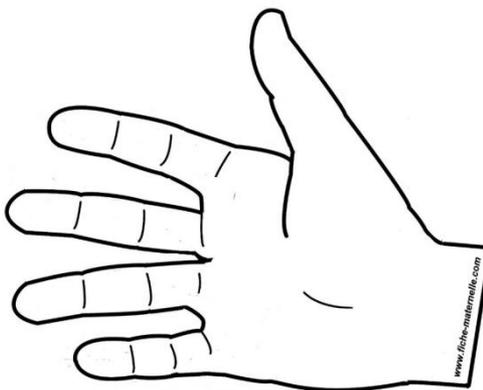
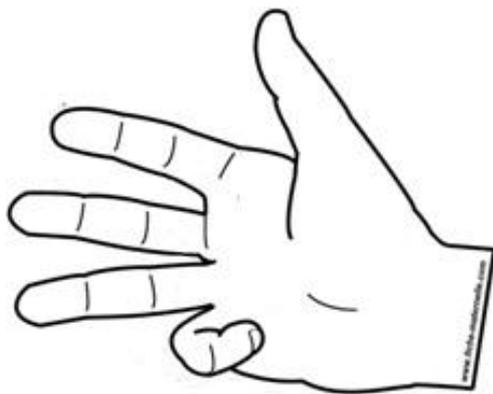
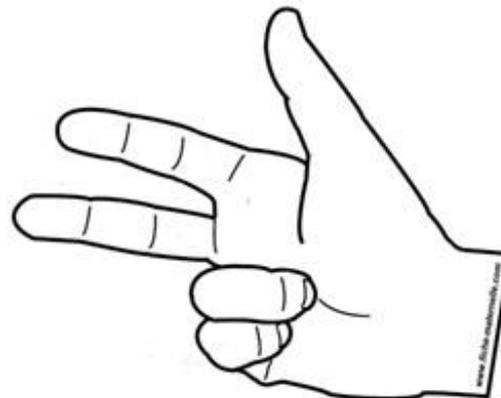
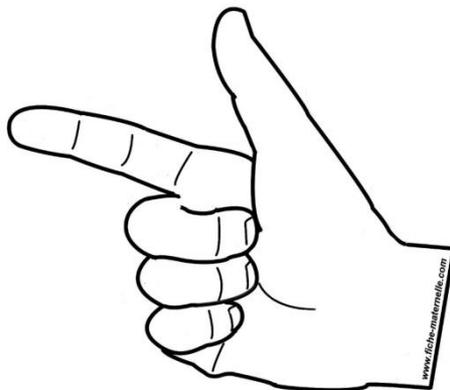
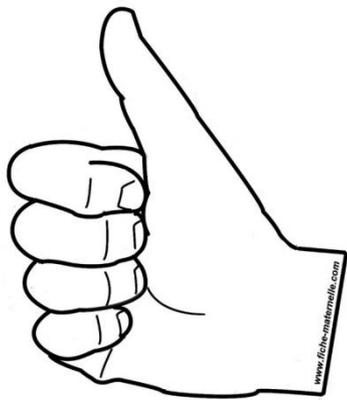
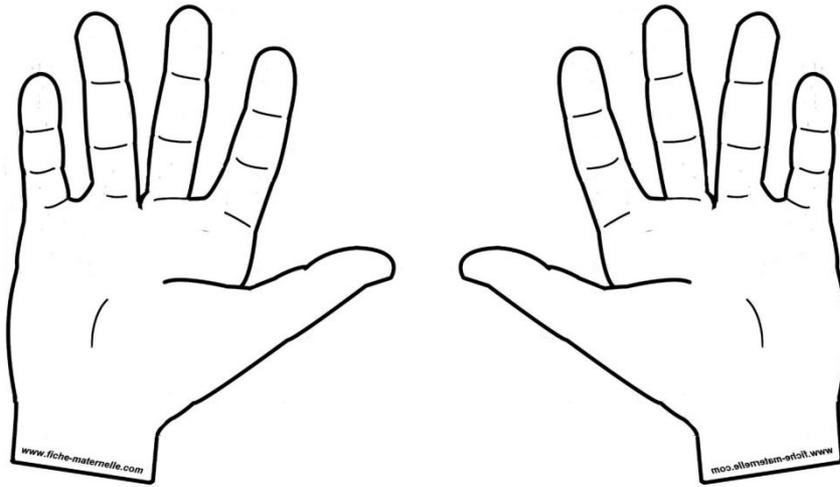


Introduction Rappel sur la base binaire

Q: Comment un enfant fait-il généralement pour compter?



Etc...



10 doigts
!!

➔ **Monde occidental: on compte en base 10**

➔ **Base: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9**

10 caractères

➔ **Tous les nombres en base 10 sont des mots composés de ces 10 caractères (0-9) !!!**

What about the machine?



➔ Ne reconnaît que 2 états



Allumé
État 1



Eteint
État 0



Base binaire !!

➔ Tous les nombres en base 2 sont des mots composés de ces 2 caractères (0 ou 1) !!!

➔ **1 caractère binaire = 1 bit**

➔ **Mot de 8 bits = 1 octet**

Q: Jusqu'à combien peut-on compter en base 10 avec 1 octet ?

Base binaire

0 0 0 0 0 0 0 0



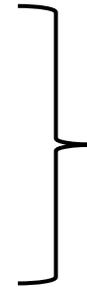
Base 10

000

1 1 1 1 1 1 1 1



255



**256
valeurs**

➔ **L'octet est l'unité de stockage en mémoire**

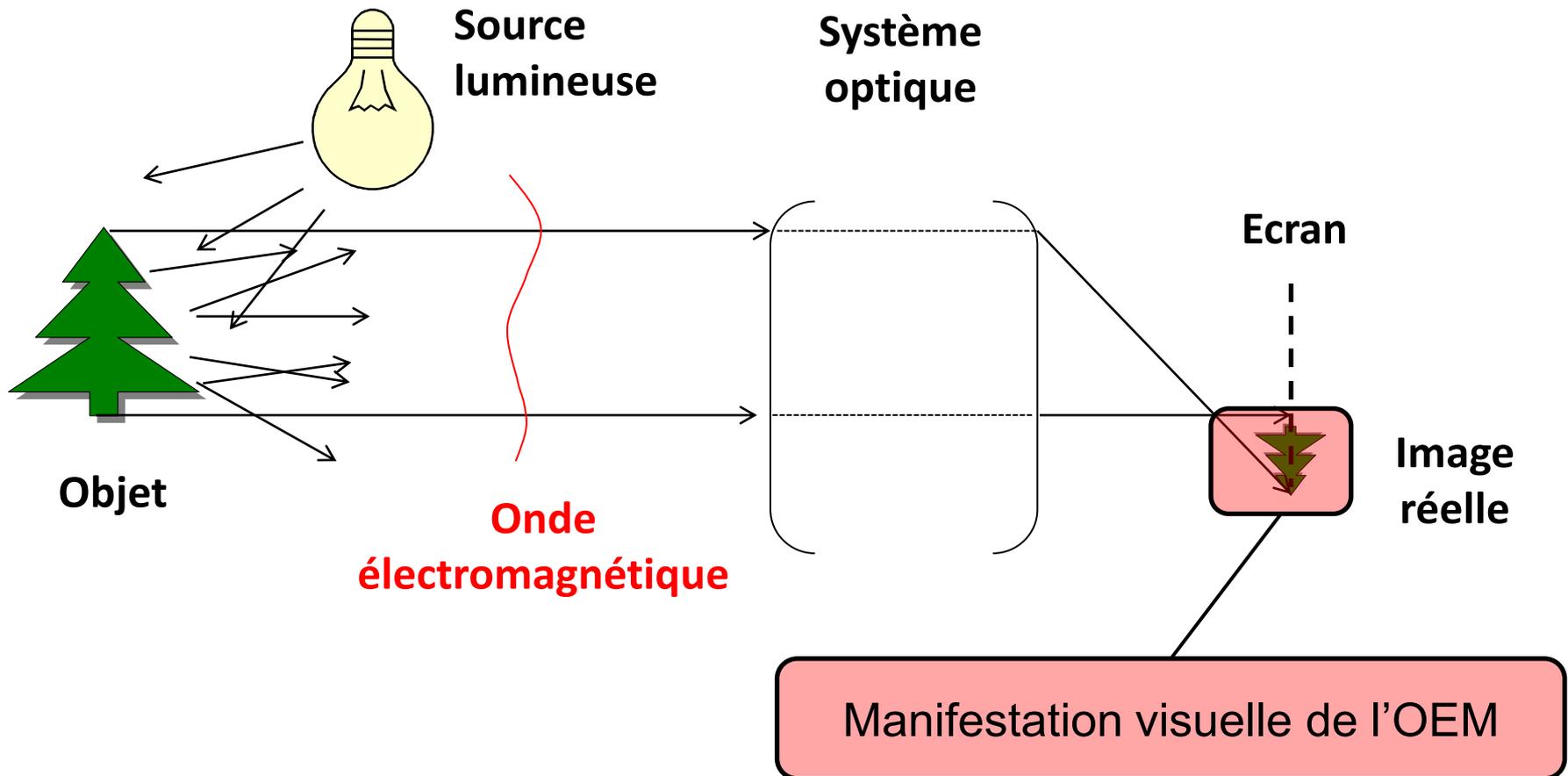
I. Images en niveaux de gris

1.1. De l'image physique à l'image numérique

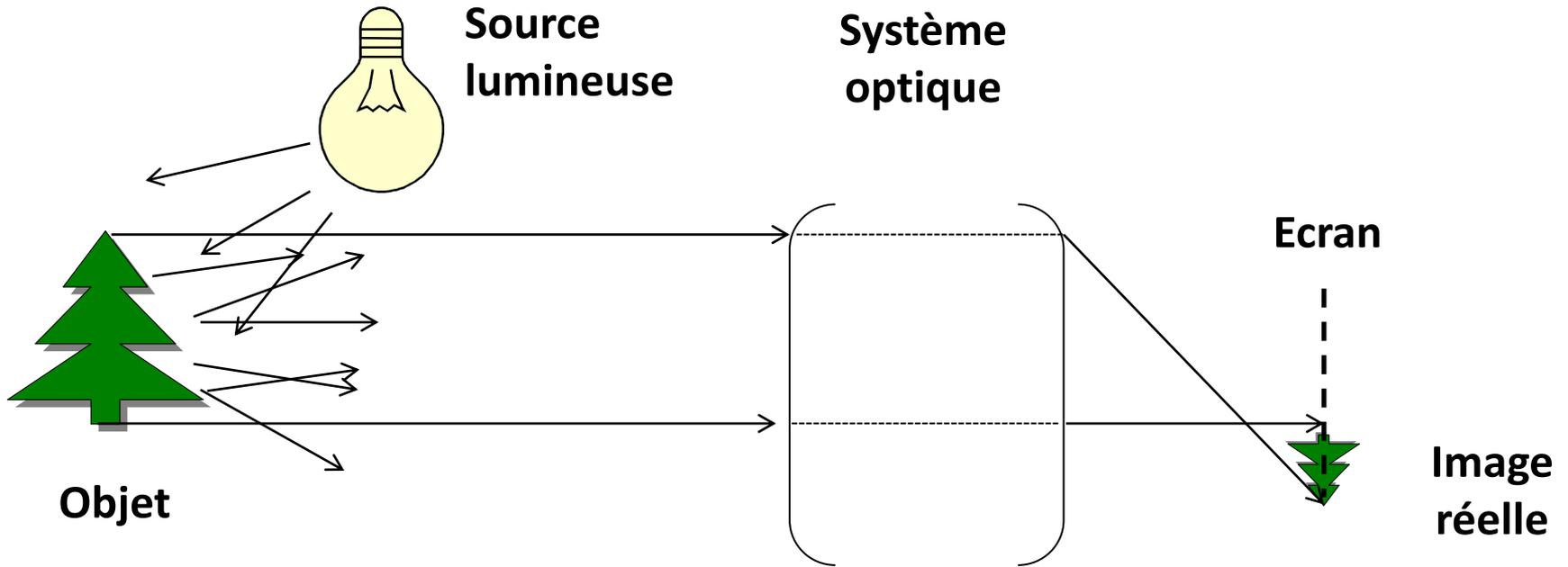
Q: comment former une image sur un écran ?



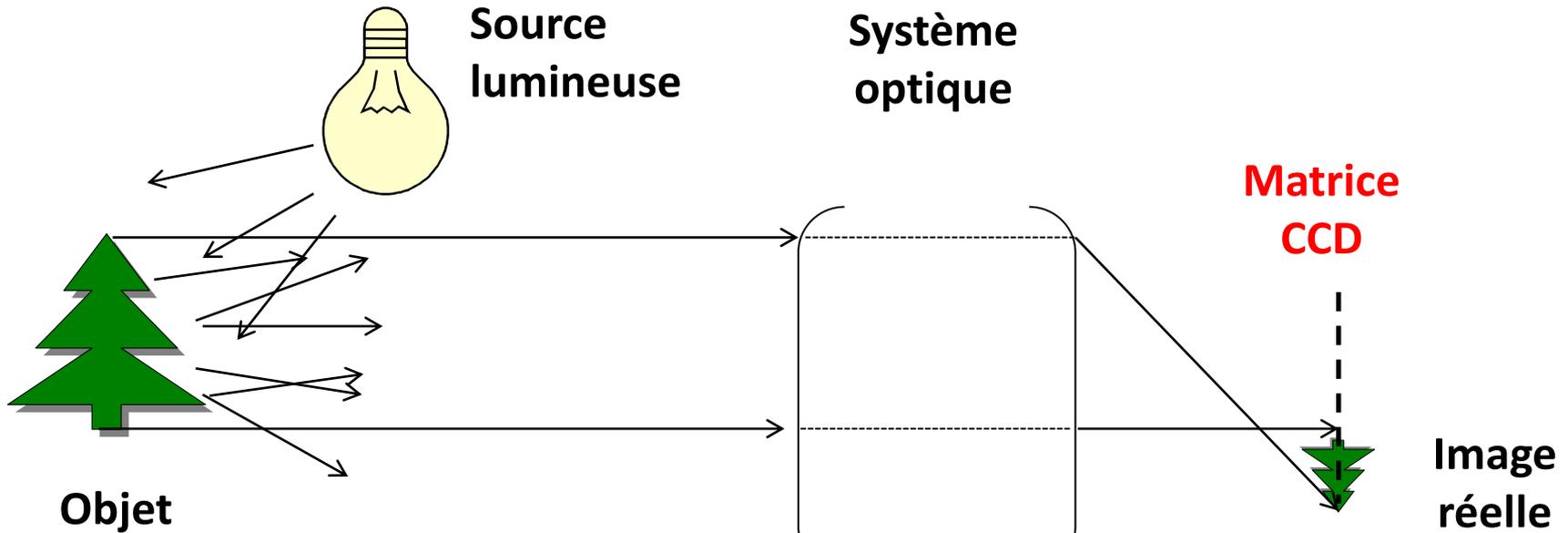
Avec un système optique (objectif, lentille, etc...)



Acquisition d'une image numérique



Acquisition d'une image numérique



Quantification de la tension enregistrée pour chaque pixel sur 8bits, 16bits, etc.... en fonction de la qualité de l'image souhaitée

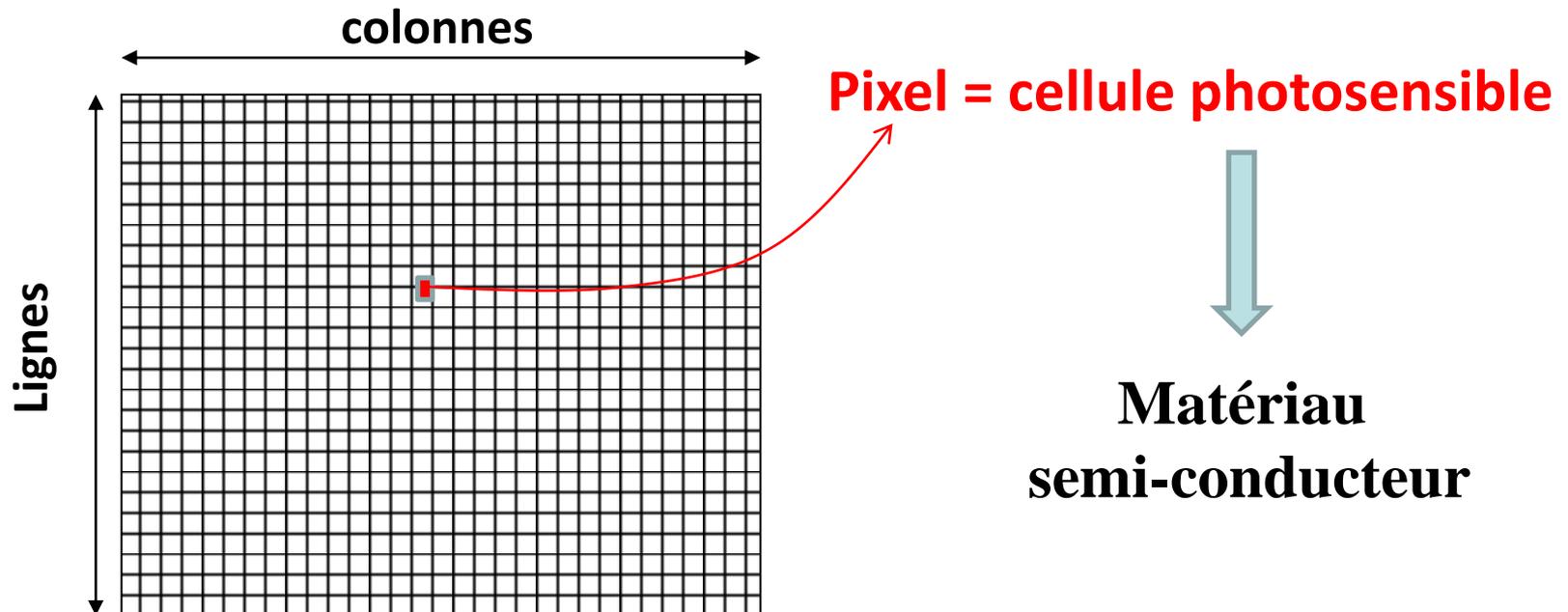
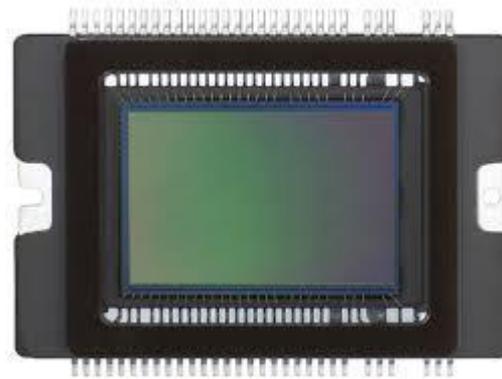
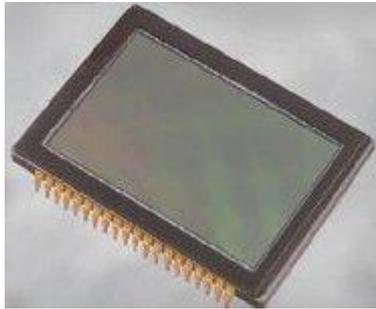
Convertisseur courant-tension

Etape de numérisation de l'image

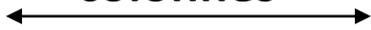
Quantification: Passage d'une tension analogique mesurée à une valeur numérique codée sur un nombre donné de « bit » correspondant à un nombre discret et fini de niveaux de gris entre le noir (intensité lumineuse nulle reçue par le pixel) et le blanc (intensité maximale qu'il est possible d'enregistrer sur un pixel).

1.2. Fonctionnement de la matrice CCD

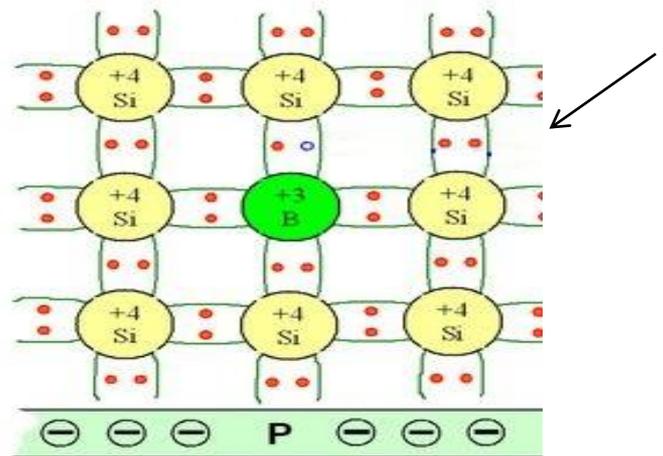
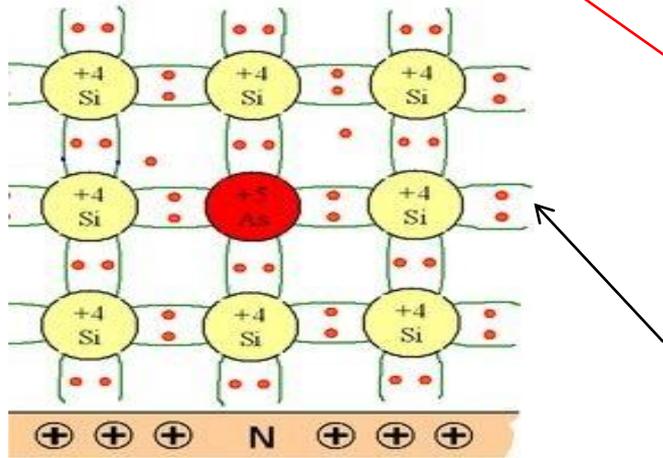
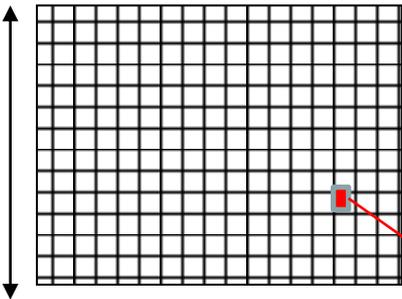
➔ **Matrice CCD**: composée de plusieurs pixels (« Picture element ») disposés sous forme de tableau.



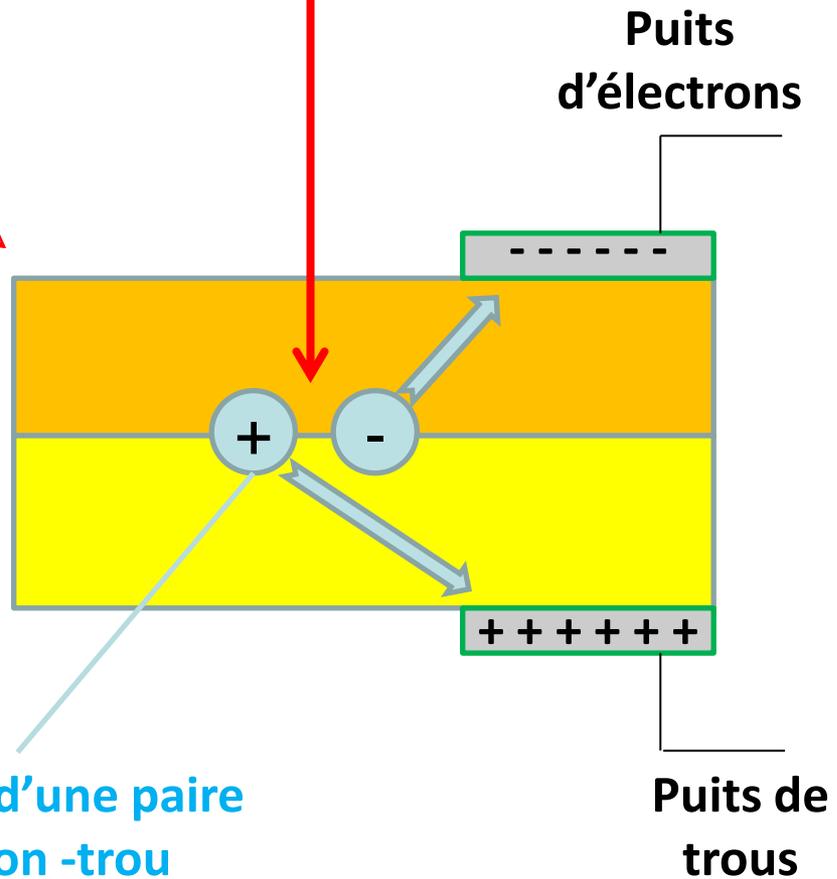
colonnes



Lignes



Photon incident



Silicium dopé N

Silicium dopé P

Création d'une paire électron-trou

Puits d'électrons

Puits de trous

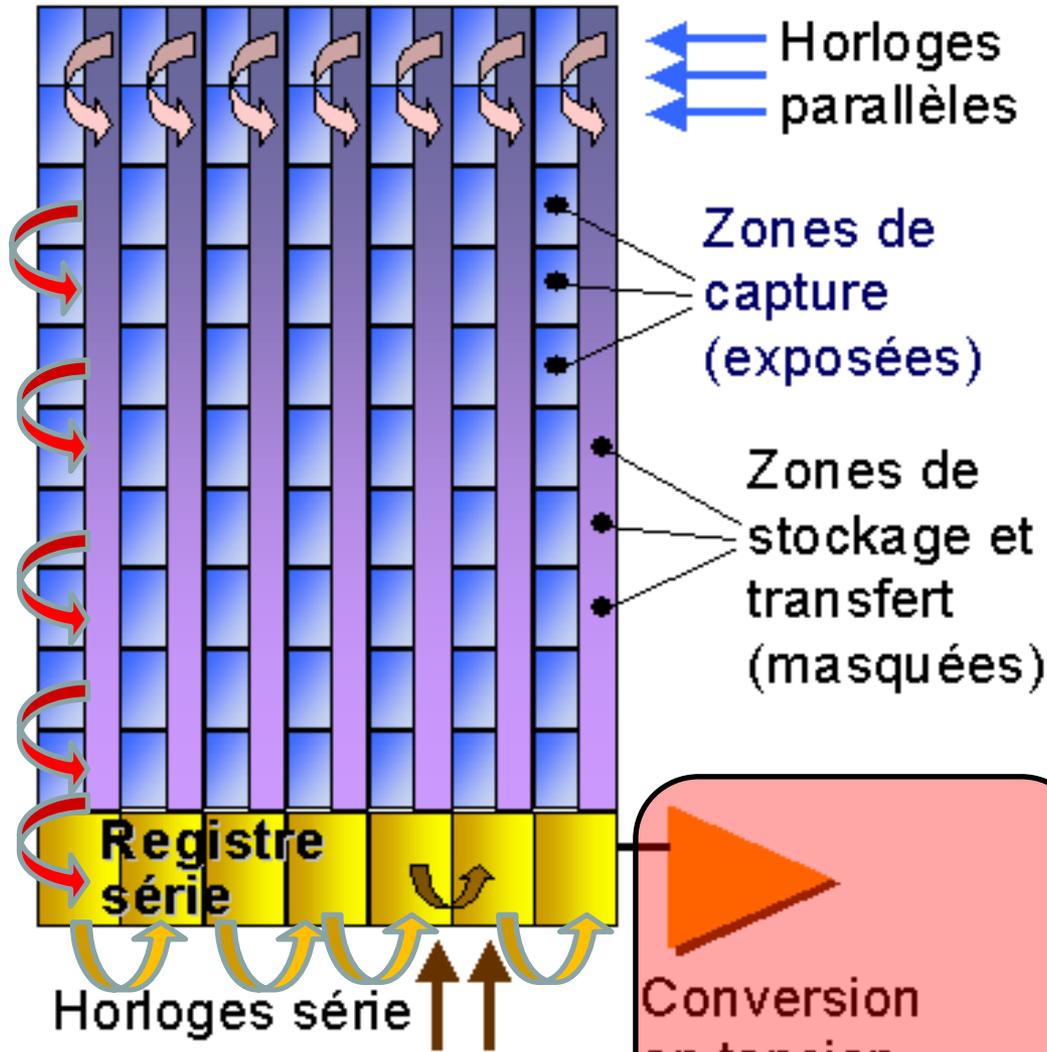


Nombre d'électrons/trous créés est proportionnel à la quantité de lumière reçue pendant toute la durée de l'exposition lumineuse.

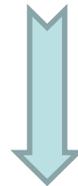
Problème technique :

**Comment compter les
charge créées sur chaque
pixel ?**

➔ **Transfert des charges de chaque pixel de proche en proche**



Mouvement de charges = courant électrique

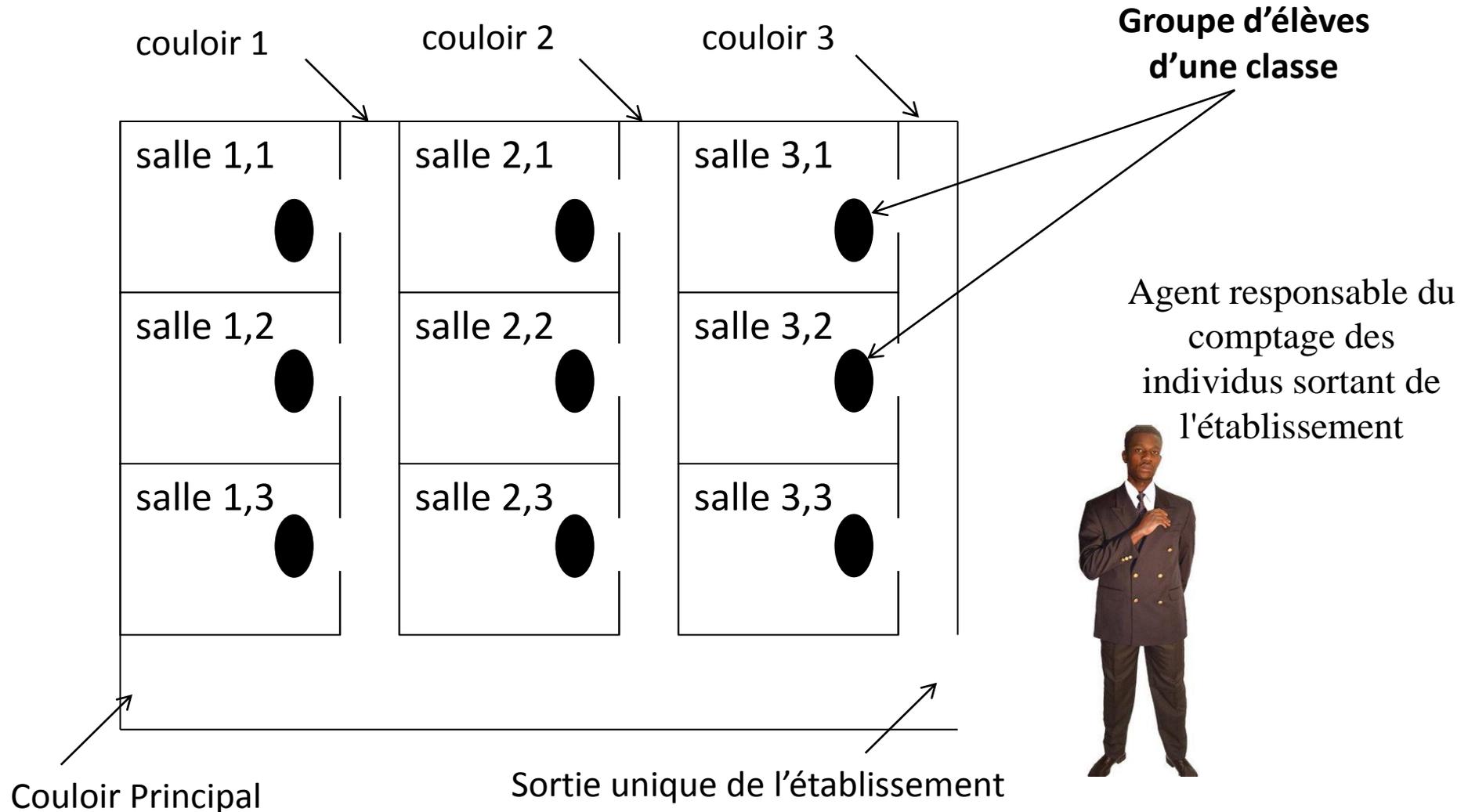


Nécessité de convertir le courant en tension pour la mesure

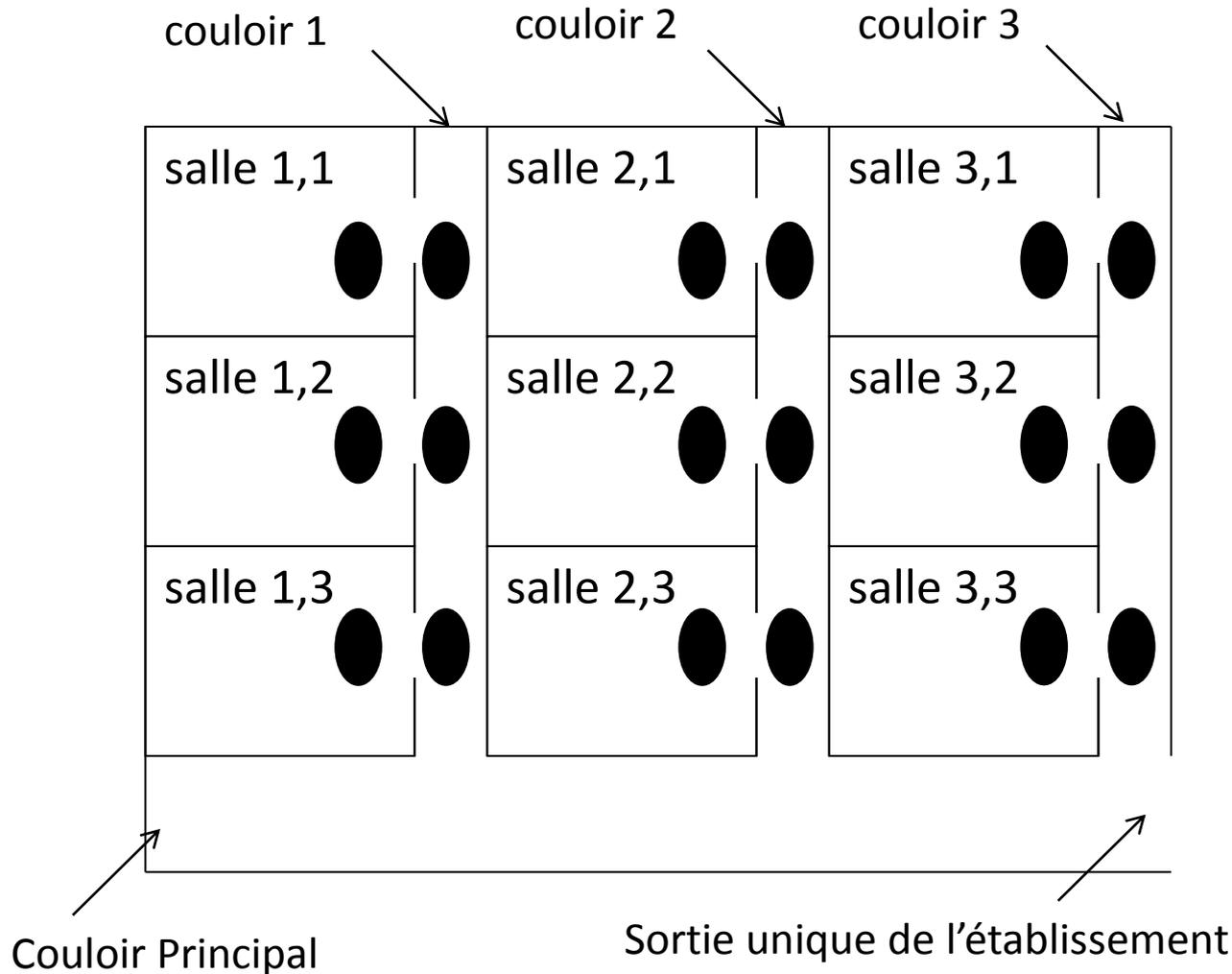
Conversion en tension

Processus de comptage

Par analogie: évacuation d'un établissement scolaire



Etape 1: Les élèves de la classe sortent dans le couloir et restent groupés dans le couloir

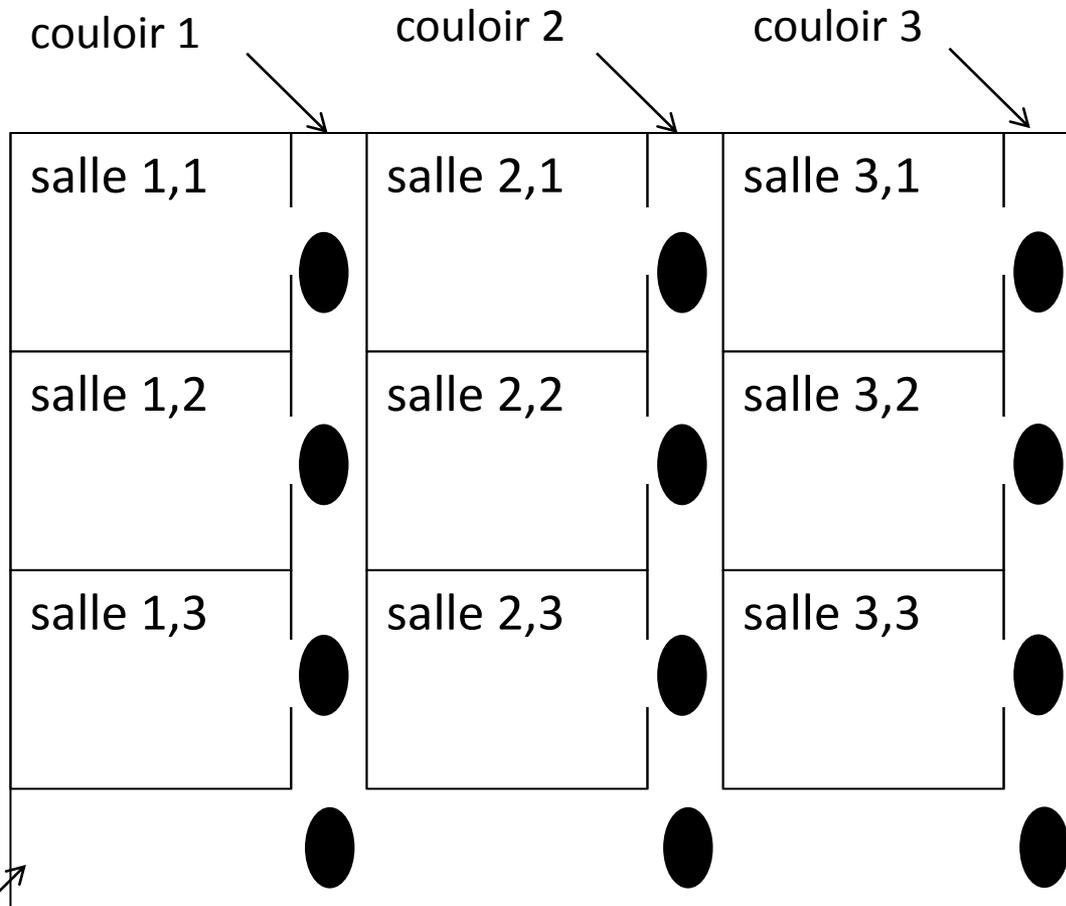


Agent responsable du
comptage des
individus sortant de
l'établissement



Etape 2: On procède à l'évacuation des élèves dans un ordre bien défini, couloir par couloir.

➔ Dans chaque couloir, les élèves des salles 3 de chaque couloir passent dans le couloir principal

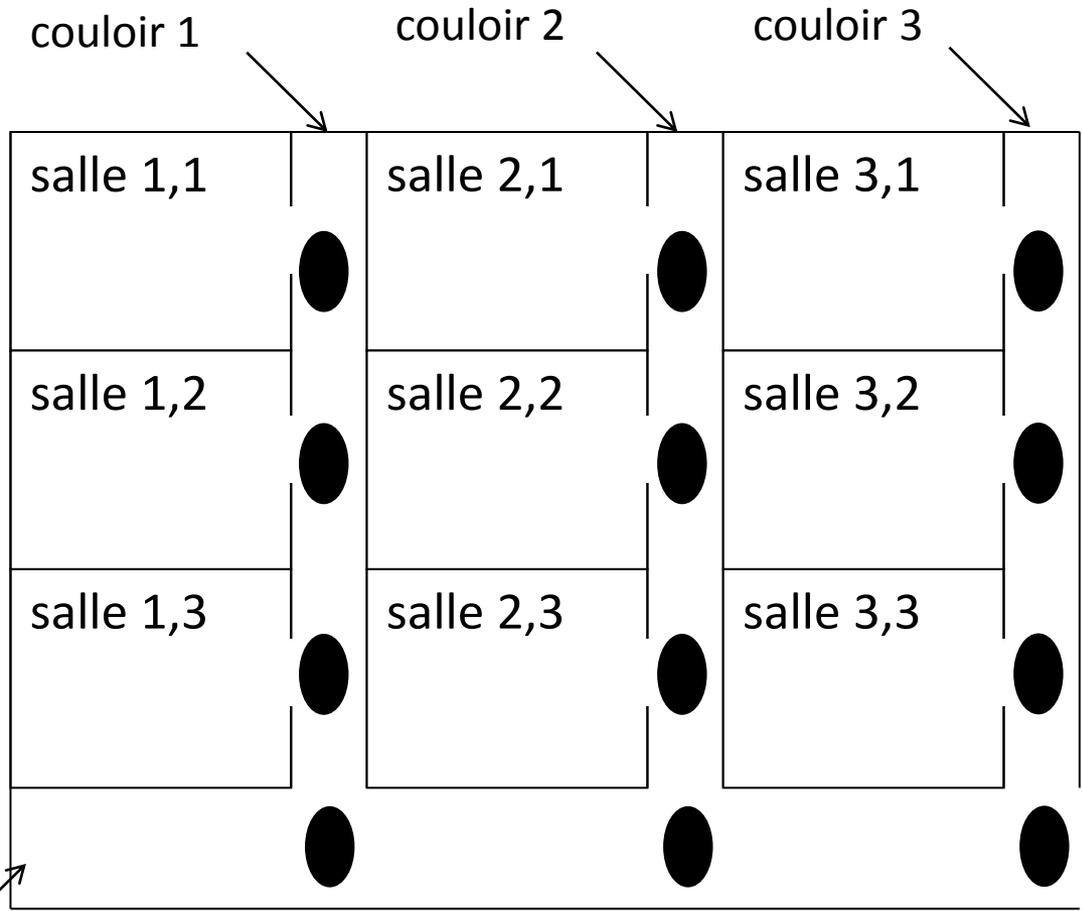


Agent responsable du
comptage des
individus sortant de
l'établissement



Couloir Principal

➔ Dans chaque couloir, les élèves des salles 2 occupent la place des élèves des salles 3, et les élèves des salles 1 occupent la place des élèves des salles 2.

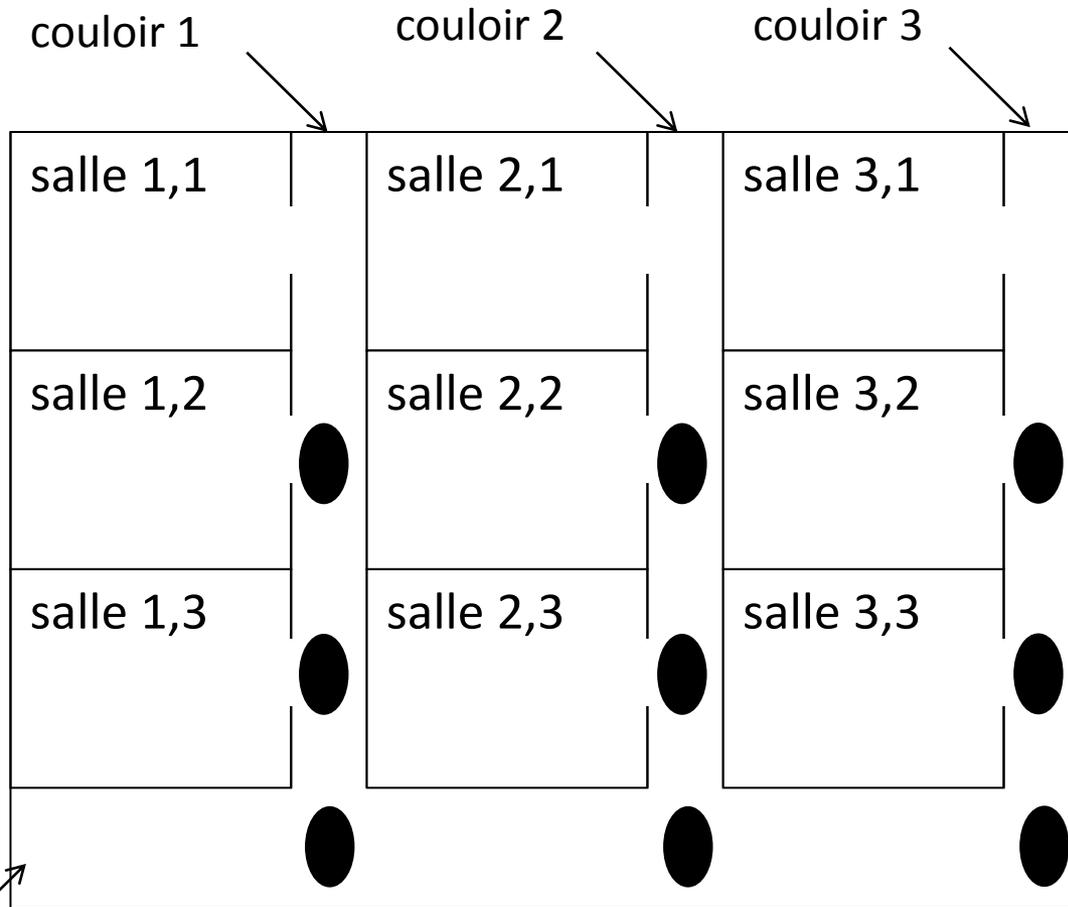


Agent responsable du comptage des individus sortant de l'établissement



Couloir Principal

➔ Les élèves des salles 3 sont redirigés groupe par groupe, les vers la sortie d'évacuation où il sont comptabilisés dans un ordre bien défini:

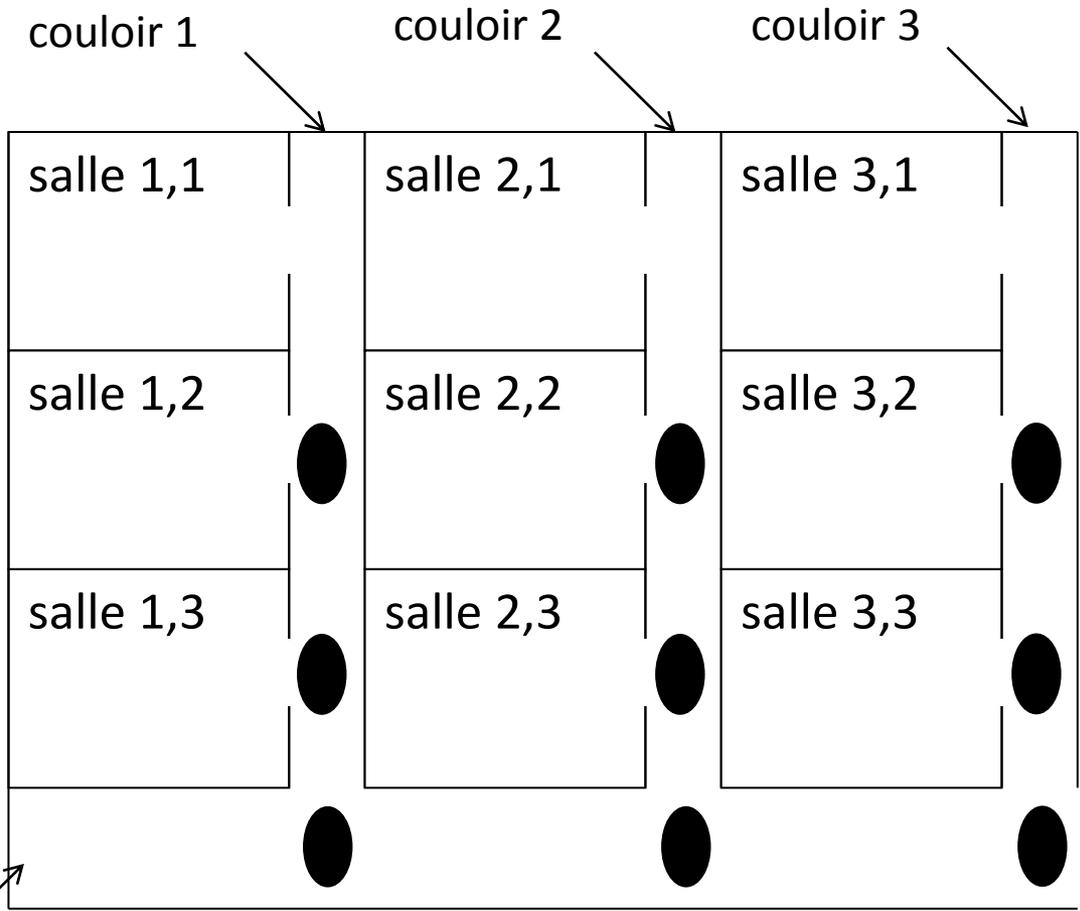


Agent responsable du comptage des individus sortant de l'établissement



Couloir Principal

➔ Ainsi de suite ...

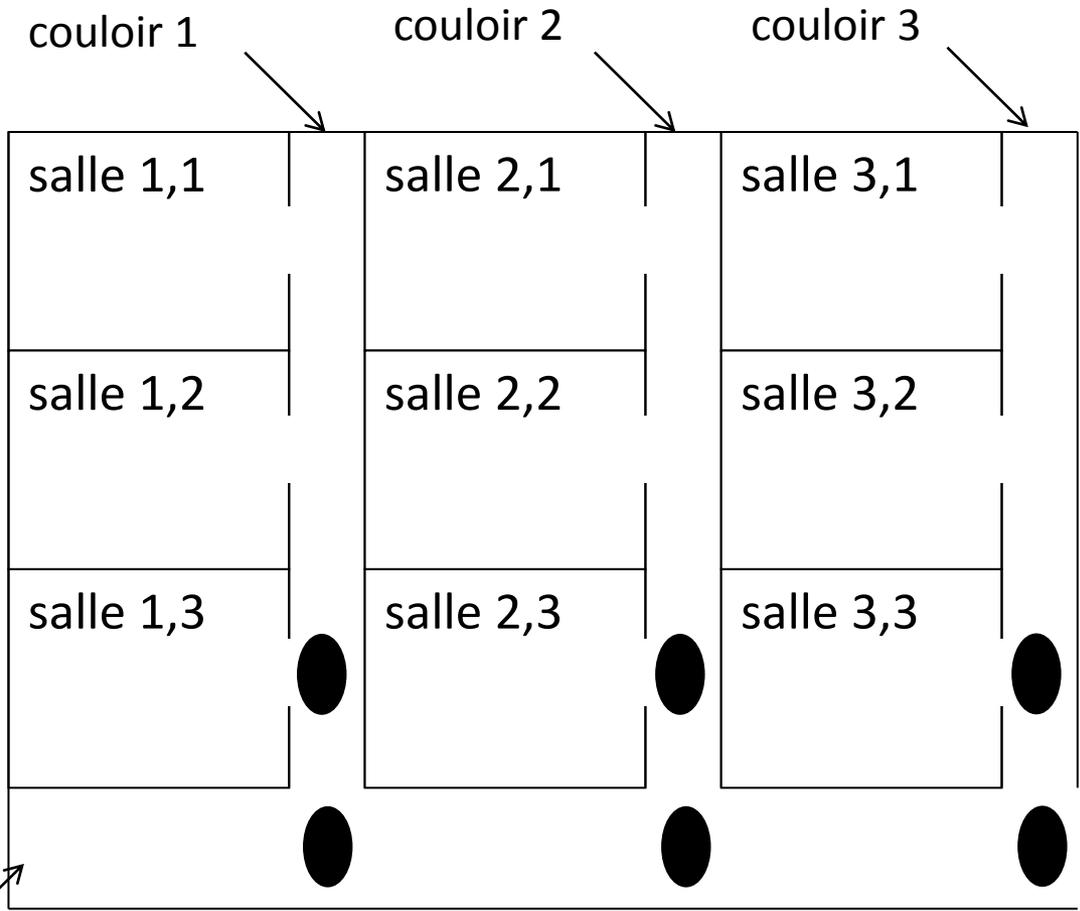


Agent responsable du comptage des individus sortant de l'établissement



Couloir Principal

➔ Ainsi de suite ...



Agent responsable du
comptage des
individus sortant de
l'établissement



Couloir Principal

 **Ce protocole d'évacuation permet de connaître à chaque instant le nombre d'élève qui se situe dans une classe donnée!!**

 **Généralisable à un nombre quelconque de couloirs et de salles...**

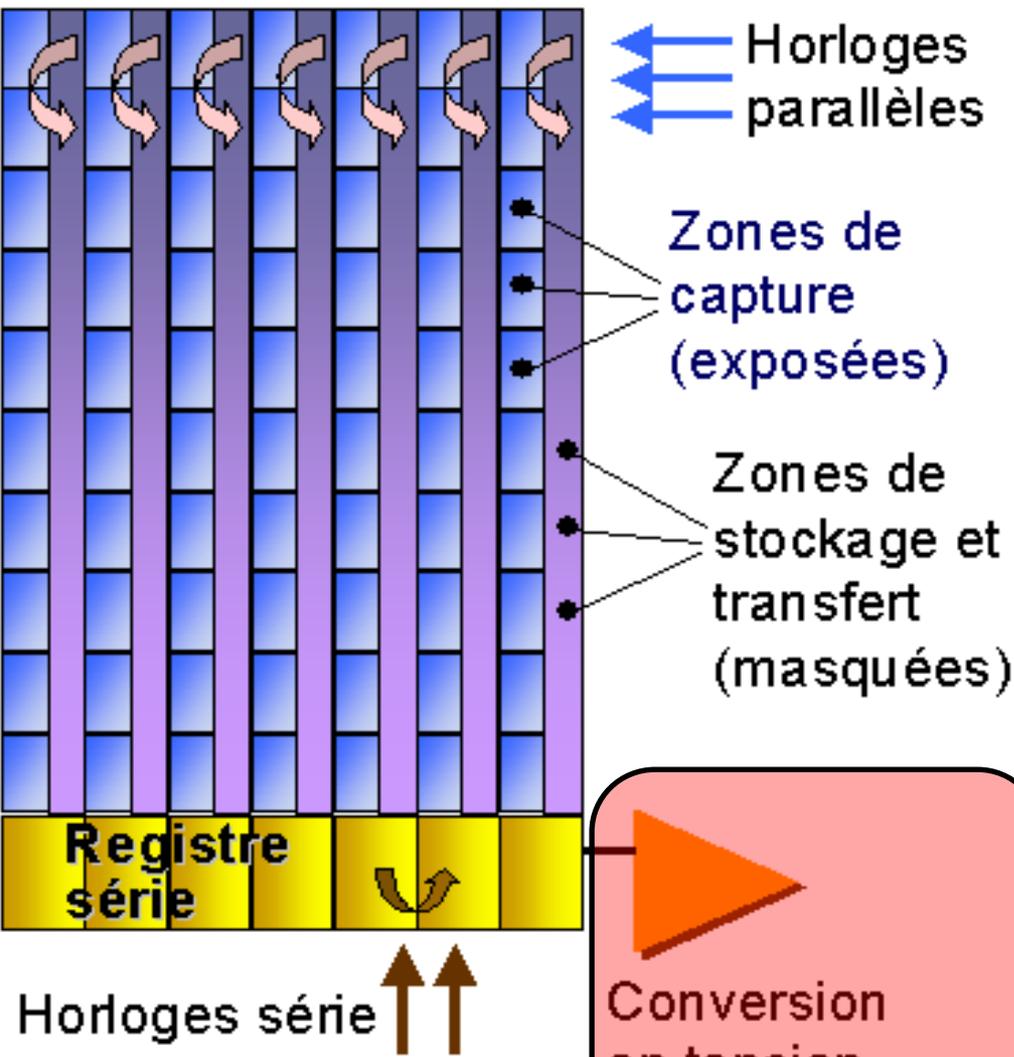


Comptabilisation identique des charges créées sur chaque photosites...

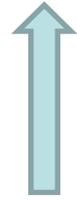


Procédé qui permet de connaître, l'intensité lumineuse reçue par chaque pixel après chaque exposition !!!

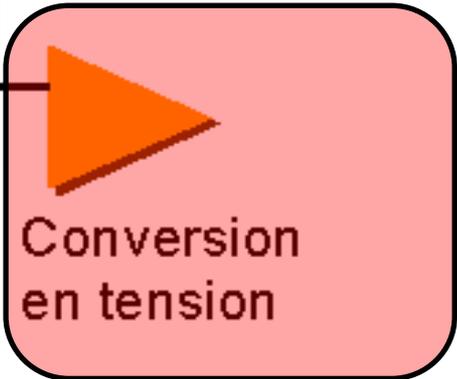
1.3. Quantification de l'image numérique



Enregistrement de la valeur sur un nombre de bits correspondant à un nombre discret et fini de niveaux de gris entre le noir et le blanc



**Pour un pixel:
Tension de sortie proportionnelle à l'intensité lumineuse**



Codage du niveau de gris et qualité d'image

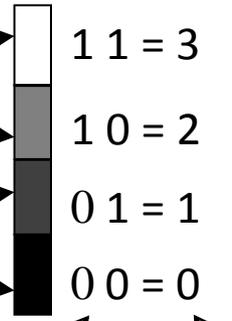
Codage sur 1 bit :

2 niveaux de gris



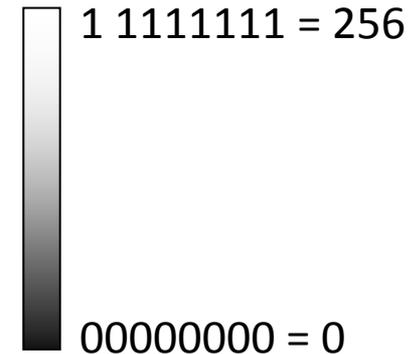
Codage sur 2 bits :

4 niveaux de gris



Codage sur 8 bits = 1 octet :

256 niveaux de gris



➔ Plus le nombre de bit utilisé est grand, plus le niveau de gris varie de façon continu.

➔ Qualité de l'image d'autant meilleur que le nb de bit alloué par pixel est important !!

↳ Inconvénient: place mémoire importante



Trouver un compromis entre qualité et place mémoire

Exemple:

1 bit / pixel :
2 niveaux de gris



2 bits/ pixel :
4 niveaux de gris



8 bits = 1 octet / pixel :
256 niveaux de gris



Remarque et Définition: Le nombre de bit utilisé pour coder le niveau de gris d'une image est appelé « profondeur de l'image ».

1 bit / pixel :

2 niveaux de gris



2 bits/ pixel :
4 niveaux de gris



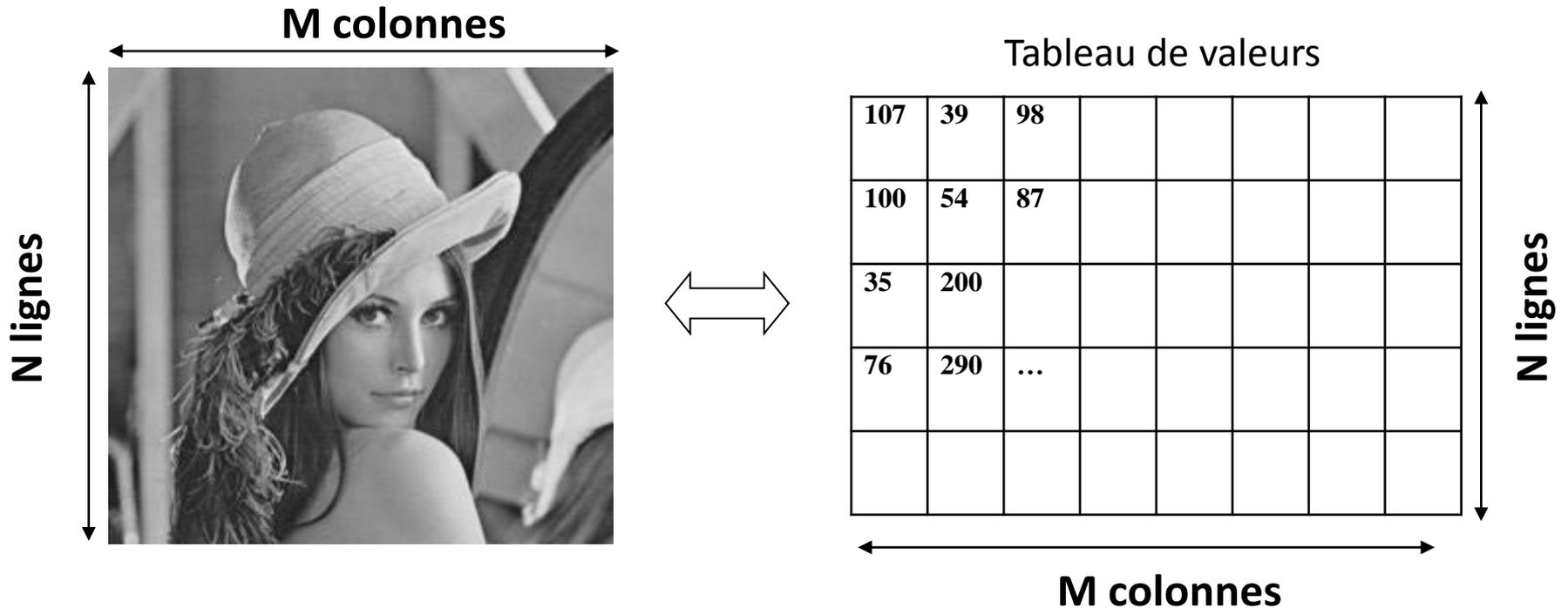
8 bits = 1 octet /pixel :
256 niveaux de gris



1.4. représentation matricielle de l'image numérique

➔ **Classiquement : 1octet/pixel.**

↳ **Valeur du niveau de gris entre 0 et 255.**



➔ **Image numérique = tableau de valeurs comprise entre 0 et 255**

Exercice1: Calculer le poids (en octet) d'une image en niveau de gris issue d'un appareil numérique dont la CCD a pour dimension 1024×768 pixels.

Exercice 2:

- 1) Ouvrir une image en noir et blanc au format PBM dans un logiciel de traitement d'image (Gimp). Réaliser le négatif de cette image.
- 2) Techniquement, réaliser le négatif d'une image revient à symétriser le niveau de gris de chaque pixel par rapport à la valeur médiane des niveaux de gris. Par exemple, le noir est transformé en blanc et vice versa. Considérons une image « I » dont le niveau de gris de chaque pixel est codé sur 1 octet. Si « x » est le veau de gris de l'image « I » considérée, et « y » le niveau de gris de la nouvelle image « J » transformée, alors la relation entre y et x est la suivante:

$$\forall x \in \{\text{pixels de l'image I}\}, y = f(x) = 255 - x$$

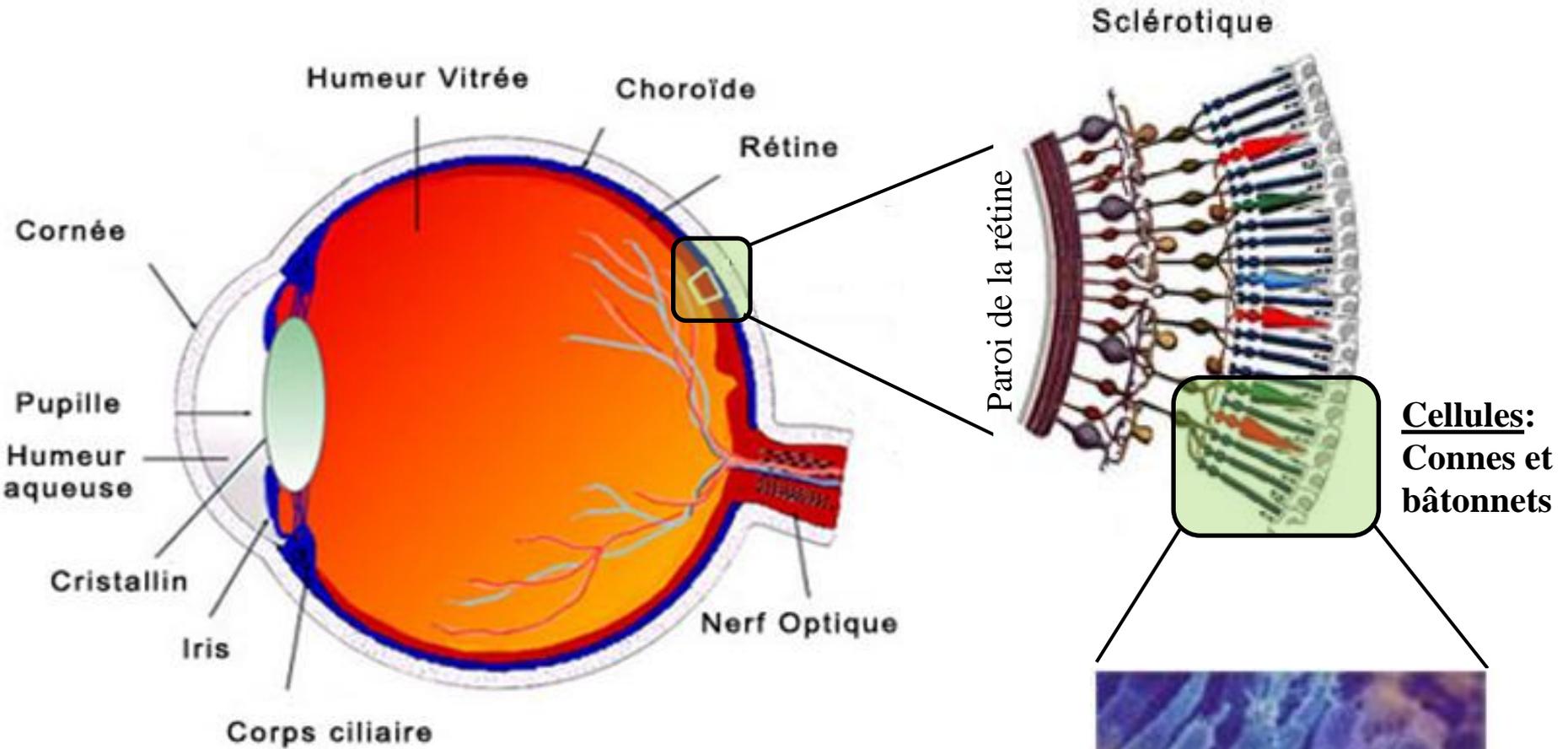
Il s'agit d'une relation affine:



➡ **Ecrire l'algorithme permettant de réaliser le négatif d'une image**

Correction:

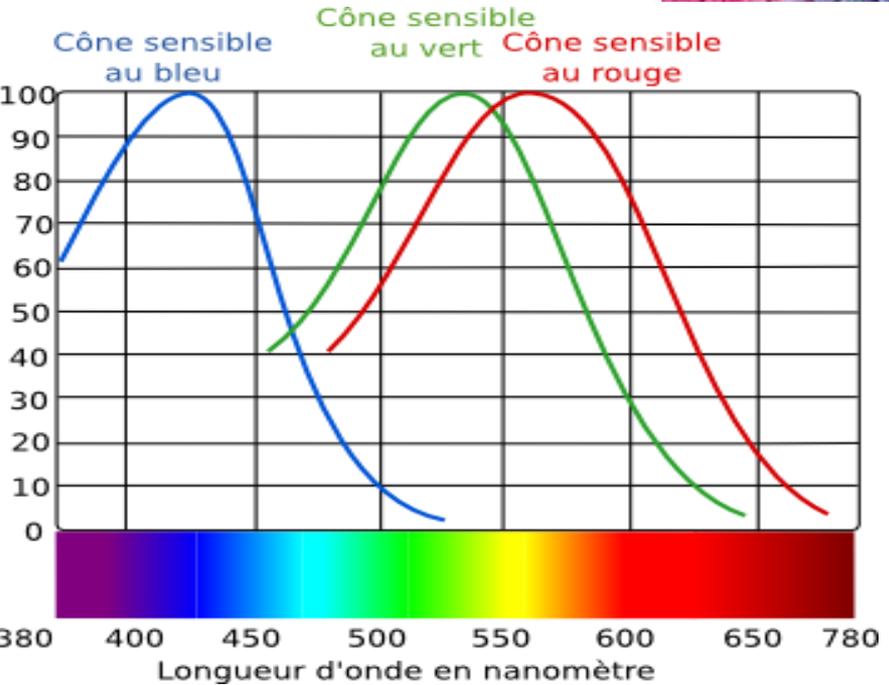
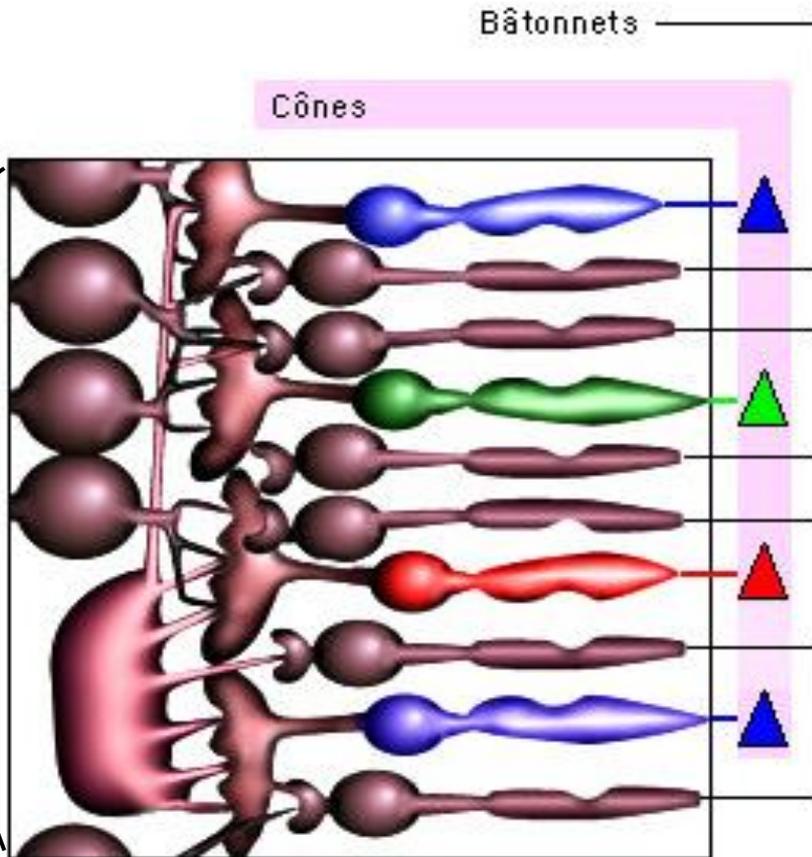
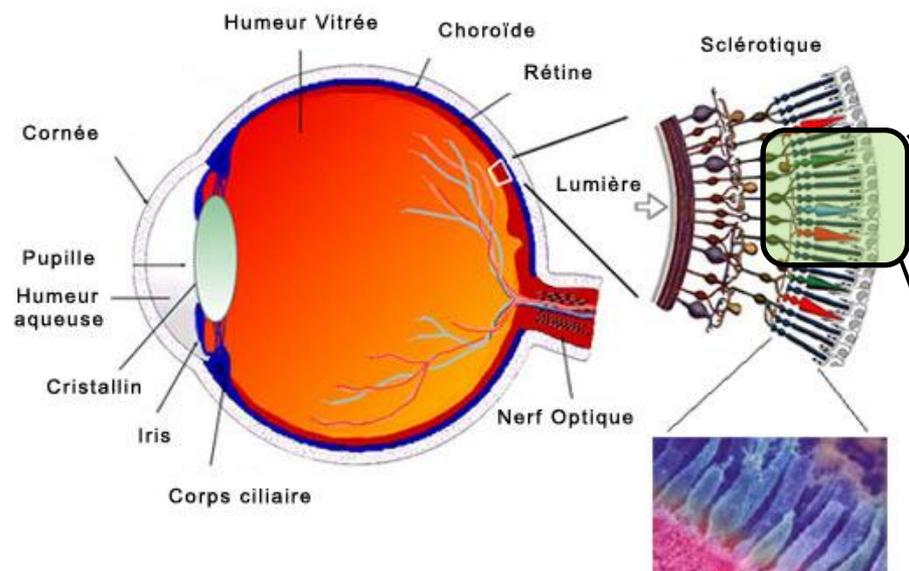
II. Images en couleurs 2.1. Perception des couleurs par l'œil humain



La sclérotique est la partie blanche et opaque de l'œil. Elle est constituée d'un tissu fibreux solide qui entoure le globe oculaire. La sclérotique contient de fins vaisseaux sanguins. Lorsque l'œil est irrité par la poussière (ou au cours d'une maladie), les vaisseaux sanguins se dilatent et le blanc de l'œil apparaît rosé ou injecté de sang.

Détail :
bâtonnets et cônes

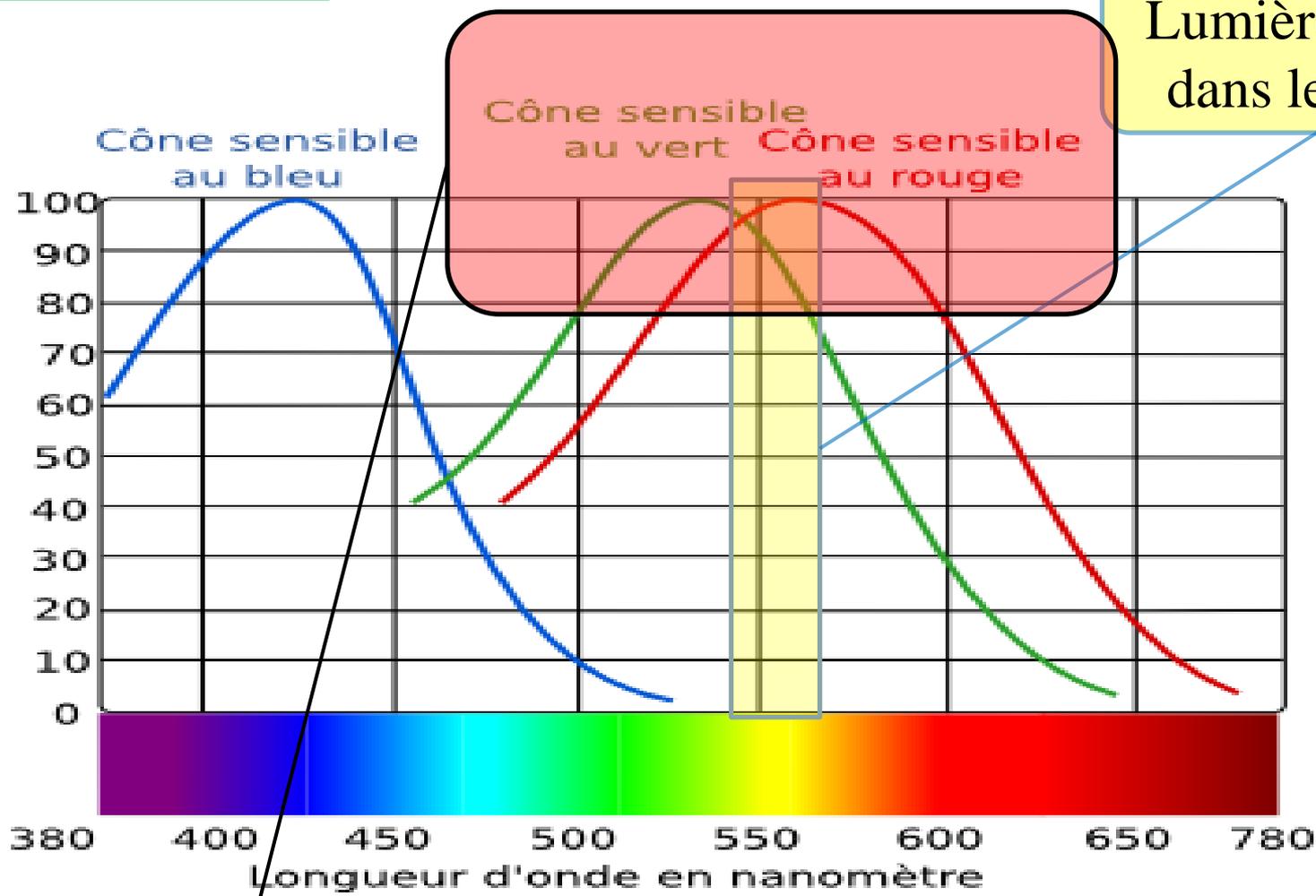
Sensibilité chromatique des cones



3 sortes de cones dont le maximum de sensibilité est situé dans :

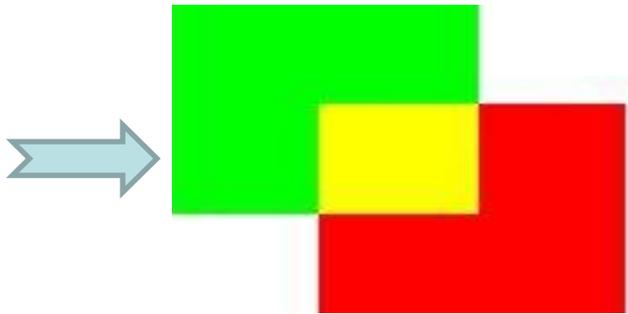
- le rouge (560nm),
- le vert (530nm),
- le bleu (424nm).

Perception des couleurs



Lumière émise dans le jaune

Cellules sensibles au vert et au rouge les plus sensibles (pas au bleu) !!

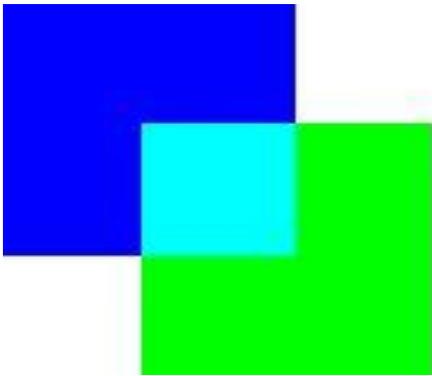


Sensation de couleur jaune identique à celle ressentie par un mélange de de vert et de rouge!

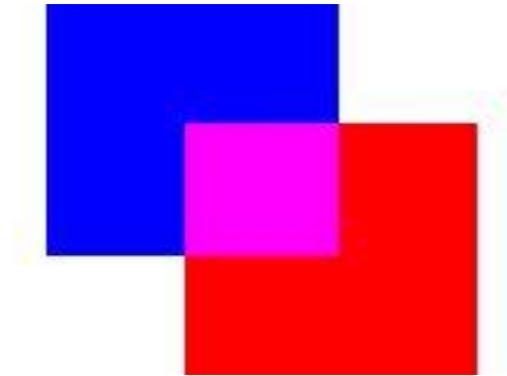
➔ Quelle que soit la longueur d'onde reçue, le cerveau reçoit uniquement les intensités du signal reçu par les cellules dans le **rouge**, le **vert** et le **bleu**.

➔ Toutes les teintes peuvent être reconstituées par une combinaison des signaux **rouge**, **vert** et **bleu**.

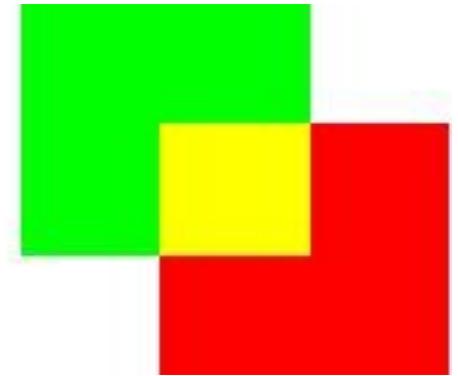
Exemple:



Vert + Bleu → Cyan



Rouge + Bleu → Magenta



Vert + Rouge → Magenta

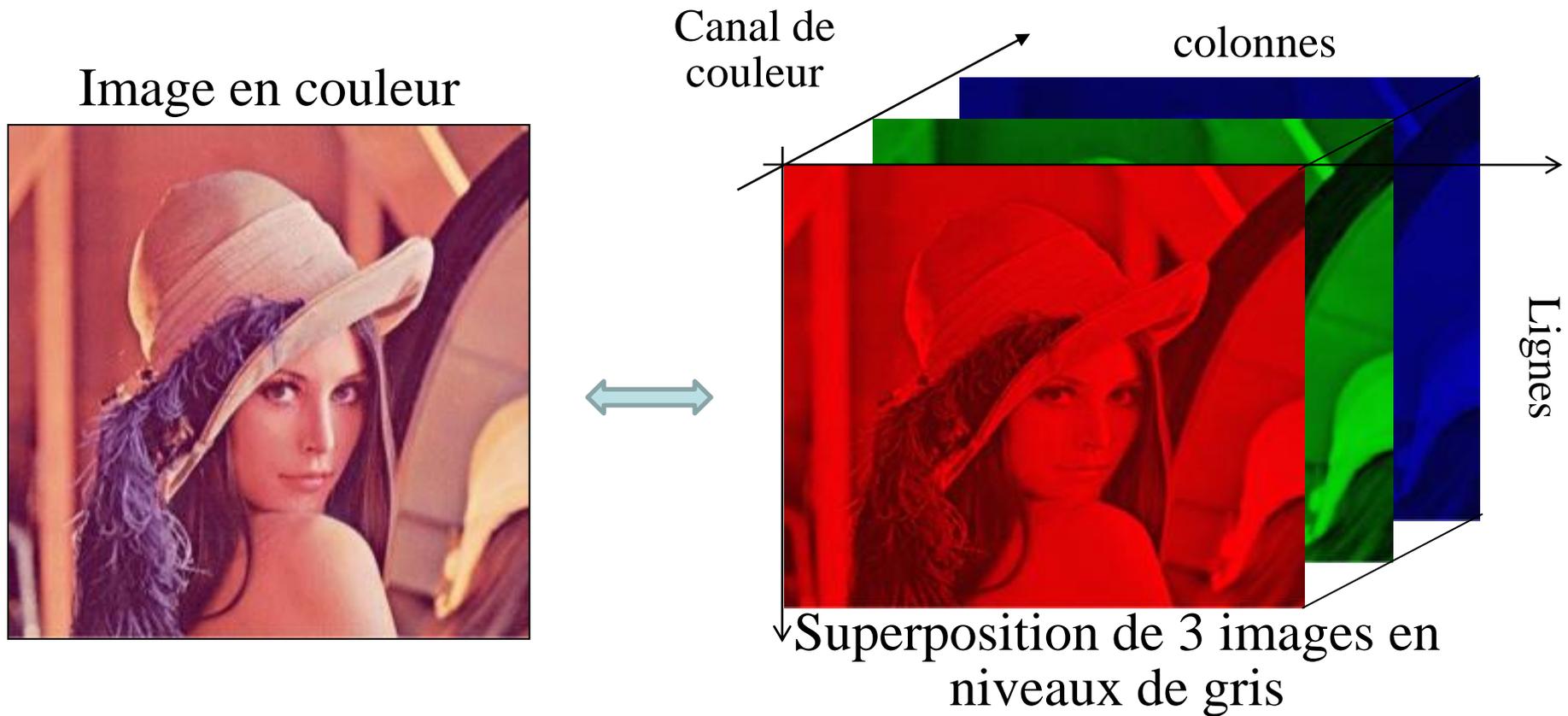


Grand intérêt technique!!!



Seulement 3 canaux RVB pour reconstituer la couleur!!!

2.2. Représentation d'une image en couleur



Modélisation : **Tableau à trois entrées (tenseur d'ordre 3) :**

⇒ Lignes: i

⇒ Colonnes : j

⇒ Couleurs : k

} $I(i,j,k)$

$0 < i < N$

$0 < j < M$

$0 < k < 3$

Exercice: Généralement, dans une image en couleur, comme pour les images en niveaux de gris, **l'intensité des pixels de chaque canal de couleurs est codé sur un octet.**

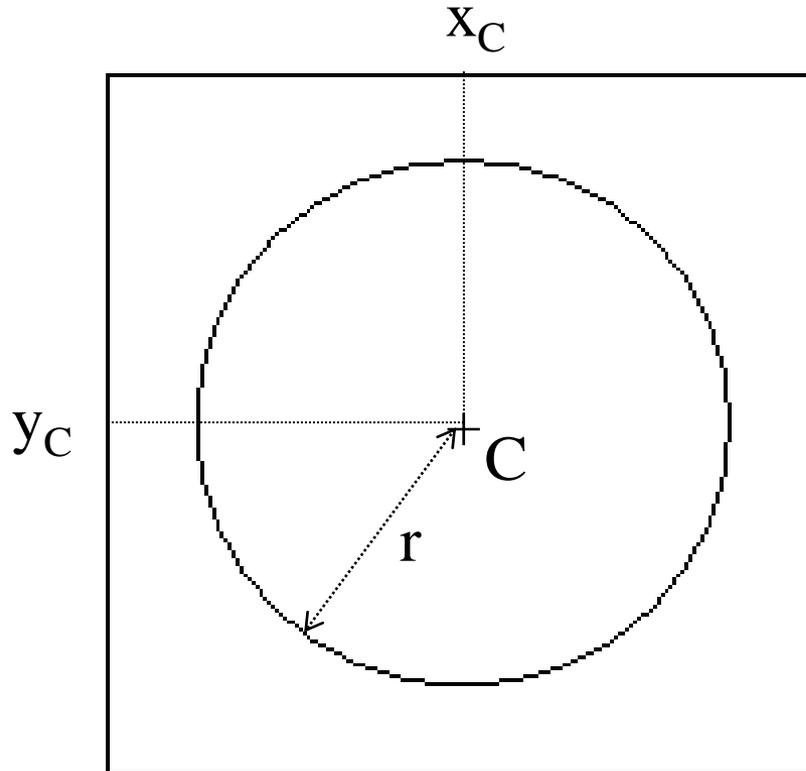
1) Sur l'image en couleur, combien d'octet attribue-on à chaque pixel ?

2) Calculer le poids (en octet) d'une image en couleur issue d'un appareil numérique dont la CCD a pour dimension 1024×768 pixels.

III. Quelques formats d'images numériques

3.1. Représentation vectorielle des images

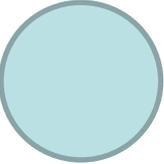
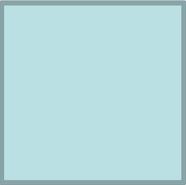
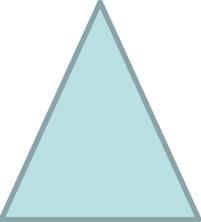
Exemple: image représentant un cercle



Représentation symbolique
=
Image vectorielle

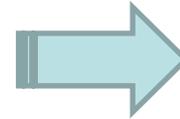
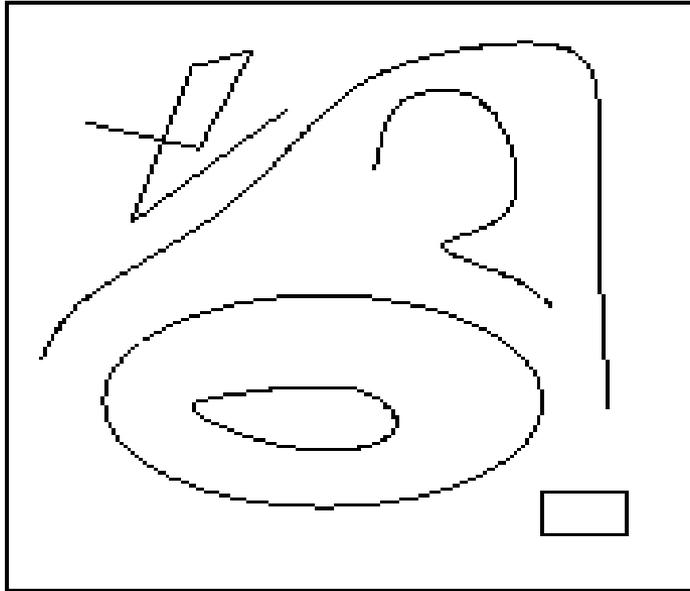
➔ **Image intégralement décrite par la donnée de 3 paramètres:**
 (x_C, y_C, z_C)

→ Beaucoup de figures géométriques simples/complexes se décrivent par une représentation vectorielle

	indicatif	paramètres	Nb paramètres
	c	(x,y,r)	3
	r	(x,y,a,b)	4
	r	(x,y,a)	3
	t	(x₁,y₁, x₂,y₂, x₃,y₃)	6

→ Avantage: poids mémoriel faible → bonne compression

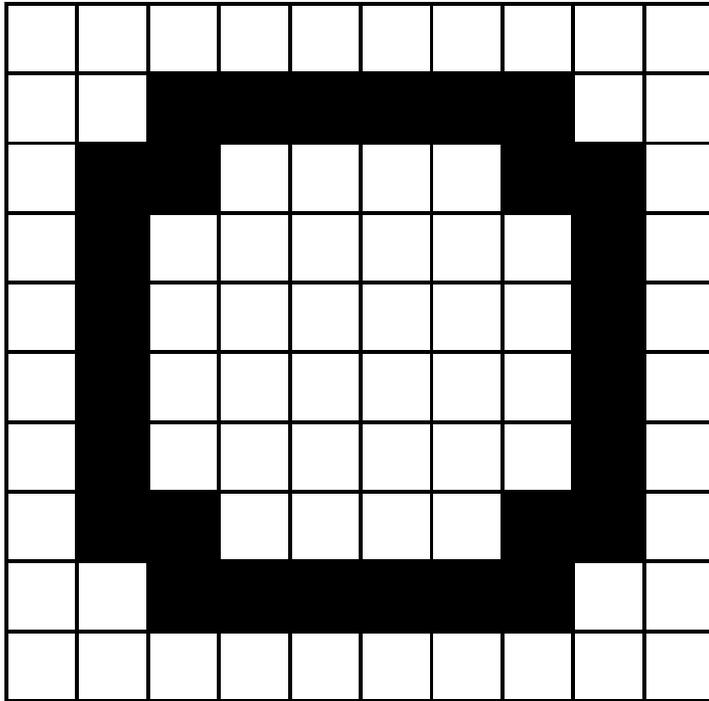
→ Problème: formes complexes



**Non représentable
par un nombre fini
de paramètres**

3.2. Représentation par pixels des images

Exemple: image représentant un cercle



➔ Superposition d'une grille
(10 × 10)

↳ 1 case = 1 pixel

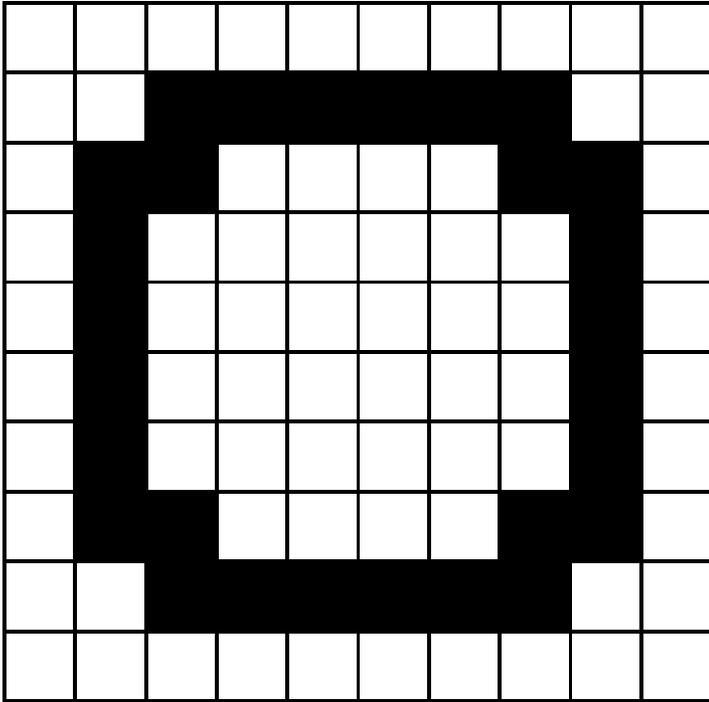
➔ On noircit les pixels qui
contiennent une portion de trait

↳ Image = succession de cases noire/blanche

↳ Lecture dans un ordre bien défini :
Gauche → Droite / Haut → Bas

➔ Représentation binaire: noir = 1 / blanc = 0

Exercice: Pour une grille (10 × 10)

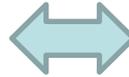
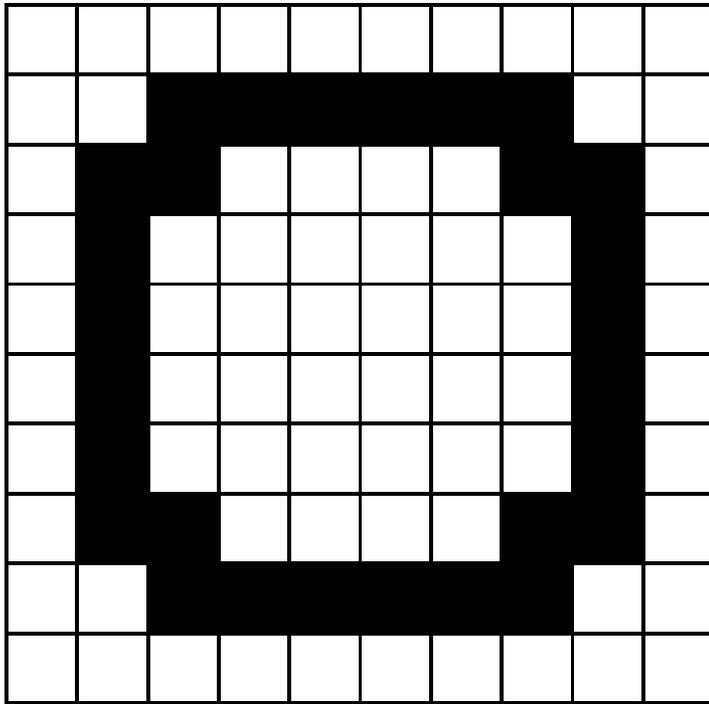


1) Donner la représentation matricielle binaire de l'image en noir et blanc.

2) En déduire le « mot binaire » correspondant à l'image en précisant son nombre de caractères.

3) Quel est ici la profondeur de l'image ?

Correction: 1) Représentation matricielle binaire



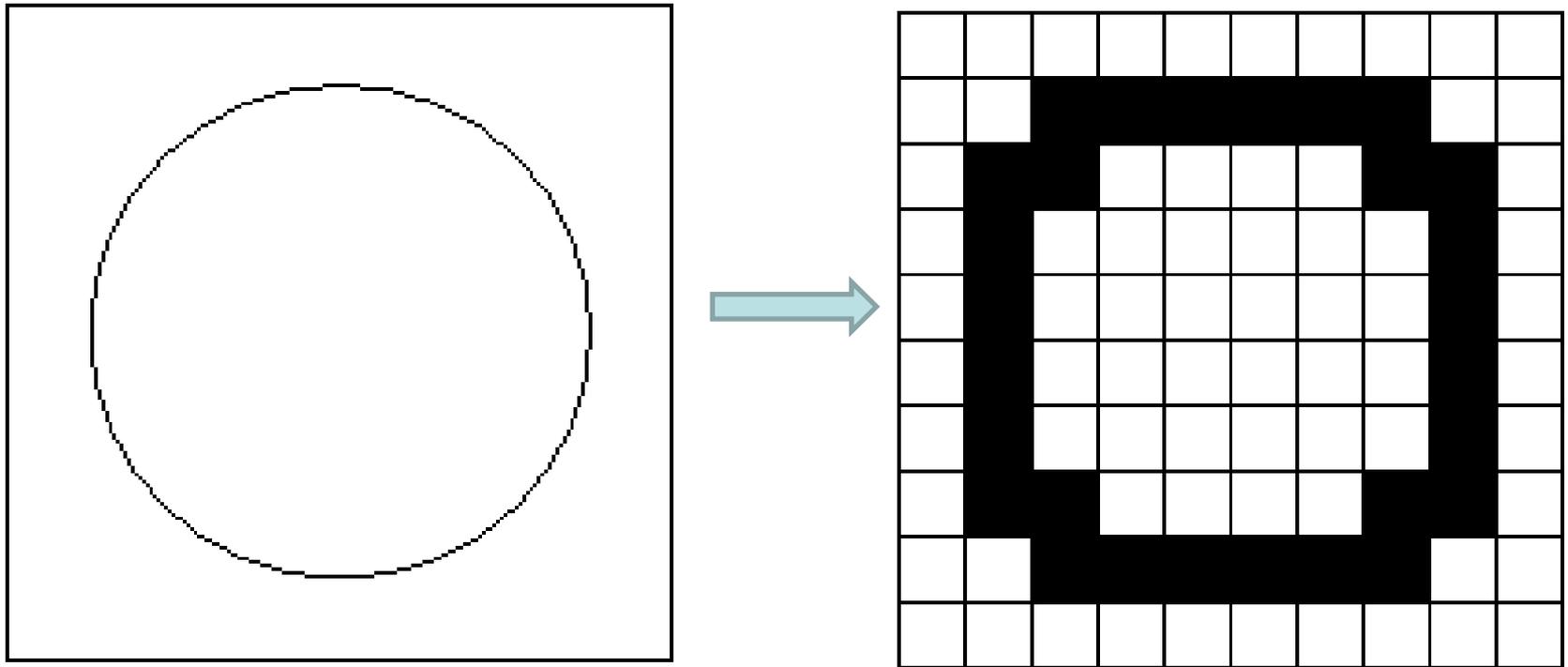
0	0	0	0	0	0	0	0	0	0
0	0	1	1	1	1	1	1	0	0
0	1	1	0	0	0	0	1	1	0
0	1	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	1	0
0	1	1	0	0	0	0	1	1	0
0	0	1	1	1	1	1	1	0	0

2) Image = mot de 100 bits :

0000000000001111100011000011001000000100100
0000100100000010010000001001100001100011111
000000000000

3) Ici 1bit/pixel (noir/blanc) ➡ Profondeur de 1 bit !!

Q: pourquoi la forme circulaire initiale n'est-elle pas bien représentée ?



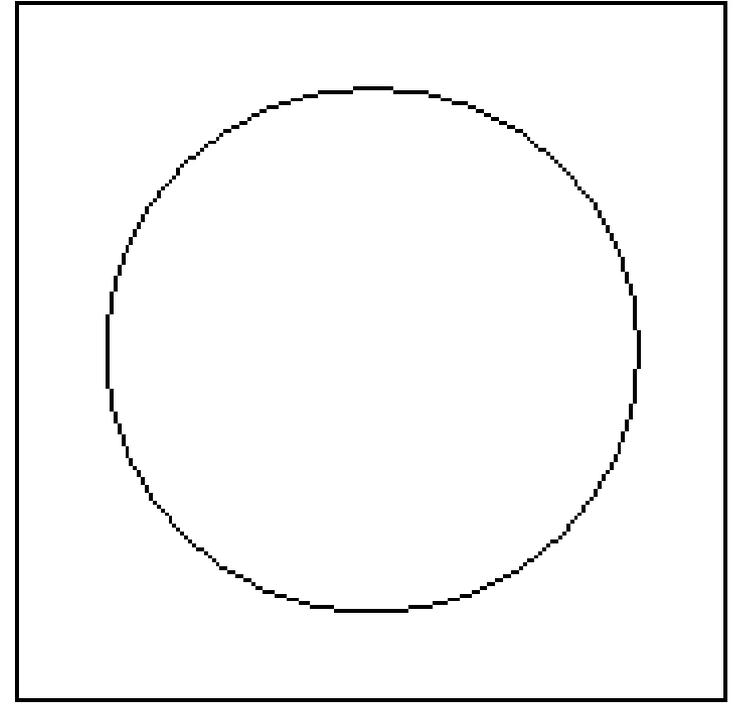
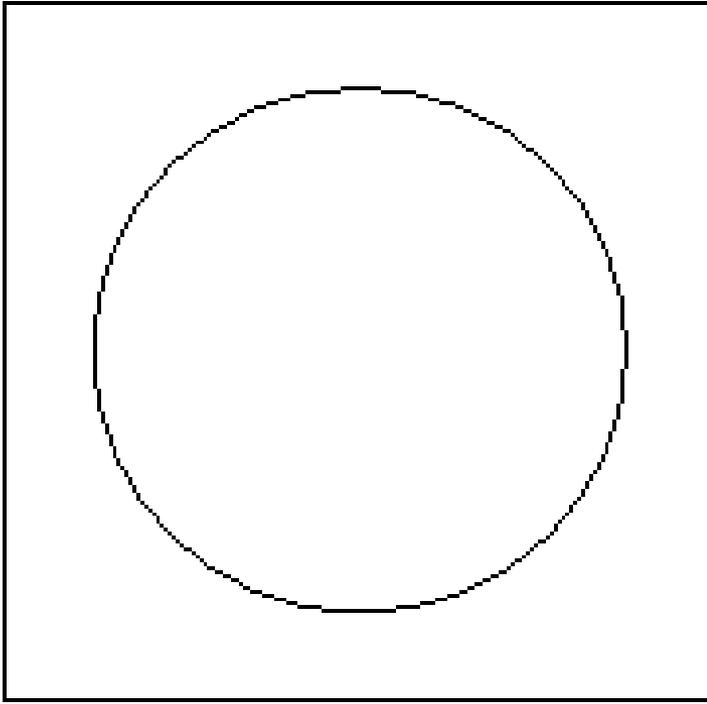
↳ **Grille trop grossière** → **Ne comporte pas assez de pixels**

↳ **Sous échantillonnage spatial !**

Q: Comment remédier à l'effet de pixellisation?

↳ **Augmenter le nombre de pixels de la grille**

Exemple: grille de (100 x 100) pixels



On n'observe plus les effets de la pixellisation



Le sur échantillonnage est inutile:

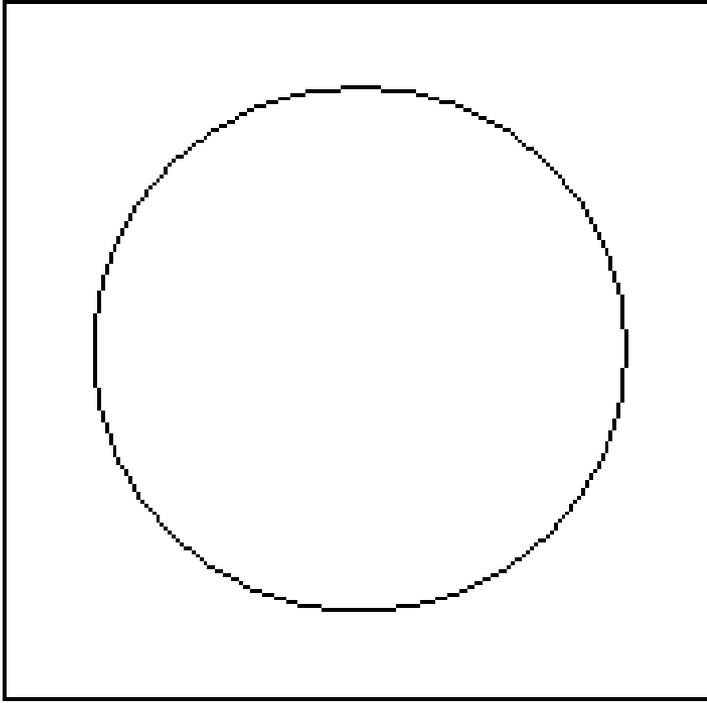


Pour une grille très fine de quelques millions de pixels l'œil est incapable de discerner deux points voisins.



Augmente inutilement le poids de l'image!!

Bilan et définition:



```
0000000000001111100011000011001  
0000001001000000100100000010010  
00000100110000110001111110000000  
0000
```

La représentation d'une image sous la forme d'une suite de pixels chacun codés sur un bit, s'appelle une bitmap. Il s'agit d'une méthode approximative, mais universelle car elle permet de décrire n'importe quelle image en noir et blanc.

3.3. Notion de format

Rappel:

**Données
numériques**

Stockées en
mémoire

fichier

**Nom = adresse de
l'emplacement en
mémoire**

**Place mémoire
(en octets)**

Q: Comment connaitre a priori la nature du fichier ?

↳ **Nécessité de définir une extension précédé d'un point**

Exemple : pour une image

PBM, PGM, PPM, PNG, JPEG, GIF, PS, PICT, TIF, etc.

Portable BitMap

Fichier image: Cercle.pbm

3.4. Image en noir et blanc : format PBM

- ➔ Le format PBM est l'un des plus simples pour exprimer des images bitmap (en noir et blanc).
- ➔ Ecrit en caractères ASCII et vérifie systématiquement l'architecture suivante:

- Le début du fichier : "**P1**" + « retour à la ligne » ou « espace »
- **Largeur de l'image en base 10**, + « retour à la ligne » ou « espace »
- **Hauteur de l'image en base 10**, + « retour à la ligne » ou « espace »
- **Liste des pixels codés sur 0 ou 1** ligne par ligne, de haut en bas et de gauche à droite **sans prendre en compte les retours à la lignes et les espaces.**
- Aucune ligne ne doit dépasser 70 caractères et toutes les lignes commençant par le caractère # sont des commentaires.

Exercice: écrire le fichier cercle.pbm de l'image précédente

```
P1 # Mon premier fichier PBM : le cercle
10 10 #taille de l'image
0000000000 # ligne 1
0011111100
0110000110
0100000010
0100000010
0100000010
0100000010
0100000010
0110000110
0011111100
0000000000 # ligne 10
```

3.5. Image en niveau de gris: format PGM

- ➔ Le format PGM est l'un des plus simples pour exprimer des images en niveaux de gris.
- ➔ Généralement l'intensité du niveau de gris est codé sur **un octet**, ce qui autorise **256 niveaux entre le noir (0) et le blanc (255)**.
- ➔ format PGM très similaire format PBM.
- ➔ Ecrit en caractères ASCII et vérifie systématiquement l'architecture suivante:

- Le début du fichier : "P2" + « retour à la ligne » ou « espace »
- **Largeur de l'image en base 10**, + « retour à la ligne » ou « espace »
- **Hauteur de l'image en base 10**, + « retour à la ligne » ou « espace »
- **Valeur maximale M pour exprimer les niveaux de gris en base 10**, + « retour à la ligne » ou « espace ».
Si chaque pixel est codé sur un octet, cette valeur est à 255 et correspond au blanc
- **Liste des pixels (entre 0 et M) ligne par ligne, de haut en bas et de gauche à droite sans prendre en compte les retours à la lignes et les espaces.**
- Aucune ligne ne doit dépasser 70 caractères et toutes les lignes commençant par le caractère # sont des commentaires.

3.6. Image en couleurs: format PPM

- ➔ Superposition de 3 images en niveaux de gris dans les trois teintes **rouge**, **verte** et **bleu**.
- ➔ Niveau de gris dans une couleur particulière est codé sur 1 octet, → **256 niveaux** variant entre le noir (0) et le rouge/vert/bleu pur intense (255).
- ➔ format PPM très similaire format PBM et PGM.
- ➔ Ecrit en caractères ASCII et vérifie systématiquement l'architecture suivante:

- Le début du fichier : "P3" + « retour à la ligne » ou « espace »
- **Largeur de l'image en base 10**, + « retour à la ligne » ou « espace »
- **Hauteur de l'image en base 10**, + « retour à la ligne » ou « espace »
- **Valeur maximale M pour exprimer les niveaux de gris en base 10**, + « retour à la ligne » ou « espace ».
Si chaque pixel est codé sur un octet, cette valeur est à 255 et correspond au blanc
- **Liste des valeurs des couleurs, trois par pixel, dans l'ordre rouge, vert, bleu (entre 0 et M)**, ligne par ligne, de haut en bas et de gauche à droite **sans prendre en compte les retours à la lignes et les espaces.**
- Aucune ligne ne doit dépasser 70 caractères et toutes les lignes commençant par le caractère # sont des commentaires.

Exemple: En mélangeant du rouge et du vert en quantités égales, on obtient du jaune en augmentant la quantité de rouge, par exemple, rouge = 237, vert = 127, bleu = 16, on obtient du orange. On peut alors écrire un fichier PPM qui représente un carré orange de 100 pixels sur 100 pixels.

P3

Mon premier fichier PPM : orange

100 100

255

237 127 16

237 127 16

237 127 16

237 127 16

...

Exercice 1:

- 1) Lequel des formats PBM, PGM et PPM est adapté pour représenter un carré noir de 10 pixels sur 10 pixels ?
- 2) Même question pour un carré rouge de même taille.
- 3) Comparer les taille des fichiers obtenus

2) Pour un carré rouge, il faut obligatoirement recourir au format PPM, seul capable de représenter de la couleur. Le rouge se représente par les trois nombres 255 0 0 : intensité maximale pour le rouge et nulle pour le vert et le bleu. On obtient le fichier suivant :

```
P3
# Un carré rouge
10 10
255
255 0 0
255 0 0
255 0 0
...
(100 lignes
identiques)
```

3) Le fichier représentant la seconde image est significativement plus gros que celui représentant la première.

Exercice 2:

- 1) Ouvrir une image en noir et blanc au format PBM dans un éditeur de texte (Bloc Note, note pad, etc...). Ecrire un carré noir de 10 par 10 pixel sur le coin en haut à gauche de l'image.
- 2) Même question pour une image en couleur au format PPM en dessinant un carré rouge de 10 par 10 pixels.