

Travaux pratiques n°1: Traitement des Images

Etudiants: Dogan Ozgur et Melissaratos Konstantinos
 Encadrant: M. Girard Félix

1 Utilisation des histogrammes

1.1 Traitement de l'image *aquitaine.png*



Figure 1: Image *aquitaine.png*

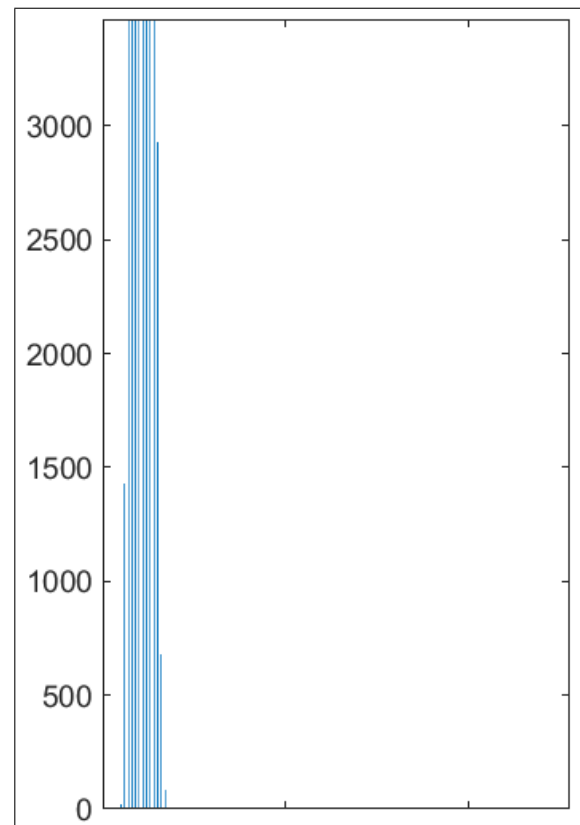


Figure 2: Histogramme de l'image

Sur la figure 2, correspondant à l'histogramme de l'image *aquitaine.png*, nous pouvons voir que les valeurs sont rassemblées à gauche. Donc les valeurs des pixels constituant l'image sont faibles. C'est cohérent avec l'image qui comme nous pouvons voir est très sombre.

La fonction *imagesc* de MatLab permet d'afficher l'image sombre dans une plage de couleurs plus adaptée. Dans notre cas, étant une image en niveau de gris, il est nécessaire aussi d'utiliser la fonction *comlormap* avec le paramètre *gray* afin d'afficher l'image dans une plage de niveau de gris adaptée.

En ajoutant 100 à la valeur de chaque pixel de l'image, cette dernière devient plus claire. Cela cause aussi un décalage de l'histogramme de l'image vers la droite. C'est le résultat que nous pouvons voir sur les figures ci-dessous.

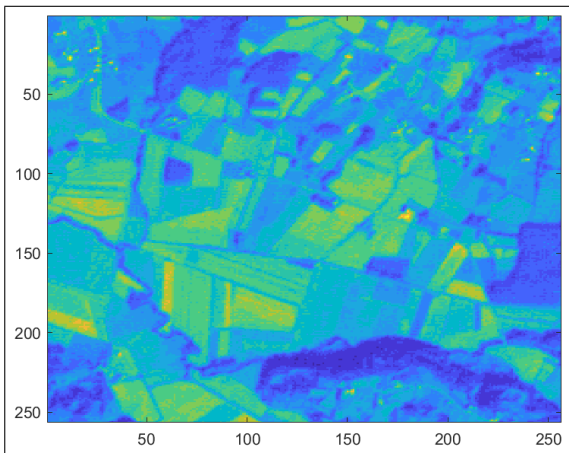


Figure 3: Image affichée avec la fonction *imagesc*

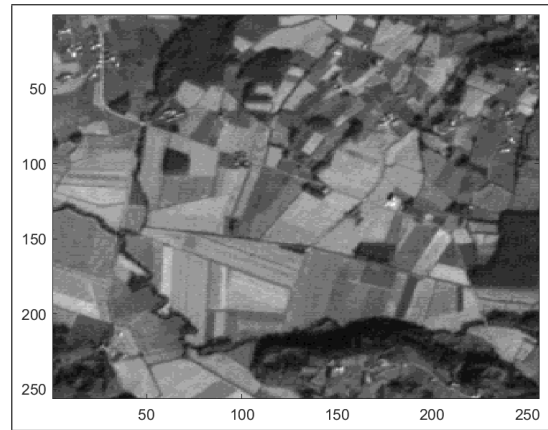


Figure 4: Image affichée avec la fonction *imagesc* dans la bonne plage de valeurs

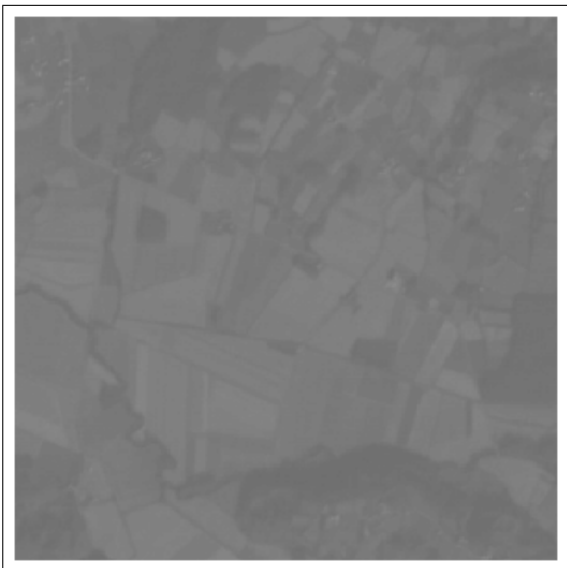


Figure 5: Image aquitaine.png après augmentation par 100 des valeurs de ses pixels

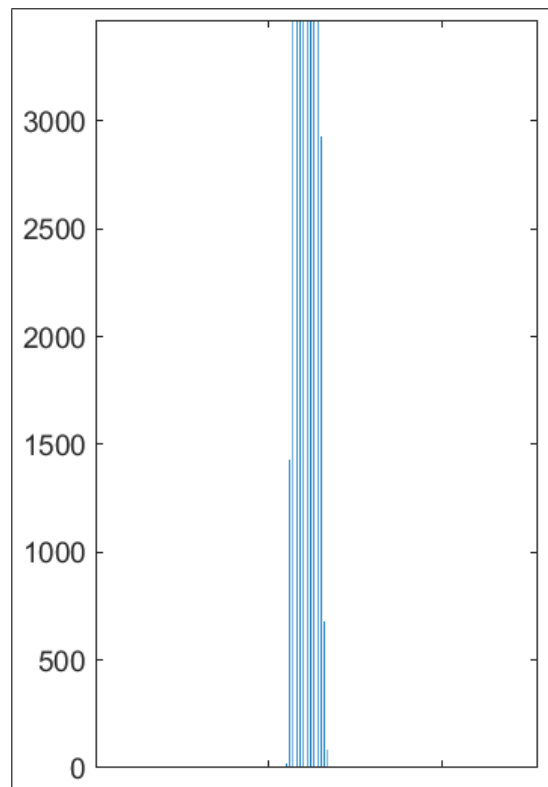


Figure 6: Histogramme de l'image aquitaine.png après augmentation par 100 des valeurs de ses pixels

1.2 Traitement de l'image lena.png

Voici, ci-dessous, l'image *lena.png*, ainsi que son histogramme.



Figure 7: Image lena.png

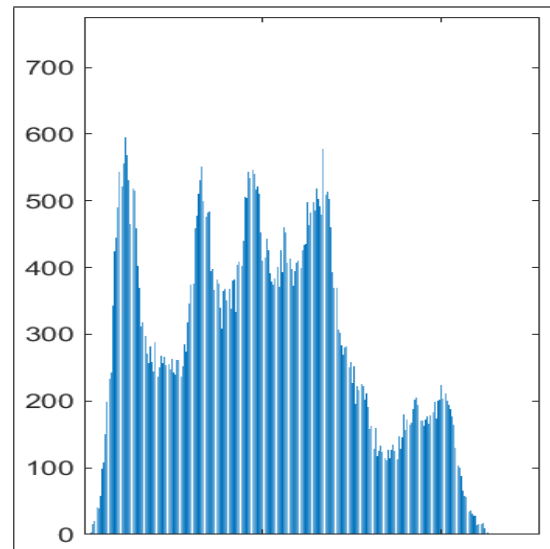


Figure 8: Histogramme de l'image lena.png



Figure 9: Image lena.png après division des valeurs des pixels par 8

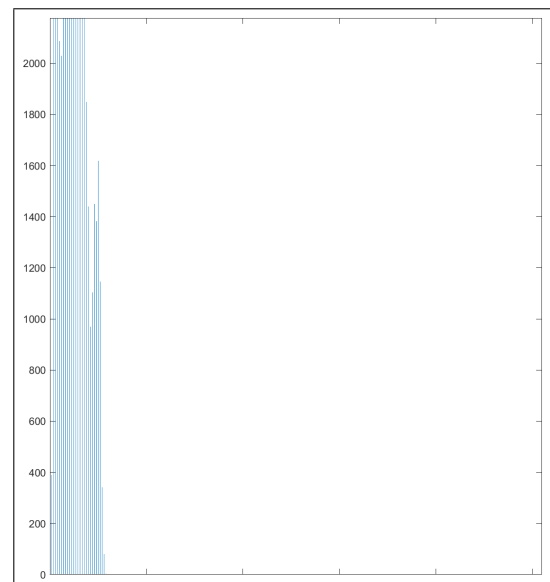


Figure 10: Histogramme de l'image lena.png après division des valeurs des pixels par 8

Sur la figure 9 ci-dessus nous pouvons voir que l'image d'origine est devenue bien plus sombre et donc son histogramme c'est rassemblé vers la gauche. Cela est bien cohérent avec l'opération que nous avons effectué qui avait pour but de diminuer les valeurs des pixels.

1.3 Traitement de l'image micro1.png

Ci-dessous sont présentes les figures 11 et 12 correspondant respectivement à l'image *micro1.png* et son histogramme.

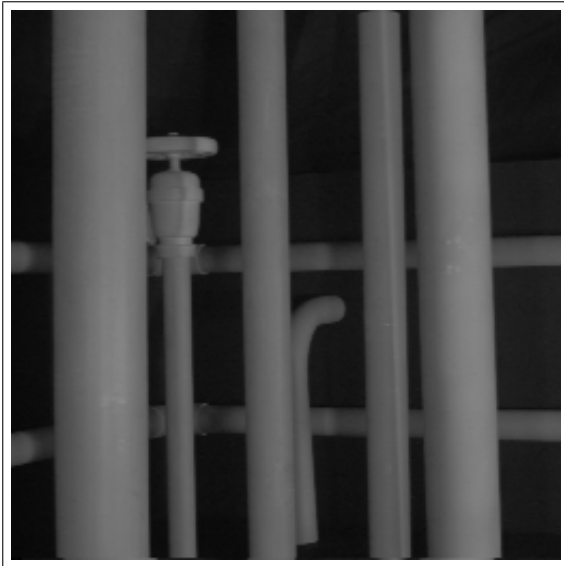


Figure 11: Image micro1.png

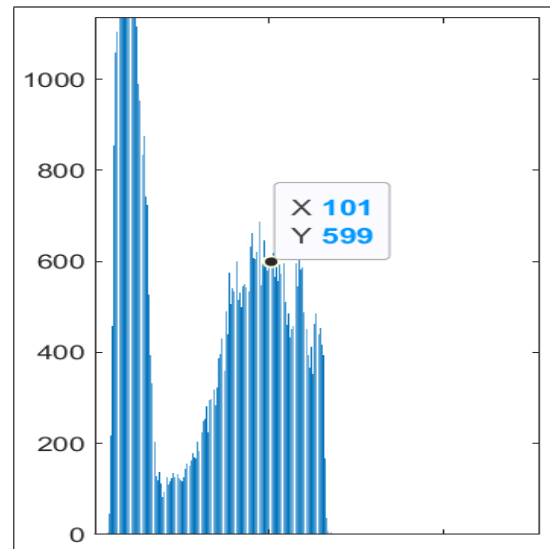


Figure 12: Histogramme de l'image micro1.png

Nous pouvons voir que bien que l'image n'ai pas sombre ou claire, son histogramme est reparti sur l'intervalle approximatif $[0, 150]$.



Figure 13: Image micro1.png normalisée

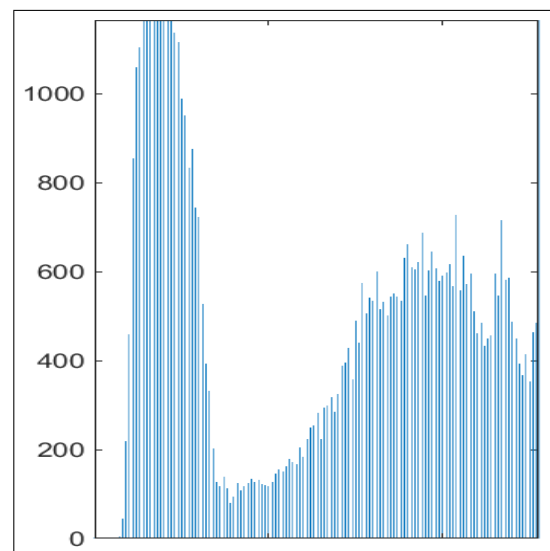


Figure 14: Histogramme de l'image micro1.png normalisée

Sur l'image normalisée ci-dessus nous pouvons voir que l'image est éclaircie par rapport à l'image d'origine. De plus, son histogramme est de la même forme que l'historgramme initial mais il est réparti presque sur la plage entière des valeurs possibles de niveau de gris.



Figure 15: Image micro1.png après égalisation

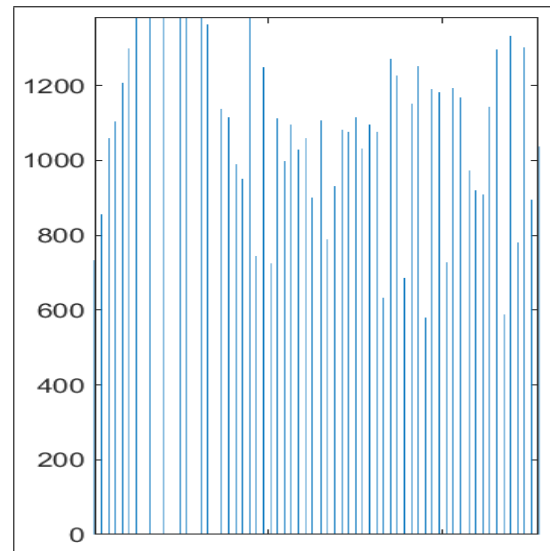


Figure 16: Histogramme de l'image micro1.png après égalisation

Nous pouvons voir que après égalisation l'image obtenue est plus claire que l'image de départ et son histogramme est étalé sur la plage entière de valeurs de niveau de gris.

En comparant les deux nouvelles images obtenues nous pouvons voir que le résultat est proche mais l'égalisation a produit une image faiblement plus claire. Celà, dans la théorie, semble cohérent étant donné que l'égalisation consiste à rapprocher les occurrences de chaque niveau de gris de l'image, tandis que la normalisation consiste à augmenter la dynamique.

2 Filtrage

2.1 Un algorithme simple de détection de contours

2.1.1 Image d'un carré blanc sur fond noir

Suite à la création de l'image d'un carré blanc sur un fond noir (c.f. figure 17), nous l'avons convolué avec les filtre suivants:

$$\nabla_x = \begin{pmatrix} 0 & 0 & 0 \\ -1 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} \quad \text{Filtre de gradient x}$$

$$\nabla_y = \begin{pmatrix} 0 & -1 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \quad \text{Filtre de gradient y}$$

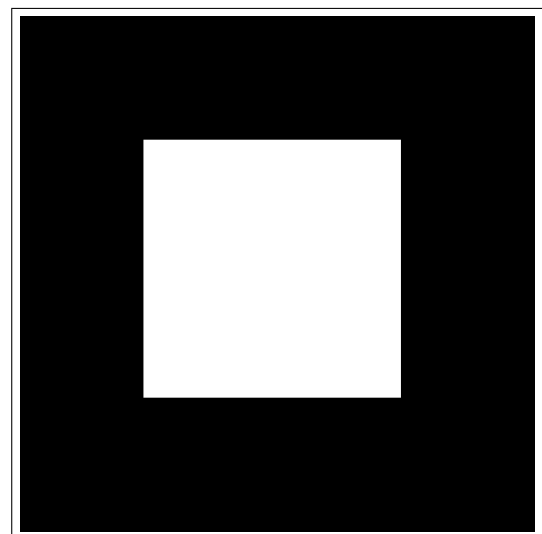


Figure 17: Image carrée construite

Ainsi après application des filtres nous obtenons les résultats figurant sur les figures 18 et 19.

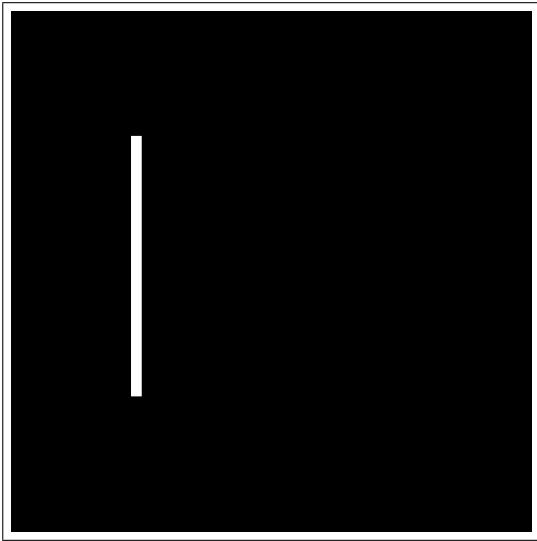


Figure 18: Image de la norme de gradient en x

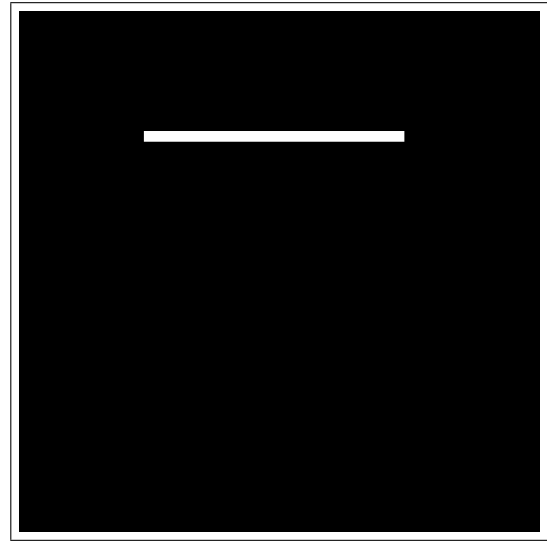


Figure 19: Image de la norme de gradient en y

2.1.2 Image cameraman.pgm

Nous avons appliqué ces mêmes filtres à l'images *cameraman.pgm* ci-dessous:



Figure 20: Image cameraman.pgm d'origine

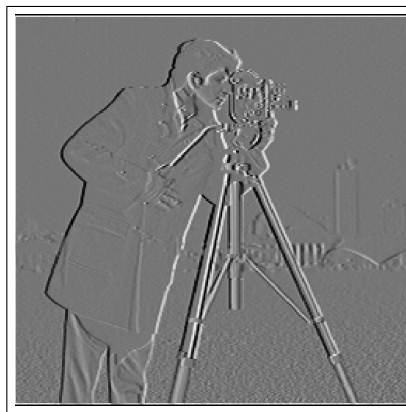


Figure 21: Image après application du filtre de gradient en x

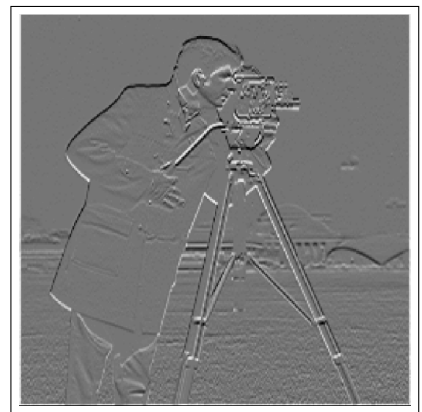


Figure 22: Image après application du filtre de gradient en y

Nous pouvons voir sur l'image de la norme de gradient (c.f. figure 23) que le cameraman est clairement visible étant l'élément structurant principal de l'image. Nous pouvons alors dire que l'image de la norme de gradient permet de voir les délimitations des objets principaux de l'image.

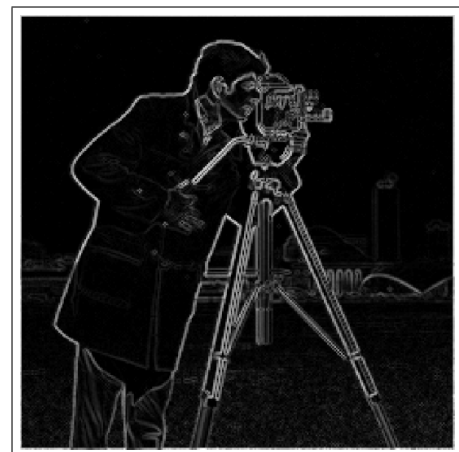


Figure 23: Image de la norme de gradient

2.2 Robustesse au bruit

Sur l'image *cameraman.pgm* nous avons ajouté un bruit blanc, qui comme précisé sur l'énoncé s'agit d'un bruit aléatoire qui suit une distribution gaussienne de moyenne nulle et de variance σ^2 qui est ajouté.

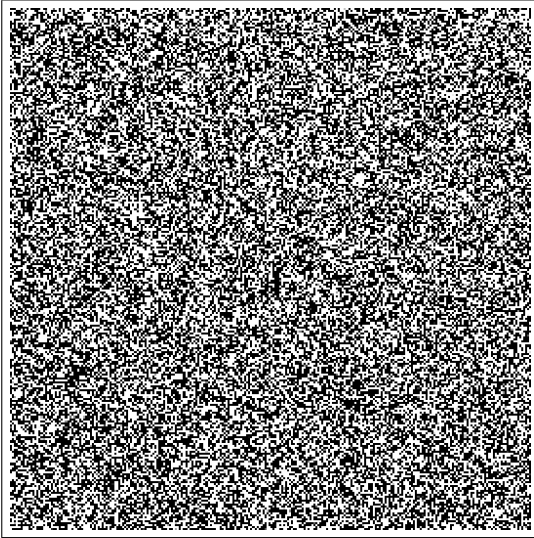


Figure 24: Bruit ajouté de variance $\sigma^2 = 20$



Figure 25: Image cameraman.pgm bruitée

2.2.1 Filtre de gradient

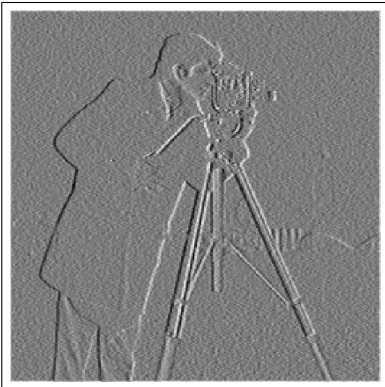


Figure 26: Image après application du filtre de Sobel en x

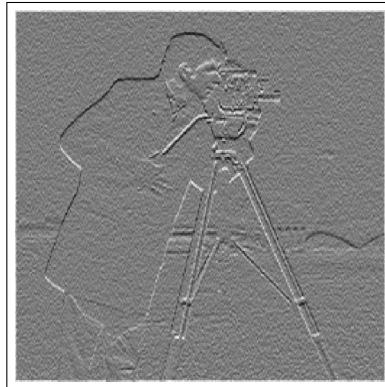


Figure 27: Image après application du filtre de Sobel en y

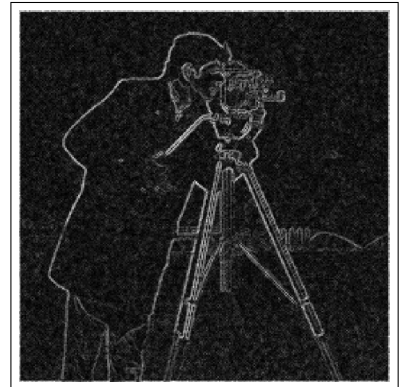


Figure 28: Image de norme de gradient de Sobel de l'image d'origine

Sur les images de gradient ci-dessus, nous pouvons voir que du bruit s'y est ajouté. Cependant les images de gradient sont bien ressemblantes aux images de gradient de l'image d'origine.

2.2.2 Filtre de Sobel

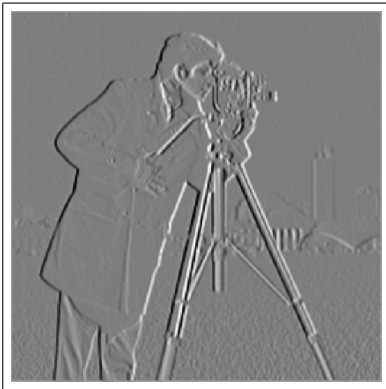


Figure 29: Image après application du filtre de Sobel en x

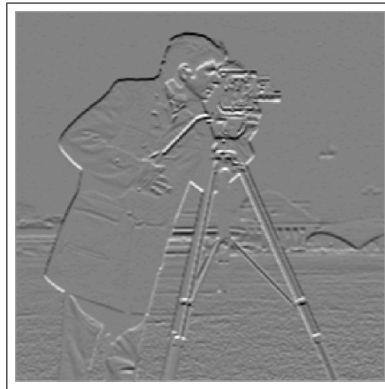


Figure 30: Image après application du filtre de Sobel en y



Figure 31: Image de norme de gradient de Sobel de l'image d'origine

Nous pouvons voir que les images obtenues en appliquant les filtres de Sobel et celles obtenues après application des filtre de gradient très ressemblantes. Toutefois, nous pouvons voir que l'image de la norme de gradient de Sobel semble plus claire, avec de meilleures délimitations de l'objet structurant.

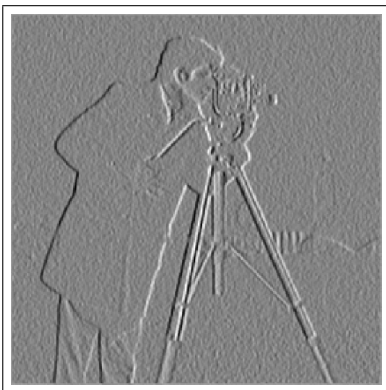


Figure 32: Image après application du filtre de Sobel en x de l'image bruitée

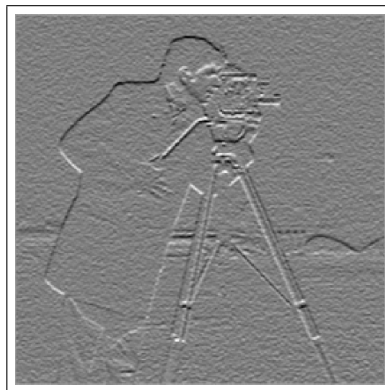


Figure 33: Image après application du filtre de Sobel en y de l'image bruitée



Figure 34: Image de norme de gradient de Sobel de l'image précédemment bruitée

En appliquant les filtres gradients de Sobel sur l'image bruitée nous pouvons voir que les images obtenues semblent elles aussi bruitées. Nous pouvons voir le même effet sur l'image de la norme du gradient obtenue. Toutefois les délimitations de l'objet structurant restent clairement visibles.

2.2.3 Conclusion

Nous pouvons conclure que les filtres gradients cités précédemment sont des filtres *robustes*. Nous pouvons voir que en ajoutant du bruit sur une image, les images des gradients, ainsi que l'image de norme de gradient sont elles aussi bruitées, mais le résultat obtenu reste clair et interprétable.

2.3 Détection de contours

Suite à la synthèse des images de gradient de l'image d'origine et de l'image bruitée nous avons procédé à un seuillage sur ces dernières dont le résultat figure ci-dessous:

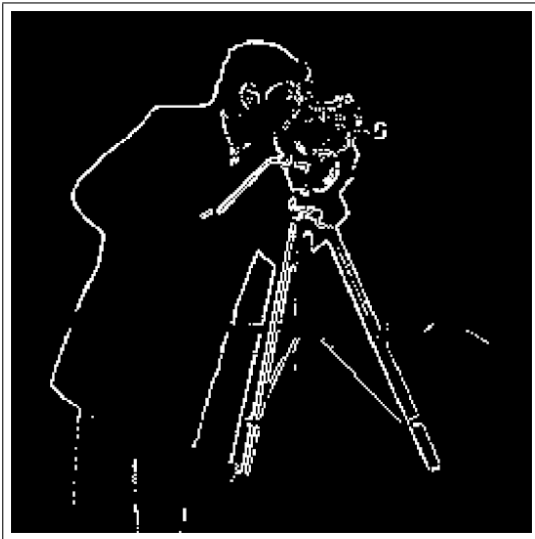


Figure 35: Image de la norme de gradient de l'image d'origine après seuillage (seuil à 150)

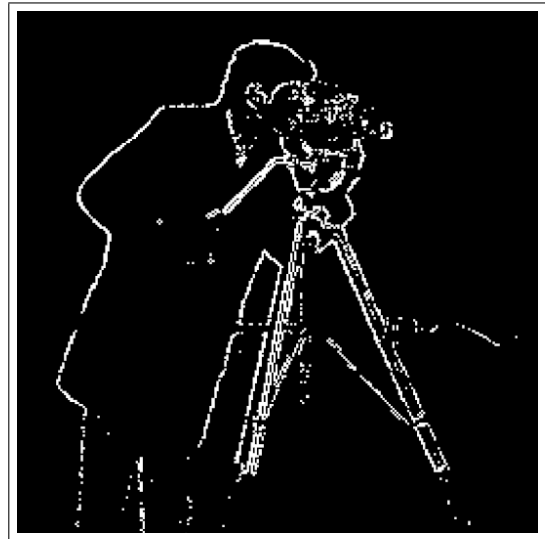


Figure 36: Image de la norme de gradient de l'image bruitée après seuillage (seuil à 150)

Nous pouvons voir que sur la figure 35, les contours du cameraman et de sa caméra sont bien visibles mais une petite partie du fond est aussi présente. Sur la figure 36, nous pouvons voir que du bruit est présent, mais l'objet structurant principal est bien mis en valeur.

Dans ces deux cas, il est possible de supprimer le bruit ainsi que les contours du fond non souhaités par modification du niveau du seuil ou bien par érosion.