

Images numériques et systèmes graphiques

M2-Pro O.S.A.E. 2008-2009 – J.F. Rabasse

- × Image numérique
- × Représentation des couleurs
- × Codage des images
- × Fichiers et formats
- × Techniques graphiques et bibliothèques
- × Le système X-Windows
- × Interfaces graphiques et toolkits

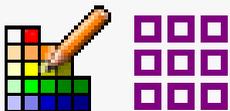
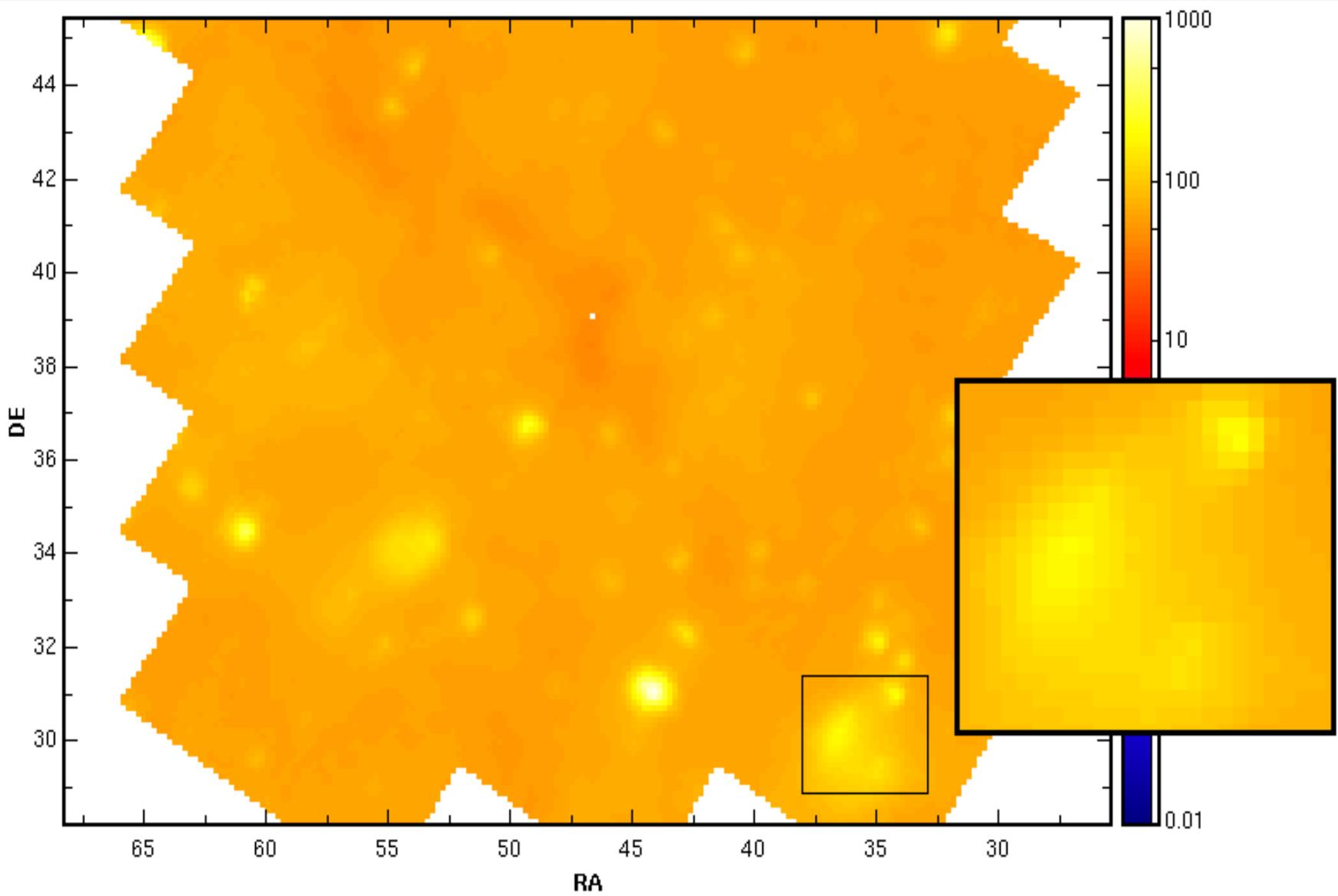
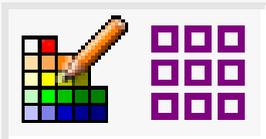


Image numérique



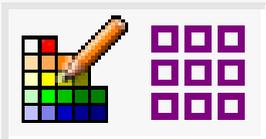


Une image numérique (*raster image*) est une représentation graphique 2D sous forme d'une matrice de *pixels* (*picture elements*).

Le *pixel* encode une information ponctuelle sous différentes implantations informatiques :

- pixel implanté sur 1 *bit*, codant une simple information noir ou blanc. On parle alors d'image *bitmap*, réservée à des graphiques au trait, gravures, alphabets, etc.
- pixel implanté sur 1 octet, codant une densité de 0 (noir) à 255 (blanc). On parle d'image en densités de gris.
- pixel implanté sur 2, 3 ou 4 octets, codant une information de couleur. Les formats graphiques les plus sophistiqués encodent également une information de *transparence*.
- pixel implanté sur 4 octets, codant un paramètre physique en format réel (e.g. images FITS). L'interprétation visuelle est à la charge de l'application.

La taille de codage des pixels s'exprime en *bpp* (*bits per pixel*).



Dimensions

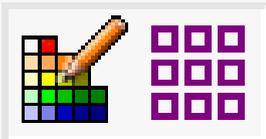
C'est la taille de la matrice image, en pixels par ligne et lignes. On parlera, par exemple, d'une image 1280x1024 (largeur x hauteur).

Définition

C'est un nombre de pixels rapporté à une unité de longueur, généralement exprimée en *dpi* (*dots per inch*).

La définition n'est PAS une caractéristique intrinsèque d'une image mais caractérise une reproduction matérielle de l'image, combinant un facteur d'agrandissement ou de réduction à la reproduction et les performances propres à un système de reproduction (ou d'acquisition, e.g. un scanner, on parle alors plutôt de résolution). On parlera de :

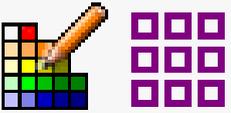
- périphériques basse définition, e.g. un écran vidéo à 75 dpi,
- périphériques moyenne définition, e.g. les imprimantes courantes à 300 ou 600 dpi,
- périphériques haute définition, e.g. les photocomposeuses d'imprimerie à 2400 dpi.



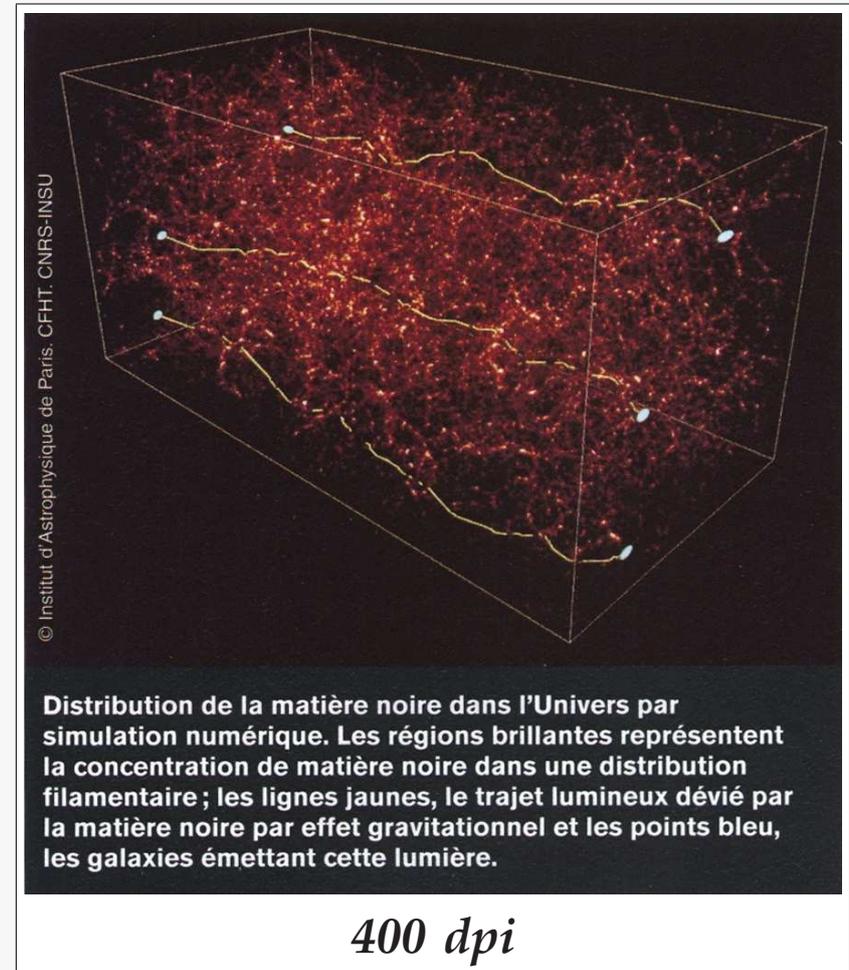
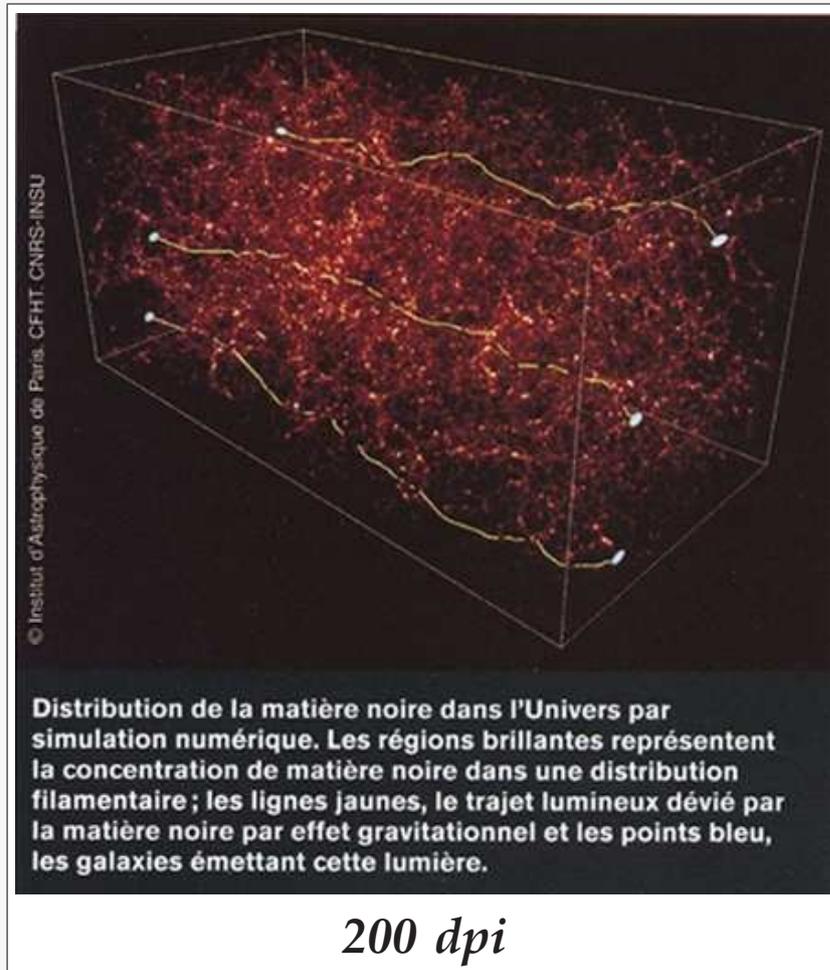
Résolution

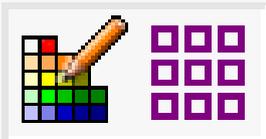
On parle habituellement de définition pour un appareil qui restitue une image (écran, imprimante) et de résolution pour un appareil (scanner, télescope) ou un organe (oeil) qui lit une image. La résolution est, en général, une propriété optique.

Un oeil d'humain (en bonne forme) a un pouvoir séparateur angulaire d'environ 30 secondes d'arc. À 40 cm (distance moyenne de lecture) l'oeil peut donc résoudre 0.06 mm, soit environ 450 dpi.



Exemples de restitutions :





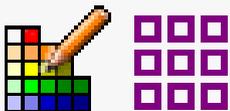
Abus de langage

Ils ne manquent pas, hélas. Ainsi on parle souvent de résolution, au lieu de définition. C'est un non sens. La résolution (ou pouvoir séparateur) est une caractéristique d'un appareil de mesure, ou optique, et non une caractéristique d'image. Une image est reproduite avec une certaine définition, à charge pour l'appareil optique ou l'oeil qui la regarde de la résoudre.

Pire, la résolution est parfois confondue avec la dimension. On peut trouver des publicités qui vantent des appareils photo numériques à "haute résolution" tout simplement parce que le capteur CCD qui équipe l'appareil est capable de construire des matrices image jusqu'à 3000x2000 !

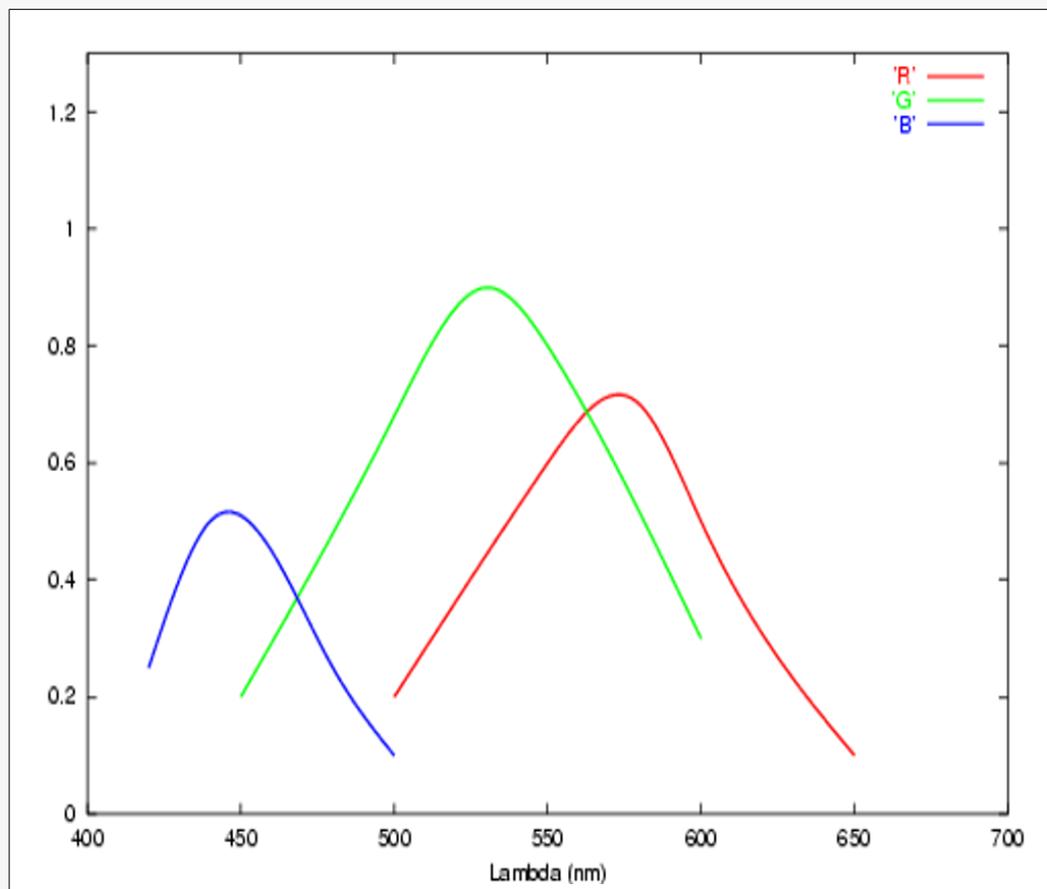
Autre abus (et barbarisme), on appelle image *pixelisée* une image reproduite avec une définition insuffisante qui fait que l'oeil distingue les pixels sous forme de petits carrés.

De fait, toute image numérique est *pixelisée* par définition.



Représentation des couleurs

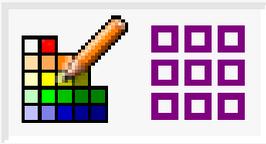
La rétine d'un oeil humain normal (non daltonien) comporte trois types de récepteurs (cônes) dont les courbes de sensibilité présentent leur maximum respectivement dans le bleu (vers 440 nm), le vert (vers 530 nm) ou dans le rouge (vers 580 nm).



On saura donc reproduire une couleur, a priori quelconque, à partir de trois composantes monochromatiques (ou presque) centrées dans le bleu, le vert, le rouge, et mélangées en proportions adéquates.

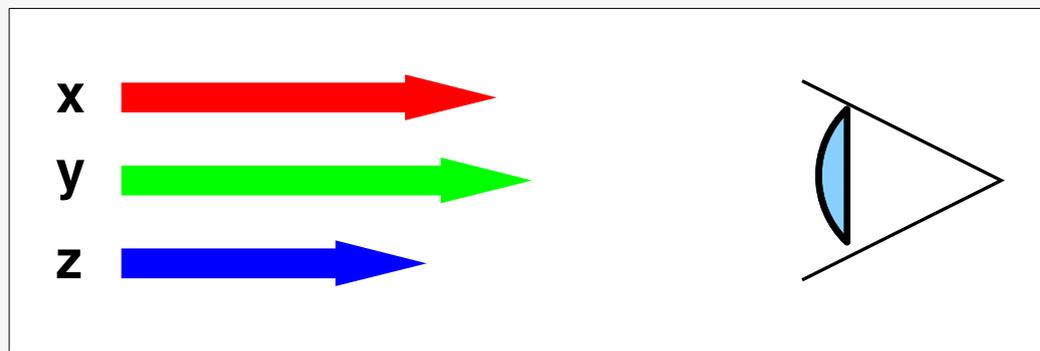
L'information de *couleur* finale est reconstruite par le cerveau à partir des trois stimuli.

(À noter que les maxima de sensibilité diffèrent pour les trois types de récepteurs.)



Synthèse additive

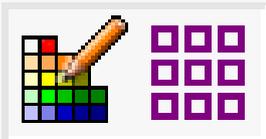
Appelée système *RGB* (*Red, Green, Blue*). C'est la technique de reproduction des couleurs utilisée par les systèmes qui génèrent eux-mêmes la lumière (tubes cathodiques d'écrans d'ordinateurs ou de téléviseurs).



Les points de l'image sont émis sous forme d'une combinaison

$$x.R + y.G + z.B$$

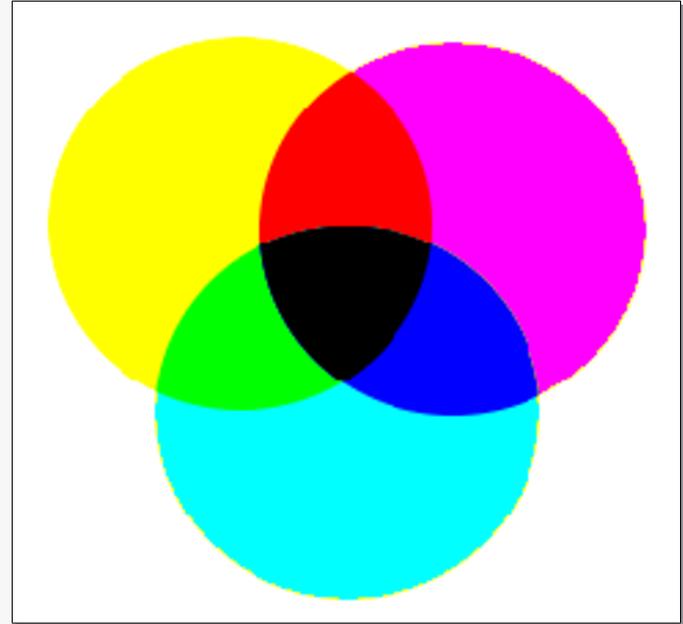
À noter que ce système encode simultanément la couleur (proportions relatives des composantes) et l'intensité (somme des composantes).



Synthèse soustractive

Ou système *YMC* (*Yellow, Magenta, Cyan*). C'est la technique de reproduction des couleurs utilisée par les systèmes qui modifient une lumière de référence - de préférence blanche - par réflexion (tirage photographique ou imprimerie) ou par transmission (photo diapositive ou transparent de rétro-projection).

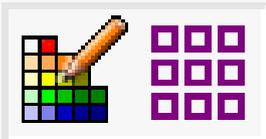
On utilise trois pigments *complémentaires* des trois primaires; un pigment jaune absorbant le bleu, un pigment magenta absorbant le vert, un pigment cyan absorbant le rouge.



En pratique, comme il est très difficile d'élaborer des pigments parfaits, le noir est en général recréé par un quatrième pigment dédié.

C'est la *quadrichromie* en imprimerie, ou le système dit *YMCK* (*Yellow, Magenta, Cyan, black*) des imprimantes couleurs à technologie laser ou jet d'encre.

A noter que le rendu des couleurs, en synthèse soustractive, dépend de la composition spectrale de la lumière incidente.



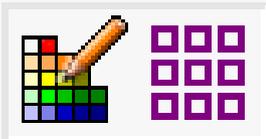
Systemes dérivés

Les systèmes de codage *RGB* ou *YMC*, s'ils correspondent à une réalité physiologique (excitation des récepteurs rétiniens), ont l'inconvénient de masquer des notions intuitives comme la teinte (composition spectrale), la saturation (mélange au blanc par ajout ou retrait d'une constante à un triplet RGB), la luminosité (coefficient multiplicatif sur un triplet RGB).

L'être humain généralise la notion de couleur bien au delà de la simple composition spectrale. Ainsi le *rose* n'est pas, à proprement parler, une couleur. C'est un *rouge* désaturé par apport de blanc. De même le *marron* n'est pas une couleur mais un *orange* sombre.

Dans les métiers des arts graphiques, le système le plus utilisé est le système *HSB* (*Hue, Saturation, Brightness*) qui permet de définir séparément la teinte d'une couleur, sa saturation, son intensité lumineuse.

C'est un système de codage auxiliaire, plus parlant pour l'artiste mais sans réalité physique ou physiologique. Les logiciels orientés arts graphiques disposent des conversions *HSB* vers *RGB* ou vers *YMC*, *YMCK*.



Images TrueColors

L'image numérique en couleurs utilise toujours le système additif *RGB*. (Le *YMC* est dépendant des conditions d'observation.)

Dans une image *TrueColors* chaque pixel de la matrice image code les trois composantes *R*, *G*, *B* du point.

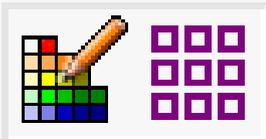
Dans une image 16 bpp chaque composante est codée sur 5 bits. Dans une image 24 bpp chaque composante est codée sur un octet.

L'image *TrueColors* est essentiellement utilisée en photographie numérique (rendu spectral dans le *visible*, recherche de fidélité).



Certains environnements ou applications orientées arts graphiques utilisent 32 bits par pixel. Trois octets codent les composantes *RGB*, le dernier octet code une information de transparence (*Alpha channel*).

On parle alors de système *RGBA*.



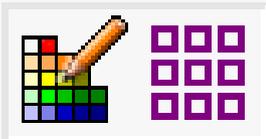
Images PseudoColors

Dans les images numériques en *fausses couleurs*, les pixels ne codent plus la couleur du point mais un index dans une table. C'est cette table, appelée *Color Lookup Table* ou *Palette*, qui contient les définitions *RGB*, ou *RGBA*, des couleurs.

En *PseudoColors* on sera limité dans le nombre de couleurs possibles par la taille de pixel utilisée (256 couleurs maximum en 8 bpp, 65536 en 16 bpp), mais on pourra avoir des définitions de couleurs de qualité convenable, sur 24 ou 32 bits. Les images et systèmes graphiques 8 bpp utilisent exclusivement un codage *PseudoColors*. Un octet ne permet pas de coder de manière satisfaisante une définition *RGB*.

Le codage *PseudoColors* est très utilisé en imagerie technique et scientifique, là où la notion de *couleur* n'a pas forcément de sens (vraies couleurs d'une image infrarouge ? vraies couleurs d'une image médicale obtenue par scanner ?).

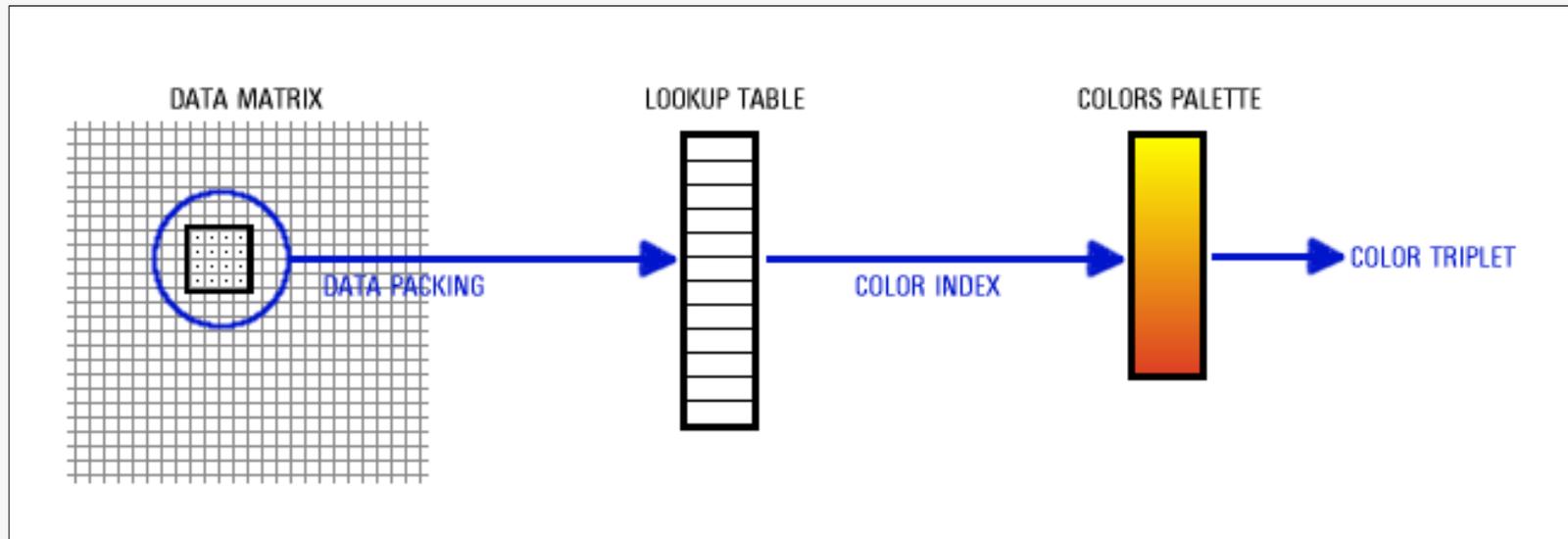
On peut également modifier le rendu visuel de l'image par simple modification de la palette.

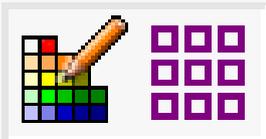


Images scientifiques

Par extension on appelle image scientifique une matrice de données à priori quelconque (grandeurs physique, valeurs en flottant) que l'on peut représenter graphiquement. La représentation (nécessairement pseudocolors) est faite en deux temps :

- Encodage de la grandeur physique en index, via une fonction de compactage (si nécessaire) et une table de choix (LUT ou Lookup Table).
- Sélection de couleurs dans une palette à partir de l'index.



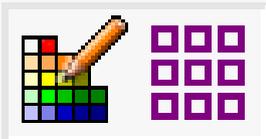


Représentations 3D

Les images *PseudoColors* sont utilisées chaque fois que l'on voudra représenter en plan un ensemble de données à trois composantes (X, Y, Z). Les grandeurs X et Y seront traduites en position colonne, ligne, dans la matrice image, la grandeur Z (qui peut être une valeur réelle) sera traduite en couleur symbolique via une table de *seuils*.

L'élaboration judicieuse d'une table de couleurs, ou palette, permet des restitutions de qualité, en fonction de la nature des données représentées. On utilise couramment :

- Division linéaire ou logarithmique de l'intervalle de variation de la grandeur Z , soit Z_{mini}, Z_{maxi} en N valeurs (par exemple 256 pour une image 8 bpp).
- Division statistique, chaque entrée de la table concerne environ le même nombre de points de l'image. Technique très utilisée pour des images astronomiques où la répartition statistique des Z est tout sauf uniforme.



Reproduction et sauvegarde des images

Il existe deux techniques (informatiques) de stockage d'images.

Langages graphiques

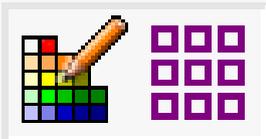
On sauve dans un fichier non pas l'image elle-même mais des commandes graphiques permettant de la reconstruire à volonté moyennant un logiciel d'interprétation.

Souvent le logiciel interpréteur est intégré à l'appareil de reproduction : traceurs HPGL, imprimantes PostScript.

L'avantage majeur des langages graphiques est que l'image est décrite géométriquement, de manière indépendante de la définition à la reproduction. On peut agrandir à volonté sans perte de qualité. L'inconvénient est de produire des fichiers souvent très volumineux.

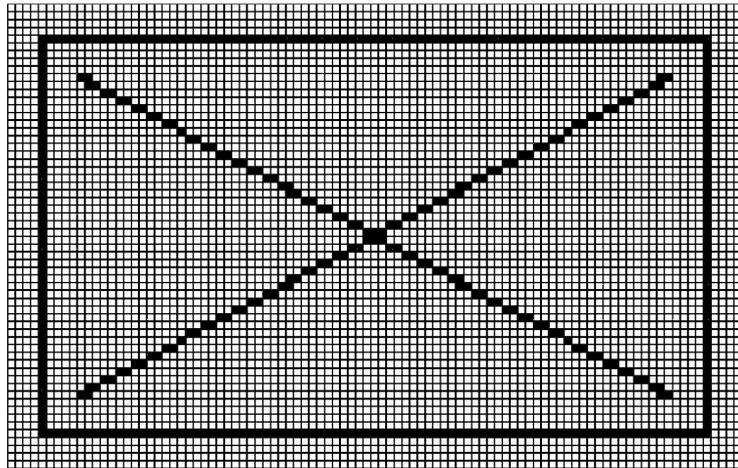
Fichiers images

Ces fichiers conservent directement une matrice image. On sera limité, pour les possibilités de reproduction, par les dimensions de la matrice et la définition du système de reproduction. La qualité se dégrade à mesure de l'agrandissement.



Reproduction et sauvegarde des images

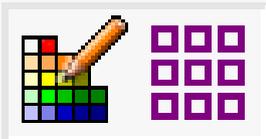
Des techniques de compression sont mises en oeuvre dans la plupart des formats de fichiers images. En général ces fichiers ont des volumes beaucoup plus faibles qu'avec les langages graphiques.



Graphique raster

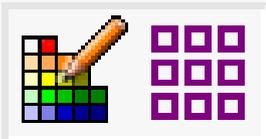
```
newpath
38 38 moveto 382 38 lineto
382 242 lineto 38 242 lineto
closepath stroke
newpath
58 58 moveto 362 222 lineto
stroke
newpath
58 222 moveto 362 58 lineto
stroke
showpage
```

Code PostScript



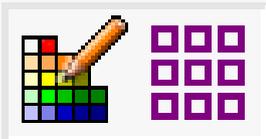
Principaux formats de fichiers images

- TIFF** Image PseudoColors, stockée avec sa palette et, parfois, des informations colorimétriques. Format très utilisé en imagerie scientifique, produisant des fichiers assez volumineux.
- GIF** Image PseudoColors 8 bpp, stockée avec compression. Le format GIF est très utilisé dans le monde des applications *web*, pour de petites icônes ou décorations, d'une part parce qu'il supporte une notion de couleur *absente* (GIFs transparents) et d'autre part parce qu'une extension du format permet le stockage d'une petite séquence d'images à afficher en boucle (G89 ou GIFs animés). Le décodage GIF est intégré dans les navigateurs *web*.
- L'inconvénient de ce format est qu'il a fait l'objet d'un dépôt légal, pour des raisons mercantiles, et que son utilisation est sujette à conditions.
- PNG** Image PseudoColors 8 bpp ou 16 bpp, très voisin de GIF mais libre ! (PNG a été créé comme une alternative au GIF commercial.) Supporte également les fonds transparents et est géré par les navigateurs *web*. Pas de possibilités d'animation.



Principaux formats de fichiers images

- JPEG** Image TrueColors 24 bpp, stockée avec compression. L'algorithme de compression est paramétrable (RLE à seuil) et on peut gagner énormément sur la compacité en acceptant une perte de qualité de l'image.
- C'est le format photographique par excellence et très utilisé dans les applications *web* (décodage intégré aux navigateurs) où une qualité moyenne suffit alors que la compacité est un atout majeur (temps de transmission/chargement).
- TARGA** Image TrueColors 32 bpp avec codage de transparence (alpha channel). TARGA est le format privilégié pour les applications imagerie et photo de haute qualité (logiciels d'images de synthèse, Ray-tracers, etc.) En dehors de ces domaines, il n'est guère utilisé à cause de la taille importante des fichiers.
- FITS** Très utilisé en astronomie, ce format est plus particulièrement destiné au stockage de *cartes* ou de *cubes*. Les matrices images sont des paramètres physiques, codés en entiers ou en réels. La représentation graphique en fausses couleurs et l'élaboration d'une table de couleurs sont à la charge de l'application qui exploite les données.



Principaux formats de fichiers images

Il existe d'autres formats, comme le BMP utilisé exclusivement dans le monde MS-Windows qui n'a aucun avantage et a la particularité de produire des fichiers monstrueusement gros. À oublier !

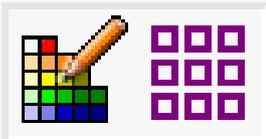
Formats étendus

Des formats de fichiers spécialisés ont été créés pour les applications vidéo, par exemple MPEG ou AVI. Le stockage est une simple suite d'images, en représentation TrueColors, mais des astuces existent pour limiter le volume qui peut devenir très important.

Par exemple, le stockage incrémental permet de ne coder que les pixels modifiés d'une image à l'autre de la séquence.

De plus, les formats vidéo stockent également une *bande son*, codée en PCM, et entrelacée avec les données images.

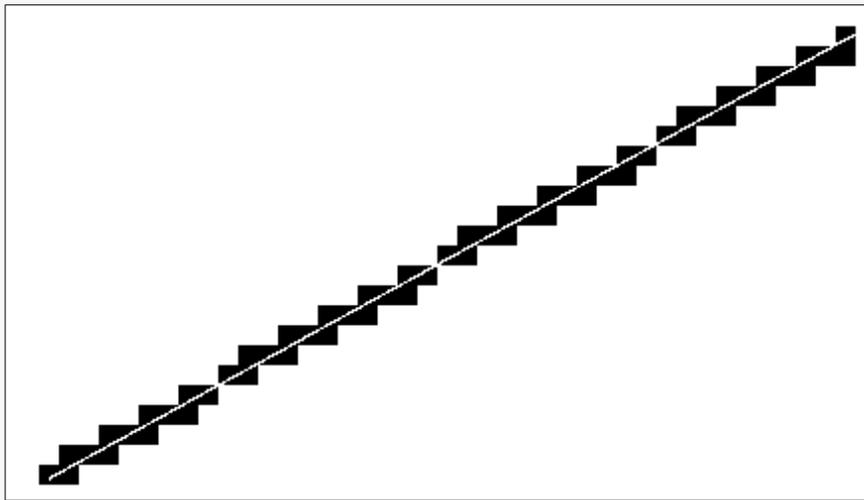
Le décodage vidéo est une opération lourde, souvent confiée à de l'électronique spécialisée (décodeurs MPEG intégrés sur les cartes vidéo haut de gamme).



Techniques graphiques et bibliothèques

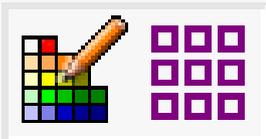
A cause de la nature discontinue d'une image numérique (matrice), les opérations graphiques les plus élémentaires mettent en oeuvre des algorithmes qui sont loin d'être triviaux.

Tracés linéiques



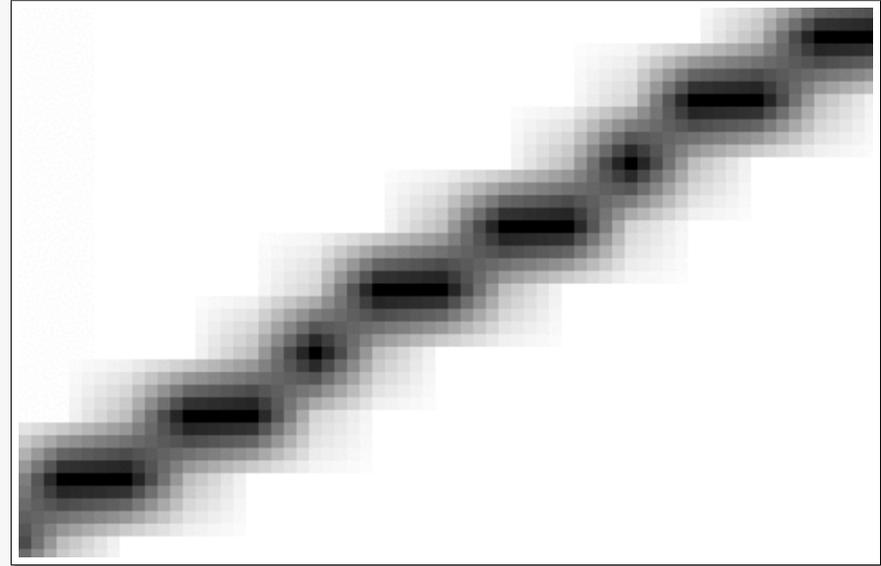
Le tracé d'un simple segment oblique suppose de déterminer la suite de pixels de la matrice image qui approchent au mieux le segment théorique.

Il existe des algorithmes spécialisés, optimisés pour tracer des segments, des arcs, des courbes (Bézier ou Splines), etc.



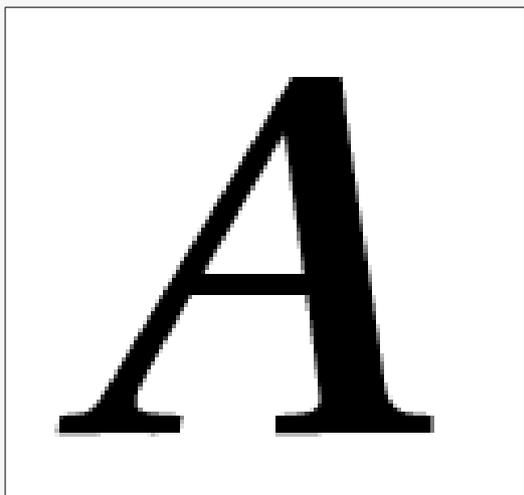
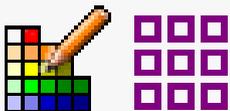
Techniques graphiques et librairies

De plus, des techniques particulières permettent d'améliorer l'aspect crénelé des tracés, par exemple l'*anti-aliasing*.



Coloriages

De même, il existe des algorithmes de *remplissages* (ou *floodfill*) de zones fermées, polygones, etc.

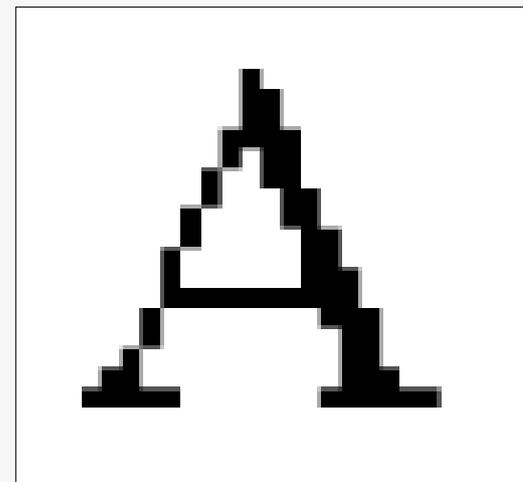


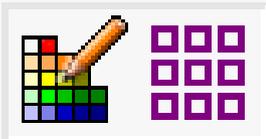
Les polices vectorielles utilisent des caractères définis géométriquement par coordonnées, équations, etc. La qualité de reproduction est excellente, en toutes tailles, et indépendante de la définition du périphérique de sortie.

Le traitement d'un texte est lourd (*dessin* des caractères un à un, avec éventuellement mise à l'échelle, rotation).

Les fontes bitmaps utilisent des caractères pré-numérisés dans différentes tailles. L'affichage est très rapide (simple transfert de pixels) mais la qualité est moyenne et les possibilités limitées (e.g. rotation de caractères par multiples de 90 degrés).

Les fontes bitmaps sont réservées à la vidéo.





Librairies et pilotes

Toute programmation graphique passe par l'utilisation de bibliothèques spécialisées qui proposent plusieurs dizaines d'appels élémentaires mettant en oeuvre les techniques ci-dessus : tracés de segments, de polygones, de courbes, écritures de textes, gestion des couleurs, etc.

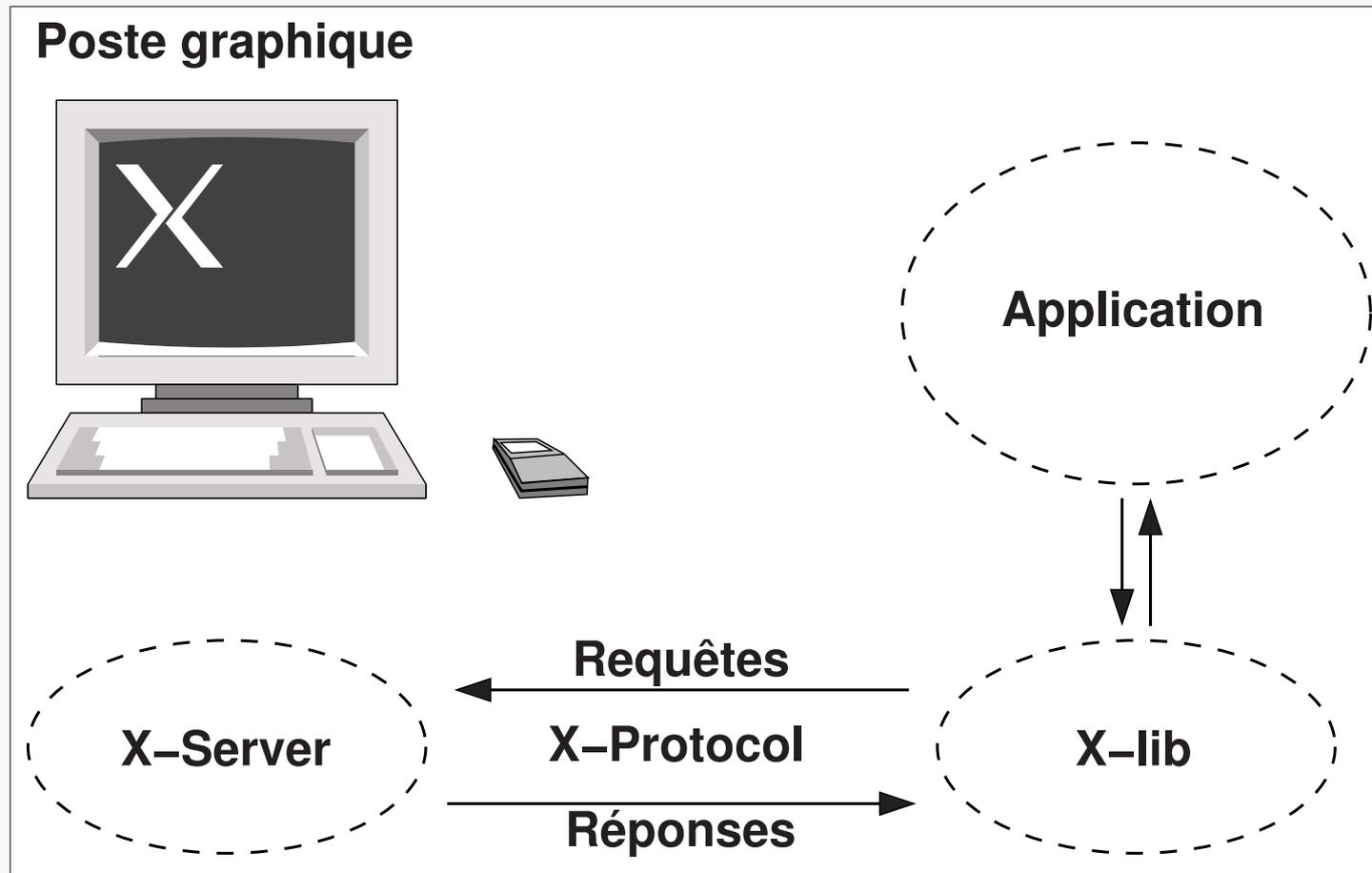
Beaucoup de bibliothèques orientées graphique technique et scientifique gèrent également des changements de coordonnées (conversion d'un espace en coordonnées flottantes vers la matrice de pixels). Par exemple, GKS, PqPlot, etc.

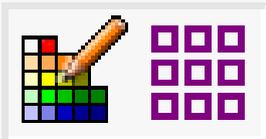
Enfin, une bibliothèque graphique digne de ce nom doit supporter plusieurs types de sorties, par l'intermédiaire de pilotes (ou *drivers*). La même application peut, par simple sélection d'un pilote, générer un même graphique vers un écran vidéo, vers un fichier PostScript, etc.



Le système X-Windows

X-Windows, ou encore X11, est un environnement graphique interactif fonctionnant sur un modèle client-serveur. C'est aujourd'hui le système graphique de référence du monde Unix.





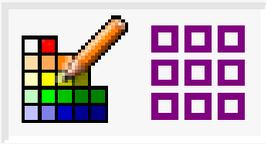
Le système X-Windows

Le logiciel serveur est installé sur un poste de travail graphique (écran, clavier, souris). Ce peut être une station de travail classique ou un terminal spécialisé (*terminal X*). Il alloue des ressources : fenêtres, fontes, tables de couleurs, etc., a des clients.

Les applications clientes s'interfaçent via une librairie locale, la X-lib, laquelle dialogue avec le serveur X (le *DISPLAY* dans la terminologie Unix) par un protocole réseau spécialisé. L'application est, de fait, indépendante du matériel graphique utilisé (carte vidéo, type de clavier, etc.).

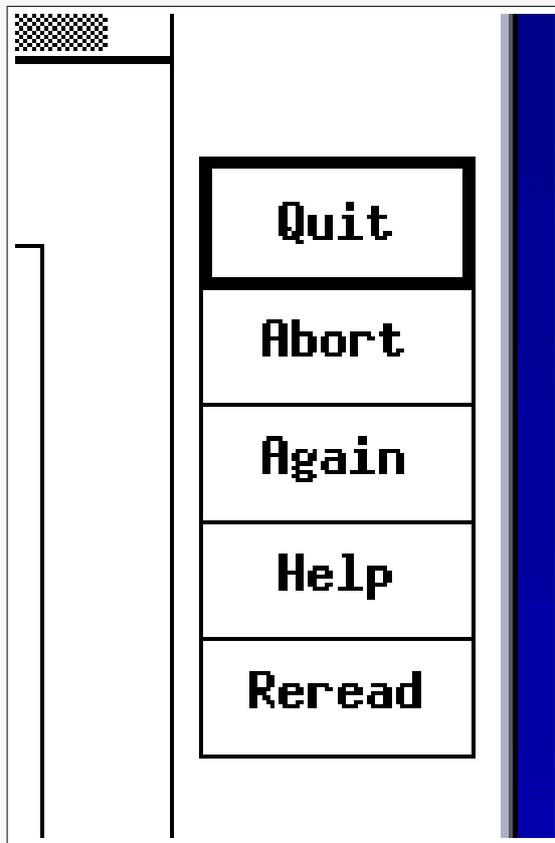
L'application envoie des requêtes graphiques, e.g. colorier une fenêtre, tracer un segment, écrire un texte, copier une image bitmap, etc.

Le serveur envoie à chacune des applications clientes les *événements* qui la concernent, e.g. déplacement de souris, frappe clavier, fenêtre masquée, etc.



Interfaces graphiques et toolkits

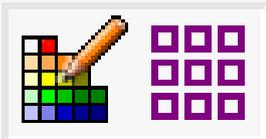
Au delà de l'utilisation strictement technique du graphique, tracés de courbes et autres, le développement des équipements graphiques, cartes vidéo, souris, a également conduit à transformer complètement les interfaces d'applications.



Les interfaces en mode texte ont laissé la place aux interfaces graphiques utilisateurs, dites GUI.

L'interface de programmation X11 (la X-lib) ne propose que des primitives graphiques de très bas niveau, tracés au trait, écritures.

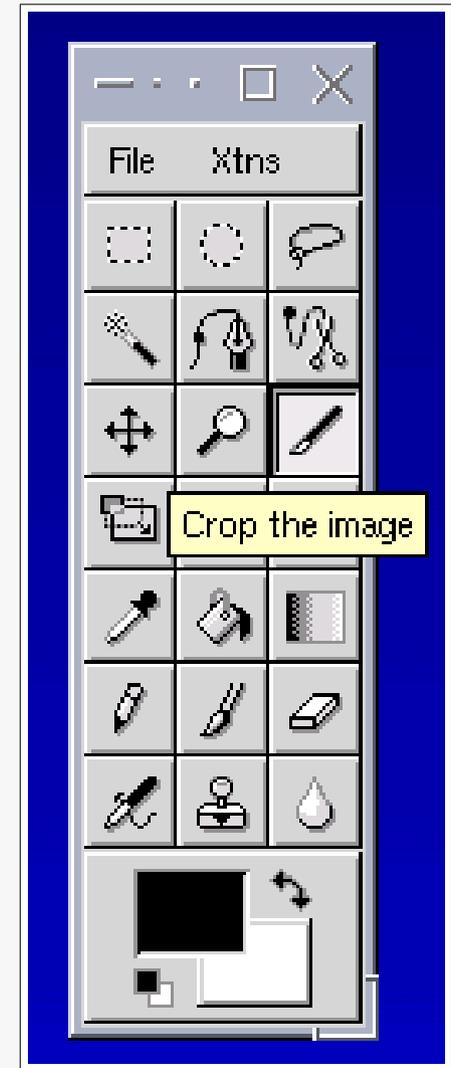
On peut utiliser directement ces appels pour programmer des interfaces utilisateurs, menus, boutons, etc., sous réserve d'accepter un *look* assez spartiate.

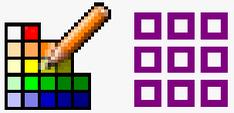


Interfaces graphiques et toolkits

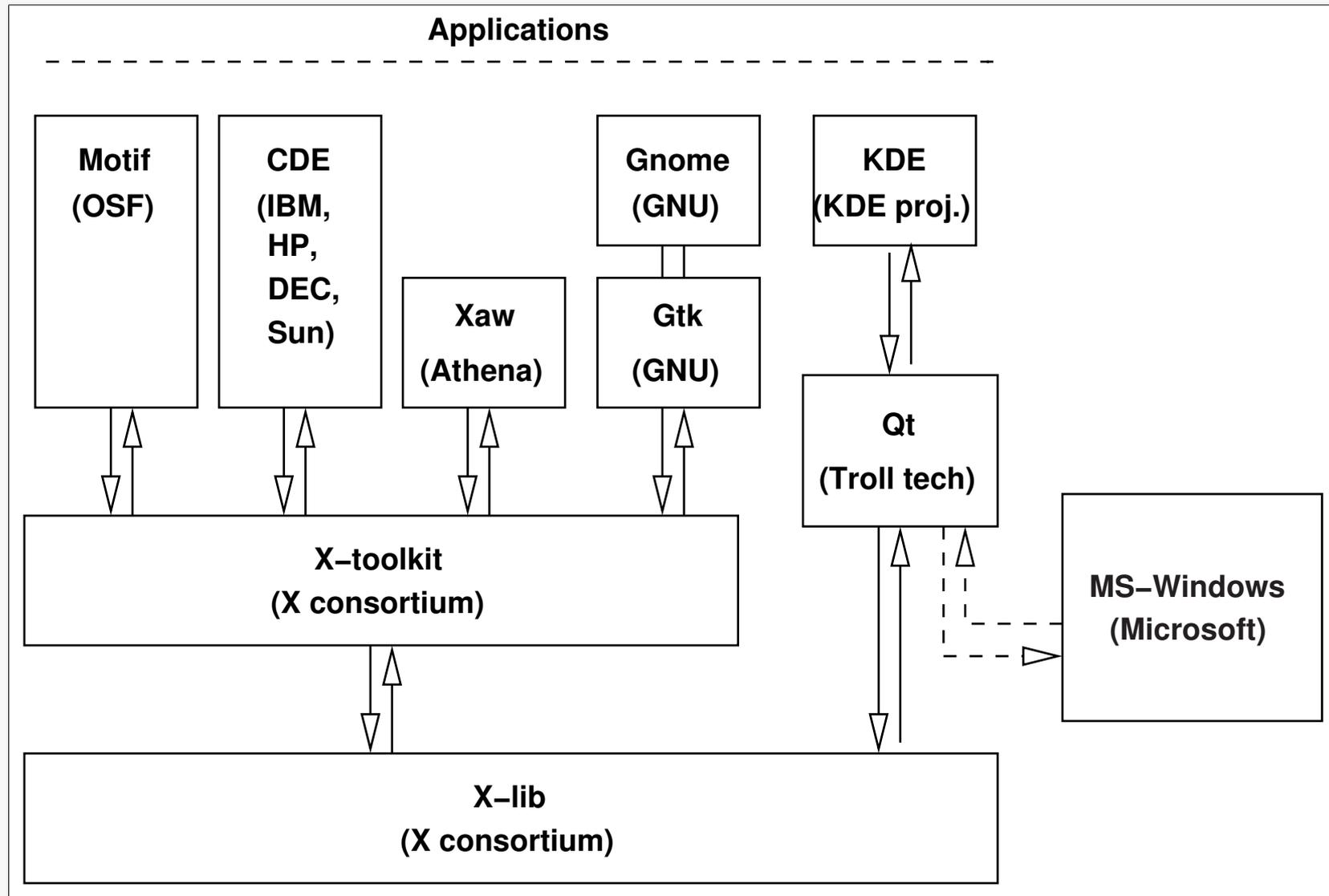
On peut souhaiter des interfaces plus sophistiquées au niveau graphique et pour la gestion des événements utilisateurs (ouverture de menus déroulants, boutons poussoirs, barres de défilement, bulles d'aide, etc.) à condition d'investir dans un énorme travail de programmation.

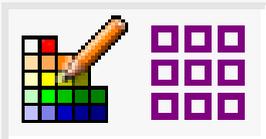
Les *toolkits* sont des bibliothèques spécialisées, élaborées au dessus de la X-lib, et qui implémentent des composants d'interface utilisateurs : menu, pushbutton, scrollbar, sous une forme déjà pré-programmée.





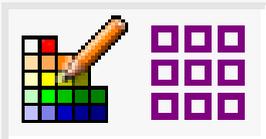
Principaux toolkits





Principaux toolkits

Le consortium X a été le premier à développer un *toolkit*, le X-toolkit, orienté objets et implémentant des composants d'interfaces graphiques sophistiqués. Ce toolkit a servi de base pour de nombreuses autres bibliothèques. A noter le cas particulier du toolkit Qt, de Troll technologies, qui est le seul pouvant s'interfacer directement avec Unix/X11, Mac OS X et MS-Windows, rendant ainsi les applications beaucoup plus facilement portables (code source compatible).



Gestionnaires

En plus des bibliothèques destinées à la programmation des applications, les toolkits les plus importants (Motif, CDE, KDE, Gnome) disposent de logiciels spécialisés dans l'intégration des applications graphiques : *windows managers* ou *session managers*.

Ces gestionnaires uniformisent l'aspect et l'ergonomie des applications (cadres de fenêtres, menus système) et en facilitent la gestion par l'utilisateur (mises en icônes, barres de tâches, gestion de bureaux, etc.)