de la perception colorée. Il est donc important de saisir que toute mesure de la couleur ne permet que d'approximer la vision humaine. Il est ainsi parfois inutile de complexifier un algorithme, si l'accroissement de ses performances n'est pas perceptible visuellement.

Les études menées en neuro-sciences et en colorimétrie permettent d'appréhender plusieurs points importants. Le signal lumineux qui excite l'oeil peut être considéré comme un mélange d'ondes électromagnétiques sinusoïdales se propageant toutes à la vitesse de la lumière c ($c \approx 3 * 10^8 m/s$). Ces ondes ou composantes spectrales peuvent être caractérisées par leurs longueur d'onde λ et leurs puissance P_λ. Un signal lumineux peut donc être décrit par une fonction $P(\lambda)$ qui associe à chaque longueur d'onde la puissance du signal associé. Ce signal lumineux excite des récepteurs photo-sensibles situés sur la rétine.

Les cellules photo-sensibles sont composées de cônes et bâtonnets. Les bâtonnets sont sensibles à de faibles niveaux de lumière mais n'atteignent le maximum de leur puissance qu'à des intensités lumineuses modérées. Les bâtonnets permettent une vision mono-chromatique ; ce sont eux qui interviennent lors de la vision nocturne. Les cônes ont une sensibilité assez faible et sont responsables de la vision diurne. Les cônes se répartissent en trois familles de sensibilités spectrales différentes (L pour *Long*, M pour *Middle* et S pour *Short*). Chaque type de cône est donc sensible à une gamme de longueur d'onde particulière et est associé à la perception d'une couleur. Ainsi, si l'oeil reçoit une lumière de faible longueur d'onde (aux alentours de 450 nanomètres), donc si l'on excite essentiellement les cônes S, le sujet percevra une couleur bleue. De même si l'oeil reçoit une lumière de longueur d'onde moyenne (entre 500 et 600 nm), il percevra une couleur verte. Enfin les longues longueurs d'ondes (entre 600 et 700 nm) correspondent à la perception du rouge. La combinaison des signaux émis par ces trois types de cônes permet la vision colorée. Cette décomposition du spectre lumineux en trois composantes est à la base de la colorimétrie. Le but de la colorimétrie est de décrire un ensemble de couleurs à l'aide de plusieurs (généralement trois) composantes réelles. Ces composantes sont habituellement déduites de la décomposition spectrale de la lumière.

Nous allons dans ce document décrire l'origine de la perception colorée à l'aide de modèles physiques (voir chapitre 2) puis expliquer, au chapitre 3, une partie des mécanismes du cerveau et de l'oeil humain utilisées pour percevoir les images. Nous verrons ensuite au chapitre 4 comment ces résultats sont exploités pour définir des quantités permettant de définir simplement une couleur.

Chapitre 2

Physique de la couleur

2.1 Les ondes électromagnétiques

La couleur est un phénomène physiologique provoqué par l'excitation de photorecepteurs situés sur la rétine par une onde électromagnétique. Une onde éléctromagnétique est induite par les électrons des atomes d'un objet émetteur. Lorsqu'un atome est excité, certains de ses électrons vont passer d'un niveau d'énergie donné á un niveau plus faible, cette perte d'énergie s'accompagne de l'émission d'une onde électromagnétique perçue par l'oeil. Une onde électromagnétique est, comme son nom l'indique, composée de deux champs électriques $E(M,t)$ et magnétiques $B(M,t)$. Ces ondes varient donc en fonction du point M d'observation et du temps t.

- Une champ electrostatique E (Volt mètre^{-1}) est créé par la presence de particules charges. Un point matériel M possédant une charge Q, exerce sur tout autre point situé à une distance r et possédant une charge q une force f égale à :

  $$\overrightarrow{f} = \frac{1}{4\pi\epsilon}\frac{qQ}{r^2}\overrightarrow{u}$$

  où $\overrightarrow{u}$ est porté par la droite portant les deux points. Le sens de $\overrightarrow{u}$ dépent des signes respectifs de q et Q.
  L'influence electrostatique du point M est alors mesuré par son champ $\overrightarrow{E} = \frac{1}{4\pi\epsilon}\frac{Q}{r^2}\overrightarrow{u}$. On a donc

  $$\overrightarrow{f} = q\overrightarrow{E}$$

  Pour toute particule de charge q dans l'espace. De même, on obtient :

$E = \frac{1}{4\pi\epsilon}\int\int\int_V \frac{\rho dv}{r^2}\overrightarrow{u}$	Pour une densité volumique de charge ρ
$E = \frac{1}{4\pi\epsilon}\int\int_S \frac{\sigma ds}{r^2}\overrightarrow{u}$	Pour une densité surfacique de charge σ
$E = \frac{1}{4\pi\epsilon}\int_L \frac{\lambda dl}{r^2}\overrightarrow{u}$	Pour une densité linéique de charge λ

- Le champ magnétique B est créé par une circulation de charges, donc par un courrant électrique. Le champ magnétique induit en un point M situé à une distance r d'un fil indéfini parcouru par un courrant I est égal à :

$$B = \frac{\mu}{2\pi}\frac{I}{r}$$

Une particule chargée q de vitesse $\overrightarrow{v}$ soumise à un champ $\overrightarrow{B}$ subit alors une force donnée par :

$$\overrightarrow{f} = q\overrightarrow{v} \wedge \overrightarrow{B}$$

Une onde électromagnétique ne doit pas se voir comme la superposition de deux ondes indépendantes E et B, mais comme une seule entité les relations entre les ondes E et B étant exprimées par les équations de Maxwell :

– Equation du flux conservatif

$$div(B) = 0$$

– Equation de Maxwell-Faraday

$$rot(E) = \frac{\partial B}{\partial t}$$

– Equation de Maxwell-Gauss

$$div(E) = \frac{\rho}{\epsilon}$$

où ϵ représente la permitivité du milieu considéré (permitivité du vide : $\epsilon_0 = \frac{1}{36\pi 10^9}$) et ρ la densité volumique de charge (Table 2.1).

– Equation de Maxwell-Ampère

$$rot(B) = \mu(i + \epsilon\frac{\partial E}{\partial t})$$

où μ représente la constante de perméabilité du milieu (perméabilité du vide : $\mu_0 = 4\pi 10^{-7}$) et i la densité volumique de courant (Table 2.1).

Nom	Symbole	Unité	Valeur dans le vide
Perméabilité magnétique	μ	$\frac{C^2}{N.m^2}$	$4\pi 10^{-7}$
Permitivité électrique	ϵ	$\frac{N.sec^2}{C^2}$	$(36\pi 10^9)^{-1}$
Conductivité	σ	$(\Omega.m)^{-1}$	0

Tab. 2.1 – Constante électro-magnétiques

Une description des différents opérateurs différentiels utilisés se trouve en Annexe (Chapitre 7). Les deux premières équations lient les champs E et B alors que les deux dernières expriment les relations entre ces champs et les phénomènes (densité de charge ou courant) qui en sont à l'origine. La permitivité et la perméabilité du vide sont reliés par l'équation :

$$\epsilon_0\mu_0 c^2 = 1$$

où $c \approx 3.10^8 m/s$ représente la vitesse de la lumière.

Si l'on utilise les théorèmes d'Ostrogradsky et de Stokes, les équations de Maxwell peuvent s'exprimer sous forme d'intégrales :

- Equation du flux conservatif

$$\iint_S B.dS = 0$$

Le champ magnétique est à flux conservatif sur n'importe qu'elle surface fermée.

- Equation de Maxwell-Faraday

$$\int_C E.dl = -\frac{d}{dt}\iint_S B.dS$$

Un champ magnétique variable (dans le temps) engendre un champ électrique à circulation non conservative.

- Equation de Maxwell-Gauss

$$\iint_S E.dS = \frac{1}{\epsilon}\iiint_\rho \rho d\tau$$

Le champ E est a flux non conservatif sur une surface fermée. Ce flux est proportionnel à la somme des charges qui l'ont engendré.

- Equation de Maxwell-Ampère

$$\int_C B.dl = \mu\iint_S (i + \epsilon\frac{\partial E}{\partial t})dS$$

Un champ électrique variable et/ou une circulation de courant engendre un champ magnétique à circulation non conservative.

où S est une surface quelconque s'appuyant sur C et τ un volume quelconque s'appuyant sur C.

2.1.1 Energie d'une onde électromagnétique

La puissance par mètre carré ($W.m^{-2}$) d'une onde éléctromagnétique est exprimée par le vecteur de Poynting défini par :

$$R = \frac{E \wedge B}{\mu}$$

décrit la direction suivant laquelle s'écoule l'énergie électromagnétique. La densité volumique d'énergie est quant à elle définie par :

$$\varpi = \frac{\epsilon E^2}{2} + \frac{B^2}{2\mu}$$

Ces deux quantités sont reliées par l'équation suivante :

$$div(R) + \frac{\partial\varpi}{\partial t} + iE = 0$$

dans le cas d'un milieu sans courant de conduction, cette équation devient :

$$div(R) = -\frac{\partial \varpi}{\partial t}$$

La puissance rayonnée par un champ électromagnétique à travers une surface fermée peut dont s'exprimer alternativement comme la dérivée par rapport au temps de l'énergie volumique ou comme le flux du vecteur de poyting à travers cette même surface :

$$\mathcal{P} = \iint_S RdS$$

L'oeil humain étant sensible à la puissance du champ électromagnétique par la longueur d'onde, l'impression colorée est provoquée par le flux de vecteurs de Poyting associées à différentes longueurs d'ondes(voir Figure 2.1) sur la surface des recepteurs de l'oeil. L'ensemble des longueurs d'ondes auxqu'elles est sensible l'oeil humain se situe entre 360 et 830 nano-mètres. On peut donc représenter une onde arrivant sur l'oeil par une courbe (voir figure 2.2) donnant la puissance associée à chaque longueur d'onde λ.

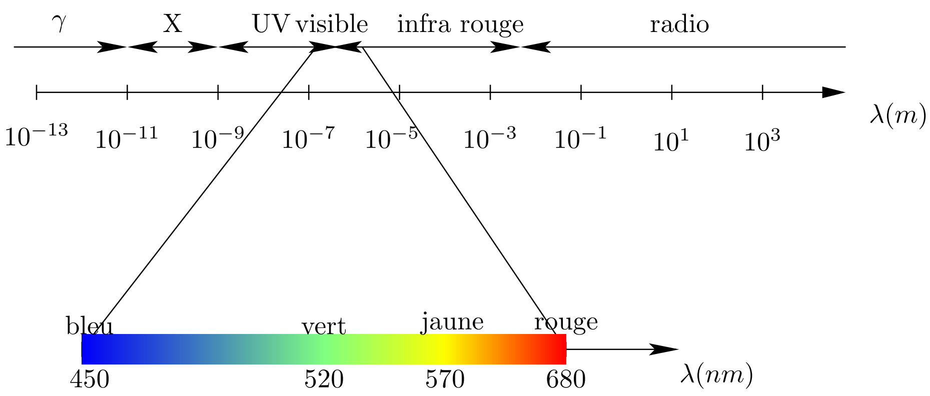


FIG. 2.1 – *Cette figure représente le classement des ondes électromagnétiques en fonction de leurs longueurs d'ondes. Le spectre visible se situe entre 450 et 680 nanomètres(*10^{-9} *mètres) et représente donc une tres faible part du spectre*

Les impressions de couleur en fonction des longueurs d'ondes se répartissent approximativement comme suit :

- Autour de 450 nm : Impression de bleu
- Entre 500 et 570 nm : impression de vert
- Entre 570 et 600 nm : impression de Jaune
- Entre 600 et 700 nm : impression de rouge.

La longueur d'onde de l'énergie électromagnétique détermine donc la chrominance alors que la puissance de l'onde détermine la luminance.

Chaque source lumineuse (lampe, néon, soleil) possède son propre spectre. Les spectres suivants sont reconnus comme des standards par la commission internationale de l'éclairage (CIE) :

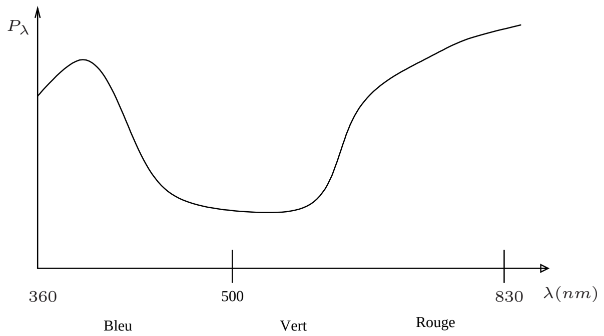


FIG. 2.2 – *Un spectre lumineux donnant la puissance en fonction de la longueur d'onde. Ce spectre est légèrement binomial avec des maximums dans les basses et hautes longueurs d'ondes. Il devrait donc correspondre à une couleur pourpre.*

– D65 et D50 correspondent à la lumière du jour, liés à des températures de 6500 et 5000 K.
– A correspond à une lampe à incandescence à 2856 K.

L'irradiance d'une onde électromagnétique est définie comme la quantité d'énergie de l'onde par unité de surface ($W.m^{-2}$) :

$$I = \frac{d\Phi_i}{dA}$$

où $d\Phi_i$ (W) représente le flux d'énergie et dA (m^2) un élément de surface.

La *radiance* ($W.m^{-2}.sr^{-1}$) d'un élément de surface dans une direction (θ_r, ψ_r) est définie comme la quantité d'énergie émise par la surface par unité de surface et unité d'angle solide. La radiance d'un patch de surface dA dans la direction (θ_r, ψ_r) est donc définie par :

$$L = \frac{d^2\Phi_r}{dA\cos(\theta_r)d\omega_r}$$

où $d\omega_r$ est l'angle solide sous lequel le patch de surface voit l'observateur et $d^2\Phi_r$ l'énergie émise dans le cône défini par $d\omega_r$.

Horn [Hor86] à montré que la radiance émise par un patch de surface était proportionnelle à l'irradiance à l'entrée des capteurs de la caméra. Plus précisément ces deux quantitées sont liées par :

$$I_r = L_r \frac{\pi}{4} \left(\frac{d}{f}\right)^2 \cos(\gamma)$$

où d et f et γ sont des paramètres de la caméra illustrés sur la Figure 2.8. Dans la plupart des applications nous pourrons supposer $\gamma = 0$.

Finallement la *réflectance* (sr^{-1}) d'un matériau est égale au rapport entre la radiance émise par un patch de surface dans une direction et l'irradiance reçu par ce même patch de surface à partir d'une autre direction

$$R = \frac{L}{I}.$$

La réflectance est souvent également désignée par BRDF (Bi Directional Reflectance Distribution Function). Du point de vue d'un utilisateur de la physique, la BRDF est sans doute la quantité la plus utile puisqu'elle nous indique ce que renvoie une surface en fonction de ce quelle reçoit. Les différentes quantitées mentionnées ci-dessus sont résumés dans la Table 2.2.

Nom	Symbole	Définition	Unité
Flux d'énergie	$d\Phi$		W
Irradiance	I	$\frac{d\Phi}{dA}$	$W.m^{-2}$
Radiance	L	$\frac{d^2\Phi}{dA\cos(\theta_r)d\omega_r}$	$W.m^{-2}.sr^{-1}$
Réflectance	R	$\frac{L}{I}$	sr^{-1}

TAB. 2.2 – Quantités utilisées en électro-magnétisme. Les symboles W, m et sr désignent respectivement des Watts, mètres et stéradians

2.1.2 Ondes planes

Une onde électromagnétique est définie par deux champs $E(M,t)$, et $B(M,t)$ fonctions de 4 variables (trois coordonnées spaciales et une temporelle). Une onde électromagnétique est qualifiée de plane lorsque ses coordonnées spaciales ne dépendent que d'un seul paramètre. On obtient alors deux champs $E(x,t)$ et $B(x,t)$ fonction d'une coordonnée spaciale et d'une coordonnée temporelle (voir Figure 2.3).

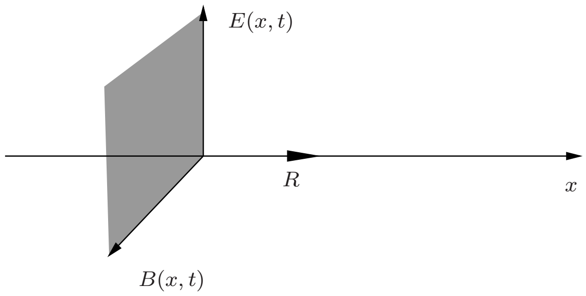


FIG. 2.3 – *Illustration d'une onde électromagnétique plane. L'onde se propage dans ce cas sur l'axe des x*

Si une onde électromagnétique se propage dans un milieu dépourvu de charges et de courant ($\rho = 0$ et $i = 0$), les équations de Maxwell se simplifient et permettent d'obtenir deux équations differentielles décrivant l'évolution des champs

E et B :

$$\triangle E = \frac{1}{c^2}\frac{\partial^2 E}{\partial t^2}, \quad \triangle B = \frac{1}{c^2}\frac{\partial^2 B}{\partial t^2}$$

Appliquées aux ondes planes ces équations deviennent :

$$\frac{\partial^2 E}{\partial x^2} = \frac{1}{c^2}\frac{\partial^2 E}{\partial t^2}, \quad \frac{\partial^2 B}{\partial x^2} = \frac{1}{c^2}\frac{\partial^2 B}{\partial t^2}$$

dont la solution générale est :

$$\begin{array}{rcl} E(x,t) & = & E_1(t-\frac{x}{c}) + E_2(t+\frac{x}{c}) \\ B(x,t) & = & B_1(t-\frac{x}{c}) + B_2(t+\frac{x}{c}) \end{array}$$

où (E_1, B_1) et (E_2, B_2) sont deux champs électromagnétiques se propageant en sens inverse sur l'axe des x à la vitesse c.

Si $\overrightarrow{u_x}$ définit le vecteur unitaire de propagation de l'onde plane, les équations de Maxwell nous permettent encore une fois de lier les champs $E(x,t)$ et $B(x,t)$:

$$E(x,t) = cB(x,t) \wedge \overrightarrow{u_x}, \quad B(x,t) = \frac{1}{c}\overrightarrow{u_x} \wedge E(x,t)$$

Les champs E et B d'une onde plane vérifient donc les propriérés suivantes :
- Ils sont transversaux ($E \perp \overrightarrow{u_x}$ et $B \perp \overrightarrow{u_x}$)
- Ils sont orthogonaux ($E \perp B$)
- Les vecteurs $(E, B, \overrightarrow{u_x})$ forment un trièdre direct
- Le rapport des modules des champs E et B est constant et égal à c ($\frac{||E||}{||B||} = c$).

Le vecteur de Poyting R et la densité volumique d'énergie ϖ sont égaux à :

$$R = \frac{E^2}{c\mu}\overrightarrow{u_x}, \quad \varpi = \epsilon E^2 = \frac{B^2}{\mu}$$

2.1.3 Onde planes mono-chromatiques

Une onde plane sinusoidale monochromatique de longueur d'onde λ se déplaçant dans le vide à la vitesse c est définie par l'équation :

$$E(x,t) = E_0 \cos(\frac{2\pi}{\lambda}(ct - x))$$

si nous introduisons le vecteur d'onde $\overrightarrow{k} = \frac{2\pi}{\lambda}\overrightarrow{u_x}$, nous obtenons :

$$E(x,t) = E_0 \cos(\frac{2\pi}{\lambda}ct - \frac{2\pi}{\lambda}x) = E_0 \cos(\omega t - \overrightarrow{k}.\overrightarrow{r})$$

avec $\omega = \frac{2\pi c}{\lambda}$ et $r = \overrightarrow{OM}$.

Le champ magnétique est alors donné par :

$$B(x,t) = \frac{1}{c}\overrightarrow{u_x} \wedge E(x,t) = \frac{\overrightarrow{k} \wedge E(x,t)}{\omega}$$

Les ondes électromagnétiques monochromatiques planes sont plus aisément manipulables lorsque l'on travaille en notation complexe ; on crée donc les champs complexes :

$$\mathcal{E} = E_0 e^{j(wt-kr)} \text{ et } \mathcal{B} = B_0 e^{j(wt-kr)}$$

où $j^2 = -1$ représente le nombre imaginaire pur.

Dans ce cas, les champs électriques et magnétiques représentent la partie réelle de ces expressions complexes. Les dérivées par rapport au temps et à l'espace s'expriment alors aisément par :

$$\frac{\partial \mathcal{E}}{\partial t} = jw\mathcal{E} \text{ et } div(\mathcal{E}) = \frac{\partial \mathcal{E}}{\partial r} = -jk\mathcal{E}$$

Les équations de Maxwell dans le vide se simplifient alors de la façon suivante :

- Equation du flux conservatif : $k.\mathcal{B} = 0$
- Equation de Maxwell-Faraday : $k \wedge \mathcal{E} = \omega\mathcal{B}$
- Equation de Maxwell-Gauss : $k.\mathcal{E} = 0$
- Equation de Maxwell-Ampère : $k \wedge \mathcal{B} = -\frac{\omega}{c^2}\mathcal{E}$

.

2.1.4 Réflection d'une onde plane monochromatique sur un métal parfait

Soit une surface Σ de densité surfacique de courant j_s et de densité surfacique de charge σ séparant deux milieux 1 et 2. Un champ électromagnétique heurtant la surface peut être décomposé en ces composantes tangencielles (E_T, B_T) et normales (E_N, B_N) à la surface. Le lien entre le champ de part et d'autre de la surface est donné par les équations suivantes :

$$\left\{ \begin{array}{rcl} E_{T_1} - E_{T_2} & = & 0 \\ B_{T_2} - B_{T_1} & = & \mu j_s \wedge n_{1\to 2} \end{array} \right. \text{ et } \left\{ \begin{array}{rcl} E_{N_1} - E_{N_2} & = & \frac{\sigma}{\epsilon} n_{1\to 2} \\ B_{T_2} - B_{T_1} & = & 0 \end{array} \right. \tag{2.1}$$

On observe donc une continuité de la composante tangencielle du champ électrique et de la composante normale du champ magnétique lors du passage entre les deux milieux. En revanche, la composante normale du champ électrique et tangencielle du champ magnétique sont discontinues. Le champ magnétique est continue uniquement dans le cas d'une densité de charge σ et d'une densité de courant j_s nulles.

Un métal parfait est un métal de conductivité $\gamma = \frac{i}{E}$ infinie. Aucune source de champ magnétique, et donc aucun champ ne peut exister à l'intérieur de ce métal. On a donc :

$$\rho = i = 0 \text{ et } E = B = 0$$

Lorqu'un champ magnétique rencontre une surface plane d'un tel métal, le champ magnétique incident à la surface du métal est égal au champ magnétique incident plus le champ magnétique réfléchi. Aucun champ ne pouvant exister à l'intérieur du métal, on obtient par les équations 2.1 :

$$\begin{array}{rcl} E_i((0,y,z),t) + E_r((0,y,z),t) & = & 0 \\ B_i((0,y,z),t) + B_r((0,y,z),t) & = & \mu j_s \wedge n \end{array}$$

en supposant que le plan $x = 0$ définit la surface de séparation entre le métal et le vide et que les vecteurs (E_i, B_i) soient inclus dans le plan Oyz. Notez que les champs électriques incident et réfléchi s'annulant à tout instant à la surface du métal doivent avoir la même fréquence, donc la réflexion sur un métal parfait ne change pas la couleur de la lumière incidente.

Soient $E_i = E_0 \cos(wt - kx)$ et $B_i = B_0 \cos(wt - kx)$ les champs électriques et magnétiques d'une onde plane monochromatique. Le champ magnétique réfléchis est donné par $E_r = -E_0 \cos(wt + kx)$ et $B_r = B_0 \cos(wt + kx)$. L'onde résultante en tout point est donc égale à :

$$\left\{ \begin{array}{lllllll} E & = & E_i + E_r & = & E_0 \cos(wt - kx) - E_0 \cos(wt + kx) & = & 2E_0 \sin(wt) \sin(kx) \\ B & = & B_i + B_r & = & B_0 \cos(wt - kx) + B_0 \cos(wt + kx) & = & 2B_0 \cos(wt) \cos(kx). \end{array} \right.$$

Les composantes spaciales et temporelles des champs magnétiques et électriques sont décorrelées. On dit dans ce cas que l'onde est stationnaire (voir Figure 2.4). Le vecteur de Poynting de l'onde résultante est égal à :

$$R = \frac{E \wedge B}{\mu} = \frac{E_0^2}{\mu c} \sin(2kx) \sin(2wt) u_x.$$

La densité volumique d'énergie est donnée par :

$$\varpi = \frac{\epsilon E^2}{2} + \frac{B^2}{2\mu}.$$

La moyenne temporelle de cette onde est indépendante de la coordonnée spaciale x :

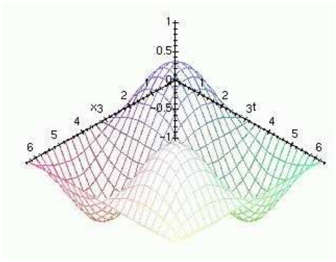
$$< \varpi >= \epsilon E_0^2.$$

Etant donnés une onde électromagnétique (E, B) et un plan d'incidence défini par $z = 0$, l'on dira que :

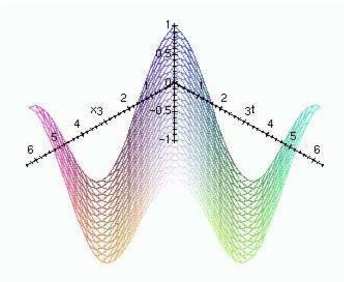
- L'onde est polarisée perpendiculairement au plan d'incidence si le champ E est perpendiculaire au plan vectoriel défini par $\overrightarrow{x}$ et $\overrightarrow{z}$.
- L'onde est polarisée perpendiculairement au plan d'incidence si le champ E est inclu dans le plan vectoriel défini par $\overrightarrow{x}$ et $\overrightarrow{z}$.

Cette notion de polarisation peut sembler un peut abstraite pour un être humain qui ne percoit pas la polarisation de la lumière. Elle est toutefois très concrète pour une grande quantité d'insectes qui sont sensibles à cette propriété. On peut notamment montrer que les différentes parties du ciel sont polarisées différement celon l'heure de la journée (et donc la position du soleil). Cette propriété permet à de nombreux insectes de s'orienter même par ciel couvert. On peut également montrer que de nombreuses ailes de papillons polarisent la lumière. Cette propriété permet notamment aux papillons de ce différencier entre sexes et entre espèces voisines lors de la reproduction.

Soit une onde électromagnétique plane d'équation $E_i(x, t) = E_0 cos(\omega t - kr)$ heurtant une surface $z = 0$ avec un angle d'incidence θ. La partie $z \leq 0$ étant occupée par un métal parfait, on peut montrer [Lum95] que l'onde réfléchie a pour équation :



(a)



(b)

FIG. 2.4 – *Tracé des deux fonctions* $z = \sin(x)\sin(t)$*(a) et* $z = cos(x - t)$*(b). Ces deux tracés ne représentent pas à proprement parler des l'amplitude d'une onde élecromagnétique, mais nous renseigne sur l'allure de cette amplitude dans le cas d'une onde stationnnaire (a) et non stationnaire (b)*

– Si l'onde électromagnétique est polarisée perpendiculairement :

$$\left\{ \begin{array}{lcl} E_r & = & -E_0 \cos(\omega t - xksin(\theta) + zkcos(\theta))\overrightarrow{y} \\ B_r & = & -\frac{E_0}{c} cos(\omega t - xk\sin(\theta) + zk\cos(\theta))(\cos(\theta)\overrightarrow{x} + \sin(\theta)\overrightarrow{z}) \end{array} \right.$$

– Si l'onde électromagnétique est polarisée parallèlement :

$$\left\{ \begin{array}{lcl} E_r & = & -E_0 \cos(\omega t - xk\sin(\theta) + zk\cos(\theta))(\cos(\theta)\overrightarrow{x} + \sin(\theta)\overrightarrow{y}) \\ B_r & = & \frac{E_0}{c} \cos(\omega t - xksin(\theta) + zkcos(\theta))\overrightarrow{y} \end{array} \right.$$

Si nous ajoutons l'onde incidente à l'onde réfléchie, nous obtenons :

– Dans le cas perpendiculaire :

$$E = 2E_0 \sin(zk\cos(\theta)).\sin(\omega t - xk\sin(\theta))\overrightarrow{y}$$

et :

$$B = \begin{pmatrix} -\frac{2E_0}{c}\cos(\theta)\cos(zk\cos(\theta))\cos(\omega t - xk\sin(\theta)) \\ 0 \\ \frac{2E_0}{c}\sin(\theta)\sin(zk\cos(\theta))\sin(\omega t - xk\sin(\theta)) \end{pmatrix}$$

– Dans le cas parallèle :

$$E = \begin{pmatrix} 2E_0\cos(\theta)\sin(zk\cos(\theta))\sin(\omega t - xk\sin(\theta)) \\ 0 \\ -2E_0\sin(\theta)\cos(zk\cos(\theta))\cos(\omega t - xk\sin(\theta)) \end{pmatrix}$$

et

$$B = \frac{2E_0}{c}\cos(zk\cos(\theta)).\cos(\omega t - xk\sin(\theta))\overrightarrow{y}$$

Dans le cas général, où l'onde n'est polarisée ni perpendiculairement ni parallèlement, le vecteur E peut être décomposé en deux composantes polarisées perpendiculairement et parallelement :

$$\begin{array}{rcl} E_{\parallel} & = & E_0 \cos(\alpha) \\ E_{\perp} & = & E_0 \sin(\alpha) \end{array}$$

Ce qui nous donne les ondes réfléchies :

$$E = \begin{pmatrix} 2\cos(\alpha)E_0 \cos(\theta) \sin(zk\cos(\theta)) \sin(\omega t - xk\sin(\theta)) \\ 2\sin(\alpha)E_0 \sin(zk\cos(\theta)). \sin(\omega t - xk\sin(\theta)) \\ -2\cos(\alpha)E_0 \sin(\theta) \cos(zk\cos(\theta)) \cos(\omega t - xk\sin(\theta)) \end{pmatrix}$$

et

$$B = \begin{pmatrix} -\sin(\alpha)\frac{2E_0}{c} \cos(\theta) \cos(zk\cos(\theta)) \cos(\omega t - xk\sin(\theta)) \\ \cos(\alpha)\frac{2E_0}{c} \cos(zk\cos(\theta)). \cos(\omega t - xk\sin(\theta)) \\ \sin(\alpha)\frac{2E_0}{c} \sin(\theta) \sin(zk\cos(\theta)) \sin(\omega t - xk\sin(\theta)) \end{pmatrix}$$

Ces dernières formules appellent plusieurs remarques :

- La fréquence du signal réfléchie n'est pas modifié (on reste en wt).
- Le facteur d'atténuation du signal réfléchie ne dépend pas de la fréquence de l'onde incidente.

Donc le métal parfait renvera la même quantitée d'énergie quelque soit la longueur d'onde de la lumière incidente. Celui ci se comporte donc comme un mirroir et n'est utile qu' en tant que modèle lorsque l'on veut modéliser les caractéristiques optiques des matériaux. Nous allons à présent étudier des modèles un peu plus réaliste.

2.2 Différents modèles de réflexion

Les mécanisme d'absorption -réflexion d'une onde électromagnétisme peuvent s'expliquer par les lois de la mécanique classique (celle des ondes électromagnétiques) ou les lois de la mécanique quantique. Attardons nous un instant sur cette dernière approche qui permet de donner une bonne intuition du phénomène. Une onde électromagnétique de longueur d'onde λ peut d'un point de vue quantique, se voir comme la propagation d'un ensemble de photons d'énergie $h\frac{c}{\lambda} = h\nu$, où h est la constante de plank. Lorsqu'un photon heurte un électron il y transmet son énergie et l'électron passe d'un état au repos à un état excité. L'ensemble des électrons susceptible de changer d'état et la quantité d'énergie que peut absorber chaque électron dans un milieu donné caractérise les longueurs d'ondes du champ électromagnétique incident qui seront absorbées et donc la couleur du milieu.

Le calcul exact du spectre réfléchi requiert donc une connaissance extrêmement précise des caractéristiques du milieu qui est rarement disponible. De plus comme nous l'avons rapidement abordé dans le paragraphe précédent le spectre réfléchie dépend de la géométrie de l'objet qui est souvent connue que d'un point de vue macroscopique quand elle n'est pas totalement inconnue (le plan infini est une abstraction mathématique).

Du fait de ces limitations de nombreux modèles ont été établis afin de rendre compte ne fusse que de façon approximative (voir empirique) des phénomènes de réfraction - absorption.

Il est relativement clair que de nombreux matériaux réfléchissent la lumière de façon très différente. Un pot de céramique et de cuivre auront par exemple des propriétés optiques très différentes. Ces différents types de réflexions seront donc caractérisés par différents modèles décrivant différent types de matériaux. On distingue notamment [Sha85, Hea89, Hun75] :

1. Les matériaux conducteurs comme les métaux. Comme nous le verrons par la suite ces matériaux atténuent rapidement l'onde incidente si bien que le phénomène de réflexion est essentiellement un phénomène de surface.
2. Les matériaux peu conducteurs également appelés *diélectriques* tels que le verre. Ces matériaux laissent au contraire pénétrer profondément l'onde incidente dans l'objet. La description de la réflexion exige donc pour ces matériaux une modélisation des phénomènes optiques dans le matériau.
3. Les matériaux optiquement homogènes. Ces matériaux ont un indice de réfraction constant à l'intérieur du matériau. Pour ce type d'objet la réflexion de l'onde incidente peut être décrite uniquement à partir de la réflexion de l'onde sur la surface du matériau. Les métaux, le verre, les cristaux sont des exemples communs de matériaux homogènes.
4. Les matériaux optiquement in-homogènes sont composés d'un matériau qui inclue de nombreuses particules colorantes dont les propriétés optiques sont différentes de celles du support. Dans ce cas, le calcul de l'onde réfléchie doit tenir compte de l'interaction de l'onde incidente avec les particules de colorant. Les plastiques, le papier, les textiles et les peintures font partie des matériaux optiquement in-homogènes.

Notez que les points 1 et 2 indiquent si le phénomène de réflexion est un phénomène de surface. Les points 3 et 4 décomposent les matériaux en fonction des phénomènes optiques se produisant à l'intérieur de ceux ci.

Un autre paramètre important d'une surface est sa rugosité. Ainsi, des matériaux homogènes et parfaitement lisses réfléchissent la lumière dans une direction symétrique au rayon incident par rapport à la normale. Ce phénomène est appelé une *réflexion spéculaire*. Inversement des matériaux homogènes plus rugueux diffusent la lumière autour de la réflexion spéculaire. L'ensemble des rayons réfléchis est appelé le *lobe spéculaire*.

L'électromagnétisme permet de décrire les propriétés optiques d'un matériau à l'aide des constantes μ, ϵ et σ décrivant respectivement la perméabilité magnétique, la permittivité électrique et la conductivité du matériau (Table 2.1). Les propriétés optiques d'un matériau peuvent également être résumées à l'aide d'une seule variable appelée *l'indice complexe de réfraction* et noté $M = n - iK_0$ où n et K_0 sont deux réels. La constante n est appelée la part réfractive de M. Dans des matériaux n'atténuant pas le signal ($K_0 = 0$), la constante n est égale au rapport entre la vitesse de la lumière dans le vide et dans le matériau. La constante

K_0 est directement impliquée dans l'atténuation de l'onde électromagnétique dans le matériau. En effet l'irradiance d'une onde planaire de fréquence ω de direction $x > 0$ qui heurte le plan (Ozy) est égale à :

$$I(x) = I(0)e^{-\dfrac{2\omega K_0 x}{c}}$$

où c est la vitesse de la lumière dans le vide.

Cette atténuation de l'énergie de l'onde incidente ce fait au bénéfice de l'apparition d'un courant appelé *courant de surface*. La profondeur $x = \frac{c}{2\omega K_0}$ est appelé la profondeur de peau du matériau. Les constantes n et K_0 peuvent être calculée en fonction de μ, ϵ et σ en utilisant les lois de Maxwell [SH81] :

$$\begin{aligned} n(\lambda) &= \frac{\mu\epsilon c^2}{2}\left[1 + \sqrt{1 + \left(\frac{\lambda\sigma}{2\pi c\epsilon}\right)^2}\right] \\ K_0(\lambda) &= \frac{\mu\epsilon c^2}{2}\left[-1 + \sqrt{1 + \left(\frac{\lambda\sigma}{2\pi c\epsilon}\right)^2}\right]. \end{aligned} \tag{2.2}$$

Nous pouvons immédiatement observer que n et K_0 (et donc M) ne dépendent que de la longueur d'onde de l'onde incidente. De plus pour des matériaux conducteurs σ sera important et donc K_0 également. L'onde électromagnétique pénètre donc faiblement dans les matériaux conducteurs pour lesquels le phénomène de la réflexion est essentiellement un phénomène lié à la surface du matériau. Inversement, des matériaux peu conducteurs auront un K_0 faible ce qui favorisera la pénétration de l'onde dans le matériau. Ce phénomène devra donc être pris en compte dans la modélisation de la réflexion.

Un dernier paramètre fondamental dans la description de la réflexion d'une onde électromagnétique est le *Coefficient de Fresnel* qui décrit la fraction de l'onde incidente réfléchie par la surface d'un matériau. Dans le cas d'un matériau isotropique et homogène, la réflexion d'une onde non polarisée heurtant une surface lisse avec un angle θ_l produit un coefficient de Fresnel égal à :

$$F(\theta_l, \lambda) = \frac{1}{2}\left(R_{\|}(\theta_l, \lambda) + R_{\perp}(\theta_l, \lambda)\right)$$

où $R_{\|}(\theta_l, \lambda)$ et $R_{\perp}(\theta_l, \lambda)$ décrivent le coefficient de Fresnel dans le cas d'une onde polarisée respectivement parallèlement et perpendiculairement au plan d'incidence (plan contenant la normale à la surface et le rayon incident).

Les termes $R_{\|}(\theta_l, \lambda)$ et $R_{\perp}(\theta_l, \lambda)$ peuvent se déduire de la théorie des ondes électromagnétiques. Toutefois la forme explicite de ces termes n'étant pas utile pour la suite de ce document nous nous contenterons de noter que $R_{\|}(\theta_l, \lambda)$ et $R_{\perp}(\theta_l, \lambda)$ peuvent s'exprimer sous forme d'une fraction de termes dépendant de θ_l, $n(\lambda)$ et $K_0(\lambda)$.

2.2.1 Le modèle Lambertien

Le modèle Lambertien est le modèle le plus utilisé pour décrire les phénomène de réflexion à l'intérieur des diélectriques. Ce modèle permet de relier l'irradiance incidente à un capteur à l'angle θ formé par l'onde incidente et la normale à la surface (Figure 2.2.1) à l'aide de l'équation suivante :

$$I = K_{diff}\cos(\theta) \tag{2.3}$$

où K_{diff} est une constante dépendant du matériau.

Notez que dans un tel modèle la direction de l'observateur n'est pas prise en compte. Intuitivement, un objet Lambertien sera donc un objet dont la couleur ne varie pas lorsque l'on se déplace. Du bitume en milieu de journée fournit un bon exemple de surface Lambertienne.

Une explication qualitative de ce modèle généralement acceptée est la suivante : l'énergie incidente à une surface pénètre dans celle-ci et est réfléchie aléatoirement à l'intérieur de l'objet par de microscopiques in-homogénéités du matériau. Au cours de ces multiples réflexions une partie de l'énergie incidente est ré-émise par la surface et ressort de l'objet suivant une direction aléatoire. Les réflexions multiples dans le matériau ne subissant aucune contrainte particulière, l'énergie est ré-émise de façon uniforme par la surface. L'intensité de l'énergie émise par un point est donc indépendante de la direction d'observation et uniquement fonction de la quantité d'énergie incidente tombant sur la surface. Cette quantité s'exprime comme un cosinus de l'angle entre la normale à la surface et la direction de la source.

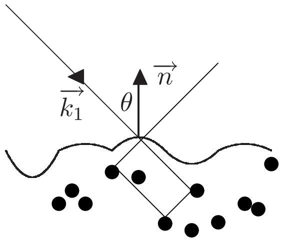


Fig. 2.5 – Réflexion Lambertienne : l'onde incidente se propage dans la direction définie par le vecteur $\vec{k_1}$. Elle est réfléchi plusieurs fois à l'intérieur du matériaux avant de ressortir de celui-ci suivant une direction aléatoire.

Une explication plus quantitative du modèle Lambertien peut être donnée dans le cas d'un diélectrique in-homogène. Ce type de matériau a été étudié par Reichmann [Rei73] qui à étendu un modèle initialement développé par Kubelka et Munk [KL31]. Reichmann modélise la répartition des pigments de colorants dans le matériau à l'aide d'un ensemble de couches superposées. L'interaction entre chaque couche élémentaire et l'onde incidente est décrite par les fonctions $\alpha(\lambda)$ et $\beta(\lambda)$ qui décrivent respectivement la fraction de l'onde incidente absorbée et réfléchie par unité de longueur.

La théorie de Kubelka et Munk repose sur la résolution de plusieurs équations différentielles du premier ordre dont la résolution fournit un coefficient de réflectance donné par :

$$R_\infty = \frac{2 - \omega(\lambda) - 2\sqrt{1-\omega(\lambda)}}{\omega(\lambda)} \text{ avec } \omega(\lambda) = \frac{\beta(\lambda)}{\alpha(\lambda)+\beta(\lambda)}$$

Toutefois, le modèle de Kubelka et Munk repose sur plusieurs hypothèses non réalistes dans le cadre d'applications réelles. Il suppose notamment que le matériau et l'air possèdent le même indice de réfraction afin d'éviter d'avoir à considérer les réflections à la surface du matériau. Ceci est visible dans la formule de R_∞ qui ne dépend d'aucun paramètre géométrique.

L'extension proposée par Reichmann permet de lever ces limitations et nous donne une expression de la réflectance plus générale :

$$R_B(\theta,\lambda) = (1-R_S)\frac{C(\theta,\lambda)(1-r_i(\lambda))\,(R_\infty(\lambda)-D(\theta))}{2\,(1-r_i(\lambda)R_\infty(\lambda))\cos(\theta)} \tag{2.4}$$

où R_S est la fraction de l'onde incidente réfléchie par la surface, $(1 - R_S)$ est donc la fraction de l'onde qui pénètre dans le matériau. Le terme $r_i(\lambda)$ représente la réflectance de surface interne du matériau [Orc69] tandis que les fonctions $C(\theta, \lambda)$ et $D(\theta)$ viennent de la résolution des équations du modèle et sont données par :

$$\begin{array}{rcl} C(\theta,\lambda) & = & \frac{\omega(\lambda)\cos(\theta)(2\cos(\theta)+1)}{1-4(1-\omega(\lambda))\cos^2(\theta)} \\ D(\theta) & = & \frac{2\cos(\theta)-1}{2\cos(\theta)+1} \end{array}$$

L'équation 2.4 semble beaucoup plus complexe que l'équation Lambertienne (équation 2.3). Toutefois dans le cas de matériaux n'absorbant pas l'onde incidente ($\alpha(\lambda) \approx 0$) nous avons $\omega(\lambda) \approx 1$ et le lecteur peut vérifier que l'équation 2.4 se réécrit $R_B(\theta, \lambda) = (1 - R_S)$. La partie de l'onde qui n'est pas réfléchie par la surface est donc renvoyée uniformément indépendamment de la longueur d'onde et de l'angle θ formé par l'onde incidente et la normale à la surface. On retrouve donc bien dans ce cas l'explication qualitative de la réflexion Lambertienne donnée au début de cette section. Notez toutefois que pour avoir une irradiance à l'entrée du capteur fonction uniquement de $\cos(\theta)$ nous devons supposer le terme $(1 - R_S)$ approximativement constant. Cette hypothèse n'est pas valable dans le cas des matériaux conducteurs (voir par exemple la Section 2.2.2). Notons de plus que quelque soit le type du matériau celui ci acquiert un comportement spéculaire pour un angle d'incidence rasant ($\theta \approx \frac{\pi}{2}$). Dans ce cas le modèle Lambertien n'est plus valable. Ce phénomène explique par exemple, les éblouissements dues à la réverbération du soleil sur le bitume au coucher du soleil.

2.2.2 Le modèle de Beckmann-Spizzichino

Le modèle de Beckmann-Spizzichino [BS87] est basé sur les lois de l'électromagnétisme et sur une modélisation des micro-aspérités de la surface.

Beckmann modélise les micro-aspérités d'une surface à l'aide d'une fonction aléatoire h des variables x et y décrivant la surface. La forme de la surface est donc déterminée par la densité de probabilité de la fonction h. Beckmann propose d'utiliser une distribution normale de moyenne nulle et d'écart type σ_h. La distribution de h est alors égale à :

$$\rho_h(h) = \frac{1}{\sqrt{2\pi}\sigma_h} e^{-\frac{h^2}{2\sigma_h^2}}.$$

Toutefois, une telle modélisation ne permet pas un contrôle efficace de la forme de la surface. En effet, plusieurs tirages aléatoires de la fonction h avec le même σ_h peuvent présenter un aspect très différent. Intuitivement, la raison de cette limitation est que le paramètre σ_h contrôle l'altitude moyenne des pics de la fonction h mais pas l'écart entre deux pics (Figure 2.6).

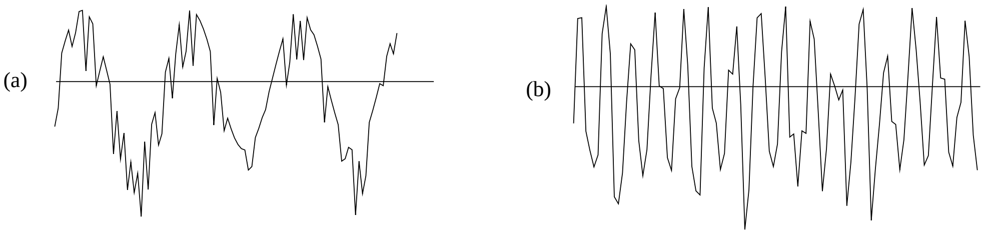


FIG. 2.6 – Deux surfaces aléatoires de valeur σ_h identiques avec une forte (a) et une faible (b) distance de corrélation

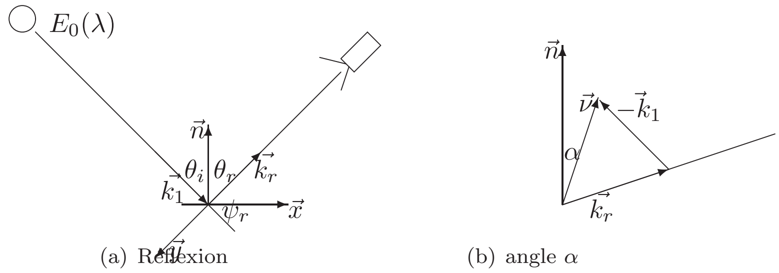
Afin de pallier à cette limitation, nous définissons un paramètre $C(\tau)$ qui représente la corrélation entre les altitudes de deux points séparés par une distance τ. Beckmann propose de représenter cette corrélation par la fonction :

$$C(\tau) = e^{-\frac{\tau^2}{T^2}}$$

où T est la *distance de corrélation* pour laquelle $C(\tau)$ est égal à e^{-1}.

Étant donnée notre modélisation de la surface par les paramètres σ_h et T, la position d'un patch de surface vis à vis de la source lumineuse et de la caméra est représentée sur la Figure 2.7(a). Le repère est choisi tel que l'axe z soit confondu avec la normale $\overrightarrow{n}$ et l'axe x est tel que le vecteur $\overrightarrow{k_1}$ décrivant la direction de la source lumineuse soit dans le plan (Ozx) avec une projection positive sur l'axe x. La puissance de la source lumineuse est décrite par la norme de son vecteur de Poynting $P(\lambda) = \frac{1}{2}\sqrt{\frac{\mu}{\epsilon}}E_0^2(\lambda)$. Le vecteur $\vec{k_r}$ décrit la direction de la caméra tandis que le vecteur $\vec{\nu}$ est défini comme la bissectrice de $-\overrightarrow{k_1}$ et $\overrightarrow{k_r}$. Ce vecteur fait un angle α avec la normale à la surface (Figure 2.7(b)). Notez que le vecteur $\overrightarrow{k_r}$ n'est pas nécessairement dans le même plan que $\overrightarrow{n}$ et $\overrightarrow{k_1}$. L'angle ψ_r représente son écart vis-à-vis de ce plan (Figure 2.7(a)).

Les paramètres σ_h, T et les variables illustrées sur la Figure 2.7 étant définis, Nayar [NIK91] en se basant sur les travaux de Beckmann, a montré que l'irradiance à l'entrée d'un capteur dA_{im} de la caméra provoquée par un patch de



(a) Réflexion (b) angle α

FIG. 2.7 – Schéma de réflexion

surface d'un conducteur parfait de cotés X et Y (Figure 2.8) pouvait être exprimée par :

$$\begin{aligned} I_r(\lambda) = & \frac{1}{2}\sqrt{\frac{\mu}{\epsilon}}E_o(\lambda)^2\frac{\pi}{4}\left(\frac{d}{f}\right)^2\cos^4(\gamma)\frac{\cos^2(\theta_i)}{\lambda^2}e^{-g} \\ & \left(\left(\frac{z}{f}\right)^2\frac{dA_{im}\cos(\gamma)}{\cos^2(\theta_r)}\rho_0^2 + \frac{\pi T^2 D^2 g}{cos(\theta_r)}\sum_{m=1}^{m=\infty}\frac{g^m}{m!m}e^{-\upsilon_{xy}^2\frac{T^2}{4m}}\right) \end{aligned} \tag{2.5}$$

Les paramètres d, f et γ dans l'équation 2.5 sont relatifs à la caméra et illustrés sur la Figure 2.8. Notez que pour une caméra placée raisonnablement loin de la scène nous pouvons supposer $\gamma \approx 0$. Les termes g, ρ_0, ν_{xy} et D sont issus de la résolution de l'intégrale d'Helmhotz et sont égaux à :

$$\left\{\begin{array}{lcl} g & = & \left(2\pi\frac{\sigma_h}{\lambda}(\cos(\theta_i) + cos(\theta_r))\right)^2 \\ \rho_0 & = & \frac{\sin(\nu_x X)}{\nu_x X}\frac{\sin(\nu_y Y)}{\nu_y Y} \\ \nu_{xy} & = & \sqrt{\nu_x^2 + \nu_y^2} \\ D & = & \frac{1+\cos(\theta_i)\cos(\theta_r) - \sin(\theta_i)\sin(\theta_r)\cos(\psi_r)}{\cos(\theta_i)(\cos(\theta_i)+\cos(\theta_r))} \end{array}\right. \tag{2.6}$$

où $\overrightarrow{\nu} = (\nu_x, \nu_y, \nu_z)$ (Figure 2.7(b)).

Le paramètre g est proportionnel au carré du rapport $\frac{\sigma_h}{\lambda}$ qui représente le rapport entre l'altitude moyenne de nos irrégularités et la longueur d'onde de la source incidente. Le facteur g représente donc la rugosité de la surface et les cas $g << 1$, $g \approx 1$ et $g >> 1$ représenteront respectivement une surface lisse, modérément rugueuse et rugueuse.

Le terme ρ_0 est une fonction qui décroît très rapidement dès que $\nu_x X$ ou $\nu_y Y$ n'est pas proche de 0. Les coordonnées ν_x et ν_y sont égales à 0 lorsque $\overrightarrow{\nu}$ est confondu avec $\overrightarrow{n}$. Dans ce cas, $\overrightarrow{n}$ est la bissectrice des vecteurs sources et destination $-\overrightarrow{k_1}$ et $\overrightarrow{k_r}$. Ce type de réflexion est appelé une réflexion spéculaire tandis que le terme décrivant ce type de réflexion est appelé un *pic spéculaire*. Le premier terme de l'équation 2.5 correspond donc à une fonction pic dont le maximum est atteint pour une réflexion spéculaire.

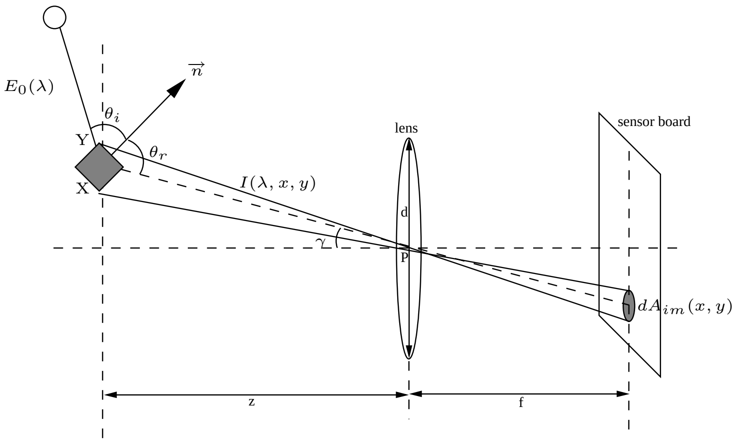


FIG. 2.8 – *Réflexion d' une onde électromagnétique frappant un des capteurs de la caméra*

Le second terme de l'équation 2.5 est appelé *le lobe spéculaire* ; ce terme correspond à une fonction qui prend son maximum lorsque la direction d'observation correspond à la direction spéculaire mais décroît plus lentement que le pic spéculaire. Notez tout de même le facteur $e^{-\upsilon_{xy}^2 \frac{T^2}{4}}$ qui décroît rapidement lorsque $\overrightarrow{k_r}$ ne correspond pas à la direction spéculaire ($\vec{\nu} = (0, 0, 1)$).

L'équation 2.5 se simplifie en fonction de la rugosité de la surface de la façon suivante :

– Pour une surface lisse ($g << 1$) :

$$I_{lisse}(\lambda) = \frac{1}{2}\sqrt{\frac{\mu}{\epsilon}}\frac{E_o(\lambda)^2}{\lambda^2}\cos^2(\theta_i)e^{-g}\left(\frac{K_{ps}}{\cos^2(\theta_r)}\rho_0^2 + \frac{K_{ls}D^2 g}{cos\theta_r}e^{-\upsilon_{xy}^2\frac{T^2}{4}}\right) \tag{2.7}$$

– Pour une surface rugueuse ($g >> 1$) :

$$I_{rugueux}(\lambda) = K_{ls}\frac{1}{2}\sqrt{\frac{\mu}{\epsilon}}\frac{E_o(\lambda)^2}{\lambda^2}\frac{e^{\left(\frac{-\upsilon_{xy}^2 T^2}{4\upsilon_z^2\sigma_h^2}\right)}}{\upsilon_z^2\sigma_h^2}\cos(\theta_i)D^2 \tag{2.8}$$

où K_{ps} etK_{ls} sont deux constantes.

Notez que pour $g = 0$, le lobe spéculaire est nul alors que le pic spéculaire est maximum. Au fur et à mesure que la rugosité de la surface augmente, le pic spéculaire diminue jusqu'à disparaître tandis que le lobe spéculaire devient prépondérant.

Une description plus qualitative du pic et du lobe spéculaire sera donnée dans la Section 2.2.4 ; Nous pouvons toutefois dès à présent indiquer les conditions de validité de l'équation 2.5 :

1. Les irrégularités de la surface sont supposées suivre une distribution uniforme de moyenne nulle. Cette hypothèse est raisonnable lorsque l'on ne

dispose d'aucune information à priori sur le type des irrégularités. Notons également que Beckmann [BS87] a défini l'irradiance pour de nombreux types d'irrégularités. Cette restriction peut donc être levée si l'on dispose d'informations plus précises sur la surface.

2. Le rayon de courbure des irrégularités doit être supérieur en tout point de la surface à la longueur d'onde du rayon incident. Cette restriction est indispensable pour pouvoir développer les équations physiques menant à l'équation 2.5. Les valeurs mesurées peuvent donc s'écarter de celles prédites par l'équation 2.5 si les irrégularités comportent des pics très étroits.

3. Le matériau doit être un conducteur parfait. Cette restriction est encore une fois dictée par des considérations pratiques afin de pouvoir intégrer l'intégrale d'Helmotz donnant le champ magnétique en un point P à partir du champ magnétique sur la surface. Supposer que la matériau est un conducteur parfait revient à supposer que le coefficient de Fresnel F est égal à $+1$ ou -1 selon la polarisation de l'onde. Beckmann propose d'approximer l'irradiance d'un matériau de conductivité finie en approximant le coefficient de Fresnel F en chaque point de la surface par sa moyenne $< F >$ sur la surface. L'irradiance est alors donnée par :

$$I_f(\lambda) =< FF^* > I_\infty(\lambda) \tag{2.9}$$

   où les indices f et ∞ représentent l'irradiance pour un matériau de conductivité finie et infinie tandis que F^* représente le conjugué de F. Cette solution revient à approximer une somme de produit par un produit de somme.

   Contrairement aux deux restrictions précédentes qui peuvent être ignorées dans la plupart des applications pratiques, cette dernière restriction interdit a priori d'utiliser l'équation 2.5 pour des isolants ou des matériaux de faible conductivité. Nous pouvons toutefois considérer que l'équation 2.5, sans nous donner une expression exacte de l'irradiance, nous fournit une bonne description qualitative du phénomène de réflexion à la surface d'un matériau. Cette hypothèse est confirmée par l'équation 2.9.

4. Les ombrages et masquages ne sont pas pris en compte par le modèle. On peut tenir compte de cet effet en substituant $S(x, y)h(x, y)$ à $h(x, y)$ où $S(x, y)$ est une fonction de masquage égale à 0 ou 1 selon que point est masqué ou pas. Il reste toutefois à définir dans ce cas un modèle pour la fonction S.

5. Les réflexions multiples ne sont également pas prises en compte. Encore une fois cette restriction est imposée pour obtenir une expression explicite de l'irradiance.

6. Le champ électromagnétique incident est supposé plan et polarisé perpendiculairement. Ces restrictions sont encore une fois imposées pour des raisons de simplicité. Beckmann a proposé plusieurs approches pour traiter le cas d'ondes de polarisation quelconque. L'hypothèse d'onde plane

est justifié lorsque la source lumineuse est à une distance importante de l'objet. Ceci sera vérifié lors de nos acquisitions.

2.2.3 Le modèle de Torrance-Sparrow

Contrairement au modèle de Beckmann-Spizzichino [BS87] (Section 2.2.2), le modèle de Torrance-Sparrow est basé sur l'optique géométrique. Ce modèle néglige donc l'aspect électromagnétique de la lumière. Cette approximation n'est valide que si les irrégularités de la surface sont bien supérieures à la longueur d'onde de la source.

Le modèle de surface utilisé par Torrance-Sparrow est basé sur une modélisation des irrégularités par une série de micro-facettes. Chaque facette est décrite par l'angle β entre sa normale et la normale à la surface macroscopique (Figure 2.9). Si nous supposons la surface isotropique, la distribution des normales de facettes est rotationnellement symétrique par rapport à $\overrightarrow{n}$. La distribution de β peut alors être modélisée par une fonction unidimensionnelle telle qu'une distribution normale de moyenne nulle et d'écart-type σ_α. Sachant que β ne peut varier qu'entre 0 et $\frac{\pi}{2}$, la fonction de densité de probabilité de β est égale à :

$$\rho_\beta(\beta) = ce^{-\frac{\beta^2}{2\sigma_\alpha^2}} \text{ avec } c = \left(\int_0^{\frac{\pi}{2}} e^{-\frac{\beta^2}{2\sigma_\alpha^2}} d\beta \right)^{-1}$$

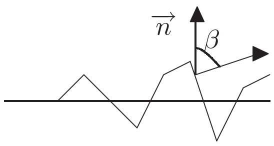


Fig. 2.9 – Modèle de surface de Torrance-Sparow

Ce modèle de surface et les lois de l'optique géométrique permettent d'obtenir une expression explicite de l'irradiance incidente à un capteur de la caméra générée par un patch de surface :

$$I = \kappa_{spec} \frac{L_i dw_i}{\cos(\theta_r)} e^{-\frac{\alpha^2}{2\sigma_\alpha^2}}$$

où α, L_i et dw_i représentent respectivement l'angle entre la normale et le vecteur $\vec{\nu}$ (Figure 2.7(b)), la radiance de la source et l'angle solide sous lequel le patch de surface voit la source. La constante κ_{spec} est donné par :

$$\kappa_{spec} = \frac{\pi}{4} \left(\frac{d}{f} \right)^2 \cos^4(\gamma) \frac{ca_f F'(\theta_i', \eta') G(\theta_i, \theta_r, \psi_r)}{4} \tag{2.10}$$

où $F'(\theta_i', \eta')$ représente le coefficient de Fresnel et $G(\theta_i, \theta_r, \psi_r)$ un facteur de visibilité entre le patch de surface et la source. Les angles θ_i, θ_r et ψ_r sont représentés sur la Figure 2.7 (Section 2.2.2). L'angle θ_i' représente l'angle entre le rayon incident et la normales aux micro-facettes susceptibles d'éclairer le capteur. La variable η' représente l'indice complexe de réfraction tandis que a_f représente la surface d'une micro-facette.

Le terme $e^{-\frac{\alpha^2}{2\sigma_\alpha^2}}$ a approximativement la même signification que le terme $e^{-v_{xy}^2 \frac{T^2}{4}}$ dans le modèle de Beckmann-Spizzichino (Section 2.2.2) et correspond à un lobe spéculaire. La modélisation de la réflexion de Torrance-Sparrow en utilisant des micro-facettes et les lois de l'optique géométrique conduisent donc à un modèle ne présentant qu'un lobe spéculaire. Ce résultat est attendu dans la mesure ou les lois de l'optique géométriques ne sont valides que pour des surfaces rugueuses. Or le pic spéculaire de Beckmann-Spizzichino n'apparaît que pour des surfaces lisses ou modérément rugueuses (Section 2.2.2).

Torrance et Sparrow ajoutent un terme Lambertien à leur équation de réflexion qui devient :

$$I = \kappa_{diff} L_i dw_i \cos(\theta_i) + \kappa_{spec} \frac{L_i dw_i}{\cos(\theta_r)} e^{-\frac{\alpha^2}{2\sigma_\alpha^2}} \text{ pour } \theta_i \in [0, \frac{\pi}{2}], 0 \text{ sinon.} \quad (2.11)$$

où κ_{diff} représente le coefficient de réflexion diffuse (ou Lambertienne).

L'utilisation de l'optique géométrique conduit à des formules moins complexes que celles induites par les lois de l'électromagnétique. Torrance et Sparrow sont donc conduits à faire moins d'hypothèses simplificatrices que Beckmann et Spizzichino. Leur modèle inclue notamment le coefficient de Fresnel et un coefficient de masquage. Ce modèle est donc applicable à des objets non conducteurs et permet de tenir compte du masquage entre différents éléments de la scène. Ce modèle a toutefois un certain nombre de limitations :

1. L'angle α est supposé avoir une distribution normale. Cette limitation est équivalente à la supposition d'une distribution normale de la hauteur h dans le modèle de Beckmann. Le modèle peut également être facilement adapté à d'autres types de distributions.
2. La taille des micro-facettes doit être beaucoup plus importante que la longueur d'onde du rayon incident. Cette contrainte correspond à la définition du domaine de rugosité de surface pour laquelle les lois de l'optique géométrique peuvent se substituer à celles de l'électromagnétique.
3. La source lumineuse est supposée être éloignée de la scène. Cette contrainte correspond à la modélisation par ondes planaires plutôt que sphériques dans le modèle de Beckmann.

2.2.4 Le modèle de Nayar

Le modèle de Nayar [NIK91] peut se concevoir comme une synthèse des modèles de Beckmann-Spizzichino et Torrance-Sparrow (Sections 2.2.2 et 2.2.3). Plusieurs expériences menées par Nayar montrent que le coefficient de Fresnel F et le facteur d'atténuation G du modèle de Torrance-Sparrow (équation 2.10) restent approximativement constants en fonction de θ_i et θ_r. Le coefficient κ_{spec}

peut donc être considéré comme constant. De plus si l'on se place dans un protocole expérimental où la source est variable tandis que la direction d'observation reste constante, les angles θ_r et ψ_r peuvent être considérés comme constants. Sous ces conditions, l'irradiance du lobe spéculaire peut s'exprimer par :

$$I_{ls} = K_{ls} e^{-\frac{\alpha^2}{2\sigma_\alpha^2}} \tag{2.12}$$

où K_{ls} est une constante dépendant du matériau et du protocole expérimental.

En revanche si l'on considère des variations simultanées de la source lumineuse et de l'observateur nous ne pouvons négliger le terme $1/cos(\theta_r)$ dans le modèle de Torrance-Sparrow (équation 2.10). L'expression du lobe spéculaire devient alors :

$$I_{ls} = \frac{C_{ls}}{\cos(\theta_r)} e^{-\frac{\alpha^2}{2\sigma_\alpha^2}} \text{ avec } K_{ls} = \frac{C_{ls}}{\cos(\theta_r)} \tag{2.13}$$

Notons que l'équation 2.13 devra être utilisée si l'on considère simultanément plusieurs pixels et donc plusieurs normales avec des angles θ_i, θ_r et ψ_r différents. L'équation 2.12 sera en revanche utilisée lorsque l'on considérera un même pixel soumis à différents illuminants. Dans ce dernier cas θ_i et α sont variables tandis que θ_r et ψ_r peuvent être considérés comme des constantes.

Le pic spéculaire du modèle de Beckmann-Spizzichino peut être approximé par une fonction δ valant 1 dans la direction spéculaire et 0 partout ailleurs. L'intensité du pic spéculaire est alors égale à :

$$I_{ss} = K_{ss}\delta(\theta_i - \theta_r)\delta(\psi_r)$$

où K_{ss} est égale à la valeur du pic spéculaire (équation 2.5) dans la direction spéculaire.

Finalement, le lobe diffus correspondant à la réflexion Lambertienne peut être ajouté au modèle de façon à avoir une intensité de pixel liée à la géométrie de la scène par :

- Si $\theta_i \in [0, \frac{\pi}{2}]$,

$$I = K_{diff}\cos(\theta_i) + K_{ls} e^{-\frac{\alpha^2}{2\sigma_\alpha^2}} + K_{ss}\delta(\theta_i - \theta_r)\delta(\psi_r) \quad \text{Observateur fixe} \tag{2.14}$$

$$I = K_{diff}\cos(\theta_i) + \frac{C_{ls}}{\cos(\theta_r)} e^{-\frac{\alpha^2}{2\sigma_\alpha^2}} + K_{ss}\delta(\theta_i - \theta_r)\delta(\psi_r) \quad \text{Observateur variable} \tag{2.15}$$

- Si $\theta_i \geq \frac{\pi}{2}$, $I = 0$.

Notez encore une fois que le cas d'un observateur variable (équation 2.15) peut s'appliquer soit :

1. à l'étude d'un pixel avec des positions successives de la caméra. Notons que si la source lumineuse est supposé d'orientation constante le terme $K_{diff}\cos(\theta_i)$ est dans ce cas également constant. Le cas de l'observateur variable s'applique également
2. à l'étude de plusieurs pixels avec une seule caméra fixe. Dans cas aucun des angles $\theta_i, \theta_r, \psi_r$ et α ne peut être considéré constant.

Nayar a de plus établi des ponts entre les deux modèles en remarquant que puisque α est l'angle entre $\overrightarrow{\nu}$ et la normale nous avons (équation 2.6) :

$$\tan(\alpha) = \frac{\nu_{xy}}{\nu_z}. \tag{2.16}$$

Donc si nous posons $\tan(\alpha_0) = \frac{2\sigma_h}{T}$ les lobes spéculaires des modèles de Beckmann et Torrance sont liés par :

$$e^{-\frac{\upsilon_{xy}^2 T^2}{4\nu_z^2 \sigma_h}} = e^{-\frac{\tan^2(\alpha)}{\tan^2(\alpha_0)}}. \tag{2.17}$$

En utilisant l'approximation $\tan(\alpha) \approx \alpha$ nous obtenons :

$$e^{-\frac{\upsilon_{xy}^2 T^2}{4\nu_z^2 \sigma_h}} = e^{-\frac{\alpha^2}{2\left(\frac{\alpha_0}{\sqrt{2}}\right)^2}}.$$

L'écart type σ_α du modèle de Torrance peut donc être relié aux paramètres σ_h e T du modèle de Beckmann par :

$$\sigma_\alpha = \frac{\alpha_0}{\sqrt{2}} = \frac{1}{\sqrt{2}} \tan^{-1}\left(\frac{\sigma_h}{T}\right)$$

Notez de plus que le modèle de Beckmann définit le lobe spéculaire à partir de $\tan(\alpha)$ plutôt que α. Le modèle de Torrance-Sparrow peut donc se comprendre comme une approximation du modèle de Beckmann avec $\tan(\alpha) \approx \alpha$.

La Figure 2.10 illustre le modèle de Nayar sur un cercle de rayon 1 éclairé par une source lumineuse placée en $\frac{\pi}{4}$ et observé en $\frac{\pi}{2}$ (Figure 2.10(a)). Le modèle utilisé est l'équation 2.15. Un point du cercle faisant un angle θ avec l'horizontale vérifiera (Figure 2.10(b)) :

$$\left\{\begin{array}{lcl} \theta_i & = & \theta - \frac{\pi}{4} \\ \theta_r & = & \frac{\pi}{2} - \theta \\ \alpha & = & \frac{\theta_r - \theta_i}{2} = \frac{3\pi}{8} - \theta. \end{array}\right.$$

La direction spéculaire ($\alpha = 0$) se situe donc en $\frac{3\pi}{8} \approx 1.2$.

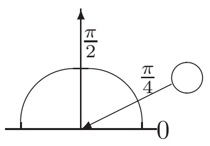
L'intensité le long du cercle en fonction de l'angle θ est représentée sur la figure 2.10(c). La contribution de chacune des composante est quand à elle illustrée sur la Figure 2.10(d). Les constantes choisies pour cette figure sont $K_{ss} = 10, C_{ls} = 5, K_{dl} = 1$ et $\sigma_\alpha^2 = \frac{1}{30}$.

2.2.5 Les modèles de Shafer et Healey

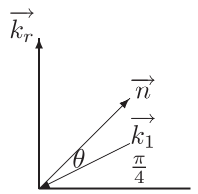
Le modèle de Shafer [Sha85] exprime la radiance d'un patch de surface comme la somme d'une composante diffuse et d'une composante spéculaire[1] :

$$\begin{aligned} L(\lambda, \theta_i, \theta_r, g) &= L_{diff}(\lambda, \theta_i, \theta_r, g) + L_{spec}(\lambda, \theta_i, \theta_r, g) &(2.18)\\ &= m_{diff}(\theta_i, \theta_r, g)c_{diff}(\lambda) + m_{spec}(\theta_i, \theta_r, g)c_{spec}(\lambda) &(2.19)\end{aligned}$$

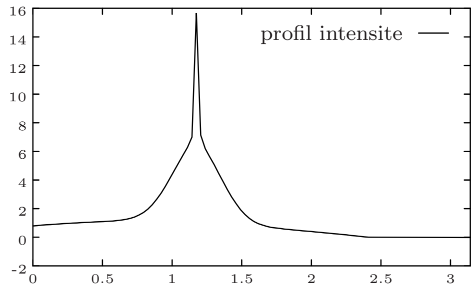
[1]Shafer préfère les termes de réflexion de l'interface et de l'intérieur du matériau aux termes de réflexion spéculaire et Lambertienne ou diffuse. Nous garderons toutefois ces dernières notations afin de rester consistant avec les notations précédentes



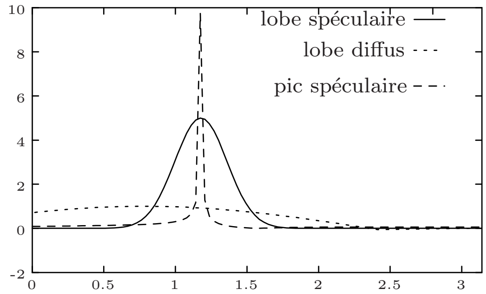
(a) Source-Observateur



(b) Un point du cercle



(c) Profil d'intensité



(d) Contribution des lobes

FIG. 2.10 – Profil d'intensité le long d'une cercle (c) et contribution des différents lobes (d). Le protocole expérimental est illustré sur les Figures (a) et (b)

où g représente l'angle entre les vecteurs $-\overrightarrow{k_1}$ et $\overrightarrow{k_r}$ (Figure 2.7).

Ce modèle est donc essentiellement qualitatif dans la mesure où il ne précise pas la valeur des fonctions m_{diff}, m_{spec} et c_{diff}, c_{spec}. Il fait toutefois deux hypothèses extrêmement fortes sur la nature de la réflexion :

1. L'irradiance incidente à un capteur est supposée pouvoir être décomposée en la somme de deux termes l'un exprimant une réflexion diffuse et l'autre une réflexion spéculaire. Une telle décomposition de l'irradiance en la somme de deux termes correspondant à deux phénomènes physiques différents a été validé par Beckmann (Section 2.2.2). Notons également que Torrance-Sparrow (Section 2.2.3) et Nayar (Section 2.2.4) font les mêmes hypothèses.
2. La seconde hypothèse, sans doute plus discutable, est que l'irradiance de chacun des termes de la somme peut être décomposée en un produit de deux termes l'un dépendant uniquement de facteurs géométriques (m_{diff} et m_{spec}) l'autre dépendant uniquement de la longueur d'onde λ (c_{diff} et c_{spec}). Une telle hypothèse a été reprise par Nayar dans le cadre de l'unification des modèles de Torrance-Sparrow et Beckmann-Spizzichino. Notons toutefois que cette supposition n'a rien d'évident au vue de l'équation de l'irradiance de Beckmann (équation 2.5).

Un autre modèle du a Healey [Hea89] exprime approximativement la même idée. Selon Healey, la réflectance d'un matériau peut être approximée par la formule suivante :

$$R(\lambda, g) = \begin{cases} M_{spec}(g)C_{spec}(\lambda) & \text{pour les métaux} \\ M_{spec}(g)C_{spec}(\lambda) + M_{diff}(g)C_{diff}(g) & \text{pour les diélectriques inhomogènes} \end{cases} \tag{2.20}$$

où g représente les facteurs géométriques tels que θ_i, θ_r et ψ_r.

On retrouve donc bien une décomposition en une réflectance spéculaire et Lambertienne avec pour chacun des termes une sous décomposition en un produit d'un terme dépendant de la géométrie et d'un terme ne dépendant que de la longueur d'onde. Toutefois, la composante Lambertienne est supprimée pour les métaux. Cette suppression s'explique par le modèle usuellement accepté pour expliquer la réflexion Lambertienne (Section 2.2.1). En effet, celle-ci est issu de réflexions à l'intérieur du matériau. Les métaux possédant une forte conductivité ont un fort coefficient d'absorption K_0 (équation 2.2), si bien que l'onde incidente pénètre peu dans le matériau et la réflexion reste un phénomène de surface décrit par le terme spéculaire. Notons toutefois que les deux modèles sont sensiblement équivalents ; en effet rien n'empêche de poser m_{diff} ou c_{diff} à 0 dans le modèle de Shafer dans le cas d'une réflexion sur un métal.

2.2.6 Conclusion

Nous avons abordé dans cette section 3 modèles décrivant les différents types de réflexion (Sections 2.2.1, 2.2.2 et 2.2.3) et 2 modèles de synthèse (Sections 2.2.4 et 2.2.5). Le modèle de Lambert (Section 2.2.1) permet de décrire les réflexions à l'intérieur d'un matériau. L'utilisation de ce modèle n'est pertinente que si une partie significative de l'onde incidente pénètre dans le matériau avant de ressortir de celui-ci. Ce modèle est donc plus adapté aux matériaux diélectriques in-homogènes. Les modèles de Beckmann et Torrance (Sections 2.2.2 et 2.2.3) décrivent quand à eux la réflexion à la surface du matériau. Ces modèles sont donc adaptés au cas où la plus grande partie de l'onde est réfléchie par la surface (cas des matériaux conducteurs) ou dans le cas où la partie de l'onde qui pénètre dans le matériau n'est pas réfléchie vers la surface (cas des matériaux homogènes). La principale différence entre les modèles de Beckmann et Torrance tient à ce que le modèle de Beckmann reste valide pour des matériaux lisse et rugueux alors que le modèle de Torrance ne peut s'appliquer pour des matériaux parfaitement lisses. La Figure 2.11 illustre les différents domaines d'applications de ces modèles. Notons toutefois que les matériaux ne sont généralement ni parfaitement conducteurs ou isolants ni parfaitement homogènes. La décomposition illustrée par cette figure reste donc schématique.

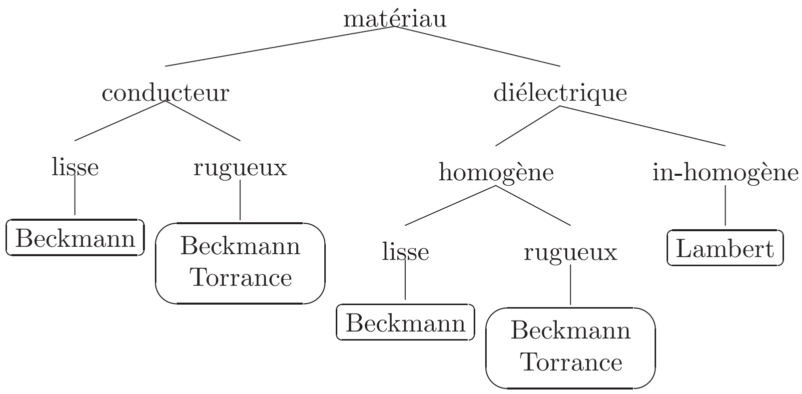


FIG. 2.11 – Utilisation des différents modèles de réflexion en fonction du type de matériau

Les modèle de Nayar (Section 2.2.4) et Safer (Section 2.2.5) décrivent respectivement le phénomène de réflexion d'un point de vue quantitatif et qualitatif. Ces modèles font une synthèse des modèles précédemment décris. Le modèle de Nayar en particulier décrit le phénomène de réflexion comme une somme de réflexions issues des modèles de Beckmann, Torrance et Lambert. Les coefficients affectés à chaque modèle de base dans le modèle de Nayar dépendent du type de matériau. Les modèles de Shafer ou Healey sont des modèles qualitatif et ne nécessitent donc pas de tels ajustements. Il faut toutefois faire attention au domaines de validités des modèles de base sur lesquels s'appuient ces modèles. Par exemple, l'utilisation de la composante Lambertienne pour des diélectrique n'est plus valide pour des sources avec un angle d'incidence rasant (Section 2.2.1).

2.3 Utilisation des spectres en Image

L'utilisation des spectre en synthèse ou traitement d'images impose tout d'abord la conversion de l'irradiance frappant les différents capteurs en triplets de valeur permettant de coder la couleur. Cette conversion est effectuée par la formule suivante qui modélise assez bien le fonctionnement des capteurs :

$$s_i(x,y) = \int_{360}^{800} f_i(\lambda) I(\lambda, x, y) d\lambda$$

où f_i représente la sensibilité du capteur i aux différentes longueurs d'ondes et $I(\lambda, x, y)$ l'irradiance frappant le capteur associé à la composante i du pixel (x, y).

Si nous négligeons les inter-réflexions, la réflection d'une onde sur un objet de réflectance R frappant un pixel de coordonnées (x, y) aura pour couleur :

$$s_i(x,y) = \int_{360}^{800} f_i(\lambda) R(\lambda, x, y) E(\lambda) d\lambda \tag{2.21}$$

où $E(\lambda)$ représente le spectre d'une source lumineuse.

2.3.1 Utilisation des spectres en synthèse d'image

L'utilisation des spectres pour la synthèse d'images est basée sur l'utilisation des fonctions de réflectance. Dans le cas du lancé de rayons, un rayons heurtant des surface de réflectance R_1, R_2,. . .,R_n avant de frapper le pixel (x, y) produira la couleur :

$$\forall i \in \{r, g, b\}\ s_i(x,y) = \int_{360}^{800} f_i(\lambda) \Pi_{j=1}^{n} R_j(\lambda) E(\lambda) d\lambda$$

notez que les termes relatifs a la géométrie des surface aux points d'impacts des rayons ont été omis pour ne pas surcharger les notations. Ils sont toutefois bien présent.

Dans le cas plus complexe de sources non ponctuelles avec des surfaces non totalement spéculaires, la puissance mono-chromatique $L'_\lambda(x, w_{xout})$ ré-émise par le matériau au point x dans la direction w_{xout} sur la longueur d'onde λ est donnée par :

$$L'_\lambda(x, w_{x_{out}}) = \int_{\mu_s} \rho_\lambda(x, w_{x_{in}}, w_{x_{out}}) I_\lambda(x_s, w_{x_{out}}) \frac{cos(\theta_{x_s})}{(x - x_s)^2} H(x_s, x) dx_s$$

où :

- $\rho_\lambda(x, w_{x_{in}})$ représente la réflectance bidirectionnelle au point x.
- $I_\lambda(x_s, w_{x_{out}})$ est la puissance du spectre dans la longueur d'onde λ quittant x_s avec un angle de , $w_{x_{out}}$
- $H(x_s, x)$est le facteur de forme entre x et x_s.

Le terme $I_\lambda(x_s, w_{x_{out}})\frac{cos(\theta_{x_s})}{(x-x_s)^2}H(x_s, x)$ de cette équation peut être considéré comme l'intensité lumineuse incidente au point x noté $E(\lambda)$ dans l'équation précédente. Alors que le terme $\rho_\lambda(x, w_{x_{in}}, w_{x_{out}})$ peut être considéré comme la réflectance du matériau sur la longueur d'onde λ. Ce terme était approximé dans l'équation précédente par $R(\lambda)$.

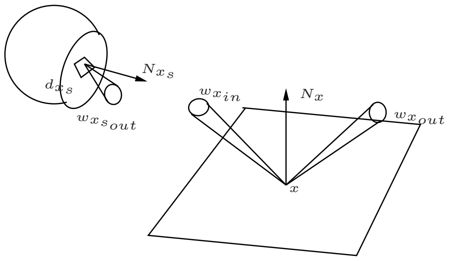


FIG. 2.12 – *Absorbtion réflexion avec source non ponctuelle*

2.3.2 Utilisation des spectres en traitement d'images

L'utilisation des spectres en traitement d'image repose beaucoup sur l'utilisation de la formule de Shafer (voir section 2.2.5) et souvent sur une approximation de celle ci pour les diélectriques qui consiste à négliger la composante spéculaire. On obtient alors :

$$R(\lambda, g) = M(g)C(\lambda)$$

aussi bien pour les métaux que pour les diélectriques. La réponse du capteur i est alors donnée par :

$$s_i(x, y) = \int_{360}^{800} f_i(\lambda)M(g)C(\lambda)E(\lambda)d\lambda$$

En traitement d'images, on cherche souvent à caractériser les objets qui compose l'image soit pour les retrouver (segmentation) soit pour les reconnaître (reconnaissance de formes). Dans cette perspective la formule précédente présente deux inconvénients majeurs. En effet, La réponse des capteurs dépend :

1. de la géométrie de l'objet. Ceci induit des problèmes lorsque l'on désire retrouver un objet indépendamment de sa géométrie.
2. de l'illuminant. Si l'on désire caractériser un objet on souhaiterais avoir une mesure indépendante des conditions d'éclairages.

le premier problème peut être résolu en faisant le rapport des coordonnées. En effet, si nous posons :

$$K_i = \int_{360}^{800} f_i(\lambda)C(\lambda)E(\lambda)d\lambda$$

qui est un terme qui ne dépend pas de la géométrie de l'objet, l'on obtient :

$$s_i(x,y) = M(g)K_i$$

autrement dit, en dénotant $s(x,y)$ le vecteur couleur :

$$s(x,y) = M(g)\begin{pmatrix} K_1 \\ K_2 \\ K_3 \end{pmatrix}$$

l'ensemble des couleurs associé a un seul matériaux parcourt donc une droite de vecteur directeur (K_1, K_2, K_3). La projection de cette droite sur le plan $s_1 + s_2 + s_3 = 1$ est égale à :

$$\frac{s_i(x,y)}{\sum_{i=1}^{3} s_i(x,y)} = \frac{K_i}{\sum_{j=1}^{3} K_j}$$

qui est un terme indépendant de la géométrie utilisé pour caractérisé le matériau. Certains auteurs préfèrent la projection sur la sphère unité donnée par :

$$\frac{s_i(x,y)}{\sqrt{\sum_{i=1}^{3} s_i^2(x,y)}} = \frac{K_i}{\sqrt{\sum_{j=1}^{3} K_j^2}}$$

La résolution du second problème est connue dans la littérature sous le nom de constance des couleurs. Ce problème assez complexe à résoudre dans le cas général se simplifie grandement si l'on dispose d'un objet de réflectance connue. Cette méthode suppose toutefois que l'on projette l'illuminant et la fonction de réflectance sur une finie de fonction ce qui induit une très forte approximation. On a alors :

$$\begin{array}{rcl} R(\lambda,x,y) & = & \sum_{i=0}^{2} \sigma_i(x,y)R_i(\lambda) \\ E(x,y,\lambda) & = & \sum_{i=0}^{2} \epsilon_i(x,y)E_i(\lambda) \end{array}$$

où $(R_i)_{i\in\{0,1,2\}}$ et $(E_i)_{i\in\{0,1,2\}}$ sont des bases de fonctions permettant la décomposition de la réflectance R et l'illuminant E.

Donné ces équations la réponse d'un capteur i associé au pixel de coordonnées (x,y) est donnée par une équation similaire à l'équation 2.21 :

$$\begin{array}{rcl} s_k(x,y) & = & \int_{360}^{800} f_k(\lambda)R(\lambda,x,y)E(\lambda,x,y)d\lambda \\ & = & \int_{360}^{800} f_k(\lambda)\left(\sum_{i=0}^{2}\sigma_i(x,y)R_i(\lambda)\right)\left(\sum_{j=0}^{2}\epsilon_j(x,y)E_j(\lambda)\right)d\lambda \\ & = & \sum_{j=0}^{2}\epsilon_j(x,y)\int_{360}^{800} f_k(\lambda)\sum_{i=0}^{2}\sigma_i(x,y)R_i(\lambda)E_j(\lambda)d\lambda \\ & = & \sum_{j=0}^{2}\Omega_{kj}^{\sigma}\epsilon_j(x,y) \end{array} \tag{2.22}$$

avec $\Omega_{kj}^{\sigma} = \int_{360}^{800} f_k(\lambda)\sum_{i=0}^{2}\sigma_i(x,y)R_i(\lambda)E_j(\lambda)d\lambda$

On obtient sous forme matricielle :

$$s(x,y) = \Omega^{\sigma}\epsilon(x,y) \tag{2.23}$$

Si nous plaçons dans la scène un objet de réflectance $(\sigma_0(x, y), \sigma_1(x, y), \sigma_2(x, y))$ connue, la détermination de l'illuminant revient simplement à inverser la matrice Ω^σ. On a de fait :

$$\epsilon(x, y) = (\Omega^\sigma)^{-1} s(x, y)$$

L'équation 2.22 peut également s'écrire sous la forme :

$$\begin{aligned} s_k(x, y) &= \int_{360}^{800} f_k(\lambda) \left(\sum_{i=0}^{2} \sigma_i(x, y) R_i(\lambda)\right) \left(\sum_{j=0}^{2} \epsilon_j(x, y) E_j(\lambda)\right) d\lambda \\ &= \sum_{i=0}^{2} \sigma_i(x, y) \int_{360}^{800} f_k(\lambda) \sum_{j=0}^{2} \epsilon_j(x, y) R_i(\lambda) E_j(\lambda) d\lambda \\ &= \sum_{i=0}^{2} \Omega_{ki}^\epsilon \sigma_i(x, y) \end{aligned}$$

avec $\Omega_{ki}^\epsilon = \int_{360}^{800} f_k(\lambda) \sum_{j=0}^{2} \sigma_j(x, y) R_i(\lambda) E_j(\lambda) d\lambda$

On obtient sous forme matricielle :

$$s(x, y) = \Omega^\epsilon(x, y) \sigma(x, y)$$

Si la valeur de ϵ est donnée en tout point (x, y) par la formule 2.23 on obtient :

$$\sigma(x, y) = (\Omega^\epsilon(x, y))^{-1} s(x, y)$$

2.4 Exercices

2.4.1 couleur et albédo

La fonction de réflectance d'un métal est, dans sa forme la plus générale fonction de 3 variables.

- λ : La longueur d'onde du spectre heurtant le métal.
- θ_{in} l'angle sous lequel le métal voit la source lumineuse (ponctuelle).
- θ_{out} l'angle sous lequel le métal voit l'observateur.

Supposons que la fonction de réflectance $R(\lambda, \theta_{in}, \theta_{out})$ puisse s'écrire sous la forme :

$$R(\lambda, \theta_{in}, \theta_{out}) = P(\lambda).G(\theta_{in}, \theta_{out})$$

où P dépend uniquement de la longueur d'onde et G, uniquement des angles θ_{in} et θ_{out}.

1. Soit $e(\lambda)$ la fonction décrivant la puissance du spectre électromagnétique de la source ponctuelle en fonction de la longueur d'onde. Donnez l'expression de la puissance du spectre réfléchi $L'(\lambda, \theta_{in}, \theta_{out})$ en fonction de e, P,G, λ, θ_{in} et θ_{out}.

2. Soit $s(\lambda)$ la fonction décrivant la sensibilité des capteurs d'une caméra monochrome en fonction de la longueur d'onde. Donnez l'expression de l'intensité d'un pixel $I(\theta_{in}, \theta_{out})$ en fonction de s et L'. On supposera que la fonction s est nulle en dehors de l'intervalle $[350, 750]$.

3. Soit une caméra couleur, on suppose que la couleur d'un pixel est définie par trois capteurs de sensibilités respectives $r(\lambda)$, $v(\lambda)$, $b(\lambda)$, mesurant respectivement la quantité de rouge, vert et bleu. On suppose de plus

que les trois capteurs sont très rapprochés, si bien qu'ils font tous le même angle θ_{out} par rapport au métal. Donnez l'expression du triplet $(R, V, B)(\theta_{in}, \theta_{out})$ en fonction de s, e, P, G, θ_{in} et θ_{out}.

4. Expliquez en quoi les rapports $\frac{R}{V}$, $\frac{V}{B}$ ou $\frac{R}{B}$ peuvent caractériser les métaux indépendamment de leurs géométrie.
5. Quelle est la courbe décrite par la couleur des pixels associés à un même métal.

2.5 Correction des Exercices

2.5.1 couleur et albédo

– Puissance du spectre réfléchi :
On applique la formule donnée dans la section 2.2.1 ce qui nous donne :

$$L'(\lambda, \theta_{in}, \theta_{out}) = e(\lambda)P(\lambda)G(\theta_{in}, \theta_{out})$$

– Intensité du pixel avec une caméra monochrome :
L'intensité finale est définie comme la somme des intensités par longueur d'onde, on a donc :

$$\begin{aligned} I(\theta_{in}, \theta_{out}) &= \int_{350}^{750} s(\lambda)L'(\lambda, \theta_{in}, \theta_{out})d\lambda \\ &= \int_{350}^{750} s(\lambda)e(\lambda)P(\lambda)G(\theta_{in}, \theta_{out})d\lambda \\ &= G(\theta_{in}, \theta_{out}) \int_{350}^{750} s(\lambda)e(\lambda)P(\lambda)d\lambda \end{aligned}$$

– Dans le cadre d'une caméra couleur, il suffit de dupliquer la dernière équation avec les sensibilités r, v et b. On obtient donc :

$$\begin{aligned} R(\theta_{in}, \theta_{out}) &= G(\theta_{in}, \theta_{out}) \int_{350}^{750} r(\lambda)e(\lambda)P(\lambda)d\lambda \\ V(\theta_{in}, \theta_{out}) &= G(\theta_{in}, \theta_{out}) \int_{350}^{750} v(\lambda)e(\lambda)P(\lambda)d\lambda \\ B(\theta_{in}, \theta_{out}) &= G(\theta_{in}, \theta_{out}) \int_{350}^{750} b(\lambda)e(\lambda)P(\lambda)d\lambda \end{aligned}$$

Le rapport $\frac{R(\theta_{in}, \theta_{out})}{V(\theta_{in}, \theta_{out})}$ est donc égal à :

$$\frac{R(\theta_{in}, \theta_{out})}{V(\theta_{in}, \theta_{out})} = \frac{\int_{350}^{750} r(\lambda)e(\lambda)P(\lambda)d\lambda}{\int_{350}^{750} v(\lambda)e(\lambda)P(\lambda)d\lambda}$$

Ce terme ne dépend plus que du spectre de la lampe, du coefficient de réflection du métal et de la sensibilité des capteurs rouges et verts de la caméra. Ce terme ne dépend donc plus de la géométrie de l'objet et doit être identique pour tous les pixels correspondant à un même métal.
Si les rapports entre les composantes sont constants, l'ensemble des pixels associés à un même métal doit décrire une droite dans l'espace (R, V, B).

Chapitre 3

Perception de la couleur

3.1 Description de l'oeil

La partie optique de la couleur se limite à l'oeil composé (entre autre) du cristallin et de la rétine (voir Figure 3.1). Le cristallin agit comme une lentille épaisse qui concentre les rayons lumineux sur la rétine. L'image inversée ainsi obtenue n'est nette que dans la partie centrale de la rétine appellée fovéa.

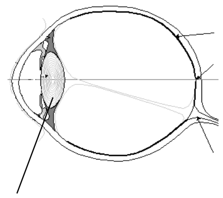


FIG. 3.1 – *shéma simplifié de l'oeil*

La rétine tient compte de cette caractéristique optique puisque les recepteurs sont beaucoup plus nombreux dans la région nette (la fovéa) que dans les régions périphériques.

3.2 Les récepteurs neuronaux

Les récepteurs neuronaux tapissant la rétine sont de deux types :

- les bâtonnets pour la vision scotopique (à faible luminance i.e. en dessous de $10^{-6} cd/m^2$) sont achromatiques et se rencontrent principalement dans les zones perifovéales et périphériques de la rétine.
- Les cônes utilisés en vision photopique (luminance élevé, au dessus de $10 cd/m^2$) sont présents principalement en région fovéale et parafovéale.

Leurs concentration est très forte dans ces zones puisque l'on peut trouver de 30.000 à 40.000 cônes sur un espace circulaire de 1 à 2 mm.

3.2.1 Les bâtonnets

Les bâtonnets sont responsables de la vision scotopiques. Ne disposant que d'un seul type de recepteur à ce niveau de luminance l'oeil ne peut fournir une vision coloré.

La vision nocturne est possible grâce à la décomposition de la Rhodopsine par une réaction photochimique. La décomposition de ce pigment induit l'exitation du bâtonnet. L'intervention de la rhodopsine dans la vision scotopique peut être mise en évidence grâce à la superposition des courbes représentant la dissociation de la rhodopsine et l'exitation des bâtonnets en fonction de la longueur d'onde (voir Figure 3.2).

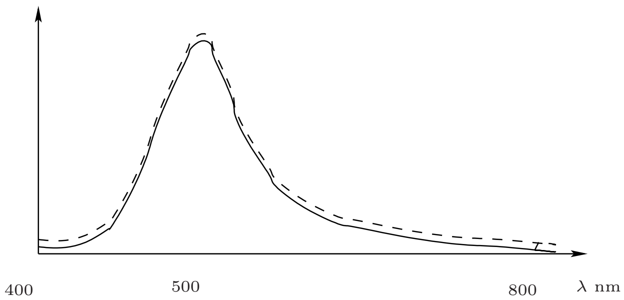


FIG. 3.2 – *Courbes de dissociation de la Rhodopsine et d'exitation des bâtonnets*

La rhodopsine est une hétéroprotéine formé par une protéine l'obsine liée à un groupement prosthétique, la rétinène (aldéhide de la vitamine A). Sousmise à un éclairement, la rhodopsine se décompose en ces deux constituants, le rétinène libéré subissant à son tour une isomérisation (forme cis à forme trans). Lorsque l'obscurité revient la reformation de la rhodopsine emprunte deux voies :

1. La voie rapide, directement à partir du rétinène
2. La voie lente a partir de la vitamine A. Ce type de resynthèse nécessite une oxidation ce qui réclame une certaine dépense métabolique. La lenteur de cette réaction chimique permet d'expliquer le phénomène d'adaptation à l'obscurité.

En effet, à la lumière toute la rhodopsine présente dans les batonnets est décomposées. Or l'exitation de ceux ci nécessite de la rhodopsine de sorte qu'il n'est pas possible de passer en vision scotopique immédiatement après un séjoure à la lumière. La vision scotopique redevient progressivement possible au fur et à mesure que le stock de rhodopsine se reconstitu, ce qui réclame de 20 à 30 minutes. La sensibilité de l'oeil s'accroie donc et n'atteint son maximum qu'au bout d'une trentaine de minutes.

Troubles de la vision scotopique

La majorité des troubles de la vision scotopique provient d'une carence en vitamine A qui comme on l'a vu est essentielle à la resynthèse de la rhodopsine. On distingue les deux cas suivants :

1. La Héméralopie : simple diminution de la teneur en vitamine A, la rétine ne pouvant plus former un taux suffisabt de rhodopsine. LA vision scotopique présente un seuil plus élevé que normalement.
2. La Xérophtalmie : elle est due a une carence sévère en vitamine A. C'est la disparition totale du sens crépusculaire. C'est une lésion irréversible de la cornée.

3.2.2 Les cônes

Les cônes se répartissent en trois grandes familles L (long), M (medium) et S (short) correspondant respectivement à leur sensibilité aux longues, moyennes et grandes longueurs d'ondes. On peut donc considérer que les cônes L sont sensibles au rouge alors que les cônes M sont sensibles au vert et les cônes S au bleu. Ces trois types de récepteurs sont inégalement répartis sur la rétine. Alors que la fovéa comporte essenssiellement des cônes L et M, les cône de types S se trouvent essentiellement en région parafovéale. De plus, la répartition des différents types de cônes sur la rétine, malgré les grandes tendance évoquées plus haut, est essentiellement aléatoire. Cette répartition aléatoire des différents types de cônes permet d'éviter des abérations visuelles si l'on regarde des motifs réguliers.

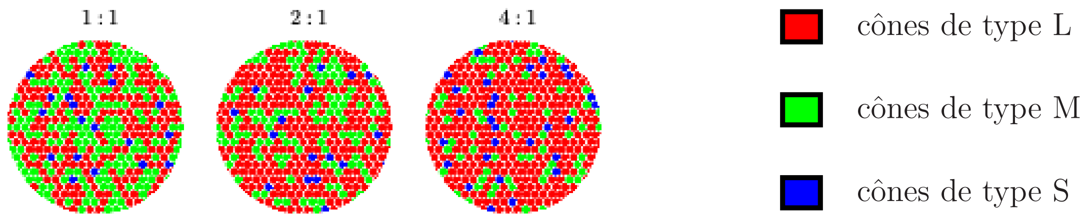


FIG. 3.3 – *Trois simulations de répartitions de cônes dans une région de la rétine de $78mm^2$. La proportion de cônes S est d'environ 5% dans chaque image. Le rapport entre le nombre de cônes L et M est indiqué au dessus de chaque image*

La réponse de chaque cône à la puissance de l'excitation lumineuse est approximativement logarithmique. Les cônes L, M et S de même nature sont reliés par des synapses electriques dites jonctions (gap). Cette mise en relation de cônes voisins de même nature induit un premier effet de lissage.

Immédiatement en arrière de la rétine se trouve un ensemble de neurones se répartissant en deux couches (voir Figure 3.4) :

1. La couche plexiforme externe. Cette couche met en relation les differents recepteurs ce qui permet d'effectuer simultanément un lissage de l'image rétinienne ainsi qu'une accentuation des contrastes. Les signaux issus de cette couche sont appellés des sigaux X.
2. La couche plexiforme interne produit des signaux appelés signaux Y permettant de signaler toute difference notable entre une image et la précédente ou à l'intérieur d'une image. On peut considérer des signaux comme des détecteurs d'évênements spaciaux temporels.

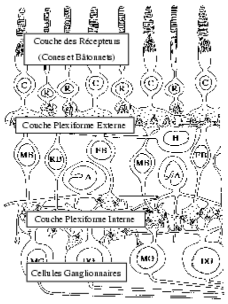


FIG. 3.4 – *La rétine avec ses récepteurs. La couche plexiforme externe comprenant les interconnexions entre les recepteurs et les cellules horizontales et bipolaires, la couche-plexiforme interne qui contient les interconnexions entre les cellules bipolaires, amacrines et ganglionaires.*

La liaison entre la couche plexiforme externe et les cellules ganglionaires est assurée par des cellules appellées cellules bipolaires. Ces cellules assurent notament une transformation du signal (L,M,S) reçu par les cônes en un signal (voir Figure 3.5) :
- Opposition rouge-vert
- Opposition bleu-Jaune
- Luminance

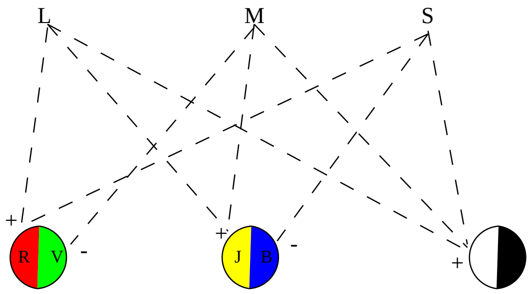


FIG. 3.5 – *Modèle de mécanisme des couleurs antagonistes*

Troubles de la vision photopique

Ces troubles sont dues à un mauvais fonctionement d'un ou plusieurs type de cône. On peut distinguer les cas suivants :

1. Les trimates anormaux.

   Pour ces personnes un type de cône ne fonctionne que partiellement. Lors d'expérience de synthèse additive ces personnes auront donc tendance à rajouter une quantité exessive de la primaire qu'elles ne percoivent que partiellement. On distingue les cas suivants :
   - Les Protanomaux, le cônes L ne fonctionne que partiellement.
   - Les Deutéranomaux, les cônes M ne foctionnent que partiellement.
2. Les dichromates ou daltoniens.

   Pour ces personnes un type de recepteur L, M ou S n'est plus du tout fonctionnel. Ces personnes ne peuvent donc pas différencier deux couleurs qui se différencient grâce au recepteur qui leur fait défaut.

   On distingue à nouveau les cas suivants :
   - Les Protanopes : Défaillance des cônes de type L.
   - Les Deutéranopes : Défaillance des cônes de type S ou M.
3. Les monochromates.

   Ces personnes ne voient plus qu'avec un type de cône. Notez qu'un nombre important de cônes défaillant induit non selement des perturbations dans la vision des couleur (résolution fréquentielle) mais également dans la vision des détails (résolution spatiale). Les monochromates ont donc également une mauvaise vision des détails.

Chapitre 4

Colorimétrie

Le chapitre précédent nous a permis d'entrevoir deux univers très différents : l'univers de la physique basé uniquement sur la notion de spectre et de puissance par longueur d'onde et le monde de la psycho-vision qui utilise les données des capteurs L, M et S situés sur la rétine pour créer l'impression colorée.

Dans le cadre de la vision par ordinateur, il est généralement couteux et peu efficace de baser des algorithmes sur la définition physique de la lumière (les spectres). En effet, utiliser un spectre impose d'échantillonner l'intervalle $[360nm - 830nm]$ des longueurs d'ondes visibles. Sachant que cette intervalle est généralement échantillonné tous les 10 nm, une couleur sera décrite par $\frac{830-360}{10} = 50$ valeurs, alors que l'on sait que l'oeil humain ne travaille qu'avec 3 valeurs correspondant aux réponses des cônes L, M et S. De plus, si le but du traitement est de simuler le comportement visuel humain (en segmentation par exemple), l'utilisation de spectres peut entrainer des erreur d'interprétations. En effet, s'il est exact que deux spectres identiques donneront la même impression colorée, du fait de la perte d'information induite par les cônes L, M et S, de nombreux spectres peuvent être différents tout en donnant la même impression colorée (voir section 4.1). On ne peut donc pas espérer différiencier des couleurs en utilisant uniquement les spectres.

D'un autre côté, les résultats en psycho-vision sont encore insuffisants pour concevoir des algorithmes complexes et efficace s'inspirant du système visuel humain. Certains chercheurs ont réussi à simuler les premiers traitements effectués par le système visuel entre l'oeil et la zone V1. Ces traitements bas niveaux permettent de concevoir des algorithmes intéressants de lissage ou de décomposition du signal mais ne permettent pas,par exemple, d'effectuer des traitements de plus haut niveau comme la segmentation.

La colorimétrie se situe donc entre ces deux extrêmes. Plutôt que d'utiliser des spectres inutilements complexes et inadéquats pour simuler le comportement humain, la colorimétrie utilise des mélanges de couleurs basés sur trois couleurs primaires tout comme l'oeil humain. Mais plutot que d'essayer de comprendre et simuler le comportement du cerveau, la colorimetrie s'attache à observer les effets de certaines stimulations lumineuses sur celui-ci et d'en tirer parti pour la définition de nouveaux espace colorimétriques.

4.1 Définition de la couleur

Nous avons vu dans la section 3.2 que chaque cône, L, M ou S est sensible à une certaine plage de longueurs d'ondes. La sensibilité de chaque cône aux différentes longueurs d'ondes peut être décrite par trois fonctions $l(\lambda)$, $m(\lambda)$ et $s(\lambda)$ décrivant la sensibilité des cônes L, M et S à chaque longueur d'onde. La réponse d'un cône étant proportionnelle à la somme de ses excitations, la réponse (c_1, c_2, c_3) des cônes L, M et S à un spectre $f(\lambda)$ donné peut être modélisée par :

$$\left\{\begin{array}{lcl} c_1 & = & \int_{360}^{830} l(\lambda)f(\lambda)d\lambda \\ \\ c_2 & = & \int_{360}^{830} m(\lambda)f(\lambda)d\lambda \\ \\ c_3 & = & \int_{360}^{830} s(\lambda)f(\lambda)d\lambda. \end{array}\right.$$

Si l'on échantillonne les fonctions f, l,m et s, on peut considérer celles-ci comme des vecteurs de taille N. L'équation précédente peut donc se réécrire sous la forme :

$$\left\{\begin{array}{lcl} c_1 & = & \sum_{i=1}^{N} l(\lambda_i)f(\lambda_i) \\ \\ c_2 & = & \sum_{i=1}^{N} m(\lambda_i)f(\lambda_i) \\ \\ c_3 & = & \sum_{i=1}^{N} s(\lambda_i)f(\lambda_i). \end{array}\right.$$

où $(\lambda_1, \ldots, \lambda_n)$ représente nos N échantillons. Si l'on pose :

$$S^t = \begin{pmatrix} l(\lambda_1) & \ldots & l(\lambda_N) \\ m(\lambda_1) & \ldots & m(\lambda_N) \\ s(\lambda_1) & \ldots & s(\lambda_N) \end{pmatrix}$$

où S^t représente la transposée de S, notre calcul intégral peut s'écrire sous la forme d'un produit matriciel :

$$c = S^t f. \tag{4.1}$$

Notez que les équations précédentes ne sont valides que si l'on suppose que tous les cônes de même type sont décrits par les mêmes fonctions de réponse, non seulement pour tous les cônes d'un même individu mais pour deux individus différents. Le lissage réalisé entre les cônes de même type de la rétine permettent d'obtenir une réponse relativement similaire pour tous les cônes. De plus, des expériences ont montré que ces réponses variaient peu entre individus.

L'équation 4.1 peut se voir comme une projection du spectre f sur le sous espace engendré par les vecteurs l, m et s. Ce sous espace appelé *Le sous espace Visuel Humain* décrit la partie des spectres que nous sommes capables de percevoir. La quantité $S^t f$ décrit les coordonnées de la projection du spectre

f dans la base l, m, s et peut s'interpréter comme une mesure de la sensation colorée lors de la perception du spectre f.

Le vecteur 3D $c = S^t f$ représentant la sensation colorée liée à la perception du spectre f, nous pourrions considérer que c représente la **couleur** associée à f. Toutefois, la sensibilité des cônes L, M et S déterminée par les fonctions l, m et s est difficile à mesurer puisqu'elle nécessite de brancher des électrodes sur différents cônes d'un observateur humain. De plus, tout changement de base dans le sous espace Visuel Humain nous donnera une mesure équivalente de la couleur. Nous allons donc essayer de déterminer des bases permettant de calculer plus facilement les triplets représentant une couleur. Ces expériences sont appelées des mises en correspondance de couleurs.

Supposons que nous disposions de trois sources lumineuses p_1, p_2, p_2 *colorimétriquement indépendantes* telles que les trois vecteurs $S^t p_1$, $S^t p_2$ et $S^t p_3$ forment une base libre de $\mathbb{R}^3$. Toute couleur $c = S^t f$ peut donc s'écrire comme une combinaison linéaire de ces trois vecteurs :

$$c = S^t f = \alpha_1 (S^t p_1) + \alpha_2 (S^t p_2) + \alpha_3 (S^t p_3)$$

où α_1, α_2 et α_3 appartiennent à $\mathbb{R}$ et représentent les coorconnées de c dans la nouvelle base. Cette dernière expression peut également d'écrire sous la forme :

$$c = S^t f = S^t (\alpha_1 p_1 + \alpha_2 p_2 + \alpha_3 p_3)$$

Si l'on note par $P = (p_1, p_2, p_3)$ la $N \times 3$ matrice représentant les spectres de nos trois sources, l'on a :

$$c = S^t f = S^t P \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{pmatrix} \tag{4.2}$$

La matrice $S^t P$ est une matrice 3×3 dont les vecteurs colonnes sont égaux à $(S^t p_1, S^t p_2, S^t p_3)$. Ces trois vecteurs formant par hypothèse une base de $\mathbb{R}^3$, la matrice $S^t P$ est inversible. Ainsi, l'impression colorée obtenue à partir d'une source lumineuse f peut être reproduite par combinaison de trois sources lumineuses p_1, p_2, p_3 d'intensités relatives $\alpha(f) = (\alpha_1, \alpha_2, \alpha_3)$. La détermination des coefficients $\alpha(f)$ s'effectue à l'aide d'expériences d'appariement de couleurs : un sujet se voit présenté deux taches colorées (voir Figure 4.1) l'une obtenue à partir d'une source f et l'autre grâce à une combinaison de trois sources p_1, p_2, p_3. Le sujet doit modifier les coefficients α de façon à ce que les deux taches colorées soient de couleur identique. Le triplet $(\alpha_1, \alpha_2, \alpha_3)$ vérifie alors l'équation 4.2.

Le vecteur $\alpha(f)$ défini par l'équation 4.2 peut comporter une ou plusieurs composantes négatives. Ces coefficients ne peuvent donc pas être déterminés par l'expérience précédente où les sujets ne peuvent envoyer que des quantités positives de lumières sur l'écran. Dans ce cas, le sujet est autorisé à déplacer un ou plusieurs projecteurs p_i de l'autre côté (voir Figure 4.2). Si l'on suppose que α_1 est le coefficient négatif, ceci revient à faire l'appariement :

$$f - \alpha_1 p_1 = \alpha_2 p_2 + \alpha_3 p_3.$$

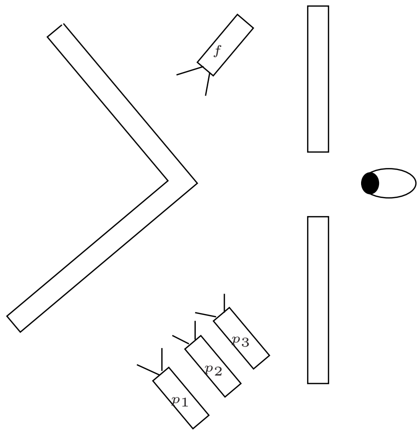


Fig. 4.1 – *L'expérience d'appariement de couleurs*

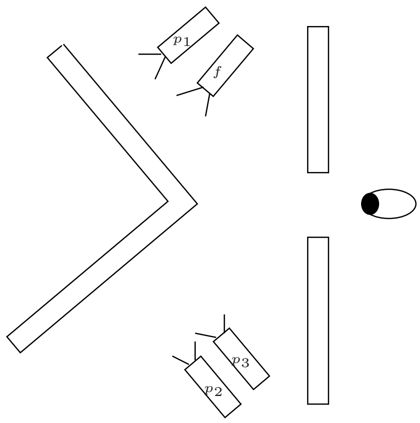


Fig. 4.2 – *Appariement avec des couleurs négatives*

On réalise alors un appariement avec uniquement des coefficients positifs.

L'équation 4.2 montre que si l'on connait les triplets correspondant à un ensemble de spectres, l'on peut connaître les triplets de n'importe qu'elle combinaison de ces spectres. En effet, si $(f_i)_{i\in\{1,\ldots,n\}}$ représente un ensemble de spectres, chaque f_i étant décrit par un triplet $\alpha(f_i)$, alors n'importe qu'elle combinaison de ces spectres $g = \sum_{i=1}^{p} \beta_i f_i$ donnera une impression colorée définie par :

$$\begin{array}{rclr} S^t g & = & S^t & \sum_{i=1}^{p} \beta_i f_i \\ & = & & \sum_{i=1}^{p} \beta_i S^t f_i \\ & = & & \sum_{i=1}^{p} \beta_i S^t P \alpha(f_i) \\ & = & S^t P & \sum_{i=1}^{p} \beta_i \alpha(f_i). \end{array} \tag{4.3}$$

On a donc $\alpha(g) = \sum_{i=1}^{p} \beta_i \alpha(f_i)$. Toute combinaison linéaire de spectres est décrite par la même combinaison de triplets associés.

Plutôt que déterminer expérimentalement les coefficients α de toutes les couleurs visibles, ce qui est matériellement impossible, on peut en utilisant la linéarité décrite dans le paragraphe précédent mesurer ces coefficients uniquement pour une base de $\mathbb{R}^N$. Etant donnés les coefficients de chaque élément de cette base, l'on peut reconstruire n'importe quel spectre et donc obtenir son triplet associé par simple combinaison linéaire. Soit $(e_i)_{i\in\{1,\ldots,N\}}$ la base canonique de $\mathbb{R}^N$. Nous pouvons mesurer pour chaque e_i le coefficient $a_i = \alpha(e_i)$ associé. On a donc pour tout i appartenant à $\{1,\ldots,N\}$:

$$S^t e_i = S^t P a_i. \tag{4.4}$$

Ces N équations peuvent s'écrire sous la forme matricielle suivante :

$$S^t I = S^t P A^t \tag{4.5}$$

où I est la matrice identité de $\mathbb{R}^N$ et $A = (a_1 \ldots a_N)$ est la matrice $3 \times N$ définie par les vecteurs de coefficients a_i de chaque vecteur unitaire e_i. Chaque coefficient a_{ik} de la matrice A représente la proportion de primaires p_i nécessaire pour réaliser l'appariement du spectre e_k. La matrice A est donc appelée la *matrice d'appariement* (ou color matching matrice). De même, les colonnes a_i de A sont appelées les *fonctions d'appariement* (color matching functions).

De l'équation 4.5 nous pouvons déduire l'expression de A :

$$A = S(P^t S)^{-1}.$$

Le théorème suivant montre que la matrice A peut être utilisée pour décrire les couleurs plutôt que S :

Théorème 1 *Deux spectres ayant la même image par A^t sont visuellement identiques. Autrement dit :*

$$\forall (f,g) \in \mathbb{R}^N \quad S^t f = S^t g \Longleftrightarrow A^t f = A^t g.$$

Preuve:

Nous avons $A^t = ((P^tS)^{-1})^tS^t$. Il est donc clair que si $S^tf = S^tg$:

$$A^tf = ((P^tS)^{-1})^tS^tf = ((P^tS)^{-1})^tS^tg = A^tg.$$

Inversement, si $A^tf = A^tg$ nous avons :

$$((P^tS)^{-1})^tS^tf = ((P^tS)^{-1})^tS^tg.$$

La matrice S^tP étant inversible, $((P^tS)^{-1})^t$ l'est également et l'équation précédente est équivalente à $S^tf = S^tg$. $\square$

Nous pouvons donc indifféremment exprimer des couleurs à l'aide de la matrice S ou de la matrice A. La différence essentielle entre ces deux modes de représentation est que la matrice S ne peut s'obtenir qu'avec des mesures extrêmement délicates de la sensibilité des cônes alors que la matrice A peut s'obtenir à l'aide d'expériences d'appariements (voir Figures 4.1 et 4.2) nettement plus simples à réaliser.

4.1.1 Le métamérisme

Nous avons vu dans les paragraphes précédents que calculer la couleur associée à un spectre pouvait s'interpréter comme le calcul de la projection de ce spectre dans un sous espace de dimension 3 appelé le sous espace visuel humain. Le passage à un espace de dimension N à un espace de dimension 3 implique une perte d'informations dès que N est plus grand que 3 (ce qui est largement le cas). Il existe donc une quantité de spectres différents possédant la même projection et donc associés à la même sensation colorée. Ces spectres sont dits *métamères*.

On peut donc se représenter l'ensemble des spectres comme deux espaces orthogonaux :

- l'espace visuel humain de dimension 3 et
- un espace de dimension $N-3$. Les spectres situés dans cet espaces ne sont pas associés à une sensation colorée. Plus exactement, un spectre défini dans cet espace donnera une sensation de noir. En conséquence, cet espace est souvent appelé *l'espace noir*.

Si A représente notre matrice d'appariement, la projection d'un spectre quelconque sur le sous espace visuel humain est donné par $P_A = A(A^tA)^{-1}A^t$ et la projection sur l'espace noir par $I - P_A$ où I représente la matrice identité de $\mathbb{R}^N$.

Tout spectre f peut donc se décomposer en une partie visible et une partie invisible :

$$f = P_Af + (I - P_A)f.$$

Le lecteur curieux peut vérifier que :

$$\forall f \in \mathbb{R}^N \left\{ \begin{array}{lcl} A^tP_Af & = & A^tf \\ A^t(I - P_A)f & = & 0. \end{array} \right.$$

Les spectres possédant la même projection sur l'espace visuel humain seront donc métamères. Etant donné un spectre f, l'ensemble de ses métamères est égal à :

$$meta(f) = \{P_A f + (I - P_A)g, \quad g \in \mathbb{R}^N\}.$$

4.1.2 Couleur des matériaux

Nous avons vu dans la section 2.2 que le spectre renvoyé par un objet pouvait être décrit par sa fonction de réflectance $r(\lambda)$ qui dépend de l'angle d'incidence de la lumière, de l'angle de réflexion et des propriétés optiques du matériel. Ces paramètres étant fixés, la couleur de l'objet est définie par :

$$c = A^t R f$$

où R est une matrice $N \times N$ diagonale telle que la valeur du i^{eme} élément sur la diagonale est égal à $r(\lambda_i)$. Si l'on develope cette dernière équation, l'on obtient :

$$\forall i \in \{1, 2, 3\} \quad c_i = \sum_{j=1}^{N} a_{ij} r_j f_j.$$

Une approximation, souvent effectuée en lancé de rayon, consiste à calculer la couleur réfléchie par un blanc d'égale énergie (dont l'énergie est constante sur chaque longueur d'onde) et à assimiler le triplet obtenu à la reflectance sur chaque longueur d'onde. Cette méthode calcule donc tout d'abord la couleur :

$$c_R = A^t R f1$$

où $f1$ est un vecteur dont toutes les composantes sont égales à 1. Les composantes du vecteur c_R et celles de c sont ensuite multipliées 2 à 2. Selon ce calcul, la lumière réféchie est donc égale à $((c_R)_1 c_1, (c_R)_2 c_2, (c_R)_3 c_3)$ où $(c_1, c_2, c_3) = (A^t f)^t$. Trancrit en terme de sommes, le calcul précédent revient à assimiler :

$$(\sum_{j=1}^{N} a_j r_j)(\sum_{i=1}^{N} a_i f_i) \text{ et } \sum_{i=1}^{N} a_i r_i f_i.$$

On confonds donc une somme de produits avec un produit de sommes. Cette aproximation permet un gain de temps appréciable, puisque l'on travaille uniquement avec des triplets de couleurs et non avec des spectres, mais conduit à des résultats inacceptables dès que l'on veut obtenir des images réalistes.

4.1.3 Constance chromatique

La détermination du triplet représentant une couleur s'est effectuée jusqu'à présent avec des conditions d'éclairement constant. Cette simplification du modèle ne permet pas de tenir compte du phénomène suivant : une feuille verte vue sous une lumière blanche (à midi) conserve sa couleur verte le soir (sous une lumière rouge).

Ce phénomène est à priori surprenant si nous nous reférons au modèle colorimétrique classique. En effet, les spectres du soleil à midi et le soir notés f_m et f_s étant très différents, l'on devrait constater des impressions visuelles $S^t f_m$ et $S^t f_s$ très différentes. Puisqu'il n'en n'est rien, nous sommes amenés à supposer que le système visuel humain (tout comme une caméra) s'adapte dynamiquement aux conditions d'éclairage.

Von Kries [Kri05, Mac70] a émis l'hypothèse que cette adaptation de l'oeil pouvait être modélisée par une fonction de gain représentée par une matrice 3×3 diagonale. On a donc :

$$c' = DS^t f$$

où D représente la matrice 3×3 diagonale uniquement fonction des conditions d'éclairage. Supposons à présent que les fonctions de gain de notre oeil à midi et le soir soient décrites par les deux matrices D_m et D_s. Les spectres réfléchis par la feuille à midi et le soir f_m et f_s nous donneront la même impression colorée si :

$$D_m S^t f_m = D_s S^t f_s.$$

Si nous raisonnons en terme d'appariement de couleur plutôt qu'en terme de réponse des cônes, nous sommes amenés à effectuer des apparariements asymétriques, ou autrement dit des appariements avec des conditions d'eclairage différents. Si nous reprenons l'équation 4.4, cet appariement est décrit par les équations :

$$\forall i \in \{1, \dots, N\} \quad D_1 S^t e_i = D_2 S^t P a_i^{12}$$

où D_1 et D_2 représentent les deux matrices de gains correspondant aux deux conditions d'éclairement différents. Le membre droit de ces équations décrit la proportion de primaires permettant d'apparier le spectre e_i sous le second éclairage alors que le membre gauche nous donne l'impression visuelle de ce même spectre sous le premier type d'éclairage. Transcrites en terme matriciel, ces équations donnent :

$$D_1 S^t I = D_2 S^t P (A^{12})^t = D_1 S^t P A^t$$

où A représente la matrice d'appariement sous les mêmes conditions d'illumination (voir equation 4.5) et A^{12} la matrice d'appariement tenant compte des différents illuminants. La relation entre A et A^{12} est donc :

$$A^t = (S^t P)^{-1} D_1^{-1} D_2 (S^t P) A^{12}.$$

On a donc :

$$\forall f \in \mathbb{R}^N \quad (S^t P) A^t f = D_1^{-1} D_2 (S^t P) A^{12} f.$$

Si nous reprenons l'exemple des lumières de midi et du soir, et si l'éclairage 1 correspond au soleil de midi et 2 à celui du soir, le triplet $A^{12} f$ définit la proportion de primaires nécessaire à midi pour avoir la même impression visuelle que f vue le soir. A un changement de base près (défini par la matrice $S^t P$), ce triplet se déduit du triplet défini à luminosité constante $A^t f$ par une simple homothétie $D_2^{-1} D_1$.

Cette modélisation de l'adaptation du système visuel humain à l'éclairement par une matrice 3×3 diagonale reste une approximation qui ne rend qu'imparfaitement compte de tous les phénomènes de constance chromatique. Toutefois, cette méthode reste très populaire en raison de sa simplicité.

4.2 Les espaces de couleur

Comme nous l'avons vu dans la section précédente, toute couleur peut être représentée comme une combinaison linéaire de trois primaires p_1, p_2, p_3. La donnée d'un triplet représentant une couleur n'a de sens que si l'on connait les 3 primaires qui ont servis à les définir. La *Commission Internationale de l'Eclairage* a donc défini un ensemble de standards permettant une communication entre les différents utilisateurs de la colorimétrie.

4.2.1 Les espaces de couleurs de la Commission Internationale de l'Eclairage(CIE)

L'espace CIE RGB de 1931

Cet espace est défini à partir de trois primaires monochromatiques de couleurs rouge, vertes et bleu. Les longueurs d'ondes associées à chacune des primaires sont les suivantes :
- 700,0 nanomètres pour le rouge,
- 546,1 pour le vert,
- 435,8 pour le bleu

De plus la puissance de chacune de ses primaire est ajustée de façon à obtenir trois triplets identiques pour tout spectre d'égale énergie. Les expériences qui ont permies d'obtenir les fonctions d'appariement $r(\lambda)$,$g(\lambda)$ et $b(\lambda)$ (Voir Figure 4.3) ont été réalisé avec un écart α (voir Figure 4.4) d'environs 4 degré (donc pour des couleurs relativement proches). La CIE à définie en 1964 un autre espace de couleur RGB dont les fonctions d'appariement ont étées obtenues avec un écart de 10 degré (donc pour des taches colorées plus éloignées).

L'espace CIE XYZ

Nous avons vu dans la section 4.1.1 que l'ensemble des spectres pouvait être décomposer en deux espaces vectoriels orthogonaux : L'espace visuel humain $P_A\mathbb{R}^N$ correspondant à la partie visible de l'ensemble des spectres, l'espace noir $(I - P_A)\mathbb{R}^N$ correspondant à la partie invisible.

Si nous utilisons trois spectres visibles comme base de l'espace visuel humain, il existera toujours des spectres ayant des coordonnées négatives sur cette base. Si nous utilisons par exemple l'espace RGB ou les trois primaires sont vivibles la Figure 4.3 nous monre que les couleurs bleu-vert(450-550 nm) ne peuvent être reproduites par superposition des trois spectres. Ceci peut poser des problèmes dès que l'on désire travailler en synthèse additive (en n'ajoutant que des quantitées positives de couleur). La CIE à donc conçu un espace de couleur basé sur

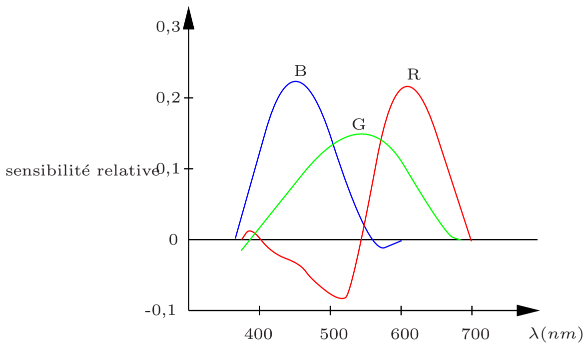


FIG. 4.3 – *Les courbes d'appariement $r(\lambda)$, $g(\lambda)$ et $b(\lambda)$*

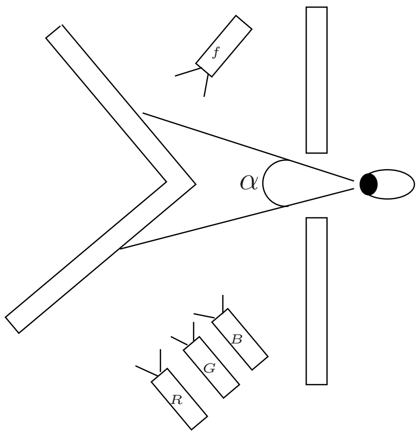


FIG. 4.4 – *L'angle α entre les deux taches colorées est une donnée importante des expériences d'appariement*

trois primaires X, Y et Z non visibles. Cet espace possède plusieurs propriétés intéressantes :

- Les triplets décrivant chaque couleur en fonction de ses primaires ont tous des valeurs positives pour les spectres visibles.
- La fonctions $y(\lambda)$ représente approximativement la sensibilité de l'oeil humain à la luminosité. La composante Y du triplet (X,Y,Z) est usuellement assimilée à la luminance du spectre incident.
- Tout spectre d"égale énergie est associé à un triplet dont toutes les composantes sont égales.

La transformation de l'espace RGB à l'espace XYZ peut s'interpreter comme un changement de base donné par la matrice :

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} 0.166 & 0.125 & 0.093 \\ 0.060 & 0.327 & 0.005 \\ 0.000 & 0.004 & 0.46 \end{pmatrix} * \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

Un ensemble de vecteurs 3D étant difficile à manipuler, et à visualiser. Un façon élégante de résoudre ce problème consiste a projeter les veteurs 3D sur *le plan unitaire* (le plan sur lequel la somme des composantes est égale à 1). La somme des trois composantes ne représantant que l'intensité de la couleur et non la proportion de chacune des primaires, l'on obtient ainsi un diagrame appellé *diagrame de chromaticité* les projetions des vecteurs 3D représentant les couleurs étant appellées des *coordonnées chromatiques*.

Le diagramme de chromaticité sans doute le plus utilisé est le *diagramme chromatique xy* de la CIE. Les coordonnées chromatiques x et y sont obtenues à partir des coordonnées X, Y et Z par la transformation suivante :

$$\begin{aligned} x &= \frac{X}{X+Y+Z} \\ y &= \frac{Y}{X+Y+Z} \\ z &= \frac{Z}{X+Y+Z} \end{aligned}$$

Notez que l'on à $x + y + z = 1$, seules deux coordonnées sont donc nécessaires pour représenter l"ensemble des couleurs.

La courbe représentée sur la Figure 4.5 représente l'ensemble des spectres monochromatiques et est appellée le *spectrum locus*. Soit (X_i, Y_i, Z_i) les coordonnées XYZ du spectre monochromatique e_i (pour i appartenant à $\{1, \dots, N\}$). Tout spectre f étant une somme pondérée de e_i, on à :

$$f = \sum_{i=1}^{N} \beta_i e_i$$

si $c = (X, Y, Z)$ dénote la couleur associée à f on obtient (voir equation 4.3) :

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \sum_{i=1}^{N} \beta_i \begin{pmatrix} X_i \\ Y_i \\ Z_i \end{pmatrix}$$

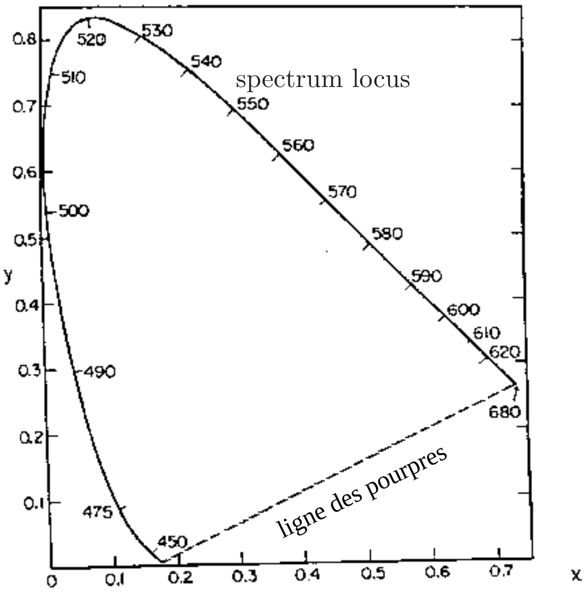


FIG. 4.5 – *Le diagramme de chromaticité xy*

Donc :

$$x = \frac{X}{X+Y+Z} = \frac{\sum_{i=1}^{N} \beta_i X_i}{\sum_{i=1}^{N} \beta_i (X_i + Y_i + Z_i)} = \frac{\sum_{i=1}^{N} \beta_i (X_i + Y_i + Z_i) x_i}{\sum_{i=1}^{N} \beta_i (X_i + Y_i + Z_i)}$$

La coordonnée x peut donc se voir comme une moyenne pondérée des différents x_i. Le même raisonnement pouvant être trenu pour y, le point (x, y) est une moyenne des differents points (x_i, y_i) du spectrum locus et se trouve donc à l'intérieur de son envelope convexe.La droite fermant cette envelope convexe joint les points correspondant aux longueurs d'ondes minimales et maximales et est appellée *la ligne des pourpres* (De fait les couleurs situées sur cette droite sont pourpres).

Notons enfin que l'espace XYZ (et plus précisément son plan unitaire) est souvent utilisé pour désigner un espace couleur (voir exercices sections 4.3.2, 4.3.3 ainsi que 4.4.2, 4.4.3). Les principaux avantages de ce type de définition est de n'utiliser qu'un nombre réduit de valeurs et de permettre une définition explicite du blanc de coordonnées $(1, 1, 1)$ dans le nouvel espace.

4.2.2 Les espaces uniformes

La description des couleurs en terme de vecteur 3D pose le problème de la pertinence du calcul des distances entre ces vecteurs. En effet, on a vue au tout début de ce chapitre que la donné d'un vecteur 3D modélise correctement la réponse des cônes à un spectre donné. En revanche, on a aucune assurance que la distance euclidienne entre ces deux vecteurs modélise fidèlement notre notion de distance entre ces couleurs. En d'autres termes l'on voudrait définir un espace tel que si trois vecteurs c, c_1 et c_2 vérifient :

$$d(c, c_1) = nd(c, c_2)$$

où $d(.,.)$ représente la distance euclidienne, la couleur c_1 apparaisse n fois plus éloignées de la couleur c que la couleur c_2.

Il est relativement clair que cete notion de distance n'a de sens que pour des couleurs relativement proches. Il est en effet assez rare, de se demander laquelle des couleurs rouge ou verte est la plus proche de la couleur bleu. Ces trois couleurs étant trop différentes pour être comparées. En revanche, cette notion de distance psycho-visuelle est pertinente et utile pour des couleurs proches. Des algorithmes de détection de contour, par exemple doivent avoir une notion de distance entre les couleurs de façon à pouvoir déterminer les contours.

Les expériences réalisées par MacAdam destinées à mesurer l'adéquation entre les espaces de couleurs et la notion de distance ont consistées à mesurer expérimentalement l'ensemble des couleurs *justes disernables* d'un ensemble de couleur donnée. Ces expériences réalisées dans l'espace XYZ ont montrées que l'ensemble des couleurs juste disernable d'une couleur donnée pouvait être approximé par une ellipse de taille et d'orientations variables (voir Figure 4.6).

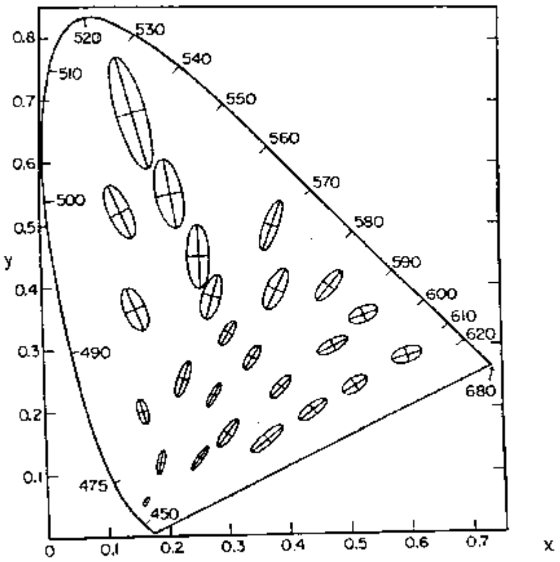


FIG. 4.6 – *Les ellipses de MacAdam représentées dans le diagramme chromatique xy*

La Figure 4.6 nous permet de tirer plusieurs conclusions. Tout d'abord, l'ensemble des couleurs à égale distance d'une couleur donnée étant une ellipse, l'espace XYZ ne peut être uniforme (voir [Wri41]).

De fait, Alain Trémeau [Tre93] a montré que la vision des couleurs obéissait plus à une géométrie Riemannienne qu'Euclidienne. Or plonger un espace de Rieman dans un espace Euclidien nécessite de passer de la dimension n à la dimension m avec :

$$m = \frac{n(n+1)}{2}$$

. L'espace de couleurs étant de dimension 3, on ne peut donc espérer obtenir une distance euclidienne uniforme qu'en passant en dimension 6.

Une uniformisation absolue des espaces de couleurs étant donc impossible, on peut toutefois chercher à déterminer des espaces pseudo uniformes. Cette recherche passe généralement par deux étapes distinctes :

- La recherche d'une représentation de l'intensité à peu près uniforme
- La recherche d'une représentation uniforme des distances entre couleur à intensité constante.

Une fois ces deux étapes achevées ont construit généralement l'espace en établissant des facteurs de pondérations entre les distances de luminosité et les distances colorimétriques.

Ces recherches ont abouti en 1976 à l'établissement de deux standards : les espaces CIE $L^*u^*v^*$ et CIE $L^*a^*b^*$.

Les espaces CIE $L^*u^*v^*$et CIE $L^*a^*b^*$

Des expériences psycho-visuelles ont montré que l'appréciation des distances entre des stimuli mono-chromatiques pouvait être approximées par une racine cubique. La luminosité dans les deux espaces de couleurs $L^*u^*v^*$ et $L^*a^*b^*$ et donc définie par :

$$L^* = 116 f(\frac{Y}{Y_w}) - 16 \text{ avec } f(x) = \begin{cases} x^{\frac{1}{3}} & \text{si } x \geq 0.008856 \\ 7.787x + \frac{16}{116} & \text{si } x \leq 0.008856 \end{cases} \tag{4.6}$$

Le caractère trop abrupt des variations de $x^{\frac{1}{3}}$ autour de zéro est supprimé par l'interpolation linéaire. Le point 0.008856 a été choisi de façon à assurer une continuité C^2 entre la courbe et la droite (voir Figure 4.7). Le triplet (X_w, Y_w, Z_w) représente un blanc de référence W. L'introduction du rapport $\frac{Y}{Y_w}$ permet de simuler très grossièrement l'adaptation de l'oeil à une luminosité donnée.

La courbe représentée sur la Figure 4.7 présente plusieurs propriétés intéressantes : tout d'abord, l'on peut observer une pente importante aux faibles luminances. Cette propriété permet de tenir compte de l'intervention des bâtonnets (voir section 3.2) pour cet ordre de luminance. On observe également un effet de saturation pour les fortes luminances. De fait, de trop grandes luminances saturent les recepteurs de l'oeil et atténuent les différences de luminosité.

Les composantes chromatiques de l'espaces $L^*u^*v^*$ sont définies par :

$$\begin{aligned} u^* &= 13L^*(u' - u'_w) \\ v^* &= 13L^*(v' - v'_w) \end{aligned}$$

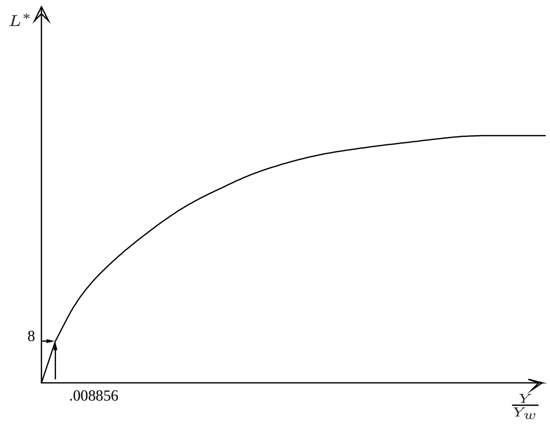


FIG. 4.7 – *La luminosité L^* en fonction de $\frac{Y}{Y_w}$*

avec :

$$\begin{aligned} u' &= \frac{4X}{X+15Y+3Z} \\ v' &= \frac{9Y}{X+15Y+3Z} \end{aligned}$$

$$\begin{aligned} u'_w &= \frac{4X_w}{X_w+15Y_w+3Z_w} \\ v'_w &= \frac{9Y_w}{X_w+15Y_w+3Z_w} \end{aligned}$$

Les coordonnées u^* et v^* peuvent donc s'interpréter comme des distances au blanc dans un espace déduit de XYZ par une transformation non linéaire.

La coordonnée L^* du modèle $L^*a^*b^*$ est définie de la même façon que dans le modèle $L^*u^*v^*$. Les coordonées chromatiques sont définies par :

$$\begin{aligned} a^* &= 500\left[f\left(\frac{X}{X_w}\right) - f\left(\frac{Y}{Y_w}\right)\right] \\ b^* &= 200\left[f\left(\frac{Y}{Y_w}\right) - f\left(\frac{Z}{Z_w}\right)\right] \end{aligned} \tag{4.7}$$

où la fonction f est définie par l'équation 4.6.

Les coordonnées a^* et b^* représentent respectivement une opposition entre les axes X et Y et Y et Z. Les sensibilités à chacun de ses axes sont à nouveau modélisées par la fonction f.

La distance entre deux couleurs (C_1) et (C_2) de coordonnées (L_1^*, u_1^*, v_1^*) et (L_2^*, u_2^*, v_2^*) dans le système $L^*u^*v^*$ et de coordonnées (L_1^*, a_1^*, b_1^*) et (L_2^*, a_2^*, b_2^*) dans le système $L^*a^*b^*$est alors définie par la distance euclidienne :

$$d(C_1, C_2) = \sqrt{(L_1^* - L_2^*)^2 + (u_1^* - u_2^*)^2 + (v_1^* - v_2^*)^2}$$

dans le système $L^*u^*v^*$ et

$$d(C_1, C_2) = \sqrt{(L_1^* - L_2^*)^2 + (a_1^* - a_2^*)^2 + (b_1^* - b_2^*)^2}$$

dans le système $L^*a^*b^*$.

Suivant une étude menée sur le sujet par Pointer [Poi81], il semble qu'aucun des deux espaces CIE $L^*a^*b^*$et $L^*u^*v^*$ ne soit significativement plus uniforme que l'autre. Il semble toutefois que l'espace $L^*u^*v^*$ tombe peu à peu en désuétude au bénéfice de l'espace $L^*a^*b^*$.

La conversion entre les espaces RGB et $L^*u^*v^*$ ou $L^*a^*b^*$ impose de passer par l'espace XYZ, de calculer une racine cubique et d'effectuer plusieurs divisions. Cette transformation implique donc souvent un surcoût de calcul non négligeable qu'il est souvent important de considérer lorsque l'on envisage le choix d'un espace de couleurs.

Inversion de l'espace $L^*a^*b^*$

L'espace $L^*a^*b^*$ étant souvent utilisé on a souvent besoin de convertir un triplet (L, a, b) en un triplet affichable tel que (R,G,B). Cette transformation nécessite de passer par l'espace XYZ puis d'utiliser l'inverse de la matrice de conversion de RGB vers XYZ.

L'inverse de la fonction f (équation 4.6) se calcule aisément est est égal à :

$$f^{-1}(x) = \begin{cases} x^3 & \text{si } x \geq 0.206893 \\ \frac{1}{7.787}\left(x - \frac{16}{116}\right) & \text{si } x \leq 0.206893 \end{cases}$$

Si nous simplifions les notations en notant $f\left(\frac{Y}{Y_w}\right)$, $f\left(\frac{X}{X_w}\right)$ et $f\left(\frac{Z}{Z_w}\right)$ respectivement f_Y, f_X et f_Z, il vient immédiatement à partir de l'équation 4.6 :

$$f_Y = \frac{L^* + 16}{116}$$

De même, l'équation 4.7 constitue un système de deux équations à deux inconnues dont les solutions sont :

$$\begin{array}{rcl} f_X & = & \frac{a^*}{500} + f_Y \\ f_Z & = & f_Y - \frac{b^*}{200} \end{array}$$

On en déduit donc :

$$\left\{ \begin{array}{rcl} X & = & X_w f^{-1}(f_X) \\ Y & = & Y_w f^{-1}(f_Y) \\ Z & = & Z_w f^{-1}(f_Z) \end{array} \right.$$

4.2.3 Les espaces de couleurs non standards

Le modèle CMY

Les modèles CMYet RGB sont tous deux établis à partir de trois couleurs fondamentales. Le modèle CMY est également représenté sous la forme d'un cube, mais l'origine est dans ce cas le blanc et les trois axes principaux le cyan, le magenta et le jaune. A la différence du modèle RGB, les contributions de chaque couleur primaire sont plus souvent exprimées en pourcentage.

Le nom de ce modèle rappelle les trois primaires soustractives : Cyan, Magenta, Yellow (ou CMJ, en français). Ce modèle est surtout répandu dans le monde de l'imprimerie.

Tout comme l'espace RGB, ce modèle ne permet pas d'afficher toutes les couleurs par synthèse additive. Ce modèle est parfois étendu au modèle CMYK (k pour Key black) pour répondre à l'incapacité du mélange des pigments CMY de produire un noir parfait. A notre connaissance, il n'existe pas de représentation visuelle de cette extension. En fait, la valeur de la composante noire est déduite de la valeur des trois autres composantes selon des tables de conversion incorporées dans les logiciels (procédé UCR : Under Color Removal).

Le modèle YIQ

Le modèle YIQ est une variante du modèle RGB établie par le NTSC[1] pour rendre plus efficace la transmission des signaux de télévision et la compatibilité avec les écrans noir et blanc. La composante Y contient l'information concernant la luminosité de l'image. L'information chromatique est quant à elle codée par les axes I et Q correspondant respectivement aux oppositions cyan-orange et magenta-bleu. La relation entre les modèles YIQ et RGB est la suivante :

$$\begin{pmatrix} Y \\ I \\ Q \end{pmatrix} = \begin{pmatrix} 0.30 & 0.59 & 0.11 \\ 0.60 & -0.27 & -0.32 \\ 0.11 & -0.52 & 0.31 \end{pmatrix} * \begin{pmatrix} R \\ G \\ B \end{pmatrix}.$$

L'espace $I_1I_2I_3$

L'espace $I_1I_2I_3$ introduit par Otha et al. [OKS80] répond à une approche totalement différente. Otha, Kanade et Sakai ont cherché l'espace de couleurs présentant le plus d'intérêt pour la segmentation et le traitement d'images. Ils ont constaté que l'on obtenait de bons résultats en utilisant l'espace de couleurs défini par les trois axes de plus grande variance de l'ensemble des couleurs associé à l'image. Un résultat bien connu en analyse de données établit que ces axes correspondent aux *vecteurs propres* de la *matrice de covariance* de l'ensemble de couleurs associé à l'image (voir Figure 4.8).

Ce calcul des vecteurs propres de la matrice de covariance est souvent appelé la transformation de Karhunen-Loève dans la littérature anglo-saxone. Elle consiste à calculer tout d'abord la matrice de covariance définie par :

$$\begin{pmatrix} var(x_1) & cov(x_1,x_1) & cov(x_1,x_3) \\ cov(x_1,x_2) & var(x_2) & cov(x_2,x_3) \\ cov(x_1,x_3) & cov(x_2,x_3) & var(x_3) \end{pmatrix}$$

où $var(x_i)_{i\in\{1,2,3\}}$ représente la variance du ı^{eme} axe de l'espace de couleurs et $cov(x_i,x_j)_{i<j\in\{1,2,3\}}$ représente la covariance entre les axes i et j du même espace.

[1]National Television Standards Committee

FIG. 4.8 – *Multi-ensemble associé à l'image Lenna. Les vecteurs propres sont notés V_1, V_2 et V_3 dans l'ordre décroissant de leurs valeurs propres. La longueur des vecteurs est proportionnelle à leurs valeur propre.*

Cette matrice peut être calculée efficacement en utilisant les moments d'ordre 0, 1 et 2 de l'image :

$$\begin{array}{rcl} M_0 & = & \sum_{P\in I} 1 = |I| \\ M_1 & = & \sum_{P\in I} I(P) \\ M_2 & = & \sum_{P\in I} I(P)I(P)^t \end{array}$$

où I représente l'image, $|I|$ le nombre de pixels de l'image et $I(P)$ la couleur du pixel P.

Notez que M_0 représente une valeur réelle alors que M_1 est un vecteur 3D et M_2 une matrice de taille 3×3. Etant donné les moments M_0, M_1 et M_2 (calculés en une passe de l'image), la matrice de covariance est définie par [OB91] :

$$C = \frac{1}{M0}M2 - \frac{M1M1^t}{M0^2}.$$

Cette matrice réelle et symétrique peut être diagonalisée sur une base orthogonale égale aux vecteurs propres de la matrice de covariance. De plus, chaque valeur propre de la matrice est égale à la variance de la projection des couleurs de l'image sur le vecteur propre associé. L'information portée par un vecteur propre v_i est classiquement mesurée par :

$$\frac{\lambda_i}{\sum_{j=1}^{3} \lambda_j}$$

où λ_i représente la valeur propre du vecteur propre v_i.

Les tests établis par Otha et confirmés par nos propres expériences (voir Table 4.1) montrent que les vecteurs propres d'une image "naturelle" s'écartent très peu de trois directions constantes. Le terme "image naturelle" s'oppose ici à “image de synthèse”. On désignera par “image naturelle” une image habituellement perçue par l’oeil.

	V_1			V_2			V_3		
	R	G	B	R	G	B	R	G	B
Zelda	0.38	0.33	0.27	0.44	0.25	-0.30	-0.06	0.45	-0.47
Lenna	0.35	0.40	0.23	0.44	0.16	-0.39	-0.20	0.40	-0.39
Fleurs	0.41	0.38	0.20	0.32	0.01	-0.67	-0.36	0.47	-0.16
Anemone	0.31	0.37	0.31	0.49	0.01	-0.48	-0.26	0.45	-0.28
Mandrill	0.18	0.34	0.47	0.68	-0.02	-0.28	-0.15	0.52	-0.32
Moyennes	0.33	0.36	0.3	0.47	0.08	-0.42	-0.21	0.46	-0.32
σ	0.09	0.03	0.11	0.13	0.12	0.16	0.11	0.04	0.12

Tab. 4.1 – Les vecteurs propres V_1, V_2 et V_3 sont ordonnés dans l’ordre décroissant de leurs valeurs propres. Leurs coordonnées sont exprimées dans l’espace RGB pour chacune des images de test. Les moyennes et les écarts types calculés sur chacune des coordonnées sont affichés sur la dernière et l’avant dernière ligne.

On constate sur la Table 4.1 que le vecteur propre de plus grande valeur propre V_1 peut être approximé par le vecteur $(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$. Ce vecteur définit un axe vectoriel correspondant à la luminosité. Le second vecteur propre V_2 peut quant à lui être approximé par $(\frac{1}{2}, 0, -\frac{1}{2})$. Ceci correspond à l’opposition rouge-bleu. Le dernier vecteur propre V_3 peut être approximé par : $(-\frac{1}{4}, \frac{1}{2}, -\frac{1}{4})$ ce qui correspond à l’opposition vert-violet. Ces trois vecteurs fournissent une nouvelle base permettant de définir un nouvel espace de couleurs déduit de l’espace RGB par la transformation suivante :

$$\begin{cases} I_1 &= \frac{R+G+B}{3} \\ I_2 &= R-B \\ I_3 &= \frac{2G-(R+B)}{2}. \end{cases}$$

Étant donnés une image naturelle et son multi-ensemble associé, les axes I_1 I_2 et I_3 sont par construction proches des vecteurs propres de la matrice de covariance du multi-ensemble. Ceci a deux conséquences intéressantes en analyse d’image :

- Les variances des axes I_1, I_2 et I_3 seront importantes. En terme d’analyse de données, un axe de forte variance correspond à un axe contenant beaucoup d’informations.
- Les covariances entre les axes I_1, I_2 et I_3 seront faibles. Ceci signifie que chaque axe contient un seul type d’information. Par exemple, l’axe R du système RGB contient une information de luminance (l’intensité du rouge) et une information de chrominance. Cette double information luminance/chrominance est partagée entre l’axe I_1 qui code la luminance et les axes I_2 et I_3 codant la chrominance (voir [SB85] pour plus de détails sur ce point).

Le système $I_1I_2I_3$ contient donc un axe représentant la luminosité et deux axes représentant la chromaticité. Remarquons que les axes I_2 et I_3 ont été multipliés par un facteur 2. Ceci permet de renforcer l'importance de la chromaticité par rapport à la luminosité afin d'être plus en adéquation avec la vision humaine.

Les espaces de type LHS

Tous les espaces de couleurs que nous avons décrits jusqu'a présent étaient basés sur des espaces cartésiens (un point est défini par ses projections sur les trois axes). Une autre façon de représenter une couleur consiste à passer dans un espace cyclindrique (voir Figure 4.9).

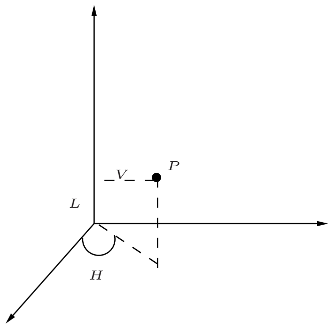


FIG. 4.9 – *Coordonnées d'un point P en représentation cyclindrique*

Dans ce cas, le triplet représentant la couleur contiendra la luminosité (L), un écart angulaire (H) par rapport à une couleur donnée et une distance (S) par rapport au blanc.

- La luminance L, également appelée brillance(Lightness ou value en anglais), indique si une couleur est claire ou sombre et dans quelle mesure elle se rapproche du blanc ou du noir.
- la teinte H(Hue en anglais) permet de déterminer la couleur souhaitée (rouge, vert, jaune, ...).
- La saturation S également appelée intensité, (ou chroma en anglais) représente la distance d'une couleur à l'axe des blancs. Elle mesure la pureté des couleurs et permet de distinguer les couleurs "vives" des couleurs "pastels" ou "délavées".

Si nous basons notre espace LHS sur l'espace $L^*a^*b^*$, les composantes L, H et S sont définies de la façon suivante :

$$
\begin{aligned}
L &= L^* \\
H &= arctan(\tfrac{a^*}{b^*}) \\
S &= \sqrt{(a^*)^2 + (b^*)^2}
\end{aligned}
$$

On peut, de même, définir un espace LHS sur l'espace $L^*u^*v^*$ en remplaçant respectivement a^* et b^* par u^* et v^* dans l'équation précédente. Plus généralement, un espace LHS peut être construit sur tout espace de couleurs à coordonnées cartésiennes dès que l'on a identifié l'axe représentant l'intensité lumineuse.

Les espaces LHS sont souvent très utiles dans les interfaces graphiques où ils permettent de définir une couleur de façon intuitive(voir Figure 4.10). L'utilisateur définit tout d'abord la luminosité de sa couleur(L), puis son écart par rapport à une couleur de réference (H) avant de définir son degré de blanc (S). Toutefois, le passage en coordonnées cyclindriques n'est pas très adapté au traitement d'images dans la mesure où il nous amène à traiter simultanément des grandeurs homogènes à des longueurs (L et S) et des angles (H). De plus, il est souvent algorithmiquement délicat de manipuler des différences d'angles. De fait, deux angles de valeurs ϵ et $2\pi - \epsilon$ seront très éloignés tout en définissant des couleurs très proches.

FIG. 4.10 – *Un exemple d'interface graphique permettant de choisir une couleur à partir du modèle HSV.*

Le modèle de Munssel

Le modèle colorimétrique de Munsell fut conçu par Albert Munsell en 1898 et révisé par la société américaine d'optique (*Optical Society of America*) en 1943. Les trois axes de ce modèle se nomment la teinte, la saturation, et la luminosité. Ils correspondent aux trois axes définies dans un espace de type LHS. L'originalité de cet espace par rapport aux espaces de type LHS classique consiste dans le faible nombre de valeurs définies sur chacun des axes. Cette faible valeur d'échantillonage permet de donner un nom à chacune des couleurs, ce qui serait totalement irréaliste dans la plupart des espaces de couleurs.

Le nom des différentes couleurs est fixé par la teinte en fonction du procédé suivant : La teinte est représenté par un anneau circulaire décomposé en 10 sections égales (voir Figure 4.11). Munsell à nomé ces sections : rouge, jaune-rouge, jaune, vert-jaune,vert, bleu-vert,bleu, pourpre-bleu, pourpre, et rouge-pourpre (R, YR, Y, GY,G,BG,B,PB,P et RP en abrégé et en anglais). Chaque section peut être divisé en sous-sections si une plus gande précision est souhaitée. Dans ce cas le numéro de la sous section précède le nom de la section (par exemple 5R ou 8BG).

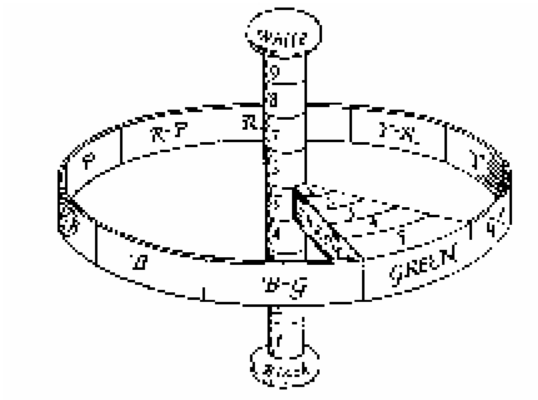


FIG. 4.11 – *Modèle colorimétrique de Munsell. Dans cette figure tiré d'une publication anglaise la hue correspond à la teinte, la chroma à la saturation et la value à la luminosité*

4.3 Exercices

4.3.1 Questions de compréhension

- Soit les espaces Lu^*v^*, La^*b^*, $I_1I_2I_3$ et HSV, indiquez les espaces les plus adaptés à chacune des applications suivantes (justifiez la réponse en quelques lignes) :
  1. Définir interactivement des couleurs.
  2. Travailler sur des coordonnées non corrélées.
  3. Avoir une notion de distance qui correspond aux notions physiologiques.
- Définissez en quelques lignes le spectrum locus et la ligne des pourpres.
- Que représentent les ellipses de MacAdam ? Ont-elles toutes la même taille et la même orientation ? Quelle est la conséquence de cette dernière propriété.
- Les espaces uniformes correspondent ils exactement à la perception humaine des distances entre couleur ? Justifiez la réponse en quelques lignes.

4.3.2 Conversion entre espaces couleurs

On désigne par $P = (p_1, p_2, p_3)$ et $Q = (q_1, q_2, q_3)$ deux systèmes de primaires.

1. Justifiez en quelques phrases, l'existence pour tout i appartenant à $\{1, 2, 3\}$ d'un unique triplet $(a_{1,i}, a_{2,i}, a_{3,i})$ tel que :

$$\forall i \in \{1,2,3\} \quad S^t p_i = a_{1,i} S^t q_1 + a_{2,i} S^t q_2 + a_{3,i} S^t q_3$$

2. Soit f un spectre quelconque. On désigne par $\alpha^P = (\alpha_1^P, \alpha_2^P, \alpha_3^P)$ et $\alpha^Q = (\alpha_1^Q, \alpha_2^Q, \alpha_3^Q)$ les coordonnées associées au spectre f dans les systèmes P et Q. On a donc :

$$\left\{ \begin{array}{rcl} S^t f & = & \alpha_1^P S^t p_1 + \alpha_2^P S^t p_2 + \alpha_3^P S^t p_3 \\ & = & \alpha_1^Q S^t q_1 + \alpha_2^Q S^t q_2 + \alpha_3^Q S^t q_3 \end{array} \right.$$

   Montrez que :

$$\alpha^Q = \begin{pmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{pmatrix} \alpha^P$$

   Cette matrice est appelée la matrice de conversion de l'espace P à l'espace Q.

4.3.3 Définition d'un espace couleur à partir de l'espace XYZ

1. Soient (X_1, Y_1, Z_1), (X_2, Y_2, Z_2) et (X_3, Y_3, Z_3) les coordonnées dans l'espace XYZ de trois primaires p_1, p_2, p_3. Donnez la matrice de conversion de l'espace $P = (p_1, p_2, p_3)$ à l'espace XYZ. Indiquez en quelques lignes pourquoi la matrice de conversion est inversible.
2. Montrez que l'ensemble des triplets (X, Y, Z) dont la projection sur le plan chromatique est égale au point (x, y) s'écrit :

$$\begin{pmatrix} tx \\ ty \\ tz \end{pmatrix}$$

   où $t = X + Y + Z$ et $z = 1 - x - y$.
   Que décrit cet ensemble de points ? (une ou deux lignes)
3. On désigne par (x_1, y_1),(x_2, y_2) et (x_3, y_3) les projections dans le plan chromatique de trois primaires de coordonnées (t_1x_1, t_1y_1, t_1z_1),(t_2x_2, t_2y_2, t_2z_2) et (t_3x_3, t_3y_3, t_3z_3). On impose que le point de coordonnées $(1, 1, 1)$ dans ce nouveau système corresponde à un blanc W de coordonnées (w_1, w_2, w_3) dans le système XYZ. Montrez que l'on a :

$$\begin{pmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ z_1 & z_2 & z_3 \end{pmatrix} \begin{pmatrix} t_1 \\ t_2 \\ t_3 \end{pmatrix} = \begin{pmatrix} w_1 \\ w_2 \\ w_3 \end{pmatrix}$$

4. Expliquez en quelques lignes pourquoi (et comment) si $\{(x_1, y_1, z_1), (x_2, y_2, z_2), (x_3, y_3, z_3)\}$ forme un système libre de $\mathbb{R}^3$ nous pouvons construire la matrice de conversion de l'espace XYZ au nouvel espace à partir des points (x_1, y_1), (x_2, y_2), (x_3, y_3) et des coordonnées du blanc W.
5. Expliquez en quelques lignes les avantages de ce mode de spécification d'un espace couleur par rapport à :
   (a) Une spécification des trois spectres p_1, p_2 et p_3.
   (b) La donnée des coordonnées de p_1, p_2 et p_3 dans l'espace XYZ.

4.3.4 Optimisation des transformations

Donné un nombre réel n, on désigne par n_0, sa valeur entière inférieure, et n_1 sa valeur entière supérieure. Le point n est donc compris dans le segment $[n_0, n_1]$ (figure 4.12). De même, un point de coordonnées $n = (x, y)$ est compris entre 4 points $(n_{00}, n_{01}, n_{10}, n_{11})$. Dans ce cas, n_{01} désigne le point de coordonnées entière $([x], [y]+1)$ où $[x]$ est la partie entière inférieure. Ce schéma ce généralise en $3D$ où un point est inclus dans un cube défini par 8 points n_{000} à n_{111} (figure 4.13).

En dimension 1(figure 4.12(a)) : On suppose que n_0 et n_1 sont associés à des valeurs p_0 et p_1. La valeur p associée à n peut être définie par interpolation linéaire. Si l'on appelle cette fonction $L^1(t)$ on a :

$$L^1(t) = p_0 + (p_1 - p_0)t$$

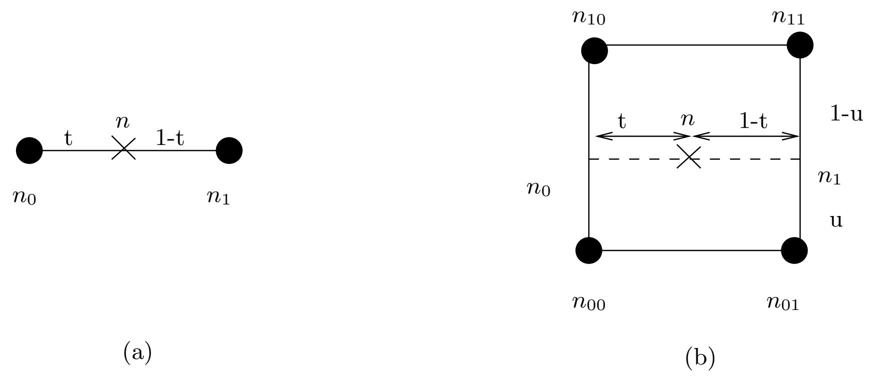


FIG. 4.12 – *Interpolation en dimensions 1 et 2*

1. En dimension 2 (figure 4.12(b)) : Chaque point n_{ij} est associé à une valeur p_{ij}. Donnez la valeur p associée à n en fonction de t et u.
   – Calculer la valeur associé aux points n_0 et n_1 par interpolation linéaire.
   – Interpolez linéairement entre n_0 et n_1 pour touver la valeur de n.
   La fonction obtenue s'appellera $L^2(t, u)$.

2. En dimension 3(figure 4.13) : On désigne par L_0^2 l'interpolation bilinéaire sur le rectangle $(n_{000}, n_{001}, n_{011}, n_{010})$ et par L_1^2 l'interpolation bilinéaire sur le rectangle $(n_{100}, n_{101}, n_{111}, n_{110})$. n_0 et n_1 représentent la projection de n sur chacun de ces rectangles. Donnez la valeur de p_0 et p_1 en utilisant L_0^2 et L_1^2. Déduisez en la valeur p de n à l'aide d'une interpolation linéaire entre n_0 et n_1. La fonction s'appellera $L^3(t, u, v)$.

On suppose que l'on a trois tableaux de float c_1, c_2 et c_3 de taille [32][32][32]. L'entrée $c_l[i][j][k]$ code la composante l de la transformée du triplé RGB $(i * 8, j * 8, k * 8)$ dans un espace couleur donné.

1. A t'on calculé la transformée de tous les triplés RGB ? Pourquoi ?
2. Utilisez les question précédentes pour déduire la transformée de tout triplé (R, G, B).On donnera pour cela :
   - Les coordonnées des points n_{000} à n_{111} que l'on doit utiliser ;
   - la valeur de t, u et v ;
   - la valeur associée à un point n_{ijk} pour la composante 1 ;
   - Une description en quelques lignes de l'algorithme calculant la valeur de la composante 1 d'un triplet RGB.
3. Si l'on doit calculer les tableaux c_1, c_2 et c_3 au début de l'algorithme, cette méthode est elle efficace pour calculer la transformée des couleurs d'une image 256×256 ? Proposez une solution (simple).

4.3.5 Constance Chromatique des matériaux

On suppose que la réflécion d'un rayon lumineux monochormatique sur une surface atténue sa puissance sans en altérer la longueur d'onde. On suppose de plus que cette atténuation s'exprime comme un produit de deux termes :

$$\beta_k^r = m(x, y) d_k \beta_k$$

où β_k et β_k^r représentent respectivement la puissance du rayon incident et réfléchi pour le spectre monochormatique e_k. Le facteur d_k est un facteur d'atténuation dépendant de la longueur d'onde tandis que $m(x, y)$ ne dépend que de facteurs géométriques (normale à la surface, angle du rayon incident, angle du rayon réfléchi,...).

1. En vous basant sur le cours exprimez par une relation matricielle le lien entre le spectre réfléchi f^r et le spectre incident f.
2. Soit A une matrice d'appariement définissant un espace colorimétrique. Indiquez le triplé associé à la couleur réfléchie $A^t f^r$ en fonction du spectre incident f.
3. En vous appuyant sur la question précédente montrez que les rapports de coordonnées $\frac{(A^t f^r)_i}{(A^t f^r)_j}, i \neq j \in \{1, 2, 3\}$ sont indépendant de l'atténuation géométrique $m(x, y)$.
4. Toujours en se basant sur la question précédente, indiquez pourquoi une image exprimée dans le diagramme de chormaticité (x, y) ne prend pas en compte la géométrie des objets de la scène.

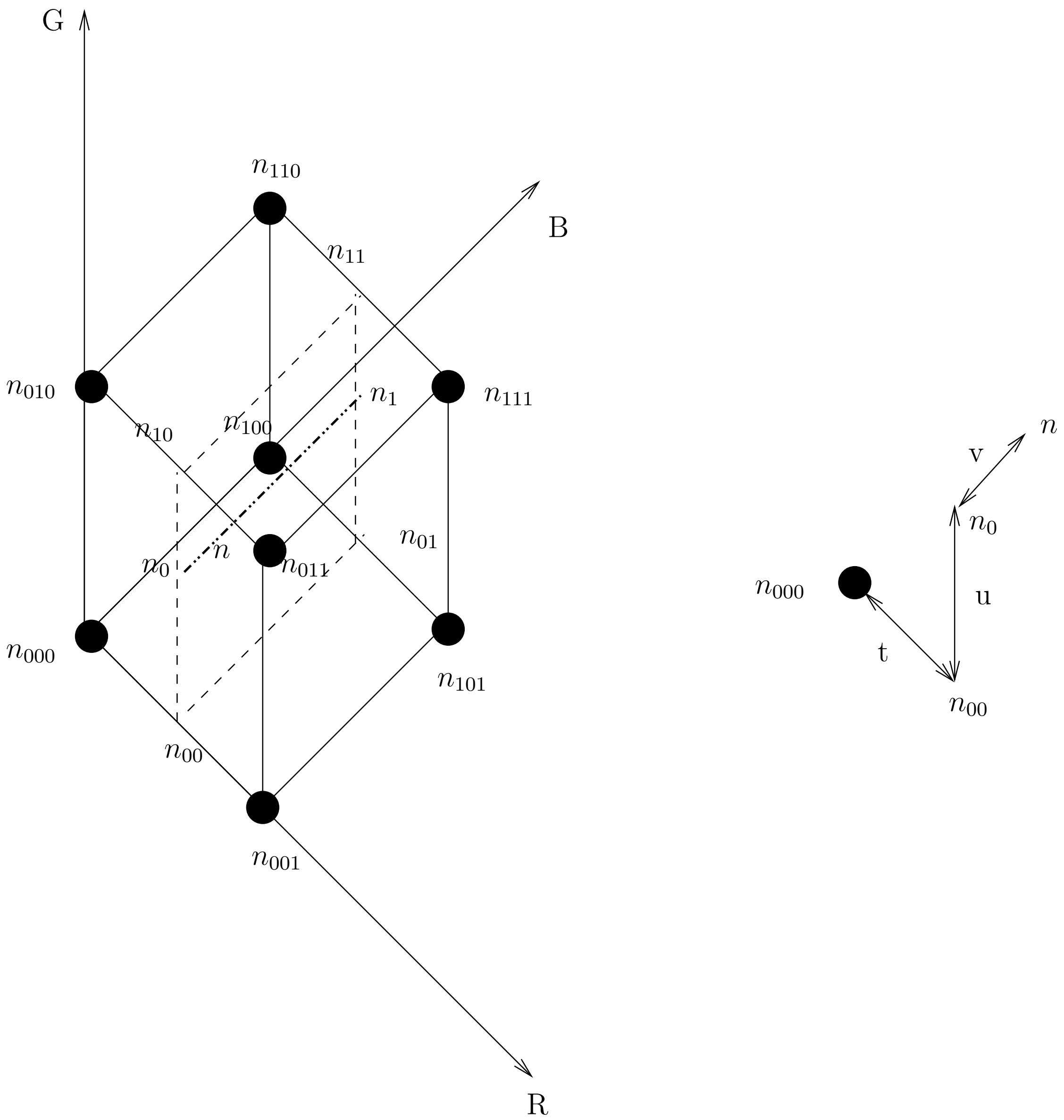


FIG. 4.13 – *Interpolation 3D*

4.4 Correction des Exercices

4.4.1 Questions de compréhension

- Soit les espaces Lu^*v^*, La^*b^*, $I_1I_2I_3$ et HSV, indiquez les espaces les plus adaptés à chacune des applications suivantes (justifiez la réponse en quelques lignes) :
  1. Définir interactivement des couleurs.
  2. Travailler sur des coordonnées non corrélées.
  3. Avoir une notion de distance qui correspond aux notions physiologiques.

  Les espaces de types LHS permettent de sélectionner une couleur en précisant
  - Sa luminosité (L)
  - Sa distance par rapport au blanc ou sa saturation (S)
  - Une certaine distance exprimée en mesure d'angle vis à vis d'une couleur de référence.

  Ces trois paramètres apparaissent suffisamment non corrélés pour permettre un positionnement itératif de chacun de ces paramètres par l'utilisateur. Ceci n'est pas le cas, par exemple, de l'espace RGB ou toute modification d'une des composantes entraîne une modification de la luminance qui peut ne pas être souhaitée par l'utilisateur.
  L'espace $I_1I_2I_3$ est conçu pour être une approximation des vecteurs propres de la matrice de covariance du multi-ensemble d'une image naturelle. Ici image "naturelle" est employé par opposition au terme image de synthèse. Les vecteurs propres de la matrice de covariance étant par définition non corrélés l'espace $I_1I_2I_3$ peut être considéré comme un des espace couleur dont la corrélation entre les axes est minimum.
  Les espaces Lu^*v^* et La^*b^* sont des espaces qualifiés de pseudo-uniformes. Ceci signifie que ces espaces sont conçus pour que la distance euclidienne entre deux couleurs approxime la notion de distance physiologique. Ne perdons toutefois pas de vue que ces espaces ont été définis à partir d'expérience présentant deux taches colorées. Ces expérience ne rendent donc que très approximativement compte de la vision usuelle des couleurs où les deux couleurs dont on apprécie la distance sont inclues dans un environnement coloré.
- Définissez en quelques lignes le spectrum locus et la ligne des pourpres. Le

  spectrum-locus est la courbe définie dans le diagramme xy représentant l'ensemble des couleurs mono-chromatique. Toute couleur visible est une combinaison de points de cette courbe. La ligne des pourpres est la ligne joignant les points de la courbe de longueur d'onde minimale (les bleus) et maximales (les rouges).

- Que représentent les ellipses de MacAdam ? Ont-elles toutes la même taille et la même orientation ? Quelle est la conséquence de cette dernière propriété. Les ellipses de Mac Adam représentent l'ensemble des couleurs juste

  discernables à partir d'une couleur donnée. Ces ellipses n'ont ni la même taille ni la même orientation. Ceci implique qu'il est impossible d'établir une transformation globale de l'espace qui transforme toutes ces ellipses en cercles de même taille. Il est donc impossible de définir un espace de couleur où la distance euclidienne corresponde à la distance physiologique.
- Les espaces uniformes correspondent ils exactement à la perception humaine des distances entre couleur ? Justifiez la réponse en quelques lignes.

  Les espaces uniformes sont conçues pour que des couleurs à la même distance *visuelle* d'une couleur de référence soient à la même distance Euclidienne de celle-ci dans l'espace considéré. Hors comme l'a montré les expériences de Mac Adam l'ensemble des couleurs à une distance visuelle donnée d'une couleur de référence décrit approximativement une ellipse dans le diagramme de Chromaticité xy. Ces ellipses ayant toutes des taille et des orientations différentes il est mathématiquement impossible de trouver une transformation de $\mathbb{R}^3$ dans $\mathbb{R}^3$ qui transforme toutes ces ellipses en cercle. Les transformées des ellipses dans les espaces uniformes ne sont donc qu'approximativement circulaire et la distance entre couleur dans ces espaces ne correspond approximativement à la distance visuelle. De plus, ces calculs de distance n'ont de sens qu'entre des couleurs proches (mais ceci est un autre débat).

4.4.2 Conversion entre espaces couleur

L'existence d'un triplet unique est justifiée par le fait que $(S^t p_1, S^t p_2, S^t p_3)$ forme une base de $\mathbb{R}^3$. Donc tout vecteur s'écrit de manière unique dans cette base.

On déduit du cours que :

$$S^t f = \alpha_1^P S^t p_1 + \alpha_2^P S^t p_2 + \alpha_3^P S^t p_3$$

On obtient en exprimant $S^t p_1, S^t p_2, S^t p_3$ en fonction de $S^t q_1, S^t q_2, S^t q_3$:

$$S^t f = (\alpha_1^P a_{1,1}+\alpha_2^P a_{1,2}+\alpha_3^P a_{1,3}) S^t q_1 + (\alpha_1^P a_{2,1}+\alpha_2^P a_{2,2}+\alpha_3^P a_{2,3}) S^t q_2 + (\alpha_1^P a_{3,1}+\alpha_2^P a_{3,2}+\alpha_3^P a_{3,3}) S^t q_3$$

Or :

$$S^t f = \alpha_1^Q S^t q_1 + \alpha_2^Q S^t q_2 + \alpha_3^Q S^t q_3$$

cette décomposition étant unique. On a donc :

$$\left\{ \begin{array}{lcl} \alpha_1^Q & = & \alpha_1^P a_{1,1} + \alpha_2^P a_{1,2} + \alpha_3^P a_{1,3} \\ \alpha_2^Q & = & \alpha_1^P a_{2,1} + \alpha_2^P a_{2,2} + \alpha_3^P a_{2,3} \\ \alpha_3^Q & = & \alpha_1^P a_{3,1} + \alpha_2^P a_{3,2} + \alpha_3^P a_{3,3} \end{array} \right.$$

Autrement dit :

$$\alpha^Q = \begin{pmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{pmatrix} \alpha^P$$

4.4.3 Définition d'un espace couleur à partir de l'espace XYZ

1. Soit $Q = (q_1, q_2, q_3)$ le système de primaires de l'espace XYZ. On a en reprenant les notations de la question précédente :

$$\forall i \in \{1,2,3\} \quad S^t p_i = X_i S^t q_1 + Y_i S^t q_2 + Z_i S^t q_3$$

On a donc, toujours en reprenant les notations de la question précédente :

$$\forall i \in \{1,2,3\} \quad a_{1,i} = X_i;\ a_{2,i} = Y_i;\ a_{3,i} = Z_i$$

La matrice de conversion est donc égale à :

$$\alpha^{XYZ} = \begin{pmatrix} X_1 & X_2 & X_3 \\ Y_1 & Y_2 & Y_3 \\ Z_1 & Z_2 & Z_3 \end{pmatrix} \alpha^P$$

où α^{XYZ} et α^P représentent les coordonnées d'une couleur dans l'espace XYZ et dans le nouvel espace.
Cette matrice de conversion peut s'interpreter comme une matrice de changement de base entre $(S^t p_1, S^t p_2, S^t p_3)$ et $(S^t q_1, s^t q_2, S^t q_3)$, elle est donc inversible.

2. On a d'après la définition de la projection sur le plan chromatique :

$$x = \frac{X}{X+Y+Z} \Longrightarrow X = x(X+Y+Z) = tx$$

de même pour chacune des composantes. L'ensemble des points obtenus décrit une droite de pente (x, y, z) passant par l'origine.

3. Si nous exprimons les coordonnées des trois primaires (p_1, p_2, p_3) dans le système XYZ nous avons :

$$\forall i \in \{1,2,3\} \quad S^t p_i = t_i x_i S^t q_1 + t_i y_i S^t q_1 + t_i z_i S^t q_3$$

Soit f_W un spectre associé au blanc W. On a :

$$S^t f_W = 1.S^t p_1 + 1.S^t p_2 + 1.S^t p_3$$

En exprimant à nouveau chaque $S^t p_i$ en fonction de $(S^t q_1, S^t q_2, S^t q_3)$ on obtient :

$$\begin{aligned} S^t f_W &= (t_1 x_1 + t_2 x_2 + t_3 x_3) S^t q_1 + (t_1 y_1 + t_2 y_2 + t_3 y_3) S^t q_2 + (t_1 z_1 + t_2 z_2 + t_3 z_3) S^t q_3 \\ &= w_1 S^t q_1 + w_2 S^t q_2 + w_3 S^t q_3 \end{aligned}$$

On a donc :

$$\begin{cases} t_1x_1 + t_2x_2 + t_3x_3 & = & w_1 \\ t_1y_1 + t_2y_2 + t_3y_3 & = & w_2 \\ t_1z_1 + t_2z_2 + t_3z_3 & = & w_3 \end{cases}$$

Ou sous forme matricielle :

$$\begin{pmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ z_1 & z_2 & z_3 \end{pmatrix} \begin{pmatrix} t_1 \\ t_2 \\ t_3 \end{pmatrix} = \begin{pmatrix} w_1 \\ w_2 \\ w_3 \end{pmatrix}$$

4. Si $\{(x_1, y_1, z_1), (x_2, y_2, z_2), (x_3, y_3, z_3)\}$ forme un système libre, la matrice formée à partir de ces vecteurs est inversible. On peut donc déterminer (t_1, t_2, t_3) en fonction de (w_1, w_2, w_3). Ce qui nous donne les coordonnées des trois sources p_1, p_2 et p_3 dans le système XYZ. La réponse à la question 1 nous permet alors de construire la matrice de conversion du système $P = (p_1, p_2, p_3)$ au système XYZ. Cette matrice étant inversible, il nous suffit de l'inverser pour obtenir la matrice de transformation de l'espace XYZ au nouvel espace. Le lecteur currieux pourra vérifier qu'il suffit que les points (x_1, y_1), (x_2, y_2) et(x_3, y_3) ne soient pas alignés pour que $\{(x_1, y_1, z_1), (x_2, y_2, z_2), (x_3, y_3, z_3)\}$ forme un système libre.
5. Cette façon de spécifier un espace couleur permet de ne spécifier que 9 valeurs (3×2 pour (x_i, y_i) et 3 pour W) ce qui est bien plus petit que la donnée des $3 \times N$ valeurs définissant les spectres p_1, p_2, p_3. De plus ce mode de spécification permet de spécifier explicitement le blanc de l'espace couleur ce qui n'est donnée que de manière implicite par une spécification des coordonnées de p_1, p_2 et p_3 dans l'espace XYZ.

4.4.4 Optimisation des transformations

Cas de la dimension 2

Les valeurs de p_0 et p_1 respectivement associées à n_0 et n_1 se déduisent par une interpolation $1D$.

$$\begin{array}{rcl} p_0 & = & L_0^1(t) = p_{00} + (p_{10} - p_{00})u \\ p_1 & = & L_1^1(t) = p_{01} + (p_{11} - p_{01})u \end{array}$$

où L_0^1 et L_1^1 correspondent respectivement aux interpolations sur les segments $[n_{00}, n_{10}]$ et $[n_{01}, n_{11}]$.

On en déduit la valeur p de n par interpolation $1D$:

$$\begin{array}{rcl} p & = & L_2(t, u) = p_0 + (p_1 - p_0)t \\ & = & p_{00} + (p_{10} - p_{00})u + (p_{01} - p_{00})t + (p_{11} + p_{00} - p_{01} - p_{00})tu \end{array}$$

Cas de la dimension 3

On a d'après les notations de la question :

$$\begin{array}{rcl} p_0 & = & L_0^2(t,u) = p_{000} + (p_{010} - p_{000})u + (p_{001} - p_{000})t + (p_{011} + p_{000} - p_{001} - p_{000})tu \\ p_1 & = & L_1^2(t,u) = p_{100} + (p_{110} - p_{100})u + (p_{101} - p_{100})t + (p_{111} + p_{100} - p_{101} - p_{100})tu \end{array}$$

La valeur de p se déduit de p_0 et p_1 par interpolation linéaire :

$$p = p_0 + (p_1 - p_0)v$$

Je laisse au lecteur currieux le soins de développer le calcul. On peut toutefois montrer [BK02] que ce calcul peut s'effectuer avec 7 additions et 7 multiplications en utilisant 12 valeurs stockées dans chaque sommet.

4.4.5 Application à la couleur

On n'a pas pu transformer tous les triplets RGB puisque l'on a calculé que 32^3 valeurs au lieu des 256^3 triplets RGB usuellement utilisés.

Étant donné un triplet (R,G,B), notons par (r, g, b) le triplet $(\frac{R}{8}, \frac{G}{8}, \frac{B}{8})$ les points n_{000} et n_{111} ont à ce moment là pour coordonnées :

$$\begin{array}{rcl} n_{000} & = & ([r], [g], [b]) \\ n_{111} & = & ([r]+1, [g]+1, [b]+1) \end{array}$$

Les valeurs de t, u et v sont égales à :

$$\begin{array}{rcl} t & = & r - [r] \\ u & = & g - [g] \\ v & = & b - [b] \end{array}$$

Étant donnée une transformée de l'espace RGB dans un espace quelconque on calcule pour tout triplet $(i, j, k) \in \{0, \ldots, 32\}^3$ la transformée (c_1, c_2, c_3) du triplet $(R = i * 8, G = j * 8, b = k * 8)$. On a donc calculé la transformée de nos triplets RGB par pas de 8 sur chacune de nos composantes. Les valeurs c_1, c_2 et c_3 sont respectivement stockées dans tab_c_1, tab_c_2 et tab_c_3. Un point n de coordonnées entières $(i, j, k) \in \{0, \ldots, 32\}^3$ se voit donc associer les trois valeurs $tab\_c_1[i][j][k]$, $tab\_c_2[i][j][k]$ et $tab\_c_3[i][j][k]$.

Donné un triplet (R,G,B) on calcule le triplet (r, g, b) associé auquel on associe une valeur en utilisant les valeurs des tableaux tab_c_1, tab_c_3, tab_c_3 codant les valeurs de la transformée sur les points (r,g,b) de coordonnées entières.

Ce type de calcul est bien évidement parfaitement inefficace si l'on ne stocke pas dans un fichier les résultats de la transformation une fois pour toute.

4.4.6 Constance Chromatique des matériaux

En se basant sur le cours, un spectre incident s'exprime dans la base $(e_k)_{k \in \{1,\ldots,N\}}$ par :

$$f = \sum_{k=1}^{N} \beta_k e_k$$

On a donc pour le spectre réfléchi :

$$f^r = \sum_{k=1}^{N} \beta_k^r e_k = m(x,y) \sum_{k=1}^{N} d_k \beta_k e_k$$

En notation vectorielle on passe donc de $f^t = (\beta_1, \ldots, \beta_N)$ à $(f^r)^t = m(x,y)(d_1\beta_1, \ldots, d_N\beta_N)$ où ^{t} représente la transposé. La transformation matricielle permettant de multiplier chaque terme d'un vecteur par une certaine valeur est bien évidement la multiplication par une matrice diagonale. On a donc :

$$f^r = m(x,y)Df \text{ avec } D = \begin{pmatrix} d_1 & 0 & \ldots & 0 \\ 0 & d_2 & \ldots & 0 \\ \vdots & & & \vdots \\ 0 & 0 & \ldots & d_N \end{pmatrix}$$

Si A est une matrice d'appariement, la couleur c d'un spectre réfléchie est donc égale à :

$$c = A^t f = m(x,y)A^t Df$$

Posons à présent pour simplifier les notations : $c^t = (c_1, c_2, c_3)$ et $(c^m)^t \underset{not.}{=} (A^t Df)^t = (c_1^m, c_2^m, c_3^m)$. On a $c = m(x,y)c^m$ et donc si $m(x,y) \neq 0$:

$$\forall (i,j) \in \{1,2,3\}^2,\ i \neq j\ \frac{(A^t f^r)_i}{(A^t f^r)_j} = \frac{c_i}{c_j} = \frac{m(x,y)c_i^m}{m(x,y)c_j^m} = \frac{c_i^m}{c_j^m}$$

Ce résultat est intéressant, car l'absorption de chaque longueur d'onde d'un rayon incident (donc la matrice D) dépend uniquement du matériaux sur lequel les rayons sont réfléchis. Le résultat précédent, indique que pour une source lumineuse donnée, le rapport des coordonnées colorimétriques reste constant sur tous les pixels correspondant à un même matériaux. Notons, au passage que pour des raisons de stabilité numérique on utilise en général plutôt les rapport :

$$\frac{c_i}{\sqrt{\sum_{i=1}^{3} c_j^2}}, i \in \{1,2,3\}$$

sans que cela change quoi que ce soit au principe de la méthode.

Soit $c^t = (X, Y, Z)$ la couleur du spectre réfléchi f^r dans l'espace XYZ et $(c^m)^t = (X_m, Y_m, Z_m)$ la couleur du spectre Df dans le même espace. On a d'après les questions précédentes : $c = m(p,q)c^m$. On note ici les facteurs géométriques (p,q) pour éviter toutes confusions avec les coordonnées (x,y) de l'espace colorimétrique. On a donc :

$$x = \frac{X}{X+Y+Z} = \frac{m(p,q)X_m}{m(p,q)(X_m+Y_m+Z_m)} = \frac{X_m}{X_m+Y_m+Z_m}$$

de même pour y. Les coordonnées (x,y) du diagramme de chormaticité sont donc insensibles aux facteurs géométriques.

Chapitre 5

Prétraitements d'images couleur

Les prétraitements d'images permettent d'améliorer la qualité de l'image en vue de traitements ultérieurs. Ces prétraitements peuvent classiquement être décomposés en plusieurs grandes familles :

- Les manipulations d'histogrammes dont le but est soit de mettre en relief telle ou telle partie de l'image soit d'augmenter la dynamique de celle-ci.
- Les opérations de filtrages (linéaires ou non) dont le but est de réduire le bruit présent dans l'image et usuellement assimilés à de hautes fréquences.
- Les réhaussements d'images dont le but est d'augmenter les contrastes entre les différentes régions de l'image.

Nous allons dans la suite de ce chapitre étudier la mise en oeuvre de ces traitements dans le cas des images couleurs et le précautions qui doivent être prises.

5.1 Les images couleur

5.1.1 Spécificité des images couleur

Les deux caractéristiques des images couleurs qui vont nous amener à modifier les méthodes conçues pour des images en niveaux de gris sont :

- Le nombre des valeurs possibles que peut prendre un pixel. En effet, en niveau de gris le niveau de gris d'un pixel varie usuellement entre 0 et 255. Dans le cas d'une image couleur et si nous utilisons l'espace RGB, chaque composante étant stockée sur 8 bits, un pixel peut prendre $256^3 \approx 16.10^6$ valeurs possibles. Cette taille importante va nous obliger à revoir les méthodes codant explicitement l'ensemble des valeurs que peut prendre un pixel.
- Tout espace de couleurs peut se voir comme un sous ensemble de $\mathbb{R}^3$ qui est par nature non ordonné. Or, de nombreux algorithmes (notamment les

algorithmes de morphologie mathématique) utilisent explicitement l'ordre des niveaux de gris. Cet ordre ayant disparu dans l'espace de couleurs, ces méthodes devront être modifiées.

5.1.2 Choix de l'espace de couleurs

Le choix d'un espace particulier dépend souvent de la méthode envisagée et du résultat recherché. L'espace RGB étant souvent utilisé pour coder les images, l'utilisation d'un espace particulier par une méthode nécessite d'utiliser une transformation T permettant de passer de l'espace RGB à l'espace désiré et une transformation inverse T^{-1} de façon à stocker l'image résultat (voir Figure 5.1).

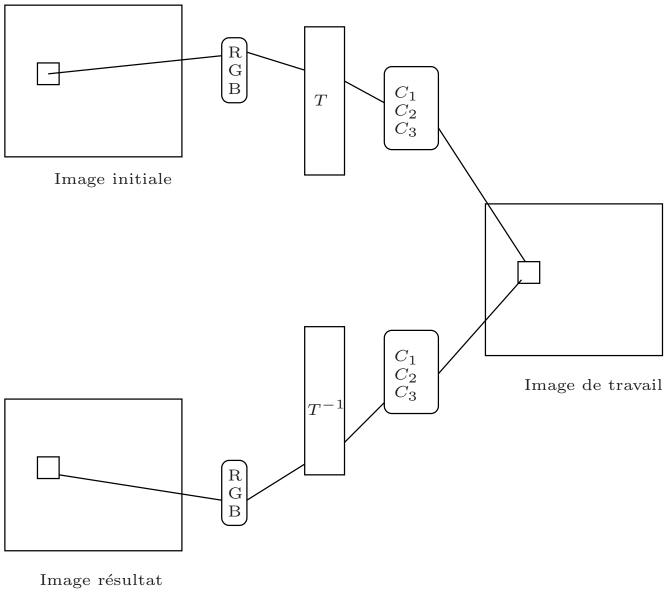


FIG. 5.1 – *Traitement dans un espace de couleurs $C_1C_2C_3$. Une transformations T et son inverse T^{-1} sont nécessaires pour créer l'image de travail et stocker le résultat.*

5.1.3 Approche Marginale et Approche Vectorielle

La plupart des algorithmes de traitement des images couleurs ne sont apparus que plusieurs dizaines d'années après les algorithmes de traitement des images en niveaux de gris. De nombreux chercheurs ont donc essayé d'adapter les algorithmes conçus pour des images en niveaux de gris à des images couleur. Cette approche est souvent appelée l'approche marginale. Elle consiste à appliquer les traitements indépendamment sur chaque composante, puis à fusionner les trois images obtenues de façon à former l'image couleur (voir Figure 5.2).

Cette approche suppose donc que les informations réparties sur les 3 axes ne sont pas corrélées et que l'on peut donc traiter une composante sans tenir compte des autres. Nous avons vu dans la section 4.2.3 que cette supposition est fausse dans le cas de l'espace couleur *RGB* et souvent discutable dans les autres espaces de couleurs. Cette approche permet toutefois de récupérer à moindre frais de nombreuses méthodes éprouvées dans le cadre des images en niveaux de gris.

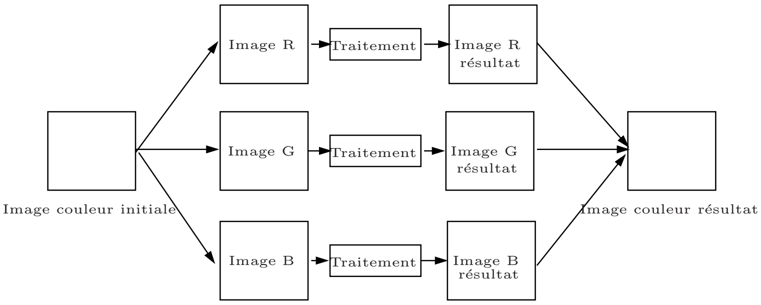


FIG. 5.2 – *Illustration de l'approche marginale dans l'espace RGB*

Inversement, la méthode vectorielle traite la couleur comme une seule entité et non comme la somme de 3 composantes indépendantes. Une couleur est à ce moment là considérée comme un vecteur 3D. Cette méthode a priori plus générale induit toutefois des traitements généralement plus complexes.

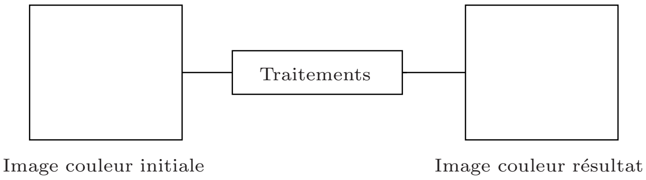


FIG. 5.3 – *Illustration de l'approche vectorielle*

5.2 Histogramme d'images couleurs

5.2.1 Multi-ensembles

De façon très générale, un multi-ensemble est un élément de $\mathcal{P}(\mathbb{R}^n) \times \mathcal{F}(\mathbb{R}^n, \mathbb{R})$, où $\mathcal{P}(\mathbb{R}^n)$ désigne l'ensemble des parties de $\mathbb{R}^n$ et $\mathcal{F}(\mathbb{R}^n, \mathbb{R})$ l'ensemble des applications de $\mathbb{R}^n$ dans $\mathbb{R}$. Un multi-ensemble est donc un couple (C, f) où C est un sous-ensemble de $\mathbb{R}^n$ et f une application de $\mathbb{R}^n$ dans $\mathbb{R}$. Cette définition étant un peu trop générale pour nos besoins, nous avons adopté la définition suivante :

Définition 1 *Soit $\mathcal{PB}(\mathbb{R}^n)$ l'ensemble des parties bornées de $\mathbb{R}^n$. Nous définissons $\mathcal{ME}_n$ l'ensemble des* **multi-ensembles** *de $\mathbb{R}^n$ comme suit :*

$$\mathcal{ME}_n = \{(C, f) \in \mathcal{PB}(\mathbb{R}^n) \times \mathcal{F}(\mathbb{R}^n, \mathbb{R}_+)/f_{|\mathbb{R}^n - C} == 0\}.$$

En d'autres termes, un multi-ensemble est la donnée d'un ensemble borné et d'une application réelle positive nulle hors de cet ensemble. Sauf mention contraire, tous les ensembles considérés dans ce chapitre seront des ensembles discrets. D'un point de vue informatique, les multi-ensembles peuvent être vus comme une extension de la notion d'histogramme à des espaces de dimension n. La fonction associée à un ensemble est souvent appelée la **fonction de fréquence** ou de **distribution**. L'application de ce concept à l'image est assez immédiate. En effet, si nous considérons une image I, on peut y associer un ensemble C, égal à l'ensemble de valeurs v associées aux pixels de l'image. Nous pouvons alors définir $f(v)$ comme le nombre de fois où le n-uplet $(v_1, \ldots, v_n)$ apparaît dans l'image. Par exemple, dans le cas d'une image couleur, $f(0, 0, 0)$ représente le nombre de pixels noirs contenus dans l'image. La fonction de fréquence associe un poids à chaque élément du multi-ensemble. Ces poids peuvent être utilisés pour définir les moments, la moyenne ou la variance d'un multi-ensemble.

Définition 2 *Soit $\mathcal{ME}_n$ l'ensemble des multi-ensembles de $\mathbb{R}^n$. On définit sur $\mathcal{ME}_n$ les fonctions $\boldsymbol{M_0}$, $\boldsymbol{M_1}$ et $\boldsymbol{M_2}$ appelées* **moments** *d'ordre 0, 1 et 2 par :*

$$\boldsymbol{M_0}\left(\begin{array}{ccc} \mathcal{ME}_n & \to & \mathbb{R} \\ (C, f) & \mapsto & \sum_{v \in C} f(v) \end{array}\right.$$

$$\boldsymbol{M_1}\left(\begin{array}{ccc} \mathcal{ME}_n & \to & \mathbb{R}^n \\ (C, f) & \mapsto & \sum_{v \in C} f(v)(v_1, \ldots, v_n) \end{array}\right.$$

$$\boldsymbol{M_2}\left(\begin{array}{ccc} \mathcal{ME}_n & \to & \mathbb{R}^n \\ (C, f) & \mapsto & \sum_{v \in C} f(v)(v_1^2, \ldots, v_n^2) \end{array}\right.$$

où $(v_1, \ldots, v_n)$ sont les n coordonnées du vecteur v.

On définit également le vecteur R_2 par :

$$R_2\left(\begin{array}{ccc} \mathcal{ME}_n & \to & \mathbb{R}^n \\ (C, f) & \mapsto & \sum_{\boldsymbol{c} \in \boldsymbol{C}} f(\boldsymbol{c})\boldsymbol{c}.\boldsymbol{c}^t \end{array}\right.$$

Il est bien sûr possible de définir des moments d'ordres supérieurs, mais les ordres 0, 1 et 2 seront suffisants pour notre étude. Afin d'alléger les notations, on omettra généralement la fonction de fréquence lors du calcul de l'image d'un multi-ensemble. L'image de (C, f) par $\boldsymbol{M_0}$, $\boldsymbol{M_0}((C, f))$, sera donc notée plus simplement $\boldsymbol{M_0}(C)$. On peut remarquer que $\boldsymbol{M_0}(\mathrm{C})$ également noté $|C|$ représente un scalaire alors que $\boldsymbol{M_1}(\mathrm{C})$ et $\boldsymbol{M_2}(\mathrm{C})$ sont des vecteurs de $\mathbb{R}^n$ où n est la dimension de l'espace dans lequel est plongé l'ensemble C. Usuellement, cette dimension est égale à 1 pour les images en niveaux de gris et 3 pour les images couleur. À partir des moments, on définit la moyenne, les variances et la matrice de covariance d'un multi-ensemble de la façon suivante :

Définition 3 *Avec les notations précédentes, les fonctions* **moyenne** *et* **variance** *sont définies sur* $\mathcal{ME}_n$ *par :*

$$\mu \left(\begin{array}{ccc} \mathcal{ME}_n & \rightarrow & \mathbb{R}^n \\ (C,f) & \mapsto & \dfrac{M_1(C)}{M_0(C)} \end{array} \right.$$

$$\forall i \in \{1, \dots n\}, \quad var_i \left(\begin{array}{ccc} \mathcal{ME}_n & \rightarrow & \mathbb{R}_+ \\ (C,f) & \mapsto & \dfrac{\boldsymbol{M_2}(C)_i}{\boldsymbol{M_0}(C)} - \mu^2(C)_i \end{array} \right.$$

$$Cov \left(\begin{array}{ccc} \mathcal{ME}_n & \rightarrow & \mathbb{R}_+ \\ (C,f) & \mapsto & \frac{R_2(\boldsymbol{C})}{|\boldsymbol{C}|} - \mu.\mu^t \end{array} \right.$$

où $\boldsymbol{M_2}(C)_i$ *et* $\mu(C)_i$ *représentent respectivement la* i^{eme} *coordonnée de* $\boldsymbol{M_2}(C)$ *et* $\mu(C)$.

La fonction var_i nous permet de mesurer la variance d'un multi-ensemble le long d'un axe et peut servir à mesurer l'homogénéité d'un multi-ensemble. La mesure d'homogénéité que nous allons utiliser, appelée **erreur quadratique**, peut être vue comme une extension de la variance pour les dimensions supérieures à 1.

Définition 4 *L'***erreur quadratique** *d'un multi-ensemble* (C, f) *notée* $\boldsymbol{SE}(C)$ *est donnée par :*

$$\boldsymbol{SE}(C) = \sum_{v \in C} f(v) \|v - \mu(C)\|^2.$$

L'erreur quadratique d'un multi-ensemble (C, f) est donc une mesure de l'écart des éléments de C par rapport à la moyenne $\mu(C)$. Si nous appliquons ce concept à l'image, la moyenne $\mu(C)$ représente le niveau de gris moyen (ou la couleur moyenne) de l'image et $\boldsymbol{SE}(C)$ représente l'écart de l'ensemble des couleurs de l'image par rapport à cette moyenne.

Des calculs relativement simples montrent que l'erreur quadratique peut s'exprimer en fonction des moments et des variances. De fait, nous avons pour un multi-ensemble (C, f) donné :

$$\boldsymbol{SE}(C) = \sum_{i=1}^{n} \boldsymbol{M_2}(C)_i - \frac{\boldsymbol{M_1}(C)_i^2}{\boldsymbol{M_0}(C)} = \boldsymbol{M_0}(C) \sum_{i=1}^{n} var_i(C). \tag{5.1}$$

L'erreur quadratique d'un multi-ensemble (C, f) peut donc également être vue comme la somme de ses variances pondérées par le cardinal de (C, f). Cette pondération permet de tenir compte du fait qu'une même erreur est a priori plus importante si elle se produit sur un grand ensemble que sur un petit.

Jusqu'à présent, nous avons vu la définition d'un multi-ensemble et les différentes mesures que l'on peut associer à celui-ci. Les multi-ensembles peuvent également être combinés pour créer de nouveaux multi-ensembles.

Définition 5 *Soit (C_1, f_1) et (C_2, f_2) deux éléments de $\mathcal{ME}_n$. La somme, l'union, l'intersection et la différence de (C_1, f_1) et (C_2, f_2) sont définies comme suit :*

$$\begin{array}{rcl}(C_1, f_1) + (C_2, f_2) & = & (C_1 \cup C_2, f_1 + f_2) \\ (C_1, f_1) \cup (C_2, f_2) & = & (C_1 \cup C_2, \max(f_1, f_2)) \\ (C_1, f_1) \cap (C_2, f_2) & = & (C_1 \cap C_2, \min(f_1, f_2)) \\ (C_1, f_1) - (C_2, f_2) & = & (C_1 - C_2, \max(0, f_1 - f_2))\end{array}$$

La combinaison de plusieurs multi-ensembles prend tout son intérêt en segmentation lorsque l'on manipule une image partitionnée en régions. Supposons par exemple que l'on désire fusionner deux régions R_1 et R_2 adjacentes dans l'image. Les multi-ensembles (C_1, f_1) et (C_2, f_2) correspondant à R_1 et R_2 ne sont a priori pas distincts. Le multi-ensemble correspondant à la fusion des régions R_1 et R_2 est alors égal à la somme des multi-ensembles (C_1, f_1) et (C_2, f_2). Inversement, si on désire partitionner une région R en deux sous-régions R_1 et R_2, nous avons la relation : $(C, f) - (C_1, f_1) = (C_2, f_2)$ où (C, f) désigne le multi-ensemble associé à R. Le problème est alors d'évaluer l'erreur quadratique de l'union ou de la différence de deux multi-ensembles. Un premier pas vers l'évaluation de l'erreur quadratique est donné par la proposition suivante :

Proposition 1 *Soient $\{(C_1, f_1), \ldots, (C_p, f_p)\}$ un ensemble de multi-ensembles de $\mathcal{ME}_n$, et $C = \sum_{i=1}^{p} (C_i, f_i)$. On a :*

$$\forall i \in \{0, 1, 2\} \quad \boldsymbol{M_i}(C) = \sum_{j=1}^{p} \boldsymbol{M_i}(C_j)$$

Preuve:

Nous allons établir la propriété pour le cas $p = 2$, le cas général s'en déduisant trivialement par récurrence. Afin de simplifier, on notera dans cette démonstration v^0 la constante 1, v^1 le vecteur v et v^2 le vecteur $(v_1^2, \ldots, v_n^2)$ si v est égal à $(v_1, \ldots, v_n)$. Soient donc (C_1, f_1) et (C_2, f_2) deux multi-ensembles de $\mathcal{ME}_n$ et $C = (C_1, f_1) + (C_2, f_2)$. Nous avons pour tout i dans $\{0, 1, 2\}$:

$$\begin{array}{rcl}\boldsymbol{M_i}(C) & = & \sum_{v \in C_1 \cup C_2} f_1(v) v^i + f_2(v) v^i \\ & = & \sum_{v \in C_1 - C_2} f_1(v) v^i + \sum_{v \in C_2 - C_1} f_2(v) v^i + \sum_{v \in C_1 \cap C_2} f_1(v) v^i + f_2(v) v^i \\ & = & \sum_{v \in C_1 - C_2} f_1(v) v^i + \sum_{v \in C_1 \cap C_2} f_1(v) v^i + \sum_{v \in C_2 - C_1} f_2(v) v^i + \sum_{v \in C_1 \cap C_2} f_2(v) v^i \\ & = & \sum_{v \in C_1} f_1(v) v^i + \sum_{v \in C_2} f_2(v) v^i \\ \boldsymbol{M_i}(C) & = & \boldsymbol{M_i}(C_1) + \boldsymbol{M_i}(C_2)\end{array}$$

$\square$

Une propriété importante de la moyenne se déduit trivialement de l'additivité des moments :

Lemme 1 *Soient* (C_1, f_1) *et* (C_2, f_2) *deux éléments de* $\mathcal{ME}_n$*. On a :*

$$\mu(C_1 + C_2) = \frac{|C_1|\mu(C_1) + |C_2|\mu(C_2)}{|C_1| + |C_2|}.$$

Corollaire 1 *Soient* (C_1, f_1)*,* (C_2, f_2) *et* $(C, f) = (C_1, f_1) + (C_2, f_2)$ *des éléments de* $\mathcal{ME}_n$*. On a :*

$$\mu(C_1) - \mu(C_2) = \frac{|C_1| + |C_2|}{|C_2|}\left(\mu(C_1) - \mu(C)\right) = \frac{|C_1| + |C_2|}{|C_1|}\left(\mu(C) - \mu(C_2)\right).$$

Preuve:

En utilisant le lemme 1 on a :

$$\mu(C_1) - \mu(C_2) = \mu(C_1) - \frac{(|C_1| + |C_2|)\mu(C) - |C_1|\mu(C_1)}{|C_2|} = \frac{|C_1| + |C_2|}{|C_2|}\left(\mu(C_1) - \mu(C)\right).$$

De même :

$$\mu(C_1) - \mu(C_2) = \frac{(|C_1| + |C_2|)\mu(C) - |C_2|\mu(C_2)}{|C_1|} - \mu(C_2) = \frac{|C_1| + |C_2|}{|C_1|}\left(\mu(C) - \mu(C_2)\right)$$

□

L'additivité des moments et le lemme précédent nous permettent de définir un théorème important permettant d'exprimer l'erreur quadratique de la somme de deux multi-ensembles.

Théorème 2 *Soient* (C_1, f_1) *et* (C_2, f_2) *deux multi-ensembles de* $\mathcal{ME}_n$*. L'erreur quadratique du multi-ensemble* $(C, f) = (C_1, f_1) + (C_2, f_2)$ *est donnée par l'équation suivante :*

$$\boldsymbol{SE}(C) = \boldsymbol{SE}(C_1) + \boldsymbol{SE}(C_2) + \frac{|C_1||C_2|}{|C_1| + |C_2|}\|\mu(C_1) - \mu(C_2)\|^2. \tag{5.2}$$

Preuve:

Si nous utilisons l'équation 5.1, l'erreur quadratique du multi-ensemble (C, f) s'exprime de la façon suivante :

$$\begin{aligned} \boldsymbol{SE}(C) &= \sum_{i=1}^{n} \boldsymbol{M_2}(C)_i - \frac{\boldsymbol{M_1}(C)_i^2}{|C|} \\ &= \sum_{i=1}^{n} \boldsymbol{M_2}(C)_i - |C|\mu(C)_i^2. \end{aligned}$$

En utilisant la proposition 1 et le lemme 1 nous obtenons :

$$\begin{aligned} \boldsymbol{SE}(C) &= \sum_{i=1}^{n} \boldsymbol{M_2}(C_1)_i + \boldsymbol{M_2}(C_2)_i - |C|\mu(C_1 + C_2)_i^2 \\ &= \sum_{i=1}^{n} \boldsymbol{M_2}(C_1)_i + \boldsymbol{M_2}(C_2)_i - (|C_1| + |C_2|)\frac{\left(|C_1|\mu(C_1)_i + |C_2|\mu(C_2)\right)^2}{\left(|C_1| + |C_2|\right)^2}. \end{aligned}$$

Si nous rajoutons et enlevons $\frac{1}{|C_1|}\sum_{i=1}^{n} \boldsymbol{M_1}^2(C_1)_i$ et $\frac{1}{|C_2|}\sum_{i=1}^{n} \boldsymbol{M_1}^2(C_2)_i$ à la somme, nous obtenons :

$$\boldsymbol{SE}(C) = \boldsymbol{SE}(C_1)+\boldsymbol{SE}(C_2)+\sum_{i=1}^{n} |C_1|\mu(C_1)_i^2+|C_2|\mu(C_2)_i^2-\frac{(|C_1|\mu(C_1)_i + |C_2|\mu(C_2)_i)^2}{|C_1|+|C_2|}.$$

Le développement du carré présent en fin d'égalité donne :

$$\begin{aligned}
\boldsymbol{SE}(C) &= \boldsymbol{SE}(C_1) + \boldsymbol{SE}(C_2) + \frac{\sum_{i=1}^{n} |C_1||C_2|\mu(C_1)_i^2 + |C_1||C_2|\mu(C_2)_i^2 - 2|C_1||C_2|\mu(C_1)_i\mu(C_2)_i}{|C_1|+|C_2|} \\
&= \boldsymbol{SE}(C_1) + \boldsymbol{SE}(C_2) + \frac{|C_1||C_2|}{|C_1|+|C_2|}\sum_{i=1}^{n} \mu(C_1)_i^2 + \mu(C_2)_i^2 - 2\mu(C_1)_i\mu(C_2)_i \\
&= \boldsymbol{SE}(C_1) + \boldsymbol{SE}(C_2) + \frac{|C_1||C_2|}{|C_1|+|C_2|}\sum_{i=1}^{n} (\mu(C_1)_i - \mu(C_2)_i)^2 \\
\boldsymbol{SE}(C) &= \boldsymbol{SE}(C_1) + \boldsymbol{SE}(C_2) + \frac{|C_1||C_2|}{|C_1|+|C_2|}\|\mu(C_1) - \mu(C_2)\|^2
\end{aligned}$$

□

Comme nous l'avons vu, l'erreur quadratique $\boldsymbol{SE}(C)$ mesure l'homogénéité de l'ensemble C. Afin de mesurer l'homogénéité d'une partition, $\{C_1, \ldots, C_n\}$, on définit **l'erreur quadratique d'une partition** de la façon suivante :

Définition 6 *Soient $(C_1, f), \ldots, (C_p, f)$ des éléments de $\mathcal{ME}_n$ tels que :*

$$\begin{array}{l} C = \bigsqcup_{i=1}^{p} C_i \\ \forall (i,j) \in \{1, \ldots, p\}^2 \quad i \neq j \Longrightarrow C_i \cap C_j = \emptyset \end{array}.$$

On définit alors **l'erreur quadratique de la partition** *notée $\boldsymbol{E}(C)$ par :*

$$\boldsymbol{E}(C) = \sum_{i=1}^{p} \boldsymbol{SE}(C_i).$$

La partition de C en sous-ensembles minimisant $\boldsymbol{E}(C)$ est un problème $\mathcal{NP}$ complet pour des espaces de dimension supérieure à 1 [WZ91]. Dans le cas mono-dimensionnel, donc pour des images en niveaux de gris, Wong, Wan et Prusinkiewicz ont élaboré un algorithme permettant de trouver la partition optimale avec une complexité en $O(NlogN)$ [WWP89].

Le découpage d'un multi-ensemble s'effectue souvent en considérant un de ses sous-ensembles et en faisant croître celui-ci jusqu'à obtenir une partition qui induit une erreur quadratique minimale. Du fait de la complexité du problème on ne choisit généralement pas la partition qui minimise globalement l'erreur quadratique, mais plus simplement la partition induisant une erreur quadratique minimale sur un ensemble de partitions envisagées. La partition la plus couramment utilisée est la *partition par plans*. Le multi-ensemble est alors découpé en deux sous-ensembles de par et d'autre du plan. Le nombre de plans susceptibles de couper un multi-ensemble étant encore trop grand, on restreint les possibilités en fixant la normale à celui-ci. L'ensemble des partitionnements possibles est alors égal au nombre de plans parallèles susceptibles de couper le multi-ensemble. Ce type de découpe utilise **la projection** d'un multi-ensemble définie de la façon suivante :

Définition 7 *Soit* (C, f) *un élément de* $\mathcal{ME}_n$. *La* **projection** *de* (C, f) *sur l'axe vectoriel défini par le vecteur* $\vec{n} \in \mathbb{R}^n$ *est égale à* $([m, M], F) \in \mathcal{ME}_1$ *avec :*

$$\begin{aligned} m &= \min_{v \in C} v.\vec{n} \\ M &= \max_{v \in C} v.\vec{n} \\ D_t &= \{v \in C / v.\vec{n} = t\} \\ F(t) &= \sum_{v \in D_t} f(v). \end{aligned}$$

La découpe du multi-ensemble 1D $([m, M], F)$ *au point* $t \in [m, M]$ *induit un partitionnement de* (C, f) *en* (C_t, f) *et* $(C, f) - (C_t, f)$ *avec :*

$$C_t = \{v \in C \ / \ v.\vec{n} \leq t\}$$

où $v.\vec{n}$ désigne le produit scalaire des vecteurs v et $\vec{n}$. La variable t joue dans ce cadre le rôle d'une abscisse le long de la direction $\vec{n}$.

5.2.2 Structures de données

Dans le cas d'une image en niveau de gris, l'histogramme de l'image peut aisément être stocké avec un tableau d'entiers de taille 256. Or, comme nous l'avons vu, dans le cadre de l'imagerie couleur, un pixel peut prendre $256^3 \approx 16.10^6$ valeurs différentes. Cette taille beaucoup plus importante de l'histogramme nécessite quelques précautions. De plus, une image de taille 512×512 ne comportant au plus que $4.256^2 << 16.10^6$ couleurs, de nombreuses cases de l'histogramme codant le nombre de pixels affecté à une couleur seront nécessairement égales à 0. La résolution de ces deux problèmes à susciter de nombreuses structures de données dont nous allons donner un aperçu.

L'Histogramme de Thomas

L'utilisation d'un énorme tableau essentiellement composé de 0 étant peu commode, beaucoup d'auteurs comme Thomas [Tho91] effectuent une première quantification en utilisant uniquement les 5 bits de poids fort pour coder chaque composante R, G ou B. L'histogramme obtenu de taille $32 \times 32 \times 32$ nécessite beaucoup moins de mémoire et, du fait de la préquantification, est beaucoup plus plein que le tableau initial de taille 256^3. Cette structure de données, permet donc d'obtenir un stockage de l'histogramme grâce à une structure de données

```
histogramme faire_histogram(image I)
{
    int hist[32][32][32]

    // Initialiser hist a 0

    Pour chaque pixel P de l'image
    {
        R_quant = I(P).R/8 ;
        G_quant = I(P).G/8 ;
        B_quant = I(P).B/8 ;
        hist[R_quant][G_quant][B_quant]++ ;
    }

    retourner hist ;
}
```

Algorithm 1: *Initialisation de l'histogramme de Thomas*

peu couteuse en mémoire et permet d'accéder rapidement au nombre de pixels d'une couleur donnée. De plus, cette structure permet d'obtenir un histogramme plus compact et donc plus manipulable que l'histogramme brut formé par le tableau de 256^3 entiers. Notons toutefois deux inconvénients majeurs de cette structure :

1. L'utilisation de la pré-quantification implique une perte d'informations par rapport à l'histogramme initial.
2. La préquantification ne garantit en aucune façon que la matrice $32 \times 32 \times 32$ ne comportera pas de nombreuses cases nulles. Elle ne fait qu'atténuer ce risque.

L'Histogramme de Xiang

L'approche de Zhigang Xiang [XJ94, Xia97], consiste à stocker le nombres d'occurrences de chaque couleur de l'image à l'aide d'un tableau *hist* de taille 256×256 composé de listes (voir Figure 5.4).

– Chaque liste, $hist(R, G)$, code les différentes valeurs de bleu prises par les pixels de l'image dont les composantes rouges et vertes sont égales à R et G.
– Chaque élément de la liste $hist(R, G)$ code une composante bleue B prise par un pixel de l'image et le nombre de pixels de couleur (R, G, B) dans l'image. Chaque liste est triée par ordre croissant sur la composante bleue.

Cette structure de données permet un codage plus compact que celle de Thomas (voir section 5.2.2). De fait, seules les couleurs présentes dans l'image apparaissent dans la structure de données. Un élément du tableau $hist(R, G)$ égal à une liste vide correspond à des composantes (R, G) absentes de l'image. Le

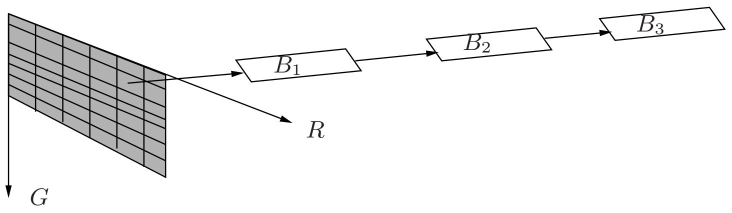


FIG. 5.4 – *Tableau de listes utilisées pour stocker l'histogramme 3D*

calcul du nombre de pixels d'une couleur (R, G, B) donnée implique toutefois un parcourt de liste et est donc légèrement plus coûteux que son équivalent effectué avec la structure de données de Thomas.

L'Histogramme de Balasubramanian

L'histogramme utilisé par Balasubramanian [BA91] est très proche de celui de Xiang (Section 5.2.2). Balasubramanian utilise également un tableau de 256×256 pour stocker les occurences correspondant à un couple (R, G) mais stocke les valeurs de B dans un arbre binaire plutot que dans une liste. Cette structure est un peu plus lourde qu'une liste simplement chainée mais lui permet de retrouver une valeur de B en un temps logarithmique (par rapport au nombre de valeurs stockées dans l'arbre).

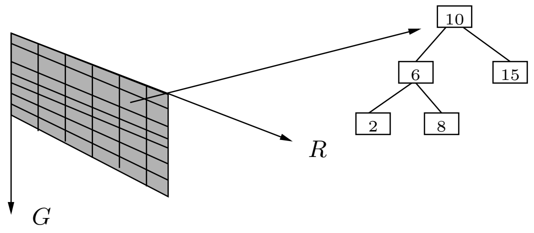


FIG. 5.5 – *Tableau d'arbres binaires utilisées pour stocker l'histogramme 3D*

Histogramme pour la quantification descendante

Les méthodes de quantification descendantes n'ont généralement pas besoin de retrouver rapidement le nombre de pixels correspondant à un triplet (R, G, B) donné. Ces méthodes se contentent donc de stocker un tableau contenant l'ensemble des couleurs de l'image avec leurs fréqence (voir Figure 5.6). Ce tableau sera ensuite découpé en sous tableaux au cours de la quantification.

```
histogramme faire_histogram(image I)
{
    liste * hist[256][256]

    // Initialiser hist a NULL

    Pour chaque pixel P de l'image
    {
        Si(hist[I(P).R][I(P).G] == NULL)
            hist[I(P).R][I(P).G]= creer_liste() ;

        Si( I(P).B abscent de hist[I(P).R][I(P).G])
        {
            Noeud NB=creer_noeud() ;
            NB.val=I(P).B ;
            NB.compeur=1 ;
            inserer_dans_liste(hist[I(P).R][I(P).G],NB) ;
        }
        sinon
        {
            Noeud NB=donner_noeud(hist[I(P).R][I(P).G],I(P).B) ;
            NB.compeur++ ;
        }

    }
    retourner hist ;
}
```

Algorithm 2: *Initialisation de l'histogramme de Xiang*

L'Histogramme de Brun et Braquelaire

Le but de cette structure de données est de permettre un accès rapide au nombre de pixels dans l'image possédant une composant de valeur donnée, i.e. vérifiant l'équation 5.3 :

$$c_i = \alpha \quad \text{et} \begin{cases} m_j \leq c_j < M_j \\ m_k \leq c_k < M_k \end{cases} \quad \text{avec } i, j, k \in \{1, 2, 3\} \tag{5.3}$$

où :
- $[m_i, M_i[\times[m_j, M_j[\times[m_k, M_k[$ définit les interalles de variations des couleurs de l'image.
- $\alpha \in [m_i, M_i[$

Résoudre (5.3) consiste à trouver le nombre de pixels de coordonnée α sur l'axe i dans l'ensemble C des couleurs de l'image.

On sent intuitivement assez bien que des données triées résoudraient le problème. Partant de cette idée, Brun et Braquelaire reprennent la structure de

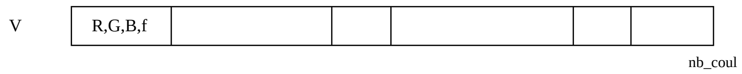


FIG. 5.6 – *Histogramme utilisé par les méthodes de quantification descendantes*

données utilisée par les méthodes de quantification descendantes en y adjoignant des tableaux d'index tab_{c_1}, tab_{c_2} et tab_{c_3} triés suivant chacune des composantes. Chacun de ces vecteurs est lui même indexé par un autre tableau permettant d'accéder rapidement au nombre de couleurs dans l'image dont la projection selon un des trois axes est supérieure à une valeur donnée (voir Figure 5.7).

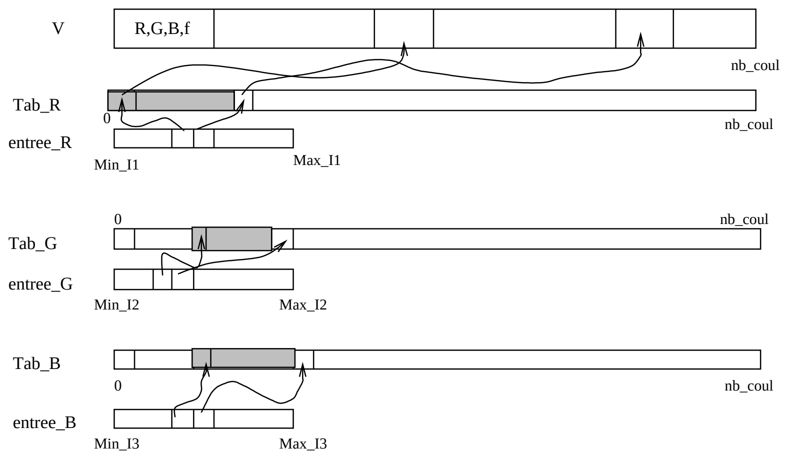


FIG. 5.7 – *Tableaux utilisés pour stocker l'histogramme par Braquelaire et Brun*

Plus précisément, on utilise les constructions suivantes :
- Un tableau V contenant l'ensemble des triplets de couleurs contenus dans l'image et pour chaque triplet le nombre de pixels correspondant à ce triplet.
- Trois vecteurs tab_c_1, tab_c_2, tab_c_3 triant les valeurs de V suivant c_1, c_2 et c_3
  - les valeurs constantes de c_1 dans tab_c_1 sont triées suivant c_2,
  - les valeurs constantes de c_2 dans tab_c_2 sont triées suivant c_1,
  - les valeurs constantes de c_3 dans tab_c_3 sont triées suivant c_1.

  Les plages grisées sur la Figure 5.7 visualisent des zones référençant des triplets ayant une même valeur de c_1 dans le premier cas, de c_2 dans le second et de c_3 dans le troisième.
- Trois vecteurs entrée_c_1, entrée_c_2 et entrée_c_3 référençant les tableaux tab_c_1, tab_c_2,tab_c_3 et vérifiant la propriété suivante :

$$\forall j \in \{1,2,3\} \quad \forall k \in [Min_j, Max_j] \quad \forall\, l \geq entre\_c_j[k] \quad V[tab\_c_j[l]].c_j \geq k.$$

Ils permettent donc d'accéder directement à l'ensemble des couleurs dont la composante c_j choisie est supérieure à k.

La résolution de l'équation 5.3 s'effectue alors en trois temps (posons pour fixer les idées i=1, j=2 et k=3)

1. Trouver l'ensemble des couleurs vérifiant $(c_1 = \alpha)$ à l'aide de entrée_c_1. Cet ensemble est codé de façon continue dans tab_c_1.
2. Chercher par dichotomie dans la partie de tab_c_1 définie précédemment les indices vérifiant $m_2 \leq V[j].c_2 < M_2$. Ces indices sont stockés de façon continue dans tab_c_1.
3. Parcourir l'ensemble des indices trouvés dans (2) pour ne retenir que ceux vérifiant $m_3 \leq V[j].c_3 < M_3$.

Cette structure de données permet donc :

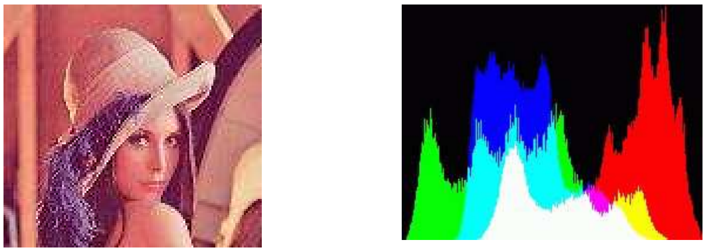
– de stocker uniquement les couleurs effectivement présentes dans l'image,

– de supprimer l'étape de préquantification.

Notons toutefois que cette structure de données relativement complexe est assez peu adaptée à tout autre requête que celle exprimée par l'équation 5.3. Son utilisation reste donc assez limitée.

Histogrammes marginaux

L'utilisation d'un histogramme 3D peut s'avérer inutilement complexe pour certains types de traitements voulant agir séparément sur chacune des composantes. Dans ce cas, trois histogrammes représentants les histogrammes marginaux de chacune des composantes sont souvent suffisants (Voir Figure 5.8). Ces histogrammes peuvent être calculés par l'Algorithme 3



(a) (b)

FIG. 5.8 – *Lenna (a) et ses trois histogrammes marginaux sur imprimés (b)*

5.3 Opérations ponctuelles

Les opérations ponctuelles sont les opérations travaillant uniquement sur la couleur de chaque pixel, sans tenir compte ni de la position du pixel dans

```
histogramme faire_histogram(image I)
{
    int hist_R[256],hist_G[256],hist_B[256]

    // Initialiser hist_R, hist_G et hist_B a 0

    Pour chaque pixel P de l'image
    {
        hist_R(I(P).R)++;
        hist_G(I(P).G)++;
        hist_B(I(P).B)++;
    }
    retourner hist_R, hist_G et hist_B
}
```

Algorithm 3: *Calcul des histogrammes marginaux*

l'image ni du voisinage de celui-ci. Les méthodes basées sur les manipulations d'histogrammes forment une grande partie de ces méthodes.

5.4 Filtrages linéaires

5.5 Filtrages non linéaires

5.6 Réhaussement

Chapitre 6

Traitements d'images couleur

6.1 Opérateurs différentiels

Ces méthodes considèrent généralement l'image comme la discrétisation d'un signal continu 2D. On a donc :

$$I = Q \circ f$$

où I est l'image discrète, f le signal continu, et Q un opérateur d'échantillonnage. La recherche de contours dans I (donc de sauts de valeurs) se caractérise dans f par des maxima de la dérivé. On introduit donc l'opérateur différentiel D. Nous verrons dans la section 6.1.5 comment l'information couleur peut être traitée par ce type d'approche. La plupart des méthodes de détection de contours étant plus spécifiquement dédiées au traitement d'images mono-dimensionnelles, nous allons dans cette section nous limiter à ce cas. Nous supposerons de plus que f appartient à $C^2(\mathbb{R}^2, \mathbb{R})$, l'ensemble des fonctions continuement deux fois différentiables de $\mathbb{R}^2$dans $\mathbb{R}$. L'ensemble des fonctions susceptibles de donner une même discrétisation I étant important, cette dernière condition n'est pas très restrictive.

La différentielle de f peut donc s'exprimer de la façon suivante :

$$Df(p).n = \frac{\partial f}{\partial x}(p).n_x + \frac{\partial f}{\partial y}(p).n_y = \lim_{h \to 0} \frac{f(p) - f(p + (hn_x, hn_y))}{h} \tag{6.1}$$

$Df(p).n$ est donc la dérivée de f dans la direction n (comme seule la direction de n nous intéresse, on peut prendre $\|n\| = 1$). Si nous introduisons l'opérateur gradient défini par :

$$\bigtriangledown f = \begin{pmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{pmatrix}$$

l'équation 6.1 s'écrit :

$$Df(p).n = \bigtriangledown f(p) \bullet n \tag{6.2}$$

où $\bullet$ désigne le produit scalaire.

On a, de plus, par l'équation 6.2 :

$$\max_{\|n\|=1} |Df(p).n| = \max_{\|n\|=1} |\bigtriangledown f(p) \bullet n| = \|\bigtriangledown f(p)\|.$$

Donc, en tout point la norme du gradient permet de connaître la variation maximum de la différentielle. De plus, le maximum étant atteint pour n colinéaire à $\bigtriangledown f(p)$, la direction du gradient donne la direction de plus grande variation de la fonction f.

Les directions de variation de f correspondant à des maxima de la différentielle de f, on peut également chercher ces directions par les zéros de la différentielle seconde de f. Si $D_2 f$ désigne cette différentielle seconde, on a :

$$D_2 f(p).n = \tfrac{\partial^2 f}{\partial x^2}(p).n_x.n_x + \tfrac{\partial^2 f}{\partial y^2}(p).n_y.n_y + 2\tfrac{\partial^2 f}{\partial x \partial y}(p).n_x n_y \tag{6.3}$$

La fonction f présentera une variation locale suivant une direction n, si à n fixé la fonction $Df(p).n$ présente un maximum local. Ce maximum local se traduira, toujours à n fixé, par un zéro de la fonction $D_2 f(p).n$ en ce point.

Afin d'alléger les calculs, on cherche les zéros du laplacien plutôt que les zéros de la différentielle seconde. Le laplacien de la fonction f, noté $\triangle f$, est défini par :

$$\triangle f(p) = \tfrac{\partial^2 f}{\partial x^2}(p) + \tfrac{\partial^2 f}{\partial y^2}(p)$$

Le laplacien étant invariant par rotation, le calcul des zéros du laplacien ne fait pas intervenir la direction n de plus grande variation de $Df(p)$. L'utilisation du laplacien évite donc de calculer le gradient en plus de la différentielle seconde.

Les zéros du laplacien ne correspondent généralement pas aux zéros de $D_2 f(p).n$, toutefois Marr [MH80] a montré qu'il y avait coïncidence entre les zéros des deux fonctions au point p si les variations d'intensité étaient linéaires sur la ligne de passage par zéro et sur les lignes parallèles dans un voisinage de p. Le laplacien peut donc être vu comme une bonne approximation de la différentielle seconde.

6.1.1 Les opérateurs de convolution

Dans les cas usuels, la fonction f comporte souvent du bruit. Ce bruit peut parasiter la détection des contours de f en créant des sauts de valeurs artificiels. Un moyen simple de diminuer le bruit est de convoluer la fonction f avec un filtre passe-bas. La convolution de deux fonctions f et g de $\mathbb{R}^2$dans $\mathbb{R}^2$définit la fonction $f * g$:

$$f*g(x,y) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(u,v)g(x-u,y-v)dt = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(x-u,x-v)g(u,v)dt$$

Le filtre passe-bas le plus connu est sans doute la gaussienne notée G et définie par :

$$G(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}}$$

Le paramètre σ contrôle l'aplatissement de la gaussienne et permet donc de contrôler la puissance du filtre. Une forte valeur de σ induit un fort lissage et vice-versa. La convolution d'un signal avec la gaussienne s'effectue en limitant celle-ci à un support fini $[-M_\epsilon, M_\epsilon]$ avec :

$$\forall x \in [-M_\epsilon, M_\epsilon] \quad G(x) < \epsilon.$$

On peut remarquer que la valeur M_ϵ est une fonction croissante de σ, donc plus l'on voudra lisser le signal plus l'on devra agrandir le support $[-M_\epsilon, M_\epsilon]$ de la gaussienne.

Une des principales raisons de la popularité de la gaussienne est sa **séparabilité**. Si $G(x, y)$ représente une gaussiennne 2D on a :

$$\begin{array}{rcl} G(x,y) & = & \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{x^2+y^2}{2\sigma^2}} \\ & = & \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}} e^{-\frac{y^2}{2\sigma^2}} \\ G(x,y) & = & \frac{1}{\sigma\sqrt{2\pi}} g(x)g(y). \end{array}$$

Donc si nous convoluons une fonction 2D f avec G, on a :

$$\begin{array}{rcl} f * G(x,y) & = & \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x-u, y-v) G(u,v) du dv \\ & = & \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x-u, y-v) g(u) g(v) du dv \\ & = & \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{+\infty} g(u) \left(\int_{-\infty}^{+\infty} f(x-u, y-v) g(v) dv \right) du \\ & = & \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{+\infty} g(u) f *_y g(x-u, y) du \\ f * G(x,y) & = & \frac{1}{\sigma\sqrt{2\pi}} g *_x f *_y g(x,y) \end{array}$$

où $*_x$ et $*_y$ représentent les convolutions par rapport aux variables x et y.

La convolution de f avec la gaussienne 2D G peut donc être réalisée grâce à deux convolutions de signaux 1D. Ceci permet d'utiliser deux convolutions avec des masques 1D $[-M_\epsilon, M_\epsilon]$ plutôt qu'avec un masque 2D de taille $[-M_\epsilon, M_\epsilon]^2$. Convoluer une image comportant m pixels avec une gaussienne dont la taille du masque est égale à n réclame donc $\mathcal{O}(2mn)$ opérations, l'utilisation d'un masque 2D réclamant quant à elle $\mathcal{O}(mn^2)$ opérations.

Une des propriétés essentielles du produit de convolution, en segmentation, est la possibilité de placer alternativement la dérivée sur un membre ou sur l'autre. Plus précisément on a :

Propriété 1 *Soient f et g deux fonctions de $\mathbb{R}$ dans $\mathbb{R}$; si f ou g est k fois différentiable alors $f * g$ l'est également. De plus, si f et g sont toutes deux au moins k fois différentiables on a :*

$$(f * g)^{(k)} = (f^{(k)}) * g = f * (g^{(k)}).$$

Donc si $G * f$ représente l'image filtrée par une gaussienne, le gradient de l'image filtrée peut tout simplement s'effectuer en calculant la convolution de f avec le gradient de G, $\bigtriangledown G$, qui peut être précalculé une fois pour toute. De même, le filtrage d'une image par une gaussienne G, puis le calcul de son laplacien peuvent s'effectuer grâce à une seule convolution $\triangle G * f$.

6.1.2 Passage au discret

La transposition de ces méthodes au modèle discret se fait par l'approximation des différentielles partielles de la fonction f. Si l'on utilise les différences finies l'on obtient par exemple :

$$\frac{\partial f}{\partial x}(i,j) \approx \frac{\Delta I}{\Delta i}(i,j) = I(i+1,j) - I(i,j)$$
$$\tfrac{\partial^2 f}{\partial x^2}(i,j) \approx 2I(i,j) - I(i+1,j) - I(i-1,j)$$

Des méthodes plus évoluées [Pre70, Kir71] approximent le gradient par une convolution avec un masque $3 \star 3$. Dans le cas discret et pour un masque M à support fini, la convolution de M avec I est définie par :

$$M * I(i,j) = \sum_{k=-p}^{p} \sum_{l=-q}^{q} M[k][l]I[i-k][j-l]$$

Marr et al. [MH80] et Huertas et al. [HM86] approximent les zéros de la différentielle seconde en convoluant l'image avec le laplacien de la gaussienne. Une approche légèrement différente introduite par Cocquerez [CD85] consiste à utiliser un filtre passe-bas non linéaire [NM79] suivi d'une dérivation.

6.1.3 Les opérateurs optimaux

Canny [Can86] modélise un contour C comme la superposition d'un saut d'amplitude S et d'un bruit blanc B.

$$C(x) = S_0 S(x) + B(x).$$

La fonction S est modélisée par la fonction $\chi_{[0,+\infty)}$ où χ est la fonction caractéristique.

Suivant ce formalisme, le détecteur de contour idéal h est celui qui, convolué à C, présente un maximum en 0. Cette contrainte n'étant pas suffisante pour déterminer h, Canny impose trois conditions supplémentaires à son détecteur de contour :

– Une bonne détection
– Une bonne localisation
– Une faible multiplicité des maxima dûs au bruit.

La formalisation de ces trois conditions impose à la fonction h de respecter l'équation différentielle suivante :

$$2h(x) - 2\lambda_1 h''(x) + 2\lambda_2 h^{(3)}(x) + \lambda_3 = 0.$$

Les conditions aux limites imposées à la fonction h permettent de fixer les paramètres $\lambda_1, \lambda_2, \lambda_3$. Les conditions imposées par Canny permettent d'obtenir une fonction à support fini $[-M, M]$:

$$h(0) = 0 \;\;;\;\; h(M) = 0 \;\;;\;\; h'(0) = S \;\;;\;\; h'(M) = 0.$$

Malheureusement, ces conditions donnent une fonction h très coûteuse à implémenter. Partant de ce constat, Deriche [Der87] a défini d'autres conditions aux limites donnant à la fonction h un support infini :

$$h(0) = 0 \;\;;\;\; h(+\infty) = 0 \;\;;\;\; h'(0) = S \;\;;\;\; h'(+\infty) = 0.$$

Ces conditions initiales donnent la fonction :

$$h \left(\begin{array}{ccc} \mathbb{R} & \rightarrow & \mathbb{R} \\ x & \mapsto & ce^{-\alpha|x|} sin\omega x \end{array} \right. .$$

L'analyse des critères donnés par Canny [Can86] montre que les meilleures performances du filtre sont obtenues pour ω tendant vers 0. On obtient à ce moment là $h(x) = Cxe^{-\alpha|x|}$ avec $C = \omega c$. Le paramètre C est un paramètre de normalisation, le paramètre α contrôle quant à lui la sensibilité de l'opérateur de détection de contour. Une augmentation de α favorise la détection au détriment de la localisation et vice-versa. Le paramètre α joue ici exactement le même rôle que le paramètre σ dans la définition du gradient de la gaussienne.

Deriche construit à partir de h une fonction de lissage 1D l définie par :

$$l(x) = \int_0^x h(x)dx = b(\alpha|x| + 1)e^{-\alpha|x|}.$$

On définit alors la fonction de lissage 2D $L(x, y)$ par :

$$L(x, y) = l(x)l(y).$$

La fonction L joue ici le rôle de la gaussienne, et le filtrage de l'image s'obtient en convoluant celle-ci avec L. Une fois l'image lissée, on peut calculer son gradient ou son laplacien. Ces deux opérations s'effectuent en convoluant l'image avec le gradient ou le laplacien de L.

L'avantage des opérateurs de Deriche par rapport à ceux de Canny vient du fait que les techniques de la transformée en z [Der87], appliquées à la fonction L et à ses dérivées, permettent de calculer de façon récursive la convolution des opérateurs de Deriche avec l'image. Si, par exemple, nous effectuons la convolution d'un signal mono-dimensionnel $x(m)$ avec la fonction échantillonnée $l(m)$, le signal final $y(m)$ se déduit des équations suivantes [DC87] :

$$\begin{array}{lcll} y^+(m) & = & a_0x(m) + a_1x(m-1) - b_1y^+(m-1) - b_2y^+(m-2) & m = 1, \ldots, M \\ y^-(m) & = & a_2x(m+1) + a_3x(m+2) - b_1y^-(m+1) - b_2y^-(m+2) & m = M, \ldots, 1 \\ y(m) & = & y^+(m) + y^-(m) & m = 1, \ldots, M \end{array}$$

où M représente la taille du signal et où les coefficients a_i et b_i se déduisent du paramètre α.

L'application de l'algorithme décrit par les équations précédentes nécessite 8 opérations par point quelle que soit la valeur de α. Les filtres de Deriche présentent donc les avantages suivants :

- Ils s'appuient sur une étude théorique permettant de juger et comparer différents détecteurs de contours
- Leur expression, plus simple que la gaussienne, permet, grâce à la transformée en z, d'obtenir une définition récursive de la convolution. Cette récursivité permet d'effectuer l'opération de convolution en un nombre fixe d'opérations par point de l'image, indépendamment du paramètre α.

Shen et Castan [SC92, CZS89] ont adopté une démarche similaire à celle de Canny [Can86]. Les différences entre les deux méthodes portent sur la formalisation des critères décrivant un détecteur de contour optimal. La fonction obtenue par Shen est égale à $\beta e^{-\beta|x|}$. Cette fonction permet de définir des opérateurs de lissage, de gradient et de laplacien et d'effectuer une transformée en z [Mon90] permettant une implémentation récursive.

6.1.4 Mise en oeuvre

Les recherches des points de contours à partir du gradient, des zéros de la dérivée seconde ou des zéros du laplacien sont très similaires. La démarche générale de ces trois méthodes peut se décomposer comme suit [CP95] :

La recherche à partir du gradient

- Calculer le gradient en chaque point de l'image
- Créer l'image de la norme du gradient
- Extraire les maxima locaux dans la direction du gradient.
- Effectuer un seuillage à effet hystérésis (voir ci-dessous) de l'image des maxima locaux.

La recherche à partir de la différentielle seconde

- Calculer $D_2f(p).n$ pour tout point p de l'image (n représente la direction du gradient, donc la direction de plus grande variation de la différentielle première)
- Rechercher les passages par zéros de $D_2f(p).n$ dans la direction n
- Créer l'image des passages par zéros et de la norme du gradient
- Effectuer un seuillage à effet hystérésis de l'image des maxima locaux.

La recherche à partir du laplacien

- Calculer le laplacien

- Rechercher les passages par zéros du laplacien
- Créer l'image des passages par zéros et de la norme du gradient
- Effectuer un seuillage à effet hystérésis de l'image des maxima locaux.

Le seuillage à effet hystérésis consiste à ne conserver que :

- Les points dont la norme du gradient est supérieure à un seuil haut (sh).
- Les points dont la norme du gradient est supérieure à un seuil bas (sb avec $sb < sh$) et appartenant à un bout de contour dont au moins un point possède une norme du gradient supérieure à sh.

Ce type de seuillage permet de diminuer le nombre de bouts de contour non fermés.

6.1.5 La couleur et les algorithmes de détection de contours

Nous avons vu dans la section 6.1 qu'une méthode classique de segmentation d'image consiste à considérer une image en niveau de gris comme la discrétisation d'une fonction f de $\mathbb{R}^2$ dans $\mathbb{R}$. À ce moment là les contours de l'image sont définis comme les maxima de la différentielle première Df ou les zéros de la différentielle seconde D_2f.

Si nous appliquons la même démarche à l'analyse d'images couleurs, une image I est considérée comme la discrétisation d'une fonction f de $\mathbb{R}^2$ dans $\mathbb{R}^3$. Nous avons alors :

$$f \left(\begin{array}{ccc} \mathbb{R}^2 & \rightarrow & \mathbb{R}^3 \\ (x,y) & \mapsto & (f_1(x,y), f_2(x,y), f_3(x,y)) \end{array} \right.$$

La différentielle première de f en p dans la direction n s'écrit alors :

$$Df(p).(n_x, n_y) = \begin{pmatrix} \frac{\partial f_1}{\partial x}(p).nx + \frac{\partial f_1}{\partial y}(p).n_y \\ \frac{\partial f_2}{\partial x}(p).nx + \frac{\partial f_2}{\partial y}(p).n_y \\ \frac{\partial f_3}{\partial x}(p).nx + \frac{\partial f_3}{\partial y}(p).n_y \end{pmatrix}$$

Comme dans le cas mono-dimensionnel, nous recherchons les fortes valeurs de la différentielle première. Cette différentielle étant cette fois-ci un vecteur de $\mathbb{R}^3$, nous calculons le carré de sa norme :

$$S(p,n) = \|Df(p).(n_x, n_y)\|^2 = E{n_x}^2 + 2Fn_xn_y + G{n_y}^2 \tag{6.4}$$

avec :

$$E = \sum_{i=1}^{3} \left(\frac{\partial f_i}{\partial x}\right)^2 \quad F = \sum_{i=1}^{3} \frac{\partial f_i}{\partial x}\frac{\partial f_i}{\partial y} \quad G = \sum_{i=1}^{3} \left(\frac{\partial f_i}{\partial y}\right)^2.$$

L'équation 6.4 nous donne la norme de la différentielle première dans la direction n. Cette mesure nous indique s'il existe ou non un contour au point p dans la direction n. L'existence d'un contour en p indépendamment de n s'évalue en cherchant le maximum de l'équation 6.4 pour n appartenant à la boule unité (seule la direction de n nous intéresse). Si nous posons $n = (cos(\theta), sin(\theta))$ les extrema de l'équation 6.4 sont atteints pour :

$$\theta_0 = \frac{1}{2} arctan\left(\frac{2F}{E-G}\right).$$

Le maximum correspondant est égal à :

$$\lambda(x,y) = \frac{E+G+\sqrt{(E-G)^2+4F^2}}{2} = S(p,(cos(\theta_0), sin(\theta_0))).$$

La valeur λ peut être vue comme l'extension du concept de gradient aux images couleurs. En effet, on constate facilement que dans le cas mono-dimensionnel, λ est égal au carré de la norme du gradient. Il est donc naturel de considérer les passages par zéro de la différentielle de la fonctions S, au point p, dans la direction n. Cette dernière valeur est donnée par l'expression suivante :

$$D_S(p).n = E_x(p){n_x}^3 + (2F_x(p)+E_y(p))n_x^2 n_y + (G_x(p)+2F_y(p))n_x n_y^2 + G_y(p)n_y^3$$

où E_x, E_y,F_x, F_y et G_x, G_y représentent les dérivées partielles de E, F et G par rapport à x et y.

La valeur $D_S(p).n$ représente l'extension de l'opérateur laplacien aux images multi-dimensionnelles.

L'ensemble des techniques que nous venons de présenter semblent avoir été découvertes en parallèle par Zenzo [Zen86] et Cumani [Cum89, Cum91, CGG91]. Zenzo approxime l'image en chaque pixel par trois fonctions linéaires (une pour chaque composante). Ces approximations lui permettent de calculer les dérivées partielles et d'en déduire la valeur de λ et l'angle pour lequel le maximum est réalisé. Chapron [Cha92] utilise les filtres de Deriche [Der87] pour calculer les dérivées partielles de l'image. Il combine ensuite ces dérivées pour obtenir la fonction λ. Cumani [Cum89, Cum91, CGG91] donne un ensemble d'outils théoriques permettant d'exploiter les passages par zéros de la fonction $D_S(p)$.

6.1.6 Conclusion

Les méthodes de détection de contours permettent d'obtenir très rapidement (de l'ordre de la seconde) un ensemble de contours qui serviront de base à des algorithmes de fermeture de contours. En raison de leur caractère local, ces méthodes doivent diminuer le bruit en appliquant un filtre passe-bas comme la gaussienne ou les filtres de Deriche [Der87] et Shen [SC92]. L'utilisation de ces filtres pose le problème du réglage des paramètres. Si l'on filtre trop, on risque de "rater" des bouts de contours importants alors que si on ne filtre pas assez, l'on risque de se retrouver avec une quantité de contours non significatifs. De plus un même contour peut avoir deux localisations légèrement différentes pour des filtrages voisins. Un début de solution à ce dernier problème a été apporté par Witkin [Wit84] dans le cas de signaux mono-dimensionnels mais à notre connaissance, il n'a pas pu être étendu aux images 2D. Williams [WS90] a toutefois utilisé une partie des travaux de Witkin pour calculer les déplacements d'un contour de type "escalier" en fonction du filtrage. Cette étude lui fournit des informations supplémentaires utilisées lors de l'étape de fermeture de contours.

De plus, le caractère local de ces méthodes rend difficile l'insertion du concept de régions. Il est par exemple délicat d'insérer dans des algorithmes de détection de contours des règles telles que :

– Cette région homogène contient des détails significatifs, il faut donc baisser le paramètre de lissage à l'intérieur de celle-ci.
– On est en train de créer la frontière entre deux régions de couleurs moyennes données. Ce pixel possédant une couleur éloignée de ces moyennes ne peut correspondre qu'à du bruit.

6.2 Segmentation d'une image en matériaux

La *segmentation* d'une image en matériaux consiste à créer une partition de l'image en *régions* telle que chaque région soit composée d'un seul matériau. La segmentation d'une image est usuellement définie comme la partition de celle-ci en régions chaque région codant un objet de la scène. Une définition formelle d'un algorithme de segmentation a été donné par Horowitz et Pavlidis [HP76, HP75] en 1975.

Définition 8 *Soit X le domaine de l'image et f la fonction qui associe à chaque pixel une valeur $f(x, y)$. Si nous définissons un prédicat P sur l'ensemble des parties de X, la segmentation de X est définie comme une partition de X en n sous-ensembles $\{R_1, ..., R_n\}$ tels que :*

1. $X = \bigsqcup_{i=1}^{m} R_i$
2. $\forall i \in \{1, \ldots, n\}$ R_i *est connexe.*
3. $\forall i \in \{1, \ldots, n\} P(R_i) = vrai$
4. $\forall i, j \in \{1, \ldots, n\}^2$ / R_i *est adjacent à* R_j *et* $i \neq j \Rightarrow P(R_i \cup R_j) = faux$

où $\bigsqcup$ représente une union d'ensemble disjoints.

Le prédicat P est utilisé pour tester l'homogénéité des ensembles R_i. Ces sous-ensembles constituent les régions de l'image. Une segmentation de l'image est donc sa décomposition en un ensemble de régions homogènes, le critère d'homogénéité P restant à déterminer. Zucker [Zuc76] a résumé les conditions de la définition 8 comme suit : la première condition implique que tout pixel de l'image appartienne à une région et une seule. Cela signifie que l'algorithme de segmentation ne doit pas se terminer avant d'avoir traité tous les points. La seconde condition implique que toute région doit être connexe. La connexité des régions étant induite par le voisinage défini sur l'image. La troisième condition implique que chaque région doit être homogène. Enfin, la quatrième condition est une condition de maximalité indiquant que la fusion de deux régions ne doit pas être homogène. Il est important de remarquer que le nombre n de régions formant la partition de l'image reste indéterminé. Il peut donc exister plusieurs segmentations possibles pour un prédicat P donné.

Intuitivement un “bon” prédicat P doit nous fournir une région pour chaque objet de l'image. Toutefois, la notion d'objet est généralement assez vague si bien que la problématique de la segmentation est par essence mal définie. La segmentation en matériaux au moins prise comme pré-traitement peut permettre une meilleure définition du problème. La restriction de la problématique de la segmentation à une segmentation en matériaux possède en effet deux avantages fondamentaux :

1. Tout d'abord le but a atteindre par l'algorithme peut être décrit précisément.
2. De plus, notre décomposition de l'image en matériaux peut s'appuyer sur les modèles physiques vus dans la Section 2.2 pour interpréter les valeurs des pixels.

Ce dernier avantage a de nombreuses conséquences pratiques. En effet le prédicat P utilisé en segmentation est souvent défini à l'aide des distances entre les couleurs de pixels. On peut par exemple imposer qu'une région soit homogène si la couleur de chacun de ses pixels est à une certaine distance de la couleur moyenne de la région. On peut également imposer que la distance entre deux pixels adjacents de la région ne dépasse pas un certain seuil. Toutefois, nous avons vu dans la section 2.2 que deux pixels adjacents correspondant au même matériau et au même objet peuvent avoir des intensités très différentes (voir par exemple le profil d'intensité sur la Figure 2.10). Ces limitations du processus de segmentation sont essentiellement dues à l'absence d'un modèle nous permettant d'interpréter la distance entre deux pixels adjacents. La segmentation en matériaux permet de ce baser sur des modèles physiques permettant d'interpréter les valeurs des pixels et leur différences. Ce processus ne fournit pas une segmentation en objets mais peut être utilisée comme première étape d'un processus de segmentation plus complet fournissant un ensemble de régions pas simplement homogènes mais composées d'un unique matériau que l'on peu éventuellement identifier. Les processus de plus haut niveau sont dans ce cas grandement simplifiés.

La majorité des algorithmes de segmentation en matériaux sont basés sur les modèles de Shafer [Sha85] ou Healey [Hea89] (Section 2.2.5). Nous allons dans cette Section établir les fondements de ces méthodes.

Le modèle de Shafer [Sha85](Section 2.2.5) exprime l'irradiance sur un capteur de la caméra par la somme d'une composante Lambertienne et d'une composante Spéculaire :

$$L(\lambda, \theta_i, \theta_r, g) = m_{diff}(\theta_i, \theta_r, g)c_{diff}(\lambda) + m_{spec}(\theta_i, \theta_r, g)c_{spec}(\lambda)$$

Si nous considérons que les angles θ_i, θ_r et g sont fonctions de la position (x, y) du pixel cette équation se réécrit :

$$L(\lambda, x, y) = m_{diff}(x, y)c_{diff}(\lambda) + m_{spec}(x, y)c_{spec}(\lambda) \tag{6.5}$$

Notez que dans un tel modèle les valeurs respectives de m_{diff} et C_{diff} ainsi que celles de m_{spec} et C_{spec} ne peuvent être déterminées qu'à un facteur multiplicatif prés. Shafer pose donc, sans perte de généralité : $0 \leq m_{diff}(x, y) \leq 1$ et $0 \leq m_{spec}(x, y) \leq 1$.

Étant donné une caméra couleur, la couleur d'un pixel $I(x, y)$ se déduit de l'irradiance sur ses capteurs par :

$$\forall i \in \{1, 2, 3\} \quad C_i(x, y) = \int_{\lambda_1}^{\lambda_2} L(\lambda, x, y) c_i(\lambda) d\lambda$$

où $C_i(x, y)$ représente la $i^{\text{ème}}$ composante de $I(x, y)$ et $(c_i(\lambda))_{i \in \{1,2,3\}}$ représentent la sensibilité de la caméra aux 3 composantes (usuellement le rouge, le vert et le bleu). Les deux longueurs d'ondes λ_1 et λ_2 représentent les deux bornes de sensibilités de la caméra ($\lambda_1 \approx 450nm$ et $\lambda_2 \approx 700nm$).

Si nous reprenons l'équation 6.5 nous obtenons :

$$\forall i \in \{1, 2, 3\} \quad C_i(x, y) = m_{diff}(x, y) \int_{\lambda_1}^{\lambda_2} c_{diff}(\lambda) c_i(\lambda) d\lambda + m_{spec}(x, y) \int_{\lambda_1}^{\lambda_2} c_{spec}(\lambda) c_i(\lambda) d\lambda +$$

Donc :

$$I(x, y) = m_{diff}(x, y) C_{diff} + m_{spec}(x, y) C_{spec} \tag{6.6}$$

avec

$$\forall i \in \{1, 2, 3\} \quad (C_{diff})_i = \int_{\lambda_1}^{\lambda_2} c_{diff}(\lambda) c_i(\lambda) d\lambda \text{ et } (C_{spec})_i = \int_{\lambda_1}^{\lambda_2} c_{spec}(\lambda) c_i(\lambda) d\lambda$$

L'équation 6.6 nous montre donc que pour un matériau l'information couleur a priori tridimensionnelle est en fait bidimensionnelle puisque l'ensemble des couleurs d'un matériau est contenu dans le plan défini par C_{diff}, C_{spec}. Plus précisément, puisque nous supposons m_{diff} et m_{spec} bornés entre 0 et 1, l'ensemble des couleurs d'un matériau est contenu dans le parallélogramme défini par les vecteurs C_{diff} et C_{spec} (Figure 6.1). Notons que dans le cas d'un seul illuminant un des coins du parallélogramme doit être l'origine de l'espace (cas où $m_{diff} = m_{spec} = 0$).

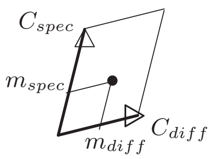


FIG. 6.1 – Représentation d'une couleur dans le parallélograme de Shafer.

Le modèle de Healey est basé sur une décomposition supplémentaire en métaux et diélectriques. On obtient en en effet à partir de l'équation 2.20 :

$$I(x, y) = \begin{cases} M_{spec}(x, y) C_{spec} & \text{Pour les métaux} \\ M_{spec}(x, y) C_{spec} + M_{diff}(x, y) C_{diff} & \text{Pour les diélectriques inhomogènes} \end{cases}$$

Donc l'ensemble des couleurs d'un métal peut être décrit par une droite. Alors qu'a priori l'ensemble des couleurs d'un diélectriques inhomogène peut être d'écrit par un parallélogramme.

Klinker [HSW92a] a affiné cette description en remarquant que dans le cas d'un diélectrique non homogène les points tels que le vecteur C_{spec} n'est pas négligeable correspondent à des zones de l'image fortement illuminées. De tels points correspondent à des zones très localisées dans l'image pour lesquelles on peut négliger les variations de la composante diffuse. Si nous nous référons au modèle de Nayar (Section 2.2.4). Un telle approximation, revient à négliger les variations de la composante Lambertienne lorsque le lobe spéculaire ou le pic spéculaire n'est pas négligeable.

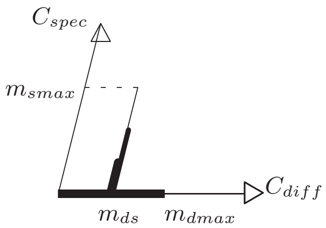


Fig. 6.2 – Répartition des couleurs dans le parallélogramme de Shafer celon Klinker.

La répartition des couleurs dans le parallélogramme de Shafer n'est donc pas uniforme mais suit une distribution similaire à celle représentée sur la Figure 6.2. L'ensemble des couleurs d'un diélectrique non homogène dans des régions non hautement éclairées peut donc être décrit uniquement par le vecteur C_{diff}. Dans le cas d'une zone de l'image hautement illuminée la composante de ce vecteur sur C_{diff} reste constante (valeur m_{ds} sur la Figure 6.2) alors que les variations se produisent sur le vecteur C_{spec}. L'histogramme des couleurs d'un matériau a donc la forme d'un T renversé et légèrement distordu. On peut également obtenir un histogramme en L si m_{ds} est égal à la valeur maximum m_{dmax} sur l'axe C_{diff}. Une étude de l'illumination d'un cylindre à amené Klinker [HSW92a] à conclure que le seuil m_{ds} ne pouvait se trouver que dans la seconde moitié de l'intervalle des variations $[0, m_{dmax}]$. Cette hypothèse confirmée par des expériences sur des images réelles est appelée *l'hypothèse des 50% supérieurs*. Nous appellerons respectivement les lignes définies par C_{diff} et $m_{ds}C_{diff} + C_{spec}$ la *ligne diffuse* et la *ligne spéculaire*.

Cette modélisation de l'ensemble des couleurs d'un matériau nous permet d'interpréter l'ensemble des couleurs du voisinage d'un point. En effet, étant donné un pixel de l'image et un fenêtre centrée sur celui-ci, l'ensemble des couleurs des pixels contenus dans cette fenêtre peut avoir une dimension 0, 1, 2 ou 3. La dimension de cet ensemble de couleurs peut être testée en utilisant les valeurs propres de la matrice de covariance de l'ensemble de couleurs (Section 5.2.1, équation 3).

- Dans le cas d'une fenêtre de dimension 0, le point considéré ne peut se trouver que sur un matériau avec une faible courbure ou sur une zone mal éclairée de ce même matériau.
- Dans le cas ou la fenêtre a pour dimension 1, la couleur du pixel se trouve sur la ligne diffuse ou sur la ligne spéculaire d'un matériau. La fenêtre peut également intersecter deux matériaux de telle façon que les couleurs de l'un deux sont situées sur la ligne diffuse alors que celles de l'autre corresponde à un seul point de la ligne diffuse. Le fenêtre intersecte alors une zone plate ou peu éclairée du second matériau.
- Si la fenêtre à pour dimension 2 la fenêtre est soit incluse dans un seul matériau avec les lignes diffuse et spéculaire soit intersecte deux matériaux de telle façon qu'une seule des deux droites de chaque matériau est comprise dans la fenêtre.
- Dans le cas de fenêtre de dimension 3, le pixel peut se trouver
  - soit dans un seul matériau. La troisième dimension est dans cas créée par le bruit d'acquisition.
  - soit à l'intersection de deux matériaux avec les deux droites diffuses et spéculaires pour l'un et une seule de ces droites pour l'autre.
  - Enfin la fenêtre peut également avoir intersecté une frontière entre au moins 3 matériaux.

Le cas ou le pixel est dans une fenêtre de dimension 3 est évidemment celui pour lequel on a le moins d'information. Il est toutefois remarquable de pouvoir faire de telles hypothèses à partir d'une fenêtre centrée sur un pixel.

Klinker [HSW92a] utilise cette classification pour décomposer l'image en matériaux. Schématiquement, sa méthode de segmentation se déroule de la façon suivante.

1. Décomposer l'image en fenêtre et étiqueter chaque fenêtre en fonction de sa dimension.
2. Fusionner les fenêtres adjacentes de même dimensions si l'ensemble obtenu a la même dimension.
3. Raffiner la segmentation de toutes les régions de dimension 1 par une croissance de région lancée au centre de la région. L'ensemble des couleurs de chaque région de dimension 1 correspond alors à une ligne diffuse ou à une ligne spéculaire.
4. Fusionner des régions de dimension 1 entre elles si l'ensemble des couleurs obtenues à une configuration en T ou en L. Ces fusions sont contraintes par l'hypothèse des 50% supérieurs. Les régions obtenues ont pour dimension 2.
5. Lancer un algorithme de croissance de région dans chaque région générée à l'étape précédente. Cet algorithme agrège à la région les pixels des régions de dimension 2 compatibles avec la configuration des couleurs de la région initiale.

Notons que les pixels de dimensions 0 et 3 ne sont pas directement utilisés par l'algorithme. En effet l'information liée à ces pixels est difficilement exploitable et ces pixels sont absorbés par les étapes de croissance de région dans des régions de dimension 1 ou 2.

La méthode présenté par Klinker utilise une approche ascendante. Healey [Hea95] a présenté une méthode comparable basée sur une approche descendante.

La méthode d'Healey néglige initialement la composante spéculaire des diélectriques inhomogènes. On a donc pour les métaux comme pour les diélectriques une équation de la forme :

$$L(\lambda, x, y) = M(x, y)c(\lambda)$$

où M représente M_{spec} pour les métaux et M_{diff} pour les diélectriques (équation 2.20). De même c représente C_{spec} pour les métaux et C_{diff} pour les diélectriques.

Donné les fonctions de sensibilité $(c_i)_{i\in\{1,2,3\}}$ de la caméra, la couleur $(C_i)_{i\in\{1,2,3\}}$ d'un pixel est donnée par :

$$\forall i \in \{1,2,3\} \quad C_i = \int_{\lambda_1}^{\lambda_2} M(x,y)c(\lambda)c_i(\lambda)d\lambda = M(x,y)K_i \text{ avec } K_i = \int_{\lambda_1}^{\lambda_2} c(\lambda)c_i(\lambda)d\lambda$$

Si nous notons par K le vecteur (K_1, K_2, K_3), la couleur I(x,y) d'un pixel est donc égale à :

$$I(x, y) = M(x, y)K$$

L'ensemble des couleurs d'un matériau appartient donc à une droite de vecteur directeur K.

Si nous normalisons les coordonnées de chaque pixel par sa norme dans L_2 nous obtenons :

$$\forall i \in \{1,2,3\} \quad \frac{C_i}{\|I(x,y)\|} = \frac{M(x,y)K_i}{\sqrt{M(x,y)^2\left(K_1^2+K_2^2+K_3^2\right)}} = \frac{K_i}{\|K\|}$$

Les coordonnées normalisées d'un pixel sont donc indépendantes de la géométrie. Notons qu'un résultat similaire aurait été obtenu en utilisant la norme L_1 plutôt que la norme L_2. Toutefois, la norme L_2 permet de définir une distance entre les droites qui ne dépend que de l'angle formé par celles-ci. Inversement la norme L_1 conduit à la définition d'une distance dépendant simultanément de l'angle entre les deux droites et de l'angle que forme une de ces deux droites avec un des axes de coordonnée. La norme L_2 est donc utilisée préférentiellement à la norme L_1.

La segmentation d'une image se ramène donc à une reconnaissance de droites dans l'espace couleur ou à la reconnaissance d'un ensemble de distributions normales dans l'espace normalisé par L_2. C'est cette dernière option que choisi Healey. Un matériau m_i est caractérisé par sa moyenne μ_i et sa matrice de covariance Σ_i (Section 5.2.1, équation 3). La distribution des couleurs dans le matériau est alors modélisée par la distribution normale $N(\mu_i, \Sigma_i)$ définie par

$$p(C/m_i) = \frac{1}{2\pi\sqrt{2\pi}\sqrt{|\Sigma_i|}} e^{-\frac{1}{2}(C-\mu_i)^t \Sigma_i^{-1}(C-\mu_i)}$$

La normalisation et effectué en considérant plutôt la distribution $N(\hat{\mu_i}, \widehat{\Sigma_i})$ avec :

$$\hat{\mu_i} = \frac{\mu_i}{\|\mu_i\|} \text{ et } \widehat{\Sigma_i} = \frac{\Sigma_i}{\|\mu_i\|^2}$$

Donné la modélisation de chaque matériau, la classification d'une couleur C a un matériau parmi M s'effectue en utilisant les fonctions suivantes basées sur la théorie de la décision de Bayes [DH73] :

$$g_i(\widehat{C}) = \log(p(\widehat{C}/m_i)) + \log(p_i)$$

où $\widehat{C} = \frac{C}{\|C\|}$ et p_i représente la probabilité qu'une couleur prise au hasard appartienne à m_i.

Si nous supposons que tous les matériaux sont équitablement représentés p_i devient indépendant de i et l'équation précédente peut se simplifier en :

$$g_i(\widehat{C}) = -\frac{1}{2}(\widehat{C} - \hat{\mu_i})^t \widehat{\Sigma_i}^{-1} (\widehat{C} - \hat{\mu_i}) - \frac{1}{2} \log(|\widehat{\Sigma_i}|). \tag{6.7}$$

La couleur C est alors affectée au matériau i pour lequel $g_i(\widehat{C})$ est maximum.

Donné cette méthode de classification, Healey commence par appliquer un opérateur gradient sur l'image. L'idée sous-jacente est de caractériser les zones correspondant à seul matériau comme des zones de faible gradient. Notons toutefois que cette supposition est discutable. En effet, l'opérateur gradient va non seulement réagir à des changements de matériaux mais également à de brusque changement de géométrie sur un objet composé d'un seul matériau.

Donné l'image de gradient Healey décompose l'image à l'aide d'un quadtree [CP95] et initialise une liste vide de matériaux. Tout noeud du quadtree couvrant une zone de l'image de faible gradient est supposé composé d'un seul matériau. Healey calcule donc sa couleur moyenne normalisée :

$$\hat{\mu} = \frac{\mu}{\||\mu\|} \text{ avec } \mu = \sum_{(x,y)\in R} I(x,y)$$

où R représente la région couverte par le noeud du quadtree.

La moyenne $\hat{\mu}$ est alors comparée aux M matériaux présents dans la liste à l'aide de l'équation 6.7. Si la valeur maximum de $g_i(\hat{\mu})$ est supérieure à un seuil T la région est affectée à un des matériaux de la liste. Sinon Healey considère que la région est composée d'un nouveau matériau et celui-ci est ajouté à la liste en utilisant les données de la région R.

La méthode de Healey segmente donc l'image en un ensemble de régions, tel que l'ensemble des couleurs de chaque région décrit une droite dans l'espace de couleurs. Il convient ensuite de fusionner les régions adjacentes composée d'un seul matériau telle que l'une des régions correspond à la droite diffuse tandis que l'autre correspond à la droite spéculaire. Schématiquement, une fusion est réalisée entre deux régions si :

1. Les couleurs des pixels de la région supposée correspondre à la ligne spéculaire ont des composantes supérieures ou égales à celle de la région correspondant à la ligne diffuse.

2. Les droites des deux régions se coupent en un point vérifiant l'hypothèse des 50% supérieurs.

La méthode de Healey repose donc sur une découpe récursive de l'image afin d'obtenir des patchs de surface composés d'un seul matériau. Malgré l'approche différente de celle de Klinker, ces deux méthodes sont basés sur les mêmes idées de base :

1. Exprimer la réflectance d'un matériau comme un produit d'un terme dépendant de la géométrie et d'un terme dépendant des longueurs d'ondes,
2. utiliser cette modélisation pour décrire la structure de l'ensemble des couleurs d'un matériau.

Donné une telle modélisation, l'algorithme de segmentation fusionne ou découpe des régions en se guidant sur les modèles décrivant les matériaux. De nombreuses adaptations de ces algorithmes ont été développées [Hea89, Bri90, HSW92b, FF95, Sha02]. Toutefois, l'idée principale de ces algorithmes repose généralement sur les deux points mentionnés précédemment.

6.3 Quantification

6.3.1 Objectifs

Le principal objectif des méthodes de quantification d'images couleurs est de réduire le nombre de couleurs de l'image originale en créant une distorsion minimale entre l'image quantifiée et l'image originale. Plus formellement, considérons une image I, nous pouvons y associer le multi-ensemble (C, f) où C représente l'ensemble des couleurs présentes dans l'image et $f(c)$ le nombre de pixels de couleur c dans I. La quantification de I en K couleurs, avec $K < |C|$ et usuellement $K << |C|$, consiste à sélectionner un ensemble de K **couleurs représentatives** et à remplacer chaque pixel de l'image originale par sa couleur représentative la plus proche. L'ensemble des couleurs représentatives $\{c_1, \dots, c_K\}$ est communément appelé la **table de couleurs**. La fonction qui associe à chaque couleur de l'image sa couleur représentative est appelée la fonction **d'inversion de table de couleurs** et est notée $\boldsymbol{Q}$.

La quantification d'image peut être utilisée pour afficher une image comportant un nombre important de couleurs, comme les images 24 bits sur des terminaux ne pouvant en afficher qu'un nombre réduit. Plus généralement, la quantification de couleurs peut être vue comme un processus permettant une compression des données de l'image. Il est généralement possible de construire une image extrêmement proche de l'original avec moins de 256 couleurs. Malgré leurs faibles taux de compression, les algorithmes de quantification restent très populaires. Wu [Wu92] a souligné une importante raison de cette popularité en remarquant que ces heuristiques combinées avec des tables de couleurs matérielles permettent un décodage de l'image en temps réel.De plus, la quantification d'une image en K couleurs permet de trouver les K couleurs les plus significatives de celle-ci. Cette propriété peut être utilisée par les algorithmes de segmentation pour retrouver les principales régions d'une image.

6.3.2 Choix d'un espace de couleurs pour la quantification

Jusqu'à présent, il n'a pas été effectué de tests systématiques des avantages et inconvénients des différents espaces de couleurs pour les algorithmes de quantifications. Ceci est sans doute dû à la grande diversité de ces algorithmes. Une telle démarche a été adoptée par Otha [OKS80] dans le cadre des algorithmes de segmentation. Ceci l'a conduit à la définition de l'espace $I_1I_2I_3$ (voir section 4.2.3).

Faute d'une étude exhaustive des avantages et inconvénients des différents espaces de couleurs pour la quantification, la plupart des méthodes de quantification ne font que préconiser un espace de couleurs. Les espaces de couleurs les plus recommandés sont les espaces RGB, YIQ, $L^*u^*v^*$ et$L^*a^*b^*$(voir section 4.2.2). L'espace RGB est souvent choisi pour sa simplicité. De plus, de nombreux formats d'images codent la couleur de chaque pixel dans cet espace. Utiliser celui-ci pour la quantification évite donc de convertir les coordonnées de chaque pixel. L'espace YIQ se déduit facilement de l'espace RGB. Il permet de plus de séparer l'information de luminance de l'information de chrominance. Les espaces uniformes $L^*u^*v^*$ et $L^*a^*b^*$ séparent également ces deux types d'informations. Ils présentent en outre l'avantage d'être plus en adéquation avec la vision humaine. Les conversions de l'espace RGB aux espaces $L^*u^*v^*$ ou $L^*a^*b^*$ sont les plus coûteuses des transformations vues au chapitre 4. Ces espaces de couleurs sont donc essentiellement utilisés par des algorithmes privilégiant la qualité des images obtenues au détriment des temps de calculs.

6.3.3 Utilisation de l'erreur quadratique

Nous avons vu à la section 5.2.1 que l'erreur quadratique $\boldsymbol{SE}(C)$ d'un multi-ensemble (C, f) permet de mesurer l'homogénéité de (C, f). De même, l'erreur quadratique d'une partition permet de mesurer l'homogénéité des ensembles formant la partition. L'utilisation de l'erreur quadratique pour la quantification repose sur le fait qu'une partition en un ensemble de multi-ensembles homogènes fournira une image quantifiée visuellement proche de l'original. Une des justifications de ce postulat repose sur la propriété suivante :

Proposition 2 *Soient I une image de taille $m \times n$, (C, f) le multi-ensemble associé à I et I' l'image issue de la quantification de I en K couleurs. Si l'ensemble des couleurs représentatives $\{c_1, \dots, c_K\}$ vérifie :*

$$\forall i \in \{1, \dots, K\} \quad c_i = \mu(Q^{-1}(c_i)) \tag{6.8}$$

alors l'erreur quadratique de la partition de (C, f) peut s'exprimer de la façon suivante :

$$E(C) = \sum_{i=1}^{m} \sum_{j=1}^{n} \|c_I(i, j) - c_{I'}(i, j)\|^2$$

où $c_I(i, j)$ et $c_{I'}(i, j)$ représentent les couleurs du pixel (i, j) dans les images I et I'.

Preuve:

On a pour tout pixel (i,j) de I' :

$$c_{I'}(i,j) = Q(c_I(i,j))$$

Donc,

$$\begin{array}{rcl}
\sum_{i=1}^{m}\sum_{j=1}^{n}\|c_I(i,j) - c_{I'}(i,j)\|^2 & = & \sum_{i=1}^{m}\sum_{j=1}^{n}\|c_I(i,j) - Q(c_I(i,j))\|^2 \\
& = & \sum_{c\in C}\|c - Q(c)\|^2 \\
& = & \sum_{i=1}^{K}\sum_{c\in Q^{-1}(c_i)}\|c - c_i\|^2 \\
& = & \sum_{i=1}^{K}\sum_{c\in Q^{-1}(c_i)}\|c - \mu(Q^{-1}(c_i)\|^2
\end{array}$$

Ce qui est la formulation de l'erreur d'une partition fournie par la définition 6.
□

La pré-condition 6.8 étant généralement vérifiée, l'erreur de partition peut se voir comme la somme pixel à pixel des différences de couleurs au carré entre l'image originale et l'image quantifiée. Il est toutefois dangereux de voir l'erreur quadratique comme une fonction de distance entre l'image originale et l'image quantifiée. Ceci est confirmé par l'expérience suivante. L'image 6.4-(a) est obtenue à partir de l'image 6.3 par une quantification en 8 couleurs. L'image originale 6.3 comporte 15 738 couleurs. Si nous appliquons l'algorithme de tramage (ou dithering) de Floyd-Steinberg [FS76] sur l'image quantifiée 6.4-(a) nous obtenons l'image 6.4-(b) qui est visuellement beaucoup plus proche de l'image originale 6.3. L'erreur quadratique de l'image 6.4-(b) est pourtant plus élevée que celle de l'image 6.4-(a), la différence relative entre les deux erreurs étant d'environ 30%. Ce phénomène a priori surprenant est dû au fait que l'algorithme de tramage a brisé la partition créée par l'algorithme de quantification. La nouvelle partition créée par l'algorithme de tramage tient compte de propriétés locales de l'image ce qui n'est pas pris en compte dans l'erreur quadratique. Les multi-ensembles créés par l'étape de tramage ne sont plus forcément connexes et sont a priori moins homogènes. L'erreur quadratique de ces multi-ensembles est donc plus importante que celle produite par les multi-ensembles issus de la quantification. Orchard [OB91] a défini une erreur quadratique pondérée permettant de tenir compte des propriétés locales de l'image durant l'étape de quantification. Orchard remplace dans la définition de l'erreur quadratique (voir définition 4) la fonction de fréquence f par la fonction de **fréquence pondérée** W, où $W(c)$ est définie comme une somme d'attributs calculés localement sur tous les pixels de couleur c de l'image. Dans ce cadre, l'erreur quadratique définie dans la section 5.2.1 peut se voir comme un cas particulier d'erreur quadratique pondérée où l'attribut d'un pixel de couleur c est égal à 1. Cette fonction de pondération permet de donner plus de poids aux couleurs de l'image majoritairement situées dans des régions sensibles aux erreurs de quantification.

FIG. 6.3 – *Rochers : image originale*

(a) (b)

FIG. 6.4 – *Application de l'algorithme de Floyd-Stenberg sur l'image (a)*

6.3.4 Classification des algorithmes

Une première approche pour définir une table de couleurs consiste à fixer l'ensemble des couleurs représentatives $\{c_1, \ldots, c_K\}$ de façon à couvrir un large spectre de couleurs. On parle dans ce cas de **quantification uniforme**, l'ensemble des couleurs représentatives est prédéterminé [GAW90, Pae91] et reste identique pour chaque image à quantifier. Il est bien évident que si le nombre de couleurs finales est réduit ($K = 8$ ou 16), ce type d'algorithme donne des résultats nettement moins intéressants qu'une quantification construisant un ensemble de couleurs représentatives adapté à chaque image. Les algorithmes de ce type seront appelés algorithmes de **quantification adaptative**. Ces algorithmes partitionnent le multi-ensemble (C, f) associé à l'image en un ensemble $\{(C_1, f), \ldots, (C_K, f)\}$ d'éléments de $\mathcal{ME}_3$. Ils associent ensuite à chaque multi-ensemble (C_i, f) une couleur représentative $\mu(C_i)$.

Ces méthodes se scindent en plusieurs familles. On distingue notamment les méthodes de quantification

1. par popularité
2. Les méthodes ascendantes
3. Les méthodes descendantes
4. Les méthodes mixtes

Nous allons par la suite étudier chacune de ces méthodes.

6.3.5 Les méthodes par popularité

Les méthodes de quantification par popularité sont historiquement parmi les premières à avoir été employées. Ces méthodes sélectionnent les K couleurs de plus grande fréquence de l'image (voir par exemple [Hec82]). Ces méthodes sont rapides et fonctionnent correctement pour un grand nombre d'images. Elles ont toutefois deux inconvénients majeurs :

Tout d'abord le nombre de couleurs affichables est bien plus grand que le nombre de couleurs d'une image (voir Section 5.2.2). De nombreuses cellules de l'histogrammes $3D$ sont donc affectées de valeurs faibles ou nulles. Outre des temps de calculs prohibitifs, cette dernière propriété remet en cause la pertinence des couleurs sélectionnées par l'algorithme. En effet, peut on dire qu'une couleur de fréquence 4 ou 5 est réellement représentative de l'ensemble des couleurs de l'image ? Un façon simple de remédier à ce problème est d'effectuer une pré quantification sur l'espace couleur. Si nous utilisons l'espace RGB on peut par exemple ne conserver que les 5 bits de poids fort de chaque composante.

Le second problème inhérent à ce type de méthode est que les couleurs de fréquence élevées ont tendance à être regroupées dans les même régions de l'espace couleur. Ceci conduit les méthodes par popularité à sur-représenter certaines partie de l'espace.

Braudeway [Bra86] résout partiellement ces deux problèmes en :

1. Effectuant une pré-quantification de l'espace couleur en L^3 sous cube de taille $\frac{M}{L}$ où M est la valeur maximale d'une composante correspondant à

une couleur affichable. Notons que ce type de méthode fonctionne particulièrement bien pour des espaces cubiques type RGB. On obtient donc un multi-ensemble :

$$(C, f) = \{(c_1, f), \ldots, (c_{L^3}, f)\}$$

où chaque élément c_i est le centre d'un sous cube et $f(c_i)$ représente la somme des fréquences des couleurs tombant dans ce sous cube.

2. Sélectionnant les K couleurs représentative par l'itération des deux étapes suivantes :
   (a) Sélectionner la couleur c de plus grande fréquence du multi-ensemble.
   (b) multiplier la fréquence de chaque couleur c' du multi-ensemble par un facteur $(1 - e^{\alpha d(c,c')})$ où $d(c, c')$ est la distance euclidienne entre c et c' et α un facteur de pondération.

6.3.6 Les méthodes ascendantes

Les méthodes ascendantes sélectionnent K couleurs de l'image afin d'initialiser K multi-ensembles, avec K le nombre de couleurs finales. Les autres couleurs de l'image sont alors lues et fusionnées aux K multi-ensembles courants selon différentes heuristiques [GP90, XJ94, Xia97]. De telles méthodes n'essaient généralement pas de minimiser l'erreur de partition (voir définition 6) et s'appuient sur des heuristiques qu'il est souvent difficile de justifier d'un point de vue théorique.

6.3.7 Étude détaillée de l'approche descendante

Introduction

Les **approches descendantes** ont quant à elles été beaucoup plus explorées [Hec82, WWP88, WZ91, Wu92, BBA94]. Ces méthodes initialisent un multi-ensemble (C, f) à partir de l'image et partitionnent celui-ci jusqu'à obtenir K multi-ensembles formant une partition de (C, f). Comme nous l'avons vu dans la section 5.2.1, trouver le partitionnement idéal en K multi-ensembles minimisant l'équation de la définition 6 est un problème $\mathcal{NP}$ complet pour des images couleurs. Selon Anderberg [And73], le nombre de partitionnements possibles est égal à $\frac{1}{K!}\sum_{i=0}^{K}(-1)^{K-i}C_K^i i^{|C|}$ où K est le nombre final de couleurs et (C, f) le multi-ensemble associé à l'image. L'algorithme trivial testant chaque partitionnement n'est donc pas réaliste et des heuristiques de découpe doivent être employées. Balasubramanian [BBA94] réalise un histogramme 1-D suivant une des composantes de l'image (voir définition 7) puis partitionne (C, f) par des plans perpendiculaires à cet axe en N_1 multi-ensembles. Il sélectionne ensuite un autre axe de découpe, et redécoupe chaque multi-ensemble perpendiculairement à cet axe à l'aide d'histogrammes 1D calculés sur chacun des N_1 multi-ensembles. Il obtient ainsi N_2 multi-ensembles. Une dernière itération de ce processus fournit les $N_3 = K$ multi-ensembles finaux. Cette méthode semble prometteuse mais laisse de nombreuses questions en suspend. Tout d'abord on ne connaît pas, sauf à faire une énumération exhaustive, la séquence optimale d'axes suivant lesquels on doit réaliser le partitionnement. De plus, l'extension de résultats valables pour K proche de l'infini à des valeurs de K comprises entre 4 et 256 semble pour le moins hasardeuse. Enfin, cet algorithme travaillant avec un ensemble d'histogrammes 1D tient a priori moins compte de l'information 3D des couleurs qu'un algorithme travaillant directement sur les données 3D. Une autre méthode présentée par Wu [Wu92] utilise la programmation dynamique pour partitionner le multi-ensemble (C, f) en κ multi-ensembles ($\kappa < K$). Les multi-ensembles restants sont alors récursivement découpés en deux jusqu'à obtenir les K multi-ensembles finaux. Le problème dans ce cas est de définir la valeur optimale de κ.

La plupart des algorithmes de quantification basés sur une approche descendante [Hec82, WWP88, WZ91, Wu92, BBA94] initialisent un multi-ensemble (C, f) à partir de l'image et subdivisent (C, f) récursivement en deux jusqu'à obtenir K multi-ensembles formant une partition de (C, f). Le découpage récursif consiste à choisir un multi-ensemble parmi les multi-ensembles déjà créés et à le découper en deux. Ce processus doit être itéré $K + 1$ fois de façon à obtenir les K multi-ensembles finaux. Le découpage de chaque multi-ensemble est réalisé par un plan appelé **plan de découpe** orthogonal à une direction appelée **direction de découpe**. Ce processus peut se subdiviser en 4 étapes élémentaires communes à tous les algorithmes de cette famille :

1. La sélection d'une stratégie de découpe
2. La sélection du multi-ensemble à découper
3. La sélection d'une direction de découpe
4. La recherche de la position du plan de coupe orthogonal à la direction de coupe.

Les heuristiques utilisées lors de ces 4 étapes nécessitent un stockage approprié du multi-ensemble 3D associé à l'image. Nous allons à présent étudier ces heuristiques et les structures de données utilisées pour stocker le multi-ensemble associé à l'image.

Sélection de la stratégie de découpe

Une stratégie de découpe possible consiste à découper récursivement (C, f) jusqu'à l'obtention des K multi-ensembles finaux. Cette séquence de découpes peut être représentée par un arbre binaire complet (un arbre dont chaque noeud possède exactement deux fils). Cet arbre est appelé l'**arbre de découpe** (voir Figure 6.5).

Le nombre de découpes possibles par cette stratégie est donc égal au nombre d'arbres binaires complets possédant exactement K feuilles soit : $\frac{1}{K} C_{2(K-1)}^{K-1}$ [FGS90]. Ce nombre est trop important pour envisager une énumération exhaustive de toutes les stratégies. Chou et al. [CTM89], Lin et al. [LSC91] et Balasubramanian et al. [BA91] ont suggéré de créer un arbre intermédiaire avec N feuilles

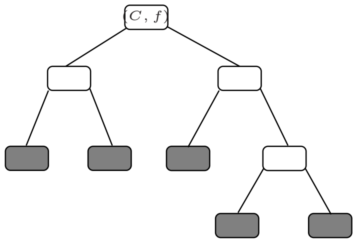


FIG. 6.5 – *Arbre de découpe représentant une stratégie de découpe de (C, f) en 5 multi-ensembles.*

(où $K < N << \frac{1}{K} C_{2(K-1)}^{K-1}$) et d'élaguer cet arbre de façon à ne conserver que les K feuilles qui conduisent à une erreur quadratique minimale. Selon Wu [WZ91], cette stratégie implique un surcoût trop important par rapport à l'amélioration apportée. Il recommande donc de ne générer que les K multi-ensembles nécessaires.

Sélection du multi-ensemble à découper

La stratégie la plus simple pour sélectionner le multi-ensemble à découper a été énoncée par Heckbert [Hec82]. Elle consiste à découper le multi-ensemble de plus grand cardinal. Cette stratégie peut être grandement améliorée en utilisant l'erreur de la partition (voir définition 6). Wan et al. [WWP88] ont proposé de sélectionner le multi-ensemble d'erreur quadratique maximale. Cette stratégie repose sur le fait que l'erreur quadratique de la partition est définie comme la somme des erreurs quadratiques des multi-ensembles formant cette partition. Donc en découpant le multi-ensemble dont la contribution à la somme est la plus importante, on espère faire diminuer celle-ci. Une stratégie légèrement différente a été proposée par Wu [WZ91]. Celui-ci découpe le multi-ensemble dont la partition va entraîner la plus grande diminution de l'erreur quadratique. Cette stratégie repose sur le raisonnement suivant :

Supposons que le multi-ensemble (C, f) contenant l'ensemble des couleurs de l'image ait été découpé k fois en $k+1$ multi-ensembles $\{(C_0, f), \ldots, (C_k, f)\}$. L'erreur $E(C)$ associée à la partition est alors égale à $\sum_{i=0}^{k} \boldsymbol{SE}(C_i)$. Supposons à présent que le prochain multi-ensemble à découper soit C_i et considérons C_i^1 et C_i^2 les deux multi-ensembles issus du découpage de C_i. L'erreur associée à la partition après le découpage est alors égale à :

$$E(C) = \boldsymbol{SE}(C_i^1) + \boldsymbol{SE}(C_i^2) + \sum_{j=0, j \neq i}^{k} \boldsymbol{SE}(C_j).$$

Le découpage du multi-ensemble (C_i, f) a donc modifié $E(C)$ de $\boldsymbol{SE}(C_i^1) + \boldsymbol{SE}(C_i^2) - \boldsymbol{SE}(C_i)$. On peut remarquer que le théorème 2 assure que $\boldsymbol{SE}(C_i^1) + \boldsymbol{SE}(C_i^2) - \boldsymbol{SE}(C_i)$ est négatif et donc que l'erreur de la partition est effectivement décroissante. L'heuristique consistant à couper le multi-ensemble dont le découpage entraînera une décroissance maximum de l'erreur de la partition est donc a priori meilleure que celle consistant à couper le multi-ensemble d'erreur quadratique maximale. Toutefois cette heuristique impose de couper à chaque itération chaque multi-ensemble afin de détecter celui qui réduira l'erreur au maximum. Cela implique que l'on va générer plus de multi-ensembles que nécessaire. De plus, selon Wu, cette heuristique ne fait diminuer l'erreur quadratique globale que de façon marginale.

Il apparaît ainsi que découper le multi-ensemble d'erreur quadratique maximum à chaque étape soit une bonne heuristique permettant de réaliser un bon compromis entre le temps de calcul et la diminution de l'erreur de la partition.

De plus, cette stratégie peut s'implémenter de manière très efficace en utilisant l'arbre de découpe. Il suffit d'associer à chaque noeud de l'arbre un pointeur sur la branche comportant la feuille d'erreur quadratique maximale. A chaque découpe de multi-ensemble, deux fils sont rajoutés au noeud correspondant et le pointeur est mis à jour le long du chemin menant de ce noeud à la racine de l'arbre. Cette méthode permet donc de retrouver rapidement, à chaque étape de l'algorithme, le multi-ensemble d'erreur maximum. Des études expérimentales nous ont montré que bien que l'arbre produit par les stratégies de Wan [WWP88] et Wu [WZ91] ne soit pas parfaitement équilibré, le temps de recherche du multi-ensemble à découper est quasiment logarithmique.

Sélection de l'axe de découpe d'un multi-ensemble

Une fois que le multi-ensemble à découper a été sélectionné, il faut choisir la normale du plan qui va couper celui-ci. La décroissance de l'erreur de la partition induite par la coupe étant égale à $\boldsymbol{SE}(C_i^1)+\boldsymbol{SE}(C_i^2)-\boldsymbol{SE}(C_i)$, le meilleur plan de découpe est celui qui minimise la somme $\boldsymbol{SE}(C_i^1)+\boldsymbol{SE}(C_i^2)$. Malheureusement, nous ne disposons pas d'outils mathématiques permettant de calculer le plan optimal et une énumération de tous les plans possibles est là encore irréaliste. On simplifie donc le problème en déterminant d'abord la normale au plan qui défini l'axe de coupe puis la position du plan sur cet axe.

Un choix simple de l'axe de coupe a été proposé par Heckbert [Hec82]. Celui-ci propose de découper l'axe de coordonnée sur lequel le multi-ensemble à découper est le plus étendu. Wu [WZ91] a étendu cette idée en choisissant la direction de plus grande variance du multi-ensemble. Cette direction est donnée par le vecteur propre associé à la plus grande valeur propre de la matrice de covariance du multi-ensemble à découper. Ce vecteur est plus simplement appelé **l'axe principal** du multi-ensemble. La justification de cette heuristique apparaît clairement lorsque l'on examine l'équation 5.1 que nous rappelons ci-dessous :

$$\boldsymbol{SE}(C) = \boldsymbol{M_0}(C)\sum_{i=1}^{3} var_i(C). \tag{6.9}$$

Si l'on effectue un changement de repère en prenant pour nouvelle base la base des vecteurs propres, l'équation 6.9 devient :

$$\boldsymbol{SE}(C) = \boldsymbol{M_0}(C)\sum_{i=1}^{3}\lambda_i \tag{6.10}$$

où λ_i représente la i^{eme} valeur propre et est égale à la variance du multi-ensemble le long de la direction définie par le i^{eme} vecteur propre.

Si nous découpons le multi-ensemble perpendiculairement à un axe, nous allons faire décroître majoritairement la variance suivant cette axe. L'idée de base de l'heuristique de Wu consiste à supposer que l'on obtient une plus grande décroissance de l'erreur quadratique en faisant décroître en priorité la plus grande des variances, c'est à dire en coupant perpendiculairement à l'axe principal. Néanmoins, même si elle est très souvent vérifiée pour des images réelles, cette heuristique peut être mise en défaut. En effet, considérons l'exemple de la figure 6.6 où le multi-ensemble (C, f) est défini par :

$$\begin{array}{lcl} C & = & \{(i,\sqrt{7}), i \in \{1,2,..,10\}\} \cup \{(i,-\sqrt{7}), i \in \{1,2,..,10\}\} \\ f(x) & = & 1 \ \ \forall x \in C. \end{array}$$

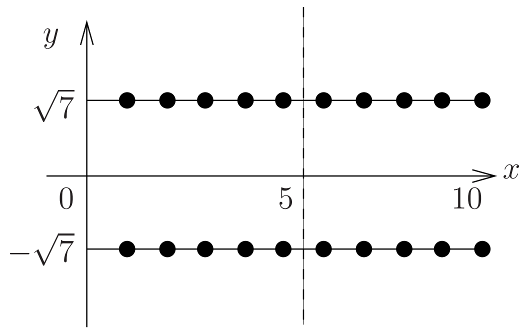


FIG. 6.6 – *Les points noirs sont les éléments de (C, f) dont l'axe principal est le vecteur $\vec{x}$.*

Dans cet exemple, la matrice de covariance de (C, f) est alors égale à :

$$\begin{pmatrix} 8.25 & 0 \\ 0 & 7 \end{pmatrix}.$$

Ceci signifie que l'axe principal de (C, f) est le vecteur $\vec{x}$. Pourtant, si nous découpons (C, f) perpendiculairement au vecteur $\vec{x}$, l'erreur quadratique $\boldsymbol{SE}(C)$ ne décroîtra que de $6,25$ alors qu'un découpage perpendiculaire à $\vec{y}$ entraîne une diminution de 7. Dans ce cas, l'axe principal du multi-ensemble est donc orthogonal à la direction de coupe optimale. C'est donc la direction la plus éloignée de l'optimum.

Outre qu'elle ne conduit pas nécessairement à la meilleure solution, l'heuristique de Wu implique plusieurs contraintes algorithmiques :

– Le fait de couper perpendiculairement à une direction variable complique le traitement des données nécessaire au découpage. Par exemple, avant d'effectuer un découpage, l'algorithme de Wu doit trier les données de chaque multi-ensemble suivant leur projection sur la direction de coupe.
– Le découpage d'un multi-ensemble réclame le calcul de la matrice de covariance du multi-ensemble et le calcul de la plus grande valeur propre de cette matrice. Le calcul du vecteur propre associé s'effectue par une méthode itérative ce qui s'avère relativement coûteux.

Sélection du plan de coupe

Une fois sélectionnés le multi-ensemble à découper (C, f) et l'axe de coupe A, nous devons définir la position du plan de coupe le long de l'axe A. La stratégie la plus simple est encore une fois proposée par Heckbert [Hec82]. Celui-ci propose de couper (C, f) par un **plan médian**, c'est à dire un plan tel que $|C_1| = |C_2|$ où C_1 et C_2 sont les deux multi-ensembles issus de la découpe de (C, f).

Cette stratégie peut être grandement améliorée en tenant compte de l'erreur de la partition. En effet, soient m et M les deux extrémités de (C, f) le long de l'axe A (voir Figure 6.7 et définition 7) et t un réel de l'intervalle $[m, M]$. Le plan orthogonal à A et de position t sur l'axe A découpe (C, f) en deux multi-ensembles (C_t, f) et $(C - C_t, f)$. L'erreur associée à cette partition est égale à :

$$E_t(C) = \boldsymbol{SE}(C_t) + \boldsymbol{SE}(C - C_t) \tag{6.11}$$

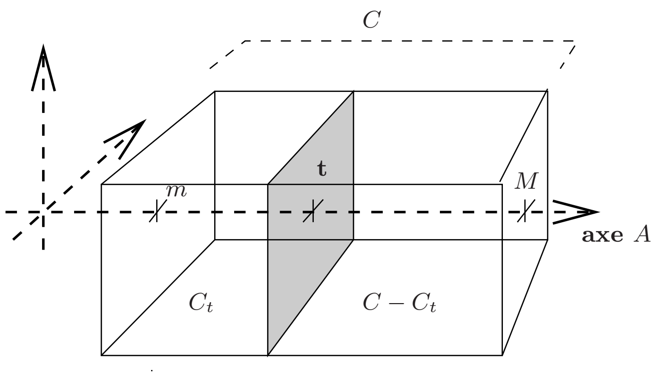


FIG. 6.7 – *Le multi-ensemble sélectionné est découpé orthogonalement à l'axe de coupe A positionné à la coordonnée t sur l'axe A.*

Wu [Wu92] a donné une autre formulation de l'erreur de la partition :

$$E_t(C) = \sum_{v \in C} f(v) \|v\|^2 - \left(\frac{\|M_1(C_t)\|^2}{M_0(C_t)} + \frac{\|M_1(C) - M_1(C_t)\|^2}{M_0(C) - M_0(C_t)} \right). \tag{6.12}$$

Cette nouvelle formulation est plus efficace puisqu'elle ne fait apparaître que le multi-ensemble C_t que nous allons faire évoluer jusqu'à trouver la valeur t_{opt}

minimisant $E_t(C)$. Elle présente toutefois quelques inconvénients. Tout d'abord elle impose de manipuler des grands nombres tels que les moments d'ordre un au carré. Du point de vue pratique, la présence de grands nombres oblige à stocker les variables sur des entiers ou réels longs qui ralentissent l'algorithme. De plus, cette formule se prête peu à des manipulations et peut donc difficilement être optimisée.

Wan et al. ont simplifié l'équation 6.12 en minimisant non pas l'erreur de la partition mais la somme des variances $var_A(C_t)$ et $var_A(C-C_t)$. Intuitivement, cette simplification revient à approximer le multi-ensemble (C, f) par sa projection sur l'axe A (voir définition 7). Afin de limiter la perte d'information liée à cette simplification, Wan et al. définissent A comme l'axe principal de (C, f). On peut montrer que la recherche de la valeur t_{opt} peut à ce moment là ce limiter à l'intervalle $[\frac{\mu+m}{2}, \frac{\mu+M}{2}]$. Cette approche permet donc de limiter l'intervalle de recherche mais ne permet pas d'obtenir la position du plan minimisant l'erreur de la partition.

Le calcul de la valeur t_{opt} par les méthodes de Wu ou de Wan et al. nécessite le calcul des moments $\boldsymbol{M_0}(C_t)$ et $\boldsymbol{M_1}(C_t)$ pour t appartenant à l'intervalle $[m, M]$. Ces moments peuvent être calculés rapidement grâce à la propriété suivante. Supposons que le multi-ensemble (C, f) défini par $C = [a_1, a_2[\times[b_1, b_2[\times[c_1, c_2[$ doit être découpé perpendiculairement au premier axe de coordonnée. Pour tout t nous avons alors : $C_t = [a_1, t[\times[b_1, b_2[\times[c_1, c_2[$. Si nous définissons :

$$D_t = \{t\} \times [b_1, b_2[\times[c_1, c_2[$$

C_t peut se définir comme l'union des D_t. Le multi-ensemble (C, f) étant discret, l'intervalle $[a_1, a_2]$ est égal à $\{a_1, a_1+1, \ldots, a_2\}$ et les moments peuvent être calculés incrémentalement par les formules suivantes :

$$\forall i \in \{0,1,2\} \left\{ \begin{array}{l} M_i(C_{a+1}) = M_i(D_a) \\ \forall t \in]a+1, b[, M_i(C_{t+1}) = M_i(C_t) + M_i(D_t). \end{array} \right. \tag{6.13}$$

Les moments du multi-ensemble $(C_{t_{opt}}, f)$ peuvent donc être calculés incrémentalement à l'aide des équations 6.13. Les moments du multi-ensemble $(C - C_{t_{opt}}, f)$ se déduisent des moments de $(C_{t_{opt}}, f)$ par la proposition 1 sur l'additivité des moments. Une fois les moments des deux multi-ensembles calculés, on peut calculer leurs moyennes, leurs variances et leurs erreurs quadratiques grâce à la définitions 3 et à l'équation 5.1. Les moments, moyennes et erreurs quadratiques de chacun des multi-ensemble sont ensuite stockés dans les feuilles de l'arbre de découpe correspondant aux multi-ensembles.

6.4 Inversion de table de couleur

6.4.1 Problématique

Nous avons vu dans la section 6.3.1 que l'objectif d'un algorithme de quantification est de partitionner le multi-ensemble (C, f) contenant l'ensemble des

		1	2	3	4
Choix du multi-ensemble	Erreur quadratique		$\star$	$\star$	$\star$
	Cardinal	$\star$			
Choix d'un ensemble d'axes	Axes de coordonnées	$\star$			$\star$
	Axe principal		$\star$	$\star$	
Choix de l'axe	Plus grande longueur	$\star$			
	Plus grande variance		$\star$	$\star$	$\star$
Choix du plan	Plan médian	$\star$			
	Minimisation de la variance		$\star$		
	Minimisation de l'erreur			$\star$	$\star$

(1)	:	Méthode de Heckbert
(2)	:	Méthode de Wan and Wong
(3)	:	Méthode de Wu
(4)	:	Méthode de Braquelaire et Brun.

TAB. 6.1 – Ce tableau résume les principales heuristiques employées par les algorithmes de quantification basés sur une division récursive du multi-ensemble de départ.

couleurs de l'image en un ensemble de multi-ensembles $\{(C_1, f), \dots, (C_K, f)\}$. Une fois cette partition déterminée, il convient d'afficher l'image avec le nouvel ensemble de couleurs et ce avec une distorsion visuelle minimale. Partant de cette partition, l'on construit un ensemble de couleurs représentatives $\{c_1, \dots, c_K\}$ défini par :

$$\forall i \in \{1, \dots, K\} \quad c_i = \mu(C_i).$$

La fonction $\boldsymbol{Q}$ définie dans la section 6.3.1 associe à chaque couleur de l'image sa couleur représentative la plus proche. On a donc :

$$\boldsymbol{Q}(c) = ArgMin_{z \in \{c_1, \dots, c_K\}} \|z - c\| \tag{6.14}$$

où $Arg\ Minf(t)$ représente la valeur d'un paramètre t réalisant le minimum, et $\|z - c\|$ la norme du vecteur $z - c$ définie dans un espace métrique donné (par exemple la norme euclidienne dans l'espace RGB).

Cette étape est usuellement appelée l'inversion de table de couleurs. Elle consiste à affecter à chaque couleur de l'image sa couleur la plus proche dans l'ensemble $\{c_1, \dots, c_K\}$. Le calcul de l'ensemble des couleurs plus proche d'une couleur représentative donnée peut s'interpréter comme le calcul du diagramme de Voronoï 3D [Ber94, Aur91, CP95] de germes $\{c_1, \dots, c_K\}$. Les méthodes utilisant explicitement le calcul du diagramme de Voronoï permettent d'obtenir très rapidement la plus proche couleur représentative d'une couleur donnée. Cependant, cette efficacité est compensée par le pré-calcul du diagramme de Voronoi qui induit un sur-coût de calcul important.

6.4.2 La méthode triviale

Le calcul des valeurs prises par la fonction $\boldsymbol{Q}$ pour l'ensemble des couleurs d'une image peut être réalisé par une recherche exhaustive du minimum de $\|z - c\|$ pour tout z dans $\{c_1, \ldots, c_K\}$. Si l'on réalise une quantification d'une image de taille $256 \star 256$ en 256 couleurs, nous devrons effectuer pour chaque pixel de l'image 256 calculs de distances. L'affichage de l'image avec sa nouvelle table de couleurs implique donc 256^3, soit plus de 16 millions de calculs de distances. Cette méthode nous permet donc d'obtenir une valeur exacte de la fonction $\boldsymbol{Q}$ mais reste peu utilisée en raison du nombre élevé de calculs qu'elle implique.

6.4.3 Améliorations de la méthode triviale

Les améliorations de la méthode triviale sont nombreuses : Poskanzer [Pos91] à proposé d'améliorer la recherche en utilisant une table de hachage de façon à ne pas recalculer la couleur représentative d'une couleur déjà rencontrée. Toutefois cette amélioration reste inefficace pour des images comportant un nombre important de couleurs différentes. Une autre approche consiste à approximer la norme L_2 par une norme moins coûteuse en temps de calculs. Chaudhuri et al. [CCW92] ont proposés la norme L_α en tant qu'approximation de la distance euclidienne définie par L_2. La norme L_α d'une couleur $\mathbf{c}$ étant définie par :

$$
\begin{array}{rcl}
\|\mathbf{c}\|_\alpha & = & (1-\alpha)\|\mathbf{c}\|_1 + \alpha\|\mathbf{c}\|_\infty \\
& = & (1-\alpha)\sum_{j=1}^{3} |\mathbf{c}^j| + \alpha \max_{i\in\{1,2,3\}} |\mathbf{c}^j|
\end{array}
$$

D'après les expériences menées par Verevka [VB95] la norme $L_{\frac{1}{2}}$ accélère la recherche de façon significative sans introduire une perte notable de la qualité de l'image résultat.

La recherche peut être encore réduite en utilisant les considérations suivantes [Hod88] :

- Somme partielle :
  Si nous utilisons le carré de la norme L_2, ou les normes L_1 ou $L_{\frac{1}{2}}$ la distance entre la couleur d'entrée et une couleur représentative est définie comme la somme de trois termes. La somme partielle doit être comparée à la distance minimale avant chaque addition. Le calcul de distance n'a en effet aucune raison de continuer si une somme partielle est plus grande que la distance minimale.
- Tri sur une coordonnée (voir exercices sections 6.5.2 et 6.6.2) :
  Supposons que les couleurs représentatives sont triés suivant un des axes de coordonné (par ex. le premier) Alors la recherche débute sur la couleur représentative dont la première coordonnée est la plus proche de celle de la couleur d''entrée et continue ensuite dans l'ordre croissant des distances sur la première coordonnée. La recherche se termine quand la distance sur la première coordonnée entre la couleur représentative courante et la couleur d'entrée est plus grande que la distance minimale. Notons que le

temps moyen de recherche sera d'autant plus court que la variance sur l'axe des coordonnée choisis sera grand.

– Distance du plus proche voisin : Donné une couleur représentative $\mathbf{c}_j$ avec $j \in \{1, \ldots, K\}$ sa plus proche couleur représentative $\mathbf{c}_{q(j)}$ est définie par :

$$q(j) = \arg \min_{k \in \{1,\ldots,K\}, k \neq j} d(\mathbf{c}_j, \mathbf{c}_k)$$

Supposons que la couleur représentative courante $\mathbf{c}_i$ parcourue par l'algorithme soit la plus proche de la couleur d'entrée $\mathbf{c}$. Supposons de plus que la distance minimale courante $\Sigma_{min} = d(\mathbf{c}, \mathbf{c}_i)$ soit plus petite que la moitié de $d(\mathbf{c}_i, \mathbf{c}_{q(i)})$. On a donc :

$$\begin{cases} \Sigma_{min} & = & d(\mathbf{c}, \mathbf{c}_i) \\ \Sigma_{min} & \leq & \frac{1}{2} d(\mathbf{c}_i, \mathbf{c}_{q(i)}) \end{cases}$$

Donné n'importe quelle autre couleur représentative $\mathbf{c}_k$:

$$\begin{array}{rrcll} & 2\Sigma_{min} & \leq & d(\mathbf{c}_i, \mathbf{c}_{q(i)}) & \leq d(\mathbf{c}_i, \mathbf{c}_k) \leq d(\mathbf{c}_i, \mathbf{c}) + d(\mathbf{c}, \mathbf{c}_k) \\ \Rightarrow & 2\Sigma_{min} & \leq & \Sigma_{min} + d(\mathbf{c}, \mathbf{c}_k) & \\ \Rightarrow & \Sigma_{min} & \leq & d(\mathbf{c}, \mathbf{c}_k) & \end{array}$$

L'algorithme d'inversion de table de couleur peut donc retourner directement $\mathbf{c}_i$ puisque aucune autre couleur représentative n peut être plus proche de $\mathbf{c}$ que $\mathbf{c}_i$.

6.4.4 La recherche par tri local d'Heckbert

La méthode d'Heckbert [Hec82], basée sur une découpe uniforme de l'espace de couleurs, permet de rejeter a priori des couleurs représentatives qui ne peuvent réaliser le minimum de l'équation (6.14). Heckbert effectue une découpe du cube RGB en N sous cubes. Chaque cube contient une liste de couleurs représentatives pouvant être la couleurs représentative d'une couleur du cube. Chaque liste est définie en calculant la distance r entre la couleur représentative la plus proche du centre du cube et le coin du cube le plus éloigné de la couleur représentative(Figure 6.8). Cette distance donne une limite supérieure de la distance entre une couleur du cube et sa couleur représentative. Donc toute couleur représentative dont la distance au cube est plus grande que r est rejetée de la liste.

Les tests effectués par Heckbert [Hec82] montrent que le nombre de tests nécessaires pour calculer l'image $\boldsymbol{Q}(c)$ d'une couleur c est d'environ $\mathcal{O}(\frac{K}{23})$, où K est le nombre de couleurs représentatives. Cette méthode permet donc de diminuer la complexité du calcul de la fonction $\boldsymbol{Q}$ sans toutefois changer l'ordre de grandeur du temps de calcul de cette fonction.

6.4.5 La méthode de Thomas

On peut aussi diminuer l'ordre de complexité de la fonction $\boldsymbol{Q}$ en utilisant un diagramme de Voronoi 3D [Tho91]. Chaque cellule de Voronoi V_i est alors

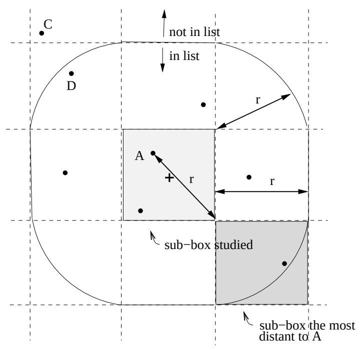


FIG. 6.8 – La recherche par tri local d'Heckbert. La couleur représentative A est la plus proche du centre de la boite courante. La distance entre A et le coin de la boite le plus éloigné défini r. La couleur représentative D appartient à la liste du cube alors que C situé à une distance supérieure à r du centre du cube est rejeté de la liste

définie comme l'ensemble des points de l'espace de couleurs plus proches de la couleur c_i que de toute autre couleur représentative. Ce type de méthode permet de pré-calculer de façon incrémentale l'ensemble des valeurs de la fonction $\boldsymbol{Q}$ pour tout point discret c de l'espace de couleurs. L'inconvénient majeur de cette méthode est que l'on effectue bien plus de calculs que nécessaire. En effet, si l'on travaille dans l'espace de couleurs RGB, le pré-calcul de la fonction $\boldsymbol{Q}$ impose d'initialiser $256^3 \approx 16.000.000$ valeurs. Si l'image à quantifier a une taille de $256 \star 256$, nous aurons besoin de tester la valeur de la fonction $\boldsymbol{Q}$ pour au plus $256^2 \approx 65.000$ valeurs différentes. Cette méthode peut donc nécessiter de nombreux calculs inutiles. De plus, le stockage de l'ensemble des valeurs de la fonction $\boldsymbol{Q}$ requiert, si l'on utilise 256 couleurs représentatives, 16 méga octets de mémoire. Afin de diminuer la place mémoire requise par l'algorithme on effectue généralement une pré-quantifiquation en ne prenant en compte que les 5 bits de poids fort de chaque composante (R, G, B). Cette méthode diminue la place mémoire requise par l'algorithme et le nombre de valeurs à calculer. Toutefois, malgré cette simplification, le pré-calcul de la fonction $\boldsymbol{Q}$ nécessite environ 24 secondes sur une station de travail SUN 3/60 [Tho91]. Sachant que l'obtention d'une partition de l'espace de couleur s'effectue généralement en moins d'une minute, le sur-coût de calcul qu'implique le pré-calcul de la fonction $\boldsymbol{Q}$ est important.

6.4.6 La méthode de Friedman

Il est également possible d'approcher le calcul de la fonction $\boldsymbol{Q}$ en affectant à chaque couleur c le barycentre c_i de son ensemble englobant C_i. Cette méthode proposée par Friedman et al. [BFF77] est utilisée par la majorité des algorithmes de quantification. L'approximation des cellules de Voronoi $\{V_1, \ldots, V_K\}$ par les ensembles $\{C_1, \ldots, \boldsymbol{C}_K\}$ utilise le fait que la partition de l'espace de couleur en cellules de Voronoi est celle qui minimise l'erreur quadratique pour un ensemble de couleurs représentatives donné. La partition produite par les algorithmes de quantification étant proche de celle qui minimise l'erreur de partition, chaque ensemble C_i est proche de la cellule de Voronoi V_i de centre c_i où c_i est le barycentre de C_i. Les algorithmes utilisant ce type de méthode utilisent généralement la structure de données utilisée pour stocker la partition de l'ensemble des couleurs de l'image. Par exemple, les méthodes de quantification descendantes stockent généralement l'ensemble des ensembles formant la partition à l'aide d'un arbre binaire. L'utilisation de cet arbre permet alors de retrouver l'ensemble contenant une couleur donnée en $\mathcal{O}(\log_2(K))$ tests. Toutefois, ce type de méthode ne peut s'appliquer que lorsque l'étape d'inversion de la table de couleur suit immédiatement l'étape de quantification. Dans le cas contraire, reproduire les structures intermédiaires utilisées par les algorithmes de quantification est souvent plus coûteux que la méthode d'Heckbert [Hec82] ou la recherche exhaustive.

6.4.7 La méthode de Brun

Nous avons vu dans la section 6.4.5 que le calcul de la fonction $\boldsymbol{Q}$ pouvait être réalisé à l'aide d'un diagramme de Voronoi 3D discret. Les principales limitations de cette méthode viennent du pré-calcul de la couleur représentative de chaque couleur de l'espace RGB. Les travaux menés par Otha [OKS80] et confirmés par nos propres expérimentations montrent que l'essentiel de l'information d'une image naturelle (par opposition à une image de synthèse) est contenu dans le plan vectoriel P_{princ} défini par les deux premiers vecteurs propres de la matrice de covariance de l'image. Le plan P_{princ} nous permet donc d'approximer efficacement l'ensemble des couleurs de l'image. De plus, la matrice de passage de l'espace de couleurs à l'espace des vecteurs propres étant une matrice orthogonale, les calculs de distances effectués dans l'espace des vecteurs propres seront similaires aux calculs de distances effectués dans l'espace de couleur initial. Il est donc possible d'approximer de manière efficace le diagramme de Voronoi 3D discret de centres $\{c_1, \ldots, c_K\}$ par un diagramme de Voronoi 2D discret défini par les centres $\{p(c_1), \ldots, p(c_K)\}$ où $p(c_i)$ représente la projection de la couleur représentative c_i sur le plan P_{princ}. Cette projection est définie par les équations suivantes :

$$\begin{cases} p(c)_{v_1} & = & v_1 \bullet c \\ p(c)_{v_2} & = & v_2 \bullet c \end{cases}$$

où v_1 et v_2 représentent les deux premiers vecteurs propres de la matrice de covariance et $\bullet$ le produit scalaire.

L'utilisation d'un diagramme de Voronoi 2D discret impose de délimiter le domaine sur lequel les calculs de distances d'un point $p(c)$ aux centres $\{p(c_1), \ldots, p(c_K)\}$ seront effectués. A cette fin, nous calculons la boite englobante des projections $p(c)$ des couleurs de l'image sur le plan P_{princ}. Le calcul de cette boite englobante s'effectue en déterminant les valeurs max_1, max_2 et min_1, min_2 définies par :

$$\forall i \in \{1,2\} \begin{cases} min_i &= Min_{c\in C}\ p(c)_{v_i} \\ max_i &= Max_{c\in C}\ p(c)_{v_i} \end{cases}$$

où C désigne l'ensemble des couleurs de l'image.

Le calcul du diagramme de Voronoi 2D discret peut alors s'effectuer en allouant une image de taille $(max_1 - min_1) \star (max_2 - min_2)$ et en affectant à chaque point discret de l'image l'indice de son centre le plus proche parmis les points $\{p(c_1), \ldots, p(c_K)\}$. Cette initialisation du diagramme de Voronoi 2D discret est effectuée grâce à la méthode incrémentale de Danielson [Dan80] de complexité $\mathcal{O}(|V_I|)$ où $|V_I|$ représente la taille du diagramme de Voronoi 2D discret associé à I.

Le calcul du diagramme V_I impose donc de parcourir une fois l'image pour calculer les deux vecteurs propres v_1 et v_2, puis de reparcourir celle ci afin de déterminer les points (min_1, min_2) et (max_1, max_2). La compléxité totale du calcul du diagramme V_I est donc égale à $\mathcal{O}(2|I| + |V_I|)$.

Le diagramme de Voronoi V_I associé à une image I nous permet donc d'approximer le calcul de la fonction $\boldsymbol{Q}$ définie dans la section 6.4.1. Un algorithme utilisant cette approximation pour réaliser l'inversion de la table de couleurs est donné sur la figure 4.

L'utilisation d'un diagramme de Voronoi 2D discret induit deux types d'approximation : Une approximation due à la projection de l'ensemble des donnés 3D de l'image sur le plan P_{princ} et une approximation due à la discrétisation du plan P_{princ} pour calculer le diagramme de Voronoi 2D. Du fait de ces deux approximations l'algorithme 4 est suceptible de renvoyer un indice différent de celui fourni par la fonction $\boldsymbol{Q}$. Toutefois, les erreurs dues aux approximations induisant de petites erreurs pour le calcul des distances, les erreurs commises par l'algorithme 4 se situent souvent entre deux indices dont les cellules sont adjacentes. Afin de supprimer ce type d'erreur nous déterminons durant le calcul de V_I le diagramme de Delaunay D_I associé à V_I. Ces deux structures sont alors combinées de la façon suivante. Soit $V_I[p(c)]$ l'indice de la cellule contenant la projection de c. L'utilisation du diagramme de Delaunay nous permet d'acceder à la structure $D_I[V_I[p(c)]]$ contenant l'ensemble des cellules adjacentes à $V_I[p(c)]$. L'utilisation du nouveau diagramme D_I nous permet d'affecter à c la couleur la plus proche de celui-ci parmis l'ensemble des couleurs représentatives dont l'indice appartient à $D_I[V_I[p(c)]]$. Ceci définit une nouvelle fonction $\mathbf{q}$ donnée par l'équation :

$$\mathbf{q}(c) = ArgMin_{i\in D_I[V_I[p(c)]]} \|c_i - c\|$$

où $V_I(p(c))$ représente l'indice de la cellule contenant c et $D_I(k)$ l'ensemble des indices dont les cellules sont adjacentes à la cellule d'indice k.

Un algorithme utilisant le diagramme de Delaunay D_I et la fonction $\mathbf{q}$ pour inverser la table de couleurs d'une image est représenté sur la figure 5.

La complexité moyenne de la fonction $\mathbf{q}$ est déterminée par le nombre moyen de cellules adjacentes à une cellule donnée. Les travaux d'Etienne Bertin [Ber94] ont montré que le nombre moyen d'arrêtes d'une cellule de Voronoi est indépendant du nombre de germes et approximativement égal à 6.

```
inverse_table_couleur1(image I,diagramme Voronoi V_I)
{
    Pour tout pixel P ∈ I
    {
        proj = p[I(P)]
        ind = V_I[proj]
        I(P) = c_ind
    }
}
```

Algorithm 4: *Premier approche pour l'inversion de la tables de couleurs. Les symboles $I(P)$ et $V_I[proj]$ représentent respectivement la couleur du pixel P et l'indice de la cellule de Voronoï contenant $proj$. La projection sur le plan P_{princ} est notée p.*

```
inverse_table_couleur(image I,diagramme Voronoi V_I,diagramme Delaunay D_I)
{
    Pour tout pixel P ∈ I
    {
        proj = p[I(P)]
        ind = ArgMin_{i∈D_I[V_I[proj]]} ||c_i − c||
        I(P) = c_ind
    }
}
```

Algorithm 5: *Second algorithme d'inversion de tables de couleurs. Les symboles $I(P)$ et $V_I[proj]$ représentent respectivement la couleur du pixel P et l'indice de la cellule de Voronoi contenant $proj$.La projection sur le plan P_{princ} est notée p.*

6.5 Exercices

6.5.1 Quantification par fusion

On suppose que l'espace RGB est contenu dans un cube de taille $256 \times 256 \times 256$. On désire partionner ce cube en $N = 2^{3p}$ (avec $p < 8$) sous-cube de tailles égales.

1. En combien d'intervalles faut il partionner chaque axe ? Quelle est la taille de chaque sous-cube ?

   Les cubes sont numérotés dans le sens des R, puis des G, puis des B. Ainsi, si α représente le coté de chaque sous-cube, le cube $[0, \alpha] \times [0, \alpha] \times [0, \alpha]$ aura le numéro 0, puis le cube, $[\alpha, 2\alpha] \times [0, \alpha] \times [0, \alpha]$ aura le numéro 1 et ainsi de suite (faire un dessin dans le plan $R - G$).
2. Donnez la formule donnant le numéro du cube contenant un triplet (R, G, B) donné.
3. Étant donnée une image couleur codée en RGB, on désire associer à chaque

sous-cube les moments d'ordre 0, 1 et 2 des couleurs contenues dans ce sous cube. Ecrire la procédure qui réalise cette opération. On supposera pour cela que l'on possède les classes suivantes :
- La classe image avec les méthodes :
  - debut() : positionne un pointeur courant en haut, à gauche de l'image.
  - lire_pixel(unsigned char*) : rempli le tableau de taille 3 passé en paramètre avec la valeur du pixel.
  - suivant(), avance le pointeur courant vers la droite, puis vers le bas, renvoie faux lorsqu'il arrive en bas à droite de l'image vrai sinon.
- Une classe sous_cube contenant les champs publics :
  - M_0, M_1 et M_2 qui correpondent aux moments d'ordre 0, 1 et 2. Le constructeur par défaut initialise ces champs à 0. M_0 sera considéré comme un float, alors que M_1 et M_2 seront considérés comme des tableaux de float de taille 3.
  - Un champ entier père qui sera utilisé plus tard.

N'oubliez pas d'allouer le tableau de sous-cubes.

4. Si l'on associe à chaque couleur de l'image la moyenne du sous cube qui le contient, combien de couleurs (au maximum) seront présentes dans l'image résultat ?
5. Donnez la formule indiquant l'erreur quadratique d'un sous cube, d'indice k. On notera le tableau de sous cubes tab_cube. Indiquez l'augmentation de l'erreur quadratique de partition provoquée par la fusion de deux sous cube.
6. On suppose que l'on a fusionné des sous-cubes de façon à se ramener à un nombre $K < N$ fixé de sous-cubes. Si l'on effectue les fusions deux à deux quels sont selon vous les deux sous-cubes qu'il convient de fusionner à chaque étape ?
7. A chaque étape de fusion entre deux sous-cubes, le sous-cube d'indice minimum est invalidé et on positionne son champ père à l'indice du sous-cube d'indice maximum. Les moments de celui-ci sont mis à jour. Ecrire la fonction $void\ merge\_sous\_cubes(sous\_cubes*, int\ i, int\ j)$ qui réalise cette opération. On supposera que le champ père est initialisé à l'indice du sous-cube.
8. Notez qu'apres plusieurs fusions, les sous-cubes deviennent des multi-ensembles quelconques. Ecrivez la procédure $give\_mean(sous\_cube*, unsignedint * color, unsignedint * moyenne, intp)$ qui recopie dans le tableau de taille 3 moyenne la moyenne du multi-ensemble contenant $color$.
9. Supposons que l'on fusionne le sous-cube 3 avec le sous-cube 5, puis le sous-cube 5 le sous-cube 10. Combien d'indirections doit effectuer la procédure give_mean ? Proposez une méthode pour supprimer ces multiples indirections.

6.5.2 Inversion de table de couleur par tri sur une coordonnée

On considère la classe suivante :

```
class colormap
{
private:
  unsigned char* tab;
  unsigned int nb_color;
public:
  colormap(image&);
  ~colormap();
  int give_entry(unsigned char);
  typedef enum {forward,backward} iterator_mode;

  class iterator
  {
  private:
    unsigned char* tab;
    unsigned int nb_color;
    int current;
    iterator_mode mode;
  public:
    iterator(colormap& map,int index=-1,iterator_mode mode=forward):
    tab(col.tab),nb_color(col.nb_color),mode(dir)
    {
        if(index==-1) {current=index;return}
        current=(mode==forward?0:nb_color-1);
    }
    iterator(iterator&):
    tab(ite.tab),nb_color(ite.nb_color),current(ite.current),mode(ite.mode)
    {
    }

    unsigned char operator [](int);
    iterator& operator ++();
    iterator operator ++(int);
    bool operator <(const iterator&);
    iterator end();
    int index() const
    {
        return current;
    }
  };
  void inverse(unsigned char*,unsigned char*);
```

```
private:
  void oriented_search(iterator &,unsigned char *,int&,int &);
};
typedef colormap::iterator colormap_iterator;
```

Constructeur

Le constructeur prend une image de taille K colonnes et 1 ligne pour initialiser le tableau tab qui contient alors une série de valeurs de la forme $RGBRGB\ldots$. Il utilise ensuite la fonction qsort pour trier la table de façon croissante sur la première composante. La syntaxe de cette fonction est la suivante :

> `void qsort (void *base, size_t nmemb, size_t size, int (*compar)(const void *, const void *));`
>
> **DESCRIPTION**
>
> La fonction qsort() trie une table contenant nmemb éléments de taille size. L'argument base pointe sur le début de la table.
>
> Le contenu de la table est trié en ordre croissant, en utilisant la fonction de comparaison pointée par compar, laquelle est appelée avec deux arguments pointant sur les objets à comparer.
>
> La fonction de comparaison doit renvoyer un entier inférieur, égal, ou supérieur à zéro si le premier argument est respectivement considéré comme inférieur, égal ou supérieur au second. Si la comparaison des deux arguments renvoie une égalité (valeur de retour nulle), l'ordre des deux éléments est indéfini.

Donnez le code de la fonction `compar` et indiquez les paramètres de l'appel à la fonction qsort.

Recherche de l'entrée dans la table

En supposant la table triée, donnez le code de la fonction :

```
int give_entry(unsigned char value,unsigned char* colormap)
```

qui renvoie l'indice (dans tab) de la couleur dont la première composante est la plus proche de `value`. On fera une recherche dichotomique.

L'itérateur

L'itérateur parcourt la table suivant les indices croissants ou décroissants selon la valeur du mode (forward ou backward). La méthode end() permet d'effectuer des boucles du type :

```
for(colormap_iterator ite(map);ite<ite.end();ite++)
```

Notez que le sens de parcourt de la table dépend du mode.

1. Donnez le code de la méthode end.
2. Donnez le code des opérateurs ++, ++(int) et <.

Recherche orientée

La méthode :

```
void oriented_search(iterator &,unsigned char *,int&,int &);
```

prend en paramètre : un itérateur, un pointeur sur un tableau de 3 unsigned char contenant les valeurs RGB de la couleur d'entrée, un entier codant une distance minimale et l'index de l'entrée la plus proche.

Cette méthode utilise l'itérateur pour parcourir une partie de la table. Si une couleur de la table à une distance à input inférieure à min_dist, celui-ci est mis à jour et l'index est initialisé à la valeur de l'index de l'itérateur.

Donnez le code de cette méthode en utilisant le trie sur la première coordonnée de façon à abandonner le parcourt dès qu'aucune couleur restant à parcourir ne peut avoir une distance à input inférieure à min_dist.

Inversion de la table

En utilisant les méthodes et classes précédemment définies donnez le code de la méthode inverse qui prend en premier paramètre une couleur d'entrée et en second paramètre la couleur la plus proche dans la table de la couleur d'entrée. Le code obtenu ne doit pas excéder une quinzaine de lignes et doit utiliser le fait que la table est triée sur la première composante.

6.6 Correction des Exercices

6.6.1 Quantification par fusion

Si $N = 2^{3p}$, il faudra partitionner chaque axe en 2^p intervalles. La longueur de chaque intervalle est alors égale au côté d'un sous cube et est égale à : $\alpha = 2^{8-p}$.

La quantité $\frac{R}{\alpha}$ nous donne le numéro de l'intervalle sur l'axe R. De même, les quantités $\frac{G}{\alpha}$ et $\frac{B}{\alpha}$ nous donnent respectivement les numéros d'intervalles sur les axes G et B. Compte tenu de notre numérotation et sachant que le plan $R - G$ comporte 2^{2p} carrés, le numéro d'un sous cube est :

$$\begin{aligned} ind &= B.2^{p-8}.2^{2p} + G.2^{p-8}.2^p + R.2^{p-8} \\ &= B.2^{3p-8} + G.2^{2p-8} + R.2^{p-8} \end{aligned}$$

Notez que ce calcul peut s'effectuer efficacement avec des décalages.

```
init_partition(const image &I, int p)
{
        unsigned char couleur[3];
        int N=(1<<3*p);
        sous_cube * tab_cube = new sous_cube[N];

        I.debut();
```

```
        do
        {
                I.lire_pixel(couleur);
                int ind =couleur[2]*2^(3*p-8)+couleur[1]*2^(2*p-8)+couleur[0]*2^(p-8)
                tab_cube[ind].M0++;
                for(int i =0;i<3;i++)
                {
                        tab_cube[ind].M1[i] += couleur[i];
                        tab_cube[ind].M2[i] += couleur[i]*couleur[i];
                }
        }
        while(I.suivant());
}
```

Le nombre de couleurs de l'image résultat sera égal à N si tous les sous cubes contiennent au moins une couleur. En pratique, on trouve généralement des valeurs bien inférieures à N.

L'expression de l'erreur quadratique du cluster k se trouve dans le cours et est égale à :

$$\sum_{i=0}^{2} tab\_cube[k].M_2[i]^2 - \frac{tab\_cube[k].M_1[i]^2}{tab\_cube[k].M_0}$$

L'erreur quadratique de partition avant la fusion est donnée par :

$$E = \sum_{i=1}^{N} SE_i$$

où N représente le nombre de sous cubes et SE_i l'erreur quadratique du sous cube i. La fusion de deux sous cubes m et n crée un multi-ensemble d'erreur quadratique :

$$SE = SE_m + SE_n + \frac{M_{0m}M_{0n}}{M_{0m} + M_{0n}} \|\mu_m - \mu_n\|^2$$

où M_{0m}, M_{0n}, μ_m et μ_n représentent respectivement les moments d'ordres 0 et les moyennes des sous-cubes m et n.

L'erreur quadratique de partition après la fusion est donc égale à :

$$\begin{aligned}
E' &= \textstyle\sum_{i=1, i \notin \{m,n\}}^{N} SE_i + SE \\
&= \textstyle\sum_{i=1, i \notin \{m,n\}}^{N} SE_i + SE_m + SE_n + \frac{M_{0m}M_{0n}}{M_{0m}+M_{0n}} \|\mu_m - \mu_n\|^2 \\
&= \textstyle\sum_{i=1}^{N} SE_i + \frac{M_{0m}M_{0n}}{M_{0m}+M_{0n}} \|\mu_m - \mu_n\|^2 \\
&= E + \frac{M_{0m}M_{0n}}{M_{0m}+M_{0n}} \|\mu_m - \mu_n\|^2
\end{aligned}$$

L'augmentation de l'erreur quadratique de partition est donc de : $\frac{M_{0m}M_{0n}}{M_{0m}+M_{0n}}\|\mu_m - \mu_n\|^2$. Il convient donc de fusionner à chaque étape les deux sous-cubes qui induisent une augmentation minimum de l'erreur quadratique de partition. L'augmentation induite par la fusion étant décrite plus haut.

```
void merge_sous_cubes(sous_cube * tab_cube, int i,int j)
{
        int max = MAX(i,j);
        int min = MIN(i,j);

        tab_cube[min].pere = max;

        tab_cube[max].M0+=tab_cube[min].M0;
        for(int i=0;i<3;i++)
        {
                tab_cube[max].M1[i] += tab_cube[min].M1[i];
                tab_cube[max].M2[i] += tab_cube[min].M2[i];
        }
}
```

```
void give_mean(sous_cube * tab_cube, unsigned int* color,unsigned int* moyenne,int p)
{
        int ind =couleur[2]<<(3*p-8)+couleur[1]<<(2*p-8)+couleur[0]<<(p-8);

        while (ind != tab_cube[ind].pere)
                ind = tab_cube[ind].pere;
        for(int i=0;i<3;i++)
                moyenne[i] = tab_cube[ind].M1[i]/tab_cube.M0;
}
```

La suppression des indirections peut s'effectuer grâce à une simple boucle descendante du type suivant :

```
        for(int i=N-1;i>0;i--)
        {
                int ind =tab_cube[i].pere;
                tab_cube[i].pere= tab_cube[ind].pere;
        }
```

qui positionne le père de chaque sous cube au père de son père. Etant donné que les relations père-fils vont toujours de l'indice le plus petit vers l'indice le plus grand, la boucle descendante supprime toutes les indirections en une passe.

6.6.2 Inversion de table de couleur par tri sur une coordonnée

Constructeur

Le code de la fonction compar peut simplement être défini par :

```
int compar(const void* pt1,const void* pt2)
{
  short val1=*((unsigned char*)pt1);
  short val2=*((unsigned char*)pt2);
  return val1 -val2;
}
```

Il suffit ensuite d'appeller la fonction qsort comme suit :

```
qsort(tab,nb_color,3*sizeof(unsigned char),compar);
```

Recherche de l'entrée dans la table

Il s'agit d'une recherche dichotomique classique. Il faut simplement faire attention à la multiplication par 3 des indices pour retomber toujours sur la première composante. De plus, lorsque l'intervalle $[min, max]$ est réduit à 1, on renvoie l'indice le plus proche de la valeur d'entrée.

```
int colormap::give_entry(unsigned char value)
{
  int min(0),max(nb_color);

  int midle=(min+max)/2;

  while (max-min>1)
    {
      if(tab[3*midle]==value)
        return midle;

      if(tab[3*midle] > value)
        max=midle;
      else
        min=midle;
      midle=(min+max)/2;
    }
  if(value-tab[3*min] < tab[3*max]-value)
    return min;
  else
    return max;
}
```

L'itérateur

la méthode end : Le test étant un test strict, il faut renvoyer des indices en dehors de la zone allouée. Ce qui nous donne :

```
colormap::iterator colormap::iterator::end()
{
```

```
  iterator ite(*this);
  if(mode==forward)
    ite.current=nb_color;
  else
    ite.current=-1;

  return ite;
}
```

Les opérateurs ++ et ++(int) : L'unique petite originalité tient ici au fait que l'on incrémente ou décrémente l'indice en fonction du mode. Le reste est assez classique est s'écrit :

```
colormap::iterator colormap::iterator::operator++(int nothing)
{
  colormap::iterator tmp(*this);
  if(mode==forward)
    current++;
  else
    current--;
  return tmp;
}
```

et

```
colormap::iterator& colormap::iterator::operator++()
{
  if(mode==forward)
    current++;
  else
    current--;
  return *this;
}
```

L'opératueur < : On obtient en prennant en compte le mode :

```
bool colormap::iterator::operator <(const colormap::iterator &ite)
{
  if(mode==forward)
    return current < ite.current;

  return current > ite.current;
}
```

Recherche orientée

Le code de la méthode est le suivant :

```
void colormap::oriented_search(iterator & ite,
                               unsigned char * input,
```

```
                                  int& min_dist,
                                  int & index)
{
  int current_dist;

  for(;ite<ite.end();ite++)
    {
      current_dist=(input[0]-ite[0])*(input[0]-ite[0]);
      if(current_dist > min_dist)
        break;
      for(int i=1;i<3;i++)
        current_dist+=(input[i]-ite[i])*(input[i]-ite[i]);
      if(current_dist<min_dist)
        {
          index=ite.index();
          min_dist=current_dist;
        }
    }
}
```

Notez le test sur la première composante pour sortir de la boucle dès que la distance sur la première composante est supérieure à min_dist. Pour le reste on fait avancer l'itérateur jusqu'à la fin en mettant à jour à chaque étape la distance minimale et l'indice correspondant.

Inversion de la table

La méthode finale d'inversion de couleur est simplement une concaténation des méthodes précédemment définies :

```
void colormap::inverse(unsigned char* input,
                       unsigned char* output)
{
  int start=give_entry(input[0]);
  iterator to_beg(*this,start-1,backward);
  iterator to_end(*this,start+1,forward);
  int min_dist(0),current_dist;
  int index(start);

  for(int i=0;i<3;i++)
    min_dist+=(input[i]-tab[3*start+i])*(input[i]-tab[3*start+i]);

  oriented_search(to_beg,input,min_dist,index);
  oriented_search(to_end,input,min_dist,index);

  for(int i=0;i<3;i++)
    output[i]=tab[3*index+i];
```

```
}
```

On récupère tout d'abord l'indice (dans la table) de la couleur dont la distance suivant la première composante est la plus proche de la couleur d'entrée. On initialise ensuite deux itérateurs nous permettant de parcourir la table dans le sens des indices croissants et décroissants à partir de l'indice précédemment calculé. On initialise également, la distance minimale et l'index associé à la valeur retournée par la fonction `give_entry`. Il suffit ensuite de lancer deux `oriented_search` permettant d'effectuer la recherche dans les deux directions. Le résultat est ensuite recopié dans la variable de sortie.

Chapitre 7

Annexe

7.1 Quelques opérateurs différentiels

L'opérateur nabla $\nabla = (\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z})$ est un pseudo-vecteur utile pour retenir les équations définissant les différents opérateurs différentiels utilisées en électomagnétisme

Ces opérateurs sont définis de la façon suivante :

– Le gradient $grad(f)$ est un opérateur s'appliquant à une fonction f dérivable de $\mathbb{R}^3$ dans $\mathbb{R}$. Il est défini par :

$$grad(f) = \begin{pmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \\ \frac{\partial f}{\partial z} \end{pmatrix}.$$

On peut retenir la formule du gradient en notant que : $grad = \nabla(f)$.

– La divergence $div(f)$ est un opérateur s'appliquant également à une fonction dérivalble de $\mathbb{R}^3$ dans $\mathbb{R}^3$. Il est défini par :

$$div(f) = \frac{\partial f_x}{\partial x} + \frac{\partial f_y}{\partial y} + \frac{\partial f_z}{\partial z}.$$

On peut retenir la formule de la divergence en notant que : $div = \nabla.f$ où $''.''$ représente le produit scalaire.

– Le rotationnel $rot(f)$ s'applique à une fonction définie de $\mathbb{R}^3$ dans $\mathbb{R}^3$. Il est défini par :

$$rot(f) = \begin{pmatrix} \frac{\partial f_z}{\partial y} - \frac{\partial f_y}{\partial z} \\ \frac{\partial f_x}{\partial z} - \frac{\partial f_z}{\partial x} \\ \frac{\partial f_y}{\partial x} - \frac{\partial f_x}{\partial y} \end{pmatrix}.$$

On peut retenir l'opérateur rot en notant que : $rot(f) = \nabla \wedge f$ où $\wedge$ représente le produit vectoriel.

– Le laplacien scalaire $\triangle$ s'applique à une fonction de $\mathbb{R}^3$ dans $\mathbb{R}$ et est défini par :

$$\triangle f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} + \frac{\partial^2 f}{\partial z^2}.$$

Le laplacien scalaire se retrouve à partir de l'opérateur nabla en remarquant que : $\triangle(f) = \nabla.\nabla(f)$.

7.1.1 Identitées vectorielles

$$\begin{aligned}
div \circ grad &= \triangle \\
rot \circ grad &= 0 \\
div \circ rot &= 0 \\
rot \circ rot &= grad \circ div - \triangle
\end{aligned}$$

où m et n sont des champs scalaires de $\mathbb{R}^3$ dans $\mathbb{R}$ et A et B des champs vectoriels de $\mathbb{R}^3$ dans $\mathbb{R}^3$.

7.1.2 Formules sur les opérateurs

$$\begin{aligned}
grad(mn) &= mgrad(n) + ngrad(m) \\
div(mA) &= mdiv(A) + grad(m).A \\
div(A \wedge B) &= B.(rot(A)) - A.(rot(B)) \\
rot(mA) &= mrot(A) + grad(m) \wedge A
\end{aligned}$$

7.2 Les théorèmes de Green-Ostrogradsky et Stokes-Ampère

Théorème de Green-Ostrogradsky :

$$\int\int_{(S)} bdS = \int\int\int_{(V)} div(b)dv$$

Théorème de Stokes-Ampère :

$$\int_L a.dl = \int\int_S rot(a)dS$$

où a et b sont des fonctions vectorielles de $\mathbb{R}^3$ dans $\mathbb{R}^3$.

7.2.1 Formules dérivées

$$\begin{aligned}
\int_L \varphi.dM &= \int\int_S dS \wedge grad(\varphi) \\
\int\int_S \varphi dS &= \int\int\int_{(V)} grad(\varphi)d\tau \\
\int\int_S dS \wedge A &= \int\int\int_{(V)} rot(A)d\tau
\end{aligned}$$

où A est une fonction vectorielle de $\mathbb{R}^3$ dans $\mathbb{R}^3$ et φ un champ scalaire de $\mathbb{R}^3$ dans $\mathbb{R}$.

Index

Bibliographie

[And73] M.R. Anderberg. *Cluster analysis for applications.* Academic Press, New York, 1973.

[Aur91] F. Aurenhammer. Voronoi diagrams : a survey of fundamental geometric data structure. *ACM Computing surveys*, 33(3) :345–405, 1991.

[BA91] Raja Balasubramanian and J. Allebach. A New Approach to Palette Selection for Color Images. *Human Vision, Visual Processing, and Digital Display III (1991)*, SPIE 1453 :58–69, 1991.

[BBA94] Raja Balasubramanian, Charles A. Bouman, and Jan P. Allebach. Sequential scalar quantization of color images. *Journal of Electronic Imaging*, 3(1) :45–59, January 1994.

[Ber94] E. Bertin. *Diagrammes de Voronoi 2D et 3D : application en analyse d'images.* PhD thesis, Université J. Fourier, Grenoble, 1994.

[BFF77] Jon Louis Bentley, Jerome H. Friedman, and R. A. Finkel. An algorithm for finding best matches in logarithmic expected time. *ACM Trans. Math. Software*, 3 :209–226, September 1977.

[BK02] Raja Bala and R. Victor Klassen. *Digital Color Imaging Handbook*, chapter Efficient color transformation implementation. Electrical and Applied Signal Processin. CRC Press, 2002.

[Bra86] Gordon W. Braudeway. A procedure for optimum choice of a small number of colors from a large color palette for color imaging. In *proc. of Electronic Imaging 86*, pages 75–79, november 1986.

[Bri90] Michael H. Brill. Image segmentation by object color : a unifying framework and connection to color constancy. *Opt. Soc. Am. A.*, 7(10) :2041–2047, October 1990.

[BS87] P. Beckmann and A. Spizzichino. *The scattering of electromagnetic waves from rough surfaces.* Artech House Inc, 2nd edition, 1987.

[Can86] J.F. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6) :679–698, 1986.

[CCW92] C. A. Chaudhuri, W. T. Chen, and J.S. Wang. A modified metric to compute distance. *Pattern Recognition*, 7(25) :667–677, 1992.

[CD85] J.-P. Cocquerez and J. Devars. Détection du contour dans les image aériennes : nouveaux opérateurs. *Traitement du signal*, 2(1), 1985.

[CGG91] A. Cumani, P. Grattoni, and A. Guiducci. An edge-based description of color images. *Computer Vision, Graphics, and Image Processing*, 53(4) :313–323, July 1991.

[Cha92] M. Chapron. A new chromatic edge detector used for color image segmentation, 1992.

[CP95] J.-P. Cocquerez and S. Philipp. *Analyse d'images : Filtrage et segmentation*. Masson, 1995.

[CTM89] P. A. Chou, Lookabaugh T., and Gray R. M. Optimal pruning with applications to tree-structured source coding and modeling. *IEEE Trans. Inf. Theory*, 2 :299–315, 1989.

[Cum89] A. Cumani. A second order differential operator for multispectral edge detection. In *5th Int. Conf. Image Anal. Process.*, pages 54–58, Positano, 1989.

[Cum91] A. Cumani. Edge detection in multispectral images. *Computer Vision, Graphics, and Image Processing*, 53 :40–51, 1991.

[CZS89] S. Castan, J. Zhao, and J. Shen. Une famille de détecteurs de contours basée sur le filtre exponentiel optimal. In 7^{eme} *congrès AFCET-RFIA*, Paris, 1989.

[Dan80] Per-Erik Danielson. Euclidean distance mapping. *Computer Vision, Graphics, and Image Processing*, 14 :227–248, 1980.

[DC87] R. Deriche and J.-P. Cocquerez. Extraction de composantes connexes basée sur une détection optimale des contours. In *MAR87*, Paris, la vilette, mai 1987.

[Der87] R. Deriche. Usinf canny's criteria to derive a recursively implemented optimal edge detector. *International Journal of Computer Vision*, 2(1) :167–187, 1987.

[DH73] R.O. Duda and P.E. Hart. *Pattern classification and scene analysis*. Wiley New York, 1973.

[FF95] Brian V. Funt and Graham D. Finlayson. Color constant color indexing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(5) :522–529, May 1995.

[FGS90] Chritine Froidevaux, Marie-Claude Gaudel, and Michèle Soria. *Types de données et algorithmes*. Mc Graw-Hill, 1990.

[FS76] R.W. Floyd and L. Steinberg. An adaptative algorithm for spatial greyscale. *Proc. SID*, 17(2) :75–77, 1976.

[GAW90] Ronald S. Gentile, Jan P. Allebach, and Eric Walowit. Quantization of color images bases on uniform color spaces. *Journal of imaging technology*, 16 :11–21, 1990.

[GP90] M. Gervautz and W. Purgathofer. A Simple Method for Color Quantization : Octree Quantization. In A. S. Glassner, editor, *Graphics Gems*, pages 287–293. Academic Press Professional, Cambridge, MA, 1990.

[Hea89] Glenn E. Healey. Using color for geometry insensitive segmentation. *J. Opt. Soc. Am. A.*, 6 :920–937, June 1989.

[Hea95] Glenn Healey. Segmenting images using normalized color. *IEEE Transactions on Systems, Man and CYBERNETICS*, 22(1) :64–73, January 1995.

[Hec82] Paul Heckbert. Color image quantization for frame buffer display. In *Proceedings of SIG-GRAPH'82*, pages 297–307, 1982.

[HM86] A. Huertas and G. Medioni. Detection of intensity changes with subpixel accuracy using laplacian-gaussian masks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(5) :651–661, 1986.

[Hod88] Michael H. Hodgson. Reducing the computation requirements of the minimum distance classifier. *Remote sencing of Environnement*, 25 :117–128, 1988.

[Hor86] B. P. K. Horn. *Robot vision.* Cambridge MA, MIT Press, 1986.

[HP75] S. L. Horowitz and T. Pavlidis. Picture segmentation by a directed split-and-merge procedure. Technical report, Departement of Electrical Engineering, Princeton University , N. J. 08540, 1975.

[HP76] S. L. Horowitz and T. Pavlidis. Picture segmentation by a tree traversal algorithm. *Journal of The Association for Computing Machinery*, 23(2) :368–388, April 1976.

[HSW92a] Glenn E. Healey, Steven A. Shafer, and Lawrence B. Wolff, editors. *Color*, chapter A Physical Approach to Color Image Understanding, pages 134–165. Jones and Bartlett, 1992.

[HSW92b] Glenn E. Healey, Steven A. Shafer, and Lawrence B. Wolff, editors. *Color*, chapter Color Image Segmentation. Jones and BArtlett, 1992.

[Hun75] R. S. Hunter. *The measurement of Appearance.* Wiley, New York, 1975.

[Kir71] R. Kirsch. Computer determination of the constituent structure of biological images. *Computer Biomedical Research*, 4 :315–328, 1971.

[KL31] P. Kubelka and F. Lunk. Ein beitrag zur optik der farbanstriche. *Z. Techn. Physik*, 12(593), 1931.

[Kri05] J. A. von Kries. *Handbuch der Physiologie der Menschen*, volume 3, chapter III, pages 109–282. Vieweg, Brunswick, 1905.

[LSC91] J. Lin, J. Storer, and M. Cohn. On the complexity of the optimal tree pruning for source coding. In *Proc. of data compression conference*, pages 63–72. IEEE computer society Press Los Angeles, 1991.

[Lum95] H. Lumbroso. *Problèmes résolus sur les ondes électromagnétiques.* Bordas, 1995.

[Mac70] D. L. MacAdam. *Sources of Color Science*. MA :MIT, Cambridge, 1970.

[MH80] D. Marr and E. Hildreth. Theory of edge detection. *Proccedings of Royal Society of London*, 207 :187–217, 1980.

[Mon90] O. Monga. *Segmentation d'images : Où en sommes nous ?* INRIA, Domaine de voluceau Rocquancourt B.P. 105 78153 Le chesnay Cedex France, Avril 1990. Rapports de Recherche.

[NIK91] Shree K. Nayard, Katsuchi Ikeuchi, and Takeo Kanade. Surface reflection : Physical and geometrical perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(7) :611–634, July 1991.

[NM79] M. Ngao and T. Matsuyama. Edge preserving smoothing. *Computer Vision, Graphics, and Image Processing*, 9 :394–407, 1979.

[OB91] Michael T. Orchard and Charles A. Bouman. Color quantization of images. *IEEE transactions on signal processing*, 39(12) :2677–2690, 1991.

[OKS80] Yu-Ichi Ohta, Takeo Kanade, and Toshiyuki Sakai. Color information for region segmentation. *Computer Vision, Graphics, and Image Processing*, 13 :222–241, 1980.

[Orc69] S. Orchard. Reflection and transmission of light by diffusing suspensions. *J. Opt. Soc. Am.*, 59 :1584–1597, 1969.

[Pae91] Alan Paeth. Mapping rgb triples onto 16 distinct values. In James Arvo, editor, *Graphics Gems II*, volume II, chapter 3, pages 143–146. Academic Press, 91.

[Poi81] M.R. Pointer. A comparison of the cie 1976 color spaces. *COLOR research and application*, 6(2) :108–118, 1981.

[Pos91] Jef Poskanzer. Pbm+ image processing software package, 1991.

[Pre70] J.M.S. Prewitt. *Picture Processing and Psychopictorics*, chapter Object Enhancement and Extraction, pages 75–149. Academic Press, London, 1970.

[Rei73] J. Reichmann. Determination of absorption and scattering coefficients for non homogeneous media. *Applied Optics*, 12 :1811–1815, 1973.

[SB85] M.H. Savoji and R.E. Burge. Note : on different methods based on the karhunen-loève expansion and used in image analysis. *Computer Vision, Graphics, and Image Processing*, 29 :259–269, 1985.

[SC92] Jun Shen and Serge Castan. An optimal linear operator for step edge detection. *Computer Vision, Graphics, and Image Processing*, 54(2) :112–133, march 1992.

[SH81] Rand Siegel and J. Howell. *Thermal Radiation heat transfer*. Mc Graw-Hill, New York, 1981.

[Sha85] S. A. Shafer. Using color to separate reflection components. *Color Research and Application*, 10(4) :210–218, 1985.

[Sha02] Gaurav Sharma, editor. *Digital Color Imaging Handbook*. Electrical and Applied Signal Processing. CRC Press, 2002.

[Tho91] S. W. Thomas. Efficient Inverse Color Map Computation. In J. Arvo, editor, *Graphics Gems II*, pages 116–125, 528–535. Academic Press Professional, Cambridge, MA, 1991.

[Tre93] A. Tremeau. *Contribution des modèles de la perception visuelle à l'analyse d'image couleur*. PhD thesis, Université Jean Monnet de Saint Etienne, Octobre 1993.

[VB95] O. Verevka and J. Buchanan. Local K-Means Algorithm for Color Image Quantization. *Graphics/Vision Interface '95*, May 1995.

[Wit84] A. P. Witkin. *Image Understanding*, chapter Scale Space Filtering : a new approach to multi-scale description, chapter 3, pages 79–95. Ablex, 1984.

[Wri41] W.D. Wright. The sensitivity of the eye to small colour differences. *Proc. Phys. Soc.*, 53(296) :93–112, 1941. Part 2.

[WS90] D. J. Williams and Mubarak Shah. Edge contours using multiple scales. *Computer Vision, Graphics, and Image Processing*, 51 :256–274, 1990.

[Wu92] Xiaoling Wu. Color quantization by dynamic programing and principal analysis. *ACM Transaction on Graphics*, 11(4) :348–372, 1992.

[WWP88] S.J. Wan, S.K.M Wong, and P. Prusinkiewicz. An algorithm for multidimensional data clustering. *ACM Trans. Math. Softw.*, 14(4) :153–162, 1988.

[WWP89] S.K.M. Wong, S.J. Wan, and P. Prusinkiewicz. Monochrome image quantization. In *Canadian conference on electrical and computer engineering*, pages 17–20, September 1989.

[WZ91] Xiaolin Wu and K. Zhang. A better tree-structured vector quantizer. In *Proceedings of the IEEE Data Compression Conference*, pages 392–401. IEEE Computer Society Press, 1991.

[Xia97] Zhigang Xiang. Color image quantization by minimizing the maximum intercluster distance. *ACM Transactions on Graphics*, 16(3) :260–276, July 1997.

[XJ94] Zhigang Xiang and Gregory Joy. Color image quantization by agglomerative clustering. *IEEE Computer Graphics and Applications*, 14(3) :44–48, May 1994.

[Zen86] S.D. Zenzo. A note on the gradient of a multi-image. *Computer vision, Graphics and Image Processing*, 33 :116–125, 1986.

[Zuc76] Steven W. Zucker. Region growing : Childhood and adolescence. *Computer Graphics, and Image Processing*, 5(3) :382–399, September 1976. ZUCKER76b.