

LE GAMMA ET LES ESPACES COLORIMÉTRIQUES

SOMMAIRE

Qu'est ce que le gamma ?	Page 3
Les appareils photos et les caméras	page 3
L'oeil humain	page 4
Les écrans	page 5
La correction de gamma	page 6 - 7
Le linear workflow	page 8 - 10
 Les couleurs	 Page 11
Généralités sur la lumière	page 11
Pourquoi voyons nous comme nous voyons ?	page 12
Les espaces couleurs	page 13 - 16
Le calibrage	page 17 - 18
 Annexes	 Page 19

Qu'est ce que le gamma ?

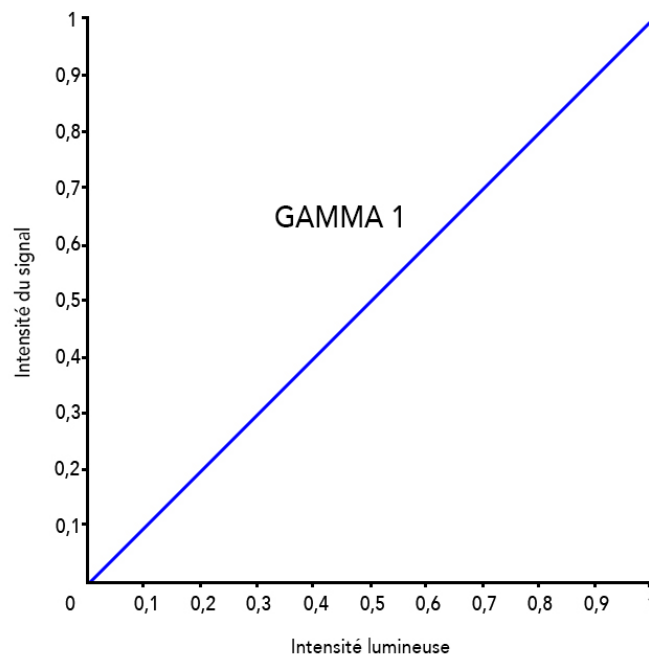
Le **gamma**, c'est une courbe mathématique (une fonction), qui permet de connaître le lien entre un signal lumineux émis (ampoule, lumière du jour, écran, signal électrique) et la réponse d'un capteur (l'oeil par exemple).

On peut écrire cette courbe comme ceci :

Signal de sortie = Signal d'entrée puissance gamma

Les appareils photos et les caméras

Quand un nombre de photons atteint le capteur d'un appareil photo, celui-ci détecte exactement ce nombre de photons, ni plus ni moins. C'est une **réponse linéaire à la lumière**. Ci-dessous le gamma d'un appareil photo.



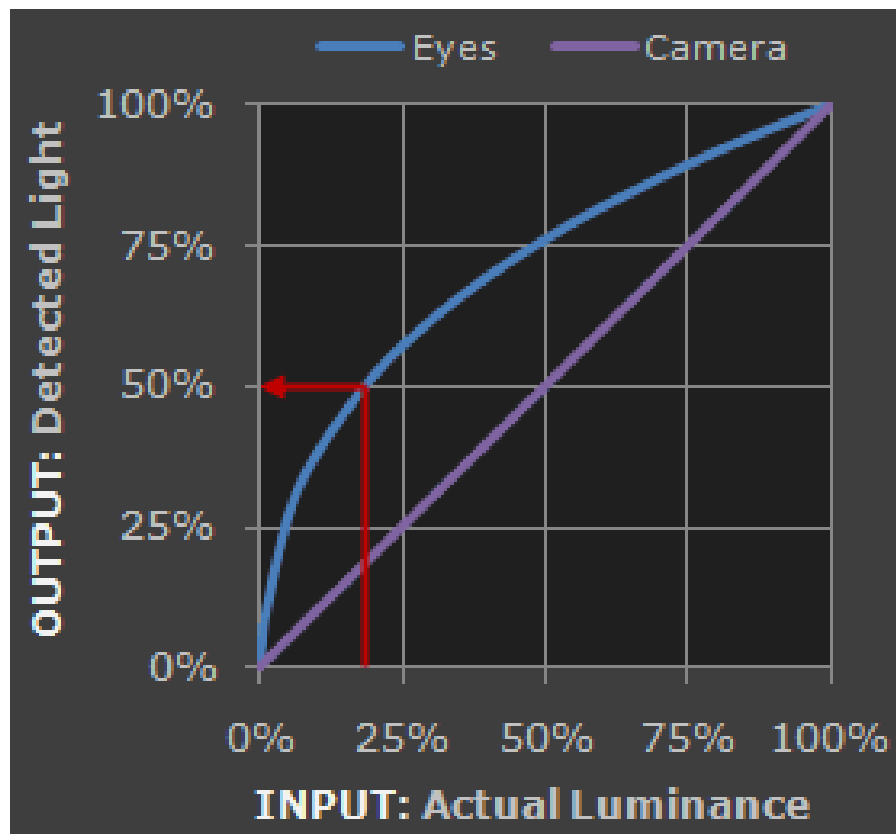
L'oeil humain

Contrairement aux appareils photo, l'oeil a une réponse **non linéaire à la lumière**. Nous percevons la lumière comme étant un peu plus lumineuse, ce qui nous permet d'avoir une plus grande plage dynamique.

Cette particularité varie selon notre environnement. En effet, l'oeil est beaucoup plus sensible aux changements dans les tons sombres que dans les hautes lumières.

Par exemple dans une pièce sombre, il suffit qu'on rajoute une bougie pour voir significativement plus de lumière, mais dans une pièce en plein jour, on ne verrait pas la différence, alors qu'on a apporté la même quantité de lumière.

Ci-dessous, la courbe de réponse de l'oeil (en bleu), comparée à la courbe de réponse d'une caméra. **Pour la même luminance, l'oeil perçoit plus de lumière que la caméra.**



Les écrans

Les écrans ont, tout comme l'oeil, une courbe de réponse **non linéaire à la lumière, mais inversée.**

Les écrans affichent les images plus sombres et contrastées que celles qu'on leur soumet en entrée. La raison est mécanique : la luminosité affichée par l'écran n'est pas proportionnelle à la tension électrique que reçoit cet écran.

Autrement dit, si 1 volt = 100% luminance (blanc pur) et 0 volt = 0% (noir pur), la logique serait 0,5 volt = 50% (gris moyen). Or, ce n'est pas le cas. Voyons pourquoi.

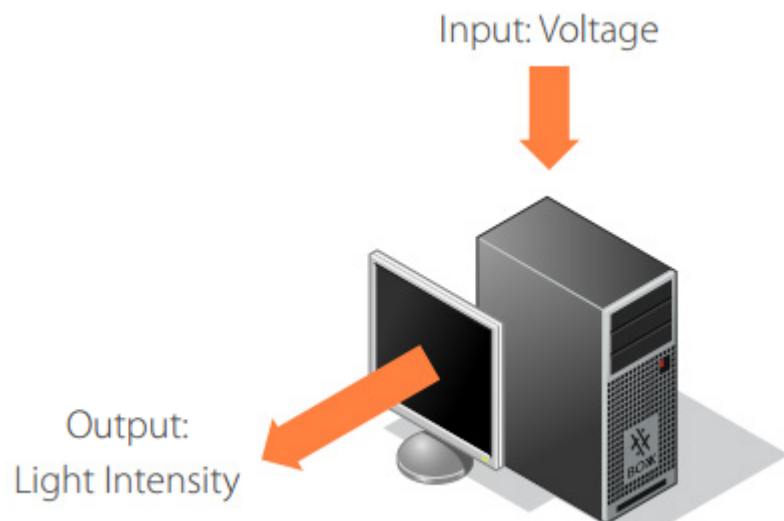
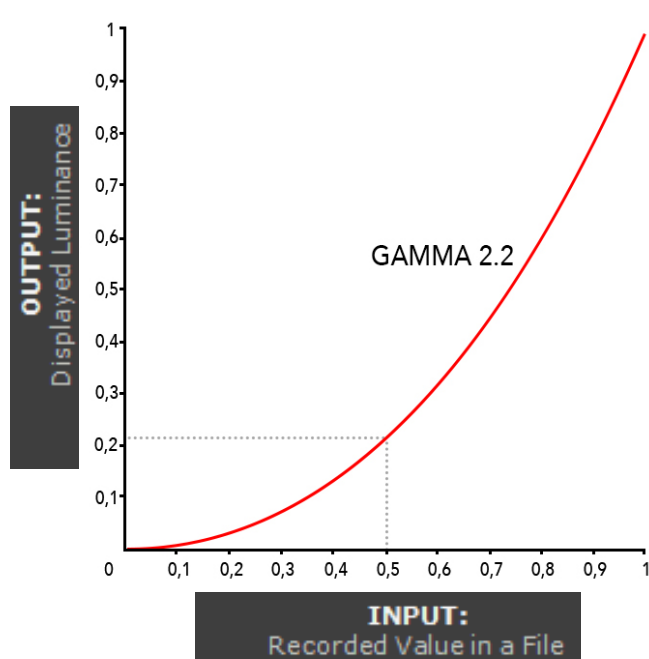
Voici la courbe de réponse d'un écran, également appelée **courbe gamma**. Il en existe plusieurs sortes. Ici c'est un **GAMMA 2.2, CE QUI CORRESPOND À LA PLUPART DES ÉCRANS.**

Sur cette courbe on peut voir qu'un gris moyen 0,5 est affiché par l'écran comme un gris 0,21 donc beaucoup plus sombre.

Petit rappel, le gamma c'est :

Signal de sortie = Signal d'entrée puissance gamma

$0,5^{2.2} = 0,21$. Le signal de sortie correspond bien à ce qui s'affiche sur le graphique.

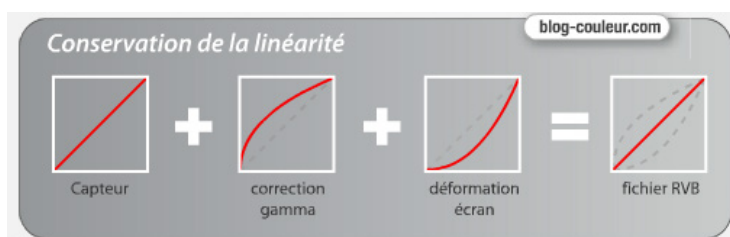
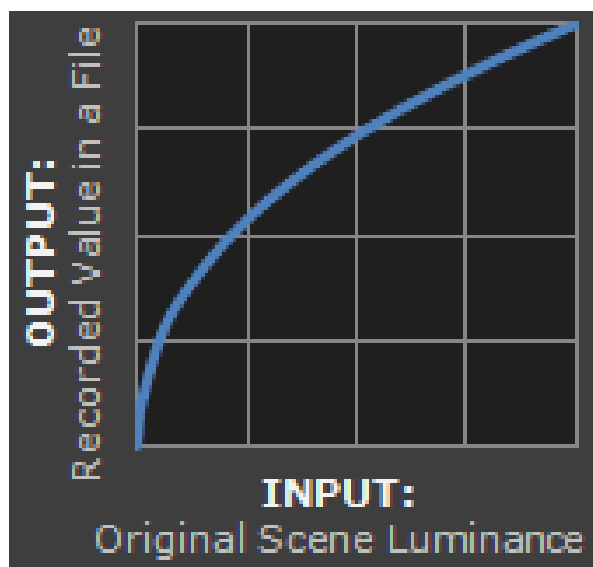


La correction de gamma

Pour afficher correctement une image sur un écran, on est obligé de lui appliquer une **correction de gamma**. Cette correction s'applique directement sur le fichier qui sort d'une caméra ou appareil photo. Un fichier raw (fichier brut) a un gamma de 1, c'est-à-dire linéaire, mais la conversion dans un espace sRGB (en jpeg par exemple) lui applique automatiquement cette courbe (en bleu).

En gros on «éclaircit» volontairement une image pour palier aux défauts des écrans, qui eux, assombrissent les images.

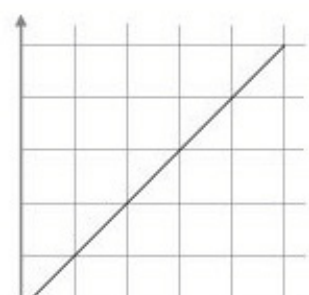
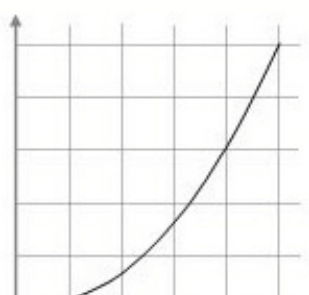
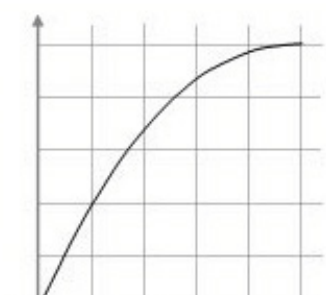
Une image standardisée est une image qui s'est vu appliquée un traitement automatique qui correspond à une courbe inverse de celle du gamma de l'écran. D'où le nom «correction de gamma».



Gamma de l'image
sortant de votre APN

X Gamma de votre écran =

Votre image



Que se passe-t-il lorsque l'on corrige le gamma ?

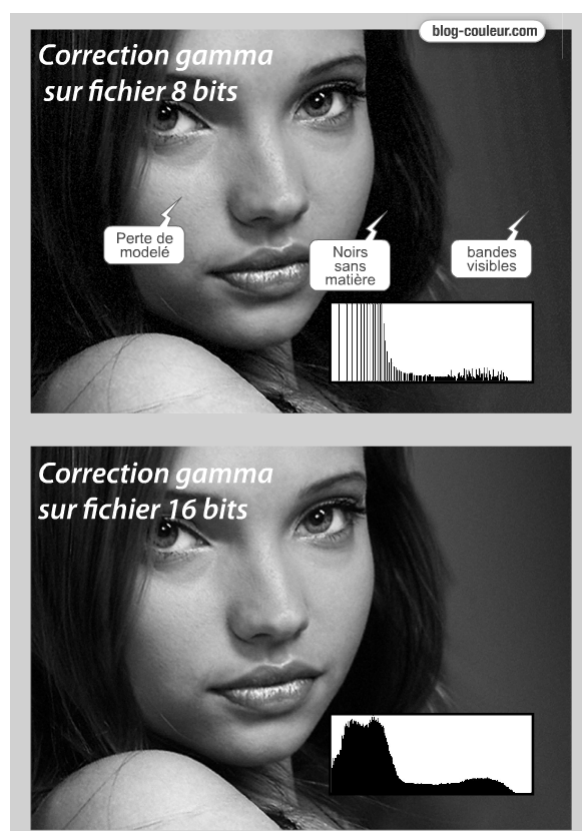
La correction de gamma est une redistribution tonale et en pratique elle est **destructrice**, parce qu'elle cherche à créer de la lumière qui n'existe pas. Plus on éclaircit une image, plus on perd de l'information.

Lorsque la valeur d'entrée avoisine le zéro (noir) la valeur de sortie a du mal à décoller, parce que la courbe démarre dans une direction verticale qui ne transmet pas d'information.

Pour atténuer cela, la courbe de correction gamma est souvent remplacée par une fonction logarithmique qui lui ressemble beaucoup, mais qui est moins pentue dans les basses lumières, ce qui permet de limiter les pertes d'informations. Chaque espace de couleur (voir suite du dossier), à sa propre courbe de transfert.

Par exemple, celle du sRGB est moins destructrice que celle de l'adobe 98 et certains logiciels comme lightroom ont aussi leur propre courbe de correction.

Cependant, même si l'on réduit la correction de gamma, les pertes restent importantes. **La solution est d'augmenter la profondeur de couleur** (voir suite du dossier).



<i>Pertes de données et profondeur de couleur</i>	
Quantification	Pertes
8 bits	8 %
10 bits	4 %
12 bits	2 %
14 bits	1 %
16 bits	0,5 %

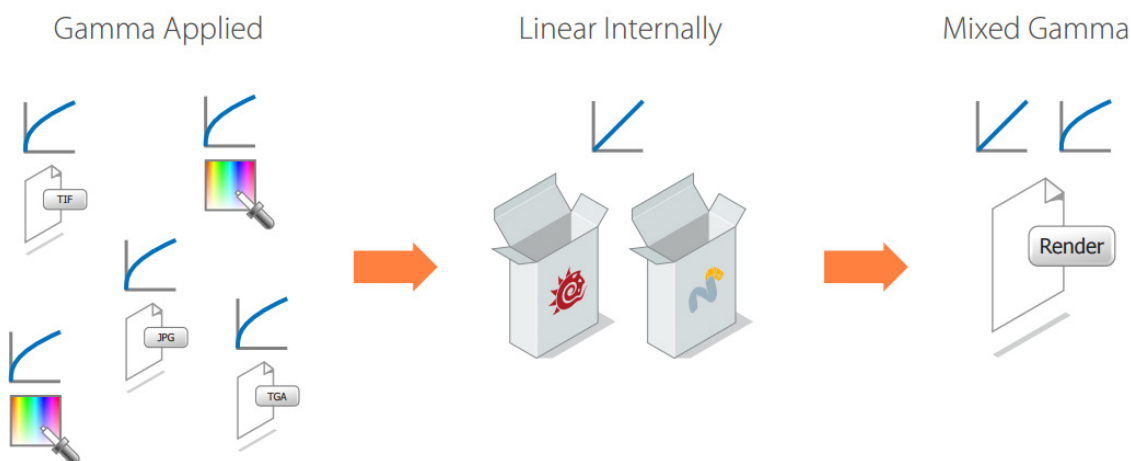
Travailler une image dans un profil de gamma 1,8 est moins destructeur que dans un profil 2,2. Mais la véritable solution, reste de travailler en 16bits ou plus.

Bon à savoir : la plupart des écrans LCD ont à peu près un gamma de 2,2, mais parfois il peut y avoir de petites déviations. Il est donc utile d'étalonner nos écrans pour être sûrs que le gamma soit bien de 2,2, ce qui correspond à l'exacte inverse de la correction de gamma appliquée sur nos images.

Le linear workflow

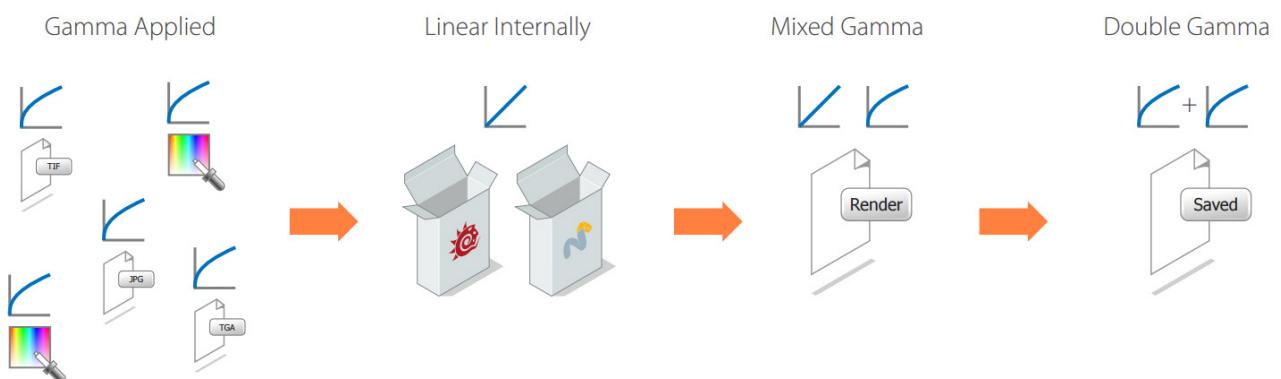
Le **Linear Workflow** est une méthode de travail qui consiste à **supprimer les compensations Gamma des images utilisées** (les textures principalement) et produites (les rendus) dans notre logiciel 3D.

Les logiciels 3D fonctionnent de manière linéaire en interne. Mais les textures et les sélecteurs de couleurs ont une correction de gamma pour être vu correctement à l'écran. Ca signifie que les rendus 3D ont en fait des éléments avec des gammas différents. Le lighting est en linéaire mais

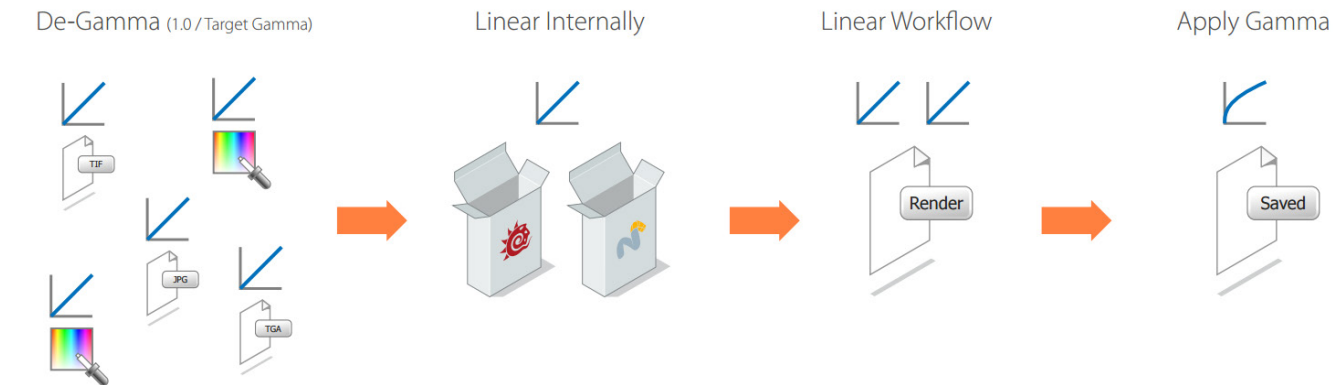


Tout comme un fichier raw, une image qui sort d'une carte graphique (un rendu) est linéaire. Mais dès lors que ce rendu est exporté du logiciel 3D, donc converti en fichier image (jpeg, png, tiff etc.), une courbe de gamma lui est appliquée pour que l'image puisse s'afficher correctement à l'écran.

Donc si on ne supprime pas les courbes de gammas des textures et des couleurs qu'on utilise dans nos scènes 3D, au final on va simplement doubler le gamma sur les textures qui en ont déjà et ça donnera un rendu trop lumineux, délavé.



L'idée, c'est donc de supprimer les corrections de gammas des textures qu'on utilise quand on les importe dans le logiciel. De cette façon toute la chaîne reste en linéaire, jusqu'à l'export final où l'image se voit appliquer une correction de gamma.



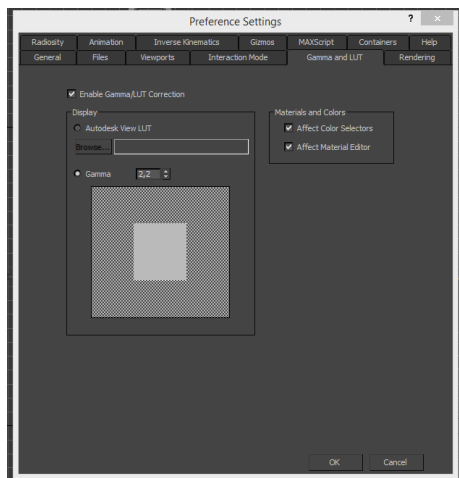
L'image de rendu à gauche est linéaire, l'écran l'affiche trop sombre. Quand elle est enregistrée (image du milieu), la correction de gamma lui est appliquée. La lumière est bonne, mais le parquet paraît délavé, car la texture de bois était déjà gamma-corrigée. A l'export, la correction gamma appliquée à l'image s'est ajoutée à la correction de la texture. A droite, un rendu issu d'un workflow linéaire : c'est le bon rendu.

Pour travailler en linear workflow, on doit donc dé-gamma corriger les textures qui sont importées dans le logiciel 3D.

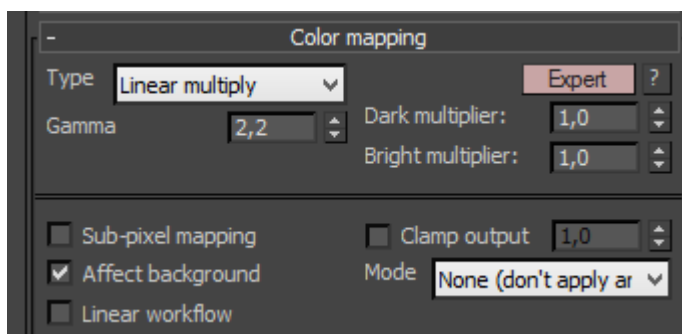
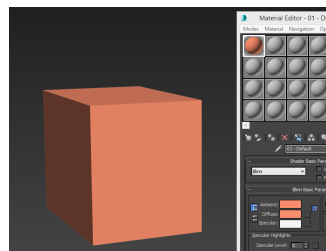
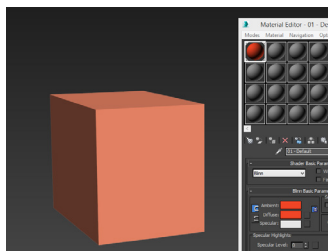
Pour cela on applique aux textures une correction de gamma de 0,454 (l'inverse de 2.2, donc $1/2.2$). Selon le logiciel et le moteur de rendu, ça se fait plus ou moins manuellement.

DANS 3DS MAX

Quand on importe une texture dans max, si l'image a une correction de gamma, max la linéarise en lui appliquant un gamma 2.2. Si c'est un fichier HDR, un fichier 32 bit, déjà en gamma 1, max ne fait rien. Si max n'a pas l'information, il se base sur la profondeur de bits. Pour les 8 et 16bits, il applique une correction.

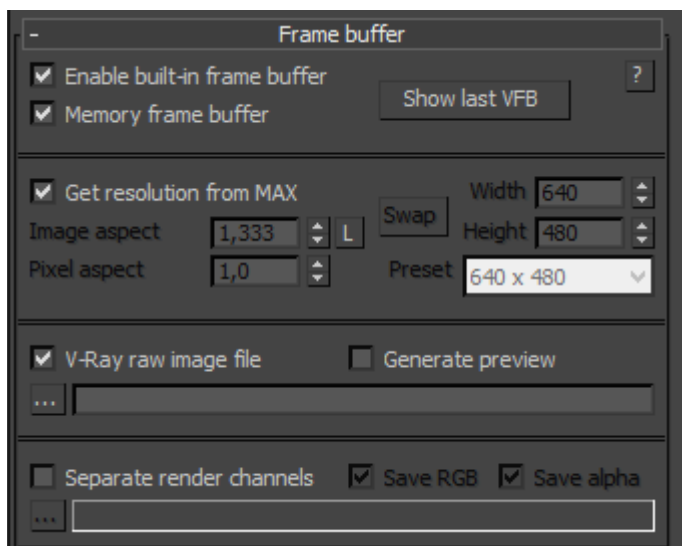


La fenêtre gamma and Lut ne concerne que le display, c'est à dire l'affichage dans le viewport. Ca n'a rien à voir avec les rendus. En interne le logiciel est toujours en linéaire, mais pour travailler confortablement, on utilise cette correction. Le mieux est d'également appliquer cette correction sur le material editor et les sélecteurs de couleurs, de façon à avoir une cohérence entre les matériaux et le viewport.

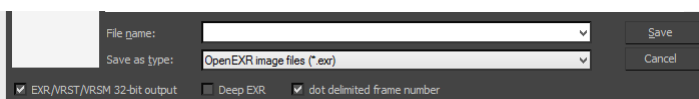


Dans les paramètres de vray, l'onglet color mapping va nous permettre de travailler en linéaire. Le menu déroulant *type* nous permet d'appliquer différents filtres au rendu, un peu comme les filtres qu'on met sur les objectifs d'appareils photos, pour compenser les trop fortes luminosités. En *linear multiply*, rien n'est compensé. On veut obtenir une image brute, pour avoir un meilleur contrôle en post production.

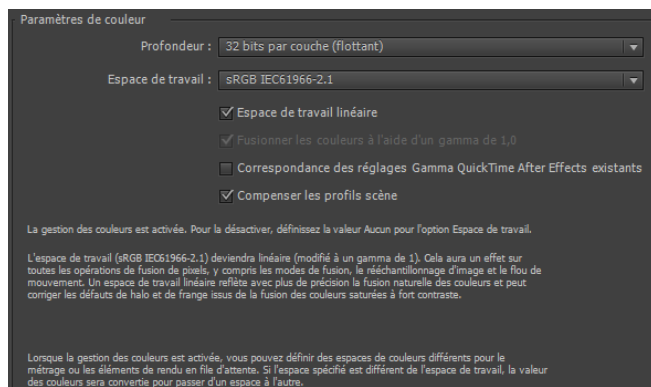
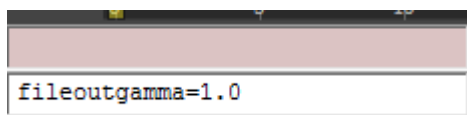
Dans gamma on laisse 2,2. Ca permet d'éviter les artefacts dans les ombres.



On active le frame buffer et on coche vray raw image file, à partir duquel on peut choisir d'exporter en exr 32bits



Petite astuce, dans 3ds max la commande `maxscript fileoutgamma=1.0` permet de passer l'export des images en gamma1, même pour les jpeg.



LES COULEURS

Généralités sur la lumière

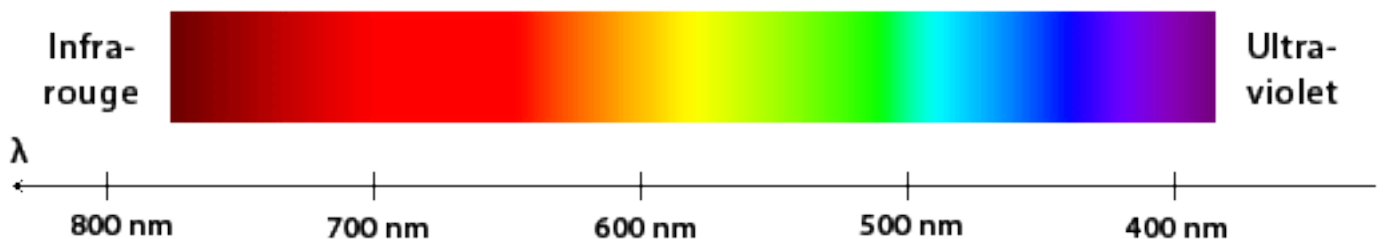
La lumière est une onde mesurée en hertz (fréquence) et en nanomètre (longueur d'onde).

La longueur d'onde définit la couleur.

Grande longueur d'onde = infrarouges

Faible longueur d'onde = ultraviolets

Entre les infrarouges et les ultraviolets, il y a le spectre visible : l'ensemble des couleurs visibles du rouge à l'indigo.



Toutes les couleurs bien mélangées et équilibrées donnent la lumière blanche.

Un rayon à lui tout seul ne peut donc pas être blanc.

Petite précision, le bleu est une couleur froide et le rouge une couleur chaude (lol). C'est la **température de couleur**. A ne pas confondre avec **température de la lumière**. En effet une lumière à dominante bleue a une température plus élevée qu'une lumière à dominance rouge ou jaune. Les étoiles les plus chaudes sont bleues (environ 12 000 K°), tandis que les étoiles les plus froides sont rouges (environ 1200K°).

Pourquoi voyons nous comme nous voyons ?

Il y a 3,5 milliards d'années, la vie apparaît dans l'eau sous la forme de bactéries. L'eau est un élément qui n'absorbe presque pas le spectre visible : l'eau nous apparaît comme transparente.

En revanche, plus on descend en profondeur, plus l'eau absorbe la lumière en partant des grandes longueurs d'ondes, c'est-à-dire les infrarouges, puis les rouges, les oranges etc. Les milieux sous-marins sont teintés de bleu et de vert : le rouge est intégralement absorbé.

Certaines bactéries ont mutées pour être capables d'absorber la lumière (la base de la photosynthèse). Elles ont une petite tâche noire, qui est en réalité l'ancêtre de l'oeil.

Plus tard les animaux terrestres développent une vision en bichromie : bleu et rouge, soit les extrêmes du spectre visible. Dans la nature, le bleu correspond au ciel et à la mer et le rouge à la Terre. Les primates vont évoluer et développer la vision du vert, couleur de la chlorophylle, qui constitue les plantes, à la base de l'alimentation des mammifères.

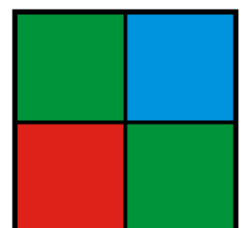
Aujourd'hui, le vert reste la couleur que l'on capte le plus : on peut voir une quantité incroyable de nuances de vert, qui parfois se confondent avec le bleu.

Le rouge n'étant pas une couleur à laquelle nous étions ancestralement habitués (sous l'eau pas de rouge), cette couleur attire d'avantage notre attention.

L'oeil voit la lumière grâce aux **cellules photosensibles** sur sa rétine

- *Les bâtonnets* : sensibles à la luminosité.
- *Les cônes* : sensibles aux couleurs. Il y a trois types de cônes
 - **Cyanolabes** : sensibles au **bleu** (437nm)
 - **Chlorolabes** : sensibles au **vert** (533nm)
 - **Erytholabes** : sensibles au **rouge** (564nm)

Comme nous voyons deux fois plus de vert, pour imiter la nature, les écrans et les capteurs d'appareils photos ont deux pixels verts pour un pixel rouge et un pixel bleu.



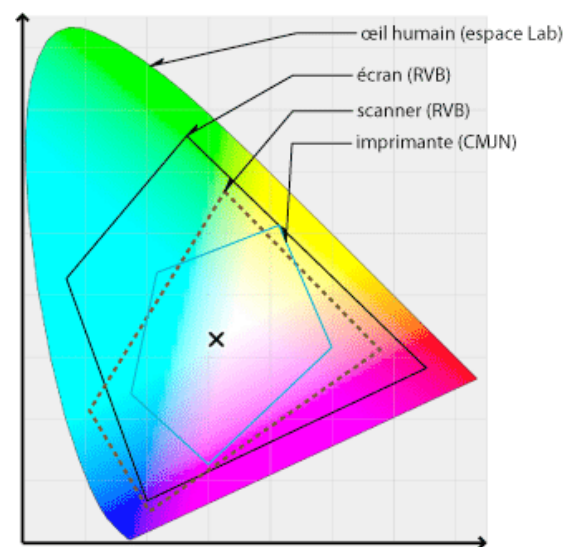
Les espaces de couleurs

Un espace de couleurs est **un ensemble de couleurs**, également appelé **gamut**. Il représente l'ensemble des couleurs reproductibles par une machine.

L'espace de couleur LAB est l'espace de référence à partir duquel on compare les gamuts. Il s'agit de **l'ensemble des couleurs que peut voir un Homme**. On estime à **8 millions** le nombre de couleurs visibles par un humain standard. C'est une moyenne, en réalité tout le monde n'a pas exactement la même vision.

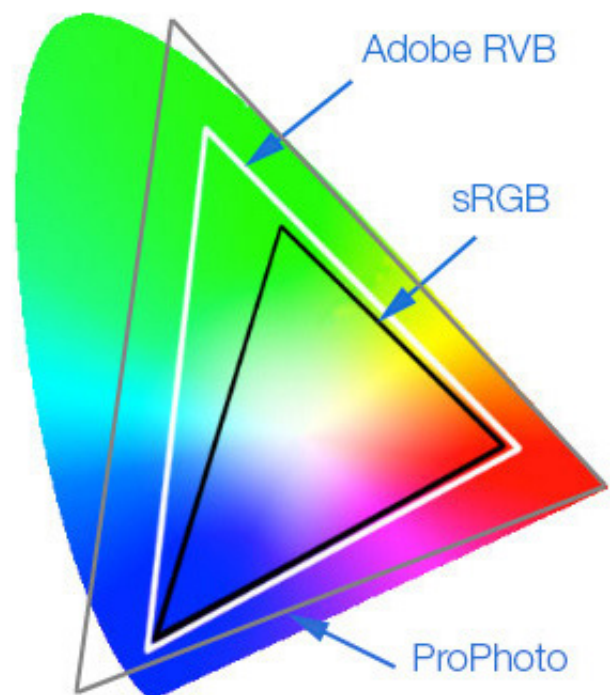
L'espace LAB représenté sur ce diagramme de chromacité permet de comparer différents gamuts. On projette dessus des gamuts de périphériques, également appelés **Profils ICC**, pour les comparer entre eux.

On peut voir qu'ils ont des couleurs communes, mais le gamut de l'écran contient des couleurs qui n'appartiennent pas au gamut de l'imprimante et inversement. Chaque appareil possède son espace couleur propre.



Il existe aussi des espaces de couleurs qui ne dépendent pas d'appareils. C'est un autre type de gamut. Ils ont été inventés pour travailler facilement, basés sur un modèle RVB ou encore CMJN.

Les différents profils sont projetés sur l'espace LAB, c'est ce qui définit la quantité de couleurs qu'ils peuvent afficher et quelle portion de l'espace LAB ils peuvent afficher.



Le standard RGB, ou sRGB, ou encore sRGB IEC61966-2.1, a été inventé en 1995 par microsoft. Il contient 35% des couleurs LAB.

L'espace sRGB est conçu pour correspondre à un affichage d'écran d'ordinateur standard. Son gamut est considéré comme suffisamment grand pour la plupart des applications. On considère que tout fichier image qui ne possède pas un profil couleur intégré est a priori à visualiser sous le profil sRGB. C'est un bon espace de couleur pour le web, mais assez limité pour l'impression. Dans le sRGB est inclu une courbe de correction de gamma, qui est l'inverse de 2,2, soit la plupart des écrans domestiques.

Adobe 98, inventé par adobe en 1998 : Contient 50% des couleurs LAB, il est plus grand que le sRGB, mais surtout dans les verts. Il est utile si on photographie du vert et qu'on veut imprimer sur du papier brillant. Il permet de voir plus de nuances.

Rappel : la vision humaine, c'est 8 millions de couleurs. On le sait parce qu'il faut 200 nuances dans un dégradé d'une seule couleur pour qu'il soit perçu comme continu, du plus sombre au plus clair.

200 nuances par couleur primaire, c'est $200 \times 200 \times 200 = 8$ millions.

Sur le dégradé du bas, on voit des cassures de tons, par ce qu'il n'y a pas suffisamment de nuances.



Comment cela se traduit informatiquement ?

Il faut comprendre la **notion de profondeur de BITS**

Le code informatique repose sur des bits, des impulsions électriques.

1 bit = 1 possibilité = 0 ou 1
2 bits = 4 possibilités = 00, 11, 01, 10
7 bits = 128 possibilités.
8 Bits = 256 possibilités.
16bits = 65536 possibilités.

Tous les pixels colorés d'une image sont créés grâce à la combinaison de trois couleurs primaires : rouge, vert et bleu. **Chaque couleur primaire est associé à un canal de couleur.** Une couleur primaire s'affiche à l'écran lorsque son canal est à 100 % et que les deux autres sont à 0%.

La profondeur de Bits quantifie combien de couleurs uniques ou de nuances de gris (dans le cadre d'une image en niveau de gris) sont possibles. Les images avec une grande profondeur de bit peuvent encoder plus de nuances ou de couleurs, puisqu'il y a plus de combinaisons possibles de 0 et 1.

Bits par canal et Bits par pixel

Bits par canal (BPC), c'est la profondeur de Bits pour chaque couleur primaire. Dans une image codée en 8 BPC, on parle donc de 8 bits par canal, autrement dit 256 valeurs possibles pour chaque couleur primaire. $256 \times 256 \times 256 = 16\,777\,216$ couleurs possibles (deux fois plus que la vision humaine).

Bits par pixel (BPP), c'est la profondeur de Bits pour chaque pixel. Une image codée en 24 BPP, c'est la même chose qu'une image codée en 8 BPC, car un pixel est composé des trois canaux RVB ($3 \times 8 = 24$).

On peut soit parler d'une image en 8 BPC ou en 24 BPP. C'est pareil.

Bits Per Pixel	Number of Colors Available	Common Name(s)
1	2	Monochrome
2	4	CGA
4	16	EGA
8	256	VGA
16	65536	XGA, High Color
24	16777216	SVGA, True Color
32	16777216 + Transparency	
48	281 Trillion	

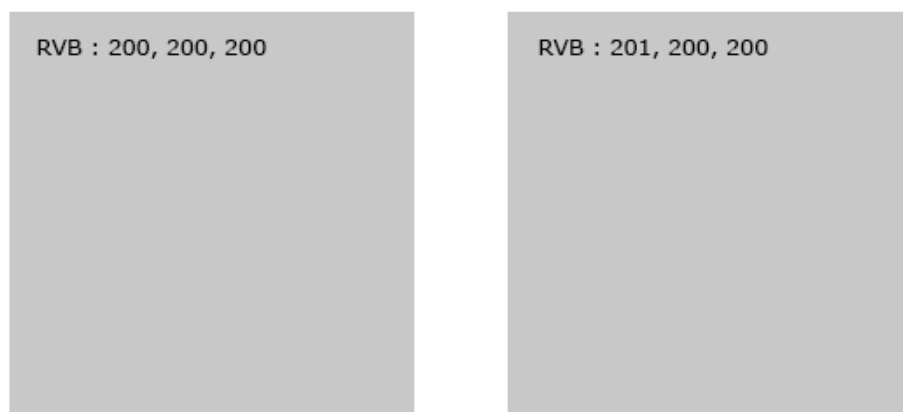
IMPORTANT : Il faut distinguer les **définitions de couleurs** et les **couleurs réellement perçues**.

Une définition de couleur, c'est sa valeur RVB.

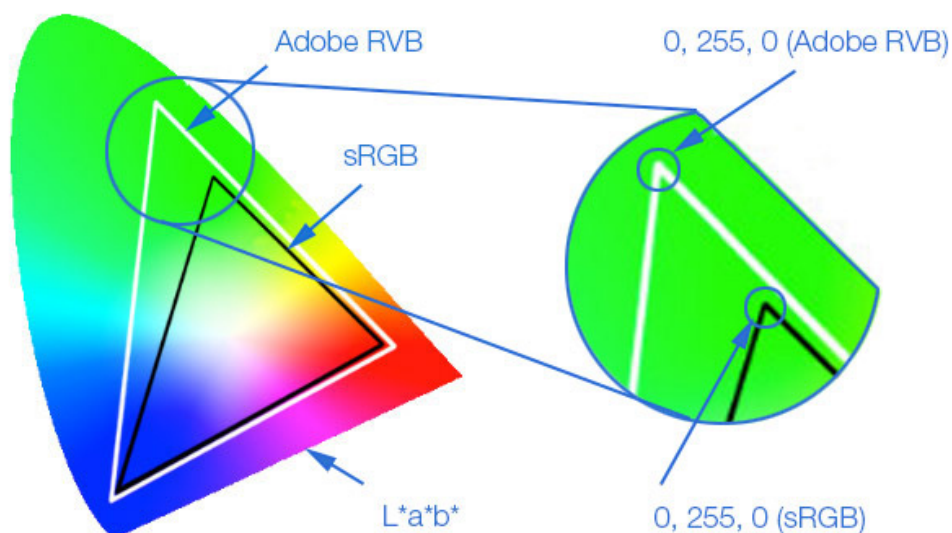
Exemple [R : 144, V : 88, B : 227]

Comme il y a deux fois plus de valeurs en 8 bits que dans l'espace LAB, **une même couleur perçue peut correspondre à plusieurs définitions de couleurs**.

Les deux carrés ci-dessous ont **deux valeurs RVB différentes, pour une même couleur perçue**. La vision humaine n'est pas capable de différencier ces deux gris, tout simplement parce qu'on est pas capable de distinguer autant de nuances qu'il en existe dans le 8 bits.



Enfin, **selon l'espace colorimétrique, une même valeur RVB ne correspondra pas à la même couleur LAB**. Ci-dessous, on voit bien que 0,255,0, qui correspond au vert le plus extrême, est plus saturé dans l'Adobe 98 que dans le sRGB, parce qu'il n'est pas projeté au même endroit sur l'espace LAB.



Le calibrage



236, 181, 74

Selon l'écran, ce carré orange paraîtra plus ou moins saturé, plus ou moins jaune. Pourtant il s'agit d'une seule et même valeur RVB.

Puisque **chaque écran interprète différemment les valeurs RVB**, il y a donc pleins de couleurs potentielles pour une seule valeur.

Quelque soit la couleur affichée à l'écran, il s'agit forcément d'une couleur LAB, une des 8 millions visible par l'être humain.

Les écrans ne sont pas capables de reproduire toutes les couleurs que voit un être humain.

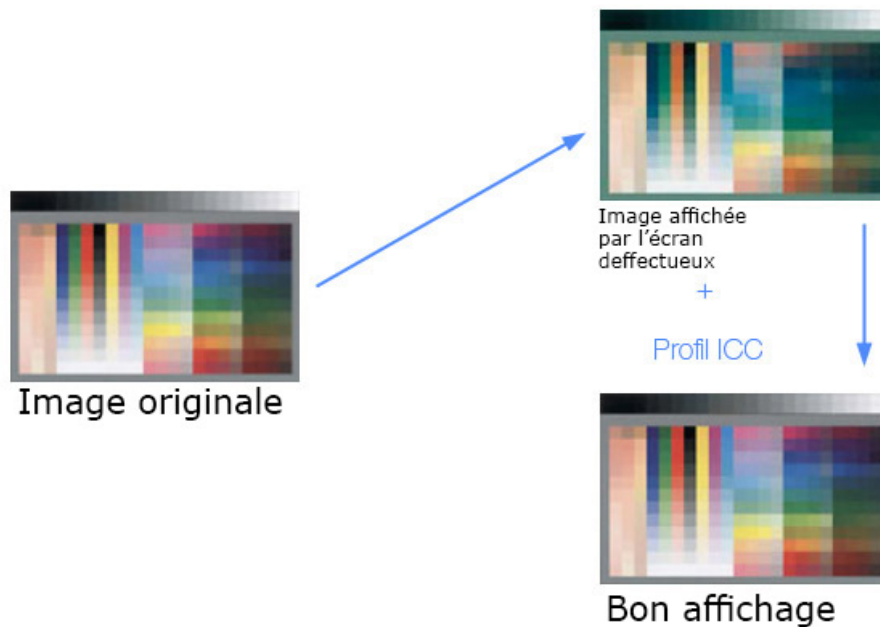
Le calibrage permet de connaître le gamut d'un appareil, c'est-à-dire l'ensemble des couleurs qu'un appareil peut reproduire **et de mesurer ses défauts**.

C'est l'espace LAB qui est pris pour référence. Il y a une norme qui applique une couleur LAB à chaque valeur RVB.

Ainsi, on sait que **telle valeur RVB est censée afficher telle couleur précise** sur l'écran. C'est sur ce principe que se base le calibrage d'un écran. On utilise une sonde de calibrage, qui envoie une valeur RVB à l'écran et lui demande de l'afficher. Cette valeur RVB est censée correspondre à une couleur précise de l'espace LAB. Si l'écran n'affiche pas exactement cette couleur, alors il y a un défaut. **La sonde permet de voir à quel point la couleur que l'écran affiche est différente de celle qu'elle devrait afficher**.

Ensuite la sonde enregistre ces informations dans un fichier qu'on appelle **profil ICC**. (.icc ou .icm).

Le profil ICC contient le gamut et les défauts de l'appareil. C'est en quelques sortes sa carte d'identité. Il va permettre de réajuster et de compenser toutes les erreurs de l'écran pour qu'il affiche les bonnes couleurs.



Un profil ICC peut également s'appliquer sur une photo. Dans un appareil photo, le profil ICC est déjà existant. Le profil ICC s'applique sur la photo qui vient d'être prise, un peu comme un post it. Du coup la photo est réajustée, en fonction des défauts de l'appareil, pour nous afficher les bonnes couleurs.

Le problème c'est que si on importe cette photo dans photoshop par exemple, ce qui nous apparaîtra comme une bonne couleur aura toujours les mauvaises valeurs RVB (comme sur l'image ci-dessous).

Heureusement, ça n'arrive jamais puisque l'appareil photo convertit la photo dans un espace de couleur neutre, tel que le sRGB, en écrasant le profil ICC et en réassociant les bonnes valeurs RVB à ce qu'on voit. Tout ça se fait automatiquement, soit dans l'appareil photo pour les jpeg, soit en post traitement avec un raw.

Une image a forcément besoin d'un profil pour s'afficher. Si elle n'en a pas, photoshop lui attribue le profil de son espace de travail par défaut. En gros, **un fichier qui contient des valeurs RVB ne correspond à aucune couleur tant qu'il n'est pas associé**

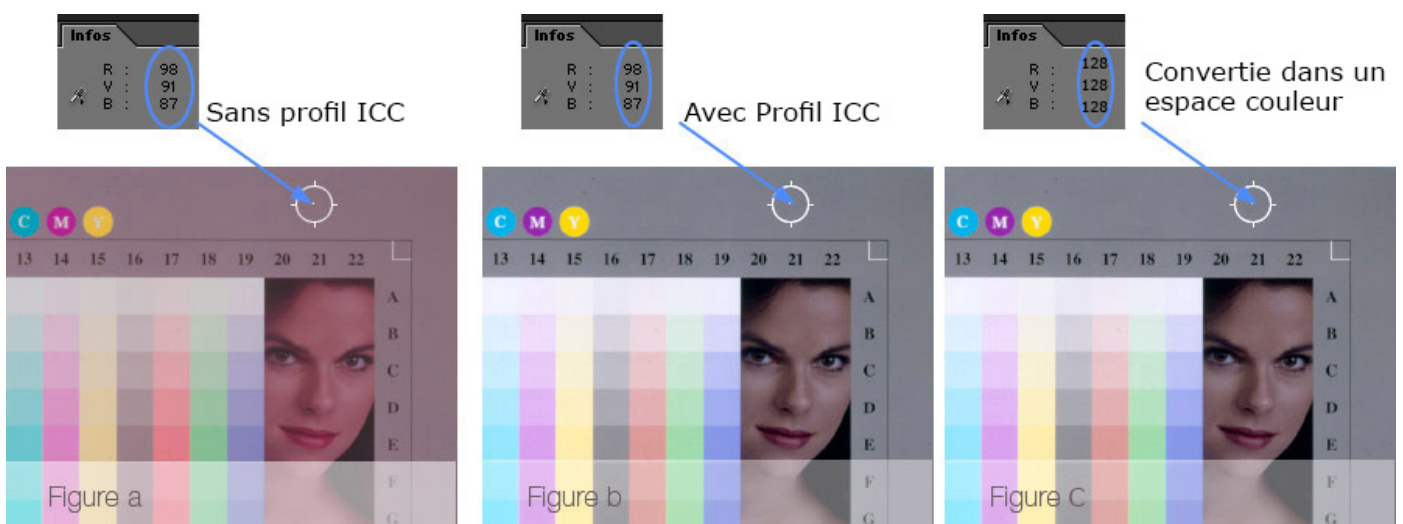


Image de gauche : image avec défauts. Image du milieu : corrigée avec un profil icc. Les mêmes valeurs RVB se sont vues associer d'autres couleurs LAB. Image de droite : Convertie dans espace de couleur neutre pour que les valeurs rvb correspondent aux nouvelles couleurs LAB affichées.

ANNEXES

Les informations et les illustrations utilisées pour réaliser ce petit dossier sont accessibles à ces liens :

<http://www.profil-couleur.com/tp/216-correction-gamma.php>

http://www.blog-couleur.com/modale/gamma-lineaire.htm?keepThis=true&TB_iframe=true&height=660&width=420

<https://www.mattguetta.com/le-linear-workflow/>

https://en.wikipedia.org/wiki/Gamma_correction

<http://www.pixine.fr/creer-des-fichiers-raw-avec-3dsmax/>

http://www.3dbunk.com/index.php?p=t&t=gamma_et_linearWorkflow&lang=FR

<http://www.blog-couleur.com/?Qu-est-ce-que-le-sRGB>

<http://www.cambridgeincolour.com/tutorials/bit-depth.htm>

<http://www.cambridgeincolour.com/tutorials/gamma-correction.htm>

<https://www.youtube.com/watch?v=iSwnXjADO9U>

<http://www.blog-couleur.com/?Qu-est-ce-que-le-gamma#gamma>

<http://www.blog-couleur.com/?Qu-est-ce-que-la-correction-du>

<http://www.blog-couleur.com/?Qu-est-ce-qu-une-courbe-de>

<http://www.blog-couleur.com/?Qu-est-ce-que-le-seuil>

http://www.blog-couleur.com/modale/courbe-logarithme.htm?keepThis=true&TB_iframe=true&height=730&width=440

<http://www.cambridgeincolour.com/tutorials/sRGB-AdobeRGB1998.htm>

<http://www.cambridgeincolour.com/tutorials/color-spaces.htm>

<https://www.focus-numerique.com/comprendre/dossiers/le-raw-le-log-et-le-rec709-en-video-723.html>