

# Rapport TP4 - PRALG

## *Quadtree* et compression d'image

Dans le cadre du TP4 du module PRALG, j'ai développé des méthodes pour réaliser l'encodage et la compression avec perte d'image (en noir et blanc ou en couleur RGB) sous forme de *Quadtree* et de graphe orienté acyclique (DAG) correspondant, ainsi que pour le décodage de ses structures de données. Les méthodes développées sont listées et expliquées brièvement ci-après :

- **dfs()** : algorithme de parcours en profondeur qui permet de convertir une image en noir et blanc (soit carrée soit rectangulaire) en *Quadtree*.
- **quaddag()** : méthode qui convertit une image en noir et blanc (soit carrée soit rectangulaire) en graphe orienté acyclique. L'arbre partage la feuille noire et la feuille blanche (c'est à dire qu'une seule feuille noire et une seule feuille blanche pour toutes les feuilles d'arbre).
- **dfsLossyCompression()** : compression avec perte pour image en noir et blanc. Cette méthode a été développée sur base de **dfs()**.
- **dfsLossyCompressionRGB()** : compression avec perte pour image en couleur RGB.
- **sameLeaves()** : méthode qui vérifie si 4 feuilles de *Quadtree* sont identiques.
- **similarLeaves()** : méthode qui vérifie si 4 feuilles de *Quadtree* sont similaires (nuance inférieure au seuil).
- **thresholdAccordingToLength()** : méthode qui détermine le seuil de compression avec perte en fonction de taille de région.
- **quadtreeDecoding()** : réalisation de décodage de *Quadtree*.
- **nextPow2()** : méthode qui cherche pour une valeur donnée la puissance de 2 supérieur.
- **sizeQuadTree()** : calcul de la taille de *Quadtree* en octets dans le mémoire.

- **imageCompressionRatio()** : calcul du taux de compression par le nombre de feuilles d'arbre divisé par le nombre de pixels d'image.