

TP2 : Filtrage non linéaire.

Abdelmajid Koubach
Abdellah Belaid

Group 2

April 29, 2022

Abstract

L'objectif du TP est de tester et comparer différentes méthodes de lissages non linéaires.

1 Estimation de bruit

La méthode proposée par le sujet exige de calculer l'écart type à partir d'un très faible pourcentile de l'histogramme des écarts-types locaux. Cette valeur d'écart-type est ensuite corrigée d'un facteur qui dépend de la taille du bloc sur lequel la variance est calculée et du pourcentile utilisé.

Cette méthode est implémentée dans le fichier **algorithme.c** principalement par la fonction "**estimation_bruit(...)**" qui retourne l'écart type corrigé.

Voici quelques exemples sur quelques images:

Exemple de bruit blanc avec $\sigma = 70$ avec taille de bloc de 10 :

```
$ ./main Archive/formes2bb70.pgm
L'ecart-type du pourcentile 0.5 est : 67.691911

$ ./main Archive/formes2bb70.pgm
L'ecart-type du pourcentile 50.0 est : 71.141503
```

```
$ ./main Archive/formes2bb70.pgm  
L'ecart-type du pourcentile 20.0 est : 70.203906
```

Exemple de bruit blanc avec $\sigma = 20$ avec taille de bloc de 10 :

```
$ ./main Archive/formes2bb20.pgm  
L'ecart-type du pourcentile 0.5 est : 20.266990
```

et avec une taille de bloc de 5 :

```
$ ./main Archive/formes2bb20.pgm  
L'ecart-type du pourcentile 0.5 est : 17.470499
```

On remarque que la taille des blocs et la valeur de pourcentile dépendent de l'écart type réel du bruit.

2 Les filtres non linéaire

2.1 Implémentation

2.1.1 Filtre médian

Il consiste à remplacer le pixel central par la valeur médiane du voisinage (taille impaire) dans lequel il se trouve.

Ce filtre est implémenté dans le fichier **filtre_median.c**

2.1.2 Filtre adaptatif récursif

le filtrage consiste d'une part à réduire la variance du bruit dans les zones homogènes, d'autre part à conserver les fronts correspondant à bords des objets.

Ce filtre est implémenté dans le fichier **filtre_recuratif.c** .

2.1.3 Filtre adaptatif : méthode à base de patch

L'idée du filtre **Nlmeans** est d'utiliser la redondance présente dans la plupart des images, de ce fait le SNR est amélioré, mais les contours sont atténués.

Ce filtre est implémenté dans le fichier **filtre_patch.c**.

2.2 Comparaison

Pour cela on utilise le fichier **"performance.c"** qui compare le temps d'exécution et la qualité de chaque filtre.

La comparaison se fait avec l'image original **"forme2.pgm"** ou **"forme2g.pgm"**.

On commence avec le bruit blanc avec $\sigma = 70$ et $\sigma = 20$:

```
$ → ./perf Archive/formes2bb20.pgm Archive/formes2.pgm 20.0
sigma=20.000000    h=0.400000    patch=2    region=10
Filtre median :: Time = 0.171276, PSNR (Quality) = 18.960101.
recursive k = 13
Filtre recursif :: Time = 0.826816, PSNR (Quality) = 30.365828.
Filtre patch :: Time = 28.732659, PSNR (Quality) = 32.305548.
```

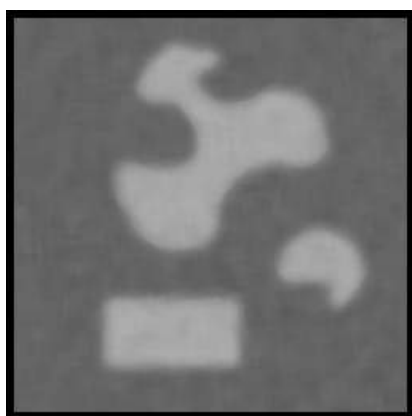


Figure 1: filtrage médian.



Figure 2: filtrage récursif.

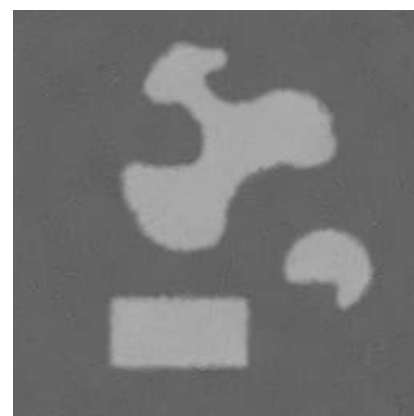


Figure 3: filtrage Nlmeans.

```
$ ./perf Archive/formes2bb70.pgm Archive/formes2.pgm 70.0
sigma=70.000000    h=0.350000    patch=4    region=10
Filtre median :: Time = 0.135152, PSNR (Quality) = 18.630022.
recursive k = 20
Filtre recursif :: Time = 1.186938, PSNR (Quality) = 27.738908.
Filtre patch :: Time = 228.163598, PSNR (Quality) = 27.554113.
```

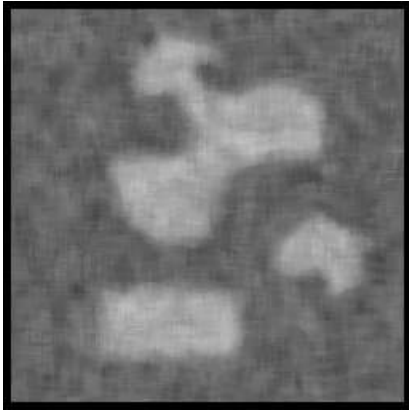


Figure 4: filtrage médian.

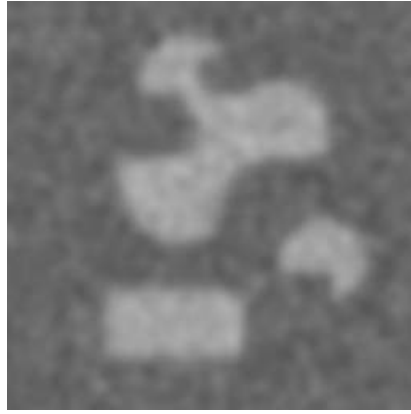


Figure 5: filtrage récursif.

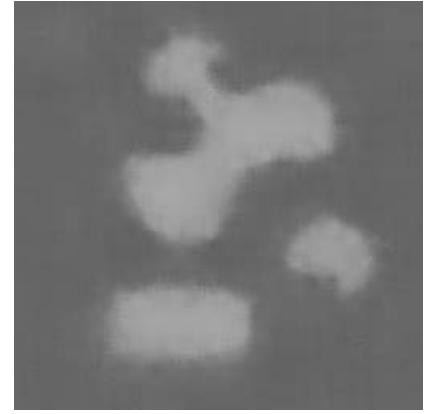


Figure 6: filtrage Nlmeans.

On remarque que le troisième algorithme est largement plus lent que les 2 autres or avec un **PSNR** qui est bien plus grand pour $\sigma = 20$ mais ce n'est pas le cas pour $\sigma = 70$, ce qui rend le deuxième algorithme (récursif) beaucoup plus favorable grâce à sa performance même si il est plus lent que le filtre médian.

Or qualitativement, le résultat du Nlmeans apparaît mieux aux yeux, aussi il faut mentionner que le temps d'exécution de cet algorithme dépend clairement de sigma en remarquant la différence de temps entre $\sigma = 20$ et $\sigma = 70$.

Maintenant avec le bruit poivre et sel , pour cela il faut d'abord estimer le sigma pour les images **formes2sp11** et **formes2sp29**:

```
$ ./main Archive/formes2sp11.pgm
L'ecart-type du pourcentile 0.5 est : 22.110862
$ ./perf Archive/formes2sp11.pgm Archive/formes2.pgm 22.0
22.000000 0.400000
Filtre median :: Time = 0.230231, PSNR (Quality) = 19.083503.
recursive k = 18
Filtre recursif :: Time = 1.138375, PSNR (Quality) = 27.574097.
Filtre patch :: Time = 20.553455, PSNR (Quality) = 16.218770.
```



Figure 7: filtrage médian.



Figure 8: filtrage récursif.

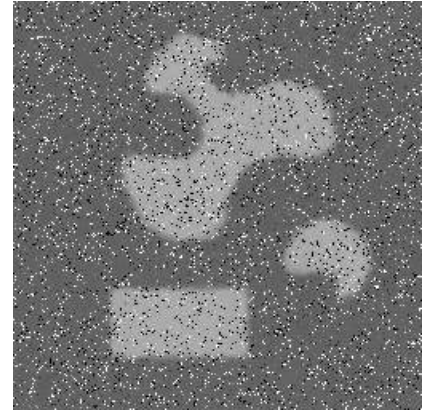


Figure 9: filtrage Nlmeans.

```
$ → ./main Archive/formes2sp29.pgm
L'ecart-type du pourcentile 0.5 est : 52.718897
$ ./perf Archive/formes2sp29.pgm Archive/formes2.pgm 52.0
sigma=52.000000 h=0.350000 patch=4 regi
Filtre median :: Time = 0.167925, PSNR (Quality) = 19.070503.
recursive k = 22
Filtre récursif :: Time = 1.289152, PSNR (Quality) = 23.967012.
Filtre patch :: Time = 191.592801, PSNR (Quality) = 13.837260.
```



Figure 10: filtrage médian.

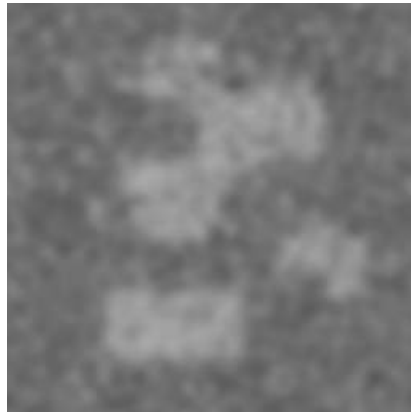


Figure 11: filtrage récursif.

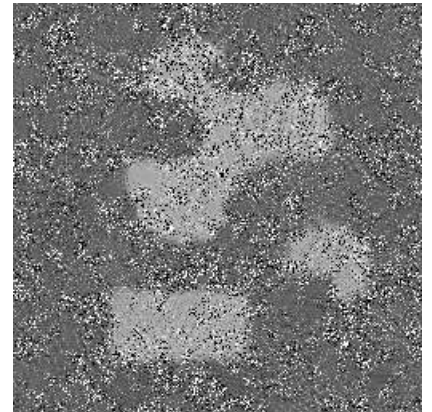


Figure 12: filtrage Nlmeans.

On remarque que le filtre **Nlmeans** est moins performant pour ce type de bruit que dans le cas précédent avec l'apparition de bruit, on a aussi un filtre médian qui donne un résultat très bon même si le **PSNR** est atténué à cause des bords qui sont supprimés, et un filtre récursif qui comme dans le cas précédent donne un résultat satisfaisant.

Alors qualitativement le filtre médian est le meilleur choix pour ce type de bruit.

3 Exemple d'image réelle

Pour tester nos filtres sur des images de la vie réelle , on a choisit l'image bruitée suivante:



Figure 13: Image bruitée.

Puisqu'on a pas d'image original non bruitée, le **PSNR** ne signifie pas grand chose dans ce cas.

On estime d'abord le bruit de cette image :

```
$ ./main reel.pgm
L'ecart-type du pourcentile 0.5 est : 17.558026
```

Alors avec un $\sigma = 17.55$ on exécute notre algorithme:

```
$ ./perf reel.pgm reel.pgm 17.55
sigma=17.550000          h=0.400000          patch=2          regi
Filtre median :: Time = 0.919005, PSNR (Quality) = 18.597325.
recursive k = 31
Filtre recursif :: Time = 11.062210, PSNR (Quality) = 21.900475.
Filtre patch :: Time = 170.859240, PSNR (Quality) = 24.608009.
```



Figure 14: filtrage médian.



Figure 15: filtrage récursif.



Figure 16: filtrage Nlmeans.

Pour les images réelles, on remarque que le filtre médian donne des images avec plus de bruit mais plus de détails or le filtre récursif donne une image lisse avec moins de bruit mais il sacrifie pour cela beaucoup de détails , et enfin le filtre patch semble être le plus équilibré avec des détails conservés comme le médian mais aussi très peu de bruit comme le récursif.