

AGOUZAL NOUHAILA
ABDESSAMAD BADAoui
NASR ALLAH AGHELIA



Traitement d'images

TP1 : Lissage et détection de contours

MARCH 2023

Contents

1	Lissage linéaire	3
1.1	FFT, filtrage fréquentiel et convolution	3
1.1.1	Traçage de l'évolution du PSNR	3
1.1.2	Conclusion	7
1.2	Convolution spatiale	7
1.2.1	Calcul de EQM	8
1.2.2	La loi empirique $W(\sigma)$	11
1.3	Complexité et comparaison des 2 méthodes	12
2	Détection de contours	13
2.1	Opérateurs différentiels du premier ordre	13
2.2	Opérateurs différentiels du deuxième ordre	16
2.3	Comparaison	19

1 Lissage linéaire

1.1 FFT, filtrage fréquentiel et convolution

Le filtrage fréquentiel gaussien est implémenté dans "fonction_filtrage_gaussien.c". "filtrage_gaussien.c" permet de tracer le PSNR et créer l'image lissée qui correspond aux PSNR maximal.

1.1.1 Traçage de l'évolution du PSNR

Dans cette partie, on trace l'évolution du PSNR entre l'image débruitée par un filtre gaussien et l'image théorique "forme2g.pgm".

Bruit gaussien de variance nulle

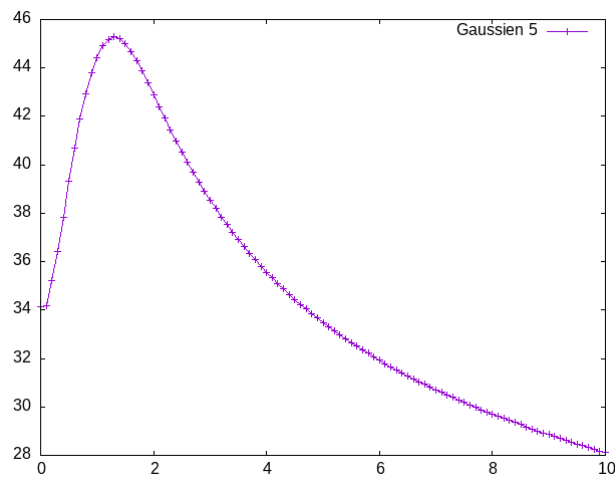


Figure 1: PSNR pour un bruit gaussien faible

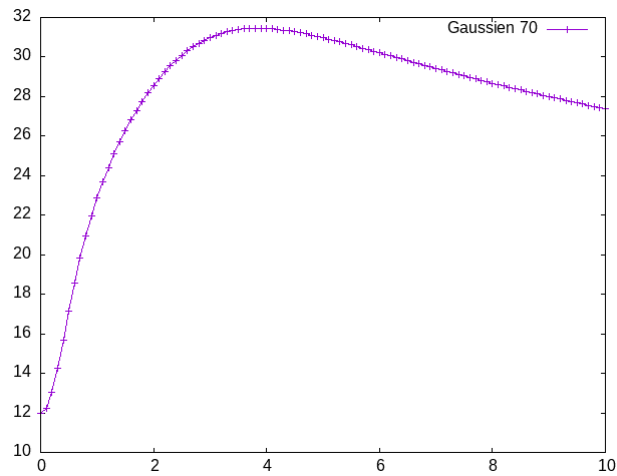


Figure 2: PSNR pour un bruit gaussien fort

On voit bien que pour une image débruitée avec un bruit gaussien faible (variance = 5), le FFT permet un PSNR maximal de valeur 45.3. Pour un bruit fort (variance = 70), la valeur maximale de PSNR est réduite à 31.45. On peut donc conclure que le PSNR maximale diminue au fur et à mesure que l'intensité du bruit augmente.

Bruit poissonien

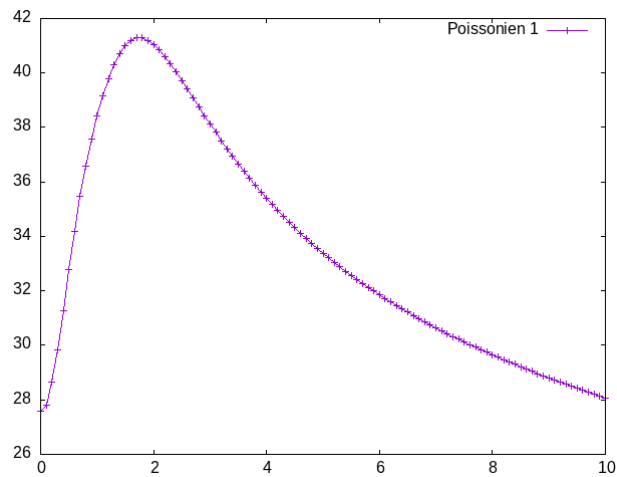


Figure 3: PSNR pour un bruit poissonien (première réalisation)

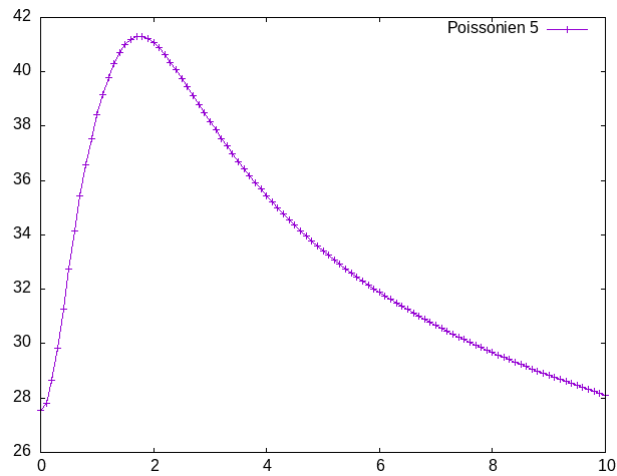


Figure 4: PSNR pour un bruit poissonien (deuxième réalisation)

Les deux figures ne représentent que deux réalisations différentes pour un bruit poissonien. Le niveau de bruit n'intervient pas dans ce cas. Dans les deux cas, on a le même PSNR maximal qui vaut 41.3.

Bruit poivre et sel

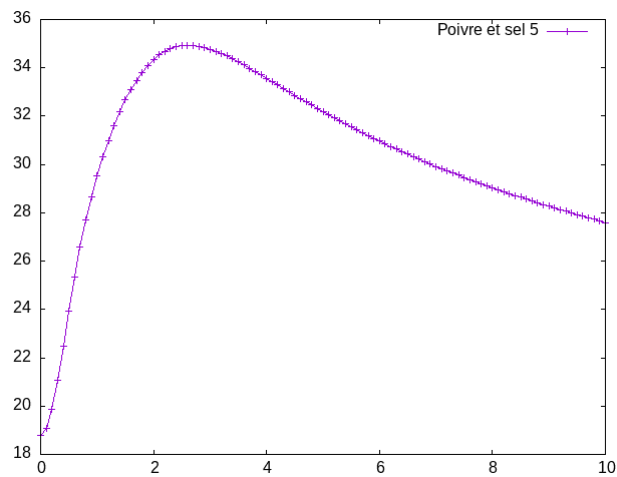


Figure 5: PSNR pour un bruit poivre et sel faible

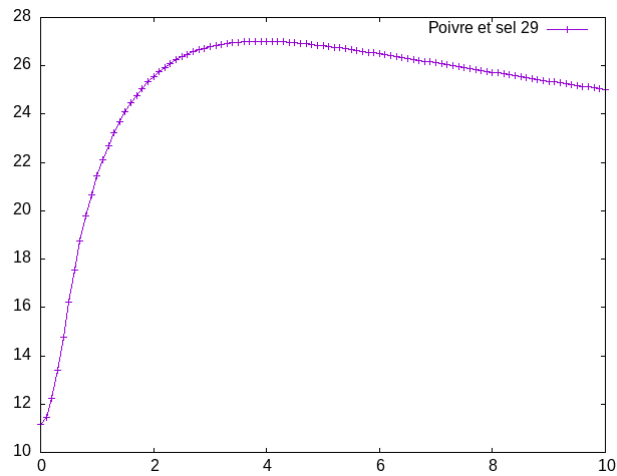


Figure 6: PSNR pour un bruit poivre et sel fort

Pour ce type de bruit, le PSNR dépend de la densité de pixels défectueux. Pour une densité faible, le PSNR maximal vaut 34.9, alors que pour une densité forte, celui-ci vaut 27. Le PSNR maximal diminue au fur et à mesure que cette densité augmente.

Bruit speckle

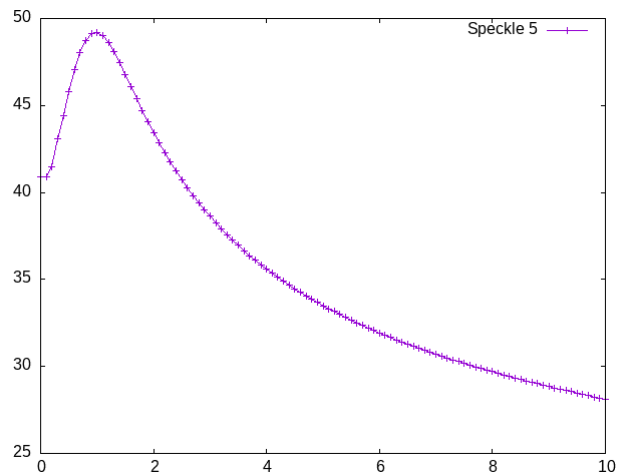


Figure 7: PSNR pour un bruit speckle faible

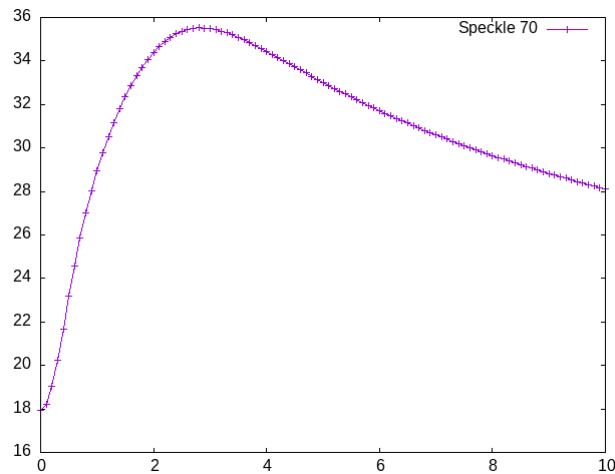


Figure 8: PSNR pour un bruit speckle fort

Pour le bruit Speckle, le PSNR maximal vaut 49.2 pour une densité de bruit faible, alors que pour une densité fort, il vaut 35.5. De même, on constate que le PSNR maximal diminue avec l'augmentation de la densité du bruit.

1.1.2 Conclusion

D'après les courbes précédentes, on constate que:

- Pour un type de bruit donné, la valeur de σ qui correspond à la valeur maximale du PSNR augmente si la densité de ce bruit augmente.
- Le filtre gaussien est efficace pour le bruit Gaussien de variance nulle et le bruit Speckle. Il donne aussi des résultats satisfaisants dans le cas du bruit Poissonien. Mais il paraît moins efficace pour le bruit Poivre et sel. Ceci peut être expliqué par le fait que ce bruit est de type impulsionnel.

1.2 Convolution spatiale

La convolution gaussienne par un masque 2D et celle par un masque séparable sont implémentées dans "fonction_convolution_spaciale.c". "convolution_spaciale.c" permet de calculer l'écart quadratique moyen pour ces deux types de convolution.

1.2.1 Calcul de EQM

On trace l'écart quadratique moyen (EQM) entre la convolution par filtrage fréquentiel par FFT et celle obtenue par un masque spatial en fonction de la taille du masque.

Pour $\sigma = 1$:

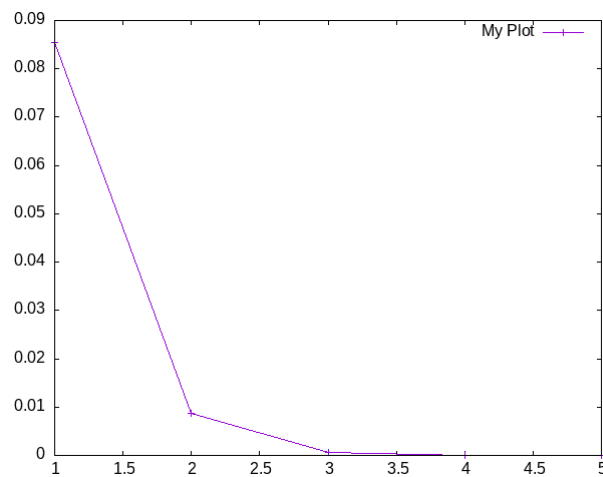


Figure 9: EQM pour l'image formes2g.pgm

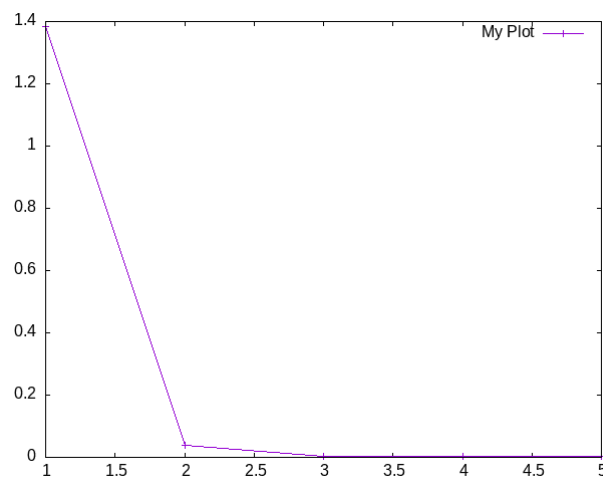


Figure 10: EQM pour l'image radio1024.pgm

Pour $\sigma = 5$:

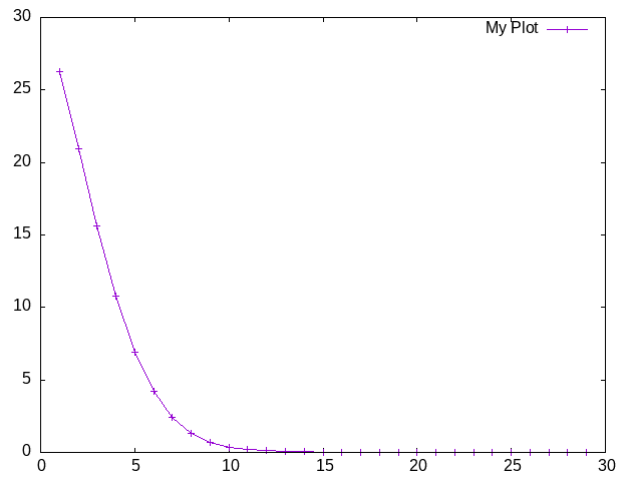


Figure 11: EQM pour l'image formes2g.pgm

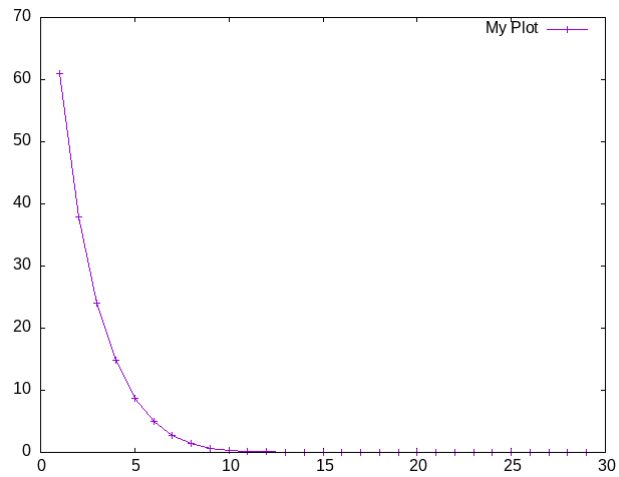


Figure 12: EQM pour l'image radio1024.pgm

Pour $\sigma = 10$:

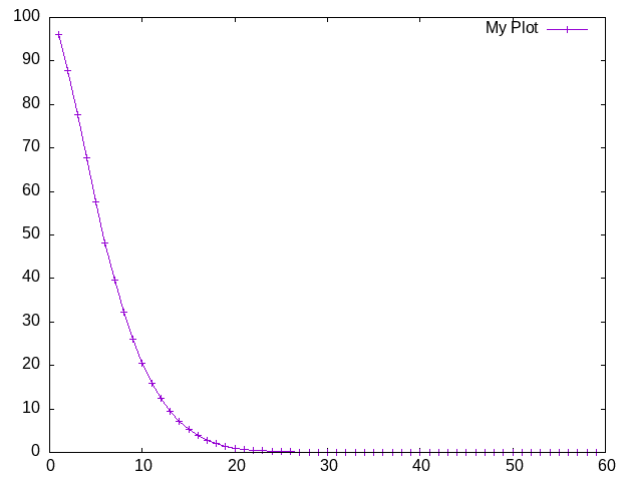


Figure 13: EQM pour l'image formes2g.pgm

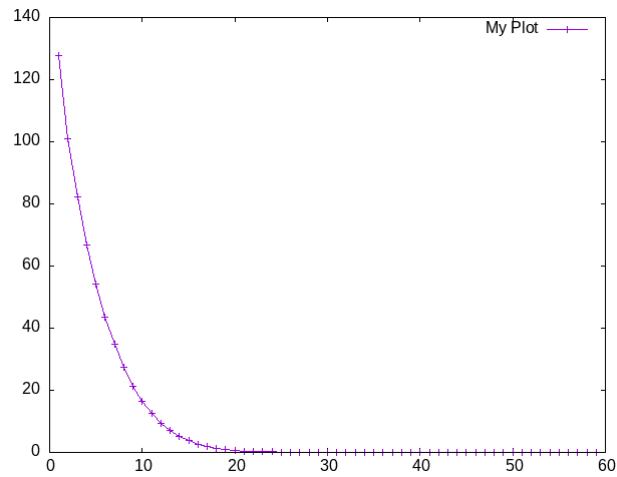


Figure 14: EQM pour l'image radio1024.pgm

On remarque que la taille du masque requise pour avoir une erreur quadratique nulle est proportionnelle à la valeur de σ .

1.2.2 La loi empirique $W(\sigma)$

On détermine approximativement la loi empirique $W(\sigma)$ pour laquelle les résultats sont identiques entre le filtrage fréquentiel par FFT et le filtrage spatial par un masque. Pour ce faire, on fixe une valeur de σ , et on cherche la taille du masque qui minimise l'erreur quadratique moyen entre ces deux filtrages. Ceci est implémenté dans "loi_W_taille_masque.c".

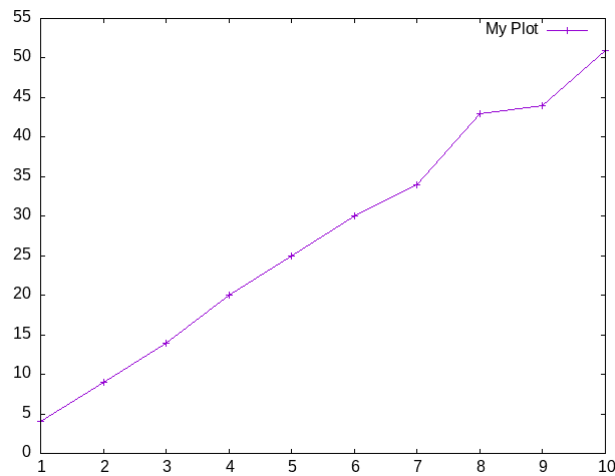


Figure 15: $W(\sigma)$ pour l'image formes2g.pgm

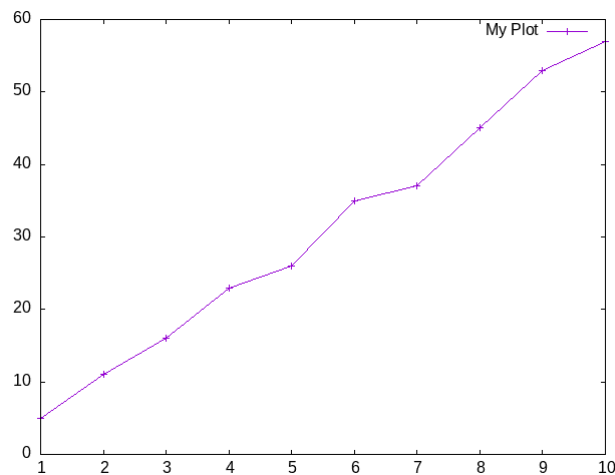


Figure 16: $W(\sigma)$ pour l'image radio1024.pgm

D'après ces deux courbes, on peut faire l'approximation suivante: $W(\sigma) = 4 \times \sigma - 1$

1.3 Complexité et comparaison des 2 méthodes

La comparaison des temps de calcul des 3 implémentations pour des filtrages gaussiens de paramètre croissant de 1 à 10 pour l'image formes2g.pgm donne le résultat suivant :

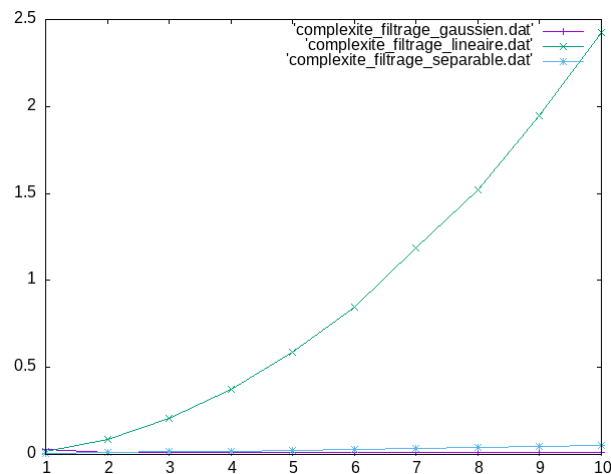


Figure 17: Le temps d'exécution des 3 méthodes en fonction de sigma (formes2g.png)

Pour l'image radio1024.pgm :

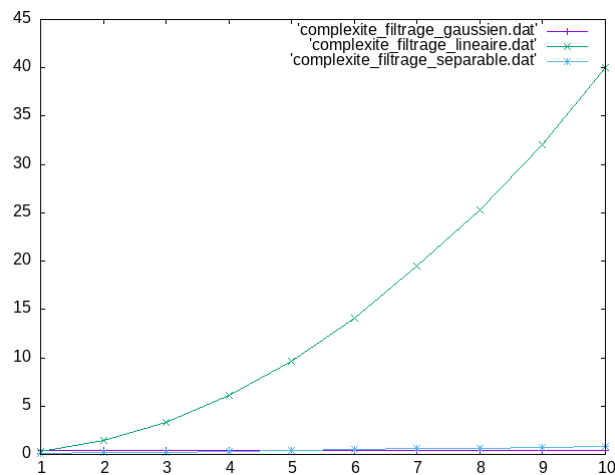


Figure 18: Le temps d'exécution des 3 méthodes en fonction de sigma (radio1024.png)

-Pour les images de grande taille, la méthode de la transformée de Fourier rapide (FFT) est souvent préférable car elle permet de réduire considérablement le temps de calcul nécessaire. En

effet, l'opération de convolution dans l'espace est très coûteuse en terme de temps de traitement, surtout pour des images volumineuses.

- La méthode de filtrage linéaire par masque 2D n'est pas adaptée pour les images de grande taille car elle est très coûteuse en terme de temps de calcul.

- La méthode de filtrage séparable est très efficace en comparaison avec les autres méthodes pour des sigma petites (c'est à dire lorsque'on utilise des tailles de masques petites), et pour des valeurs de sigma plus grandes, la méthode reste toujours efficace mais le filtrage gaussien par FFT devient meilleur.

2 Détection de contours

2.1 Opérateurs différentiels du premier ordre

Pour calculer le gradient, on a adopté la méthode du "Gradient lissé séparable" qui est implémentée dans "gradient_lissé_séparable.c". La figure suivante montre le résultat de ce calcul.

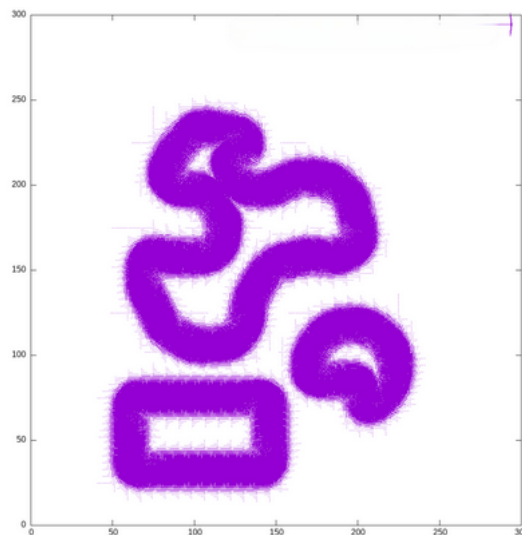


Figure 19: Les vecteurs gradient pour formes2g.pgm

Cet opérateur est implémenté dans "opérateur_premier_ordre.c"

Impact de la valeur de σ

On fixe $s_{bas} = 10$ et $s_{haut} = 20$ et on raisonne sur le même niveau de bruit.

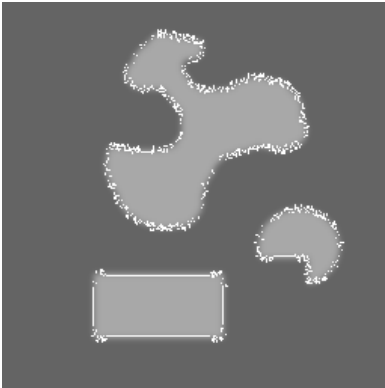


Figure 20: Détection pour formes2g.pgm pour sigma=1

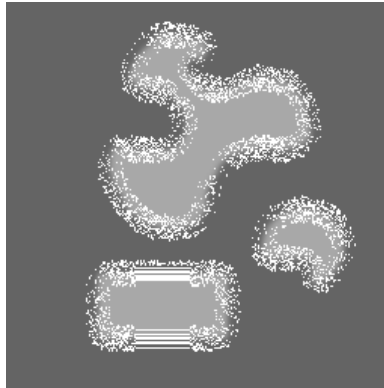


Figure 21: Détection pour formes2g.pgm pour sigma=5

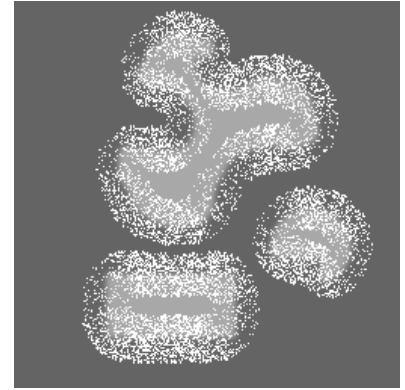


Figure 22: Détection pour formes2g.pgm pour sigma=10

Le sigma du filtre de lissage gaussien contrôle la variance de la fonction de Gauss utilisée pour calculer le filtre. Plus le sigma est grand, plus le flou est important car la fonction de Gauss est plus étalée. En conséquence, les contours détectés seront plus larges et flous, ce qui peut être adapté pour les contours larges et peu précis. Cependant, en diminuant la valeur de sigma, la détection des contours sera plus précise et moins floue, ce qui est plus approprié pour les contours fins.

Impact du seuillage

On fixe $\sigma = 5$ et $s_{bas} = 10$ et on raisonne sur le même niveau de bruit.

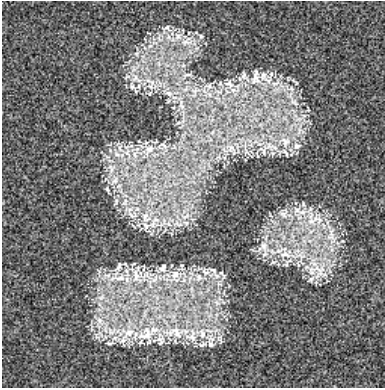


Figure 23: Détection pour un seuil haut=20

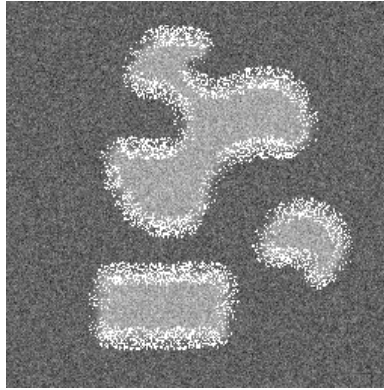


Figure 24: Détection pour un seuil haut=30

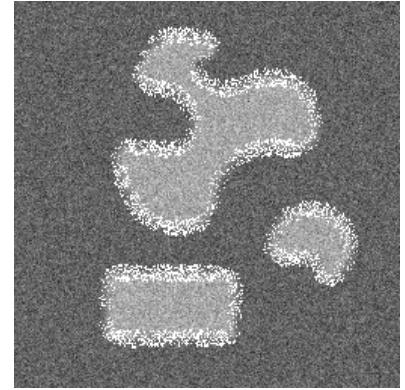


Figure 25: Détection pour un seuil haut=40

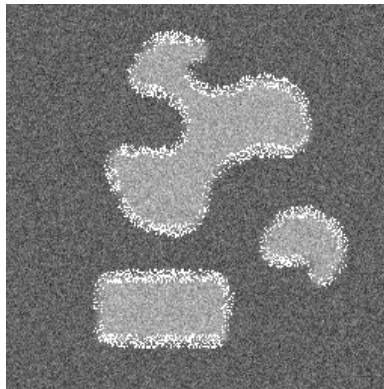


Figure 26: Détection pour un seuil haut=50

En ajustant le seuil de détection, on peut influencer la quantité de gradients considérés comme des contours. Si le seuil est augmenté, moins de gradients seront considérés comme des contours, ce qui peut réduire la sensibilité et la précision de la détection. Au contraire, si le seuil est diminué, davantage de gradients seront inclus dans la détection, ce qui peut augmenter la sensibilité, mais également introduire plus de bruit dans le résultat final.

Impact du niveau de bruit

On fixe $s_{bas} = 10$, $s_{haut} = 20$ et $\sigma = 5$.

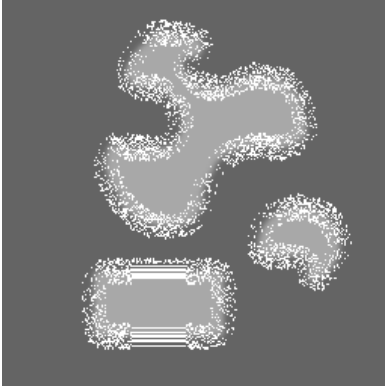


Figure 27: Détection pour formes2g.pgm

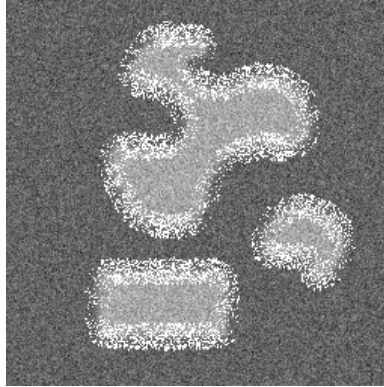


Figure 28: Détection pour formes2bb20.pgm

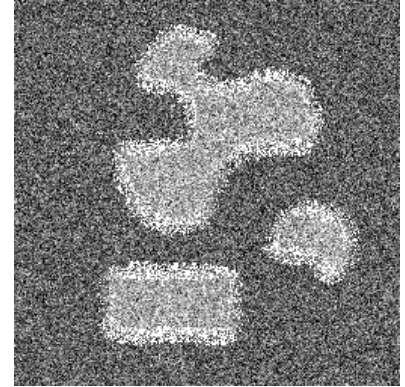


Figure 29: Détection pour formes2bb50.pgm

la détection de contours avec cette méthode n'est pas sensible aux bruit. Ceci est dû au fait que le lissage a réduit considérablement cette sensibilité en réduisant le bruit dans l'image avant cette détection.

2.2 Opérateurs différentiels du deuxième ordre

Pour cet opérateur, on a implémenté deux méthodes pour le calcul du laplacien. La première est celle par convolution spatiale (implémenté dans "laplacien_gaussien.c" et le main correspondant est "opérateur_deuxieme_ordre.c) et la deuxième est par FFT (dans "laplacien_lissage_fft.c").

Laplacien par FFT pour formes2bb20.pgm

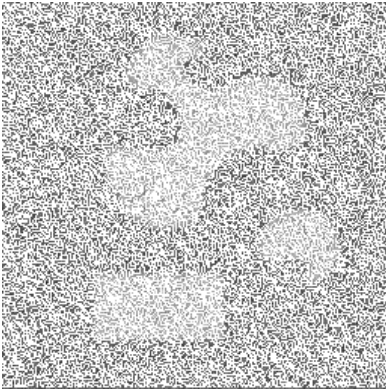


Figure 30: FFT pour formes2bb20.pgm sigma=1

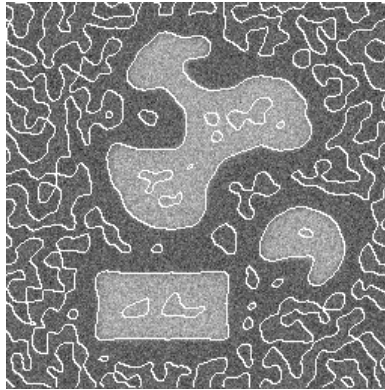


Figure 31: FFT pour formes2bb20.pgm sigma=5

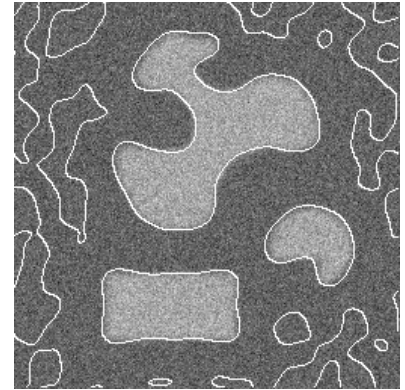


Figure 32: FFT pour formes2bb20.pgm sigma=10

Laplacien par convolution spatiale pour formes2bb20.pgm

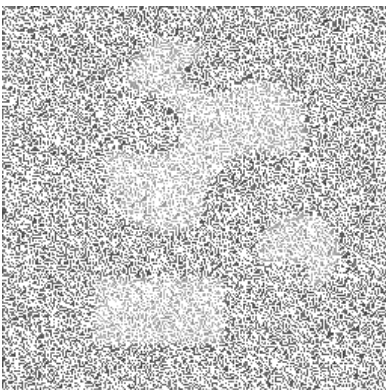


Figure 33: Laplacien gaussien formes2bb20.pgm pour sigma=1

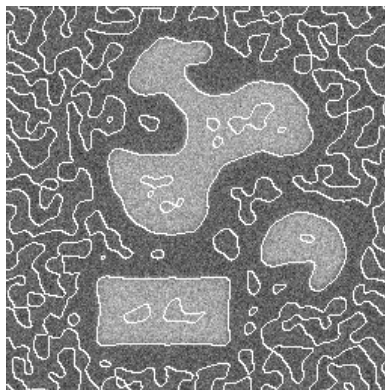


Figure 34: Laplacien gaussien formes2bb20.pgm pour sigma=5

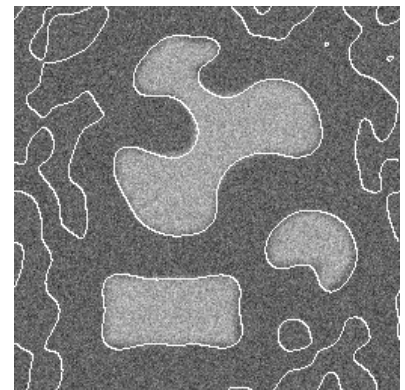


Figure 35: Laplacien gaussien formes2bb20.pgm pour sigma=10

Laplacien par FFT pour formes2g.pgm

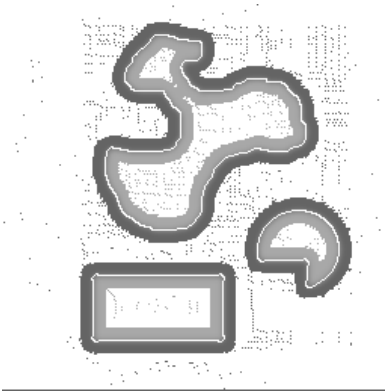


Figure 36: FFT formes2g.pgm pour sigma=1

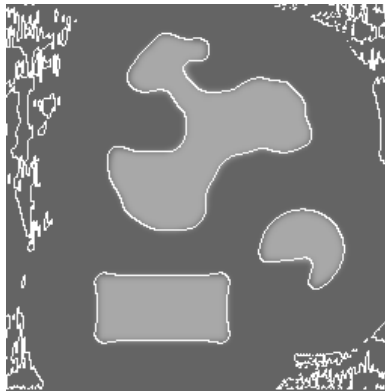


Figure 37: FFT formes2g.pgm pour sigma=5

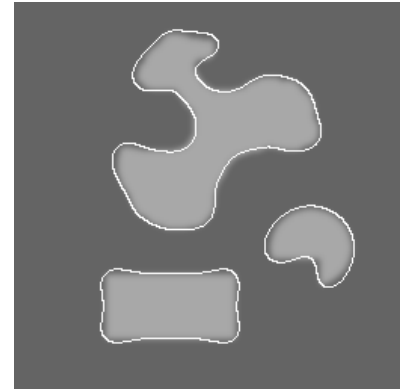


Figure 38: FFT formes2g.pgm pour sigma=10

Laplacien par convolution spatiale pour formes2g.pgm



Figure 39: Laplacien gaussien formes2g.pgm pour sigma=1



Figure 40: Laplacien gaussien formes2g.pgm pour sigma=5

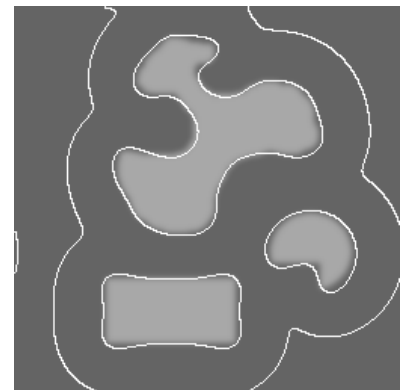


Figure 41: Laplacien gaussien formes2g.pgm pour sigma=10

formes2bb20.pgm

Les deux méthodes (le Laplacien par convolution spatiale et le Laplacien par FFT) se comporte d'une manière équivalente pour toutes les 3 valeurs de σ (1, 5, 10). Lorsque σ augmente, la sensibilité au bruit diminue. En outre, la qualité de la détection des contours devient un peu moins exacte.

formes2g.pgm

Néanmoins, pour l'image **formes2g.pgm**, on remarque des différences.

La méthode du Laplacien par convolution spatiale duplique les contours à l'extérieur du contour visé. Ce contour dupliqué s'éloigne du vrai contour au fur et à mesure que la valeur de σ augmente.

La méthode du Laplacien par FFT engendre un bruit pour les valeurs de σ 1 et 5.

2.3 Comparaison

On peut donc conclure que l'opérateur du premier ordre est plus robuste aux bruits et aux détails fins, mais peut manquer de précision. Il dépend fortement de la valeur de σ choisie ainsi que du seuillage.

L'opérateur du deuxième ordre est plus sensible aux bruits et aux détails fins, mais il permet de détecter les vrais contours d'une manière plus exacte en le comparant à l'opérateur du premier ordre. Il faut aussi noter que l'augmentation de la valeur de σ permet d'éviter la détection du bruit et des détails fins.