

Modèles de couleur, modèles d'affichage et matériel graphique

Physiologie humaine de la vision

- perception de l'intensité lumineuse
 - la rétine humaine possède deux types de capteurs:
 - de 75 M à 150 M bâtonnets, sensibles à l'intensité lumineuse
 - environ 7 M cônes (rouges, verts et bleus), sensibles à la couleur
 - l'œil humain est donc beaucoup plus sensible aux intensités qu'aux couleurs: un contraste d'intensité sera plus facilement perçu qu'un contraste de couleur

moins facile

facile

facile

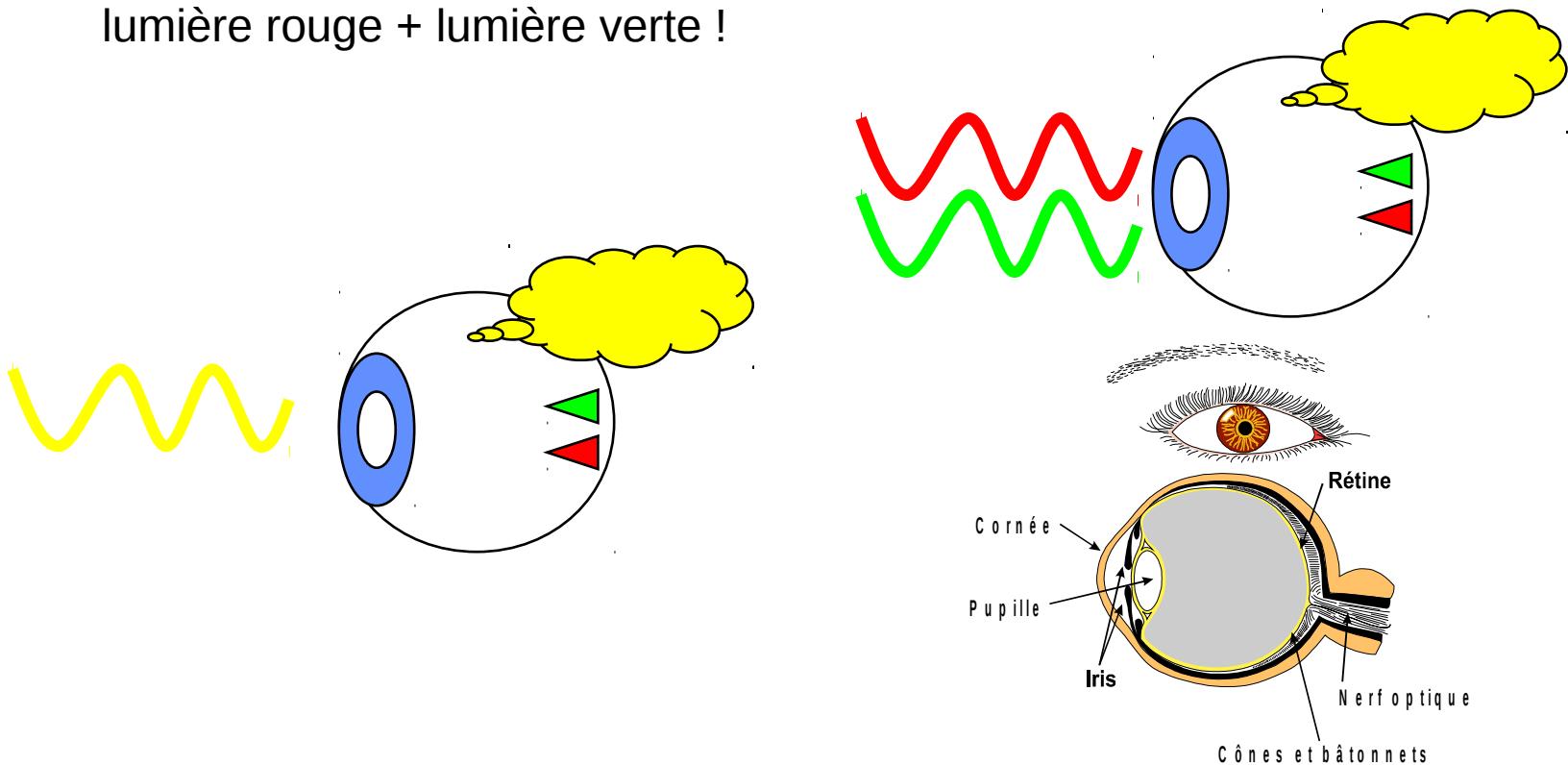
- la sensibilité aux différences d'intensité est moins bonne aux extrémités du spectre visible

facile

moins facile

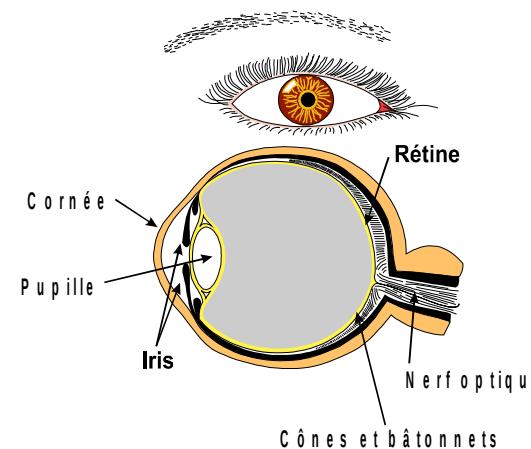
Physiologie humaine de la vision

- perception de la couleur
 - pour nous, lumière jaune = lumière rouge + lumière verte !



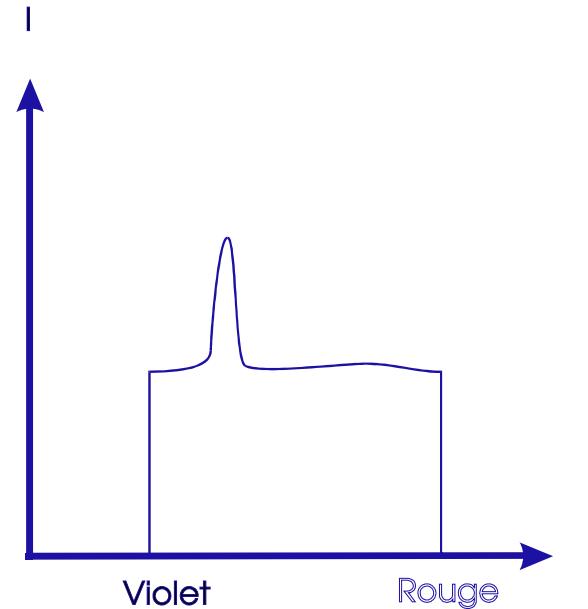
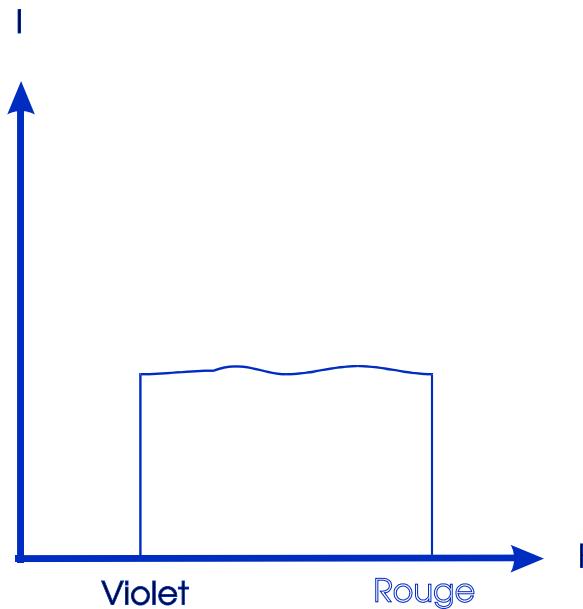
Physiologie humaine de la vision

- acuité visuelle
 - le centre de la rétine a une forte densité de cônes et n'a pas de bâtonnets
 - c'est la raison pour laquelle il est très difficile, la nuit, de bien voir un faible point lumineux (p.e. une étoile)
 - si on le regarde droit en face (centre de la rétine)
 - mais on peut le voir si l'on regarde un peu à côté.
 - de plus, on n'arrive pas à lire ailleurs que vers le centre du champ de vision



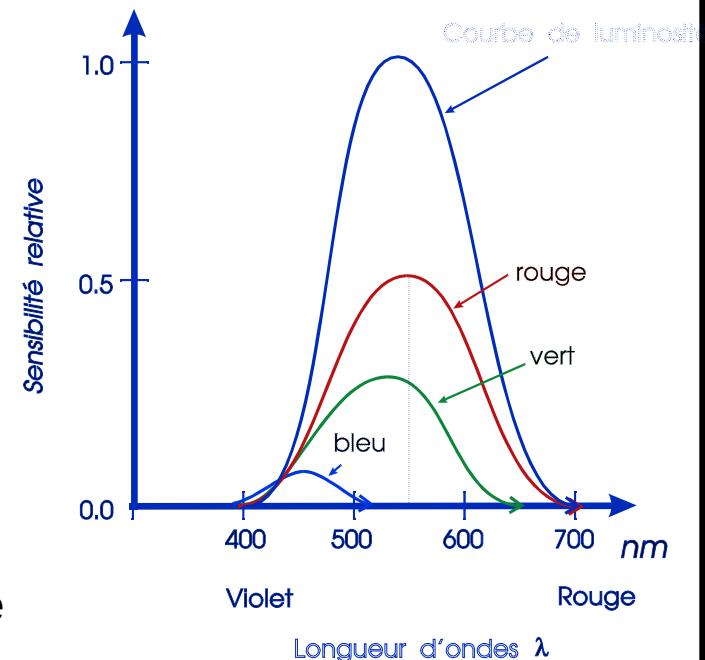
Modèles de couleur

- La perception de la lumière ou des différentes couleurs: liée à la sensibilité du système oculaire aux ondes électromagnétiques à l'intérieur d'une plage de fréquence.



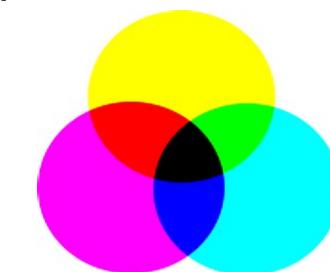
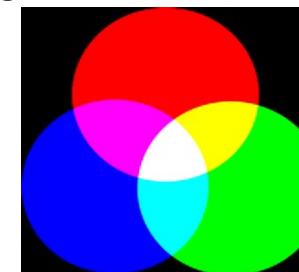
Modèles de couleur

- Une source de lumière: émet toutes les fréquences pour produire la couleur blanc.
 - Certaines fréquences sont absorbées, ou réfléchies => les ondes ré-émises par l'objet sont une certaine combinaison du spectre donnant lieu à une couleur que l'on associe à l'objet.
 - On divise le spectre en trois zones: les ondes courtes, moyennes et longues donnant respectivement bleu, vert et rouge.
 - Caractéristiques:
 - couleur ou la teinte.
 - l'énergie qui donne lieu à l'intensité lumineuse.
 - pureté ou saturation est une mesure relative de l'intensité de l'onde (ou couleur) dominante par rapport à la distribution donnant la couleur blanche.



Modèles de couleur : norme de la CIE

- Lumière monochromatique : peu dans la pratique.
 - Couleurs perçues : mélanges.
 - La théorie des trois couleurs primaires du mélange des couleurs se base sur la supposition qu'il existe trois types de cônes sensibles à la couleur dans la portion centrale de la rétine de l'oeil.
 - Deux systèmes importants de mélange de couleurs primaires en infographie:
 - ▶ le système rouge-vert-bleu [RVB] additif,
 - ▶ le système cyan-magenta-jaune [CMJ] soustractif.
- Les trois sensibilités de l'œil humain : fonctions de la longueur d'ondes des émissions lumineuses correspondant respectivement au bleu, vert et rouge.
- La CIE (Commission Internationale de l'Éclairage) (1931) : modèle basé sur ces trois sensibilités; permet en infographie de comparer les différents moniteurs sur le marché.



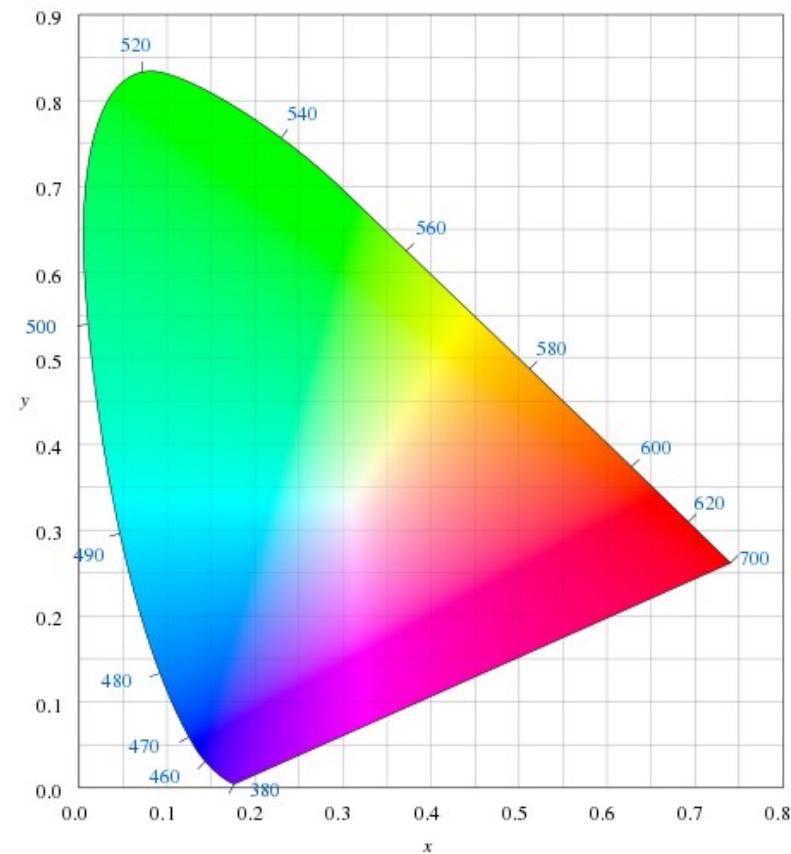
Modèles de couleur : norme de la CIE

● L'œil distingue trois stimuli => la nature tridimensionnelle de la couleur :

- la longueur d'onde (teinte),
- la pureté (saturation)
- la luminance (luminosité)

ou bien

- le rouge,
- le vert,
- le bleu.



Modèles de couleur : norme de la CIE

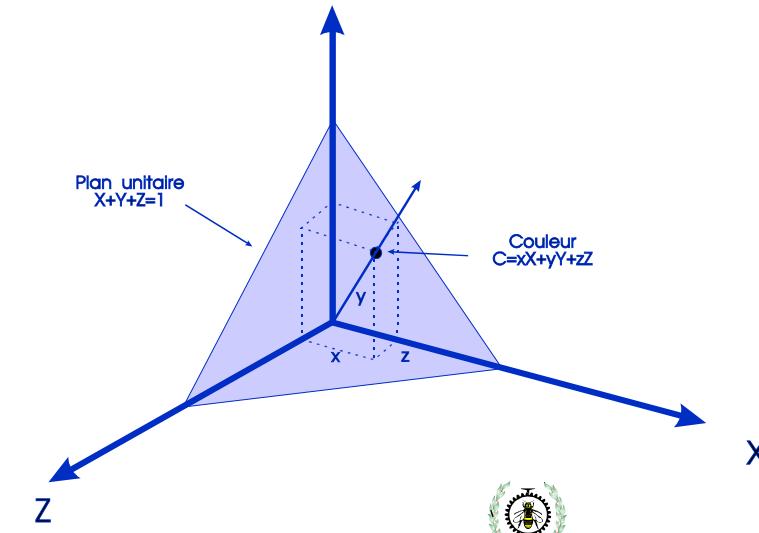
- Une couleur C comme une combinaison linéaire de trois couleurs primaires.
- Dans le modèle CIE, les trois couleurs primaires XYZ (pas des couleurs réelles).
 - X est une primaire rougeâtre ayant une saturation supérieure au rouge.
 - Y est un vert sursaturé ayant la même longueur d'ondes que le vert,
 - Z un bleu sursaturé ayant la même longueur d'ondes que le bleu.
- Poids des couleurs:

$$x = \frac{X}{X + Y + Z}$$

$$y = \frac{Y}{X + Y + Z}$$

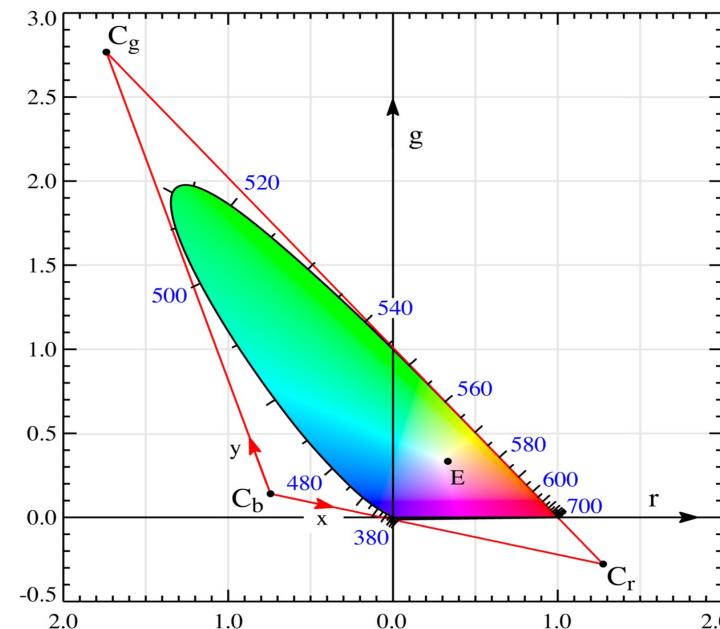
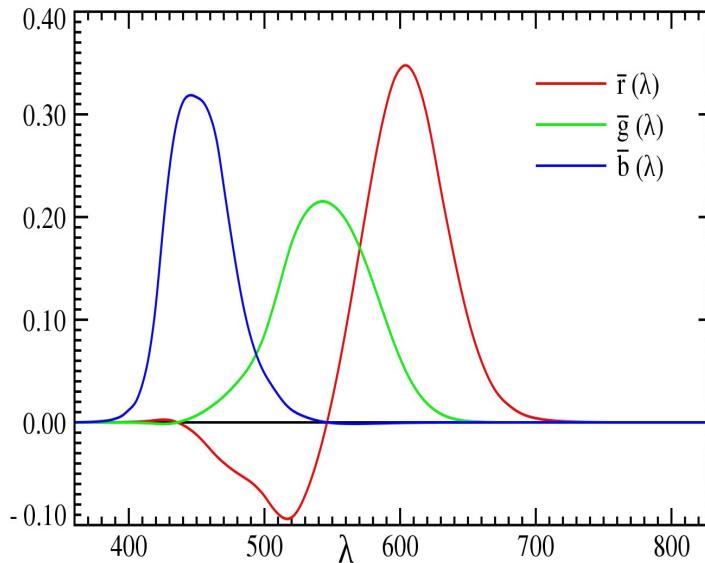
$$z = \frac{Z}{X + Y + Z}$$

- x, y, z : proportions des couleurs primaires
- X, Y, Z dans une couleur et $x + y + z = 1$



Modèles de couleur : norme de la CIE

- ➊ La nature tridimensionnelle de la couleur suggère de porter les valeurs de chacune des composantes des trois couleurs primaires sur trois axes orthogonaux.



Modèles de couleur : norme de la CIE

- Les couleurs primaires peuvent être représentées par un diagramme chromatique. La droite de luminosité zéro, à laquelle appartiennent les primaires X et Z est définie comme suit:

$$0 = 0.177 R + 0.812 V + 0.0106 B$$

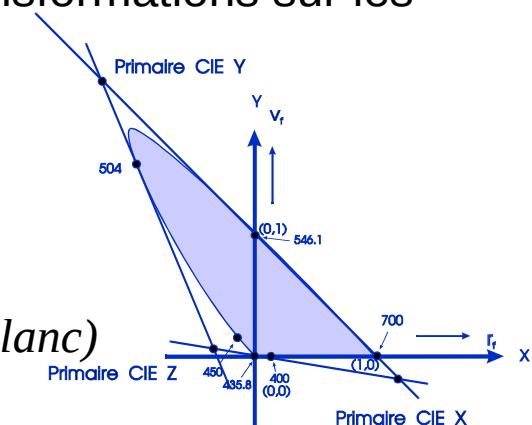
Puisque $z = 1 - x - y$, on obtient $y = -0.207 x - 0.0132$. En ajoutant une autre droite tangente à la longueur d'ondes 504 nm, et un troisième coupant l'axe des x au point 1.0, les intersections de ces lignes permettent de définir les couleurs primaires XYZ du modèle CIE. Si on retrace ce diagramme en utilisant ces droites comme axe, on obtient une courbe en forme de selle de cheval qui constitue les couleurs pures à 100%. L'effet de ces transformations sur les couleurs primaires RGB est:

$$R \text{ (700 nm)}: (1, 0, 0) \rightarrow (0.735, 0.265, 0)$$

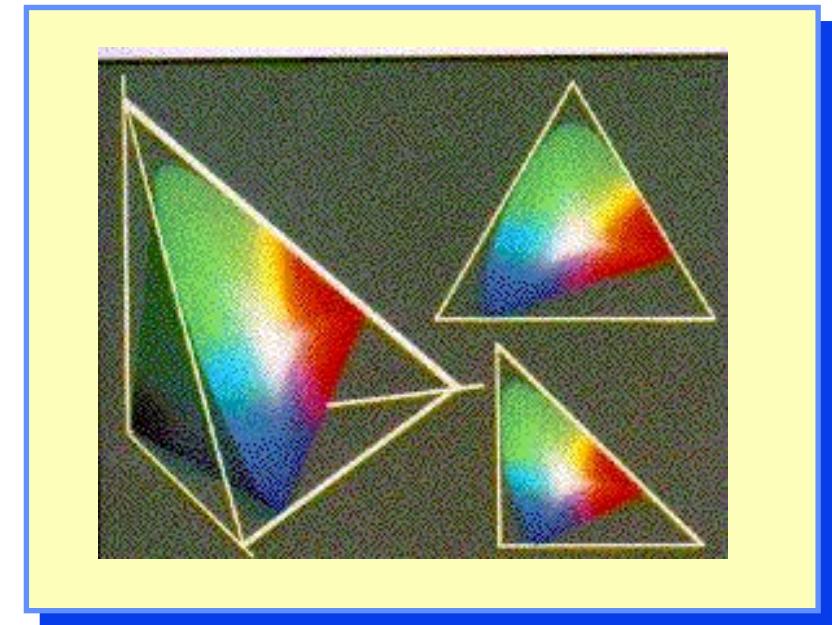
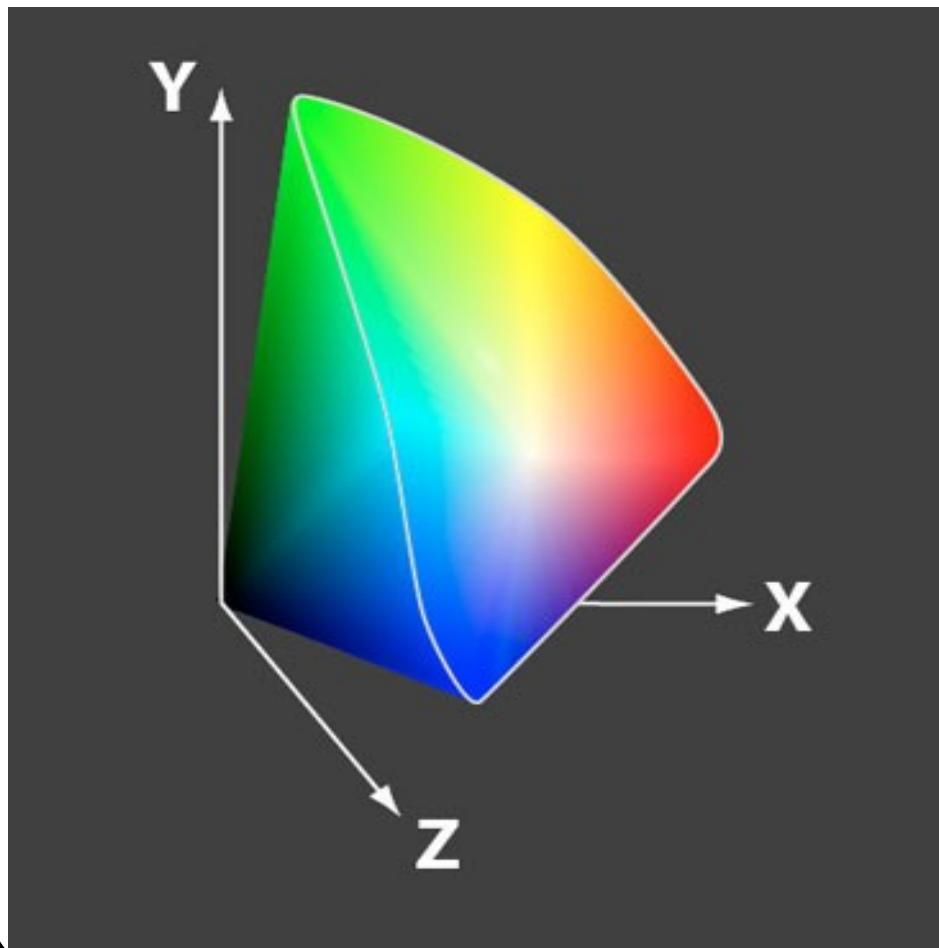
$$V \text{ (546 nm)}: (0, 1, 0) \rightarrow (0.273, 0.717, 0.009)$$

$$B \text{ (435.8 nm)}: (0, 0, 1) \rightarrow (0.167, 0.009, 0.825)$$

$$E \text{ (énergie égale)} (0.333, 0.333, 0.333) \rightarrow (0.333, 0.333, \text{blanc})$$



Modèles de couleur : norme de la CIE

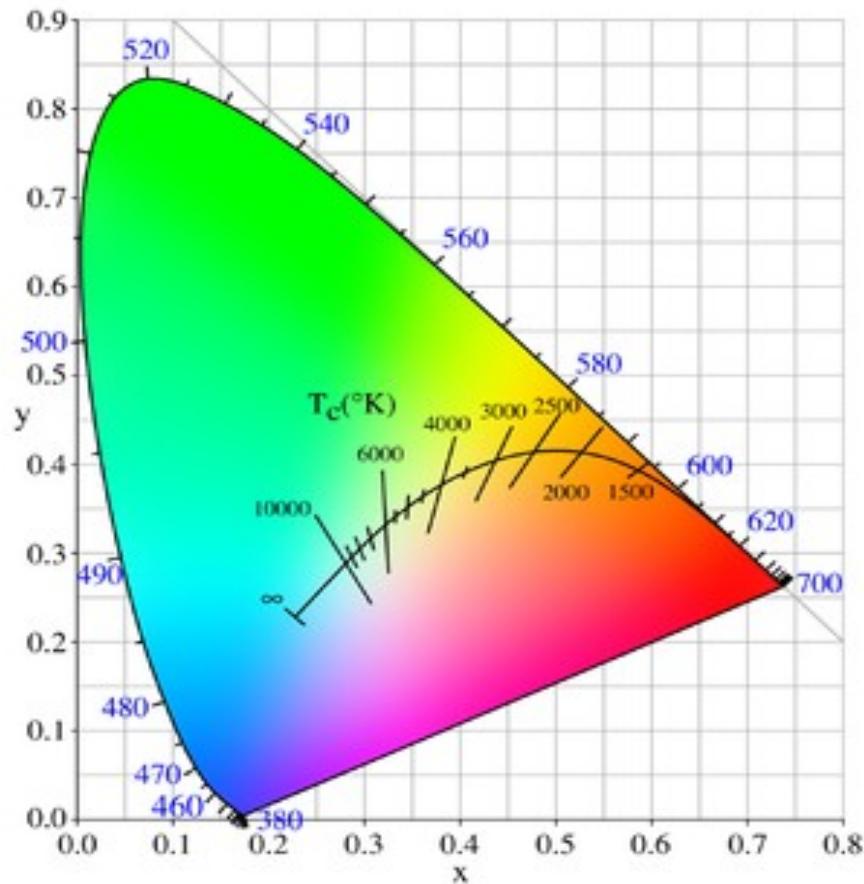
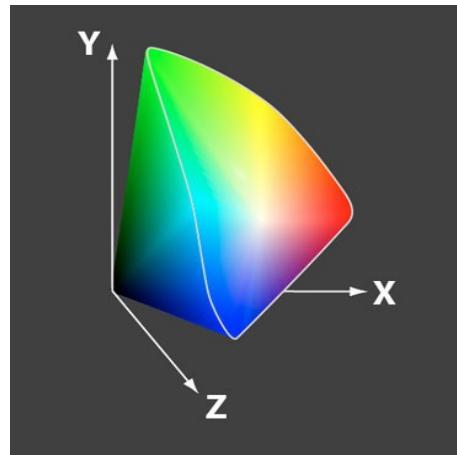


Modèles de couleur : norme de la CIE

Utilisation

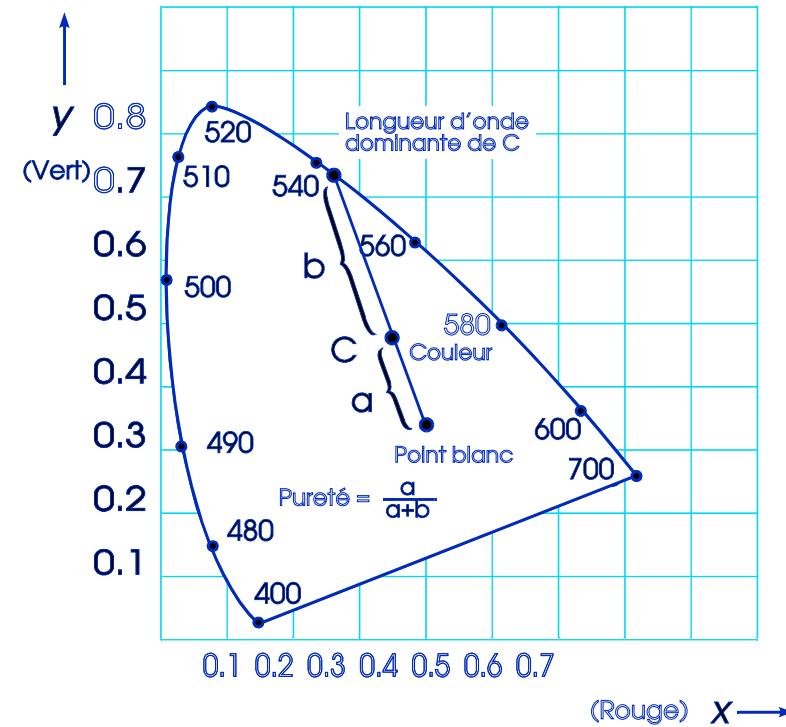
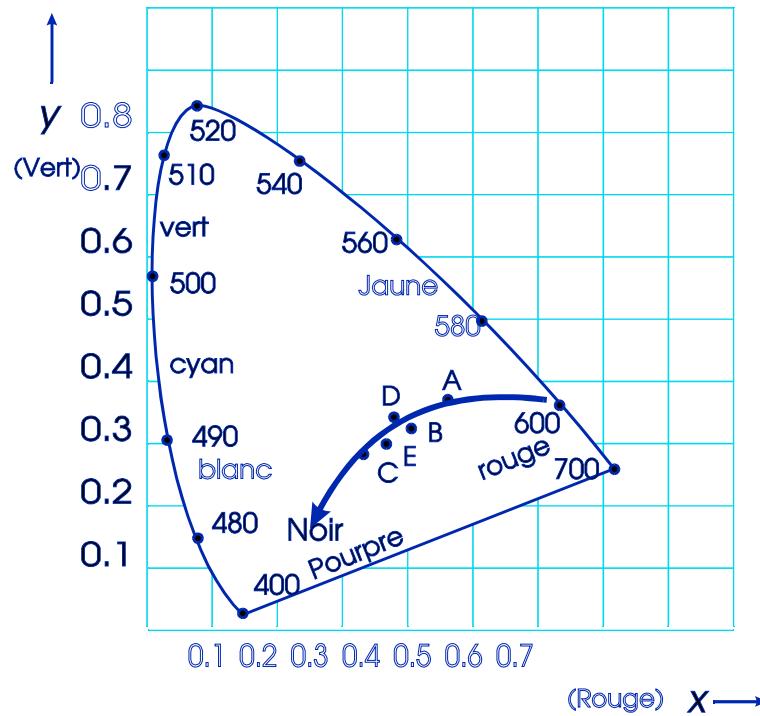
- Comparer les couleurs des moniteurs des différents fabricants.

Chaque fabricant peut fournir les coordonnées chromatiques des trois phosphores qu'il utilise.



Modèles de couleur :

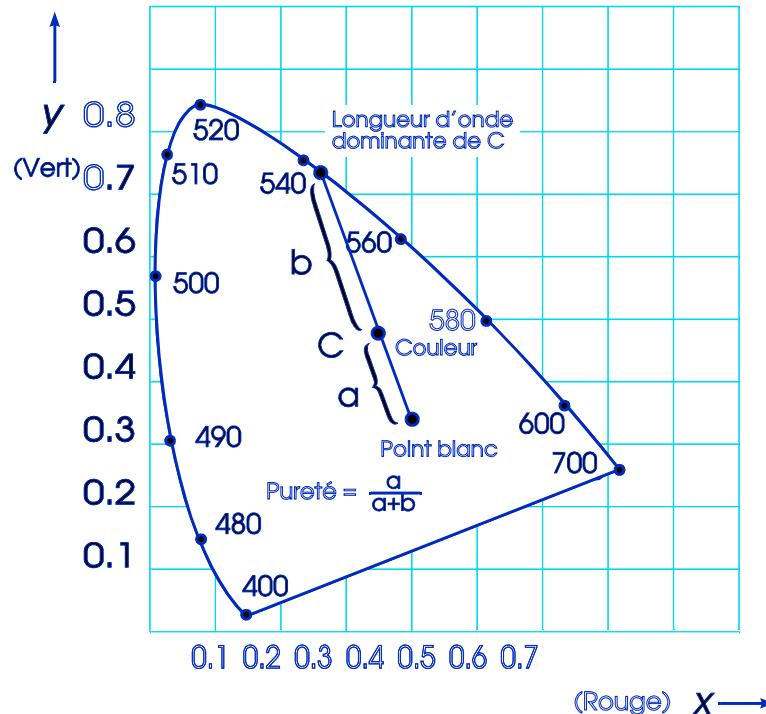
norme de la CIE



Modèles de couleur : norme de la CIE

Utilisation

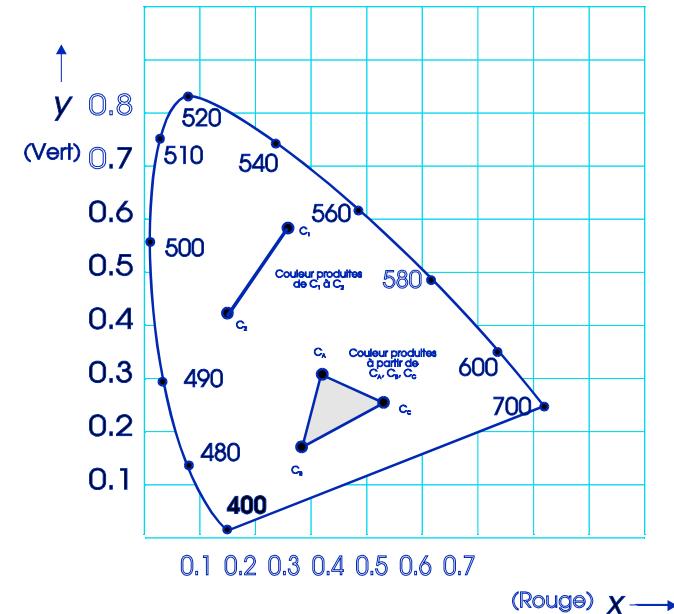
- Retrouver la longueur d'onde et la pureté.
Le blanc est 0% pur et les couleurs spectrales sont 100% pures. Le rapport $a / (a + b)$ donne la pureté de la couleur C .



Modèles de couleur : norme de la CIE

Utilisation

- Prédire les mélanges des couleurs.



Pour mélanger deux couleurs C_1 et C_2 , il suffit d'additionner leurs valeurs XYZ du diagramme CIE:

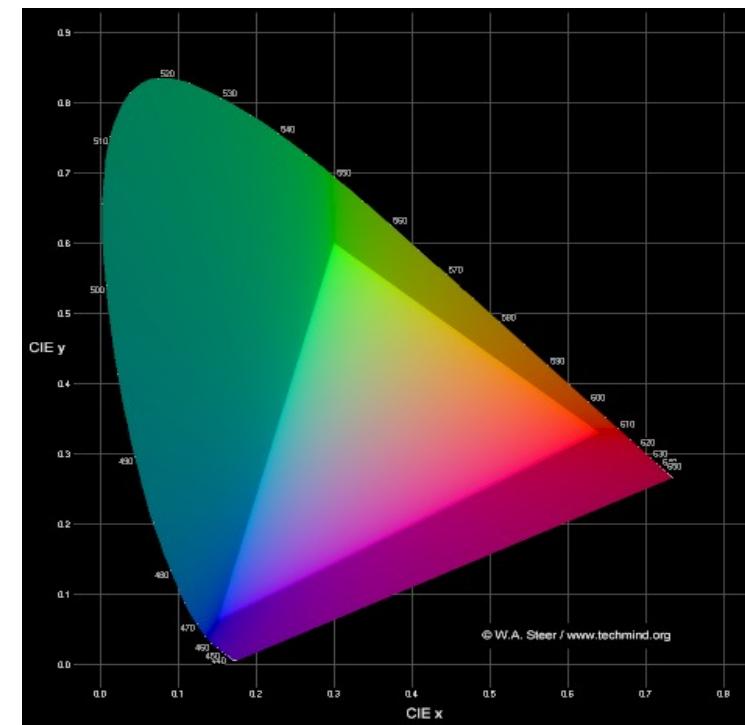
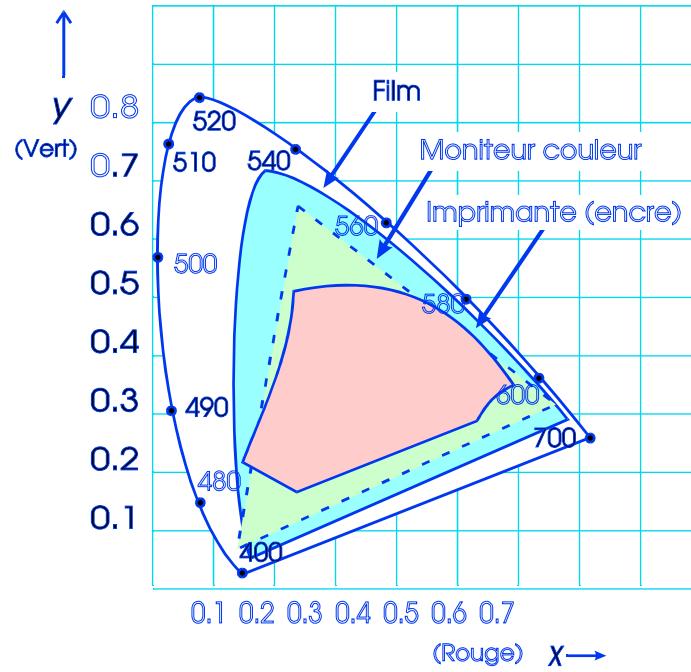
$$C_{12} = C_1 + C_2$$

Sur la base de cette théorie, différents modèles de couleur ont été proposés pour les applications informatiques.

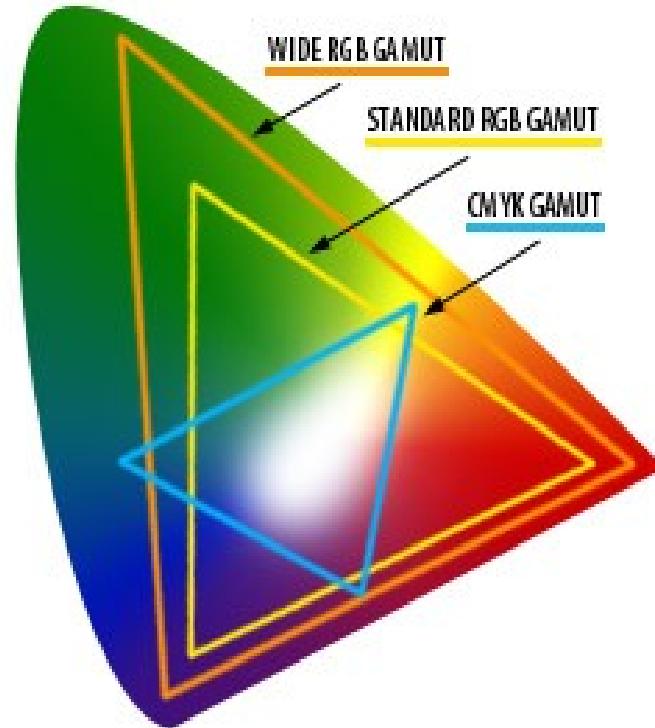
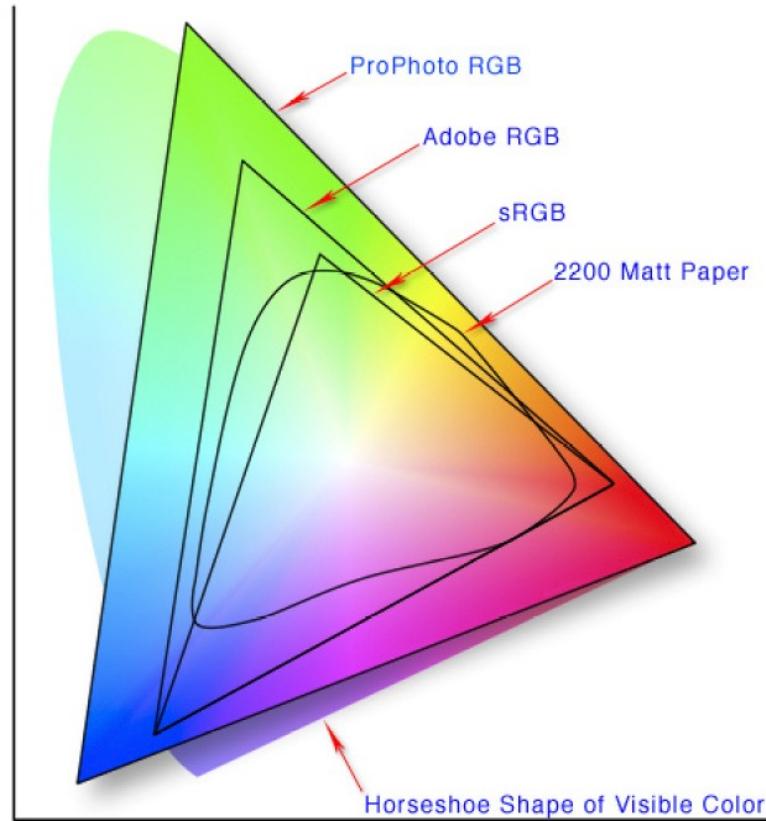
Modèles de couleur :

norme de la CIE

Phosphores à courte rémanence			Phosphores à longue rémanence			
Rouge	Vert	Bleu	Rouge	Vert	Bleu	
(x)	(y)	(z)	(x)	(y)	(z)	
x	0.61	0.29	0.15	0.62	0.21	0.15
y	0.35	0.59	0.063	0.33	0.685	0.063



Modèles de couleur : norme de la CIE



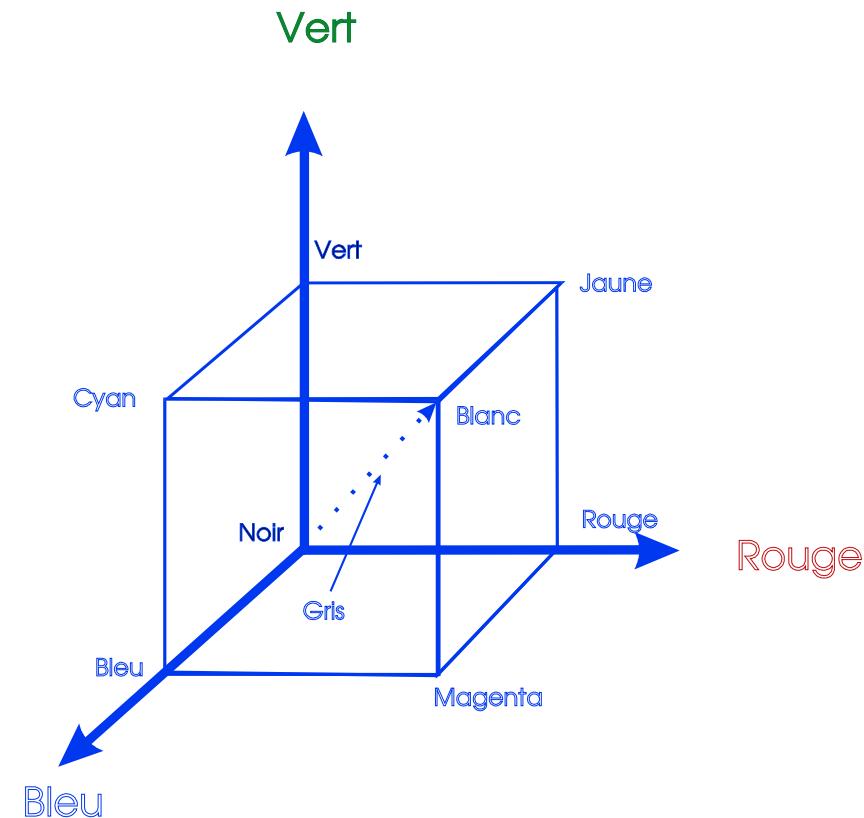
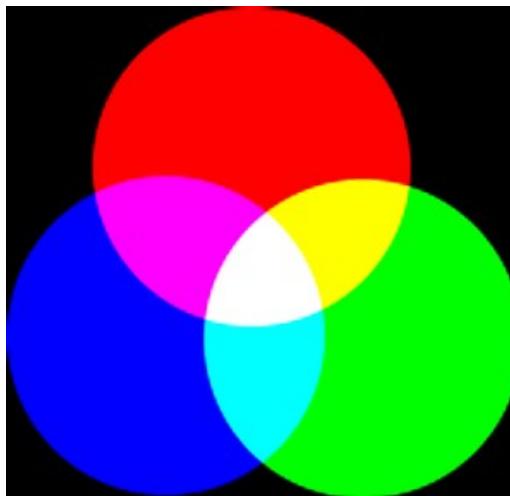
Modèles de couleur : modèle RVB [RGB]

- Le modèle RVB utilise les composantes rouge, vert et bleu où une couleur est représentée par le point (R,V,B).

(0,0,0) : noir

(1,1,1) : blanc

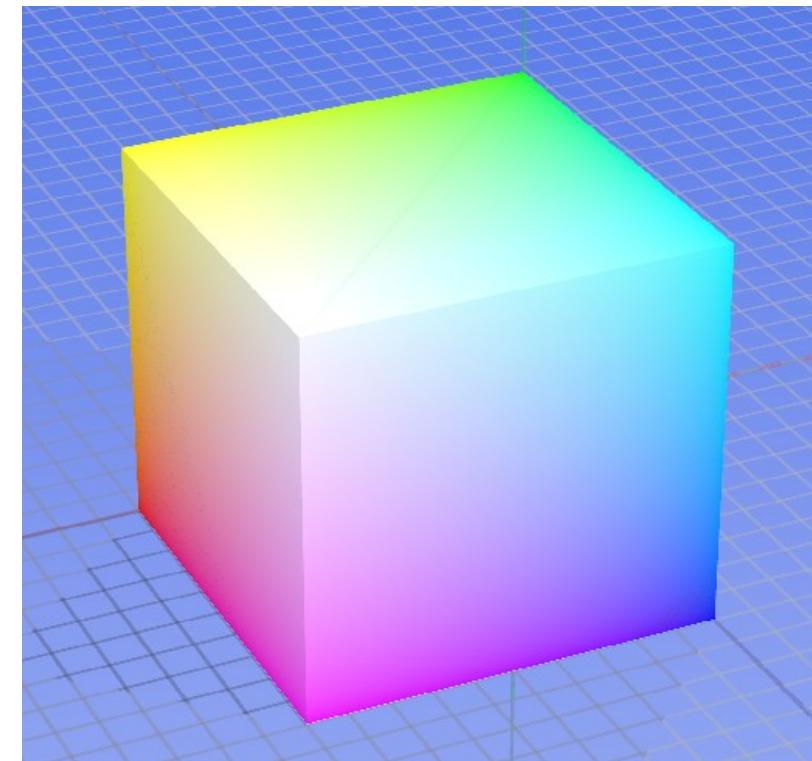
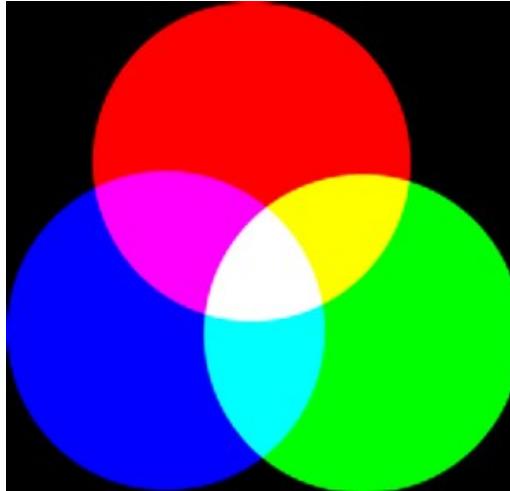
diagonale: teintes de gris



Modèles de couleur : modèle RVB [RGB]

- Modèle additif: la surface d'affichage qui émet des radiations, de la lumière. C'est celui utilisé par les moniteurs des systèmes graphiques.

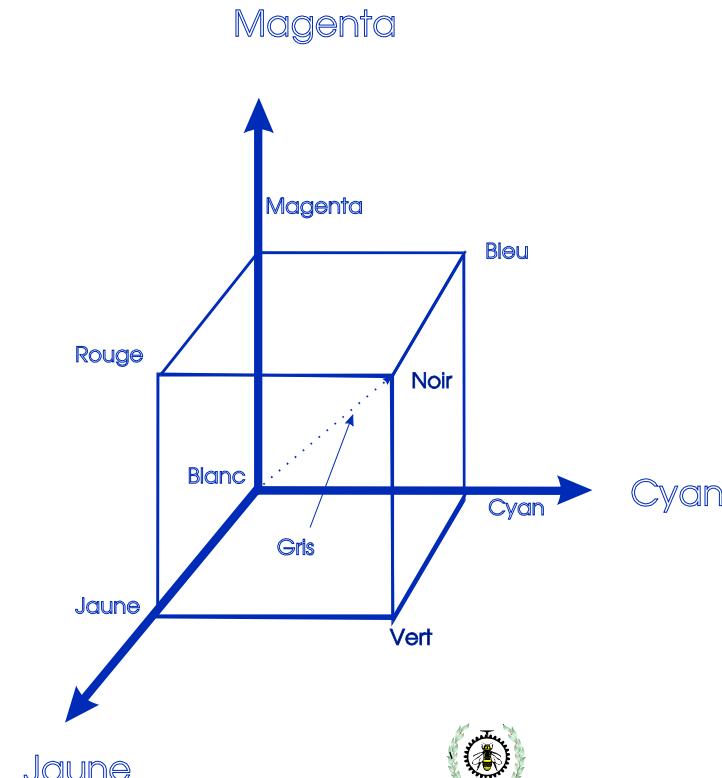
Le volume est:



Modèles de couleur : modèle CMJ [CMY]

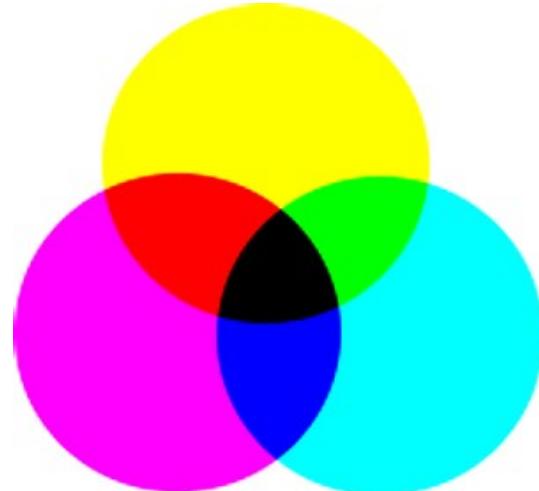
- **Imprimantes à jet d'encre:** la couleur est produite par réflexion c'est-à-dire qu'une couleur est obtenue en retranchant les couleurs primaires de la lumière blanche qui est la somme des couleurs rouge, vert, bleu.
- Par exemple, une encre magenta soustrait la composante verte de la lumière blanche tandis qu'une encre cyan soustrait la composante rouge de la lumière blanche. Ainsi une surface encrée de cyan, magenta et jaune absorbe des rouge, vert et bleu, et est noire. On désigne ce modèle cyan, magenta, jaune.
Ces relations peuvent être représentées par les équations

$$\begin{bmatrix} Cyan \\ Magenta \\ Jaune \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} Rouge \\ Vert \\ Bleu \end{bmatrix}$$

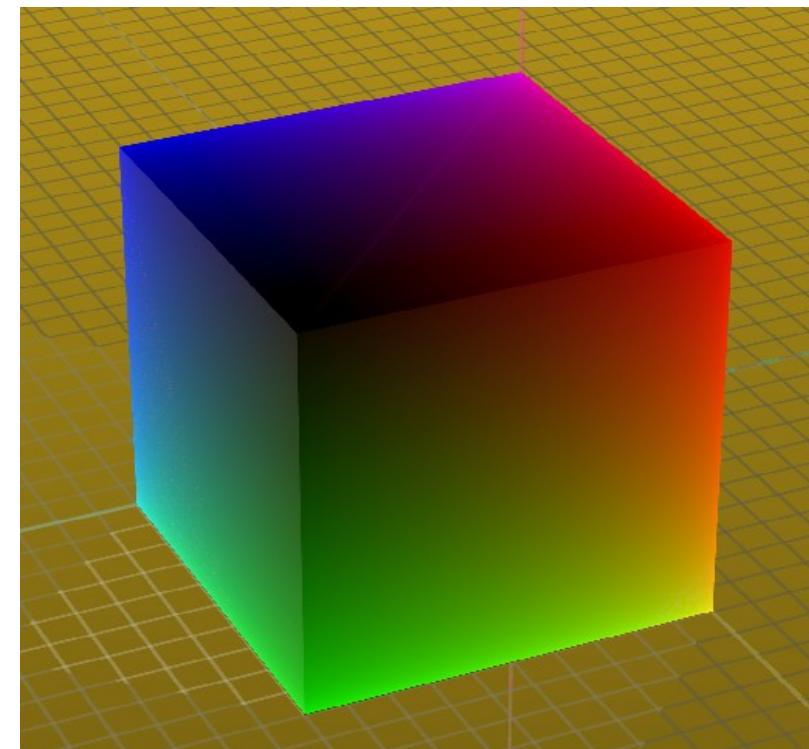


Modèles de couleur : modèle CMJ [CMY]

- Modèle soustractif: la surface encrée *absorbe* les couleurs, la lumière. C'est celui utilisé par les imprimantes couleur.

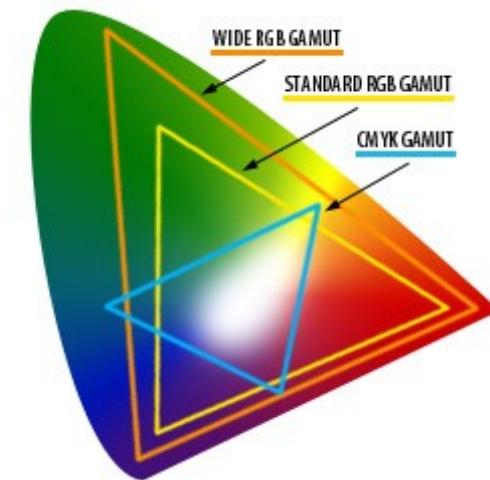
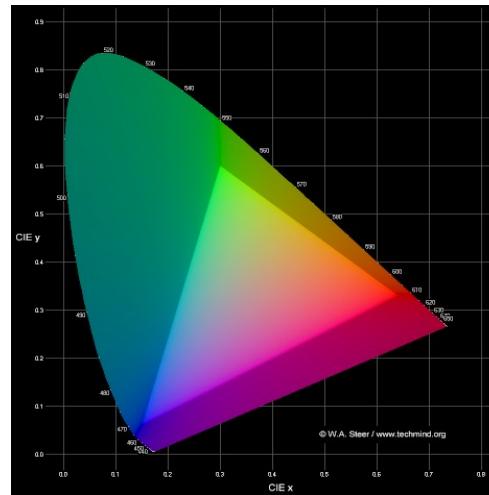


Le volume est:



Modèles de couleur : (norme de la CIE)

- Un moniteur (LCD, CRT) utilise un modèle de couleur **additif**; les coins d'un triangle RGB permettent le maximum de couleurs.



- Une imprimante utilise un modèle **soustractif**; le diagramme CIE ne s'applique pas !

Modèles de couleur : modèle HSL

Modèle HSL (/HSI) utilise :

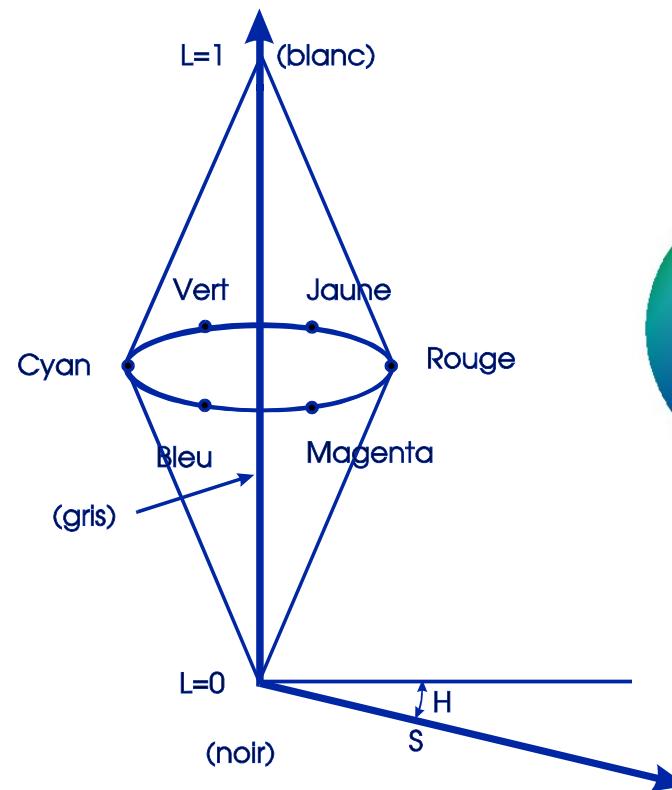
- **teinte [hue]** : position dans le spectre
- **saturation** (ou pureté): intensité (quantité de "couleur" par rapport au gris)
- **clarté [lightness ou intensity]**: position entre le blanc et le noir.

Représentation graphique :

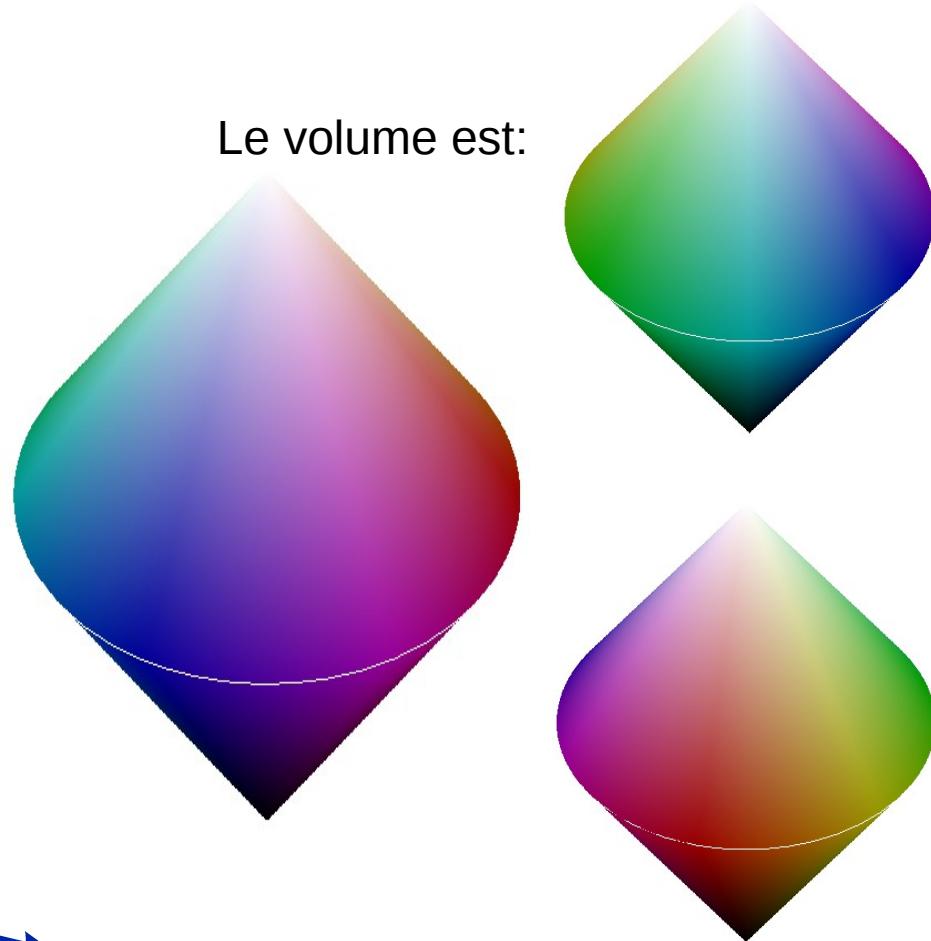
- La variation de la teinte (H):
- l'axe vertical pour la clarté (L).
- Sur ce cercle appelé chromatique, les couleurs disposées à 180 à 240 degrés donnent le blanc et sont appelées complémentaires.
- La saturation ou pureté (ton): 0 (pas de couleur) et 1 (la couleur pure)- la valeur du rayon.
- clarté (L) : position le long de l'axe vertical - teinte de gris. Ainsi l'ouverture du cône S, donne la quantité de couleur par dessus une teinte de gris, L sa hauteur.
- L'angle circonférenciel précise la nuance de couleur, c'est-à-dire la longueur d'onde.

Rouge	= 0
Jaune	= 60
Vert	= 120
Cyan	= 180
Bleu	= 240
Magenta	= 300

Modèles de couleur : modèle HSL



Le volume est:



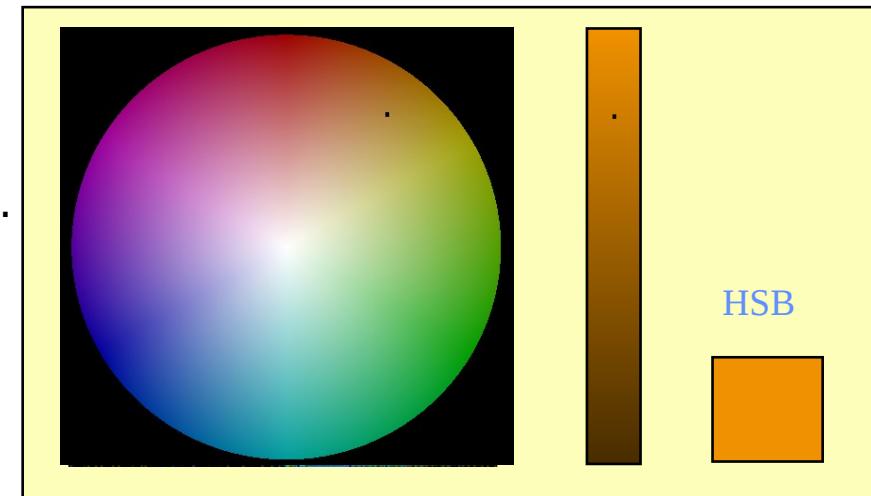
Modèles de couleur : modèle HSV

- Le modèle HSL (/HSI) utilise:

- teinte [*hue*] -- position dans le spectre.
- saturation (ou pureté) -- intensité par rapport à l'intensité moyenne.
- clarté [*lightness* ou *intensity*] -- position entre le blanc et le noir.

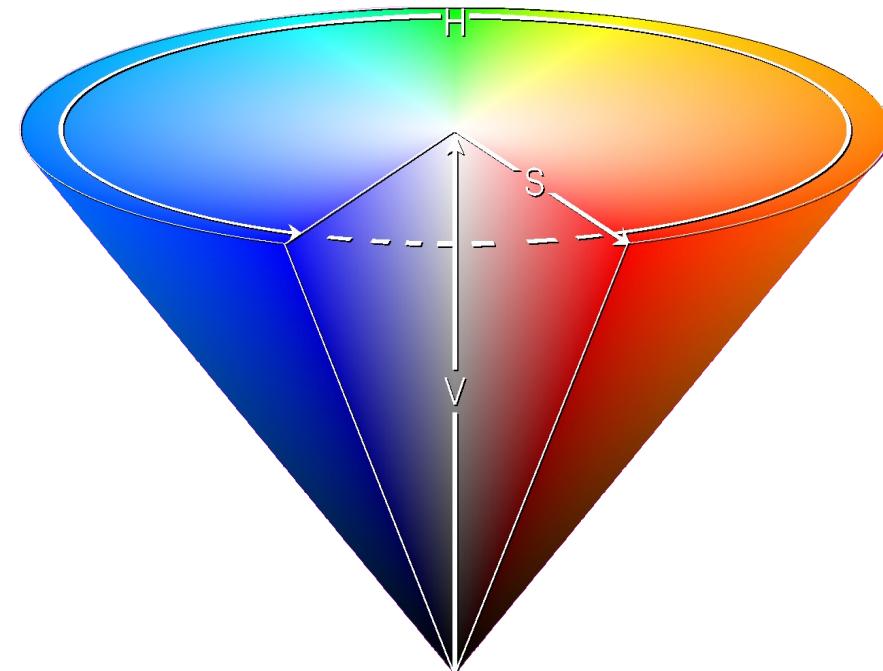
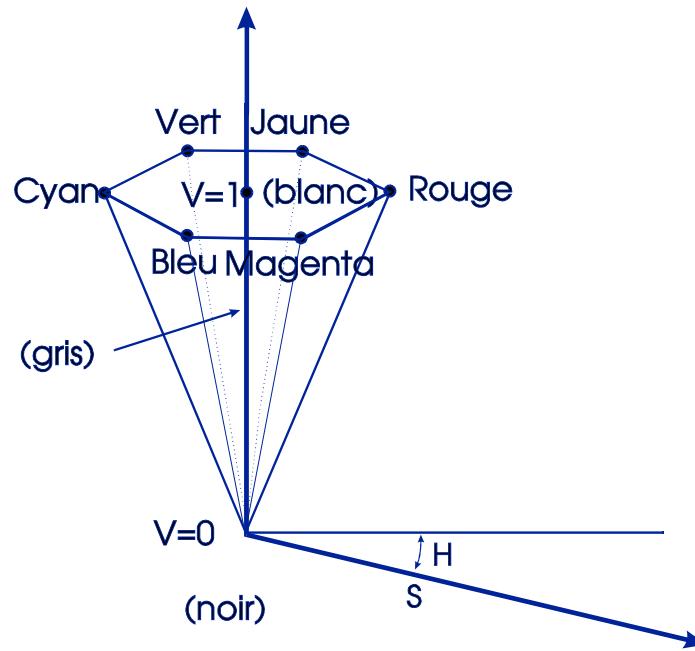
et le modèle HSV (/HSB) utilise:

- la teinte,
- la saturation
- brillance [*value* ou *brightness*].



Modèles de couleur : modèle HSV

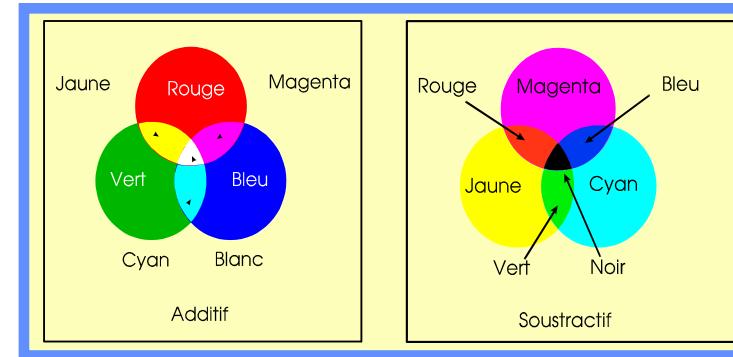
Le volume est:



Modèles de couleur : conversion

Conversion RGB → CMY

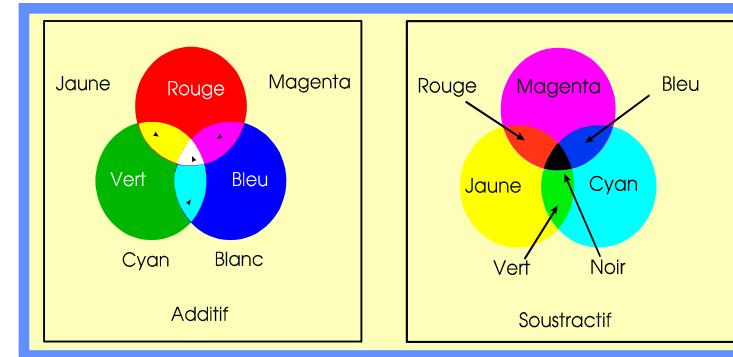
```
void RGBtoCMY( double coulR, double coulG, double coulB,
                double *coulC, double *coulM, double *coulY )
{
    *coulC = 1 - coulR;
    *coulM = 1 - coulG;
    *coulY = 1 - coulB;
    return;
}
```



Modèles de couleur : conversion

Conversion CMY → RGB

```
void CMYtoRGB( double coulC, double coulM, double coulY,  
                double *coulR, double *coulG, double *coulB )  
{  
    *coulR = 1 - coulC;  
    *coulG = 1 - coulM;  
    *coulB = 1 - coulY;  
    return;  
}
```



Modèles de couleur : conversion

Relation entre HSV-RGB

- N'importe quelle intensité [RED, GREEN, BLUE] peut être produite pas la somme d'une couleur pure (Hue) et blanc
- Une composante blanche : Proportion égale pour R, G, B
 - ▶ $\text{RED}_{\text{Blanc}} = \text{GREEN}_{\text{Blanc}} = \text{BLUE}_{\text{Blanc}} = \text{Min}(\text{RED}, \text{GREEN}, \text{BLUE})$
- Couleur pure est un mélange d'au plus 2 couleurs primaires

$$\text{RED}_{\text{Hue}} = \text{RED} - \text{Min}(\text{RED}, \text{GREEN}, \text{BLUE})$$

$$\text{GREEN}_{\text{Hue}} = \text{GREEN} - \text{Min}(\text{RED}, \text{GREEN}, \text{BLUE})$$

$$\text{BLUE}_{\text{Hue}} = \text{BLUE} - \text{Min}(\text{RED}, \text{GREEN}, \text{BLUE})$$

Au moins une de ces composantes est zéro

- Intuitivement

$$\text{Hue}^\circ = \frac{120^\circ \text{GREEN}}{\text{RED}_{\text{Hue}} + \text{GREEN}_{\text{Hue}}}$$

Modèles de couleur : conversion

$$\text{Saturation} = \frac{\text{Max}(RED, GREEN, BLUE) - \text{Min}(RED, GREEN, BLUE)}{\text{Max}(RED, GREEN, BLUE)}$$

$$\text{Value} = \text{Max}(RED, GREEN, BLUE)$$

Modèles de couleur : conversion

Conversion RGB → HSV

```
void RGBtoHSV( double coulR, double coulG, double coulB,
                double *coulH, double *coulS, double *coulV )
{
    // trouver 'value'
    double min = coulR, max = coulR;

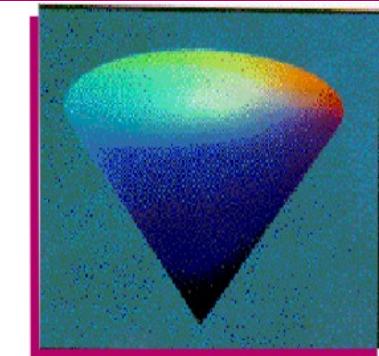
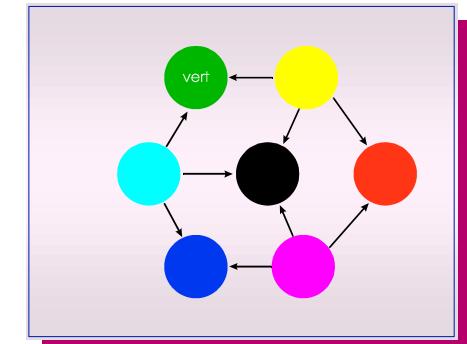
    if ( max < coulG ) max = coulG;
    else if ( min > coulG ) min = coulG;

    if ( max < coulB ) max = coulB;
    else if ( min > coulB ) min = coulB;

    *coulV = max;

    // trouver 'saturation'
    *coulS = max ? ( max - min ) / max : 0;

    // ...
}
```



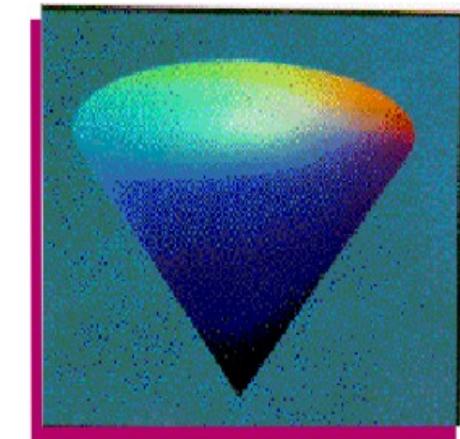
Modèles de couleur : conversion

```
// trouver 'hue'  
if ( *coulS ) // la saturation n'est pas zéro; trouver la teinte  
{  
    // cas chromatique  
    double rc = ( max - coulR ) / ( max - min ); // on mesure la distance  
    double gc = ( max - coulG ) / ( max - min ); // de la couleur  
    double bc = ( max - coulB ) / ( max - min ); // couleur au rouge  
  
    if ( coulR == max ) *coulH = bc - gc; // entre jaune et magenta  
    else if ( coulG == max ) *coulH = 2 + rc - bc; // entre cyan et jaune  
    else if ( coulB == max ) *coulH = 4 + gc - rc; // entre magenta et cyan  
  
    *coulH *= 60; // convertir en degrés  
    if ( *coulH < 0 ) *coulH += 360; // h ne doit pas être négatif  
}  
else  
    *coulH = 0; // cas achromatique: indéfini  
return;  
}
```

Modèles de couleur : conversion

Conversion HSV → RGB

```
void HSVtoRGB( double coulH, double coulS, double coulV,
                double *coulR, double *coulG, double *coulB )
{
    if ( coulS == 0 ) // cas achromatique
        *coulR = *coulG = *coulB = coulV;
    else // il y a une teinte; cas chromatique
    {
        double h = ( coulH == 360 ) ? 0 : coulH / 60;
        int i = (int) h;
        double f = h - i;
        double p = coulV * ( 1 - coulS );
        double q = coulV * ( 1 - coulS * f );
        double t = coulV * ( 1 - coulS * ( 1 - f ) );
    }
}
```



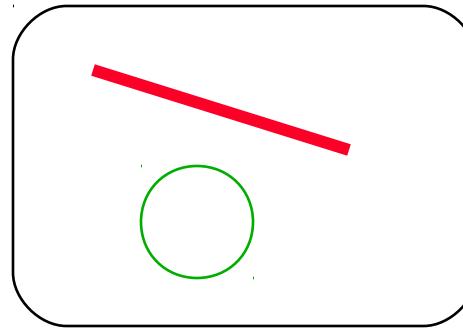
Modèles de couleur : conversion

```
switch ( i )
{
default:
    case 0: *coulR = coulV; *coulG = t;      *coulB = p;      break;
    case 1: *coulR = q;          *coulG = coulV; *coulB = p;      break;
    case 2: *coulR = p;          *coulG = coulV; *coulB = t;      break;
    case 3: *coulR = p;          *coulG = q;        *coulB = coulV; break;
    case 4: *coulR = t;          *coulG = p;        *coulB = coulV; break;
    case 5: *coulR = coulV;     *coulG = p;        *coulB = q;      break;
}
return;
}
```

Modèles d'affichage

Modèles d'affichage

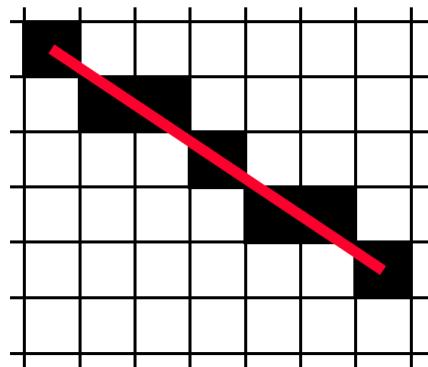
modèle trait (vectoriel)



- une image est décomposée en un ensemble de traits d'une certaine épaisseur et couleur
- le plus ancien
- utilisé p.e. pour écran à balayage cavalier, table traçante, logiciel de dessin
- mal adapté au rendu d'images réalistes
- la taille du modèle dépend de la complexité de l'image

Modèles d'affichage

modèle pixel



- une image est décomposée en une matrice (finie) de points (*pixels*) d'une certaine couleur
- utilisé par pratiquement tout matériel d'affichage moderne
- *résolution*: nombre de lignes \times nombre de colonnes
 - ➡ Écran: 1280×1024 , 1600×1200 , ...
 - ➡ imprimante laser: 6000×4800 , ...

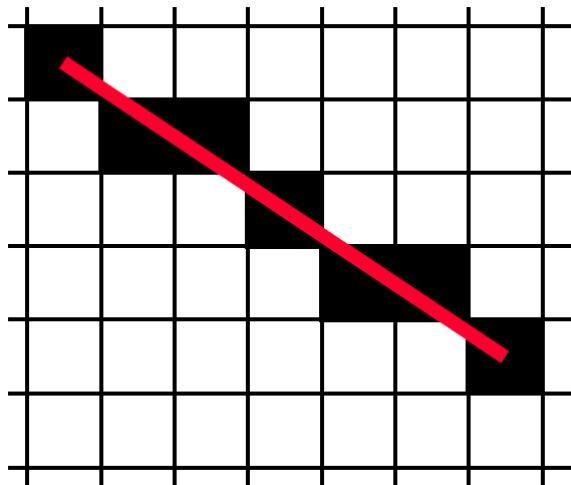
Modèles d'affichage

modèle pixel (suite)

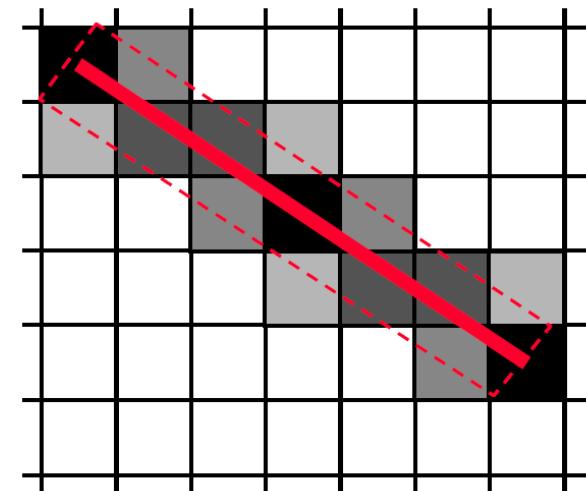
- bien adapté au rendu d'images réalistes
- la taille du modèle (\approx résolution) est indépendante de la complexité de l'image
- nécessite une étape intermédiaire de *conversion* puisqu'une image est souvent plus naturellement exprimée dans un ou l'autre des deux autres modèles
- la discréétisation d'une image peut causer des problèmes de *crénelage* (« aliasing »)

Anti-crénelage

Ligne en escalier

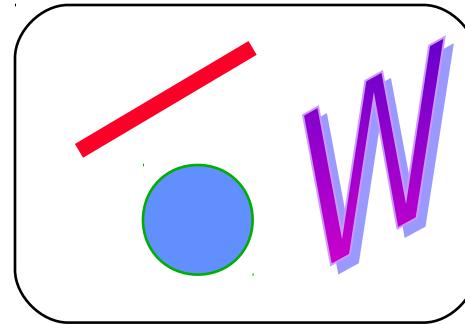


Ligne anti-crénelée



Modèles d'affichage

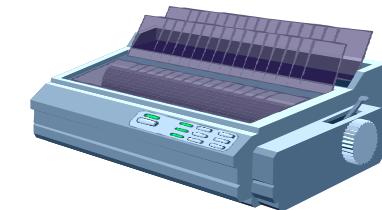
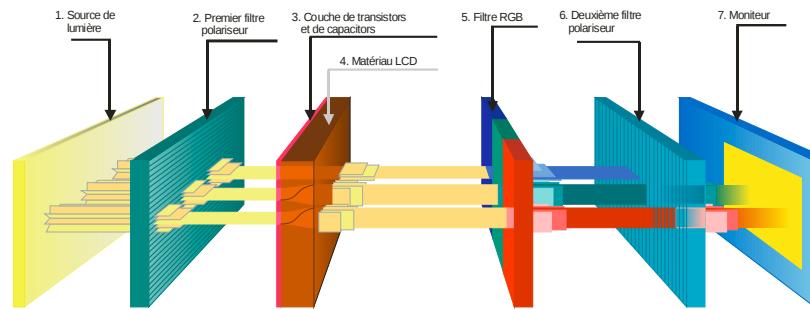
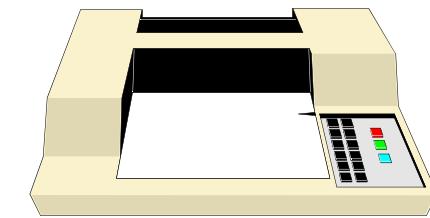
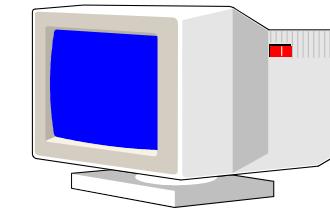
- ➊ modèle région (vectoriel)



- une image est décomposée en régions dont la frontière est définie par des traits et, pour lesquelles on spécifie la façon de les remplir
- utilisé p.e. par le langage PostScript pour les imprimantes laser
- la taille du modèle présente un bon compromis, surtout pour communiquer des données à du matériel d'affichage à haute résolution

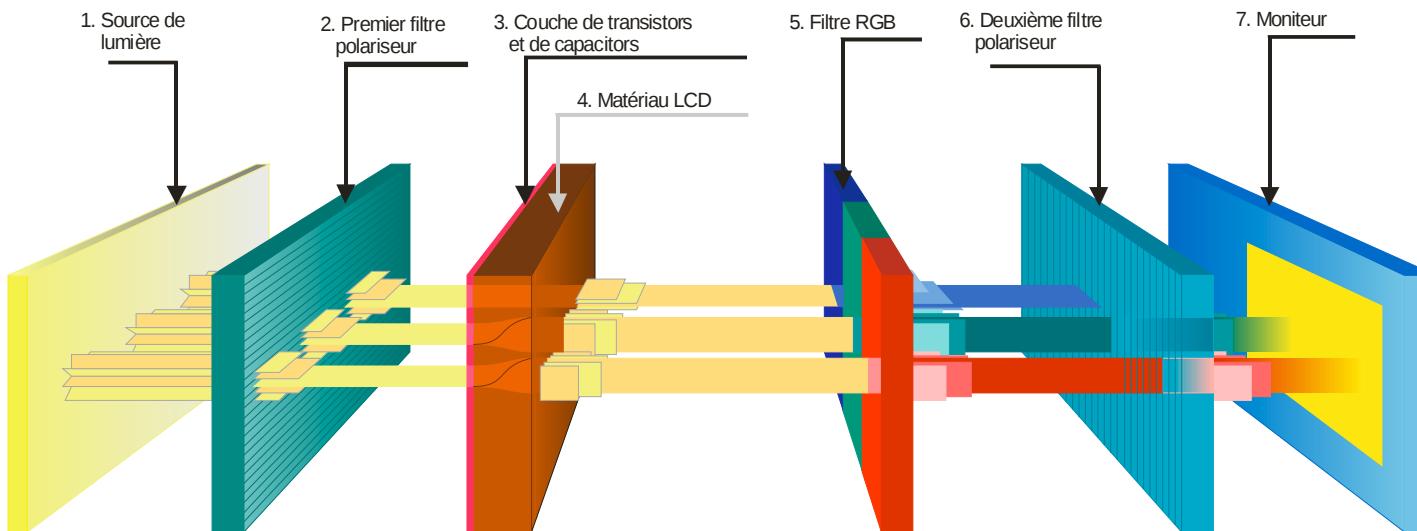
Matériel graphique

Matériel graphique



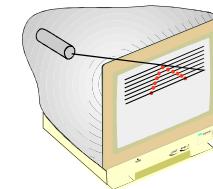
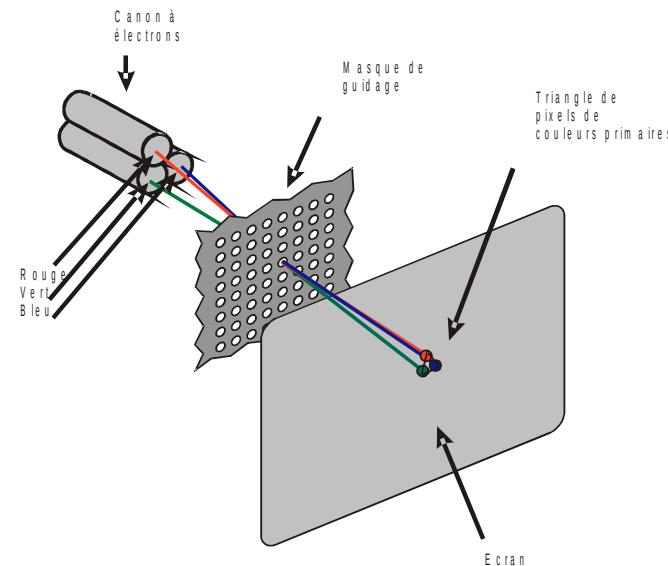
Matériel graphique

- affichage à cristaux liquides (LCD)
 - le choix moderne
 - ▶ écran plat
 - ▶ faible consommation d'énergie (idéal pour les ordinateurs portables)
 - basé sur un contrôle de l'opacité de cristaux
 - utilise le modèle pixel



Matériel graphique

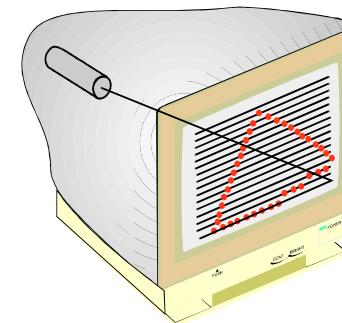
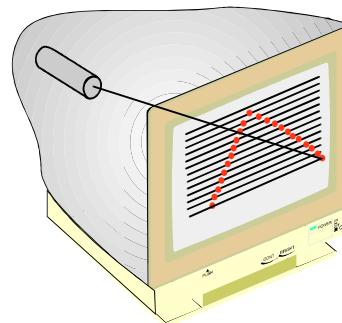
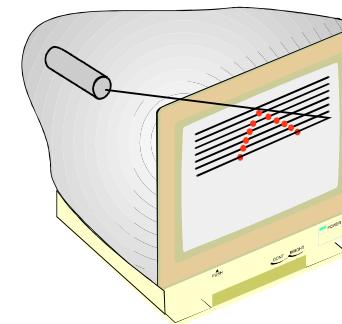
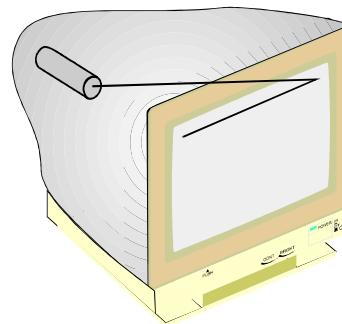
écran cathodique



- écran avec revêtement de phosphore et canon(s) d'électrons émetteur(s)
- basé sur l'intensité d'un faisceau d'électrons
- rafraîchissement périodique de l'affichage (60Hz et plus)

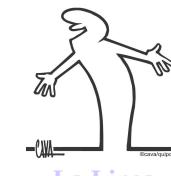
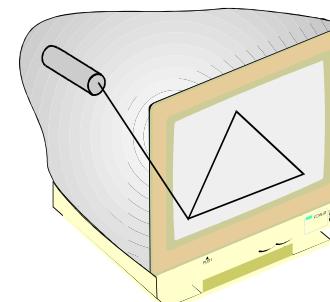
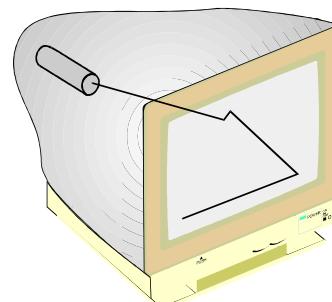
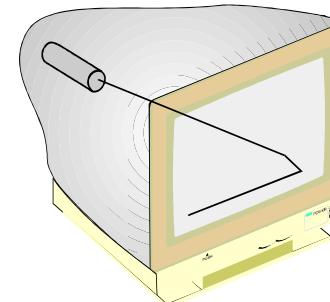
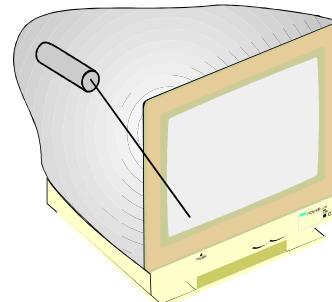
Matériel graphique

- ➊ écran cathodique
 - écran à balayage linéaire (« raster display »)
 - ➡ modèle pixel



Matériel graphique

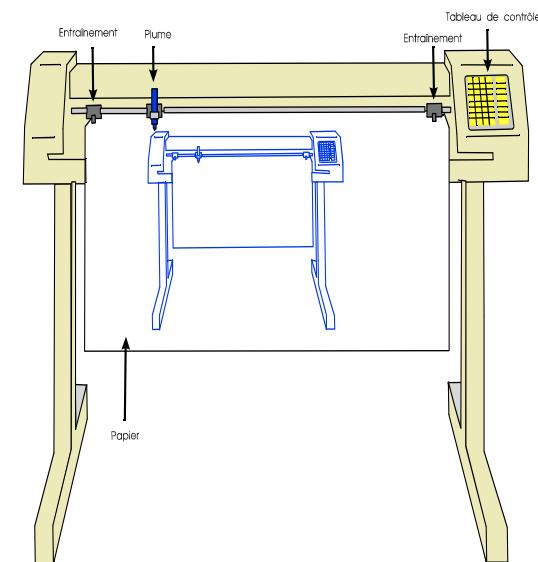
- ➊ écran cathodique
 - écran à balayage cavalier (« vector display »)
 - ➡ modèle trait



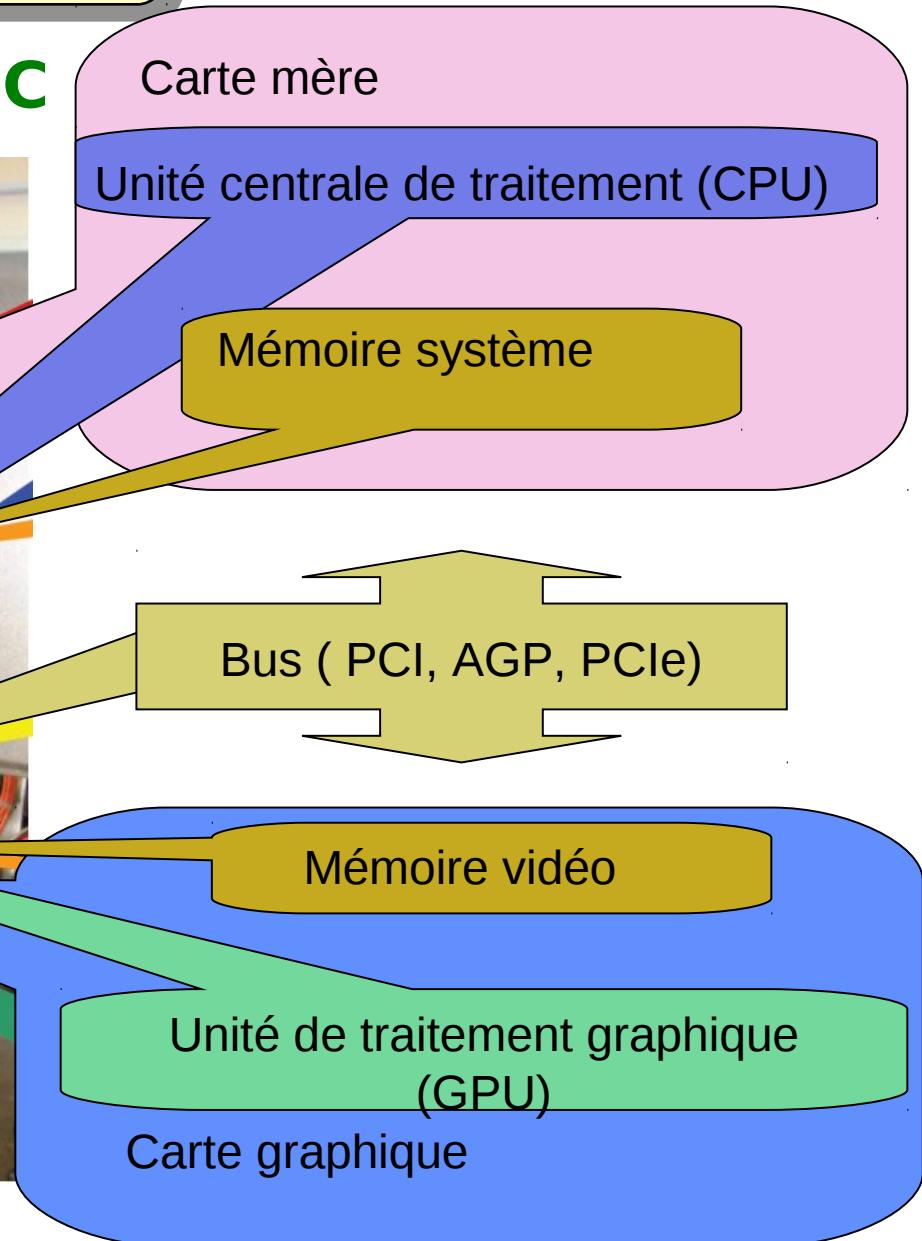
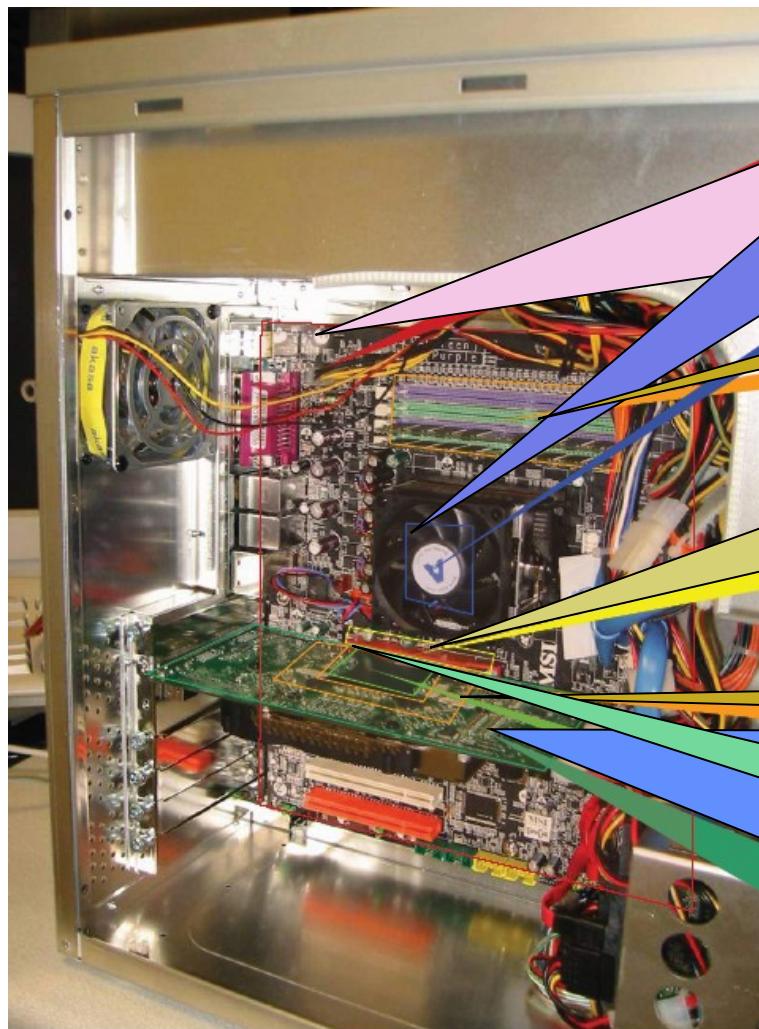
Matériel graphique

imprimantes

- table traçante: utilise modèle trait
- imprimante laser: possède mémoire de trame mais utilise modèle région et exceptionnellement modèle pixel
- imprimante à jet d'encre
- imprimante couleur

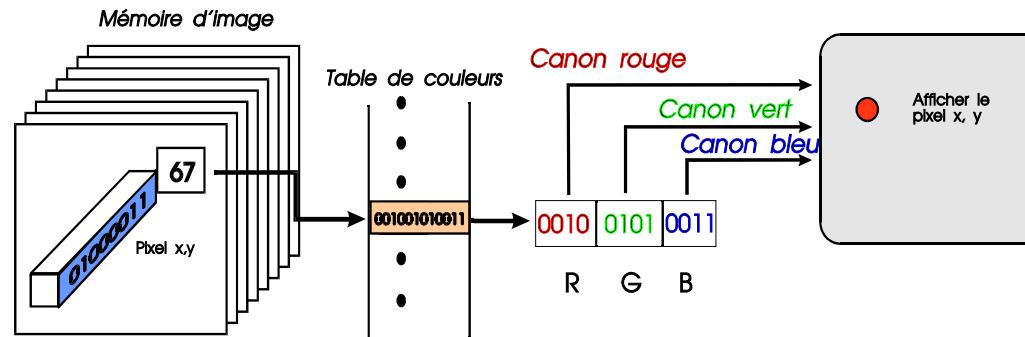


Architecture d'un PC



Matériel graphique

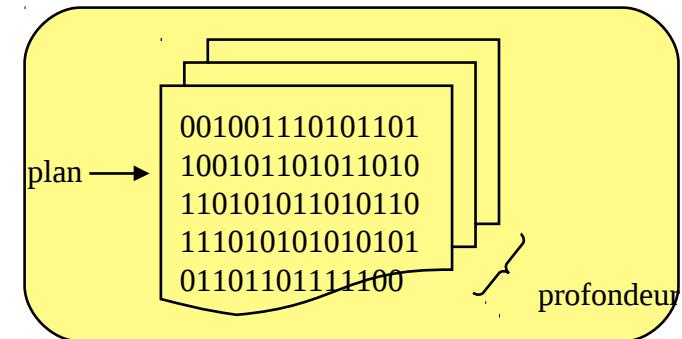
- ➊ mémoire de trame (« frame buffer », « refresh buffer »)



- propre au modèle pixel
- espace mémoire dont le contenu est une représentation logique de ce qui apparaît à l'écran
 - ▶ deux des dimensions de cette mémoire définissent la résolution spatiale de l'image et forment un *plan*
 - ▶ une troisième dimension appelée *profondeur* spécifie le nombre de bits utilisés pour représenter un élément de l'image (pixel)
- intermédiaire entre l'écran et la librairie graphique

Codification de la couleur RGB

- La troisième dimension de la mémoire de trame
- 3 types d'images (mode direct)
 - monochrome
 - ▶ 1 plan de « couleur »
 - ▶ tons de gris obtenus par effet de damier
 - ▶ p.e. imprimante laser
 - tons de gris
 - ▶ typiquement 8 plans de « couleur »
 - ▶ 8 bits \Rightarrow 256 tons de gris
 - couleur
 - ▶ typiquement 8 plans *par couleur additive primaire*
 - ▶ donc 24 plans de couleur, en général



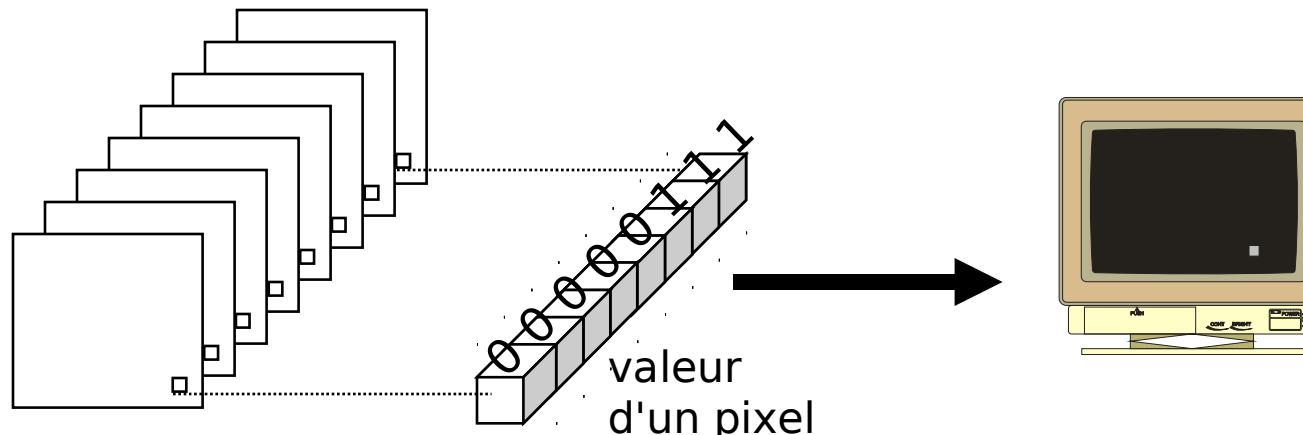
- 2 modes
 - ◆ direct
 - ◆ indexé

Codification de la couleur RGB mode direct

- monochrome

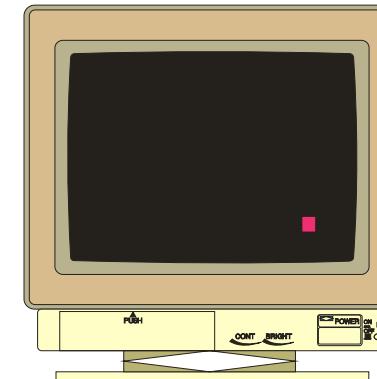
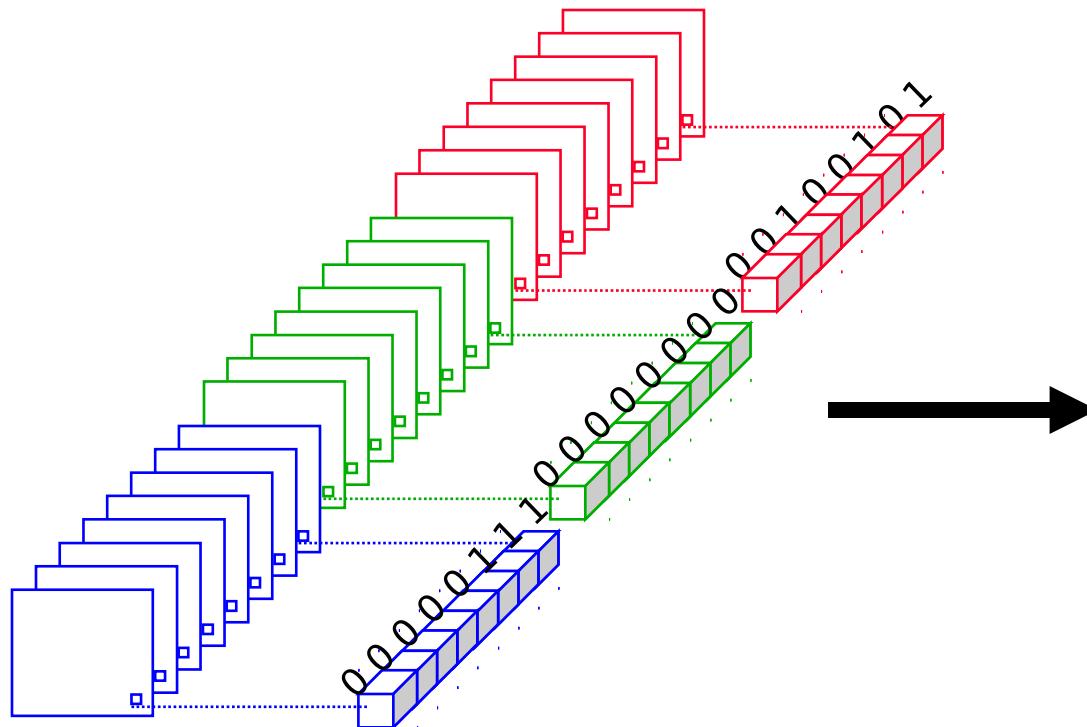


- tons de gris



Codification de la couleur RGB mode direct

couleur



Codification de la couleur RGB mode indexé

● monochrome, tons de gris

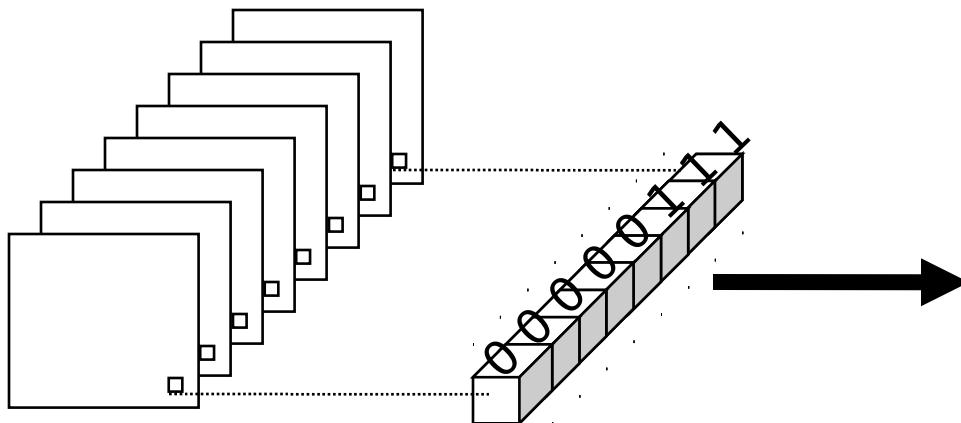
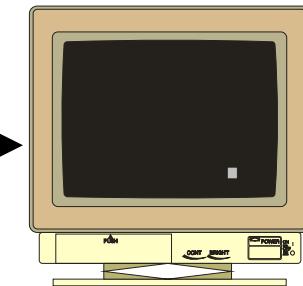
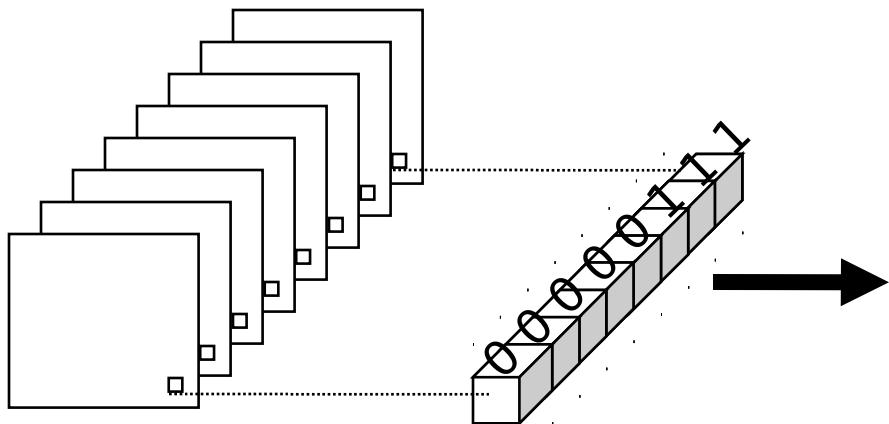


table de couleur

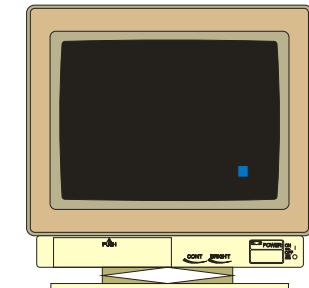
...
9
8
7
158
...



● couleur

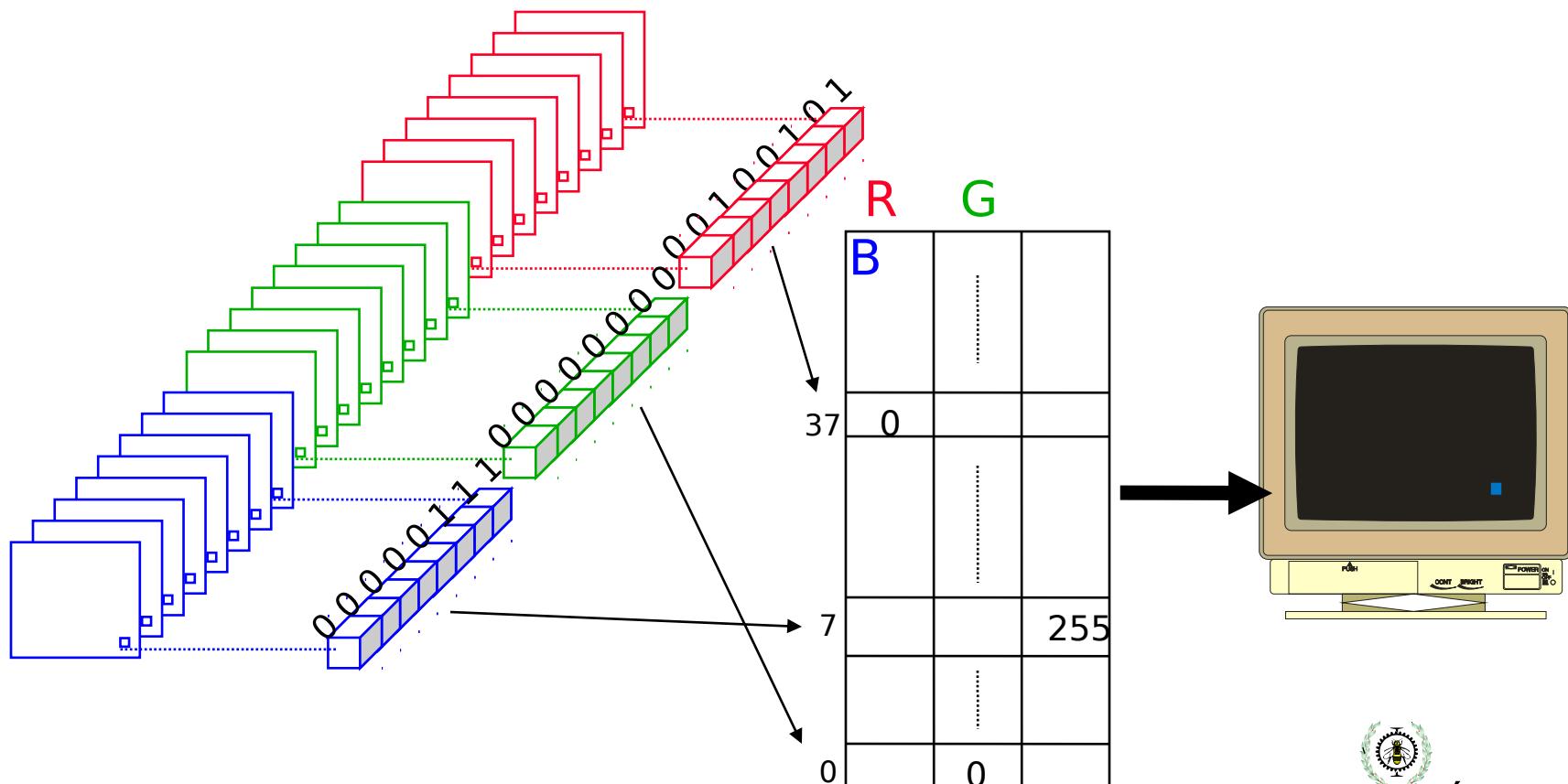


B	R	G
...
9		
8		
7	0	0
6	255	
5		



Codification de la couleur RGB mode indexé

- couleur, index décomposé



Couleur -- OpenGL

● Couleur -- mode indexé

```
void glIndex[s,i,f,d](TYPE Index);
void glIndex[s,i,f,d]v( const TYPE *Index);
où Index indique l'index dans la table de couleur
```

Index	Rouge	Vert	Bleu	
0	0	0	0	noir
1	1	1	1	blanc
2	1	0	0	rouge
3	0	1	0	vert
4	0	0	1	bleu
5	1	1	0	jaune
6	1	0	1	magenta
7	0	1	1	cyan

s -- short
 i -- int
 f -- float
 d -- double

Couleur -- OpenGL

● Couleur -- RGBA

```
void glColor3{b,s,i,f,d,ub,us,ui}(TYPE r,TYPE g,TYPE b );  
void glColor4{b,s,i,f,d,ub,us,ui}(TYPE r,TYPE g,TYPE b,TYPE a);  
void glColor{3 4}{b,s,i,f,d,ub,us,ui}v(const TYPE *v );
```

où

r rouge

g vert

b bleu

a alpha

v vecteur de 3 ou 4 valeurs

b -- byte
s -- short
i -- int
f -- float
d -- double
ub-- unsigned byte
us-- unsigned short
ui-- unsigned int

Vue d'ensemble (simplifiée!)

