

Chapitre 3

Traitements de bases sur l'image (suite)

3.3 Correction de la dynamique de l'image par les transformations affines sur l'histogramme

On peut appliquer une transformation T sur les intensités des pixels d'une image pour en modifier les valeurs tel que :

$$j = T(i)$$

Où, j et i représentent respectivement les intensités des pixels de la nouvelle image et de l'image originale. L'application d'une telle transformation (**LUT : Look Up Table**) a alors une incidence sur l'histogramme.

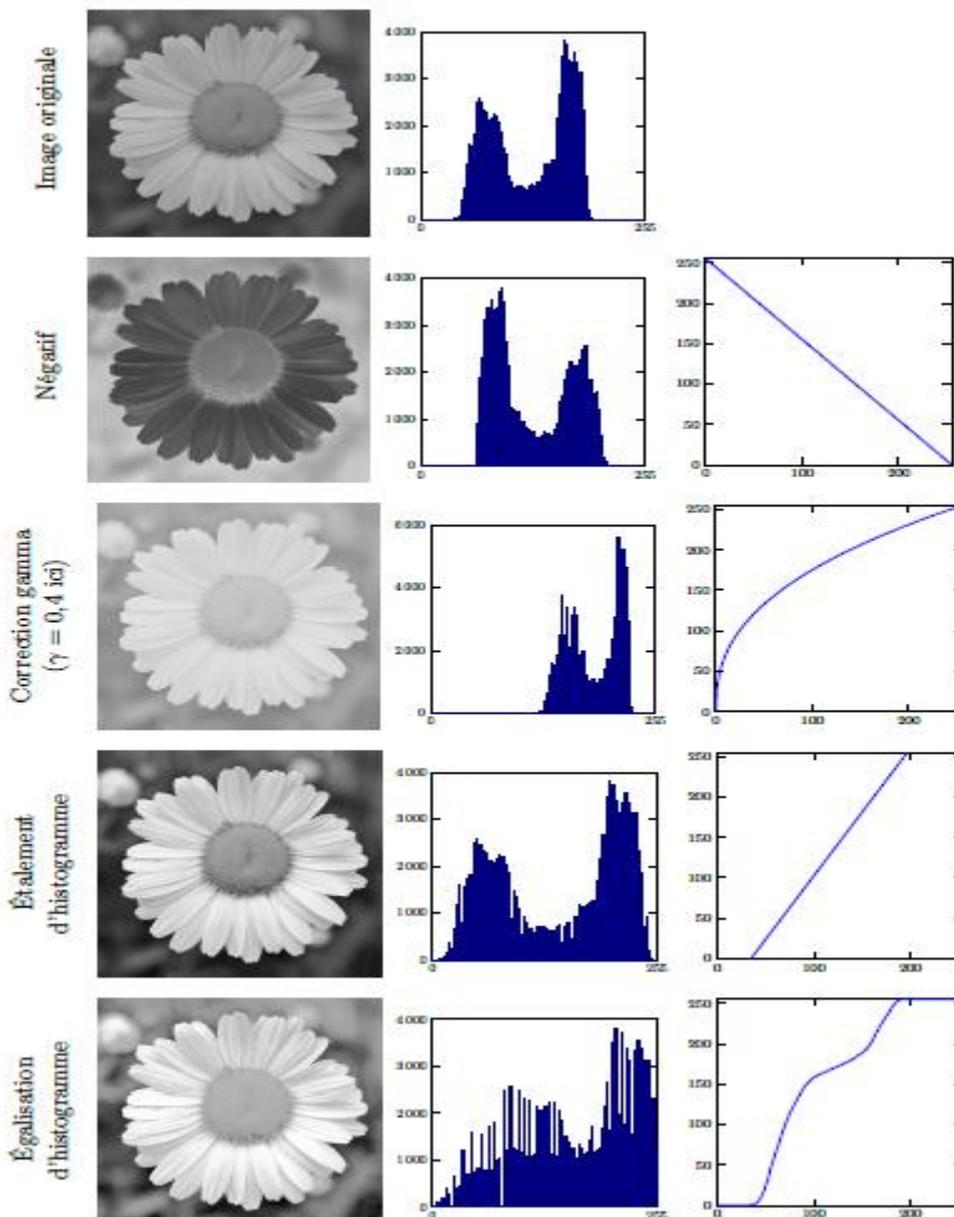


Figure 3.1 : Exemples de transformations. De gauche à droite : image (originale ou transformée) et son histogramme, transformation T .

Quatre transformations classiques sont décrites (en supposant que les intensités sont à valeurs dans $[0;1]$).

- négatif : $T(i) = 1 - i$;
- correction gamma : $T(i) = i^\gamma$;
- étalement (normalisation) :

$$T(i) = \frac{i - i_{min}}{i_{max} - i_{min}}$$

Où ; i_{min} et i_{max} sont respectivement les intensités minimale et maximale de l'image originale ;

- égalisation (equalization) :

$$T(i) = \frac{1}{MN} \sum_{k=0}^i n_k$$

Où ; M et N sont les dimensions de l'image et n_k est le nombre de pixels d'intensité k. Cette transformation cherche à étaler au mieux l'histogramme sur toute la dynamique des intensités, et à rendre l'histogramme le plus plat possible : cela a pour conséquence d'augmenter le contraste de l'image.

Exercice d'application

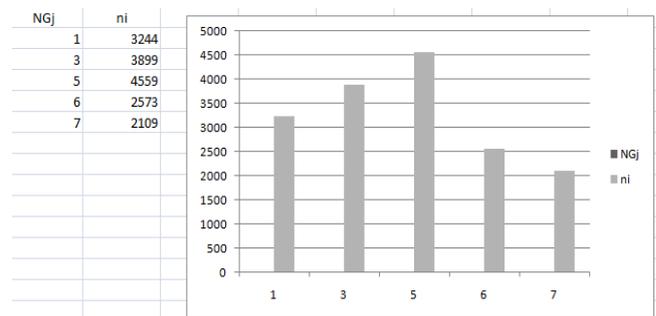
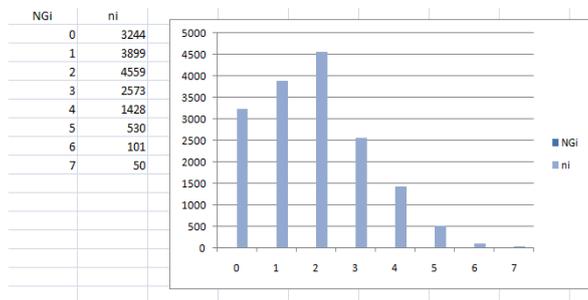
La table suivante donne le nombre de pixels pour les niveaux de gris de 0.....7 d'une image:

NG _i	0	1	2	3	4	5	6	7
n _i	3244	3899	4559	2573	1428	530	101	50

Tracer l'histogramme correspondant, puis faites une égalisation d'histogramme et tracer le résultat obtenu.

Solution :

Tracé de l'histogramme de l'image



Egalisation de l'histogramme de l'image :

L=8 ; le nombre de niveaux de gris, n=taille de l'image=M.N=nombre total de pixel

NG _i	n _i	$\sum n_i$	$\frac{L-1}{n} \sum n_i$	Valeur arrondie = NG _j
0	3244	3244	1.39	1
1	3899	7143	3/05	3
2	4559	11702	5	5
3	2573	14275	6.10	6
4	1428	15703	6.71	7
5	530	16233	6.97	7
6	101	16334	6.98	7
7	50	16384	7	7

NG _j	1	3	5	6	7	7	7	7
n _i	3244	3899	4559	2573	1428	530	101	50

Pour le niveau 7 on a 1428+530+101+50=2109

NG _j	1	3	5	6	7
n _i	3244	3899	4559	2573	2109

Rq :Les tracés ont été fait avec Excel vous pouvez utiliser d'autres logiciel pour le tracé(pour les examens le tracé sera manuel)

3.4 Seuillage (déjà vu au cours)

Dans certains cas, l'histogramme peut être un outil efficace pour segmenter une image en deux classes, c'est-à-dire pour distinguer les objets de l'image suivant leur luminosité. En effet, lorsque l'histogramme présente deux modes distincts, on peut définir un seuil (**threshold**) *S* entre ces deux modes, et appliquer ensuite un seuillage (thresholding) sur les pixels de l'image :

- si le pixel a une valeur inférieure au seuil, le pixel est dans la classe 0 ;
- si le pixel a une valeur supérieure au seuil, le pixel est dans la classe 1.

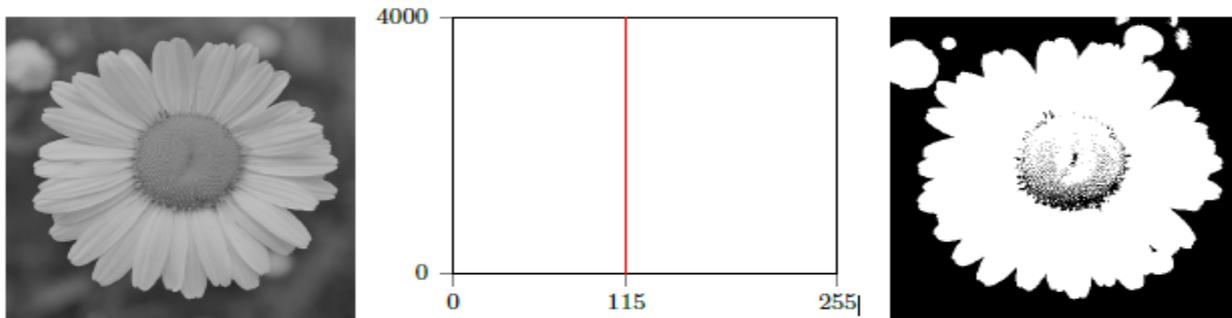


Figure 3.2 : Seuillage avec un seuil égal à 115.

Le seuillage permet d'obtenir une image binaire qui ne contient que deux valeurs. Il existe des méthodes pour calculer automatiquement le seuil.

3.5. Opérations arithmétiques

Des opérations mathématiques simples peuvent être effectuées sur les éléments de deux images. Pour simplifier, on considère dans cette partie des images 2D, la généralisation à des images de dimension différente est évidente.

3.5.1 Addition d'images

L'addition de deux images *f* et *g* de même taille est une nouvelle image *h* de même taille dont chaque pixel correspond à la somme des valeurs des pixels des images originales :

$$\forall m, n \quad h(m, n) = f(m, n) + g(m, n).$$

La figure 3.3 donne un exemple d'utilisation de l'addition. L'addition peut également être utilisée pour débruiter une série d'images.



Figure 3.3 : L'image de droite est la somme des deux images de gauche. Notez que cette somme a des valeurs entre 0 et 510.

3.5.2 Soustraction d'images

La soustraction de deux images est utilisée par exemple pour détecter des changements (figure 3.4).

$$\forall m, n, \quad h(m, n) = f(m, n) - g(m, n)$$

$$\text{ou } \forall m, n \quad h(m, n) = |f(m, n) - g(m, n)|$$



Figure 3.4 : L'image de droite est la différence des deux images de gauche. Notez que l'image de différence a des valeurs entre -255 et 255.

3.5.3 Division d'images

La division de deux images permet notamment de corriger une illumination non homogène. La figure 3.5 illustre la suppression de l'ombre sur une image.

$$\forall m, n, \quad h(x, y) = \frac{f(m, n)}{g(m, n)}$$



Figure 3.5 : L'image de droite est la division de l'image de gauche par l'image de droite.

3.6. Opérations logiques

On applique, pixel à pixel les opérateurs logiques(ET, OU,.....), à deux(ou plusieurs) images de même taille.

Les opérateurs logiques sont appliqués aux images binaires.

A	B	NOT A	A ET B	A OR B	A XOR B	A - B	A NXO B
1	1	0	1	1	0	0	1
1	0	0	0	1	1	1	0
0	1	1	0	1	1	0	0
0	0	1	0	0	0	0	1

- Non logique : "Not"
- Ou logique : "Or"
- Différence logique : "logical-sub"
- Addition logique : "And"
- Ou exclusif logique : "Xor"
- Equivalence logique : "nxo"

On peut utiliser ce types d'opérations dans la morphologie mathématique et pour réaliser des masquages sur les images

Exemple sur le ET logique :

- Pour les images en niveaux de gris, l'opérateur logique est appliqué sur la représentation binaire des niveaux de gris, en comparant les bits correspondants.
- 2 NG 47 et 252 codés sur 8 bits : $(47)_{10}=(00101111)_2$ $(252)_{10}=(11111100)_2$

Le ET logique entre ces deux niveaux donne $(00101100)_2 = (44)_{10}$

Application : masque pour isoler une région

