

AGOUZAL NOUHAILA
ABDESSAMAD BADAoui
NASR ALLAH AGHELIA



Traitement d'images

TP2 : Filtrage non linéaire

MAY 2023

Contents

1	Estimation du bruit	3
2	Filtre médian	4
2.1	Estimation de la demie-taille du filtre adéquate	4
2.1.1	Bruit gaussien de moyenne nulle	4
2.1.2	Bruit poissonien	4
2.1.3	Bruit poivre et sel	5
2.1.4	Bruit speckle	5
2.2	Conclusion	6
3	Filtres adaptatifs	6
3.1	Filtre adaptatif récursif	6
3.1.1	Bruit gaussien de moyenne nulle	6
3.1.2	Bruit poissonien	7
3.1.3	Bruit poivre et sel	7
3.1.4	Bruit speckle	8
3.2	Conclusion	8
3.3	Méthodes à base de patch : moyenne non locale	9
4	Comparaison du PSNR et du temps CPU	9
5	Comparaison qualitative	11

1 Estimation du bruit

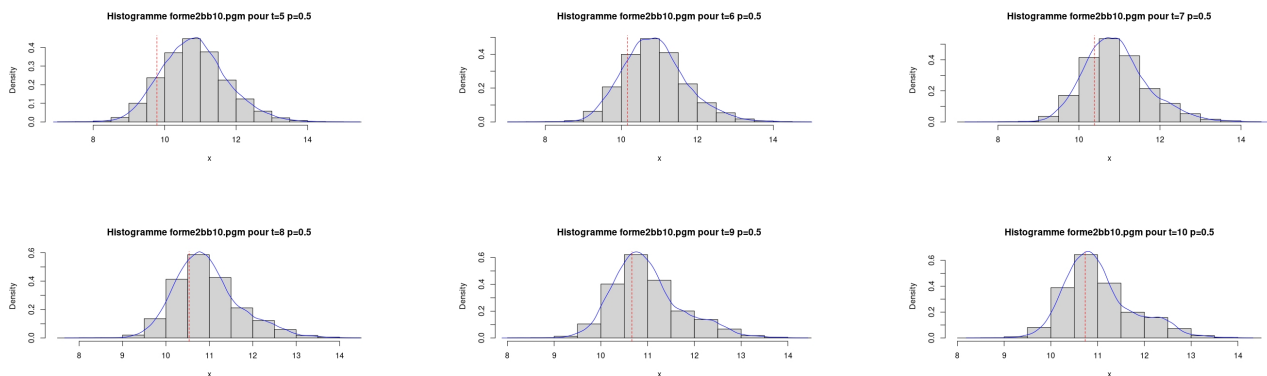
Le calcul de l'écart type local est implémentée dans "**ecart_type_bloc.c**". On se base sur un prolongement miroir pour la gestion des bords. La fonction "**ecart_type_bloc**" renvoie un vecteur qui contient les valeurs triées de tous les écarts-type locaux.

Dans **test_conv_discrete_miroir.c**, on affiche la valeur estimée du bruit sur la console pour une entrée, une taille de bloc **t** et une valeur du pourcentage **p** passés comme arguments en ligne de commande.

Pour **formes2bb10.pgm**, on trace les histogrammes pour différents valeurs de **t** et **p**.

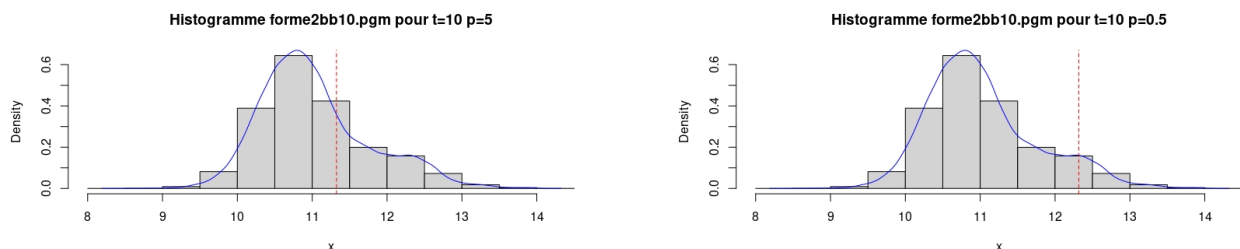
pour p=0.5:

Dans ce cas, on voit bien que le meilleur résultat est obtenu avec **t=6** pour cet exemple.



Maintenant, on fixe **t=10** et on teste avec **p=5** et **p=50**.

On constate que le résultat obtenu avec **p=0.5** est nettement meilleur.



Le choix des paramètres **p** et **t** adéquats n'est pas évident, ce qui nous amène à travailler avec les valeurs suggérées dans l'énoncé.

2 Filtre médian

L'algorithme "Median par histogramme" est implémenté dans **"filtre_median_fonction.c"**. **"filtre_median_fonction.c"** permet d'extraire les arguments de la ligne de commande: image d'entrée, image de sortie et demie-taille du filtre.

Néanmoins, les bords ne sont pas traités avec cet algorithme. Une option consiste à les ignorer complètement mais il faut aussi les ignorer lors de la comparaison entre les différents filtres. On définit donc une fonction **"filtre_median_psnr"** qui permet de calculer le PSNR sur une partie de l'image.

2.1 Estimation de la demie-taille du filtre adéquate

Toutes les comparaisons sont faites par rapport à **formes2g.pgm**.

2.1.1 Bruit gaussien de moyenne nulle

On trace la demie-taille du filtre qui maximise le PSNR en fonction de la variance du bruit.

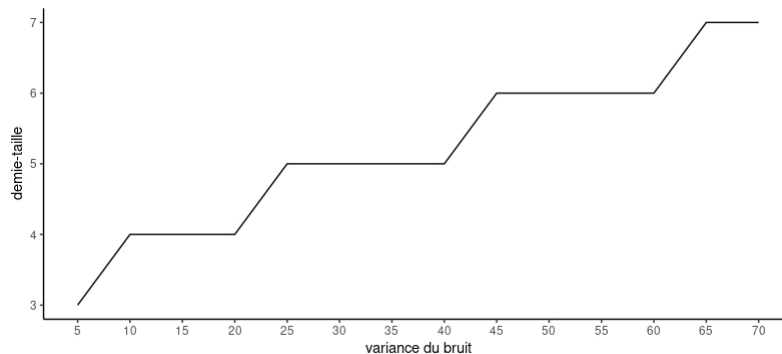


Figure 1: Demie taille du filtre adéquate pour le bruit gaussien

2.1.2 Bruit poissonien

On trace la demie-taille du filtre qui maximise le PSNR en fonction du numéro de réalisation.

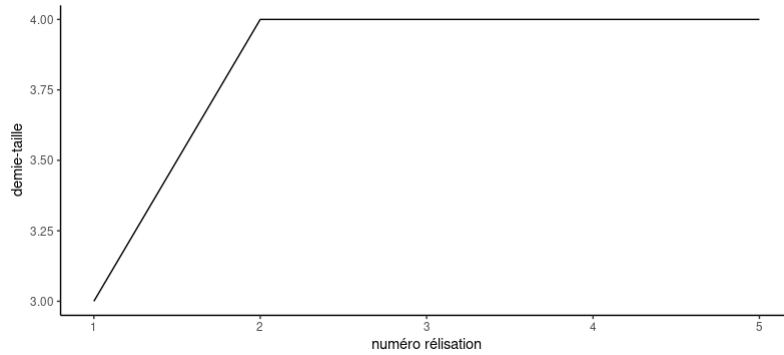


Figure 2: Demie taille du filtre adéquate pour le bruit poissonien

2.1.3 Bruit poivre et sel

On trace la demie-taille du filtre qui maximise le PSNR en fonction de la densité de pixels défectueux.

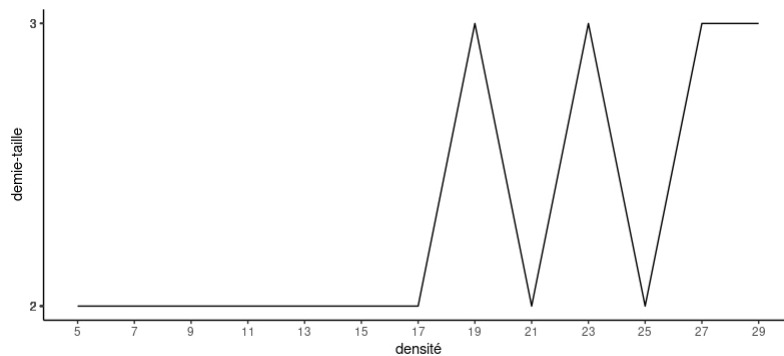


Figure 3: Demie taille du filtre adéquate pour le bruit poivre et sel

2.1.4 Bruit speckle

On trace la demie taille du filtre qui maximise le PSNR en fonction de la variance.

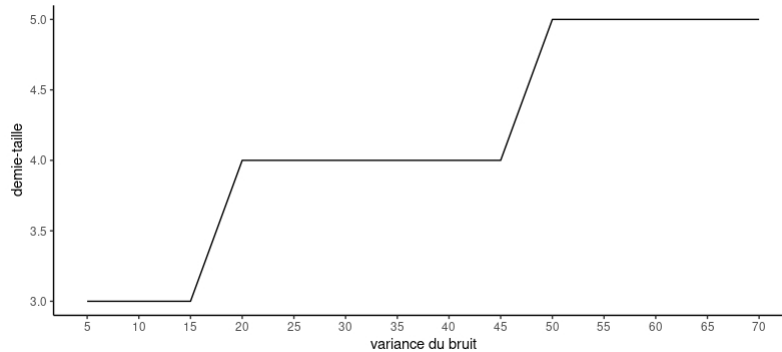


Figure 4: Demie taille du filtre adéquate pour le bruit speckle

2.2 Conclusion

Pour le bruit gaussien et le bruit speckle, on voit bien que pour des niveaux de bruit plus élevés, il faut utiliser un filtre médian avec une taille plus grande pour obtenir de meilleurs résultats de débruitage. Cela est dû au fait que les niveaux de bruit plus élevés peuvent entraîner des variations plus importantes entre les pixels voisins, ce qui nécessite une fenêtre de filtrage plus grande pour capturer les structures et réduire les effets du bruit.

3 Filtres adaptatifs

3.1 Filtre adaptatif récursif

Ce filtre est implémenté dans "`filtre_adaptatif_recuratif.c`". `test_filtre_adaptatif_recuratif.c` permet de récupérer l'image d'entrée et de sortie.

On trace la courbe qui donne le meilleur paramètre k (coefficient de lissage) en fonction des caractéristiques de chaque bruit pour **100** et **200** itérations.

3.1.1 Bruit gaussien de moyenne nulle

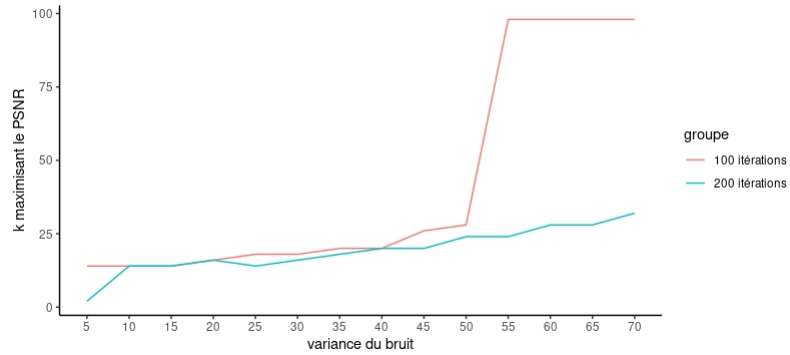


Figure 5: Paramètre k adéquat pour le bruit gaussien

3.1.2 Bruit poissonien

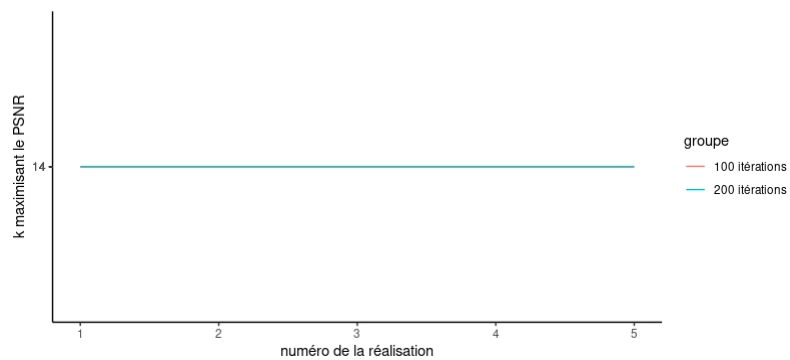


Figure 6: Paramètre k adéquat pour le bruit poissonien

3.1.3 Bruit poivre et sel

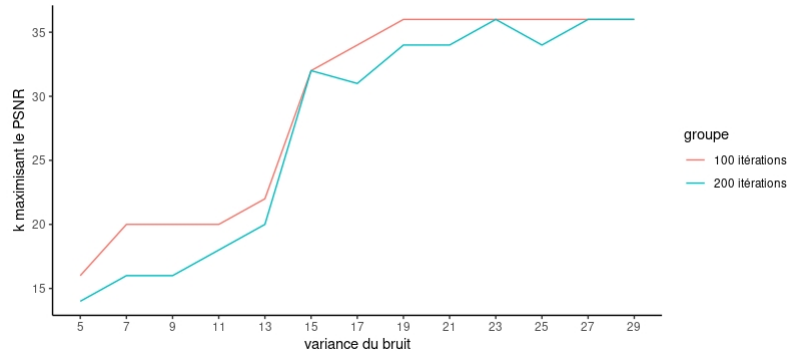


Figure 7: Paramètre k adéquat pour le bruit poivre et sel

3.1.4 Bruit speckle

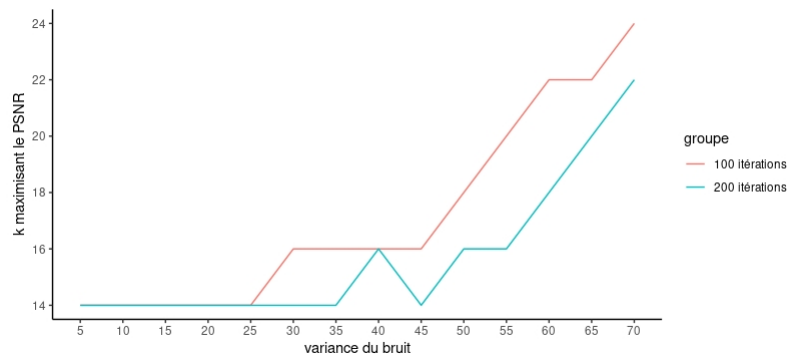


Figure 8: Paramètre k adéquat pour le bruit speckle

3.2 Conclusion

Le choix des paramètres k et du **nombre d'itérations** pour un filtre adaptatif récursif dépendra des caractéristiques spécifiques de chaque type de bruit. Il n'y a pas de valeurs universelles qui fonctionnent parfaitement pour tous les cas. Pour le bruit gaussien, la valeur de k qui maximise le PSNR augmente au fur et à mesure que la variance du bruit augmente. On constate également que si on augmente le nombre d'itérations, la valeur de k nécessaire diminue. La même remarque s'applique pour le bruit speckle. Pour le bruit poissonien, on voit bien que pour les différentes réalisations on a la même valeur de k même si on augmente le nombre d'itérations.

3.3 Méthodes à base de patch : moyenne non locale

Cet algorithme est implémenté dans `non_local_means.c`. `test_moyenne_non_locale.c` permet de récupérer l'image d'entrée et de sortie à partir de la ligne de commande.

Cette méthode a plusieurs paramètres et son exécution prend pas mal de temps. Par conséquent, l'estimation des paramètres adéquats n'est pas évident. On propose donc d'estimer le niveau de bruit avec le premier algorithme et d'utiliser la valeur estimée pour trouver les bons paramètres grâce au tableau donné dans l'énoncé.

4 Comparaison du PSNR et du temps CPU

On compare ces filtres pour différents types de bruit. On choisit l'image `formes2bb30.pgm` pour le bruit gaussien, `formes2p3.pgm` pour le bruit poissonien, `formes2sp15.pgm` pour le bruit poivre et sel et finalement `formes2s35.pgm` pour le bruit speckle.

Cette comparaison est implémentée dans `"comparaison_methodes_non_lineaire.c"`

Pour le PSNR, on voit bien que le "filtre médian" donne le meilleur PSNR pour tous les types

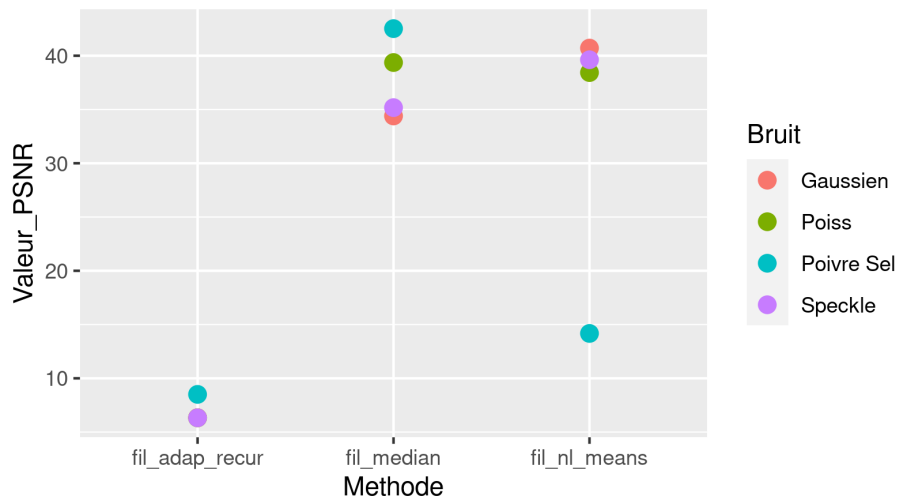


Figure 9: Comparaison PSNR

de bruit, tandis que le "filtre à base de patch" ne donne pas un bon PSNR pour le bruit "Poivre et Sel". En se basant sur le PSNR, on classe ces filtres comme suit:

1 - Filtre médian.

- 2 - Filtre à base de patch.
- 3 - Filtre adaptatif récursif.

Maintenant, on compare les trois filtres en se basant sur le temps CPU.

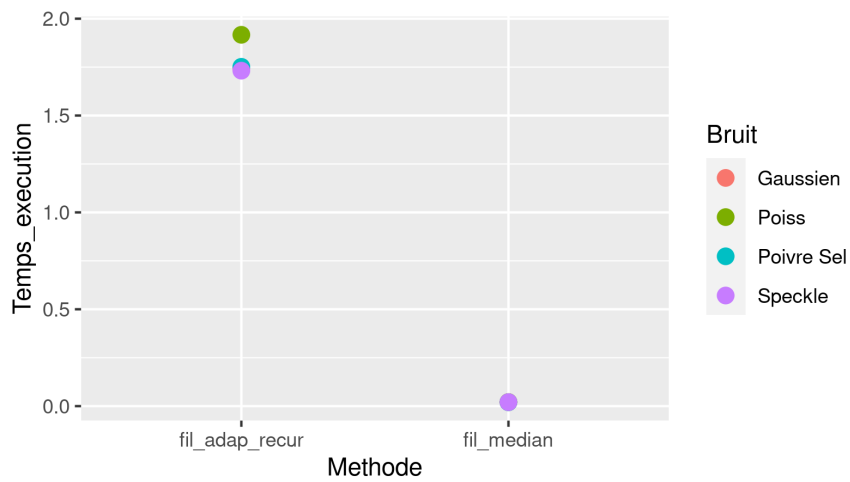


Figure 10: Comparaison temps CPU

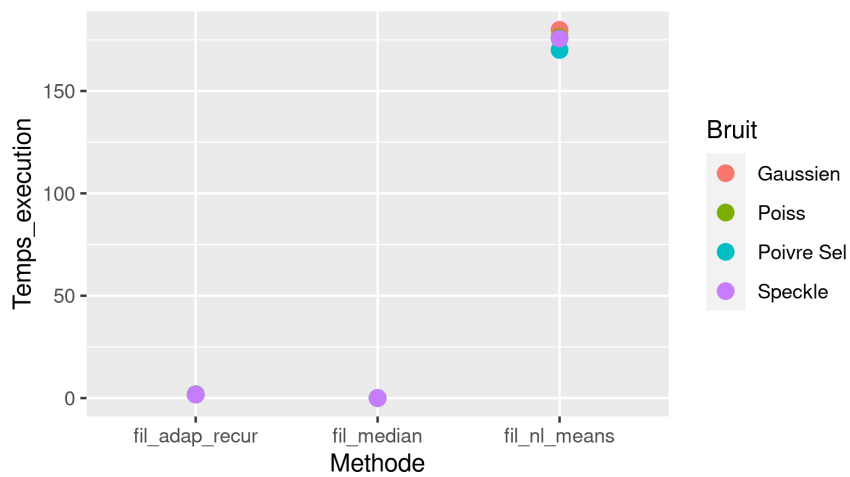


Figure 11: Comparaison temps CPU

La complexité temporelle du "filtre à base de patch" est très grande. Celle du filtre adaptatif récursif est généralement plus grande que celle du filtre médian et elle dépend fortement du nombre d'itérations.

En se basant sur le temps CPU, on classe ces filtres comme suit:

- 1 - Filtre médian.
- 2 - Filtre adaptatif récursif.
- 3 - Filtre à base de patch.

Il faut donc trouver un compromis entre ces deux critères lors du choix du filtre.

5 Comparaison qualitative

Filtre médian :

- Avantages : Le filtre médian est très efficace pour éliminer les impulsions et le bruit de type "salt-and-pepper". Il préserve avec précision les contours de l'image et les détails importants.
- Inconvénients : L'image peut perdre de sa netteté ou devenir floue. Il est moins efficace pour éliminer les autres types de bruit, comme le bruit gaussien.

Filtre adaptatif récursif :

- Avantages : Un filtre adaptatif récursif peut adapter automatiquement son comportement au contenu local de l'image. Il offre un bon compromis entre réduction du bruit et préservation des détails.
- Inconvénients : Il peut y avoir une légère perte de netteté dans les zones à fort contraste. Il est sensible aux changements soudains du contenu de l'image et peut introduire des artefacts indésirables.

NL-Means (Non-Local Means) :

- Avantages : Le NL-Means est très efficace pour réduire le bruit gaussien, y compris le bruit de bas niveau. Il utilise des informations provenant de régions similaires dans l'image pour estimer et supprimer de manière adaptative le bruit.
- Inconvénients : Le NL-Means peut être relativement lent en raison de son calcul intensif, en particulier pour les images de grande taille. Il peut également introduire un léger flou dans les régions à fort contraste.

En résumé, les filtres médians sont idéaux pour supprimer le bruit impulsif, tandis que les filtres adaptatifs récursifs offrent un bon équilibre entre la réduction du bruit et la préservation des détails. La NL-Means est particulièrement efficace pour réduire le bruit gaussien, mais peut être lente et introduire un peu de flou. Le choix de la méthode dépendra du type de bruit présent dans l'image et des compromis souhaités entre réduction du bruit et préservation des détails.