

*UNIVERSITÉ TUNIS EL MANAR*  
-----O-----  
*INSTITUT SUPÉRIEUR D'INFORMATIQUE*

*ANALYSE D'INFORMATION MULTIMÉDIA*



# ANALYSE D'INFORMATION MULTIMÉDIA

3<sup>ème</sup> année Licence SIL

AU 2015-2016

# PLAN

- ❖ Rappel
  - Qu'est ce qu'une image
  - Le pixel
  - La définition
  - La résolution
  - Codage des couleurs
- ❖ Problématique
- ❖ Introduction
  - Histogramme
  - Dynamique
  - Luminance
  - Contraste
- ❖ Rehaussement de l'image
  - Binarisation
  - Recadrage de la dynamique
  - Egalisation de l'histogramme
  - Négatif de l'image
- ❖ Filtrage
  - Convolution
  - Effet de bord
  - Filtrage passe-bas
    - ❑ Filtre moyenneur
    - ❑ Filtre médian
    - ❑ Filtre gaussien

# PLAN

- Rappel
  - Qu'est ce qu'une image
  - Le pixel
  - La définition
  - La résolution
  - Codage des couleurs
- Problématique
- Introduction
  - Histogramme
  - Dynamique
  - Luminance
  - Contraste
- Rehaussement de l'image
  - Binarisation
  - Recadrage de la dynamique
  - Egalisation de l'histogramme
  - Négatif de l'image
- Filtrage
  - Convolution
  - Effet de bord
  - Filtrage passe-bas
    - Filtre moyennneur
    - Filtre médian
    - Filtre gaussien

# RAPPEL (1/6)

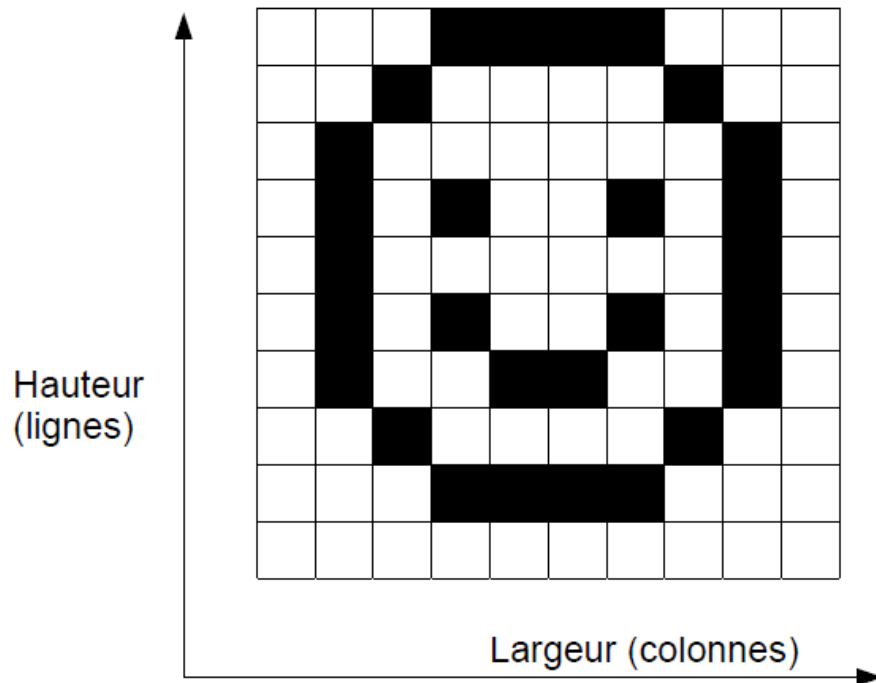
## QU'EST CE QU'UNE IMAGE

- ❖ Souvent, cette image représente une réalité 3D  $(x,y,z)$
- ❖ Une image est avant tout un signal 2D  $(x,y)$
- ❖ D'un point de vue mathématique : Une image est un matrice de nombres représentant un signal.
- ❖ D'un point de vue humain :
  - Une image contient plusieurs informations sémantiques.
  - Il faut interpréter le contenu au-delà de la valeur des nombres.

# RAPPEL (2/6)

## LE PIXEL

- ❖ Une image numérique est constituée d'un ensemble de points appelés **pixels** (abréviation de *PIC*ture *E*lement) pour former une image.
- ❖ *Le pixel représente ainsi le plus petit élément constitutif* d'une image numérique.
- ❖ L'ensemble de ces pixels est contenu dans un tableau à deux dimensions constituant l'image



# RAPPEL (3/6)

## LA DÉFINITION

- ❖ On appelle **définition** le nombre de points (pixels) constituant une image:
  - c'est le nombre de colonnes de l'image que multiplie son nombre de lignes. Une image possédant 10 colonnes et 11 lignes aura une définition de 10 x 11.

	1	2	3	4	5	6	7	8	9	10
1										
2										
3										
4										
5										
6										
7										
8										
9										
10										
11										

- ❖ Formule de calcul du nombre total des pixels dans une image:
  - *Nombre total des pixels = colonnes x lignes.*

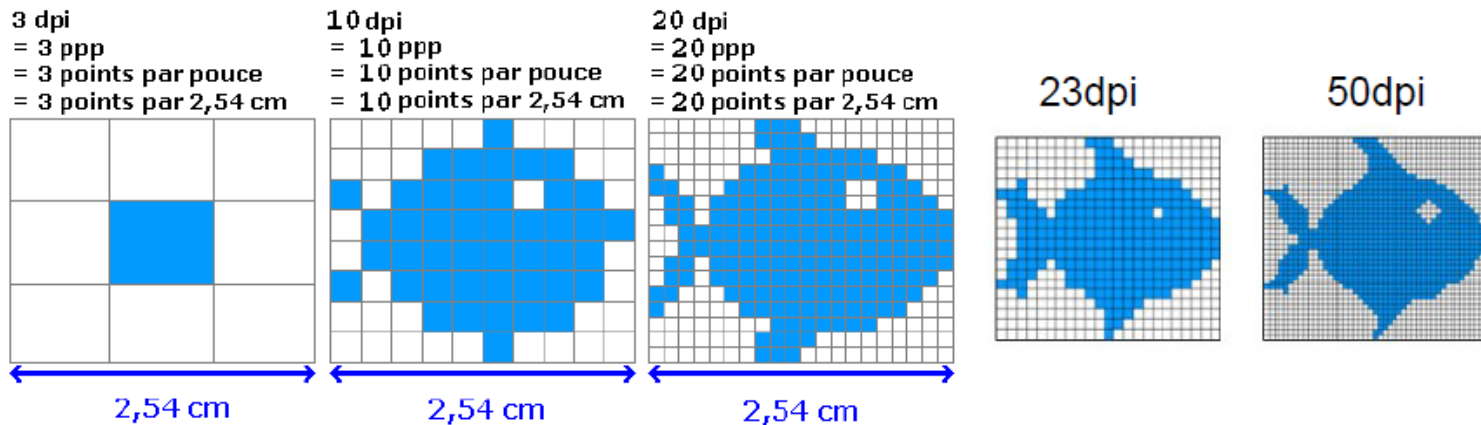
Exemple:  $10 \times 11 = 110$  pixels au total pour l'image ci-dessus.

# RAPPEL (4/6)

## LA RÉOLUTION

- C'est le nombre de points contenu dans une **longueur donnée** (en pouce). Elle est exprimée en **points par pouce** (PPP, en anglais: DPI pour *Dots Per Inch*). Un pouce mesure 2.54 cm, c'est une unité de mesure britannique.

- La **résolution** permet ainsi d'établir le rapport entre la **définition en pixels** d'une image et la **dimension réelle** de sa représentation sur un support physique (écran, papier...)



**Formule :** Calculer la résolution à partir de la définition et de la dimension

**Résolution** = définition / dimension

# RAPPEL (5/6)

## CODAGE DES COULEURS

- ❖ Une image numérique utilise plus ou moins de mémoire selon le codage des informations de couleur qu'elle possède. C'est ce que l'on nomme le **codage de couleurs ou profondeur des couleurs, exprimé en bit par pixel (bpp): 1, 4, 8, 16 bits...**
- ❖ En connaissant le **nombre de pixels d'une image et la mémoire nécessaire à l'affichage d'un pixel**, il est possible de définir exactement le poids que va utiliser le fichier image sur le disque dur
- ❖ Formule de calcul du poids d'une image en octet:

$$\text{Poids (octet)} = \text{Nombre de pixel total} \times \text{codage couleurs(octet)}$$



# RAPPEL (6/6)

## RAPPEL DU CODE BINAIRE

### ❖ On sait que:

- 1bit = 2 états; (0 ou 1) =  $2^1$
- 2bits = 4 états, =  $2^2$
- 4bits = 16 états, =  $2^4$
- 8bits = 256 états, =  $2^8$  etc...

### ❖ Un ensemble de 8bit = 1 *Octet*:

- 1024 Octets = 1 kilo-octet (Ko).
- 1024 Kilo-Octets = 1 Mega-Octet (Mo)
- 1024 Mega-Octets = 1 Giga-Octet (Go)
- 1024 Giga-Octets = 1 Tera-Octet (To)

# PLAN

- ❖ Rappel
  - Qu'est ce qu'une image
  - Le pixel
  - La définition
  - La résolution
  - Codage des couleurs
- ❖ Problématique
- ❖ Introduction
  - Histogramme
  - Dynamique
  - Luminance
  - Contraste
- ❖ Rehaussement de l'image
  - Binarisation
  - Recadrage de la dynamique
  - Egalisation de l'histogramme
  - Négatif de l'image
- ❖ Filtrage
  - Convolution
  - Effet de bord
  - Filtrage passe-bas
    - ❑ Filtre moyenneur
    - ❑ Filtre médian
    - ❑ Filtre gaussien



## PROBLÉMATIQUE (1/2)

- ❖ L'acquisition d'images s'accompagne toujours d'une distorsion/dégradation.
  - ❖ Différentes sources de dégradation (bruit) d'une image:
    - ***Bruit lié au contexte de l'acquisition :***
      - ▣ Bougé, mauvaises conditions d'éclairage,...
    - ***Bruit lié au capteur :***
      - ▣ Capteur de mauvaise qualité, mauvaise mise au point, etc...
    - ***Bruit lié à l'échantillonnage :***
      - ▣ Une mauvaise fréquence d'échantillonnage peut introduire dans l'image des points blancs ou noirs connus souvent sous l'appellation «poivre et sel».
    - ***Bruit lié à la nature de la scène :***
      - ▣ Présence de fumée, de nuage, etc...
- ➔ Il faut corriger l'image par un procédé algorithmique:
- Rehaussement
  - Restauration & filtrage

## PROBLÉMATIQUE (2/2)

### ❖ *Rehaussement:*

- Donner à l'image un aspect visuellement correct.
- Le rehaussement repose donc sur des critères subjectifs. Il est de nature empirique

### ❖ *Restauration & filtrage:*

- Retrouver autant que possible l'image originale telle qu'elle était avant sa dégradation.
- La restauration repose sur des critères objectifs : Chercher une estimation de telle sorte qu'elle minimise une fonction d'erreur.
- La restauration peut s'appuyer sur une fonction d'erreur précise par exemple l'erreur quadratique moyenne :

$$EQM = \frac{1}{P * Q} \cdot \sum_{i=0}^{P-1} \sum_{j=0}^{Q-1} [I(i,j) - \hat{I}(i,j)]^2$$

# PLAN

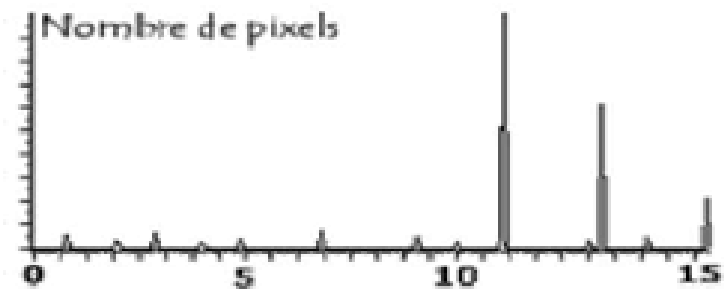
- ❖ Rappel
  - Qu'est ce qu'une image
  - Le pixel
  - La définition
  - La résolution
  - Codage des couleurs
- ❖ Problématique
- ❖ Introduction
  - Histogramme
  - Dynamique
  - Luminance
  - Contraste
- ❖ Rehaussement de l'image
  - Binarisation
  - Recadrage de la dynamique
  - Egalisation de l'histogramme
  - Négatif de l'image
- ❖ Filtrage
  - Convolution
  - Effet de bord
  - Filtrage passe-bas
    - ❑ Filtre moyenneur
    - ❑ Filtre médian
    - ❑ Filtre gaussien

# INTRODUCTION

## HISTOGRAMME (1/5)

- ❖ Soit une image définie par une fonction  $f(x,y)$  appartenant à l'intervalle  $[0, L-1]$
- ❖ L'histogramme de  $f$  est une fonction  $hist(l)$  qui est égale au nombre de pixels ayant le niveau de gris  $l$ . (i.e. la distribution des valeurs de niveaux de gris dans l'image)

hist:[0,L-1]    N  
      L    hist(l)



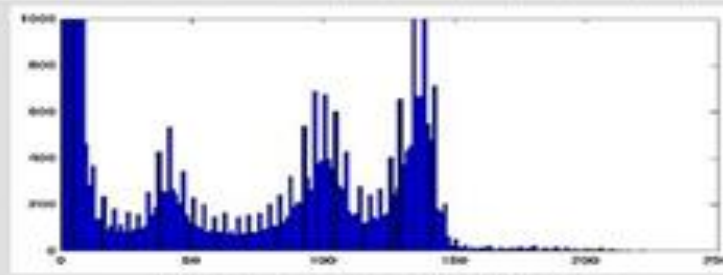
# INTRODUCTION

## HISTOGRAMME (2/5)

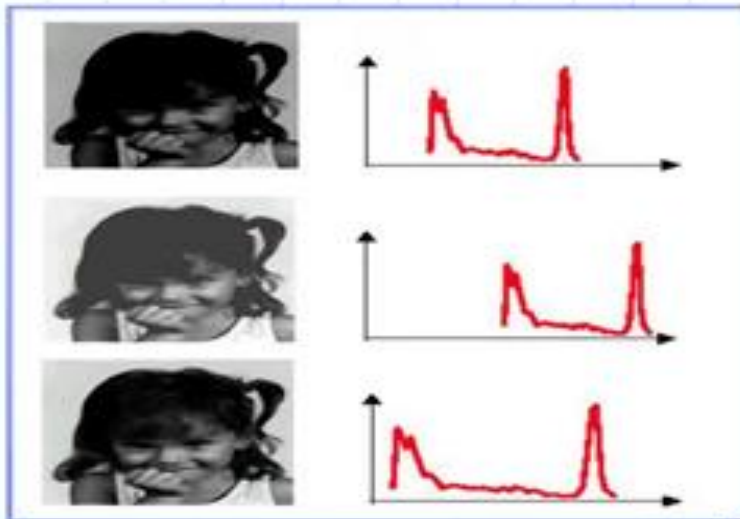
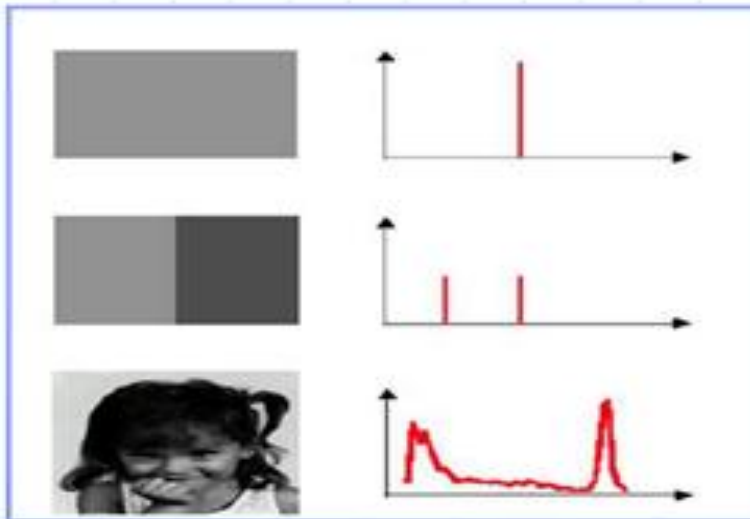
Exemples d'histogramme des niveaux de gris



Image originale



Histogramme de l'image



# INTRODUCTION

## HISTOGRAMME (3/5)

Algorithme de l'histogramme

*Histogramme (Matrice I [ nbcol][nbligne ], entier nbcol, entier nbligne)*

*Debut*

*H = new Array [0..255]*

*/\*Initialisation du tableau\*/*

*Pour i= 0 à 255*

*H[i]= 0;*

*Fin pour*

*Pour i= 0 à nbcol*

*Pour j= 0 à nbligne*

*H[ I[i][j] ] ++;*

*Fin pour*

*Fin pour*

*dessiner(H);*

*Fin*



# INTRODUCTION

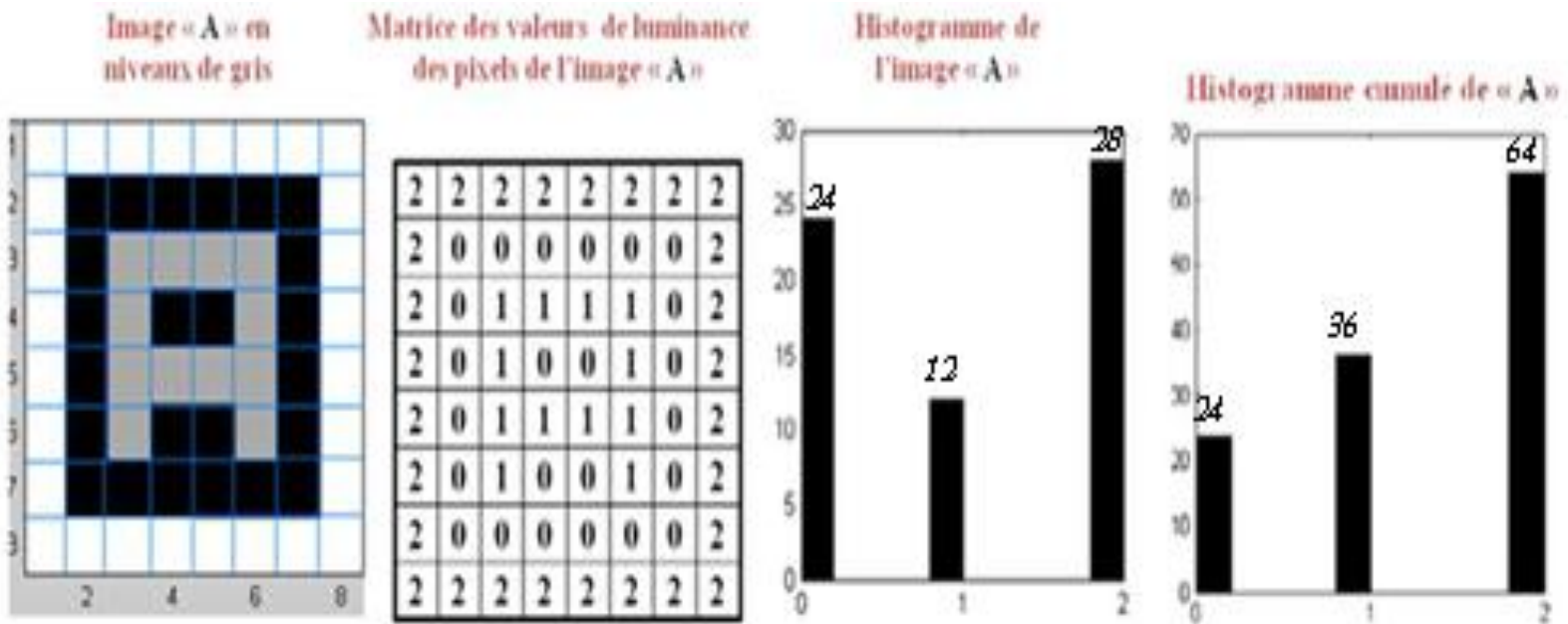
## HISTOGRAMME (4/5)

### Histogramme cumulé

- ❖ L'histogramme cumulé est défini par:

$$Hist\_Cumulé(l) = \sum_{k=0}^l Hist(k)$$

- ❖ Sachant que  $hist(k)$  est non négatif, l'histogramme cumulé est donc une fonction monotone et croissante.
- ❖ Exemple:



# INTRODUCTION

## HISTOGRAMME (5/5)

Algorithme de l'histogramme cumulé

*HistogrammeCumulé (Tableau HistOrigin[255])*

*Debut*

*Tableau HCumul = new Array [0..255]*

*HCumul[0] = HistOrigin[0];*

*Pour i= 1 à 255*

*HCumul[i] = HCumul[i-1] + HistOrigin[i];*

*Fin pour*

*dessiner(HCumul);*

*Fin*

# INTRODUCTION

## DYNAMIQUE

### ❖ *Dynamique de l'image*

- Généralement, les valeurs des pixels occupent une partie de cet intervalle comprise entre une valeur minimale et une valeur maximale. L'intervalle ainsi défini  $[\min, \max]$  est appelé la dynamique de l'image.

### ❖ *Dynamique maximale*

- La valeur d'un pixel d'une image en niveaux de gris appartient à l'intervalle  $[0, 2N-1]$  généralement  $[0, 255]$ . On dit que cet intervalle est la dynamique maximale d'une image.

# INTRODUCTION

## LUMINANCE OU BRILLANCE D'UNE IMAGE

- ❖ La luminance (ou brillance) est définie par la moyenne de tous les pixels de l'image:

$$lum = Moy(P(i, j)) = \sum_i \sum_j \frac{P(i, j)}{M \times N}$$

- ❖ Dans les deux images suivantes, seule la luminance est différente :



# INTRODUCTION

## CONTRASTE D'UNE IMAGE

- ❖ Le contraste est défini de la manière suivante:

$$C = \frac{\text{Max}(P(i, j)) - \text{Min}(P(i, j))}{\text{Max}(P(i, j)) + \text{Min}(P(i, j))}$$

➔ *C'est la variation entre niveau de gris min et max*

- ❖ Les deux images suivantes possèdent un contraste différent:



# PLAN

- ❖ Rappel
  - Qu'est ce qu'une image
  - Le pixel
  - La définition
  - La résolution
  - Codage des couleurs
- ❖ Problématique
- ❖ Introduction
  - Histogramme
  - Dynamique
  - Luminance
  - Contraste
- ❖ Rehaussement de l'image
  - Binarisation
  - Recadrage de la dynamique
  - Egalisation de l'histogramme
  - Négatif de l'image
- ❖ Filtrage
  - Convolution
  - Effet de bord
  - Filtrage passe-bas
    - ❑ Filtre moyenneur
    - ❑ Filtre médian
    - ❑ Filtre gaussien

# REHAUSSEMENT DE L'IMAGE

- ❖ Supposons que  $f(x,y) \in [0,L-1]$
- ❖ Le rehaussement consiste généralement à appliquer une fonction :

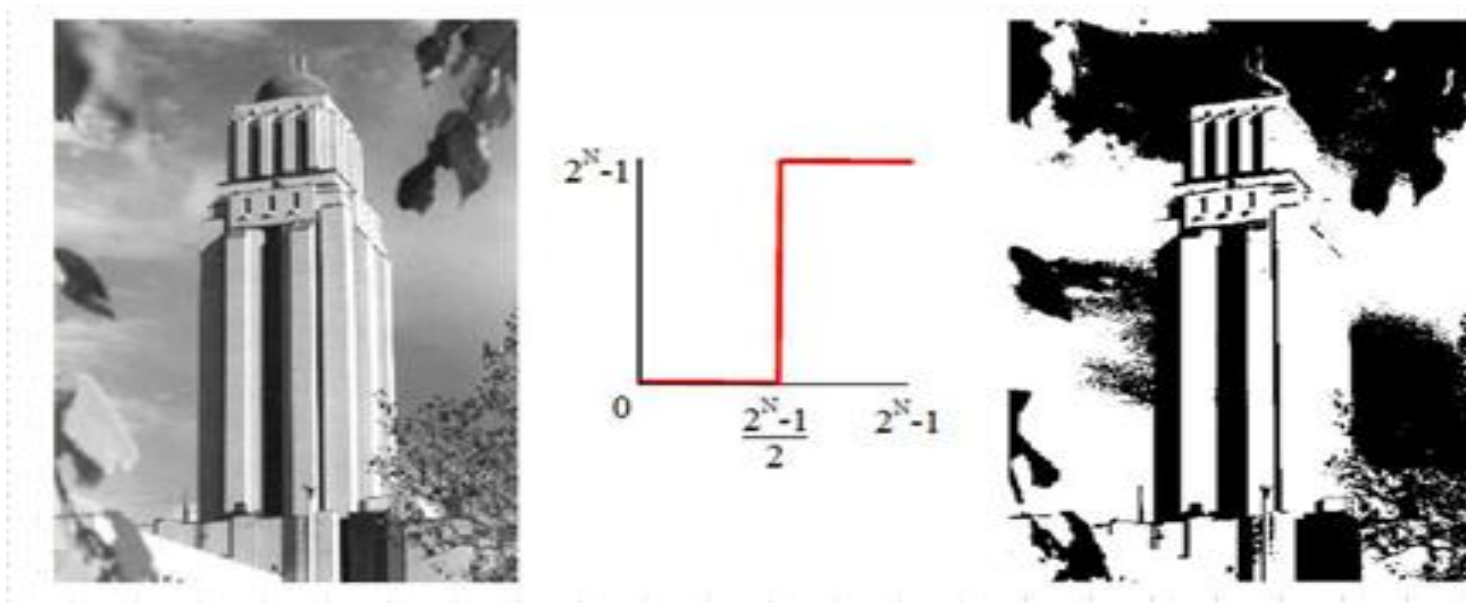
$$T : [0,L] \rightarrow [0,L]$$

$$L \rightarrow T(l)$$

- $T$  croissante pour que la relation d'ordre soit maintenue,
  - Le problème revient donc à trouver une telle fonction  $T$
- ❖ Pour trouver une telle fonction, il existe différentes méthodes:
  - Binarisation de l'image (seuillage)
  - Etirement de contraste ou recadrage de la dynamique
  - Egalisation de l'histogramme
  - Négatif d'une image

# REHAUSSEMENT DE L'IMAGE

## BINARISATION (1/2)



- ❖ Un seuil fixe à  $((2N-1)/2)$  → Risque de ne rien présenter  
→ Perte de l'information
- ❖ Généralement on choisit la valeur médiane entre deux seuils  $(Max+Min)/2$



# REHAUSSEMENT DE L'IMAGE

## BINARISATION (2/2)

Algorithme de la binarisation

*Binarisation (Matrice Origin[ nbcol] [nbligne ], seuil)*

*Debut*

*Matrice Binaire[ nbcol] [ nbligne]*

*Pour i= 0 à nbcol*

*Pour j= 0 à nbligne*

*Si (Origin[i][j]>= seuil)*

*Binaire[i][j]= 255;*

*Sinon*

*Binaire[i][j]= 0;*

*Fin Si*

*Fin pour*

*Fin pour*

*Fin*

# REHAUSSEMENT DE L'IMAGE

## RECADRAGE DE LA DYNAMIQUE (1/6)

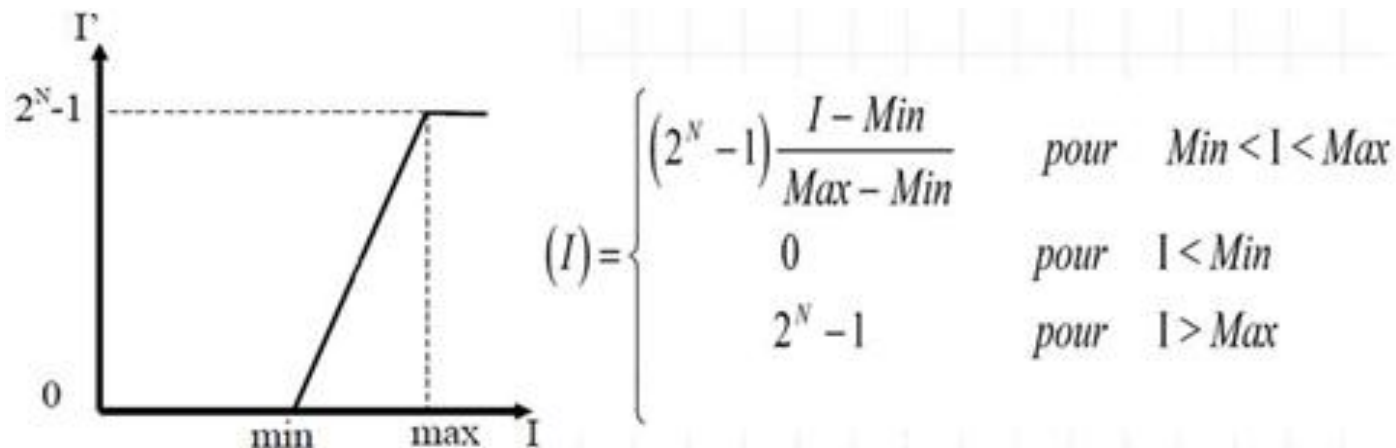
- ❖ On suppose une image de départ présentant un histogramme concentré dans l'intervalle  $[a, b]$ . Les valeurs  $a, b$  correspondent aux niveaux de gris extrêmes présents dans cette image. Le recadrage de dynamique consiste à étendre la dynamique de l'image transformée à l'étendue totale  $[0, 2^N-1]$ . La transformation de recadrage est donc une application affine du niveau de gris des pixels de telle sorte que l'image utilise toute la dynamique de représentation.



# REHAUSSEMENT DE L'IMAGE

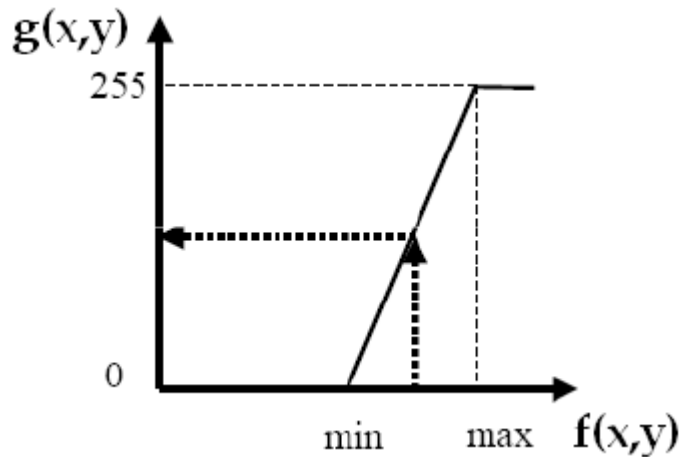
## RECADRAGE DE LA DYNAMIQUE (2/6)

- ❖ Déterminer l'histogramme de l'image
- ❖ Déterminer la dynamique de l'image [min,max]
- ❖ Choisir T telle que:
  - $T[0,\min]=0, T[\max,2N-1]=2N-1$
  - T est continue et linéaire par morceau sur [0, 2N-1]



# REHAUSSEMENT DE L'IMAGE

## RECADRAGE DE LA DYNAMIQUE (3/6)



$$g(x, y) = \alpha f(x, y) + \beta$$

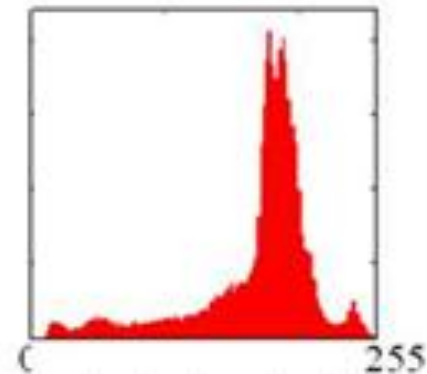
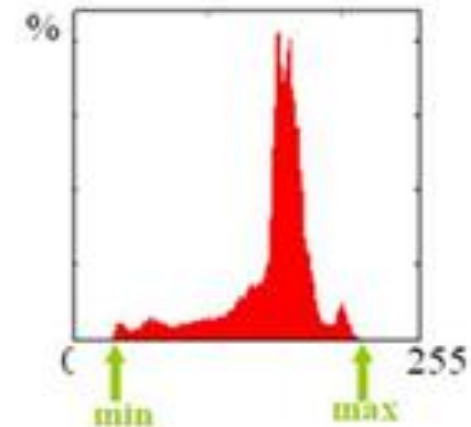
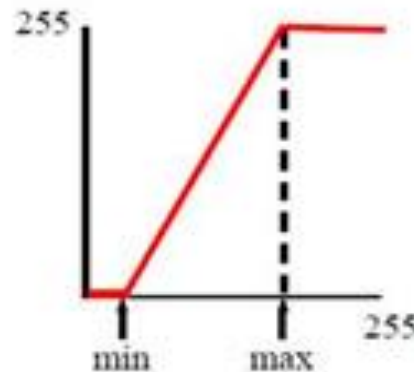
$$\begin{cases} 0 = \alpha \min + \beta \\ 255 = \alpha \max + \beta \end{cases}$$

$$\begin{cases} \alpha = \frac{255}{\max - \min} \\ \beta = -\frac{255}{\max - \min} \min \end{cases}$$

# REHAUSSEMENT DE L'IMAGE

## RECADRAGE DE LA DYNAMIQUE (4/6)

Exemple (1/2)



# REHAUSSEMENT DE L'IMAGE

## RECADRAGE DE LA DYNAMIQUE (5/6)

Exemple (2/2)



Original



Recadrage :  $a = 30$ ,  $b = 200$

# REHAUSSEMENT DE L'IMAGE

## RECADRAGE DE LA DYNAMIQUE (6/6)

Algorithme de l'histogramme recadré

*Recadrage de la dynamique*(Matrice Origin[ nbcol] [nbligne ], min, max)

*Debut*

*Matrice Recadrage*[ nbcol] [ nbligne]

*Pour i= 0 à nbcol*

*Pour j= 0 à nbligne*

*Si (Origin[i][j]== min)*

*Recadrage[i][j]= 0;*

*Si (Origin[i][j]== max)*

*Recadrage[i][j]= 255;*

*Sinon*

*Recadrage[i][j]= 255\*(Origin[i][j]-min)/(max-min);*

*Fin Si*

*Fin pour*

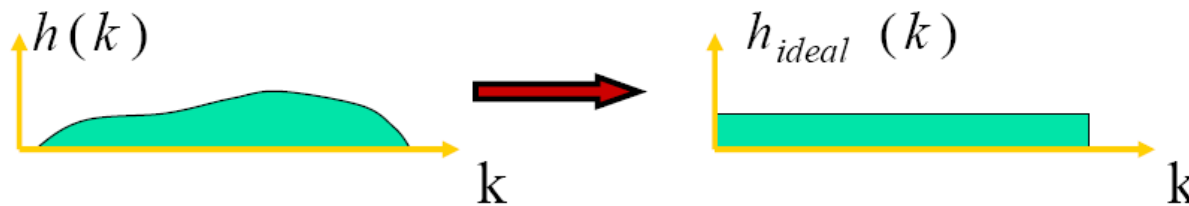
*Fin pour*

*Fin*

# REHAUSSEMENT DE L'IMAGE

## EGALISATION DE L'HISTOGRAMME (1/5)

- ❖ L'égalisation d'histogramme a pour objectif d'obtenir une image rehaussée ayant un histogramme linéarisé ou égalisé: c'est-à-dire uniforme (constant) sur tout l'intervalle  $[0,255]$ . Un histogramme "uniforme" signifie que tous les niveaux de gris sont présents en même nombre dans l'image.



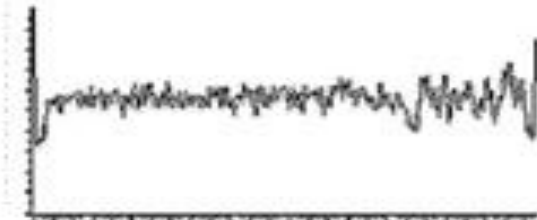


# REHAUSSEMENT DE L'IMAGE

## EGALISATION DE L'HISTOGRAMME (2/5)

### ❖ Pseudo-algorithme d'égalisation d'histogramme

- Soit  $M(i,j)$  l'image source
- Déterminer son histogramme  $h$
- Déterminer son histogramme cumulé  $h_c$
- Normaliser  $h_c$
- Déterminer  $M'(i,j)$  par :  $m'(i,j) = hc[m(i,j)]$



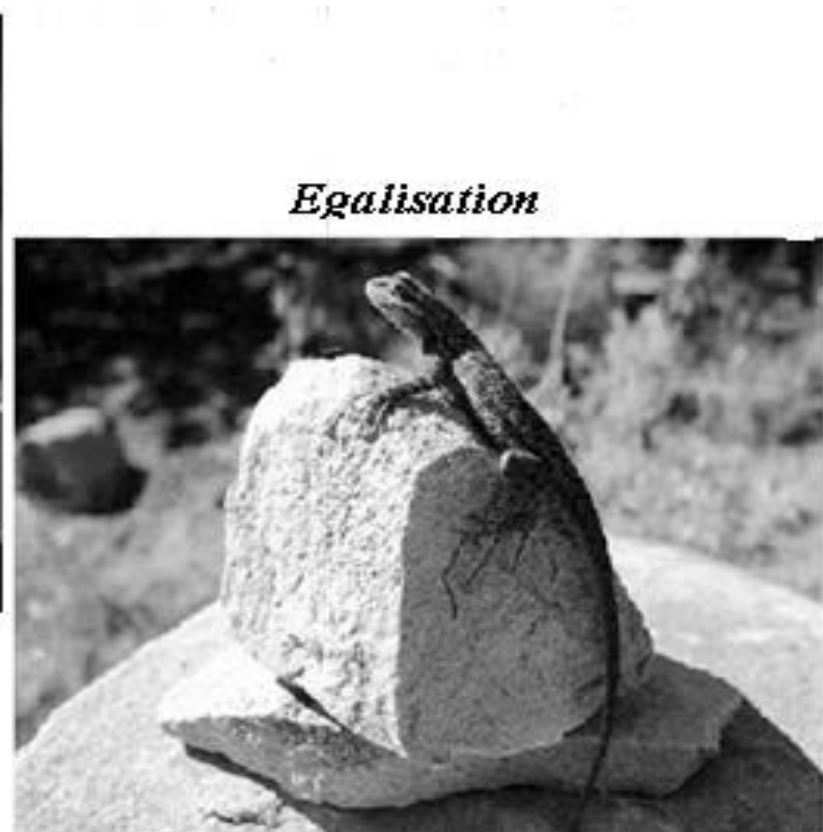
# REHAUSSEMENT DE L'IMAGE

## EGALISATION DE L'HISTOGRAMME (3/5)

Exemple (1/3)



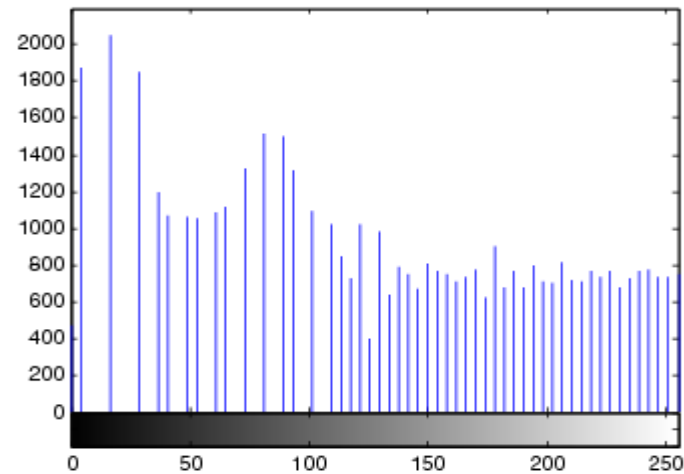
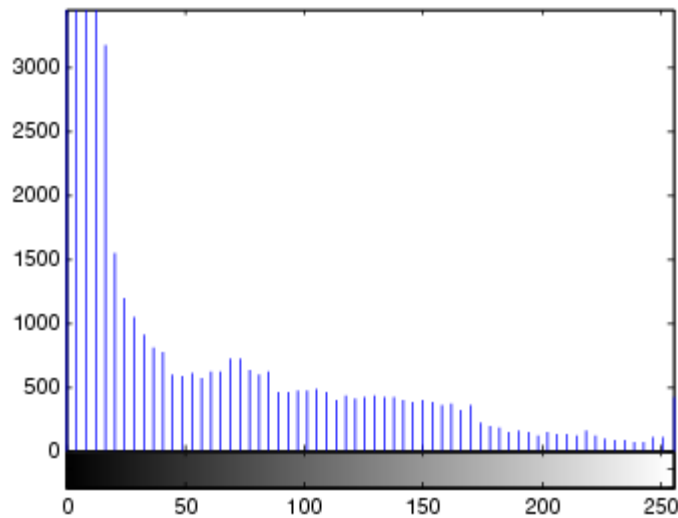
*Original*



*Egalisation*

# REHAUSSEMENT DE L'IMAGE

## EGALISATION DE L'HISTOGRAMME (4/5)



# REHAUSSEMENT DE L'IMAGE

## EGALISATION DE L'HISTOGRAMME (5/5)

Exemple (3/3)



Original

-

Étirement d'histogramme

-

Égalisation

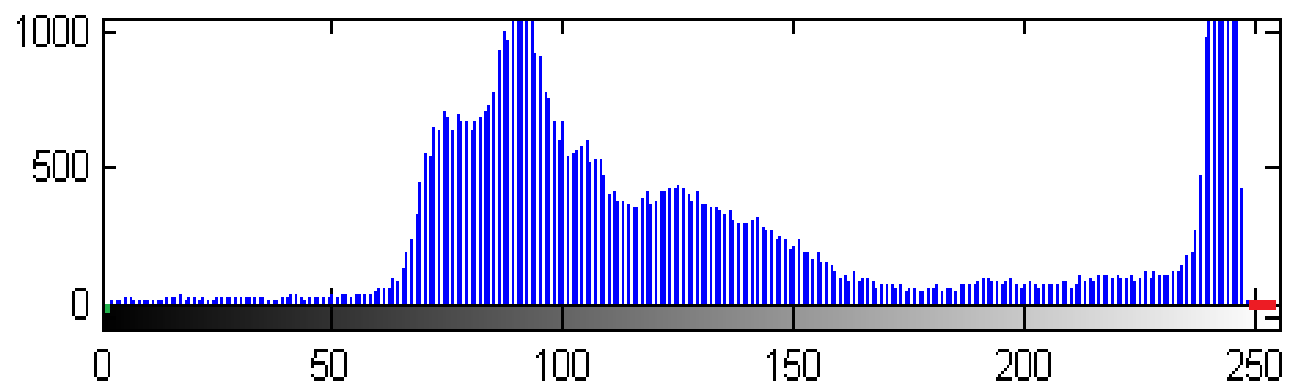
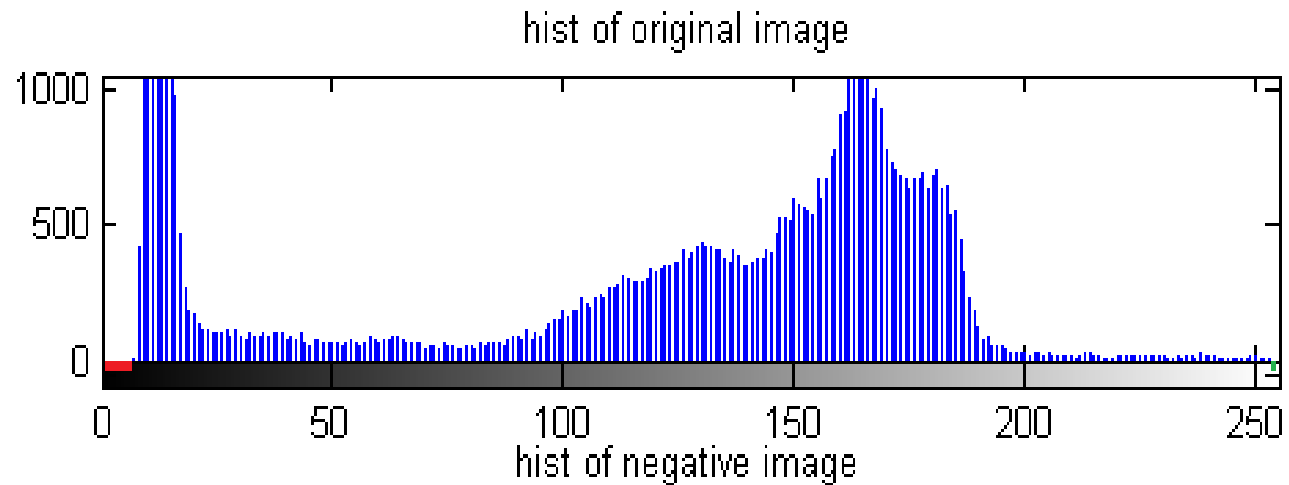
# REHAUSSEMENT DE L'IMAGE

## NÉGATIF DE L'IMAGE (1/2)

original image



negative image



# REHAUSSEMENT DE L'IMAGE

## NÉGATIF DE L'IMAGE (2/2)

*Négatif(Matrice Origin[ nbcol] [nbligne ], seuil)*

*Début*

*Matrice Négatif[ nbcol] [ nbligne]*

*Pour i= 0 à nbcol*

*Pour i= 0 à nbligne*

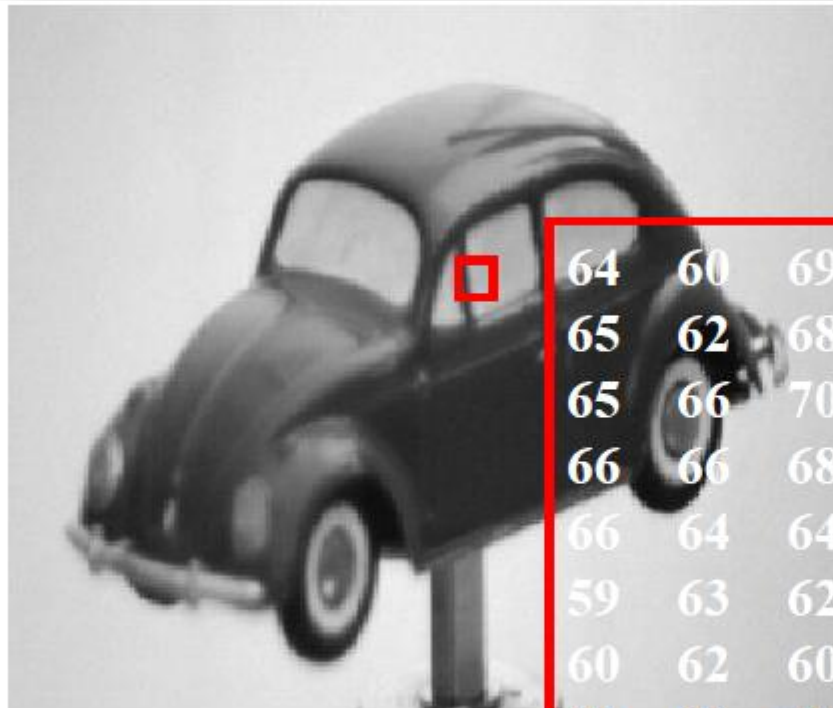
*Négatif[i][j]= 255 - Origin[ i] [j], ;*

*Fin pour*

*Fin pour*

*Fin*

# EXERCICE 1(1/2)



*Niveaux de gris - 8 bits:*

*0 - noir*

*255 - blanc*

64	60	69	100	149	151	176	182	179
65	62	68	97	145	148	175	183	181
65	66	70	95	142	146	176	185	184
66	66	68	90	135	140	172	184	184
66	64	64	84	129	134	168	181	182
59	63	62	88	130	128	166	185	180
60	62	60	85	127	125	163	183	178
62	62	58	81	122	120	160	181	176
63	64	58	78	118	117	159	180	176

## EXERCICE 2 (2/2)

1. Calculer pour l'image précédente:
  - a. la définition.
  - b. La taille.
  - c. Le contraste.
  - d. La luminance.
2. Dessiner l'histogramme de l'image.
3. Donner la matrice de sortie de l'image suite à:
  - a. Une binarisation.

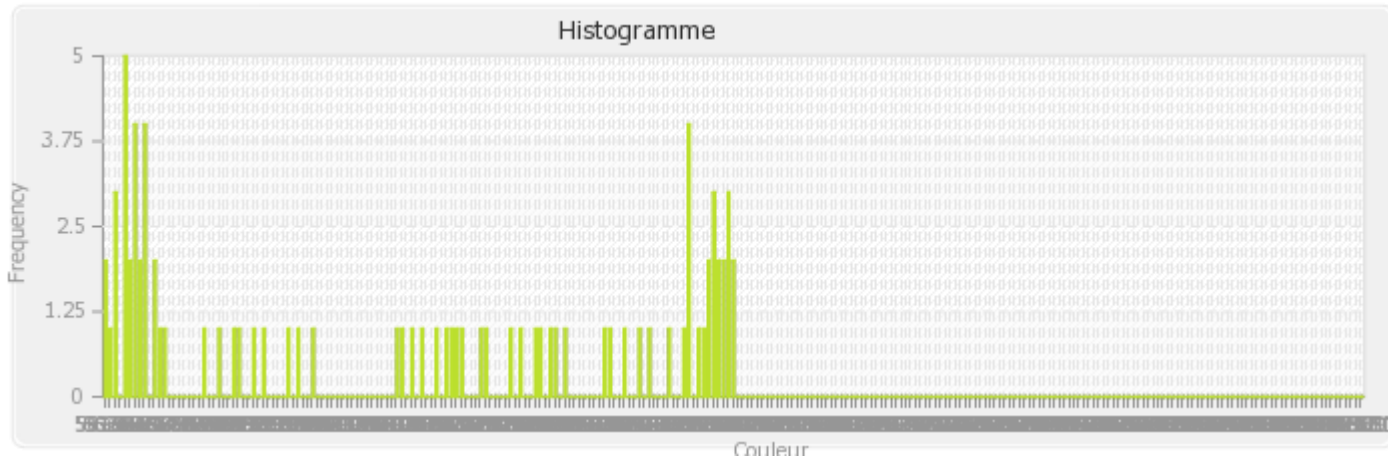


# ELÉMENTS DE CORRECTION (1/3)

- Définition = 9x9
- Taille =  $(9*9)*8 = 648 \text{ bit} = 81 \text{ octets}$
- Contraste =  $(185-58)/(185+58) = 0.522$
- Luminance =  
$$\frac{(64+60+69+100+149+151+176+182+179+65+62+68+97+145+148+175+183+181+65+66+70+95+142+146+176+185+184+66+66+68+90+135+140+172+184+184+66+64+64+84+129+134+168+181+182+59+63+62+88+130+128+166+185+180+60+62+60+85+127+125+163+183+178+62+62+58+81+122+120+160+181+176+63+64+58+78+118+117+159+180+176)/(9*9) = 9699/81 = 119.74$$

# ELÉMENTS DE CORRECTION (2/3)

## ○ Histogramme



## ○ Matrice binarisée

- $\text{Seuil} = (\text{Max} + \text{Min}) / 2 = (185 + 58) / 2 = 121$

0	0	0	0	1	1	1	1	1
0	0	0	0	1	1	1	1	1
0	0	0	0	1	1	1	1	1
0	0	0	0	1	1	1	1	1
0	0	0	0	1	1	1	1	1
0	0	0	0	1	1	1	1	1
0	0	0	0	1	1	1	1	1
0	0	0	0	1	0	1	1	1
0	0	0	0	0	0	1	1	1

## EXERCICE 2

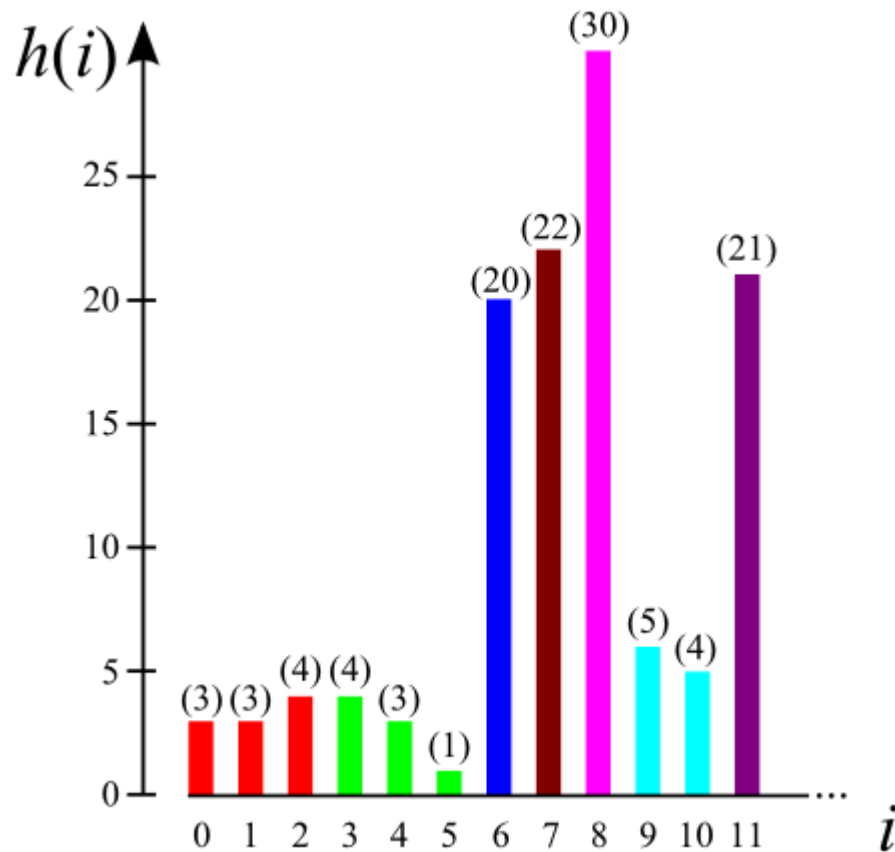
1. Soit la matrice ci-dessous.

0	1	2	3	6	6	6	7	7	7
0	1	2	3	6	6	6	7	7	7
0	1	2	3	6	6	7	7	7	7
4	4	2	3	6	6	7	7	7	7
4	5	6	6	6	6	7	7	7	7
6	6	6	6	6	6	7	7	7	7
9	9	9	9	9	10	10	10	10	11
11	11	11	11	11	11	11	11	11	11
11	11	11	11	11	11	11	11	11	11
8	8	8	8	8	8	8	8	8	8
8	8	8	8	8	8	8	8	8	8
8	8	8	8	8	8	8	8	8	8

Tracer son histogramme après égalisation. Expliquer.  
Pour simplifier les calculs on prend 12 (0 à 11) niveaux de gris seulement au lieu de 256 (0 à 255)

# ELÉMENTS DE CORRECTION (1/6)

- Histogramme



# ELÉMENTS DE CORRECTION (2/6)

- Histogramme

i	h(i)
0	3
1	3
2	4
3	4
4	3
5	1
6	20
7	22
8	30
9	5
10	4
11	21

## Histogramme cumulé

hc(i)	Explication
3	3
6	3+3
10	6+4
14	10+4
17	14+3
18	17+1
38	18+20
60	38+22
90	60+30
95	90+5
99	95+4
120	99+21

$$hc(0) = h(0)$$

$$hc(i) = hc(i-1) + h(i)$$

(pour i de 1 à 255)

## ELÉMENTS DE CORRECTION (3/6)

- Transformation

$$i' = E \left( \frac{256}{N_x \times N_y} h_c(i) - 1 \right), \text{ avec } h_c(i) = \sum_{k=0}^i h(k)$$

- $N_x \times N_y = 120$  pixels
- Prenons un cas simplifié avec seulement 12 couleurs on remplace 256 par 12 dans la formule ci-dessous.

# ELÉMENTS DE CORRECTION (4/6)

hc(i)	i'=t(i)	Explication de i'
3	0	PartieEntière(Abs((12/120)* <b>3</b> -1))
6	0	PartieEntière(Abs((12/120)* <b>6</b> -1))
10	0	PartieEntière(Abs((12/120)* <b>10</b> -1))
14	0	PartieEntière(Abs((12/120)* <b>14</b> -1))
17	0	PartieEntière(Abs((12/120)* <b>17</b> -1))
18	0	PartieEntière(Abs((12/120)* <b>18</b> -1))
38	2	PartieEntière(Abs((12/120)* <b>38</b> -1))
60	5	PartieEntière(Abs((12/120)* <b>60</b> -1))
90	8	PartieEntière(Abs((12/120)* <b>90</b> -1))
95	8	PartieEntière(Abs((12/120)* <b>95</b> -1))
99	8	PartieEntière(Abs((12/120)* <b>99</b> -1))
120	11	PartieEntière(Abs((12/120)* <b>120</b> -1))

# ELÉMENTS DE CORRECTION (5/6)

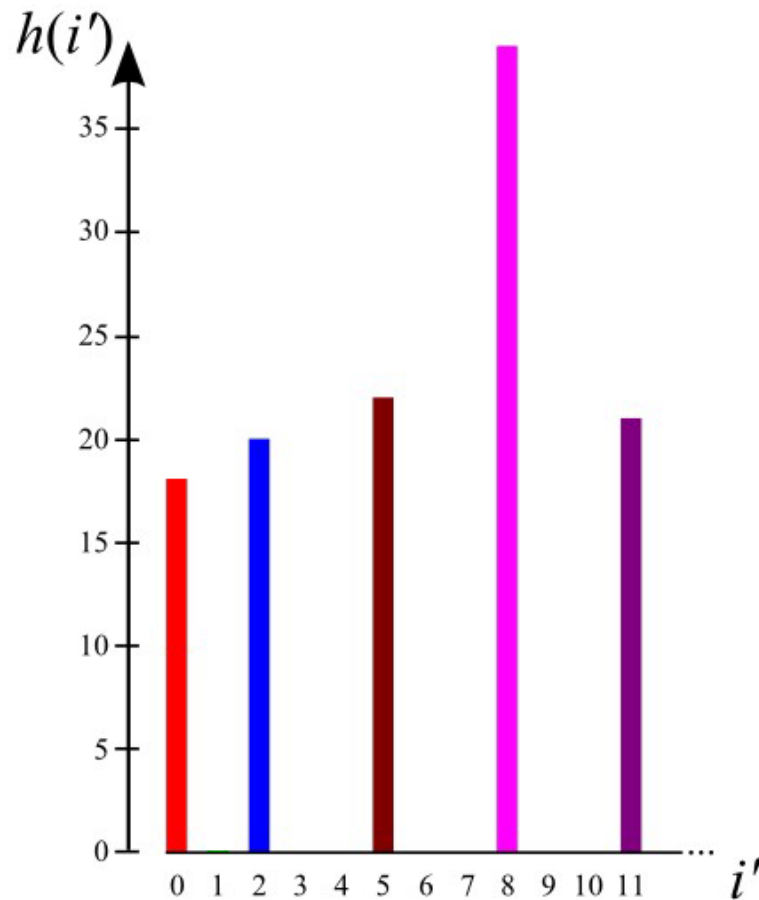
h(i)	i'=t(i)
3	0
3	0
4	0
4	0
3	0
1	0
20	2
22	5
30	8
5	8
4	8
21	11

i'	h(i')	Explication de h(i')
0	18	3+3+4+4+3+1
1	0	0
2	20	20
3	0	0
4	0	0
5	22	22
6	0	0
7	0	0
8	39	30+5+4
9	0	0
10	0	0
11	21	21



# ELÉMENTS DE CORRECTION (6/6)

- Histogramme égalisé

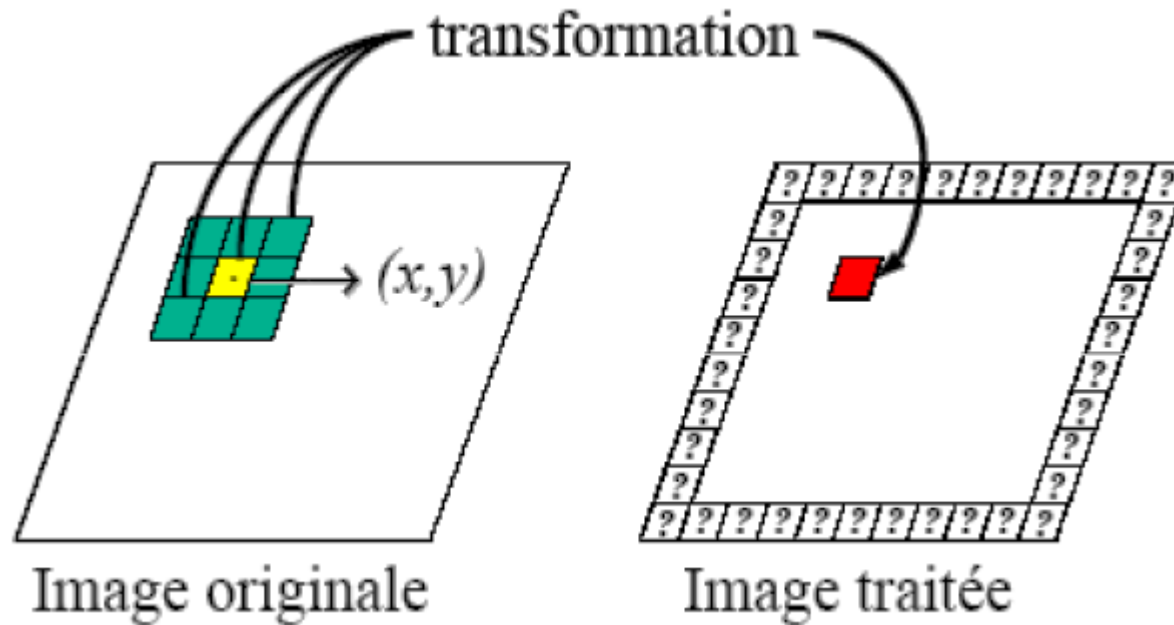


# PLAN

- ❖ Rappel
  - Qu'est ce qu'une image
  - Le pixel
  - La définition
  - La résolution
  - Codage des couleurs
- ❖ Problématique
- ❖ Introduction
  - Histogramme
  - Dynamique
  - Luminance
  - Contraste
- ❖ Rehaussement de l'image
  - Binarisation
  - Recadrage de la dynamique
  - Egalisation de l'histogramme
  - Négatif de l'image
- ❖ Filtrage
  - Convolution
  - Effet de bord
  - Filtrage passe-bas
    - ❑ Filtre moyennneur
    - ❑ Filtre médian
    - ❑ Filtre gaussien

# FILTRAGE

## TRANSFORMATION BASÉE SUR LE VOISINAGE D'UN PIXEL (X,Y)



? : Effet de bord

# FILTRAGE

## FILTRAGE SPATIAL ET CONVOLUTION (1/7)

100	130	110	120	110
110	90	100	90	100
130	100	90	130	110
120	100	130	110	120
90	110	80	120	100

Image initial

$W_1$	$W_2$	$W_3$
$W_4$	$W_5$	$W_6$
$W_7$	$W_8$	$W_9$

Noyau de convolution  
(Masque)

?	?	?	?	?
?	V			?
?				?
?				?
?	?	?	?	?

Image résultat

$$V = 100 * W_1 + 130 * W_2 + 110 * W_3 + 110 * W_4 + 90 * W_5 + 100 * W_6 + 130 * W_7 + 100 * W_8 + 90 * W_9$$

# FILTRAGE

## FILTRAGE SPATIAL ET CONVOLUTION (2/7)

$$g(x, y) = f(x, y) \otimes \text{filtre}(x, y)$$

$$g(x, y) = \sum_i \sum_j f(x+i, y+j) \cdot \text{filtre}(i, j)$$

$$g(x, y) = f(x-1, y-1) * W_1 + f(x-1, y) * W_2 + f(x-1, y+1) * W_3 + \\ f(x, y-1) * W_4 + f(x, y) * W_5 + f(x+1, y) * W_6 + \\ f(x+1, y-1) * W_7 + f(x+1, y) * W_8 + f(x+1, y+1) * W_9$$

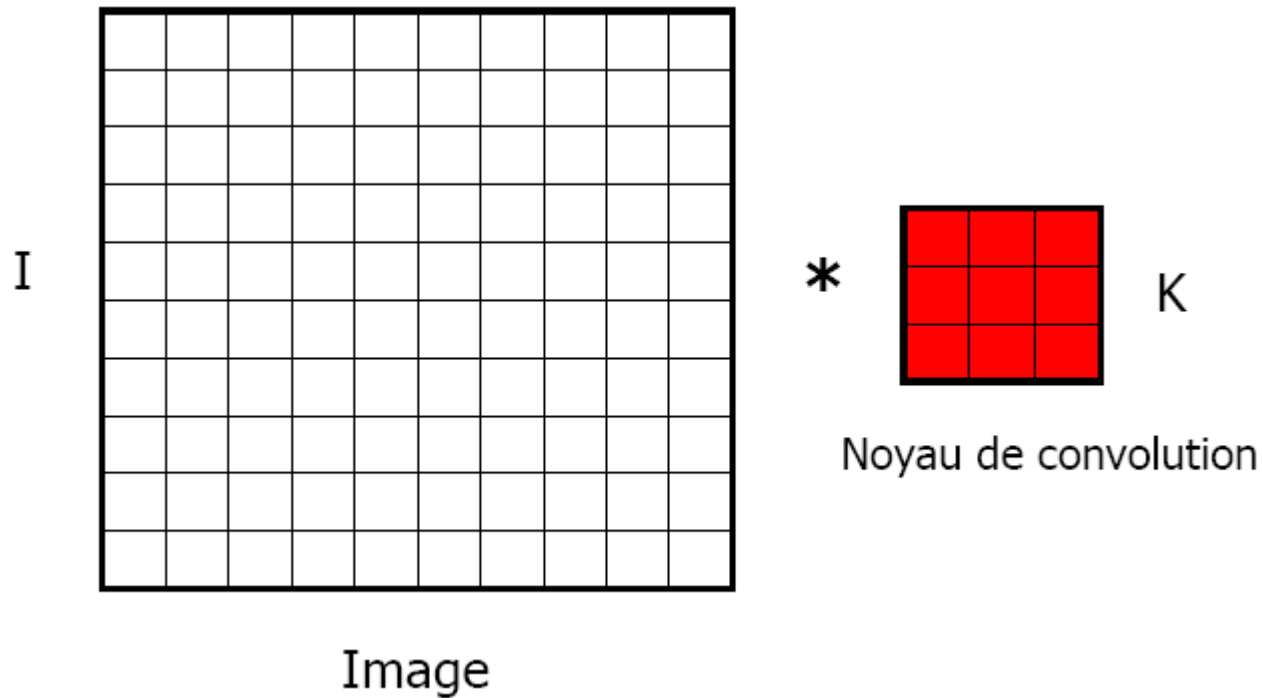
$w_1$	$w_2$	$w_3$
$w_4$	$w_5$	$w_6$
$w_7$	$w_8$	$w_9$

$\text{filtre}(0,0) = w_5$

Généralement, le masque est de dimension impaire et symétrique.

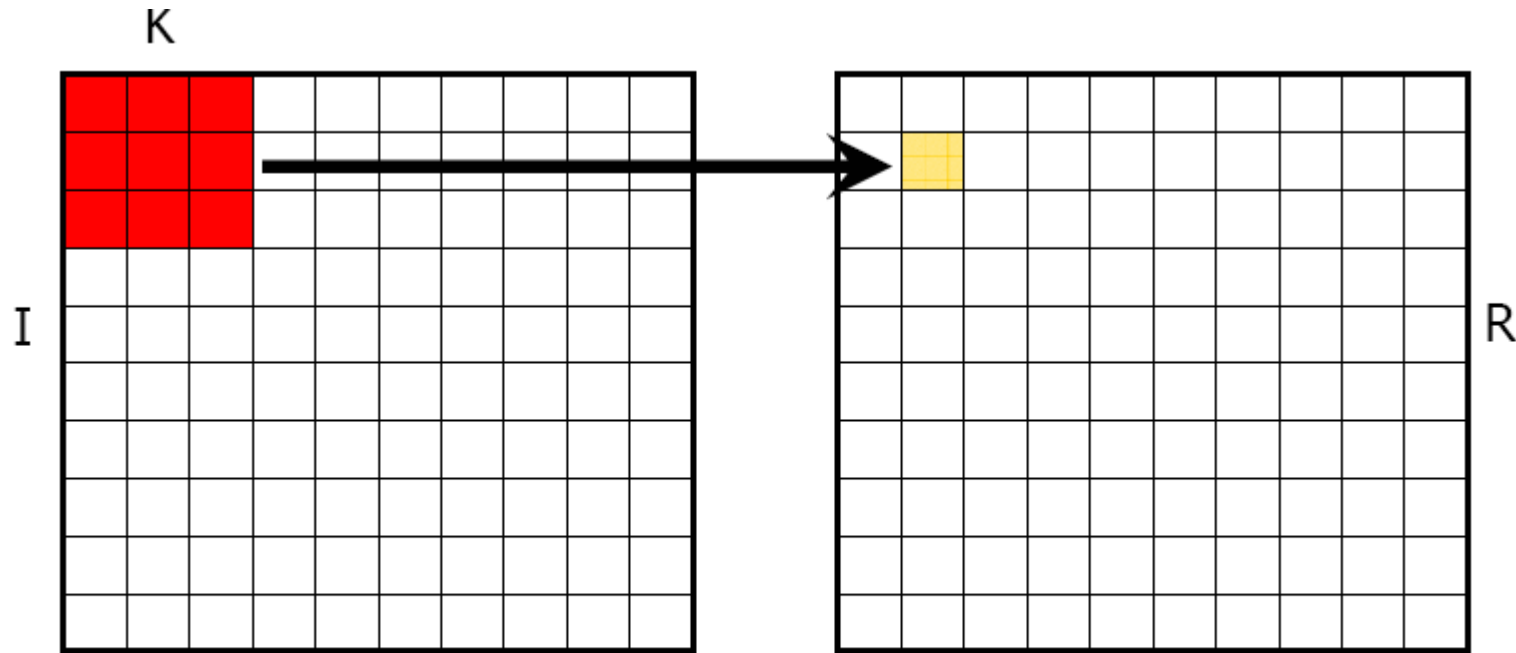
# FILTRAGE

## FILTRAGE SPATIAL ET CONVOLUTION (3/7)



# FILTRAGE

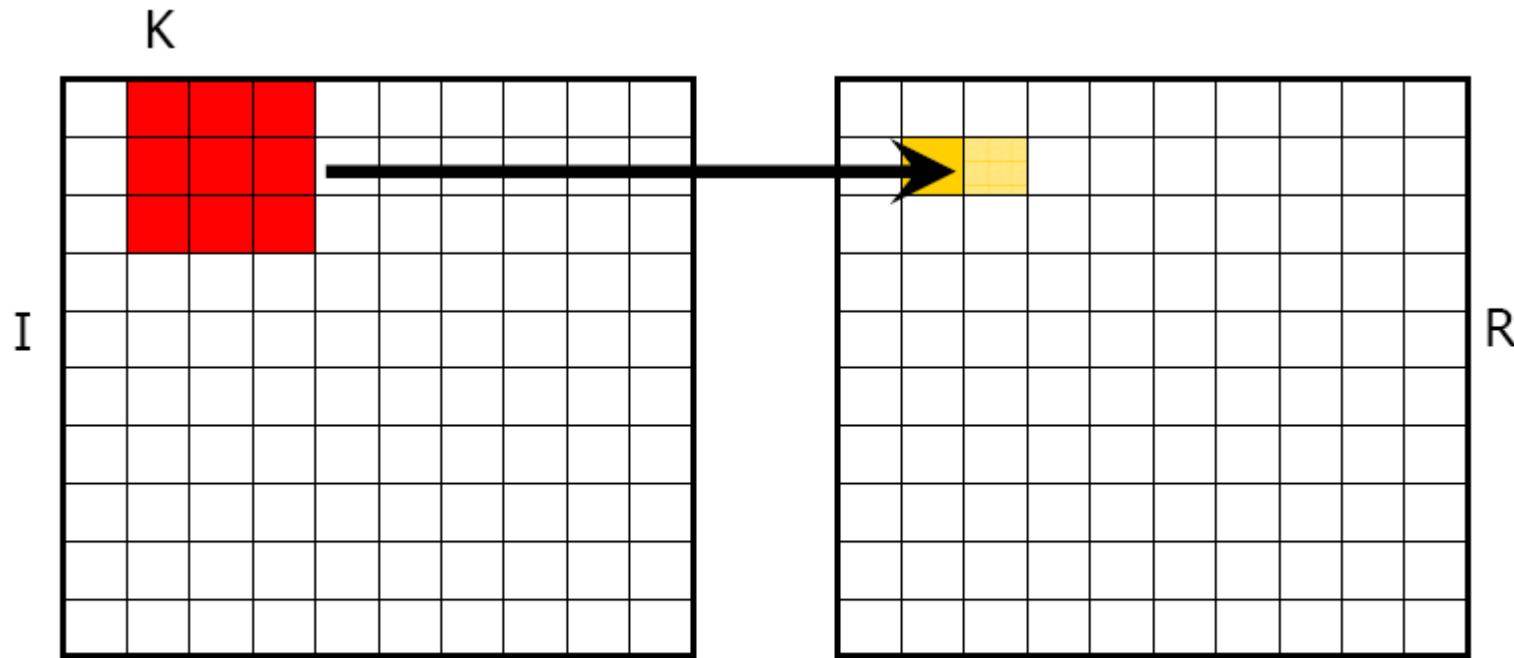
## FILTRAGE SPATIAL ET CONVOLUTION (4/7)



$$\begin{aligned} R(1,1) = & I(0,0) K(0,0) + I(1,0) K(1,0) + I(2,0) K(2,0) \\ & + I(0,1) K(0,1) + I(1,1) K(1,1) + I(2,1) K(2,1) \\ & + I(0,2) K(0,2) + I(1,2) K(1,2) + I(2,2) K(2,2) \end{aligned}$$

# FILTRAGE

## FILTRAGE SPATIAL ET CONVOLUTION (5/7)

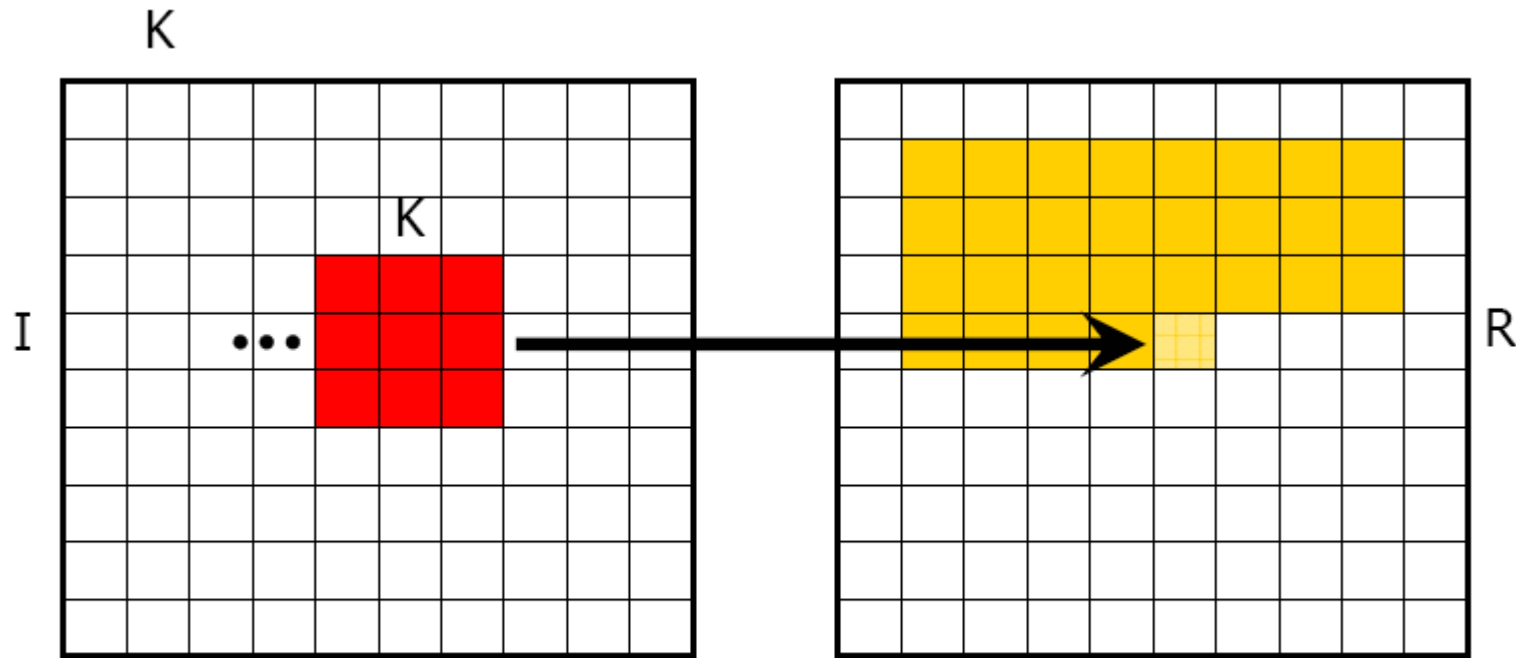


$$\begin{aligned} R(2,1) = & I(1,0) K(0,0) + I(2,0) K(1,0) + I(3,0) K(2,0) \\ & + I(1,1) K(0,1) + I(2,1) K(1,1) + I(3,1) K(2,1) \\ & + I(1,2) K(0,2) + I(2,2) K(1,2) + I(3,2) K(2,2) \end{aligned}$$



# FILTRAGE

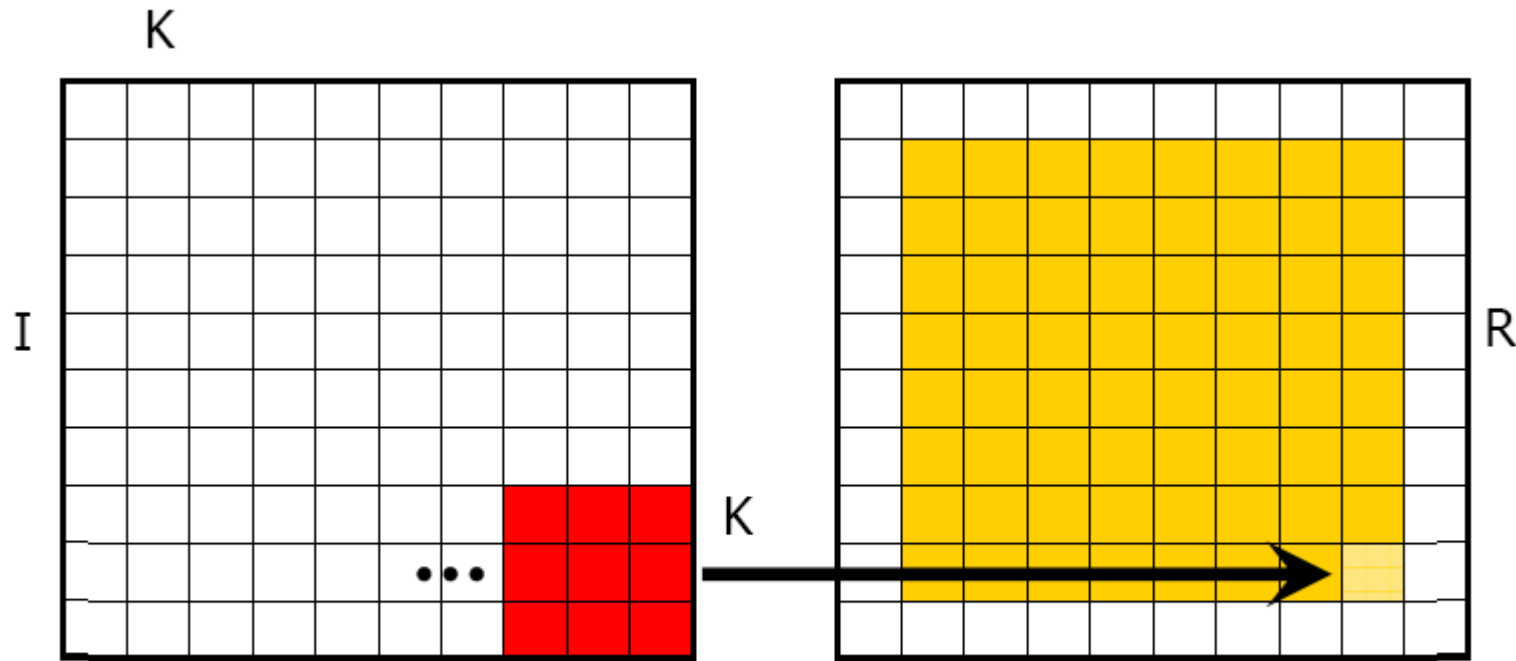
## FILTRAGE SPATIAL ET CONVOLUTION (6/7)



$$\begin{aligned}
 R(x,y) = & I(x-1,y-1) K(0,0) + I(x, y-1) K(1,0) + I(x+1, y-1) K(2,0) \\
 & + I(x-1,y) K(0,1) + I(x,y) K(1,1) + I(x+1,y) K(2,1) \\
 & + I(x-1,y+1) K(0,2) + I(x,y+1) K(1,2) + I(x+1,y+1) K(2,2)
 \end{aligned}$$

# FILTRAGE

## FILTRAGE SPATIAL ET CONVOLUTION (7/7)



$$\begin{aligned}
 R(N-2, M-2) = & I(N-3, M-3) K(0,0) + I(N-2, M-3) K(0,1) + I(N-1, M-3) K(0,2) \\
 & + I(N-3, M-2) K(1,0) + I(N-2, M-2) K(1,1) + I(N-1, M-2) K(1,2) \\
 & + I(N-3, M-1) K(2,0) + I(N-2, M-1) K(2,1) + I(N-1, M-1) K(2,2)
 \end{aligned}$$

# FILTRAGE

## EFFET DE BORD (1/4)

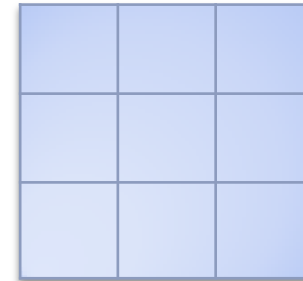
Enlever les bords

100	130	110	120	110
110	90	100	90	100
130	100	90	130	110
120	100	130	110	120
90	110	80	120	100

**Image initiale**

$w_1$	$w_2$	$w_3$
$w_4$	$w_5$	$w_6$
$w_7$	$w_8$	$w_9$

$$\text{filtre}(0,0) = w_5$$



**Image résultat**

# FILTRAGE

## EFFET DE BORD (2/4)

Ne pas filtrer les bords

$w_1$	$w_2$	$w_3$
$w_4$	$w_5$	$w_6$
$w_7$	$w_8$	$w_9$

$$\text{filtre}(0,0) = w_5$$

100	130	110	120	110
110	90	100	90	100
130	100	90	130	110
120	100	130	110	120
90	110	80	120	100

**Image initiale**

100	130	110	120	110
110				100
130				110
120				120
90	110	80	120	100

**Image résultat**

# FILTRAGE

## EFFET DE BORD (3/4)

Mise à zéro de la couronne

0	0	0	0	0	0	0
0	100	130	110	120	110	0
0	110	90	100	90	100	0
0	130	100	90	130	110	0
0	120	100	130	110	120	0
0	90	110	80	120	100	0
0	0	0	0	0	0	0

Image initiale

$w_1$	$w_2$	$w_3$
$w_4$	$w_5$	$w_6$
$w_7$	$w_8$	$w_9$

$$\text{filtre}(0,0) = w_5$$

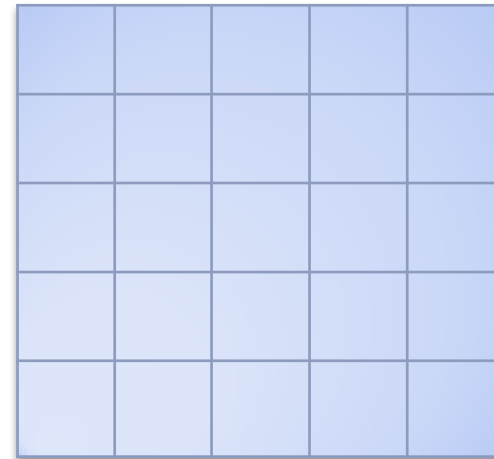


Image résultat

# FILTRAGE

## EFFET DE BORD (4/4)

Réaliser un effet de miroir

<b>100</b>	<b>100</b>	<b>130</b>	<b>110</b>	<b>120</b>	<b>110</b>	<b>110</b>
<b>100</b>	100	130	110	120	110	<b>110</b>
<b>110</b>	110	90	100	90	100	<b>100</b>
<b>130</b>	130	100	90	130	110	<b>110</b>
<b>120</b>	120	100	130	110	120	<b>120</b>
<b>90</b>	90	110	80	120	100	<b>100</b>
<b>90</b>	<b>90</b>	<b>110</b>	<b>80</b>	<b>120</b>	<b>100</b>	<b>100</b>

Image initiale

$w_1$	$w_2$	$w_3$
$w_4$	$w_5$	$w_6$
$w_7$	$w_8$	$w_9$

$$\text{filtre}(0,0) = w_5$$

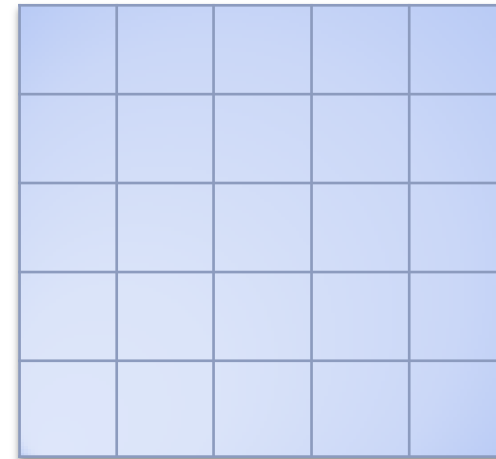


Image résultat

# FILTRAGE

## **Filtrage passe-bas**

Lissage et élimination du bruit

## **Filtrage passe-haut**

Détection des contours

# FILTRAGE PASSE-BAS (1/7)

## Filtre moyennneur (1/3)

- ❖ C'est un filtre passe-bas
  - Lisse l'image (effet de flou)
  - Réduit le bruit
  - Réduit les détails
- ❖ Filtre dont tous les coefficients sont égaux (chaque pixel est remplacé par la moyenne de ses voisins)

$$\frac{1}{9}$$

1	1	1
1	1	1
1	1	1

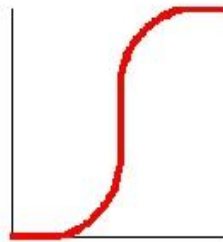
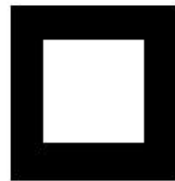
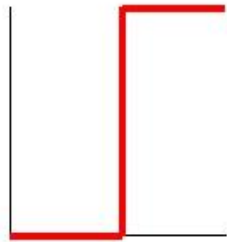
$$\frac{1}{25}$$

1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

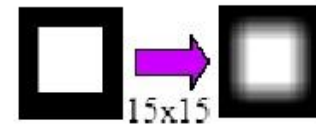
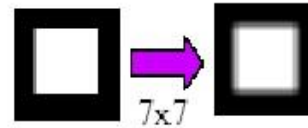
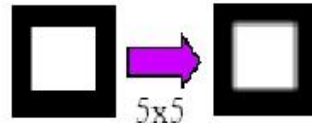
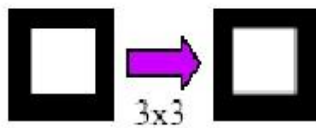


# FILTRAGE PASSE-BAS (2/7)

## Filtre moyenneur (2/3)



**Lissage flou !!**



**Plus le filtre grossit, plus le lissage devient important et plus le flou s'accroît.**

# FILTRAGE PASSE-BAS (3/7)

## Filtre moyenneur (3/3)



Original



Moyenne 5x5



Moyenne 11x11

# FILTRAGE PASSE-BAS (4/7)

## Filtre gaussien (1/2)

- ❖ Le filtre gaussien donne un meilleur lissage et une meilleure réduction du bruit que le filtre moyenne.

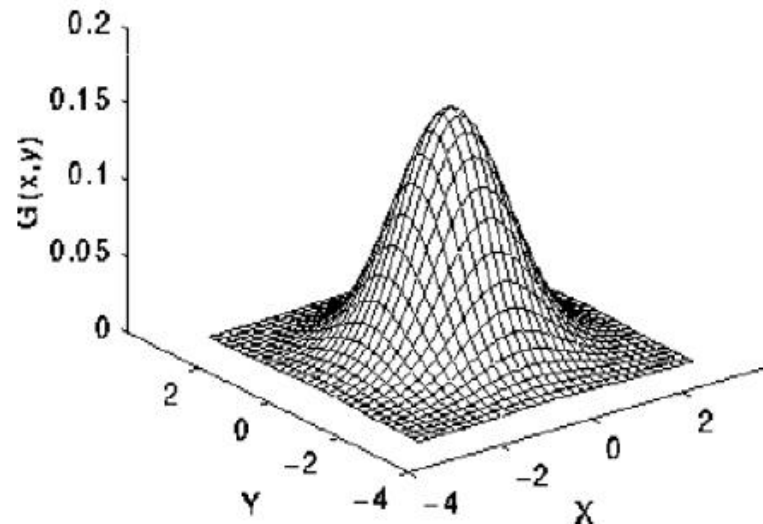
$$\frac{1}{16}$$

1	2	1
2	4	2
1	2	1

$$\frac{1}{246}$$

1	4	6	4	1
4	16	24	16	4
6	24	36	24	6
4	16	24	16	4
1	4	6	4	1

$$G(x, y) = \frac{1}{2\pi\sigma_x\sigma_y} e^{-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2}\right)}$$



# FILTRAGE PASSE-BAS (5/7)

## Filtre gaussien (2/2)



Original



Gauss 5x5

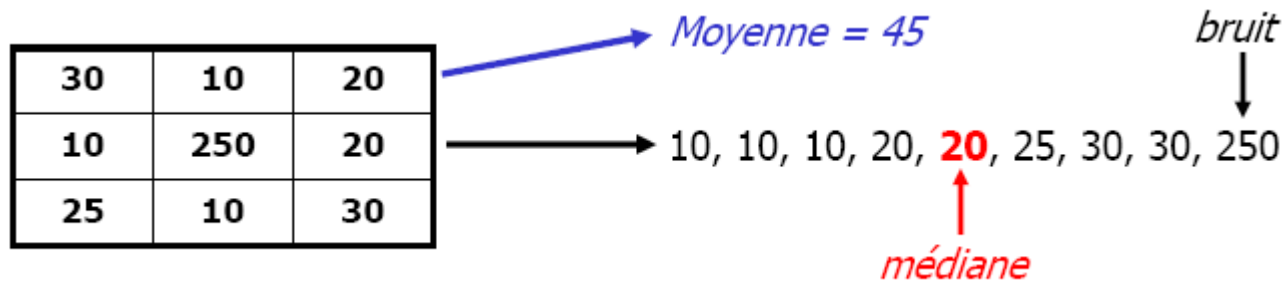


Gauss 11x11

# FILTRAGE PASSE-BAS (6/7)

## Filtre médian

- ❖ Pour nettoyer le bruit dans une image, il existe mieux que le filtre moyenneur ou le filtre gaussien : le **filtre médian**.
  - C'est un filtre non-linéaire, qui ne peut pas s'implémenter comme une convolution
  - On remplace la valeur d'un pixel par la valeur médiane dans son voisinage  $2n + 1 \times 2n + 1$

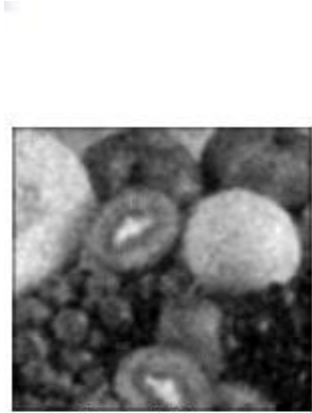


Moyenne 45 à cause du 250

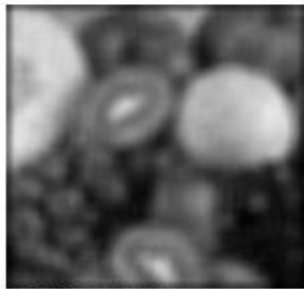
- ❖ Particulièrement utile pour un bruit de type poivre & sel (0 et 255)

# FILTRAGE PASSE-BAS (7/7)

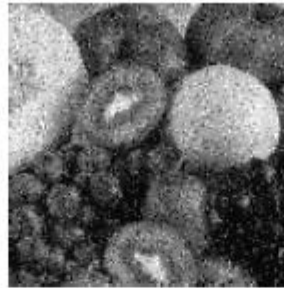
## Moyenneur VS Médian



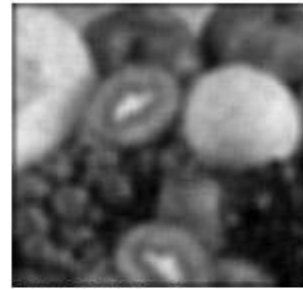
3 X 3 Moyenne



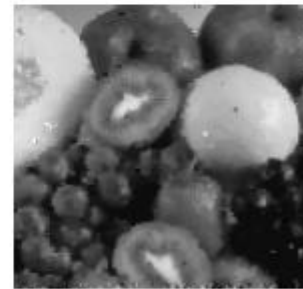
7 X 7 Moyenne



Bruit "poivre et sel"



5 X 5 Moyenne



Filtre médian

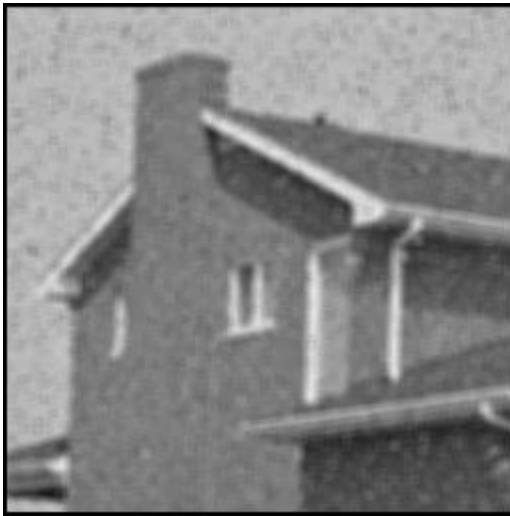
## EXERCICE (1/3)

- On considère l'image initiale suivante :



- Cette image a été traitée avec :
  - un filtre médian 3x3
  - un filtre moyen 3x3
  - un filtre moyen 5x5
- Associez chaque image résultat au traitement qui a permis de l'obtenir. Expliquez.

## EXERCICE (2/3)



(a)



(b)



(c)



## ELÉMENTS DE CORRECTION (1/2)

- (a): Filtre moyennneur 5x5 car il représente plus de flou que l'image (b) et moins de bruits
- (b): Filtre moyennneur 3x3 car il représente moins de flou que l'image (a) et plus de bruits
- (c): Filtre médian 3x3 c'est le filtre qui est le plus approprié à ce type de bruit « poivre et Sel »

## EXERCICE (3/3)

- Soit l'image suivante:

0	140	51	191	140	51
0	51	191	140	140	51
51	140	20	20	140	0
51	140	20	20	20	140
0	140	191	0	20	51
0	10	51	10	140	51

- Si on convolue cette image avec le filtre:

$$\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

- Quel est le type de ce filtre?
- Quelle sera la nouvelle valeur du pixel de coordonnées (2; 3) ? Détaillez votre calcul. Quel est l'effet général de ce filtre sur une image ?

## ELÉMENTS DE CORRECTION (2/2)

0	140	51	191	140	51
0	51	191	140	140	51
51	140	20	20	140	0
51	140	20	20	20	140
0	140	191	0	20	51
0	10	51	10	140	51

- Filtre gaussien.
- $$I(2,3) = 140(1/16) * 1 + 20(1/16) * 2 + 20(1/16) * 1 + 140(1/16) * 2 + 20(1/16) * 4 + 20(1/16) * 2 + 140(1/16) * 1 + 191(1/16) * 2 + 0(1/16) * 1$$
$$= 70$$
- Voir cours.

# RÉFÉRENCES

- ❖ Cours Traitement numérique de l'image - Raphaël Isdant 2009
- ❖ Cours Traitement d'images - Alain Boucher
- ❖ [www.pixel-dargent-74.fr/documents/base.pdf](http://www.pixel-dargent-74.fr/documents/base.pdf)
- ❖ Cours TIM - Ezzeddine Zagrouba 2010/2012
- ❖ Cours Analyse d'Inf. Multimédia: Nadhem Nemri 2012/2013
- ❖ LPro GPI Vision Industrielle – UE3 : Analyse des Images – O. Losson & F. Cabestaing