

## TP4

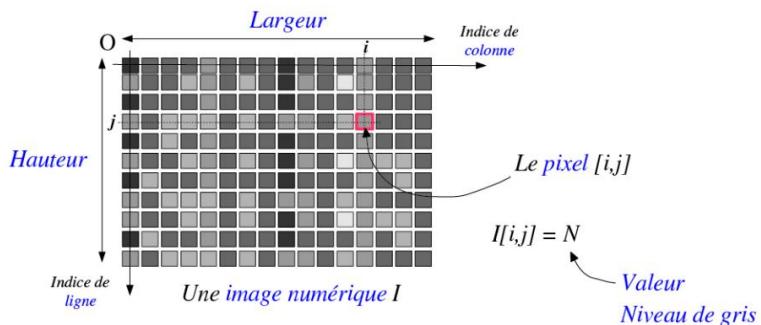
### Objectifs

Manipulation des tableaux de deux dimensions.

### Définition :

Une image numérique est associée à un pavage. Le pavage est une discréétisation (ou un maillage) de l'espace. En général, le pavage est rectangulaire mais il existe d'autres pavages qui peuvent être mieux adaptés à la nature de l'image (hexagonal, triangulaire, ...).

On appelle pixel un élément du pavage associé à l'image numérique.



En Java pour initialiser un tableau de deux dimensions, on utilise :

```
int [][] twoDimArray = {{5,7,3,17}, {7,0,1,12}, {8,1,2,3}};
```

### Exercice 1.

Le Filtre moyenneur consiste à remplacer un pixel par la moyenne de ses voisins et de lui-même. Il existe deux types de voisinages : connexion 4 et connexion 8.

$\frac{1}{5}$ <table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> </table> connexion 4	0	1	0	1	1	1	0	1	0	$\frac{1}{9}$ <table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </table> connexion 8	1	1	1	1	1	1	1	1	1
0	1	0																	
1	1	1																	
0	1	0																	
1	1	1																	
1	1	1																	
1	1	1																	

Ainsi pour la connexion 4 nous avons :

$$g * h(a, b) = 1/5[g(a-1, b) + g(a, b-1) + g(a, b) + g(a, b+1) + g(a+1, b)]$$

et pour la connexion 8 nous avons :

$$g * h(a, b) = 1/9[g(a-1, b-1) + g(a-1, b) + g(a-1, b+1) + g(a, b-1) + g(a, b) + g(a, b+1) + g(a+1, b-1) + g(a+1, b) + g(a+1, b+1)]$$

Écrivez un programme Java qui applique ces deux filtres à l'image  $I$ .

### Exercice 2.

**Filtre conservatif** : Le principe de ce filtre consiste à conserver la valeur du pixel si cette valeur est dans l'intervalle déterminé par les valeurs des 8 pixels voisins. Sinon la valeur du pixel sera remplacée par la valeur la plus proche parmi celles des pixels voisins

15	18	20
21	19	17
22	22	16

*filtre conservatif* →  
*Intervalle=[15,22]*

15	18	20
21	19	17
22	22	16

15	18	20
21	10	17
22	22	16

*filtre conservatif* →  
*Intervalle=[15,22]*

15	18	20
21	15	17
22	22	16

15	18	20
21	25	17
22	22	16

*filtre conservatif* →  
*Intervalle=[15,22]*

15	18	20
21	22	17
22	22	16

Donnez en java le programme *Filtre\_conservatif* et qui applique ce filtre à l'image *I*

### Exercice 3.

**Filtre médian** : Le principe de cette méthode consiste à remplacer un pixel par la médiane de ses voisins. Cela consiste à ranger par ordre croissant les valeurs des pixels voisins et prendre pour nouvelle valeur du pixel celle se trouvant au milieu.

15	18	20
21	10	17
22	22	16

*filtre médian* →  
*10 15 16 17 18 20 21 22 22*

15	18	20
21	18	17
22	22	16

Donnez en java le programme *Filtre\_median* et qui applique ce filtre à l'image *I*