# Traitement d'images*

David Monniaux
monniaux@clipper.ens.fr

23 juin 1998

## 1 Qu'est-ce qu'une image sur ordinateur ?

### 1.1 Images matricielles (ou *bitmap*)

#### 1.1.1 Définition

Les images les plus couramment manipulées sur ordinateur sont constituées de grilles (ou **matrices**) de points, appelés **pixels**, ayant chacun une certaine couleur ; elles sont en premier lieu déterminées par leur taille en points (*largeur* × *hauteur*).

#### 1.1.2 Systèmes d'impression ou d'affichage

☞ *Dans cette partie, de nombreuses données, en particulier numériques, vont être données à titre d'information. Vous n'avez pas besoin de les connaître pour manipuler des images !*

La taille de l'image est bien sûr à considérer relativement à la précision, ou **résolution** du système d'affichage ou d'impression considéré. Cette résolution est le plus souvent indiquée en points par pouce (*dots per inch*, dpi). Les systèmes d'impression courants sont :

**écrans** typiquement, 75 ou 100 dpi ; les tailles en points sont souvent, sur PC, $640 \times 480$, $800 \times 600$, $1024 \times 768$ ; sur les Sun, c'est $1152 \times 900$ ;

**imprimantes**

**à aiguilles** elles comportent une tête d'impression qui se déplace horizontalement sur le papier, formée d'un certain nombre d'aiguilles disposées verticalement (typiquement, 9 ou 24) et appuyant sur un ruban

*Ce document a été réalisé avec HyperLaTeX et les figures ont été faites avec XFig. Les transformations d'images ont été effectuées avec XV et NetPBM. Une version hypertexte est disponible à http://www.eleves.ens.fr:8080/tuteurs/images.

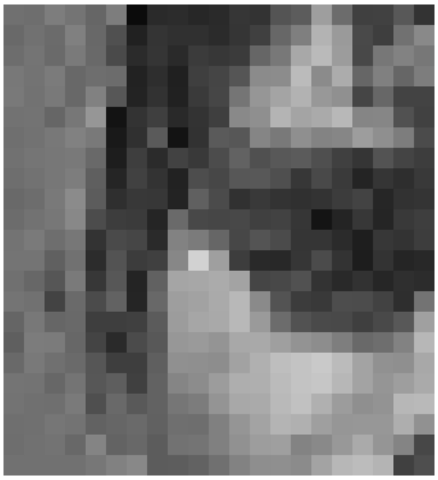

FIG. 1: Une *bitmap* agrandie (c'est un bout d'une photo scannée).

(semblable à celui des machines à écrire) ; elles sont assez bruyantes et le résultat est assez médiocre ; une résolution typique est 240 dpi ; elles sont parfois appelées **imprimantes matricielles**, par opposition aux **imprimantes à marguerite**, qui fonctionnaient comme les machines à écrire à marguerite (outil sur lequel les caractères sont gravés et viennent frapper un ruban), qui ne sont plus guère utilisées de nos jour ;

**imprimantes laser** par un procédé semblable à celui des photocopieuses, elles fixent une poudre, le *toner*, sur le papier ; les résolutions typiques sont 300 et 600 dpi ; les imprimantes `lwsalles` (Salle S), `lwinfi` (Infirmatique) et `lwsallet` (Salle T) sont de ce type ;

**imprimantes à jet d'encre** moins coûteuses que les imprimantes laser, elles fonctionnent par envoi d'encre liquide par de petites buses ; bien que la qualité souffre quelquefois de la déformation du papier sous l'effet de l'encre ou des bavures, la qualité s'approche de celle des imprimantes laser, et il existe aujourd'hui des imprimantes à jet d'encre couleur peu onéreuses (résolution typique 300 dpi) ;

**photocomposeuses** elles sont à très haute résolution (1200, 2400 dpi...) et servent à préparer les films pour impression *offset* (méthode d'impression usuelle des journaux, magazines et livres)

Il faut également noter que, pour un travail fin, il est nécessaire d'adapter l'image au système d'impression considéré, certains ayant tendance, par exemple, à faire les traits plus gras que spécifié, ou à ne pas imprimer les déliés trop fins...

## 1.2 Tons de gris et couleur

### 1.2.1 Tons de gris

**Échelles de gris** Les images les plus simples sont **monochromes** : à chaque point est associée une couleur : soit blanc, soit noir. Il suffit donc de stocker pour chaque point un petit drapeau 0 ou 1 ; or, on vous a sûrement déjà dit que les ordinateurs calculaient en interne sur de tels petits drapeaux, appelés **bits**. Bref, une image monochrome est à 1 bit par pixel (1 bpp).

Les images en ton de gris contiennent pour chaque point une valeur dans une échelle de gris ; généralement, cette échelle comprend 256 niveaux de gris (0=noir, 255=blanc). Cette valeur vous semble sans doute étrange, mais c'est parce que c'est exactement l'échelle de gris qui peut tenir sur 8 bits par pixel (8 bpp) ($2^8 = 256$). Sur certains matériels, il n'y a que 6 bits de résolution ; on n'a alors que 64 tonalités disponibles ($2^6 = 64$).

Cela serait encore simple si nous voyions un gris à 120 comme deux fois moins intense qu'un gris à 240 ; mais le matériel (carte graphique, écran, yeux...) n'est pas aussi bien fait que cela ! On est donc quelquefois amené à appliquer une correction dite **gamma-correction**[1].

**Tramage** Afin de restituer les gris sur des dispositifs ne permettant pas d'en afficher directement mais qui ont une bonne résolution (les imprimantes, par exemple), on applique des méthodes de tramage : on va imprimer des points noir sur blanc de façon a ce qu'en moyenne cela fasse le gris demandé. Il y a deux grandes méthodes pour cela :

- des petits motifs réguliers (on parle alors de trame à 33%, 75%...) — vous pouvez en voir dans la presse en regardant de près ;
- la dispersion d'erreur — on met un point noir si le point est plutôt noir, un point blanc s'il est plutôt blanc et on disperse l'erreur commise sur les points alentours — les motifs obtenus sont alors irréguliers, bien qu'assez caractéristiques.

### 1.2.2 Couleur

**Modèle RVB** Comme vous pouvez le remarquer en vous approchant d'un téléviseur, les pixels sur un écran vidéo sont en fait constitués de la juxtaposition de trois petits pixels rouge, vert, bleu — sur un moniteur d'ordinateur, c'est beaucoup plus difficile de la remarquer, en raison de la résolution bien supérieure. Cette méthode d'affichage s'appelle **synthèse additive**, parce que la couleur affichée résulte de l'addition des trois couleurs élémentaires. Le blanc s'obtient ainsi en mettant les trois couleurs au maximum. Notons que l'œil humain est sensible, dans l'ordre, au vert, plus au rouge et au bleu. On aura donc quelquefois des systèmes ayant une

[1] Pour les plus curieux, cette correction est de prendre $x' = x^{1/\gamma}$ avec $x$ et $x'$ dans $[0,1]$.

meilleure qualité dans le vert que dans les autres composantes. Géométriquement, les couleurs forment alors un cube sur trois axe.

Une image bitmap couleur est alors formée de trois images bitmap en tons de gris, chacune pour une couleur élémentaire ; il y a ainsi la composante rouge, la composante verte, la composante bleue. On parle donc de composantes RVB ou RGB (*red, green, blue*). Typiquement, on utilisera des systèmes dans lesquels chaque composante de chaque pixel est stockée sur 8 bits (aussi dits ***true color***) ; chaque pixel est donc stocké sur **24 bits** ; il y a alors environ 16,7 millions de teintes possibles.[2] On utilise aussi des systèmes avec 5 bits par composante 32768 couleurs), voire 6 bits pour le vert et 5 bits pour le bleu et 5 bits pour le rouge (65536 couleurs).

**Modèle CMYK - quadrichromie** C'est le principe utilisé par les imprimantes couleur et les imprimeurs. Les couleurs sont alors exprimées par soustraction au blanc de couleurs fondamentales (cyan, magenta, jaune, noir — *Cyan, Magenta, Yellow*) ; on a ajouté le noir (*blacK*) afin de pouvoir obtenu des tons sombres soutenus et sans trop utiliser d'encre. On a ainsi imprimer 4 images monochromes à imprimer ; des procédés de tramage sont utilisés pour obtenir les nuances.

**Modèle HLS** Les modèles précédents sont peu « parlants » pour l'utilisateur. On préfère souvent utiliser le modèle **teinte, luminance, saturation** (*Hue, Light, Saturation*). La luminance est la luminosité de la couleur — c'est elle qui est conservée lors du passage en noir et blanc ; la teinte est prise dans un cercle (rouge - jaune - vert - cyan - bleu - violet) ; la saturation varie entre 0 (gris) et 1 (teinte très saturée), elle exprime le délayage de blanc dans la couleur (fig. 2). Géométriquement, les couleurs forment un cône dont l'axe est formé des gris et dont la pointe est le noir.

### 1.2.3 Couleurs normalisées

Le problèmes des espaces de couleurs précédents est que les couleurs peuvent apparaître légèrement différentes suivant le périphérique utilisé (voir le fait que la même image diffusée sur deux moniteurs différents, ou par deux chaînes de télévision différentes sur le même téléviseur apparaissent avec des couleurs différentes). On est alors amené à appliquer des corrections (gamma par composante).

Il existe par ailleurs des répertoires de couleurs normalisées (Pantone...).

### 1.2.4 Palettes

Certaines cartes graphiques, dont celles de nos Suns, ne peuvent afficher plus de 256 couleurs à la fois sur un écran. Ces couleurs peuvent par contre être choisies

[2] Il existe également des cartes avec des couleurs sur 32 bits ; les 8 bits supplémentaires servent alors à stocker des informations de transparence ($\alpha$-channel).

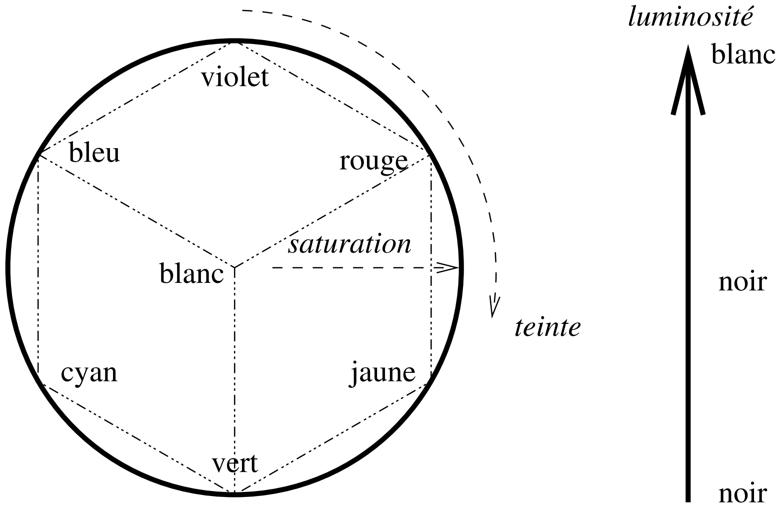


FIG. 2: Le modèle HLS. On voit le cube RGB s'inscrire dedans.

dans une palette 24 bits. Il est par contre possible d'avoir des palettes différentes suivant les fenêtres ; la palette utilisée pour afficher tout l'écran sera alors celle de la fenêtre courante. Cela explique que certaines applications ont des couleurs étranges si l'on ne met par la souris dans leur fenêtre et que les couleurs de lécran entier varient violemment suivant la position de la souris. Cela correspond à l'utilisation d'une palette privée (*private colormap*).

### 1.2.5 Effet de crénage

Lorsque l'on remarque une ligne noire oblique sur un fond blanc sur un écran d'ordinateur (affichage à faible précision, relativement à une imprimante), on est gêné par l'effet de crénage (*aliasing*), qui est que les formes carrées des pixels font que la forme globale apparaît irrégulière. C'est encore plus sensible pour l'affichage des caractères dans des grandes tailles.

Une méthode efficace est le lissage ou *antialiasing*, consistant a mettre des pixels gris dans les contours touchés par l'*aliasing* (fig. 3) pour les adoucir. Signalons que c'est cette technique qui rend l'affichage de `xdvi` agréable et améliore celui de `gv`.

## 1.3 Images vectorielles

Nous l'avons vu, les images *bitmap* ont l'inconvénient de dépendre de la résolution du périphérique d'impression. Ce n'est donc pas pratique pour de nombreuses applications ; on aimerait pouvoir dire « je veux un titre en Times, corps 12, à 1 cm à partir du haut et 3 cm à partir de la gauche de la page » au lieu de fournir un motif de pixel pour cela. C'est l'objet de la description vectorielle des

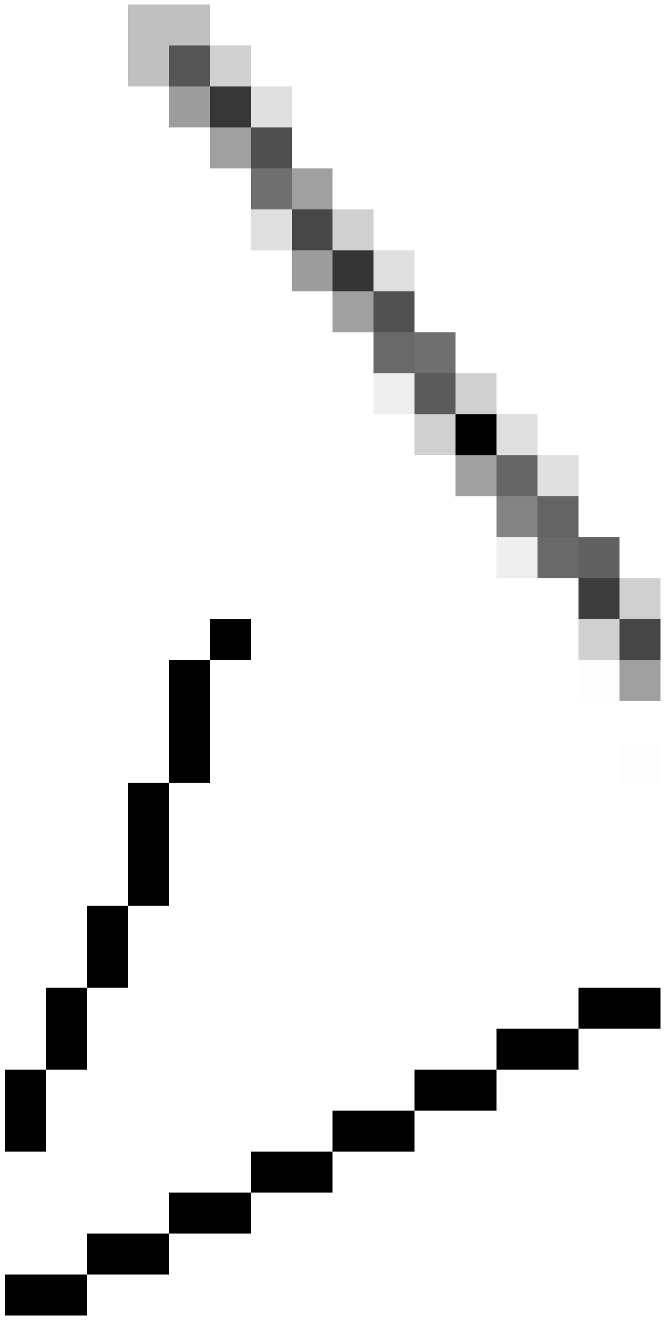

FIG. 3: Exemple de traits bruts et de traits lissés.

dessins. On mémorise des informations du genre « trait entre les points $(x_1, y_1)$ et $(x_2, y_2)$ ». Le dessin ainsi obtenu peut être imprimé à différentes résolutions sans problème.

### 1.4 Dimensions

Nous rappelons qu'un pouce anglais (*inch*) vaut 2,54cm ; un point peut, notamment dans les mesures de hauteur de caractères, désigner une unité de longueur d'environ 0,351 mm (environ $1/72$ de pouce).

### 1.5 Animations

Il s'agit tout bonnement de suites d'images.

## 2 Stockage des images — formats de fichiers

### 2.1 Images bitmap

#### 2.1.1 Compression

Un gros problème avec les images est leur taille. On mesure la taille d'un fichier informatique en **octets** (*bytes*), chacun constitué de 8 bits (*bits*).[3] Les multiples sont le **kilo-octet** (*kilobyte*), qui vaut 1024 octets[4], et le **méga-octet** (*megabyte*), qui vaut $1024 \times 1024$ octets. Faisons le calcul. Une image couleur d'assez bonne qualité (300 dpi), plein format A4, contient

$$\frac{21}{2,54} \times \frac{29,7}{2,54} \times 300^2 = 8\ 700\ 632$$

points. Chacun est stocké sur 24 bits, donc 3 octets. L'image prendrait donc environ 24 méga-octets ! C'est très gros.

Des techniques spécifiques de compression d'image ont donc été mises au point.[5]

On distingue deux classes de méthodes :

**sans perte de qualité** après compression puis décompression, on a la même image qu'à l'origine ; ces méthodes ont des performances limitées ;

[3] Attention à ne pas confondre *byte* et *bit* !

[4] On s'attendrait à 1000 et non 1024, vu le préfixe kilo. Mais, d'un point de vue informatique, 1024 est bien plus « rond » que 1000, car c'est $2^{10}$. Remarquons que les publicités mesurent quelquefois les tailles de disques durs etc... en divisant par 1000 et non 1024, afin de gonfler légèrement les chiffres.

[5] Il est évident qu'il n'existe pas de méthode universelle de compression de données. Toute méthode de compression de données doit donc utiliser des redondances dans les données ; la nature des redondances à exploiter dépend du type de données à compresser. Ainsi, du texte ne se compresse pas de la même façon qu'un programme exécutable ou qu'une image. Plus fin encore, différents types d'images ne se compressent pas pareil, et cela dépend de plus de la qualité désirée.

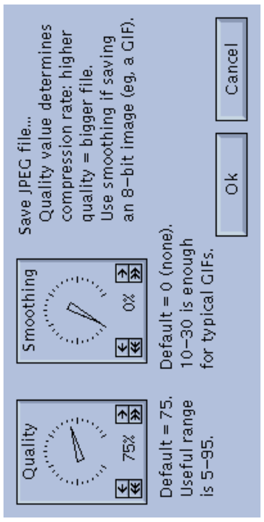


FIG. 4: Xv demande des paramètres pour la sauvegarde JPEG. Le **smoothing** sert à adoucir les images, ce qui est nécessaire sur les images à palette pour reconstituer les dégradés fins et avoir une image de bonne qualité pour la compression JPEG.

**avec perte de qualité** l'image est légèrement différente après compression puis décompression ; des méthodes comme le JPEG permettent, pour une perte de qualité infime, d'obtenir des taux de compression considérables (typiquement, 10% de la taille originale) ; il est cependant recommandé de ne pas sauvegarder une image avec une de ces methodes, la recharger, la traiter, puis la resauvegarder — la qualité finirait par s'en ressentir ; on travaillera donc avec des formats intermédiaires sans perte de qualité ;

### 2.1.2 Formats de fichiers

Il y a différentes façons de stocker les images dans des fichiers, et tous les logiciels ne supportent pas tous les formats. On distingue les formats de fichier à leur extension. Les principaux formats sont :

**GIF** (extension `.gif`) était originellement réservé aux images sur des palettes de 256 couleurs ; il a une compression sans perte de qualité ; il est très répandu pour mettre des images sur le WWW, en particulier grâce à ses fonctions de couleur transparente (définir une couleur comme étant la couleur de fond de l'image, au travers de laquelle on verra l'image placée en dessous) et d'animation ;

**JPEG** (extension `.jpg` ou `.jpeg`) est un format d'images en niveaux de gris et *true color* avec perte de qualité, très efficace ; il faut le réserver aux images comme les photographies[6], les images de synthèse réalistes, et ne pas l'em-

[6]Ce format s'appuie sur les particularités des images du monde réel (les couleurs de pixels

ployer sur des images présentant des petits détails à fort contraste et précis, comme des dessins au trait ; la perte de qualité est sélectionnable à la sauvegarde (une meilleure qualité peut être obtenue au prix d'une taille accrue ; fig. 4) — cette perte est d'ailleurs négligeable pour un réglage de qualité raisonnable ; ce format est très répandu pour mettre des photos sur le WWW ;

**PNG** (extension `.png`) est un nouveau format destiné à remplacer GIF[7], reconnu par la plupart des logiciels récents (Netscape à partir de la version 4...)

**TIFF** (extension `.tif` ou `.tiff`) est un format d'images en niveaux de gris et *true color* ; il a les méthodes de compression des télécopieurs mais la plupart des logiciels produisent du TIFF non compressé, très volumineux.

Il existe une multitude d'autres formats, moins utilisés. Il y a des raisons techniques à cette multiplicité. Certains formats sont très simples à lire et à écrire, mais sont limités en performances de compressions et dans la nature des informations qu'ils peuvent stocker (limités à certaines tailles, certains modèles de couleur...) ; d'autres sont plus complets, mais sont plus difficiles à supporter complètement pour les programmeurs. Il peut arriver qu'un programme sorte un fichier utilisant des fonctionnalités d'un format qui ne soient pas supportées par d'autres logiciels.

☞ *Il est important d'utiliser le bon format au bon moment. Le format GIF est très adapté pour les petits dessins, comme les boutons des pages WWW. En revanche, il donnera des résultats assez mauvais pour les photographies. JPEG, qui a été conçu pour compresser des images photographiques, fera merveille sur ce type d'images ; en revanche, il faudra autant que possible le réserver à la version finale de l'image après retouches, puisque les compression répétées avec pertes pourraient amoindrir la qualité. TIFF, compressé si possible, pourra être un format intéressant pour les versions intermédiaires.*

## 2.2 Images vectorielles

### 2.2.1 Postscript

Le langage Postscript™ est un langage de description de pages, permettant de donner des pages à imprimer indépendantes des caractéristiques du système d'impression. Au delà d'un simple système de description de pages vectoriel, c'est un véritable langage de programmation... dans lequel il est rare que l'on ait à programmer soi-même, vu que les applications comme les traitements de texte etc... génèrent du Postscript tout prêt — c'est d'ailleurs en Postscript que nos traitements de textes commandent nos imprimantes !

---

proches sont proches, les formes sont relativement douces) et de la perception humaine (plus grande sensibilité aux contrastes qu'aux teintes).

[7]Les raisons de ce remplacement sont de natures technique et économique. Le format GIF utilise un algorithme de compression breveté aux États-Unis, et la société qui le possède a décidé de l'exploiter, ce qui gêne son utilisation. Par ailleurs, ce format a des limitations techniques ennuyeuses.

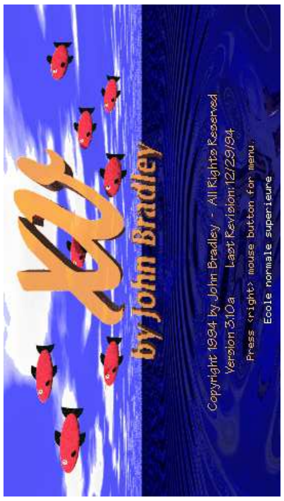


FIG. 5: La fenêtre d'affichage d'images de XV, affichant ici l'image de titre (qui sera remplacée par l'image sur laquelle l'utilisateur travaillera). Cliquer dessus avec le bouton droit pour obtenir le menu.

Les fichiers Postscript ont généralement une extension `.ps`. Il existe également le format *encapsulated Postscript*, qui est adapté à l'inclusion dans d'autres documents (extension `.eps`).

# 3 Les logiciels

## 3.1 Dessin *bitmap*

### 3.1.1 Xv

`xv`[8] est un programme de conversion de format d'image, de manipulation de palette, d'agrandissement et rétrécissement et propose quelques filtres simples.

Lorsqu'il se lance, il affiche une jolie image d'accueil (fig. 5). Cliquez avec le bouton droit dessus (comme d'ailleurs écrit dans l'image) pour afficher le menu (fig. 6). Ce menu présente une liste d'images de travail ; on sélectionne parmi elle l'**image courante** (fond grisé). Les opérations présentées dans ce menu sont :

– des opérations de fichiers

  **load** chargement d'une nouvelle image, et insertion dans la liste,

  **save** sauvegarde de l'image courante,

  **print** impression de l'image courante,

  **delete** effacer l'image

[8]Beaucoup de programmes graphiques sous Unix™ ont un nom commençant par X parce que le système graphique standard sous Unix s'appelle X.

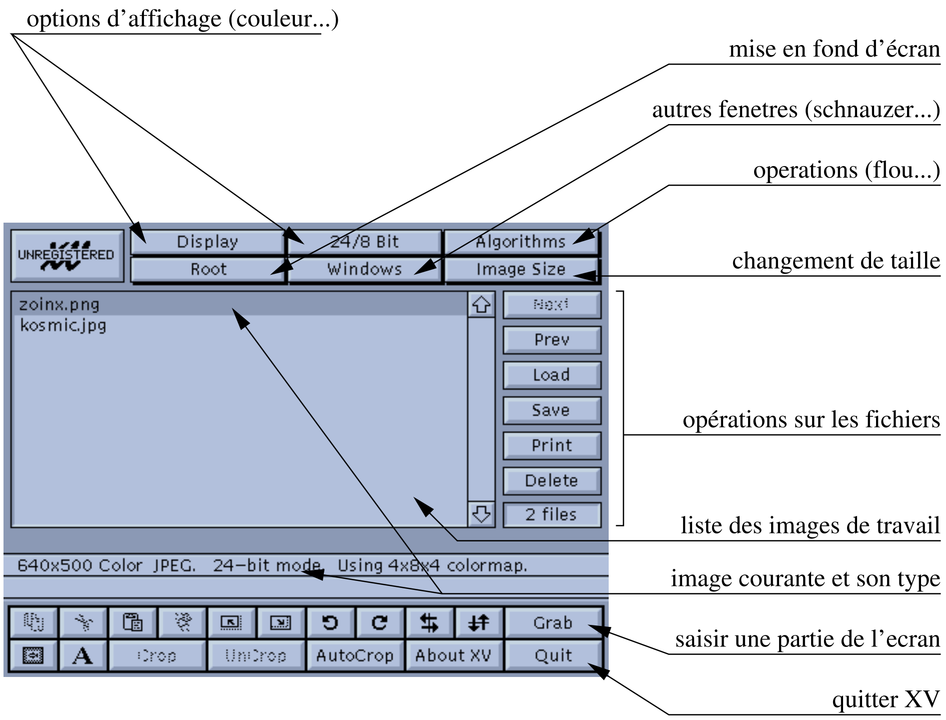


FIG. 6: Le menu principal d'XV, accessible en cliquant avec le bouton droit sur une image visualisée par ce logiciel.

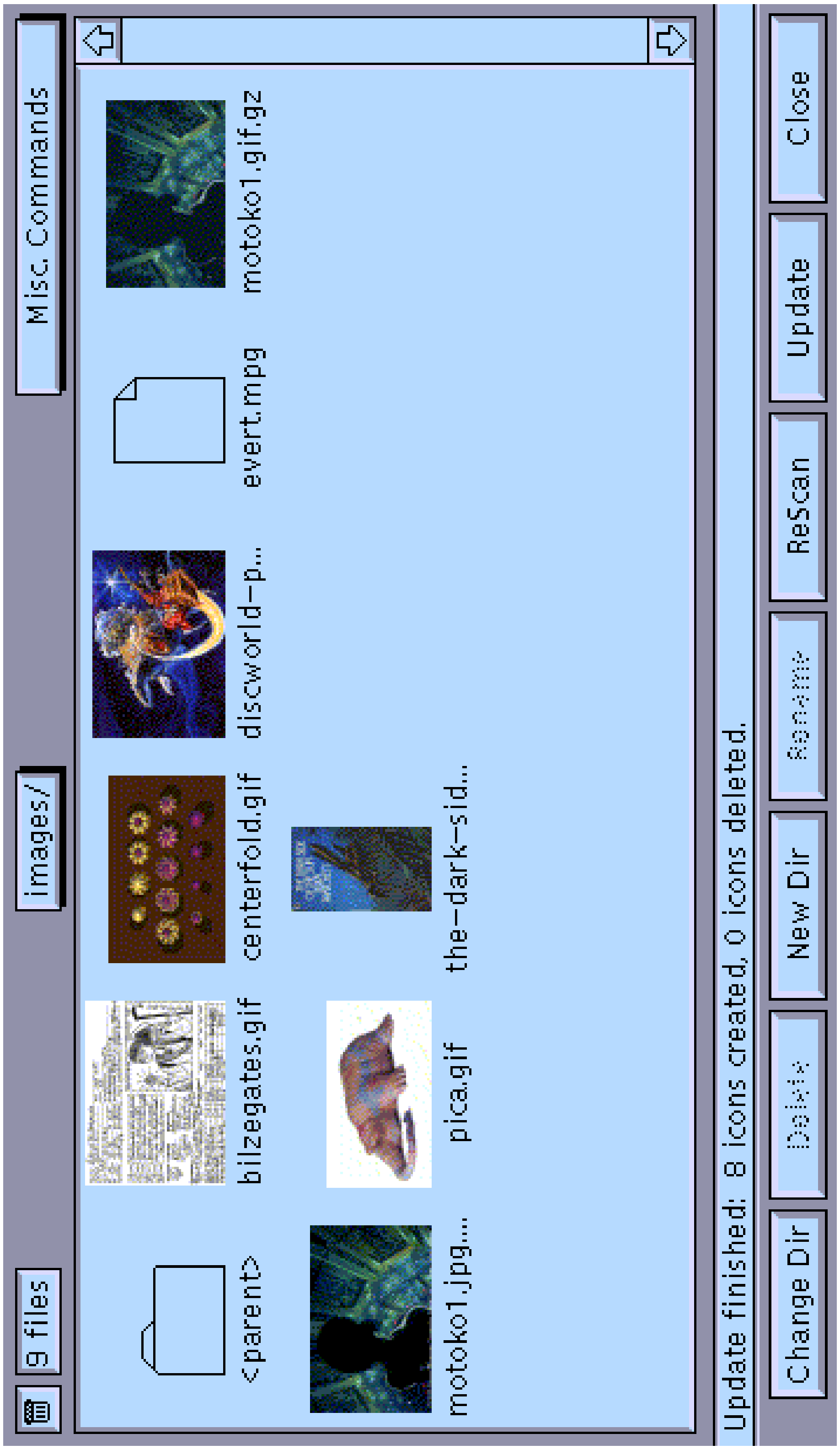


FIG. 7: Le *visual schnauzer* permet de trouver rapidement une image dans un répertoire.

  - de la liste,
  - du disque ;

  bien entendu, de très nombreux formats de fichiers sont supportés ;
- filtres algorithmiques,

  **blur** rendre flou,

  **sharpen** accentuer les contours,

  **edge detect** detecter les contours,

  **emboss** effet de métal embossé,

  **oil painting** effet de peinture à l'huile...

  **undo all** permet d'annuler les filtres déjà appliqués appliqués ;
- changements de taille (voir 4.1)
- affichage : différentes options, en particulier pour le rendu de la couleur ;
- affichage dans le fond de l'écran, selon différents modes ;
- fenêtres complémentaires :

  **visual schnauzer** , qui donne la vision de toutes les images d'un répertoire, chacune au format timbre-poste (cliquer sur *update* pour les obtenir) ; (fig. 7) ;

  **color editor** , qui permet d'éditer les couleurs ;

  **image info** , données sur l'image ;

Des raccourcis-clavier sont disponibles pour beaucoup d'options ; un ∧représente la touche Control et un d' la touche Meta (représentée par un ◆ sur les claviers Sun) ; si le symbole correspondant à l'une de ces touches est indiquée à côté d'une autre touche, on pressera les deux touches simultanément (en fait, on maintient la touche Control ou Meta appuyée, et on appuie brièvement sur l'autre).

Une séquence d'utilisation typique est :
- chargement de l'image,
- application d'un filtre,
- sauvegarde de l'image.

☞ *Les modifications sur une image ne deviennent définitives que lorsque celle-ci a été sauvegardée et XV vous avertit avant de remplacer une image existant déjà sur disque. Il ne faut donc pas avoir peur de jouer avec.*

*Si vous cherchez une image parmi d'autres, utilisez le* visual schnauzer *(Control-V).*

### 3.1.2 Xpaint

`xpaint` est un logiciel simple de dessin dans une image avec palette. Il n'est pas capable de retoucher des images *true color*.

### 3.1.3 Gimp

`gimp` est un logiciel assez gros et puissant de retouche d'images (essentiellement niveaux de gris et *true color*, mais aussi avec palette), semblable à Photoshop™. Vous trouverez plus d'informations dessus dans le serveur du projet`http://www.gimp.org`.

### 3.1.4 NetPBM

☞ *Cette section suppose que vous connaissiez un peu Unix. Une initiation rapide aux connaissances demandées ici est disponible dans l'annexe 5.2.*

NetPBM est un ensemble d'utilitaires de manipulation d'images. Il n'y a pas d'interface graphique, tout se fait en ligne de commande Unix. Ces utilitaires se divisent en trois catégories :

**Convertisseurs d'entrée,** qui convertissent de nombreux formats vers le format PNM. Leurs noms sont de la forme *format*`top`*x*`m`, où *x* est une lettre parmi {`b`,`g`,`m`,`n`,`p`}. On peut ajouter le filtre `djpeg`, qui décompresse le format JPEG.

**Filtres,** qui appliquent des opérations sur les fichiers PNM.

**Convertisseurs de sortie** , qui convertissent le format PNM vers de nombreux formats. Leurs noms sont de la forme `p`*x*`mto`*format*, où *x* est une lettre parmi {`b`,`g`,`m`,`n`,`p`}. On peut ajouter le filtre `djpeg`, qui compresse vers le format JPEG.

Le format PNM est un format non compressé et doc peu adapté au stockage sur disque ; il est par contre très facilement lu et écrit. Il se divise en fait en formats `pbm`, `pgm`, `ppm`, respectivement pour les images monochromes, en niveau de gris et couleur. Le format en couleur est *true color*.

Tous les programmes prennent optionnellement un argument de fichier d'entrée (en son absence, ils prendront l'entrée standard) et sortent leur résultat sur la sortie standard. Ils seront donc particulièrement utiles en *pipes*.

Pour plus d'informations, on se réfèrera à pnm(5)[9]

## 3.2 Dessin vectoriel

**XFig** est un logiciel de dessin adapté à la réalisation de schémas, diagrammes... ; il fournit une sortie Postscript.

# 4 Comment faire pour... ?

Cette section présente des solutions à des problèmes concrets.

[9] Cela veut dire qu'il faut taper `man 5 pnm` pour avoir des informations dessus. Pour plus de renseignements sur la commande `man` et sur la façon de lire sa sortie, voir l'annexe 5.1.

## 4.1 Changement de tailles

? *Je veux changer une image de taille.*

☞ C'est une tâche assez courante, et il existe plusieurs solutions.

**Première solution** Chargez l'image sous XV, sélectionnez sa nouvelle taille et sauvez-la.

Un problème est que lors des rapetissements, l'algorithme utilisé est un peu simpliste (pour un rapport d'agrandissement de $1/n$, il prend un pixel sur $n$), notamment le texte devient illisible. Une méthode satisfaisante est de d'abord l'effet *pixelize* (dans le menu *algorithms*) pour une taille de $n$ — cela va remplacer les blocs $n \times n$ de pixels par leur moyenne. Appliquer alors un facteur d'agrandissement de $1/n$ va donner le résultat escompté. Si l'on veut spécifier des facteurs différents pour les axes des abscisses et des ordonnées (respectivement $1/n_1$ et $1/n_2$), demander une pixélisation de $n_1$ x $n_2$.

**Seconde solution** Utiliser un *pipe-line* avec NetPBM (voir 3.1.4) ; par exemple, pour appliquer un agrandissement d'un facteur 0,3 sur l'image `zoinx.gif` et obtenir `petitzoinx.gif`, faire :

```
giftopnm zoinx.gif | pnmscale 0.3 | ppmtogif >petitzoinx.gif
```

Cette solution donne des résultats satisfaisants y compris au rapetissement et ne demande pas d'astuce spécifique contrairement à la précédente.

---

? *XV rapetisse très mal certaines de mes images, en particulier le texte devient illisible.*

Voir la réponse à la question précédente, au sujet du simplisme de l'algorithme utilisé par XV.

---

? *Un logiciel me sort une courbe en Postscript. Je voudrais changer sa taille pour l'inclure dans un autre logiciel. J'ai essayé avec XV mais le résultat est tout moche.*

Le problème est que XV est un logiciel de dessin *bitmap* et va donc convertir le Postscript en une *bitmap* à la (faible) résolution de l'écran. C'est cette *bitmap* que vous sauvegardez ensuite, ce qui donne un résultat très peu satisfaisant à l'impression.

☞ Les logiciels qui permettent d'inclure du Postscript permettent généralement d'en choisir la taille. XFig permet de définir la taille d'un *picture object*. En LaTeX, c'est facile aussi :

```
\usepackage{graphicx}
...
\includegraphics[scale=0.3]{fichier.ps}
```

# 5 Quelques détails sur Unix

☞ *Les notions présentées ici le sont de façon rapide ; cette annexe ne prétend pas remplacer le cours d'initiation à Unix !*

## 5.1 Le manuel en ligne : la commande `man`

Les systèmes Unix disposent d'un manuel en ligne. On l'obtient par la commande `man`. L'usage le plus courant est `man` *commande*. `man -k` *chaîne* recherche une chaîne dans les titres des pages du manuel (ce qui permet d'obtenir les pages qui parlent d'un sujet).

Le manuel est divisé en sections, dont voici celles qui vous intéressent :

1 commandes utilisateur

5 formats de fichiers

Lorsque l'on réfère à une page du manuel, on la note *nom* `(`*section*`)`. Pour la voir, on utilise `man` *section nom*[10]. Par exemple, `man 1 man` envoie la page de manuel sur la commande `man`.

## 5.2 Entrée et sortie standards, redirections

Les programmes sous Unix ont chacun une entrée et une sortie standards et une sortie d'erreur standard. Lorsqu'on les lance depuis le *prompt* d'un *shell*[11] , l'entrée et la sortie standard se font dans le terminal (au clavier pour l'entrée), ainsi que la sortie d'erreur.

On peut **rediriger** ces entrées et sorties en mettant sur la ligne de commande des instructions :

- > *fichier* dirige la sortie standard vers un fichier
- < *fichier* prend l'entrée standard dans un fichier
- `2>` *fichier* dirige la sortie d'erreur standard vers un fichier[12]

Une autre fonctionnalité très importante est le ***pipe***. *commande1* | *commande2* lance les deux commandes en même temps, avec la sortie standard de la première reliée par un tuyau à l'entrée standard de la deuxième[13]. Il est possible de mettre plus de deux commandes et de créer ainsi de longs pipe-lines.

[10]Ce genre de choses varie suivant le genre d'Unix considéré, mais on s'est arrangé à l'ENS, zone élèves, pour que cela soit cohérent.

[11]Un *shell*, c'est le truc en mode texte dans lequel on tape des commandes, un peu comme sous MS-DOS. Et le prompt c'est l'affichage d'un *shell* interactif qui invite l'utilisateur à taper une commande. Par exemple, ici, je demande au *shell* de lancer la commande `zoinx` : `brick-~ $ zoinx`

[12]`2>` fonctionne dans tous les *shells* raisonnables, compatibles avec `sh`. Par contre, il ne fonctionne pas dans les C-shells, qui sont de toutes façon des bouses. De toute façon, par défaut tout le monde à l'ENS, zone élèves, a `zsh`, donc le problème ne se pose pas ici.

[13]Pour ceux qui ont utilisé MS-DOS : les *pipes* Unix sont des vrais, et non des émulations passant par des fichiers temporaires.