

# Comresseur universel d'images 4K

BERNARDON Vincent, BIREMBAUT Mateusz



UNIVERSITÉ DE  
MONTPELLIER



# Sommaire

*01*

**Contexte**

*02*

**État de l'art et  
Méthodes proposées**

*03*

**Comparaison et  
analyse**

*04*

**Démonstration  
et conclusion**

01

# Contexte

# Contexte

Besoin de compressions qui :

- Fonctionnent sur tout type d'image
- Plutôt rapide
- Bon taux de compression
- PSNR > 30



02

## État de l'art et Méthodes proposées

# État de l'art

---

# Transformation en Cosinus Discrète

Avantages :

- Standard universel,
- Performant pour des images “naturelles”

Désavantages :

- Artéfacts en blocs pour des taux de compression élevés

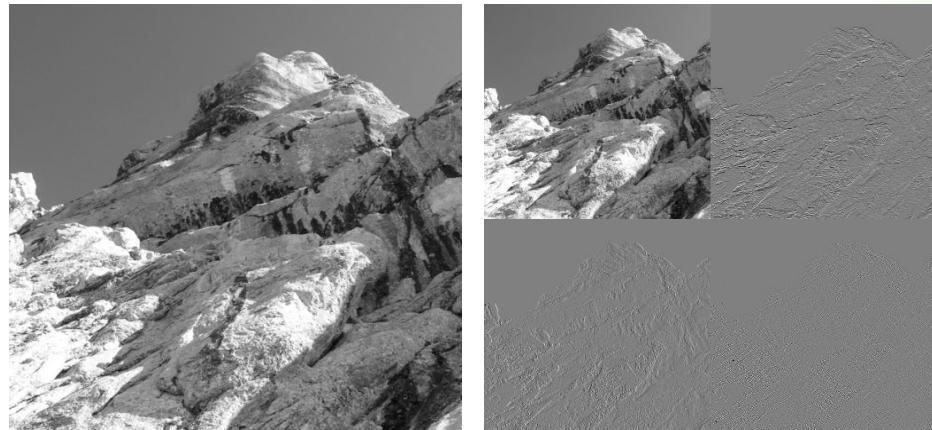


Three levels of JPG compression. The left-most image is the original. The middle image offers a medium compression, which may not be immediately obvious to the naked eye without closer inspection. The right-most image is maximally compressed.

# Compression par Ondelettes

Avantages :

- Meilleure qualité d'image pour des taux de compression élevés



Désavantages :

- Plus complexe en terme d'implémentation / calculs

# Quantification Vectorielle

Avantages :

- Décodeur simple
- Pas de quantification de coefficients

Désavantages :

- Long à encoder,
- moins adapté aux faibles taux de bits



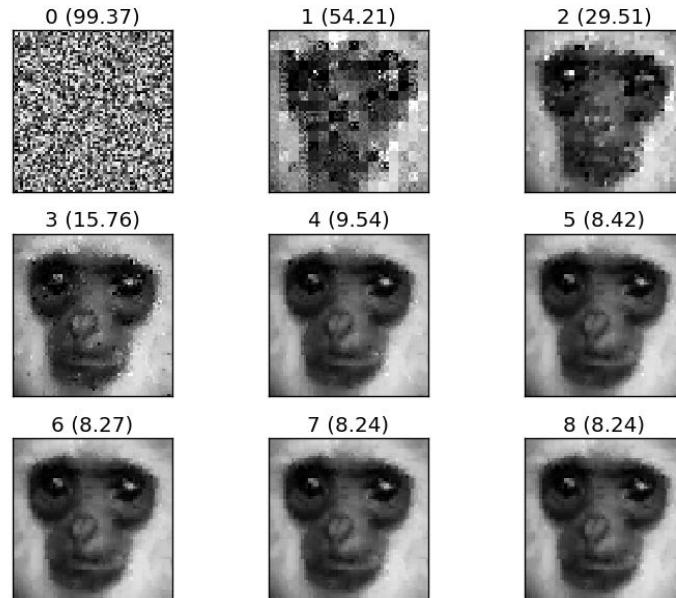
# Compression Fractale

Avantages :

- Potentiellement, des taux de compression élevés

Désavantages :

- L'encodage extrêmement lent
- Qualité moins bonne comparée aux autres



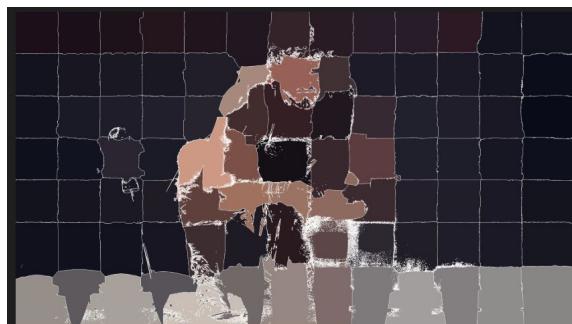
# 1<sup>re</sup> Méthode proposée : SLIC

---

# SLIC : Simple Linear Iterative Clustering

2 paramètres :

- $k$  : Le nombre de superpixels
- $m$  : La compacité des superpixels



$k=100 ; m=250$



$k=100 ; m=20$

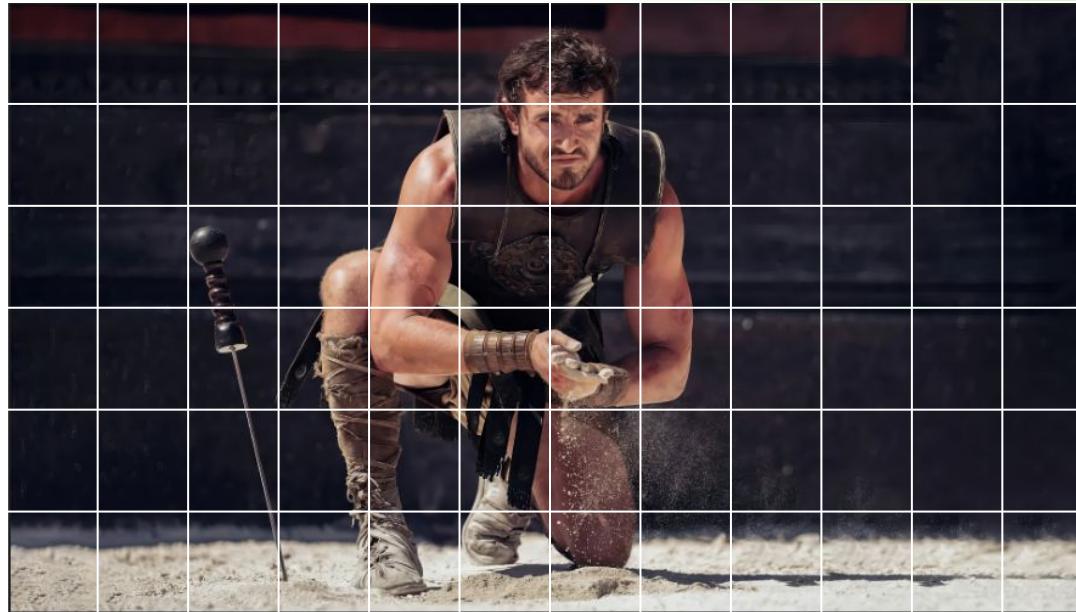


$k=10\ 000 ; m=250$

# SLIC : Simple Linear Iterative Clustering



# SLIC : Simple Linear Iterative Clustering

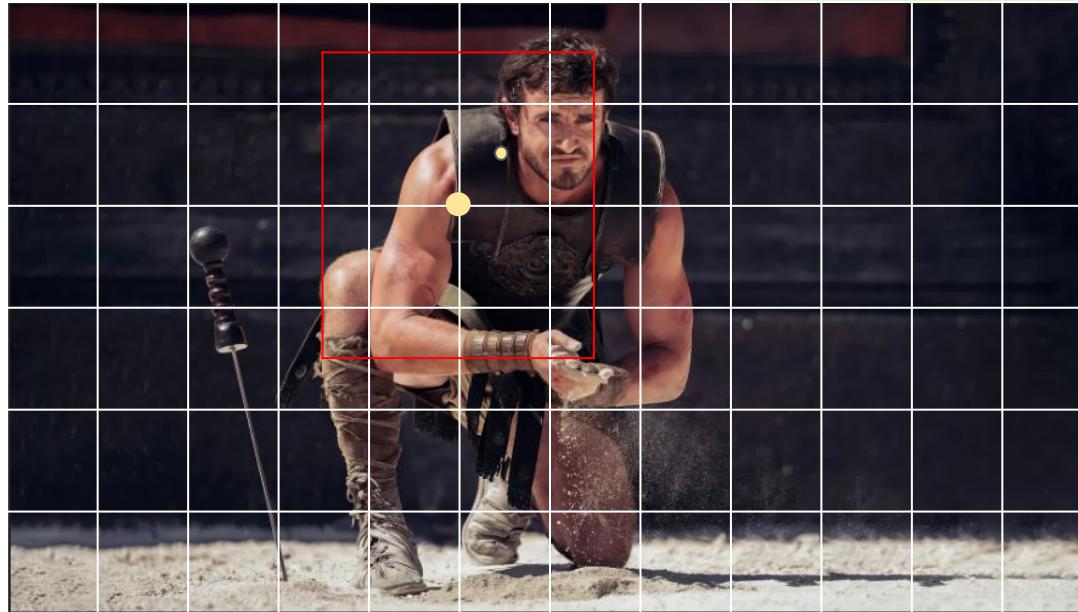


# SLIC : Simple Linear Iterative Clustering



Distance = distance couleur + distance dans l'espace

# SLIC : Simple Linear Iterative Clustering



# SLIC : Simple Linear Iterative Clustering



# SLIC : Simple Linear Iterative Clustering



# SLIC : Simple Linear Iterative Clustering



# 2<sup>e</sup> Méthode proposée : SDGT

---

# SDGT : Superpixel-Driven Graph Transform

4 paramètres :

- k : Le nombre de superpixels
- m : La compacité des superpixels
- **k' : Le nombre de k après réduction**
- **c : Coefficient de quantification**

**Fracastoro et al., ICIP 2017 – "Superpixel-Driven Graph Transform for Image Compression"**

# SDGT : Superpixel-Driven Graph Transform

Étape 1 :

- Segmentation de l'image avec SLIC (sans modifier la couleur des pixels)



# SDGT : Superpixel-Driven Graph Transform

## Étape 2 : (optionnel)

- Passe les superpixels de l'espace RGB à CIELAB (Standard CIEDE 2000)
- Fusionne les 2 pixels voisins ayant la plus petite distance dans l'espace CIELAB. (boucle jusqu'à que  $k = k'$ )

# SDGT : Superpixel-Driven Graph Transform

Étape 3 : Pour chaque superpixels :

- On récupère les pixels associés
- On calcule la matrice d'Adjacence A :

<b>0</b>	<b>1</b>
<b>2</b>	<b>3</b>



	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>
<b>0</b>	0	1	1	0
<b>1</b>	1	0	0	1
<b>2</b>	1	0	0	1
<b>3</b>	0	1	1	0

# SDGT : Superpixel-Driven Graph Transform

## Étape 4 :

- À partir de la matrice  $A$ , on calcule la matrice Laplacienne :  $L$
- On récupère la matrice des vecteurs propres  $U$  à partir de  $L$ .
- On applique une transformée de Fourier notre graphe grâce à  $U$ .

# SDGT : Superpixel-Driven Graph Transform

Étape 4 : Visuellement

$$\begin{array}{c} - \\ \begin{array}{|c|c|c|c|c|} \hline & \mathbf{0} & \mathbf{1} & \mathbf{2} & \mathbf{3} \\ \hline \mathbf{0} & 0 & 1 & 1 & 0 \\ \hline \mathbf{1} & 1 & 0 & 0 & 1 \\ \hline \mathbf{2} & 1 & 0 & 0 & 1 \\ \hline \mathbf{3} & 0 & 1 & 1 & 0 \\ \hline \end{array} \end{array} + \begin{array}{|c|c|c|c|c|} \hline & \mathbf{0} & \mathbf{1} & \mathbf{2} & \mathbf{3} \\ \hline \mathbf{0} & 2 & 0 & 0 & 0 \\ \hline \mathbf{1} & 0 & 2 & 0 & 0 \\ \hline \mathbf{2} & 0 & 0 & 2 & 0 \\ \hline \mathbf{3} & 0 & 0 & 0 & 2 \\ \hline \end{array} = \begin{array}{|c|c|c|c|c|} \hline & \mathbf{0} & \mathbf{1} & \mathbf{2} & \mathbf{3} \\ \hline \mathbf{0} & 2 & -1 & -1 & 0 \\ \hline \mathbf{1} & -1 & 2 & 0 & -1 \\ \hline \mathbf{2} & -1 & 0 & 2 & -1 \\ \hline \mathbf{3} & 0 & -1 & -1 & 2 \\ \hline \end{array}$$

-A                    +                    D                    =                    L

# SDGT : Superpixel-Driven Graph Transform

## Étape 5 :

- Transformé de Fourier
- Quantification similaire à une DCT
- Transformé inverse

# SDGT : Superpixel-Driven Graph Transform

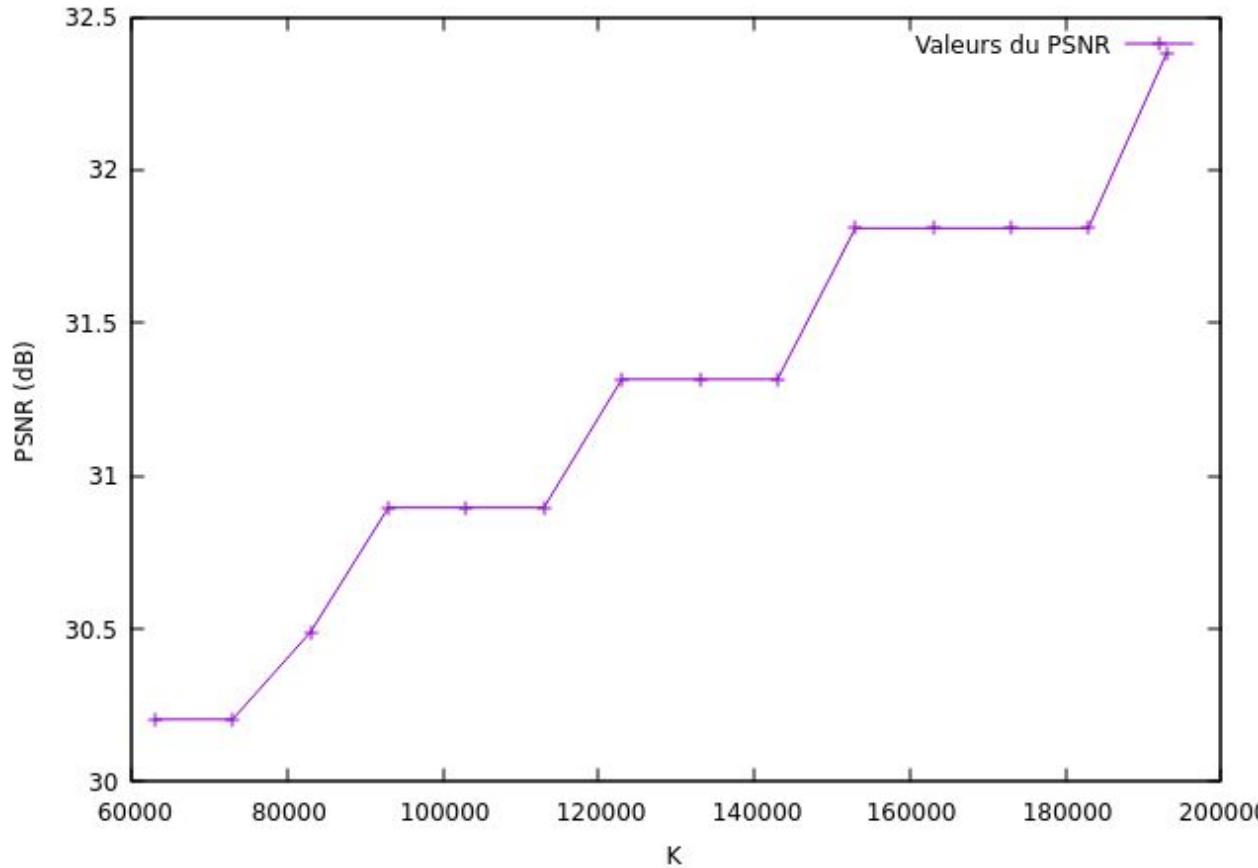


Facteur de quantification : 40 et 100 000 superpixels

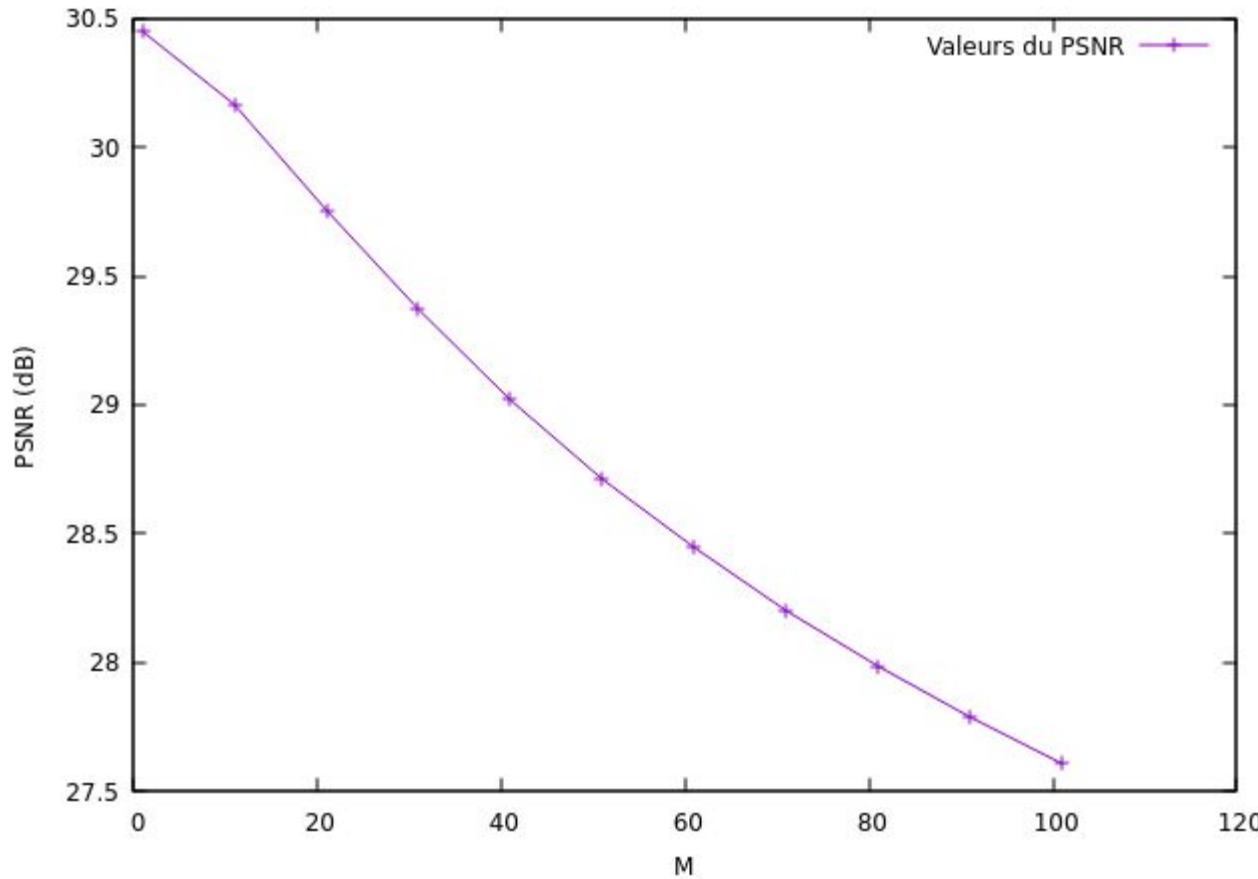
03

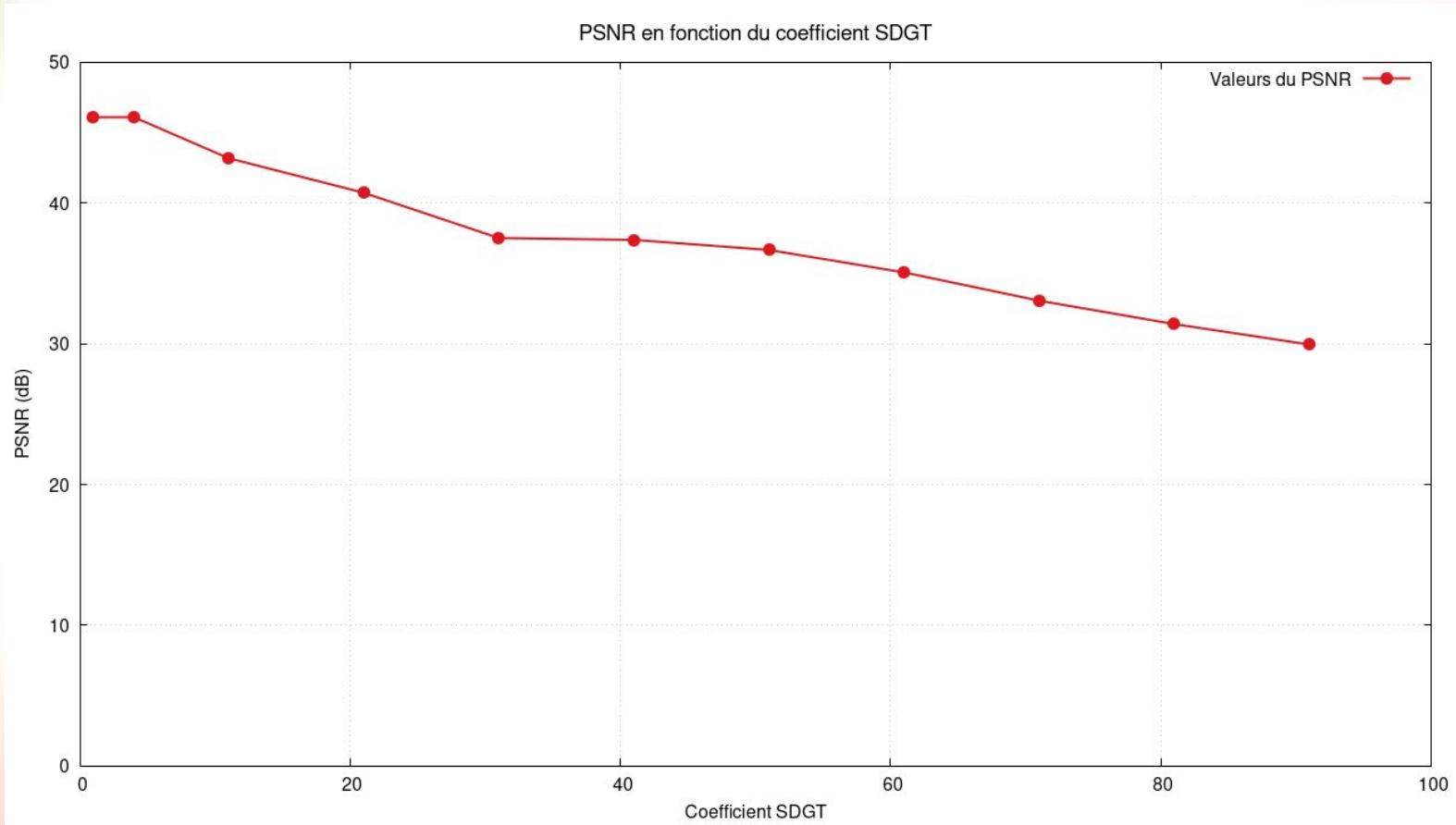
# Comparaison et analyse

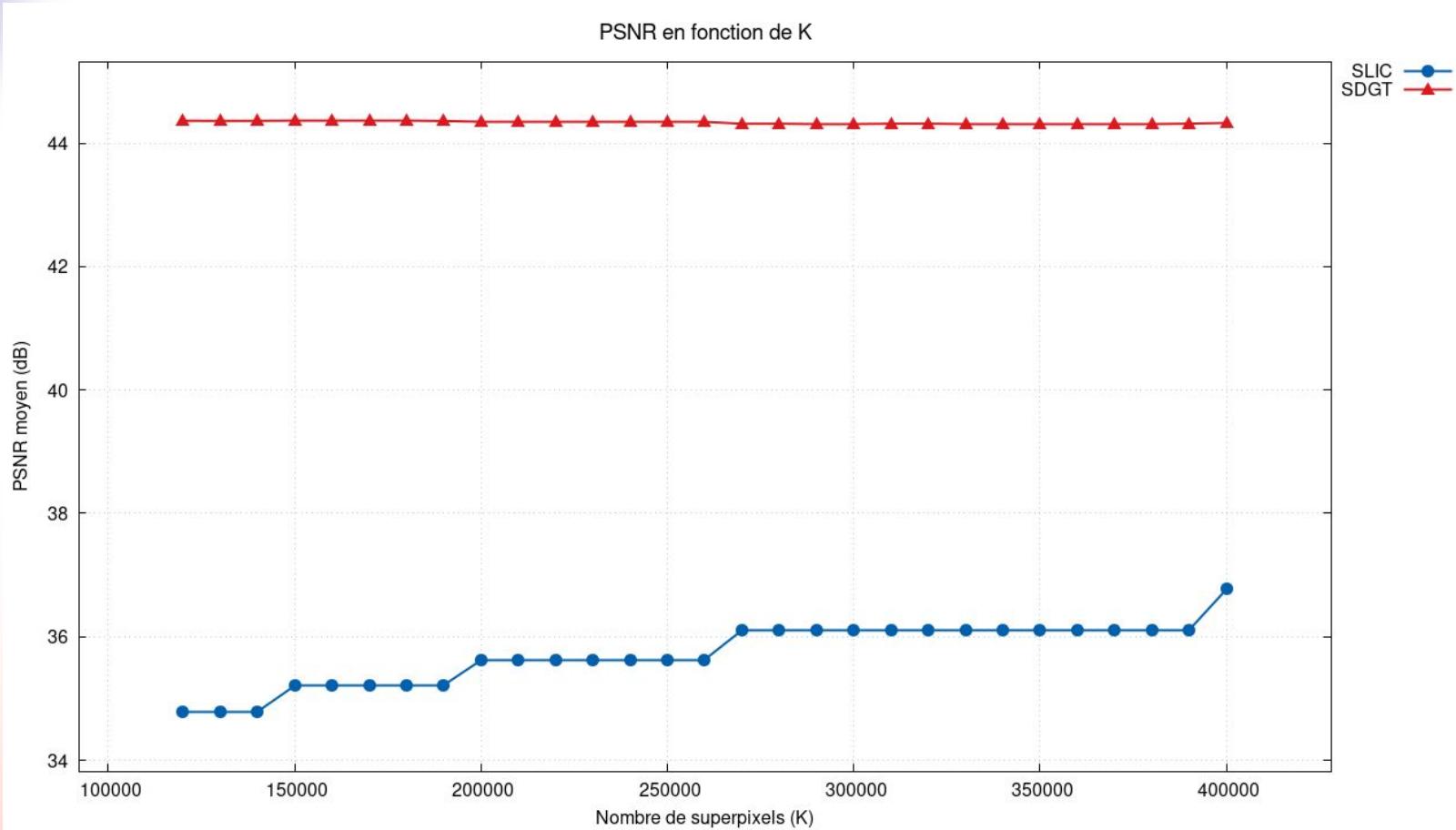
PSNR en fonction de K



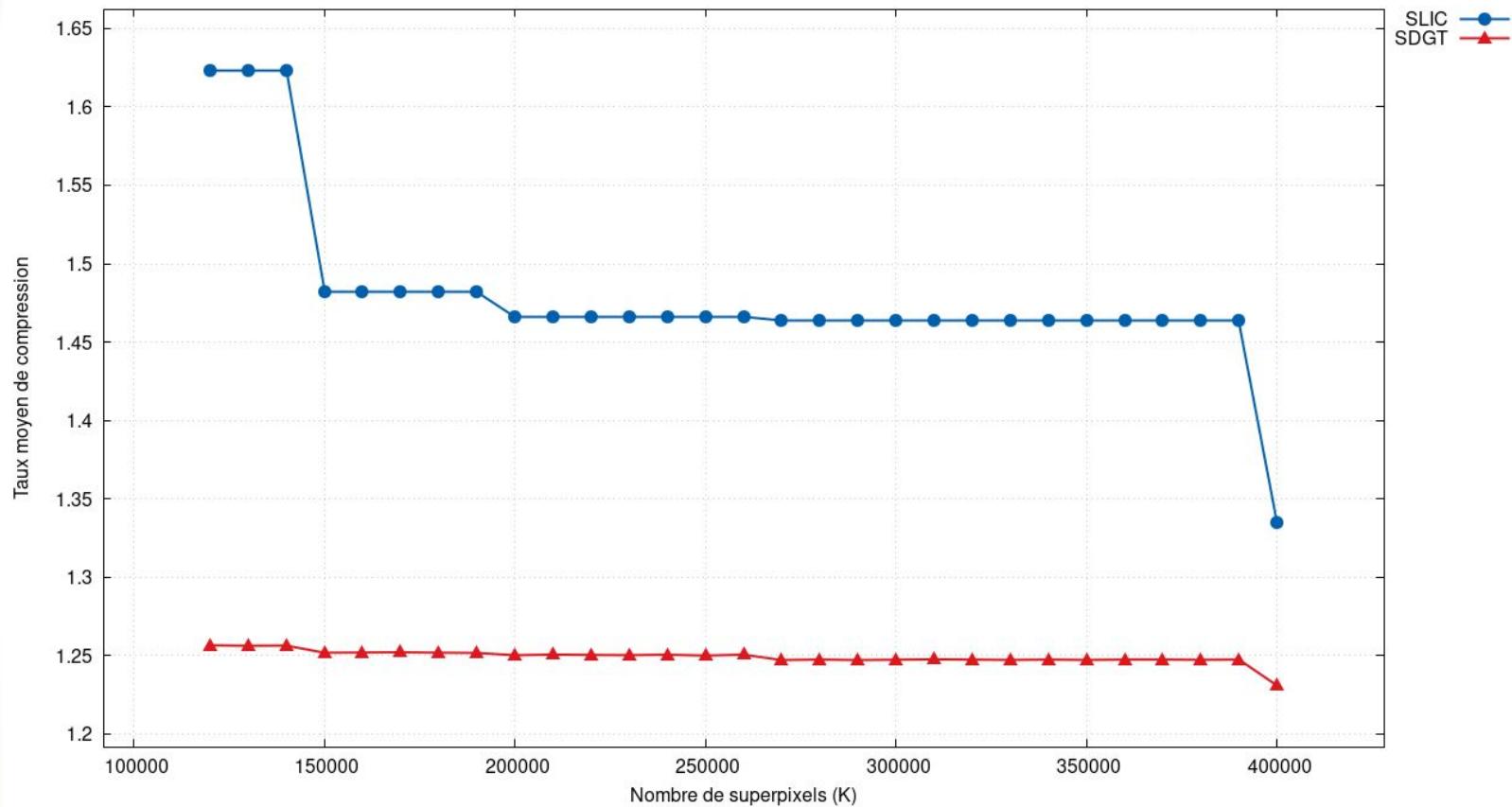
PSNR en fonction de M



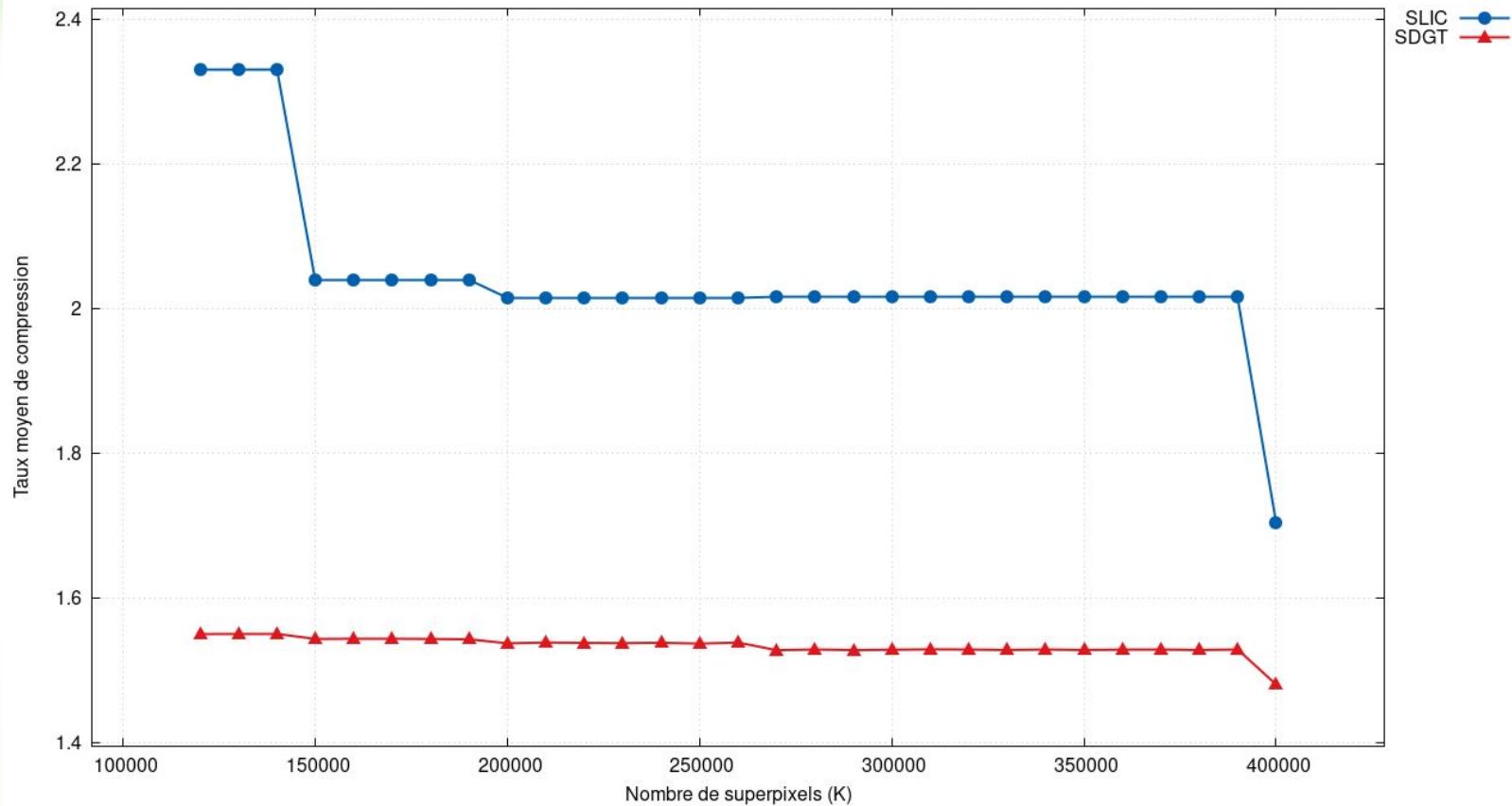




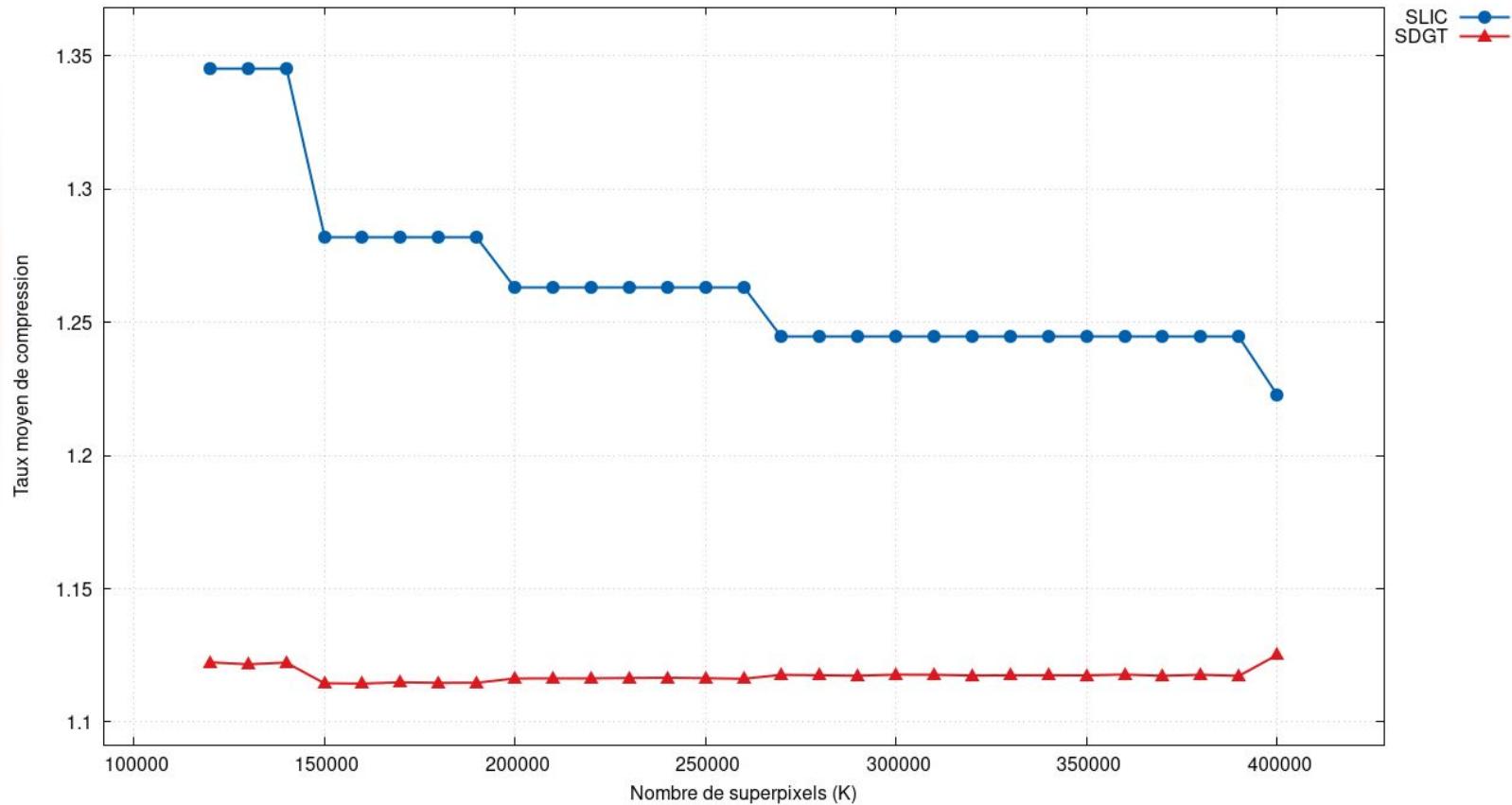
Taux de compression en fonction de K



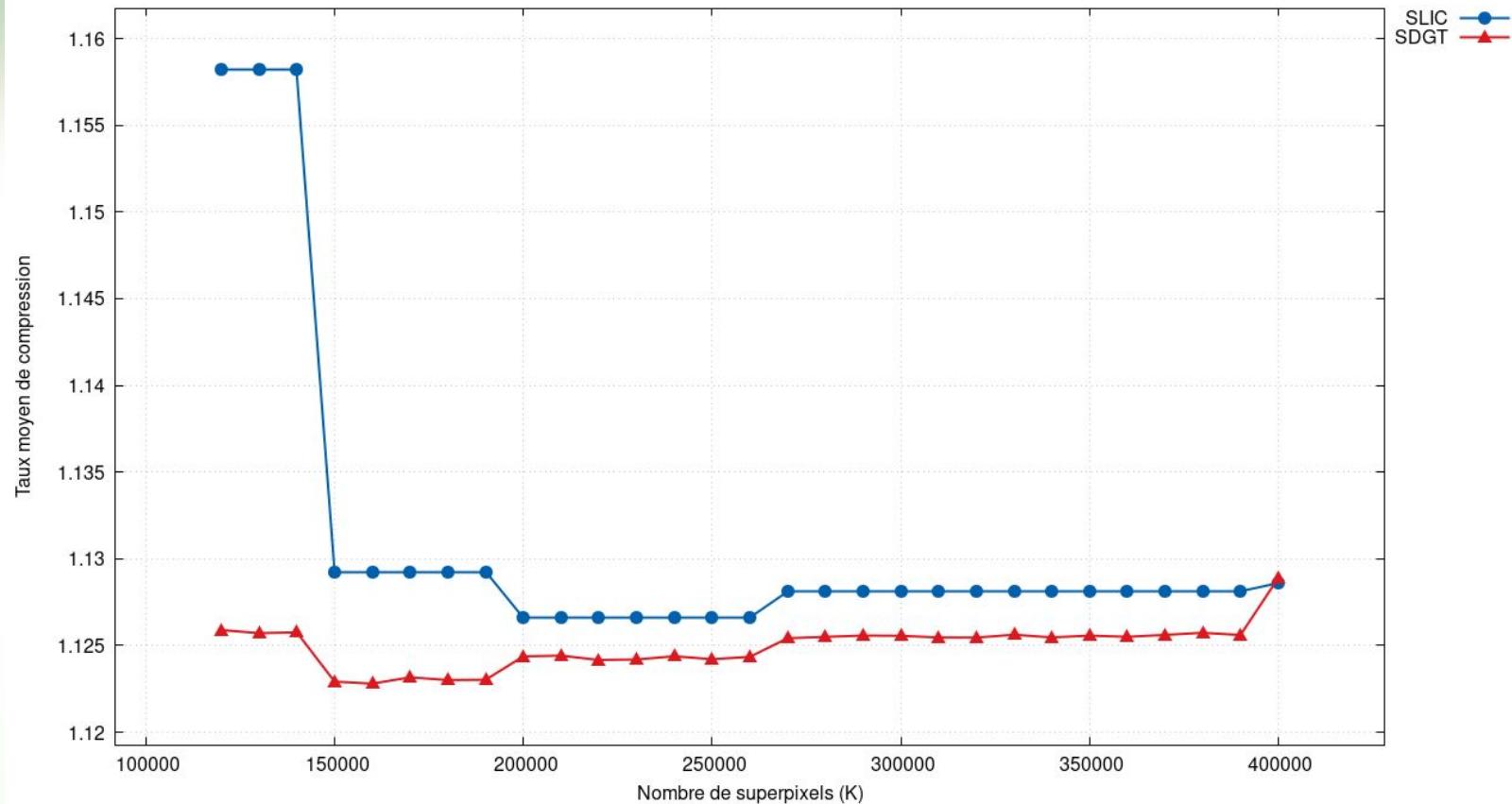
Taux de compression en fonction de K - dessin



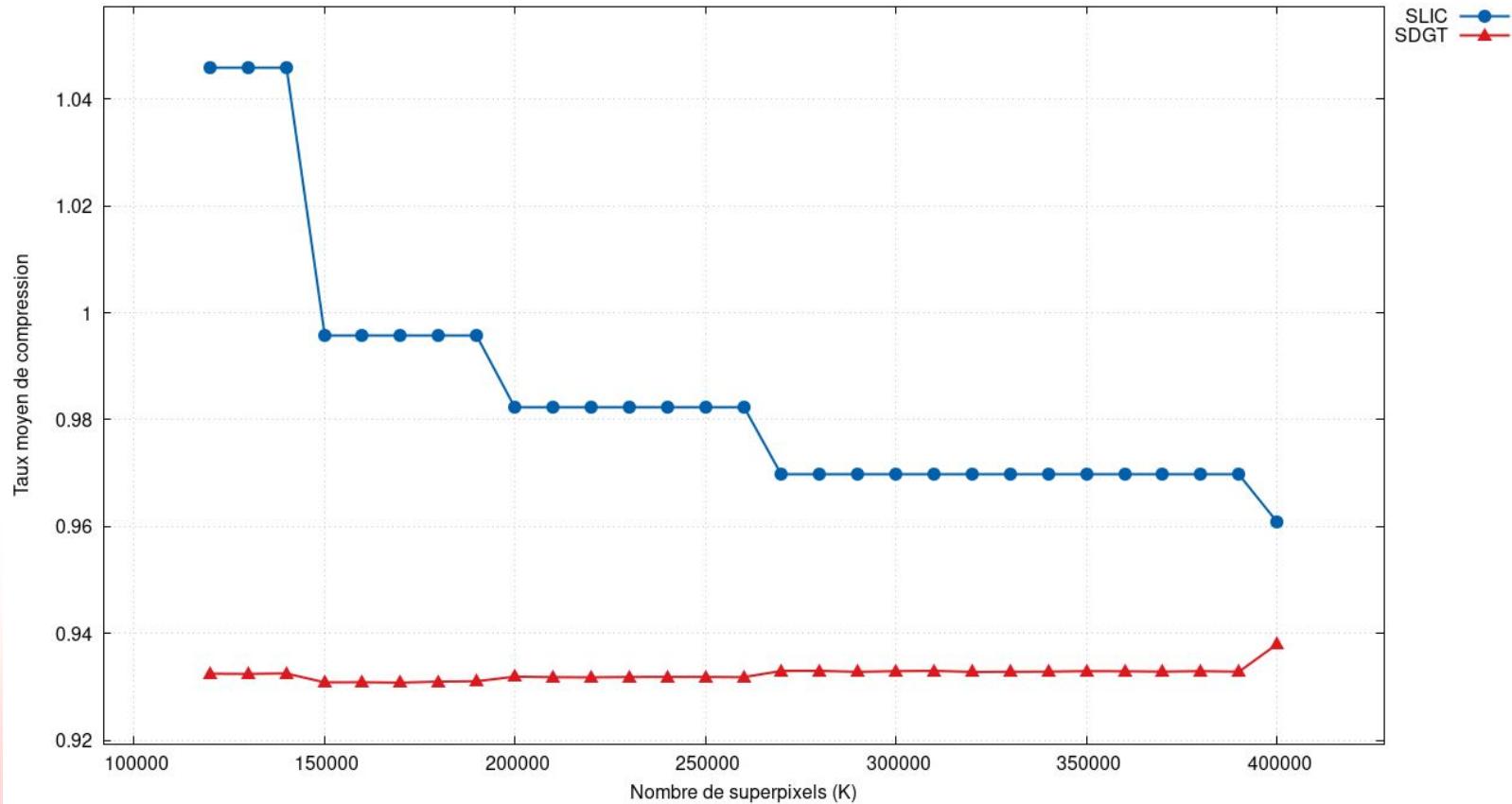
Taux de compression en fonction de K - images-numériques



Taux de compression en fonction de K - film



Taux de compression en fonction de K - photos



04

# Démonstration et conclusion



**Merci de votre écoute !**

# Bibliographie

[1] « Review Paper on Image Compression ». Disponible sur :

<https://ceng460.cankaya.edu.tr/uploads/files/file/Review%20paper.pdf>

[2] « SLIC Superpixels ». Disponible sur :

[https://www.researchgate.net/publication/44234783\\_SLIC\\_superpixels](https://www.researchgate.net/publication/44234783_SLIC_superpixels)

[3] « Superpixel-driven Graph Transform for Image Compression ». Disponible sur :

[https://www.researchgate.net/publication/282290290\\_Superpixel-driven\\_Graph\\_Transform\\_for\\_Image\\_Compression](https://www.researchgate.net/publication/282290290_Superpixel-driven_Graph_Transform_for_Image_Compression)

[4] UHD Paper, « Free Ultra HD Wallpapers ». Disponible sur : <https://www.uhdpaper.com/>