

## Rapport TP1 Analyse d'Image

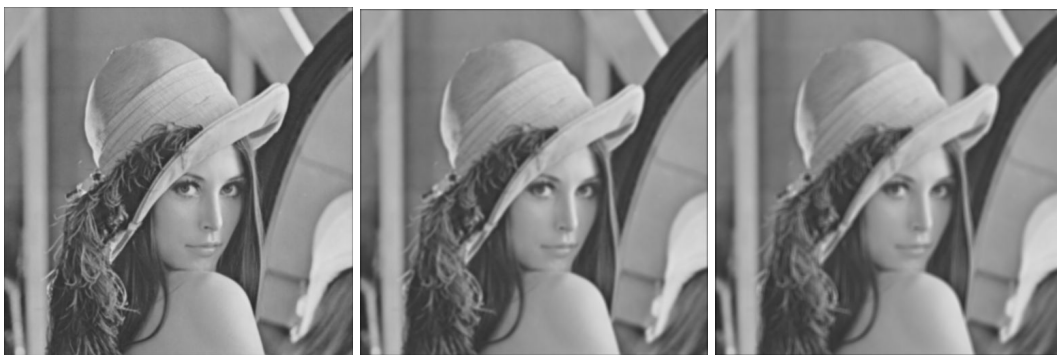
### Détection de contour

Remarques général sur le code :

- Hormis pour l'ouverture et l'affichage des images, la librairie Opencv n'est pas utilisé.
- La classe Matrix ("src/utls/Matrix.h" et "src/utls/Matrix.cpp") est une ré-implémentation de matrice avec quelques ajouts permettant notamment la conversion en Matrice OpenCV (et inversement) ou la création de sous matrice sans risque (gestion de dépassement de la taille de la matrice initial) par exemple.
- La classe MatrixFilter ("src/utls/MatrixFilter.h" et "src/utls/MatrixFilter.cpp"), héritant de la classe Matrix représente tous les types de filtres possible. C'est une matrice avec un facteur à utiliser lors des convolutions.
- La plupart des fonctions importante (convolution, seuillage et raffinement de contour) se trouve dans les fichiers "src/Computations.h" et "src/Computations.cpp".
- Divers tests sont trouvables dans le fichier "main.cpp" et exécutables en lisant les instructions dans le "README.md".

### I] Convolution

Il y a 3 fonctions de convolution. Une fonction pour les convolutions simples (filtre de lissage), une pour les convolutions bidirectionnelles et une pour les convolutions quadri-directionnelles. Les fonctions ont été séparées car la rotation de matrices, nécessaire pour les cas multidirectionnels, ne permettait pas la création d'une seul fonction.



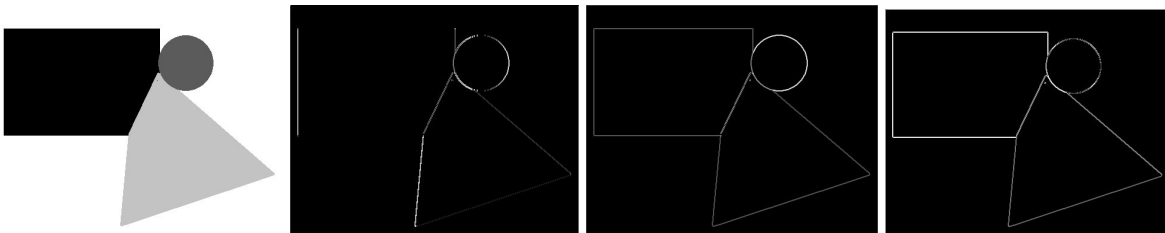
*Images montrant l'utilisation de la convolution simple sur la fameuse image de Lenna. De gauche à droite : image original et convolution simple avec filtre gaussien 7x7 sigma = 0.5, convolution simple avec filtre gaussien 7x7 sigma = 0.01*

### II] Détection de contour

L'application des différents filtres montre bien l'importance des filtres différentiels directionnels. Plus le nombre de directions est grand plus on obtient des contours remarquables.

#### a) Sobel

Les exemples ci dessous mettent en évidence les différences entre les trois convolutions. La convolution uni-directionnelle omet complètement les deux côtés horizontaux du rectangle. On remarquera aussi que les côtés du triangle sont plus marqués sur la convolution quadri-directionnelle que sur les deux autres.

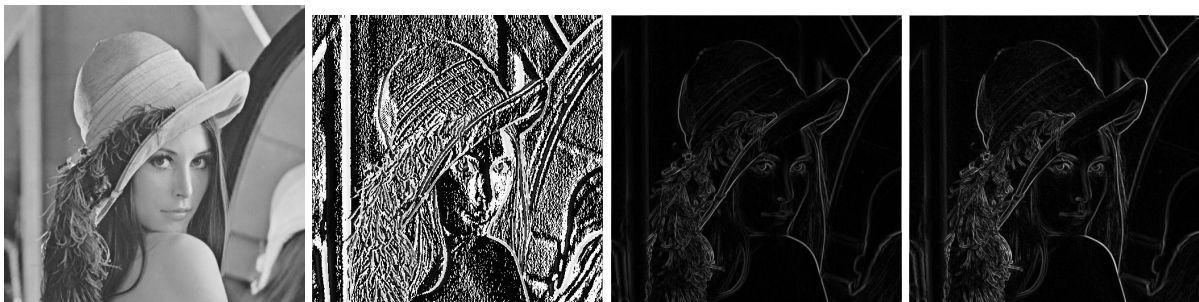


*Images montrant l'application du filtre de Sobel sur un exemple simple.*

*Images de gauche à droite : image original, convolution uni-directionnelle (horizontal), convolution bidirectionnelle et convolution quadri-directionnelle.*

#### b) Prewitt

Les exemple ci-dessous montre très clairement la faiblesse du filtre uni-directionnel sur de vraies images, en effet il y a trop de variation pour obtenir des contours intéressant. On remarquera aussi, concernant les filtres multi-directionnel, des contours plus marqués s'il y a plus de directions testées.



*Images montrant l'application du filtre de Prewitt sur un la photo de Lenna.*

*Images de gauche à droite : image original, convolution uni-directionnelle (horizontal), convolution bidirectionnelle et convolution quadri-directionnelle.*

### c) Kirsch

Les constats des deux dernières parties sont les même avec le filtre de Kirsch. L'image obtenu avec la convolution quadri-directionnelle possède un contraste plus fort.

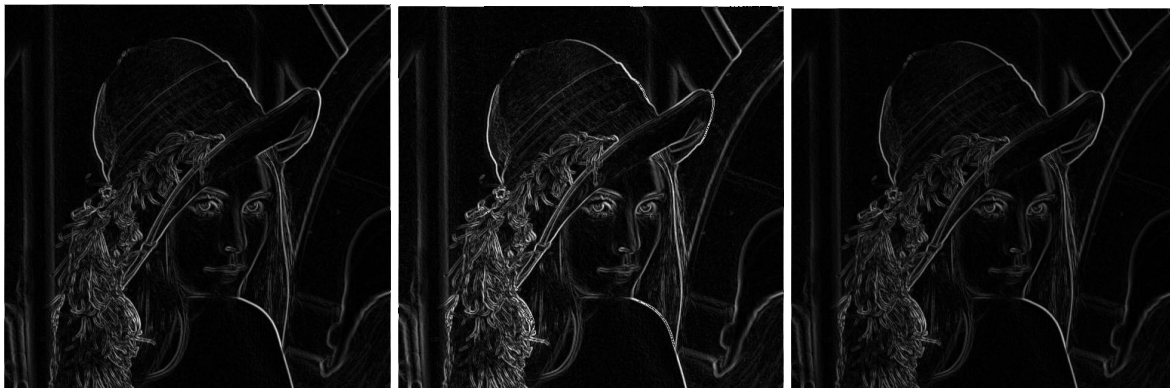


*Images montrant l'application du filtre de Kirsch sur un la photo de Barbara.*

*Images de gauche à droite : image original, convolution uni-directionnelle (horizontal), convolution bidirectionnelle et convolution quadridirectionnelle.*

### d) Comparatifs

La comparaison de trois filtres sur une même image met en évidence un contraste plus fort (très visible sur les yeux et les cheveux) pour le filtre de sobel en revanche, les contours sont moins continue (cela est assez visible sur l'épaule).



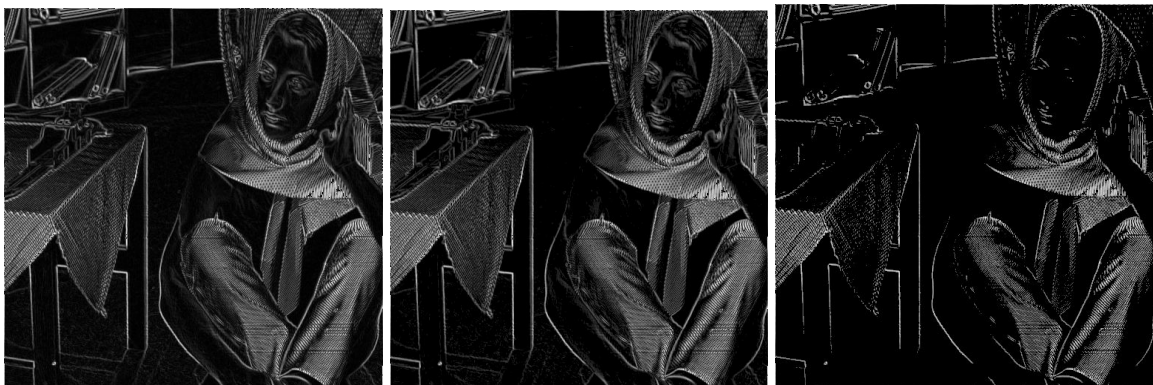
*Images comparant les 3 filtres sur une même image avec un filtre quadridirectionnel. Images de gauche à droite : filtre de Prewitt, filtre de Sobel et filtre de Kirsch*

### III] Etape de seuillage

Le seuillage peut être obtenu en utilisant la moyenne ou la médiane des différents pixels d'une image. Les filtres ayant été déjà appliqués, la plupart des pixels de l'image ont une valeur à 0 ou une valeur proche de 200. La moyenne sera donc faussée à cause de l'écart entre les deux "classes" de valeurs. La médiane semble être un choix plus pertinent.

#### a) Seuillage global

Le seuillage global proposé se base sur les quantiles de l'image. C'est à dire que l'on omet un certain pourcentage des valeurs de l'image pour ne garder que les plus grandes. On voit bien ici les limites d'un seuillage global. Les contours ne sont pas affinés, les zones très contrastées le reste. On perd notamment le visage de Barbara avec un quantile grand tandis que son pantalon reste très visible.



*Images montrant l'impact du seuillage global sur l'image de Barbara avec un filtre de Sobel quadridirectionnel puis un post traitement avec un seuillage global.*

*Image de gauche à droite : Image sans seuillage, Image avec un seuillage avec un quantile égal à la médiane, Image avec un seuillage global avec pour quantile 85 (c'est à dire que l'on a gardé les 15% des valeurs les plus grandes le reste étant noir).*

#### b) Seuillage local

L'idée du seuillage local sera la même que celle du seuillage globale (quantile) mais sur une plus petite zone. Les premières observations du seuillage local amènent trois constats. Dans un premier temps, un seuillage trop gros fragmente totalement l'image (nuage de points sur l'image la plus à droite). Il permet néanmoins de conserver plus facilement les zones avec des contours plus faibles (visage de Barbara en comparaison avec le seuillage global). Enfin le seuillage global permet d'affiner certains contours simples si le seuillage n'est pas trop grand (pieds de la table).



*Images montrant l'impact du seuillage local sur l'image de Barbara avec un filtre de Prewitt quadridirectionnel puis un post-traitement avec un seuillage local.*

*Image de gauche à droite : Image sans seuillage, Image avec un seuillage local avec un quantile égal à 70, Image avec un seuillage avec pour quantile 80, Image avec un seuillage avec un quantile égal à 95.*

### c) Seuillage par hystérésis

Le seuillage par hystérésis semble tout d'abord vouloir effacer les problèmes de nuages de points en conservant les contours, relativement faibles, proche de contours plus fort. Il permet de garder la continuité de certains contours tout en éliminant efficacement le bruit.



*Images montrant l'impact du seuillage par hystérésis sur l'image de Lenna avec un filtre de Kirsch quadridirectionnel puis un seuillage par hystérésis.*

*Image de gauche à droite : Image sans seuillage, Image avec un seuillage par hystérésis avec un quantile égal à 70, Image avec un seuillage avec pour quantile 80, Image avec un seuillage avec un quantile égal à 95.*

*Les valeurs de seuillage pour les images ci dessus ont été obtenu avec un seuillage global pour le seuillage haut. Le seuillage bas a été calculé proportionnellement au seuillage haut ( $0.7 \cdot Sh$ ).*

Le seuillage par hystérésis est assez ouvert dans son utilisation. Il peut en exister de toute sorte. Il peut être intéressant de travailler avec deux seuils obtenus différemment. Par exemple, le premier (seuil bas) pourrait être global pour éliminer les trop petites valeurs (une forme de bruit) et le second seuil (seuil haut) pourrait être local pour certes éliminer les points correspondant à du bruits mais surtout pour

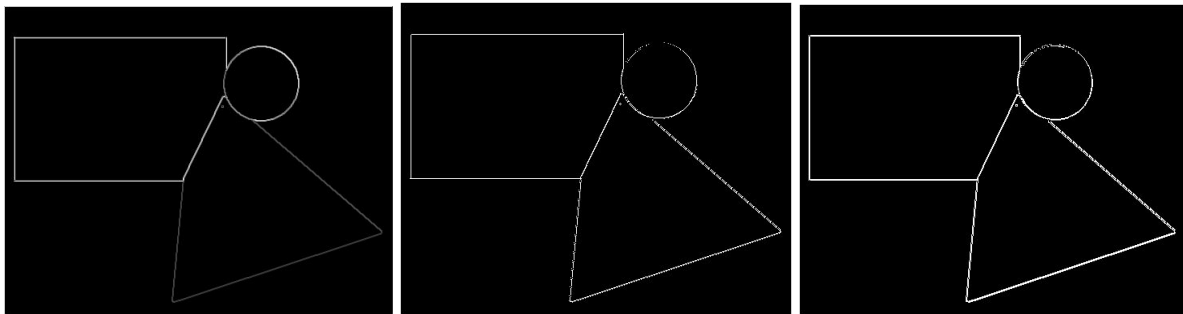
conserver la continuité des contours avec des valeurs plus faibles à certains endroits.

#### IV] Affinage des contours

##### a) Méthodes de maxima locaux

La méthode a été implémentée pour marcher dans les cas de convolution bidirectionnelle et quadri-directionnelle.

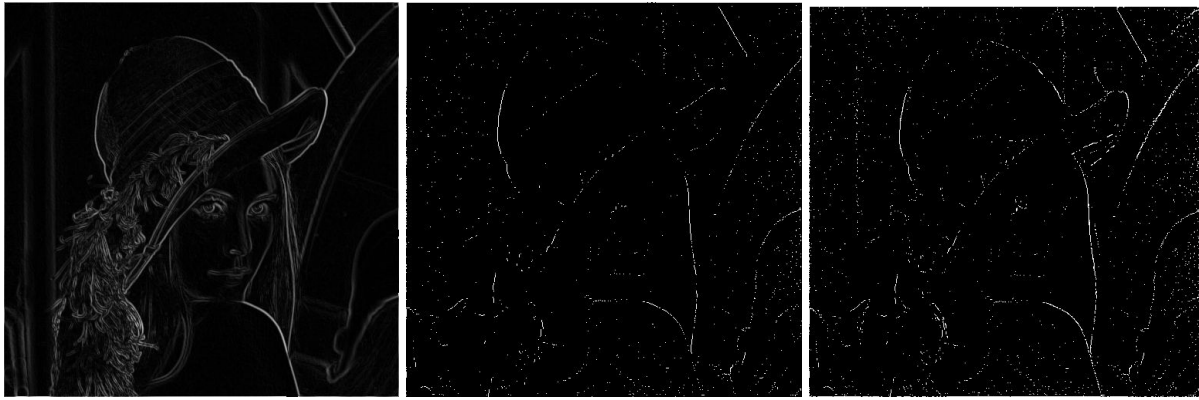
Les premiers tests sur un exemple simple montre l'efficacité de la méthode dans les cas bidirectionnels. Les contours sont bien de 1 pixel. En revanche, l'application à des cas quadri-directionnels entraînent un épaissement du trait.



*Images montrant l'application du raffinement de contour sur un exemple simple.*

*Images de gauche à droite : image après application d'un filtre de Kirsch, image après application d'un filtre de Kirsch et raffinement des contour sur les axes x et y, images après filtrage de Kirsch et affinement des contour quadridirectionnel.*

Bien que les résultats soit assez probants sur des images simples, le raffinement de contour est décevant sur des images complexes. Cela s'explique par un trop gros nombre de valeurs de gradients directionnels. Les maxima locaux ne sont pas les bons car il y a trop de valeurs localement.



*Images montrant l'application du raffinement de contour sur Lenna.*

*Images de gauche à droite : image après application d'un filtre de Kirsch, image après application d'un filtre de Kirsch et raffinement des contour sur les axes x et y, images après filtrage de Kirsch et affinement des contour quadridirectionnel.*

*Toutes les illustrations présentes dans ce document sont trouvables dans le dossier du projet : "data/Results"*