

Rapport : Floutage et défloutage d'images

Ali OKOU et Mathusan SELVAKUMAR

31 Mai 2024

1 Introduction

Le floutage d'images est un problème courant en photographie et en imagerie scientifique. Il peut être causé par divers facteurs tels que le mouvement de la caméra, la mise au point incorrecte, ou la diffusion de la lumière.

La déconvolution¹ est une technique utilisée pour inverser le processus de floutage et restaurer l'image originale. Parmi les nombreuses méthodes de déconvolution, l'algorithme de Richardson-Lucy est particulièrement populaire pour sa capacité à produire des résultats de haute qualité, même en présence de bruit.

Alors que l'algorithme de Richardson-Lucy nécessite une connaissance préalable de la fonction de floutage, nous avons exploré une variante de l'algorithme, Blind Richardson-Lucy, qui peut estimer la fonction de floutage à partir de l'image floutée elle-même.

Toutefois, il est important de noter que ces différentes méthodes de déconvolution ne sont pas parfaites et présentent des limites d'utilisation.

2 Déconvolution de Richardson-Lucy

L'algorithme de Richardson-Lucy². est une méthode itérative utilisée pour restaurer les images floues. Il est surtout utile dans les cas où le flou peut être modélisé par une fonction d'étalement du point (Point Spread Function, PSF). Cet algorithme maximise la probabilité que l'image résultante, lorsqu'elle est convoluée avec la PSF, soit une copie de l'image floue. Cette fonction peut être efficace lorsque vous connaissez bien la PSF mais peu le bruit additif de l'image.

2.1 Principe Mathématique

L'algorithme vise à résoudre l'équation de convolution suivante :

$$I = J * P \tag{1}$$

où I est l'image observée (floue), J est l'image à restaurer (non floue) et P est la PSF. La convolution est représentée par l'opérateur $*$.

L'objectif est de trouver J en maximisant la probabilité $\mathbb{P}(I | J)$, qui est la probabilité de l'image observée étant donnée l'image à restaurer et la PSF.

2.2 Algorithme de Richardson-Lucy

L'algorithme de Richardson-Lucy procède de manière itérative comme suit :

1. La convolution de deux fonctions f et g est définie par :

$$(f * g)(t) = \int_{-\infty}^{\infty} f(\tau)g(t - \tau) d\tau$$

2. Le PSF doit être connu pour pouvoir appliquer l'algorithme de Richardson-Lucy.

```

Input: Image floue  $I$ , PSF  $P$ , nombre d'itérations  $n\_it$ 
Output: Image restaurée  $J$ 
Initialiser  $J_0 = I$ ;
for  $n = 1$  to  $n\_it$  do
    Convolution  $J_n$  avec  $P$  pour obtenir une estimation floutée  $I_{\text{estimée}}$ ;
     $I_{\text{estimée}} = J_n * P$ ;
    Calculer le ratio de flou relatif  $\text{Ratio} = \frac{I}{I_{\text{estimée}} + \epsilon}$ ;
    Convolution ce ratio avec le miroir de la PSF;
    Correction =  $\text{Ratio} * P_{\text{miroir}}$ ;
    Mettre à jour l'estimation  $J_{n+1} = J_n \times \text{Correction}$ ;
end
return  $J_{n+1}$ ;

```

Algorithm 1: Déconvolution de Richardson-Lucy

2.2.1 Initialisation

L'algorithme commence par une estimation initiale de l'image défloutée J_0 , souvent prise comme étant l'image observée I .

2.2.2 Convolution

À chaque itération, l'estimation actuelle J_n est convoluée avec la PSF P pour obtenir une estimation floutée $I_{\text{estimée}}$.

$$I_{\text{estimée}} = J_n * P \quad (2)$$

2.2.3 Ratio de flou relatif

Le ratio de flou relatif est calculé pour mesurer combien l'estimation floutée diffère de l'image observée.

$$\text{Ratio} = \frac{I}{I_{\text{estimée}} + \epsilon} \quad (3)$$

où ϵ est une petite constante pour éviter la division par zéro.

2.2.4 Correction de l'estimation

Ce ratio est ensuite convolué avec le miroir de la PSF P_{miroir} pour corriger l'estimation.

$$\text{Correction} = \text{Ratio} * P_{\text{miroir}} \quad (4)$$

2.2.5 Mise à jour de l'estimation

L'estimation est mise à jour en multipliant l'estimation actuelle par la correction obtenue.

$$J_{n+1} = J_n \times \text{Correction} \quad (5)$$

2.2.6 Explication de l'utilisation du miroir de la PSF

La convolution avec le miroir de la PSF est essentielle pour corriger l'estimation de l'image. Cette étape simule l'effet inverse de la convolution effectuée par la PSF initiale. De plus, elle s'assure de la symétrie dans la convolution, orientant correctement les mises à jour de l'image.

2.3 Résultats

2.3.1 Détails sur l'implémentation

L'algorithme de Richardson-Lucy a été implémenté en Python en utilisant la bibliothèque NumPy pour effectuer les calculs matriciels de base, sans recourir à des bibliothèques tierces souvent considérées comme des *boîtes noires*. Cependant, pour des raisons de performance, la bibliothèque SciPy a été utilisée pour les calculs de convolution. Les étapes suivantes ont été suivies :

- Les images floues ont été générées en utilisant la fonction `scipy.signal.convolve2d` avec un noyau de convolution approprié.
- Ces images floues ont ensuite été défloutées en appliquant l'algorithme de Richardson-Lucy avec un nombre variable d'itérations.
- Les images traitées ont été enregistrées au format PNG pour une visualisation facile.

Ainsi, deux versions de l'implémentation de l'algorithme de Richardson-Lucy ont été réalisées :

- Une version manuelle, qui ne bénéficie pas des optimisations et est donc plus lente.
- Une version rapide (*fast*), qui utilise la bibliothèque SciPy pour les calculs de convolution, offrant ainsi des performances nettement supérieures.

La différence de performance entre ces deux versions est significative. Par exemple, pour une image de taille 250×250 , la version manuelle prend environ 15 secondes pour 5 itérations, tandis que la version rapide prend moins de 1 seconde pour le même nombre d'itérations.

2.3.2 Petite précision sur le PSF

Par ailleurs, plutôt que le PSF, un noyau de convolution a été utilisé pour générer les images floues. Cela a permis de simuler le flou sans avoir besoin de connaître la PSF exacte.

2.3.3 Images Originales



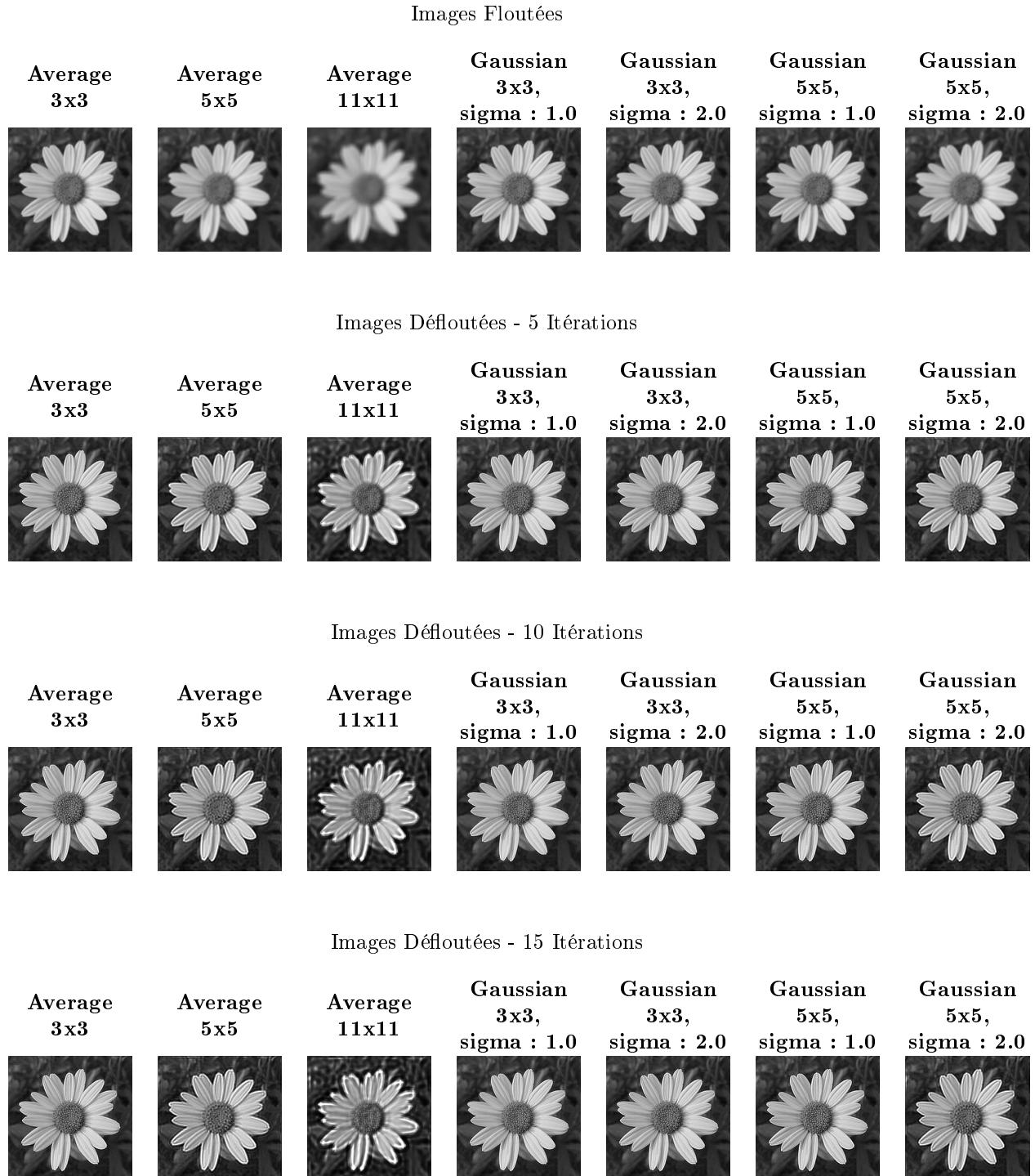
(1) Fleur (niveaux de gris)



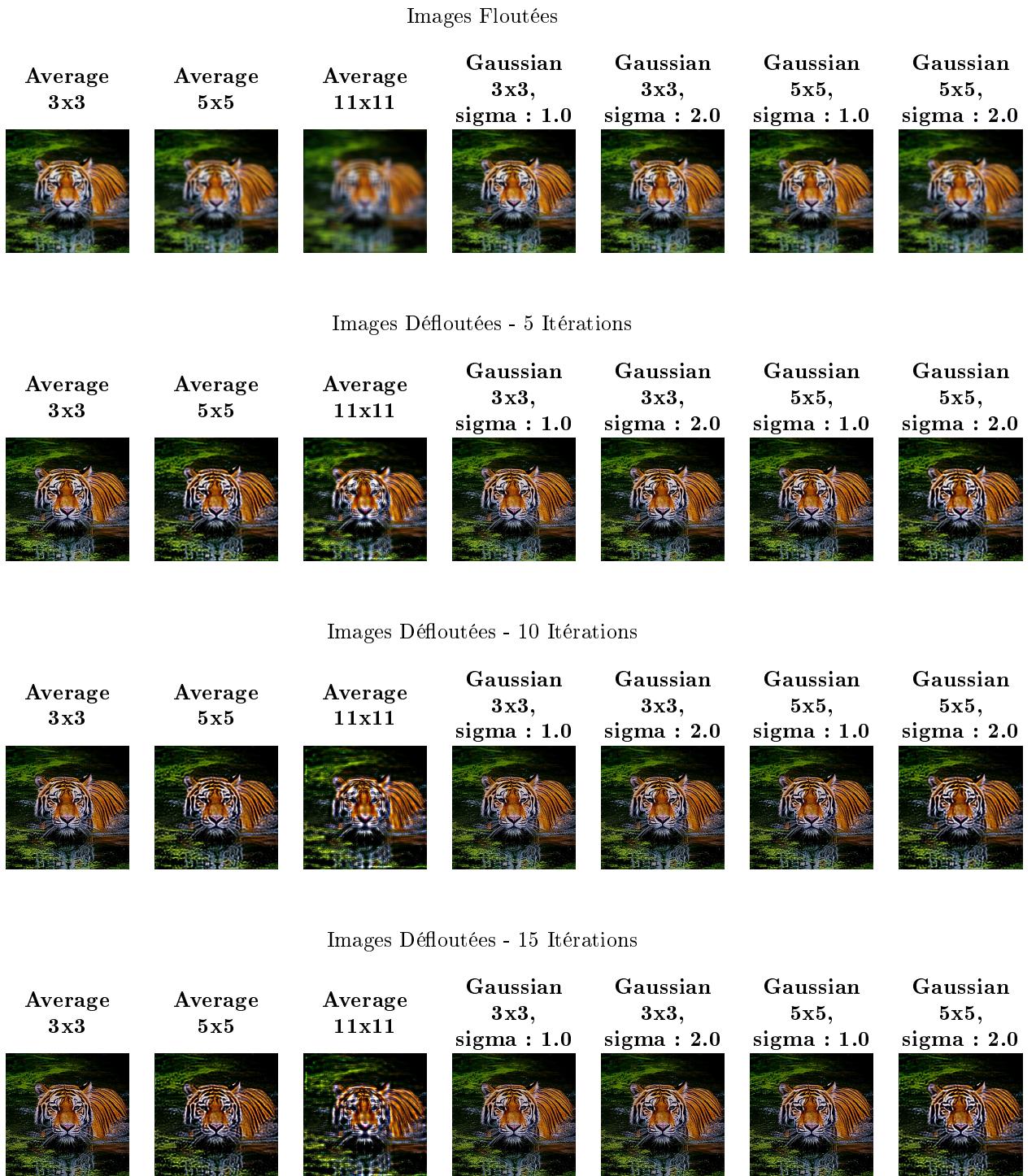
(2) Tigre

2.3.4 Images Traitées

(1) Fleur (niveaux de gris)



(2) Tigre



2.4 Conclusion sur l'algorithme de Richardson-Lucy

Grâce à l'application de l'algorithme de Richardson-Lucy, nous avons pu restaurer des images floues avec succès. Toutefois, il est important de noter que l'algorithme est sensible au bruit et peut produire des artefacts si le bruit est trop important. Un très grand nombre d'itérations peut également entraîner un surajustement et produire des résultats de mauvaise qualité. Il est donc essentiel de trouver un équilibre entre le nombre d'itérations et la qualité de l'image restaurée.

3 Déconvolution Aveugle de Richardson-Lucy

L'algorithme de Richardson-Lucy est une méthode de déconvolution qui nécessite une connaissance préalable de la fonction de floutage. Cependant, dans de nombreuses applications pratiques, cette fonction est inconnue et doit être estimée à partir de l'image floutée elle-même. Pour résoudre ce problème, nous avons exploré une variante de l'algorithme de Richardson-Lucy, appelée déconvolution aveugle de Richardson-Lucy, qui peut estimer la fonction de floutage à partir de l'image floutée. Toutefois, cela nécessite une initialisation précise avec une estimation raisonnable de la fonction de floutage. L'algorithme se charge ensuite de raffiner cette estimation pour restaurer l'image originale.

3.1 Principe Mathématique

L'algorithme de déconvolution aveugle de Richardson-Lucy vise à résoudre l'équation de convolution suivante :

$$I = J * P \quad (6)$$

où I est l'image observée (floue), J est l'image à restaurer (non floue) et P est la fonction d'étalement du point (PSF). La convolution est représentée par l'opérateur $*$.

L'objectif est de trouver J et P en maximisant la probabilité $\mathbb{P}(I | J, P)$, qui est la probabilité de l'image observée étant donnée l'image à restaurer et la PSF. Pour ce faire, l'algorithme itère entre la mise à jour de J et la mise à jour de P .

3.2 Algorithme de Richardson-Lucy Aveugle

L'algorithme de Richardson-Lucy aveugle procède de manière itérative comme suit :

```

Input: Image floue  $I$ , PSF initial  $P$ , nombre d'itérations pour l'image  $n\_it$ , nombre d'itérations
       pour la PSF  $psf\_it$ 
Output: Image restaurée  $J$ , PSF affinée  $P$ 
Initialiser  $J_0 = I$ ;
for  $n = 1$  to  $n\_it$  do
    Convoyer  $J_n$  avec  $P$  pour obtenir une estimation floutée  $I_{\text{estimée}}$ ;
     $I_{\text{estimée}} = J_n * P$ ;
    Calculer le ratio de flou relatif  $\text{Ratio} = \frac{I}{I_{\text{estimée}} + \epsilon}$ ;
    Convoyer ce ratio avec le miroir de la PSF;
    Correction =  $\text{Ratio} * P_{\text{miroir}}$ ;
    Mettre à jour l'estimation  $J_{n+1} = J_n \times \text{Correction}$ ;
    for  $m = 1$  to  $psf\_it$  do
        Convoyer  $J_{n+1}$  avec  $P$  pour obtenir une nouvelle estimation floutée  $I_{\text{estimée}}$ ;
        Calculer le ratio d'erreur  $E = \frac{I}{I_{\text{estimée}} + \epsilon}$ ;
        Convoyer ce ratio avec le miroir de l'image restaurée  $J_{n+1}$ ;
        PSF_Update =  $E * J_{n+1,\text{miroir}}$ ;
        Mettre à jour la PSF  $P = P \times \text{PSF\_Update}$ ;
        Normaliser  $P = \frac{P}{\sum P}$ ;
    end
end
return  $J_{n+1}, P$ ;
```

Algorithm 2: Déconvolution Aveugle de Richardson-Lucy

3.3 Résultats

3.3.1 Image Originale

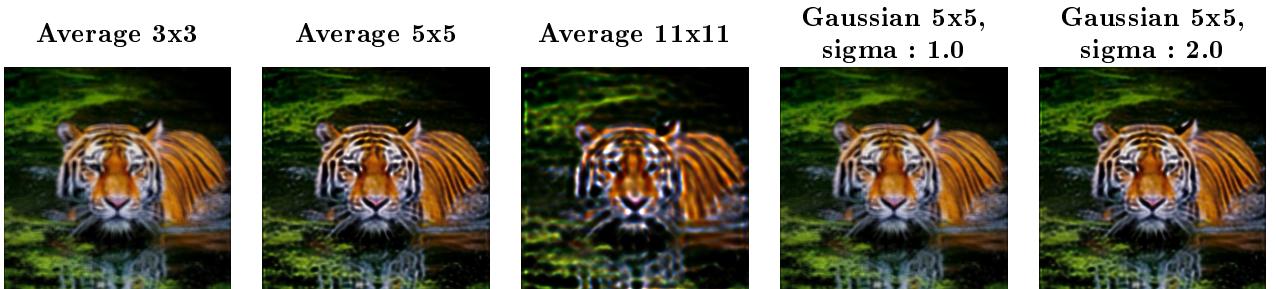


(A) Tigre flou (Noyau inconnu)

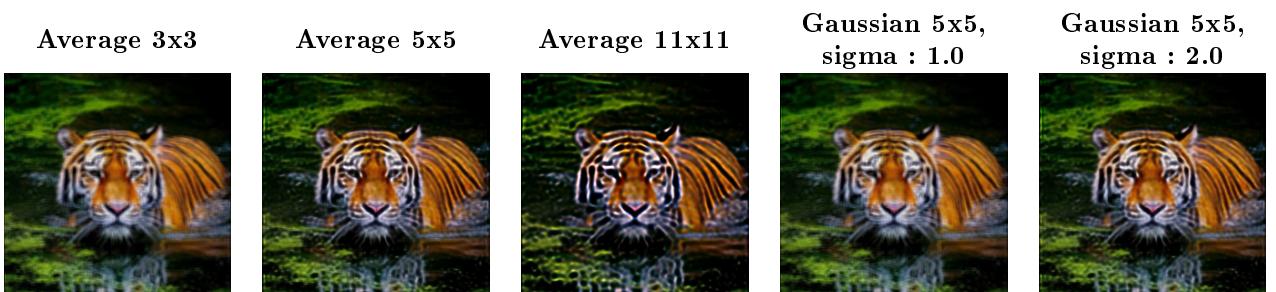
3.3.2 Images Traitées

(A) Tigre Flouté (Noyau inconnu)

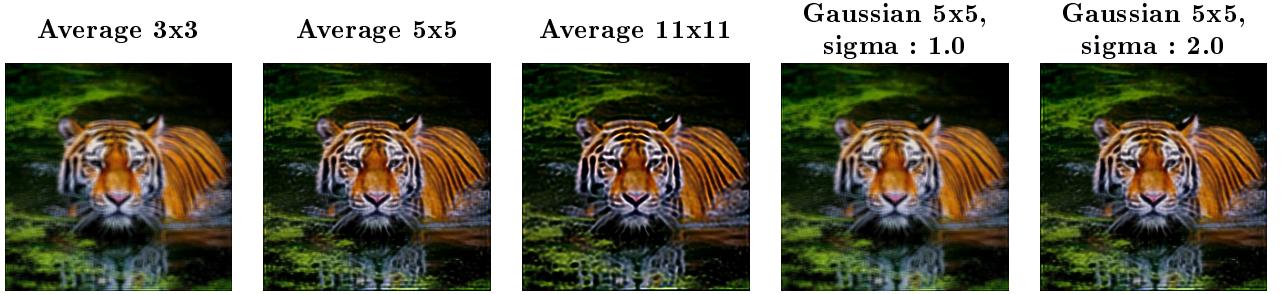
Images Défloutées - 15 Itérations, 25 Itérations PSF



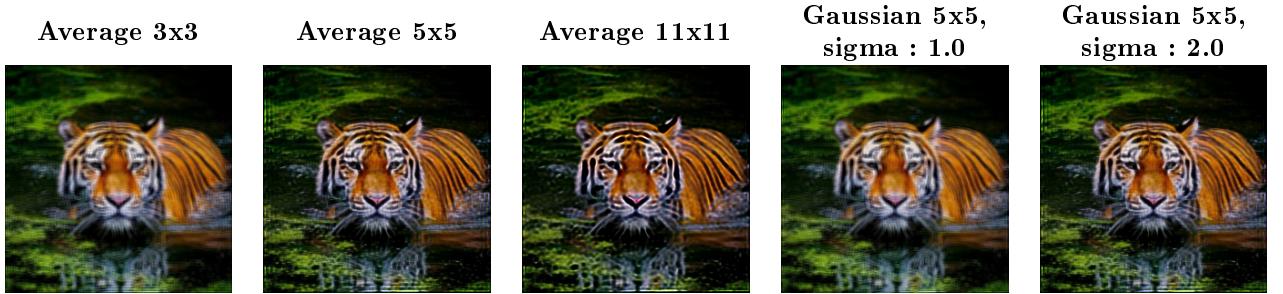
Images Défloutées - 30 Itérations, 25 Itérations PSF



Images Défloutées - 60 Itérations, 25 Itérations PSF



Images Défloutées - 120 Itérations, 25 Itérations PSF



3.4 Conclusion sur l'algorithme de Richardson-Lucy Aveugle

Grâce à l'application de l'algorithme de Richardson-Lucy aveugle, nous avons pu restaurer des images floues avec succès en ayant moins de contraintes sur la connaissance de la fonction de floutage. Toutefois, il est important de noter que l'algorithme est sensible au bruit et peut produire des artefacts si le bruit est trop important. Il est beaucoup plus sensible aux conditions initiales que l'algorithme de Richardson-Lucy classique, et une mauvaise initialisation peut entraîner une mauvaise restauration de l'image. La taille du noyau de convolution est également un paramètre critique qui peut affecter la qualité de la restauration.

4 Conclusion

Grâce à l'application de l'algorithme de Richardson-Lucy et de l'algorithme de Richardson-Lucy aveugle, nous avons pu restaurer des images floues. Dans le premier, nous devions connaître la fonction de floutage, tandis que dans le second, nous n'avions pas besoin de cette connaissance. Une simple initialisation de la PSF était suffisante pour restaurer l'image. Dans les résultats fournis, les images restaurées étaient de petite taille, mais l'algorithme peut être appliqué à des images de plus grande taille avec utilisation de bibliothèques de calculs parallèles qui permettent de réduire le temps de calcul.