

Travaux pratiques de Traitement d'Images

Ce TP sera réalisé à l'aide du logiciel MATLAB et de sa toolbox Image Processing. Le but est manipuler des notions d'histogrammes et de filtrage pour des images en niveaux de gris. Vous réaliserez un compte-rendu succinct dans lequel vous présenterez vos résultats obtenus.

1 Utilisation d'histogrammes

- Affichez l'image *aquitaine.png* ainsi que son histogramme. Vous pourrez utiliser pour cela les fonctions **imread**, **imhist**, **imshow** et **imagesc**. Prenez bien garde à la plage des valeurs utilisée (notamment pour la fonction **imagesc**).
- Modifiez l'image en additionnant 100 à l'intensité de chaque pixel et affichez de nouveau l'image et son histogramme. Expliquez les différences observées au niveau de l'histogramme et de l'image.
- Affichez l'image *lena.png* ainsi que son histogramme. Modifiez l'image en divisant par 8 l'intensité de chaque pixel et affichez de nouveau l'image et son histogramme. Expliquez les différences observées au niveau de l'histogramme et de l'image.
- Affichez l'image *micro1.png* ainsi que son histogramme. Que remarquez-vous ?
- Normalisez l'histogramme afin d'utiliser toute la dynamique possible ($[0 - 255]$). Comparez l'image et l'histogramme obtenus avant et après normalisation.
- Egalisez l'histogramme afin d'utiliser toute la dynamique possible ($[0 - 255]$). Pour cela vous utiliserez la fonction d'égalisation **histeq**. Comparez l'image et l'histogramme obtenus avant et après égalisation.
- Conclusions sur la normalisation et l'égalisation d'histogramme.

2 Filtrage

2.1 Un algorithme simple de détection de contours

- Créez une image de carré blanc sur fond noir.

- Implémentez l'opération de convolution d'une image avec un filtre 3×3 sans gérer les problèmes de bord. (**N'utilisez pas la fonction de matlab qui fait la convolution, faites la vôtre!**)

- Convoluez l'image de carré avec chacun des filtres gradient suivants:

$$\nabla_x = \begin{pmatrix} 0 & 0 & 0 \\ -1 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} \quad \text{et} \quad \nabla_y = \begin{pmatrix} 0 & -1 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \quad (1)$$

- Estimez la norme du gradient. Commentez vos résultats.
- Recommencez pour l'image *cameraman.pgm*.

2.2 Robustesse au bruit

- Bruitez l'image *cameraman.pgm* avec un bruit blanc et reprenez les questions 2.1 avec l'image bruitée. On rappelle que le bruit blanc est un bruit aléatoire qui suit une distribution gaussienne de moyenne nulle et de variance σ^2 qui est ajouté à l'ensemble de l'image. Pour cela vous générerez une image de bruit avec la fonction **randn** et vous l'ajouterez à l'image d'origine.
- Mêmes questions (image bruitée et non bruitée) avec le filtre de Sobel :

$$S_x = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \quad \text{et} \quad S_y = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix} \quad (2)$$

- Quelles différences remarquez-vous entre les images de la norme du gradient et de la norme de Sobel (cas bruité et non bruité) ?
On rappelle que les filtres de Sobel sont obtenus par les relations suivantes :

$$S_x = \nabla_x * \begin{pmatrix} 0 & 1 & 0 \\ 0 & 2 & 0 \\ 0 & 1 & 0 \end{pmatrix} \quad \text{et} \quad S_y = \nabla_y * \begin{pmatrix} 0 & 0 & 0 \\ 1 & 2 & 1 \\ 0 & 0 & 0 \end{pmatrix} \quad (3)$$

- Conclusion sur la robustesse au bruit des filtres gradients.

2.3 Détection de contours

- Seuillez l'image de norme du gradient pour obtenir les contours de l'image (image bruitée et non bruitée). Que remarquez-vous?