Compression d'image et triangulation

Nicolas Ehrhardt et David Hoffmann

6 juin 2012

1 Introduction

La compression d'image est dominée par des algorithmes basés sur les transformées de Fourier et plus récemment sur la théorie des ondelettes. L'un des désavantages souvent critiqué de la compression par stockage des coefficients de Fourier est son caractère non-localisant. La théorie des ondelettes se propose de résoudre ce problème.

Si la théorie de Fourier est valable pour tout type de signal elle ne prend pas assez en compte le caractère ordonné d'une image. En effet, les objets dans l'image ont une structure particulière : des bords et des surfaces homogènes notamment. Pour mieux rendre compte de cette structure interne, un nouveau type de compression a été introduit, il consiste à stocker un nombre de points fixés de l'image avec leur couleur sous forme de matrice creuse, ainsi qu'une triangulation associée.

1.1 Décompression d'une triangulation

On obtient alors l'image décodée de manière suivante. Pour tout point de l'image, chercher le triangle de la triangulation qui le contient, obtenir alors ses coordonnées barycentriques x_1, x_2, x_3 associées aux trois sommets du triangle. On note dans le même ordre c_1, c_2, c_3 les couleurs des trois sommets. La couleur du point en question est alors $\sum x_i c_i$.

Il est assez simple de voir que l'image reconstruite sera alors composée de triangles dans lesquels la couleur sera un dégradé entre les sommets. Ainsi, on s'attend à avoir une image lissée en quelque sorte et non pas "bruitée" dans le cas d'une compression type Fourier de mauvaise qualité.

1.2 Erreur

Plusieurs types d'erreurs peuvent être utilisés pour comparer les qualités des images. Considérons notre image comme une surface, pour chaque pixel, sa couleur (coordonnée z) est fonction de son abscisse et de son ordonnée dans dans l'image (x, y). On notera $c(x, y) = I(x, y)$ l'image originale, et $c_d(x, y) = I_d(x, y)$ l'image décompressée.

Erreur L^p

Comme indiqué, il suffit de calculer la quantité suivante pour obtenir l'erreur L^p :

$$(\sum_x \sum_y |c(x,y) - c_d(x,y)|^p)^{\frac{1}{p}}$$

Ceci étant généralisable à L^∞.

Erreur L^p en projection

Pour cette erreur, il s'agit de projeter tout point de l'image I sur l'image I_d. Dans notre cas, puisque l'image décompressée est une union de triangles on considère la distance au plan associé au triangle.

$$(\sum_x \sum_y |\pi_d c(x,y) - c_d(x,y)|^p)^{\frac{1}{p}}$$

Cette erreur est contestable nous en ferons la démonstration.

2 Algorithme de compression adaptatif

Si s'intéresser à une triangulation a des avantages sur le plan visuel, il en a aussi sur le plan théorique. On a vu en introduction que l'on stockait les points sous forme de matrice sparse pour gagner en place, mais cela ne nous affranchi pas du stockage de la triangulation, sous forme d'une matrice $n \times 3$. Hormis si l'on travaille avec une triangulation de Delaunay... Évidemment, cette triangulation étant unique, il est inutile de la stocker, elle peut très bien être reconstruite juste avant la décompression en $O(Nln(N))$.

2.1 Principe de l'algorithme

Initialisation : On construit un maillage qui inclut tous les points de l'image (N). Chaque point ayant comme erreur associée la couleur maximale de ses quatres voisins moins la couleur minimale de ses quatre voisins.

Étape i : On dispose de $N - i$ points dans notre maillage. On choisit le point auquel est associé la plus petite erreur et on l'enlève du maillage. On recalcule la triangulation de delaunay pour ses points voisins. Et on met à jour leur erreur associée.

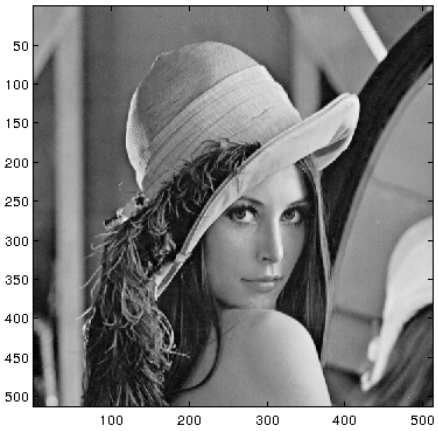
Critère d'arrêt : On peut se donner un nombre de point à enlever, ou demander à s'arrêter lorsque l'erreur atteint un seuil.

Calcul de l'erreur associée : L'erreur associée à un point est l'erreur entre l'image décompressée à partir du maillage auquel on aurait enlevé ce point et l'image initiale. Celle-ci dépend du choix de la méthode avec laquelle nous calculerons l'erreur.

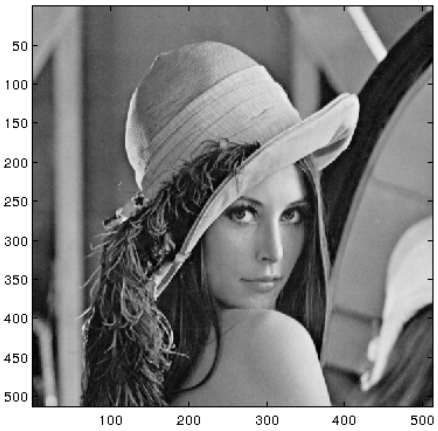
2.2 Résultats

Prenons pour commencer une image avec un certain nombre de bords et de la texture. Comme d'habitude, notre amie Lena fera parfaitement l'affaire. Il faudra regarder de plus près les cheveux et le haut du chapeau pour voir apparaître les premières erreurs.

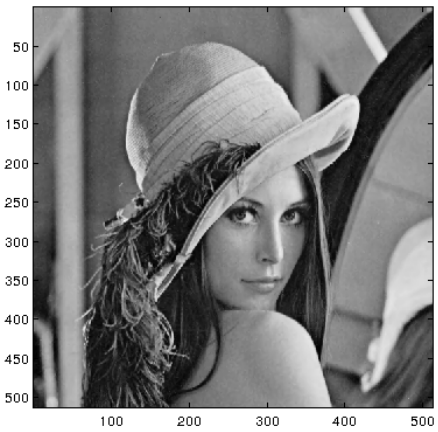
3 Algorithme des bissections



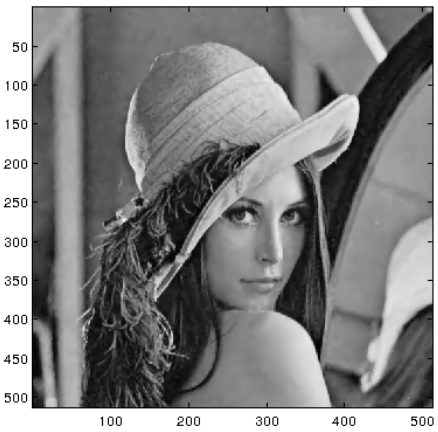
(a) Image originale



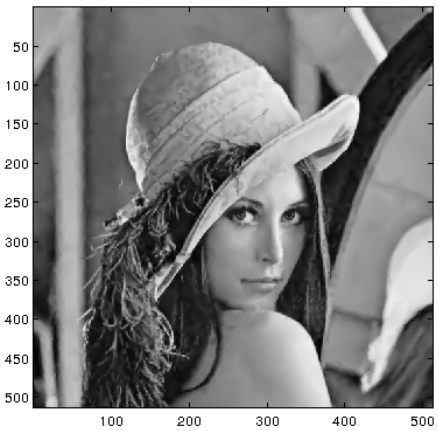
(b) 102144 Noeuds ; snr : 3.4489



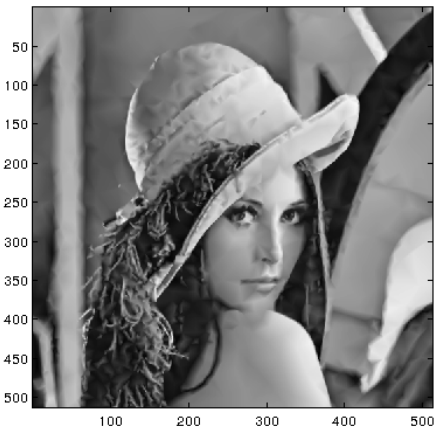
(c) 42144 Noeuds ; snr : 9.9287



(d) 12144 Noeuds ; snr : 25.9734



(e) 7144 Noeuds ; snr : 39.5056



(f) 4144 Noeuds ; snr : 63.0030

FIGURE 1 – Résultat pour l'algorithme de compression adaptatif