Rapport de TP Quaddtree – Ilyass Ramdani

Questions

3. Notre fonction est construite de manière à prendre en input l'image, sa taille et renvoyer l'arbre de l'image encodé.

Cela se fait en deux étapes.

- Etape 1 : Descendre récursivement dans les sous régions carrées de l'image (NW, NE, SE, SW) jusqu'à la taille d'un pixel.
- Etape 2 : Une fois arrivé sur la taille du pixel, on remplit l'arbre de la valeur du pixel.
- Etape 3 : Si on est face à 4 feuilles de la même couleur, on remplace le nœud par une feuille de la couleur.

4. Tout comme notre fonction encodeImage, notre fonction Quaddag est construite de manière à prendre en input l'image, sa taille et renvoyer l'arbre de l'image encodé. La seule difference c'est que nous faisons pointer tous les pixels blancs vers le pixel blanc global et tous les pixels noirs vers le pixel noir global.

Aussi, ici nous sommes obligés d'ajouter protect_leaves_from_destruction pour ne pas delete le pointeur partagé.

5. Notre foction **quadtreeDecoding** nous permet de descendre recursivement dans l'arbre et de recreer l'image. Dès qu'on arrive à la taille de la feuille, on associe la couleur d la feuille à toute la région representée par la feuille.

6. L'image du cheval est encodée sur un arbre de 2060 nœuds. Sachant que chaque nœud est binaire, la taille de l'image compressée est seulement de 2060 bits ce qui est équivalent à 0.25ko.

On pourrait mesurer le taux de compression en faisant le calcul suivant :

1-Taille compressée/Taille initiale = (1-nb de nœuds/nb de pixels) *100%

7. L'image étant rectangulaire nous devons construire une fonction qui respecte les 4 étapes suivantes :

Hors de la fonction, on fait ce petit traitement :

- D'abord on prend W et H de la photo et on essaye de trouver le max de W et de H
- Disons que c'est H le max, alors on trouve la puissance de 2 la plus proche
- Une fois qu'on a trouvé une puissance de 2 supérieure, on applique notre fonction à une image implicitement définie comme de taille (2N, 2N)

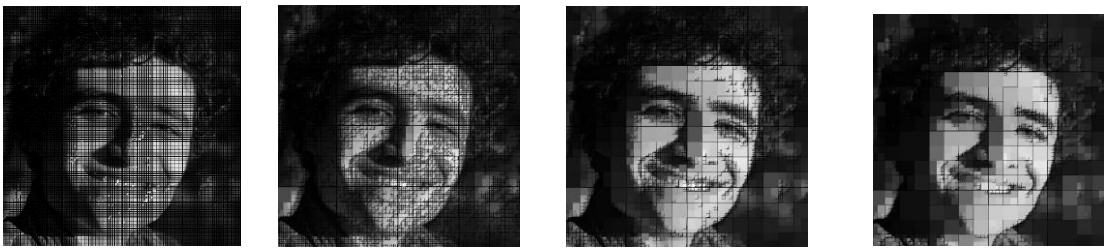
Dans la fonction :

- On descend récursivement dans les sous images comme dans Quaddag.
- Une fois arrivé à la taille du pixel
    - Si nos coordonnees (x,y) sont hors de l'image rectangulaire, on fait pointer sur le BLANC
    - Sinon on fait comme Quaddag.

8. Renvoi le tree de l'image encoded en faisant la moyenne des intensités des 4 pixels. Cela se decompose en deux etapes :

- Descendre Recursivement dans les sous regions carrees de l'image jusque la taille du pixel
- Une fois arrivé sur la taille du pixel, on rempli l'arbre de la valeur du pixel
- On introduit la fonction nearValueLeaves qui prend en argument les 4 feuilles pour retourner true si les valeurs des 4 feuilles sont inférieures à un certain seuil.

Voici le résultat des images pour un seuil qu'on a fait varier (de gauche à droite : 1, 3, 10, 20)



On voit clairement que plus le seuil est élevé moins les détails sont visibles. Et plus les changements de contrastes sont violents.

Pour faire dépendre le seuil en fonction de la taille des régions, on ajoute un seuil = seuil - 3 au niveau de la suppression des 4 feuilles « semblables ». Ainsi, en remontant les couches successivement, le seuil va décroitre.