

HUITIEME COLLOQUE SUR LE TRAITEMENT DU SIGNAL ET SES APPLICATIONS

NICE du 1^{er} au 5 JUIN 1981

OPTIMALISATION DE PROCÉDES CLASSIQUES SIMPLES DE COMPRESSION D'IMAGES NUMERISEES

MM. O de CLERCQ et A. MARGUINAUD

THOMSON-CSF/DITM , 6 rue Adolphe LALYRE - 92400 COURBEVOIE

RESUME

On connaît une grande variété d'algorithmes pour réduire la quantité de données à stocker ou à transmettre. Les procédés de réduction de données peuvent se classer selon la taille mémoire nécessaire : les taux de réduction les plus élevés sont obtenus en utilisant une mémoire de trame ; cela permet de ne transmettre que les zones nouvelles d'une image.

Quelle que soit la classe des performances recherchées, il est donc essentiel de disposer de procédés efficaces pour coder des zones d'images entièrement nouvelles. La valeur du rayon de corrélation des brillances d'une image incite à se limiter aux codages n'utilisant que la connaissance de la ligne précédente. Trois méthodes de réduction paraissent intéressantes :

- la codage par plage,
- le codage par pente,
- le codage par transformée.

Nous décrivons et présentons ici les méthodes et résultats obtenus par simulation sur images réelles : des taux de réduction de l'ordre de 10 avec un écart maximum de 1 ou 2 unité(s) entre les valeurs réelles et transmises des brillances.

Cette amélioration notable de l'efficacité par rapport aux résultats classiques résulte de l'extension du codage par plage de une à deux dimensions, ce qui permet d'exploiter simultanément la corrélation en ligne et la corrélation en colonne. En outre, certains aspects de la mise en oeuvre des traitements à l'émission ou à la réception permettent d'assurer une bonne résistance aux erreurs de transmission.

SUMMARY

OPTIMIZATION OF CLASSICAL DIGITAL IMAGE COMPRESSION TECHNIQUES

We have at our disposition a considerable variety of algorithms for reducing the number of bits to be transmitted or stored. These can be classified according to the amount of memory required : the maximum reduction is obtained by memorizing an entire frame at a time; which permits us to transmit only those portions of the image which have changed since the last frame.

Regardless of the degree of compression desired, it is essential to have efficient coding techniques for those portions of the image which have changed. The range of brightness correlations which are encountered leads us to adopt only those coding techniques which require a knowledge of the preceding line.

Under these circumstances, three methods of compression seem to be of interest :

- run-length encoding of discrete luminosity values
- slope encoding as a function of incremental luminosity changes
- transform coding.

We describe herein the methods and results obtained by computer simulation on real images. A compression ratio of 10 is obtained with a maximum error of 1 or 2 values of brilliance in the reconstituted image.

The considerable improvement over classical techniques is obtained by extending the run-length technique to two dimensions, which permits us to take advantage of both line and column brilliance correlations. Moreover, our implementation of the run-length algorithm provides an excellent image quality in the presence relatively high bit error rates.



1. INTRODUCTION

On connaît une grande variété d'algorithmes permettant de réduire la quantité de données à stocker ou à transmettre. Les procédés de réduction de données peuvent se classer selon la taille de mémoire nécessaire : les taux de réduction les plus élevés sont obtenus en utilisant une mémoire de trame. Cela permet de ne transmettre que les zones nouvelles d'une image.

Quelles que soient les performances recherchées, il est donc essentiel de disposer de procédés efficaces pour coder des zones d'images entièrement nouvelles. La valeur du rayon de corrélation des brillances incite à se limiter aux codages n'utilisant que la connaissance de la ligne précédente. Trois méthodes paraissent intéressantes : le codage par plage, le codage par pente, le codage par transformée.

Le principe du codage par transformée consiste à transmettre pour chaque ligne une version tronquée d'une transformée \mathcal{F} de la différence entre la ligne nouvelle et la ligne précédente reconstituée.

On désigne par L_i la suite des brillances de la i ème ligne, \mathcal{C} l'opérateur de troncature qui consiste par exemple à ne transmettre que certaines composantes de la transformée et par \mathcal{F}^{-1} la transformée inverse de \mathcal{F} .

La suite des opérations est la suivante :

On transmet :

$$\begin{aligned} T_1 &= \mathcal{C} \mathcal{F} L_1 \\ T_2 &= \mathcal{C} \mathcal{F} [L_2 - \mathcal{F}^{-1} T_1] \\ T_i &= \mathcal{C} \mathcal{F} [L_i - \mathcal{F}^{-1} T_{i-1}] \end{aligned}$$

On décode ligne à ligne de la façon suivante :

$$\begin{aligned} \hat{L}_1 &= \mathcal{F}^{-1} T_1 \\ \hat{L}_2 &= \hat{L}_1 + \mathcal{F}^{-1} T_2 \\ \hat{L}_i &= \hat{L}_{i-1} + \mathcal{F}^{-1} T_i \end{aligned}$$

L'intérêt des transformations en réduction de données provient de ce que les composantes ont des dynamiques très différentes entre elles, de sorte que l'on peut coder chaque composante avec un nombre de bits adapté à sa dynamique (troncature).

L'utilisation des transformées donne des facteurs de compression inférieurs à ceux du codage par plage, de plus la complexité de mise en oeuvre reste importante malgré l'utilisation d'algorithmes rapides.

Le principe du codage par pente consiste à coder en une seule fois un intervalle de points dont la

fonction représentative des brillances est une droite de pente croissante ou décroissante. L'intérêt de ce procédé provient de ce que l'on peut utiliser un critère de tolérance adapté à la pente locale ; les écarts les plus grands sont sur les pentes les plus grandes. En pratique, les facteurs de réduction sont de l'ordre de la moitié de ceux obtenus avec le codage par plage, car il faut en plus coder les variations de pente dues à des sauts brusques de valeurs.

La dernière méthode étudiée est le codage par plage (run-length).

A partir de cette méthode, on a cherché des procédés simples permettant d'améliorer l'efficacité du codage par plage.

On a ainsi défini une famille d'un millier d'algorithmes en combinant les procédés élémentaires.

Les critères d'appréciation des résultats sont le facteur de réduction, la complexité des algorithmes en nombre d'opérations élémentaires par point et l'écart-type de l'image par rapport à l'image originale.

L'efficacité de la méthode résulte de l'extension du codage par plage de une à deux dimensions. Ce procédé peut être étendu au cas d'images animées ; il faut alors ajouter un contrôle du seuil de tolérance pour uniformiser le facteur de réduction. L'utilisation d'une mémoire de trame permet d'espérer des résultats encore meilleurs en raison de la corrélation entre les images.

2. RAPPEL SUR LE CODAGE PAR PLAGE

Le codage par plage consiste à coder en une seule fois les brillances de points le long d'une ligne. Un intervalle de ce type ou plage est constitué par une suite de brillances dont les valeurs sont égales.

Exemple : La suite des nombres suivants :

$$2, 2, 2, 2, 2, 5, 5, 5, 7, 7, 7, 7, \dots$$

└──────────┘
plage

peut se réduire à la séquence de couples :

$$(5, 2), (3, 5), (4, 7), \dots$$

↑ ↑
longueur valeur

On définit le taux de réduction k par le rapport entre la longueur de la suite numérique avant et après codage.

OPTIMALISATION DE PROCÉDES CLASSIQUES SIMPLÉS DE COMPRESSION D'IMAGES
NUMÉRISÉES

Les améliorations apportées au codage par plage proviennent des propriétés de la source, de la corrélation, de la répartition statistique des sous-suites à coder et de la tolérance visuelle des utilisateurs.

On a analysé onze procédés élémentaires permettant d'améliorer notablement la méthode simple.

La figure 1 donne un exemple des résultats obtenus en fonction du nombre d'opérations arithmétiques par point selon les procédés utilisés (a à k) définis dans la suite du texte.

Remarque : Le codage par plage sans introduire de seuil de tolérance donne un facteur de réduction de 1,1.

Algorithmes	Nombre d'opérations	Tolérance 1 %		Tolérance 2 %	
		Facteur de réduc.	Ecart de type (%)	Facteur de réduc.	Ecart de type (%)
1 : a+b	3	1,4	0,6	1,7	0,68
2 : a+b+c+d	5	3,8	0,64	5,5	0,92
3 : a+b+c+d+e	6	4,3	0,72	5,8	0,92
4 : a+b+c+d+e+f	7	4,0	0,72	5,5	0,96
5 : a+b+c+d+e+f+g	9	4,3	0,8	5,8	1,0
6 : a+b+c+d+e+f+g+h	11	4,8	0,84	6,1	1,0
7 : a+b+c+d+e+f+g+h+i	12	5,0	0,84	6,3	1,0
8 : a+b+c+d+e+f+g+h+i+j	13	5,2	0,84	6,5	1,04
9 : a+b+c+d+..... k	14	6,5	0,76	9,1	0,84

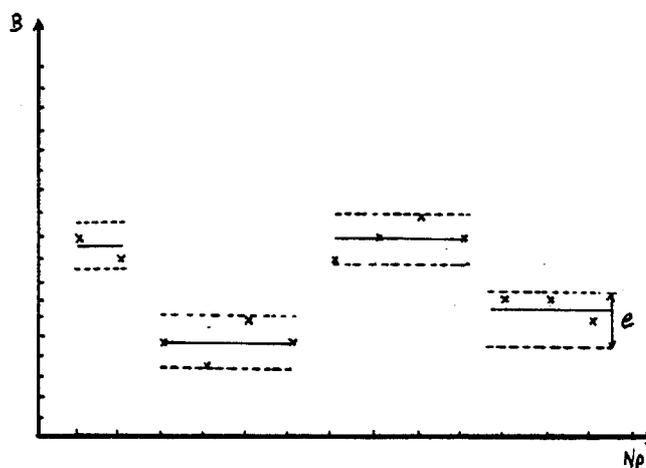
Figure 1 : Exemple de résultats obtenus en fonction du nombre d'opérations arithmétiques par point selon les différents algorithmes utilisés.

3.- PRESENTATION DES ALGORITHMES ELEMENTAIRES

a) Tolérance visuelle sur la valeur des brillances le long d'une ligne

On considère que la brillance est constante aussi longtemps qu'elle reste dans un intervalle de valeurs caractéristiques du niveau de tolérance donné. La valeur codée correspond à la valeur médiane des valeurs comprises dans l'écart tolérable (figure 2).

Chaque point de l'image originale est codé sur 8 bits, ce qui correspond à 256 niveaux de brillances.



B : Niveaux de brillance
Np : Points sur une ligne
e : Tolérance

Figure 2 : codage par plage ; tolérance sur la valeur des brillances

b) Optimisation statistique du nombre de bits du champ alloué au codage de la longueur

La statistique des longueurs de plage pour un ensemble d'images montre que 80 % des longueurs sont inférieures à 8. On code les longueurs sur 3 bits. Les plages de plus grande longueur sont codées en plusieurs fois.

c) Corrélation entre les valeurs de brillance le long d'une ligne

En considérant qu'il n'y a pas de grands écarts de niveaux de brillance entre plages successives, on code la différence avec la valeur de la plage précédente (figure 3).

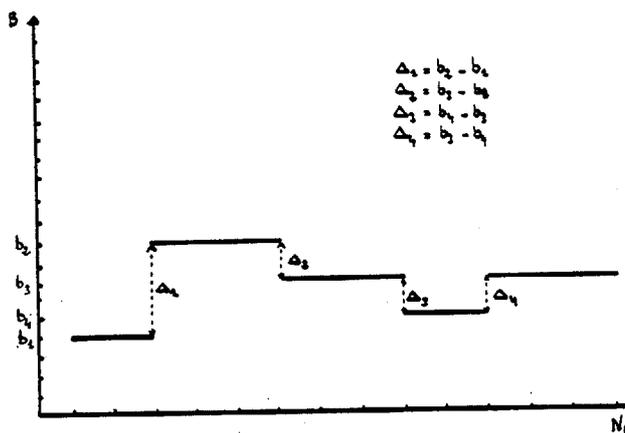


Figure 3 : Codage de la différence



OPTIMALISATION DE PROCÉDES CLASSIQUES SIMPLES DE COMPRESSION D'IMAGES NUMÉRISÉES

d) Optimalisation statistique du nombre de bits du champ alloué au codage de la différence des brillances

La statistique sur les différences donne un écart rarement dépassé de ± 16 , on code la différence sur 5 bits. Lorsque la différence est supérieure à $|16|$, on code la valeur réelle sur 8 bits. On distingue la valeur réelle de la différence par un bit rajouté au champ réservé au codage de la valeur des plages.

e) Utilisation du coefficient de corrélation entre les valeurs de brillance le long d'une ligne (limitation de la propagation des erreurs)

Si λ est ce coefficient, b_i la brillance du $i^{\text{ème}}$ point de la ligne, on code la différence $b_{i+1} - \lambda b_i$ au lieu de $b_{i+1} - b_i$. Le calcul donné ci-dessous montre que le facteur de réduction est multiplié par : $\frac{2}{1+\lambda}$.

$$E = \sum (b_i - b_{i+1})^2$$

$$E_\lambda = \sum (b_{i+1} - \lambda b_i)^2$$

$$E_\lambda = \sum b_{i+1}^2 - 2\lambda \sum b_i b_{i+1} + \lambda^2 \sum b_i^2$$

Cette quantité est minimale pour $\lambda = \frac{\sum b_i b_{i+1}}{\sum b_i^2}$

En remplaçant cette quantité dans E_λ , on trouve :

$$E_\lambda = \sum b_i^2 (1 - \lambda^2)$$

En introduisant λ dans E , on trouve :

$$E = 2 \sum b_i^2 (1 - \lambda)$$

Le rapport est donc : $\frac{E}{E_\lambda} = \frac{2}{1+\lambda}$

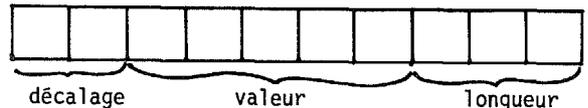
Si $\lambda = 0,5$, l'amélioration vaut 4/3.

f) Corrélation avec la ligne précédente sans biais (cf codage par transformée)

On calcule la différence entre la ligne nouvelle et la ligne précédente reconstituée après décodage. Ce procédé apporte peu d'améliorations sur le taux de réduction, en revanche le confort visuel est considérablement amélioré.

g) Corrélation avec la ligne précédente avec biais

La corrélation locale ligne à ligne n'est pas forcément verticale. On effectue des comparaisons avec la ligne précédente reconstituée, décalée d'un point à droite, à gauche, ou non décalée. Chaque plage est représentée par 3 champs de bits:



- Champ décalage : 2 bits

- | | |
|---|---|
| 1 | 0 |
|---|---|

 pour en face
- | | |
|---|---|
| 0 | 0 |
|---|---|

 pour à gauche
- | | |
|---|---|
| 0 | 1 |
|---|---|

 pour à droite
- | | |
|---|---|
| 1 | 1 |
|---|---|

 pour la valeur réelle codée

- Champ valeur : 5 bits

Différence avec la valeur de la plage précédente ; si le premier champ est

1	1
---	---

 alors le 2ème champ comporte 8 bits.

- Champ longueur : 3 ou 4 bits

Pour coder chaque plage, on utilise comme référence la version de la ligne précédente qui donne la plus grande longueur pour cette plage. Cette dernière méthode permet à la fois d'obtenir des images de bonne définition et un taux de réduction amélioré.

h) Corrélation entre les longueurs de plages sur des lignes successives

On peut exploiter la corrélation ligne à ligne en remarquant que, dans une zone de deux lignes consécutives, la longueur des plages varie peu. On peut alors coder non pas la longueur des plages mais la différence avec la longueur de la plage correspondante sur la ligne précédente. On associe les plages de lignes consécutives par les centres de chaque plage. Cette méthode permet de diviser par deux environ le nombre de bits du champ de longueur.

OPTIMALISATION DE PROCÉDES CLASSIQUES SIMPLES DE COMPRESSION D'IMAGES NUMÉRISÉES

i) Utilisation du coefficient de corrélation entre brillances de lignes successives

Les valeurs de brillance sont corrélées le long d'une ligne mais aussi entre lignes successives. On peut donc pondérer la différence entre les lignes par le coefficient de corrélation λ entre les brillances, comme dans le cas e), le facteur de réduction est multiplié par $\frac{2}{1+\lambda}$

Les algorithmes sont alors les suivants :

au codage : $T_1 = \mathcal{C}(L_1)$

$T_2 = \mathcal{C}(L_2 - \lambda \mathcal{C}^{-1} T_1)$

$T_i = \mathcal{C}(L_i - \lambda \mathcal{C}^{-1} T_{i-1})$

au décodage : $\hat{L}_1 = \mathcal{C}^{-1} T_1$

$\hat{L}_2 = \lambda \hat{L}_1 + \mathcal{C}^{-1} T_2$

$\hat{L}_i = \lambda \hat{L}_{i-1} + \mathcal{C}^{-1} T_i$

avec

L_i : ligne à coder

\hat{L}_i : ligne reconstituée après décodage

T_i : ligne codée

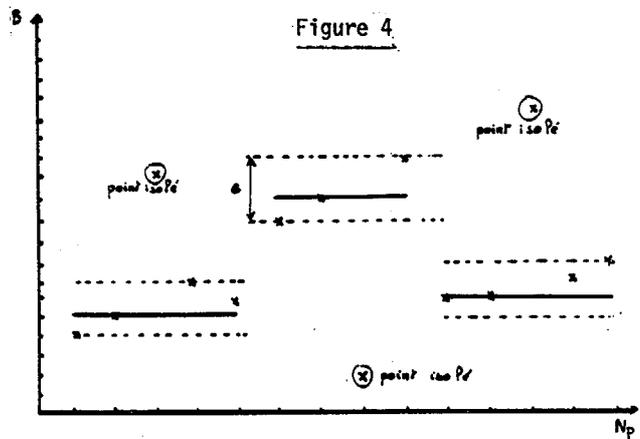
\mathcal{C} : codage

\mathcal{C}^{-1} : décodage.

j) Débruitage des images par élimination des points isolés des plages

Lorsqu'on détermine la longueur d'une plage, on considère que la brillance reste constante aussi longtemps qu'elle reste dans un intervalle de valeurs e . Cependant, on constate que certains points isolés s'écartent notablement de cet intervalle, ce qui diminue la longueur des plages. Ces points isolés sont dus au bruit des circuits électroniques qui sont à plus large bande que ceux de l'optique.

La méthode consiste à ne pas tenir compte de ces points lorsqu'ils sont isolés et que leur différence avec la valeur médiane des points constituant la plage est supérieure à une valeur (Figure 4).



h) Utilisation de codes à préfixe (codes de HUFFMAN)

Dans un code de HUFFMAN, les mots du code les plus courts sont affectés aux mots les plus fréquents. La longueur moyenne d'un mot du code est donnée par $L = \sum_j l_j p_j$ dans lequel p_j est la probabilité d'occurrence du mot de longueur l_j

On utilise cette méthode pour coder les deux suites longueur de plages, et valeur des plages correspondantes. On applique ensuite les algorithmes de codage par plage en remplaçant les longueurs et les valeurs de plage par les mots du code correspondant.

4.- CONCLUSIONS

L'utilisation des procédés décrits, à complexité comparable, conduit à des performances meilleures que celles obtenues à partir du codage par pentes ou de techniques plus savantes comme les transformées. Par ailleurs, l'algorithme de compression étant obtenu par combinaison de procédés élémentaires simples, il convient de l'adapter à chaque type d'image considérée (croquis au trait, plans, fac-similé, ...) par des simulations.

Pour éviter la propagation des erreurs, on peut envisager d'ajouter sur chaque ligne un code détecteur d'erreur ; en cas d'erreur détectée, on prend la ligne à l'instant précédent. Comme la longueur des représentations de chaque ligne



OPTIMALISATION DE PROCEDES CLASSIQUES SIMPLES DE COMPRESSION D'IMAGES
NUMERISEES

est variable, on utilise un conditionnement type HDLC, ce qui augmente la longueur moyenne seulement de quelques pour cent , ce qui peut être largement compensé par la récupération des périodes inter-lignes et interimages des transmissions TV classiques.
