

L'IMAGERIE INFORMATIQUE

1 – Rappels sur les procédés classiques d'expression graphique

- dessin *au trait* (charbon, stylet, plume) sur paroi, roche, pierre, tablette d'argile, papier ...
- dessin au doigt, au pinceau..., sur paroi, parchemin, papyrus, cuir, toile, papier ...
- peinture par remplissage de formes dessinées au trait,
- peinture directe (sans canevas de traits),
- peinture par points juxtaposés,
- projection d'émulsion (pistolet, ébulliseur...)

En informatique, on peut ramener toutes les possibilités actuelles de graphisme à deux types :

- l'**image pixellisée**,
- le **dessin vectoriel**.

On va les étudier ci-après en accordant davantage d'attention à l'image pixellisée (ou *bitmap*). Le dessin vectoriel, plus complexe, justifierait une étude à part.

2 – Image pixellisée

L'image pixellisée, c'est un **tableau** rectangulaire de n lignes avec m pixels par ligne, côte à côte. Le **pixel** est la plus petite surface sur laquelle on peut agir ; c'est l'**élément de base** du dessin, la tache élémentaire.

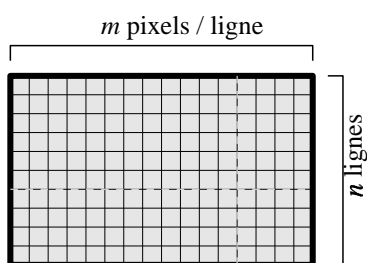


Fig. 1 – Image pixellisée schématique

Avantage : **simple** à créer, ne nécessite aucune compétence en programmation.

Inconvénients : l'image **dépend** fortement des moyens (écran, imprimante, carte graphique), donc actuellement images de petite taille seulement (< 20×30 cm), mais on pourrait en assembler plusieurs.

Cette image se modifie difficilement et s'agrandit très mal (sa *résolution* est finie). Elle pourrait convenir au

tachisme ou au *pointillisme* avec un logiciel conçu dans cette optique. C'est la forme des images de télévision (625 lignes de 830 pixels environ, 25 images/seconde).

En informatique, le pixel est une valeur numérique (ou un groupe de 3 valeurs si l'image est en couleurs). Elles s'échelonnent en général de 0 à 256 (256 valeurs de gris ou 256^3 couleurs, soit plus de 16 millions). Les valeurs de couleur sont celles des **3 teintes fondamentales**. L'échelle des valeurs peut être réduite parfois à 0 et 1.

Pour produire une image pixellisée, on utilise :

- soit un logiciel de dessin courant (gratuit ou bien distribué avec le système d'exploitation, comme **Paint**),
- soit un logiciel commercial plus élaboré (**CorelDraw**, **Illustrator**, ...),
- soit la numérisation (*scanning*, *scannage* en français, *bélino*) d'une image réelle ou d'une photo,
- soit la photographie numérique directe d'une scène ou d'un objet réel,
- soit une photo numérisée retouchée avec un logiciel comme **Photoshop**,
- soit la numérisation d'une image vectorielle.

3 – Mémorisation des images

L'image est contenue dans un fichier, qui est écrit soit en mémoire vive (la mémoire de travail de la machine, mémoire rapide, mais transitoire), soit dans un disque ou une disquette (mémoires lentes mais permanentes).

L'informatique ne sait manipuler ou conserver que des 0 ou des 1. Son unité fondamentale est le bit, égal soit à 1, soit à 0. La mémoire de base est une bascule mise à 1 ou à 0 (*qui dit oui, qui dit non*, le courant électrique passe ou ne passe pas, l'interrupteur est ouvert ou fermé). Mais on peut grouper les bits en **octets** (ensemble de 8 bits), en kilooctets (1 ko = 1024 octets), en mégaoctets (1 Mo = 1024 ko) ou en gigaoctets (1 Go = 1024 Mo).

Pour conserver un pixel en noir OU blanc, il suffit d'un bit, mis à 0 si le pixel est noir, à 1 s'il est blanc. C'est le cas du dessin au trait (si on ne tient pas compte de la plus ou moins grande intensité du noir). Dans ce cas, un octet contient 8 pixels.

Pour conserver un pixel dans lequel on veut distinguer 256 **teintes de gris**, on lui attribue un octet (un ensemble de 8 bits qui peut prendre $2^8 = 256$ valeurs). L'intensité de gris est donc conservée sous la forme d'un nombre. On ne serait pas obligé d'utiliser 256 niveaux de gris, mais c'est plus pratique et universel (parce qu'alors le pixel occupe juste un octet) ; l'œil moyen, paraît-il, ne distinguerait que 64 niveaux de gris.

Pour conserver un pixel en couleur, on utilise généralement 3 octets (ce qui autorise 16 millions de couleurs).

Donc, il existe **3 grands types d'images pixellisées** : en N&B, en gris, en couleurs.

Une image de télévision ordinaire N&B comporte $625 \times 625 \times 4/3$, soit plus de 520 000 pixels ; en couleur, elle *pèserait* près de 1,5 Mo si elle était mise en mémoire numérique (mais les cassettes des magnétoscopes actuels ne sont pas numériques). C'est énorme, sachant qu'une disquette dite *haute densité* contient 1,4 Mo ; elle ne peut même pas contenir une seule image de télévision *brute* (un disque optique en contiendrait environ 600).

Pourquoi *brutes* ? Parce qu'heureusement, on sait compresser les fichiers images. On en reparlera plus tard. Les *taux de compression* peuvent atteindre 10 à 50. La disquette peut alors contenir 10 à 50 images de télévision.

Dans la mémoire, tous les octets se suivent indifférenciés. Pour pouvoir reconstruire l'image, le logiciel doit connaître au moins les informations suivantes : nombre n de lignes, nombre m de pixels par ligne et nombre k de bits par pixel. La taille du fichier image non comprimée

est de $mnk/8$ octets, mnk ou $3mnk$ (selon le type) pour sa partie image ; le fichier doit comporter aussi une partie **descriptive** donnant au moins les valeurs m , n , k , et occupant encore quelques dizaines ou centaines d'octets.

La description de l'image peut être faite de différentes façons ; des conventions ou même de normes ont été établies. En tête du fichier, un mot doit immédiatement permettre de reconnaître le procédé employé pour conserver l'image. De plus, en général (mais ce n'est pas obligatoire), le nom du fichier devrait comporter un *suffixe* identifiant ce procédé.

Par exemple, les recopies d'écran sous Windows (qu'on obtient en appuyant sur la touche *Impr-Ecran*) provoquent une sorte de photographie de l'écran actuel, ou plus exactement la mise en mémoire (dans le *presse-papier* ou *clipboard*) de tout le contenu de la page en cours dans la carte graphique. Ce fichier non comprimé a pour suffixe **bmp** et occupe, selon la finesse et la taille de l'écran, environ un mégaoctet. On peut le compresser, mais il faut vérifier si sa qualité reste correcte (on en reparlera dans les dernières sections).

4 – Résolution

Il peut paraître peu naturel d'admettre qu'une image est composée de points. Pourtant, c'est la règle ! Toutes les images imprimées sont faites de points, cartes postales comprises ; il suffit de les regarder à la loupe (elles possèdent au plus 10 points par millimètre ; on en parlera dans la section 5). Même les photos sont faites de grains (cristaux de sels d'argent qu'on peut les voir au microscope : taille de l'ordre du micron). On appelle **résolution** l'intervalle séparant les points élémentaires d'une image.

Et l'image *naturelle* ? La nature elle-même est faite d'atomes, mais si petits qu'on ne les discerne pas ! La référence, **c'est l'œil**. Sa rétine est composée de cellules telles qu'on ne distingue que des objets séparés par un angle supérieur à son **pouvoir séparateur**. Ce pouvoir dépend des individus, mais, en moyenne, il est voisin d'**une minute d'arc**, i.e. 0,3 milliradians (ou millièmes). Ceci veut dire qu'on ne distingue à 35 cm (distance de lecture) que des traits espacés d'au moins 0,1 mm.

5 – Demi-teintes et tramage dans l'imprimerie traditionnelle

Comme on l'a dit, l'image est conservée en mémoire ; mais elle n'est visible qu'avec un **écran** ou bien après **impression** sur papier. Il faut donc parler des problèmes d'impression et spécialement de celle des demi-teintes.

On appelle **demi-teinte** une teinte *non saturée*. On se limitera d'abord au *gris* (ce qui sera dit sera valable pour n'importe quelle couleur de base). Le *blanc* (ou bien la couleur du papier) et le *noir* (ou toute couleur d'encre) sont considérés comme des teintes limites (ou saturées) et la demi-teinte se situe entre les deux.

Les artistes peuvent représenter le gris en appuyant plus ou moins fort sur le crayon. Mais l'imprimerie ne dispose que d'encre noire, qu'on dépose ou non sur le papier. On dit que c'est une technique *binnaire* (tout ou rien). Quand la *gravure* a été découverte (gravure au burin, ou à la pointe sèche, sur cuivre), on n'a pu rendre les gris qu'avec le procédé des **hachures** (traits parallèles tous de même épaisseur, mais d'espacement variable, voir la figure 2). Quand les **procédés photo** se sont imposés en imprimerie, il a fallu trouver une méthode analogue pour rendre les gris.

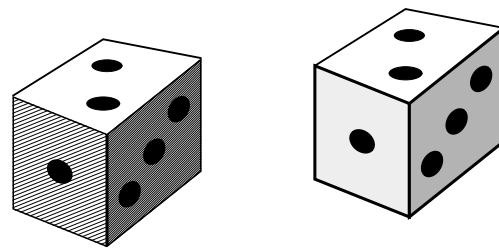


Fig. 2 – Rendu des gris :
à gauche, par rayures ; à droite, par tramage

On a alors inventé le **tramage**. Depuis lors, toutes les photos imprimées, même les cartes postales, sont tramées. On remplace l'image initiale ou la photo par un tableau de points équidistants (un maillage à pas constant).

On appelle ces points des **taches**. Elles sont forcément imprimées en noir pur, mais leur **diamètre** varie avec le niveau de gris. Le *pas* du réseau de taches est inférieur à la *résolution de l'œil*. On ne voit qu'une **moyenne** et donc un gris plus ou moins foncé selon le diamètre des points. A la limite, quand le diamètre des taches est maximum, elles se touchent et le papier à cet endroit est complètement noir (schéma page suivante).

Le tramage de la photo est toujours obligatoire dans l'imprimerie professionnelle. Il était réalisé au début du siècle dans une *chambre noire* (gros appareil de photo) en interposant entre la photo d'origine et le cliché d'imprimerie une *trame cristal* ou réseau de fils fins parallèles de pas constant et croisés. La trame cristal décompose (le mot correct est **échantillonne**) la photo en taches *équidistantes* dont le rayon varie avec la transparence locale de l'échantillon (les mécanismes physiques en cause sont la diffraction de la lumière par les petits trous et le seuil de sensibilité du cliché). Actuellement, la trame cristal est remplacée par une pellicule impressionnée avec un réseau de franges laser (même résultat).

Toutes les imprimantes informatiques (à laser ou à jet d'encre) sauf celles à sublimation thermique (rares) utilisent le procédé du tramage pour représenter les demi-teintes. Avec une loupe, on distingue très bien le tramage sur une photo imprimée, même de haute qualité.

6 – La couleur

Chaque pixel possède une couleur propre et une seule. Or on a découvert que la lumière blanche est formée d'un mélange de couleurs. Pour le prouver, on utilise un *disperseur*, tel qu'un prisme de verre ; c'est l'explication de l'arc-en-ciel, lumière du soleil *dispersée* par les gouttes d'eau.

On a trouvé également qu'on pouvait reproduire pratiquement toute couleur en mélangeant d'autres couleurs (ex: bleu+jaune → vert). On a alors tenté de ramener toutes les nuances à **3 couleurs fondamentales**.

C'est un procédé approché et non univoque : il y a en effet plusieurs moyens d'y parvenir (ou plusieurs systèmes de couleurs) :

- **RVB** (ou, en anglais, RGB). La télévision produit la couleur en composant les fondamentales rouge, vert, bleu.
- **CYM** : la photo (à cause du passage obligatoire par un négatif) construit ses couleurs à partir des fondamentales cyan, jaune (yellow) et magenta.
- **CYMK** : le noir obtenu avec le système CYM étant de mauvaise qualité (marron foncé), l'imprimerie de type photogravure utilise une couleur supplémentaire, le noir (K pour *black*).
- **CIE** : une *société savante*, la Compagnie internationale de l'éclairage a, en 1931, défini une carte triangulaire des couleurs à partir de notions mieux adaptées à la perception humaine de la couleur et plus *commode* à utiliser que les précédents.

7 – Tramage dans l'impression informatique

Les machines à écrire il y a 20 ans frappaient directement des caractères sur le papier comme, d'ailleurs, les premières imprimantes informatiques.

Les imprimantes bureautiques actuelles impriment *par points*. Au début, elles comportaient des aiguilles frappant le papier ; celles à laser et celles à jet d'encre produisent des points beaucoup plus *serrés* que celles à aiguilles. Mais elles ne savent déposer que **du noir** (ou la

Il existe des normes ou *conventions* : les photos des quotidiens sont imprimées à 70 lpi (lignes de taches par pouce), mais les hebdomadaires le sont à 150 ou 250 lpi. Pour atteindre la qualité photo, il faut 1000 lpi au moins. Le nombre de lignes par pouce s'appelle la **linéature**.

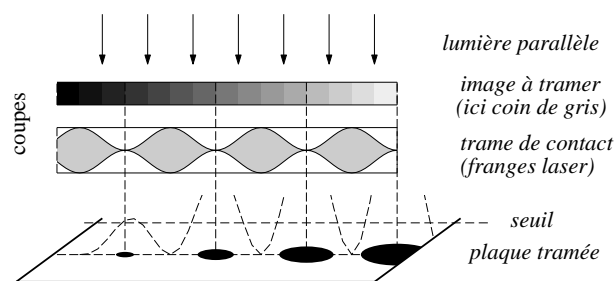


Fig. 3 : Tramage par contact

– **HSB** : système US de couleurs proche de celui de la CIE, avec les notions de **teinte** (*hue*), de **pureté** (*saturation* en anglais) et **intensité** (*brillance*).

Des relations mathématiques permettent de passer d'un système à l'autre (de manière approchée).

Pour obtenir une image de télévision couleurs, on dispose 3 éléments colorés (RVB), ou **pixels**, l'un près de l'autre ; chaque pixel d'une couleur donnée est excité par un canon donné (sélection du canon par une grille très fine appelée *masque*). En regardant l'écran à la loupe, on aperçoit bien les 3 composantes (en triangle ou en ligne). Cependant, l'image télé est *émise* selon le système CIE (seule façon d'assurer la compatibilité avec les récepteurs N&B). Pour la photo couleur, 3 couches sensibles identiques sont superposées et séparées les unes des autres par des filtres sélectionnant la couleur appropriée.

En imprimerie, on utilise un **triple réseau de taches** : une trame pour chaque couleur fondamentale en quinconce l'une par rapport à l'autre (donc base triangulaire comme en TV). On module l'intensité de chaque couleur comme avec les niveaux de gris. On ne *mélange* jamais intimement les encres de couleur (elles réagiraient chimiquement entre elles et n'obéiraient donc plus aux lois de composition énoncées ci-dessus). Seuls les procédés de transfert thermique, avec des encres très grasses (des *cires*) autorisent le mélange intime (les réactions chimiques sont bloquées). Toutes les autres imprimantes utilisent la **juxtaposition** (le tramage est une juxtaposition).

couleur de l'encre sélectionnée). Elles ne peuvent pas directement imprimer du gris ou une nuance de couleur. Le problème est le même que celui de l'imprimerie classique. Il est résolu par un **tramage calculé**. On décompose l'image en pixels ; on mesure la couleur moyenne au droit de chaque pixel et on représente ce pixel par une tache grossièrement circulaire d'un diamètre *a* proportionnel à l'intensité du gris à cet endroit. Les points–

machine (*dots*), beaucoup plus denses que les pixels, ne sont imprimés que s'ils sont situés à l'intérieur de la tache de diamètre a . Le gris est donc rendu par le rayon variable des taches situées sur réseau fixe (la linéature). La couleur est rendue grâce à 3 réseaux similaires de même pas p et disposés en quinconce, chacun affecté à l'une des 3 ou 4 couleurs fondamentales.

On devine qu'il y a une relation entre ces paramètres : en effet, une tache de dimension a pourra contenir en gros a^2/r^2 points machine ou n^2a^2 , si leur diamètre est r ou si leur densité est n ($n=1/r$, $n = 300, 600, 720, 1200 \dots$ ppi ou dpi, *dots per inch*). Le nombre N de niveaux de gris possibles est égal au nombre maximum de ces points noirs dans une tache, plus 1 (correspondant au blanc pur).

Or la tache de diamètre maximum avec une linéature L est $a = p = 1/L$. Le nombre de niveaux de gris est donc $N = 1 + (n/L)^2$. Pour obtenir $N=64$ niveaux de gris, il faut $n=8L$ ($n=600$ dpi $\rightarrow L=70$ lpi et $L=150$ exige $n=1200$).

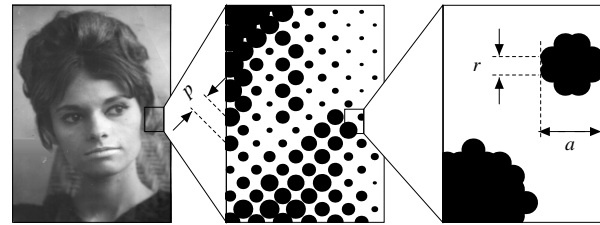


Fig. 4 photo entière photo tramée (détail) taches agrandies

8 – Le matricage

On imprime donc les nuances de gris par tramage et les nuances de couleur avec trois réseaux tramés en quinconce. Mais, en informatique, on met au point les images sur un **écran**. L'écran n'est pas seulement un auxiliaire, il peut aussi servir d'organe de sortie des images (par ex. projection vidéo, ou grand écran, ou photo d'écran). L'écran n'utilise pas le procédé du tramage pour représenter les demi-teintes. Les premiers écrans informatiques n'avaient que trois nuances : le pixel était *éteint*, *allumé* ou *demi-allumé*. Puis vinrent les écrans couleur EGA qui ne pouvaient produire que 8 teintes pâles et 8 teintes brillantes (trois couleurs, trois canons avec, chacun, 3 possibilités : éteint, allumé, ou demi-allumé).

C'était un procédé *ternaire*, donc un peu plus souple que l'imprimerie. Hélas, la résolution des écrans (nombre de points par unité de longueur) est bien inférieure à celle de l'imprimerie : la résolution 300 dpi des premières imprimantes-laser correspond à des pas de 85 microns entre points machine ; l'écran de télévision n'a qu'un pas de 0,5 mm, les écrans informatiques VGA des pas de 0,35 mm et les superVGA actuels des pas de 0,25 mm. A cause du trop faible pas du point écran, on n'a pas pu utiliser le *tramage*, mais un procédé appelé **matricage**. Voisin du tramage, ce procédé consiste à consacrer à chaque pixel image non pas un point écran, mais un rectangle contenant n^2 points écran, de sorte qu'en allumant ou non ces points (et on ne savait faire que ça), on obtient n^2+1 niveaux de gris (ou nuances si on est en couleurs). L'écran ne peut alors contenir qu'un nombre restreint de pixels. L'écran SVGA comporte 640 points en largeur, il ne peut donc afficher que $N = 640/n$ pixels ; si on exige 65 teintes, $n=8$, on est limité à 80 pixels en largeur (pour une soixantaine en hauteur) ! Si on se limite à 17 teintes, on double le nombre de pixels dans la largeur de l'écran. Ce procédé est encore utilisé par des logiciels mal adaptés, bien que les écrans EGA soient devenus rares. En outre, c'est un procédé d'emploi délicat, car il fait vite apparaître des zébrures ou des *moirés*.

Avec la norme VGA, la carte graphique sait gérer 256 intensités par pixel (donc 16 millions de couleurs). Il n'y a plus besoin de tramage ni de matricage : un pixel écran peut *faire* un pixel image. Le nombre de pixels est peu élevé. Il croît avec la taille de l'écran (1280×1024 pour un bon écran de 17"). Ce nombre se calcule ainsi : la taille commerciale D annoncée est celle de sa diagonale technique, dont environ 93% forme la diagonale utile ; le rapport largeur-hauteur étant de 4 à 3, la largeur L est égale à $0,93 \times 4/5 D$; si le pas (*pitch*) est p , le nombre maximum de points réellement disponible est $n=L/p$, i.e. $n=18,8 D/p$ (si D est en pouces et p en mm). Ex.: un pas de 0,25 mm autorise juste 1280 points sur un 17".

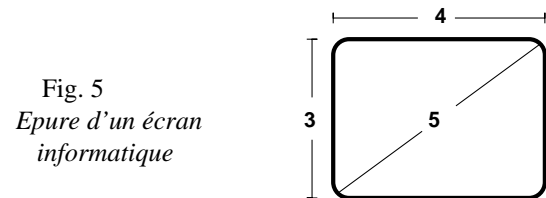


Fig. 5 Epure d'un écran informatique

L'écran VGA retrouve donc la souplesse de la télévision (256 modulations pour chaque pixel). Malheureusement, les créateurs du PC n'avaient pas prévu ça et la mémoire attribuée à l'écran (150 ko) ne pouvait pas contenir autant d'informations (640×480 points avec 3 couleurs de 256 valeurs chacune exigent déjà 1 Mo de mémoire). Les cartes graphiques récentes pour écrans SuperVGA contiennent de 1 à 8 Mo de mémoire et permettent donc d'afficher des images détaillées. Mais tous les logiciels en service n'en tiennent pas compte et certains pratiquent encore le matricage, limitant le nombre utile de pixels à guère plus de 300×200. On peut calculer la valeur M , capacité mémoire nécessaire pour contenir une image de largeur n pixels (donc $0,75n^2$ au total) avec N nuances par couleur : c'est $M = 0,28n^2 \log_2 N$. Une carte graphique de 4 Mo peut donc afficher 16 millions de couleurs avec une résolution de 1330 points. Avec moins de mémoire, il faut réduire au moins l'un des paramètres.

9 – Création d'image pixellisées

On peut d'abord échantillonner à l'écran une image vectorielle par copie d'écran.

On peut **échantillonner** une image papier par béliographie (*scanning* en anglais ou même *scannage* en français). L'appareil décompose l'image en m lignes et cha-

que ligne en n pixels. Chacun de ses n capteurs mesure l'éclairement du point visé avec une précision de $1/N$ ou de $\log_2 256$, soit 8 bits. Si l'appareil accepte la couleur, il comporte $3n$ capteurs. Une majorité des images informatiques actuelles sont issues de ce procédé.

On peut **photographier** une scène avec un appareil de photo numérique. Son optique est de même type que celle des appareils classiques (de milieu de gamme). La pellicule amovible est remplacée par un détecteur multi-point appelé CCD produisant autant de pixels par image qu'il comporte de points sensibles (*mn*). Les images sont conservées sur une petite carte, qu'il faut décharger sur un gros disque (disque dur ou, mieux, disque optique).

Enfin, on peut produire soi-même des images pixelisées avec un calculateur et un **logiciel** approprié. Il existe des logiciels simples, gratuits ou bien distribués avec le système d'exploitation (comme *MSPaint*) et des logiciels plus élaborés mais plus coûteux (*CorelDraw*, *Illustrator*, etc ...).

Tous proposent au graphiste des outils : traceurs de formes géométriques, pinceau, pistolet, seau à peinture ... On peut tracer des figures géométriques faites de lignes droites, de rectangles, d'arcs de cercles ou d'ellipses. Ces lignes auront la couleur et l'épaisseur désirée. Avec la souris, on sélectionne un outil dans la barre d'outils, puis on clique sur l'emplacement souhaité pour l'objet tracé. Après sélection d'une zone d'image (rectangulaire), on peut changer sa dimension, la déplacer ou la dupliquer. Les emplacements et dimensions sont peu précis.

Si la figure construite est fermée, on peut la remplir d'une couleur avec le seau à peinture. La couleur est choisie généralement en cliquant sur une teinte dans une palette. Avec les meilleurs logiciels, on peut définir soi-même une couleur quelconque en fixant le pourcentage de chacune des composantes.

On peut également passer le pinceau (avec la couleur choisie) sur l'image ; la largeur de la trace et sa couleur sont fixes ; elles ne tiennent pas compte d'une plus ou moins grande force d'appui (sauf peut-être dans les logiciels haut de gamme, spécialement ceux exploitant les récentes *tablettes graphiques*). Il y a peu de nuances ; les dégradés sont difficiles. On obtient souvent un dessin un peu schématique, mais certains artistes en tirent de belles compositions dans la tendance *art moderne*. L'emploi du pistolet (*aérographe*) est plus délicat, mais cet outil autorise plus de nuance et des formes plus libres. Il colorie seulement une partie des pixels sous son passage : plus on le passe lentement, plus denses sont les pixels colorés. Un beau résultat exige cependant beaucoup de dextérité.

10 – Le dessin vectoriel

C'est un dessin très **abstrait** (au sens technique, pas au sens artistique) et totalement **indépendant** des machines, de leur limitations et de leur résolution. Au moment d'être mis sur papier (ou sur pellicule), il est, bien sûr, soumis à des limitations (dimensions, procédé), mais il ne l'est pas dans sa conception. Quand les machines évoluent, ce genre de dessin utilise toujours toutes leurs performances (ce n'est pas le cas de l'image pixelisée).

En outre, ce dessin abstrait n'a pas de dimensions précises : on peut le sortir à n'importe quelle échelle, sous

Une utilisation plus courante de ces logiciels consiste à **modifier de photos** numérisées. La modification affectera soit les couleurs (qu'on peut améliorer), soit les formes, globales ou locales (on déforme en transformant en courbes les lignes droites sous-tendant une partie d'un dessin : déformation des traits d'un visage par exemple).

On peut modifier l'**amplitude** de tous les pixels (toutes les valeurs du fichier) : augmenter le *contraste* revient à les **multiplier** toutes par un même facteur ; augmenter la *luminosité*, c'est leur **ajouter** la même quantité.

La découpe rectangulaire de l'image en modifie les dimensions *l* et *h*, tandis que le **détourage** ne les change pas, mais met à blanc tous les pixels à l'extérieur de la partie d'image *détournée* (de forme quelconque). On est aidé dans cette sélection par le *lasso* (qui crée avec la souris une forme libre) ou par la *baguette magique* qui ajoute ou retranche de la sélection des zones d'égale couleur.

Il est possible de **remplacer une zone** d'image par une autre, pour supprimer un élément non désiré ou bien en ajouter un : cette facilité, pas toujours innocente, permet toute sorte de manipulation et a contribué à enlever toute crédibilité aux images informatisées.

Agrandir une image est un terme ambigu. On peut se contenter de modifier sa *présentation* à l'écran (fonction souvent appelée *zoom*) : les pixels sont simplement représentés par des matrices plus étendues, mais uniformes (il n'y a pas interpolation) ; c'est cependant une fonction très utile parce qu'elle permet des travaux plus précis sur l'image, en particulier la correction pixel par pixel.

L'autre type d'agrandissement consiste à faire calculer une **nouvelle image** qui remplacera la précédente dans le fichier. C'est une opération très délicate : par exemple, diviser par 2 les dimensions de l'image revient à grouper les pixels par 4, à calculer la moyenne de leurs couleurs et à remplacer ce groupe par un seul pixel avec la couleur moyenne. Augmenter la taille revient à inventer des points artificiels dont on calcule la couleur à partir de (en interpolant) celle des voisins (de même si on divise la taille de l'image par un facteur autre qu'un multiple de 2). De même, la déformation des lignes d'une image est une opération mathématiquement très compliquée et réservée aux meilleurs logiciels de traitement d'images.

n'importe quel grandissement. Enfin, il est très peu encombrant en mémoire ; il est très **économique**. Ce type de dessin possède donc de sérieux atouts. Le prix à payer ? Il faut le programmer, d'où un savoir-faire technique important et un minimum de connaissances mathématiques.

Quand on dit programmer, ça veut dire rédiger pour le calculateur (ou ordinateur) une liste d'ordres dans un **langage** donné. Plus le dessin est perfectionné, plus le langage est complexe. On va dire quelques mots de quelques uns de ces langages.

11 – Le dessin vectoriel de base

Pourquoi de base ? Parce que presque tous les langages informatiques de base possèdent des instructions de dessin vectoriel : le Pascal, le C, le Java, le Basic ...

On écrit donc un programme qui comporte des ordres de dessin. Comme avec les outils des logiciels de dessin pixellisé, on peut tracer des lignes droites, des courbes ou des figures géométriques (cercles, secteurs, rectangles, ellipses, ...) ; on colore à sa guise les traits et les surfaces. Le résultat est bien meilleur qu'avec les logiciels de dessin, parce qu'on maîtrise beaucoup mieux *la plume* (virtuelle, bien sûr). Les emplacements et dimensions fixés sont extrêmement précis. Ce n'est plus la souris qui les attribue. On écrit des nombres, les coordonnées des points. On sait exactement ce qu'on a fait, ce qu'on a programmé ; il est beaucoup plus facile de garder ce qui

nous convient et d'éliminer les tracés malheureux. Mais, en gros, le dessin aura tout de même peu de nuances, il restera schématique et un peu froid.

Citons la possibilité pour ces logiciels de créer des **images animées** ; en effet, très souvent, on peut dessiner un objet avec une sorte d'identité propre : il peut être conservé et servir à nouveau à un autre emplacement, avec une autre couleur, une autre taille ou une autre inclinaison (c'est fondamental dans le dessin animé si l'on veut réduire la quantité de travail à fournir).

Ces logiciels ne produisent des dessins en principe que pour l'écran. Si on veut les imprimer, il faudra passer par une copie d'écran, dont on a dit qu'elle fournissait du dessin de qualité médiocre (parce qu'à faible résolution).

12 – Le dessin HPGL

C'est le langage que comprennent les **tables traçantes** (maintenant démodées, ou bien chères, car très grandes) ainsi que les imprimantes **laser de type HP**. Le jeu d'instructions HPGL (Hewlett-Packard graphical language) fait partie du PCL, langage de base des imprimantes HP, depuis sa version 5. Le HPGL comporte 2 niveaux.

Le niveau 1 avait été conçu pour les petites tables traçantes, lesquelles avaient un pas de 50 microns au mieux

et utilisaient des plumes (7 au maximum en général). On ne disposait ainsi que de 7 types de traits (en couleur ou en épaisseur). C'était donc plutôt limité pour les imprimantes actuelles. Aussi, le HPGL de niveau 1 n'est plus guère utilisé. Le niveau 2 améliore le premier ; il sait gérer la finesse des imprimantes laser et écrire des traits variés, en forme et en taille. N'ayant jamais utilisé le niveau 2 de ce langage, nous n'en dirons rien de plus (il est d'ailleurs très peu *documenté*).

13 – Le dessin en langage PostScript

C'est le langage de création de dessin vectoriel le plus perfectionné. On écrit là aussi un **programme** qui sera contenu dans un fichier. La taille du fichier n'a rien à voir avec la taille de l'image créée, mais elle croît avec sa complexité. Par exemple un logo programmé en PostScript (**PS**) peut occuper 2 à 5 Ko. Si on numérise cette image (à partir de celle en PS ou en passant au scanner l'image initiale de l'artiste), la taille du fichier sera de plusieurs centaines de kilooctets.

Les ordres qu'on peut écrire dans ce langage sont très élaborés. Ils comprennent tous ceux disponibles dans les langages cités §10. De plus, la notion de trait est plus perfectionnée : il peut se terminer en arrondi ou en rectangle, les jonctions de traits peuvent être arrondies PS passe toujours par les figures géométriques. En plus des polygones, des cercles, secteurs, ellipses, il dispose des **courbes de Bézier**, élégantes figures mathématiques du 3e degré mises au point par Pierre Bézier, ingénieur chez Renault, pour représenter des profils de carrosserie. On peut inclure dans le dessin tout texte, avec n'importe quelle taille et dans n'importe quelle direction.

On dispose de 16 millions de couleurs, soit sous forme RVB, soit sous forme HSB. Ces couleurs servent à colorer des traits (toutes épaisseurs autorisées) ou à remplir des surfaces. La surface doit toujours être complètement fermée par un trait, mais ce trait peut être virtuel, c-à-d.

non tracé. Par exemple, pour dessiner une boule éclairée, on décompose la boule en cercles élémentaires qui reçoivent chacun une nuance dégradée de la couleur nominale.

PS est employé – souvent à l'insu du dessinateur – par les logiciels de dessin haut de gamme mentionnés ci-dessus. Il permet de créer des logos, des schémas, des dessins techniques de très haute qualité et toujours avec la finition ultime permise par l'imprimante. Un des gros avantages de PS consiste en son aptitude à piloter n'importe quelle imprimante de bureau ou professionnelle avec la plus haute qualité disponible. Ces avantages sont importants, le prix à payer également. Non pas le prix commercial, car tout est gratuit. Le langage n'est pas breveté et il existe des logiciels gratuits qui aident puissamment le programmeur en représentant à l'écran le résultat de son programme (Ghostscript, Ghostview, Gsview...). Le seul prix à payer est intellectuel, car c'est un langage relativement difficile. Cependant, quand il est bien expliqué, on le maîtrise rapidement avec un peu de volonté.

On peut conclure en disant que, si, au prix d'un investissement intellectuel certain, PS permet le dessin technique d'une très haute qualité, il n'est pas spécialement prévu pour le dessin artistique ou la peinture. Cependant, c'est probablement à partir de ce langage que pourraient être développés des logiciels de dessin artistique, laissant plus de liberté à l'opérateur.

14 – Logiciels de dessin vectoriel

Il existe des logiciels conviviaux permettant de créer du dessin vectoriel. Leur nom en général comporte le mot *draw*, comme *MayuraDraw*. En apparence, les barres d'outils et les menus sont semblables à ceux des logiciels de dessin pixellisé, mais leur action est très différente. Ces outils créent **des objets** qu'on peut ensuite reprendre, transformer, déplacer, dupliquer... car ils ont une expression abstraite (mathématique). Lorsqu'on clique sur eux, leurs points de construction (*handles*) apparaissent et rendent possibles ces transformations. On peut également **grouper** (ou dégroupier) ces objets pour leur appliquer des transformations globales.

Ces logiciels, apparentés en réalité à ceux de la **CAO** (*conception assistée par ordinateur*), sont bien adaptés au dessin technique, mais toujours un peu froids et géométriques pour un travail artistique, à moins que leurs prestations ne s'enrichissent à l'avenir. Ils créent souvent des fichiers en langage PostScript, mais n'ont malgré tout pas accès à la totalité du puissant arsenal d'instructions dont dispose ce langage.

15 – Echanges

On peut facilement passer d'un dessin vectoriel à un dessin pixellisé. Certains logiciels s'en chargent (Ghostscript par exemple pour PS). On peut également représenter le dessin à l'écran (avec le logiciel convenable) et faire une **copie d'écran**. Le fichier pixellisé ainsi obtenu peut avoir une taille plus de 100 fois supérieure à celle du fichier vectoriel d'origine avec de surcroît une qualité dégradée. On n'opère ainsi seulement quand c'est strictement nécessaire (pour un site Internet par exemple).

Passer d'un dessin pixellisé à un dessin vectoriel est beaucoup plus difficile. L'exécutant peut analyser lui-même le dessin, en extraire les traits les plus significatifs et les programmer. Il existe quelques rares logiciels transformant partiellement des dessins pixellisés en dessin PS. De toutes façons, ils n'en extraient que peu de lignes et il faut reprendre le travail avec soin. Une méthode – laborieuse mais efficace – consiste à créer un dessin vectoriel par dessus une image pixel présentée en toile de fond.

16 – La compression

On a vu que les fichiers d'images pixellisées sont volumineux ; leur taille est en général exagérée par rapport à l'information contenue. Par exemple, dans un dessin au trait, il y a beaucoup de pixels blancs ; ainsi une ligne de 600 pixels peut ne comporter que quelques pixels noirs (passages des traits par cette ligne) ; il n'est pas judicieux de conserver les 600 pixels ; l'information est dite redondante. De nombreuses méthodes ont été inventées pour comprimer ces fichiers. Nous en citerons quatre.

– Procédé **RLE** (*run length encoding* ou *packbit*). On étudie le fichier ligne par ligne. Si dans l'une, m octets se suivent avec une valeur identique n , on les remplace par les deux symboles successifs mn . Cette méthode très simple (actuellement peu employée), ne comprime bien que les dessins au trait ou ceux créés par informatique.

– Méthode de **Huffmann** (ou statistique). On lit une première fois le fichier pour procéder à des **statistiques** ; on cherche si des suites de caractères se répètent et, ce, combien de fois. On remplace ces suites par un code numérique simple (un nombre), d'autant plus court que la séquence est fréquente. Bien sûr, il faut conserver ces codes dans un dictionnaire qu'on joint au fichier. Les fax sont des images pixellisées comportant surtout des traits et lues par béliño. Tous sont maintenant comprimés par une méthode de Huffmann, mais avec un dictionnaire des codes fixé une fois pour toutes et qu'on ne joint pas au fax. La norme CCIT3 comprime les fax d'un facteur atteignant 15 ; la norme CCIT4 atteint le facteur 30.

– Méthodes **LZ77** et **LZW** (ou *progressives*). Avec ces méthodes, on n'effectue pas d'étude préalable du fichier. On code au fur et à mesure, remplaçant les séquences par des nombres, mais en utilisant les codes déjà définis

quand une nouvelle suite de caractères répète la même séquence. Le dictionnaire se construit ainsi petit à petit, il n'a pas besoin d'être joint au fichier, le logiciel lecteur peut le reconstruire au vu des codes comprimés. Le taux de compression peut dépasser 20. Cette méthode est la plus employée en informatique, pour tout type de fichier et pas seulement pour les images. Relèvent de ce procédé les logiciels **zip**, **unzip**, **gzip**, **winzip** ainsi que les fichiers ***.zip** et ***.gif** (à noter que le procédé **gif** est breveté et qu'on ne peut pas l'utiliser pour des travaux lucratifs, sauf après contrat avec Unisys, détentrice du brevet).

– Méthode **JPEG**. Cette méthode complexe utilise plusieurs procédés. En particulier, elle remplace les images par leur *spectre de Fourier* numérique ; comme ceux-ci s'étendraient à l'infini, on les tronque ; ce faisant, on perd l'information sur les détails les plus fins de l'image (souvent du bruit). Parmi toutes les méthodes de compression, c'est la seule qui **perde un peu d'information**. Toutes les autres permettent de reconstituer exactement l'image de départ. La méthode JPEG donne de bons résultats avec les images à tons continus et spécialement les photos, avec lesquelles le taux de compression peut dépasser 100. On peut régler le seuil à partir duquel on perd les détails ; la méthode exige un peu de doigté, mais elle est **extrêmement employée** pour comprimer les **photos**.

D'autres méthodes existent ou sont en cours d'étude, comme la *compression fractale*. Signalons qu'en ce qui concerne les images de **télévision numérique**, chaque image est comprimée bien sûr ; mais, en outre, on émet seulement les fractions d'image différant de la précédente (on appelle cela la **compression temporelle**). Le taux global de compression peut alors dépasser 1000).

17 – La cryptographie

Nous terminerons par un aperçu sur le **chiffrement** ou *cryptage* des fichiers images en vue de les dérober aux regards indiscrets, surtout si on les transmet par Internet. Ce problème n'est pas réservé aux fichiers images.

Quand on désire conserver dans ses disques un ou des fichiers confidentiels, on peut utiliser l'un des trois logiciels ci-après.

– **Winzip**, logiciel commercial, dont il existe sur Internet des versions gratuites dites *de démonstration* (2,2 Mo environ), valables quelques semaines seulement. On peut, grâce à lui, *archiver* des fichiers et chiffrer l'archive avec un **mot de passe**. **Archiver** signifie *comprimer et réunir* en un même fichier tout un ensemble de fichiers.

– **Powerarchiver**, gratuit, disponible sur Internet et totalement compatible avec Winzip (même taille) : il peut même décrypter et *désarchiver* une archive Winzip si on connaît le mot de passe ayant servi au chiffrement.

– **Security Box**, gratuit et également disponible sur Internet, ne produit pas d'archive, mais crypte les fichiers un à un à l'aide d'un mot de passe ; toutefois, il refuse certains fichiers, spécialement s'ils sont de type Jpeg.

Ces logiciels peuvent créer des archives sur disque multiple (comme **Backup**, voir ci-après). Cependant ils n'utilisent pas des algorithmes très puissants et on peut assez facilement les *casser* (des logiciels commerciaux existent pour ce faire !). Pour obtenir plus de sûreté – en particulier pour envoyer un message secret sur Internet, il faut utiliser un outil beaucoup plus fiable comme PGP.

PGP utilise le cryptage à **double clé**. Avec un chiffrement à clé simple, votre destinataire doit connaître votre clé (ou *mot de passe*), mais il serait imprudent de la lui envoyer, soit avec le fichier, soit en différé (les espions sont aux écoutes). Dans le cryptage à **double clé**, on uti-

lise une *clé publique* et une *clé privée*. Vous demandez à votre correspondant sa clé publique (qu'il ne refuse à personne) et vous cryptez votre message à la fois avec sa clé publique et votre clé privée. Vous lui envoyez votre clé publique (vous pouvez aussi la diffuser largement sur Internet). Pour décrypter le fichier, il faut la clé privée du destinataire et votre clé publique. Il est, paraît-il, quasi impossible de remonter aux clés privées à partir des clés publiques. Il faut cependant être sûr que personne n'a *falsifié* la clé publique utilisée, ce qui exige son *authentification* par un *certificat* d'une personne sûre et connue.

C'est le chiffrement considéré comme le plus sûr, mais **il n'est pas** (encore ?) **autorisé en France**, ni la police ni l'armée n'étant parvenues à le *casser*. PGP (*pretty good privacy*), est disponible gratuitement sur Internet.

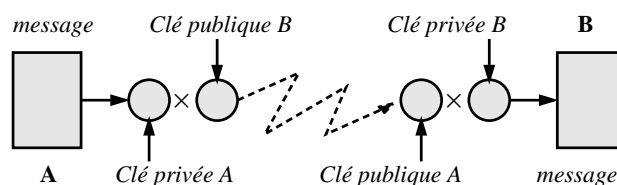


Fig. 6 – Chiffrement par double clé

Sur Internet, ces logiciels sont fournis sous forme d'un gros fichier lui-même étant une archive (auto-décompressible si elle a pour suffixe **exe**). Ces fichiers dépassent généralement 1,4 Mo. Si on n'a pas de graveur et qu'on veut les transporter, il faut utiliser plusieurs disquettes. Dans ce cas, il est nécessaire de passer par un utilitaire approprié. Les *archiveurs* décrits ci-dessus (**Winzip** et **PowerArchiver**) savent archiver sur plusieurs disquettes successives (option *archivage multiple*). On peut également employer l'utilitaire **Backup** de Windows en suivant avec attention la procédure indiquée.

18 – Comparaison avec la musique

On peut esquisser un parallèle entre les fichiers d'image et les fichiers de musique. Ces derniers existent également sous deux formes : les fichiers de **musique échantillonnée** et ceux de **musique programmée**.

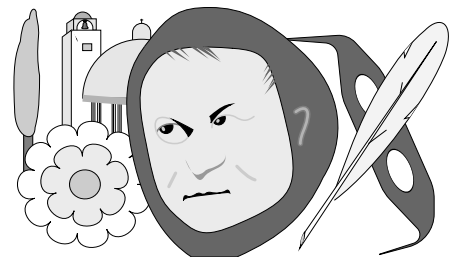
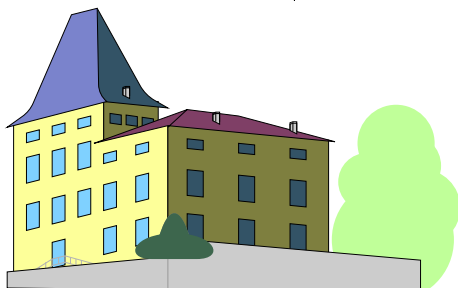
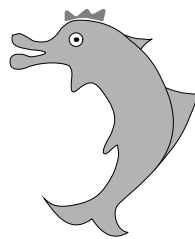
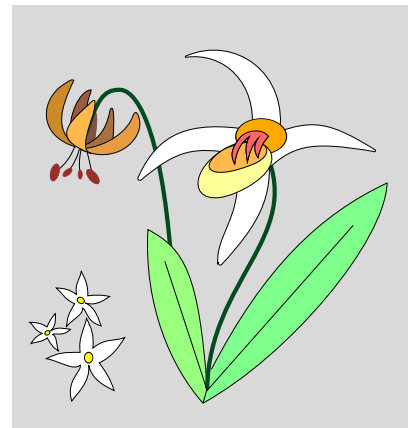
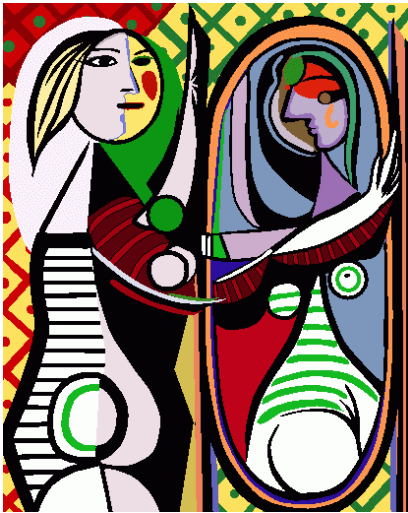
La musique échantillonnée est une suite d'octets donnant à chaque instant (44 000 fois par seconde) le son (la forme de l'*onde sonore*) parvenant à un micro. De tels fichiers sont volumineux : ainsi une heure de musique stéréo exige de mémoriser $3600 \times 44\,000 \times 4$ octets si le son est *mesuré* avec une grande précision (2 octets). En calculant, on trouve 634 Mo ; c'est la capacité des disques optiques, qui – comme on le sait – contiennent bien une heure de musique. Cette façon d'enregistrer la musique est tout fait similaire à l'enregistrement des images sous forme de points. Dans les deux cas, on sait heureusement compresser de tels fichiers (méthodes Jpeg, LZW... pour les images, MP3 pour la musique).

Mais on sait également représenter de la musique par des symboles : ceux du solfège... une heure de musique *remplit* alors seulement quelques feuillets de papier (à condition de savoir les décoder). En informatique, la notation symbolique des partitions musicales fait l'objet de la **norme MIDI** : quelques octets seulement indiquent pour chaque note sa hauteur, sa durée, son *volume*, le ou les instruments qui la jouent ..., leur *expression*. Cette note occupe une dizaine d'octets au lieu des 44 000 exigés par l'échantillonnage s'il s'agit d'une noire (durée une seconde environ). Le gain en volume est considérable (plus de 1000), tout comme avec une image **vectorielle**. Dans les deux cas, il faut d'une part, savoir décoder ces formes **symboliques** et, d'autre part, bien savoir les utiliser pour que l'œuvre ne paraisse pas trop pauvre, schématique, voire étriquée.

Pierre Blanc, janvier 2002

Plusieurs exemples de dessins informatiques **pixelisés**, dus à Georges Jallibert, peuvent être trouvés sur le site Internet de la MPT de Biviers. Les dessins **vectoriels** ci-dessous sont tous programmés en PostScript. A gauche on peut voir une traduction PS de *la Fille au miroir* de Picasso (exemple donné dans le

logiciel **MayuraDraw**). Au centre, *le tigre* fait partie du logiciel **Ghostscript**. Les autres dessins ont été programmés par l'auteur, mais deux d'entre eux reproduisent en PS des logos *d'artiste*, celui de la MPT, créé par G. Jallibert, et celui du *hibou*, propriété et emblème du Parc naturel de Chartreuse.



Voici, pour donner un exemple, le programme PostScript complet qui dessine le logo de la MPT de Biviers ci-dessus. (les 3 premières lignes définissent des abréviations utilisées pour raccourcir le nom des *fonctions de dessin* du langage).

```

/db {gsave} def /rt {grestore} def /M {moveto} def /L {lineto} def /R {rlineto} def
/Z {translate} def /rot {rotate} def /T {stroke} def /col {sethsbcolor} def
/Q {setgray} def /F {setgray fill} bind def /bz {curveto} def /fer {closepath} def
db 140 148 Z .3 .2 scale % mise en place et à l'échelle
db 20 0 M 125 0 R 0 140 R -125 0 R fer .17 1 1 col fill rt 0 Q % rect. jaune
130 88 M 120 112 120 127 130 154 bz 142 127 142 112 130 88 bz 0 F % arbre
60 0 M 60 70 L 0 70 L 0 90 L 70 102 L 70 120 L 140 95 L 140 80 L 160 80 L 160 70 L % toit
102 70 64 40 64 0 bz 0 F % base
db 58.5 68.5 Z 0 0 3 0 360 arc 1 Q fill T % soleil
-20 rot 7 {4 0 M 15 0 5 20 -20 arcn 4 0 L 1 F T 30 rot} repeat % rayons
rt 0 Q
12 /Helvetica-Bold findfont 12 scalefont setfont % police utilisée
0 50 M (M P T) show 0 35 M 1.4 0 (BIVIERS) ashow % texte
rt showpage

```