

Subspace Iteration Methods

Application to image compression

Ayoub Bouchama
Oussama ElGuerraoui

Le 19 avril 2023

Contents

1	Introduction	2
2	Question-1	2
3	Question-2	3
3.1	Reconstruction avec subspace_iter_v3	3
3.2	Reconstruction avec subspace_iter_v2	8
4	Reconstruction de l'image colorée	16
5	Conclusion	18

List of Figures

1	Images reconstruites par la méthode subspace_iter_v3	3
2	Différences entre image réelle et image reconstruite	4
3	Images reconstruites par la méthode subspace_iter_v3	5
4	Différences entre image réelle et image reconstruite	6
5	Images reconstruites par la méthode subspace_iter_v2	8
6	Différences entre image réelle et image reconstruite	9
7	Images reconstruites par la méthode subspace_iter_v2	10
8	Différences entre image réelle et image reconstruite	11
9	Images reconstruites par la méthode subspace_iter_v1	13
10	Différences entre image réelle et image reconstruite	14
11	Images colorées reconstruites	17

1 Introduction

Nous allons d'étudier les méthodes itératives de sous-espace pour la compression d'images. Nous allons appliquer la méthode itérative de sous-espace pour obtenir la décomposition en valeurs singulières (SVD) de l'image et ainsi réaliser la meilleure approximation de rang faible. Nous étudierons également les différences de performances entre les différentes fonctions qui calculent les paires d'éléments propres des matrices. Enfin, nous aborderons brièvement la question de la compression des images en couleur.

2 Question-1

$\Sigma_k \in R^{k \times k}$: contient les k valeurs singulières de l'image triées par ordre décroissant sur sa diagonale.

$U_k \in R^{q \times k}$: contient les vecteurs singuliers à gauche de l'image correspondant aux k valeurs singulières les plus grandes.

$V_k \in R^{p \times k}$: contient les vecteurs singuliers à droite de l'image correspondant aux k valeurs singulières les plus grandes.

3 Question-2

3.1 Reconstruction avec subspace_iter_v3

SIMULATION-1: SUBSPACE V3

Paramètres :

$$\text{tolerance} = 10^{-8}$$

$$m = 400$$

$$\text{percentage} = 0.995$$

$$\text{puiss} = 1$$

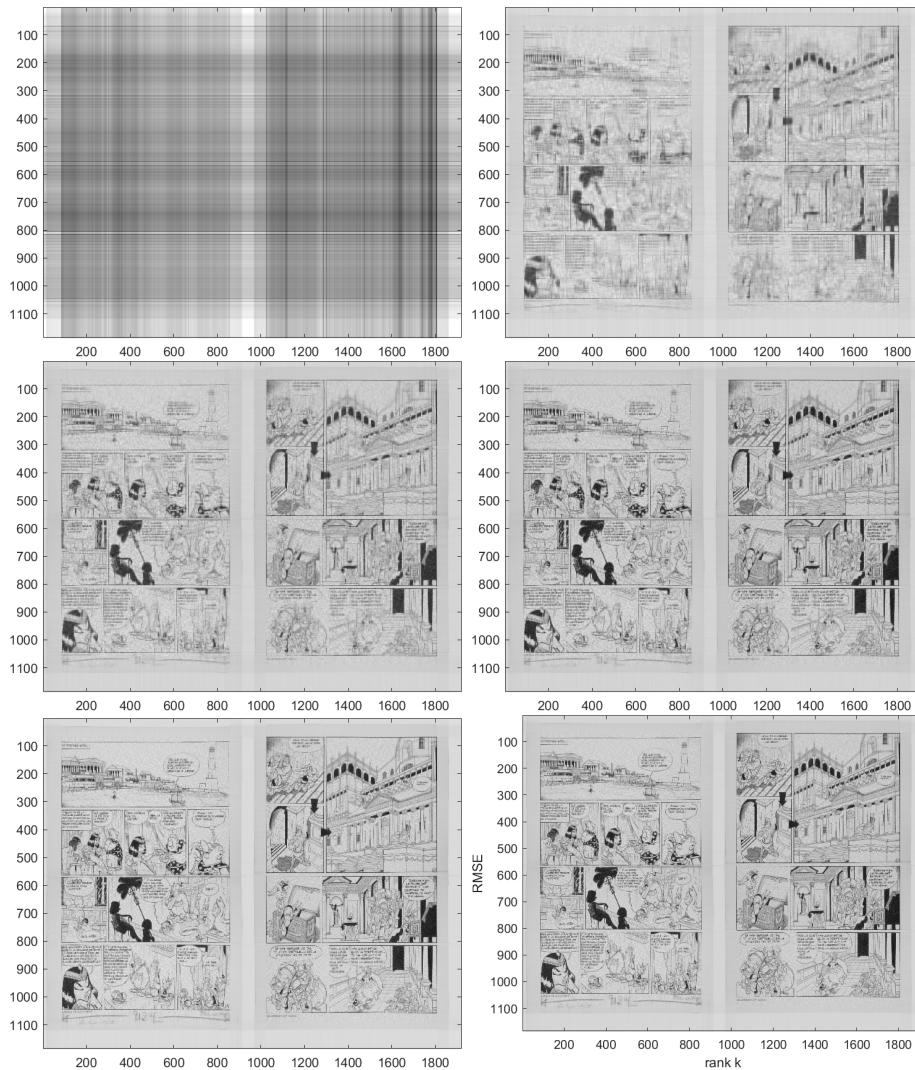


Figure 1: Images reconstruites par la méthode subspace_iter_v3

Nous remarquons que l'image a été reconstruite correctement, pourtant l'image finale est toujours un peu flou.

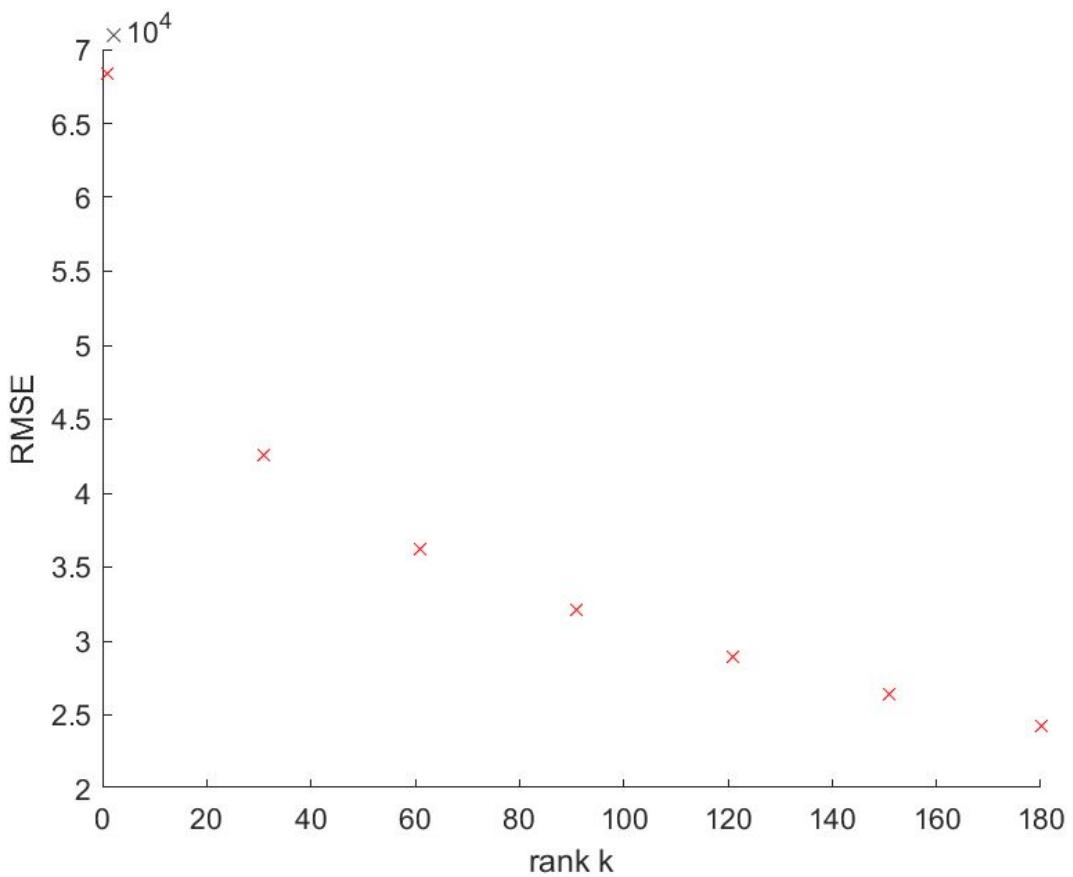


Figure 2: Différences entre image réelle et image reconstruite

D'après la figure tracée représentant la différence de SVD, il apparaît que l'augmentation du rang de la matrice de l'image reconstruite à l'aide de la SVD est corrélée à une diminution de la RMSE entre l'image originale et l'image reconstruite. Autrement dit, une reconstruction d'image plus précise peut être obtenue en utilisant une matrice de rang plus élevé. Pour augmenter le rang de la matrice de l'image reconstruite, il faut utiliser davantage de valeurs singulières. Cependant, cette méthode peut entraîner une augmentation du coût de calcul et de stockage. Ainsi, il faut trouver un juste équilibre entre la précision de la reconstruction et les ressources nécessaires pour la réaliser. En somme, la figure montre que l'augmentation du rang de la matrice de l'image reconstruite à l'aide de la SVD peut améliorer la précision de la reconstruction d'image, mais cela nécessite une utilisation accrue des ressources de calcul et de stockage.

SIMULATION-2: SUBSPACE V3

Paramètres :

$$\text{tolerance} = 10^{-8}$$

$$m = 400$$

$$\text{percentage} = 0.995$$

$$\text{puiss} = 2$$

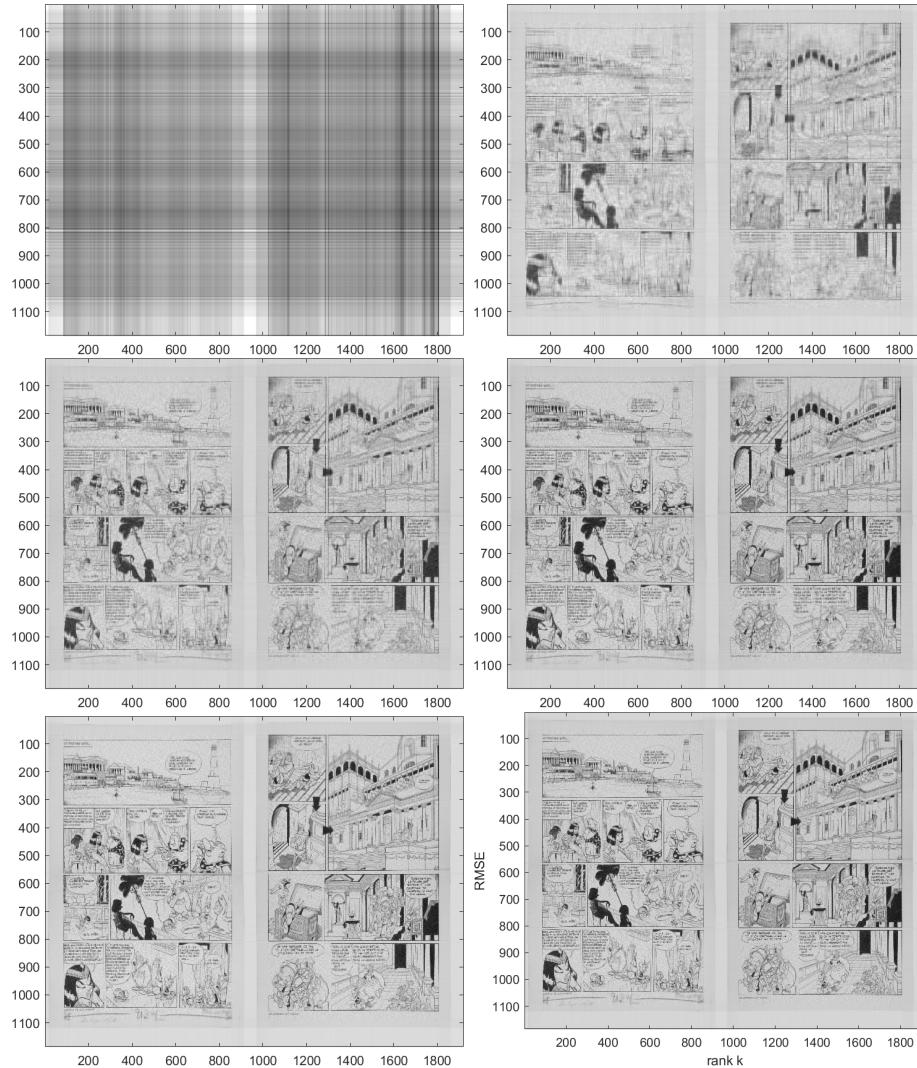


Figure 3: Images reconstruites par la méthode subspace_iter_v3

Nous remarquons que l'image a été reconstruite correctement, pourtant l'image finale est toujours un peu flou.

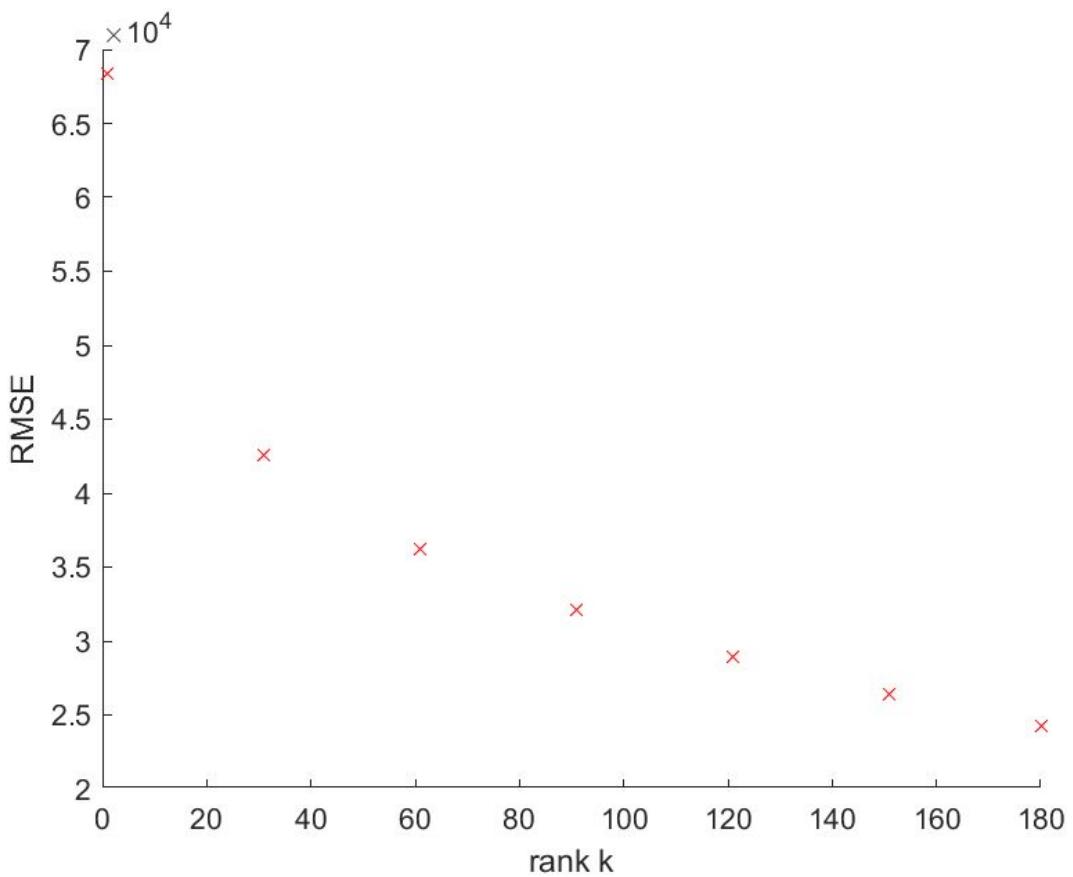


Figure 4: Différences entre image réelle et image reconstruite

Nous remarquons les différences de SVD entre l'image originale et l'image reconstruite lors'on change le paramètre p qui modélise le nombre de produit $Y = A^p \cdot V$, donc ce paramètre n'a pas d'effet sur la qualité de l'image et sa précision mais plutot impacte la rapidité des calculs des couples propres.

SIMULATION-3: SUBSPACE V3

Paramètres :

tolerance = 10^{-8}

m = 400

percentage = 0.995

puiss = 3

Le processus de reconstruction de l'image peut prendre beaucoup de temps en raison de la taille de l'image ainsi que de la complexité du calcul de I^3 . En conséquence, notre programme semble ne pas répondre pendant un certain temps, voire ne pas être capable de reconstruire l'image du tout. Cela est dû au fait que le traitement de données complexes nécessite plus de temps et de ressources pour être effectué de manière efficace. Il est possible que l'utilisation d'un matériel informatique plus performant ou d'une méthode de traitement plus avancée puisse améliorer les performances du programme de reconstruction d'image ce qui n'est pas à notre disposition.

3.2 Reconstruction avec subspace_iter_v2

SIMULATION-1: SUBSPACE V2

Paramètres :

$$\text{tolerance} = 10^{-8}$$

$$m = 400$$

$$\text{percentage} = 0.995$$

$$\text{puiss} = 1$$

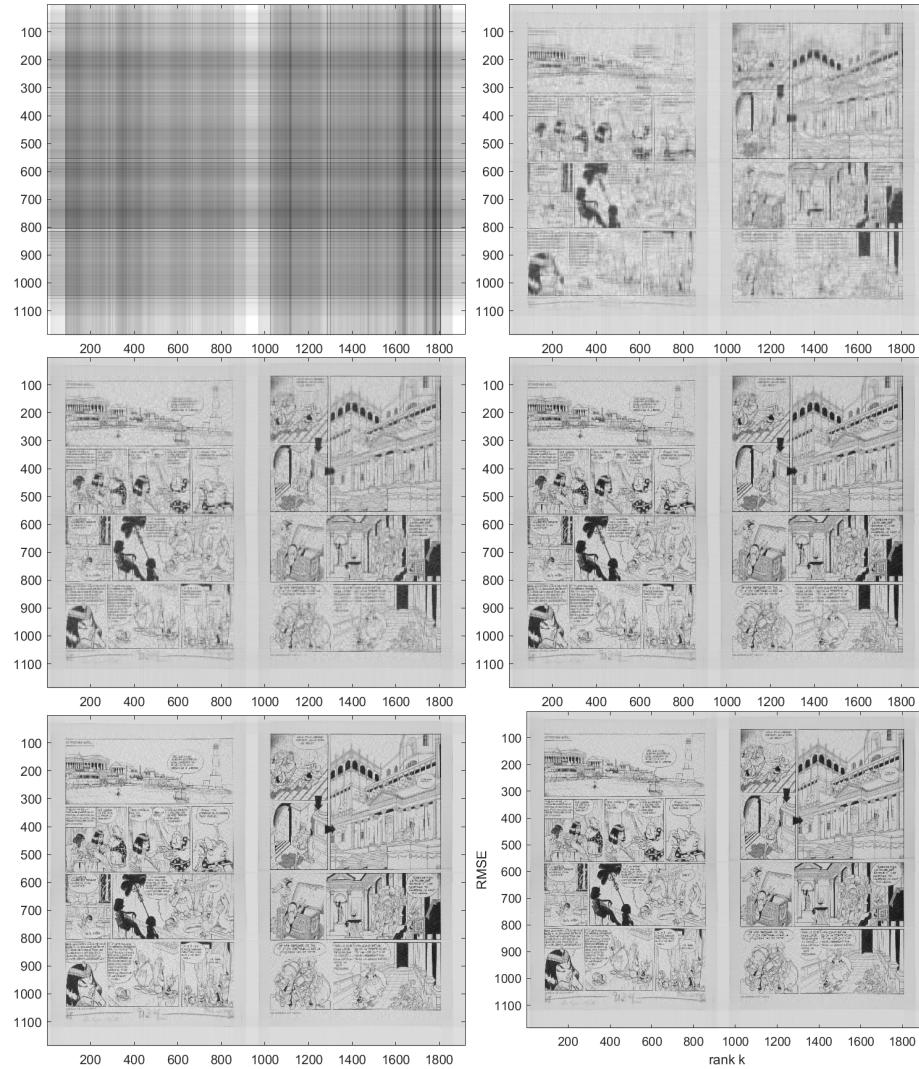


Figure 5: Images reconstruites par la méthode subspace_iter_v2

Nous remarquons que l'image a été reconstruite correctement, pourtant l'image finale est toujours un peu flou.

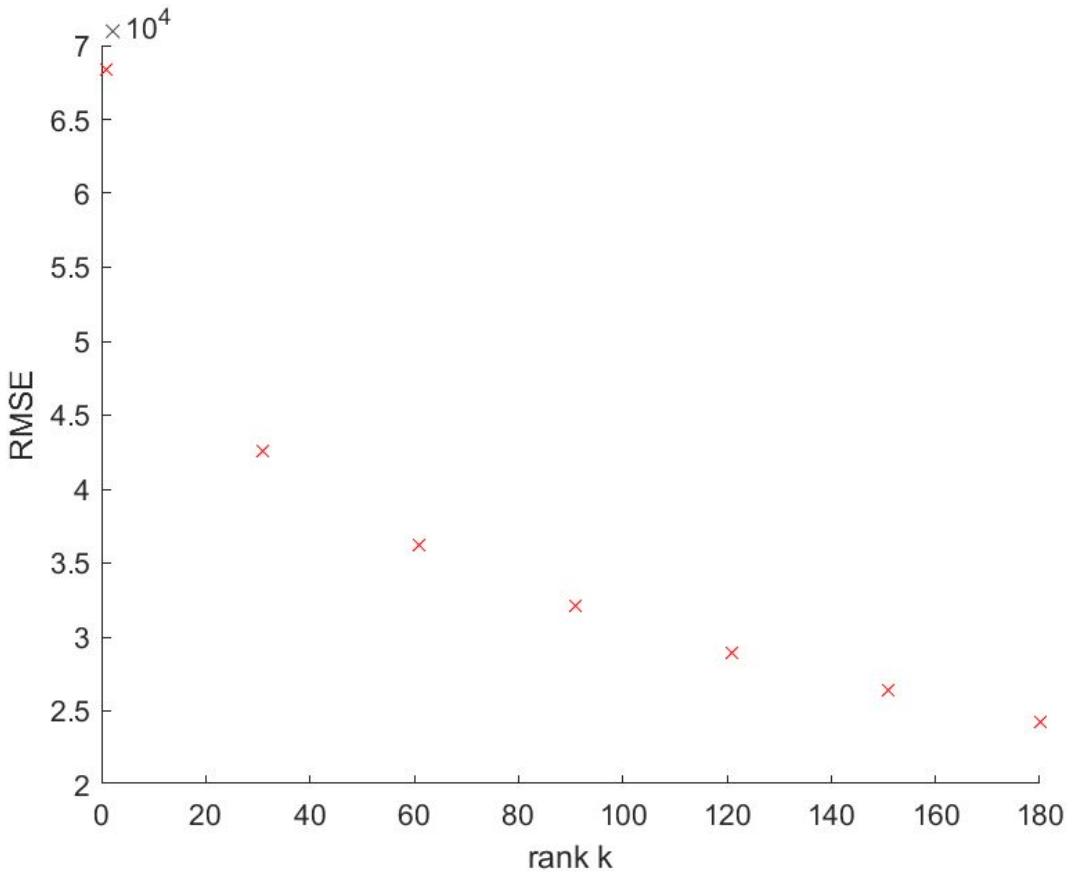


Figure 6: Différences entre image réelle et image reconstruite

D'après la figure de la différence de SVD tracée, il semble que notre figure montre une décroissance de la racine carrée de l'erreur quadratique moyenne (RMSE) entre l'image originale et l'image reconstruite en fonction de l'augmentation du rang de la matrice de l'image reconstruite à l'aide de la SVD.

Cela signifie que lorsque le rang de la matrice de l'image reconstruite augmente, la RMSE entre l'image originale et l'image reconstruite diminue. En d'autres termes, une reconstruction d'image plus précise peut être obtenue en utilisant une matrice de rang plus élevé.

Le rang de la matrice de l'image reconstruite peut être augmenté en utilisant plus de valeurs singulières lors de la reconstruction. Cela permet de capturer plus d'informations sur l'image originale et de mieux reproduire sa structure. Cependant, l'utilisation de plus de valeurs singulières peut également conduire à une augmentation du coût de calcul et de stockage, il faut donc trouver un équilibre entre la précision de la reconstruction et les ressources nécessaires pour la réaliser.

En conclusion, la figure montre que l'augmentation du rang de la matrice de l'image reconstruite à l'aide de la SVD peut améliorer la précision de la reconstruction d'image, mais cela nécessite une utilisation accrue des ressources de calcul et de stockage.

SIMULATION-2: SUBSPACE V2

Paramètres :

$$\text{tolerance} = 10^{-8}$$

$$m = 400$$

$$\text{percentage} = 0.995$$

$$\text{puiss} = 2$$

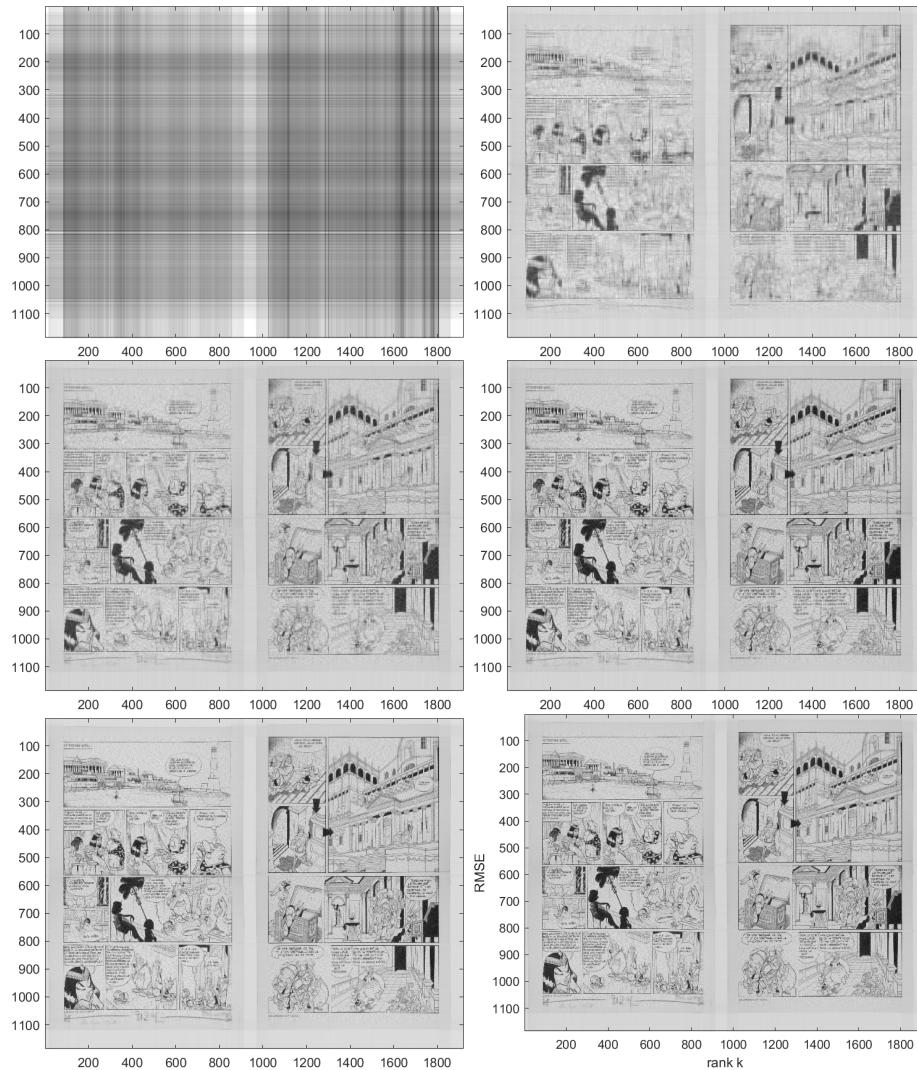


Figure 7: Images reconstruites par la méthode subspace_iter_v2

Nous remarquons que l'image a été reconstruite correctement, pourtant l'image finale est toujours un peu flou.

Différence de SVD

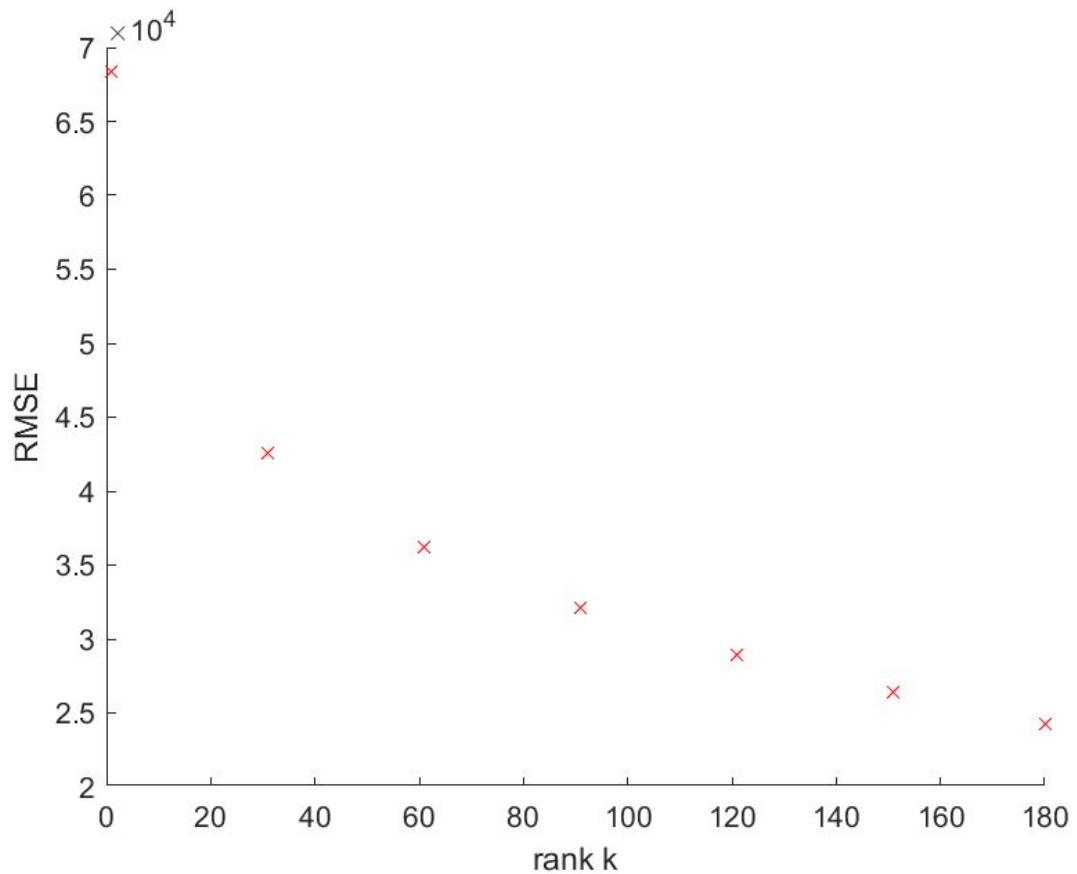


Figure 8: Différences entre image réelle et image reconstruite

Nous remarquons qu'il n'y a pas de différence entre les différences de SVD entre l'image originale et l'image reconstruite entre $p = 1$ et $p = 2$. Les versions subspace_iter v3 et v2 ont presque les mêmes performances dans la reconstruction de l'image en gris.

SIMULATION-3: SUBSPACE V2

Paramètres :

tolerance = 10^{-8}

m = 400

percentage = 0.995

puiss = 3

La reconstruction de l'image peut s'avérer être un processus chronophage en raison de la taille de l'image et de la complexité des calculs nécessaires, ce qui peut entraîner des temps de réponse prolongés voire l'échec de la reconstruction de l'image. Ce problème est causé par la nature complexe des données qui requiert une quantité de temps et de ressources importantes pour être traitée efficacement. Malheureusement, nous ne disposons pas d'un matériel informatique plus performant ni de méthodes de traitement plus avancées pour améliorer les performances du programme de reconstruction d'image.

SIMULATION: SUBSPACE V1

Paramètres :

$$\text{tolerance} = 10^{-8}$$

$$m = 400$$

$$\text{percentage} = 0.995$$

$$\text{puiss} = 1$$

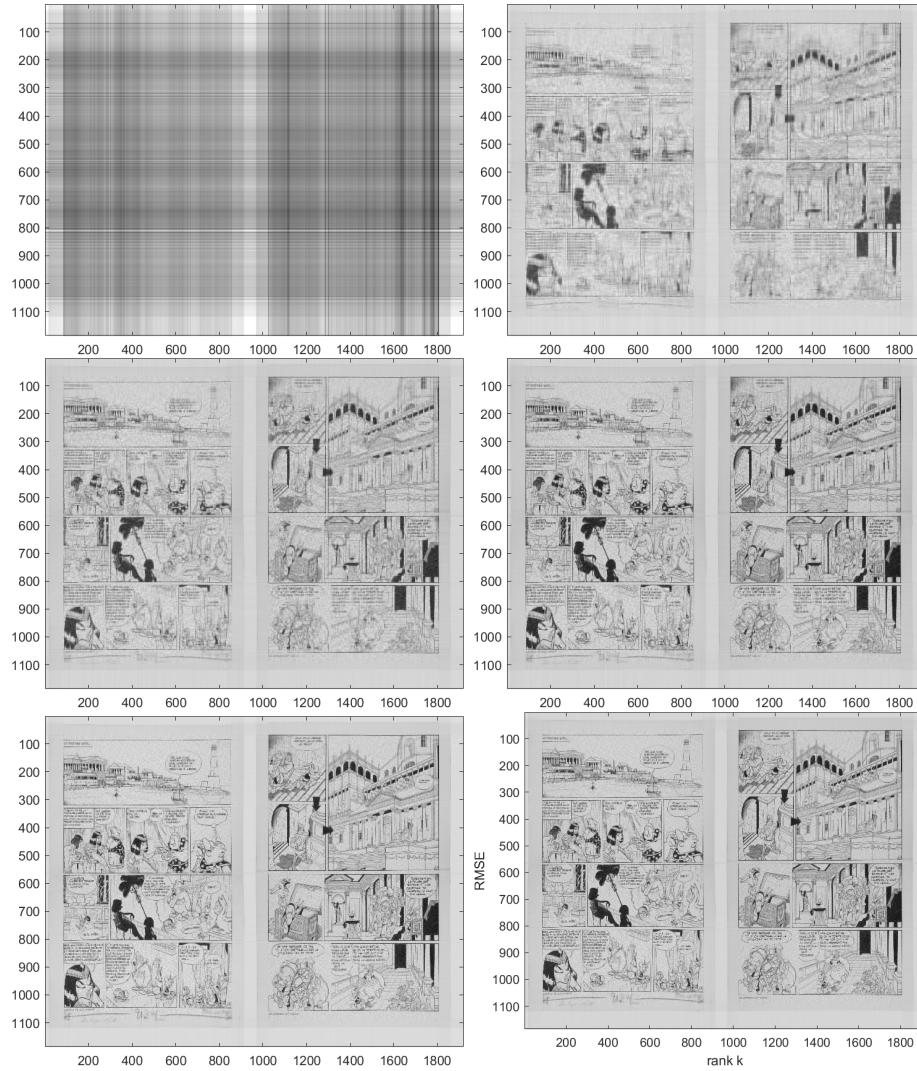


Figure 9: Images reconstruites par la méthode subspace_iter_v1

Nous remarquons que l'image a été reconstruite correctement, pourtant l'image finale est toujours un peu flou.

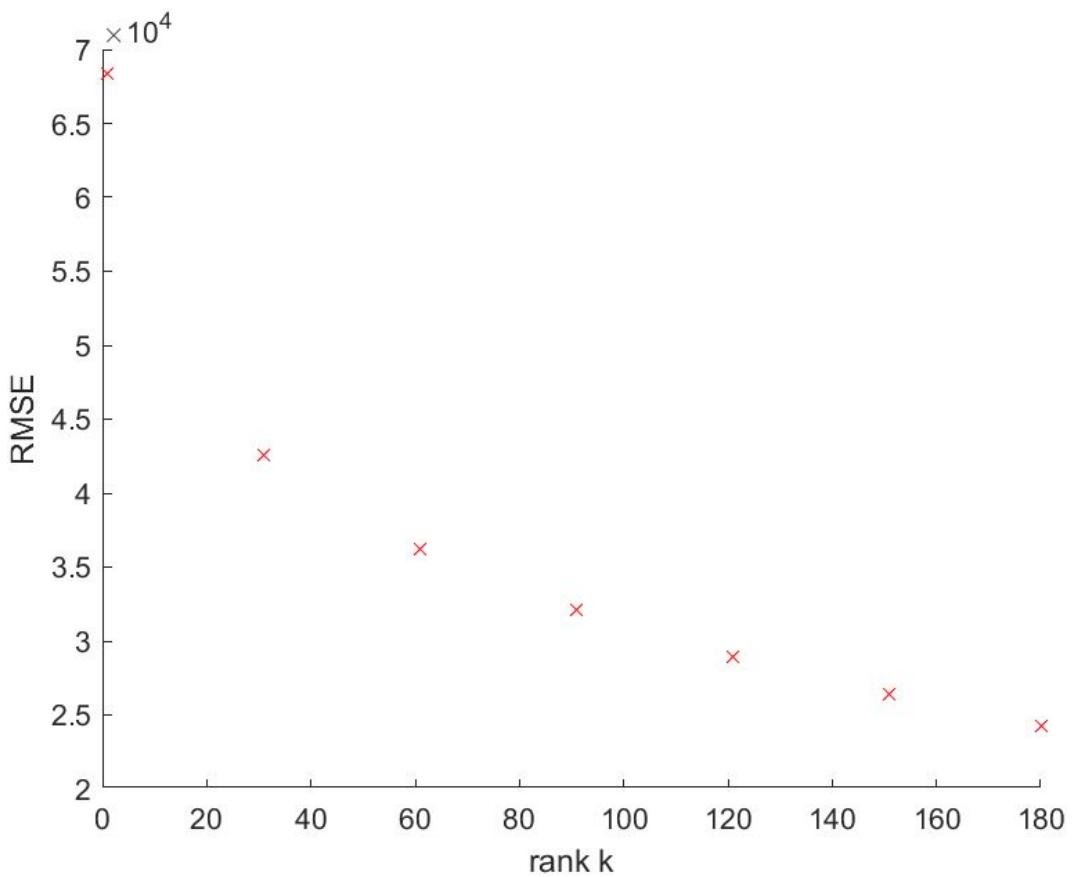


Figure 10: Différences entre image réelle et image reconstruite

Les résultats obtenus sont très proche des simulations précédentes ce qui montre qu'il n'y a pas une différence dans la précision des différentes méthodes mais il différent du temps de calcul et de leur capacité de reconstruire l'image.

SUBSPACE V0 — POWER V11 — POWER V12

L'image n'est malheureusement pas reconstruite quelque soit les paramètres utilisés, le programme ne converge pas et prend un temps infini d'exécution, du coup ces méthodes ne sont pas efficaces dans les traitements des grandes matricess.

4 Reconstruction de l'image colorée

Dans cette section, nous avons exploré la reconstruction d'une image couleur en utilisant différentes méthodes. Pour ce faire, nous avons adapté les implémentations précédentes utilisées pour la reconstruction d'une image en niveaux de gris, en appliquant ces méthodes aux trois matrices de couleurs qui composent l'image colorée, à savoir Ir, Ib et Iv.

Une fois que chaque matrice a été traitée, nous avons assemblé les résultats pour reconstruire l'image finale en utilisant différentes valeurs de rang k pour les matrices. Nous avons toutefois constaté que la reconstruction d'une image couleur prend beaucoup plus de temps que celle d'une image en niveaux de gris, étant donné que les images couleur sont plus volumineuses et que l'on doit traiter trois matrices au lieu d'une seule. En somme, le temps d'exécution pour reconstruire une image couleur est supérieur à trois fois celui nécessaire pour traiter une image en niveaux de gris.

Cela suggère qu'une image couleur est plus difficile à reconstruire avec précision, et que d'autres méthodes plus rapides et plus précises doivent être envisagées pour garantir une bonne reconstruction. En effet, même si les méthodes que nous avons implémentées sont capables de reconstruire une image couleur, leur temps d'exécution élevé peut être un obstacle pour leur utilisation dans des applications pratiques qui requièrent une reconstruction rapide et précise des images couleur.

Le code matlab que nous avons implémenté dans `ReconstructionImageCouleur.m` a pour but de compresser des images colorées en utilisant la SVD (décomposition en valeurs singulières). Dans un premier temps, l'image est lue et affichée. Ensuite, la SVD est calculée pour chaque canal de couleur (rouge, vert et bleu). Ensuite, une boucle est utilisée pour reconstruire l'image en utilisant un nombre croissant de vecteurs singuliers (de 1 à 200, avec un pas de 40). Pour chaque nombre de vecteurs, l'image reconstruite est affichée et la différence entre l'image originale et l'image reconstruite est calculée. Finalement, la différence est affichée pour chaque canal de couleur. Dans un second temps, le code utilise les différentes méthodes de compression pour calculer des matrices de compression, des couples propres et des vecteurs singuliers pour chaque canal de couleur. La reconstruction de l'image est ensuite réalisée en utilisant un nombre croissant de vecteurs singuliers (de 1 à $6 \times E(n_{ev}/6)$). Encore une fois, la différence entre l'image originale et l'image reconstruite est calculée et affichée pour chaque canal de couleur.

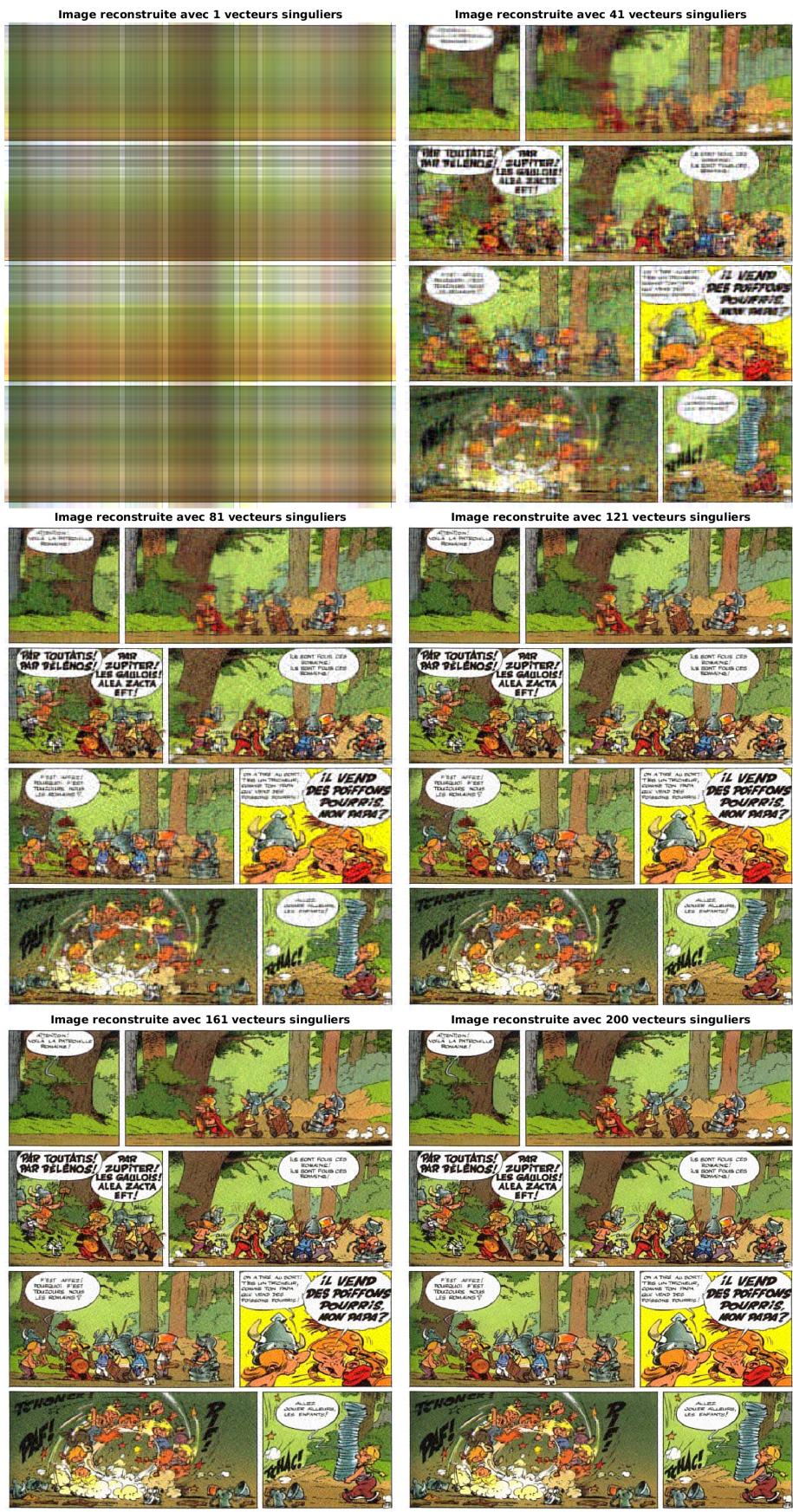


Figure 11: Images colorées reconstruites

5 Conclusion

En conclusion, ce projet nous a permis d'exploiter différentes méthodes vue en cours pour le calcul des couples propres (Eig, PowerV11, PowerV12, SubSpaceIterV0, SubSpaceIterV1, SubSpaceIterV2, SubSpaceIterV3), et on a pu les tester dans la construction d'images grise en premier lieu et les images colorées en deuxième. Bref, on a pu conclure sur l'efficacité et la rapidité de chaque méthode pour tester leurs performances, ainsi, en conclure sur la meilleure méthode de point de vue rapidité et précision.