

DIXIEME COLLOQUE SUR LE TRAITEMENT DU SIGNAL ET SES APPLICATIONS

NICE du 20 au 24 MAI 1985

ETUDE COMPARATIVE DE PLUSIEURS ALGORITHMES ADAPTATIFS DE COMPRESSION D'IMAGE PAR TRANSFORMATION UNITAIRE

Serge MULLER

Université P. & M. CURIE (PARIS 6), Place Jussieu, 75005 PARIS (stage à la S.A. MATRA - 78146 VELIZY Cedex)

RESUME

Pour les applications de la télédétection, les qualités demandées à un algorithme de compression d'images sont :

- taux de compression important
- faible distorsion des images reconstruites
- immunité au bruit de transmission
- rapidité d'exécution

Parmi les méthodes de compression par transformation, la méthode de quantification par blocs associée à un codage par zones, semble offrir le meilleur compromis vis à vis des exigences formulées. Les performances de cette technique peuvent être substantiellement améliorées en l'adaptant aux variations de la statistique de l'image.

Trois types d'adaptativité sont étudiés :

- l'adaptativité de la transformation
- l'adaptativité de l'allocation des bits
- l'adaptativité de la distribution des niveaux de quantification

Une évaluation et une comparaison des différentes étapes de la chaîne de compression sont présentées. Le nombre optimum de classes d'adaptativité et la taille optimum des blocs à transformer sont déterminés. La réduction de l'écart entre les performances de la transformation en cosinus et de la transformation de Hadamard avec l'utilisation de méthodes adaptatives est montrée. Les simulations permettent d'établir que la répartition des points transformés est gaussienne pour une compression adaptative et laplacienne pour une compression non adaptative. Le codage vectoriel est introduit. Il permet d'obtenir une diminution de la distorsion par rapport au codage scalaire usuel, sans augmentation du temps calcul. Diverses propositions permettent d'augmenter la rapidité d'exécution qui est un des principaux problèmes à l'implantation de ce type d'algorithme sur un système temps réel.

SUMMARY

In remote sensing applications, the qualities needed for image compression algorithms are :

- important compression rate
- reconstructed images with small distortion
- good immunity to transmission noise
- fast computation

Within the transform coding methods, bloc quantization used with zonal coding seems to be a good choice for the specified goals. Performance of transform coding schemes could be improved substantially by adapting them to changes in image statistics.

Three types of adaptation are studied :

- adaptation of transform
- adaptation of bit allocation
- adaptation of quantization levels

The performance obtained by the different steps of the compression chain are evaluated and compared. Optimal number of adaptivity classes and optimal block size are determined. It is shown that adaptivity reduce the distortion gap between cosine transform and Hadamard transform. The distributions of the transform coefficients are shown to be gaussian with adaptive compression and laplacian with non adaptive compression. Vector coding is introduced. Its use allows us to obtain a distortion reduction with regard to the usual scalar coding without increasing the processing time. Various suggestions allows us to increase execution rapidity that is one of the more important problems for real time implementation.



I - INTRODUCTION

Pour les applications de télédétection, les qualités demandées à un algorithme de compression d'images sont :

- taux de compression important
- faible distorsion des images reconstruites
- immunité au bruit de transmission
- rapidité d'exécution

El ressort de l'excellente revue de Jain (1) que les méthodes de compression par transformation unitaire satisfont mieux aux trois premiers critères énoncés ci-dessus simultanément, que les méthodes prédictives (DPCM par exemple). Les autres méthodes de compression rencontrées dans la littérature (2) apparaissent trop complexes pour être implantées dans un système temps réel, ou bien conduisent à des distorsions inacceptables pour les applications visées ici.

Parmi les méthodes de compression par transformation, la méthode de quantification par blocs associée à un codage par zones, semble offrir le meilleur compromis vis à vis des exigences formulées. Les performances de cette technique peuvent être substantiellement améliorées en l'adaptant aux variations de la statistique de l'image.

La figure 1 décrit le schéma général de la chaîne de compression. On distingue essentiellement trois types d'adaptativité :

- adaptativité de la transformation concernant :
 - . la transformation unitaire des blocs de l'image
- adaptativité de l'allocation des bits regroupant :
 - . le calcul des variances des points de l'espace transformés
 - . la construction des cartes de bits
 - . le codage
- adaptativité de la distribution des niveaux de quantification comprenant :
 - . la normalisation des points transformés
 - . la quantification

Le calcul des activités et leur classification sont deux étapes communes aux trois types d'adaptativité.

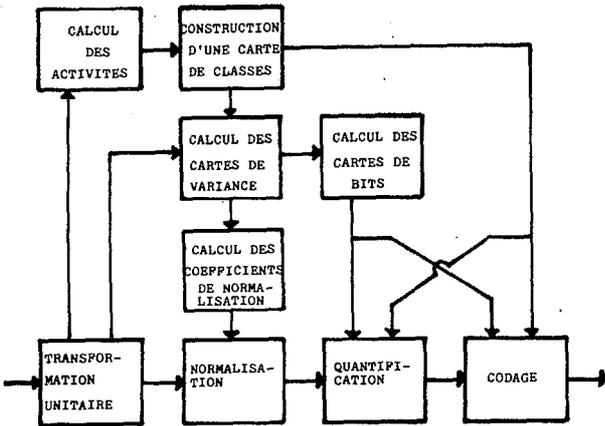


Figure 1 : Chaîne de compression

La présente étude compare à chaque étape de la chaîne de compression, diverses méthodes. Le chapitre II traite de la taille des blocs et de la détermination du nombre optimum de classes d'adaptativité. Le traitement des grandes images est évoqué. Le chapitre III est consacré à l'adaptativité de la transformation. Une comparaison des performances obtenues en utilisant la transformation de Hadamard, la transformation en cosinus, ou les deux simultanément est exposée. Le chapitre IV regroupe les étapes du traitement relatives à l'adaptativité de l'allocation des bits. Le codage vectoriel des points transformés d'un bloc est introduit. Il est comparé au codage scalaire classique. Plusieurs estimateurs de variance sont proposés et l'influence de leur choix sur les performances de

l'algorithme analysé. Le chapitre V expose l'adaptativité de la distribution des niveaux de quantification. Plusieurs modèles de répartition utilisés pour la quantification des points transformés sont confrontés. Dans le chapitre VI quelques propositions sont faites pour améliorer la rapidité d'exécution de l'algorithme. Une synthèse des résultats obtenus est présentée dans la conclusion. L'accent est mis tout au long de l'exposé sur le problème du temps calcul qui est un des principaux obstacles à l'implantation de ce type d'algorithme sur un système temps réel.

II - NOMBRE OPTIMUM DE CLASSES D'ADAPTATIVITE ET TAILLE DES BLOCS

Adapter la compression aux variations de la statistique de l'image ne peut que diminuer la distorsion. Pour cela, il faut d'abord regrouper les blocs en classes de statistique homogène. Le problème qui se pose immédiatement est de savoir combien de classes considérer. Les auteurs qui effectuent une classification des blocs image à partir d'une mesure d'activité prennent un nombre de classes arbitraire (3). Le choix de la taille des blocs pour une taille d'image donnée n'est pas non plus effectué sur des critères objectifs.

Ces problèmes sont étudiés ci-après, et des solutions sont proposées. Le traitement des grandes images est évoqué dans le dernier paragraphe.

1 - FRAIS GENERAUX

Outre les points transformés codés, la restitution de l'image nécessite :

- la carte d'appartenance de chaque bloc à une des classes
 - la carte des écarts-types de chaque classe
- Ce sont les frais généraux dont il faut s'acquitter pour effectuer une compression adaptative.

Pour C classes d'adaptativité, on doit repérer chaque bloc par un mot de longueur $\log_2 C$ bits. La carte d'appartenance de chaque bloc de $M \times M$ pixels à l'une des C classes nécessite donc, pour une image de $N \times N$ pixels, l'emploi de B_c bits/pixel. Les cartes des écarts-types associées aux C classes sont codées quand à elles sur B_v bits/pixel, avec :

$$B_v = \frac{3 C N^2}{N^2} \quad B_c = \frac{\log_2 C}{M^2} \quad B = B_v + B_c$$

chaque écart-type disposant de 8 bits.

2 - TAILLE OPTIMUM DES BLOCS

Les frais généraux peuvent être minimisés par un choix judicieux de la taille des blocs. La minimisation de B, donne :

$$M = \sqrt[4]{\frac{N^2 \log_2 C}{8C}}$$

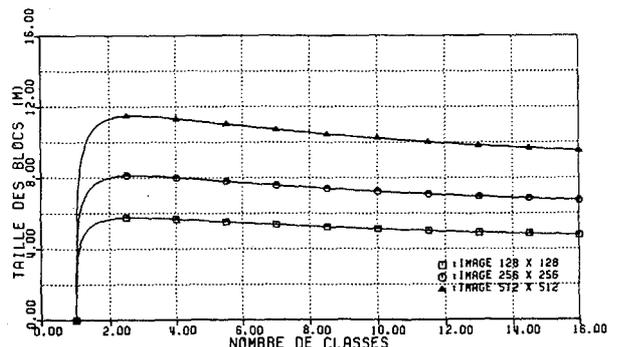


Figure 2 : Taille optimum des blocs



ETUDE COMPARATIVE DE PLUSIEURS ALGORITHMES ADAPTATIFS
DE COMPRESSION D'IMAGE PAR TRANSFORMATION UNITAIRE

La figure 2 donne le paramètre M en fonction du nombre de classes C pour trois tailles d'image. On voit par exemple que la minimisation des frais généraux pour une image 256x256 pixels avec 4 classes d'adaptativité, nécessite un découpage en blocs de 8x8 pixels.

3 - DETERMINATION DU NOMBRE OPTIMUM DE CLASSES

Il a été tracé pour plusieurs images des courbes représentant la distortion en fonction du taux de compression pour plusieurs valeurs du nombre de classes. Ces résultats ont été obtenus en utilisant une transformation de Hadamard suivie d'une quantification de chaque coefficient transformé à l'aide d'un quantificateur de Max (4) dans l'hypothèse d'une répartition gaussienne. La mesure d'activité adoptée est l'énergie du bloc transformé moins le continu.

On utilise le rapport signal sur bruit exprimé en décibels pour mesurer la distortion :

$$S/B = 10 \log \frac{\sigma^2}{\epsilon^2}$$

σ^2 : variance centrée de l'image

ϵ : erreur quadratique moyenne de reconstruction

Pour C = 1 on retrouve le cas bien connu de la compression non adaptative. On observe alors une diminution de la distortion lorsque la taille des blocs augmente.

Cependant, pour la compression adaptative (C différent de 1) on trouve expérimentalement que les résultats sont indépendants de la valeur de M.

Ainsi pour une image de 256x256 pixels, un découpage en blocs de 8x8 pixels semble être le meilleur choix puisqu'il minimise les frais généraux et que la distortion est très peu sensible à ce choix pour un codage adaptatif.

La figure 3 montre, pour une image 256x256 découpée en blocs 8x8, que l'augmentation du nombre de classes d'adaptativité entraîne, à qualité constante :

- une diminution du nombre de bits à transmettre correspondant à l'image codée.
- une augmentation du nombre de bits pour les frais généraux

Lorsque, pour une valeur de la distortion fixée, l'augmentation des frais généraux compense la diminution du nombre de bits nécessaires au codage de l'image, la valeur optimum du nombre de classes est atteinte.

On ne considère que les valeurs de C puissances de 2 pour utiliser au mieux les bits dont on dispose pour la carte de classes. On trouve alors les valeurs de C optimum suivantes :

- C = 4 pour des images 128x128 ou 256x256
- C = 8 pour des images 512x512

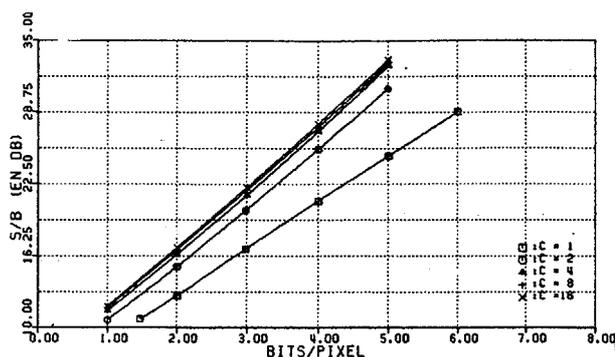


Figure 3 : Nombre optimum de classes

4 - TRAITEMENT DES GRANDES IMAGES

Il est important vis à vis des algorithmes de transformation unitaire rapides que le nombre de points à transformer soit une puissance de 2. Il ressort de la figure 2 que pour les valeurs de C optimum, seule une image 256x256 permet d'obtenir des blocs dont le nombre de points soit puissance de 2 et dont la taille minimise les frais généraux. Pour des images 128x128 ou 512x512, on ne peut minimiser qu'imparfaitement la charge des frais généraux puisqu'il faut approcher la taille des blocs à la puissance de 2 la plus proche.

On remarquera d'autre part qu'il y a un compromis à faire sur le choix de la taille des blocs. L'utilisation de blocs de grande taille permet :

- une décorrélation des données plus importante (essentiellement jusqu'à 16x16)

L'utilisation des blocs de petite taille permet :

- un calcul plus rapide de l'image transformée
- un traitement de l'information plus local

On voit que des blocs de 8x8 pixels semblent offrir un bon compromis.

Ainsi, pour une image de grande taille, une solution pour obtenir simultanément :

- une minimisation des frais généraux
- un nombre de classes optimum
- un bon compromis pour la taille des blocs

est de découper l'image en sous-images de 256x256 pixels, puis de découper chaque sous-image en blocs de 8x8 pixels. La compression adaptative utilise alors 4 classes.

Dans la suite de l'étude, les résultats proviennent de sous-images possédant ces caractéristiques.

III - ADAPTATIVITE DE LA TRANSFORMATION

L'hypothèse de stationnarité de l'image, invoquée lors de l'utilisation de transformations unitaires n'est que très rarement vérifiée. Plutôt que d'effectuer une transformation de Karhunen - Loève (connue pour être optimum vis à vis de la décorrélation des données) sur chaque zone de statistique homogène, on peut essayer de définir une stratégie de choix entre plusieurs transformations possédant un algorithme de calcul rapide. Deux d'entre elles sont particulièrement intéressantes :

- la transformation en cosinus, parce qu'elle donne des résultats comparables à K.L.
- la transformation de Hadamard parce qu'elle est particulièrement rapide et simple à implanter.

Dans le premier paragraphe, une comparaison des performances obtenues par chacune de ces transformations est donnée pour 1 et 4 classes d'adaptativité. Le second paragraphe évoque l'utilisation simultanée de ces deux transformations pour la compression adaptative d'une image.

1 - TRANSFORMATION DE HADAMARD - TRANSFORMATION EN COSINUS

Le problème est de savoir si l'on est disposé à perdre en rapidité d'exécution pour bénéficier du gain en qualité de reconstruction obtenu en passant de Hadamard à Cosinus.

Des simulations ont été menées pour le cas non adaptatif (C = 1) et pour le cas adaptatif (C = 4) sur diverses images de télédétection de 256x256 pixels. La figure 4 donne les résultats pour l'une d'entre elles.



ETUDE COMPARATIVE DE PLUSIEURS ALGORITHMES ADAPTATIFS
DE COMPRESSION D'IMAGE PAR TRANSFORMATION UNITAIRE

Globalement, on retiendra que :

- le gain en qualité pour une compression non adaptative est de l'ordre de $1.5 \text{ dB} \pm 0.5 \text{ dB}$, quel que soit le taux de compression
- le gain en qualité pour une compression adaptative est de l'ordre de $0.5 \text{ dB} \pm 0.5 \text{ dB}$ quel que soit le taux de compression
- la diminution de la distorsion consécutive à l'utilisation de la transformation en cosinus plutôt que la transformation de Hadamard est moins importante pour une compression adaptative que pour une compression non adaptative.

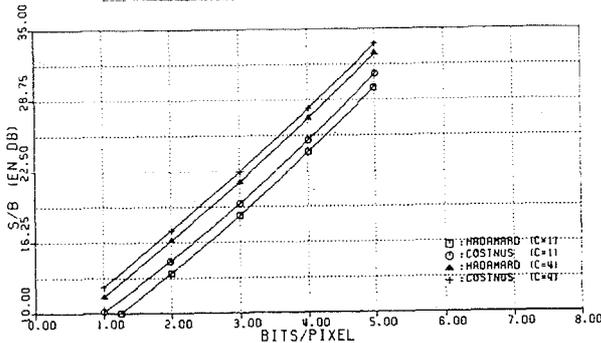


Figure 4 : Transformations unitaires

La rapidité d'exécution est mesurée en évaluant le nombre d'instructions par pixel. Cette mesure est basée sur les performances du TMS 320 qui est le processeur de base du calculateur parallèle CAPITAN développé pour le traitement d'images par MATRA (5). Pour plus de détails on renvoie le lecteur à la référence (6).

Pour une image 256×256 divisée en blocs de 8×8 pixels, l'algorithme de transformation de Hadamard nécessite 6 instructions par pixel. L'algorithme de transformation en cosinus impose 8.72 instructions par pixel (7).

2 - UTILISATION SIMULTANEE DES TRANSFORMATIONS EN COSINUS ET DE HADAMARD

Les distorsions résultant d'une compression par transformation de Hadamard et d'une compression par transformation en cosinus ont été comparées bloc par bloc. On trouve alors que pour certains blocs, Hadamard conduit à une distorsion plus faible que cosinus. Cependant sur une image, le gain apporté par l'utilisation simultanée de ces deux transformations est très faible (inférieur à 0.2 dB).

IV - ADAPTATIVITE DE L'ALLOCATION DES BITS

Après avoir regroupé les blocs image transformés, par classification d'une mesure de leur activité, les cartes de variances des points transformés sont calculées. C'est en fonction de ces variances, ou des écarts-types correspondants, que l'on attribue un nombre de bits plus ou moins important à chaque point transformé appartenant à une des C classes. La minimisation de la distorsion amène classiquement (1) à une allocation de bits optimum selon la relation :

$$b_k = \log_2 a \sigma_k$$

σ_k : écart-type du point k

a : constante reliée à la distorsion

Le paragraphe 1 expose le codage scalaire et introduit un codage vectoriel. Une évaluation et une comparaison de ces deux méthodes sont présentées. Le paragraphe 2 étudie la sensibilité au bruit inhérente à ces deux techniques de codage. Le paragraphe 3 est consacré aux différents estimateurs de variance qui peuvent être utilisés.

1 - CODAGE SCALAIRE-CODAGE VECTORIEL

Le codage employé usuellement est un codage scalaire. À chaque point transformé on attribue un mot de code. Cela implique que :

- le nombre de bits alloués à chaque point est entier
- le nombre de niveaux de quantification est une puissance de 2.

Le codage scalaire conduit donc à une approximation grossière du nombre optimum de niveaux de quantification n_k donné par :

$$n_k = 2^{b_k} = a \sigma_k$$

Pour obtenir une répartition plus correcte du nombre total de niveaux de quantification disponibles pour un taux de compression donné, il convient d'allouer \hat{n}_k niveaux de quantification à chaque point transformé, \hat{n}_k étant égal à la valeur entière la plus proche de n_k . On voit qu'alors, le nombre de bits attribué à chaque point n'est pas un nombre entier. Il n'est donc pas possible d'utiliser le codage scalaire.

Une solution consiste à former la valeur x telle que :

$$x = x_1 + \hat{n}_1 (x_2 + \hat{n}_2 (x_3 + \dots + \hat{n}_{M-1} x_M \dots))$$

x_k : niveau occupé par le point k

M : nombre de points dans un bloc image

et à coder x avec un mot de longueur b bits, avec :

$$b = \prod_{k=1}^M \hat{n}_k + 1$$

On a donc attribué un seul mot de code à l'ensemble des points d'un bloc image. On dira que l'on a effectué un codage vectoriel (ne pas confondre avec quantification vectorielle (3)). Le décodage s'effectue par une cascade de divisions arithmétiques de x puis des quotients par $\hat{n}_1, \hat{n}_2, \dots, \hat{n}_{M-1}$. Pour plus de détails sur cette méthode, le lecteur se reportera à l'article original (9).

La simulation de ces deux techniques de codage, a été menée sur plusieurs images. On retient essentiellement que le codage vectoriel diminue la distorsion par rapport au codage scalaire de 0.25 dB environ quelque soit le taux de compression. Ces résultats sont dus à la meilleure utilisation du nombre de bits disponibles par la méthode de codage vectoriel.

2 - FORMATTAGE DES DONNEES ET BRUIT DE TRANSMISSION

Pour le codage scalaire, les mots de code des points transformés sont concaténés en paquets de 16 bits. Pour le codage vectoriel, les points transformés sont regroupés jusqu'à obtention de paquets de 16 bits. A chaque groupe de points on applique le codage vectoriel et l'on attribue un mot de code que l'on transmet. Ce découpage des points en paquets de 16 bits n'a que des conséquences négligeables sur les performances de la méthode par rapport au codage du bloc entier. Codages vectoriel et scalaire nécessitent tous deux 1.98 instructions par pixel au maximum.

Le regroupement des points transformés en paquets de 16 bits permet, en outre, de rendre le codage vectoriel moins sensible au bruit de transmission (9).

3 - ESTIMATEURS DE VARIANCES OU D'ECARTS-TYPES

Pour une répartition gaussienne, la variance ou l'écart-type d'une variable aléatoire peut être estimé de diverses façons. La plus usuelle est de calculer :

$$\sigma^2 = \frac{1}{S} \sum_{i=1}^S (x_i - m)^2$$

ETUDE COMPARATIVE DE PLUSIEURS ALGORITHMES ADAPTATIFS DE COMPRESSION D'IMAGE PAR TRANSFORMATION UNITAIRE

x_i : valeur de la i /ème réalisation
 S : nombre total de réalisations
 m : moyenne des réalisations x_i

Cet estimateur oblige à calculer les $\hat{\sigma}$ correspondant pour l'utilisation du codage vectoriel et pour le calcul des coefficients de normalisation.

Afin de réduire le temps calcul, on propose (6) l'utilisation de l'estimateur d'écart-type suivant :

$$\hat{\sigma} = \sqrt{\frac{\pi}{2}} \cdot \frac{\sum_{i=1}^S |x_i - m|}{S}$$

La différence de qualité obtenue par l'utilisation du second estimateur par rapport au premier est absolument négligeable à 2 bits/pixel. Elle croit avec le débit mais reste faible (inférieure à 0.4 dB à 5 bits/pixel).

Le calcul des cartes d'allocation de bits nécessite, pour le codage scalaire, 0.05 instruction par pixel quel que soit l'estimateur utilisé. Le calcul des cartes d'allocation de niveaux demande 0.13 instruction par pixel avec le premier estimateur et 0.004 instruction avec le deuxième estimateur.

Le calcul des coefficients de normalisation nécessite 2.28 instructions par pixel avec le premier estimateur et 0.25 instruction par pixel avec le second.

V - ADAPTATIVITE DE LA DISTRIBUTION DES NIVEAUX DE QUANTIFICATION

La minimisation de la distortion implique l'utilisation d'un quantificateur de Max sur la répartition expérimentale des réalisations de chaque point transformé. Cependant cette procédure est très coûteuse en temps calcul. Deux modèles de répartition sont communément utilisés :

- la répartition gaussienne
- la répartition laplacienne

Le problème du choix entre ces deux modèles est très controversé (10). Dans le premier paragraphe, un critère de choix est proposé. Les résultats obtenus pour les compressions adaptatives et non adaptatives sont exposés. On évoque l'utilisation des deux modèles de répartition pour quantifier différents point transformés d'un même bloc image. Le paragraphe 2 donne une évaluation des performances de divers quantificateurs.

1 - REPARTITION GAUSSIENNE OU LAPLACIENNE

Le critère de maximum de vraisemblance est utilisé pour choisir entre le modèle gaussien et le modèle laplacien (6).

Les résultats des simulations sont exposés sur la figure 5 pour une des images de télédétection comprimée par une transformation de Hadamard. On retient essentiellement que :

- le modèle gaussien est meilleur pour une compression adaptative
- le modèle laplacien est meilleur pour une compression non adaptative

Ceci peut être vu comme une conséquence du regroupement des blocs image en classes de statistiques homogènes. En effet, les points transformés d'un groupe de blocs de statistique homogène ont une répartition gaussienne conformément au théorème de la limite centrale. Pour une compression non adaptative on a superposition de plusieurs répartitions gaussiennes d'écart-types différents, ce qui conduit à une répartition d'allure laplacienne.

Pour la compression non adaptative, le nombre de points transformés gaussiens augmente lorsque la taille des blocs augmente. C'est une conséquence

du théorème de la limite centrale. Pour la compression adaptative le nombre de points transformés gaussiens diminue lorsque la taille des blocs diminue. La raison provient de la plus grande adaptativité des blocs de petite taille. En augmentant la taille des blocs, on sépare moins finement les zones de statistiques homogènes.

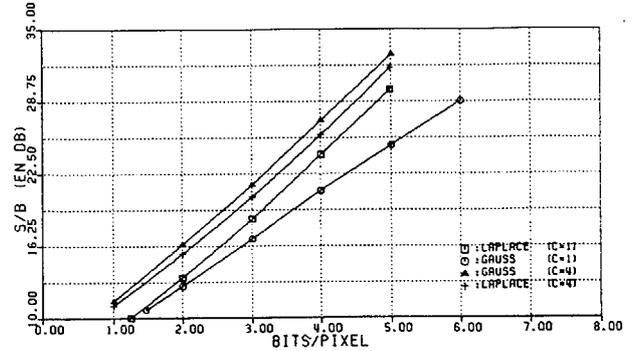


Figure 5 : Modèles de répartition

Ces résultats sont dépendant de la transformation unitaire utilisée. On observe que le nombre de points transformés gaussiens est plus grand pour la transformation de Hadamard que pour la transformation en cosinus lors de l'utilisation d'une compression adaptative. L'adaptativité est donc moins efficace avec la transformation en cosinus, qu'avec la transformation de Hadamard. Cette observation permet de mieux comprendre la figure 4.

L'utilisation des deux modèles de répartition pour quantifier différents points transformés d'un même bloc image n'apporte pas d'amélioration significative de la qualité de reconstruction de l'image comprimée (inférieure à 0.02 dB pour la transformation en cosinus).

2 - QUANTIFICATEURS

Le quantificateur de Max et le quantificateur uniforme ont été testés sur des images télédétection pour une compression adaptative.

La figure 6 montre que :

- le quantificateur de Max donne une distortion plus faible que le quantificateur uniforme pour un taux de compression donné
- le quantificateur uniforme s'approche d'autant plus des performances du quantificateur de Max que le taux de compression est important. A 2 bits/pixel, la différence en distortion est de l'ordre de 0.1 dB.

Sur le plan temps calcul, le quantificateur uniforme est le plus rapide puisqu'il nécessite 1 instruction par pixel contre 7.88 instructions pour le quantificateur de Max.

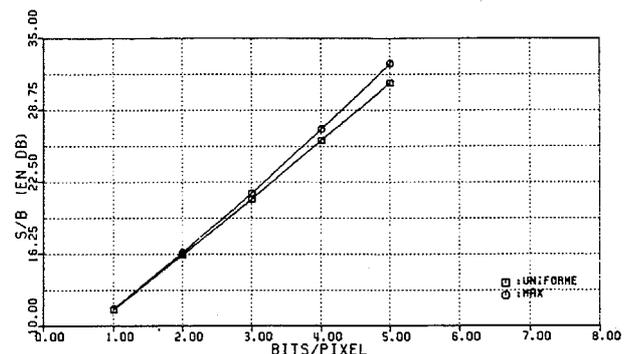


Figure 6 : Quantificateurs



ETUDE COMPARATIVE DE PLUSIEURS ALGORITHMES ADAPTATIFS
DE COMPRESSION D'IMAGE PAR TRANSFORMATION UNITAIRE

VI - AUGMENTATION DE LA RAPIDITE D'EXECUTION

Deux propositions sont faites pour diminuer le temps calcul.

1 - MESURE D'ACTIVITE

La mesure d'activité utilisée est donnée par :

$$A_i = \sum_{n=0}^{M-1} \sum_{m=0}^{M-1} F_{nm}^2 - F_{00}^2 \text{ pour le bloc } i$$

Le calcul des A_i pour des blocs 8x8 nécessite 1.97 instructions par pixel. L'utilisation de la mesure MACE(R,S) (11) permet de réduire notablement le temps calcul puisqu'elle consiste en une mesure localisée de l'énergie

$$\text{MACE}(R,S) = \sum_{n=0}^{R-1} \sum_{m=0}^{R-1} F_{nm}^2 - \sum_{n=0}^{S-1} \sum_{m=0}^{S-1} F_{nm}^2$$

Pour des blocs 8x8, avec des valeurs de R et S égales respectivement à 4 et 1, les performances de la chaîne de compression sont pratiquement identiques à celles obtenues par la mesure A_i . La mesure MACE(4,1) nécessite 0.66 instructions par pixel.

Le nombre d'opérations nécessaires à la classification des activités par segmentation de la cumulative (3) est indépendant de la mesure d'activité adoptée. Pour une image 256x256 avec un découpage en blocs 8x8, elle demande 0.53 instructions par pixel.

2 - NOMBRE DE BLOCS UTILISES POUR LA STATISTIQUE

La construction des cartes d'allocation de bits ou de niveaux nécessite l'estimation des écarts-types de chaque point de l'espace transformé. Au lieu d'effectuer cette estimation sur l'ensemble des blocs d'une classe, on peut ne considérer qu'une partie de ces blocs. La figure 7 montre l'évolution des performances avec le nombre de blocs pris en compte. On voit que la réduction par un facteur 4 du nombre de calculs peut être obtenu pour une dégradation mineure de la qualité de reconstruction à 2 bits/pixel. On passe ainsi de 1.97 instructions par pixels à 0.49 instructions par pixel pour le calcul des écarts-types.

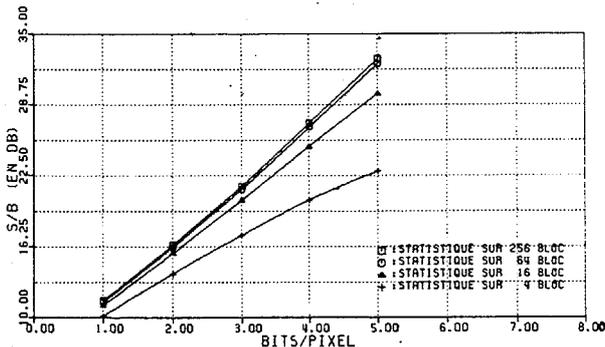


Figure 7 : Statistiques partielles

VII - CONCLUSION

Trois types d'adaptativité ont été étudiés pour la compression par transformation utilisant la quantification par blocs associée à un codage par zones :

- l'adaptativité de la transformation
- l'adaptativité de l'allocation du nombre de bits
- l'adaptativité de la distribution des niveaux de quantification

Une évaluation et une comparaison des différentes

étapes de la chaîne de compression ont été présentées. On retiendra les principaux résultats :

- Pour obtenir simultanément :
 - . une minimisation des frais généraux
 - . un nombre optimum de classes d'adaptativité
 - . un bon compromis pour la taille des blocs il convient de traiter les images par morceaux de 256x256 pixels découpés eux-mêmes en blocs de 8x8 pixels, répartis en 4 classes d'adaptativité.
- L'adaptativité de l'allocation du nombre de bits est la source principale de l'amélioration des performances de la chaîne de compression
- La séparation en classes de statistiques homogènes sur un critère d'activité est plus efficace avec la transformation de Hadamard, qu'avec la transformation en cosinus. La conséquence est que l'écart entre les performances obtenues par cosinus par rapport à Hadamard diminue avec l'adaptativité
- Le codage vectoriel des points transformés permet une meilleure distribution du nombre de bits disponibles. On observe alors une diminution de la distortion pour un temps calcul identique par rapport au codage scalaire usuel
- La répartition des points transformés dans les zones de statistique homogène est gaussienne. Pour la compression non adaptative, la répartition de chaque point de l'espace transformé est une somme de gaussiennes centrées en zéro et d'écarts-types différents, ce qui conduit à un profil laplacien.

L'optimisation du choix des paramètres rencontrés dans la chaîne de compression adaptative permet d'améliorer les performances. On peut obtenir ainsi, par rapport à l'algorithme bien connu de Chen et Smith (3), un gain en qualité de 0.3 dB et une amélioration du temps calcul d'un facteur 1.25, pour un débit de 2 bits/pixel. L'utilisation des étapes les plus rapides proposées dans cet article, en conservant toutefois la transformation en cosinus, permet d'améliorer le temps calcul d'un facteur 2.5 pour une dégradation minime (inférieure à 0.1 dB).

BIBLIOGRAPHIE

- (1) A.K. Jain, "Image data compression : a review", Proc. IEEE, vol.69, pp 349-389, Mar. 1981
- (2) M. Kunt, A. Ikonopoulos, et M. Kocher, "Compression d'images : méthodes de la deuxième génération", Premier colloque image GRETSI-CESTA, pp 93-105, Mai 1984
- (3) W.H. Chen and C.H. Smith, "Adaptive coding of monochrome and color images", IEEE Trans. Commun., vol. COM-25, pp 1285-1292, Nov. 1977
- (4) J. Max, "Quantizing for minimum distortion", IRE Trans. Inform. Theory, vol. IT-6, pp 7-12, 1960
- (5) G. Gaillat, "Le calculateur parallèle CAPITAN : 600 Mips pour l'imagerie temps réel", Premier colloque image GRETSI-CESTA, pp 473-481, Mai 1984
- (6) S. Muller, "Comparaison et évaluation d'algorithmes pour la compression d'images multispectrales", Thèse de doctorat de l'université de Paris 6, Septembre 1985
- (7) F.A. Kamangar, and K.R. Rao, "Fast algorithms for the 2-D discrete cosine transform", IEEE Trans. Comput., vol. C-31, pp 899-906, Sept. 1982
- (8) R.M. Gray, "Vector quantization", IEEE ASSP magazine, pp 4-29, Apr. 1984
- (9) A. Ayoun, G. Gaillat, et S. Muller, "Codage vectoriel pour la compression par transformation unitaire", Colloque sur le traitement du signal et ses applications GRETSI, Mai 1985
- (10) R.C. Reiningger, and J.D. Gibson, "Distributions of the two-dimensional DCT coefficients for images", IEEE Trans. Commun., vol COM-31, pp 835-839, June 1983
- (11) J.K. Wu, and R.E. Burge, "Adaptive bit allocation for image compression", G.G.I.P., vol 19, pp 392-400, Aug 1982