

Quantification

1 Introduction

But du TP

Le but du TP consiste à observer l'effet d'une compression par **quantification** des niveaux de chaque pixel d'une image bidimensionnelle (en niveau de gris ou en couleur (codage RGB ou HSV)) sur la qualité visuelle de l'image : la quantification peut être **uniforme** (Exercice 1) ou **adaptative** (Exercice 2). L'exercice 3 aborde la méthode de Floyd et Steinberg, perturbant le signal mais donnant le meilleur résultat visuel.

Manipulation des données

Le traitement des données se fera en *langage C*. Les images manipulées seront mises au format PGM (niveaux de gris) ou PPM (couleur RGB), qui sont caractérisés par :

1. un en-tête de **15 octets**¹ du type :

```
P5
350 286
255
```

où P5 désigne le type de données (ici du PGM; pour du PPM, P6 remplace P5) ;
350 286 spécifient la taille de l'image; et 255 est l'intensité maximale possible;

2. un **bloc de données** écrites en **binaire** (de haut en bas, et de gauche à droite) :
 - pour le format PGM : 1 octet lu correspond au niveau de gris entre 0 et 255 (codage en **unsigned char**) du pixel considéré ;
 - pour le format PPM : un triplet d'octets lu correspond aux trois valeurs RGB (un octet pour chaque couleur) du pixel considéré.

¹Pour observer cet en-tête de 15 caractères, il suffit d'ouvrir le fichier sous **emacs**. Si l'en-tête de votre image est de longueur différente, libre à vous de modifier le programme en conséquence.

2 Travail demandé

Preliminaires

Choisir une image de taille raisonnable, quel que soit son format, et sauver là au format PGM et au format PPM, par le biais de votre afficheur d'image préféré, suivant les disponibilités. Par exemple, si l'image à traiter s'appelle `mon_image.jpg`, sauver là sous `mon_image.pgm` et `mon_image.ppm`. Profitez-en pour vérifier la cohérence de la taille des deux fichiers `mon_image.pgm` et `mon_image.ppm`.

Exercice 1 : compression par quantification uniforme

- 1) Appliquer une transformation du type $pixel = n \times E(\frac{pixel}{n})$ à votre image PGM, où n est un entier fixé, désignant la largeur de l'intervalle de quantification (plus ce nombre augmente, plus le nombre de valeur possibles au résultat, à partir de la valeur initiale comprise entre 0 et 255, diminue).
- 2) Comparer le résultat obtenu avec l'image originelle. Faites différents essais pour des valeurs de n croissantes, et déduisez-en l'inconvénient de cette quantification.
- 3) Pour chaque image générée, visualiser l'histogramme, et comparer-le à celui de l'image de départ.
- 4) Conclure quant à la possibilité de compresser l'image d'origine par cette méthode de quantification.
- 5) Recommencer l'opération avec une image PPM.

Exercice 2 : compression par quantification adaptative

Afin de ne pas découper la plage d'intensité 0-255 de manière uniforme sur l'image (Exercice 1), on se propose d'écrire un algorithme permettant de s'adapter à la forme de l'histogramme, i.e. de créer un découpage par intervalles de l'ensemble des niveaux de gris possibles de telle sorte que chaque interval concerne sensiblement autant de pixels que les autres (rappelons qu'une image de X pixel sur Y comporte au total XY pixels, et que donc une division en m niveau possible devrait idéalement laisser XY/m pixels par interval).

Ecrire une fonction (ou des fonctions) permettant d'effectuer ce type de compression par quantification, adaptative. Comparer avec l'approche précédente.

Exercice 3: quantification de Floyd-Steinberg

faire, à partir de l'exercice 1, une version utilisant le process de quantification de Floyd-Steinberg (voir le cours pour le détail, cours sur l'image fixe).