

# Pyramide résiduelle pour la compression d'images

Didier Calle et Annick Montanvert

Equipe INFODIS, TIMC-IMAG, URA CNRS D1618  
 Institut Albert Bonniot, Domaine de la Merci, 38706 La Tronche Cedex, France  
 Email: Didier.Calle@imag.fr, Annick.Montanvert@imag.fr

## RÉSUMÉ

Dans cet article, nous révisons la pyramide laplacienne introduite initialement pour la compression des images. Nous approfondissons le principe d'expansion des niveaux d'une pyramide gaussienne qui est essentiel à la construction d'une pyramide laplacienne. L'expansion classique consiste en une simple interpolation. Nous montrons qu'une interpolation ne se justifie pas dans un contexte de compression, et qu'il faut finalement procéder par induction. Ce nouveau principe d'expansion inductive donne naissance à un modèle pyramidal que nous appelons "pyramide résiduelle". Nous appliquons ce nouveau modèle à la compression d'images sans perte d'information.

## ABSTRACT

We review here the Laplacian pyramid originally defined for image compression. We study the expansion principle of Gaussian pyramid layers which is essential to build a Laplacian pyramid. The standard expansion is a simple interpolation scheme. We show that such scheme is unsuited, and that a layer must be expanded by induction. This new scheme based on inductive expansion yields a new pyramidal model called "residual pyramid". We use this new model for lossless image compression.

## 1 Introduction

Dans cet article, nous révisons la pyramide laplacienne introduite par Burt et Adelson pour la compression des images [2]. Sa construction est basée sur la manipulation d'une pyramide gaussienne dont l'objet est de représenter une image à des niveaux de résolution décroissants [1]. Plus précisément, un niveau  $k$  de la pyramide laplacienne est le résultat d'une soustraction d'un niveau  $k$  d'une pyramide gaussienne et de sa version estimée par expansion du niveau supérieur  $k - 1$ . Chaque niveau de la pyramide laplacienne représente une erreur d'estimation, due à l'algorithme d'expansion utilisé. L'efficacité d'une compression réalisée à partir d'une pyramide laplacienne passe donc par l'amélioration de l'algorithme d'expansion : notre contribution se situe à ce niveau.

Nous montrons qu'une expansion par interpolation, telle que décrite dans [2], est injustifiée et sous-optimale dans un contexte de compression. Nous montrons que dans un tel contexte, l'algorithme d'expansion doit procéder par induction.

Tout d'abord, nous rappelons précisément le schéma de construction d'une pyramide gaussienne, puis celui d'une pyramide laplacienne.

### 1.1 Pyramide gaussienne

Considérons une image de résolution<sup>1</sup>  $K$ . Une représentation multirésolution de cette image, peut se décrire par le procédé itératif suivant :

$$G_k = \Delta(G_{k+1}) \quad k = K - 1, \dots, 0$$

$$G_k(i, j) = \sum_{m=-2}^2 \sum_{n=-2}^2 W(m, n) \cdot G_{k+1}(2i + m, 2j + n) \quad (1)$$

$$i = 0, \dots, 2^k - 1 \quad j = 0, \dots, 2^k - 1$$

- où :
- $G_k$  représente une image de résolution  $k$ , sachant que  $G_K$  est l'image de départ.
  - $W$  est un noyau gaussien de taille  $5 \times 5$ .
  - $\Delta$  symbolise l'opérateur de réduction.

La collection d'images  $G_K, \dots, G_0$  est appelée pyramide gaussienne. Dans l'équation (1), nous remarquons que chaque niveau de la pyramide gaussienne est le résultat d'une convolution du niveau inférieur par un noyau gaussien combinée à une opération de sous-échantillonnage de facteur 2.

Le noyau gaussien  $W$  est formulé à partir d'un vecteur  $w$  de taille 5, tel que :

$$W(m, n) = w(m) \cdot w(n)$$

$$m = -2, \dots, 2 \quad n = -2, \dots, 2$$

Certaines contraintes imposées à  $w$  (voir [1]) conduisent à exprimer le noyau gaussien à partir d'un seul paramètre  $a$ , d'où :

$$\frac{1}{4} \leq w(0) = a \leq \frac{1}{2}$$

$$w(1) = w(-1) = \frac{1}{4} \quad w(2) = w(-2) = \frac{1}{4} - \frac{a}{2}$$

<sup>1</sup>Une image est considérée comme étant une matrice de taille  $2^k \times 2^k$ . Dans ce cas, nous parlons d'une image de résolution  $k$ .

## 1.2 Pyramide laplacienne

La différence entre un niveau de la pyramide gaussienne et sa version estimée par expansion du niveau supérieur, conduit à un niveau de la pyramide laplacienne. Ce principe se décrit par :

$$L_k = G_k - \Xi(G_{k-1}) = G_k - \hat{G}_k$$

- où :
- $L_k$  représente un niveau  $k$  de la pyramide laplacienne.
  - $\Xi$  symbolise l'opérateur d'expansion.

$L_k$  représente donc un résidu d'estimation associé au niveau de résolution  $k$ . Une pyramide laplacienne étant un codage exact de l'image originale, possédant une entropie et une variance bien plus faibles, elle est utilisée avec un codage entropique pour compresser une image.

## 1.3 Expansion gaussienne (interpolation)

L'algorithme d'expansion, proposé dans [2], est un simple procédé d'interpolation. Plus exactement, le niveau estimé  $\hat{G}_k$  résulte d'une convolution de  $G_{k-1}$  par un noyau gaussien combinée à une opération de sur-échantillonnage, c'est-à-dire :

$$\hat{G}_k(i, j) = 4 \sum_{m=-2}^2 \sum_{n=-2}^2 W(m, n) \cdot G_{k-1} \left( \frac{i-m}{2}, \frac{j-n}{2} \right)$$

Nous appellerons "expansion gaussienne" la transformation décrite par cette équation.

## 2 Expansion inductive

Le paramètre  $a$  du noyau gaussien étant fixé, le taux de compression effectif dépendra de la qualité de l'estimation des niveaux de la pyramide gaussienne, c'est-à-dire de l'algorithme d'expansion.

Dans un contexte de compression, une expansion gaussienne est injustifiée et sous-optimale. En effet, ce type d'expansion ne tient pas compte d'une information importante :  $G_{k-1}$  est issue d'une réduction de  $G_k$  par un opérateur  $\Delta$  fixé (cf. équation (1)). Pour respecter cette contrainte, il faut que l'algorithme d'expansion génère une version estimée de  $G_k$  ayant la propriété suivante :

$$\Delta(\hat{G}_k) = G_{k-1} \quad (2)$$

C'est-à-dire qu'une nouvelle réduction de  $\hat{G}_k$  doit aboutir à une image strictement identique à  $G_{k-1}$ . Il est évident qu'une expansion gaussienne ne permet pas d'aboutir à une telle propriété.

Dans la suite, nous décrivons un algorithme d'expansion respectant la propriété donnée par l'équation (2).

### 2.1 Principe

Notons  $\Omega_k$  l'ensemble fini des images de résolution  $k$  et considérons un niveau  $G_k$  de la pyramide gaussienne. L'application de l'opérateur de réduction  $\Delta$  (le paramètre  $a$  du

noyau étant fixé) sur  $G_k$  génère le niveau  $G_{k-1}$ . L'ensemble des images de  $\Omega_k$  qui génèrent exactement  $G_{k-1}$ , après une réduction par  $\Delta$ , est appelé "ensemble induit" de  $G_{k-1}$  et  $\Delta$ . Cet ensemble est noté  $\Gamma_k$  et peut être décrit formellement par :

$$\Gamma_k = \{I \in \Omega_k \mid \Delta(I) = G_{k-1}\}$$

L'algorithme d'expansion que nous proposons génère uniquement des images appartenant à l'ensemble induit. Nous savons que  $\Gamma_k$  n'est pas vide, puisque  $G_k \in \Gamma_k$ .

Sur la figure 1, nous résumons les expansions gaussienne (à gauche) et inductive (à droite). Pour obtenir théoriquement une

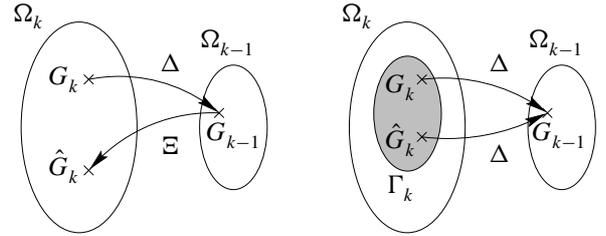


FIG. 1 — Expansion gaussienne et expansion inductive.

image quelconque de l'ensemble induit de  $G_{k-1}$  et  $\Delta$ , nous commençons par initialiser  $\hat{G}_k$  avec une image quelconque de  $\Omega_k$ . Puis nous réduisons cette image par  $\Delta$ , et nous calculons l'erreur entre  $G_{k-1}$  et  $\Delta(\hat{G}_k)$ . Cette erreur est alors utilisée pour corriger  $\hat{G}_k$ , de manière à réduire l'erreur entre  $G_{k-1}$  et une nouvelle réduction de  $\hat{G}_k$ . Cette procédure est itérée jusqu'à l'obtention d'une erreur nulle. Dans ce cas, l'image  $\hat{G}_k$  appartient à l'ensemble induit  $\Gamma_k$  (voir la figure 2).

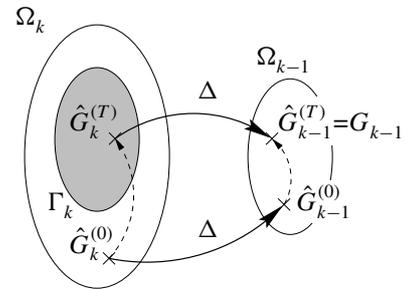


FIG. 2 — Procédure itérative pour obtenir une image de l'ensemble induit. A l'itération  $T$ ,  $\hat{G}_k$  appartient à l'ensemble induit  $\Gamma_k$ .

L'image de l'ensemble induit que nous obtenons dépend uniquement de l'image de  $\Omega_k$  que nous choisissons pour initialiser  $\hat{G}_k$ . Dans le contexte de compression, qui nous occupe dans cet article, il est important de bien choisir l'image initiale pour minimiser l'erreur entre  $G_k$  et sa version estimée  $\hat{G}_k$ . Par conséquent, nous choisirons le résultat d'une expansion gaussienne de  $G_{k-1}$  pour initialiser  $\hat{G}_k$ . Ainsi nous régularisons le résultat d'une expansion gaussienne vis-à-vis de l'ensemble induit et nous pouvons espérer obtenir une image plus proche de  $G_k$ . Cette hypothèse se confirme dans la pratique.

Dans la suite, nous décrivons sommairement l'algorithme d'induction que nous avons élaboré pour réaliser le principe

précédent. Pour plus de détails, le lecteur est invité à consulter [3], ou de préférence [4] qui détail en profondeur l'algorithme présenté ci-après.

## 2.2 Algorithme d'induction

L'algorithme d'induction d'un niveau  $k - 1$  de la pyramide gaussienne se compose des étapes suivantes :

1. Initialiser  $\hat{G}_k$  avec le résultat d'une expansion gaussienne de  $G_{k-1}$ .

2.  $t \leftarrow 0$   
 $\Theta_{k-1}^{(t)} \leftarrow G_{k-1} - \Delta(\hat{G}_k)$

3. **Pour**  $j \leftarrow 0$  **jusqu'à**  $2^{k-1} - 1$   
**Pour**  $i \leftarrow 0$  **jusqu'à**  $2^{k-1} - 1$

$$\hat{G}_{k-1}(i, j) \leftarrow \sum_{m=-2}^2 \sum_{n=-2}^2 W(m, n) \cdot \hat{G}_k(2i + m, 2j + n)$$

$$\varepsilon_{k-1}(i, j) \leftarrow G_{k-1}(i, j) - \hat{G}_{k-1}(i, j)$$

**Pour**  $n \leftarrow -2$  **jusqu'à** 2

**Pour**  $m \leftarrow -2$  **jusqu'à** 2

$$\hat{G}_k(2i + m, 2j + n) \leftarrow \hat{G}_k(2i + m, 2j + n) + \frac{\varepsilon_{k-1}(i, j) \cdot W(m, n)}{\sum_{u=-2}^2 \sum_{v=-2}^2 W(u, v)^2}$$

4.  $t \leftarrow t + 1$   
 $\Theta_{k-1}^{(t)} \leftarrow G_{k-1} - \Delta(\hat{G}_k)$   
**Si**  $\Theta_{k-1}^{(t)} < \Theta_{k-1}^{(t-1)}$  **alors** recommencer l'étape 3.  
**Sinon** l'algorithme est terminé.

La deuxième étape consiste à calculer l'erreur globale entre  $G_{k-1}$  et  $\Delta(\hat{G}_k)$ . Ce calcul servira de critère d'arrêt la première fois où la quatrième étape sera exécutée.

Dans la troisième étape, il s'agit de corriger les éléments de  $\hat{G}_k$  en fonction d'une erreur locale ( $\varepsilon_{k-1}$ ) dans  $\Delta(\hat{G}_k)$ .

Enfin, la quatrième étape consiste en un test d'arrêt sur l'erreur globale entre  $G_{k-1}$  et  $\Delta(\hat{G}_k)$ . L'algorithme se termine lorsque cette erreur est supérieure ou égale à l'erreur calculée à l'itération précédente.

## 3 Résultats expérimentaux

Pour illustrer les résultats, nous avons utilisé l'image de *Lena* ( $256 \times 256$  pixels).

### 3.1 Comparaison entre expansion gaussienne et expansion inductive

Pour comparer les performances entre les expansions gaussienne et inductive, nous avons calculé le PSNR dans le cas de l'expansion des niveaux  $G_7$ ,  $G_6$  et  $G_5$  de la pyramide gaussienne (voir figure 3). Nous voyons que pour l'expansion gaussienne, les meilleurs résultats sont obtenus pour un noyau trimodal avec  $a$  variant entre 0.5 et 0.6. Alors que pour l'expansion inductive, les résultats sont nettement meilleurs pour un noyau unimodal avec  $a$  variant entre 0.3 et 0.35.

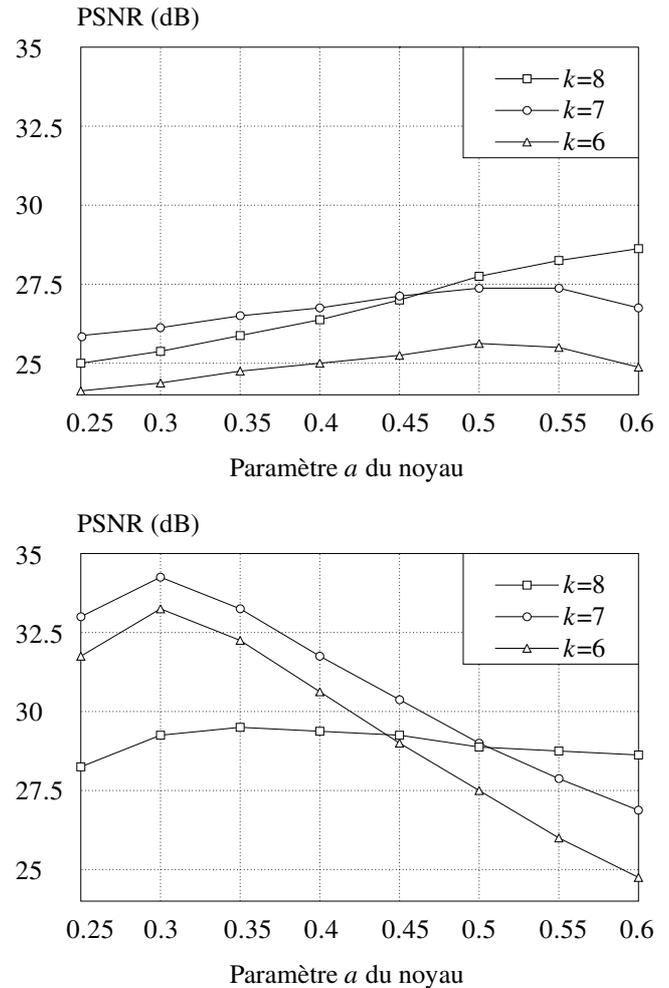


FIG. 3 — Evolution du rapport signal à bruit de crête (PSNR) en fonction du paramètre  $a$  du noyau gaussien dans le cas d'une expansion gaussienne (en haut) et dans le cas d'une expansion inductive (en bas). Nous illustrons ici les résultats obtenus à partir de l'image de *Lena* (version  $256 \times 256$  pixels) pour  $\hat{G}_8$ ,  $\hat{G}_7$  et  $\hat{G}_6$ .

### 3.2 Compression sans perte

La pyramide laplacienne générée en utilisant l'expansion inductive est rebaptisée "pyramide résiduelle" : le terme "laplacien" n'est plus justifié.

La compression atteinte est évaluée par un codage entropique des niveaux de la pyramide. Ainsi, nous obtenons une compression sans perte de 6.49 bits/pixel ( $a=0.6$ ) avec la pyramide laplacienne et 6.23 bits/pixel ( $a=0.35$ ) avec la pyramide résiduelle (voir la figure 4). Un codage entropique sur l'image originale de *Lena* donne 7.44 bits/pixel.

### 3.3 Quantification

La quantification des niveaux de la pyramide résiduelle est un problème très délicat. Une reconstruction de l'image initiale à partir de la pyramide résiduelle nécessite de nouvelles expansions inductives. La quantification perturbe les inductions et les erreurs se propagent de manière exponentielle du haut vers le bas de la pyramide. Ceci est dû au fait qu'une

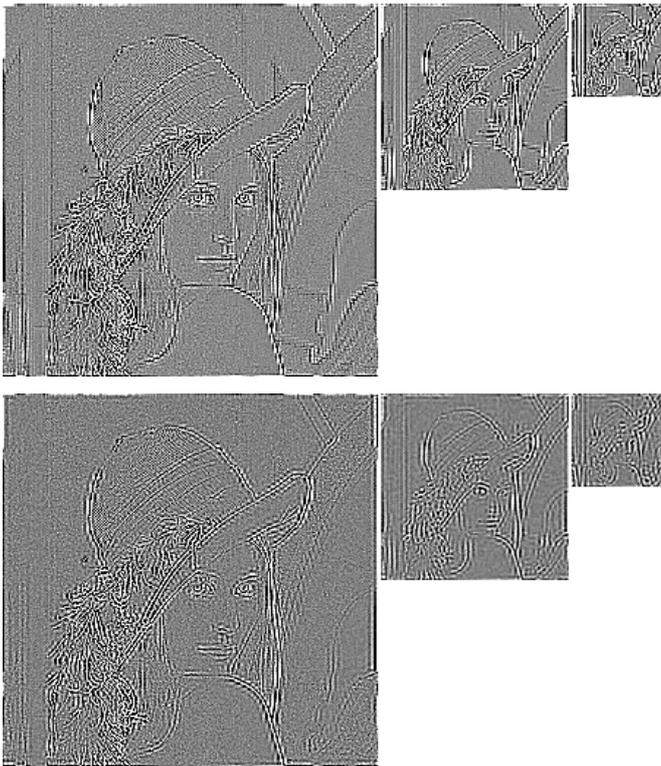


FIG. 4 — Pyramide laplacienne avec  $a=0.6$  (en haut) et pyramide résiduelle avec  $a=0.35$  (en bas). Les erreurs ont été multipliées par un facteur 10 pour permettre une comparaison visuelle.

quantification ne garantit plus une solution pour l'induction ( $\Gamma_k$  peut-être vide) et il y a risque de divergence de l'image induite vis-à-vis de l'image initiale.

Contrairement à la pyramide laplacienne, la pyramide résiduelle ne semble pas adaptée pour la compression d'images avec perte. Seul le niveau  $K$  de la pyramide résiduelle peut être quantifié puisque ce dernier n'est pas utilisé comme point de départ d'une induction.

## 4 Conclusion

Nous avons présenté un nouvel algorithme, pour l'expansion des couches de la pyramide gaussienne, procédant par induction plutôt que par interpolation. Ceci nous a conduit à définir un nouveau modèle de pyramide : la pyramide résiduelle. Nous avons montré sa supériorité à une pyramide laplacienne pour une compression d'images sans perte.

Le concept d'induction s'inscrit dans un domaine plus large qui est celui de la super-résolution : étude des relations inter-résolutions en fonction des connaissances. L'objectif à terme serait de fournir des images haute-résolution à partir d'images basse-résolution.

Le nouveau format d'image FlashPix [5], qui tend à devenir un standard dans l'industrie du traitement de l'image, est en partie basé sur une pyramide d'images. Ce format intègre une option de compression JPEG. Nous pensons que la pyramide résiduelle pourrait constituer une alternative à JPEG et

s'insérer à terme dans un tel format, en exploitant directement la structure pyramidale.

C'est sur la base de ces perspectives que nous continuons nos travaux.

## Références

- [1] P. J. Burt. "Fast filter transforms for image processing". *Computer Graphics and Image Processing*, 16 :20-51, 1981.
- [2] P. J. Burt and E. H. Adelson. "The Laplacian pyramid as a compact image code". *IEEE Transactions on Communications*, 31(4) :532-540, April 1983.
- [3] D. Calle et A. Montanvert. "Pyramide d'images résiduelle par expansion inductive". *10ième congrès Reconnaissance des Formes et Intelligence Artificielle (RFIA'96)*, pages 278-287, Rennes, Janvier 1996.
- [4] D. Calle et A. Montanvert. "Induction d'une image réduite". *Rapport de recherche IMAG*, 1997. A paraître.
- [5] Kodak. "FlashPix format and architecture white paper". <http://www.kodak.com>, June 1996.