**TP 2**

**COMPRESSION ET TRANSMISSION D’IMAGES FIXES**

**Master 1 RT et ST**

1. **Avant propos**

La compression des images est une étape essentielle aussi bien lors du stockage d’image que lors de sa transmission à travers des canaux de bandes passantes limitées. Plusieurs techniques et normes de compression d’images fixes ont été proposées et adoptées depuis presque une quarantaine d’années. Parmi les standards les plus utilisées jusqu’à nos jours la norme JPEG (Joint Photographic Expert Group) mais aussi la norme JPEG2000 et les techniques telles que EZW ou encore SPIHT. Alors que la norme JPEG est basée sur la transformation DCT (Discret Cosine Transform) JPEG2000, EZW et SPIHT sont basées sur la transformation DWT (Discret Wavelet Transform)

Toutes ces normes et ces techniques de compression d’images sont de type avec perte, car les images reconstruites (après compression / décompression) subissent une certaine distorsion par rapport à l’image originale au profit d’un taux de compression élevé. Pour cette raison les méthodes de compression d’images sont évaluées à l’aide de plusieurs critères et métriques dont certaines sont subjectives, basées sur l’appréciation visuelle des êtres humains comme MOS (Mean Opinion Score), et d’autres objectives où basées sur des expressions mathématiques telles que MSE (Mean Square Error), PSNR (Peak Signal to Noise Ration) ou encore SSIM (Structural SIMilarity). Tous ces critères vont dépendre bien évidemment du taux de compression CR (Compression Ratio) représentant le rapport entre la taille de l’image originale et la taille des données compressées. Souvent, au lieu d’utiliser le ***CR=Taille de l’image originale/taille des données compressées***, on utilise plutôt le débit binaire ou Bitrate noté bpp (bit per pixel) qui représente une mesure absolue du nombre moyen de bits nécessaires pour coder chaque pixel d'image en tenant compte de la taille de la donnée compressée obtenue. On peut dire tout simplement que

**bpp = Taille des données compressées / Nombre de pixels total = 8 / CR**

Où 8 représente généralement le nombre de bits original par pixel d’une image en niveaux de gris non compressée.

Enfin, il faut noter que d’autres critères entrent aussi en jeu pour évaluer les performances d’une techniques de compression d’images en particulier les critères liés à la complexité de calcul de la méthode de compression. En effet, la complexité de calcul et un critère pertinent car il influe beaucoup sur la possibilité d’utiliser une technique de compression dans la pratique surtout pour un travail en temps réel ou encore pour l’économie énergétique du système sensé exécuter ces algorithmes.

1. **Objectifs et outils**

Pour étudier la compression/décompression, codage/décodage et transmission des images numériques, nous allons nous intéresser à un outil de compression d’images et de vidéo **VcDemo**.

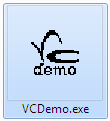
VcDemo est un logiciel de compression d’images et de vidéo, dédié à l’enseignement et développé par **Delft University of Technology**. Il fonctionne sous Windows et peut être téléchargé à partir du lien suivant :

<http://siplab.tudelft.nl/content/image-and-video-compression-learning-tool-vcdemo>

Une aide est fournie avec le logiciel et une version pdf de cette aide est également disponible.

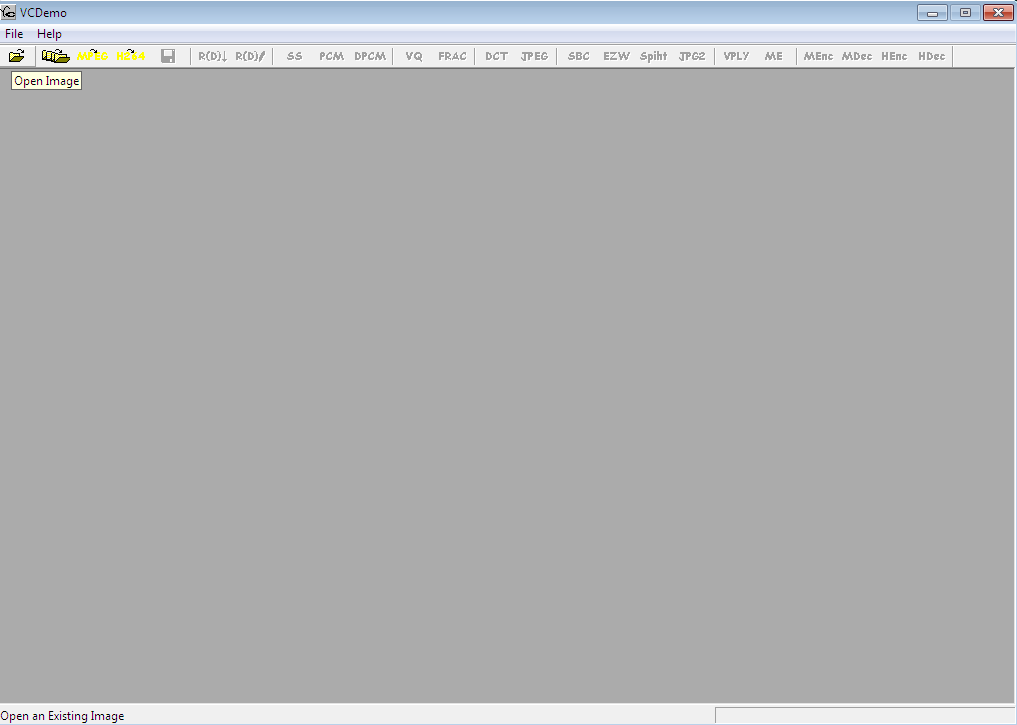
1. **Exécution de VcDemo**

Après avoir décompressé VcDemo, de préférence dans un dossier séparément, il faut exécuter l’application (l’exécutable) désigné par le fichier suivant :



A la première exécution de VcDemo.exe, un message d’erreur apparaît, puis une fenêtre apparait où vous devez renseigner les 4 répertoires comme étant le répertoire de travail (en faisant parcourir jusqu’au dossier choisi), où se trouve les images et vidéos à traiter (par exemple le dossier où vous avez décompressé votre outils **VcDemo**. Fermer le programme et redémarrer. Ces répertoires peuvent changés par la suite dans Help/Settings.

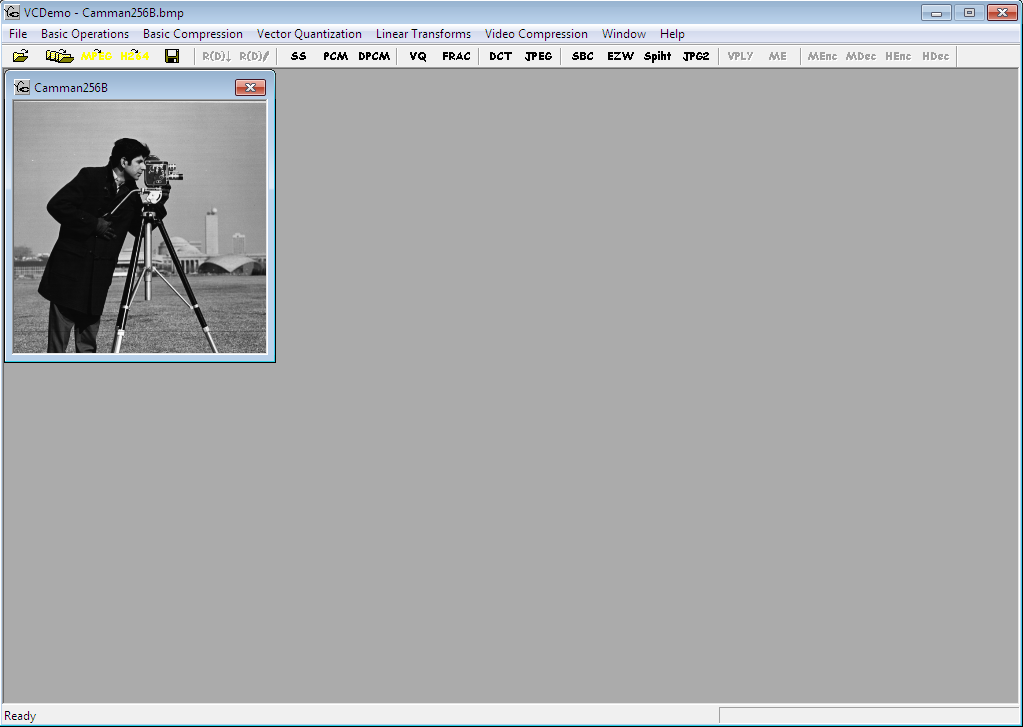
Dans ce TP, nous allons nous intéresser uniquement aux images et non à la vidéo.



**Pour ouvrir une image on clique sur**

**Puis on fait parcourir jusqu’au Dossier qui contient les images tests**

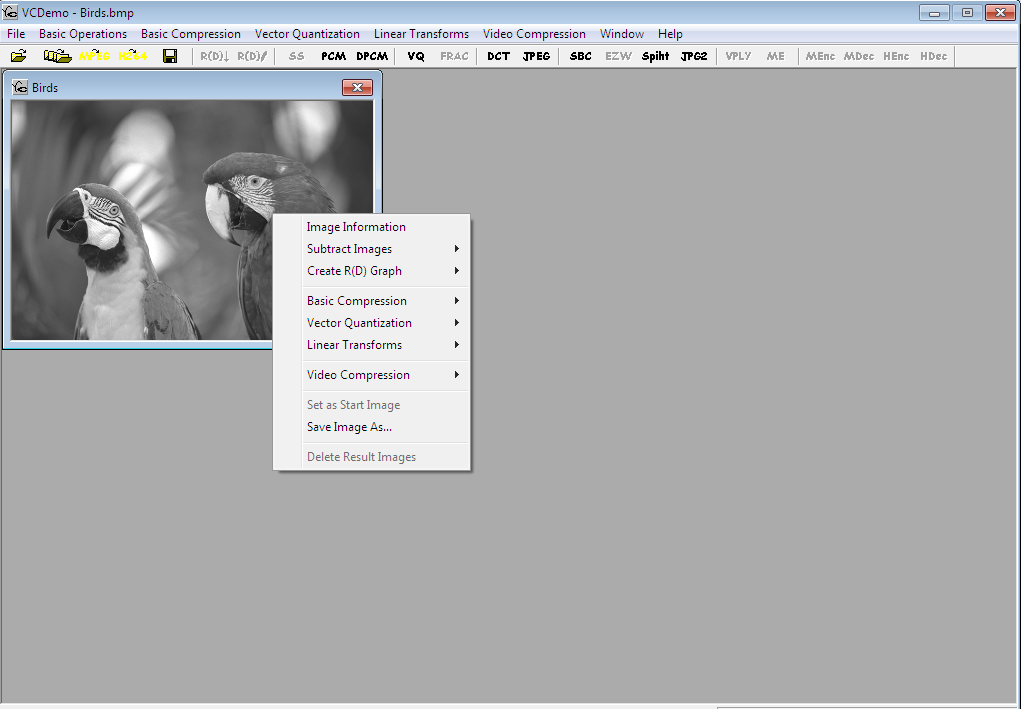
Par exemple, ci-dessous, on charge l’image Cameraman

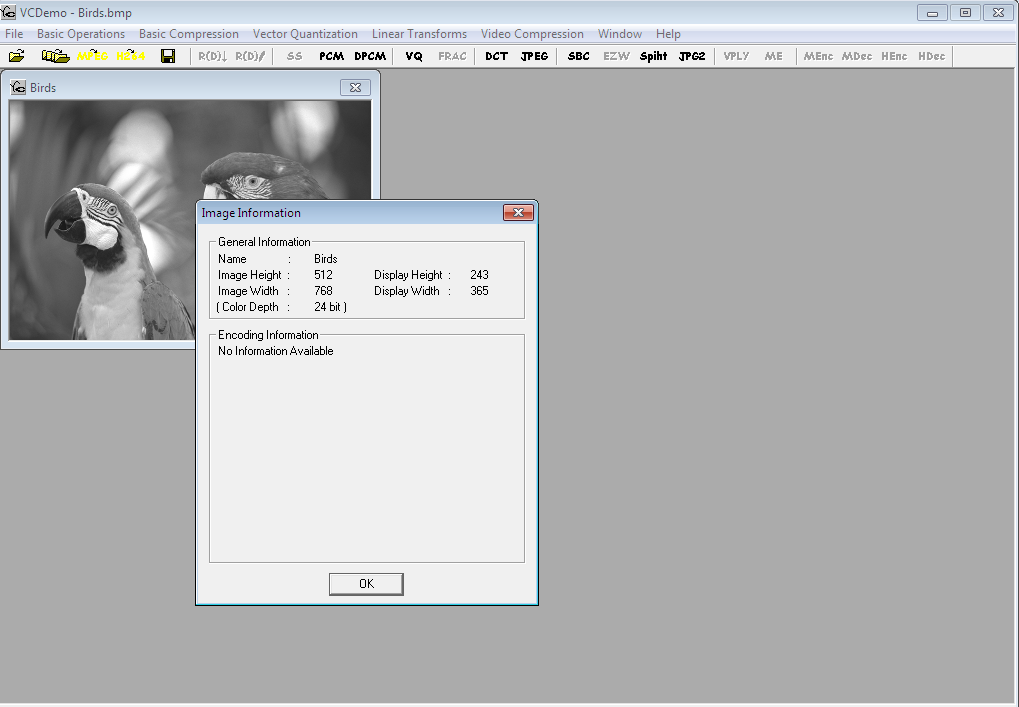


Dans le menu de ce logiciel, nous avons alors différentes fonctions et instructions que nous pouvons appliquer sur l’image chargée. Les résultats seront donnés instantanément

**Travail à faire**

1. Chargez plusieurs images différentes (Cameraman, Bird, …etc) mêmes certaines que vous pouvez apportez vous-même. Pour chacune il faut lire ses propriétés et ses informations. Ceci en faisant un clic droit sur l’image affichée. Alors un menu contextuel apparait dont ‘’Image Information’’. Alors en cliquant sur ‘’Image Information’’ une fenêtre apparait pour vous donnez les informations demandées. Comprendre ces informations et les notez pour chacune des images testées.

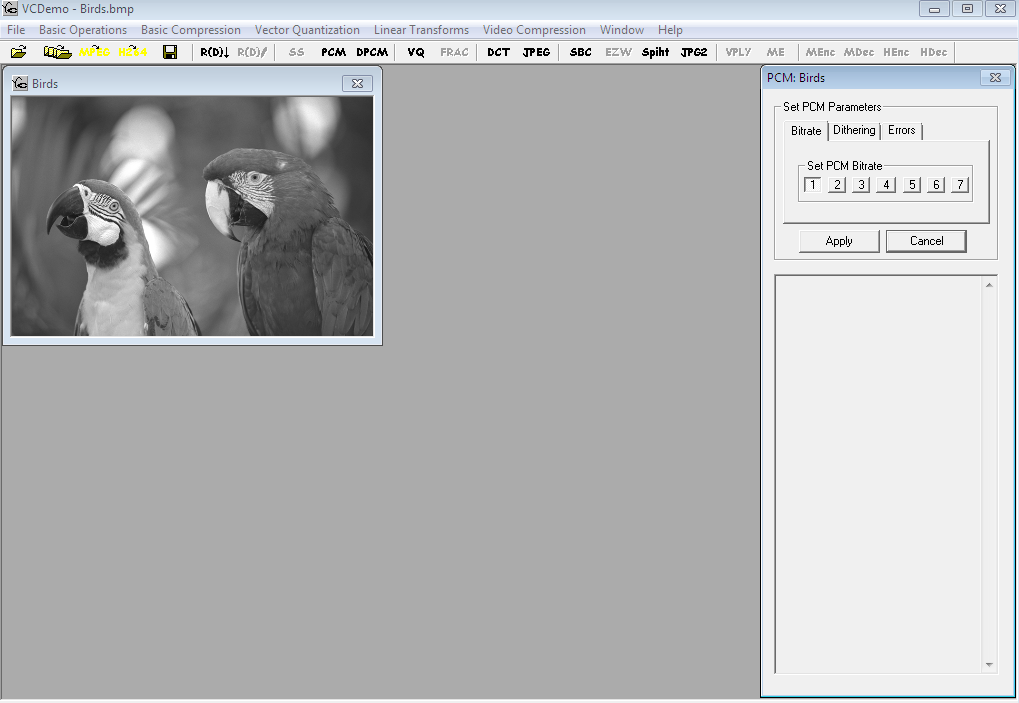




1. **Compression de base**

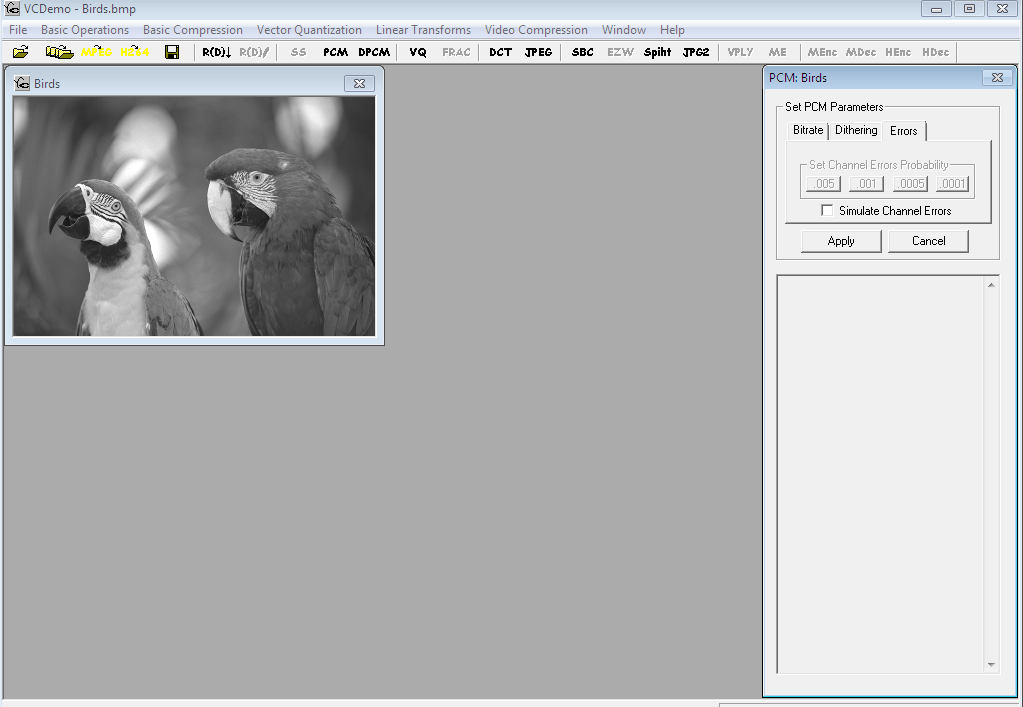
Refaire le même clic droit sur les différentes images utilisées dans les tests mais cette fois ci pour faire une compression basique (PCM et puis DPCM)

* 1. **PCM (Pulse-code modulation) CODING :** 
     1. Commencez par comprendre de quoi il s’agit en faisant une recherche bibliographique.
     2. Sur chaque image en cliquant sur PCM nous obtenons :



**Bitrate**

* Où à droite de cette fenêtre vous avez ‘’Set PCM Bitrate’’ qui s’agit du taux de compression choisi en bpp (Bit Per Pixel). Vous avez le choix de 1 à 7 bpp, avec bpp=1 est le taux de compression le plus élevé et 7 le moins élevé. Si bpp=1 on aura la distorsion la plus importante et bpp=7 nous aurons la distorsion la plus faible.
* Pour chacune des sept valeur de ce Bitrate on l’applique en cliquant sur ‘’Apply’’ ce qui va nous donner image reconstruite et les métriques d’évaluation en particulier le MSE et le PSNR. En relevant ces valeurs de MSE et PSNR pour les sept valeurs de Bitrate nous pourrons réaliser, par Matlab, deux courbes pour chacune des images testées, la première courbe MSE en fonction de Bitarte et la seconde PSNR en fonction de Bitrate.
* Après avoir tracez ces courbes essayez de les discuter
  1. **Simulation d’un canal de transmission**
* Toujours avec le PCM mais en cliquant sur l’onglet Errors

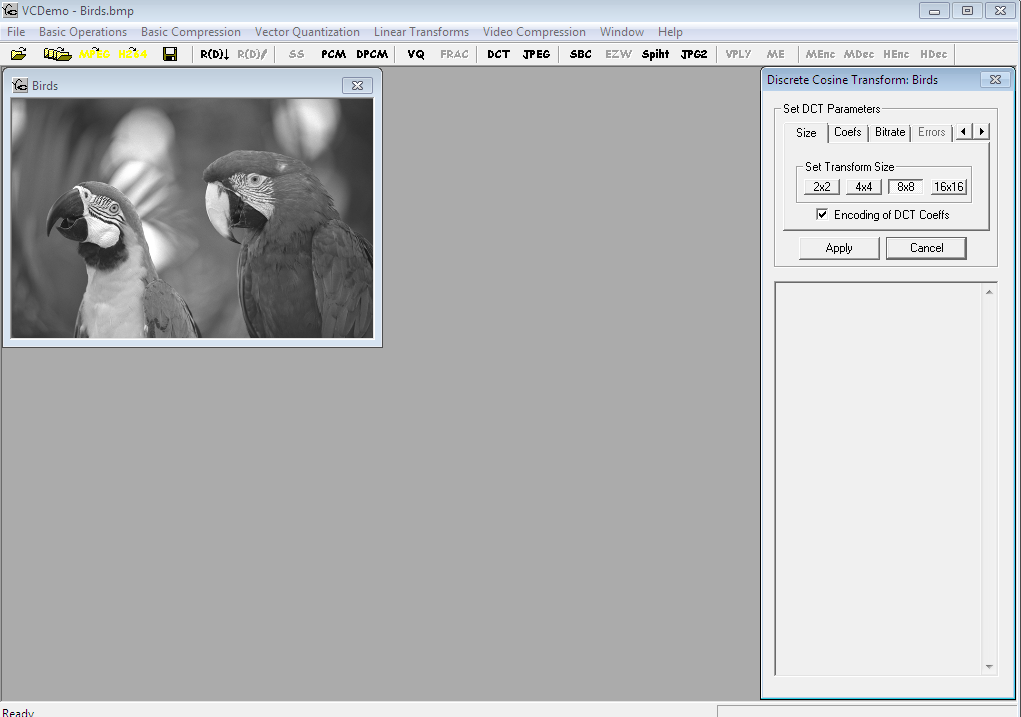


**Errors**

* Il s’agit de simuler un canal qui introduit des erreurs avec une certaine probabilité qui va de 0.005 jusqu’à 0.0001. Pour chacune de ces probabilités et en fixant au début le Bitrate de 1 à 7, nous obtenons de nouvelles valeurs des métriques d’évaluation en particulier MSE et PSNR et bien entendu l’affichage de l’image testée (transmise à travers ce canal simulé et reconstruite).
* Tracez, par Matlab, pour chaque valeur de la probabilité de l’erreur les courbes MSE et PSNR en fonction du Bitrate. Comparer les aux précédentes courbes et discutez les résultats obtenus
* Réalisez le même travail pour les différentes images tests

1. **Transformations linéaires**

* Refaire le même clic droit sur les différentes images utilisées dans les tests mais cette fois ci pour faire appliquer sur les images tests une transformation linéaire. Nous allons nous intéresser uniquement à la DCT ou Discret Cosine Tranform (non pas à la DWT ou Subband coding).



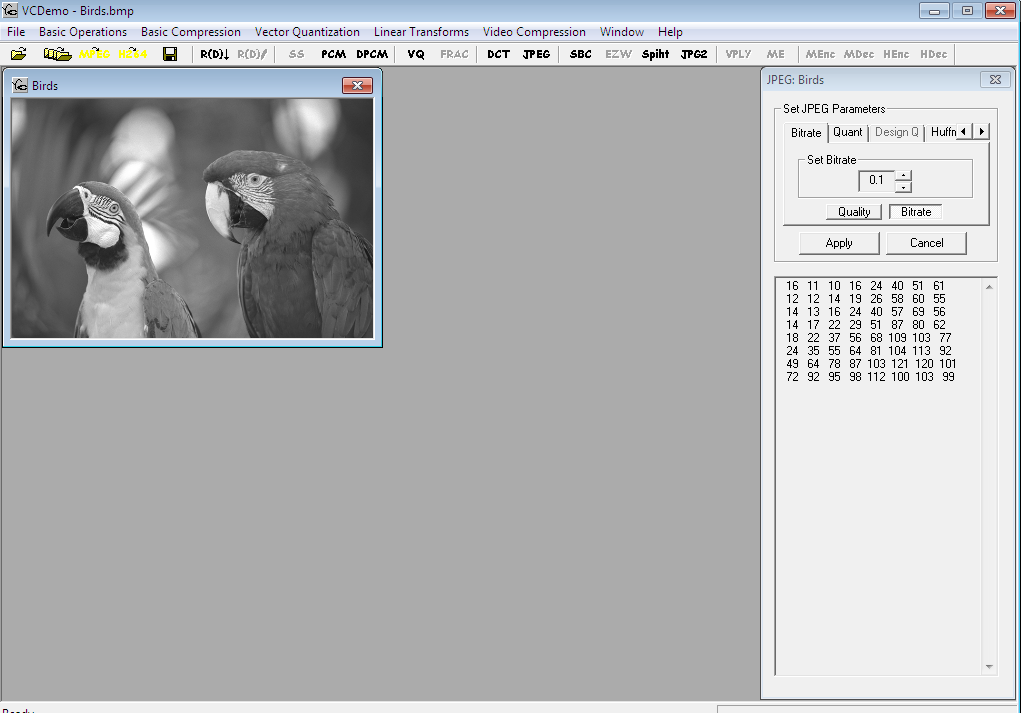
**Bitrate**

**Size**

* Dans l’onglet ‘’**Size**’’ on choisit la taille des blocs d’images 2×2, 4×4, 8×8 ou 16×16. Dans l’onglet ‘**Bitrate**’ on choisit le taux de compression en bpp qui va de 0.25 à 4 soit dix valeurs. **bpp=0.25** est le taux de compression le plus élevé correspondant à la qualité la moins bonne et bpp=4 est le taux de compression le moins élevé pour une qualité la plus élevée. Ainsi pour chaque taille de bloc vous devez appliquer la compression avec les dix valeurs de **bpp** et de noter le **PSNR** et le **MSE**. Ainsi, en Matlab, vous devez tracer deux courbes PSNR et MSE en fonction de bpp pour chaque image test et pour chaque taille de bloc.

1. **JPEG**

* Refaire le même clic droit sur les différentes images utilisées dans les tests mais cette fois ci pour faire appliquer sur les images tests la compression JPEG.



* Nous avons cette fois plusieurs onglets dont ‘’Bitrate’’ mais cliquez sur le bouton ‘’Bitrate’’ au lieu du bouton ‘’Quality’’. Vous pouvez alors choisir une valeur sur valeur bitrate de 0.1 à 8. Pour chaque valeur de bitrate vous appliquez la compression pour obtenir le PSNR et la MSE. Ceci va vous permettre de tracer, par Matlab, des courbes MSE et PSNR en fonction du bitrate pour les différentes images tests.
* Discutez les résultats obtenus et comparez les aux précédents.
* Reprenez la même travail mais en simulant un canal de transmission avec différentes probabilités d’erreurs de 0.001 à 0.00005. Pour chacune de ces probabilités d’erreurs vous appliquez la compression JPEG pour les différents Bitrate utilisés précédemment. Ainsi, vous pouvez tracer les mêmes courbes PSNR et MSE en fonction du Bitrate pour les différentes probabilités d’erreurs.
* Discutez et comparez les résultats obtenus

1. **Conclusion générale**

Vous devez proposer une conclusion générale pour présenter succinctement ce que vous avez pu obtenir et aussi proposer d’éventuels perspectives à ce travail.

**Professeur Noureddine DOGHMANE**