Fabriquer un reflet numérique d'un objet réel

M . Arnaud, V. Dageville,
D. Larrieu, P. Lucaud

Questions :

Sur quel support les anciens appareils photos enregistraient-ils les images ? Et de nos jours ? Pourquoi dit-on appareils photos numériques ? Où sont les nombres ? Comment sont-ils organisés ? Que fait-on avec ?

.....
.....
.....

Décrire l'information avec des nombres binaires :

La plus petite unité d'information manipulable par une machine numérique est le **bit** *« binary digit »*.
Un bit prend deux états auxquels on attribue les symboles 0 ou 1.
Il est possible de représenter physiquement cette information :

- par un signal électrique (le courant passe ou ne passe pas) ou magnétique,
- par des aspérités géométriques dans une surface (le principe du braille)
- par des sons courts et longs (le morse) ...

Grâce à 2 bits, on peut obtenir 4 états différents (2×2) : 00 01 10 11.
Avec 3 bits, on peut obtenir .....=..... états différents : 000 001 .....
Avec 4 bits, on peut obtenir .....=..... états différents : .....
.....

Les octets :

Avec 8 bits, il est possible d'obtenir …….=……. états différents.
Cet ensemble de 8 bits est appelé un **octet (**en anglais Byte noté B).
En général les informations sont regroupées par paquets de 8, 16, 32 ou 64 bits, c'est-à-dire 1, 2, 4 ou 8 octets.
Avec 2 octets (16 bits) il est possible d'obtenir .....états différents.
Avec 4 octets (32 bits) il est possible d'obtenir .....états différents.

Unités : Au début de l'informatique, on travaillait avec des langages proches des micro-processeurs qui eux-mêmes travaillaient avec les bits. Les unités de stockages allaient donc selon les puissances de deux amenant la confusion suivante : 1 kilo-octet = 1024 octets. Maintenant que la science informatique a de plus en plus de liens avec les autres sciences (biologie ...) un besoin de clarification s'est fait sentir. Le préfixe kilo qui est indépendant de l'unité choisie (grammes, mètres...) représente 10^3 = 1000. Depuis 1998 le Bureau International des Poids et Mesures a fixé la règle suivante :

- Un kilooctet (ko ou kB) = 1000 octets
- Un mégaoctet (Mo ou MB) = 1000 ko = 1 000 000 octets
- Un gigaoctet (Go ou GB) = 1000 Mo = 1 000 000 000 octets
- Un téraoctet (To) = 1000 Go = 1 000 000 000 000 octets

En binaire :

- Un kibioctet = 1 Kio = 1024 octets
- Un mébioctet = 1 Mio = 1024^2 octets = 1 048 576 octets
- Un gibioctet = 1 Gio = 1024^3 octets = 1 073 741 824 octets...

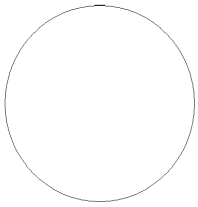
Les vendeurs de supports informatiques utilisent cette ambigüité pour « gonfler » la capacité des supports de stockages vendus. Un disque dur d'une capacité de 2^{40} octets, soit 1 024 Gio, s'affiche à 1 100 Go. Mais au fait, de combien est la différence ?.....

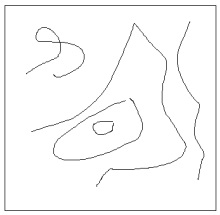
Quelques ordres de grandeur:

- Une disquette a une capacité d'environ .....
- Les clés USB font maintenant entre .....
- La mémoire vive (RAM) d'un ordinateur est comprise entre ..... Mo et ..... Go.
- Un CD-Rom contient .....
- Un DVD fait .....
- Les disques durs actuels font souvent 150, 250 ou 500 Go, allant jusqu'à 1 To (téraoctets) pour certains.

Appliquons maintenant nos nouvelles connaissances au codage d'images :

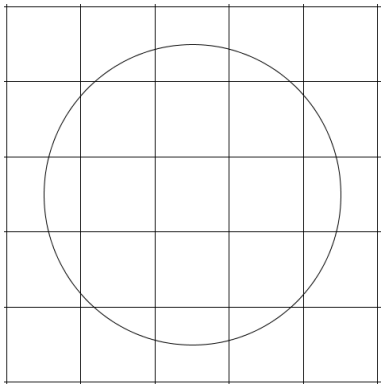
Codage d'une image noir et blanc : Comment décrire ces dessins ? Quel est celui qui demande le moins d'informations ?

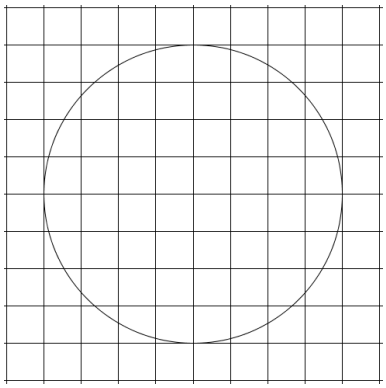


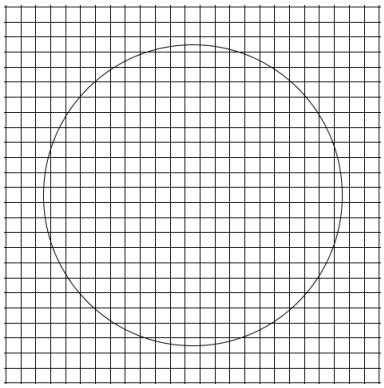


....
....
....

Nous allons maintenant utiliser un autre procédé. On superpose au tracé un quadrillage. On noircit les carrés qui contiennent une portion de trait. On obtient une **image matricielle** formée d'un tableau de points ou **pixels** *« Picture element ».*

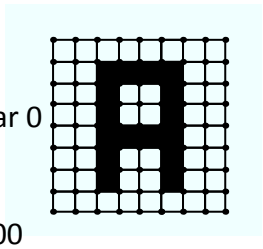






Si on convient de coder le blanc par 0 et le noir par 1, on obtient :
Pour le dessin 1 : .....
Pour le dessin 2 : .....

Un petit exercice à faire à deux :



Le codage de la lettre A dessinée ci-contre, si l'on décide qu'un carré blanc est codé par 0 et un carré noir par 1, serait le suivant :

00000000 00111100 00100100 00100100 00111100 00100100 00100100 00000000

On a regroupé les 0 et 1 par paquets de 8, qui constituent des octets et qui correspondent aux lignes !

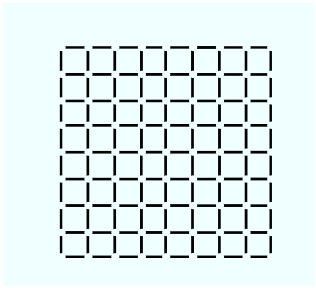
Un tel dessin, composé de 64 cases blanches (0) ou noires (1) se code donc sur 8 octets (soit 64 bits).

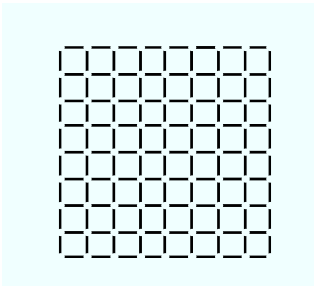
Plutôt que de mettre les zéros et les uns en ligne, on aurait pu les organiser comme ceci :

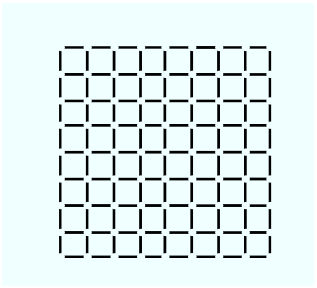
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .

C'est une **matrice**, un objet mathématique avec lequel on peut faire de nombreux calculs.

Travail à faire :







1. Après avoir transposé la matrice ci-dessus (échanger les lignes et les colonnes) faire le dessin dans la première grille. Comment est la nouvelle figure ?
2. Dessiner un pictogramme dans le second carré, le coder, passer le code au voisin qui en déduira le pictogramme.
3. Récupérer le code du pictogramme de son voisin et le dessiner dans la troisième grille. Comparer avec l'original !

Dans ce travail, qui joue le rôle du fichier, qui joue le rôle du logiciel ?.....

.....

Combien de pictogrammes différents peut-on faire ?.....

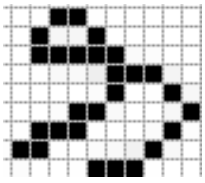
Paint et les Pixels :

Nous avons vu que certaines figures peuvent être décrites simplement avec un vocabulaire adapté (en supposant que l'interlocuteur à qui on s'adresse connaît ce vocabulaire) pour être reproduites à l'identique, c'est-à-dire obtenir une figure parfaitement superposable à l'originale.
Cependant d'autres figures sont beaucoup plus complexes et il est très difficile voir quasiment impossible de décrire le plus précisément possible et à l'aide d'un vocabulaire adapté, ce qu'il faut faire pour que soit reproduites à l'identique ces figures (Cependant certaines personnes arrivent à approcher la description donnée, par exemple dans le cas des portraits robots).

Nous allons maintenant nous intéresser à une figure qu'il sera assez difficile de décrire précisément pour pouvoir la reproduire à l'identique :

1. Essayer de décrire avec un vocabulaire adapté le plus précisément cette figure afin que quelqu'un puisse la reproduire à l'identique :.....
   .....
   .....

2. Nous allons utiliser le langage binaire pour faire la même chose, pour cela on a quadrillé cette figure avec une matrice de 9x12. On décide de coder les carrés blancs par 0 et les carrés noirs par 1.
   En codant de gauche à droite ligne par ligne la figure quadrillée, indiquer le codage de cette figure en langage binaire :.....



   .....
   .....

3. Nous voulons obtenir une figure plus proche de celle donnée. Copier et coller la première figure dans « Paint » (pour trouver « Paint », aller dans le menu « Démarrer », « Tous les programmes » puis « accessoires »). Dans « Paint », cliquer sur « affichage », aller sur « zoom » puis « personnalisé ». Sélectionner ensuite « 800% », et suivre le même chemin pour « afficher la grille ».
   A l'aide de la grille que vous obtenez indiquer le nombre total de 0 et de 1 qui seront nécessaires pour coder cette figure.

4. Que pensez-vous de l'efficacité du langage binaire pour la reproduction de figure?
   .....

5. Soit le codage binaire d'une figure donné ci-dessous:

```
1111100111110011110010001001111
1100100100110010010010011001001
1111100111100010010011010001001
1001100101100011110001110001001
1001100100100010010001100001001
1111000100100010010001100001111
```

a. Déterminer le nombre de lignes et de colonnes nécessaires dans une matrice pour reproduire la figure dont le codage est donné :.....

b. Insérer ci-dessous un tableau ayant les dimensions requises puis à l'aide de l'icône remplissage, remplir les cases codées 1 en noir, qu'obtenez-vous comme figure ?

Codage des images (source Wikidia)

Images monochromes

Les images sur les écrans des premiers ordinateurs étaient composés de Pixels [1] monochromes qui pouvaient être soit allumés (en vert ou en orange par exemple), soit éteints.

Le codage de telles images était plutôt simple : pour représenter un pixel allumé, on utilisait un 1, et pour un pixel éteint, c'était un 0.

Les images noir et blanc de type fax (sans nuances de gris) sont toujours codées ainsi. Elles occupent très peu de place et leur acheminement sur un réseau est très rapide.

Exemple : une image de 800 pixels de large et 600 pixels de hauteur (on dit aussi *une définition de 800x600*) *codée en 1 bit* (c'est-à-dire *monochrome*) a une taille de :

800 × 600 × 1 = ……. bits = ……. octets = …… ko

Quand on n'avait que des images monochromes, **un bit** suffisait pour les **2 couleurs** (noir et blanc par exemple). Mais une fois les ordinateurs capables d'afficher des couleurs, il a fallu coder les pixels sur plusieurs bits.

Images en couleurs

L'une des premières normes d'affichage, la norme CGA, permettait d'afficher des pixels dans une palette de **4 couleurs** différentes (prises parmi les 16 disponibles). Pour cela, il fallait que chaque pixel soit codé par **2 bits**. Par exemple :

Numéro (décimal)	Numéro (binaire)	Couleur
0	00	noir
1	01	vert
2	10	rouge
3	11	jaune

Ensuite, le nombre de couleurs qu'on pouvait représenter à la fois a continué à augmenter. On peut à présent avoir des images de **16 couleurs**. Et puisque $16 = 2^4$, il suffit pour cela de coder chaque pixel par une suite de **4 bits**. Voici un exemple de palette courante :

N° décimal	N° binaire	Couleur	N° décimal	N° binaire	Couleur
0	0000	noir	8	1000	gris foncé
1	0001	bleu	9	1001	bleu clair
2	0010	vert	10	1010	vert clair
3	0011	cyan	11	1011	cyan clair
4	0100	rouge	12	1100	rouge clair
5	0101	magenta	13	1101	magenta clair
6	0110	marron	14	1110	jaune
7	0111	gris	15	1111	blanc

Et on peut continuer ainsi pour découvrir des images de plus en plus belles grâce à leur grand nombre de couleurs : avec **8 bits** par pixels, on obtient **256 couleurs** (car $2^8 = 256$) ; avec **16 bits**, **65536 couleurs** ; avec **24 bits**, plus de **16 millions** de couleurs ; avec **32 bits**, plus de **4 milliards**. À partir de 24 bits, on parle d'ailleurs d'affichage en *vraies couleurs* tant le réalisme est parfait. Voici deux exemples :

Images en teintes (ou niveaux) de gris

On ne code ici plus que le niveau de l'intensité lumineuse, généralement sur un octet (256 valeurs). Par convention, la valeur zéro représente le noir (intensité lumineuse nulle) et la valeur 255 le blanc (intensité lumineuse maximale) :

000 008 016 024 032 040 048 056 064 072 080 088 096 104 112 120 128 136 144 152 160 168 176 184 192 200 208 216 224 232 240 248 255

Ce procédé est fréquemment utilisé pour reproduire des photos en noir et blanc ou du texte.

Images en couleurs

Dans le codage RVB (ou RGB en anglais) les 24 bits d'une couleur se décomposent en 3 fois 8 bits, c'est-à-dire 3 octets :

- 8 bits sont consacrés à la teinte primaire **rouge** ;
- 8 bits sont consacrés à la teinte primaire **vert** ;
- 8 bits sont consacrés à la teinte primaire **bleu**.

C'est le codage utilisé dans de nombreux périphériques numériques (Scanners, appareils photos, écrans...). Le principe est simple, un octet sert à coder un nombre compris entre 0 et 255 (d'où 256 valeurs) qui correspond à la valeur de la composante rouge, verte ou bleue du pixel (0 étant l'absence de cette composante, 255 la saturation). Le mélange de ces trois composantes donne la couleur finale.

> **Application :** ouvrir « Paint » et sélectionner Couleurs, Modifier les couleurs, Définir les couleurs personnalisées. En cliquant sur une des couleurs de base puis dans la palette de couleurs on voit les trois composantes RVB varier. On voit aussi les trois composantes d'un autre système de codage très utilisé en informatique et proche de la perception visuelle humaine, le codage TSL (pour teinte, saturation, lumière).

Mais attention ! Émerveillés par tant de couleurs, on peut oublier que plus le nombre de bits pour coder un pixel est grand, plus la taille en mémoire de l'image le sera : en 32 bits, un pixel a besoin de 4 octets pour être codé... Voici donc quelques exemples :

- Une image de 800x600 en 16 couleurs (4 bits) fait …… bits = …… octets = ……. ko
- Une image de 800x600 en 256 couleurs (8 bits) fait …… bits = ……. octets = …… ko
- Une image de 800x600 en 16 millions de couleurs (24 bits) fait …… bits = ……. octets = ……. Mo
- Avec un affichage réglé à 1280 x 1024 et 32 bits (ce qui est courant pour un simple moniteur 17" actuel), l'image du papier-peint (fond d'écran) du bureau aura une taille d'à peu près ……

Heureusement des **algorithmes de compression** permettent de réduire considérablement la taille des fichiers.

On se contentera parfois de **8 bits** par pixels, pour coder une *palette* de **256 couleurs** et afficher des images très correctes et moins volumineuses (Chercher définition de palette).

Formats d'image

Un format d'image est une représentation numérique de l'image, associée à des informations sur la façon dont l'image est codée et fournissant éventuellement des indications sur la manière de la décoder et de la manipuler.

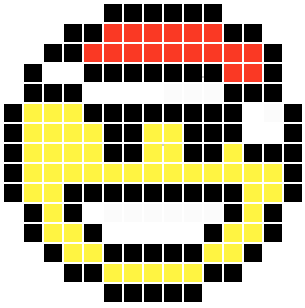
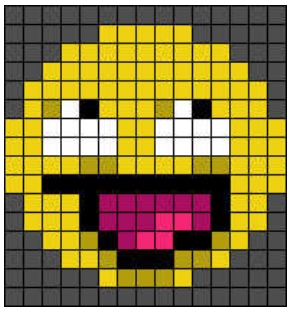
Tableau comparatif

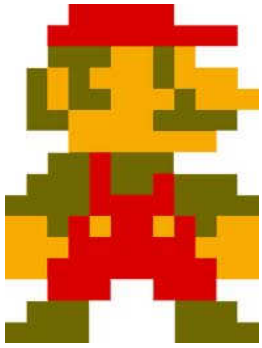
	Type (matriciel/ vectoriel)	Compression des données	Nombre de couleurs supportées
JPEG	matriciel	Oui, réglable (avec perte)	16 millions
JPEG2000	matriciel	Oui, avec ou sans perte	32 millions
GIF	matriciel	Oui, Sans perte	256 maxi (palette)
PNG	matriciel	Oui, sans perte	Palettisé (256 couleurs ou moins) ou 16 millions
TIFF	matriciel	Compression ou pas avec ou sans pertes	de monochrome à 16 millions
SVG	*vectoriel*	*compression possible*	*16 millions*

PhotoFiltre et les Pixels :

1. Ouvrir un nouveau fichier, choisir la hauteur et la largeur (15*15 par exemple). Quelle remarque peut-on faire ?.....
2. Enregistrer trois fois dans « Mes documents » le fichier sous le nom : PixelArt.jpg, PixelArt.bnp et PixelArt.bmp. Noter à chaque fois la taille du fichier (qui peut être obtenue en passant la souris sur l'icône du fichier ou en cliquant avec le bouton droit et en sélectionnant « Propriétés »). Consigner les valeurs obtenues dans le tableau ci-dessous. Pourquoi remarque-t-on une différence entre la taille sur le disque et la taille affichée ?
3. Après avoir agrandi le carré (zoom à 1600%), remplir entièrement le carré en noir puis en jaune. Prenez soin de noter à chaque fois après l'avoir enregistré la taille du fichier. Remplir le carré au quart puis à la moitié et enfin au trois quarts (environ bien sûr). Que remarque-t-on ?

   .....
4. Dessiner pixel par pixel un dessin de votre choix (smiley, pacman ….). Utiliser les couleurs que vous voulez. Noter la taille de l'image dans le tableau.



Format	.jpg	.png	.giff
Carré blanc			
Carré noir - jaune			
Un quart colorié			
La moitié			
Trois quarts coloriés			
Multicolore			
Mon dessin			