Introduction au codage multimédia et la compression

Sebastien.Kramm@univ-rouen.fr

IUT SRC Rouen

2012-2013

Sommaire

Généralités

- Un fichier informatique est un ensemble d'octets, stocké sur un support de stockage de masse.
- On distingue les fichiers texte (au contenu lisible par l'homme), et les fichiers binaires.
- La machine ne fait absolument **aucune distinction** (mais l'OS [1] peut "présenter" les données d'une certaine façon).
- Le **type** du fichier peut être reconnu par l'OS par plusieurs mécanismes :
  - l'**extension** du nom du fichier (`machintruc.doc`, `bidule.exe`, ...)
  - Une analyse de son contenu (Mac/Linux only...)
  - Un type-MIME, rattaché au fichier en tant que métadonnée (transfert de fichier par mail ou http).

1. OS : *Operating System* (système d'exploitation)

Forme d'un fichier

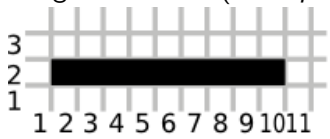
- En général, un fichier binaire est constitué d'un en-tête, et des données.

- Par exemple, une image png est identifiée par l'en-tête de 8 octets : `89 50 4E 47 0D 0A 1A 0A`
  suivi de un ou plusieurs "*chunks*" (blocs) de données, eux-même identifiés par un en-tête.
- On parlera de fichier *raw* ("brut") en cas de fichier binaire sans aucun en-tête.

Sommaire

Images bitmap vs. vectorielle

- 2 façon de concevoir une image :
  - Image matricielle (*bitmap*="champ de pixel") :

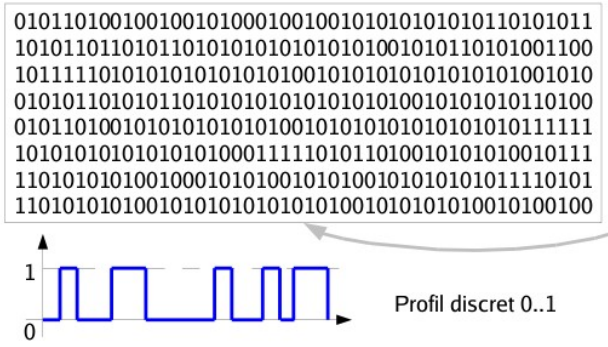


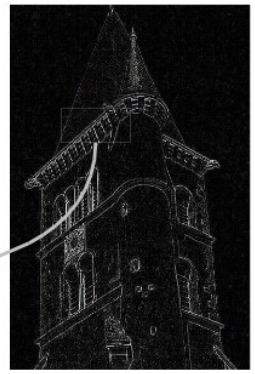
  - description du contenu :
    "une droite du point (2,2) au point (10,2)"

  ⇒ 2 types de format de fichier :
  - Format "bitmap" : bmp, gif, png, jpg, tiff, ppm, ...
  - Format vectoriel : svg (xml), ai (Adobe), ps & eps (postscript), ...
- Mais la visualisation (écran ou papier) sera fera sous forme de pixels.
- Selon le type d'image, l'un ou l'autre sera à privilégier :
  - photo, dessin complexe : bitmap
  - illustration, DAO, CAO, logos, ... : vectoriel

Images binaires

- Principe : un pixel = 1 bit

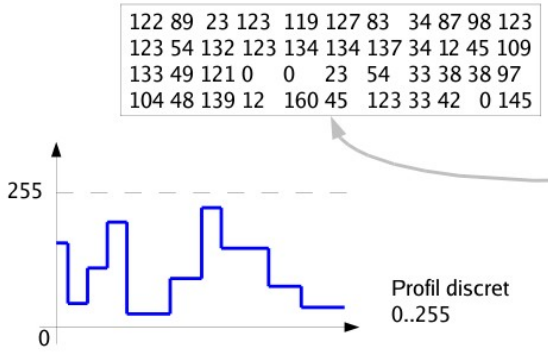




Source : J. Landré

Images en niveau de gris (*"grayscale"*)

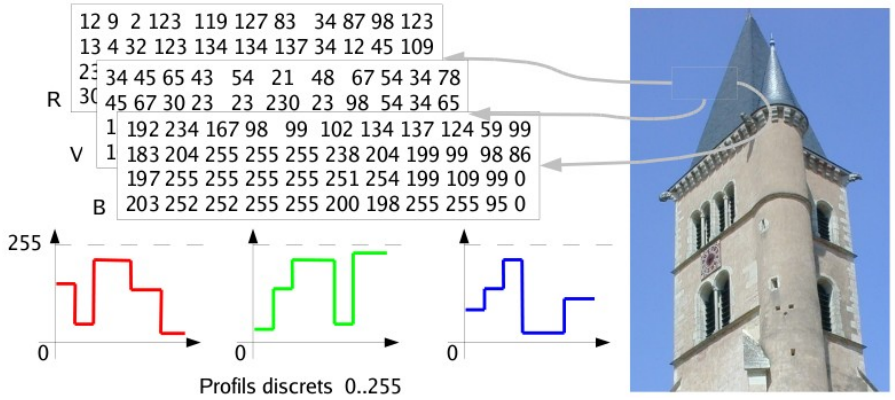
- Principe 1 pixel = 1 octet (256 valeurs)
  - Noir → 0
  - Blanc → 255



Source : J. Landré

Images couleurs

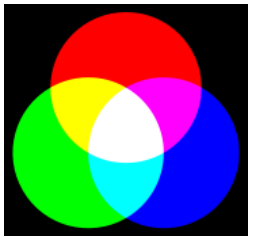
- Principe 1 pixel = 3 pixels RGB = 3 x 1 octet
- on parle d'espace "16 millions de couleurs" : $2^{3\times8} = 2^{24} = 16.777.216$



Source : J. Landré

Notion d'espace de couleur

- Une couleur réelle correspond à un signal complexe, composé de plusieurs longueurs d'ondes.
- Les machines en font une visualisation dans un espace de couleur donné :
- Ecrans : espace RGB (*Red Green Blue*)
- Impression : espace Jaune-Cyan-Magenta



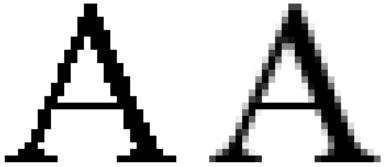
- De multiples autres espaces de couleurs peuvent être utilisés en interne.
  - TSL (Teinte-Saturation-Luminance), HSV en anglais
  - Lab
  - ...

Problématique du bitmap

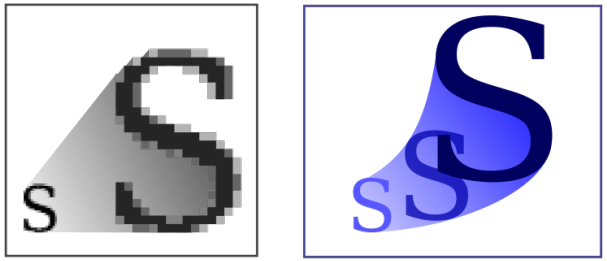
- En cas d'agrandissement de l'image ("zoom"), l'image conserve sa résolution initiale, et des déformations invisibles à petites échelle deviennent visible.

Aliasing d'image matricielle

- On peut compenser en procédant à un suréchantillonnage (*aliasing*)



- Les formats vectoriels éliminent cet inconvénient :



Taille des images de type bitmap

- Le poids des images est directement lié à leur **résolution**.
  - Une image 1000 x 1000 pixels contient... 1 million de pixels.
  - Si on divise par 2 la résolution (500 x 500), elle en contiendra 4 fois moins !
- Ceci est important dans un contexte "web", ou chaque image est chargée avec la page.
- Remarques : les formats d'images permettent parfois une compression des données.

Format SVG

- Les informations sont stockées sous forme lisible dans un fichier XML :

```
<marker,
     inkscape:stockid ="Arrow1Mend"
     orient ="auto"
     refY="0" refX="0"
     id="Arrow1Mend-7-65"
     style ="overflow:visible">
   <path
      inkscape:connector-curvature="0"
      id="path3276-4-7"
      d="M 0,0 5,-5 -12.5,0 5,5 0,0 z"
      style ="fill-rule:evenodd;stroke:#000000;stroke-
          width:1pt;marker-start:none"
      transform="matrix(-0.4,0,0,-0.4,-4,0)" />
</marker>
```


Format vectoriel

- Le fichier stocke l'image sous forme d'informations géométriques.
- L'affichage ou l'impression passe par un **moteur de rendu**, qui va "dessiner" les pixels à l'échelle désirée.
- Formats libre : SVG (xml)
- Logiciels : Inkscape, Adobe Illustrator, ...

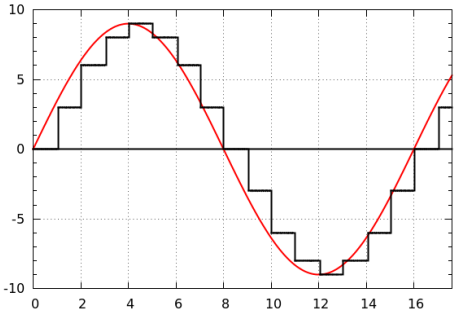
Sommaire

Comment ?

- Le son est un signal **monodimensionnel**, qui est stocké sous forme numérique :
  - CD (qualité hifi) : échantillonnage à 44,1 kHz, quantification linéaire sur 16 bits, sur 2 canaux (stéréo).
  - téléphone (bande limitée à 300-3,4 kHz) : échantillonnage à 8000 Hz, quantification logaritmique sur 8 bits.
- Problème du débit : CD : en 1s, il faut transmettre 44100 x 2 octets x 2 canaux = 176 ko = 1,4 Millions de bits / s. !
- Apparition de techniques de compression avec pertes dédiées à l'audio (mp3, aac, ogg, ...)

Codage PCM

- Un fichier audio non-compressé (format .WAV) code les échantillons en PCM (*Pulse Coded Modulation*).
- Par exemple, le signal suivant :



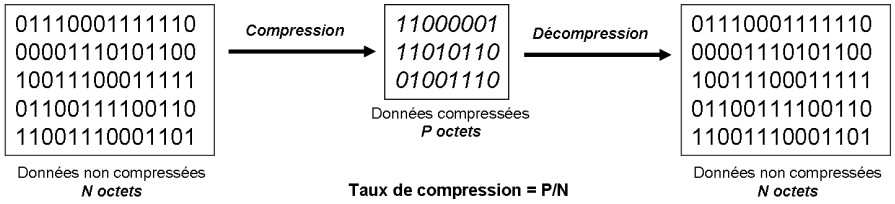
va être codé par la séquence de valeurs :
0 - 3 - 6 - 8 - 9 - 8 - 6 - 3 ...

Sommaire

Compression

Définition

- Compression : opération informatique qui consiste à transformer une suite de bits A en une suite de bits B plus courte, contenant les mêmes informations.
- Décompression : opération informatique inverse, permettant de retrouver l'information d'origine à partir de la suite de bits B.



Remarque : Ces opérations sont parfois transparentes pour l'utilisateur final. Exemple : lecture d'une vidéo : la décompression est **intégrée** dans le lecteur de média.

Comment compresser ?

- Il n'existe pas de méthode universelle, pour avoir les meilleurs résultats, on doit considérer :
  - La nature des données (image, texte, son, ...)
  - Le contexte général : quelle qualité attendue ? Combien de temps accorde-t-on à la compression ? à la décompression ?
- On distingue :
  - Compression sans pertes : on peut retrouver l'information de départ **exactement.**
  - Compression avec pertes : on accepte une **dégradation** de l'information, plus ou moins perceptible.

Compression sans perte

- On veut pouvoir récuperer l'information dans toute son intégrité : une comparaison bit-à bit doit donner donner un résultat identique.
- Trois techniques principales :
  1. compression par encodage des répétition de motifs (RLE),
  2. compression par codage de Huffman,
  3. compression par dictionnaire.

1 - Run Length Encoding (*RLE*)

- Idée générale : en cas de répétition d'une valeur, on stocke :
  - Le nombre de répétition
  - La valeur répétée
- Exemple 1 : soit le message AAAAAAAAAAAAAAA de longueur 15 ; on va le coder par les deux valeurs 15-A
  $\Rightarrow$ Taux de compression $= 15/2 = 7,5$
- Exemple 2 : soit le message AAAAAAbbbXXXXXt
  on va le coder par les 8 valeurs : 6-A 3-b 5-X 1-t
  $\Rightarrow$ Taux de compression $= 14/8 = 1,75$
- Exemple 3 : soit le message ABCDEF
  on va le coder par les 12 valeurs : 1-A 1-B 1-C 1-D 1-E 1-F
  $\Rightarrow$ Taux de compression $= 6/12 = 0,5$! ! !

1 - Run Length Encoding (*RLE*)

Conclusion

- Facile à mettre en oeuvre.
- N'est efficace que pour des des données possédant de nombreux éléments consécutifs redondants, notamment les images possédant de large parties uniformes.
- En pratique : format d'images BMP (Windows), Fax (CCIT)

2 - Codage de Huffman

- Le code est déterminé à partir d'une estimation des probabilités d'apparition des symboles de source, un code court étant associé aux symboles de source les plus fréquents (voir cours / codage).
- Inconvénient : il faut faire au préalable une lecture complète des données à compresser pour contruire la table des fréquences.
- Solution : codage de Huffman adaptatif : on recalcule la table des fréquences et on modifie le code en temps réel.

3 - Algorithmes de type Lempel-Ziv-Welch

- Principe général : on construit au fur et à mesure de la lecture du fichier un **dictionnaire**, via une fenêtre glissante.

le	petit	chat	est	mort	vive	le	petit	chat
le	petit	chat	est	mort	vive	1	2	3

(en général, la fenêtre est de taille fixe)

- Ce dictionnaire référence chaque mot rencontré, et si le mot est déja dans le dictionnaire, alors l'algorithme le remplace par sa position dans le dictionnaire.
- Le dictionnaire n'est **pas** transmis avec le fichier encodé : le décompresseur va le reconstruire de la même façon.
- De nombreuses variantes existent (LZ77, LZ78, LZW, LZMA, ...)

3 - Algorithmes de type Lempel-Ziv-Welch

Conclusion

- Très efficace, pour tout type de fichiers.
- Implémenté dans de nombreux logiciels.
- Paramétrable : ratio temps/taux de compression réglable.

- En pratique : format zip, rar, 7z, etc.
- Application : Le format d'image png implémente une compression qui couple une compression de type LZ avec un codage de Huffman.

Principe

- On admet ici que l'information reste pertinente malgré une dégradation de son contenu
- Applications : image, son (pas le texte ! ! !)

Compression avec pertes pour le texte

Qll st l clr d chvl blnc d'hnr lV

Images : compression JPEG

- Adapté aux fichiers de type "photo"
- Fondement théorique : transformation DCT (*Discrete Cosine Transform*).
- Facteur de compression réglable en %.
- Exemples (résolution : 190 x 190) :

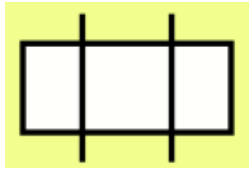
Sans compression, taille=53 ko

Compression q=50, taille=5,3 ko

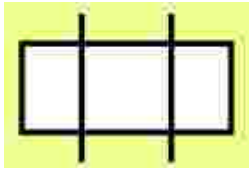
Compression q=10, taille=1,7 ko

Images : compression JPEG

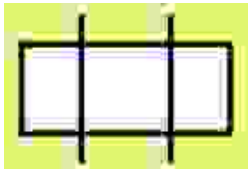
- Pas adapté aux images comportant beaucoup de zones homogènes ("aplats"), type logo ou schémas.



Sans compression



Compression q=10



Compression q=2

Son : compression MP3

- Nom exact : MPEG-1/2 Layer 3, correspond à la spécification sonore du standard vidéo MPEG-1/MPEG-2.
  (MPEG : *Moving Picture Experts Group*)
- Utilise les imperfections du système auditif humain pour réduire le volume des informations.
- Utilise l'effet de masque :
  - Masque fréquentiel : l'audition humaine analyse le son perçu en bandes de fréquences. A l'intérieur de ces bandes, un signal peut en masquer un autre d'intensité moins élevée.
  - Masque temporel : Un signal d'intensité elevée va masquer les sons qui suivent pendant quelques ms.

  $\Rightarrow$ La compression MP3 supprime/atténue ces signaux qui sont masqués par d'autres.