

# Traitement d'image

TP n°04

**BARCHID** Sami

**CARTON** Floriane

# Introduction

---

La matière vue dans ce TP traite des transformations appliquées à des images monochromes. L'objectif de ce TP est d'implémenter les différentes transformations vues en cours (utilisation de LUT, égalisations, etc) ainsi que d'observer et interpréter les résultats obtenus.

Ce TP est divisé en trois parties :

- **Transformation ponctuelle invariante par translation :** implémenter une transformation ponctuelle invariante par translation au moyen d'une table de correspondance (LUT) et interpréter les résultats obtenus.
- **Transformation ponctuelle non invariante par translation :** implémenter une transformation ponctuelle non invariante par translation au moyen d'une transformation affine et interpréter les résultats obtenus.
- **Égalisation d'histogramme :** implémenter l'égalisation des niveaux de gris d'une image et interpréter les résultats obtenus.

# Avant-propos

---

Les différentes expériences et observations rencontrées sur ce TP ont été réalisées grâce au logiciel « **ImageJ** ».

# Transformation ponctuelle

## invariante par translation

---

### Objectif de l'expérience

L'exercice posé dans cette partie du TP est de modifier les niveaux de gris des pixels d'une image de sorte que la dynamique des niveaux de gris soit entre 0 et 255 (noir et blanc).

En d'autres termes, le but est de faire en sorte que les niveaux de gris des pixels d'une image puissent être répartis entre les niveaux 0 et 255.

### Identification du problème

La transformation requise pour arriver à notre objectif est donc une **transformation ponctuelle invariante** car l'image initiale et l'image transformée sont de dimensions identiques et les composantes d'un pixel de l'image transformé (ici, le niveau de gris) ne dépendent que des composantes du pixel de l'image initiale à la même position.

De plus, cette transformation ponctuelle est **invariante en translation**, car la transformation ponctuelle est une fonction d'un intervalle borné des niveaux de gris minimum et maximum de l'image initiale vers un autre intervalle borné, qui est celui des niveaux de gris de 0 à 255.

### Explication du procédé

Pour réussir à implémenter cette transformation ponctuelle invariante en translation, nous utiliserons une LUT (Look Up Table) de sorte que, pour chaque valeur entière du niveau de gris initiale, nous y associons une valeur entière de niveau de gris transformé.

Nous pouvons traduire l'élaboration de notre LUT de cette manière :

$$y=a*x+b \text{ où :}$$

- $x$  est la valeur entière de niveau de gris de l'image initiale.
- $y$  est la valeur entière de niveau de gris de l'image transformée.
- $a$  et  $b$  sont des constantes permettant d'obtenir le niveau de gris de l'image transformée.

C'est pourquoi nous devons trouver les constantes  $a$  et  $b$  afin d'appliquer la transformation à chaque niveau de gris de l'image initiale.

D'autre part, la transformation que nous essayons d'implémenter doit pouvoir transformer les valeurs entières de niveau de gris minimale et maximale de l'image initiale respectivement en 0 et en 255, c'est-à-dire :

- $0=a*x_{min}+b$
- $255=a*x_{max}+b$
- où  $X_{max}$  et  $X_{min}$  sont respectivement les valeurs de niveau de gris maximale et minimale de l'image initiale.

Après une courte résolution du système d'équation à deux inconnues, nous pouvons trouver les valeurs des constantes  $a$  et  $b$  :

- $a=\frac{255}{X_{max}-X_{min}}$
- $b=-a*X_{min}=-\frac{255}{X_{max}-X_{min}}*X_{min}$

Ainsi, nous pouvons appliquer l'équation de notre LUT ( $y=ax+b$ ) sur chaque pixel de l'image.

## Implémentation sous ImageJ

La macro implémentant la transformation ponctuelle recherchée sera constituée des instructions suivantes :

- Trouver le niveau de gris minimum de l'image initiale
- Trouver le niveau de gris maximum de l'image initiale
- Calculer les constantes  $a$  et  $b$
- Pour chaque pixel de l'image
  - Appliquer la formule de la LUT ( $y=ax+b$ ) pour obtenir le pixel de l'image transformée
  - Modifier le pixel de l'image avec celui qui a été calculé

Vous trouverez, ci-après, le code source de la transformation

```

image = getImageID();
W = getWidth();
H = getHeight();
xmin = findGreyLevMin();
xmax = findGreyLevMax();

// equation de la LUT :
//  $y = a * x + b$  où
//      $y$  est le niveau de gris de l'image transformée
//      $x$  est le niveau de gris de l'image initiale

// application de la formule trouvée (c.f. rapport)
a = 255/(xmax - xmin);
b = -a*xmin;

// POUR CHAQUE [pixel]
for (j=0; j<H; j++)
{
    for (i=0; i<W; i++)
    {
        p = getPixel(i,j);
        // calculer la valeur transformée avec l'équation de la LUT ( $y = ax + b$ )
        pnew = a*p + b;

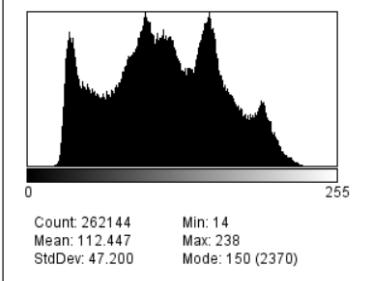
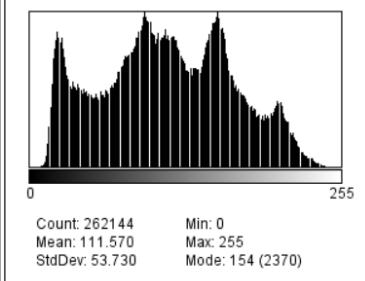
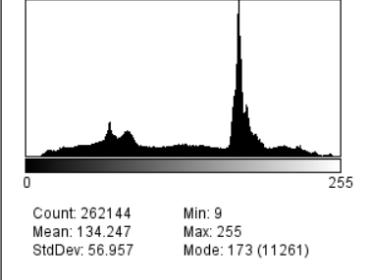
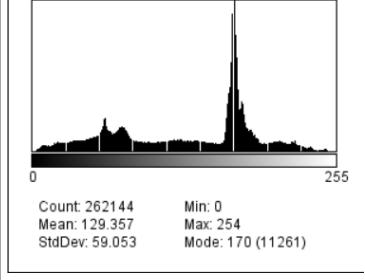
        // remplacer le pixel de l'image initiale par celui de l'image transformée
        setPixel(i,j,pnew);
    }
}

```

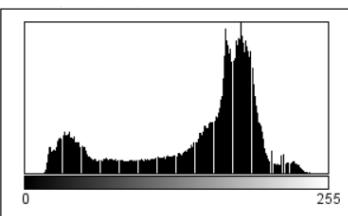
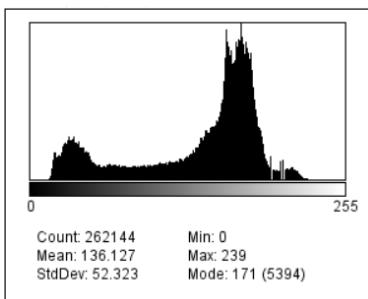
où les fonctions « `findGreyLevMin` » et « `findGreyLevMax` » permettent de retrouver  $x_{min}$  et  $x_{max}$  (l'implémentation de ces fonctions étant triviales, nous n'avons pas souhaité alourdir le rapport en rajoutant le code de ces fonctions).

## Interprétation des résultats

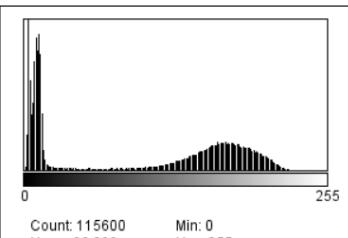
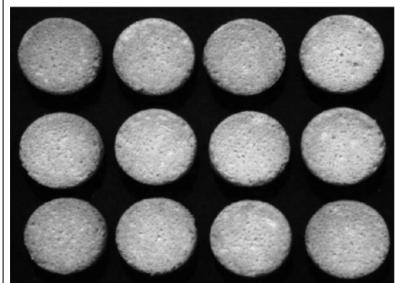
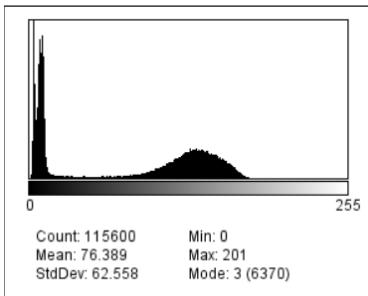
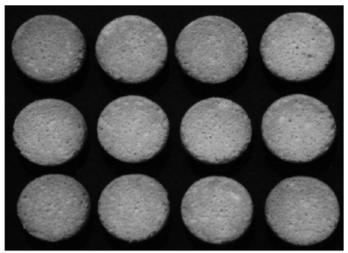
Le tableau ci-dessous montre des images sur lesquelles nous avons testé la transformation ponctuelle implémentée.

Nº	Image initiale	Histogramme des niveaux de gris	Image transformée	Histogramme des niveaux de gris (transformé)
1		<p>300x240 pixels; RGB; 281K</p>  <p>Count: 262144 Mean: 112.447 StdDev: 47.200</p> <p>Min: 14 Max: 238 Mode: 150 (2370)</p>		 <p>Count: 262144 Mean: 111.570 StdDev: 53.730</p> <p>Min: 0 Max: 255 Mode: 154 (2370)</p>
2		 <p>Count: 262144 Mean: 134.247 StdDev: 56.957</p> <p>Min: 9 Max: 255 Mode: 173 (11261)</p>		 <p>Count: 262144 Mean: 129.357 StdDev: 59.053</p> <p>Min: 0 Max: 255 Mode: 170 (11261)</p>

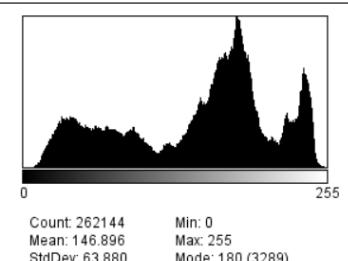
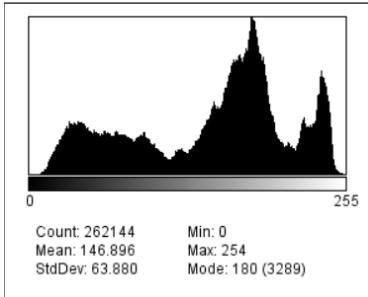
3



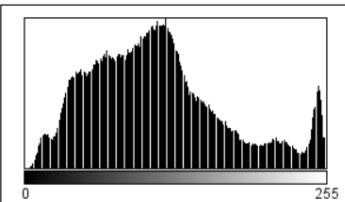
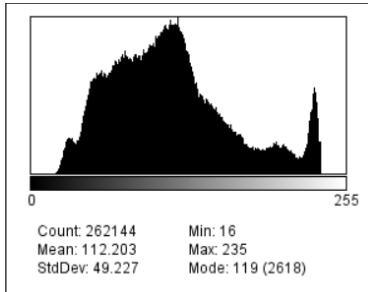
4



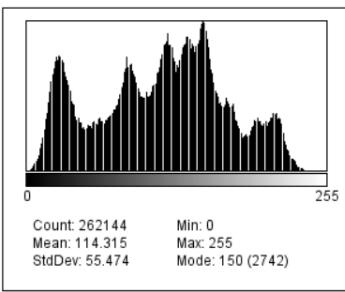
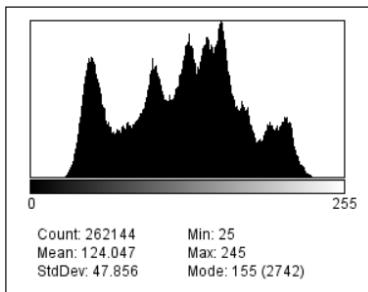
5



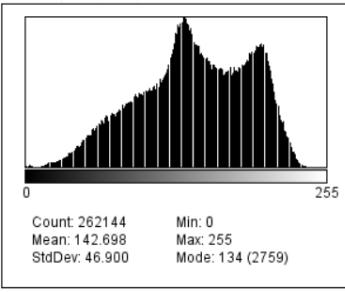
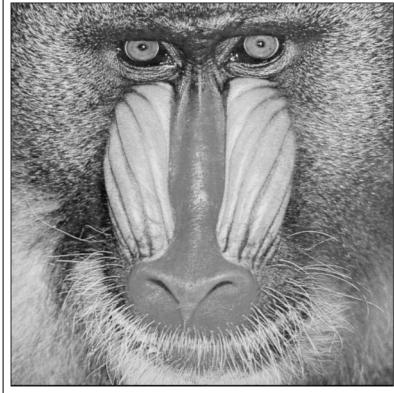
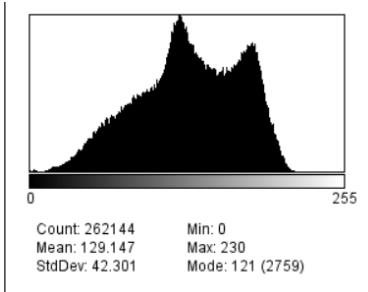
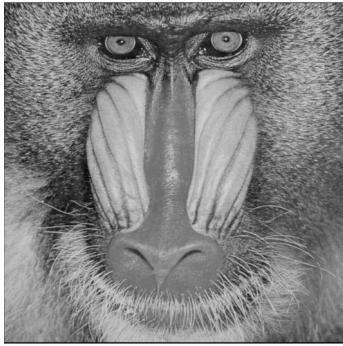
6

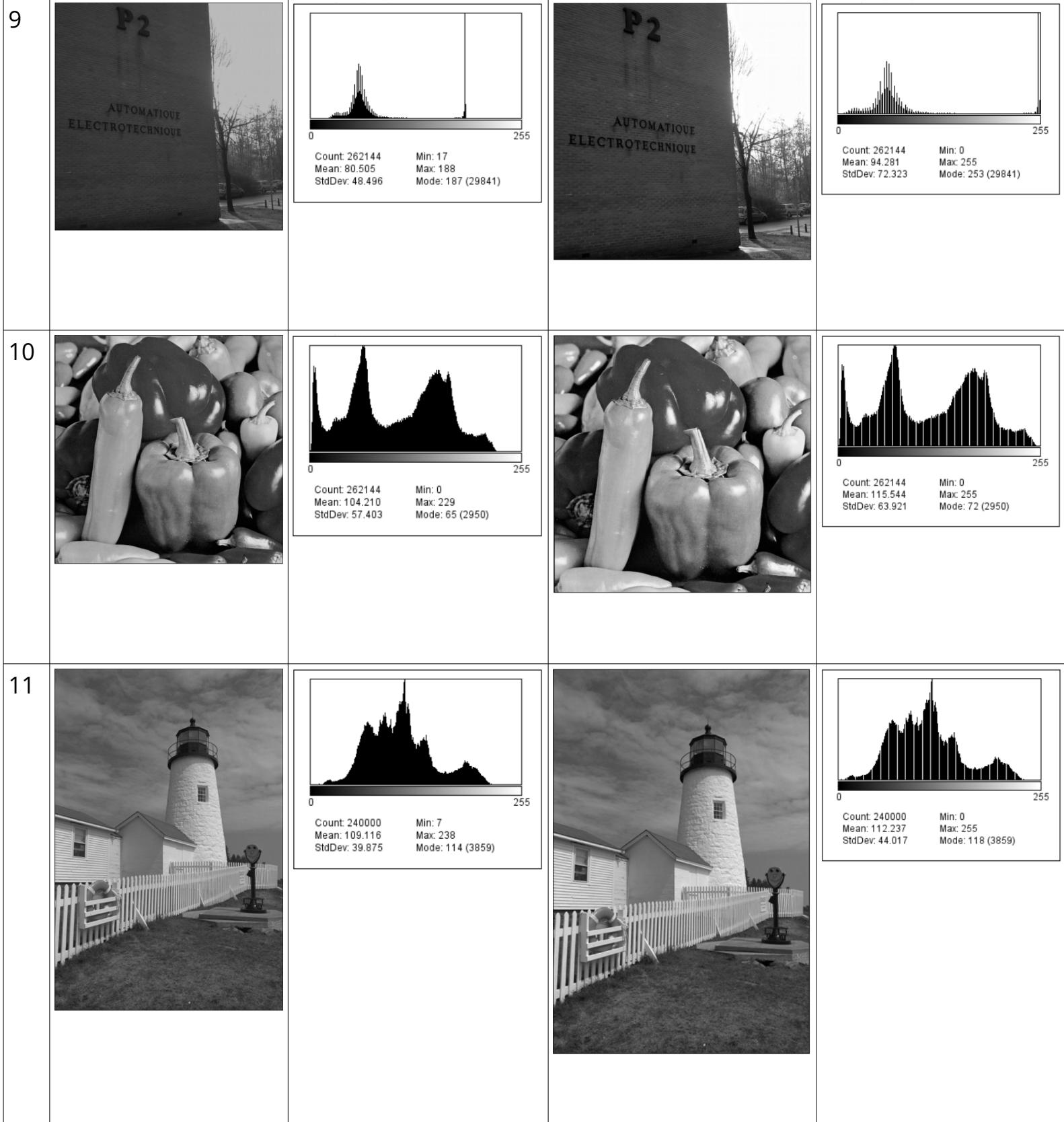


7



8





**N.B.** : Le fait que des images transformées et initiales n'aient pas les mêmes dimensions dans le tableau est dû à une différence d'affichage à cause de l'éditeur de texte utilisé pour ce rapport. Dans la réalité, les images initiales et transformées sont de mêmes dimensions.

Nous pouvons observer plusieurs choses

# Transformation ponctuelle non invariante par translation

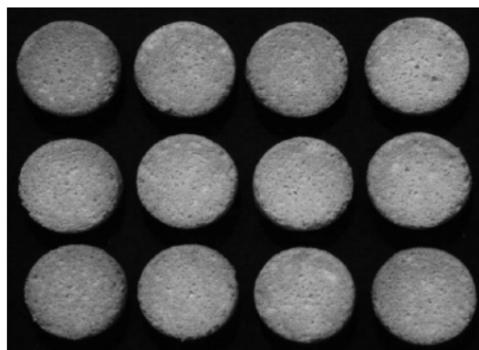
---

## Objectif de l'expérience

L'objectif de cette partie du TP est d'appliquer une transformation affine sur une image (en niveaux de gris) et d'interpréter les résultats obtenus.

Pour ce faire, un travail sera réalisé :

- Essayer d'obtenir une image transformée avec un fond complètement noir (niveau de gris 0) et des objets au contraste maximale pour l'image suivante :



## Identification du problème

Le sujet du problème pour cette partie du TP est donc les transformations affines dont l'expression générale est :

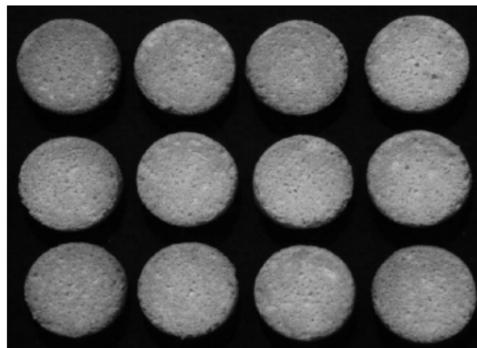
$$g(x, y) = a + b * I(x, y) \text{ , où :}$$

- $g(x, y)$  est le niveau de gris du pixel situé à la ligne  $x$  et la colonne  $y$  sur l'image transformée.
- $I(x, y)$  est le niveau de gris du pixel situé à la ligne  $x$  et la colonne  $y$  sur l'image initiale.
- $a$  est l'**offset**
- $b$  est le **gain**

L'objectif de cette partie du TP est d'identifier comment l'offset et le gain de la transformation affine peuvent influencer l'image obtenue.

## Explication du procédé

Nous commencerons par travailler sur l'image des « cakes », à savoir transformer l'image (ci-dessous) pour que l'image transformée ait un fond noir et un contraste maximal au niveau des objets.

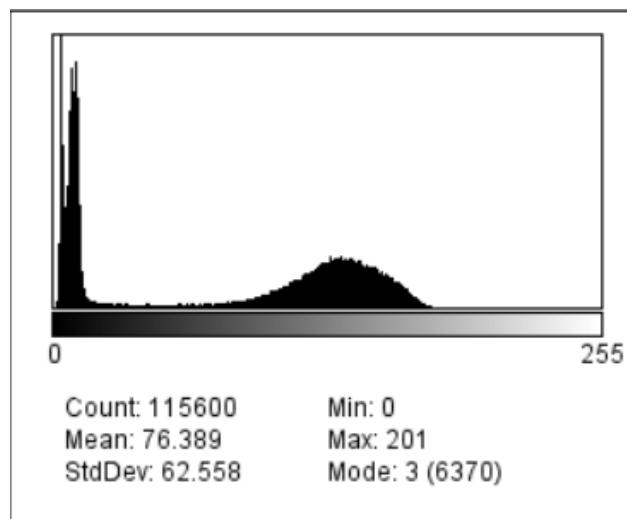


*L'image « cakes » à transformer*

Plusieurs observations peuvent être faites sur cette image :

- Le fond de l'image est très sombre et est donc composé de pixel qui ont des niveaux de gris bas.
- Les objets (les « cakes ») sont plus claires que le fond et possèdent donc des valeurs de niveaux de gris plus hautes que le fond.

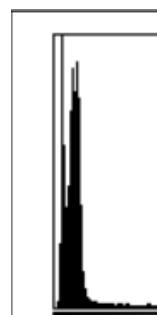
L'histogramme des niveaux de gris de cette image est le suivant :



*Histogramme des niveaux de gris de « cakes »*

Nous y observons l'apparition de deux grands « regroupements » de pixels pour des niveaux de gris faibles et plus élevés. Comme nous savons que le fond est composé de pixels de niveaux de gris bas et que les pixels des objets ont des niveaux de gris plus élevés, nous sommes en mesure d'assumer que :

- Le regroupement de pixels de niveaux de gris faibles est composé essentiellement des pixels du fond.



*Regroupement des pixels du fond*

- Le regroupement de pixels de niveaux de gris plus élevés est composé essentiellement des pixels des cakes.



*Regroupement des pixels des cakes*

Ce que nous voulons faire est donc de faire en sorte que les niveaux de gris des pixels du fond soient tous à 0 et de répartir le regroupement des pixels des cakes sur des niveaux de gris entre 0 et 255 (contraste maximale).

Pour ce faire, nous devons :

- Définir les niveaux de gris maximum et minimum des pixels composant les cakes
- Modifier les niveaux de gris de l'image de telle sorte que la dynamique des niveaux de gris entre les niveaux de gris minimum et maximum soit comprise entre 0 et 255 dans l'image transformée (comme vue dans la partie précédente).

Pour les bornes minimales et maximales des niveaux de gris des cakes, nous choisissons (à la main) les valeurs suivantes :

- Niveau de gris minimum  $I_{min}(x, y) = 80$
- Niveau de gris maximum  $I_{max}(x, y) = 180$

Nous obtenons, comme pour la partie précédente, un système d'équation à deux inconnues suivant :

$$\begin{aligned}0 &= 80b + a \\255 &= 180b + a\end{aligned}$$

Après une résolution du système d'équation, nous obtenons les valeurs suivantes pour b (le gain) et a (l'offset) :

- $a = -204$
- $b = 2,55$

Ce qui nous donne la formule à appliquer pour chaque pixel de l'image :

$$g(x, y) = 2,55I(x, y) - 204$$

Il ne nous reste plus qu'à appliquer cette transformation sur tous les pixels de l'image. De cette manière :

- Les pixels du fond de l'image seront mis à 0
- Les pixels des cakes seront répartis entre 0 et 255 (contraste maximale)

## Implémentation sous ImageJ

```
image = getImageID();
W = getWidth();
H = getHeight();

// définir l'offset (à la main)
offset = -204;

// définir le gain (à la main)
gain = 2.55;

// POUR CHAQUE [pixel]
for (j=0; j<H; j++)
{
    for (i=0; i<W; i++)
    {
        p = getPixel(i,j);
        // appliquer la transformation affine avec le gain et l'offset définis
        // g(x,y) = b*i(x,y) + a
        pnew = offset + gain*p;

        // modifier
        setPixel(i,j,pnew);
    }
}
```

L'implémentation de la macro permettant d'appliquer une transformation affine est simple : il suffit de définir à la main un gain et un offset et d'appliquer la transformation ponctuelle sur chaque pixel.

## Interprétation des résultats

La transformation affine appliquée sur les cakes donne les résultats suivants :

	Initial	Transformé
Image		
Histogramme des niveaux de gris	<p>Count: 115600 Mean: 76.389 StdDev: 62.558</p>	<p>Count: 115600 Mean: 71.010 StdDev: 76.549</p>

# Égalisation d'histogramme

---

## Objectif de l'expérience

Le but de cette partie du TP est d'égaliser l'histogramme des niveaux de gris d'une image et d'interpréter les résultats obtenus grâce à l'histogramme cumulé de l'image résultante de l'égalisation.

## Explication du procédé

L'égalisation d'histogramme de niveaux de gris pour une image  $g_j$  est calculée par la formule suivante :

$$g_j = \frac{g_{max}}{N} h_c(f) \text{ où}$$

- $N$  est le nombre total de pixel
- $g_{max}$  est le niveau de gris maximale de l'image transformée (ici, ce sera toujours 255)
- $h_c(f)$  l'histogramme cumulé pour le niveau de gris  $f$

Les actions à réaliser pour ce TP sont donc :

- Calculer l'histogramme cumulé des niveaux de gris pour l'image (pour effectuer la transformation).
- Transformer l'image

- Calculer et afficher l'histogramme et l'histogramme cumulé de l'image résultante

## Implémentation sous ImageJ

L'implémentation sous ImageJ est la suivante :

- Calcul de l'histogramme cumulé de l'image à transformer

```
// tableau représentant l'histogramme cumulé pour chaque niveau de gris
histogrammeCumule = newArray(bins);
```

```
for (i=0; i<bins; i++) {
    count = counts[i];
    if (count>0) {
        n += count;
        sum += count*i;
        if (i<min) min = i;
        if (i>max) max = i;
    }
    // L'histogramme cumulé d'un niveau de gris est le
    // nombre de pixels ayant un niv de gris inférieur
    // au niveau de gris calculé
    histogrammeCumule[i] = n;
}
```

- Transformation de l'image (égalisation)

```

W = getWidth();
H = getHeight();
image = getImageID();

// realisation de l'égalisation
for(j=0; j<H; j++)
{
    for(i=0; i<W; i++)
    {
        // POUR CHAQUE pixel
        p = getPixel(i,j);

        // application de la formule vue en cours
        // #####
        // La nouvelle valeur du niveau de gris du pixel est
        // La valeur maximale du niveau de gris dans l'image résultante (ici, on veut 255)
        // normalisée par le nombre total de pixel, le tout divisé par la valeur de l'histogramme
        // cumulé pour le niveau de gris initial calculé.
        egalisation = (255/n)*histogrammeCumule[p];
        setPixel(i,j,egalisation);
    }
}

```

- Calculer et afficher l'histogramme et l'histogramme cumulé des niveaux de gris de l'image résultante

```

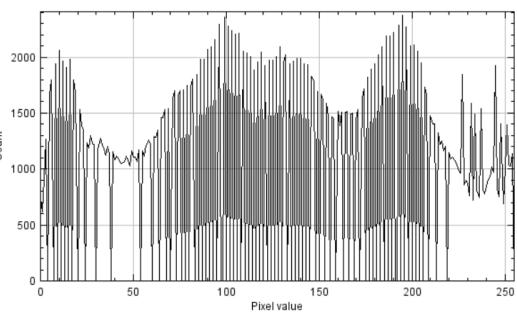
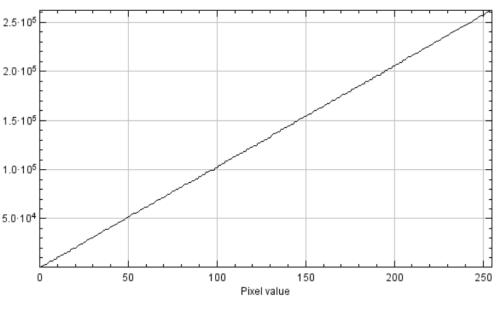
// calcul de l'histogramme transformé
getHistogram(values, counts, bins)
Plot.create("Histogramme normalisé de l'image résultante", "Pixel value", "Count", values, counts);
n = 0;
// calcul de l'histogramme cumulé du résultat
for (i=0; i<bins; i++) {
    n += counts[i];
    // L'histogramme cumulé d'un niveau de gris est le
    // nombre de pixels ayant un niv de gris inférieur
    // au niveau de gris calculé
    histogrammeCumule[i] = n;
}

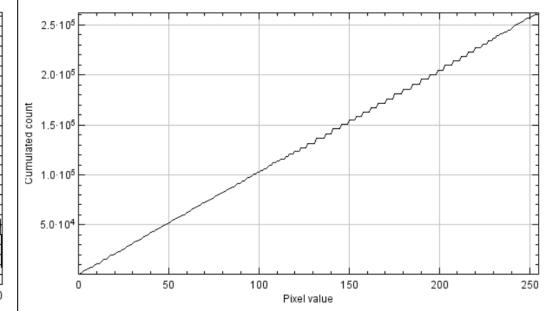
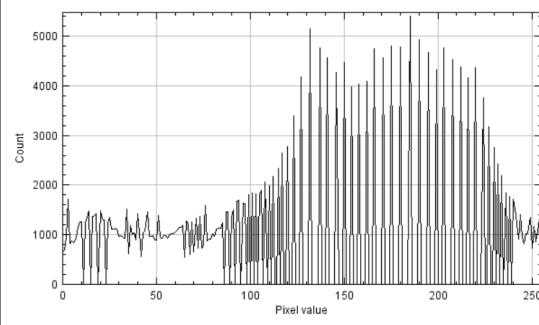
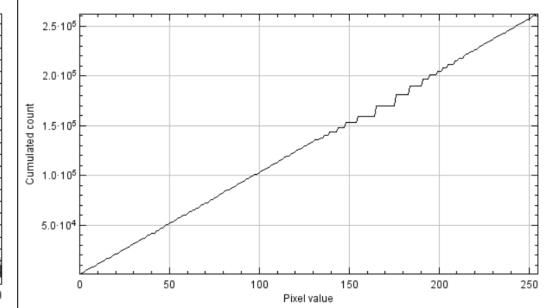
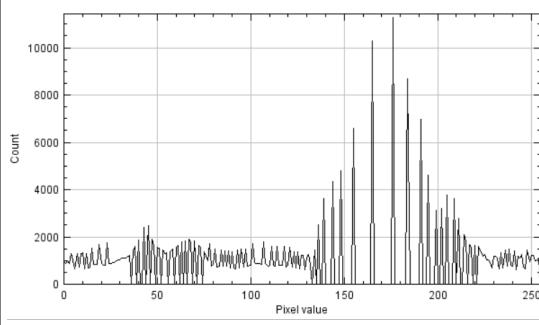
Plot.create("Histogramme cumulé après transformation", "Pixel value", "Cumulated count", values, hi

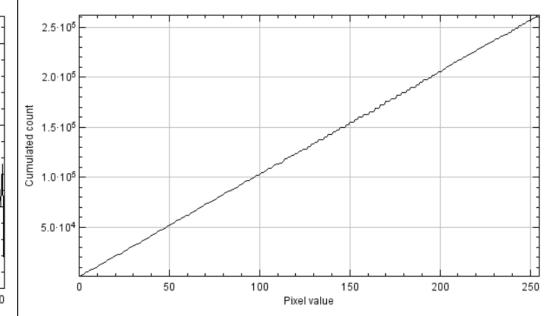
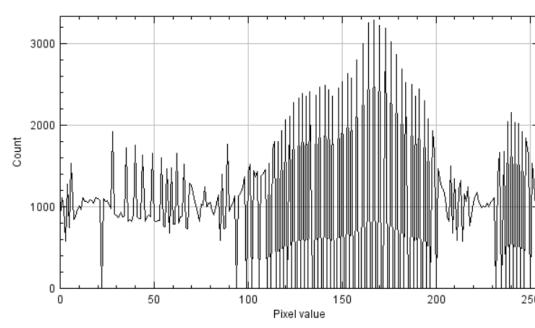
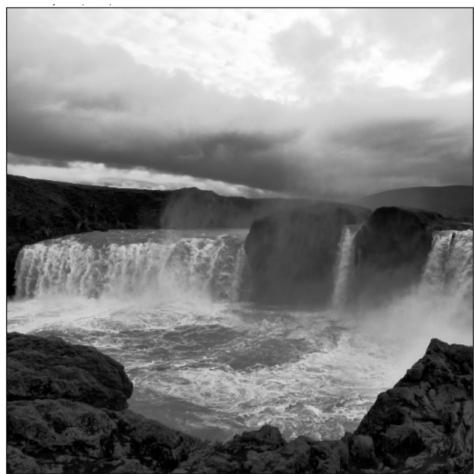
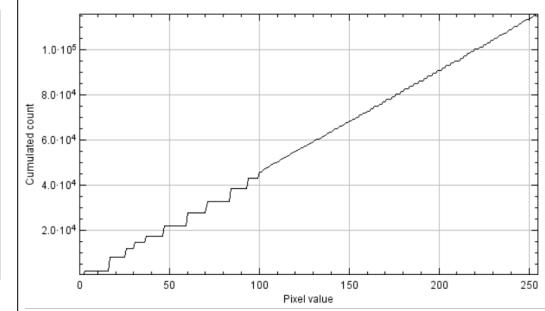
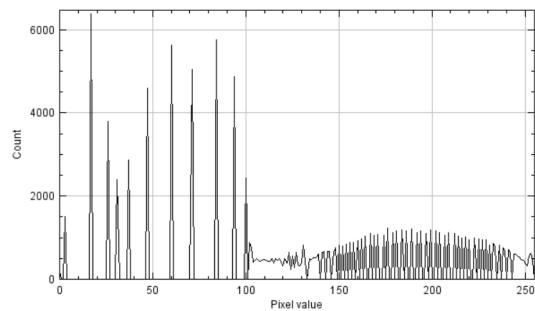
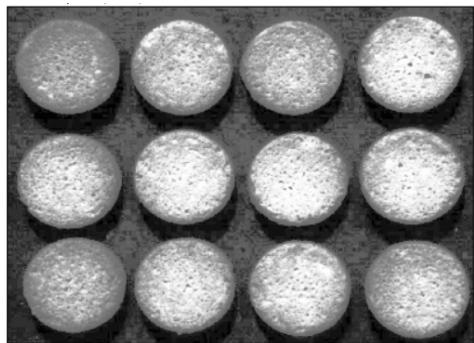
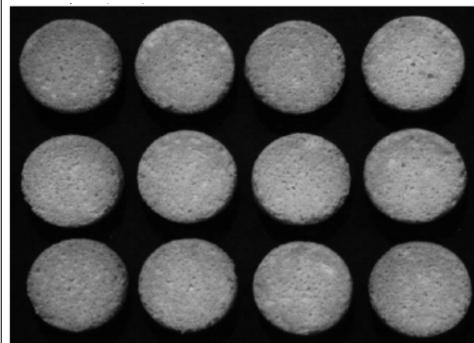
```

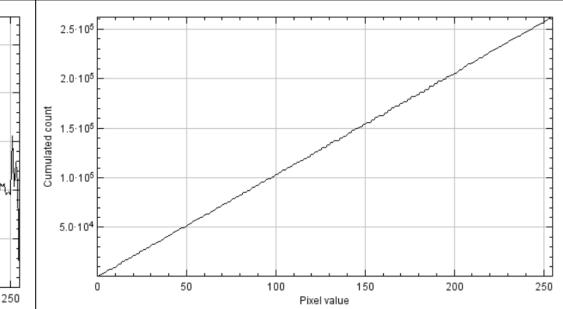
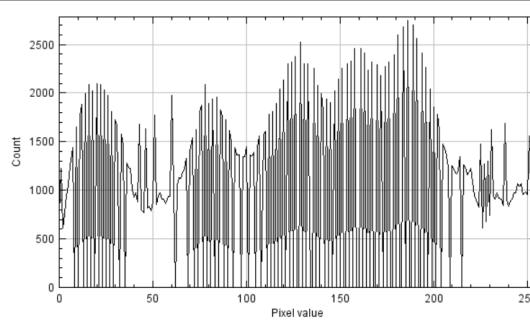
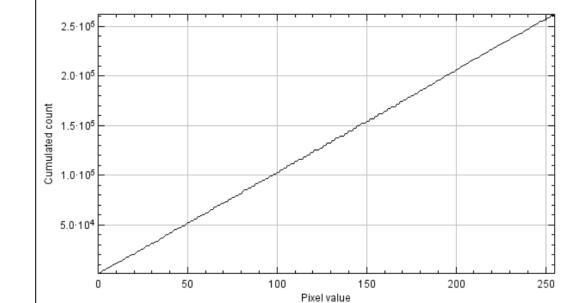
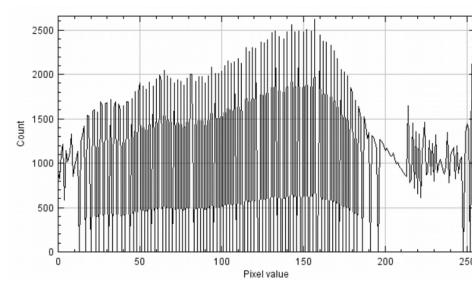
## **Interprétation des résultats**

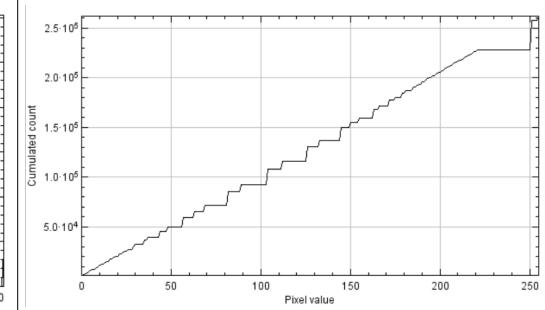
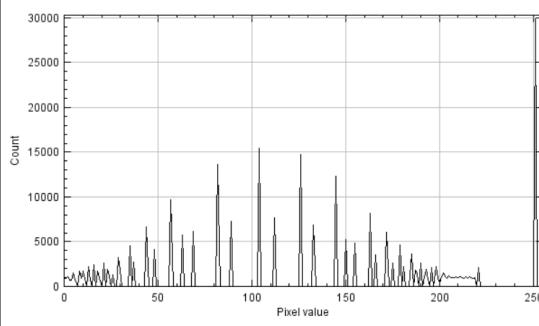
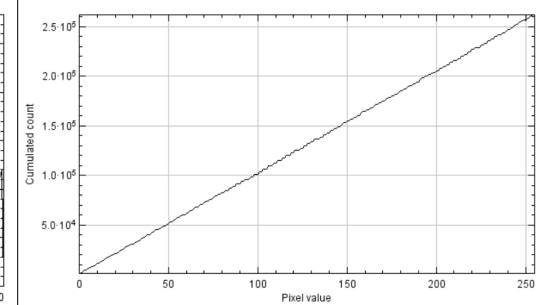
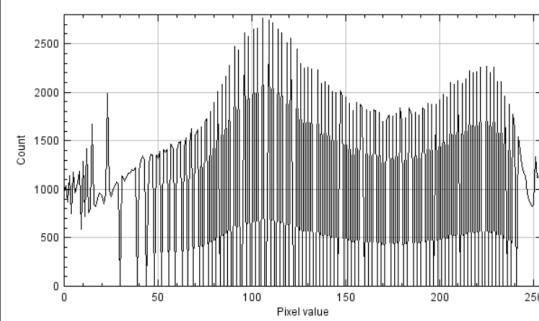
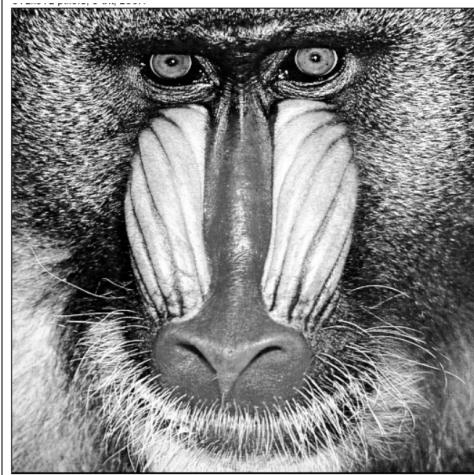
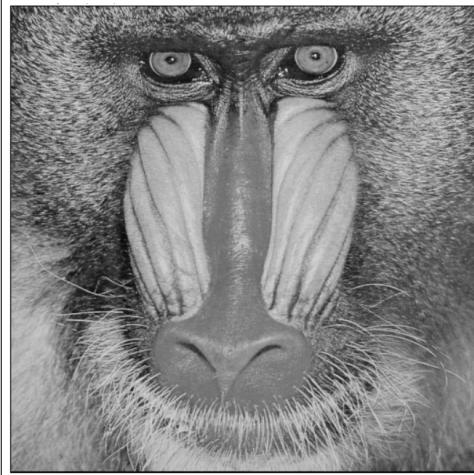
Ci-dessous se trouve le tableau des observations et mesures faites par rapport à l'égalisation implémentée :

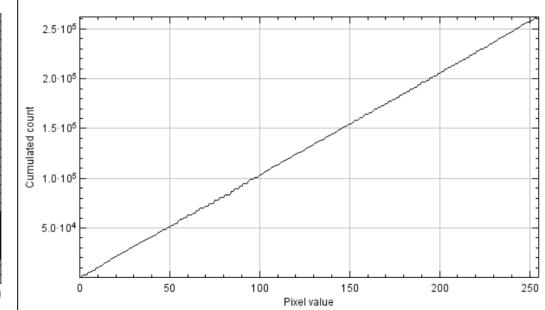
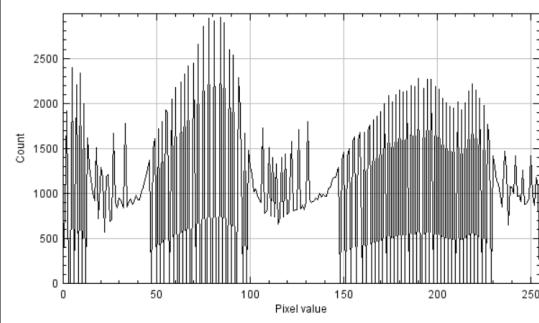
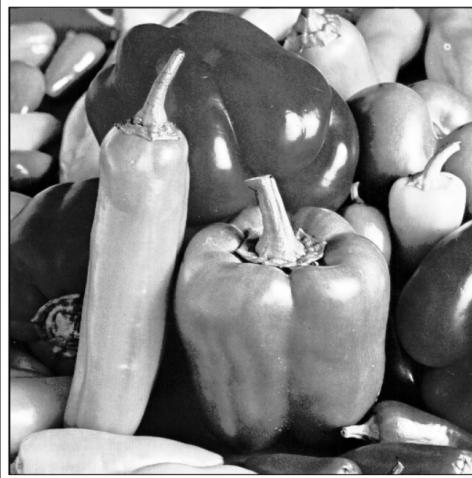
Image	Image transformée	Histogramme de l'image résultante	Histogramme cumulé de l'image résultante
			

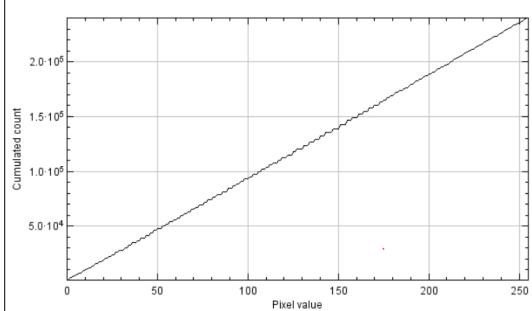
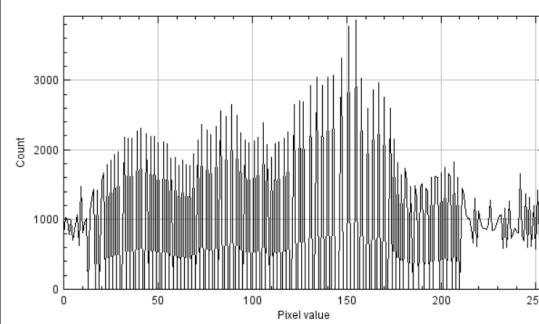












On remarque que, pour toutes les images, nous obtenons un histogramme cumulé qui ressemble à une droite, ce qui est normal puisque l'égalisation est censé affecter le même nombre de pixels pour chaque niveau de gris.

Cependant, pour des images dont la répartition des niveaux de gris est moins uniforme (exemple : la photo du P2), on constate que l'histogramme cumulé décrit, à un certain endroit, un « escalier ». Ceci est dû au fait qu'en pratique, il est impossible d'obtenir une parfaite uniformité de la distribution des niveaux de gris car l'égalisation ne porte que sur la répartition globale des niveaux de gris.