

Interpolation simple et efficace d'un bloc dans une image à partir de son contour

Ziad Al kachouh et Maurice G. Bellanger

CNAM - ÉLECTRONIQUE
292, rue Saint-Martin
75141 - Paris cedex 03 - France

RÉSUMÉ

Dans certaines techniques de traitement et de compression, les images sont décomposées en blocs de pixels qui sont ensuite transformés par DCT. Par la suite, des blocs peuvent être abandonnés ou perdus et il faut les reconstituer, à partir de leur voisinage dans l'image. On montre qu'une interpolation exacte des blocs d'image peut être effectuée dans le domaine spatial, en utilisant la DCT. Les conditions d'existence d'une solution sont données et l'algorithme correspondant est optimisé, il nécessite 8 multiplications par pixel restitué. Des simulations montrent que l'interpolation exacte proposée présente de meilleures performances que d'autres méthodes, avec un moindre coût de calcul.

ABSTRACT

In many processing and coding techniques, images are decomposed into blocks of pixels and the DCT is used to transform those blocks. In some situations, blocks are dropped or lost, and, at some point in the processing, they must be restored from their neighborhood in the images. It is shown that an exact interpolation of image blocks can be performed in the spatial domain, using the DCT. The conditions for a solution to exist are derived and the corresponding algorithm is optimized, requiring 8 multiplies per restored pixel. The exact interpolation approach outperforms the alternative methods in most of the simulations presented, at a significantly smaller computational cost.

1 Introduction

La transformation de blocs par DCT (Discrete Cosine Transform) est d'utilisation courante en traitement et codage d'images. Dans certaines opérations, comme le stockage ou la transmission, des blocs peuvent être éliminés ou perdus et il faut les restituer lors de la reconstitution de l'image.

Le principe d'interpolation est à la base des techniques de restitution et différentes approches ont été proposées. Par exemple, dans le domaine fréquentiel, les coefficients DCT du bloc à restituer peuvent être interpolés à partir des coefficients DCT des blocs voisins disponibles [1]. Des approches dans le domaine spatial ont aussi été proposées, utilisant les moindres carrés, avec de fortes complexités de calculs et des effets de lissage [2, 3].

L'objet de la présente contribution est d'étudier la faisabilité de l'interpolation exacte et directe d'un bloc à partir de son contour. Cette approche s'appuie sur l'hypothèse de base utilisée en compression d'images, à savoir que les composantes de fréquences élevées peuvent être négligées. C'est à dire que pour faire une interpolation dite exacte, on annule un nombre de coefficients DCT de fréquences élevées suffisant pour obtenir un système d'équations déterminé. En fait, il faut considérer comme nuls autant de coefficients DCT qu'il y a de pixels à restituer.

Le principe de l'interpolation est explicité dans la section 2 et les conditions pour que le système ait une solution unique sont étudiées dans la section 3. L'algorithme de calcul est analysé et des résultats de simulations sont donnés dans les sections 4 et 5, respectivement. Enfin, une conclusion et des perspectives d'applications sont évoquées dans la section 6.

2 Principe de l'interpolation

La transformation DCT, appliquée à un bloc de $N \times N$ pixels est définie par l'équation :

$$Y(k,l) = \frac{2C(k)C(l)}{N} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \cos\left[\frac{\pi(2i+1)k}{2N}\right] \cos\left[\frac{\pi(2j+1)l}{2N}\right] \cdot X(i,j) \quad (1)$$

avec $i, j, k, l \in \{0, 1, \dots, N-1\}$.

L'interpolation d'un bloc à partir de son contour correspond au calcul de $L \times L$ pixels, avec $L = N - 2$, en utilisant les $4(N+1)$ pixels du contour comme le montre la figure 1-a. Dans la transformation d'une image, les coefficients DCT $Y(k,l)$ sont généralement négligeables pour les valeurs des paramètres k et l associées aux fréquences hautes. L'annulation de ces coefficients amène la compression. Dans le cas présent, pour calculer $L \times L$ pixels inconnus, il faut annuler autant de valeurs $Y(k,l)$. Pour écrire le système d'équations, on désigne par X_1 et X_2 les vecteurs des pixels connus et inconnus respectivement et par A et B les matrices des éléments de la DCT qui les multiplient pour obtenir l'ensemble des $Y(k,l)$ nuls. D'où l'équation suivante :

$$A \cdot X_1 + B \cdot X_2 = 0 \quad (2)$$

Les matrices A et B sont respectivement de dimensions $L^2 \times 4(N+1)$ et $L^2 \times L^2$. Pour qu'une solutions X_2 existe, il faut et il suffit que $\det[B] \neq 0$, c'est à dire que la matrice B associée au système (2) soit de

rang plein. Les éléments de la matrice B sont fonctions des valeurs $Y(k, l)$ annulées en sortie de la DCT. On désigne par domaine d'annulation l'ensemble de ces valeurs. En pratique, il est intéressant de disposer d'une certaine latitude pour choisir ces valeurs, par exemple en utilisant des connaissances a priori sur les statistiques d'une image. Il apparaît cependant que les conditions d'inversion de la matrice B sont très restrictives.

3 Détermination du domaine d'annulation

La démarche utilise une décomposition trigonométrique, un partitionnement de la matrice et une factorisation liée à la DCT. Les éléments de la DCT peuvent être réécrits par :

$$M_{k,l}(i, j) = E(k, i) \cdot E(l, j) \quad (3)$$

avec $i, j, k, l \in \{0, 1, \dots, N-1\}$, et

$$E(k, i) = \cos\left[\frac{\pi(2i+1)k}{2N}\right] \quad (4)$$

Le facteur $\frac{2C(k)C(l)}{N}$ n'est pas pris en compte dans ce calcul dans un but de simplification ; de plus, ce facteur est éliminé par la mise à zéro des coefficients DCT.

$E(k, i)$ est une fonction paire, elle est symétrique par rapport à l'axe $\frac{N-1}{2}$. En utilisant la décomposition trigonométrique suivante :

$$\cos(a-b) = \cos a \cdot \cos b + \sin a \cdot \sin b \quad (5)$$

et pour $h = 0, 1, \dots, \frac{N}{2} - 1$, nous pouvons écrire :

$$E(k, N-1-h) = \cos(k\pi) \cdot E(k, h) \quad (6)$$

Une réécriture de la relation (1) de transformation DCT est réalisée par le réarrangement des coefficients DCT en sous-blocs et dans un ordre comme indiqué sur la figure 1-b et rotation des sous-blocs 2, 3 et 4. D'où :

$$Y(k, l) = [F \cos(l\pi) \cdot F \cos(k\pi) \cdot F \cos(k\pi) \cdot \cos(l\pi) \cdot F] \quad (7)$$

Où,

$$F = F(k, l) = (E(k, i) \cdot E(l, j))_{0 \leq i, j \leq \frac{N}{2}-1} \quad (8)$$

Les coefficients dans la matrice F sont parcourus de gauche à droite et de haut en bas dans le sous bloc 1 de la figure 1-b pour avoir une représentation unicolonne des coefficients DCT et des pixels. Un réarrangement en fonction de k et l paire et impaire des coefficients $Y(k, l)$, donne :

$$\begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \\ Y_4 \end{bmatrix} = \begin{bmatrix} P_1 & P_1 & P_1 & P_1 \\ P_2 & -P_2 & -P_2 & P_2 \\ P_3 & -P_3 & P_3 & -P_3 \\ P_4 & P_4 & -P_4 & -P_4 \end{bmatrix} \cdot \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \end{bmatrix} + \begin{bmatrix} Q_1 & Q_1 & Q_1 & Q_1 \\ Q_2 & -Q_2 & -Q_2 & Q_2 \\ Q_3 & -Q_3 & Q_3 & -Q_3 \\ Q_4 & Q_4 & -Q_4 & -Q_4 \end{bmatrix} \cdot \begin{bmatrix} W_1 \\ W_2 \\ W_3 \\ W_4 \end{bmatrix} \quad (9)$$

Soient ϕ_1 et ϕ_2 deux ensembles de (i, j) correspondants respectivement aux zones grise et blanche du sous bloc 1. P_r et Q_r sont des matrices dont les coefficients de transformations DCT correspondent aux vecteurs des pixels connus V_r et des pixels inconnus W_r dans chaque sous-bloc et en fonction de k et l paire et impaire ($1 \leq r \leq 4$), comme indiqué sur la figure 1-b.

$$Y_1 = (Y_{2m, 2n})_{0 \leq m, n \leq \frac{N}{2}-1}$$

$$Y_2 = (Y_{2m+1, 2n+1})_{0 \leq m, n \leq \frac{N}{2}-1}$$

$$Y_3 = (Y_{2m, 2n+1})_{0 \leq m, n \leq \frac{N}{2}-1}$$

$$Y_4 = (Y_{2m+1, 2n})_{0 \leq m, n \leq \frac{N}{2}-1}$$

$$P_1 = (M_{2m, 2n}(i, j))_{0 \leq m, n \leq \frac{N}{2}-1; (i, j) \in \phi_1}$$

$$P_2 = (M_{2m+1, 2n+1}(i, j))_{0 \leq m, n \leq \frac{N}{2}-1; (i, j) \in \phi_1}$$

$$P_3 = (M_{2m, 2n+1}(i, j))_{0 \leq m, n \leq \frac{N}{2}-1; (i, j) \in \phi_1}$$

$$P_4 = (M_{2m+1, 2n}(i, j))_{0 \leq m, n \leq \frac{N}{2}-1; (i, j) \in \phi_1}$$

$$Q_1 = (M_{2m, 2n}(i, j))_{0 \leq m, n \leq \frac{N}{2}-1; (i, j) \in \phi_2}$$

$$Q_2 = (M_{2m+1, 2n+1}(i, j))_{0 \leq m, n \leq \frac{N}{2}-1; (i, j) \in \phi_2}$$

$$Q_3 = (M_{2m, 2n+1}(i, j))_{0 \leq m, n \leq \frac{N}{2}-1; (i, j) \in \phi_2}$$

$$Q_4 = \left(M_{2m+1,2n}(i,j) \right)_{0 \leq m, n \leq \frac{N}{2}-1; (i,j) \in \phi_2}$$

Les matrices Q_r , pour $1 \leq r \leq 4$, peuvent être simplifiées en utilisant le produit matriciel de Kronecker symbolisé par \otimes , soit :

$$E_1 = \left(E(2m, i) \right)_{0 \leq m \leq \frac{N}{2}-1; 0 < i \leq \frac{N}{2}-1} \quad (10)$$

$$E_2 = \left(E(2m+1, i) \right)_{0 \leq m \leq \frac{N}{2}-1; 0 < i \leq \frac{N}{2}-1} \quad (11)$$

D'où :

$$Q_1 = E_1 \otimes E_1, \quad Q_2 = E_2 \otimes E_2,$$

$$Q_3 = E_1 \otimes E_2, \quad Q_4 = E_2 \otimes E_1$$

Les matrices E_1 et E_2 sont de dimensions $\frac{N}{2} \times \frac{L}{2}$ et contiennent donc plus de lignes que des colonnes. Des sous matrices carrées de rang plein doivent être extraites. La sélection des lignes dans chacune des sous matrices est fonction des paramètres k et l associées aux fréquences. Dans le cas présent, l'interpolation se fait sans connaissances a priori sur les statistiques de l'image ou du voisinage du bloc à interpoler. Le choix se porte donc, sur les valeurs k et l les plus élevées. Le nombre de lignes doit être suffisamment élevé pour former des matrices inversibles à partir des matrices E_1 et E_2 . Soient :

$$H_1 = \left(E(2m, i) \right)_{0 < m \leq \frac{N}{2}-1; 0 < i \leq \frac{N}{2}-1} \quad (12)$$

$$H_2 = \left(E(2m+1, i) \right)_{0 < m \leq \frac{N}{2}-1; 0 < i \leq \frac{N}{2}-1} \quad (13)$$

Ceci donne une configuration du domaine d'annulation comme il est indiqué sur la figure 1-c.

En utilisant l'hypothèse que les coefficients DCT de hautes fréquences sont faibles, correspondant aux domaine d'annulation, permet leur substitution par des zéros et le système d'équations déterminé est le suivant :

$$[0] = \begin{bmatrix} R_1 & R_1 & R_1 & R_1 \\ R_2 & -R_2 & -R_2 & R_2 \\ R_3 & -R_3 & R_3 & -R_3 \\ R_4 & R_4 & -R_4 & -R_4 \end{bmatrix} \cdot \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \end{bmatrix} + \begin{bmatrix} S_1 & S_1 & S_1 & S_1 \\ S_2 & -S_2 & -S_2 & S_2 \\ S_3 & -S_3 & S_3 & -S_3 \\ S_4 & S_4 & -S_4 & -S_4 \end{bmatrix} \cdot \begin{bmatrix} W_1 \\ W_2 \\ W_3 \\ W_4 \end{bmatrix} \quad (14)$$

Avec :

$$S_1 = H_1 \otimes H_1, \quad S_2 = H_2 \otimes H_2$$

$$S_3 = H_1 \otimes H_2, \quad S_4 = H_2 \otimes H_1$$

$$R_1 = \left(M_{2m,2n}(i,j) \right)_{0 < m, n \leq \frac{N}{2}-1; (i,j) \in \phi_1}$$

$$R_2 = \left(M_{2m+1,2n+1}(i,j) \right)_{0 < m, n \leq \frac{N}{2}-1; (i,j) \in \phi_1}$$

$$R_3 = \left(M_{2m,2n+1}(i,j) \right)_{0 < m, n \leq \frac{N}{2}-1; (i,j) \in \phi_1}$$

$$R_4 = \left(M_{2m+1,2n}(i,j) \right)_{0 < m, n \leq \frac{N}{2}-1; (i,j) \in \phi_1}$$

La résolution du système d'équations déterminé peut se faire directement par inversion de la matrice associée au système. Mais, cette tâche est facilitée par le partitionnement et la factorisation matricielle réalisée dans l'étape précédente. Les avantages des propriétés de produit matriciel de Kronecker est utilisé [4]. Il suffit d'inverser les matrices H_1 et H_2 qui sont d'ordre $\frac{L}{2}$. Après quelques manipulations il vient :

$$\begin{bmatrix} W_1 \\ W_2 \\ W_3 \\ W_4 \end{bmatrix} = \begin{bmatrix} A & B & C & D \\ B & A & D & C \\ C & D & A & B \\ D & C & B & A \end{bmatrix} \cdot \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \end{bmatrix} \quad (15)$$

avec,

$$A = \frac{1}{4}(a+b+c+d), \quad B = \frac{1}{4}(a-b-c+d)$$

$$C = \frac{1}{4}(a-b+c-d), \quad D = \frac{1}{4}(a+b-c-d)$$

$$a = -H_1^{-1} \otimes H_1^{-1} \cdot R_1, \quad b = -H_2^{-1} \otimes H_2^{-1} \cdot R_2$$

$$c = -H_1^{-1} \otimes H_2^{-1} \cdot R_3, \quad d = -H_2^{-1} \otimes H_1^{-1} \cdot R_4$$

4 Masque d'interpolation et complexité

L'avantage de cette approche est que le masque d'interpolation peut être pré-calculé avant de commencer le processus. Soit G ce masque :

$$G = \begin{bmatrix} A & B & C & D \\ B & A & D & C \\ C & D & A & B \\ D & C & B & A \end{bmatrix} \quad (16)$$

Cette matrice (masque) est constituée de L^2 lignes de $4 \cdot (N + 1)$ éléments. Chaque ligne compte 8 éléments non nuls, ce qui réduit à 8 le nombre de multiplications pour restituer un pixel.

Quelques aspects supplémentaires doivent être considérés. D’abord, les coefficients du masque sont normalisés à 1, ils conservent donc l’intensité globale de l’image. De plus, les pixels restitués doivent rester dans l’intervalle de définition $[0,255]$. Par exemple, dans certaines images on observe un taux de débordement de 0.1%. Si un débordement se produit, il faut effectuer une saturation.

5 Résultats

En pratique, le codage d’image se fait par blocs de taille 8×8 . Pour interpoler un bloc erroné de cette taille par son contour il faut que $N=10$ ($L=8$). Les résultats de simulation de l’algorithme proposé ainsi que l’algorithme donné dans [3] sont montrés sur la figure 2. L’algorithme proposé donne de meilleurs résultats subjectifs, surtout les contours sont mieux restitués. Concernant la mesure objective PSNR (rapport signal sur bruit crête), les valeurs obtenues confirment la supériorité de cette approche.

Les blocs contenant des contours verticaux, horizontaux ou légèrement déviés sont correctement reconstitués. Cependant, des difficultés pour restituer un bloc de fortes activités spatiales (une forte amplitude d’un coefficient DCT de hautes fréquences mis à zéro), et des contours diagonaux sont observées. Une zone de la barre oblique dans l’image est mal restituée.

6 Conclusion et perspectives

Cette méthode d’interpolation est performante, car l’interpolation est exacte au sens de résolution d’un système d’équations déterminé et en exploitant les pixels de contour.

Cette technique s’étend au cas d’un contour non fermé, par exemple limité à 3 ou même 2 cotés pour un bloc. L’interpolation basée sur des contours non carrés est également possible.

L’approche proposée permet une amélioration de la qualité objective et subjective de l’image avec un coût de calculs plus réduit par rapport aux algorithmes donnés dans [2, 3]. Elle s’applique à la compression d’images, par la non transmission de blocs «masquables». Elle s’applique également au masquage des erreurs de transmission dans le cadre des normes de transmission des images animées dans un milieu bruité (MPEGx, H26x).

Références

[1] L. T. Chia et al., "On the Use of Transform Domain Information for Concealment of Errors in JPEG Images", Proceedings of the Conference EUSIPCO-94, Edinburg (Scotland), pp. 616-619, September 1994.
 [2] Yao Wang, Qin-Fan Zhu, and Leonard Shaw, "Maximally Smooth Image Recovery in Transform Coding", IEEE Transactions on Communications, Vol. 41, No. 10, pp. 1544-1551, October 1993.
 [3] J. W. Park, D. S. Kim and S. U. Lee, "On the Error Concealment Technique for DCT Based Image Coding", Proceedings of the Conference ICASSP-94, pp. III-293-296, 1994.
 [4] F. Rotella, P. Borne, "Méthode et pratique de l’ingénieur, Mathématiques: Théorie et pratique du calcul matriciel", EDITIONS TECHNIP, Paris, 1995.

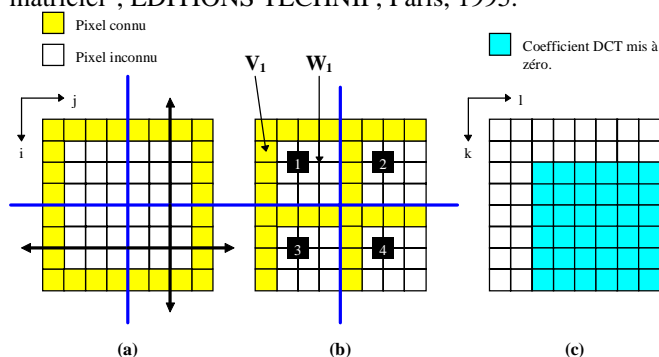


Figure 1 : (a) Pixels du bloc à interpoler et son contour. (b) pixels dans 4 sous blocs avec des rotations. (c) coefficients DCT dans le domaine transformé.

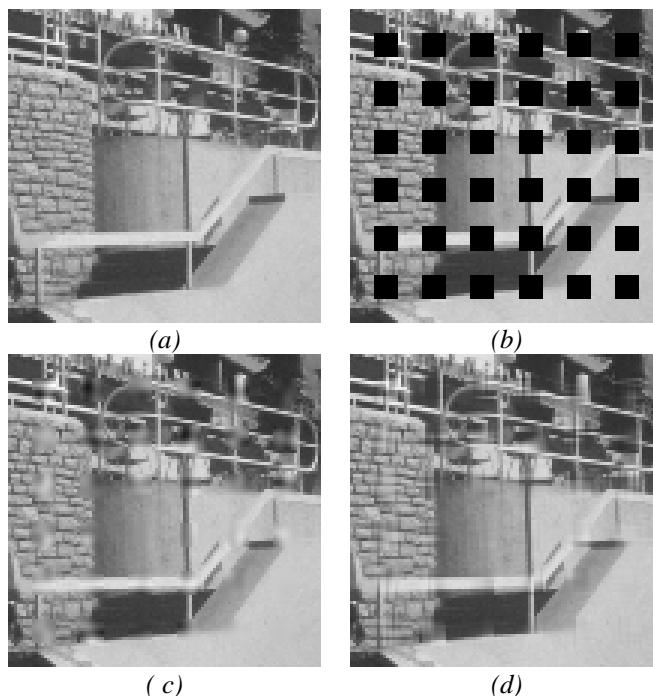


figure-2 : (a) image originale, (b) image avec blocs erronés, (c) image avec interpolation par l’algorithme proposé dans [3], PSNR=24,9 dB, (d) image avec interpolation par l’algorithme proposé, PSNR=25,4 dB.