Reconstruction d'image basée sur la complétion de matrices

ATTALI, Hugo
hugo.attali@univ-lyon2.fr

DJAALEB, Tom
tom.djaaleb@univ-lyon2.fr

20 octobre 2022

Énoncé Etudier et reproduire les expériences proposées dans le blog ici sur le problème de complétion de matrices par méthode d'optimisation alternée. Reproduire les expériences et en étudier l'influence du choix des divers paramètres.

1 Introduction

L'objectif de ce projet est de reproduire les expériences dans le blog précédemment cité et d'en étudier l'influence des paramètres d'initialisation. Il s'avère qu'après quelques essais et une idée intuitive, nous proposons d'aller plus loin pour appliquer la méthode à la reconstruction d'image. Nous détaillons le protocole ainsi que les conditions pour que la méthode fonctionne et décrivons quelques résultats expérimentaux.

2 Problème

2.1 Minimisation alternée

Soit $\mathbf{M}$ une matrice de rang k, celle-ci peut être écrite en forme factorisée :

$$\mathbf{M} = \mathbf{X}\mathbf{Y}^T, \quad \mathbf{X} \in \mathbb{R}^{m \times k}, \mathbf{Y} \in \mathbb{R}^{n \times k} \tag{1}$$

En prenant $\mathbf{X}_0, \mathbf{Y}_0$ les états initiaux de nos deux matrices, nous pouvons alterner l'optimisation de $\mathbf{X}$ et $\mathbf{Y}$ comme suit :

$$\mathbf{X}_t = \arg\min_{\mathbf{X}} f(\mathbf{X}\mathbf{Y}_{t-1}^T) \tag{2}$$

$$\mathbf{Y}_t = \arg\min_{\mathbf{Y}} f(\mathbf{X}_t\mathbf{Y}^T) \tag{3}$$

2.2 Complétion de matrice

Soit $\mathbf{M} \in \mathbb{R}^{n \times p}$ une matrice partiellement observable, nous souhaitons compléter cette matrice avec les entrées manquantes. Pour ce faire, nous devons imposer des hypothèses concernant la matrice $\mathbf{M}$ de départ et la matrice $\mathbf{M}^*$ complète :

— Le rang de la matrice $\mathbf{M}^*$ est "petit".
— Les entrées observable de la matrice $\mathbf{M}$ sont uniformément et aléatoirement sélectionnées (on dénote l'ensemble des entrées observables par Ω).
— les entrées de $\mathbf{M}$ sont "étalées" de sorte qu'un échantillon aléatoire prélève une part proportionnelle de $\mathbf{M}$ avec une bonne probabilité.

Soit $\mathbf{M} \in \mathbb{R}^{n \times p}$ une matrice partiellement observable dont les coefficients appartiennent à Ω, P_Ω la projection d'une matrice sur l'ensemble Ω et $||\cdot||_F$ la norme de Frobenius qui pour la cas réel est égale à $\sqrt{\mathrm{tr}(\mathbf{A}^T\mathbf{A})}$ pour une matrice $\mathbf{A}$ quelconque. Notre problème d'optimisation est défini ainsi :

$$\min_{\mathbf{X} \in \mathbb{R}^{m \times k}, \mathbf{Y} \in \mathbb{R}^{n \times k}} \frac{1}{2}||P_\Omega(\mathbf{M} - \mathbf{X}\mathbf{Y}^T)||_F^2 \tag{4}$$

Pour résoudre ce problème, nous utiliserons la méthode des **Moindres Carrés Alternée**.

3 Expériences et applications

3.1 Exemple simple

La première expérience est celle proposée dans le notebook cité de l'énoncé. La matrice cible $\mathbf{M}^* \in \mathbb{R}^{40\times 70}$ est de la forme

$$\mathbf{M}^* = \begin{pmatrix} -2 & -4 & -2 & -4 & ... & -4 \\ 4 & 2 & 4 & 2 & ... & 2 \\ -2 & -4 & -2 & -4 & ... & -4 \\ ... & ... & ... & ... & ... & ... \\ 4 & 2 & 4 & 2 & ... & 2 \end{pmatrix}$$

Cette matrice est de rang 2. Les auteurs représentent visuellement cette matrice par l'image de la Figure 1.

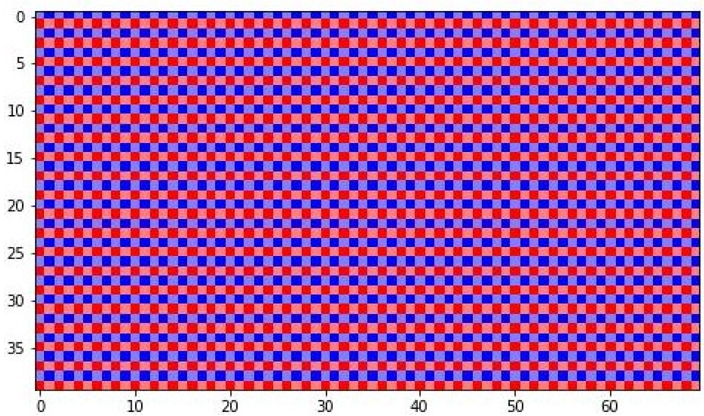
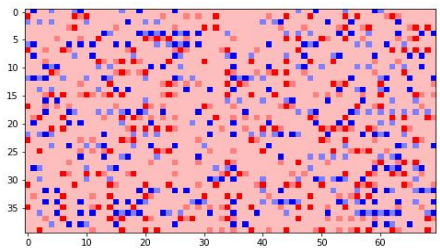
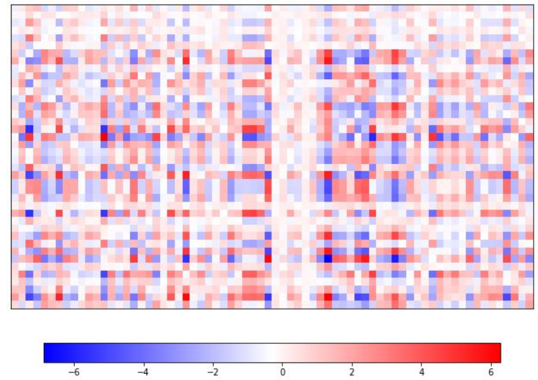


FIGURE 1 – Représentation visuelle de la matrice $\mathbf{M}^*$.

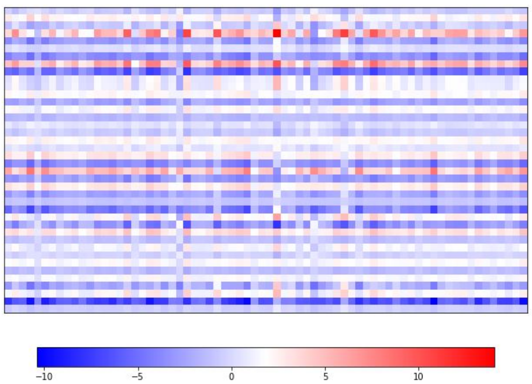
Pour réaliser notre expérience, nous devons prendre aléatoirement une partie de cette matrice, et remplacé les coefficients non sélectionnés par des 0. Un premier paramètre qui est apparaît est la proportion de coefficients sélectionnées. Pour cet exemple, l'auteur choisi d'avoir une matrice de départ avec 25% d'informations. Nous verrons l'impact de ce paramètre sur le résulat par la suite mais regardons en premier lieu l'exécution de l'algorithme itération par itération visualisables sur la Figure 2. L'image 11a représente notre matrice incomplète de départ, comprenant 25% de l'information de notre matrice référence en Figure 1. Nous voyons ensuite que notre algorithme converge rapidement vers ce qui est attendu. En seulement 7 itérations nous retrouvons la matrice complète. Ce qui est contre intuitif est que la matrice ne résout pas un problème d'optimisation sur une matrice de dimensions 40×70 mais sur deux matrices de dimensions 40×2 et 2×70 où 2 est le rang de la matrice de départ ce qui accélère grandement le processus.



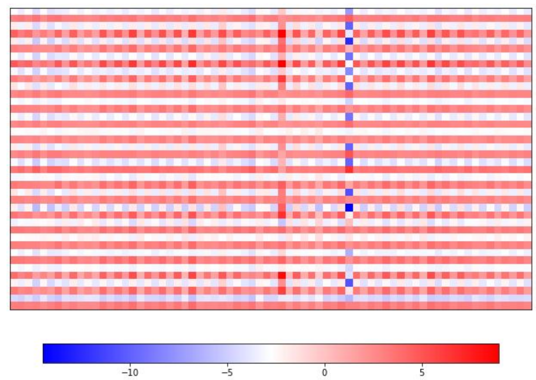
(a) Itération 0



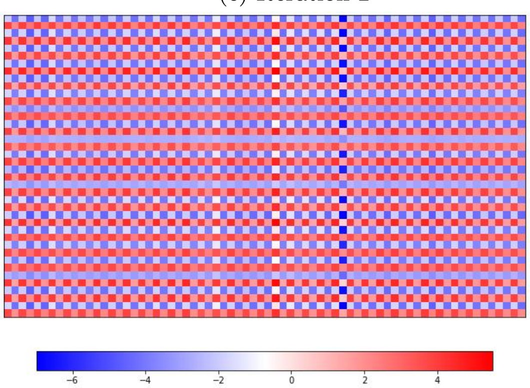
(b) Itération 1



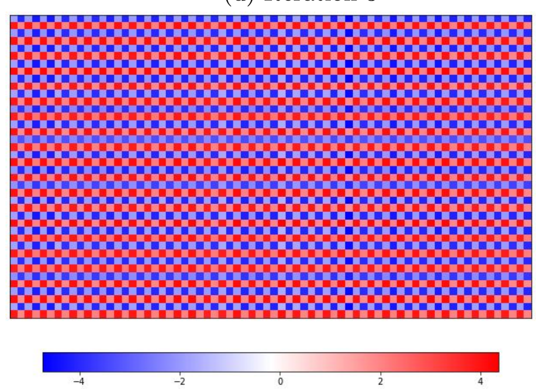
(c) Itération 2



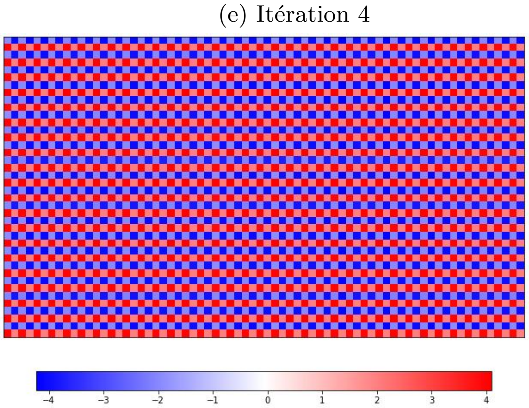
(d) Itération 3



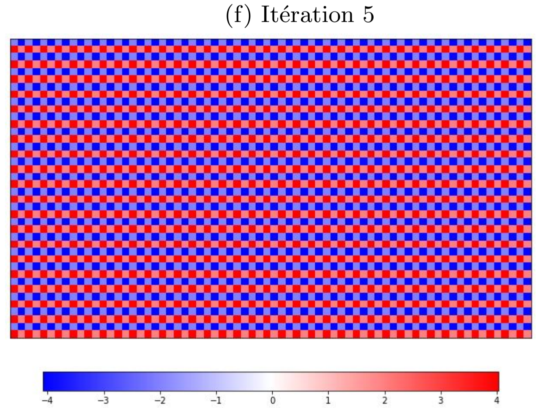
(e) Itération 4



(f) Itération 5



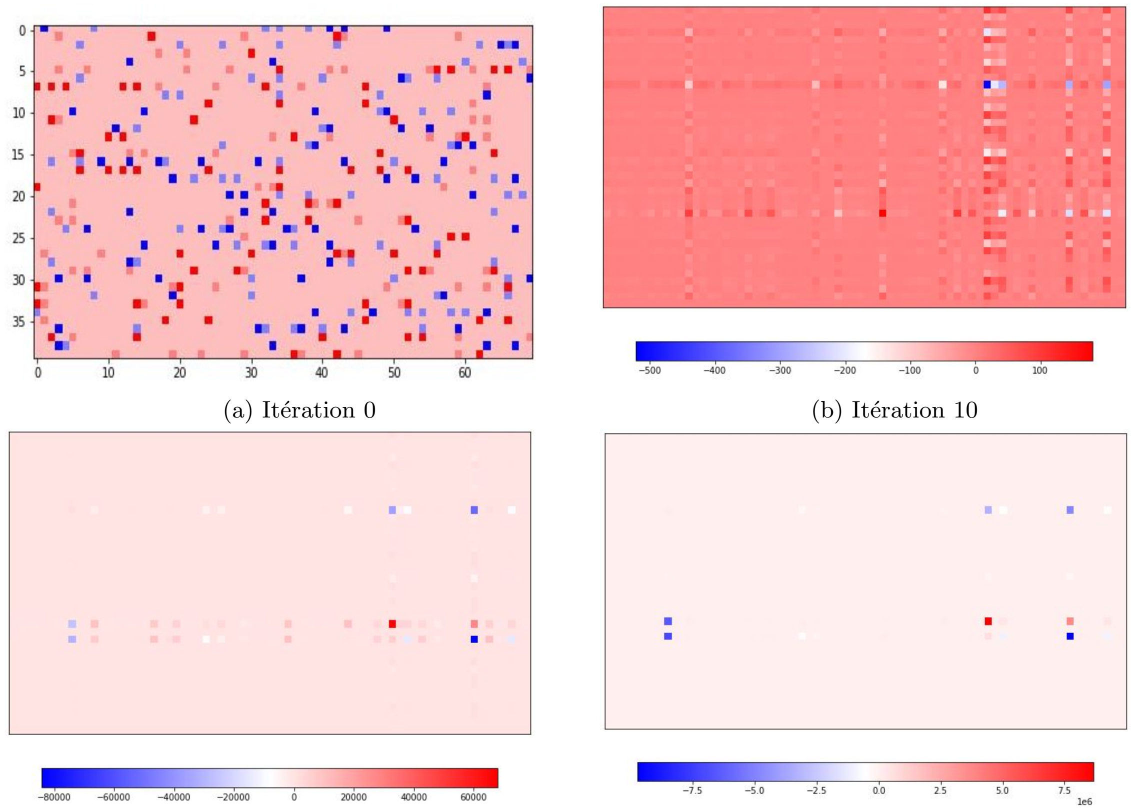
(g) Itération 6



(h) Itération 7

FIGURE 2 – Décomposition des étapes de l'optimisation alternée avec 25% d'information initale.

Regardons maintenant ce qu'il se passe si nous retirons de l'information au départ. Dans le cas où nous ne prenons que 10% de $\mathbf{M}^*$, l'algorithme n'est pas en capacité de retrouver la matrice complète et ce peu importe le nombre d'itération. La Figure 3 illustre bien le problème. Cette non convergence est dûe au manque d'information initiale de notre matrice incomplète. Inversement, lorsque la matrice de départ à une grande part d'information, l'algorithme convergera plus rapidement vers la matrice voulue, à condition de respecter les hypothèses de départ.



(a) Itération 0

(b) Itération 10

(c) Itération 100

(d) Itération 1000

FIGURE 3 – Décomposition des étapes de l'optimisation alternée avec 10% d'information initale.

3.2 Reconstruction d'images

L'auteur ayant utilisé un exemple visuel pour montrer le fonctionnement de l'algorithme, nous nous sommes interrogé sur la possible utilité de celui-ci pour reconstruire des images. Premièrement, nous devons convertir les coefficients de notre matrice en pixel de couleur, nous n'utiliserons pas la norme RGB car cela impliquerait d'avoir un tableau en 3 dimensions et donc compliquerais la tâche (si celle-ci est possible). Nous allons donc travailler en nuance de gris (`plt.cm.gray` de la libraire *matplotlib*).

3.2.1 Visage

Le premier cas concerne une image simple représentée en Figure 4, un visage de résolution 25×25.

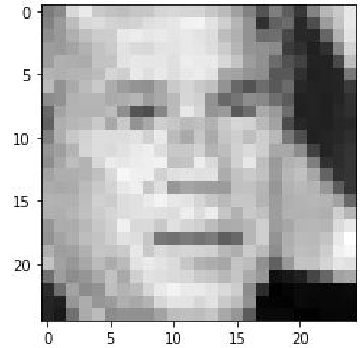
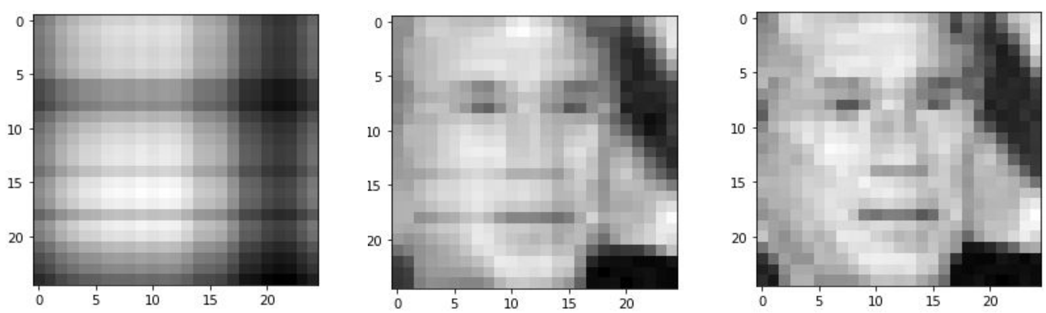


Figure 4 – Visage.

Le rang de la matrice associée au visage est égal à 25, ce qui pose un problème pour nos hypothèses. Nous pouvons faire une réduction de rang sur la matrice de départ, voyons comment cela se comporte visuellement sur la Figure 5.



(a) Rang = 1 (b) Rang = 5 (c) Rang = 10

Figure 5 – Visualisation de l'image en fonction de la réduction au rang associé.

Il est clair qu'une matrice réduite au rang 1 n'est pas envisageable car nous ne voulons pas que notre algorithme converge vers une image illisible. En revanche les deux autres images semblent perdre en information par rapport à la première mais elles sont tout de même interprétables. Les résultats pour ces deux cas ne sont pas assez satisfaisants pour être proposés pour les raisons suivantes :

- Dans les deux cas, l'algorithme ne converge pas toujours vers l'image.
- Augmenter l'information de départ augmente les chances de converger, cependant nous devons avoir au minimum 70% de la matrice cible pour pouvoir espérer que l'algorithme converge vers celle-ci.

3.2.2 Tigre

N'étant pas satisfaits de nos précédents résultats, nous avons décidé de tester notre protocole sur une photo de la vie courante ayant une plus grande résolution (480×640), nous avons choisi arbitrairement une photo de tigre (cf Figure 6). La matrice associée à l'image est de rang 480.

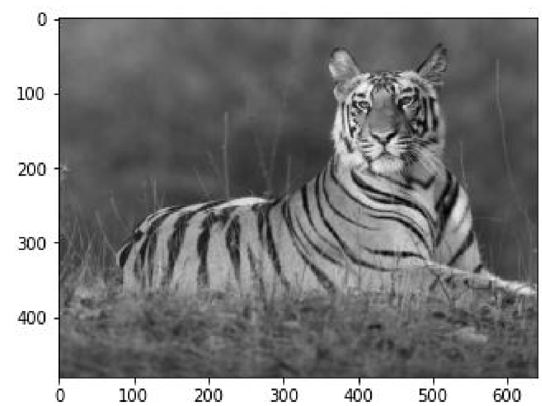
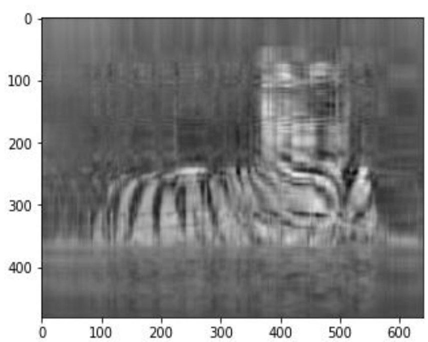
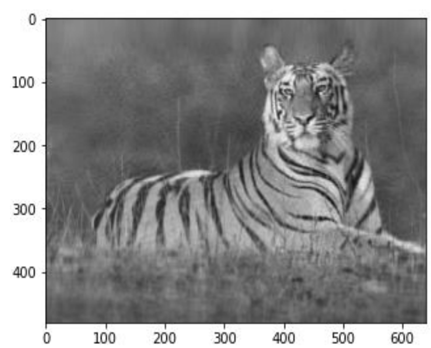


FIGURE 6 – Tigre.

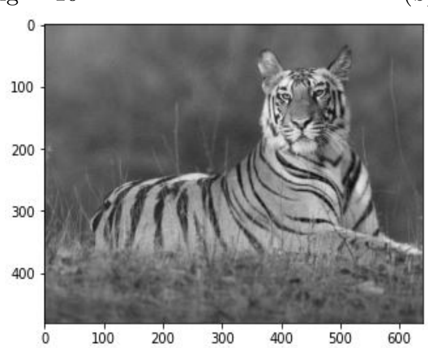
Comme en témoigne la Figure 7, une réduction de rang 10 perd trop d'information par rapport à l'image de départ mais les réductions de rang 50 et 100 renvoient des images lisibles pour l'homme, avec de meilleure nuance de couleurs ainsi que plus de détails pour la seconde.



(a) Rang = 10



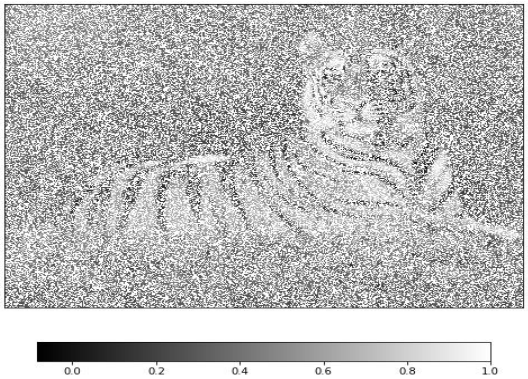
(b) Rang = 50



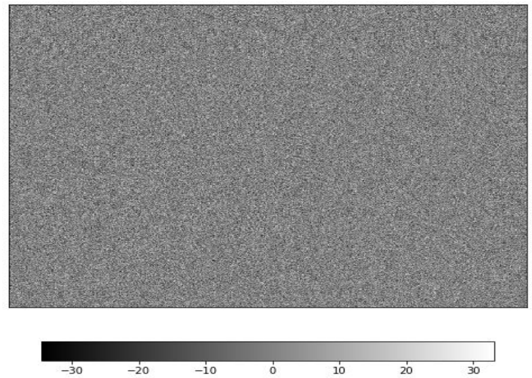
(c) Rang = 100

FIGURE 7 – Visualisation de l'image en fonction de la réduction au rang associé.

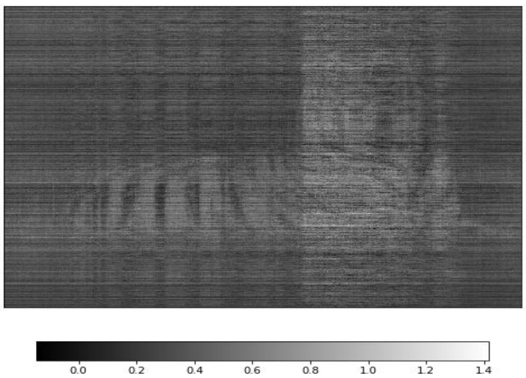
La décomposition de l'algorithme pour une réduction de rang à 50 et 50% d'information au départ est proposée en Figure 8



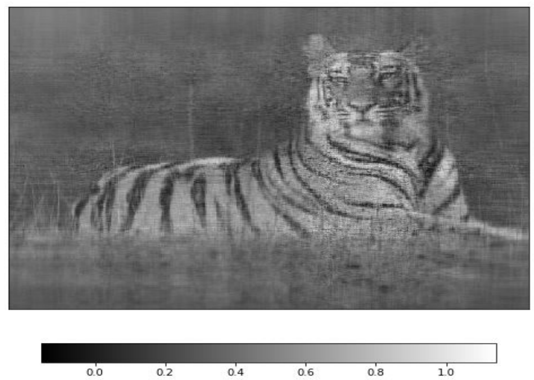
(a) Itération 0



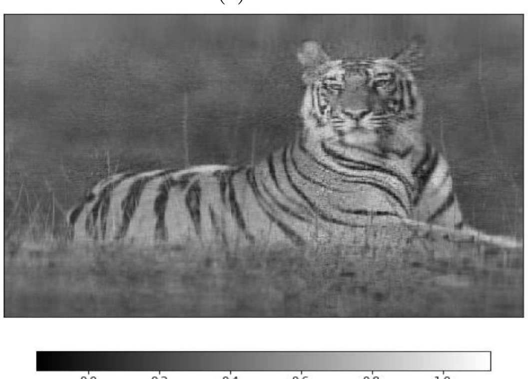
(b) Itération 1



(c) Itération 2

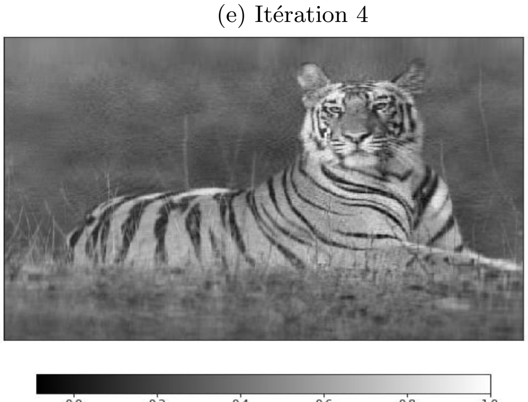


(d) Itération 3

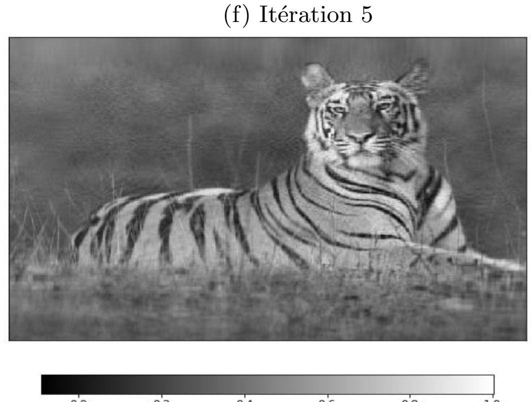


(e) Itération 4

(f) Itération 5



(g) Itération 6



(h) Itération 7

FIGURE 8 – Décomposition des étapes de l'optimisation alternée avec 50% d'information initale.

Pour alléger le rapport, nous ne montrons pas tous les tests que nous avons effectué mais nous remarquons quelques informations notables. La première est que plus le rang est grand, plus l'algorithme est long. Ceci est dûes aux matrices X et Y qui sont de plus grandes dimensions lorsque le rang augmente. La deuxième information est que plus le rang est grand, plus il est nécessaire d'avoir d'information au départ pour que l'algorithme retrouve l'image de départ. Enfin, le nombre d'itérations de l'algorithme n'a pas besoin d'être très important car l'algorithme converge rapidement, soit vers la bonne solution, soit vers une solution éronnée selon les autres paramètres définis au départ.

3.2.3 Immeubles

La dernière expérience cherche à démontrer si la méthode fonctionne pour un paysage entier. En effect, la section précédente était appliquée à une image qui possède un objet au milieu mais le reste du décor n'a pas vraiment d'importance et donc facilite le déroulement de l'algorithme. Pour ce cas là, nous étudierons une photo de Lyon (Figure 10). La matrice associée à cette image est de rang 768.

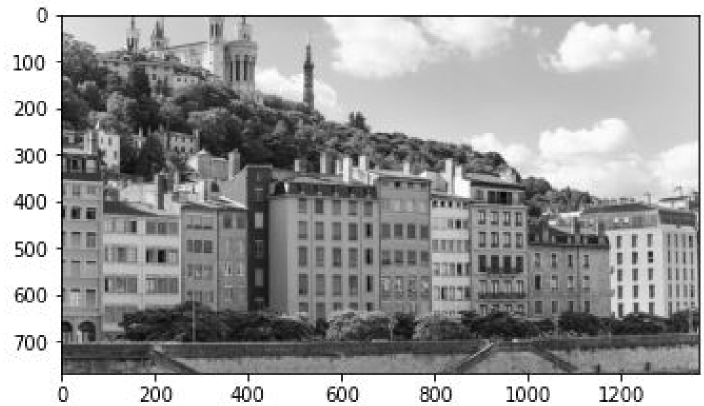


Figure 9 – Lyon.

Pour notre expérience, nous réduisons la matrice jusqu'au rang 100 et prenons comme l'expérience précédente 50% de la matrice souhaitée pour notre initialisation.

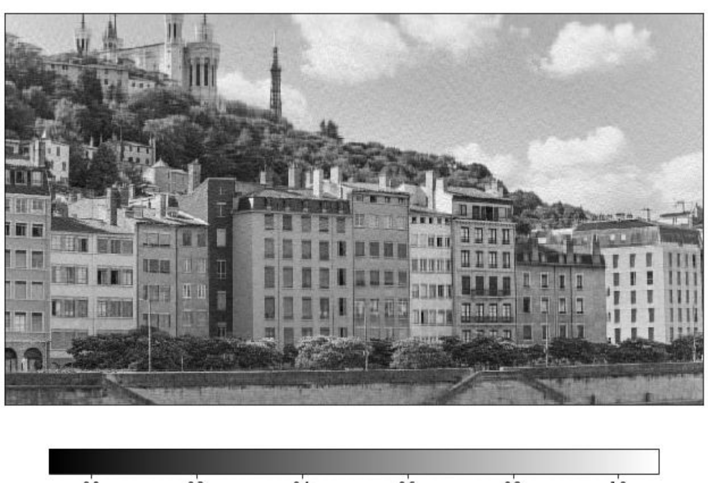
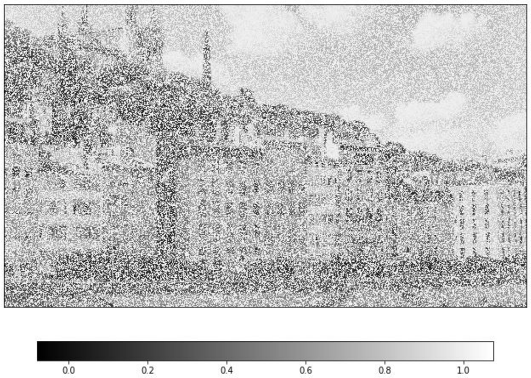
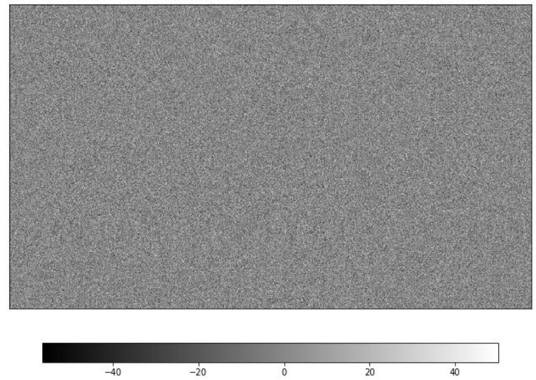


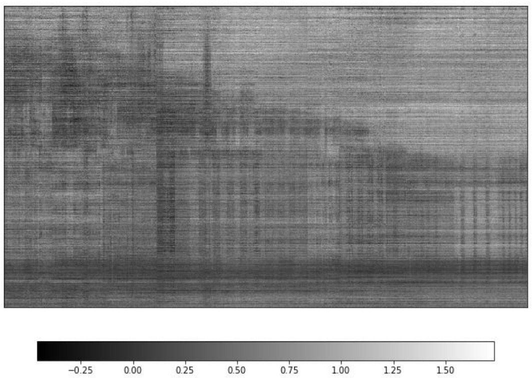
Figure 10 – Rang = 100.



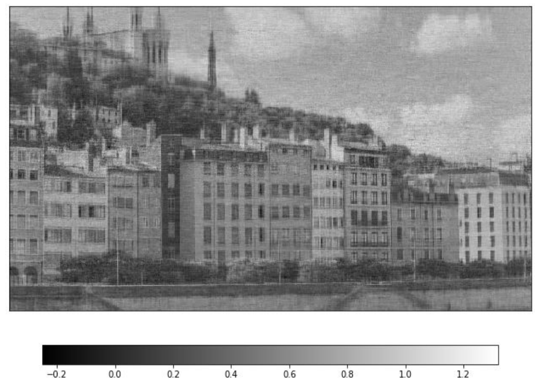
(a) Itération 0



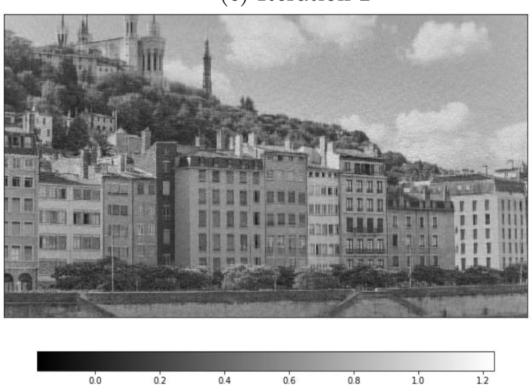
(b) Itération 1



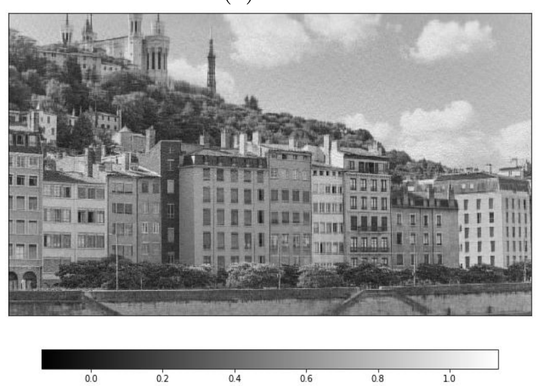
(c) Itération 2



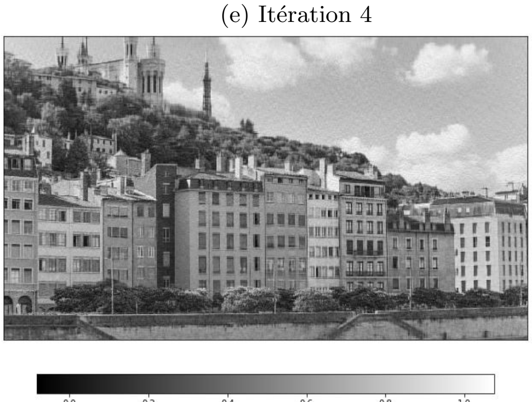
(d) Itération 3



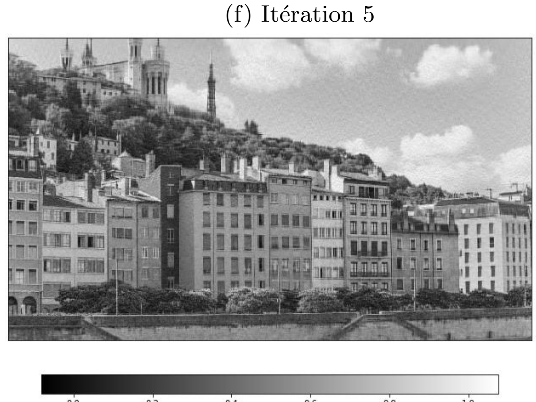
(e) Itération 4



(f) Itération 5



(g) Itération 6



(h) Itération 7

FIGURE 11 – Décomposition des étapes de l'optimisation alternée avec 50% d'information initale.

La Figure 11 décrit les étapes de notre algorithme pour la photo de Lyon. L'image de départ a pu être très bien approximée malgré les détails comme la distinction des arbres.

4 Conclusion

Ce projet nous a permis de mettre en évidence que la méthode de complétion de matrices basée sur l'optimisation alternée peut être applicable sur la reconstruction d'images à partir d'un échantillon. De plus, le nombre d'itération ne doit pas nécessairement être grand pour obtenir de bonne performance. Le rang de la matrice à reconstruire est un facteur important sur le temps d'exécution de l'algorithme et sur la proportion d'information que l'itération 0 doit contenir pour que celui-ci converge vers la bonne solution.