

# 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. La réalisation se fera en C.

## 2 Travail demandé

### Préliminaires

On se servira pour le chargement et la sauvegarde de l'image de la base de code précédemment réalisée pour le TP de stéganographie.

### 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). Rappel, en langage C, la division entre deux entiers donne un résultat entier, donc l'expression  $pixel = n \times E(\frac{pixel}{n})$  se ramène en C à  $pixel = n \times (\frac{pixel}{n})$
- 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 (vecteur de 255 entrées contenant pour chaque entrée le nombre de pixels du niveau de gris correspondant à l'entrée). Pour ce faire on va ici 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  $m$  niveaux de gris possibles devrait idéalement laisser  $XY/m$  pixels par interval).

Ecrire une fonction permettant d'effectuer ce type de compression par quantification, adaptative. visualiser le résultat sur votre image test. Comparer avec l'approche précédente.

### Exercice 3: quantification de Floyd-Steinberg

faire, à partir du code réalisé pour l'exercice 1, une version utilisant le process de diffusion d'erreur de Floyd-Steinberg.