

Images, couleurs et numérisation

Lycée Monge, Chambéry

2 Décembre 2015

Ce document est mis à disposition selon les termes de la licence Creative Commons "Attribution - Pas d'utilisation commerciale - Partage dans les mêmes conditions 4.0 International".



Table des matières

Comment voyons nous ?

- L'oeil : une perception physiologique sans analyse.
- Le cerveau : analyse des données renvoyées par l'oeil.

Quatre problématiques.

- Représenter des images.
- Capter des images.
- Coder les images (à émettre ou captées).
- Analyser des images pour leur donner du sens.

Problématique 1 - Représenter des images

Nécessité de mieux connaître le fonctionnement de l'oeil.

Un fait important, les photorécepteurs : les cônes et les bâtonnets.

- Les cônes pour la couleur - Intensité lumineuse correcte.
 - Environ 7 millions principalement au centre de la rétine.
 - Sensibles à trois type de couleurs : bleu (420 nm), vert (535 nm) et jaune-rouge (570 nm).
- Les bâtonnets - Faible intensité lumineuse.
 - Environ 120 millions principalement en périphérie de la rétine.
 - Incapables de différencier plusieurs teintes.
 - Bien plus sensibles à la lumière que les cônes (vision nocturne).

Problématique 1 - Représenter des images

Solution technique adoptée :
émission de lumières rouge, verte et bleue « mélangées ».
De petits émetteurs pour chacune des trois couleurs.

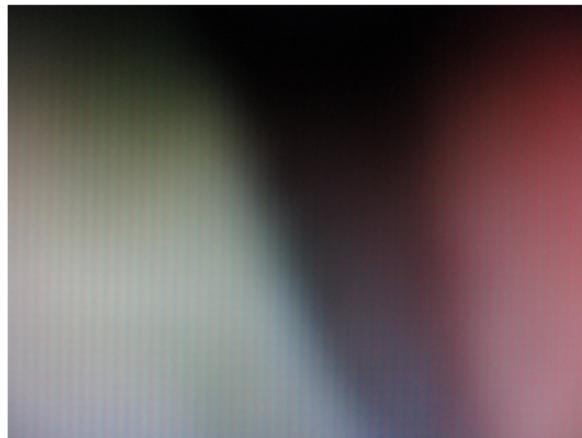


Photo en gros plan d'une télévision

Problématique 1 - Représenter des images

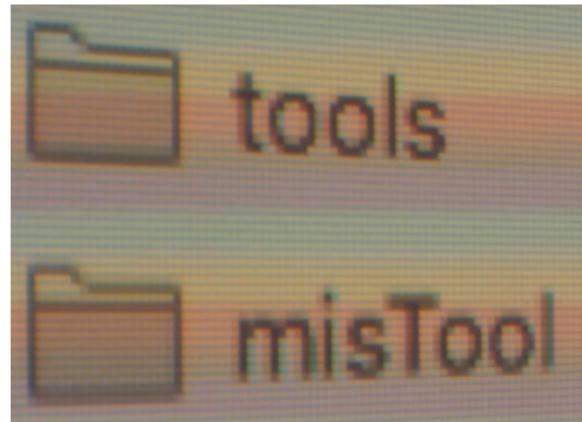


Image extraite d'une vidéo filmant le rendu d'un vidéo projecteur
(le fond était visuellement blanc)

Problématique 2 - Capter des images

Principe « inverse » de la représentation.

- Utilisation de petits capteurs spécialisés dans une couleur.
- Application des résultats de la physique quantique.
- Mesures électriques traduites en nombres.

Problématique 3 - Coder les images

- Comment coder une image ?
- Quelles conventions utilisées ?
- Nécessité d'établir des standards.

Problématique 4 - Analyser des images

Problème difficile. Pourquoi ?

Imaginons que l'on code par 1 le noir et 0 le blanc.

Seriez-vous me dire ce que représente l'image noir et blanc de dimension 13 « cases » horizontalement sur 17 verticalement qui est codée ci-dessous ?

```
000000000000000000001000000000000111000000000101010000001000001000001  
000000010001110000011100101111110100100100010010010010010010010010010  
0010010010010001001001000100100100100011000001100000111111  
10000000000000000000
```

Problématique 4 - Analyser des images

Réorganisation des chiffres (respect des dimensions).

```
00000000000000  
0000001000000  
0000011100000  
0000101010000  
0001000001000  
0010000000100  
0111000001110  
010111111010  
0100100010010  
0100100010010  
0100100010010  
0100100010010  
0100100010010  
0011000001100  
0001111110000  
00000000000000
```

Problématique 4 - Analyser des images

Voici l'image (ajout du quadrillage pour les sceptiques).



On devine un crayon... Pas si simple même en noir et blanc !

Imaginez les problématiques soulevées avec des images en couleur.

Problématique 4 - Analyser des images

Attention ! Codons par # le noir et par . le blanc.

On obtient visuellement un crayon.

```
.....  
.....#.....  
....###....  
....#.#.#....  
...#.....#...  
..#.....#..  
.###.....###.  
.#.#####.#.  
.#.#.##.#..#.  
.#.#.##.#..#.  
.#.#.##.#..#.  
.#.#.##.#..#.  
.#.#.##.#..#.  
.#.#.##.#..#.  
..##.....##..  
....#####...  
.....
```

Problématique 4 - Analyser des images

Ceci ne signifie pas que ce nouveau codage est meilleur.

Pourquoi ? Tout est dans le « visuellement ».

Problématique 4 - Analyser des images

Ne pas oublier qu'un ordinateur est juste un calculateur !

Quel codage choisir qui réponde aux contraintes technologiques ?

- Images représentées.
 - Quels émetteurs rouges, verts et bleus activés ?
 - Quelle intensité pour un émetteur donné ?
- Images captées.
 - Quantifier les mesures physiques.
- Nécessité de définir des conventions.

Le modèle RVB (Rouge-Vert-Bleu) ou RGB (Red-Green-Blue).

- Souhait de modéliser les outils physiques utilisés.
- Pour chaque couleur : valeurs allant de 0 (intensité nulle) à 255 (intensité maximale).
- Combien de chiffres nécessaires pour un codage binaire ?
 $256 = 2^8$ donc besoin d'au moins $3 \times 8 = 24$ chiffres en écriture binaire.

Coder les couleurs

Exemples de couleur RGB avec la notation (Red , Green , Blue).

- Quelle couleur correspond à (255, 0, 0) ?

Avec L^AT_EX, on obtient  qui est un rouge.

- Quelle couleur correspond à (0, 255, 0) ?

Avec L^AT_EX, on obtient  qui est un vert.

- Quelle couleur correspond à (0, 0, 255) ?

Avec L^AT_EX, on obtient  qui est un bleu.

- Quel type de couleur correspond à (200, 200, 200) ?

Avec L^AT_EX, on obtient  qui est un gris.

Plus généralement, 256 nuances de gris grâce à (n, n, n).

Coder les couleurs

Exemples de couleur RGB avec la notation (Red , Green , Blue).

(la suite)

- Comment obtenir du blanc ?
 $(255, 255, 255)$ nous donne avec \LaTeX : .
- Comment obtenir du noir ?
 $(0, 0, 0)$ nous donne avec \LaTeX : .
- Proposons un codage correspondant à un jaune.
Mélangeons du rouge et du vert (pas simple a priori).
 $(255, 255, 0)$ nous donne avec \LaTeX : .

Coder les couleurs

Une autre convention pour le format RGB.

$256 = 2^8 = (2^4)^2 = 16^2$ donc deux chiffres en écriture hexadécimale pour chacune des couleurs.

Collage successif de ces écritures dans l'ordre R-G-B.

$3 \times 2 = 6$ chiffres au total pour les trois couleurs.

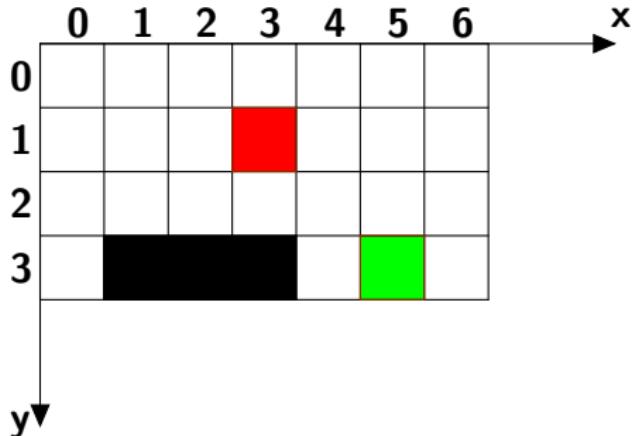
Par exemple, $(0, 141, 255)$ s'écrit 008DFF ou juste 8DFF en utilisant $141 = 128 + 13 = 8 \times 16 + 13$.

Un autre exemple : $(30, 173, 5)$ s'écrit 1EAD05.

Repérage des émetteurs et des récepteurs

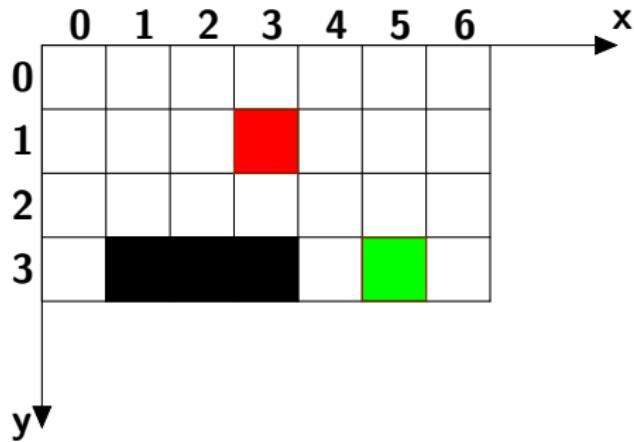
Reste à indiquer où une couleur sera activée, ou bien à savoir où elle a été mesurée (quantifiée).

Attention ! Convention des écritures manuscrites « latines » ou de la machine à écrire.



Graphique produit avec GeoGebra

Repérage des émetteurs et des récepteurs



Cellule **rouge** en $(x, y) = (3, 1)$.

Cellule **verte** en $(x, y) = (5, 3)$.

Couleur des cellules en $(1, 3)$, $(2, 3)$ et $(3, 3)$: le noir.

Formats ASCII

Caractéristiques générales.

- Formats ASCII « manipulables à la main ».
 - Windows : utilisation de Notepad++.
 - Linux : Geany.
 - Mac OS : Atom.
- Formats pris en charge par Gimp.
- Trois versions.
 - Portable Bitmap (PBM) : image en noir et blanc.
 - Portable Graymap (PGM) : image en niveau de gris.
 - Portable Pixmap (PPM) : image en couleur.
- Pour ceux présentés, pas de compression : les infos de chaque pixel directement dans le fichier image.

Formats ASCII : Portable Bitmap (PBM)

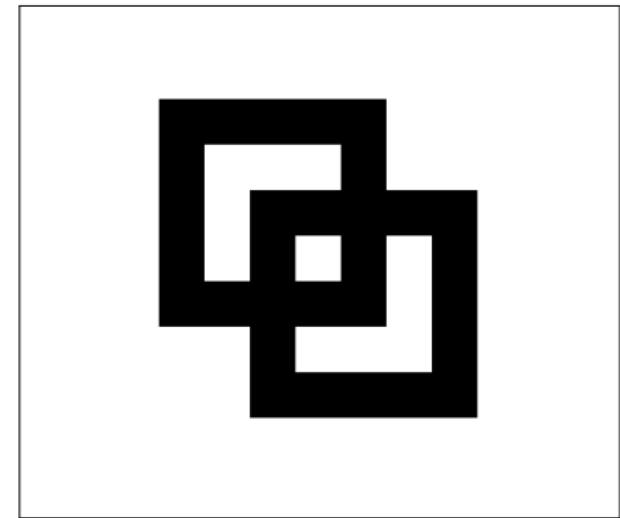
Spécifications techniques du fichier.

- Extension du fichier : .pbm.
- 1ère ligne ne contenant que le texte P1.
- 2ème ligne : dimensions sous la forme LARGEUR HAUTEUR.
- À partir de la 3ème ligne.
 - 1 code le noir et 0 le blanc.
 - Espaces possibles mais pas indispensables.
 - Pas plus de 70 caractères par ligne.
- Des commentaires : texte précédé par le symbole #.

Formats ASCII : Portable Bitmap (PBM)

Un exemple.

```
P1      # Image en noir et blanc
13 11  # Largeur = 13 Hauteur = 11
00000000000000
00000000000000
0001111100000
0001000100000
0001011111000
0001010101000
0001111101000
0000010001000
0000011111000
00000000000000
00000000000000
```



ATTENTION !
Image encadrée et agrandie.

Formats ASCII : Portable Graymap (PGM)

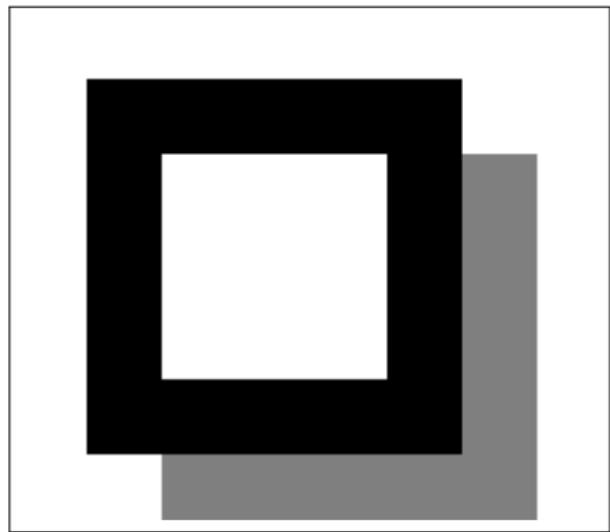
Spécifications techniques du fichier.

- Extension du fichier : .pgm.
- 1ère ligne ne contenant que le texte P2.
- 2ème ligne : dimensions sous la forme LARGEUR HAUTEUR.
- 3ème ligne : MAX la valeur maximale de niveau de gris.
On doit avoir $\text{MAX} \leqslant 65\ 536$.
- À partir de la 4ème ligne.
 - Gris codé de 0 pour le noir à MAX pour le blanc.
 - Espaces indispensables.
 - Pas plus de 70 caractères par ligne.
- Des commentaires : texte précédé par le symbole #.

Formats ASCII : Portable Graymap (PGM)

Un exemple.

```
P2 # Image en niveau de gris
8 7 # Largeur = 8 Hauteur = 7
10 # 11 Niveaux de gris
10 10 10 10 10 10 10 10 10
10 0 0 0 0 0 10 10
10 0 10 10 10 0 5 10
10 0 10 10 10 0 5 10
10 0 10 10 10 0 5 10
10 0 0 0 0 0 5 10
10 10 5 5 5 5 5 10
10 10 10 10 10 10 10 10
```



ATTENTION !
Image encadrée et agrandie.

Formats ASCII : Portable Pixmap (PPM)

Spécifications techniques du fichier.

- Extension du fichier : .ppm.
- 1ère ligne ne contenant que le texte P3.
- 2ème ligne : dimensions sous la forme LARGEUR HAUTEUR.
- 3ème ligne : MAX la valeur maximale pour le rouge, le vert et le bleu. On doit avoir $\text{MAX} \leqslant 65\,536$.
- À partir de la 4ème ligne.
 - Couleur codé au format R G B.
 - Espaces indispensables.
 - Pas plus de 70 caractères par ligne.
- Des commentaires : texte précédé par le symbole #.

Formats ASCII : Portable Pixmap (PPM)

Un exemple.

```
P3 # Image en couleur
7 2 # Largeur = 7 Hauteur = 2
10 # 11 niveaux pour R, V ou B
0 0 0 5 0 0 0 0 0 0 0 5 0 0 0 0 0 0 8 8 8 8 8 8 0 0 0
0 0 0 5 0 0 0 0 0 0 0 5 0 0 0 0 0 0 8 8 8 8 8 8 0 0 0
```



ATTENTION !
Image encadrée et agrandie.

« Non manipulables à la main » : utilisation de pillow (Python 3).

- Graphics Interchange Format (GIF)

- Extension du fichier : .gif.
- Transparence et compression (avec perte).
- Affichage progressif.

- Portable Network Graphics (PNG)

- Extension du fichier : .png.
- Transparence et compression sans perte.
- Affichage progressif.

- Joint Photographic Experts Group (JPEG)

- Extension du fichier : .jpg ou .jpeg.
- Pas de transparence mais compression réglable (avec perte).
- Affichage progressif.

Références

- « *Introduction au traitement d'images [2e édition]* » de Dian LINGRAND édité par Vuibert.
- « *Formation Lyon ISN 2015 : Modules & Représentation d'images* » de Nicolas-François BUYLE-BODIN.
- « *Numérisation [Cours ISN 2012-13]* » de Xavier OUVRARD.
- fr.wikipedia.org/wiki/Portable_pixmap
- fr.wikipedia.org/wiki/Graphics_Interchange_Format
- fr.wikipedia.org/wiki/Portable_Network_Graphics
- fr.wikipedia.org/wiki/JPEG