

TD compression d'image

30/09/2021

1. Mise en jambe

Charger les images *Lena.png*, *Lena_comp.jpg*, et *Sierpinski.png*. Affichez-les.

Il faudra éventuellement transformer les images en uint8.

Affichez leur histogramme (fonction *imhist*). Que remarquez-vous ?

2. Entropie de Shannon

Écrire une fonction *entropy.m* qui calcul l'entropie d'un signal 1D ou 2D. Vous pourrez vous aider du calcul de l'histogramme. Calculer l'entropie des images *Sierpinski* et *Lena*. Attention aux probabilités nulles !

3. RLE

Écrire une fonction de « Run-Length Encoding » *rle.m*.

Vous pourrez vous aider du fichier *rle-empty.m*

4. RLD

Écrire une fonction de décodage *rld.m*.

Vous pourrez vous aider du fichier *rld-empty.m*

5. Parcours d'image

Écrire une fonction de parcours d'image (scan) qui transforme une image 2D en un signal 1D. Cette fonction doit permettre d'effectuer différents types de parcours : en ligne, ou en colonne.

6. Compression RLE

Compresser l'image *Sierpinski* avec le codage RLE. Calculez le taux de compression.

7. Transformée en cosinus discrète

- A l'aide de la fonction *dct2*, calculer la transformée en cosinus discrète (dct) de l'image *house256.png*, puis recréer l'image en utilisant la transformée inverse.
- Simplifier la dct en mettant à zéro les coefficients dont la valeur absolue est inférieure à un seuil T . Utilisez différentes valeurs de T ($T=0.1$, $T=2$, $T=50$).

8. Seuillage

- Créer une fonction *value_from_percent.m* pour calculer T en fonction d'un pourcentage de valeur donnée.
- Trouvez une valeur de pourcentage de coefficients gardés partir de laquelle vous observez une différence dans l'image reconstruite.

9. Pipeline

Compresser l'image *Sierpinski_binary* en utilisant le RLE sur la dct simplifiée. Calculez le taux de compression.

10. BONUS

Compléter la fonction *scan* pour prendre en compte le parcours en zigzag.