

---

# Compression 1 — SI 201

Avant de commencer le TP, il faut :

1. Une fois ouverte la session de travail, créer un répertoire sur le Bureau : si vous êtes sur Windows, H:\Desktop\TP
2. Ouvrir un navigateur Web et télécharger le matériel pour le TP  
<http://cagnazzo.wp.mines-telecom.fr/>
3. Extraire tous les fichiers des archives compressés dans le répertoire de travail.

**NB. Les comptes rendus doivent être envoyés avant la fin de la P2 à :**  
[cagnazzo@telecom-paristech.fr](mailto:cagnazzo@telecom-paristech.fr)

## 1 Quantification uniforme

Ouvrez le fichier `tp_qu.m`. Exécutez les parties du fichier avec `Ctrl+Entrée`. Complétez les parties manquantes comme demandé.

1. Chargez l'image `lena` et visualisez-la.
2. Affichez l'histogramme de l'image. Justifiez la commande utilisée.
3. Ouvrez la fonction `qu.m` et vérifiez qu'elle implémente un quantificateur uniforme (QU). Quel est le débit ? Qu'est-ce que `delta` représente ?
4. Tracez la courbe entrée/sortie du QU pour différents valeurs du débit `bitrate`. Quelle est la relation entre `bitrate` et `delta` ?
5. Quantifiez l'image de test avec 3 bpp. Calculez la distorsion (erreur quadratique moyenne) et le PSNR<sup>1</sup>. Affichez l'histogramme de l'image quantifiée.
6. Répétez les opérations au point précédent avec d'autres valeurs de débit.
7. Calculez la courbe débit-distorsion du QU appliqué à l'image de test. Utilisez des intervalles de quantification entiers entre 128 et 1. Commentez la façon d'évaluer `R` en fonction de `delta`. Quelle la pente de la courbe `R-PSNR` à haut débit ? Pourquoi ?
8. Répétez l'expérience précédente (courbe `RD`) en utilisant un source aléatoire uniforme iid  $\mathcal{U}(0, 255)$ . Utilisez la fonction `rand`.

## 2 Quantification prédictive

Ouvrez le fichier `tp_qp.m`. Exécutez les parties du fichier avec `Ctrl+Entrée`. Complétez les parties manquantes comme demandé.

1. Ouvrez la fonction `prediction.m` et décrivez son fonctionnement.
2. Ouvrez la fonction `reconstruct.m`. Qu'est-ce que `reconstruct(prediction(x))` donne comme résultat ? Pourquoi ?
3. Effectuez la prédiction de l'image de test et affichez l'histogramme de l'erreur de prédiction. Commentez.
4. Deux quantificateurs uniformes sont fournis : `qu.m` et `qu_dz.m`. Quel est la différence entre les deux quantificateurs ? Tracez la courbe entrée-sortie du quantificateur uniforme à zone morte (*deadzone*).
5. Quantifiez l'erreur de prédiction avec `qu.m` et avec `qu_dz.m`. Quel est le quantificateur le mieux adapté à ce cas ? Pourquoi ?

---

1.  $\text{PSNR} = 10 \log_{10} \frac{255^2}{\text{MSE}}$

6. Calculez les variances de l'image et de l'erreur de prédiction ; calculez le gain de prédiction.
7. Quantifiez l'image de test avec  $\Delta=3$ . Calculez la distorsion (erreur quadratique moyenne) et le PSNR. Calculez l'MSE sur l'image et sur l'erreur de prédiction. Les deux valeurs sont-elles identiques ? Pourquoi ?
8. Calculez la courbe débit-distorsion du QP appliqué à l'image de test. Utilisez des intervalles de quantification entiers entre 128 et 1. Comparez au cas QU.
9. La validité des tests faits ici est limitée par le fait que nous n'avons pas pris en compte l'effet du *codage sans pertes* qui suit la quantification et qui réduit le débit de codage. Une estimation du vrai débit de codage s'obtient avec le calcul de l'entropie. Retracer donc les courbes RD du QU et du QP en remplaçant l'estimation de R avec l'entropie de la distribution de l'image (pour le QU) et de l'erreur de prédiction (pour le QP). Tirez vos conclusions<sup>2</sup>.

### 3 Quantification et transformée

Ouvrez le fichier `tp_qt.m`. Exécutez les parties du fichier avec `Ctrl+Entrée`. Complétez les parties manquantes comme demandé.

1. Calculez et affichez la transformée de l'image entière. Commentez le résultat.
2. Pour calculer la transformée en blocs, on utilise la commande `blkproc.m`. Tapez `help blkproc.m` pour justifier cette démarche.
3. Stockez les blocs de la transformée dans un "cube" `blockSize × blockSize × numBlocks`. On considère l'ensemble des coefficients d'un bloc comme la réalisation d'un vecteur aléatoire de `blockSize2` éléments. Calculez le gain de codage associé à la transformée via les variances des éléments de ce vecteur.
4. Quantifiez l'image transformée avec `qu_dz.m` et  $\Delta=20$ . Calculez la transformée inverse et vérifiez l'isométrie de la TCD. Affichez l'image après transformée - quantification - transformée inverse, calculez son PSNR.
5. Calculez la courbe débit-distorsion de l'approche Q+T appliquée à l'image de test. Utilisez des intervalles de quantification entiers entre 128 et 1. Utilisez l'entropie pour l'estimation du débit et comparez aux cas précédents.
6. Proposez une méthode de codage qui unit prédiction et transformée. Estimez la courbe RD de cette méthode.
7. Mettez en oeuvre une version simplifiée de JPEG avec TCD  $8 \times 8$ , quantification à zone morte avec le tableau de quantification  $Q$ . Le codage sans perte de JPEG, relativement complexe sera remplacé par une estimation du débit par l'entropie des coefficients quantifiés. Utilisez le facteur de qualité pour générer une courbe RD.

$$Q = \begin{array}{|c|c|c|c|c|c|c|c|} \hline 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ \hline 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ \hline 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ \hline 14 & 17 & 22 & 29 & 51 & 87 & 81 & 61 \\ \hline 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ \hline 24 & 35 & 55 & 64 & 81 & 104 & 111 & 90 \\ \hline 49 & 63 & 78 & 87 & 101 & 121 & 120 & 100 \\ \hline 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \\ \hline \end{array}$$

<sup>2</sup> Une autre limitation de ces simulations est que les prédicteurs sont calculés avec les échantillons de l'image originale. Le décodeur n'aurait pas accès à ces données. Pour simplicité on néglige cet aspect.