Compression d'images avec les fractales

Alexandre Boucher & Pierre-Olivier Parisé

Concepts avancés en mathématiques et informatique appliquées
11 janvier 2016

Introduction

Remise en contexte :

- Les images : supports omniprésents dans notre quotidien.
- Nécessité d'algorithmes pour la compression.
- L'apparition des fractales en 1980.

Problématique :

- Algorithme JPEG dépend du spectre de Fourier et de la résolution de l'image.
- Comment les fractales peuvent-elles servir le domaine de la compression d'images ?

Sur les images

Définition d'une image

Une image :

- Composée de pixels ;
- Affectée de trois composantes
  - R : rouge ;
  - G : vert ;
  - B : bleu.

Notation :

$$\mathcal{I} := \{(x, y, R(x, y), G(x, y), B(x, y)) : 0 \leq x \leq l,\ 0 \leq y \leq h, 0 \leq R, G, B \leq 255\}$$

Sur les images

Les systèmes de couleurs

Autres représentations :

- *YUV* ;
- *YIQ* ;
- *HSV* .

Sur les images

Les systèmes de couleurs

Passage de *RGB* à *YUV*

$$\begin{pmatrix} Y \\ U \\ V \end{pmatrix} = \begin{pmatrix} 0,299 & 0,587 & 0,114 \\ -0,147 & -0,289 & 0,436 \\ 0,615 & -0,515 & -0,100 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}.$$

Passage de *RGB* à *YIQ*

$$\begin{pmatrix} Y \\ I \\ Q \end{pmatrix} = \begin{pmatrix} 0,299 & 0,587 & 0,114 \\ 0,596 & -0,274 & -0,322 \\ 0,211 & -0,523 & 0,312 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}.$$

Sur les IFS

Définition d'un IFS

IFS : Iterated function system.

Formé :

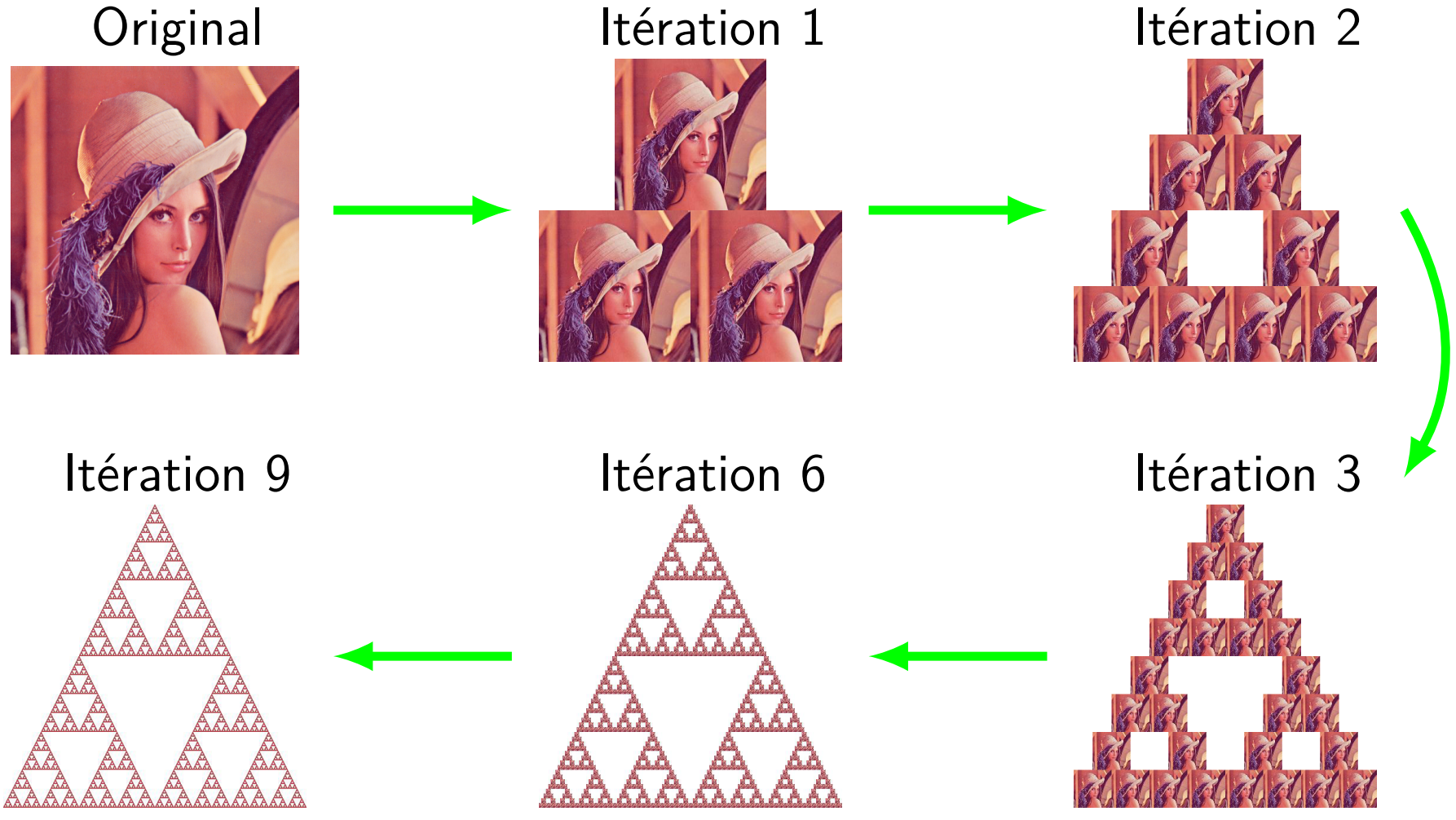
- de contractions w_i ;
- des facteurs de contractions s_i.

Notation :

$$\{\mathbb{R}^n; w_i, i = 1, 2, \ldots, m\}$$

$$W(I) = \bigcup_{i=1}^{m} w_i(I).$$

Exemple : le triangle de Sierpinski



Première étape

Une image en entrée de taille $2^k \times 2^k$.

Passer de *RGB* à *YIQ* :

$$\begin{pmatrix} Y \\ I \\ Q \end{pmatrix} = \begin{pmatrix} 0,299 & 0,587 & 0,114 \\ 0,596 & -0,274 & -0,322 \\ 0,211 & -0,523 & 0,312 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}.$$

Utilisation de *YIQ* :

- œil moins sensible à I et Q ;
- variance I et Q plus petite que Y ;
- réduction de la mémoire utilisée.

Deuxième étape

Partitionner l'image

- Construire les « range blocks » R_i.
- Construire les « domain blocks » D_i.

Les R_i :

- taille de $2^k \times 2^k$;
- sont disjoints.

Les D_k :

- deux fois la taille des R_i ;
- se chevauchent.

Deuxième étape

Les « range blocks »

R_i

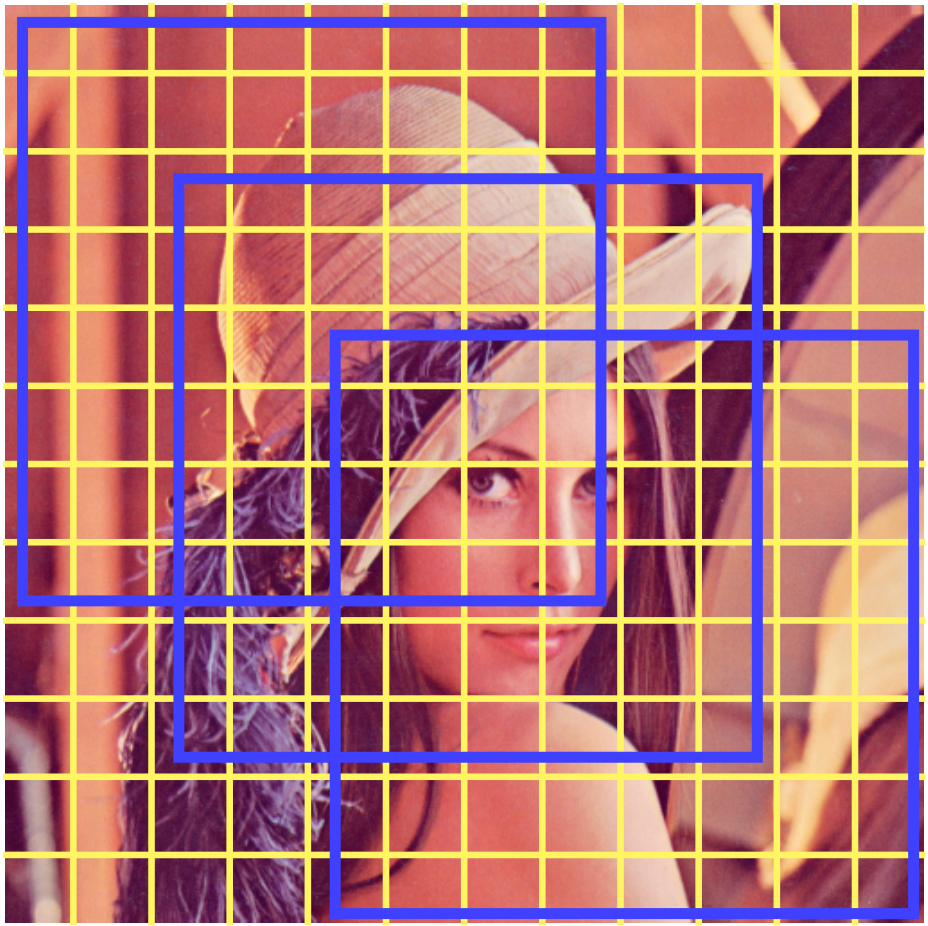
pixel

Deuxième étape

Les « domain blocks »

D_k

pixel



Troisième étape

Recherche du meilleur D_k :

1. Fixer un R_i.
2. Abaisser les dimensions des D_k à celles du R_i.
3. Calculer les coefficients s_u et o_u :

$$\min \left\{ \chi^2 := \sum_{j=1}^{n} (s_u \cdot a_j + o_u - b_j)^2 \right\} \qquad \text{où } u = Y,\ I \text{ ou } Q.$$

4. Garder les coefficients minimisant l'erreur χ^2.
5. Générer la transformation associée aux coefficients optimums.

Troisième étape

Les symétries

Point 4 : appliquer les 7 symétries et rotations aux D_k.

- Réflexion selon x ;
- Réflexion selon y ;
- Réflexion selon $y = x$;
- Réflexion selon $y = -x$;
- Rotation de $90°$;
- Rotation de $180°$;
- Rotation de $270°$.

Troisième étape

La transformation affine

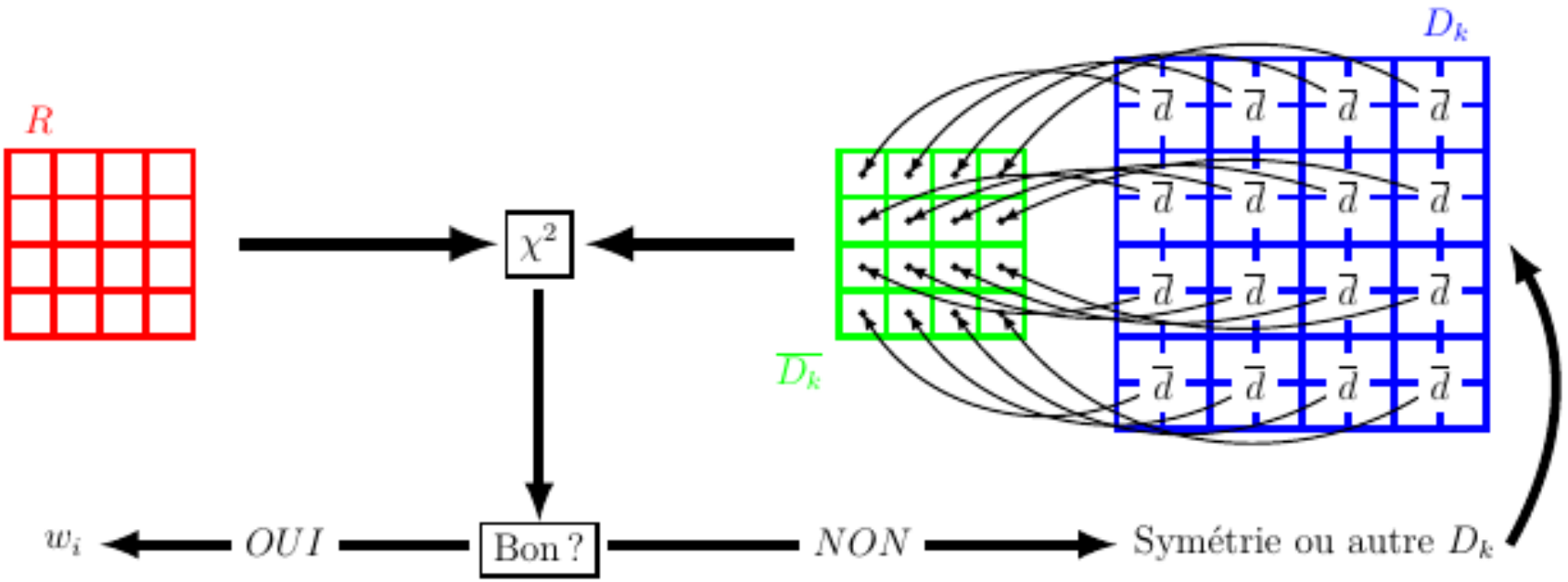
Point 5 : la transformation w_i sur chacune des composantes.

$$w(x, y, u) := \begin{pmatrix} \frac{1}{2} & 0 & 0 \\ 0 & \frac{1}{2} & 0 \\ 0 & 0 & s_u \end{pmatrix} \cdot \begin{pmatrix} D_x \\ D_y \\ u \end{pmatrix} + \begin{pmatrix} x_0 \\ y_0 \\ o_u \end{pmatrix}$$

où $u := Y(x, y)$, $I(x, y)$ ou $Q(x, y)$.

Troisième étape

Illustration des étapes



Quatrième étape

Enregistrer toutes les contractions : un minimum de bits.

- les s_u et o_u gardés ;
- la position du D_k ;
- la symétrie utilisée.

Cinquième étape

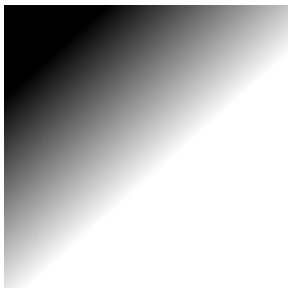
Décompression :

- Une image quelconque en entrée.
- Application des contractions à l'image.
- Possibilité de lissage.
- Retour à la représentation *RGB*.

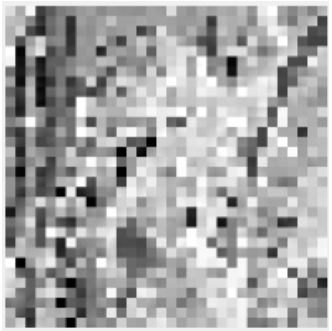
Pour une image en niveau de gris

Illustration du procédé

Image départ



Itération 1



Itération 2

Itération 3

Itération 4

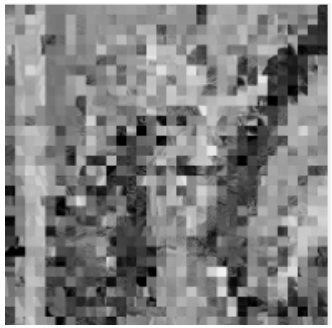
Itération 9

Pour une image en niveau de gris

Illustration de l'invariance de l'image de départ

Image départ

Itération 1



Itération 2

Itération 3

Itération 4

Itération 9

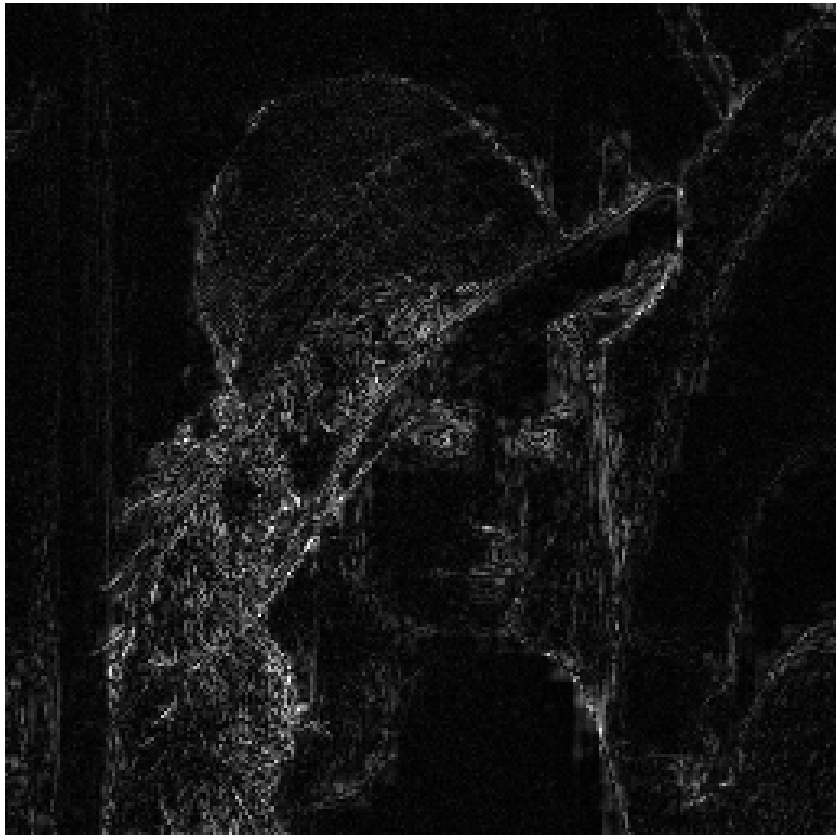
Pour des images en couleur

Image de Lenna 256 $\times$ 256 originale

Pour des images en couleur

Taille 256×256 et $R = 8$

(a) Adoucie

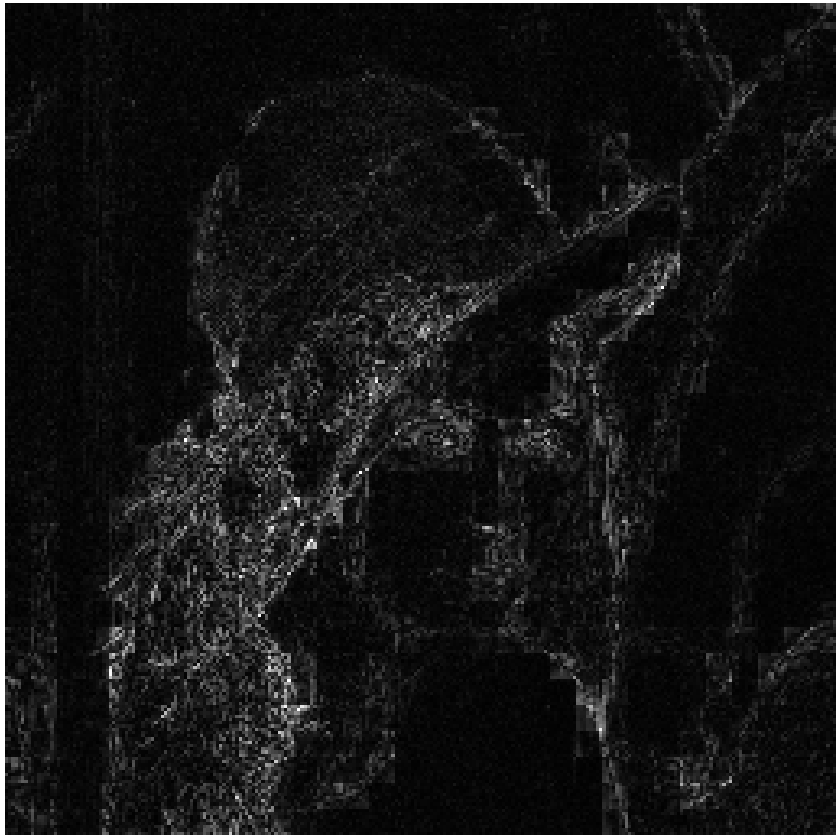


(b) Différence d'images

Pour des images en couleur

Taille 256×256 et $R = 8$

(a) Brut



(b) Différence d'images

Pour des images en couleur

Taille 256×256 et $R = 4$

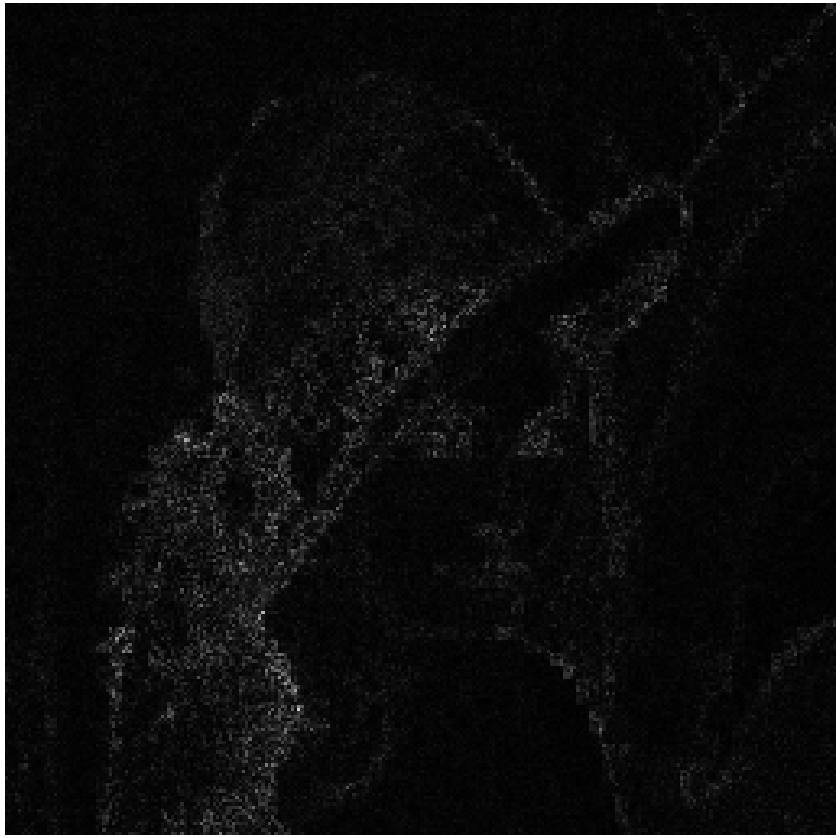
(a) Adoucie

(b) Différence d'images

Pour des images en couleur

Taille 256×256 et $R = 4$

(a) Brut



(b) Différence d'images

Pour des images en couleur

Taille 256×256 et $R = 2$

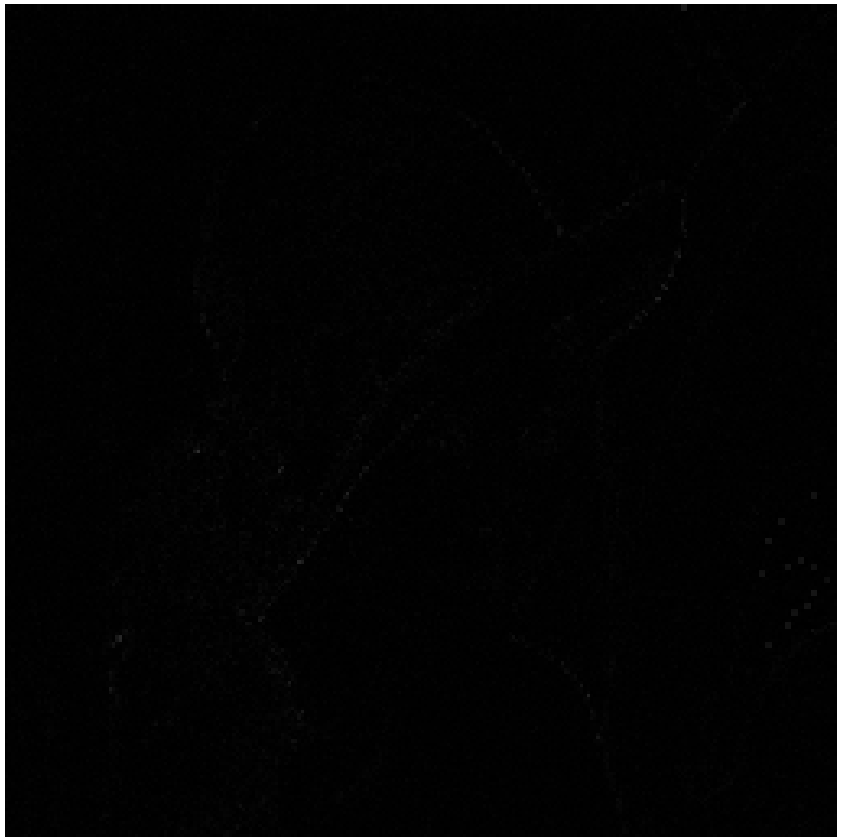
(a) Adoucie

(b) Différence d'images

Pour des images en couleur

Taille 256×256 et $R = 2$

(a) Brut



(b) Différence d'images

Pour des images en couleur

Aspects visuels des carrés des algorithmes

(a) Format JPEG

(b) Format FCCI

Pour des images en couleur

Taille des fichiers

Image	R	Taille FCCI	Taille JPEG	Taille PNG	Ratio
L-64x64	8	650	3255	9030	13.89 :1
L-64x64	4	2594	3255	9030	3.48 :1
L-64x64	2	10370	3255	9030	0.87 :1
L-128x128	8	2786	8994	33101	11.88 :1
L-128x128	4	11138	8994	33101	2.97 :1
L-128x128	2	44546	8994	33101	0.74 :1
L-256x256	8	11906	27212	125271	10.52 :1
L-256x256	4	47618	27212	125271	2.63 :1
L-256x256	2	190466	27212	125271	0.66 :1
P-1920x960	8	399602	548790	7385829	18.48 :1

Temps d'encodage et fidélité à l'image originale

Image	R	Encodage	Déc. adouci	Déc. brut	PSNR Adouci	PSNR Brut
L-64x64	8	272 ms	9	6	8.7679	8.59116
L-64x64	4	506 ms	12	6	12.86739	15.29051
L-64x64	2	1420 ms	21	6	12.37583	NaN
L-128x128	8	5.67 s	39	24	12.64667	12.73227
L-128x128	4	9.61 s	55	24	15.40411	18.32838
L-128x128	2	24.87 s	81	23	14.37288	NaN
L-256x256	8	2m 17.2s	245	133	14.86215	15.2005
L-256x256	4	3m 17.62s	277	139	17.46362	20.96746
L-256x256	2	7m 18.73s	425	135	16.34114	NaN
P-1920x960	8	18h 43m 28.59s	7440	5027	15.50346	15.48109

Conclusion

Méthode pertinente :

- Taux de compression très bon, spécialement avec des méthodes plus avancées.
- Temps de décompression très rapide.
- Par contre, le temps de compression est un facteur négatif.

Exploration de techniques plus avancées :

- *Quadtree*
- *HV Partitioning*

Pourrait être utilisé à l'avenir comme alternative à JPEG

- Pour diminuer l'effet de pixellisation.