



Traitement numérique des images

Notes de cours (C++)

Steven Pigeon

Professeur

Département de mathématiques,
informatique et génie
UQAR

Traitement numérique des images
Notes de cours (C++)

Traitement numérique des images

Notes de cours (C++)

par

Steven Pigeon, professeur

Département de mathématiques,
informatique et génie
UQAR

Steven Pigeon
Département de mathématiques, informatique, et génie
Université du Québec à Rimouski
Rimouski, Québec
G5L 3A1
steven_pigeon@uqar.ca

Le document a été typographié par l'auteur avec la police libre URW-Garamond par Mathdesign et le système L^AT_EX. Les illustrations, sauf indications contraires, sont de la main de l'auteur. Les autres illustrations sont libres de droits ou utilisées selon les dispositions de l'article 29 de la loi sur le droit d'auteur. La photo « *Inoxtubes* » de la couverture est © Steven Pigeon.

V 0.9 Juillet 2020

005.7xxx
QA76.xx 2018
ISBN 978-0-0000-0000-2

© 2020 Steven Pigeon

Tous droits de traduction totale ou partielle réservés pour tous les pays. La reproduction d'un extrait quelconque de ce livre, sauf selon les dispositions prévue par la loi, par quelque procédé que ce soit, tant électronique que mécanique, en particulier par photocopie, est interdite sans l'autorisation écrite de l'auteur.

Dernièrement, un ami [...] me disait ceci : « Le savoir, si tu te donnes complètement à lui, il te donne un peu. Si tu lui donnes un peu, il ne te donne rien ».

Ali Benmakhlof
À quoi sert le savoir ?

Table des matières

Table des matières	i
Table des figures	iii
Liste des tableaux	v
Table des notations	vii
Conventions typographiques	ix
1 Structures des images	1
1.1 Structure des images	1
1.2 Les pixels	1
1.2.1 Systèmes de couleurs	1
1.2.1.1 Monochromie et tons de gris	1
1.2.1.2 Dichromie	2
1.2.1.3 Trichromie	2
1.2.1.4 Quadrichromie	3
1.2.1.5 Multichromie	3
1.2.1.6 Couleurs hyperspectrales	3
1.2.1.7 Bande dynamique étendue et non-linéarité	5
1.2.2 Encodage de la couleur	7
1.2.2.1 <i>Direct Colors et Bit Depth</i>	7
1.2.2.2 Palettes	11
1.3 Organisation des pixels dans l'image	13
1.3.1 Résolutions	13
1.3.2 Organisation en mémoire	14
1.3.3 Sous-échantillonnage	17
1.4 Exemples de formats d'image	19
1.5 Remarques bibliographiques	21
2 Saisie des images	25
2.1 Introduction	25
2.2 Senseurs et couleurs	25
2.2.1 Technologies de senseurs	25
2.2.2 Prismes et filtres	27
2.2.3 Systèmes à un seul senseur	28
2.2.3.1 Bayer et ses amis	29
2.3 Remarques bibliographiques	31
3 Espaces de couleurs	33
3.1 Introduction	33
3.1.1 Des couleurs primaires	33
3.1.2 Familles d'espaces de couleurs	36

TABLE DES MATIÈRES

3.2	L'espace de couleurs CIÉ 1931	37
3.2.1	De l'origine du triangle des couleurs	38
3.2.2	L'espace des couleurs CIÉ comme un espace vectoriel	40
3.2.3	Contraintes imposées sur l'espace CIÉ 1931	41
3.2.4	CIÉ 1931 et les autres espaces de couleurs.	41
3.2.5	Conversions entre CIÉ XYZ 1931 et RGB	42
3.3	Les espaces de couleurs linéaires	42
3.3.1	Kodak 1	43
3.3.2	Kodak <i>Lossless</i>	45
3.3.3	Kodak <i>YCC</i>	46
3.3.4	Xerox <i>YES</i>	48
3.3.5	Espaces Ohta	48
3.3.6	$YC_o C_g$	50
3.3.7	JPEG 2000 : $YVU-R$	53
3.3.8	Série <i>S</i>	54
3.3.9	Les espaces de couleurs de la télévision	61
3.3.9.1	<i>Transmission primaries</i>	61
3.3.9.2	YUV et YIQ	63
3.3.9.3	Télévision numérique : BT.601, BT.709 et sRGB	65
3.3.9.4	$YC_b C_r$	67
3.3.9.5	$YP_b P_r$	69
3.3.9.6	$YD_b D_r$	70
3.3.9.7	$Y'C'_b C'_r$ alias $YC_b C_r$ HDTV	70
3.3.9.8	BT.2020 UHDTV	71
3.3.9.9	BT.2100 HDRTV	73
3.3.9.10	Un portrait de famille	76
3.3.10	<i>CMY</i> et <i>CMYK</i>	77
3.4	Les espaces de couleurs non linéaires	82
3.4.1	<i>HSV</i> et <i>HSL</i>	83
3.4.1.1	<i>HSV</i>	83
3.4.1.2	<i>HSL</i>	88
3.4.2	CIÉ $L^*a^*b^*$, $L^*u^*v^*$ et CIÉDE2000	91
3.4.2.1	$L^*a^*b^*$	92
3.4.2.2	$L^*u^*v^*$	94
3.4.2.3	CIÉDE2000	95
3.4.3	Autres espaces de couleurs	95
3.4.3.1	Munsell	95
3.4.3.2	Ostwald	96
3.4.3.3	NCS	98
3.5	Remarques bibliographiques	99
	Bibliographie	105
	Index	119

Table des figures

1.2.1	Dithering et halftoning	2
1.2.2	Trois couleurs	2
1.2.3	Tramage	4
1.2.4	Les doigts de la création	6
1.2.5	Correction gamma	6
1.2.6	Corrections plus souples	7
1.2.7	Contraste simultané	8
1.2.8	Faux contours	10
1.2.9	La réponse psychovisuelle, d'après [108].	10
1.2.10	Palettes CGA	12
1.2.11	Le roi Toutânkhamon par Avril Harrison (1986), version Amiga. Source : Retroshowcase [25].	12
1.3.1	L'organisation en plage contiguë	15
1.3.2	Organisation en plans de bits	15
1.3.3	L'organisation en plans avec transparence	16
1.3.4	Organisation Entrelacée	17
1.3.5	Sous-échantillonnage des couleurs	19
1.3.6	Des images sous-échantillonées en 4:2:0	19
1.5.1	Caractères semigraphiques	22
2.2.1	Rolling shutter	26
2.2.2	Prisme de type Philips	27
2.2.3	Prismes chroïques	28
2.2.4	Prisme dichroïque	28
2.2.5	Prisme trichroïque de caméra	29
2.2.6	Senseur « pleine couleur »	29
2.2.7	Senseur et microlentilles	29
2.2.8	Motifs de Bayer	30
2.2.9	Image saisie avec un motif de Bayer	30
3.1.1	Sensibilité de l'œil aux longueurs d'ondes	34
3.1.2	Triangle de couleurs primaires	35
3.1.3	Les cônes et les bâtonnets combinent leurs signaux pour donner la luminance, les différences rouge-vert et jaune-bleu [29].	37
3.2.1	Les fonctions colorimétriques CIÉ	38
3.2.2	Triangle de couleurs primaires CIÉ	39
3.2.3	Triangle $X\ Y\ Z$ de la CIÉ	40
3.2.4	Pyramide $X\ Y\ Z$ de la CIÉ	40
3.3.1	L'espace de couleurs Kodak-1	43
3.3.2	L'espace de couleurs Kodak <i>Lossless</i>	45
3.3.3	L'espace de couleurs Kodak <i>Lossless</i> , seconde variante	46
3.3.4	L'espace de couleurs Kodak <i>YCC</i>	47
3.3.5	L'espace de couleurs Xerox <i>YES</i>	49
3.3.6	Les axes principaux d'un patatoïde	49

3.3.7	Les espaces de couleurs Ohta	50
3.3.8	L'espace de couleurs YC_oC_g	51
3.3.9	L'espace de couleurs $YVU-R$	54
3.3.10	La construction de l'espace S_1	56
3.3.11	Les espaces de couleurs S_1 et S_2	56
3.3.12	Le cube RGB « compressé » par les poids ω_r , ω_g et ω_b .	58
3.3.13	L'espace de couleurs S_3	59
3.3.14	L'espace de couleurs S'_1	60
3.3.15	Les espaces de couleurs YUV et YIQ	64
3.3.16	L'espace de couleurs YC_bC_r	68
3.3.17	Les espaces de couleurs YC_bC_r et $Y'C'_bC'_r$ comparés.	72
3.3.18	L'espace de couleurs $Y'C'_{BC}C'_{RC}$	73
3.3.19	Une image en grande gamme dynamique (HDR).	74
3.3.20	L'espace de couleurs IC_TC_P transformant LMS transformant RGB , sans les corrections non linéaires.	76
3.3.21	La famille BT.	77
3.3.22	L'espace de couleurs $CMYK$, attentes et résultats!	78
3.3.23	L'espace de couleurs CMY	79
3.3.24	L'espace de couleurs $CMYK$ (les K sont montrés par translations).	81
3.4.1	Le cercle chromatique de l'espace HSV	84
3.4.2	Géométries de l'espace HSV	84
3.4.3	HSV et une projection du cube RGB .	85
3.4.4	La projection sur l'hexagone HSV .	85
3.4.5	Sur l'intervalle $[-\frac{1}{2}, \frac{1}{2}]$, $\tan^{-1}x \approx x$.	86
3.4.6	Les régions de couleurs dans l'hexagone HSV .	87
3.4.7	Une implémentation <i>Mathematica</i> des conversions de RGB à HSV .	88
3.4.8	Les variantes de l'espace de couleurs HSV .	88
3.4.9	Les variantes de l'espace de couleurs HSL .	89
3.4.10	HSL , dans le brevet de Bergstedt [43].	89
3.4.11	Une implémentation <i>Mathematica</i> des conversions de RGB à HSL .	91
3.4.12	Les ellipses de MacAdam (dessinées 10 fois plus grandes qu'en réel), d'après [170]. Source : Wikipedia.	92
3.4.13	La portion RGB de l'espace de couleurs $L^*a^*b^*$	93
3.4.14	La portion RGB de l'espace $L^*u^*v^*$	95
3.4.15	Le cercle chromatique de l'espace de couleurs de Munsell. Source : Wikipedia.	96
3.4.16	L'espace de couleurs de Munsell (couleurs réalisables en RGB).	96
3.4.17	L'arbre des couleurs de Munsell, tiré de [66] (domaine public).	97
3.4.18	Le cercle chromatique d'Ostwald	97
3.4.19	L'espace de couleurs Ostwald.	98
3.4.20	Les coordonnées Ostwald	98
3.4.21	L'espace de couleur NCS.	99
3.5.1	L'espace d'Ostwald et sa correction logarithmique.	102

Liste des tableaux

1.2.1	Longueurs d'ondes et phénomènes physiques	5
1.2.2	Quelques exemples d'encodage de couleurs	8
1.3.1	Quelques résolutions communes	13
1.3.2	Quelques résolutions communes pour la télévision	13
1.3.3	CIF et SIF	14

LISTE DES TABLEAUX

Table des notations

\mathbb{N}	Les nombres naturels, $\{1, 2, 3, \dots\}$
\mathbb{N}_0	Les naturels incluant le zéro, $\{0, 1, 2, 3, \dots\}$
\mathbb{Z}	Les entiers, $\{\dots, -3, -2, -1, 0, 1, 2, 3, \dots\}$
\mathbb{Z}^*	Les entiers non négatifs, équivalent à \mathbb{N}_0
\mathbb{Z}^+	Les entiers positifs, équivalent à \mathbb{N}
\mathbb{Q}, \mathbb{Q}'	Les rationnels, les irrationnels
\mathbb{R}	Les réels
\mathbb{R}^*	Les réels non négatifs
\mathbb{R}^+	Les réels (strictement) positifs
$ x $	Valeur absolue
$[x]$	Le <i>plancher</i> de x : le plus grand entier plus petit ou égal à x
$\lceil x \rceil$	Le <i>plafond</i> de x : le plus petit entier plus grand ou égal à x
$\{x_i\}_{i=a}^b$	La séquence $\{x_a, x_{a+1}, \dots, x_{b-1}, x_b\}$
\perp	Séquence vide, mot vide
$ A $	Cardinalité de l'ensemble A , longueur de l'objet A
$A \times B$	Si A et B sont des ensembles, produit cartésien
A^n	Si A est un ensemble, la puissance cartésienne
A^*	$\{\perp\} \cup A \cup A^2 \cup A^3 \cup \dots$
$A \cup B$	Union de A et B
$A \cap B$	Intersection de A et B
$A \subseteq B, A \subset B$	Inclusion, inclusion stricte
$A^c, \complement_U A$	Complément de A , complément de A dans U
$P(S)$	L'ensemble de toutes les parties d'un ensemble
$\ln x$	Logarithme naturel (base e) de x
$\log_{10} x$	Logarithme à base 10 de x
$\lg x$	Logarithme à base 2 de x
$\log x$	Logarithme à base indifférente

$\exists, \exists!, \nexists$	Quantificateurs existentiels : il existe, il n'existe qu'un seul, il n'existe aucun
\forall	Quantificateur universel : pour tous
$a \ll b$	a est beaucoup plus petit que b
$a \gg b$	a est beaucoup plus grand que b
$a \sim b$	a est de la forme b , a est semblable à b
$\binom{n}{m}$	Nombre de façons de choisir m parmi n , $\binom{n}{m} = \frac{n!}{m!(n-m)!}$
$\binom{n}{n_1, n_2, \dots, n_k}$	Nombre de façons de choisir n_1, n_2, \dots, n_k symboles indistinguables parmi n
ssi	Si, et seulement si.
γ	La constante d'Euler-Mascheroni, $\gamma = 0.577215664901532\dots$
e	La constante d'Euler, $e = 2.718281828459045\dots$
π	La constante d'Archimède, $\pi = 3.141592653589793\dots$

Conventions typographiques

Le texte normal est typographié avec la police libre URW-Garamond par Mathdesign. Les programmes et les éléments de programmation sont typographiés avec la police ASCII de Syropoulos et Nickalls, qui rappelle une police de terminal. Les éléments ordinaires sont typographiés ainsi alors que les mots-clefs du langage sont typographiés en gras, comme pour **while**, **if**, **for**, etc.

Les librairies utilisées par les programmes sont tirés de la librairie standard. Par convention, les librairies introduites par `#include <librairie>` font partie du système et sont en principe disponibles en même temps que le compilateur C++. Les librairies introduites par des guillemets, comme par exemple `#include "librairie"` sont les librairies fournies par l'utilisateur, c'est-à-dire par le programme plutôt que par la librairie standard.

Les références, notées par exemple [215], vous amènent à la fin de l'ouvrage où elles sont présentées en ordre alphabétique d'auteurs, et non en ordre d'apparition.

Les démonstrations se trouvent presque toujours dans un encadré intitulé comme tel. Pour une première lecture ou pour un cours moins avancé, ces démonstrations peuvent être omises, mais nous invitons tout de même le lecteur à s'y attarder. Comme pour les démonstrations, les exemples sont encadrées.

1

Structure des images

Sommaire. Ce chapitre présente les différentes organisations des images numériques, du pixel à la structure de fichiers.

1.1 Structure des images

1.2 Les pixels

1.2.1 Systèmes de couleurs

1.2.1.1 Monochromie et tons de gris

Les images monochromes (étymologiquement « une couleur ») sont en fait toujours composées de deux couleurs, dont une fait office d'arrière-plan et l'autre d'avant-plan. Typiquement, ces deux couleurs sont le noir et le blanc, mais il peut s'agir d'autres couleurs — comme deux tons de vert épinard ou d'« ambre » ! Comme chaque pixel ne peut prendre que deux couleurs, les tons dégradés nécessaires au rendu agréable d'images naturelles sont simulés par diffusion d'erreur (*dithering*) ou par tramage (*halftone*). Ces techniques pour simuler les tons de gris sont montrées à la fig. 1.2.1.

Lorsque l'image est en tons de gris (*grayscale*) nous trouvons n (souvent $n = 2^k$) tons également répartis du noir au blanc. Si le nombre de tons est suffisamment grand, nous pouvons obtenir l'illusion d'une image naturelle en noir et blanc.

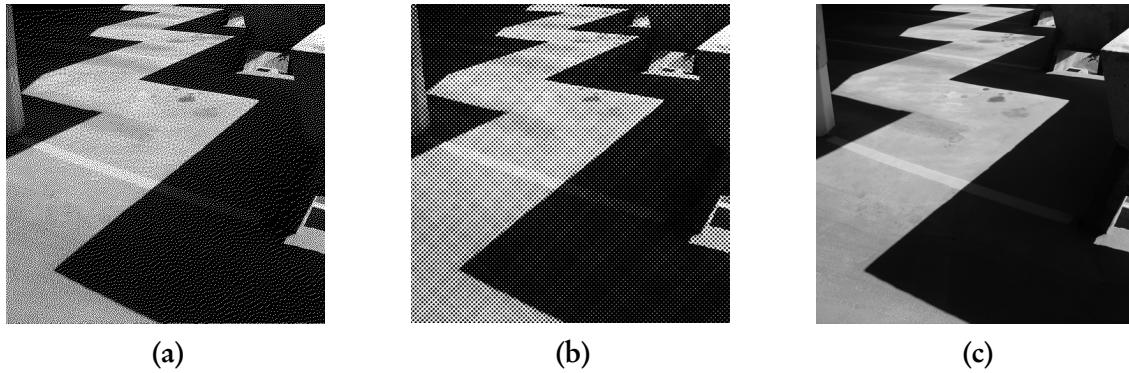


Figure 1.2.1 — Diffusion d'erreur (a) et tramage (b). En (c), l'image en tons de gris.

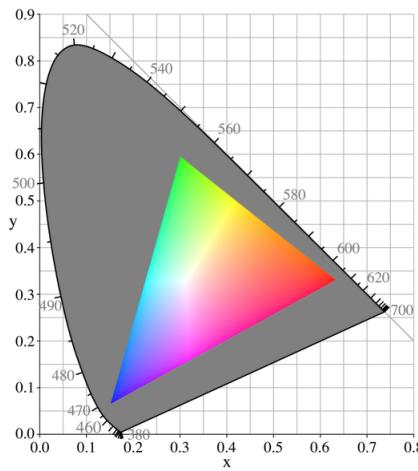


Figure 1.2.2 — La plage de couleurs réalisable à partir de trois couleurs primaires. Source : Wikipedia.

1.2.1.2 Dichromie

Nous pourrions imaginer des systèmes de couleurs avec deux couleurs primaires, mais il semble que ces systèmes s'avèrent insuffisants pour représenter des couleurs qui paraissent naturelles et convaincantes. Bien qu'il soit facile de simuler un système dichrome avec un matériel trichrome, il semble qu'il n'existe pas de système fondamentalement dichrome.

1.2.1.3 Trichromie

Il semble qu'il ne soit nécessaire que de trois composantes pour tromper l'œil et lui faire voir une plage intéressante de couleurs. Si elles sont judicieusement choisies, ces composantes peuvent être mélangées de façon à former un grand nombre de couleur, et peut-être un nombre assez vaste pour permettre une illusion convaincante d'avoir de « vraies couleurs ». La fig. 1.2.2 montre, en gris, toutes les couleurs perceptibles pour un humain typique, tandis que les couleurs simulées se retrouvent dans le triangle coloré.

Typiquement — mais ni forcément, ni tout le temps — les images sont construites à partir de trois couleurs primitives : le rouge, le vert et le bleu. Les longueurs d'ondes des couleurs primitives dépendent de la technologie (comme la télévision analogique) ou des standards (pour le cinéma 8K, par exemple).

1.2.1.4 Quadrichromie

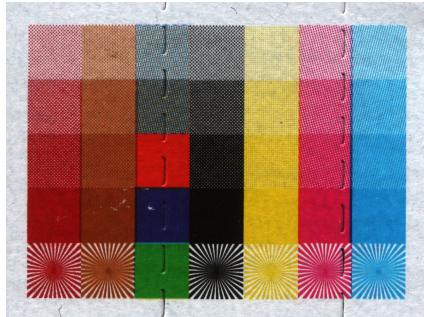
Certains systèmes proposent une quatrième couleur primaire, comme par exemple le jaune ajouté aux primaires rouge, verte et bleue. Cela a pour effet d'accroître le nombre de couleurs représentables et le triangle des couleurs réalisables (c.f. fig. 1.2.2) devient un quadrilatère. Le nombre de couleurs nouvellement représentables variera en fonction de la nouvelle composante de couleur, mais on peut tout aussi bien accroître le nombre de couleurs représentables en choisissant mieux nos trois composantes primaires — peut-être même au point d'inclure la quatrième composante primaire.

1.2.1.5 Multichromie

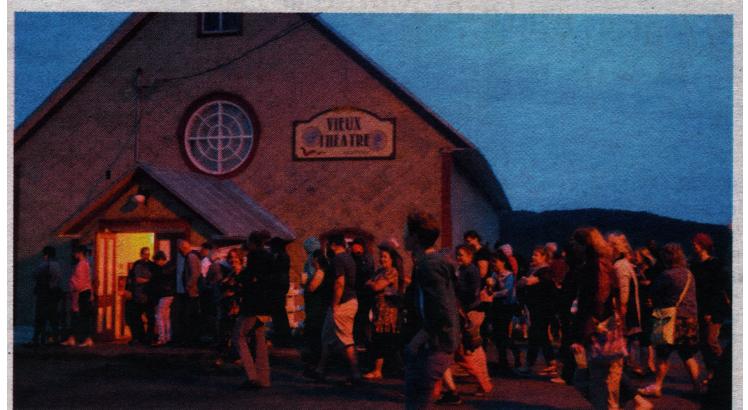
Certains procédés utilisent un grand nombre de « couleurs pures » pour le rendu des images. C'est notamment le cas en imprimerie où, s'il est toujours possible de simuler les couleurs par tramage de quatre couleurs de base (magenta, jaune, cyan et noir, comme on le voit à la fig. 1.2.3 (a)), il est souvent préférable d'utiliser des couleurs exactes pré-mélangées. Avec des couleurs pures, les effets de tramage disparaissent et il est possible de représenter des couleurs impossibles à rendre à partir des couleurs de base choisies. Cependant, les exemples de la fig. 1.2.3 montrent que le rendu par tramage peut être satisfaisant.

1.2.1.6 Couleurs hyperspectrales

Pour les images hyperspectrales, les couleurs ne sont plus représentées par seulement trois composantes mais par un grand nombre de composantes, chacune correspondant à une longueur d'onde spécifique. Les images hyperspectrales sont souvent le résultat d'un passage au spectromètre, où chaque pixel est décomposé en un « arc-en-ciel » de quelques dizaines ou quelques centaines de longueurs d'onde. Lorsqu'on a une petite dizaine de longueurs d'ondes, on parlera alors d'image *multispectrale*; bien que la distinction entre hyperspectrale et multispectrale soit floue et varie d'un auteur à l'autre.

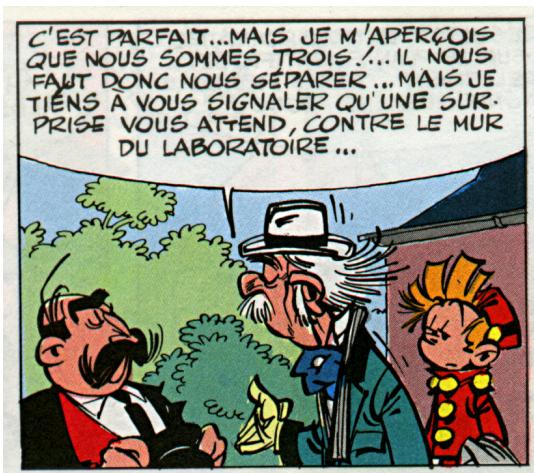


(a)

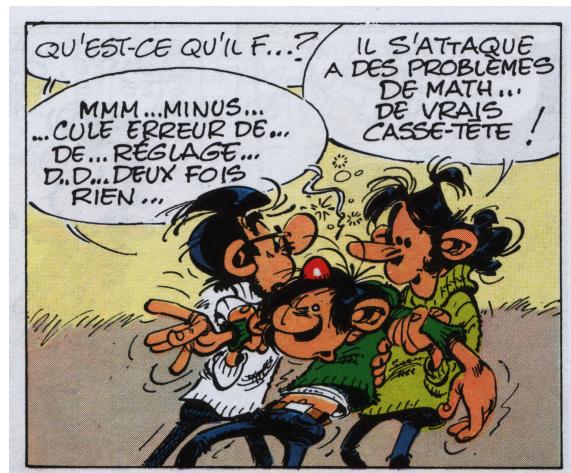


La qualité de la direction artistique du Vieux Théâtre de Saint-Fabien est reconnue par RIDEAU. (Photo gracieuseté)

(b)



(c)



(d)

Figure 1.2.3 — Technique du tramage. a) À gauche, un étalon trouvé sur une boîte de poulet surgelé. b) À droite, une photographie rendue par tramage (source : *L'avantage*, vol 24 n° 34, p.13). c) Tramage d'une image tirée de Spirou, d) de Gaston Lagaffe [100, p. 7].

Les bandes de fréquences¹ peuvent couvrir le spectre visible (400nm, violet, à 700nm, rouge profond) ou un spectre bien plus étendu, depuis l'« infrarouge profond » (*deep infrared*) jusqu'aux ultraviolets ou même les rayons X! La table 1.2.1 présente quelques bandes de longueurs d'ondes et leur correspondance physique.

Ces images jouent un rôle important en télédétection (soit à partir d'un avion, soit un satellite), et permettent de segmenter les images selon les différentes longueurs d'ondes. Il est ainsi bien plus

1. Longueur d'onde et fréquence sont liées. À une longueur d'onde λ correspond une fréquence $f = c/\lambda$ et inversement $\lambda = c/f$. Les fréquences sont exprimées en Hz, les longueurs d'ondes en nanomètres.

Nom	Longueurs d'onde	Application
Rayons X Ultra-violet	0.03nm–3nm 10nm–400nm	Radiographie Spectroscopie, photolithographie
Violets	380 ou 400nm–450nm	?
Bleus	450nm–520nm	Atmosphère, eaux (moins de 50m)
Verts/Jaune	515 ou 520nm–600nm	Végétation, eaux (moins de 30m)
Jaune/Rouges	600nm–690nm	Structures artificielles, eaux (moins de 10m)
Proche infrarouge	750nm–900nm	Végétation
Moyen infrarouge	1550nm–1750nm	Végétation, humidité des sols, feux de forêt
Infrarouge lointain	2080nm–2350nm	Végétation, humidité des sols, feux de forêt, silicates, argiles, feux
Infrarouge thermique	10000nm–12500nm	Chaleur, température des eaux, feux, « vision nocturne »

Table 1.2.1 — Longueurs d'ondes et phénomènes physiques.

facile de détecter les plans d'eau, d'estimer leur profondeur, la couverture végétale, la quantité d'eau contenue dans celle-ci... voire détecter les vilaines cultures illégales.

Si nous avons des bandes qui correspondent au spectre visible, on pourra toujours reconstruire une image visible en trois composantes en utilisant, par exemple, la sensibilité de l'œil aux différentes longueurs d'ondes¹. Cette sensibilité est montrée à la fig. 3.1.1.

Par contre, les images multi- ou hyperspectrale sont souvent rendues en *fausse couleurs*, où chaque bande reçoit une couleur (visible) aléatoire ou encore codifiée — jaune pour le sable, bleu pour l'eau, vert pour la végétation, gris pour les argiles, etc. Si les couleurs sont assignées en fonction d'une échelle, comme par exemple un gradient du bleu au rouge pour la température, nous parleront alors de *pseudo-couleurs*.

Par exemple, les superbes photographies multicolores de nébuleuses lointaines (voir fig. 1.2.4) sont composées en pseudo-couleurs en fonction de leur composition atomique : bleu-vert pour l'oxygène, rouge pour l'hydrogène, etc. [37, 71, 227, 257].

1.2.1.7 Bande dynamique étendue et non-linéarité

La réponse du médium peut être linéaire ou non. Si la réponse est linéaire (d'aucuns pourraient penser qu'il s'agit d'une réponse idéale), l'intensité rendue sera identique à l'entrée, peut-être fois une constante de proportionnalité, c'est-à-dire quelque chose comme $f(x) = \alpha x$. Il est souvent pratique de supposer que c'est le cas.

1. . Notons par ailleurs que la couleur est *perçue* par notre cerveau, et ne correspond en rien à des caractéristiques *a priori* physiques.



Figure 1.2.4 — Image composée de trois bandes : O-III (502 nm), H- α (657 nm), S-II (673 nm). Source : NASA, ESA/Hubble and the Hubble Heritage Team.

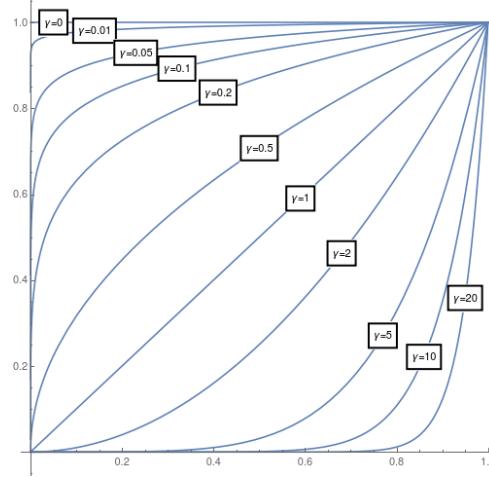


Figure 1.2.5 — Correction gamma.

Si la réponse est non-linéaire (et c'est presque toujours le cas), elle pourra atténuer ou accentuer certains niveaux, ou transformer toutes les valeurs. Dans les cas les plus simples, on parlera d'une fonction ou d'une *correction gamma* qui prend la forme $f(x) = ax^\gamma$. Les grandes valeurs de γ augmentent le contraste mais obscurcissent l'image; les petites adoucissent le contraste et rendent l'image plus claire. Les corrections résultant de plusieurs valeurs de γ sont montrées à la fig. 1.2.5.

La correction gamma est quand même assez peu expressive car les formes qu'elle produit sont peu variées. D'autres lui préféreront des fonctions dites sigmoïdes (« en forme de s »). Nous en trouvons deux à la fig. 1.2.6. Ces fonctions sont utiles pour « améliorer » le contraste en obscurcissant davantage les couleurs sombres et en augmentant les couleurs claires (comme on le voit à la fig. 1.2.6 (a)).

Nous pouvons aussi utiliser ces fonctions pour simuler la réponse de l'œil aux scènes fortement contrastées. Contrairement à ce que nous pourrions penser, nous ne percevons pas une scène dans

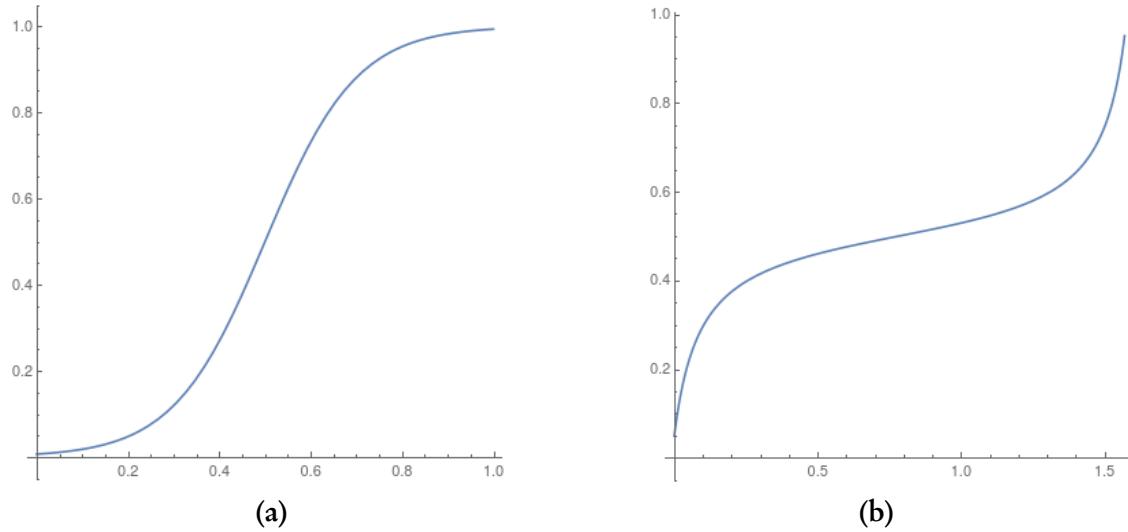


Figure 1.2.6 — Corrections plus souples. (a) À gauche, une fonction de contraste. (b) À droite, correction de bande dynamique.

sa totalité d'un seul regard : il nous faut déplacer notre attention sur ses différents éléments pour avoir l'impression de la percevoir dans son ensemble. Lorsque nous portons notre attention sur une (petite) région de la scène, nous pouvons voir clairement des éléments dont la luminosité relative ne diffère que d'environ deux ordres de grandeurs (soit 100:1), qu'il s'agisse d'ombres ou de lumières. Si le contraste dépasse (localement) ce ratio, l'ombre nous paraît impénétrable. La fig. 1.2.7 (a) nous montre une de ces images. À la fig. 1.2.7 (b), nous avons la même image, mais cette fois-ci, nous avons appliqué la fonction de la fig. 1.2.6 (b). Nous retrouvons l'effet perceptuel sans pour autant augmenter le ratio de contraste maximal de l'écran. Notons, de plus, que cette correction n'est pas nécessairement appliquée en logiciel : l'écran peut très bien prendre en charge cette correction.

1.2.2 Encodage de la couleur

1.2.2.1 Direct Colors et Bit Depth

Lorsque l'encodage d'un pixel donne directement les composantes de couleur, on dira que l'encodage est de type *direct color*. Si l'encodage réfère à une table, alors nous parlerons d'encodage par palette — nous en discuterons en détail dans la prochaine section.

Si l'encodage est de type direct, alors un certain nombre de bits sont dévolus à l'encodage de la composante rouge, de la composante verte et de la composante bleue. Le nombre de bits utilisés pour encoder la couleur est le *bit depth*, ou encore le *color depth* (traduit assez naïvement par *profondeur de couleur*). Si, pour arriver à un alignement compatible avec le matériel (typiquement, le nombre de bits est arrondi au prochain octet), l'encodage laisse quelques bits disponibles, ils peuvent encoder

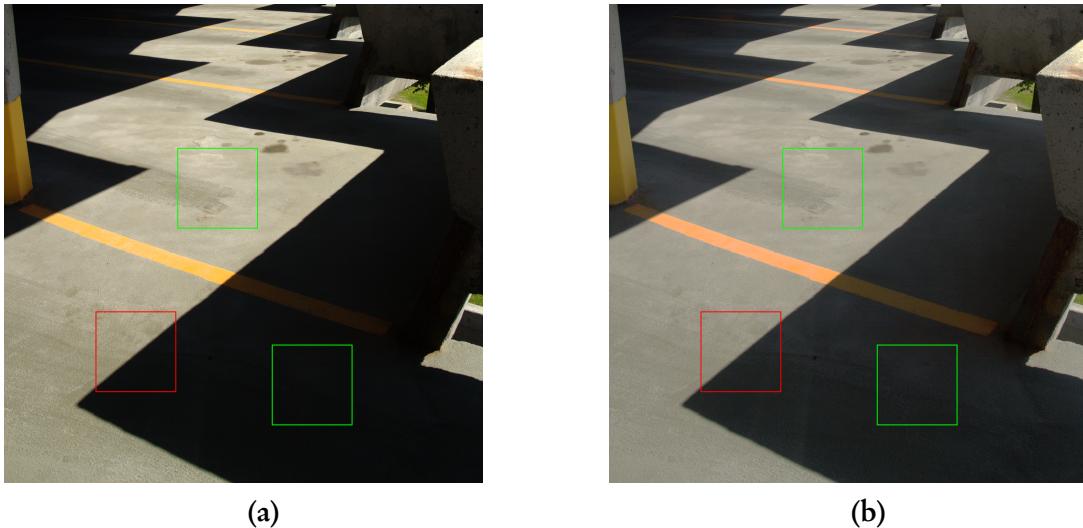


Figure 1.2.7 — Contraste simultané. (a) À gauche. En vert, les régions où l'œil perçoit bien les détails. En rouge, une région où l'ombre semble impénétrable. (b) À droite. La même scène corrigée par la fonction de la fig. 1.2.6 b).

Bits	Mode	Exemple	
3/4	D	u r g b	
6/7	D	B r g b b I r g f b f	CGA avant- et arrière-plan (I pour intensité, B pour <i>blink</i>) [68, 69]
6	D	u u r r g g b b	EGA
9	P	u u u u u r r r u g g g u b b b	Atari ST [211, p 2-13]
12	P	u u u u r r r r g g g g b b b b	Amiga 1000 [210, p 2-13]
15	D	u r r r r r g g g g b b b b b	SVGA
16	D	u r r r r r g g g g g b b b b b	SVGA
18	P	u u r r r r r u u g g g g g g u u b b b b b	VGA
24	P	r r r r r r r g g g g g g g g b b b b b b b	VESA

Table 1.2.2 — Quelques exemples d'encodage de couleurs. Légende : u (inutilisé), r,g,b, bits qui participent respectivement aux composantes rouges, vertes, bleues. P indique un mode palette, D un mode direct.

une quatrième composante, par exemple la transparence (universellement nommée α , probablement depuis 1984 [216], car avant, il semble qu'il ne soit question que de transparence [263]), ou encore un effet autre, comme la demi-brillance [157, p. 258] ou le clignotement. La table 1.2.2 montre quelques exemples d'encodage de couleur.

*

La table 1.2.2 montre des encodages où le nombre de niveaux par composante est une puissance de deux. Cela est le résultat direct de l'utilisation d'un nombre entiers de bits pour encoder chaque composante. Ainsi, nous avons 2, 8, 16, 256, etc., niveaux, mais nous ne sommes pas contraints aux

puissances de deux. Rien ne nous empêche d'avoir 19 ou 42 niveaux par composantes, ou même des nombres de niveaux différents d'une composante à l'autre !

La *web safe palette* (alias *Netscape Color Cube*) est un exemple d'un tel encodage. Chaque composante varie selon 6 niveaux, pour un total de $6 \times 6 \times 6 = 216$ couleurs. À l'époque où la plupart des cartes graphiques ne pouvaient que rendre 256 couleurs en même temps en haute résolution (souvent via un mode palette), les applications ne disposaient pas d'un grand nombre de couleurs libres, car certaines étaient réservées et fixes pour assurer que l'interface graphique demeure lisible en tout temps. Heureusement, les couleurs réservées pour l'interface étaient les 16 ou 32 premières couleurs, laissant ainsi les autres aux applications qui pouvaient les reprogrammer — et nous offrir un camaïeu de couleurs farfelues lorsque qu'on passait d'une application à l'autre.

La palette de $6 \times 6 \times 6 = 216$ couleurs n'utilise que 216 couleurs parmi les 256 couleurs possibles lorsqu'on utilise un code sur 8 bits — car, clairement, 7 bits seraient insuffisants pour représenter toutes les combinaisons. Comme un code optimal devrait contenir *tout juste assez* de bits pour représenter ces 216 combinaisons, $\log_2 216 \approx 7.75$ bits devraient être suffisants ! En utilisant 8 bits, nous « gaspillons » donc ≈ 0.25 bits par rapport à un code optimal ! Si nous voulons mieux utiliser nos 8 bits, il nous faut une palette avec plus de couleurs — idéalement 256. Or, Une variante de la palette *web safe*, présente $6 \times 7 \times 6 = 252$ couleurs, avec un niveau supplémentaire pour le vert. Cette palette s'approche plus de 256 couleurs que la palette $6 \times 6 \times 6$, et en ce sens, utilise beaucoup mieux les 8 bits du code de couleur. En effet, nous trouvons $\log_2 252 \approx 7.98$ bits!¹

*
* * *

Comme nous venons de le faire remarquer, le nombre de niveaux par composantes n'est pas nécessairement contraint. Souvent il est choisi pour exploiter efficacement le matériel sous-jacent (par exemple, un octet par composante), mais le matériel évolue constamment comme son coût de production chute constamment. Ainsi, nous voyons émerger des systèmes avec 10, 12, voire 16 bits par composantes, nous donnant des pixels sur 30, 36, voire 48 bits (on parlera alors de modes *deep color*). Cependant, même si en principe le matériel rend les couleurs sur plus de 24 bits, il est quand même encore hautement probable que le rendu final par l'écran soit bien en deçà des capacités théoriques du matériel — un œil exercé verra les limitations de l'écran où les tons en principe continus sont rendus par des couleurs discrétisées, faisant apparaître de « faux contours », comme à la fig. 1.2.8.

1. On peut d'ailleurs interpréter ces codes comme étant des codes qui utilisent des *fractions* de bits. Nous y reviendrons lorsque nous discuterons de la compression de données appliquée aux images.

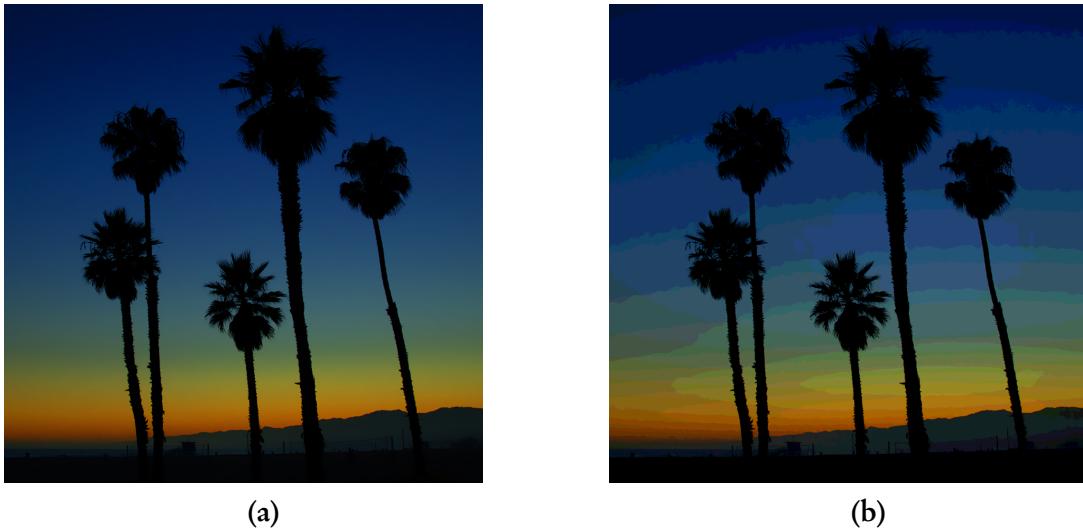


Figure 1.2.8 — Effet de faux contours dû à l'insuffisance de résolution de couleurs. (a) image originale, (b) image avec 16 niveaux par composante (4096 couleurs, sans diffusion d'erreur).

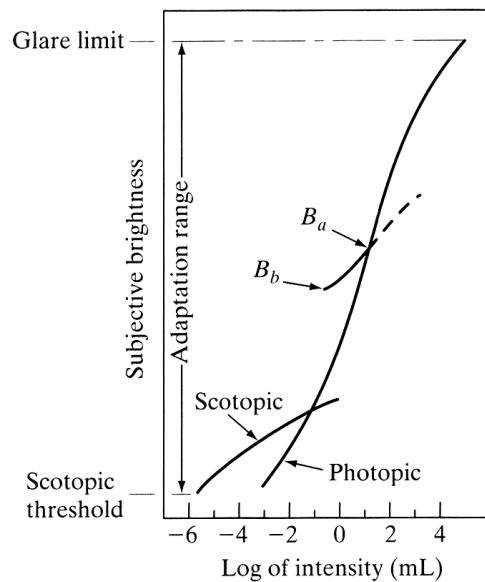


Figure 1.2.9 — La réponse psychovisuelle, d'après [108].

Mais alors, combien de bits seraient nécessaires pour un rendu parfait de n'importe quelle image? Pour répondre adéquatement, il faut d'abord comprendre comment l'œil réagit à l'intensité lumineuse. Certains estiment que cette bande dynamique s'étend sur 10 ordres de grandeur, du seuil de la perception jusqu'à l'éblouissement [108], d'autres plutôt sur 16 [29, 89]. La fig. 1.2.9 montre la

courbe de réponse de l'œil à l'intensité lumineuse. En lumière normale, ce sont les cônes qui forment la vision en couleur (la vision photopique), tandis qu'en obscurité, ce sont les bâtonnets qui forment l'image (la vision scotopique). Pour représenter les intensités lumineuses avec précision sur D ordres de grandeur, il nous faut

$$\log_2 k 10^D = \log_2 k + D \log_2 10 \approx \log_2 k + 3.322D \quad (1.2.1)$$

bits. Ici, k est une constante de finesse qui peut être ajustée en fonction de la différence minimale perceptible entre deux intensités. Si on suppose $D = 10$, on trouvera

$$\log_2 k 10^{10} \approx \log_2 k + 33.22$$

bits, tandis qu'avec $D = 16$,

$$\log_2 k 10^{16} \approx \log_2 k + 53.15$$

bits. Cela semble indiquer qu'il nous faille utiliser un grand nombre de bits pour représenter une image avec la même précision qu'avec laquelle l'œil la perçoit. Revenons à la fig. 1.2.9. Au milieu de la figure, nous avons un point indiqué B_a , qui est, pour une scène donnée, l'éclairage moyen. Autour de cette éclairage moyen, l'adaptation de l'œil (qui ne se fait qu'après un moment) ne permet de résoudre qu'une certaine bande dynamique dont la marge inférieure est notée B_b dans la figure (et, par symétrie, une marge supérieure indiquée par le pointillé). Les luminosités plus faibles que B_b sont perçues comme étant noires, et les luminosités plus grandes que son point symétrique comme « blanc », c'est-à-dire une luminosité d'où il nous est impossible de résoudre les détails.

Cette bande dynamique d'adaptation est bien plus restreinte que la bande dynamique de toutes les intensités visibles, que de toute façon, nous ne pouvons pas voir simultanément en entier. Cela nous donne donc l'idée de choisir judicieusement un B_a fixe, satisfaisant pour la télévision, ou pour l'infographie, et ajuster la bande dynamique entre B_b et le point symétrique B_c . La figure nous donne 4 ou 5 ordres de grandeur pour cette bande dynamique, ce qui, en utilisant l'éq. (1.2.1), nous donne $\approx \log_2 k + 16$ bits. En posant $k = 1$, nous arrivons à ≈ 16 bits par composante¹.

1.2.2.2 Palettes

Le mode palette est un compromis entre la mémoire utilisée pour stocker l'image et le nombre de couleurs possibles. Chaque pixel, plutôt que coder directement la couleur à afficher, devient un index dans une table de couleurs. Ainsi, on peut avoir des index assez courts (un octet, ou moins) qui pointent dans une table de couleurs riches (16 bits ou plus). Ce compromis permet de réduire

1. C'est encore un peu simplifier ; la luminosité d'une couleur est une fonction de toutes ses composantes, en plus des corrections non-linéaires qui ont pu y être apportées. Nous reviendrons sur le rôle des composantes dans la luminosité au chapitre 3, sur les espaces de couleurs.

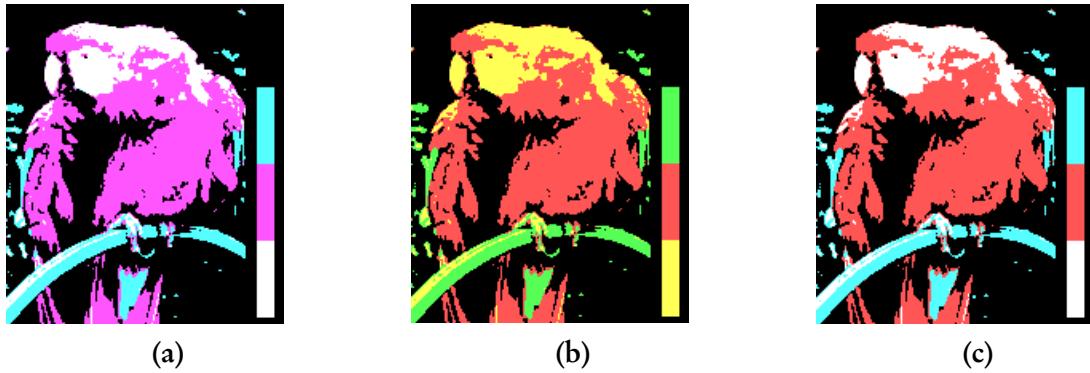


Figure 1.2.10 — Palettes CGA. (a) mode 4, palette 1, (b) mode 4, palette 2 et (c), mode 5. Source : Wikipedia.



Figure 1.2.11 — Le roi Toutânkhamon par Avril Harrison (1986), version Amiga. Source : Retroshowcase [25].

l'espace mémoire nécessaire pour retenir l'image (et aussi la largeur du bus nécessaire pour y accéder) sans sacrifier trop de l'expressivité de l'image.

La palette peut être sous le contrôle d'un programme ou être fixée par le matériel. Ainsi, les modes graphiques CGA 320×200 offraient des modes à quatre couleurs, dont une seule programmable (noir par défaut); le choix du mode déterminait les 3 autres couleurs (en deux modes : faible et haute intensité) [68, 69]. Heureusement, d'autres matériels étaient capables de bien mieux! Par exemple, l'Amiga 1000 était capable de modes avec une palette de 32 couleurs choisies parmi 4096. La fig. 1.2.11 montre une telle image. Comme changer les couleurs contenues dans la palette n'implique pas forcément de recoder l'image, le mode palette permet de réaliser des animations (simples) à peu de frais : changer cycliquement certaines couleurs peut donner des effets de chatoiement ou encore donner l'illusion du mouvement.

Résolution	Sigle	Aspect	Nom
320×200	CGA	16:10	Color Graphics Adapter
320×240	QVGA	4:3	Quarter VGA
640×480	VGA	4:3	Video Graphics Array
720×358	Hercules	4:3	Monochrome
800×600	SVGA	4:3	Super VGA
1024×768	XGA	4:3	Extended Graphics Array
1280×720	HD	16:9	HDTV
1280×1024	SXGA	5:4	Super XGA
1440×900	WSXGA	16:10	Wide SXGA
1600×1200	UXGA	4:3	Ultra SXGA
1920×1080	HD 1080	16:9	Full HDTV
2048×1536	QXGA	4:3	Quad XGA
2560×1440	QHD	16:10	Quad HD
3840×2160	UHD	16:9	Ultra HD, 4K
7680×4320	UHD2	16:9	Ultra HD 2, 8K

Table 1.3.1 — Quelques résolutions communes

Résolution	Sigle	Aspect	Modes
520×576	PAL/SÉCAM	4:3	Entrelacé
440×486	NTSC	4:3	Entrelacé
720×480	NTSC	4:3	Entrelacé (480i)
1280×720	HD	16:9	HDTV, entrelacé (720i) ou progressif (720p)
1920×1080	HD 1080	16:9	Full HD, entrelacé (1080i) ou progressif (1080p)
3840×2160	UHD	16:9	Ultra HD, 4K, progressif
7680×4320	UHD2	16:9	Ultra HD 2, 8K, progressif

Table 1.3.2 — Quelques résolutions communes pour la télévision

1.3 Organisation des pixels dans l'image

1.3.1 Résolutions

Nous entendons parfois *résolution* et *définition* utilisés de façon indifférente pour parler du nombre de pixels qui forment une image. Cependant, la *définition* d'une image désigne plutôt la qualité (précision) de chaque pixel, tandis que la *résolution* en détermine le nombre.

Les résolutions typiques ont évolué au gré des technologies. La table 1.3.1 montre les résolutions d'écran communes (ou qui l'ont été). Il en existe évidemment beaucoup d'autres, dictées par les dimensions physiques des écrans, des technologies du moment, des lignes de produits — Apple, par

Résolution	Sigle	Nom
128×96	SQCIF	Sub QCIF
176×144	QCIF	Quarter CIF
352×240	SIF	Standard Input Format
352×288	CIF	Common Intermediate Format
704×480	4SIF	Quad SIF
704×576	4CIF	Quad CIF
1408×1152	16CIF	

Table 1.3.3 — CIF et SIF

exemple, offre des écrans de 1366×768 pixels sur certains de ses modèles. Pour la télévision, numérique comme analogique, les résolutions sont fortement standardisées pour assurer l'interopérabilité des signaux [40, 92, 93, 107, 113, 219]. La table 1.3.2 montre les résolutions pour la télévision. Notons que certaines résolutions sont *entrelacées* : pour des raisons techniques liées à la bande passante limitée, ces résolutions affichent en alternance les lignes impaires d'une image puis les lignes paires de l'image suivante. Cela permet de doubler le nombre d'images *perçues* par seconde sans doubler la bande passante, camouflant ainsi une partie des saccades de la vidéo.

L'interopérabilité était un souci : les différents standards de télévision n'étaient pas tous exactement compatibles entre eux. Le nombre d'images par seconde variait d'une région à l'autre (50 ou 60, selon) ; ce qui demande des adaptations de la vidéo grâce à une gymnastique compliquée d'interpolation et de désentrelacement. Comme si cela n'était pas déjà assez compliqué, les résolutions différaient aussi, nécessitant, parfois, des formats intermédiaires facilement convertibles à l'une ou l'autre résolution standard. Le format CIF (*Common Intermediate Format*), défini par la norme H.261, tente de résoudre ce problème [3]. La table 1.3.3 montre les résolutions proposées par le standard CIF.

Si les écrans — et la télévision — ont plutôt tendance à imposer des résolutions particulières, les images, comme les vidéos, peuvent utiliser n'importe quelle résolution. L'ordinateur s'accommode fort bien des images minuscules ou immenses, et la plupart des *media players* sont indifférents à la dimension de l'image vidéo, se contentant de les redimensionner en temps réel au besoin.

1.3.2 Organisation en mémoire

L'organisation la plus naturelle est de placer les pixels les uns à la file des autres dans une plage de mémoire contiguë et sans sauts. La fig. 1.3.1 montre schématiquement cet arrangement. Traditionnellement, le pixel en haut à gauche reçoit l'adresse (relative) 0. Cela facilite le calcul

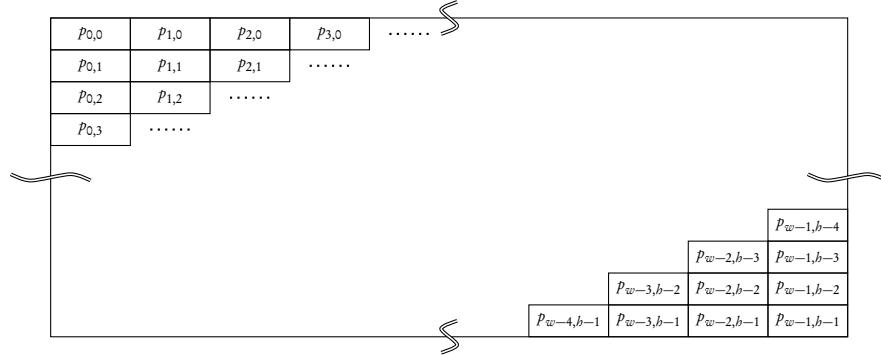


Figure 1.3.1 — L'organisation en plage contiguë.

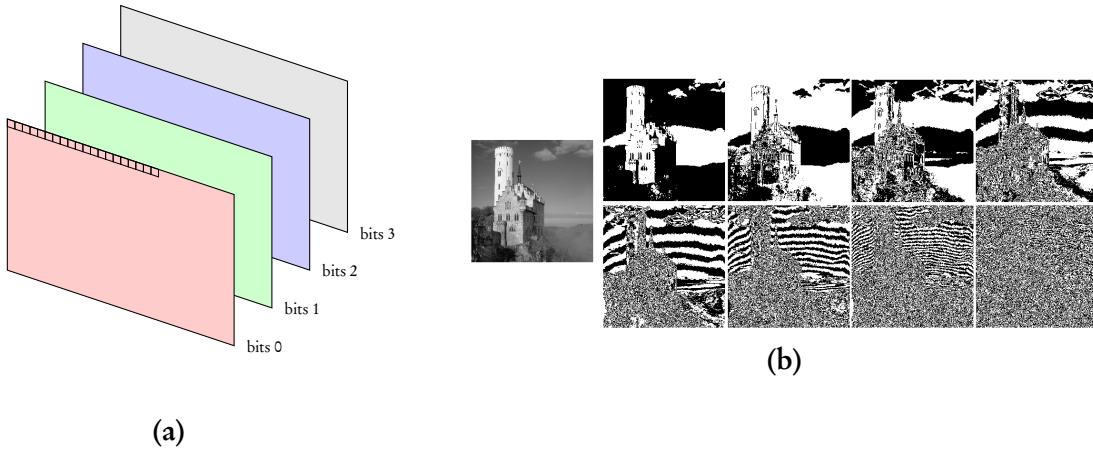


Figure 1.3.2 — Organisation en plans de bits des pixels en mémoire. (a) Plans de bits, (b) Image décomposée en plans de bits (Source : Wikipedia).

d'adresse à l'intérieur de l'image car il devient simplement

$$a(i, j) = p + s(iw + j),$$

où p est l'adresse de base, s est la taille (en octet) d'un pixel, i est la rangée et j la colonne, tous deux commençant à zéro. Cette convention quasi-universelle comporte certaines exceptions, comme, par exemple, le format d'image BMP de Microsoft qui, en plus d'une multitude de complications inutiles, commence par le pixel en bas à gauche [23, 146, 193].

L'organisation en plans de bits (*bit planes*) regroupe tous les bits de même poids dans une même plage de mémoire. La fig. 1.3.2 détaille cette organisation. Cela permettrait, en principe, d'ajouter des plans et d'augmenter la profondeur des pixels; ou en avoir moins et obtenir naturellement une image avec moins de profondeur. À la limite, on pourrait supposer que cela uniformise la gestion de l'image, plus ou moins profonde en fonction de la mémoire disponible.

Dans une organisation en plans de bits, changer un pixel demande de modifier des bits dans plusieurs plans, ce qui demande un calcul d'adresse par bit, ce qui peut se révéler assez peu efficace.

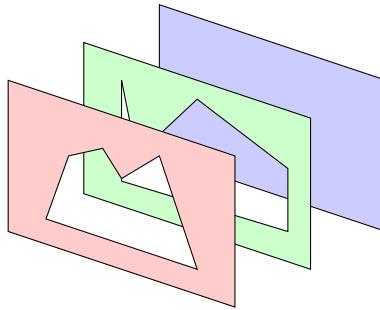


Figure 1.3.3 — L'organisation en plans avec transparence.

En effet, l'adresse de l'octet qui contient le bit b à la position (i, j) est donnée par

$$\begin{aligned} a(i, j, b) &= p + b \left\lceil \frac{w}{8} \right\rceil b + i \left\lceil \frac{w}{8} \right\rceil + \left\lfloor \frac{j}{8} \right\rfloor \\ &= p + \left\lceil \frac{w}{8} \right\rceil (bb + i) + \left\lfloor \frac{j}{8} \right\rfloor, \end{aligned}$$

où w est la largeur de l'image en pixels, h sa hauteur et p le pointeur vers le début de la plage de mémoire.

*
* * *

Les plans peuvent aussi être utilisés avec de la transparence, comme à la fig. 1.3.3. La transparence est gérée automagiquement par le matériel : une « couleur » spéciale est réservée pour coder la transparence, et la lecture d'un pixel transparent provoque la lecture du pixel correspondant dans le plan suivant. S'il est lui aussi transparent, on passe au prochain plan, et ainsi de suite, jusqu'à concurrence du nombre de plans supporté par le matériel. Cela permet de simuler une profondeur de champs, surtout avec les jeux vidéo. Pour simuler l'effet de parallaxe, il suffit de déplacer le plan le plus profond plus lentement que le plan médian, lui aussi plus lent que le premier plan. Cela rend l'illusion du mouvement tel qu'on le percevrait en train : la montagne lointaine bouge peu dans notre champ de vision mais l'avant plan défile à toute allure.

*
* * *

Le dernier arrangement que nous considérerons ici est l'organisation entrelacée des pixels. Dans cette organisation, l'image est décomposée en $n = 2^k$ banques. La première banque contient toutes les lignes de l'image qui sont congruentes à 0 modulo 2^k (donc $0, 2^k, 2 \times 2^k, 3 \times 2^k, \dots$), la seconde les lignes congruentes à 1 modulo 2^k ($1, 2^k+1, 2 \times 2^k+1, 3 \times 2^k+1, \dots$), la troisième les lignes congruentes à 2 modulo 2^k , et ainsi de suite. La fig. 1.3.4 montre une organisation en quatre plans.

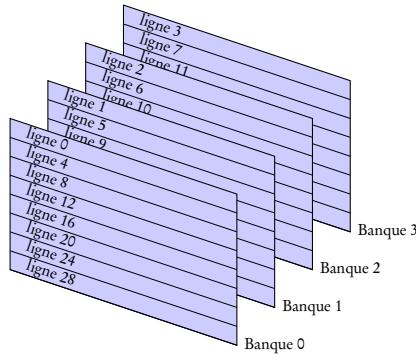


Figure 1.3.4 — Organisation Entrelacée.

Cette organisation qui paraît compliquée a pourtant une utilité. La mémoire vidéo est accédée par deux côtés : d'un côté, le processeur qui peut aller y lire et y écrire, et de l'autre, le générateur d'image qui doit y accéder en lecture (seulement) pour produire l'image vidéo. Si la mémoire ne supporte pas les accès simultanées, écrire à (ou lire de) la mémoire bloque la lecture par le générateur d'image. C'est d'ailleurs ce qu'on avait avec les premières cartes graphiques : écrire trop rapidement à la mémoire vidéo se traduisait par des lignes de « neige » dans l'écran ! Pour palier les limitations de la mémoire, les premières cartes graphiques utilisaient une organisation entrelacée de la mémoire — une solution maintes fois reprise pour la mémoire partagée [124, 139, 208]. Comme les lignes de l'image doivent être générées dans l'ordre de l'image (à cause des limitations des moniteurs d'alors [113]), l'organisation linéaire de la mémoire était présentée du côté du générateur vidéo, laissant l'organisation entrelacée du côté du processeur. Ainsi, écrire dans une banque n'affecte qu'une ligne sur quatre (comme dans la fig. 1.3.4) et si on écrit assez rapidement, on limite la quantité de neige dans l'écran [68].

1.3.3 Sous-échantillonnage

Jusqu'à maintenant, nous n'avons considéré que les couleurs *RGB*. Si ces trois couleurs primaires sont suffisantes pour recréer des couleurs réalistes, nous les percevons pas individuellement. Si l'œil humain est muni de cônes qui détectent le rouge, le vert et le bleu et de bâtonnets qui sont sensibles à l'intensité de la lumière, le signal envoyé au cerveau n'est pas directement celui mesuré par les cônes et les bâtonnets, mais un calcul qui estime la luminosité, la différence rouge-vert et la différence jaune-bleu [29]. Ainsi, l'œil est très sensible aux différences de luminosité, mais beaucoup moins aux différences de teinte et de saturation. Les formats d'images exploitent souvent cette caractéristique et utilisent plutôt un espace de couleur où une composante représente la luminosité, et où les deux autres correspondent — au moins approximativement — à la teinte et à la saturation, mais plus souvent à la différence rouge-vert et jaune-bleu. De plus, comme nous sommes beaucoup moins sensibles aux deux dernières composantes, nous pouvons nous permettre de les sous-échantillonner, c'est-à-dire

d'avoir une résolution beaucoup plus faible pour ces composantes que pour la luminosité. JPEG, par exemple, utilise cette technique [213]. L'avez-vous remarqué? Non? *My point exactly.*

Le sous-échantillonnage est noté par trois (parfois 4) nombres, de la forme $l:c_1:c_2$ ou $l:c_1:c_2:\alpha$ [219, p. 90], où :

- l est le nombre d'échantillons horizontaux de luminance. Conventionnellement 4 à cause d'une contrainte technique liée au signal de télévision, mais on trouve — rarement! — d'autres valeurs, comme 3. On suppose que l'image est toujours « pleine résolution » dans le domaine de la luminance.
- c_1 est le nombre de changements de chrominance pour l pixels horizontaux.
- c_2 est le nombre de changements de chrominance d'une ligne à l'autre.
- α , lorsque présent, représente les facteurs de transparence; habituellement le même que l , mais pas nécessairement.

Les sous-échantillonnages les plus fréquents sont

- 4:4:4 : Chaque pixel a une composante de luminance et deux composantes de chrominance — il n'y a pas de sous-échantillonnage. Supporté pour la vidéo haute définition « *studio profile* » (MPEG-4, Part 2).
- 4:2:2 : Hérité de la télévision analogique [9, 72, 75, 92, 93, 113], se retrouve encore dans les standards HDTV [17], et correspond historiquement à la transmission entrelacée.
- 4:2:0 : présente un bon compromis entre la résolution de luminance et de chrominance. Ce sous-échantillonnage est supporté par un grand nombre de formats d'images (comme JPEG [213]) et de vidéo (MPEG, H.264, HEVC, AV1, etc.).

La fig. 1.3.5 montre les sous-échantillonnages que l'on retrouve dans les standards et les produits commerciaux. La fig. 1.3.5(e) nous montre une image sous-échantillonnée en mode 4:2:0. Nous reviendrons, au chapitre 3, sur les motivations de l'utilisation du sous-échantillonnage de couleur.

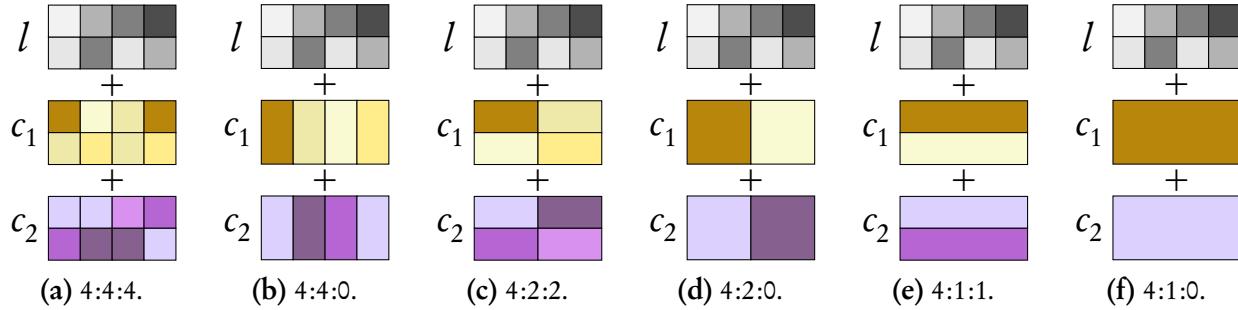


Figure 1.3.5 — Principaux motifs de sous-échantillonnage des couleurs.



(a)



(b)

Figure 1.3.6 — Des images sous-échantillonées en 4:2:0. (a) À gauche, l'image reconstruite (Gus et MIPS). (b) À droite, les composantes : en pleine résolution, la luminance, à moitié résolution, les composantes de chrominance.

1.4 Exemples de formats d'image

L'organisation en mémoire des images reflète souvent l'architecture de la machine sous-jacente, mais les formats de fichiers qui sauvegardent les images s'efforcent, au contraire, de s'en absoudre et d'assurer un rendu cohérent sur les différentes plate-formes. Les formats de fichiers proposent,

souvent de façon différente, de balancer la taille du fichier avec la qualité du rendu, c'est-à-dire que la compression peut être avec ou sans perte.

La compression sans perte restitue, bit pour bit, les données originales. Ce stockage sans perte est perçu comme essentiel pour les applications médicales, pour l'archivage, ou pour la masterisation. La compression avec perte essayera de détruire l'information peu perceptible afin de réduire davantage la taille du fichier. Nous reviendrons plus longuement sur la compression au chapitre ??.

Quelques formats usuels :

- PNM : alias *netpbm*, pour *portable any-map format*. Un format extrêmement simple avec une entête qui annonce le mode (noir et blanc, *RGB*, *RGBA*) et la résolution. Introduit par Jef Poskanzer en 1988, ce format est demeuré marginal, mais sans disparaître.
- GIF (*graphics interchange format*¹) Ce format est spécialement conçu pour les images par palette, avec au plus 8 bits par pixel. La compression utilise une variante de l'encodage LZ78 [274], LZW, que l'on doit à Welch [264, 267]. Le format prévoit aussi un mode entrelacé pour permettre d'afficher une image approximative même lorsque la transmission n'est pas complétée (un *must* lorsque la vitesse de transfert se calcule en kilobits/s!). Prévoit aussi un mode « multi-image » pour les animations. La limitation la plus importante est le nombre de couleurs, limité à 256 par image.
- TIFF (*Tagged Image File Format*) est un format originalement proposé par les fabricants de numérisateurs de documents, mais a été rapidement adopté pour la publication et l'info-graphisme. TIFF supporte un grand nombre d'espaces de couleurs, dont *RGB*, *CYMK*, YC_rC_b , CIÉ $L^*a^*b^*$,..., avec différentes profondeurs. Ce format supporte aussi plusieurs algorithmes de compression, comme RLE, CCITT G3 [12], LZW et même des modes avec perte utilisant JPEG [2].
- JPEG (*Joint Photographic Experts Group*, du nom du comité ISO/IEC chargé de développer un nouveau standard) [213]. Le format JPEG commence par changer l'espace de couleur *RGB* vers YC_rC_b , puis sous-échantillonne (de façon paramétrable) les composantes C_r et C_b (qui correspondent à la différence rouge-vert et jaune-bleu, respectivement). Ces plans sous-échantillonés sont ensuite transformés dans un domaine fréquentiel où les données jugées imperceptibles sont détruites. Suit enfin un codage entropique, de style Huffman ou codage arithmétique.

1. D'après l'auteur, GIF se prononce « jiffe »). Tout le monde dit « guif », comme *gift* ou *graphics*.

Le successeur de JPEG, JPEG2000, utilise les ondelettes comme transformée et un format d'encodage beaucoup plus sophistiqué [252]. Cependant, la relative complexité de l'implémentation et l'efficacité, ainsi que le support universel du format JPEG original ont grandement freiné l'adoption de ce nouveau standard.

- **PNG (*Portable Network Graphics*)**. Ce format sans perte supporte plusieurs espaces de couleurs, la détection d'erreur, et la compression grâce aux prédicteurs [13, 47]. L'algorithme de compression, comme celui de GIF, est un dérivé de l'algorithme LZ78, LZSS [244]. Ce format supporte aussi l'entrelacement¹ pour les connexions lentes.
- **Kodak PhotoCD**. Originalement conçu pour la photographie numérique haute résolution (6144×4096 par exemple), le format est maintenant tombé en désuétude [262]. La compression utilisée est relativement simple. Après une transformation vers un espace de couleur propriétaire, KodakYCC, l'image est codée par prédition : l'image est réduite à $\frac{1}{4}$ de la résolution (réduction par 2 dans les deux dimensions) puis réinterpolée à pleine résolution pour servir de prédicteur, à partir duquel on code les différences [90].
- **RAW**. Il existe un grand nombre de formats pour les images issues de caméras numériques (ou de numériseurs de document). Ces formats supportent des profondeurs de couleurs plus grandes qu'à l'habitude, et stockent les valeurs sans perte (parfois sans compression). De plus, ils incluent les métadonnées de la prise de vue (heure, date, géolocalisation, temps d'exposition, température de couleur, correction de couleur, etc.). Les métadonnées sont stockées selon le standard EXIF [18], tandis que les données elles-mêmes sont stockées selon l'algorithme choisi pour le format particulier (par ex. [6]) et de façon à accommoder la géométrie du capteur — sur laquelle nous reviendrons au chapitre 2. Le standard DNG (*digital negative*) d'Adobe se veut un format *raw* universel [19].

Il existe encore un grand nombre de formats de fichiers. Pour en savoir plus, vous pouvez consulter [51, 146, 193].

1.5 Remarques bibliographiques

Le format PNG a été conçu pour remplacer GIF qui risquait de ne plus pouvoir être utilisé librement. En effet, le propriétaire de l'algorithme de compression, la société Unisys (originalemen

1. L'entrelacement Adam7 a été retenu. Pourquoi ce nom ? C'est le 7^e essai... d'Adam M. Costello, un des contributeurs originaux au projet PNG.



Figure 1.5.1 — Les caractères semigraphiques du TRS-80 Color Computer. Source : Wikipedia.

Sperry Corporation), voulait exiger licences et royaumes, grâce au brevet qu'elle détenait [265]. Cela en contraria plusieurs, et dès 1995, un petit groupe se mit au développement d'un format libre de brevets [161, ch. 7]. Dès l'année suivante, les bases étaient jetées [47]. Ultimement, PNG finit par s'imposer, mais GIF continuera d'être utilisé par tous impunément et — surtout — impécuniairement.

*
* * *

Les vieux ordinateurs (pour lesquels, par ailleurs, je n'éprouve *aucune* nostalgie) nous enseignent plein de trucs astucieux d'économie de mémoire et de parcimonie d'encodage. Certains ordinateurs permettaient, par exemple, de changer la palette à chaque ligne de l'image grâce à une interruption matérielle ; d'autres encore simulaient un mode graphique grâce aux caractères « semigraphiques ». Par exemple, le TRS-80 Color Computer combinait à la fois forme et couleur des caractères (voir fig. 1.5.1) [26]. Des astuces similaires se retrouvent encore dans les algorithmes de compression de texture.

*
* * *

Les images multi- ou hyperspectrales occupent une place prépondérante en exploration terrestre [112] comme spatiale [37, 154, 164, 257].

*
* * *

L'idée du tramage n'est pas récente, car c'est une invention du XIX^e siècle! Les graines de l'idée sont plantées dans les années 1840 par William Henry Fox Talbot (1800–1870). Il obtient un premier brevet pour un procédé d'impression des images sur un papier photographique [250], mais ce n'est pas encore du tramage. Les premiers à mettre en œuvre du véritable tramage sont deux Montréalais : George-Édouard-Amable Desbarats (1838–1893) [101] et William Augustus Leggo

(1830–1915) [44]. En 1869, ils lancent les *Canadian Illustrated News*, qui montrent, pour la première fois une reproduction tramée d'une photographie (du duc Arthur de Connaught et Strathearn). Il faudra encore attendre quelques années pour que la technique se perfectionne et s'accommode de la couleur [131–133].

2

Saisies des images

Sommaire. Dans ce chapitre, nous nous intéresserons à la formation d'image dans une caméra. Nous discuterons des senseurs, de la séparation des couleurs, et finalement des *color filter arrays*.

2.1 Introduction

Dans ce chapitre, nous nous intéresserons à la formation d'image, mais du point de vue des senseurs seulement. Bien que la quincaillerie d'un appareil photo ou d'un numérisateur, les lentilles, la mise au point automatique (*autofocus*), les diaphragmes, les obturateurs, etc., soient tous des sujets fort intéressants, nous n'en parlerons pas ici.

2.2 Senseurs et couleurs

2.2.1 Technologies de senseurs

Il existe deux grandes familles de (photo)senseurs :

- Les senseurs actifs (souvent dits CMOS¹, à cause de la technologie d'intégration de circuits utilisée). Ce type de senseur utilise une grille de photodiodes (ou de photorésistances) pour

1. Pour *Complementary Metal-Oxyde Semiconductor*.



Figure 2.2.1 — L'effet du *rolling shutter*, résultant du jeu entre le déplacement de l'objet et l'ordre de lecture des pixels.
Source : Wikipedia [229].

capturer l'image. Or, pour mesurer la quantité de lumière reçue par un pixel en particulier, la photodiode (ou photorésistance) doit être traversée par un courant. Ce courant est activé au moment de la lecture du pixel ; ce qui implique que l'image n'est pas capturée dans son ensemble d'un coup, mais séquentiellement pixel par pixel [97, 98, 228]. Cette technologie présente plusieurs avantages :

- Grande densité réalisable. Des senseurs de taille très réduite sont idéaux pour les applications où on peut se contenter d'une qualité d'image moindre, comme les appareils photo intégrés aux téléphones cellulaires.
- Grande sensibilité. Tandis qu'un film argentique typique offre une sensibilité entre 100 et 10000 ISO [1, 11], les senseurs CMOS peuvent aller jusqu'à 4000000 ISO (un senseur fabriqué par Canon).
- Le fait que les pixels doivent être activés/adressés individuellement pour être lus a pour effet secondaire de permettre de lire un sous-ensemble arbitraire des pixels. Cela permet de saisir des images à plus faible résolution (soit en espaçant régulièrement les pixels lus, soit en restreignant la région lue à un rectangle) plus rapidement.

Par contre, comme chaque pixel est lu séquentiellement, les images saisies par ce type de senseur peuvent être déformées. L'effet le plus connu est celui du *rolling shutter* [24], dont un exemple est montré à la fig. 2.2.1.

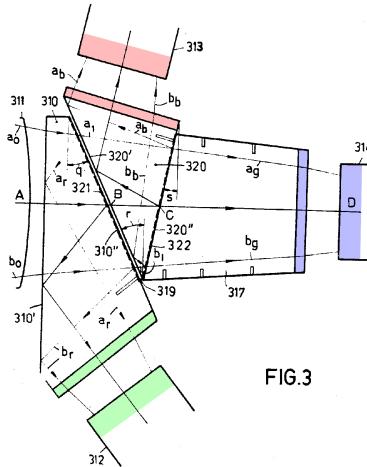


Figure 2.2.2 — Trois prismes séparant la lumière entrante en trois faisceaux, chacun filtré pour ne laisser passer qu'une plage de couleurs. Tiré de [74], couleurs ajoutées.

- Les senseurs à charge couplée. Ces senseurs utilisent une charge électrique isolée du senseur à proprement parler, et ce sont les photons qui traversent le pixel qui charrent les électrons de la réserve vers le détecteur. Conçus par Boyle et Smith en 1969 [50] et rapidement mis en œuvre par Amelio *et al.* [34], ces senseurs permettent la saisie de l'image dans son ensemble au même instant, puisque ce sont les photons reçus qui transfèrent les charges. Par contre, pour sortir les valeurs des pixels, il faut quand même les lire séquentiellement (grâce à une mécanique compliquée de registre à décalage).

Cependant, ces senseurs sont monochromatiques — ou plus exactement *panchromatiques* — et doivent être aidés de filtres et/ou de prismes pour saisir des images en couleur.

2.2.2 Prismes et filtres

Comme les senseurs eux-mêmes sont typiquement monochromatiques, il faut soit scinder la lumière en plusieurs bandes de longueurs d'onde, soit appliquer des filtres ne laissant passer que certaines longueurs d'onde devant les senseurs, voire les deux !

La première solution — séparer la lumière en bandes — est montrée à la fig. 2.2.2 [74]. Ici, des prismes séparent et reflètent la lumière, créant trois chemins qui chacun se terminent par un filtre de couleur. La lumière qui atteint un senseur est alors monochromatique (ou presque) : rouge, verte, bleue.

Une seconde solution (qui paraît plus récente) est d'utiliser des prismes dichroïques (étymologiquement *deux couleurs*) ou trichoïques (*trois couleurs*) pour séparer la lumière blanche en couleurs

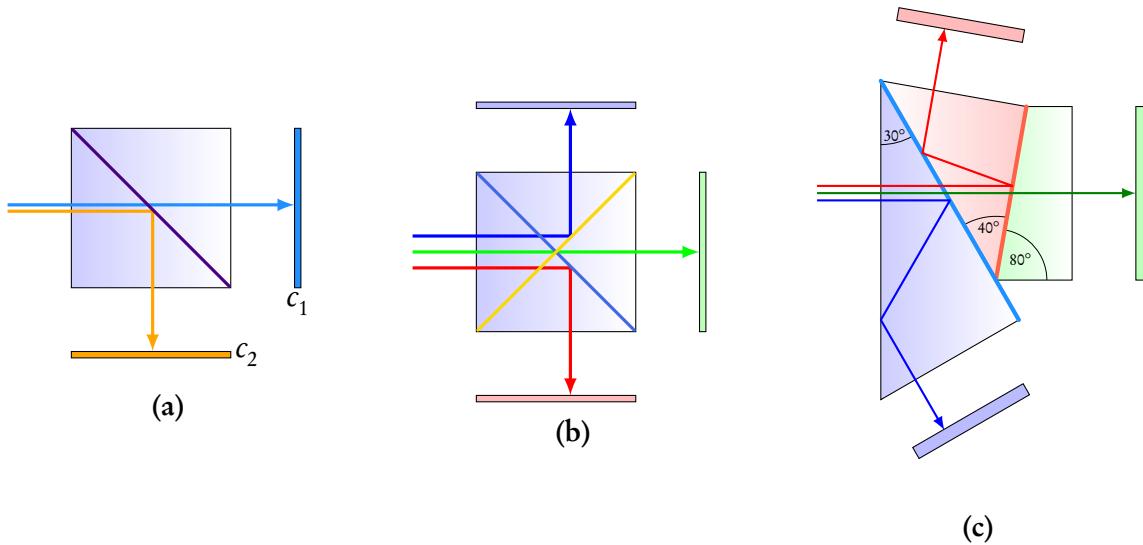


Figure 2.2.3 — Prismes pour la séparations de couleur. (a) prisme dichroïque, (b) trichroïque (*cross dichroic*) et (c) trichroïque de style Philips.



Figure 2.2.4 — Un prisme dichroïque (taille $\approx 1\text{cm}^3$) du type montré à la fig. 2.2.3 (b). Photo : Steven Pigeon.

choisies, possiblement polarisée [120]. La fig. 2.2.3 montre quelques unes des configurations. La fig. 2.2.3 (a) montre une configuration où deux couleurs sont séparées et dirigées vers deux senseurs. Les configurations qui nous intéressent le plus sont montrées aux figs. 2.2.3 (b) et 2.2.3 (c), cette dernière reprenant la configuration de la fig. 2.2.2, tandis que la première montre le schéma d'un prisme semblable à celui montré aux figs. 2.2.4 et 2.2.5. Pour toutes astucieuses qu'elles soient, les solutions basées sur des prismes di- ou trichroïque ont pour désavantage principal d'être encombrantes. Il est en effet très difficile de construire ces assemblages dans des appareils aussi compacts que des téléphones mobiles! C'est pourquoi on leur préfère souvent d'autres solutions, comme les CFAs (*color filter arrays*), sujets de la section suivante.

2.2.3 Systèmes à un seul senseur

Si nous voulons nous restreindre à un seul senseur, il faut que chaque pixel mesure les trois composantes de couleur. Cela mène à un arrangement similaire à la fig 2.2.6 où chaque région du

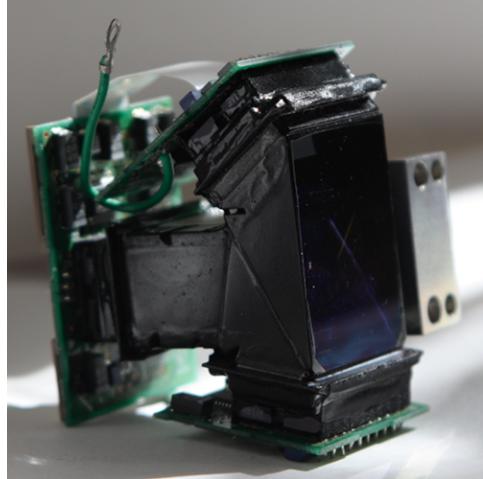


Figure 2.2.5 — Un exemple de caméra utilisant l’assemblage de type Philips. Source : Wikipedia [272].

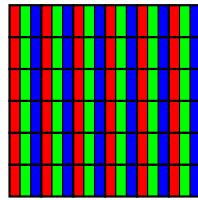


Figure 2.2.6 — Un senseur « pleine couleur », où chaque pixel possède trois types de détecteurs.

senseur (donc un « pixel ») possède trois détecteurs séparés, chacun réagissant à une couleur. La géométrie la plus évidente — mais pas la seule possible! — est d’avoir trois régions rectangulaires qui correspondront aux composantes du système d’imagerie (voir ch. 1). Cependant, cette solution comporte ses inconvénients. Comme la densité des pixels augmente, la taille de chacune des régions est d’autant plus réduite qu’elle n’occupe qu’un tiers de pixel, ce qui en réduit la sensibilité. De plus, il devient difficile de créer les lentilles qui concentrent la lumière reçue par la région rectangulaire sur le senseur lui-même (comme à la fig. 2.2.7).

2.2.3.1 Bayer et ses amis

Plutôt que d’utiliser un senseur où chaque pixel comporte trois senseurs (un pour chaque couleur), Bayer proposa d’utiliser une grille où chaque pixel n’aurait qu’une lentille et un senseur, mais pour une seule des couleurs primaires. Pour s’assurer de mesurer (au moins approximativement) les

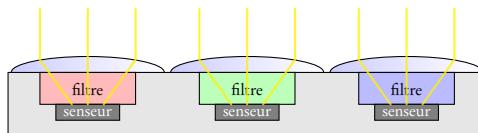


Figure 2.2.7 — Un senseur avec filtres de couleur et microlentilles.

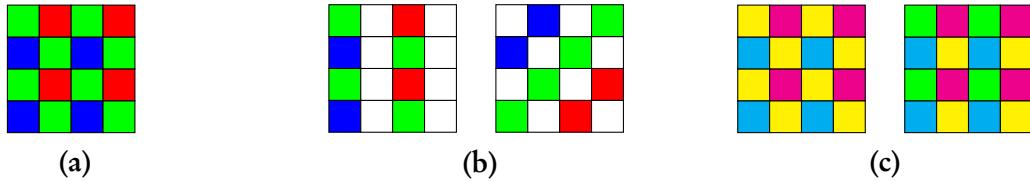


Figure 2.2.8 — Motifs de Bayer. (a) Motif original de Bayer [39], (b) variantes Kodak [67], (c) variantes avec d’autres primitives de couleur (cyan, magenta, jaune et cyan, magenta, jaune, vert).

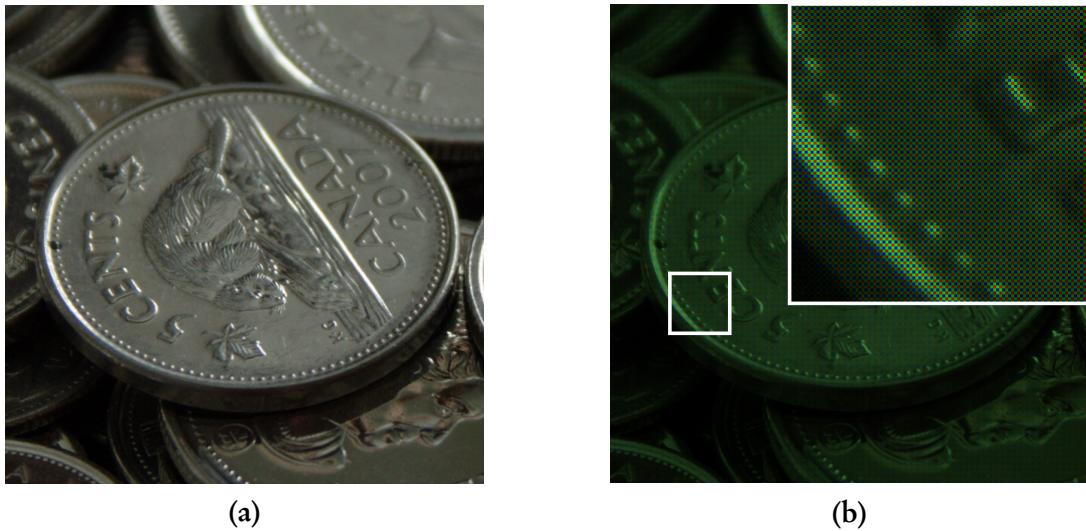


Figure 2.2.9 — Image saisie avec un motif de Bayer. (a) Image désirée, (b) image telle que saisie par le senseur, en encadré, la zone agrandie.

couleurs dans les trois composantes, les filtres de couleurs sont dispersés comme nous le montre la fig. 2.2.8 (a) [39]. La fig. 2.2.8 montre aussi quelques variantes où certains senseurs, montrés en blanc, sont « panchromatiques »; d’autres encore utilisent des primitives différentes.

Le problème évident avec cette technique, c’est que chaque pixel, au sortir du senseur d’image, ne comporte qu’une des trois composantes de couleur! C’est bien embêtant et il faut utiliser un algorithme pour « inventer » les couleurs manquantes. Certains se contentent d’algorithmes assez simples — voisin le plus près, interpolation linéaire, etc. — mais d’autres préféreront des algorithmes qui préservent la teinte et les rebords tout en évitant les effets de moiré. Parmi les algorithmes se retrouvant fréquemment dans les logiciels de traitement de photographie (dans un ordre quelconque) :

- VNG (*Variable Number of Gradients*), proposé par Chang *et al.* [62]. Cet algorithme estime les gradients dans huit directions pour procéder heuristiquement au choix des valeurs manquantes. Cet algorithme tente de préserver les rebords et leur direction. Il existe une variante, VNG4, où les pixels verts sont traités comme deux couleurs distinctes : on a rouge,

bleu, vert-1 et vert-2.

- PPG (*Patterned Pixel Grouping*), par Lin (2003) [160]. Cet algorithme estime les gradients, mais dans l'espace de la luminance. Il prend ensuite une décision en fonction du gradient le plus petit, préférant les plages au rebords.
- AHD (*Adaptive Homogeneity Directed (Demosaicing Algorithm)*), par Hirakawa et Parks [126–130]. Cet algorithme estime les variations de couleurs par direction et utilise la direction de variation minimale pour interpoler, puis utilise un filtre d'antialiasage sélectif pour adoucir les « escaliers ».

On trouve plusieurs autres algorithmes, certains assez quelconques, d'autres qu'on ne connaît qu'à travers leur implémentation, n'ayant pas fait l'objet de « vraies » publications. Parmi ceux-ci, on trouve RCD (*Ratio-Corrected Demosaicing*) et DCB par Jacek Gózdź et AMaZE (*Aliasing Minimization and Zipper Elimination*) par Emil J. Martinec.

2.3 Remarques bibliographiques

L'idée de Bayer a eu un impact initial limité, et la littérature des vingt-cinq années qui suivirent ce premier brevet [39] est assez modeste [79, 150, 249]. Par contre, un peu avant les années 2000, l'intérêt renaît et la littérature explose [10, 30–32, 35, 56, 61–63, 65, 67, 70, 82, 83, 87, 88, 96, 103, 109, 114, 116–118, 121, 122, 126–130, 134, 138, 147, 148, 158–160, 163, 165–167, 177, 185, 186, 190–192, 194, 196, 198, 199, 212, 222–226, 237, 247, 251, 253, 260, 261, 270, 273, 275]! Mais pourquoi? Simplement parce que l'évolution de la technologie rendait la photographie numérique réalisable à peu de frais, promettant ainsi des applications nouvelles et peu dispendieuses!

*
* * *

Les algorithmes de mise au point automagiques (*autofocus*) sont extrêmement mal documentés! En fait, ce n'est pas exact : comme les bons algorithmes de mise au point automatiques constituent des avantages compétitifs intéressants — voire essentiels! — entre les différents modèles et manufacturiers, la plupart sont plus ou moins des secrets industriels que l'on ne connaît qu'à travers les brevets qui les protègent [28, 105, 106, 119, 149, 153, 233, 235, 236] et quelques articles de recensement [38].

3

Espaces de Couleurs

Sommaire. Dans ce chapitre, nous nous intéresserons aux espaces de couleurs. Nous verrons qu'il existe deux grandes familles d'espaces de couleurs, les espaces linéaires et (sans surprise) les espaces non linéaires. Nous insisterons un peu plus sur les espaces de couleurs reliés à la télévision et au cinéma. Enfin, nous terminerons avec les remarques bibliographiques.

3.1 Introduction

Pourquoi avons-nous besoin d'autres espaces de couleurs que l'espace *RGB* (dont nous avons déjà discuté au ch. 1)? En partie parce que l'œil (normal) fonctionne avec quatre types de capteurs, les bâtonnets, qui sont achromatiques et réagissent à l'intensité de la lumière, et trois types de cônes qui captent chacun le rouge, le vert et le bleu, mais l'information qui est transmise au cerveau est encodée en termes de la luminosité, de la différence vert-rouge et de la différence jaune-bleu. Il tombe alors sous le sens de tenter d'exploiter ce trait physiologique et d'encoder les couleurs dans un espace qui corresponde à cette sensibilité; c'est-à-dire avec l'information principalement concentrée dans la luminosité, et avec moins d'information dans le *chroma* (c'est-à-dire la « couleur » avec sa teinte et sa pureté) auquel nous sommes beaucoup moins sensibles.

3.1.1 Des couleurs primaires

Les différents types de cônes ne répondent pas à une seule longueur d'onde de lumière spécifique, plutôt ils répondent différemment à différentes longueurs d'ondes, avec, il est vrai, une réponse plus

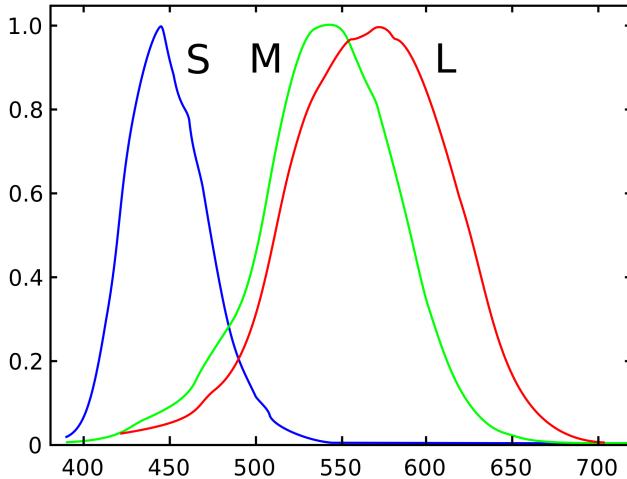


Figure 3.1.1 — Sensibilité de l’œil aux longueurs d’ondes. Source : Wikipedia.

forte autour de certaines longueurs d’ondes que nous associons aux couleurs rouge, vert et bleu. La fig. 3.1.1 montre les courbes de réponses normalisées.

Comme les courbes se recoupent de façon importante, cela implique qu’une longueur d’onde donnée (on parle d’une couleur monochromatique ou de « couleur spectrale ») peut stimuler plus d’un type de cône; nous avons alors ça sensation d’une couleur « complexe ». Par contre, deux (ou trois) longueurs d’ondes judicieusement choisies et mélangées avec des intensités appropriées pourraient stimuler les cônes exactement de la même façon ! On décrit toutes les distributions spectrales — tous les mélanges de longueur d’ondes — qui résultent dans la même couleur *perçue* comme des couleurs métamères (ou couleurs homochromes).

Il s’agit alors de choisir un petit nombre de longueurs d’ondes suffisantes, par métamérisme, pour produire une gamme de couleurs assez riche pour être intéressante. Comme l’œil dispose de trois types de cônes dont les réponses diffèrent significativement, choisir trois *couleurs primaires* paraît indiqué, et sans doute devraient-elles correspondre aux maximums des courbes de réponses de la fig. 3.1.1. Pourrait-on reproduire les couleurs de façon satisfaisante avec moins de couleurs primaires ?

Clairement, une seule couleur serait bien insuffisante et ne donnerait qu’une seule couleur en intensité variable, ce serait une image monochrome. Deux couleurs paraissent aussi insuffisantes. En effet, deux couleurs ne pourraient qu’exprimer des dégradés entre ces deux couleurs, par exemple, passer du rouge au bleu en générant différents tons de mauve, fuchsia, etc., ce qui serait bien étrange et dérangeant. Par contre, trois couleurs primaires permettent des combinaisons bien plus intéressantes. Si nous considérons l’espace des couleurs comme un volume en trois dimensions où une dimension correspond à la luminance perçue et les deux autres à des propriétés comme la teinte et la pureté (des caractéristiques que nous expliquerons plus en détail dans un instant), une unique couleur primaire correspond à un segment de droite à quelque part dans le volume orienté parallèlement à la luminance.

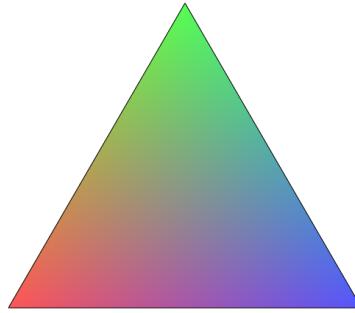


Figure 3.1.2 — Trois couleurs primaires sont mélangées pour donner une multitude de couleurs intermédiaires. Vers le centre, nous avons des tons de gris, et des couleurs maximalement saturées sur les rebords.

Deux couleurs primaires correspondent à un plan coupant le volume à un angle donné. Ce n'est qu'avec les trois composantes que nous pouvons remplir le volume.

Nous pouvons aussi visualiser les combinaisons des trois couleurs primaires comme un diagramme ternaire, dont une approximation visuelle est donnée à la fig. 3.1.2. Un diagramme ternaire (en anglais *ternary plot*) affiche un point de composantes originales a , b et c entre 0 et 1 inclusivement aux coordonnées

$$\alpha = \frac{a}{a+b+c},$$

$$\beta = \frac{b}{a+b+c},$$

$$\gamma = \frac{c}{a+b+c}$$

dans le diagramme ; lesquelles sont converties sur le plan par

$$\begin{bmatrix} 0 & \frac{1}{2} & -\frac{1}{2} \\ 1 & \frac{\sqrt{3}}{2} & \frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix}$$

où la matrice est composée des vecteurs colonnes qui correspondent aux orientations des côtés du triangle équilatéral dont le coin inférieur gauche est aux coordonnées $(0, 0)$. Cette représentation est importante car nous la réutiliserons à la § 3.1.2.

La fig. 3.1.2 devrait nous convaincre que les combinaisons de trois couleurs primaires créent des tons variés et intéressants. Au centre, nous trouvons des tons de gris, le blanc lorsque l'intensité des couleurs est suffisante et du noir lorsque l'intensité est nulle¹. Le problème devient alors

1. Une intensité suffisante ne signifie pas nécessairement maximale ; à partir d'une certaine intensité, tous les cônes saturent et même les couleurs monochromatiques apparaissent blanches. Si l'intensité dépasse un certain seuil, les dommages à l'œil sont permanents. Nous considérerons donc que l'intensité est entre des limites « modérées » : noir est une intensité de lumière insuffisante pour stimuler les cônes, et blanc une combinaison de couleurs suffisamment intense pour donner une *sensation* de blanc. Les intensités sont normalisées arbitrairement entre 0 et 1, ou quelques autres limites imposées par l'espace de couleurs considéré.

de judicieusement choisir ces trois couleurs primaires — pas forcément monochromatique. Mais comment les choisir puis les décrire? C'est le problème que nous explorerons à la section 3.2. Nous verrons aussi que le choix des couleurs primaires varient d'un standard d'image à l'autre, évoluant selon les technologies disponibles.

3.1.2 Familles d'espaces de couleurs

Les espaces de couleurs (ou colorimétriques) sont donc des représentations des couleurs qui permettent d'encoder les couleurs (stimuli perçus) de façon numérique, préféablement d'une manière utile pour les diverses opérations que nous ferons subir aux images.

Nous distinguons deux grandes familles d'espaces de couleur :

- Les *espaces linéaires* expriment les couleurs en termes de combinaisons linéaires des couleurs primaires. C'est donc qu'il est possible de les exprimer comme des produits matrice/vecteur; et les « traductions » d'un espace de couleurs linéaire à un autre s'expriment aussi par des matrices réversibles. C'est-à-dire que s'il existe une matrice T d'un espace A à un espace B , nous avons

$$T \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \quad \text{et} \quad T^{-1} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

Typiquement, l'un de ces deux espaces est l'espace *RGB* pour la raison très pragmatique que c'est typiquement l'encodage des couleurs directement supporté par le matériel de rendu.

- Les *espaces non-linéaires*. Ces espaces expriment les couleurs en utilisant des transformations généralisées, habituellement pour organiser les couleurs selon des caractéristiques qui sont elles-mêmes non linéaires, telles que la luminosité perçue, la teinte, la saturation, etc. Ces transformations peuvent être complexes et difficiles à calculer, par exemple, à cause de l'emploi de fonctions logarithmiques, trigonométriques, etc. La complexité de ces espaces les rend peu attrayants pour les applications où le temps de calcul doit demeurer modeste (comme la compression d'image), mais peuvent quand même être utilisés lorsque les propriétés de ces espaces les rendent intéressants pour un traitement en particulier.

Qu'ils soient linéaires ou non, les espaces de couleurs vont tenter d'exploiter les caractéristiques du système visuel humain. En particulier, comme nous le montre la fig. 3.1.3, que si les couleurs perçues sont bien captées par les cônes et les bâtonnets dans l'espace de couleurs primaires rouge, vert et bleus en plus de la luminance, l'information envoyée au cerveau est un encodage de la luminance,

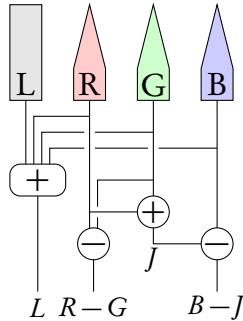


Figure 3.1.3 — Les cônes et les bâtonnets combinent leurs signaux pour donner la luminance, les différences rouge-vert et jaune-bleu [29].

d'une différence rouge-vert ($R-G$) et jaune-bleu ($B-J$). Nous ne sommes pas également sensibles à ces trois composantes. En effet, nous sommes beaucoup plus sensibles aux variations en luminance qu'aux variations dans les différences rouge-vert ou jaune-bleu. Cela pourra être exploité judicieusement pour coder la couleur avec moins de précision, mais sans dégradation visible.

*
* * *

Les prochaines sections exploreront ces deux grandes familles plus en détail, tout en tissant les différents liens de parenté entre les espaces de couleurs présentés.

3.2 L'espace de couleurs CIÉ 1931

Rappelons qu'à la base, la lumière n'est composée que de photons de différentes énergies¹ qui sont ensuite détectés par les capteurs de l'œil, les cônes des trois types et les bâtonnets. Ces capteurs ont des réponses différentes en fonction de l'énergie des photons. Pour modéliser la réponse de l'œil aux couleurs (qui n'existent par ailleurs que dans notre esprit), il faut donc estimer les *courbes de réponse (matching functions)* des différents capteurs de l'œil. C'est ce qu'à fait la Commission internationale de l'éclairage (ci-après, CIÉ) dans les années 1920–1930 [115, 217, 241, 268]. Les courbes obtenues par la CIÉ sont montrées à la fig. 3.2.1.

Le modèle (qui ne tient pas compte de la réponse des bâtonnets qui varie grandement en fonction de l'intensité de l'éclairage) a été ajusté sur un relativement petit nombre d'observations (dix sujets

1. Rappelons aussi que des photons de haute énergie possèdent des longueurs d'ondes plus courtes, car la fréquence f et longueur d'onde λ sont liées par la formule $f = \frac{c}{\lambda}$ (et donc $\lambda = \frac{c}{f}$), et que l'énergie contenue dans un photon de longueur d'onde λ est donnée par $E = h \frac{c}{\lambda}$, où $h \approx 6.626 \times 10^{-34}$ Js est la constante de Planck.

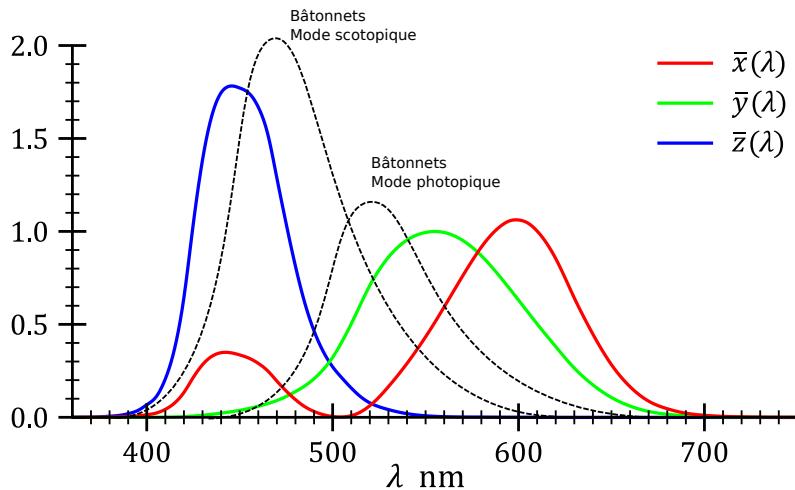


Figure 3.2.1 — Les fonctions colorimétriques de l’observateur standard selon le modèle CIÉ 1931. Modifié d’après Wikipedia pour les bâtonnets, échelles très approximatives.

dans l’étude de Wright [268] et sept dans celle de Guild [115]) et on doit s’attendre à des variations significatives d’un observateur à l’autre. Cependant, la réponse n’est pas entièrement subjective, elle est forcément proportionnelle, en intensité, à l’énergie contenue dans la lumière. Nous avons donc une réponse visuelle

$$V \propto \int v(\lambda)E(\lambda)d\lambda,$$

où $v(\lambda)$ est la sensibilité de l’œil à la longueur d’onde λ et $E(\lambda)$ l’énergie de la longueur d’onde λ [46]. Par hypothèse, si nous avons deux sources de lumière $E_1 = v(\lambda_1)E(\lambda_1) = v(\lambda_2)E(\lambda_2) = E_2$, alors elles sont de même brillance (intensité) perçue. De même, pour tous les $0 \leq \alpha \leq 1$, la mixture

$$\alpha E_1 + (1 - \alpha)E_2$$

est de brillance perçue constante (même si la couleur varie) [152].

3.2.1 De l’origine du triangle des couleurs

En mélangeant trois couleurs spectrales (monochromatiques, c'est-à-dire composées d'une seule longueur d'onde) judicieusement choisies dans des proportions calculées, nous devrions pouvoir simuler n'importe quelle couleur visible. Pour des raisons technologiques¹, ces trois couleurs judicieusement choisies par le CIÉ sont

Rouge	700 nm
Vert	546.1 nm
Bleu	435.8 nm

1. Dans les années 1920, les lampes à vapeur de mercure semblaient être un choix raisonnable, d'autant plus que trois des bandes d'émission correspondent exactement aux couleurs retenues, sauf pour le rouge, à 650 nm.

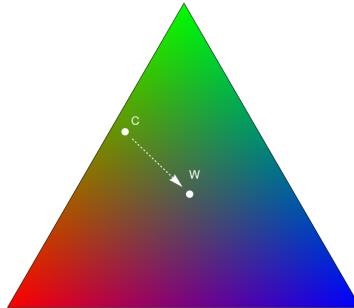


Figure 3.2.2 — Trois couleurs primaires sont mélangées. Vers le centre, nous avons un point perçu comme un gris neutre (W). Une couleur C près du bord est perçue comme saturée. Se déplacer vers le gris neutre produit des couleurs de moins en moins saturées.

Les couleurs que l'on peut obtenir par combinaisons linéaires avec des coefficients normalisés entre 0 et 1 inclusivement, sont contenues dans un triangle (revoir la fig. 3.1.2, p. 35, et la section 3.1.1) dont les sommets sont les couleurs primaires choisies. Considérez maintenant la fig. 3.2.2. Dans ce diagramme triangulaire, un point, pour une brillance moyenne donnée, sera perçu comme un gris neutre (c'est-à-dire sans teinte). Dans la fig. 3.2.2, c'est le point « blanc » marqué par un W (pour White). Ce point, en anglais *white point*, variera en fonction des couleurs primaires choisies. Une autre couleur (marquée par un C dans la figure) paraîtra plus riche, plus saturée, si elle se trouve près des bords de la figure. Faire glisser cette couleur vers le point blanc donne des couleurs de moins en moins saturées (mais notez que suivre la flèche en ligne droite ne préserve pas la *teinte*!).

Pour produire des couleurs en dehors du triangle de la fig. 3.2.2, il faudrait avoir recours à des couleurs qui dépassent l'intensité maximale permise (par hypothèse de construction du triangle) ou encore négatives, ce qui, bien entendu, est impossible physiquement¹, mais *mathématiquement cohérent*. Nous pourrions alors choisir trois couleurs primitives « imaginaires », X , Y , Z qui englobent non seulement le triangle de la fig. 3.2.2 mais aussi *toutes* les couleurs que nous pouvons percevoir.

Considérons maintenant la fig. 3.2.3. Nous trouvons maintenant un triangle dont les sommets sont les couleurs imaginaires X , Y et Z . Dans la figure, le triangle montre toutes les couleurs que l'on pourrait obtenir par des combinaisons des couleurs X , Y et Z . En gris, ce sont les couleurs impossibles à obtenir physiquement. En blanc, les couleurs visibles. Dans le triangle de couleur, les couleurs que l'on peut représenter comme une combinaison des trois couleurs primaires. La zone en blanc représente des couleurs visibles, mais impossibles à représenter à partir des trois couleurs primaires (si le blanc eût été coloré, il eusse dû l'être avec de fausses couleurs!). Il existe plusieurs

1. Et avant qu'un petit malin suggère d'utiliser des antiphotons, sachez que les photons, comme les bosons de jauge (autres que W^- et W^+), sont leurs propres antiparticules.

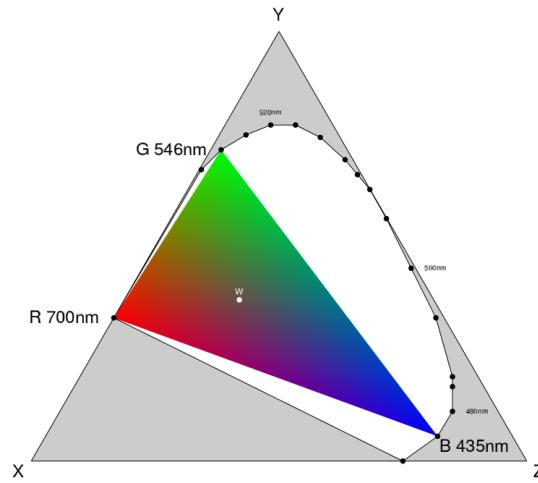


Figure 3.2.3 — Le triangle $X Y Z$ de la CIÉ 1931 (schéma très approximatif).

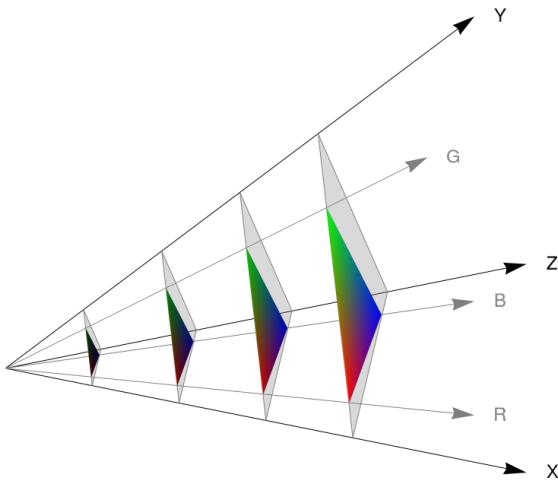


Figure 3.2.4 — La pyramide $X Y Z$ de la CIÉ 1931.

représentations équivalentes du triangle CIÉ, comme par exemple la fig. 1.2.2, p. 2, où le triangle n'est plus équilatéral mais rectangle et où l'axe X change de direction.

3.2.2 L'espace des couleurs CIÉ comme un espace vectoriel

Si on impose comme condition qu'aucune des couleurs (imaginaires ou réelles) qui correspondent aux sommets du triangle ne peut être une combinaison des deux autres (elles seraient colinéaires sinon), elles peuvent former un espace vectoriel à trois dimensions où chacun des vecteurs de la base correspond aux intensités d'une couleur primaire. Ainsi, une couleur est décrite de façon non ambiguë comme une coordonnée dans un espace à trois dimensions.

La fig. 3.2.4 montre différents plans formés par des couleurs de luminosité perçue égale. Les régions en gris correspondent aux triangles XYZ pour des luminosités données, y compris les couleurs impossibles à réaliser. Inclus dans chacun de ces plans, les triangles des couleurs réalisables à partir des trois couleurs primaires (réalisables aussi) R, G et B . À la pointe, la luminosité est nulle et toutes les couleurs sont confondues dans le noir. Lorsque X, Y et Z sont suffisamment grands, un plan apparaît avec un point vraiment blanc (plutôt que gris).

3.2.3 Contraintes imposées sur l'espace CIÉ 1931

Pour que l'espace de couleurs soit vraiment utile, il doit se comporter non seulement comme un espace vectoriel, mais aussi posséder des caractéristiques intuitives, par exemple :

- Pour une couleur C , varier sa luminosité déplace la couleur le long de la droite passant par l'origine et C (changer la luminosité ne change pas la teinte) ;
- Les couleurs spectrales (monochromatiques) sont sur le pourtour de la forme en « voile de bateau » ;
- Les couleurs maximalement saturées sont aussi sur le pourtour de la « voile » ;
- Les couleurs réalisables à partir des couleurs primaires sont maximamente saturées sur les rebords du triangle ;
- Mélanger deux couleurs C_1 et C_2 par une combinaison linéaire crée des couleurs intermédiaires sur le segment de droite dont les extrémités sont C_1 et C_2 ;

On aimerait bien aussi que

- Si on mélange le « blanc » et une couleur on obtient une version moins saturée de la couleur ; mais ce n'est pas ce que l'on observe, car la couleur se rapproche du point blanc mais en traversant des *teintes* différentes, ce qui est bien embêtant. Dans cet espace, le chemin entre le point blanc et la couleur désirée qui conserve la teinte est *courbé*.
- Qu'il soit facile d'avoir une intuition de la couleur qui résulte de X, Y et Z , mais ce n'est pas le cas.

3.2.4 CIÉ 1931 et les autres espaces de couleurs.

Tous les autres espaces de couleurs standards que nous verrons dans ce chapitre définissent (au moins) trois couleurs primaires. Si ces couleurs primaires correspondent toutes assez bien aux

couleurs primaires rouge, vert et bleu, elles ne sont pas forcément monochromatiques. Le standard CIÉ impose le rouge monochromatique à 700nm, le vert à 546.1nm et le bleu à 435.8nm, mais d'autres standards vont utiliser des couleurs primaires complexes. Pour les définir sans ambiguïté, et avec grande précision, ils les définissent très souvent en termes des coordonnées xyz obtenues à partir des coordonnées XYZ de la façon suivante :

$$\begin{aligned}x &= \frac{X}{X + Y + Z} \\y &= \frac{Y}{X + Y + Z} \\z &= \frac{Z}{X + Y + Z} = 1 - x - y\end{aligned}$$

et donc le z peut être laissé implicite.

3.2.5 Conversions entre CIÉ XYZ 1931 et RGB .

La conversion entre XYZ et RGB CIÉ 1931 (tel que défini par les couleurs monochromatiques décrite à la § 3.2.1) pour $0 \leq r, g, b \leq 1$ est donnée [46, p. 100–101] par

$$\begin{bmatrix} 0.48698 & 0.31001 & 0.20000 \\ 0.17696 & 0.81240 & 0.01064 \\ 0 & 0.01 & 0.98999 \end{bmatrix} \begin{bmatrix} r \\ g \\ b \end{bmatrix} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}. \quad (3.2.1)$$

et l'inverse par

$$\begin{bmatrix} 2.3647 & -0.90302 & -0.47143 \\ -0.51884 & 1.42781 & 0.08947 \\ 0.00524 & -0.01442 & 1.00921 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} r \\ g \\ b \end{bmatrix} \quad (3.2.2)$$

Vous pouvez vous interroger sur les valeurs suspectement précises utilisées étant donné que les observations ont été obtenues avec assez peu de sujets; d'autres auteurs ne s'en bâorent pas et arrondissent considérablement les valeurs [86].

3.3 Les espaces de couleurs linéaires

Dans cette section, nous présentons quelques exemples d'espaces de couleurs linéaires, c'est-à-dire ceux dont la transformation de et vers RGB peut s'exprimer en termes de produits matrice/vecteur. Les espaces de couleurs sont aussi présentés approximativement selon une gradation de la complexité des idées qui ont mené à leur création. Enfin, en fin de section, nous reviendrons sur ce que nous avons appris et nous présenterons une « généalogie » des espaces de couleurs linéaires qui clarifiera, nous l'espérons, les relations entre les différents standards.

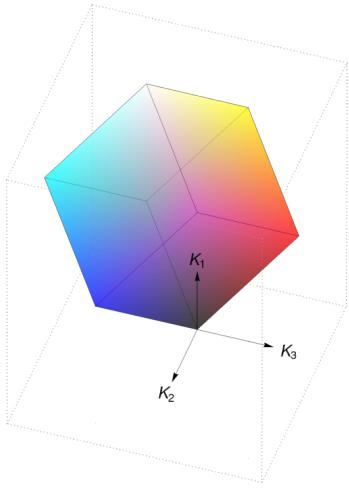


Figure 3.3.1 — L'espace de couleurs Kodak-1.

3.3.1 Kodak 1

La matrice de transformation de *RGB* à Kodak-1 est donnée¹ par

$$\begin{bmatrix} 1 & 1 & 1 \\ -1 & -1 & 1 \\ 1 & -1 & -1 \end{bmatrix} \begin{bmatrix} r \\ g \\ b \end{bmatrix} = \begin{bmatrix} k_1 \\ k_2 \\ k_3 \end{bmatrix} \quad (3.3.1)$$

et de Kodak-1 à *RGB* par

$$\frac{1}{2} \begin{bmatrix} 1 & 0 & 1 \\ 0 & -1 & -1 \\ 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} k_1 \\ k_2 \\ k_3 \end{bmatrix} = \begin{bmatrix} r \\ g \\ b \end{bmatrix} \quad (3.3.2)$$

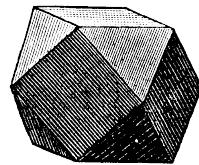
Cet espace de couleur n'est pas idéal, car k_1 est approximativement la luminance, k_2 est une différence jaune-bleu, mais k_3 est une différence rouge-cyan (plutôt que rouge-vert) comme attendu pour bien modéliser la réponse visuelle. La transformation du cube *RGB* dans l'espace de couleurs Kodak-1 est montré à la fig. 3.3.1.

Il néglige aussi de compacter les bits : si r , g et b sont des valeurs allant de 0 à 255, nous avons

$$\begin{aligned} 0 &\leq k_1 \leq 765 = 3 \times 255, \\ -510 &\leq k_2 \leq 255, \\ -510 &\leq k_3 \leq 255, \end{aligned}$$

1. J'ai pris note de cette transformation dans un cahier de laboratoire pendant mon séjour à Bell Labs en 1996. Naturellement, je n'ai pas retenu la référence de la source et il semble qu'elle soit maintenant « tombée de l'Internet », car je n'ai pu la retrouver. Mon moi du passé a appris de ses erreurs ; n'oubliez pas de noter correctement vos références !

ce qui ne tient plus sur 8 bits par composantes, mais sur 10 bits, ce qui est bien embêtant. De plus, les coordonnées dans l'espace Kodak-1 sont assez peu intuitives : si le noir est bien à $(0,0,0)$, le blanc est à la position $(3,-1,-1)$. Cependant, il a l'avantage de se calculer en nombres entiers seulement et d'être parfaitement réversible, c'est-à-dire qu'on peut partir de RGB , aller dans Kodak-1 et revenir exactement aux mêmes valeurs RGB .



Démonstration 3.3.1. *Réversibilité de Kodak-1.* La démonstration de la réversibilité est directe :

$$\begin{bmatrix} 1 & 1 & 1 \\ -1 & -1 & 1 \\ 1 & -1 & -1 \end{bmatrix} \begin{bmatrix} r \\ g \\ b \end{bmatrix} = \begin{bmatrix} r+g+b \\ -r-g+b \\ r-g-b \end{bmatrix}$$

puis

$$\begin{aligned} \frac{1}{2} \begin{bmatrix} 1 & 0 & 1 \\ 0 & -1 & -1 \\ 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} r+g+b \\ -r-g+b \\ r-g-b \end{bmatrix} &= \frac{1}{2} \begin{bmatrix} (r+g+b)+(r-g-b) \\ (-r-g+b)+(r-g-b) \\ (r+g+b)+(-r-g+b) \end{bmatrix} \\ &= \frac{1}{2} \begin{bmatrix} 2r \\ 2g \\ 2b \end{bmatrix} = \begin{bmatrix} r \\ g \\ b \end{bmatrix}. \end{aligned}$$

□

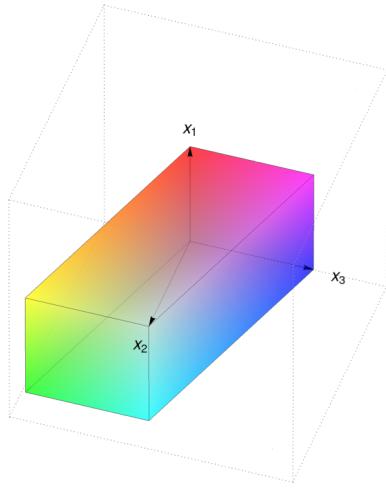


Figure 3.3.2 — L'espace de couleurs Kodak *Lossless*.

3.3.2 Kodak *Lossless*

L'intérêt pour les espaces de couleurs parfaitement réversible est clair : cela permet de compresser sans perte des images. Un de ces espaces de couleurs se retrouve dans le standard JPEG sans perte [221, 254, 266]. La transformation de *RGB* vers l'espace Kodak *Lossless* (« sans perte »), montré à fig. 3.3.2, est donnée par

$$\begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} r \\ g \\ b \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad (3.3.3)$$

et de Kodak *Lossless* à *RGB* par

$$\begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} r \\ g \\ b \end{bmatrix} \quad (3.3.4)$$

Il est alors facile de vérifier que la transformation est exactement réversible (sur le modèle de la démonstration 3.3.1). Il existe une variante qui joue sur les propriétés de l'arithmétique entière (qu'utilisent des langages de programmation comme C et C++) et qui, pour cette raison, ne s'exprime pas bien comme un produit matrice/vecteur. Nous avons

$$\begin{aligned} x_1 &= r - g \\ x_2 &= g \\ x_3 &= b - \left\lfloor \frac{r + g}{2} \right\rfloor \end{aligned} \quad (3.3.5)$$

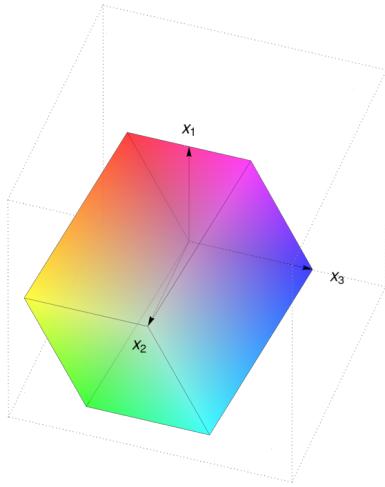


Figure 3.3.3 — L'espace de couleurs Kodak *Lossless*, seconde variante.

et l'inverse est donné par

$$\begin{aligned} r &= x_1 + x_2 \\ g &= x_2 \\ b &= x_3 + \left\lfloor \frac{x_1 + 2x_2}{2} \right\rfloor = x_3 + \left\lfloor \frac{x_1}{2} \right\rfloor + x_2 \end{aligned} \tag{3.3.6}$$

La transformation du cube *RGB* qui résulte de cette seconde transformation est montrée à la fig. 3.3.3. L'une comme l'autre donne des coefficients tels que

$$-255 \leq x_1 \leq 255,$$

$$0 \leq x_2 \leq 255,$$

$$-255 \leq x_3 \leq 255,$$

ce qui ne préserve pas le nombre original de bits.

3.3.3 Kodak *YCC*

Les transformées précédentes donnent une pondération égale à chaque composante de couleur. Cependant, comme nous le montre la fig. 3.2.1, p. 38, l'œil n'y est pas sensible également. Il convient donc d'inclure ces pondérations dans l'espace de couleurs [9, 17, 85]. Celles-ci sont

$$\begin{aligned} \omega_r &= 0.299, \\ \omega_g &= 0.587, \\ \omega_b &= 0.114. \end{aligned} \tag{3.3.7}$$

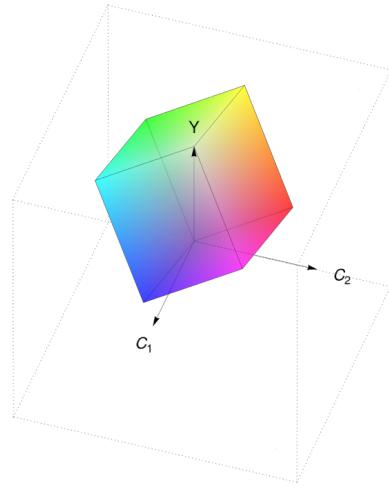


Figure 3.3.4 — L'espace de couleurs Kodak YCC .

Il paraît alors raisonnable d'inclure ces sensibilités dans l'espace de couleurs, et si possible d'une façon « équilibrée », c'est-à-dire de telle façon que les coordonnées dans l'espace de couleurs soient aussi centrées sur zéro que possible.

L'espace Kodak YCC , montré à la fig. 3.3.4, est un espace simple qui fait usage de ces pondérations [4, 193, 262]. Strictement, l'espace Kodak YCC n'est pas linéaire : les composantes RGB sont transformées par une espèce de correction gamma. Pour la composante c , on a

$$c' = \begin{cases} 1.099c^{0.45} - 0.099 & \text{si } c \geq 0.018 \\ 4.5c & \text{sinon.} \end{cases} \quad (3.3.8)$$

Une fois les composantes r , g et b « corrigées » en r' , g' et b' , la transformation est donnée par

$$\begin{bmatrix} \omega_r & \omega_g & \omega_b \\ -\omega_r & -\omega_g & 1-\omega_b \\ 1-\omega_r & -\omega_g & -\omega_b \end{bmatrix} \begin{bmatrix} r' \\ g' \\ b' \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.299 & -0.587 & 0.886 \\ 0.701 & -0.587 & -0.114 \end{bmatrix} \begin{bmatrix} r' \\ g' \\ b' \end{bmatrix} = \begin{bmatrix} Y \\ C_1 \\ C_2 \end{bmatrix} \quad (3.3.9)$$

et l'inverse est donné par

$$\begin{bmatrix} \omega_r & \omega_g & \omega_b \\ -\omega_r & -\omega_g & 1-\omega_b \\ 1-\omega_r & -\omega_g & -\omega_b \end{bmatrix}^{-1} \begin{bmatrix} Y \\ C_1 \\ C_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 \\ 1 & -\frac{\omega_b}{\omega_g} & -\frac{\omega_r}{\omega_g} \\ 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} Y \\ C_1 \\ C_2 \end{bmatrix}$$

$$\approx \begin{bmatrix} 1 & 0 & 1 \\ 1 & -0.194 & -0.509 \\ 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} Y \\ C_1 \\ C_2 \end{bmatrix} = \begin{bmatrix} r' \\ g' \\ b' \end{bmatrix}. \quad (3.3.10)$$

Cet espace de couleur utilise non seulement les pondérations pour la sensibilité aux couleurs, mais

aussi tente de maintenir l'espace balancé, c'est-à-dire approximativement centré sur l'axe Y . Ainsi, nous avons $1 - \omega_r$, qui s'oppose à $-\omega_g - \omega_b$ car $1 - \omega_r = \omega_g + \omega_b$, comme nous avons $1 - \omega_b$ qui s'oppose à $-\omega_r - \omega_g$, pour la même raison.

3.3.4 Xerox YES

Cet espace de couleurs spécifie des couleurs primaires particulières en terme de leurs coordonnées CIÉ ($x, y, z = 1$) [5, 144],

$$\begin{array}{ll} R & x = 0.630 \quad y = 0.340 \\ G & x = 0.310 \quad y = 0.595 \\ B & x = 0.155 \quad y = 0.070 \end{array}$$

probablement pour « bien jouer » avec le matériel de reproduction d'image de Xerox. La transformation de RGB (avec leurs couleurs primaires spéciales) à YES est donnée par

$$\begin{bmatrix} 0.253 & 0.684 & 0.063 \\ \frac{1}{2} & -\frac{1}{2} & 0 \\ \frac{1}{4} & \frac{1}{4} & -\frac{1}{2} \end{bmatrix} \begin{bmatrix} r_x \\ g_x \\ b_x \end{bmatrix} = \begin{bmatrix} Y \\ E \\ S \end{bmatrix} \quad (3.3.11)$$

tandis que l'inverse est donné par

$$\begin{bmatrix} 1 & 1.431 & 0.126 \\ 1 & -0.569 & 0.126 \\ 1 & 0.431 & -1.874 \end{bmatrix} \begin{bmatrix} Y \\ E \\ S \end{bmatrix} = \begin{bmatrix} r_x \\ g_x \\ b_x \end{bmatrix}. \quad (3.3.12)$$

L'espace de couleurs YES est montré à la fig. 3.3.5.

3.3.5 Espaces Ohta

Alors que la plupart des autres espaces de couleurs veulent modéliser, au moins approximativement, la réponse de l'œil aux couleurs, les espaces d'Ohta (ce n'est pas un acronyme mais un patronyme) proposent une approche radicalement différente. L'idée d'Ohta est que l'espace de couleurs devrait supporter efficacement la distribution des couleurs dans l'image et non la réponse de l'œil. Pour ce faire, nous trouvons les axes principaux du nuage de points formé par les couleurs de l'image grâce au calcul des composantes principales (voir § 3.5). Ces axes sont orthogonaux entre eux, de longueur proportionnelle à la variance du nuage de point dans la direction qui leur correspond et la fig. 3.3.6 montre un exemple en 2 dimensions.

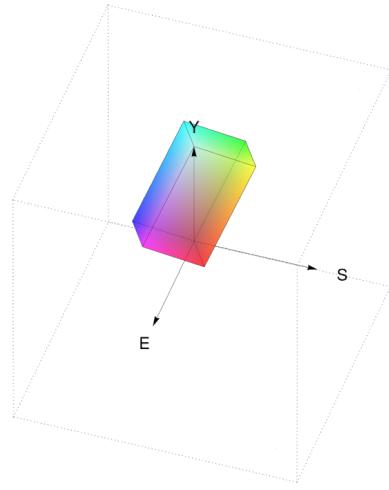


Figure 3.3.5 — L'espace de couleurs Xerox YES.

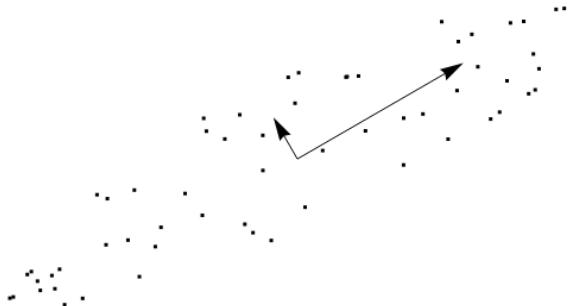


Figure 3.3.6 — Les axes principaux d'un patatoïde.

Or, plutôt que de trouver ces vecteurs optimaux pour chaque nouvelle image (ce qui s'avère pas très compliqué, mais pas trivial non plus [140]), Ohta propose simplement de choisir des vecteurs de base qui sont, en général, à peu près dans les bonnes directions. Il propose les deux espaces de couleurs suivants [203, 204]. Le premier espace qu'il propose, nommons-le Ohta-1, est donné par

$$\begin{bmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ \pm\frac{1}{2} & 0 & \mp\frac{1}{2} \\ -\frac{1}{4} & \frac{1}{2} & \frac{1}{4} \end{bmatrix} \begin{bmatrix} r \\ g \\ b \end{bmatrix} = \begin{bmatrix} I_1 \\ I_2 \\ I_3 \end{bmatrix} \quad (3.3.13)$$

avec son inverse (considérant la seconde rangée $(\frac{1}{2}, 0, -\frac{1}{2})$), car il propose d'utiliser « l'une ou l'autre » combinaisons de signes indifféremment [204, p. 228]) :

$$\begin{bmatrix} 1 & 1 & -\frac{2}{3} \\ 1 & 0 & \frac{4}{3} \\ 1 & -1 & -\frac{2}{3} \end{bmatrix} \begin{bmatrix} I_1 \\ I_2 \\ I_3 \end{bmatrix} = \begin{bmatrix} r \\ g \\ b \end{bmatrix}. \quad (3.3.14)$$

Il prétend par ailleurs que cet espace de couleurs remplace les axes calculés exactement pour

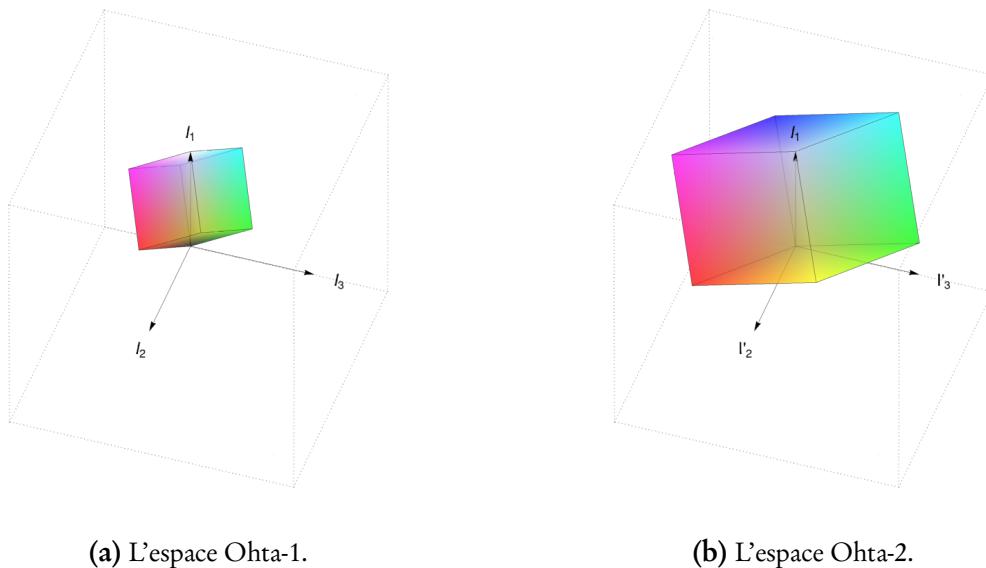


Figure 3.3.7 — Les espaces de couleurs Ohta.

le nuage de point « sans dégradation » [203, p. 28], mais sans véritablement offrir d'arguments quantifiés¹. Le second espace de couleurs qu'il propose, Ohta-2, utilise la transformation suivante :

$$\begin{bmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ 1 & 0 & -1 \\ -\frac{1}{2} & 1 & \frac{1}{2} \end{bmatrix} \begin{bmatrix} r \\ g \\ b \end{bmatrix} = \begin{bmatrix} I_1 \\ I'_2 \\ I'_3 \end{bmatrix} \quad (3.3.15)$$

où vous aurez compris que c'est essentiellement une mise à l'échelle de deux rangées de la matrice de transformation pour le premier espace. L'inverse est donné par

$$\begin{bmatrix} 1 & \frac{1}{2} & -\frac{1}{3} \\ 1 & 0 & \frac{2}{3} \\ 1 & -\frac{1}{2} & -\frac{1}{3} \end{bmatrix} \begin{bmatrix} I_1 \\ I'_2 \\ I'_3 \end{bmatrix} = \begin{bmatrix} r \\ g \\ b \end{bmatrix}. \quad (3.3.16)$$

Enfin, les cubes *RGB* transformés dans les deux espaces d'Ohta sont montrés à la fig. 3.3.7.

3.3.6 YC_oC_g

Proposé par Malvar *et al.*, l'espace de couleur YC_oC_g répond à deux besoins distincts : d'une part, un calcul extrêmement rapide de la transformation de YC_oC_g à *RGB*, d'autre part une possibilité de rendre la conversion de *RGB* à YC_oC_g puis de nouveau à *RGB* sans perte [178–180, 248].

1. Mais pour être rigoureux, et obtenir les vraies composantes principales, il faudrait aussi appliquer un biais sur les couleurs, c'est-à-dire leur soustraire la couleur moyenne. Ohta triche encore un peu en plaçant ce « centre » à (0,0,0).

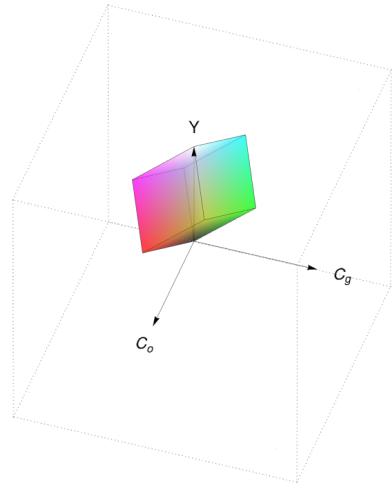


Figure 3.3.8 — L'espace de couleurs YC_oC_g .

Contrairement aux espaces Ohta qui donnent une pondération égale aux composantes rouges, vertes et bleues, l'espace YC_oC_g donne prépondérance au vert :

$$\begin{bmatrix} \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ \frac{1}{2} & 0 & -\frac{1}{2} \\ -\frac{1}{4} & \frac{1}{2} & -\frac{1}{4} \end{bmatrix} \begin{bmatrix} r \\ g \\ b \end{bmatrix} = \begin{bmatrix} Y \\ C_o \\ C_g \end{bmatrix}. \quad (3.3.17)$$

La transformée inverse, comme promis, est particulièrement simple :

$$\begin{bmatrix} 1 & 1 & -1 \\ 1 & 0 & 1 \\ 1 & -1 & -1 \end{bmatrix} \begin{bmatrix} Y \\ C_o \\ C_g \end{bmatrix} = \begin{bmatrix} r \\ g \\ b \end{bmatrix}. \quad (3.3.18)$$

L'espace de couleurs YC_oC_g est montré à la fig. 3.3.8. Curiosité, les indices o et g sont censés représenter orange et vert, respectivement, mais C_o n'est pas une « différence orange », mais une différence rouge-bleue ; C_g est la différence vert-magenta.

Par contre, la transformation de RGB à YC_oC_g n'est pas immédiatement parfaitement réversible, car il faut garder deux bits après le point (à cause des divisions par 4). Sachant que pour une matrice A et une constante $k \neq 0$, nous avons que

$$(kA)^{-1} = A^{-1}k^{-1} = \frac{1}{k}A^{-1},$$

il nous est possible de mettre les deux matrices (transformée et inverse) à l'échelle pour obtenir de l'arithmétique en nombres entiers seulement. Certes :

$$4 \begin{bmatrix} \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ \frac{1}{2} & 0 & -\frac{1}{2} \\ -\frac{1}{4} & \frac{1}{2} & -\frac{1}{4} \end{bmatrix} \begin{bmatrix} r \\ g \\ b \end{bmatrix} = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 0 & -2 \\ -1 & 2 & -1 \end{bmatrix} \begin{bmatrix} r \\ g \\ b \end{bmatrix} = \begin{bmatrix} Y' \\ C'_o \\ C'_g \end{bmatrix}. \quad (3.3.19)$$

et l'inverse est donné par

$$\frac{1}{4} \begin{bmatrix} 1 & 1 & -1 \\ 1 & 0 & 1 \\ 1 & -1 & -1 \end{bmatrix} \begin{bmatrix} Y' \\ C'_o \\ C'_g \end{bmatrix} = \begin{bmatrix} r \\ g \\ b \end{bmatrix}. \quad (3.3.20)$$

Ce qui nous donne bien une transformation parfaitement réversible (mais dont la bande dynamique est augmentée à 10 bits par composante).

Démonstration 3.3.2.

Réversibilité de YC_oC_g mis à l'échelle. Encore une fois, la démonstration est directe :

$$\begin{bmatrix} 1 & 2 & 1 \\ 2 & 0 & -2 \\ -1 & 2 & -1 \end{bmatrix} \begin{bmatrix} r \\ g \\ b \end{bmatrix} = \begin{bmatrix} r+2g+b \\ 2(r-b) \\ -r+2g-b \end{bmatrix},$$

et

$$\begin{aligned} & \frac{1}{4} \begin{bmatrix} 1 & 1 & -1 \\ 1 & 0 & 1 \\ 1 & -1 & -1 \end{bmatrix} \begin{bmatrix} r+2g+b \\ 2(r-b) \\ -r+2g-b \end{bmatrix} \\ &= \frac{1}{4} \begin{bmatrix} (r+2g+b)+2(r-b)-(-r+2g-b) \\ (r+2g+b)+(-r+2g-b) \\ (r+2g+b)-2(r-b)-(-r+2g-b) \end{bmatrix} \\ &= \frac{1}{4} \begin{bmatrix} 4r \\ 4g \\ 4b \end{bmatrix} = \begin{bmatrix} r \\ g \\ b \end{bmatrix}. \end{aligned}$$

□

Enfin, Malvar *et al.* proposent une version réversible par « lifting », essentiellement une astuce qui rétablit la parité du résultat sans avoir recours à des bits après point. Pour deux entiers x et y , posons

$$d = x - y,$$

$$m = y + \left\lfloor \frac{d}{2} \right\rfloor.$$

Nous pouvons retrouver exactement x et y car

$$y = m - \left\lfloor \frac{d}{2} \right\rfloor,$$

$$x = y + d.$$

En appliquant cette astuce :

$$\begin{aligned} C_o &= r - b, \\ t &= b + \left\lfloor \frac{C_o}{2} \right\rfloor, \\ C_g &= g - t, \\ Y &= t + \left\lfloor \frac{C_g}{2} \right\rfloor, \end{aligned}$$

ce qui nous permet de retrouver nos composantes *RGB* originales :

$$\begin{aligned} t &= y - \left\lfloor \frac{C_g}{2} \right\rfloor, \\ g &= C_g + t, \\ b &= t - \left\lfloor \frac{C_o}{2} \right\rfloor, \\ r &= b + C_o. \end{aligned}$$

3.3.7 JPEG 2000 : *YVU-R*

JPEG 2000 est un standard développé pour remplacer JPEG. Cependant, pour toutes sortes de raisons, et malgré que ce soit un meilleur standard de compression, il n'a pas su s'imposer. Il propose d'utiliser les espaces de couleurs YC_bC_r (dont nous discuterons plus loin) et $YVU-R$, une transformation exactement réversible (d'où le $-R$). Cette transformée est presque linéaire, car, comme YC_oC_g , elle exploite les propriétés de l'arithmétique entière.

La transformation de *RGB* à *YVU-R* est donnée par [239, p. 40] :

$$\begin{aligned} Y_r &= \left\lfloor \frac{r + 4g + b}{4} \right\rfloor, \\ V_r &= r - g, \\ U_r &= b - g, \end{aligned} \tag{3.3.21}$$

et de *YVU-R* à *RGB* par

$$\begin{aligned} g &= Y_r - \left\lfloor \frac{V_r + U_r}{4} \right\rfloor, \\ r &= V_r + g, \\ b &= U_r + g. \end{aligned} \tag{3.3.22}$$

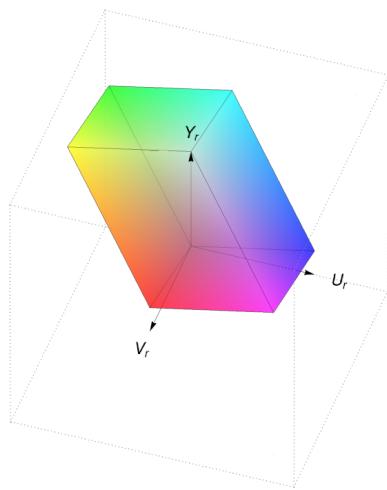


Figure 3.3.9 — L'espace de couleurs $YVU\text{-}R$.

Cette transformée est réversible mais demande un bit de plus pour les composantes V_r et U_r . L'espace $YVU\text{-}R$ est montré à la fig. 3.3.9.

Démonstration 3.3.3. *Réversibilité de $YVU\text{-}R$.* Il suffit de montrer que nous récupérons la composante g :

$$\begin{aligned}
 g &= Y_r - \left\lfloor \frac{V_r + U_r}{4} \right\rfloor \\
 &= \left\lfloor \frac{r + 2g + b}{4} \right\rfloor - \left\lfloor \frac{(r - g) + (b - g)}{4} \right\rfloor \\
 &= \left\lfloor \frac{r + 2g + b}{4} \right\rfloor - \left\lfloor \frac{r - 2g + b}{4} \right\rfloor \\
 &= \left\lfloor \frac{r + 2g + b}{4} \right\rfloor + \left\lfloor \frac{-r + 2g - b}{4} \right\rfloor \\
 &= \left\lfloor \frac{4g}{4} \right\rfloor \\
 &= g.
 \end{aligned}$$

□

3.3.8 Série S

Notes : Les espaces S viennent de mes notes préparatoires pour le doctorat, développés pour explorer le problème des espaces de couleurs linéaires. Même s'ils ne sont pas « standards », ils serviront ici à comprendre des manipulations nécessaires à la construction d'autres espaces de couleurs.

Parmi les propriétés désirables pour un espace de couleur, on trouve qu'une des composantes devrait correspondre directement à la luminosité de la couleur, et en particulier aux tons de gris lorsque les deux autres composantes sont nulles — cela permet, par exemple, de créer une image en noir et blanc à partir d'une seule composante de l'espace de couleur. Préférablement, les deux autres composantes devraient contenir moins d'information et correspondre, au moins vaguement, aux différences rouge-vert et jaune-bleu ; la teinte et la saturation n'étant vraiment réalisables qu'avec des transformations non-linéaires.

Il est alors possible de prendre le cube *RGB*, lui appliquer une rotation adéquate pour ramener la grande diagonale du cube — à laquelle correspondent les tons de gris — sur un des axes, disons s_{11} ; et peut-être lui appliquer une rotation supplémentaire pour qu'un des axes correspondent à la différence jaune-bleu ou rouge-vert. Mais pour ce faire, nous aurons besoin des matrices de rotations en trois dimensions (que nous donnons ici sans démonstration). La matrice de rotation en sens *antihoraire*¹ autour de l'axe des R est donnée par

$$R_r(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & \sin \alpha \\ 0 & -\sin \alpha & \cos \alpha \end{bmatrix}, \quad (3.3.23)$$

celle autour de l'axe des G par

$$R_g(\alpha) = \begin{bmatrix} \cos \alpha & 0 & \sin \alpha \\ 0 & 1 & 0 \\ -\sin \alpha & 0 & \cos \alpha \end{bmatrix}, \quad (3.3.24)$$

tandis que celle autour de l'axe des b est donnée par

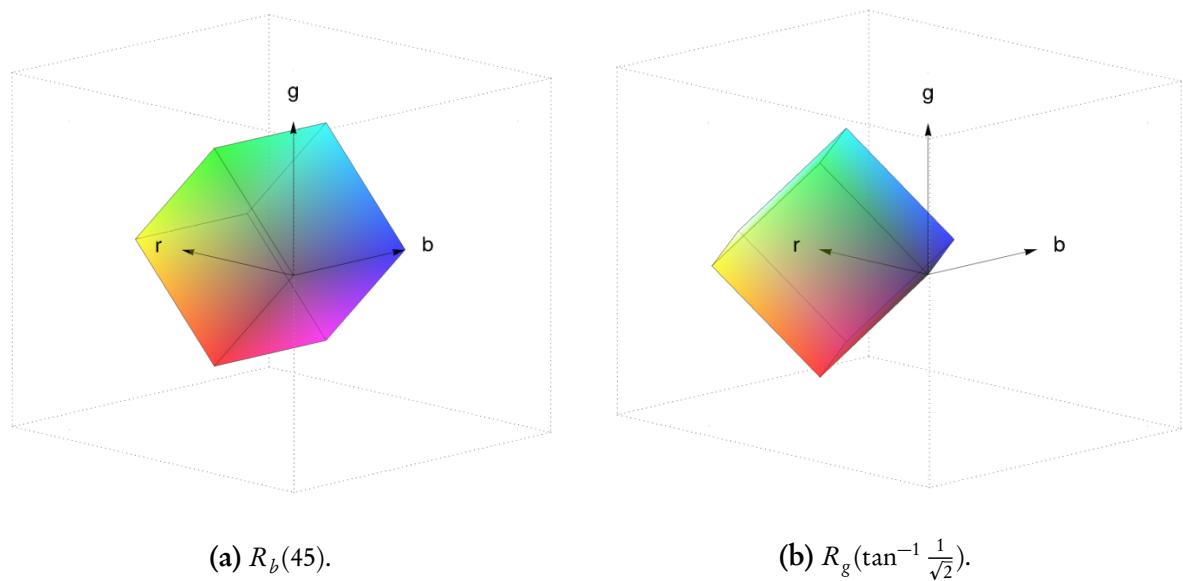
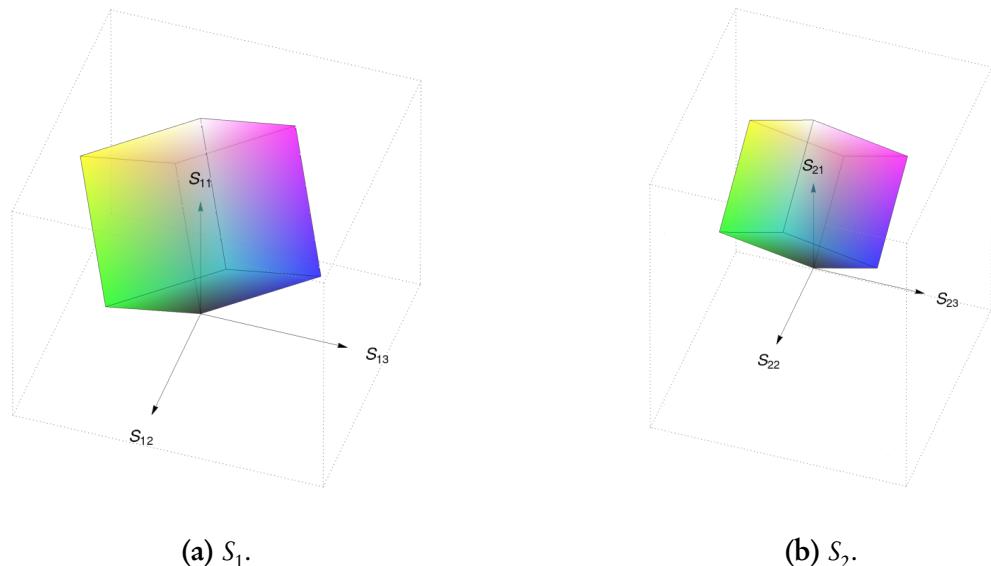
$$R_b(\alpha) = \begin{bmatrix} \cos \alpha & \sin \alpha & 0 \\ -\sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (3.3.25)$$

Utilisons donc ces rotations pour ramener la grande diagonale du cube *RGB* sur l'un des axes de notre espace — le premier. Cette rotation composée est donnée par

$$S_1 = R_g(\tan^{-1} \frac{1}{\sqrt{2}}) R_b(45^\circ). \quad (3.3.26)$$

Cette transformation applique d'abord une rotation au cube *RGB* sur l'axe B de façon à amener $(1, 1, 0)$ (rouge et vert maximum) à $(\sqrt{2}, 0, 0)$ (voir fig. 3.3.10 (a)), puis une rotation sur l'axe des G pour ramener le coin $(\sqrt{2}, 0, 1)$ à $(\sqrt{3}, 0, 0)$ (voir fig. 3.3.10 (b)). Le résultat final, en plaçant l'axe S_{11} à

1. C'est-à-dire le « bon » sens !

Figure 3.3.10 — La construction de l'espace S_1 .Figure 3.3.11 — Les espaces de couleurs S_1 et S_2 .

la verticale, est montré à la fig. 3.3.11. La matrice est donc

$$S_1 = R_g(\tan^{-1} \frac{1}{\sqrt{2}})R_b(45^\circ) = \begin{bmatrix} \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ -\frac{1}{\sqrt{6}} & -\frac{1}{\sqrt{6}} & \sqrt{\frac{2}{3}} \end{bmatrix} \quad (3.3.27)$$

donc

$$\begin{bmatrix} \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ -\frac{1}{\sqrt{6}} & -\frac{1}{\sqrt{6}} & \sqrt{\frac{2}{3}} \end{bmatrix} \begin{bmatrix} r \\ g \\ b \end{bmatrix} = \begin{bmatrix} s_{11} \\ s_{12} \\ s_{13} \end{bmatrix} \quad (3.3.28)$$

Une propriété intéressante des matrices de rotations, c'est qu'elles sont orthonormales; et une composition de matrices de rotation est une matrice de rotation. La matrice de transformation de l'espace S_1 est donc orthonormale, et nous avons $S_1^{-1} = S_1^T$. La conversion de S_1 à RGB est donc

$$\begin{bmatrix} \frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{6}} \\ \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{6}} \\ \frac{1}{\sqrt{3}} & 0 & \sqrt{\frac{2}{3}} \end{bmatrix} \begin{bmatrix} s_{11} \\ s_{12} \\ s_{13} \end{bmatrix} = \begin{bmatrix} r \\ g \\ b \end{bmatrix}. \quad (3.3.29)$$

Le premier axe, S_{11} correspond à la luminosité et si S_{13} correspond à la différence jaune-bleu, S_{12} correspond à la différence aquamarine-fuchsia, une caractéristique très quelconque. On pourrait vouloir (pour une raison ou une autre) une orientation différente, par exemple avoir un axe qui correspond à la différence rouge-cyan et un autre à la différence mauve-lime. Il faudrait alors appliquer une rotation autour de S_{11} de 30° . La transformation serait alors

$$S_2 = R_r(30^\circ)R_g(\tan^{-1}\frac{1}{\sqrt{2}})R_b(45^\circ) = \begin{bmatrix} \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} \\ -\sqrt{\frac{2}{3}} & \frac{1}{\sqrt{6}} & \frac{1}{\sqrt{6}} \\ 0 & -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} \quad (3.3.30)$$

et donc

$$\begin{bmatrix} \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} \\ -\sqrt{\frac{2}{3}} & \frac{1}{\sqrt{6}} & \frac{1}{\sqrt{6}} \\ 0 & -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} r \\ g \\ b \end{bmatrix} = \begin{bmatrix} s_{21} \\ s_{22} \\ s_{23} \end{bmatrix}. \quad (3.3.31)$$

Puisque S_2 est orthonormale, $S_2^{-1} = S_2^T$, et

$$\begin{bmatrix} \frac{1}{\sqrt{3}} & -\sqrt{\frac{2}{3}} & 0 \\ \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{6}} & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{6}} & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} s_{21} \\ s_{22} \\ s_{23} \end{bmatrix} = \begin{bmatrix} r \\ g \\ b \end{bmatrix}. \quad (3.3.32)$$

L'espace de couleur S_2 est montré à la fig. 3.3.11 (b).

Nous avons conçu deux espaces de couleurs qui ont des propriétés généralement désirables pour les espaces de couleurs : la grande diagonale du cube RGB est aligné avec un des axes de l'espace de couleurs et les autres dimensions correspondent, au moins partiellement, aux caractéristiques de

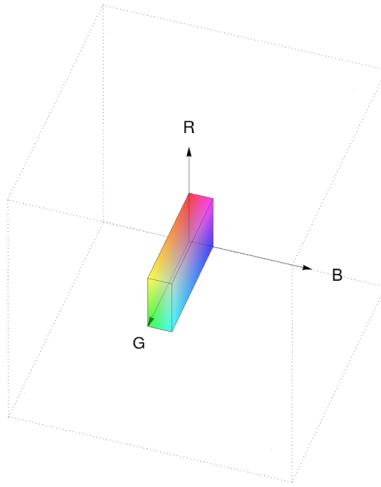


Figure 3.3.12 — Le cube *RGB* « compressé » par les poids ω_r , ω_g et ω_b .

l’œil. Cependant, les valeurs maximales ne sont pas bornées de façon intéressante. Par exemple, pour S_1 nous avons

$$\begin{aligned} 0 &\leq s_{11} \leq \sqrt{3}, \\ -\frac{1}{\sqrt{2}} &\leq s_{12} \leq \frac{1}{\sqrt{2}}, \\ -\sqrt{\frac{2}{3}} &\leq s_{13} \leq \sqrt{\frac{2}{3}}, \end{aligned}$$

ce qui est numériquement quelconque. De plus, ni S_1 ni S_2 ne tiennent compte de la sensibilité de l’œil aux différentes couleurs primaires. L’éq. (3.3.7), p. 46, donne les poids ω_r , ω_g et ω_b aux trois couleurs primaires *RGB*. La fig. 3.3.12 montre l’importance relative des couleurs primaires en utilisant le cube *RGB* « compressé » par les poids ω_r , ω_g et ω_b .

Cet espace maintenant mis à l’échelle ne correspond pas encore tout à fait à nos besoins : il faut encore amener la grande diagonale du parallélépipède rectangle (ou « brique ») sur un des axes de notre espace. Nous n’avons plus un cube de dimension $1 \times 1 \times 1$ mais $\omega_r \times \omega_g \times \omega_b$, ce qui complique un peu les choses. Posons donc

$$S_3 = \frac{1}{\|(\omega_r, \omega_g, \omega_b)\|} R_g \left(\tan^{-1} \frac{\omega_b}{\|(\omega_g, \omega_r)\|} \right) R_b \left(\tan^{-1} \frac{\omega_g}{\omega_r} \right) \begin{bmatrix} \omega_r & 0 & 0 \\ 0 & \omega_g & 0 \\ 0 & 0 & \omega_b \end{bmatrix}, \quad (3.3.33)$$

où $\|(\omega_g, \omega_b)\|$ est la longueur de la diagonale de la face vert-rouge du parallélépipède et $\|(\omega_r, \omega_g, \omega_b)\|$ est la longueur de la grande diagonale du parallélépipède. L’espace de couleurs

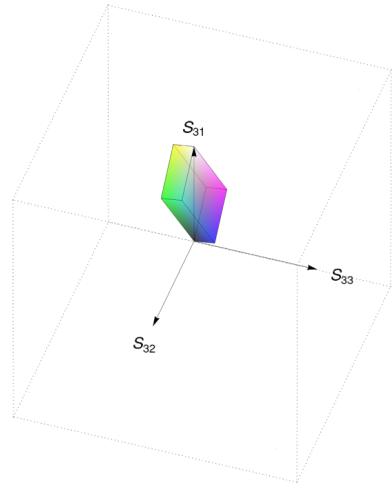


Figure 3.3.13 — L'espace de couleurs S_3 .

S_3 est montré à fig. 3.3.13. Cela nous donne les deux transformations

$$S_3 \begin{bmatrix} r \\ g \\ b \end{bmatrix} = \begin{bmatrix} 0.200 & 0.771 & 0.029 \\ -0.399 & 0.399 & 0 \\ -0.035 & -0.133 & 0.168 \end{bmatrix} \begin{bmatrix} r \\ g \\ b \end{bmatrix} = \begin{bmatrix} s_{31} \\ s_{32} \\ s_{33} \end{bmatrix} \quad (3.3.34)$$

et

$$S_3^{-1} \begin{bmatrix} s_{31} \\ s_{32} \\ s_{33} \end{bmatrix} = \begin{bmatrix} 1 & -1.990 & -0.173 \\ 1 & 0.516 & -0.173 \\ 1 & -0.006 & 5.780 \end{bmatrix} \begin{bmatrix} s_{31} \\ s_{32} \\ s_{33} \end{bmatrix} = \begin{bmatrix} r \\ g \\ b \end{bmatrix}. \quad (3.3.35)$$

Nous avons donc maintenant un espace de couleurs où la luminosité est normalisée (qui varie de 0 à 1), mais où les deux autres composantes sont quelconques ($-0.399 \leq s_{32} \leq 0.399$ et $-0.168 \leq s_{33} \leq 0.168$).

*
* * *

Si pour une raison technique (par exemple, liée à la nature du canal de transmission ou de l'algorithme de traitement d'images) nous voudrions plutôt avoir un espace de couleurs qui « remplit » notre canal, nous devrions normaliser notre espace de façon à ce que les coordonnées occupent une boîte d'encombrement (*bounding box*) donnée. Prenons par exemple l'espace S_1 . Nous

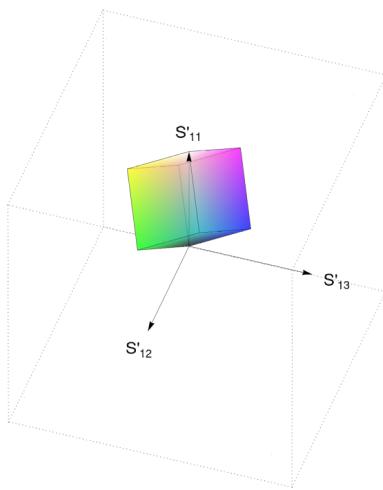


Figure 3.3.14 — L'espace de couleurs S'_1 .

avions remarqué que les coordonnées étaient

$$\begin{aligned} 0 &\leq s_{11} \leq \sqrt{3}, \\ -\frac{1}{\sqrt{2}} &\leq s_{12} \leq \frac{1}{\sqrt{2}}, \\ -\sqrt{\frac{2}{3}} &\leq s_{13} \leq \sqrt{\frac{2}{3}}. \end{aligned}$$

Pour normaliser S_1 de façon à ce que la boîte occupée varie de 0 à 1 pour s_{11} , et de $-\frac{1}{2}$ à $\frac{1}{2}$ pour les deux autres composantes, il suffit de poser

$$S'_1 \begin{bmatrix} r \\ g \\ b \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{3}} & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & \frac{1}{2\sqrt{\frac{2}{3}}} \end{bmatrix} S_1 \begin{bmatrix} r \\ g \\ b \end{bmatrix} = \begin{bmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ -\frac{1}{2} & \frac{1}{2} & 0 \\ -\frac{1}{4} & -\frac{1}{4} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} r \\ g \\ b \end{bmatrix} = \begin{bmatrix} s'_{11} \\ s'_{12} \\ s'_{13} \end{bmatrix} \quad (3.3.36)$$

Nous remarquons avec un certain bonheur que l'espace S'_1 normalisé a une matrice de transformation somme toute bien sympathique. La première matrice divise par les grandeurs désirées pour obtenir des portées voulues. Le premier coefficient doit s'assurer de normaliser la première composante sur 0 à 1, on divise donc par $\sqrt{3}$. Pour la seconde, la portée désirée est $-\frac{1}{2}$ à $\frac{1}{2}$, il faut donc diviser par deux fois le maximum (et $\frac{1}{2\frac{1}{\sqrt{2}}} = \frac{1}{\sqrt{2}}$). Enfin pour le troisième, on divise par $2\sqrt{\frac{2}{3}}$. Par un heureux concours de circonstance, tout cela se simplifie plutôt bien et donne une matrice de transformation simple.

L'inverse aussi est assez simple et est donnée par :

$$S_1'^{-1} \begin{bmatrix} s'_{11} \\ s'_{12} \\ s'_{13} \end{bmatrix} = \begin{bmatrix} 1 & -1 & -\frac{2}{3} \\ 1 & 1 & -\frac{2}{3} \\ 1 & 0 & \frac{4}{3} \end{bmatrix} \begin{bmatrix} s'_{11} \\ s'_{12} \\ s'_{13} \end{bmatrix} = \begin{bmatrix} r \\ g \\ b \end{bmatrix}. \quad (3.3.37)$$

*

* * *

La présentation de la série S nous a permis de faire le tour de la boîte à outils de création des espaces de couleurs linéaires : rotations, mise à l'échelle, normalisation. Dans ce qui suit, nous rencontrerons d'autres espaces de couleurs qui s'expliquent fort bien par la combinaisons de ces transformations de base. Il est, à notre avis, bien plus intéressant de comprendre comment un espace de couleurs est construit que de simplement connaître ses deux matrices de transformation.

3.3.9 Les espaces de couleurs de la télévision

Dans cette section, nous allons jeter un œil aux espaces de couleurs issues de la télévision, autant analogique (au temps des dinosaures) que numérique.

3.3.9.1 *Transmission primaires*

Très tôt dans l'histoire de la télévision (qui commence aussi loin que les années 1920 [91]) on a arrêté le choix des couleurs primaires [135]. Le NTSC (*National Television System Committee*, l'organisme de standardisation formé en 1941) a proposé en 1953 un standard pour la télévision en couleur et définissait les trois couleurs primaires comme étant

Rouge R_{470}	$x = 0.67$	$y = 0.33$
Vert G_{470}	$x = 0.21$	$y = 0.71$
Bleu B_{470}	$x = 0.14$	$y = 0.08$
Blanc C	$x = 0.31$	$y = 0.316$

en coordonnées xyz ($z = 1$) du CIÉ 1931. Cette proposition est plus tard devenue la recommandation BT.470 du ITU [9]. Le blanc est le blanc de l'« illuminant C », une source de lumière censée

correspondre à la lumière du jour, à 6774°K [9,52,58]. Ils établissent aussi les sensibilités relatives aux couleurs primaires, $\omega_r = 0.299$, $\omega_g = 0.587$, $\omega_b = 0.114$, déjà mentionnées à la section 3.3.3, p. 46.

Les couleurs des pixels sont représentées comme des combinaisons linéaires des couleurs primaires qu'il faut transmettre. Dans le cas particulier de la télévision couleur, il était nécessaire d'assurer la compatibilité avec la télévision noir et blanc, alors la seule télévision. Il nous faut donc transmettre les couleurs de façon à ce que le signal qui contient l'information de couleur puisse être ignoré par les appareils déjà en place, et rendre une image noir et blanc potable. Il faut donc transformer l'espace de couleurs *RGB* (avec les couleurs primaires ci-dessus) en un espace de couleurs où la luminance (l'image en noir et blanc) est séparée du signal de couleur. Ainsi, un poste noir et blanc ne décode que l'information de luminance, ce qui lui permet d'afficher une image cromulente, tandis qu'un poste couleur affichera une image couleur après avoir décodé les trois canaux¹.

Il faut donc transformer les couleurs primaires en « couleurs primaires de transmission » BT.470, en *transmission primaries*. Ces *transmission primaries* sont données par

$$\begin{aligned} Y &= \omega_r r + \omega_g g + \omega_b b, \\ P_2 &= R - Y = (1 - \omega_r)r - \omega_g g - \omega_b b, \\ P_3 &= B - Y = -\omega_r r - \omega_g g + (1 - \omega_b)b, \end{aligned} \quad (3.3.38)$$

soit donc

$$T_{p_{470}} \begin{bmatrix} r_{470} \\ g_{470} \\ b_{470} \end{bmatrix} = \begin{bmatrix} \omega_r & \omega_g & \omega_b \\ 1 - \omega_r & -\omega_g & -\omega_b \\ -\omega_r & -\omega_g & 1 - \omega_b \end{bmatrix} \begin{bmatrix} r_{470} \\ g_{470} \\ b_{470} \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.701 & -0.587 & -0.114 \\ -0.299 & -0.587 & 0.886 \end{bmatrix} \begin{bmatrix} r_{470} \\ g_{470} \\ b_{470} \end{bmatrix} = \begin{bmatrix} Y \\ P_2 \\ P_3 \end{bmatrix}. \quad (3.3.39)$$

La transformée inverse, qui nous ramène à l'espace *RGB* n'est pas difficile à calculer :

$$T_p^{-1} \begin{bmatrix} Y \\ P_2 \\ P_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & -0.509 & -0.194 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} Y \\ P_2 \\ P_3 \end{bmatrix} = \begin{bmatrix} r_{470} \\ g_{470} \\ b_{470} \end{bmatrix}. \quad (3.3.40)$$

Par la suite, d'autres transformations seront appliquées pour accommoder les différentes stratégies de transmission et l'évolution des standards.

1. Le signal de couleur, à plus faible résolution, était poussé dans les très hautes fréquences du signal de l'image, à basse amplitude. Cela avait pour effet de produire une image en noir et blanc très légèrement bruitée pour un poste qui ne savait pas interpréter les composantes de couleurs. Le détail de l'encodage dépasse le cadre du présent ouvrage, mais pour en savoir plus, consultez, par exemple, [72,75,85,107,113].

3.3.9.2 YUV et YIQ

Les *transmission primaires* sont d'abord normalisées pour permettre un encodage ultérieur par un signal en quadrature de phase [14, 78]. Les *transmission primaires* sont d'abord transformées de RGB (avec les couleurs primaires BT.470) vers l'espace de couleurs YUV .

Nous définissons¹

$$\begin{aligned} U_{\max} &= 0.436, \\ V_{\max} &= 0.615, \end{aligned}$$

et les poids

$$\begin{aligned} W_Y &= 1, \\ W_U &= \frac{U_{\max}}{1 - \omega_b}, \\ W_V &= \frac{V_{\max}}{1 - \omega_r}. \end{aligned}$$

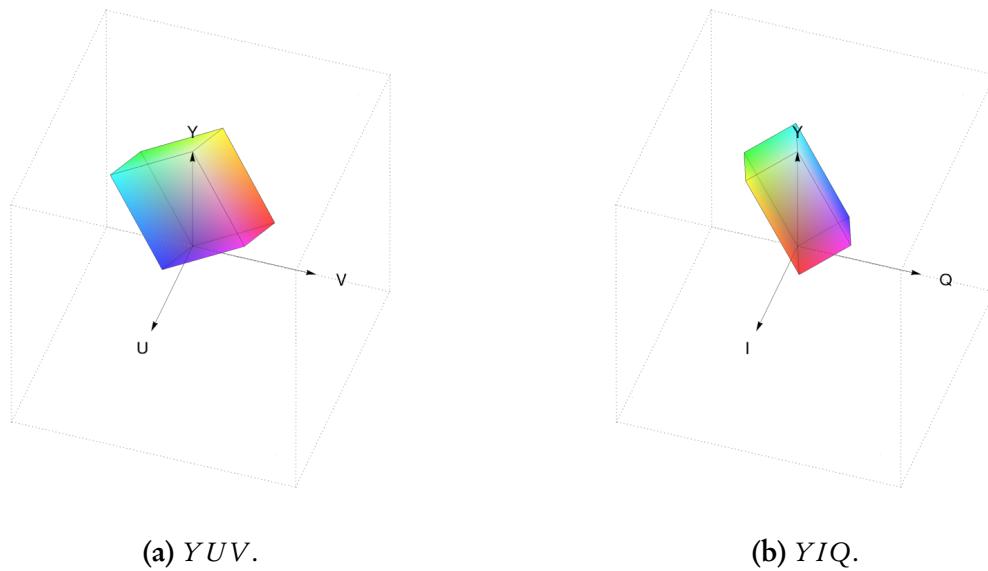
Nous définissons aussi une matrice de permutation (car l'ordre est différent des *transmission primaires*),

$$P_{YUV} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}. \quad (3.3.41)$$

Enfin, nous pouvons définir la transformation de RGB à YUV :

$$\begin{aligned} T_{YUV} \begin{bmatrix} r_{470} \\ g_{470} \\ b_{470} \end{bmatrix} &= \underbrace{\begin{bmatrix} W_Y & 0 & 0 \\ 0 & W_U & 0 \\ 0 & 0 & W_V \end{bmatrix}}_{\text{normalisation pour } YIQ} \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}}_{P_{YUV}} \underbrace{\begin{bmatrix} \omega_r & \omega_g & \omega_b \\ 1 - \omega_r & -\omega_g & -\omega_b \\ -\omega_r & -\omega_g & 1 - \omega_b \end{bmatrix}}_{T_{p470}} \begin{bmatrix} r_{470} \\ g_{470} \\ b_{470} \end{bmatrix} \\ &= \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & 0.436 \\ -0.615 & -0.515 & -0.100 \end{bmatrix} \begin{bmatrix} r_{470} \\ g_{470} \\ b_{470} \end{bmatrix} \\ &= \begin{bmatrix} Y \\ U \\ V \end{bmatrix}. \end{aligned} \quad (3.3.42)$$

1. L'origine exacte des « constantes magiques » semble liée aux circuits analogiques d'encodage et de décodage en quadrature. Cependant, bien que j'aie cherché, je n'ai pas trouvé de justification explicite pour ces constantes. Poynton donne une formule pour les calculer, mais sans la justifier [218]. Il semblerait que les constantes aient été choisies pour une compatibilité avec une unité de mesure d'amplitude, les IRE, qui varient de 0 (pour noir) à 100 (pour blanc), les valeurs négatives étant réservées pour des signaux de contrôle [16].

Figure 3.3.15 — Les espaces de couleurs YUV et YIQ .

L'inverse est donné par

$$\begin{bmatrix} 1 & 0 & 1.140 \\ 1 & -0.394 & -0.581 \\ 1 & 2.032 & 0 \end{bmatrix} \begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} r_{470} \\ g_{470} \\ b_{470} \end{bmatrix}. \quad (3.3.43)$$

L'espace YIQ transforme YUV en un espace compatible avec l'encodage par quadrature, mais il ne s'agit pas d'une simple mise à l'échelle pour occuper une boîte donnée : non ! il faut encore permuter deux rangées de YUV et lui appliquer une rotation de 33° en sens horaire. Nous revenons donc aux *transmission primaries* :

$$\begin{aligned} T_{YIQ} &= R_r(-33^\circ) \begin{bmatrix} W_Y & 0 & 0 \\ 0 & W_V & 0 \\ 0 & 0 & W_U \end{bmatrix} T_p \\ &= \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos -33^\circ & \sin -33^\circ \\ 0 & -\sin -33^\circ & \cos -33^\circ \end{bmatrix} \begin{bmatrix} W_Y & 0 & 0 \\ 0 & W_V & 0 \\ 0 & 0 & W_U \end{bmatrix} \begin{bmatrix} \omega_r & \omega_g & \omega_b \\ 1 - \omega_r & -\omega_g & -\omega_b \\ -\omega_r & -\omega_g & 1 - \omega_b \end{bmatrix}}_{\text{une permutation de } YUV} \\ &\approx \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.275 & -0.321 \\ 0.212 & -0.522 & 0.311 \end{bmatrix}. \end{aligned} \quad (3.3.44)$$

Enfin, nous avons :

$$T_{YIQ} \begin{bmatrix} r \\ g \\ b \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.275 & -0.321 \\ 0.212 & -0.522 & 0.311 \end{bmatrix} \begin{bmatrix} r_{470} \\ g_{470} \\ b_{470} \end{bmatrix} = \begin{bmatrix} Y \\ I \\ Q \end{bmatrix}, \quad (3.3.45)$$

et l'inverse

$$T_{YIQ}^{-1} \begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 1 & 0.956 & 0.621 \\ 1 & -0.275 & -0.647 \\ 1 & -1.107 & 1.704 \end{bmatrix} \begin{bmatrix} r_{470} \\ g_{470} \\ b_{470} \end{bmatrix} = \begin{bmatrix} r \\ g \\ b \end{bmatrix}. \quad (3.3.46)$$

Les deux espaces de couleurs YUV et YIQ sont montrés à la fig. 3.3.15.

*
* * *

Pourquoi faire simple quand on peut faire des standards? Le FCC (l'équivalent américain du CRTC) propose de modifier légèrement les matrices de YIQ [16, § 73.682], juste « parce que » :

$$T_{YIQ_{FCC}} = \begin{bmatrix} 0.3 & 0.59 & 0.11 \\ 0.599 & -0.2773 & -0.3217 \\ 0.213 & -0.5251 & 0.3121 \end{bmatrix} \quad (3.3.47)$$

et l'inverse est maintenant

$$T_{YIQ_{FCC}}^{-1} \approx \begin{bmatrix} 1 & 0.947 & 0.624 \\ 1 & -0.275 & -0.636 \\ 1 & -1.109 & 1.709 \end{bmatrix}. \quad (3.3.48)$$

3.3.9.3 Télévision numérique : BT.601, BT.709 et sRGB

La télévision analogique, sauf encore pour quelques rares stations, s'est éteinte. Elle a été remplacée par la télévision numérique haute définition (HDTV), rendue possible par la standardisation d'algorithmes de compression vidéo (MPEG2 qui assure la compression) et de transmission numérique (MPEG-TS qui en assure le transport par ondes ou par câble). Les résolutions supportées par la télévision numérique sont le 720p (des images de 1280×720 pixels, transmises progressivement) et le 1080p/i (des images de 1920×1080 pixels, transmises progressivement ou en mode entrelacé).

La recommandation BT.601, amendée par BT.709, définit de nouvelles couleurs primaires et de nouveaux espaces de couleurs, affranchis des mystérieux bricolages analogiques, choisis pour accommoder les nouvelles technologies des écrans plats de toutes sortes. BT.601, qui définit les paramètres pour la transition de la télévision analogique à la télévision numérique, définit les couleurs primaires de transfert [17] :

	625 lignes SÉCAM/PAL (Europe)	525 lignes NTSC (Amér./Japon)
Rouge	$x = 0.64 \quad y = 0.33$	$x = 0.63 \quad y = 0.34$
Vert	$x = 0.29 \quad y = 0.60$	$x = 0.31 \quad y = 0.595$
Bleu	$x = 0.15 \quad y = 0.06$	$x = 0.155 \quad y = 0.07$

C'est le standard BT.709 qui définit les nouvelles couleurs primaires [21] :

Rouge R_{709}	$x = 0.64$	$y = 0.33$
Vert G_{709}	$x = 0.30$	$y = 0.60$
Bleu B_{709}	$x = 0.15$	$y = 0.06$
Blanc D_{65}	$x = 0.3127$	$y = 0.3290$

Le blanc utilisé est l'illuminant D_{65} ($x = 0.3127$ et $y = 0.3290$). Cet espace RGB avec ces nouvelles couleurs primaires forment l'espace $sRGB$ (pour *standard RGB*) ou encore noté RGB_{709} . Il faut donc convertir de l'espace de couleurs analogique vers RGB_{709} à partir des couleurs primaires définies selon le standard à 625 ou 525 lignes, c'est-à-dire calculer une conversion de coordonnées de BT.470 à BT.709, avant d'appliquer la transformation de couleurs vers l'espace YC_bC_r .

*
* * *

Les couleurs primaires de BT.709 servent de base à l'espace de couleurs standard des téléviseurs mais aussi des écrans d'ordinateurs récents, $sRGB$ [7, 8]. Le standard IEC/4WD 61966-2-1 prévoit d'autres paramètres (généralement ignorés) comme la luminosité de l'écran (à 80 cd/m^2) et la couleur et l'intensité l'éclairage de la pièce environnante (350 lx de blanc D_{50} , soit $x = 0.3457$, $y = 0.3585$). Les transformations de xyz à $sRGB$ et de $sRGB$ à xyz sont données par

$$\begin{bmatrix} r_{sRGB} \\ g_{sRGB} \\ b_{sRGB} \end{bmatrix} = \begin{bmatrix} 3.2406 & -1.5372 & -0.4986 \\ -0.9689 & 1.8758 & 0.0415 \\ 0.0587 & -0.2040 & 1.0570 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (3.3.49)$$

et

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 0.4124 & 0.3576 & 0.1805 \\ 0.2126 & 0.7152 & 0.0722 \\ 0.0193 & 0.1192 & 0.9505 \end{bmatrix} \begin{bmatrix} r_{sRGB} \\ g_{sRGB} \\ b_{sRGB} \end{bmatrix}. \quad (3.3.50)$$

Le standard prévoit aussi des transformations de type correction gamma. Soient donc r' , g' et b' les valeurs $sRGB$ linéaires $0 \leq r, g, b \leq 1$ qui seront ajustées de façon non linéaire en r , g et b ¹. La

1. C'est la notation préconisée par [7].

transformation non-linéaire est, pour chacune des composantes, donnée par

$$\gamma_{sRGB}(u) = \begin{cases} u^{\frac{24}{323}} & \text{si } u \leq 0.04045 \\ \left(\frac{u + \frac{11}{200}}{1 + \frac{11}{200}}\right)^{2.4} & \text{sinon.} \end{cases} \quad (3.3.51)$$

On applique la transformation à chaque composante :

$$\begin{aligned} r &= \gamma_{sRGB}(r'), \\ g &= \gamma_{sRGB}(g'), \\ b &= \gamma_{sRGB}(b'). \end{aligned}$$

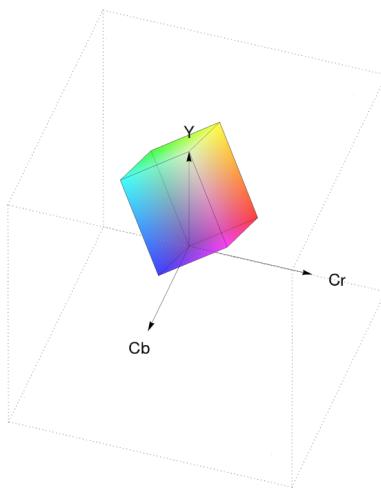
Le choix de l'exposant, 2.4 est tel qu'il est assez près de 2.2 pour maintenir une compatibilité approximative avec le matériel plus ancien [7, p. 7]. L'inverse de la fonction $\gamma_{sRGB}(\cdot)$, l'éq. (3.3.51), est donné par

$$\gamma_{sRGB}^{-1}(u) = \begin{cases} u^{\frac{323}{25}} & \text{si } u \leq 0.0031308 \\ \sqrt[2.4]{(1 + \frac{11}{200})u - \frac{11}{200}} & \text{sinon.} \end{cases} \quad (3.3.52)$$

Les transformations données par les éqs. (3.3.51) et (3.3.52) sont bricolées pour éviter que la correction ne donne des valeurs trop près de zéro lorsque la composante est petite, tout en appliquant une correction de type gamma lorsque les valeurs sont assez grandes. On remarquera aussi le grand nombre de « constantes magiques » qui ne sont ni justifiées, ni expliquées dans le standard.

3.3.9.4 YC_bC_r

L'espace de couleurs YC_bC_r , défini dans le standard BT.709 [21] est un espace dérivé de l'espace YUV mais cette-fois ci normalisé pour que nous ayons $0 \leq Y \leq 1$, $-\frac{1}{2} \leq C_b \leq \frac{1}{2}$ et $-\frac{1}{2} \leq C_r \leq \frac{1}{2}$. Comme nous avons déjà fait ce genre de normalisation avec S_3 (un espace de couleurs de la série S , ci-dessus), on déduit rapidement que la formule pour transformer de RGB_{709} à YC_bC_r est donnée

Figure 3.3.16 — L'espace de couleurs YC_bC_r

par :

$$\begin{aligned}
 T_{YC_bC_r} \begin{bmatrix} r_{709} \\ g_{709} \\ b_{709} \end{bmatrix} &= \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{2(1-\omega_b)} & 0 \\ 0 & 0 & \frac{1}{2(1-\omega_r)} \end{bmatrix}}_{\text{normalisation}} \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}}_{P_{YUV}} \underbrace{\begin{bmatrix} \omega_r & \omega_g & \omega_b \\ 1-\omega_r & -\omega_g & -\omega_b \\ -\omega_r & -\omega_g & 1-\omega_b \end{bmatrix}}_{T_p} \begin{bmatrix} r_{709} \\ g_{709} \\ b_{709} \end{bmatrix} \\
 &= \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.169 & -0.331 & 0.5 \\ -0.5 & -0.419 & -0.081 \end{bmatrix} \begin{bmatrix} r_{709} \\ g_{709} \\ b_{709} \end{bmatrix} \\
 &= \begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix}, \tag{3.3.53}
 \end{aligned}$$

tandis que l'inverse est donnée par

$$\begin{bmatrix} 1 & 0 & 1.402 \\ 1 & -0.334 & -0.714 \\ 1 & 1.772 & 0 \end{bmatrix} \begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix} = \begin{bmatrix} r_{709} \\ g_{709} \\ b_{709} \end{bmatrix}. \tag{3.3.54}$$

L'espace de couleurs est montré à la fig. 3.3.16.

*
* * *

Il existe une asymétrie entre la production de contenu (image ou vidéo) et sa consommation. Si on fait la supposition que contenu est créé une fois, mais qu'il est consommée plusieurs fois, on peut

vouloir un encodage aussi dispendieux que l'on veut, mais on tiendra quand même à un décodage aussi rapide et aussi léger que possible. Cela permet de réaliser des décodeurs numériques à faible coût. Une des façons d'accélérer le décodage des images, c'est d'avoir une transformée de l'espace de couleurs intermédiaire à RGB rapide. Avec YC_bC_r , on remarque que

$$\begin{bmatrix} 1 & 0 & 1.402 \\ 1 & -0.334 & -0.714 \\ 1 & 1.772 & 0 \end{bmatrix} \approx \begin{bmatrix} 1 & 0 & 1.5 \\ 1 & -0.25 & -0.75 \\ 1 & 1.75 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & \frac{3}{2} \\ 1 & -\frac{1}{4} & -\frac{3}{4} \\ 1 & \frac{7}{4} & 0 \end{bmatrix},$$

et il s'avère que des divisions par deux ou quatre se font bien en arithmétique entière grâce aux opérations de décalage. En effet, $\frac{3}{4}x$ c'est $\frac{x}{2} + \frac{x}{4}$, que l'on écrit $(x \gg 1) + (x \gg 4)$. Ces instructions de décalage sont presque tout le temps *beaucoup* plus rapides que des divisions entières, et considérablement plus rapides que les divisions en virgule flottante, ce qui rend l'astuce intéressante. Il ne reste plus qu'à calculer l'inverse de l'inverse pour trouver la transformée. Si

$$\tilde{T}_{YC_bC_r}^{-1} \begin{bmatrix} \tilde{Y} \\ \tilde{C}_b \\ \tilde{C}_r \end{bmatrix} = \begin{bmatrix} 1 & 0 & \frac{3}{2} \\ 1 & -\frac{1}{4} & -\frac{3}{4} \\ 1 & \frac{7}{4} & 0 \end{bmatrix} \begin{bmatrix} \tilde{Y} \\ \tilde{C}_b \\ \tilde{C}_r \end{bmatrix} = \begin{bmatrix} r \\ g \\ b \end{bmatrix}, \quad (3.3.55)$$

alors

$$(\tilde{T}_{YC_bC_r}^{-1})^{-1} = \tilde{T}_{YC_bC_r} = \begin{bmatrix} 0.304 & 0.609 & 0.087 \\ -0.174 & -0.348 & 0.522 \\ 0.464 & -0.406 & -0.058 \end{bmatrix},$$

et donc

$$\tilde{T}_{YC_bC_r} \begin{bmatrix} r \\ g \\ b \end{bmatrix} = \begin{bmatrix} 0.304 & 0.609 & 0.087 \\ -0.174 & -0.348 & 0.522 \\ 0.464 & -0.406 & -0.058 \end{bmatrix} \begin{bmatrix} r \\ g \\ b \end{bmatrix} = \begin{bmatrix} \tilde{Y} \\ \tilde{C}_b \\ \tilde{C}_r \end{bmatrix}. \quad (3.3.56)$$

J'ai proposé cet espace de couleurs pour le standard DjVu (prononcé *déjà vu*) en 1997 ou à peu près. C'est encore l'espace de couleurs utilisé [215, p. 265] et [15, p. 44]. La différence avec YC_bC_r demeure petite.

3.3.9.5 YP_bP_r

C'est un encodage analogique (pour des câbles « composantes ») de l'espace YC_bC_r , avec une correction de type gamma (*power function*) sur les composantes. Pour r , g et b entre 0 et 1 inclusivement, on applique d'abord la transformation

$$\begin{aligned} r' &= r^{0.45}, \\ g' &= g^{0.45}, \\ b' &= b^{0.45}. \end{aligned}$$

avant de poursuivre à travers YC_bC_r pour la télévision « définition standard » avec RGB_{601} ou $Y'C'_bC'_r$ pour la télévision haute définition avec RGB_{709} (nous reparlerons de $Y'C'_bC'_r$ dans un moment) [219, p. 303–318].

Ce n'est donc pas exactement un espace de couleurs linéaire. L'inverse de la transformation d'espace de couleurs est donnée par l'inverse de YC_bC_r (ou de $Y'C'_bC'_r$). L'inverse de la correction gamma est étrange. Si nous avons $r' = r^{0.45}$, l'inverse devrait être $r = r'^{\frac{20}{9}} = r'^{2.\bar{2}}$, puisque $0.45 = \frac{9}{20}$. Cependant, l'inverse est donné par $r = r'^{\frac{5}{2}}$ [218].

3.3.9.6 YD_bD_r

Le système de télévision analogique SÉCAM (pour Séquentiel couleur à mémoire¹) est, avec PAL (qui est encodé par YUV), l'un des standards compétiteurs de NTSC. SÉCAM était utilisé en France, en Afrique francophone et en Russie, et dans quelques autres pays [107]. SÉCAM utilise les couleurs primaires de BT.601 pour les images à 625 lignes (voir ci-haut). SÉCAM module les signaux de façon plus simple que NTSC :

$$\begin{aligned} T_{YD_bD_r} \begin{bmatrix} r_{\text{SÉCAM}} \\ g_{\text{SÉCAM}} \\ b_{\text{SÉCAM}} \end{bmatrix} &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{4}{3} \frac{1}{1-\omega_b} & 0 \\ 0 & 0 & \frac{4}{3} \frac{1}{1-\omega_r} \end{bmatrix} P_{YUV} T_p \begin{bmatrix} r_{\text{SÉCAM}} \\ g_{\text{SÉCAM}} \\ b_{\text{SÉCAM}} \end{bmatrix} \\ &= \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.450 & -0.833 & 1.333 \\ -1.333 & -1.116 & -0.217 \end{bmatrix} \begin{bmatrix} r_{\text{SÉCAM}} \\ g_{\text{SÉCAM}} \\ b_{\text{SÉCAM}} \end{bmatrix} \\ &= \begin{bmatrix} Y \\ D_b \\ D_r \end{bmatrix}. \end{aligned} \quad (3.3.57)$$

L'inverse est donné par

$$T_{YC_bC_r}^{-1} \begin{bmatrix} Y \\ D_b \\ D_r \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0.526 \\ 1 & -0.129 & -0.268 \\ 1 & 0.665 & 0 \end{bmatrix} \begin{bmatrix} Y \\ D_b \\ D_r \end{bmatrix} = \begin{bmatrix} r_{\text{SÉCAM}} \\ g_{\text{SÉCAM}} \\ b_{\text{SÉCAM}} \end{bmatrix} \quad (3.3.58)$$

3.3.9.7 $Y'C'_bC'_r$ alias YC_bC_r HDTV

La recommandation BT.709 introduit, en plus de nouvelles couleurs primaires, un nouvel espace de couleur, $Y'C'_bC'_r$, défini plus précisément avec des pondérations différentes pour les couleurs

1. Et non, comme le prétendent certaines langues bifides, le *système élégant contre les Américains!*

primaires. Les nouvelles pondérations sont

$$\begin{aligned} w_r &= 0.2126, \\ w_g &= 0.7152, \\ w_b &= 0.0722. \end{aligned}$$

Les coefficients sont donnés avec plus de précision pour accommoder des valeurs sur 8 et 10 bits. Comme pour YP_bP_r , les valeurs RGB sont ajustées par la même correction que Kodak YCC, l'éq. (3.3.8), avant d'être transformées vers l'espace de couleurs $Y'C'_bC'_r$. Ce dernier est bâti de la même façon que l'espace YC_bC_r , à la différence qu'on remplace ω_r , ω_g et ω_b par w_r , w_g et w_b :

$$\begin{aligned} T_{Y'C'_bC'_r}^{-1} \begin{bmatrix} r_{709} \\ g_{709} \\ b_{709} \end{bmatrix} &= \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{2(1-w_b)} & 0 \\ 0 & 0 & \frac{1}{2(1-w_r)} \end{bmatrix}}_{\text{normalisation}} \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}}_{P_{YUV}} \underbrace{\begin{bmatrix} w_r & w_g & w_b \\ 1-w_r & -w_g & -w_b \\ -w_r & -w_g & 1-w_b \end{bmatrix}}_{T_{p709}} \begin{bmatrix} r_{709} \\ g_{709} \\ b_{709} \end{bmatrix} \\ &= \begin{bmatrix} 0.2126 & 0.7152 & 0.0722 \\ -0.1146 & -0.3854 & 0.5 \\ 0.5 & -0.4542 & -0.0458 \end{bmatrix} \begin{bmatrix} r_{709} \\ g_{709} \\ b_{709} \end{bmatrix} \\ &= \begin{bmatrix} Y' \\ C'_b \\ C'_r \end{bmatrix}, \end{aligned} \tag{3.3.59}$$

L'inverse est donné par

$$T_{Y'C'_bC'_r}^{-1} \begin{bmatrix} Y' \\ C'_b \\ C'_r \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1.5748 \\ 1 & -0.1874 & -0.4681 \\ 1 & 1.8556 & 0 \end{bmatrix} \begin{bmatrix} Y' \\ C'_b \\ C'_r \end{bmatrix} = \begin{bmatrix} r_{709} \\ g_{709} \\ b_{709} \end{bmatrix}. \tag{3.3.60}$$

3.3.9.8 BT.2020 UHDTV

Pour la télévision 4K (3840×2160 pixels) et 8K (7640×4320 pixels), la recommandation BT.2020 définit *encore* de nouvelles couleurs primaires, de nouveaux poids, et un nouvel espace de couleurs [20]. Les couleurs primaires sont

Rouge R_{2020}	$x = 0.708$	$y = 0.292$
Vert G_{2020}	$x = 0.170$	$y = 0.797$
Bleu B_{2020}	$x = 0.131$	$y = 0.046$
Blanc D_{65}	$x = 0.3127$	$y = 0.3290$

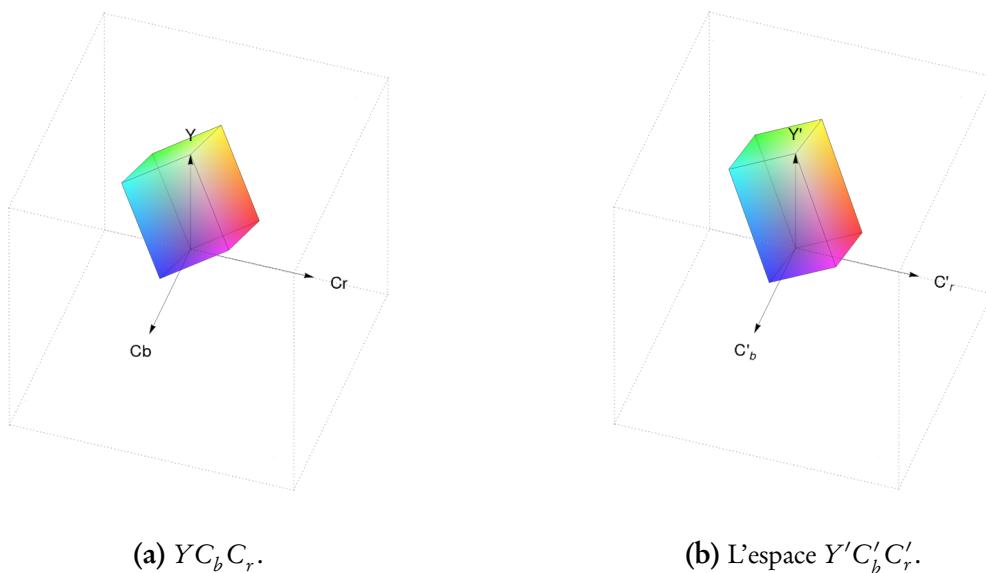


Figure 3.3.17 — Les espaces de couleurs YC_bC_r et $Y'C'_bC'_r$ comparés.

tandis que les pondérations sont données par

$$W_r = 0.2627,$$

$$W_g = 0.6780,$$

$$W_b = 0.0593.$$

La transformée de couleurs est aussi passablement plus compliquée que les précédentes. Comme pour BT.709 et KodakYCC, on commence par une correction de type gamma :

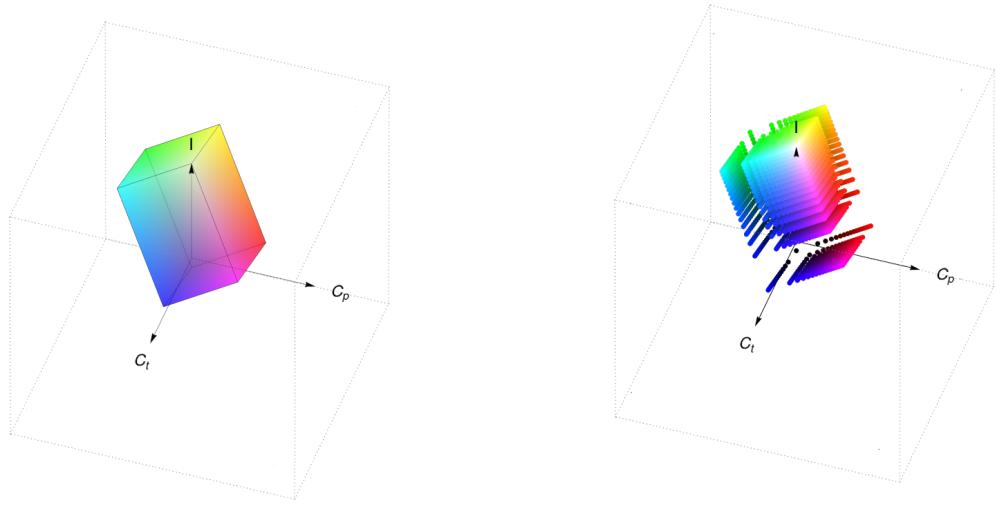
$$c' = \begin{cases} \alpha c^{0.45} - (\alpha - 1) & \text{si } \beta \leq c \geq 1 \\ 4.5c & \text{sinon.} \end{cases} \quad (3.3.61)$$

où $\alpha = 1.099$ et $\beta = 0.018$ pour les images en 10 bits par composantes, tandis que nous aurons $\alpha = 1.0993$ et $\beta = 0.0181$ pour les images en 12 bits. Après la correction (qui a pour effet de donner peu de codes aux couleurs très sombres et plus aux couleurs intenses, comme nous le montre la fig. 3.3.18 (b)), nous obtenons r' , g' et b' . La transformée de couleur est

$$Y'_c = (W_r, W_g, W_b)^T(r', g', b') = 0.2627r' + 0.6780g' + 0.0593b', \quad (3.3.62)$$

puis

$$C'_{BC} = \begin{cases} \frac{b' - Y'_C}{-2N_B} & \text{si } N_B \leq b' - Y'_C \leq 0 \\ \frac{b' - Y'_C}{2P_B} & \text{sinon,} \end{cases} \quad (3.3.63)$$

(a) L'espace $Y'C'_b C'_r$.(b) La densité de l'espace $Y'C'_b C'_r$ (effet exagéré).**Figure 3.3.18** — L'espace de couleurs $Y'C'_{BC} C'_{RC}$

et

$$C'_{RC} = \begin{cases} \frac{r' - Y'_C}{-2N_R} & \text{si } N_R \leq b' - Y'_C \leq 0 \\ \frac{r' - Y'_C}{2P_R} & \text{sinon,} \end{cases} \quad (3.3.64)$$

où

$$\begin{aligned} P_B &= \alpha(1 - 0.0593^{0.45}) \approx 0.7910 \\ N_B &= \alpha(1 - 0.9407^{0.45}) \approx -0.9720 \\ P_R &= \alpha(1 - 0.2627^{0.45}) \approx 0.4969 \\ P_N &= \alpha(1 - 0.7373^{0.45}) \approx -0.8591. \end{aligned}$$

Les indices en « C » (comme C'_{RC}) sont censés indiquer « luminosité constante », dans la mesure où ils préservent, avec une bonne précision, les valeurs r' , g' , b' . L'espace $Y'C'_{BC} C'_{RC}$ est montré à la fig. 3.3.18. La recommandation BT.2020 dit par ailleurs que pour une approximation, on pourra utiliser $Y'C'_b C'_r$, mais avec les poids W_r , W_g et W_b ; la différence entre les deux espaces est négligeable — si on néglige le fait qu'ils n'utilisent pas les mêmes couleurs primaires.

3.3.9.9 BT.2100 HDRTV

Le standard BT.2100 spécifie les caractéristiques pour un téléviseur à grande gamme dynamique (en anglais HDR pour *High Dynamic Range*), censé s'approcher de l'étendue de la vision humaine.



(a) Image exposée pour les lumières intenses.



(b) Image exposée pour les lumières moyennes.



(c) Image exposée pour les lumières faibles.



(d) Image HDR recomposée.

Figure 3.3.19 — Une image en grande gamme dynamique (HDR).

Rappelez-vous la fig. 1.2.9, p. 10, où l'on montre la réponse perçue à l'intensité de lumière reçue. Les images HDR présentent, au moins en principe, une très grande différence de luminosité entre la luminosité maximale (au seuil de l'aveuglement) et minimale (au seuil de la perception).

Dans la réalité, la bande dynamique demeure modeste, mais des traitements sont apportés aux couleurs pour donner l'impression d'une grande bande dynamique. La fig. 3.3.19 explique comment : on prend plusieurs images de la même scène, une en exposant pour les lumières intenses (fig. 3.3.19 (a)), une pour les lumières moyennes (fig. 3.3.19 (b)), et une pour les lumières faibles (fig. 3.3.19 (c)). On les combine ensuite en choisissant les pixels en fonction de l'image que notre œil percevrait — il y a un flou artistique pour réussir à produire des images vraisemblables et agréables. C'est ce processus que les standards HDR tentent d'automatiser, en approximant la courbe de réponse de la fig. 1.2.9, p. 10, en « boostant » les intensités les plus faibles et en atténuant les plus fortes.

BT.2100 définit le standard pour la télévision HDR [22]. Les transformations de couleurs sont non linéaires, mais BT.2100 est listé dans cette section par ce que le cœur de la transformée est linéaire. Il se distingue aussi des autres standards en ce qu'il utilise une composition de transformées, censée rendre la réponse du système visuel [57].

La première matrice de transformation (après des étapes de correction de type gamma que nous ne décrirons pas ici mais qui se trouvent fort bien détaillées dans la recommandation [22, p. 4+]) convertit les r , g et b (avec les couleurs primaires de BT.2020) dans l'espace LMS grâce à la transformation

$$T_{LMS} \begin{bmatrix} r_{2020} \\ g_{2020} \\ b_{2020} \end{bmatrix} = \frac{1}{4096} \begin{bmatrix} 1688 & 2146 & 262 \\ 683 & 2951 & 462 \\ 99 & 309 & 3688 \end{bmatrix} \begin{bmatrix} r_{2020} \\ g_{2020} \\ b_{2020} \end{bmatrix} = \begin{bmatrix} l \\ m \\ s \end{bmatrix}. \quad (3.3.65)$$

L'espace LMS (pour *Long, Medium, Short Wavelengths*) est censé reproduire la réponse moyenne des cônes rouges, verts et bleus. Le diviseur 4096 normalise les valeurs sur 12 bits (car le standard prévoit des images sur 10 ou 12 bits).

Par la suite, une seconde transformée est appliquée à l'espace LMS pour l'amener dans l'espace $IC_T C_p$, dont les composantes représentent la luminance, la différence jaune-bleu et la différence rouge-vert/cyan. La différence jaune-bleu est dite tritanopique, tandis que la différence rouge-vert est protanopique¹, d'où le T et le P dans $IC_T C_p$.

La transformation de $L'M'S'$ (car encore une fois une correction de type gamma est appliquée) à $IC_T C_p$ est donnée par

$$\begin{aligned} T_{IC_T C_p} \begin{bmatrix} l' \\ m' \\ s' \end{bmatrix} &= \frac{1}{4096} \begin{bmatrix} 2048 & 0 & 2048 \\ 6610 & -13613 & 7003 \\ 17933 & -17933 & -542 \end{bmatrix} \begin{bmatrix} l' \\ m' \\ s' \end{bmatrix} \\ &= \frac{1}{4096} \begin{bmatrix} 2048 & 0 & 2048 \\ 6610 & -13613 & 7003 \\ 17933 & -17390 & -542 \end{bmatrix} \sigma \left(\frac{1}{4096} \begin{bmatrix} 1688 & 2146 & 262 \\ 683 & 2951 & 462 \\ 99 & 309 & 3688 \end{bmatrix} \begin{bmatrix} r_{2020} \\ g_{2020} \\ b_{2020} \end{bmatrix} \right) \\ &= \begin{bmatrix} I \\ C_T \\ C_P \end{bmatrix}. \end{aligned} \quad (3.3.66)$$

où $\sigma(\cdot)$ représente la fonction de transfert non linéaire. La transformée inverse est donnée par

$$T_{IC_T C_p}^{-1} \begin{bmatrix} I \\ C_T \\ C_P \end{bmatrix} = \begin{bmatrix} 0.9999 & -0.2757 & 0.2158 \\ 1 & -0.2929 & -0.0062 \\ 1.0001 & 2757 & -0.2158 \end{bmatrix} \begin{bmatrix} I \\ C_T \\ C_P \end{bmatrix} = \begin{bmatrix} l' \\ m' \\ s' \end{bmatrix}, \quad (3.3.67)$$

puis enfin, après l'inverse de la transformation non linéaire qui nous amène de $L'M'S'$ à LMS , on

1. Pour être strict, il faudrait parler de tritanopie et de protanopie que lorsque nous sommes *incapables* de voir ces différences, comme par exemple, si nous souffrons d'une des formes de daltonisme. Il existe plusieurs autres conditions, dépendamment de quels récepteurs font défaut. Voir, par exemple [29, 182].

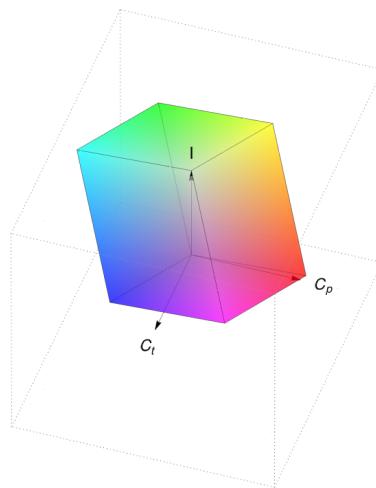


Figure 3.3.20 — L'espace de couleurs $IC_T C_P$ transformant LMS transformant RGB , sans les corrections non linéaires.

peut revenir à RGB

$$T_{LMS}^{-1} \begin{bmatrix} l \\ m \\ s \end{bmatrix} = \begin{bmatrix} 3.4361 & -2.5058 & 0.0697 \\ -0.7909 & 1.9831 & -0.1922 \\ -0.0261 & -0.0987 & 1.1248 \end{bmatrix} \begin{bmatrix} l \\ m \\ s \end{bmatrix} = \begin{bmatrix} r_{2020} \\ g_{2020} \\ b_{2020} \end{bmatrix}. \quad (3.3.68)$$

L'espace de couleurs $IC_T C_P$, sans les corrections non linéaires, est montré à la fig. 3.3.20.

*
* * *

Vous aurez compris que les transformations non linéaires de RGB à LMS et de LMS à $IC_T C_P$ sont censé modéliser la courbe de réponse de la fig. 1.2.9, et ainsi rendre l'illusion d'une image à grande gamme dynamique. Cependant, l'image reste ultimement limitée par la gamme relativement modeste de l'écran.

3.3.9.10 Un portrait de famille

La figure 3.3.21 montre les filiations entre les divers espaces de couleur de la télévision. Dans l'arbre généalogique, les liens directs se limitent à des transformations simples dans la structure des espaces de couleurs (changement des couleurs primaires ou de leur pondération, échanger des lignes de la matrice des *transmission primaries*, changement de normalisation, etc.). Quelques fois, l'ajout d'une correction de type gamma vient bouleverser la linéarité stricte des espaces de couleurs. Nous avons vu aussi que pour certains espaces (comme $YP_b P_r$), ces corrections non linéaires sont présentes pour accommoder un transport analogue, ou une autre contrainte. Parfois, (comme BT.2100), elles

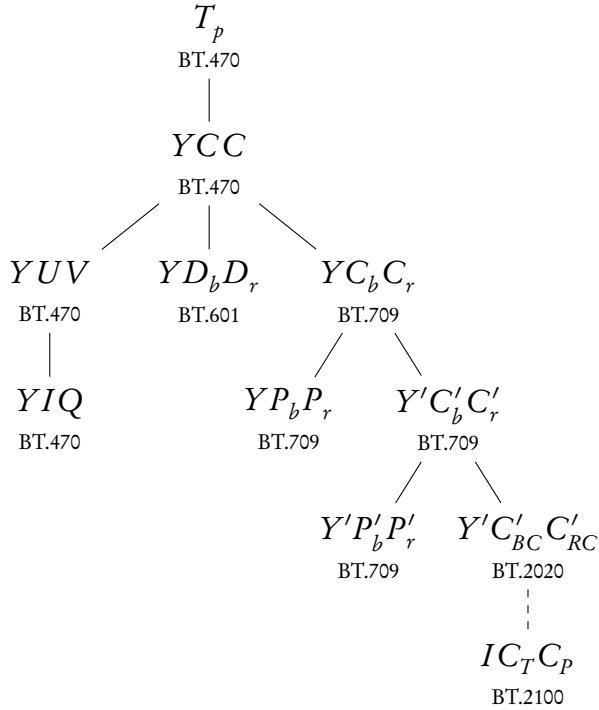


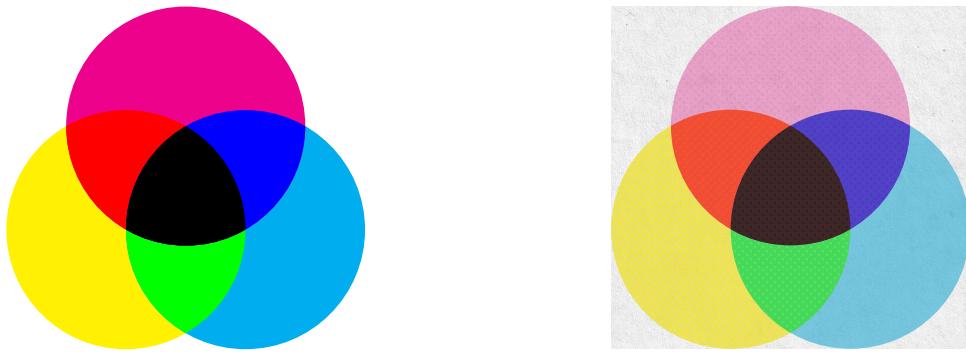
Figure 3.3.21 — La famille BT.

jouent un rôle fondamental dans l'espace de couleur. Pour en savoir plus sur ces espaces de couleurs et leurs interactions avec les différents standards de télévision, consultez [40, 92, 93, 219].

3.3.10 CMY et CMYK

Nous avons déjà rencontré l'espace de couleurs CMYK au chapitre 1. Il s'agit d'un espace linéaire, mais *soustractif*, c'est-à-dire que les « pixels » (des points imprimés) filtrent la lumière reçue plutôt qu'en émettent — ce qui fait des autres espaces de couleurs des espaces *additifs*, ils émettent de la lumière. Les quatre couleurs de base sont le cyan, C , le magenta, M , le jaune, Y (qu'il ne faut pas confondre ici avec le Y de luminance dans d'autres espaces de couleurs), et le noir, K . On pourrait penser que K a été tiré du k final de *blacK* pour le désambiguïser du bleu B , mais non ! K vient de *key*, la « clef d'alignement » en impression — qui n'est pas forcément noire ! Les couleurs sont souvent alignées contre une référence qui délimite les contours (voir, par exemple, les fig. 1.2.3 (c) et 1.2.3 (d) pour comprendre pourquoi et comment).

Si nous disposions d'encre *parfaites* sur papier transcendant, nous obtiendrions, à l'imprimé, des couleurs vives et précises, comme à la fig. 3.3.22 (a). Or, les encres et les papiers utilisés sont toujours en deçà des espérances, et les couleurs obtenues sont moins brillantes et moins précises, on obtient un résultat semblable à la fig. 3.3.22 (b), où, en plus d'utiliser des encres moins que parfaites, le tramage a

(a) L'espace $CMYK$: résultats désirés.(b) L'espace $CMYK$: résultats obtenus!Figure 3.3.22 — L'espace de couleurs $CMYK$, attentes et résultats!

été utilisé. Dans un monde idéal, le mélange des encres donnerait la couleur secondaire avec précision, or, le magenta et le cyan mélangés ne donnent pas un bleu pur, mais un bleu vaguement prussien¹, et le mélange des trois couleurs ne donne pas noir, mais un brun vaguement scatochrome. C'est une des raisons pourquoi le noir est ajouté, et assez souvent aussi des *spot colors*, des encres avec une couleur spécifique (voir fig. 1.2.3 (a)). Nous verrons comment l'ajout du noir a aussi pour effet de permettre une économie des encres colorées.

*
* * *

Si on suppose que CMY sont les couleurs parfaitement *complémentaires* à RGB (car $C = G + B$, et donc $W - R = G + B = C$) on peut obtenir une première conversion (linéaire) et naïve :

$$c = 1 - r,$$

$$m = 1 - g,$$

$$y = 1 - b,$$

que l'on peut exprimer comme un produit matrice/vecteur :

$$\begin{bmatrix} -1 & 0 & 0 & 1 \\ 0 & -1 & 0 & 1 \\ 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r \\ g \\ b \\ 1 \end{bmatrix} = \begin{bmatrix} 1 - r \\ 1 - g \\ 1 - b \\ 1 \end{bmatrix} = \begin{bmatrix} c \\ m \\ y \\ 1 \end{bmatrix}. \quad (3.3.69)$$

1. Le *bleu de Prusse*, *Preußischblau* en Allemand, est un bleu chimique à base de fer proche de l'indigo [60, p. 92–98].

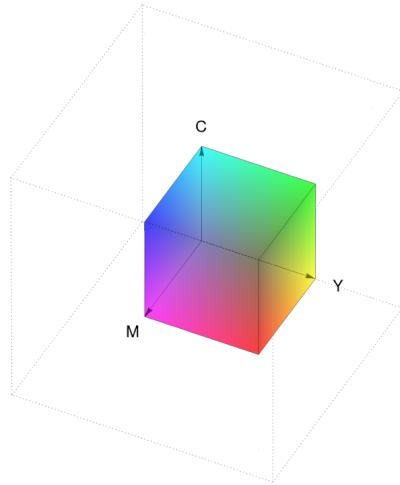


Figure 3.3.23 — L'espace de couleurs CMY .

L'inverse est donné par

$$r = 1 - c,$$

$$g = 1 - m,$$

$$b = 1 - y,$$

c'est-à-dire

$$\begin{bmatrix} -1 & 0 & 0 & 1 \\ 0 & -1 & 0 & 1 \\ 0 & 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} c \\ m \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} 1 - c \\ 1 - m \\ 1 - y \end{bmatrix} = \begin{bmatrix} r \\ g \\ b \end{bmatrix} \quad (3.3.70)$$

où on ignore joyeusement k . On remarque que l'inverse n'est plus une matrice carrée : comme c'est une projection d'un espace à 4 dimensions sur un espace à 3, la matrice est rectangulaire 3×4 ¹. L'éq. (3.3.69) ne se réduit pas à une matrice 4×3 à cause de la « translation » : il faut alors utiliser des coordonnées homogènes. L'espace de couleurs CMY est montré à la fig. 3.3.23.

Mais la transformée naïve de RGB à $CMYK$ ne fonctionne qu'à l'écran ! En imprimé, on ne peut pas avoir un cyan foncé en ajoutant plus d'encre cyan ! Un peu d'encre cyan filtre un peu de jaune, beaucoup d'encre filtrera beaucoup de jaune, mais ne devient pas cyan foncé. Il faut donc inclure le noir dans notre modèle pour permettre de faire des couleurs foncées — et les couleurs pâles

1. On pourrait voir toute matrice de transformation comme une projection d'un espace à n dimensions vers un autre espace à m dimensions. Lorsque que nous projetons un espace à $n > m$ dimensions en m dimensions, la matrice sera rectangulaire horizontalement et $m \times n$, c'est-à-dire avec m rangées de n colonnes. Si nous projetons un espace à $n < m$ dimensions dans un espace en m dimensions, la matrice sera encore rectangulaire, cette fois verticalement, $m \times n$.

sont mélangées avec du blanc, mais le blanc, c'est le papier! Il nous faut donc maintenant inclure la composante K que nous supposerons noire¹.

Pour ajouter le noir à CMY pour obtenir $CMYK$, on pose

$$\begin{aligned} k &= \min(c, m, y) \\ &= \min(1 - r, 1 - g, 1 - b) \\ &= 1 - \max(r, g, b) \\ &= 1 - \gamma. \end{aligned}$$

Autrement dit, la composante K , c'est ce qu'on a ôté à blanc. Alors, si $k \neq 1$, nous avons

$$\begin{aligned} c' &= \frac{c - k}{1 - k}, \\ m' &= \frac{m - k}{1 - k}, \\ y' &= \frac{y - k}{1 - k}, \\ k &= \min(c, m, y). \end{aligned} \tag{3.3.71}$$

sinon nous posons $(c', m', y', k) = (0, 0, 0, 1)$.

On peut comprendre que les composantes (c', m', y', k) (avec des apostrophes pour les distinguer de CMY) sont encodées différentiellement par rapport au niveau de noir qu'elles comportent déjà. Les différences sont normalisées sur $1 - k$ (une étape qui ne paraît pas essentielle, mais qui est traditionnellement présente). Pour revenir à CMY , on applique les transformations inverses :

$$\begin{aligned} c &= (1 - k)c' + k, \\ m &= (1 - k)m' + k, \\ y &= (1 - k)y' + k, \end{aligned} \tag{3.3.72}$$

que l'obtient facilement à partir des éqs. (3.3.71). On peut ensuite revenir à RGB .

*
* * *

Passer par l'espace intermédiaire — et par ailleurs plutôt inutile — CMY pour se rendre de RGB à $CMYK$ paraît bien superflu. Créons donc la transformée directe de RGB à $CMYK$ (et son inverse, de $CMYK$ à RGB). Rappelons : nous avons

$$k = 1 - \max(r, g, b) = 1 - \gamma.$$

1. Comme nous l'avons mentionné précédemment, le *key* est la couleur d'alignement, et elle n'est pas forcément noire.

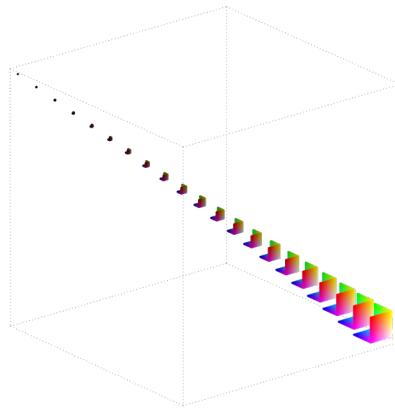


Figure 3.3.24 — L'espace de couleurs *CMYK* (les K sont montrés par translations).

Nous avons suite que

$$\begin{aligned} c' &= \frac{c-k}{1-k} \\ &= \frac{(1-r)-k}{1-k} \\ &= \frac{(1-r)-(1-\gamma)}{1-(1-\gamma)} \\ &= \frac{\gamma-r}{\gamma} = 1 - \frac{r}{\gamma}. \end{aligned}$$

En faisant la même transformation pour les autres composantes, si $\gamma \neq 0$, nous trouvons

$$\begin{aligned} c' &= \frac{\gamma-r}{\gamma} = \frac{(1-k)-r}{1-k}, \\ m' &= \frac{\gamma-g}{\gamma} = \frac{(1-k)-g}{1-k}, \\ y' &= \frac{\gamma-b}{\gamma} = \frac{(1-k)-b}{1-k}, \\ k &= 1-\gamma, \end{aligned} \tag{3.3.73}$$

sinon nous posons $(c', m', y', k) = (0, 0, 0, 1)$. Pour trouver l'inverse, on utilise les éqs. (3.3.73) pour isoler les composantes *RGB*, et on obtient :

$$\begin{aligned} r &= (1-k)(1-c'), \\ g &= (1-k)(1-m'), \\ b &= (1-k)(1-y'). \end{aligned} \tag{3.3.74}$$

*
* * *

Il est impossible de visualiser directement l'espace $CMYK$ en « cube des couleurs » puisque celui-ci est en quatre dimensions. Si on dessine, comme à la fig. 3.3.24, pour chaque K les trois autres dimensions, on découvre une structure qui rappelle les poupées russes, où chaque surface CMY est une petite *matriochka*. On voit aussi que correctement empilées, les « coquilles » correspondront au cube RGB .

3.4 Les espaces de couleurs non linéaires

Les espaces non linéaires se divisent en deux grandes familles. D'abord, nous avons la famille des espaces non linéaires seulement par leur système de coordonnées, souvent une variation sur le thème des coordonnées polaires ou sphériques. Puis nous avons la famille des espaces où les coordonnées sont non linéaires pour répondre à un critère perceptuel, par exemple, s'assurer qu'un déplacement dans l'espace des coordonnées correspond à une variation proportionnelle dans la perception. Cette dernière famille est plus difficile à concevoir : il s'agit de bien modéliser la réponse de l'observateur. Nous présenterons des espaces de chaque famille.

*
* * *

Établissons le vocabulaire que nous utiliserons pour ce qui suit :

- La *teinte*, c'est-à-dire la couleur « pure ». La plupart des espaces de couleurs utilisent un système de coordonnées polaires ou sphériques où un angle donne la teinte, avec une teinte spécifique choisie pour correspondre à l'angle zéro (typiquement rouge) ; et d'autres pour des angles spécifiques (Munsell utilise cinq couleurs également distribuées autour du cercle chromatique, Ostwald en utilise quatre, *HSV/HSL* en utilisent six). Les couleurs primaires sont choisies en fonction des applications spécifiques.
- La *saturation* ou la *pureté*, c'est-à-dire la portion de la couleur pure mélangée au gris neutre de même valeur. La plupart des systèmes dénotent la pureté comme étant maximale lorsque la couleur est le plus éloigné de l'axe central, auxquels correspondent les tons de gris ; la pureté varie donc entre la couleur maximamente distinguée et un gris neutre de même valeur.

- La *valeur*. La luminance mesure la puissance de la lumière visible, elle n'a pas limite supérieure évidente ; cependant, les espaces de couleurs la supposent limitée entre 0, noir, et 1, blanc, que l'on suppose tous deux idéaux — dans la réalité, toutefois, ni l'un ni l'autre ne sont parfaits, mais seulement limités par l'équipement de rendu.

La plupart des systèmes (mais pas tous) profitent du fait qu'une couleur ne peut pas être à la fois très saturée et très brillante, car elle est perçue comme étant « blanche » et qu'une couleur ne peut être très saturée et très foncée en même temps, car elle est perçue comme « noircie », pour restreindre l'espace de couleurs dans ces régions. Ostwald propose un système en double cône à cet effet, tandis que l'espace de couleurs *CMYK* ne donne beaucoup de place qu'aux couleurs maximalement saturées et très peu aux couleurs foncées (voir la fig. 3.3.24).

3.4.1 HSV et HSL

Les espaces (et différents systèmes de classement) de couleurs sont depuis longtemps (voir § 3.5) basés sur un cercle chromatique où un certain nombre de couleurs primaires sont placées, les teintes intermédiaires entre elles, aussi sur le cercle. Certains étendent le modèle à un disque, où couleurs maximalement saturées (« pures ») sont placé sur le rebord et les couleurs peu saturées près du centre, laissant le centre exact pour les gris achromatiques. La fig. 3.4.1 montre un tel arrangement, celui que *HSV* et *HSL* utilisent.

L'idée est donc de découper le cercle (ou le disque) chromatique en secteurs, où chaque angle spécifique indique une teinte, et où la distance au centre donne la saturation, les plus loin du centre étant les plus saturés. Un autre axe, perpendiculaire aux deux autres donne la luminosité de la couleur. Comme nous l'avons fait remarqué précédemment, une couleur ne peut être à la fois très saturée et très lumineuse, car elle est perçue comme blanche, de même qu'une couleur foncée n'apparaît justement que foncée, sans grand-couleur. Les systèmes *HSV* et *HSL* diffèrent principalement dans la façon dont ils exploitent cette observation.

3.4.1.1 HSV

Revenons à la fig. 3.4.1. L'espace *HSV* est composé de trois dimensions :

- *H*, la teinte (*hue*), est un angle (dépendamment des implémentations, de 0° à 360° , de 0 à 2π , ou même 0 à 1) relatif à la couleur de départ, presque tout le temps le rouge.
- *S*, la saturation, ou la pureté, est la distance au centre. Typiquement, elle variera de 0 (achromatique) à 1 (maximamente saturée).

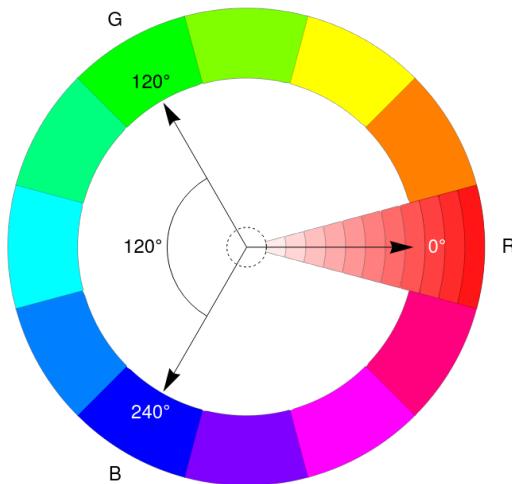


Figure 3.4.1 — Le cercle chromatique de l'espace *HSV*.

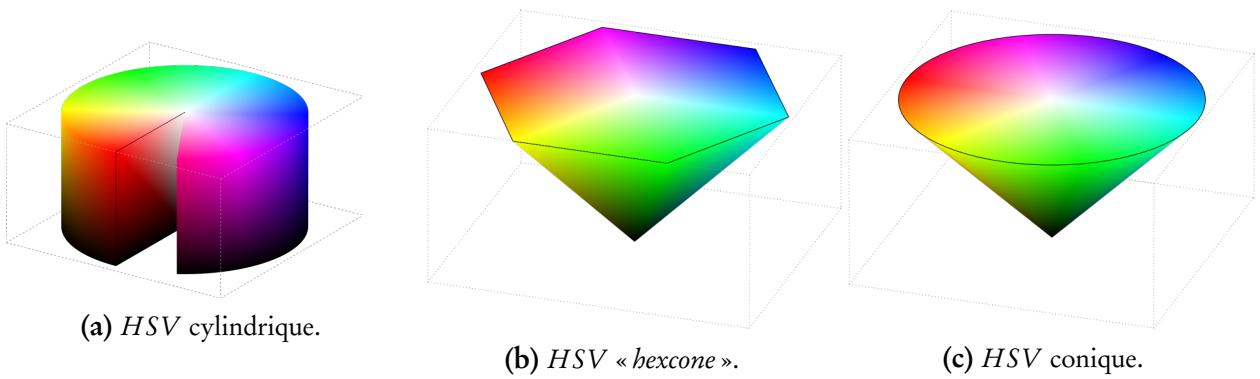
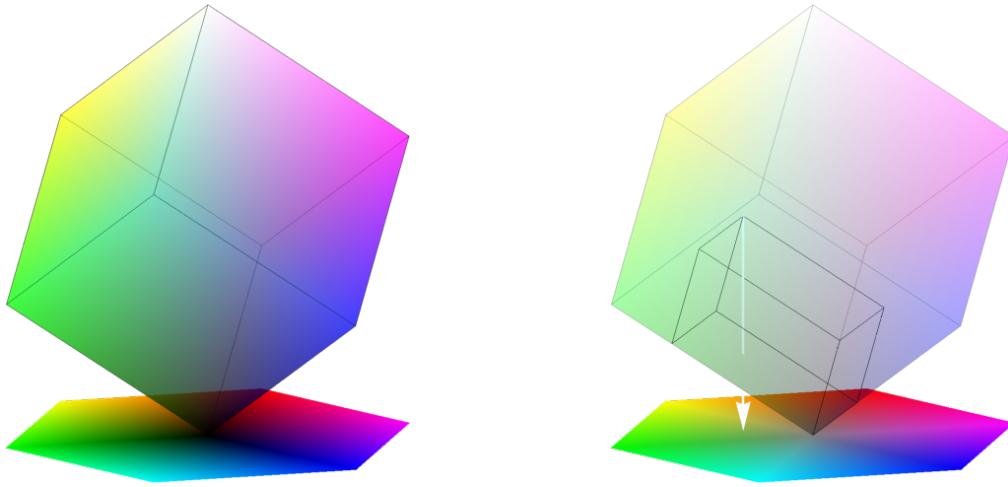


Figure 3.4.2 — Géométries de l'espace *HSV*.

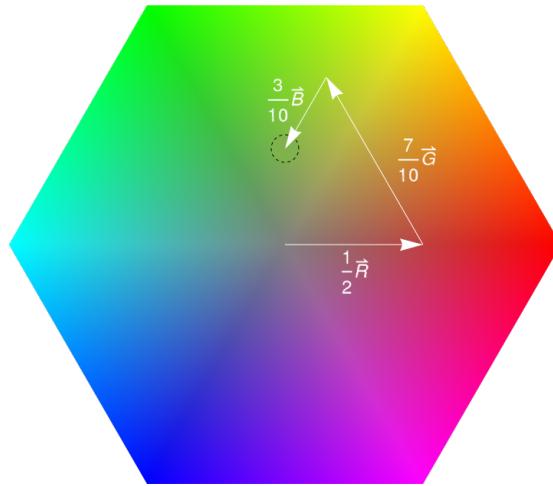
- V , la valeur, la luminosité perçue de la couleur, aussi normalisée entre 0 (noir) et 1 (blanc).

Comment implémenter cet espace de couleurs? Probablement qu'une approche naïve nous donnerait un espace comme celui montré à la fig. 3.4.2 (a), où les trois dimensions sont indépendantes. Ce cylindre, qui n'est pas sans rappeler le pudding à l'arsenic avec sa tranche retirée, donne beaucoup de codes aux couleurs très foncées difficiles à distinguer, même s'il place les tons de gris le long de son axe central. La fig. 3.4.2 (b) nous montre une projection du cube *RGB* contre un hexagone et compresse les couleurs difficiles à distinguer ensemble, pour former une pointe dont le sommet est le noir. La fig. 3.4.3 (a) nous montre qu'appliquer une rotation au cube *RGB* de façon à rendre sa grande diagonale verticale nous permet une projection qui forme un hexagone régulier. Mais comment faire cette projection?

Considérons trois vecteurs (colonne) unitaires $\vec{r} = (r_1, r_2)$, $\vec{g} = (g_1, g_2)$ et $\vec{b} = (b_1, b_2)$ dans le plan, chacun à 120° des autres. Si ces vecteurs sont les projections des vecteurs \vec{R} , \vec{G} et \vec{B} dans le plan, alors les couleurs qui sont des combinaisons linéaires de \vec{R} , \vec{G} et \vec{B} (dans le cube *RGB*) ont leur projection

(a) Une projection du cube *RGB*.

(b) La projection d'une couleur en particulier.

Figure 3.4.3 — *HSV* et une projection du cube *RGB*.Figure 3.4.4 — La projection sur l'hexagone *HSV*.

dans la combinaison linéaire de mêmes coefficients de \vec{r} , \vec{g} et \vec{b} ! Autrement dit,

$$\begin{bmatrix} \vec{r} & \vec{g} & \vec{b} \end{bmatrix} \begin{bmatrix} r \\ g \\ b \end{bmatrix} = \begin{bmatrix} r_1 & g_1 & b_1 \\ r_2 & g_2 & b_2 \end{bmatrix} \begin{bmatrix} r \\ g \\ b \end{bmatrix} = \begin{bmatrix} u \\ v \end{bmatrix}.$$

Cette projection n'est pas encore tout à fait complète : nous perdons l'information de luminance et plusieurs couleurs sont projetées au même endroit ! Les coordonnées (u, v) nous donnent, au moins, l'information sur la teinte, puisque nous pouvons obtenir l'angle que forme (u, v) avec l'origine. La saturation est proportionnelle à la norme $\|(u, v)\|$ (la longueur du vecteur) et la valeur, la luminosité, doit être une fonction de (r, g, b) . Mais ces opérations (\tan^{-1} , $\sqrt{u^2 + v^2}$, etc.) sont dispendieuses en temps de calcul, donc on cherchera à les éviter en construisant la transformée différemment.

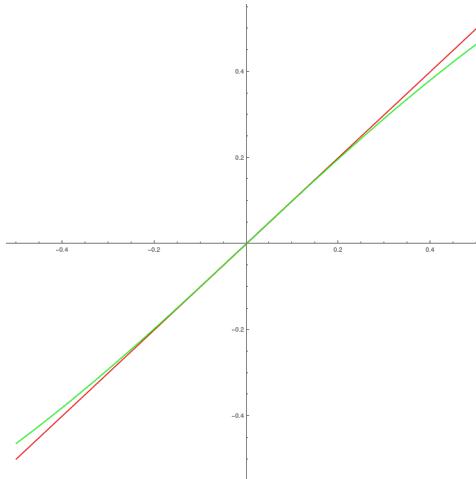


Figure 3.4.5 — Sur l'intervalle $[-\frac{1}{2}, \frac{1}{2}]$, $\tan^{-1} x \approx x..$

Clairement, calculer exactement l'angle et la saturation (la distance au centre) pour une couleur (r, g, b) est dispendieux computationnellement. Nous allons donc tricher un peu et changer le calcul tout en conservant l'esprit de la transformation. Commençons par remarquer que près de $x = 0$, on a

$$\tan^{-1} x \approx x,$$

et comme nous le montre la fig. 3.4.5, l'erreur sur l'intervalle $[-\frac{1}{2}, \frac{1}{2}]$ demeure modeste; au plus $\pm 0.0363524\dots$, une erreur relative d'au plus $\approx 7\%$. Cela nous permet de prendre la tangente x pour son angle — c'est bien entendu inexact, mais l'erreur est tolérable — et de nous dispenser de la vilaine fonction trigonométrique \tan ¹.

Ensuite, remarquons, avec l'aide de la fig. 3.4.6, que l'espace des couleurs est divisé en trois régions, où l'une des couleurs primaires domine. Dans la région 0, c'est le rouge (nous expliquerons les indices dans un instant), dans la région 2, c'est le vert, dans la 4, le bleu. Nous pourrons alors choisir comme base du triangle qui sert à calculer la pente comme la couleur dominante; et la hauteur comme la différence entre les deux autres couleurs. Plus exactement, nous utiliserons les différences au gris neutre de même luminosité (qui est au centre de la figure). Nous estimerons que la luminosité est donnée par

$$M = \max(r, g, b),$$

tandis que la portion de gris neutre est donnée par

$$m = \min(r, g, b),$$

1. Les instructions trigonométriques peuvent prendre des dizaines ou des centaines de cycles, même sur les processeurs les plus récents et les plus sophistiqués. Agner Fog rapporte de 11 à 82 cycles pour l'architecture Ryzen, alors qu'une addition en virgule flottante, en comparaison, ne prend qu'un cycle [95, p. 93].

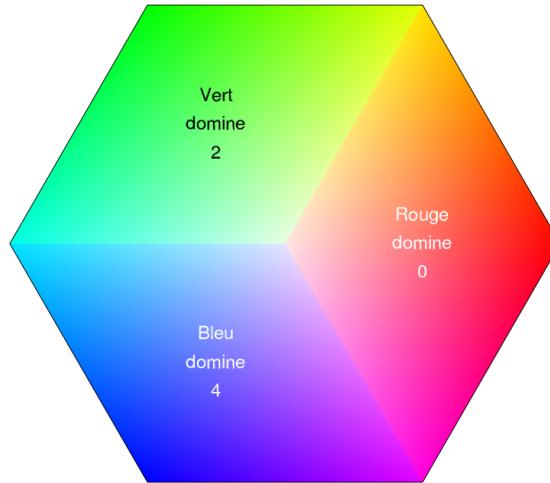


Figure 3.4.6 — Les régions de couleurs dans l’hexagone *HSV*.

et on pourra dire que la saturation est $\approx M - m$.

Ainsi, nous pouvons écrire une couleur (r, g, b) comme $m + (r - m, g - m, b - m)$; comme nous l’avions fait avec *CYMK*. La portion $(r - m, g - m, b - m)$ peut être interprétée comme la différence au gris (m, m, m) . Prenons par exemple une couleur où le rouge domine, nous aurons $M = r$, et la pente (qu’on estime être \approx l’angle) est donné par

$$\frac{(g - m) - (b - m)}{r - m} = \frac{g - m - b + m}{r - m} = \frac{g - b}{r - m} = \frac{g - b}{M - n}.$$

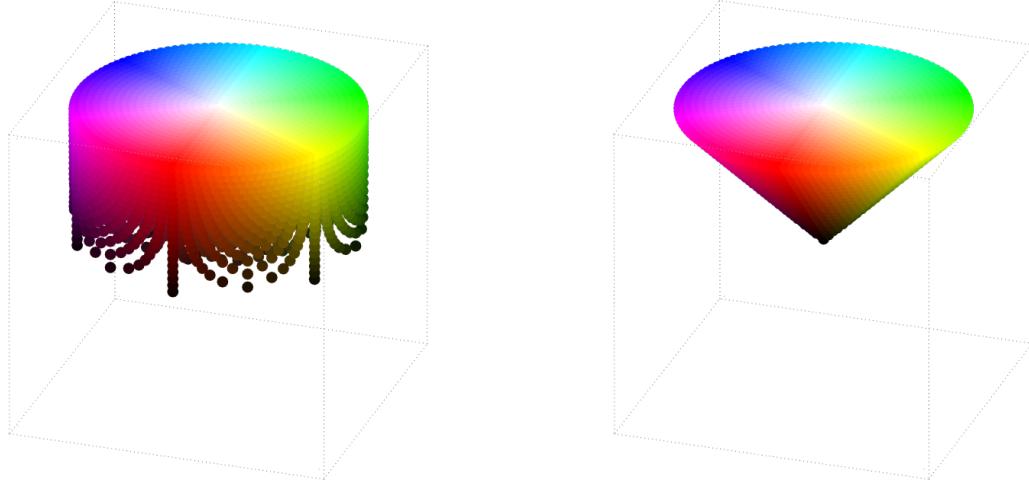
Pour cette couleur, nous avons

$$\begin{aligned} b &\approx \frac{g - b}{r - m}, \\ s &\approx M - m, \\ v &\approx M. \end{aligned}$$

On pourra par la suite normaliser b de façon à tenir, par exemple, entre 0 et 1, ou 0 et 6 (c’est un détail d’implémentation). Pour une implémentation complète, il faut bien entendu tester dans quel secteur se trouve la couleur dominante. Pour donner l’illusion d’un angle, on pourra avoir les « angles » dans $]-1, 1]$ pour le secteur rouge, dans $]1, 3]$ pour le secteur vert, et dans $]3, 5[$ pour le secteur bleu. On peut ajuster les « angles » rouges de façon à ce que tous les angles soient entre 0 et 1, ou, comme nous le ferons, entre 0 et 6. La fig. 3.4.7 montre une implémentation complète en *Mathematica*. L’implémentation est fortement inspirée de [94, 240]. Le mapping des couleurs *RGB* dans l’espace *HSV* cylindrique est montré à la fig. 3.4.8 (a); tandis que la fig. 3.4.8 (b) montre l’espace conique.

```
(* HSV cylindrique *)
RGBtoHSV[{r_, g_, b_}] :=
Module[{M = Max[r, g, b], m = Min[r, g, b], h,
s, v, d},
v = M;
s = If[M != 0,  $\frac{M-m}{M}$ , 0];
(* ôter la normalisation sur M donne le
modèle conique *)
d = M - m;
h = If[s == 0, 0,
If[r == M, Mod[(g - b) / d, 6],
(* façon de faire +6 si négatif *)
If[g == M, 2 + (b - r) / d,
4 + (r - g) / d]]];
{s, v, h} (* normalisation de h ici...
ou non *)]
]

HSVtoRGB[{h_, s_, v_}] := Module[{i, f, p, q, t},
i = Floor[h];
f = h - i;
p = v (1 - s); (* si s=0, alors achromatique! *)
q = v (1 - s f);
t = v (1 - s (1 - f));
Which[
i == 0, {v, t, p},
i == 1, {q, v, p},
i == 2, {p, v, t},
i == 3, {p, q, v},
i == 4, {t, p, v},
i == 5, {v, p, q}]
]
```

(a) Conversion de *RGB* à *HSV*.(b) Conversion de *HSV* à *RGB*.Figure 3.4.7 — Une implémentation *Mathematica* des conversions de *RGB* à *HSV*.(a) *HSV* cylindrique(b) *HSV* conique.Figure 3.4.8 — Les variantes de l'espace de couleurs *HSV*.

3.4.1.2 *HSL*

L'espace de couleurs *HSL* est si semblable à l'espace *HSV* qu'il semble que certains le confondent. En fait, même s'ils sont très semblables, ils diffèrent en deux points importants : comment ils estiment la luminosité et la saturation. Dans l'espace *HSV*, tel que présenté par Fishkin [94], on calcule la luminosité comme

$$l = \frac{1}{2}(\max(r, g, b) - \min(r, g, b)) = \frac{1}{2}(M - m)$$

tandis que d'autres implémentations posent plutôt $l = \frac{1}{3}(r + g + b)$. Le calcul de la saturation est

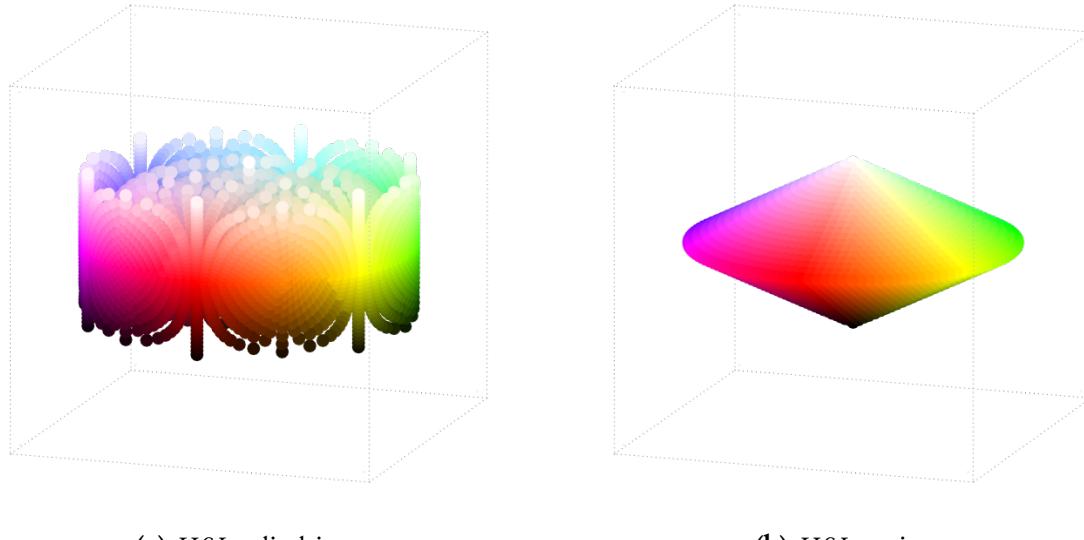


Figure 3.4.9 — Les variantes de l'espace de couleurs *HSL*.

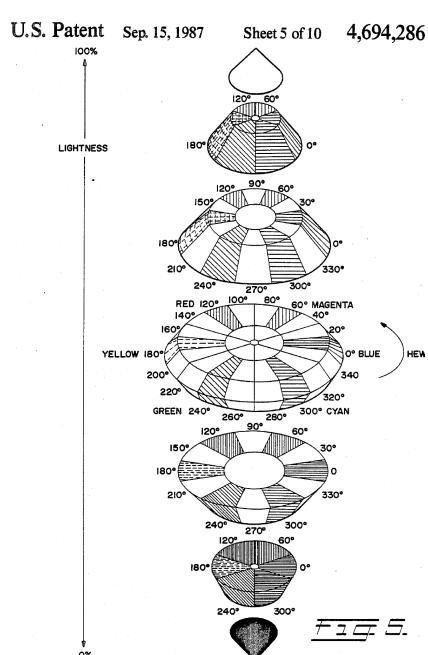


Figure 3.4.10 — *HSL*, dans le brevet de Bergstedt [43].

passablement plus compliqué :

$$s = \begin{cases} \frac{M-m}{M+m} & \text{si } l \leq \frac{1}{2} \\ \frac{M-m}{2-M-n} & \text{sinon.} \end{cases}$$

La transformation du cube *RGB* par l'algorithme de Fishkin disperse les points comme à la

fig. 3.4.9 (a), en espèce de couronne, où les couleurs très pâles ou très foncées sont dispersées dans des régions ténues de l'espace HSL . Au contraire, le centre se trouve densément peuplé de couleurs saturées. Ce que nous voulons, c'est avoir un espace également densément peuplé, comme à la fig. 3.4.9 (b) ou encore comme à la fig. 3.4.10.

Pour ce faire, nous allons modifier le calcul de la saturation, laissant le calcul de la luminosité inchangé. Si on pose plutôt

$$s = \max(r, g, b) - \min(r, g, b) = M - m,$$

nous obtenons bien la fig. 3.4.9 (b), mais il nous faut encore modifier le calcul de l'inverse. Il nous faut retrouver M et m à partir de l et s . Nous avons deux équations, deux inconnues,

$$\begin{aligned} l &= \frac{1}{2}(M - m) \\ s &= M - m. \end{aligned}$$

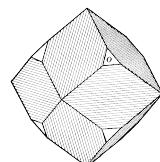
qu'il nous faut résoudre pour M et m . On trouve

$$\begin{aligned} M &= \frac{1}{2}(2l + s) \\ m &= \frac{1}{2}(2l - s). \end{aligned}$$

Il ne reste plus qu'à corriger l'implémentation de Fishkin pour obtenir le code de la fig. 3.4.11. Notons que l'implémentation revenir à l'original en changeant le calcul de s pour le code en commentaire dans la fig. 3.4.11 (a) et l'assignation de v pour le code en commentaire pour la fig. 3.4.11 (b).

*
* * *

Mais pourquoi insister sur les transformées coniques et ne pas se contenter des transformées cylindriques mêmes si elles n'utilisent pas très efficacement l'espace? Si on veut utiliser ces espaces de couleurs en compression de données, on ne veut pas assigner des codes à des combinaisons qui ne sont pas susceptibles d'être observées. Au contraire, on veut minimiser le nombre de bits émis, et donc adresser un espace aussi compact que possible. D'autres considérations entrent en jeu, comme la compaction d'énergie, dont nous discuterons amplement au chapitre ??.



```

Clear[HSLtoRGB]
HSLtoRGB[{h_, s_, l_}] :=
Module[{v, m1, m2, M1, M2, ss, i, f},
v =  $\frac{1}{2} (2 l + s)$ ;
(*If[l <  $\frac{1}{2}$ , l(1+s), l+s(1-l)];*)
If[v == 0, {0, 0, 0},
m1 = 2 l - v;
ss = 1 -  $\frac{m1}{v}$ ;
i = Floor[h];
f = h - i;
M1 = m1 + v ss f;
M2 = v (1 - ss f);
Which[
i == 0, {v, M1, m1},
i == 1, {M2, v, m1},
i == 2, {m1, v, M1},
i == 3, {m1, M2, v},
i == 4, {M1, m1, v},
i == 5, {v, m1, M2}
]
]
]
]

(* Fishkin, VIII.14 dans Graphics Gems (I),
p. 448+
(* modifiée pour double cone *)
Clear[RGBtoHSL]
RGBtoHSL[{r_, g_, b_}] :=
Module[{M = Max[r, g, b], m = Min[r, g, b], l, s, h},
l = (M + m) / 2;
If[M == 0, {0, 0, 0},
s = M - m; (* If[M!=m, If[l< $\frac{1}{2}$ ,  $\frac{M-m}{M+m}$ ,  $\frac{M-m}{2-M-m}$ ], 0];*)
h = If[s == 0, 0,
If[r == M, Mod[(g - b) / (r - m), 6],
(* façon de faire +6 si négatif *)
If[g == M, 2 + (b - r) / (g - m),
4 + (r - g) / (b - m)]];
{s, l}
]
]
]
]

(a) Conversion de RGB à HSL.

```

(b) Conversion de *HSL* à *RGB*.

Figure 3.4.11 — Une implémentation *Mathematica* des conversions de *RGB* à *HSL*.

3.4.2 CIÉ $L^*a^*b^*$, $L^*u^*v^*$ et CIÉDE2000

Les espaces de couleurs présentés jusqu'à maintenant tentaient simplement de représenter les couleurs dans un espace où une des composantes représente la luminosité tandis que les deux autres donnent de l'information sur la saturation et la teinte, soit sous la forme de différentes jaune-bleu et rouge-vert, soit sous la forme explicite de teinte et de saturation. Mais ces espaces ne sont pas perceptuellement uniformes, c'est-à-dire que la magnitude du changement perçu dans la couleur dépend de la direction du changement. Autrement dit, ces espaces ne garantissent pas une variation d'égale magnitude selon qu'un petit changement Δ soit fait dans la première, la seconde ou la troisième composante.

Ce problème a été remarqué il y a déjà fort longtemps et plusieurs ont tenté de mesurer, le plus précisément possible, la quantité de changement nécessaire pour qu'on remarque une variation dans la luminosité ou la teinte [53, 54, 73, 181, 269, 271]. Ce plus petit changement perçu est la JND (*just noticeable difference*), et dépend de la couleur, comme nous le montre la fig. 3.4.12, créée à partir des données de MacAdam [170–176], et maintes fois revisitée [64, 111, 151]. L'expérience consistait « simplement » à faire varier x et y à partir d'une couleur de départ jusqu'à ce qu'une variation soit perçue, et marquer le changement nécessaire. Ensuite, une ellipse est estimée pour chaque couleur initiale. L'orientation et la forme des ellipses nous montrent clairement que l'espace de couleur

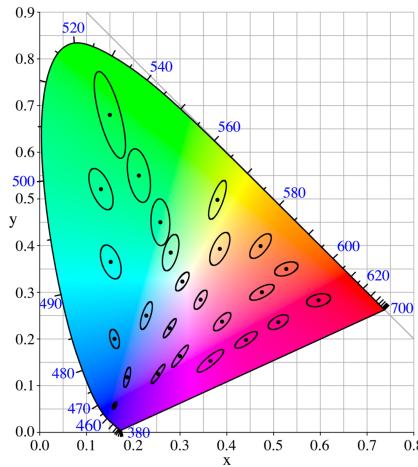


Figure 3.4.12 — Les ellipses de MacAdam (dessinées 10 fois plus grandes qu'en réel), d'après [170]. Source : Wikipedia.

CIÉ XYZ (ou xyz) n'est pas perceptuellement uniforme : on doit « marcher » moins loin dans une direction que dans une autre pour atteindre les limites des ellipses.

Un espace perceptuellement uniforme ne présenterait pas des ellipses, mais des cercles (ou des sphères si on considère les trois dimensions). Pour rendre l'espace de couleurs uniforme, il faut estimer les JND un peu partout dans cet espace, et compenser pour les déformations et les ramener à des cercles/sphères. Les espaces de couleurs $L^*a^*b^*$ et $L^*u^*v^*$ du CIÉ proposent justement une déformation de l'espace XYZ qui le rend « isotrope ».

3.4.2.1 $L^*a^*b^*$

Accepté comme standard par le CIÉ en 1976, l'espace $L^*a^*b^*$ est une transformation de l'espace XYZ pour le rendre uniforme [183, 184, 230]. L'espace $L^*a^*b^*$ dépend d'une fonction non linéaire f donnée par [104, 230] :

$$f(x) = \begin{cases} \sqrt[3]{x} & \text{si } x > \left(\frac{6}{29}\right)^3 \\ \frac{x}{3} \left(\frac{29}{6}\right)^2 + \frac{4}{29} & \text{sinon.} \end{cases} \quad (3.4.1)$$

tandis que l'inverse est donnée par

$$f^{-1}(z) = \begin{cases} z^3 & \text{si } z > \frac{6}{29} \\ 3\left(\frac{6}{29}\right)^2 \left(z - \frac{4}{29}\right) & \text{sinon.} \end{cases} \quad (3.4.2)$$

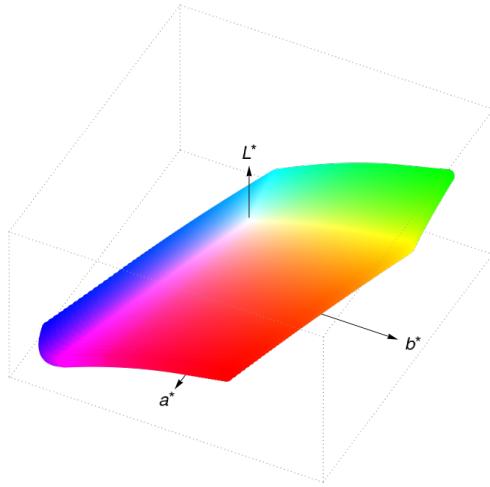


Figure 3.4.13 — La portion *RGB* de l'espace de couleurs $L^*a^*b^*$

En fonction d'un point blanc, par exemple D_{65} , dont les coordonnées XYZ sont $x_n = 95.4212$, $y_n = 100$ et $z_n = 18.8840$, et une couleur (x, y, z) on a les coordonnées (l^*, a^*, b^*) telles que

$$\begin{aligned} l^* &= 116f\left(\frac{y}{y_n}\right) - 16, \\ a^* &= 500\left(f\left(\frac{x}{x_n}\right) - f\left(\frac{y}{y_n}\right)\right), \\ b^* &= 200\left(f\left(\frac{y}{y_n}\right) - f\left(\frac{z}{z_n}\right)\right), \end{aligned} \quad (3.4.3)$$

où l^* est la luminosité, a^* correspond approximativement à la différence rouge-vert et b^* à la différence jaune-bleu. On revient à XYZ grâce à

$$\begin{aligned} x &= x_n f^{-1}\left(\frac{l^* + 16}{116} + \frac{a^*}{500}\right), \\ y &= y_n f^{-1}\left(\frac{l^* + 16}{116}\right), \\ z &= z_n f^{-1}\left(\frac{l^* + 16}{116} - \frac{b^*}{200}\right). \end{aligned} \quad (3.4.4)$$

Ces fonctions sont bricolées pour s'ajuster aux données recueillies pour mesurer les différences perçues et ont été choisies parmi plusieurs alternatives [59, 184].

La portion *RGB* de l'espace de couleurs $L^*a^*b^*$ est montrée à la fig. 3.4.13. Cet espace, donc, est censé être perceptuellement uniforme. C'est-à-dire que toutes les modifications à l^* , a^* et b^* qui donnent la même différence

$$\Delta E^* = \sqrt{(l_1^* - l_2^*)^2 + (a_1^* - a_2^*)^2 + (b_1^* - b_2^*)^2}$$

donne la même sensation de différence. De plus, dans cet espace, $\Delta E^* \approx 2.3$ correspond à la JND d'un humain normal.

3.4.2.2 $L^* u^* v^*$

L'espace de couleurs CIÉ $L^* u^* v^*$, en quelque sorte une version simplifiée de l'espace $L^* a^* b^*$, est une évolution des espaces de couleurs (dont nous ne discuterons pas plus avant) CIÉ 1960 et CIÉ 1964 [141, 169] et [232, p. 81]. Pour convertir de $X Y Z$ à $L^* u^* v^*$, nous avons d'abord

$$\begin{aligned} u' &= \frac{4x}{12y - 2x + 3} \\ u'_n &= 0.2009 \\ v' &= \frac{9y}{12y - 2x + 3} \\ v'_n &= 0.4610 \end{aligned}$$

où u'_n et v'_n sont des constantes choisies en fonction du blanc C, puis

$$\begin{aligned} l^* &= 116f\left(\frac{y}{y_n}\right) - 16, \\ u^* &= 13l^*(u' - u'_n) \\ v^* &= 13l^*(v' - v'_n) \end{aligned} \tag{3.4.5}$$

où $f(x)$ est donnée par l'éq. (3.4.1) (et son inverse par l'éq. (3.4.2)¹) et y_n le y du blanc C. L'inverse est donné par

$$\begin{aligned} u' &= \frac{u^*}{13l^*} + u'_n \\ v' &= \frac{v^*}{13l^*} + v'_n \end{aligned}$$

puis

$$\begin{aligned} y &= y_n f^{-1}\left(\frac{l^* + 16}{116}\right) \\ x &= y \frac{9u'}{4v'} \\ z &= y \frac{12 - 3u' - 20v'}{4v'} \end{aligned} \tag{3.4.6}$$

La portion *RGB* de l'espace de couleurs $L^* u^* v^*$ est montrée à la fig. 3.4.14.

1. La version de cette fonction que l'on trouve dans Wikipedia (juillet 2020) est cassée; le seuil à 8 (qui n'existe pas les versions du CIÉ) demande une constante supplémentaire pour que la partie cubique se joigne correctement à la partie linéaire.

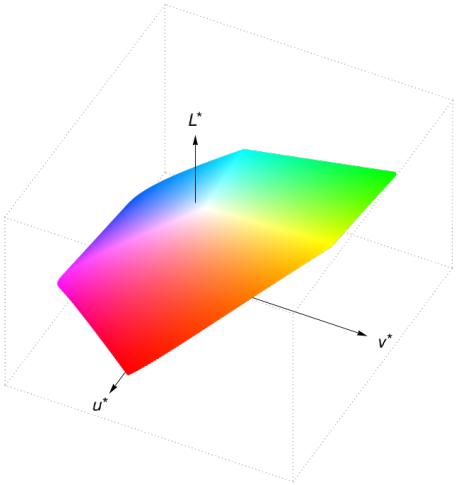


Figure 3.4.14 — La portion *RGB* de l'espace $L^*a^*b^*$

3.4.2.3 CIÉDE2000

Ce n'est pas à proprement parler un espace de couleurs, mais plutôt une métrique de différences de couleurs dans l'espace de couleurs $L^*a^*b^*$ [33, 168, 234]. La métrique est passablement plus compliquée que le ΔE^* de $L^*a^*b^*$ (elle fait une vingtaine d'équations comparativement à simple une distance euclidienne) mais est censée rendre encore beaucoup mieux les différences perçues lorsqu'on applique une petite différence de luminosité, de teinte, ou de saturation. Cette métrique a des visées industrielles où, dans certains processus, il peut être crucial de bien vérifier que les couleurs demeurent inchangées ou au moins à l'intérieur d'une certaine variation acceptable.

3.4.3 Autres espaces de couleurs

3.4.3.1 Munsell

Créé dans le but de donner des noms sans ambiguïté et très précisément aux couleurs, l'espace de couleurs de Munsell est un espace non linéaire qui découpe les couleurs selon un schéma classique de teinte, saturation (qu'il nomme « concentration ») et valeur [142, 187–189, 214]. Seule surprise, les couleurs primaires sont au nombre de 5 (rouge (R), jaune (Y), vert (G), bleu (B) et (P) pourpre) auxquelles viennent s'ajouter 5 couleurs intermédiaires (jaune-rouge (YR), jaune-vert (YG), bleu-vert (BG), bleu-pourpre (BP), et (PR) pourpre-rouge). Le cercle chromatique formé par ces couleurs est montré à la fig. 3.4.15. L'ensemble des couleurs (réalisables en *RGB*) est montré à la fig. 3.4.16, créée en utilisant les données du *Munsell Color Science Lab*, Rochester Institute of Technology.

En principe, le système de classement des couleurs dans l'espace de Munsell est méthodique (comme veut l'illustrer l'« arbre des couleurs », fig. 3.4.17), mais il comporte un défaut sérieux, du

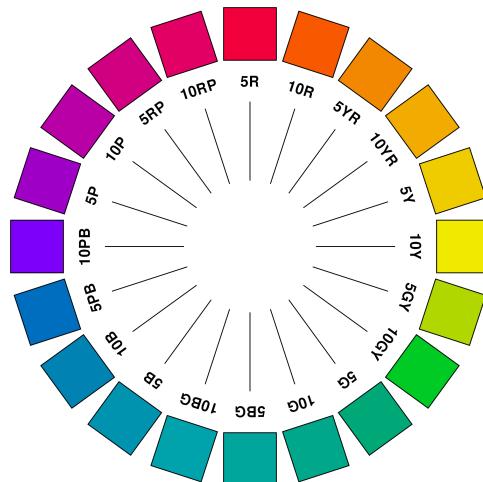


Figure 3.4.15 — Le cercle chromatique de l'espace de couleurs de Munsell. Source : Wikipedia.

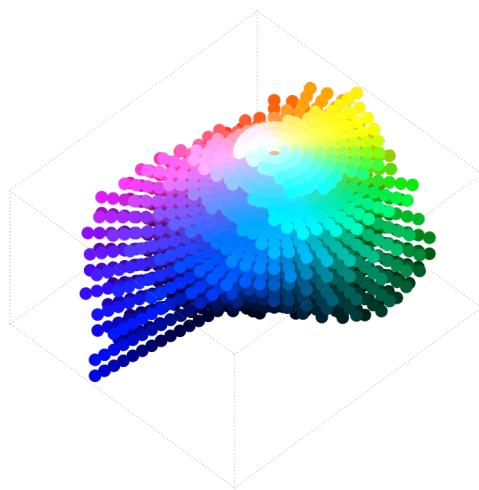


Figure 3.4.16 — L'espace de couleurs de Munsell (couleurs réalisables en *RGB*).

moins computationnellement : il n'y a pas de formule de conversion explicite, seulement une table établie entre codes de Munsell et des couleurs exprimées en coordonnées xyY (une variante de XYZ CIÉ 1931). Il faut alors construire une table de RGB à xyY à Munsell, ce qui est assez peu pratique [195].

3.4.3.2 Ostwald

Dans les années 1900–1920, Ostwald a tenté d'établir l'étude des couleurs comme une discipline scientifique sujette à la formalisation mathématique [110, 205–207]. Il divise les couleurs en teintes selon un cercle chromatique dominé par quatre couleurs primaires. Il utilise les couleurs *geld* (jaune), *rot* (rouge), *ultramarin blau* (ultra-marin) et (cyan-verdrâtre) *seegrün*, « vert marin »). Le cercle chromatique est montré à la fig. 3.4.18, tandis que le volume est montré à la fig. 3.4.19 (a). Ces couleurs

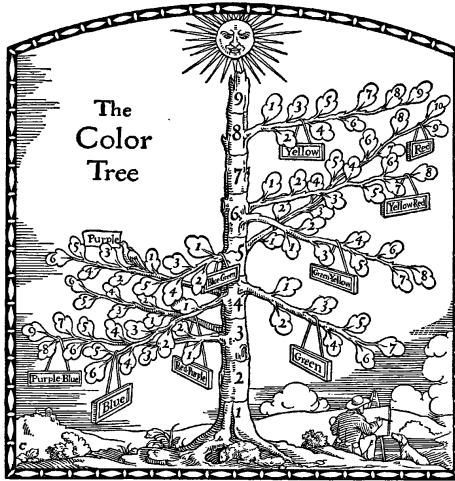


Figure 3.4.17 — L'arbre des couleurs de Munsell, tiré de [66] (domaine public).

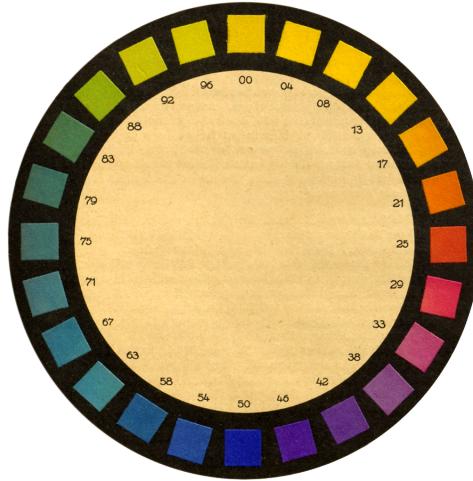


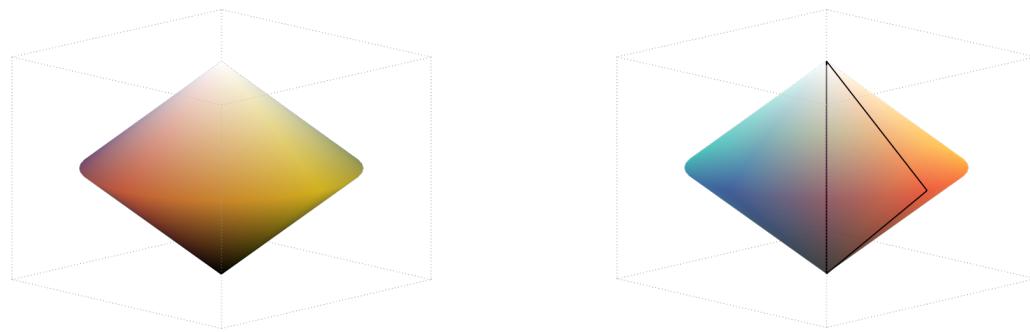
Figure 3.4.18 — Le cercle chromatique d'Ostwald, tiré de [207].

ont été vraisemblablement choisies en fonction des technologies de reproduction de couleurs qui lui étaient accessibles à l'époque.

Pour une teinte, il utilise un triangle équilatéral où un sommet est occupé par la couleur pure, la teinte, un autre par le noir (s pour *schwartz*, noir en Allemand) et le troisième par le blanc (w pour *weiß*, blanc en Allemand), ce triangle, comme le triangle de la fig. 3.1.2, interpole entre ces trois couleurs. Il impose, comme à la § 3.1.1, une normalisation des coordonnées. Ainsi, le mélange de couleur pure r , de blanc w et de noir s obéit à la contrainte

$$r + w + s = 100.$$

Les coordonnées dans le triangle sont exprimées par un système complexe et arbitraire : l'axe vertical est « numéroté » de a (blanc) à p (noir, p est la 16^e lettre de l'alphabet), la distance de la couleur pure est aussi notée de a (extérieur de la figure) à p (gris achromatique de même valeur),



(a) L'espace de couleurs Ostwald.

(b) Le triangle équilatéral.

Figure 3.4.19 — L'espace de couleurs Ostwald.

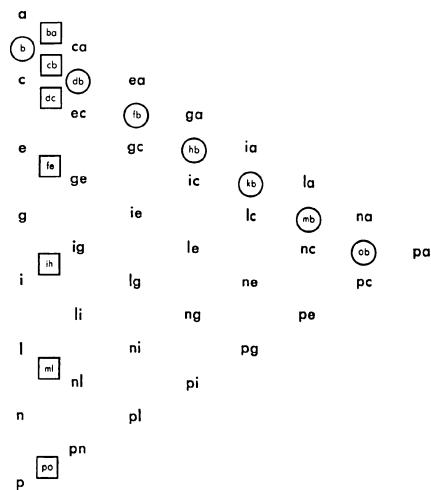


Figure 3.4.20 — Le système de coordonnées de l'espace Ostwald, d'après [110]

auxquelles s'ajoute un code pour la teinte. La fig. 3.4.20 donne une idée du système. Ce triangle, génère un volume de révolution (voir fig. 3.4.19 (b)) où le haut est dominé par le blanc, le bas par le noir, et l'« équateur » par les couleurs maximamente saturées.

3.4.3.3 NCS

L'espace de couleur NCS (pour *Natural Color System*) est basé sur quatre couleurs primaires issues de la théorie des couleurs complémentaires : une paire rouge-vert et une paire jaune-bleu [136, 137]. Comme pour *HSL*, l'espace forme un double cône dont les extrémités sont le blanc et le noir, et où les couleurs maximamente saturées sont réparties sur l'« équateur ». Les quatre couleurs primaires (qui ne sont pas sans rappeler les couleurs d'Ostwald) sont montrées à la fig. 3.4.21 (a) et sont données par

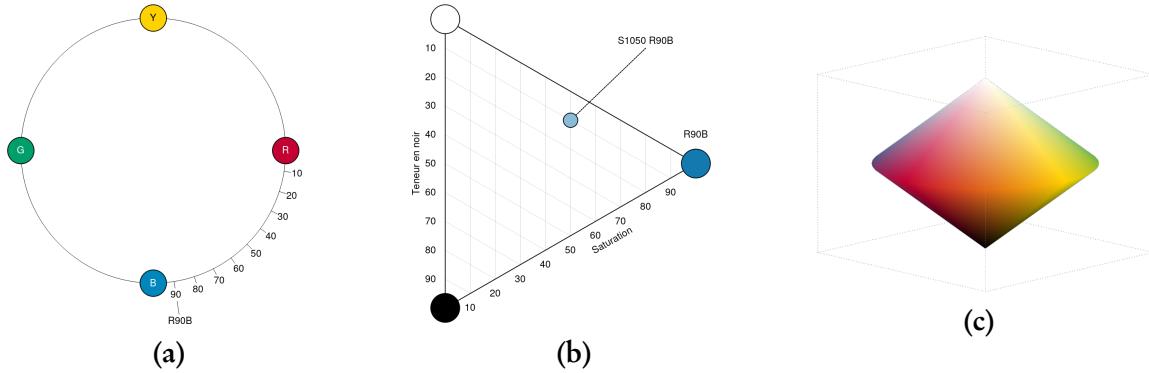


Figure 3.4.21 — L'espace de couleur NCS.

	<i>RGB</i>	Hexa
Blanc	1, 1, 1	ffffff
Rouge	0.77, 0.01, 0.2	c40233
Vert	0, 0.62, 0.42	009f6b
Bleu	0, 0.5, 0.74	0087bd
Jaune	1, 0.83, 0	ffd300
Noir	0, 0, 0	000000

Comme pour Ostwald, le système de coordonnées est basé sur la teinte, donnée par une des couleurs primaires R , Y , G , B . Ainsi la teinte $R90B$ représente 90 parts de B pour 10 de R , comme nous le montre la fig. 3.4.21 (a) qui définit un triangle entre le blanc, le noir et la couleur maximalement saturée. À l'intérieur de ce triangle (placé dans le volume de la fig. 3.4.21 (c)) comme à la fig. 3.4.19 (b)), on établit des coordonnées triangulaires, bien que plus simplement que pour le système d'Ostwald. NCS établit une grille où un axe est saturation et l'autre la teneur en noir; donnant des couleurs comme $S1050 R90B$ qu'il faut décoder comme 10% noir, 50% saturée, de la teinte $R90B$.

3.5 Remarques bibliographiques

Les espaces de couleurs CIÉ $L^*a^*b^*$ et $L^*u^*v^*$ perceptuellement uniformes ne sont pas les premiers espaces « équisensibles ». En effet, un certain nombre d'essais les ont précédés. Certains, avant MacAdam, ont tenté d'estimer la sensibilité de l'œil aux variations de couleur et d'intensité [220, 256], tandis que d'autres encore ont proposé des reparamétrisations de l'espace de couleur CIÉ de façon à rendre mieux compte de la sensibilité visuelle [141]. En particulier, le système de coordonnées « en racine cubique » qui survivra jusque dans les espaces de couleurs CIÉ 1976 $L^*a^*b^*$ et $L^*u^*v^*$ [104].

*
* * *

Pour un détail de l'histoire du développement de la télévision jusqu'à 1940, on pourra s'en référer à Fink [91]. La standardisation de la télévision couleur est arrivée en 1953, issue de travaux qui avaient débuté bien avant, dans les années 1930 [123, 276]. La standardisation a permis l'essor de la télévision, car auparavant, il y avait une pléthore de méthodes, farfelues ou astucieuses, mais toutes incompatibles entre elles [41, 255]. Après un développement des techniques qui permettent la transmission de la couleur [258, 259], il ne restait plus qu'à décider d'un standard pour rendre interopérables stations et récepteurs, indépendamment du fabriquant. Malgré tout, plusieurs standards se sont imposés : NTSC en Amérique du nord et Japon, SÉCAM en France, Russie et quelques pays francophones, et PAL ça et là [107].

*
* * *

Les espaces de couleurs dont la transformée peut être calculée sans perte, c'est-à-dire qu'une couleur *RGB* est restituée à l'identique après un passage par un second espace de couleurs, sont intéressants dans certains cas particuliers où il est primordial de restituer l'image exactement à la décompression. S'il n'est pas nécessaire de sauvegarder vos photos de voyage au bit près, d'autres images demanderont d'être restituées *exactement*. Pour les applications d'archivage muséal, médicales, de télédétection ou spatiales, les images acquises à grand coût, riches en information, doivent être stockées et compressées sans perte. Par conséquent, ces applications ne peuvent s'accommoder de transformées de couleurs approximatives. Pour un traitement plus général des transformées de couleurs sans perte, voir [245, 246].

*
* * *

Les théories sur la couleur ne datent pas d'hier. Déjà, au XII^e siècle Robert Grosseteste (c. 1168–1253), évêque de Lincoln de 1235 à 1254, s'est intéressé aux couleurs. Vers 1220 il écrit *De Colore*, un traité sur les couleurs où il propose essentiellement un modèle à deux cônes, avec le blanc et le noir aux sommets, mais avec sept couleurs primaires, les couleurs de l'arc-en-ciel sur son équateur [80, 242, 243]. Leon Battista Alberti (1404–1472), surtout connu pour son chiffre [143, 238], s'y est aussi intéressé. En 1435, dans *Della pittura*, il propose une double pyramide dont chaque coin correspond à l'une des quatre couleurs primaires, rouge, vert, bleu et jaune et dont les deux pointes correspondent au noir et au blanc [45]. L'organisation des couleurs en cercle chromatique (en « roue

des couleurs ») découle peut-être d'une observation judicieusement vague de Newton [197], reprise et formalisée par Boutet, qui sera le premier à donner explicitement le cercle chromatique [48].



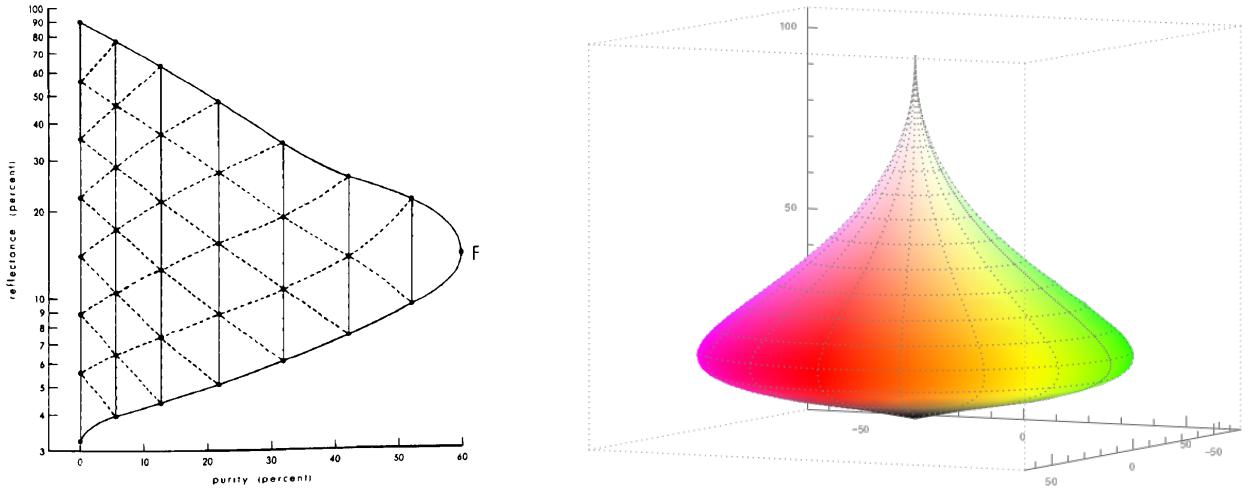
La formulation explicite du concept de couleurs primaires serait peut-être due à Boyle, au XVII^e siècle [49]. L'idée d'utiliser des couleurs primaires en imprimerie s'est imposée assez tôt. Dès 1725, le Francfortois Jakob Christoffel Le Blon fait paraître, d'abord en anglais, puis en français, « L'art d'imprimer des tableaux » où il préconise l'utilisation des couleurs cyan, magenta et jaune, en plus du noir [155, 156]. L'idée, qui s'avérera néanmoins adéquate, repose tout de même alors sur des bases pifométriques. Le premier à s'interroger et à formaliser le problème des couleurs primaires optimales sera Benson, dans les années 1860 [42].

*
* * *

La théorie des espaces de couleurs (qui nous intéresse ici par son application aux images numériques) n'est en fait qu'un sous-domaine du domaine plus vaste de la vision et de la perception des couleurs. Certains s'y intéressent à un niveau anatomique et biologique [29, 36], psychologique [76, 77, 99], physique [46, 231], esthétique voire frivole [27, 102], en général [55, 77], pratique [81], sous une perspective mathématique rigoureuse [84].

*
* * *

L'idée d'Ohta (voir § 3.3.5) d'utiliser les axes principaux du nuage de couleurs pour construire les vecteurs de base de son espace de couleur n'est pas bête du tout, mais computationnellement



(a) L'espace d'Ostwald selon [110].

(b) L'espace d'Ostwald, échelle linéaire.

Figure 3.5.1 — L'espace d'Ostwald et sa correction logarithmique.

non triviale, et il opte plutôt pour une approximation qu'il dit être satisfaisante. Cependant, il existe des techniques pour obtenir exactement les axes optimaux d'un nuage de points en un nombre quelconque de dimensions : il s'agit de l'analyse en composante principale, aussi connue sous le nom de transformée de Karhunen-Loève et transformée d'Hotelling selon les auteurs et les domaines [140, 145, 162, 209]. On pourrait argumenter que sur une machine à peu près moderne, calculer les composantes principales d'un nuage de points en trois dimensions n'est plus prohibitif, car il ne s'agit que de trouver les vecteurs propres (*eigenvectors*) d'une matrice de covariance 3×3 .

*
* * *

Pourquoi JPEG 2000 n'a-t-il pas su s'imposer malgré sa supériorité technique? D'une part, le standard JPEG 2000 est nettement plus complexe que le standard JPEG « classique », et son implémentation efficace plus difficile. D'autre part, les librairies ont mis du temps à être disponibles, étaient peu portables ou propriétaires (par opposition à « en logiciel libre »), et demandaient, pour l'époque, au tournant du millénaire, trop de mémoire et étaient lentes. Même aujourd'hui, le support pour JPEG2000 est incomplet et difficile à mettre en œuvre... Enfin, pour la plupart des gens, JPEG2000 n'apporte aucun avantage évident par rapport à JPEG. En effet, pour s'échanger des images par e-mail, JPEG fait bien l'affaire.

*
* * *

Le système NCS utilise quatre couleurs car Hering croyait que le jaune était une couleur primaire et non un mélange d'autres couleurs [125]. Le choix des couleurs m'est plus nébuleux pour l'espace de Munsell [200–202], que l'on peut voir en toutes couleurs [66]. On pourra aussi se surprendre que les couleurs primaires choisies ne soient pas tout à fait vives : *seegrün* et *geld* sont des couleurs que l'on voit comme composées dès le départ.

L'espace d'Ostwald, que nous avons présenté à la section § 3.4.3.2, est ordinairement présenté comme à la fig. 3.4.19 (a), mais si on observe attentivement les documents, on voit que l'axe vertical, qui correspond à la luminosité est toujours montré selon une échelle logarithmique [110], malgré qu'Ostwald lui-même ait initialement utilisé une échelle linéaire [205–207]. Si on montre l'espace d'Ostwald dans sa version 1942 selon une échelle linéaire, on obtient l'oignon de la fig. 3.5.1 (b), qui est nettement plus étrange.

Bibliographie

La bibliographie qui suit contient les références, présentées en ordre alphabétique des auteurs. En principe, aucun des ouvrages cités n'est un « introuvable ». Pour plusieurs références, surtout les références aux articles relativement récents, nous vous invitons à vérifier auprès de votre institution quelles ententes ont été établies auprès des éditeurs. Les journaux de l'IEEE, de l'ACM et d'Elsevier sont souvent accessibles sans frais par l'intermédiaire de votre établissement scolaire, collégial ou universitaire. Plusieurs auteurs auront aussi, comme vous le découvrirez si vous utilisez un engin de recherche, mis à disposition leurs publications sur leurs pages personnelles. D'autres références, enfin, contiendront des URL qu'il vous suffira de suivre pour atteindre l'information.

- [1] *Photography — Colour Negative Films for Still Photography — Determination of ISO Speed* — Rapport technique, ISO/IEC (novembre 1987).
- [2] *TIFF Revision 6 (FINAL)* — Rapport technique, Aldus Corporation (1992).
- [3] *ITU-T Recommendation H.261 : Video Codec For Audiovisual Services at $p \times 64$ kbits* — Rapport technique, International Telecommunication Union (1993).
- [4] *PhotoYCC Color Encoding and Compression Schemes* — Rapport technique n° 4, Kodak (1994).
- [5] *Xerox Color Encoding Standard XNSS 289005* — Rapport technique, Xerox (1994).
- [6] *CIFF Specification on Image Data File V. 1 r.4* — Rapport technique, Canon Inc. (1997).
- [7] *IEC/4WD 61966-2-1 : Colour Measurement and Management in Multimedia Systems and Equipment — Part 2-1 — Default Colour Space — sRGB* — Rapport technique, IEC (1998).
- [8] *ISO/IEC 61966-2-2 : Multimedia Systems and Equipment — Colour Measurement and Management — Part 2-2 — Colour Management — Extended Colour Space — scRGB* — Rapport technique, IEC (1998).
- [9] *Rec. ITU-R BT.470-6 : Conventional Television System* — Rapport technique, International Telecommunication Union (1998).
- [10] *Color Correction for Image Sensors* — Rapport technique n° MTD/PS-0534-2, (2003).
- [11] *Photography — Colour Reversal Camera Films — Determination of ISO Speed* — Rapport technique, ISO/IEC (octobre 2003).
- [12] *Rec. ITU-T T.4 : Standardization of Group 3 Facsimile Terminals for Document Transmission* — Rapport technique, International Telecommunication Union (juillet 2003).
- [13] *ISO/IEC 15948 :2004 — Information Technology, Computer Graphics and Image Processing — Portable Network Graphics (PNG) Functional Specification* — Rapport technique, ISO/IEC (2004).

- [14] *SMPTE Standard for Television : Composite Analog Video Signal : NTSC for Studio Applications* — Rapport technique, SMPTE (2004).
- [15] *LizardTech DjVu Reference* — Rapport technique, LizardTech (2005).
- [16] *Code of Federal Regulation (CFF) §73.682 TV Transmission Standards* — Rapport technique, US Government Publications (2010).
- [17] *Rec. ITU-R BT.601.7 : Studio Encoding Parameters of Digital Television for Standard 4:3 and Wide-Screen 16:9 Aspect Ratios* — Rapport technique, International Telecommunication Union (2011).
- [18] *CIPA DC-008-2012 Exchangeable Image File Format for Digital Still Cameras : EXIF Version 2.3* — Rapport technique, Camera & Imaging Products Association (2012).
- [19] *Digital Negative (DNG) Specification v1.4.0.0* — Rapport technique, Adobe Systems Inc. (2012).
- [20] *Rec. ITU-R BT.2020-2 : Parameter Values for the Ultra-High Definition Television Systems for Production and International Programme Exchange* — Rapport technique, International Telecommunication Union (2015).
- [21] *Rec. ITU-R BT.709-6 : Parameter Values for the HDTV Standards for Production and International Programme Exchange* — Rapport technique, International Telecommunication Union (2015).
- [22] *Rec. ITU-R BT.2100-1 : Image Parameter Values for High-Dynamic Range Television for Use in Production and International Programme Exchange* — Rapport technique, International Telecommunication Union (2017).
- [23] *Windows Metafile Format V 14.0* — Rapport technique, Microsoft (2017).
- [24] *Rolling Shutter* — https://en.wikipedia.org/wiki/Rolling_shutter (2020).
- [25] *The Atari ST and Amiga computing evolution!* — <http://retroshowcase.com/index.php?p=article&artid=6> (2020).
- [26] Tim Ahrens, Jack Browne, Hunter Scales — *What's Inside Radio Shack's Color Computer?* — BYTE, vol. 6, n° 3 (mars 1981), p. 90–130.
- [27] Josef Albers — *Interaction of color : 50th Anniversary Edition* — Yale University Press (2013).
- [28] Felix Albu, Vlad Paenaru, Alexandru Drimbarean — *US Patent 8,508,652 B2 : Autofocus Method* — US Patent Office (2013).
- [29] M. A. Ali, M. A. Klyne — *La vision chez les vertébrés* — Décarie/Masson (1986).
- [30] David Alleysson, Sabine Süsstrunk, Jeanny Hérault — *Color Demosaicing by Estimating Luminance and Opponent Chromatic Signals in the Fourier Domain* — dans *Procs IS&T/SID 10th Color Imaging Conference*, (novembre 2002), p. 331–336.
- [31] David Alleysson, Sabine Süsstrunk, Jeanny Hérault — *Linear Demosaicing Inspired by the Human Visual System* — IEEE Trans. on Image Processing, vol. 14, n° 4 (avril 2005), p. 439–449.
- [32] David Alleysson, Sabine Süsstrunk, Joanna Marguier — *Influence of Spectral Sensitivity Functions on Color Demosaicing* — dans *Procs. Color and Imaging Conference*, (2003), p. 351–357.
- [33] D. H. Alman, *et al.* (éds) — *Improvement to Industrial Colour-Difference Evaluation* — Rapport technique n° CIE 142-2001, CIÉ Central Bureau (2001).
- [34] G. F. Amelio, M. F. Tompsett, G. E. Smith — *Experimental Verification of the Charge Coupled Device Concept* — Bell System Technical Journal, vol. 49, n° 4 (1970), p. 593–600.
- [35] Deepak Aneja — *Edge Strength Based Fuzzification of Color Demosaicking Algorithms* — Mémoire de maîtrise, Dept. Information Technology, Delhi Technological University, (2013).

- [36] David A. Atchison, George Smith — *Optics of the Human Eye* — Butterworth Heinemann (2000).
- [37] David Baker — *NASA Hubble Space Telescope : 1990 Onwards (Including All Upgrades), Owner's Workshop Manual* — Haynes Publishing (2015).
- [38] Octavian Baltag — *History of Automatic Focusing Reflected by Patents* — Science Innovation, vol. 4, n° 1 (2015), p. 1-17.
- [39] Bryce E. Bayer — *US Patent 3,971,065 : Color Imaging Array* — US Patent Office (1976).
- [40] Philippe Bellaïche — *Les secrets de l'image vidéo* — 11^e éd., Eyrolles (2018).
- [41] Raymond Snyder Bennett W. Cooke, Silas C. Narland — *Coyne Television Cyclopedia* — Coyne Electrical and Television-Radio School (1951).
- [42] William Benson — *Principles of the Science of Colour, concisely stated to Aid and Promote Their Useful Application in the Decorative Arts* — Chapman & Hall (1868).
- [43] Gar A. Bergstedt — *US Patent 4,694,286 : Apparatus and Method for Modifying Displayed Color Images* — US Patent Office (septembre 1987).
- [44] Harry Black — *Canadian Scientists and Inventors : Biographies of People who Shaped our World* — 2^{de} éd., Prembrooke Publishers (2-8).
- [45] Dott. Anicio Bonucci — *Opere Volgari di Leon Battista Alberti (4 Vols)* — Typografia Galileiana (1843).
- [46] P. J. Bouma — *Physical Aspects of Color* — N. V. Philips Gloeilampenfabrieken (1947).
- [47] T. Boutell, et al. — *RFC 2083 : PNG (Portable Network Graphics) Specification* — Rapport technique, IETF (1997).
- [48] Claude Boutet — *Traité de la peinture en Mignature pour apprendre aisément à Peindre sans Maître* — Loüis & Henry van Dole, La Haye (1708).
- [49] Robert Boyle — *Experiments and Considerations Touching Colours : The Beginning of an Experimental History of Colours* — Henry Herringman (1664).
- [50] W. S. Boyle, G. E. Smith — *Charge Coupled Semiconductor Devices* — Bell System Technical Journal, vol. 49, n° 3 (1970), p. 587-593.
- [51] C. Wayne Brown, Barry J. Shepherd — *GRaphics File Formats : Reference and Guide* — Manning (1995).
- [52] George Brown, D. G. C. Luck — *Principles and Development of Color Television Systems* — RCA Review, vol. XIV, n° 2 (juin 1953), p. 144-204.
- [53] W. R. J. Brown — *Color Discrimination of Twelve Observers* — J. of the Optical Society of America, vol. 47, n° 2 (1957), p. 808-834.
- [54] W. R. J. Brown, David L MacAdam — *Visual Sensitivities to Combined Chromaticity and Luminance Differences* — J. of the Optical Society of America, vol. 39, n° 10 (octobre 1949), p. 808-834.
- [55] Manlio Brusatin — *Histoire des couleurs* — Champs/Flammarion (1986).
- [56] Antoni Buades, Bartomeu Coll, Jean-Michel Morel, Catalina Sbert — *Self-Similarity Driven demosaicing* — Image Processing Online, vol. 1 (2011), p. 51-56.
- [57] Dan Budny — *What is IC_TC_P ?* — Rapport technique n° V071, Dolby Systems (avril 2016).
- [58] Velma I. Burns — *Design and Calibration of Luminance Standards for the Three Color Phosphors used in Color Television Tubes* — Rapport technique n° 4870, National Bureau of Standards (octobre 1956).
- [59] Velma I. Burns — *CIÉ Colorimetry Committee—Working Program on Color Differences* — J. of the Optical Society of America, vol. 64, n° 6 (juin 1974), p. 896-897.

- [60] Gunter Buxbaum, *et al.* — *Colored Pigments* — dans Hans G. Völz, *et al.* (éds), *Ullmann's Encyclopedia of Industrial Chemistry*, vol. 2, John Wiley & Sons (2000), chapitre 3, p. 24–97.
- [61] Peter B. Catrysse, Bryan A. Wandell, Abbas El Gamal — *Comparative Analysis of Color Architectures for Image Sensors* — dans *Proc. Sensors, Cameras, and Applications of Digital Photography (Electronic Imaging '99)*, (1999), p. 26–35.
- [62] Ed Chang, Shiufun Cheung, Davis Pan — *Color Filter Array Recovery Using a Threshold-Based Variable Number of Gradients* — dans *Proc. Sensors, Cameras, and Applications of Digital Photography (Electronic Imaging '99)*, (1999), p. 36–43.
- [63] Lanlan Chang, Yap-Peng Tan — *Hybrid Color Filter Array Demosaicing for Effective Artifact Suppression* — *Journal of Electronic Imaging*, vol. 15, n° 1 (2006), p. 013003-1–013003-17.
- [64] Emmanuel Chevallier, Ivar Farup — *Interpolation of the MacAdam Ellipses* — Rapport technique n° HAL-01625921v2, HAL Archives Ouvertes (2018).
- [65] King-Hong Chung, Yuk-Hee Chan — *Color Demosaicing Using Variance of Color Differences* — *IEEE Trans. on Image Processing*, vol. 15, n° 10 (octobre 2006), p. 2944–2955.
- [66] T. M. Cleland — *A Grammar of Color : Arrangements of Strathmore Papers, in a Variety of Printed Color Combinations according to the Munsell Color System* — Strathmore Paper Company (1921).
- [67] John T. Compton, John F. Hamilton, Jr — *US Patent 8,139,130 B2 : Image Sensor with Improved Light Sensitivity* — US Patent Office (2012).
- [68] IBM Computers — *IBM PC Technical Reference* — Personal Computer Hardware Reference Library, IBM (1983).
- [69] IBM Computers — *IBM Color/Graphics Monitor Adapter* — Personal Computer Hardware Reference Library, IBM (1984).
- [70] Guy Côté, E. Frederiksen — *US Patent 8,638,342B2 : System and Method for Demosaicing Image Data Using Weighted Gradients* — US Patent Office (2014).
- [71] Michael A. Covington — *Astrophotography for the Amateur* — Cambridge University Press (1999).
- [72] Jan Davidse — *Transmission and Decoding in Colour Television* — Thèse de doctorat, Technische Hogeschool, Technische Uniseriteit Eindhoven (1964).
- [73] Hugh R. Davidson — *Calculation of Color Differences from Visual Sensitivity Ellipsoids* — J. of the Optical Society of America, vol. 41, n° 12 (décembre 1951), p. 1052–1056.
- [74] Hendrik de Lang, Gijsbertus Bouwhuis — *US Patent 3,202,039 : Optical System for a Color Television Camera* — US Patent Office (1965).
- [75] F. W. de Vrijer — *Fundamentals of Color Television* — Philips Technical Review, vol. 19, n° 3 (1957/1958), p. 86–97.
- [76] André Delorme — *Psychologie de la perception* — Université de Montréal (1995).
- [77] Maurice Déribéré — *La couleur* — n° 220 dans la collection *Que sais-je?*, Presses universitaires de France (1964).
- [78] V. G. Devereux — *Limiting of YUV Digital Video Signals* — Rapport technique n° BBC RD 1987/22, BBC Research Department (1987).
- [79] Peter L. P. Dillon, David M. Lewis, Frank G. Kaspar — *Color Imaging System Using a Single CCD Area Array* — *IEEE Trans. on Consumer Electronics*, vol. ED-25, n° 2 (février 1978), p. 102–107.

- [80] Greti Dinkova-Bruun, Giles E. M. Gasper, Micahel Huxatble, Tom C. b. McLeish, Cecilia Panti, Hannah E. Smithson — *The Dimesnions of Colour : Robert Goressetest's De Colore* — Institute of Medieval and Renaissance Studies, Durham University (2013).
- [81] Jean Dourgnon, Paul Kowalski — *La reproduction des couleurs* — n° 472 dans la collection *Que sais-je?*, Presses universitaires de France (1966).
- [82] Éric Dubois — *Frequency-Domain Methods for Demosaicing of Bayer-Sampled Color Images* — IEEE Signal Processing Letters, vol. 12, n° 12 (2005), p. 847–850.
- [83] Éric Dubois — *Filter Design for Adaptive Frequency-Domain Bayer Demosaicing* — dans *IEEE Int. Conf. on Image Processing (ICIP)*, (2006), p. 2705–2708.
- [84] Éric Dubois — *The Structure and Properties of Color Spaces and the Representation of Color Images* — Morgan & Claypool (2010).
- [85] Harold E Ennes — *Television Broadcasting : Equipment, Systems, and Operating Fundamentals* — Howard S. Sams & Co. (1970).
- [86] Hugh S. Fairman, Michael H. Brill, Henry Hemmendinger — *How the CIÉ 1931 Color-Matching Functions Were Derived from Wright-Guild Data* — Color Research and Application, vol. 32, n° 1 (février 1997), p. 11–13.
- [87] Sina Farsiu, Michael Elad, Peyman Milanfar — *Multi-Frame Demosaicing and Super-Resolution of Color Images* — IEEE Trans. on Image Processing, vol. 15, n° 1 (janvier 2006), p. 141–159.
- [88] Sina Farsiu, Dirk Robinson, Michael Elad, Peyman Milanfar — *Advances and Challenges in Super-Resolution* — Int. Journal of Imaging Systems and Technology, vol. 14, n° 2 (janvier 2004), p. 47–57.
- [89] Alan Fein, Ete Zoltan Suzts — *Photoreceptors : Their Role in Vision* — Cambridge university Press (1982).
- [90] Ted Felix — *Image Pac compression and JPEG compression : What's the Difference?* — <http://tedfelix.com/PhotoCD/ImagePacvsJPEG.html> (2020).
- [91] Donald G. Fink (éd.) — *Television Standards and Practice : Selected Papers form the Proceedings of the National Television System Committee and its Panels* — McGraw-Hill (1941).
- [92] Walter Fischer — *Digitale Fernsehtechnik in Theorie und Praxis* — Springer (2006).
- [93] Walter Fischer — *Digital Video and Audio Broadcasting Technology* — 3^e éd., Springer (2010).
- [94] Ken Fishkin — *A Fast HSL-to-RGB Transform* — dans Andrew S. Glassner (éd.), *Graphics Gems*, Academic Press (1990), chapitre VIII.14, p. 448–449.
- [95] Agner Fog — *Optimization Manuals, Vol 4 : Instruction Tables* — Technical University of Denmark (2019).
- [96] A. Forsey, S. gungor — *Demosaicing Images from Colour Cameras for Digital Image Correlation* — Optics and Lasers in Engineering, vol. 86 (2016), p. 20–28.
- [97] Eric R. Fossum — *Active Pixel Sensors : Are CCDs Dinosaurs?* — dans *Procs. Charged Compled Devices and Solid-State Optical Sensors III*, (1993), p. 2-14. (SPIE vol. 1900).
- [98] Eric R. Fossum, Donald B. Hondongwa — *A Review of the Pinned Photodiode for CCD and CMOS Image Sensors* — IEEE Journal of the Electron Devices Society, vol. 2, n° 3 (mai 2014), p. 33–43.
- [99] Robert Francès — *La perception* — n° 1076 dans la collection *Que sais-je?*, Presses universitaires de France (1963).
- [100] Franquin — *Gaston Lagaffe : Le gang des gaffeurs* — Dupuis (1973). (reprint 2016).

- [101] Claude Galarneau — *Les Desbarats : une dynastie d'imprimeur-éditeurs (1794–1893)* — Les cahiers des dix, vol. 46 (1991), p. 125–149.
- [102] Amandine Gallienne — *Les 100 mots de la couleur* — n° 4081 dans la collection *Que sais-je?*, Presses universitaires de France (2017).
- [103] Pascal Getreuer — *Malvar-He-Cutler Linear Image Demosaicking* — Image Processing Online, vol. 1, p. 2011.
- [104] L. G. Glassner, A. H. McKinney, C. D. Reilly, P. D. Schnelle — *Cube-Root Color Coordinate System* — J. of the Optical Society of America, vol. 48, n° 10 (novembre 1958), p. 736–740.
- [105] Micha Galor Gluskin — *US Patent 100,044,959 B2 : Mask-Less Phase Detection Autofocus* — US Patent Office (2018).
- [106] Micha Galor Gluskin — *US Patent 10,044,926 B2 : Optimized Phase Detection Autofocus (PDAF) Processing* — US Patent Office (2018).
- [107] Pierre Godou — *Télévisions du monde : guide pratique de la réception longue distance* — Eyrolles (1983).
- [108] Rafael C. Gonzalez, Richard E. Woods — *Digital Image Processing* — 2^{de} éd., Prentice-Hall (2002).
- [109] Bart Goossens, Hiep Luong, Jan Aelterman, Alexandra Pižurica — *An Overview of State-of-the-Art Denoising and Demosaicing Techniques : Toward a Unified Framework for Handling Artifacts During Image Reconstruction* — dans *Procs. Int. Image Sensor Workshop (IISW)*, (2015). Session 12.
- [110] Walter C. Granville — *The Color Harmoni Manual, a Color Atlas Based on the Ostwald Color System* — Color Research and Application, vol. 19, n° 2 (avril 1994), .
- [111] Jens Gravesen — *The Metric of Colour Space* — Graphical Models, vol. 82 (novembre 2015), p. 77–86.
- [112] Robert O. Green, Michael L. Eastwood, Charles M. Sarture, Thomas G. Chrien, Mikael Aronsson, Bruce J. chippendale, Jessica A. Faust, Betina E. Pavri, Christopher J. Chovit, Manuel Solis, Martin R. Olah — *Imaging Spectroscopy and the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS)* — Remote Sensing of Environment, vol. 65, n° 3 (1998), p. 277–320.
- [113] Bernard Grob — *Basic Television : Principles and Servicing* — MgGraw-Hill (1975).
- [114] Liron D. Grossmann, Yonina C. Eldar — *Enhancement of Color Images by Efficient Demosaicing* — Rapport technique n° CCIT #508, Technion-Israel Institute of Technology (2004).
- [115] J. Guild — *The Colorimetric Properties of the Spectrum* — Philosophical Trans. of the Royal Society of London, Series A : Containing Papers of a Mathematical or Physical Character, vol. 230 (1932), p. 149–187.
- [116] Bahadir K. Gunturk, Yucel Altunbasak, Russell M. Mersereau — *Color Plane Interpolation Using Alternating Projections* — IEEE Trans. on Image Processing, vol. 11, n° 9 (septembre 2002), p. 997–1013.
- [117] Bahadir K. Gunturk, John Glotzbach, Yucel Altunbasak, Ronald W. Schafer, Russell M. Mersereau — *Demosaicing : Color Filter Interpolation* — IEEE Signal Processing Magazine, vol. 22, n° 1 (janvier 2005), p. 44–54.
- [118] Maya R. Gupta, Ting Chen — *Vector Color Filter Demosaicing* — dans *Sensors and Camera Systems for Scientific, Industrial and Digital Applications II*, (2001). (SPIE Vol. 4306).
- [119] Kengo Hayasaka, Kenji Yamamoto — *Japan Patent JP2011-171858A : Image Processing Apparatus, Image Processing Method, Image Processing Program and Imaging Device* — Japan Patent Office (2011).
- [120] Eugene Hecht — *Optics* — 5^e éd., Pearson (2017).

- [121] Yacov Hel-Or — *US Patent 6,404,918(B1) : Image Demosaicing Method Utilizing Directional Smoothing* — US Patent Office (2002).
- [122] Yacov Hel-Or — *The Canonical Correlations of Color Images and Their Use for Demosaicing* — Rapport technique n° HPL-2003-164(R.1), HP Laboratories (février 2004).
- [123] Scott Helt — *Practical Television Engineering* — 2^{de} éd., Rinehart Books (1953).
- [124] John L. Hennessy, David A. Patterson — *Computer Architecture : A Quantitative Approach* — Morgan Kaufmann (2006).
- [125] Ewald Hering — *Zur lehre von Lichtsinne* — Druck und Verlag Carl Gerold's Sohn (1878).
- [126] Keigo Hirakawa, Thomas W. Parks — *Adaptive Homogeneity-Directed Demosaicing Algorithm* — IEEE Trans. on Image Processing, vol. 14, n° 3 (mars 2005), p. 360–369.
- [127] Keigo Hirakawa, Thomas W. Parks — *Joint Demosaicing and Denoising* — dans *IEEE Int. Conf. on Image Processing (ICIP)*, (2005).
- [128] Keigo Hirakawa, Thomas W. Parks — *Joint Demosaicing and Denoising* — IEEE Trans. on Image Processing, vol. 15, n° 8 (août 2006), p. 2146–2157.
- [129] Keigo Hirakawa, Patrick J. Wolfe — *Second-Generation Color Filter Array and Demosaicking Designs* — dans *Procs. Visual Communications and Image Processing*, (2008), p. 68221P-1–68221P-12. (SPIE Vol. 6822).
- [130] Keigo Hirakawa, Patrick J. Wolfe — *Spatio-Spectral Color Filter Array Design for Enhanced Image Fidelity* — dans *Procs. IEEE Int. Conf. on Image Processing (ICIP)*, (2010), p. II-81-II-94.
- [131] August Hoen — *US Patent 27,981 : Composition for Etching Stone* — US Patent Office (1860).
- [132] August Hoen — *US Patent 227,782 : Lithographic Process* — US Patent Office (1880).
- [133] August Hoen — *US Patent 227,730 : Lithographic Process* — US Patent Office (1883).
- [134] Jong-Uk Hou, Han-Ul Jang, Heung-Kyu Lee — *Hue Modification Estimation Using Sensor Pattern Noise* — dans *Procs. IEEE Int. Conf. on Image Processing (ICIP)*, (2014), p. 5281–5291.
- [135] P. H. Howells — *The Concept of Transmission Primaries in Color Television* — Procs. of the IRE, vol. 42, n° 1 (janvier 1954), p. 134–138.
- [136] Anders Hård, Lars Sivik, Gunnar Tonnquist — *NCS, Natural Color System - From Concept to Research and Applications, Part I* — Color Research and Application, vol. 21, n° 3 (1996), p. 180–205.
- [137] Anders Hård, Lars Sivik, Gunnar Tonnquist — *NCS, Natural Color System - From Concept to Research and Applications, Part II* — Color Research and Application, vol. 21, n° 3 (1996), p. 206–220.
- [138] Yizhen Huang, Yangjing Long — *Demosaicking Recognition with Application in Digital Photo Authentication Based on a Quadratic Pixel Correlation Model* — dans *Procs. IEEE Conf. on Computer Vision and Pattern Recognition*, (2008), p. 1–8.
- [139] Kai Hwang — *Advanced Computer Architecture : Parallelism, Scalability, Programmability* — McGraw-Hill (1993).
- [140] I.T. Jolliffe — *Principal Component Analysis* — Springer-Verlag (1986).
- [141] Deane B. Judd — *A Maxwell Triangle Yielding Uniform Chromaticity Scales* — J. of the Optical Society of America, vol. 25, n° 1 (janvier 1935), p. 24–35.
- [142] Deane B. Judd, Dorothy Nickerson — *One Set of Munsell Re-renotations* — Rapport technique n° 192693, National Bureau of Standards (juin 1967).

- [143] David Kahn — *The Codebreakers* — MacMillan (1967).
- [144] Henry R. Kang — *Color Technology for Electronic Imaging Devices* — SPIE Optical Engineering Press (1996).
- [145] Kari Karhunen — *Über Lineare Methoden in der Wahrscheinlichkeitsrechnung* — Suomalainen Tiedeakatemia (1947).
- [146] David C. Kay, John R. Levine — *Graphics File Formats* — TAB Books (1992).
- [147] Daniel Keren, Maragarita Osadchy — *Restoring Subsampled Color Images* — Machine Vision and Applications, vol. 11, n° 4 (décembre 1999), p. 197–202.
- [148] Ron Kimmel — *Demosaicing: Image Reconstruction from Color CCD Samples* — IEEE Trans. on Image Processing, vol. 8, n° 9 (septembre 1999), p. 1221–1228.
- [149] Timothy Knight, Colving Pitts, Kurt Akely, Yurly Romanenko, Carl (Warren) Craddock — *US Patent 9,300,932 B2 : Optimization of Optical Systems for Improved Lightfield Capture and Manipulation* — US Patent Office (2016).
- [150] Karl Knop, Rudolf Morf — *A New Class of Mosaic Color Encoding Patterns for Single-Chip Camera* — IEEE Trans. on Electronic Devices, vol. ED-32, n° 8 (août 1985), p. 1390–1395.
- [151] Jan Koenderink, Andrea van Doorn, Karl Gegenfurtner — *Graininess of RGB-Display Space* — iPerception, vol. 9, n° 5 (2018), p. 1–46.
- [152] Arthur König — *Über den Helligkeitswert der Spektralfarben bei Verschiedener absoluter Intensität* — Verlag von Leopold Voss (1891).
- [153] Yosuke Kusaka — *US Patent 5589909 : Automatic Focus Adjustment Camera with High-Speed Sequential Photography Capability and Method of Automatic Focus Adjustment Therefor* — US Patent Office (1996).
- [154] Emily Lakdawalla — *The Design and Engineering of Curiosity : How the Mars Rover Performs its Job* — Springer (2018).
- [155] Jakob Christoffel Le Blon — *L'art d'imprimer les tableaux* — P. G. Le Mercier, Imprimeur-libraire (1756).
- [156] Jakob Christoffel Le Blon — *Coloritto; or the Harmony of Colouring in Painting, reduced to the Mechanical Practice under Easy Precepts, and Infallible Rules, together with some Colour'd Figures, in order to render the said precepts and rules intelligible, not only to painters but even to all lovers of Painting* — (inconnu) (c. 1725).
- [157] Sheldon Leemon — *Inside Amiga Graphics* — Compute! Publications (1986).
- [158] Xin Li — *Demosaicing by Successive Approximations* — IEEE Trans. on Image Processing, vol. 14, n° 3 (mars 2005), p. 370–379.
- [159] Xin Li, Bahadir K. Gunturk, Lei Zhang — *Image Demosaicing : A Systematic Survey* — dans *Proc. Visual Communications and Image Processing*, (2008), p. 68221J-1–68221J-15. (SPIE Vol. 6822).
- [160] Chuan-Kai Lin — *Demosaic* — <https://sites.google.com/site/chklin/demosaic/> (2020).
- [161] Chuan-Kai Lin — *History of the Portable Network Graphics Format* — <http://www.libpng.org/pub/png/book/chapter07.html> (2020).
- [162] Michel Loève — *Probability Theory* — D. van Nostrand (1955).
- [163] Philippe Longère, Xuemei Zhang, Peter B. Delahunt, David H. Brainard — *Perceptual Assessment of Demosaicing Algorithm Performance* — Proc. of the IEEE, vol. 90, n° 1 (janvier 2002), p. 132–132.

- [164] Ralph Lorenz — *NASA/ESA/ASI Cassini-Huygens : 1997 Onwards (Cassini Orbiter, Huygens Probe and Future Exploration Concepts)* — Haynes Publishing (2017).
- [165] Olivier Lossen, Ludovic Macaire, Yanquin Yang — *Comparison of Color Demosaicing Methods* — dans *Advances in Imaging and Electronic Physics*, (2010), p. 173–265.
- [166] A. Lukin, D. Kubasov — *High-Quality Algorithm for Bayer Pattern Interpolation* — Programming and Computer Software, vol. 30, n° 6 (2004), p. 347–358.
- [167] Alexey Lukin, Denis Kubasov — *An Improved Demosaicing Algorithm* — dans *Proc. 14th Int. Conf. on Computer Graphics (GRAPHICON)*, (2004), p. 38–45.
- [168] Ming Ronnier Luo — *New Colour-Difference Formulae for Surface Colours* — Thèse de doctorat, School of Studies in Colour Chemistry and Colour Technology, University of Bradford (1986).
- [169] David L. MacAdam — *Projective Transformations of I.C.I. Color Specifications* — J. of the Optical Society of America, vol. 27, n° 8 (août 1937), p. 294–299.
- [170] David L MacAdam — *Visual Sensitivities to Color Differences in Daylight* — J. of the Optical Society of America, vol. 32, n° 5 (1942), p. 247–274.
- [171] David L MacAdam — *Specification of Small Chromaticity Differences* — J. of the Optical Society of America, vol. 33, n° 1 (janvier 1943), p. 18–26.
- [172] David L. MacAdam — *On the Geometry of Color Space* — J. of the Franklin Institute, vol. 238, n° 3 (septembre 1944), p. 195–210.
- [173] David L MacAdam — *Analytical Approximations for Color Metric Coefficients* — J. of the Optical Society of America, vol. 47, n° 4 (1957), p. 268–274.
- [174] David L MacAdam — *Analytical Approximations for Color Metric Coefficients II : Friele Approximations* — J. of the Optical Society of America, vol. 54, n° 2 (1964), p. 249–256.
- [175] David L MacAdam — *Analytical Approximations for Color Metric Coefficients III : Optimization of Parameters in Friele's Formulas* — J. of the Optical Society of America, vol. 54, n° 9 (septembre 1964), p. 1161–1165.
- [176] David L. MacAdam — *Color measurement : Theme and variations* — Springer-Verlag (1981).
- [177] Henrique Malvar, Li-Wei He, Ross Cutler — *High-Quality Linear Interpolation for Demosaicing of Bayer-Patterned Color Images* — dans *Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, (2004), p. III-485–III-488.
- [178] Henrique S. Malvar, Gary J. Sullivan — *$YC_oC_g - R$: A color Space with RGB Reversibility and Low Dynamic Range* — Rapport technique n° JVT-I014r3, Joint Video Team ISO/IEC (2003).
- [179] Henrique S. Malvar, Gary J. Sullivan — *Lifting-Based Reversible Color Transformations for Image Compression* — dans *Proc. SPIE XXXI (vol. 7073) Application of Digital Image Processing*, SPIE (2008), p. 707307-1–707307-10.
- [180] Pavlos Mavridis, Georgios Papaiouannou — *The Compact YC_oC_g Frame Buffer* — J. of Computer Graphics Techniques, vol. 1, n° 1 (2012), p. 19–35.
- [181] Joshua R. Maxwell — *Assessing Color Discrimination* — Mémoire de maîtrise, Dept. of Human Factors & Systems, Embry-Riddle Aeronautical University, Daytona Beach (2012).
- [182] Donald McIntyre — *Colour Blindness : Causes and Effects* — Dalton Publishing (2002).
- [183] K. McLaren — *XII : The SDC Recommended Colour-Difference Formula : Change to CIÉLAB* — J. of the Society of Dyers and Colourists, vol. 92, n° 9 (1976), p. 337–338.

- [184] K. McLaren — *XIII : The Development of the CIÉ 1976 ($L^*a^*b^*$) Uniform Colour Space and Colour-Difference Formula* — J. of the Society of Dyers and Colourists, vol. 92, n° 9 (1976), p. 338–341.
- [185] Daniele Menon, Stefano Adriani, Giancarlo Calvagno — *Demosaicing with Directional Filtering and a posteriori Decision* — IEEE Trans. on Image Processing, vol. 16, n° 1 (janvier 2017), p. 132–141.
- [186] Daniele Menon, Giancarlo Calvagno — *A Regularized Approach to Demosaicing* — dans *Procs. Visual Communications and Image Processing*, (2008), p. 68221L-1–68221L-11. (SPIE Vol. 6822).
- [187] Albert Henry Munsell — *A Pigment Color System and Notation* — The American Journal of Psychology, vol. 23, n° 2 (avril 1912), p. 256–244.
- [188] Albert Henry Munsell — *Atlas of the Munsell Color System* — Munsell Color Company (1915).
- [189] Albert Henry Munsell — *A Color Notation* — Munsell Color Company (1919).
- [190] D. Darian Muresan, Steve Luke, Thomas W. Parks — *Reconstructing of Color Images from CCD Arrays* — dans *TIDSP Fest*, (2000).
- [191] D. Darian Muresan, Thomas W. Parks — *Optimal Recovery Demosaicing* — dans *IASTED Signal And Image Processing*, (2002).
- [192] D. Darian Muresan, Thomas Porter — *Demosaaicing Using Optimal Recovery* — IEEE Trans. on Image Processing, vol. 14, n° 2 (février 2005), p. 267–278.
- [193] James D. Murray, William van Ryper — *Encyclopedia of Graphics File Formats* — 2^{de} éd., O'Reilly (1996).
- [194] Junichi Nakamura — *Image Sensors and Signal Processing for Digital Still Camera* — Taylor & Francis (2006).
- [195] Sydney M. Newhall, Dorothy Nickerson, Deane B. Judd — *Final Report of the O.S.A. Subcommittee on the Spacing of the Munsell Colors* — J. of the Optical Society of America, vol. 33, n° 7 (juillet 1943), p. 385–418.
- [196] Dev R. Newlin, Elwin Chandra Monie — *Edge Sensing Demosaicing Using Adaptive Weighted Interpolation* — American Journal of Applied Sciences, vol. 10, n° 4 (2013), p. 418–425.
- [197] Isaac Newton — *Opticks : Or, a Treatise of the Reflexions, Refractions, Inflexions and Colours of Light; also Two Treatises of the Species and Magnitude of Curvilinear Figures* — Smith & Walford, London (1704).
- [198] Nhat Xuan Nguyen — *Numerical Algorithms for Image Superresolution* — Thèse de doctorat, Dept. Scientific Computing and Computational Mathematics, Stanford University (2000).
- [199] Nhat Xuan Nguyen, Peyman Milanfar, Gene Golub — *A Computationally Efficient Superresolution Image Reconstruction Algorithm* — IEEE Trans. on Image Processing, vol. 10, n° 4 (avril 2001), p. 573–583.
- [200] Dorothy Nickerson — *History of the Munsell Color System, Company, and Foundation, I* — Color Research and Application, vol. 1, n° 1 (1976), p. 7–10.
- [201] Dorothy Nickerson — *History of the Munsell Color System, Company, and Foundation, II* — Color Research and Application, vol. 1, n° 2 (1976), p. 69–77.
- [202] Dorothy Nickerson — *History of the Munsell Color System, Company, and Foundation, III* — Color Research and Application, vol. 1, n° 3 (1976), p. 121–130.
- [203] Yu-Ichi Ohta — *A Region-Oriented Image-Analysis System by Computer* — Thèse de doctorat, Department of Information Science, Kyoto University (mars 1980).

- [204] Yu-Ichi Ohta, Takeo Kanada, Toshiyuki Sakai — *Color Information for Region Segmentation* — Computer Graphics and Image Processing, vol. 13, n° 3 (juillet 1980), p. 222–241.
- [205] Wilhelm Ostwald — *Die Farbenlehre, Vol 1 : Mathematische Farbenlehre* — Verlag Unesma (1918).
- [206] Wilhelm Ostwald — *Einführung in die Farbenlehre* — 3^e éd., Verlag von Philipp Reclam (1919).
- [207] Wilhelm Ostwald — *Die Farbenfibel* — 5^e éd., Verlag Unesma (1920).
- [208] David A. Patterson, John L. Hennessy — *Computer Organisation and Design : The Hardware/Software Interface* — 5^e éd., Morgan Kaufmann (2014).
- [209] Karl Pearson — *On Lines and Planes of Closest Fit to Systems of Points in Space* — Philosophical Magazine, vol. 2, n° 6 (1901), p. 559–572.
- [210] Robert Peck, Susan Deyl, Jay Miner — *Amiga Hardware Manual* — Commodore Amiga Inc (1985).
- [211] Katherine Peel — *The Concise Atari ST 68000 Programmer's Reference Guide* — Glentop Press (1986).
- [212] Ibrahim Pekkacuksen, Yucel Altunbasak — *Gradient-Based Threshold-Free Color Filter Array Interpolation* — dans *IEEE Int. Conf. on Image Processing (ICIP)*, (2019), p. 137–140.
- [213] William b. Pennebaker, Joan L. Mitchell — *JPEG : Still Image Data Compression Standard* — van Nostrand Reinhold (1992).
- [214] Walter Petryshyn — *Psychological Hue Relationship of the Munsell Color System* — Mémoire de maîtrise, Department of Psychology, University of Windsor, (1967).
- [215] Steven Pigeon — *Contributions à la compression de données* — Thèse de doctorat, Département d'informatique et de recherche opérationnelle, Faculté des Arts et des Sciences, Université de Montréal (décembre 2001).
- [216] Thomas Porter, Tom Duff — *Compositing Digital Images* — Computer Graphics, vol. 18, n° 3 (juillet 1984), p. 253–259.
- [217] Charles Poynton — *A Guided Tour of Color Space* — dans *Proc. SMPTE Advanced Television and Electronic Imaging Conference*, (1990), p. 167–180.
- [218] Charles Poynton — *FAQ about Color* — Autopublié? (1997).
- [219] Charles Poynton — *Digital Video and HDTV: Algorithms and Interfaces* — Morgan Kaufmann (2006).
- [220] Irwin G. Priest, Ferdinand G. Brickwedde — *Minimum Perceptible Colorimetric Purity as a Function of Dominant Wave-Length* — Rapport technique n° RP1009, National Bureau of Standards (mai 1938) Aussi dans *Journal of the National Bureau of Standards*, vol. 20.
- [221] Majid Rabbani, Craig M. Smith — *US Patent 5,412,427: Electronic Camera Utilizing Image Compression Feedback for Improved Color Processing* — US Patent Office (mai 1995).
- [222] A. Rajeev Ramanath, B. Wesley Snyder, C. David Hinks — *Image Comparison Measure for Digital Still Color Cameras* — dans *Proc. Int. Conf. on Image Processing (ICIP)*, (2002), p. I-629–I-632.
- [223] Rajeev A. Ramanath, Wesley E. Snyder — *Color Restoration in Digital Color Cameras* — dans *SPIE OPTO-SouthEast*, (2000).
- [224] Rajeev A. Ramanath, Wesley E. Snyder — *Adaptive Demosaicking* — Journal of Electronic Imaging, vol. 12, n° 4 (octobre 2003), p. 633–642.
- [225] Rajeev A. Ramanath, Wesley E. Snyder, Griff L. Bilbro, Willam A. Sander III — *Demosaicing Methods for Bayer Color Arrays* — Journal of Electronic Imaging, vol. 11, n° 3 (juillet 2002), p. 306–315.
- [226] Rajeev A. Ramanath, Wesley E. Snyder, Youngjun Yao, Mark S. Drew — *Color Imaging Processing Pipeline* — IEEE Signal Processing Magazine, vol. 22, n° 1 (janvier 2005), p. 34–43.

- [227] Travis A. Rector, Zoltan G. Levay, Lisa M. Frattare, Jayanne English, Kirk Pu'uohau-Pummil — *Image-Processing Techniques for the Creation of Presentation-Quality Astronomical images* — The Astronomical Journal, vol. 133, n° 2 (février 2007), p. 598–611.
- [228] D. Renshaw, P. B. Denyer, G. Wang, M. Lu — *ASIC Image Sensors* — dans *Procs. IEEE Int. Symposium on Circuits and Systems*, (mai 1990), p. 3038–3041.
- [229] Rich Milliron — *Turboprop Rolling Shutter* — https://commons.wikimedia.org/wiki/File:Turboprop_Rolling_Shutter.jpg (2011).
- [230] A. R. Robertson, *et al.* (éds) — *Colorimetry Part 4 : CIÉ 1976 L*a*b* Colour Space* — Rapport technique n° DS 014-4.3/E :2007, CIÉ Central Bureau (2007).
- [231] Roderick T. Ryan (éd.) — *Principles of Color Sensitometry* — 3^e éd., Society of Motion Picture and Television Engineers (1974).
- [232] János Schanda — *Colorimetry : Understanding the CIE System* — John Wiley & Sons (2007).
- [233] Gal Shabtay, Noy Cohen, Nadav Gera, Oded Gigunshinski — *US Patent 10,084,953 B1 : Thin Multi-Aperture Imaging System with Auto-Focus and Method for Using Same* — US Patent Office (2018).
- [234] Gaurav Sharma, Wencheng Wu, Edul N. Nadal — *THE CIÉDE2000 Color-Difference Formula : Implementation Notes, Supplementary Test Data, and Mathematical Observations* — Color Research and Application, vol. 30, n° 1 (février 2005), p. 21–30.
- [235] Edwin K. Shenk — *US Patent 4,199,244 : Automatic Focusinig Camera* — US Patent Office (1982).
- [236] Edwin K. Shenk — *US Patent 4,309,098 : Autofocus Cine Camera Having Automatic Variable Speed Focusing* — US Patent Office (1982).
- [237] Paul M. Simon — *Single-Shot High Dynamic Range and Multi-Spectral Imaging Based on Properties of Color Filter Arrays* — Mémoire de maîtrise, School of Engineering, Dayton University, (2011).
- [238] Simon Singh — *The Code Book : The Science of Secrecy from Ancient Egypt to Quantum Cryptography* — Anchor Books (2000).
- [239] Athanassios Skodras, Charilaos Christopoulos, Touradj Ebrahimi — *The JPEG 2000 Still Image Compression Standard* — IEEE Signal Processing Magazine, vol. 18, n° 5 (septembre 2001), p. 36–58.
- [240] Alvy Ray Smith — *Color Transform Pairs* — SIGGRAPH Computer Graphics, vol. 12, n° 3 (1978), p. 12–19.
- [241] T. Smith, J. Guild — *The CIÉ Colorimetric Standards and Their Use* — Trans. of the Optical Society, vol. XXXIII, n° 3 (1931–1932), p. 73–134.
- [242] Hannah E. Smithson, Greti Dinkova-Bruun, Giles E. M. Gasper, Michael Huxtable, Tom C. b. McLeish, Cecilia Panti — *A Three Dimensional Color Space from the 13th Century* — J. of the Optical Society of America A, vol. 29, n° 2 (2012), p. A356–A352.
- [243] Amelia Carolina Sparavigna — *Robert Grosseteste and the Colours* — Int. Journal of Science, vol. 3, n° 1 (janvier 2014), p. 1–6.
- [244] James A. Storer, Thomas G. Szymanski — *Data Compression via Textual Substitution* — Journal of the ACM, vol. 29, n° 4 (1982), p. 928–951.
- [245] Tilo Strutz — *Multiplierless reversible Color Transforms and Their Automatic Selection for Image Data Compression* — IEEE Trans. Circuits and Systems for Video Technologies, vol. 23, n° 7 (juillet 2013), p. 1249–1259.

- [246] Tilo Strutz, Alexander Leipnitz — *Reversible Colour Spaces without Increased Bit Depth and Their Adaptive Selection* — IEEE Signal Processing Letters, vol. 22, n° 9 (septembre 2015), p. 1269–1273.
- [247] Dan Su, Philip Willis — *Demosaicing of Color Images Using Pixel-Level Data-Dependent Triangulation* — dans *Procs. Theory and Practice of Computer Graphics*, (2003), p. 16–23.
- [248] Shijun Sun — *Residual Color Transform using YC_oC_g* — Rapport technique n° JVT-L014, Joint Video Team ISO/IEC (2004).
- [249] Yoshinori Takizawa, Hiroaki Kotaki, Kikuo Saito, Tadashi Sigiki, Yasuo Takemura — *Field Integration Mode CCD Color Television Camera Using a Frequency Interleaving Method* — IEEE Trans. on Consumer Electronics, vol. CE-29, n° 3 (août 1983), p. 358–364.
- [250] William Henry Fox Talbot — *US Patent 5,171 : Improvement in Photographic Pictures* — US Patent Office (1847).
- [251] Marshall F. Tappen, Bryan C. Russell, William T. Freeman — *Exploiting the Sparse Derivative Prior for Super-Resolution and Image Demosaicing* — dans *3rd Int. Workshop on Statistical and Computational Theories of Vision*, (2003).
- [252] David Taubman, Michael Marcellin — *JPEG 2000 Image Compression : Fundamentals, Standard and Practice* — Springer (2012).
- [253] Ping-Sing Tsai, Tinku Acharya, Ajay K. Ray — *Adaptive Fuzzy Color Interpolation* — Journal of Electronic Imaging, vol. 11, n° 3 (juillet 2003), p. 293–305.
- [254] Yushing T. Tsai, Scott J. Daly, Majid Rabbani — *US Patent 5,172,227 : Image Compression with Color Interpolation for a Single Sensor Image System* — US Patent Office (décembre 1992).
- [255] Kingdon S. Tyler — *Telecasting and Color* — Harcourt, Brace & Co. (1946).
- [256] E. P. T. Tyndall — *Chromaticity Sensibility to Wave-Length Differences as Function of Purity* — J. of the Optical Society of America, vol. 23, n° 1 (janvier 1933), p. 15–24.
- [257] Oli Usher, Lars Lindberg Christensen — *The Universe Through the Eyes of Hubble* — ESO & ESA/Hubble (2014).
- [258] Georges Valensi — *Brevet 841-335 : Procédé de télévision en couleurs* — Direction de la propriété intellectuelle (1939).
- [259] Georges Valensi — *US Patent 2,375,966 : System of television in Color* — US Patent Office (mai 1945).
- [260] Patrick Vandewalle, Sabine Süsstrunk, Martin Vetterli — *A Frequency-Domain Approach to Super-Resolution Imaging from Shared Low-Resolution Images* — EURASIP Journal on Advances in Signal Processing, (2006), p. 1–14.
- [261] Miguel Vega, Rafael Molina, Aggelos K Katsaggelos — *A Bayesian Superresolution Approach to Demosaicing of Blurred Images* — EURASIP Journal on Advances in Signal Processing, vol. 2006 (2006), p. 1–12. (Article ID 25072).
- [262] Heinz-W. von Bülow, Dirk Paulissen — *Le grand Livre du Kodak PhotoCD* — Éds MicroApplication (1994).
- [263] Bruce A. Wallace — *Merging and Transformation of Raster Images for Cartoon Animation* — Computer Graphics, vol. 15, n° 3 (août 1981), p. 253–262.
- [264] Terry A. Welch — *A Technique for High-Performance Data Compression* — Computer, vol. 17, n° 6 (1984), p. 8–19.

- [265] Terry A. Welch — *US Patent 4,558,302 : High Speed Data Compression and Decompression Apparatus and Method* — US Patent Office (décembre 1985).
- [266] Andrej Werner (éd.) — *JPEG 9 Lossless Coding* — Rapport technique n° JTC 1/SC 29WG 1, ISO/IEC (2012).
- [267] Steve Wilhite — *Graphics Interchange Format, Version 89a* — Rapport technique, Compuserve Inc. (1989).
- [268] W. D. Wright — *A Re-Determination of the Trichromatic Coefficients of the Spectral Colors* — Trans. of the Optical Society, vol. XXX, n° 4 (1928–1929), p. 73–134.
- [269] W. D. Wright, F. H. G. Pitt — *Hue-Discrimination in Normal Colour-Vision* — Procs. of the Physical Society, vol. 46, n° 3 (1934), p. 459–473.
- [270] Xiaolin Wu, Wai Kin Choi, Paul Bao — *Color Restoration from Digital Camera Data By Pattern Matching* — dans *Color Imaging : Device-Independent Color, Color Hard Copy, and Graphics Arts II*, (1997), p. 12–17. (SPIE Vol. 3018).
- [271] Günter Wyszecki, G. H. Fielder — *New Color-Matching Ellipses* — J. of the Optical Society of America, vol. 61, n° 9 (septembre 1971), p. 1135–1152.
- [272] Xingbo — *A 3 CCD Imaging Block* — https://en.wikipedia.org/wiki/File:A_3CCD_imaging-block.jpg (2009).
- [273] Fan Zhang, Xiaolin Wu, Xiaokang Yang, Wenjun Zhang — *Improved Color Demosaicking in Weak Spectral Correlation* — dans *Procs. Visual Communications and Image Processing*, (2008), p. 68221N-1–68221N-10. (SPIE Vol. 6822).
- [274] J. Ziv, A. Lempel — *Compression of Individual Sequences via Variable-Rate Coding* — IEEE Trans. Inf. Theory, vol. 24, n° 5 (September 1978), p. 530–536.
- [275] Assef Zomet, Shmuel Peleg — *Multi-Sensor Super-Resolution* — dans *Procs. 6th IEEE Workshop on Applications of Computer Vision*, (2002), p. 27–31.
- [276] V. K. Zworykin, G. A. Morton — *Television : The Electronics of Image Transmission* — John Wiley & Sons (1940).

Index

Les noms propres, les noms des algorithmes, et les concepts se retrouvent ici dans le même index. Les titres de livres apparaissent en italiques. Notez que les entrées sont indexées relativement au début des paragraphes qui abordent le sujet en question.

- Additives** ~ Couleurs, 77
- Analyse** ~ En composantes principales, 101
- Approximation** ~ \tan^{-1} , 86
- Arc-en-ciel**, 100
- * * *
- Blanc C**, 61
- BT.2020**, 71–73 ~ Couleurs primaires, 71
- BT.2100**, 73–76
- BT.470**, 61 ~ Couleurs primaires, 61
- BT.709** ~ Couleurs primaires, 66
- * * *
- Cercle chromatique**, 83, 84, 100
- CFR**, 65
- CIÉ**, 37 ~ Fonctions de réponse, 37 ~ *Matching functions*, 37 ~ Propriétés désirables, 41 ~ Triangle des couleurs, 38–40 ~ xyz, 41
- Compression** ~ Et espaces de couleurs, 90, 100 ~ Sans perte, 100
- Constante** ~ Magique, 63, 67
- Couleurs**, *voir aussi* Espace de couleurs ~ Additives, 77 ~ Cercle chromatique, 83, 84 ~ Complémentaires, 78 ~ Et compression, 90, 100 ~ Différence au gris, 87 ~ Homochromes, 34 ~ Luminance, 83 ~ Métamétrisme, 34 ~ Et Newton, 100 ~ Perception, 101 ~ Pondérations sur les, 46 ~ Primaires, 101 ~ Primaires optimales, 101 ~ Primaires, BT.2020, 71 ~ BT.2100, 73 ~ Primaires, BT.470, 61 ~ Primaires, BT.709, 66 ~ Pureté, 83 ~ Saturation, 82, 83 ~ Sensibilité aux, 33 ~ Soustractive, 77 ~ Teinte, 82, 83 ~ Théorie de la vision, 101 ~ Théories de la, 100 ~ Triangle des, 35 ~ Valeur, 83, 84
- CRTC**, 65
- * * *
- D**₅₀, 66
- D**₆₅, 66, 71, 93
- Diagramme ternaire**, *voir* Triangle des couleurs
- * * *
- Eigenvector**, 101
- Espaces de couleurs** ~ CIÉ 1931, 37–42 ~ CMY, 77–82 ~ CMYK, 77–82 ~ Conversion ~ XYZ à RGB, 42 ~ Couleurs primaires, 33–36 ~ D_{50} , 66 ~ D_{65} , 66, 71, 93 ~ Familles, 36–37 ~ Kodak Lossless, 45–46 ~ Kodak YCC, 46–48 ~ Kodak-1, 43–44 ~ Linéaires, 36, 42–82 ~ NCS, 103 ~ Non linéaires, 36, 82–99 ~ CIÉDE2000, 95 ~ HSL, 88–90 ~ HSL conique, 89–90 ~ HSV, 83–87 ~ $L^*a^*b^*$, 92–94, 99 ~ $L^*u^*v^*$, 94, 99 ~ Munsell, 95–96 ~ NCS, 98–99 ~ Ostwald, 96–98, 103 ~ Perceptuellement uniformes, 91 ~ Ohta, 48–50, 101 ~ Série S, 54–61 ~ sRGB, 66–67 ~ De la télévision, 61–77 ~ BT.2020, 71–73 ~ BT.2100, 73 ~ BT.601, 65–67 ~ BT.709, 65–67 ~ Famille BT, 76–77 ~ *IC_TC_P*, 74–76 ~ *LMS*, 74–76 ~ *Transmission primaries*, 61–62 ~ *YC_bC_r*, 67–69 ~ *YC_bC_r* DjVu, 68–69 ~ *YC_b'C_r'*, 70–71 ~ HDTV, 70–71 ~ *YC_b'C_r'*, 70–71 ~ *YD_bD_r*, 70 ~ *YP_bP_r*, 69–70 ~ YUV et YIQ, 63–65 ~ Xerox YES, 48 ~ XYZ, 42 ~ *YC_oC_g*, 50–53 ~ *YC_oC_g* réversible, 51–53 ~ *YVU-R*, 53–54
- * * *
- FCC**, 65
- * * *
- Gamme** ~ Grande gamme dynamique, 73
- * * *
- HDR**, 73
- High Dynamic Range**, 73
- Homochromes**, 34
- * * *
- IEC/4WD 61966-2-1**, 66
- * * *
- JPEG**, 53, 102
- JPEG 2000**, 53, 102
- * * *
- Luminance**, 84 ~ Définition, 83
- * * *
- Magique** ~ Constante, 63, 67

- Matrice** ~ De rotation 3D, 55 ~ Normalisation, 60 ~ De permutation, 63 ~ Rotation, orthonormale, 57
Matriochka, 82
Métamérisme, 34
Métrique, 93, 95
Munsell ~ Espace de couleurs de, 95–96
* * *
- NCS** ~ Espace de couleurs, 98–99
Newton ~ Et couleurs, 100
Normalisation ~ D'un espace de couleurs, 60
* * *
- Ostwald** ~ Espace de couleurs d', 96–98, 103
* * *
- Perception** ~ Des couleurs, 101
Pureté, 83
* * *
- Rotation** ~ Matrice orthonormale, 57 ~ Matrice, 3D, 55
* * *
- Saturation**, 83
saturation ~ Définition, 82
Soustractive ~ Couleurs, 77
* * *
- tan⁻¹**, 86
Teinte, 83 ~ Définition, 82
Télévision ~ Histoire de la, 61, 100
Théorie ~ De la vision, 101
Transformée ~ Hotelling, 101 ~ Karhunen-Loève, 101
Transformées ~ Linéaires de couleurs, 36
Triangle ~ Des Couleurs, 35 ~ Des Couleurs CIÉ, 38–40
Triche, 86
* * *
- Valeur** ~ D'une couleur, 83 ~ D'une couleur, 84
Vecteur ~ Propre, 101
Vision ~ Théorie de la, 101

Traitement numérique des images

Notes de cours (C++)

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam consequat vel sapien in sodales. Mauris tincidunt pulvinar enim ut dictum. Sed lectus massa, maximus a est non, gravida tincidunt mi. Non, ce n'est pas un message secret. Oui, c'est exprès. Mauris molestie fringilla nunc vitae tincidunt. Nam imperdunt mauris volutpat, consectetur orci nec, rhoncus orci. **Lorem ipsum dolor** sit amet, consectetur adipiscing elit. Vestibulum vestibulum vel tellus vel lobortis. Suspendisse fermentum malesuada purus malesuada posuere. Nullam aliquet cursus ex, et lacinia ante lobortis quis. Sed vel posuere leo.

