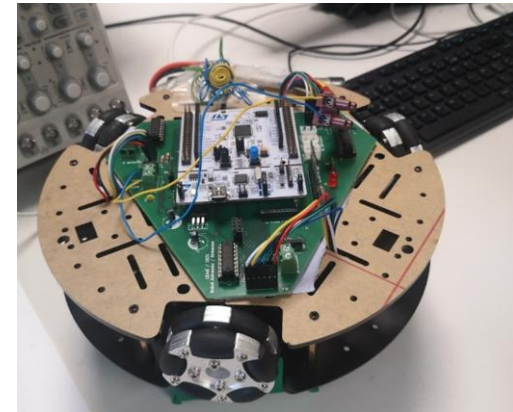
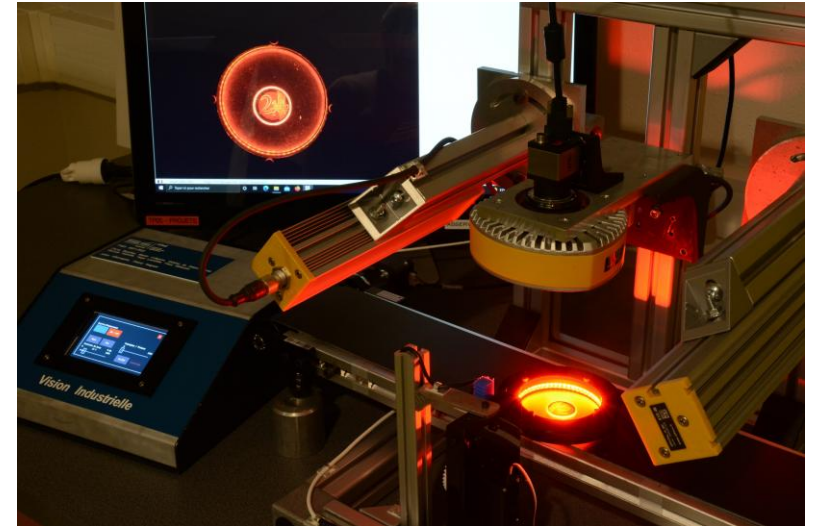
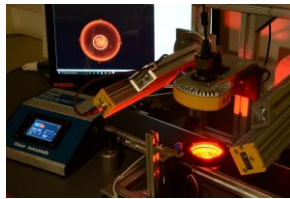


# Interfaçage Numérique

## Vision Industrielle

Julien VILLEMEJANE





# Vision Industrielle

## Machine Vision

Système basé sur un **système imageant** permettant d'**automatiser les procédés d'inspection** de produits



**Prendre une décision**

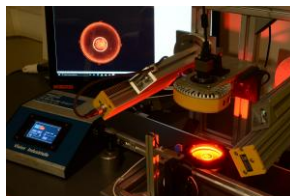
### Contrôle Qualité / Tri d'objets

- Détecter des défauts ou irrégularités
- Vérifier l'uniformité de surface
- Compter ou/et trier des objets

### Gain en efficacité et en répétabilité

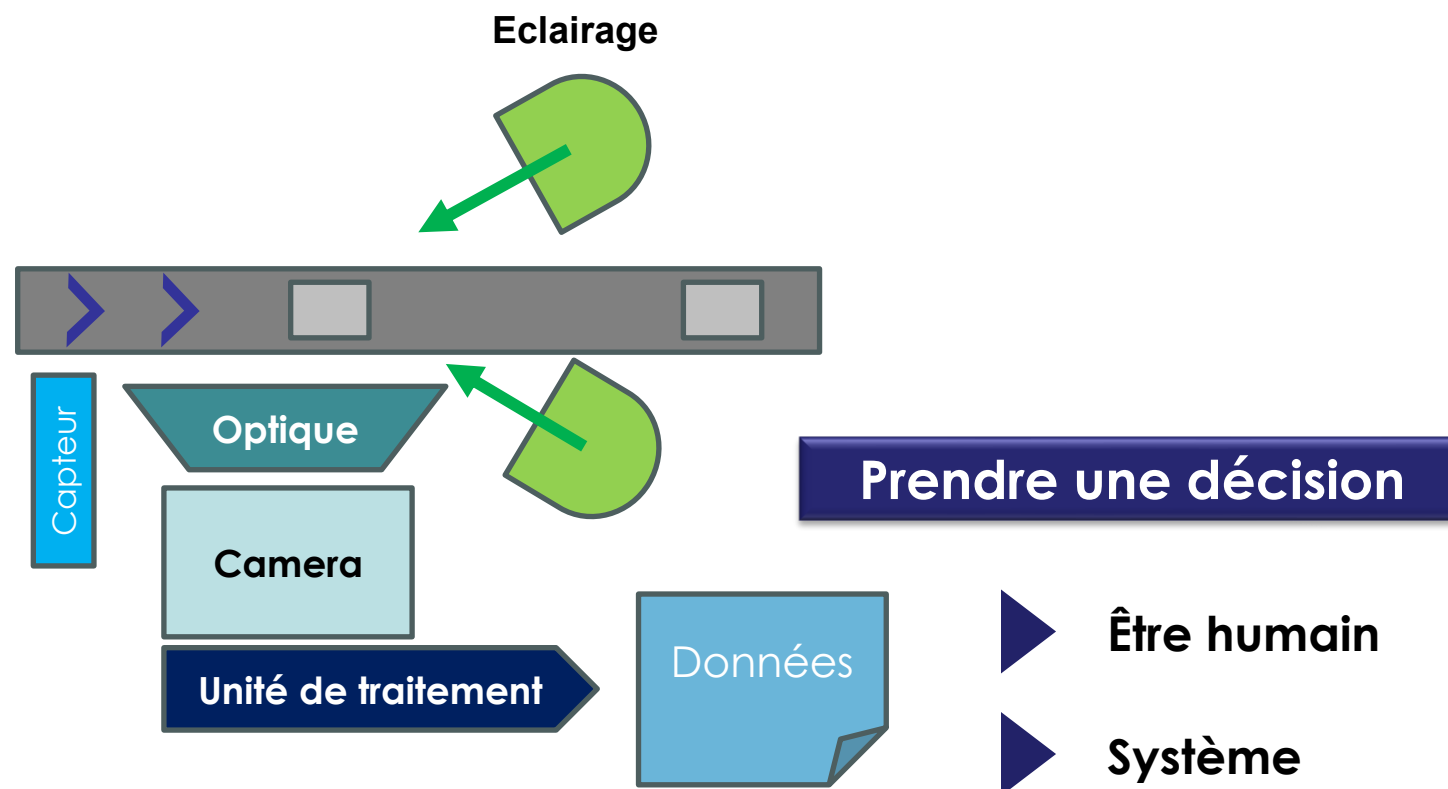
- Inspection en temps réel et à vitesse élevée
- Opérations en continu

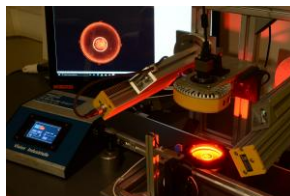




# Vision Industrielle

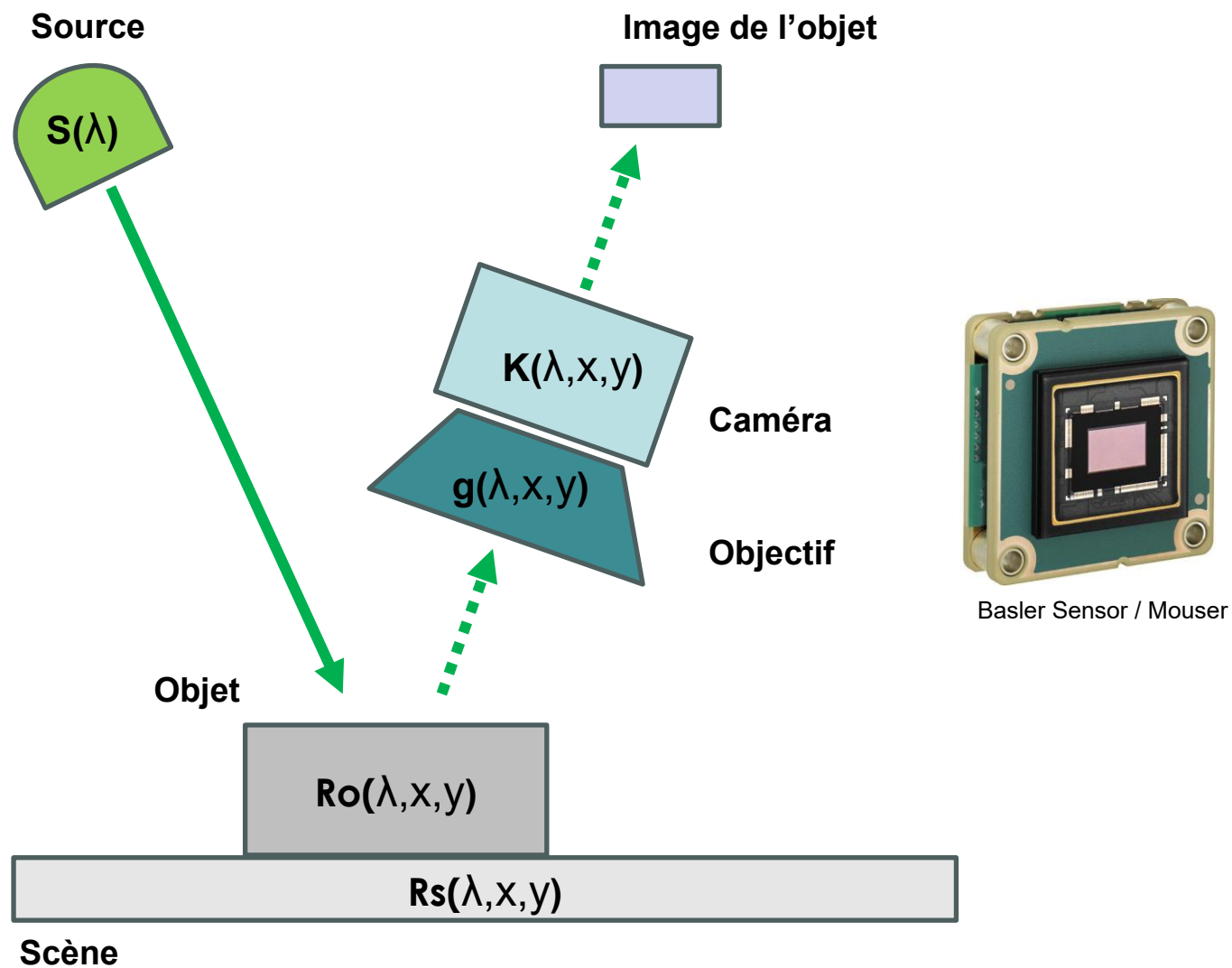
## Éléments constitutifs





# Vision Industrielle

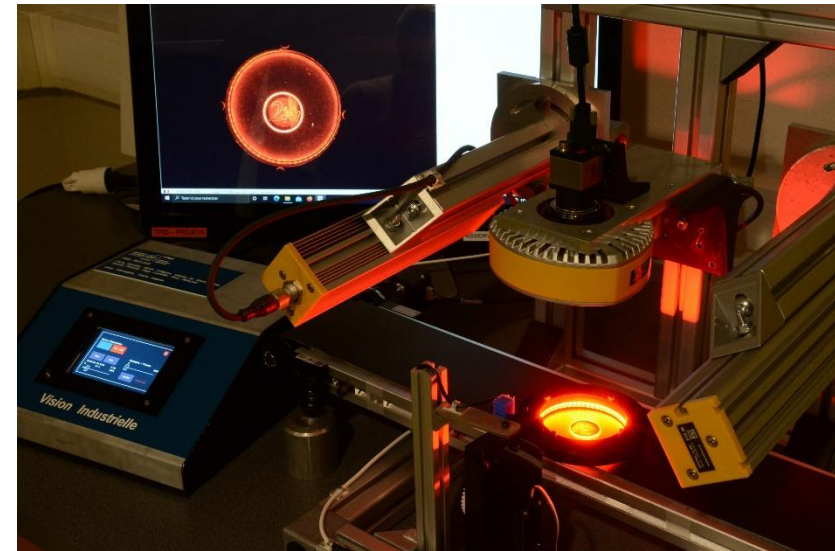
## Modélisation de la chaîne

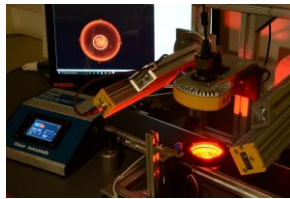




# Objets / Sources

Eclairage / Colorimétrie

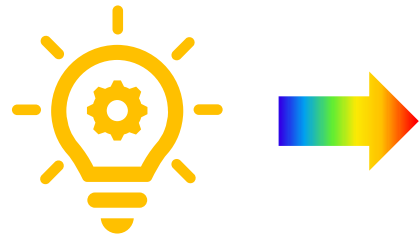




# Sources

## Sources primaires

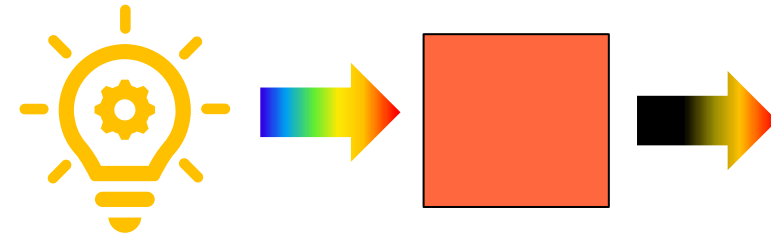
Produisent leur propre lumière



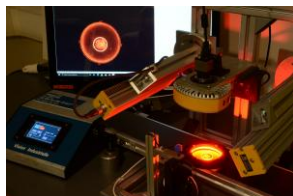
Caractérisées par leur **spectre d'émission**

## Sources secondaires

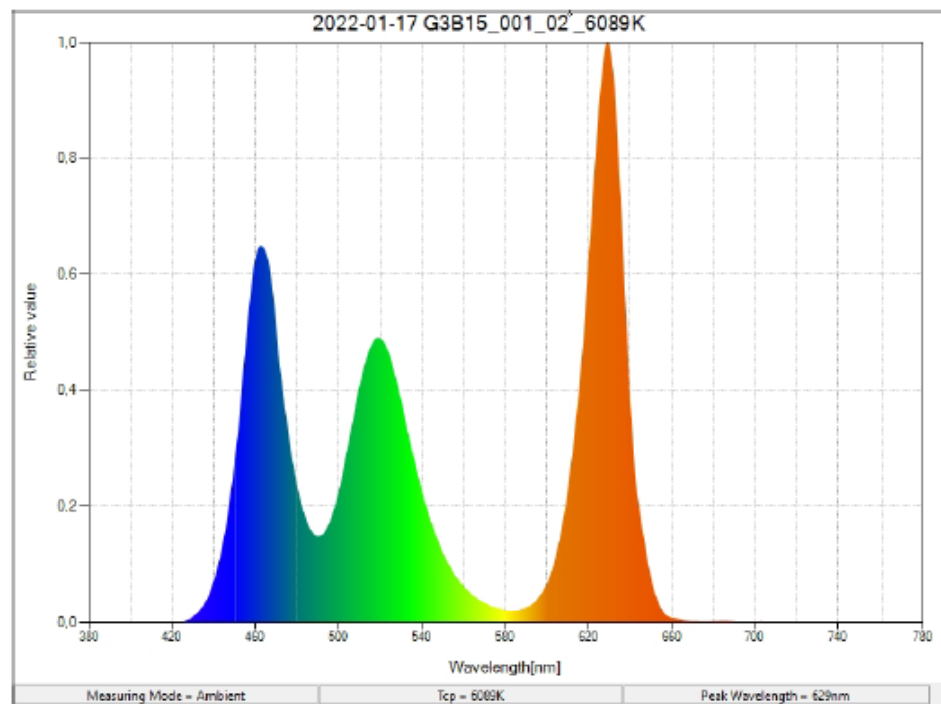
Diffusent la lumière produite par une source primaire



Caractérisées par le **spectre de l'illuminant**  
et leur **spectre en réflectance**



## Spectre d'émission

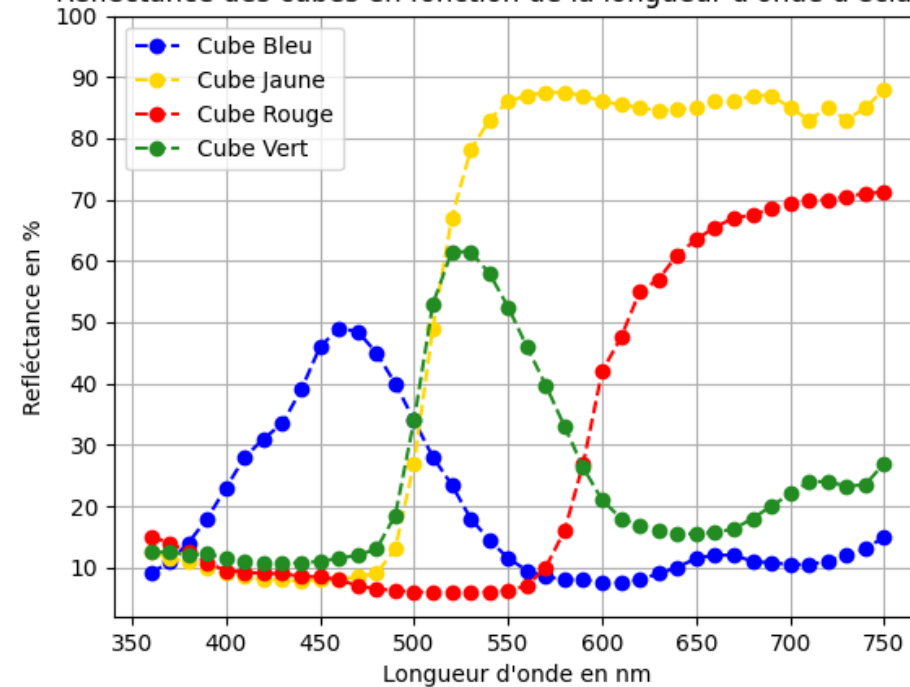


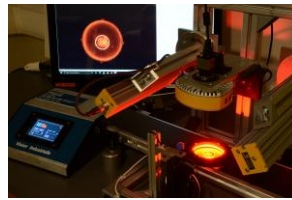
## Sources

## Réflectance



Réflectance des cubes en fonction de la longueur d'onde d'éclairage



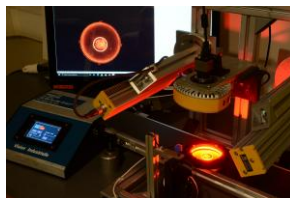


# Eclairage

---

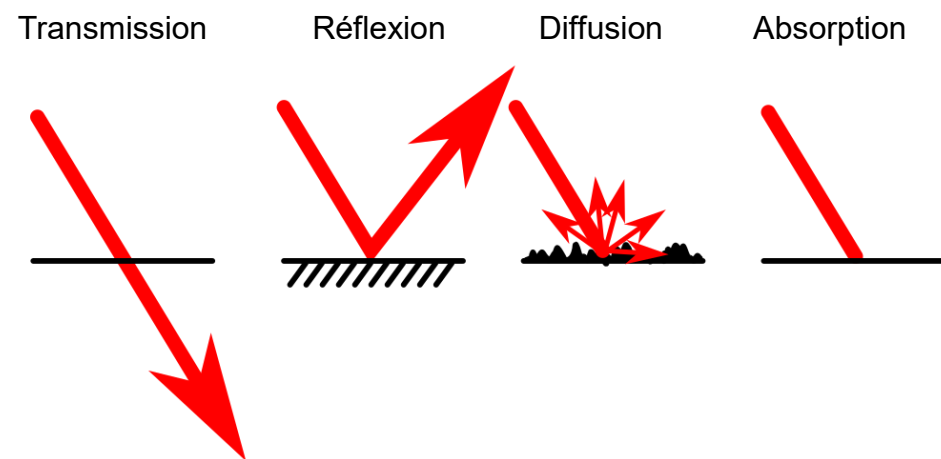
Uniformité de l'éclairage





# Eclairage

## Impact du type d'éclairage / Nature des objets



Directif



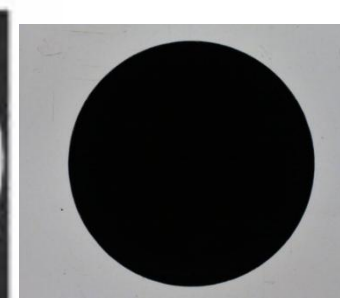
Diffus



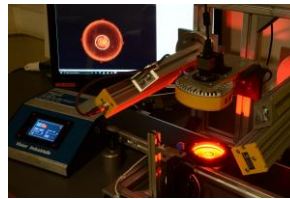
Rasant



Coaxial



Retro



# Colorimétrie

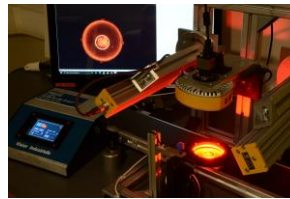
*Image prise par un capteur optique  
(sans balance des blancs)*



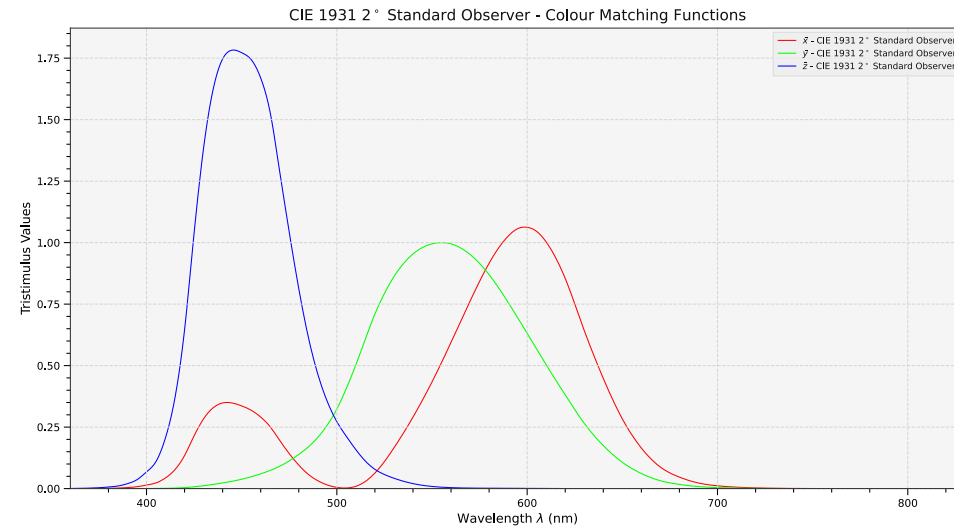
*Image vue par un humain  
(grâce à l'adaptation chromatique)*



► Fairchild, Color Appearance models

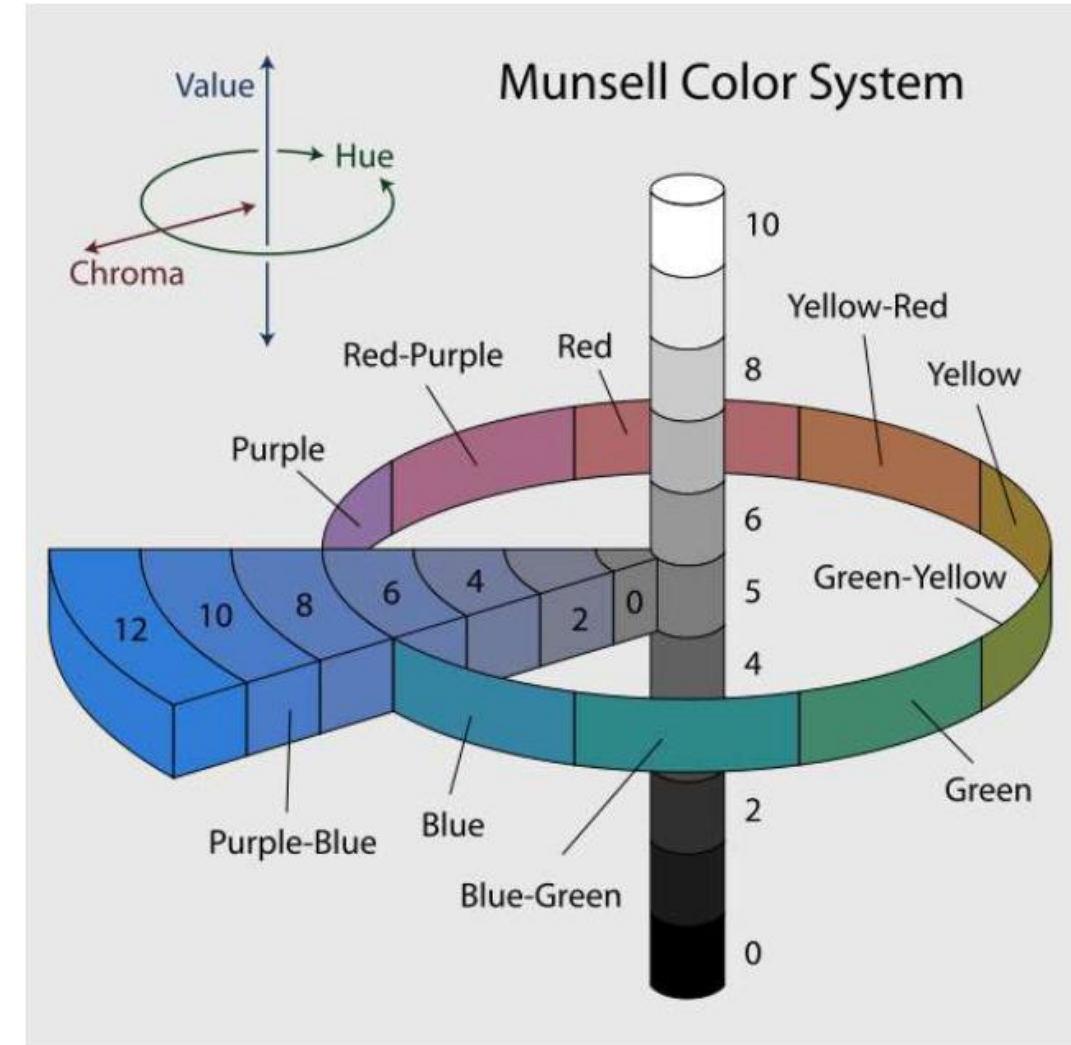


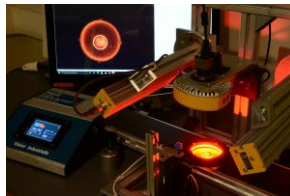
# Colorimétrie



$$K_m = 683 \text{ lm/W}$$

$$\begin{cases} X = K_m \int_0^\infty \bar{x}(\lambda) L_{e,\lambda}(\lambda) d\lambda \\ Y = K_m \int_0^\infty \bar{y}(\lambda) L_{e,\lambda}(\lambda) d\lambda \\ Z = K_m \int_0^\infty \bar{z}(\lambda) L_{e,\lambda}(\lambda) d\lambda \end{cases}$$



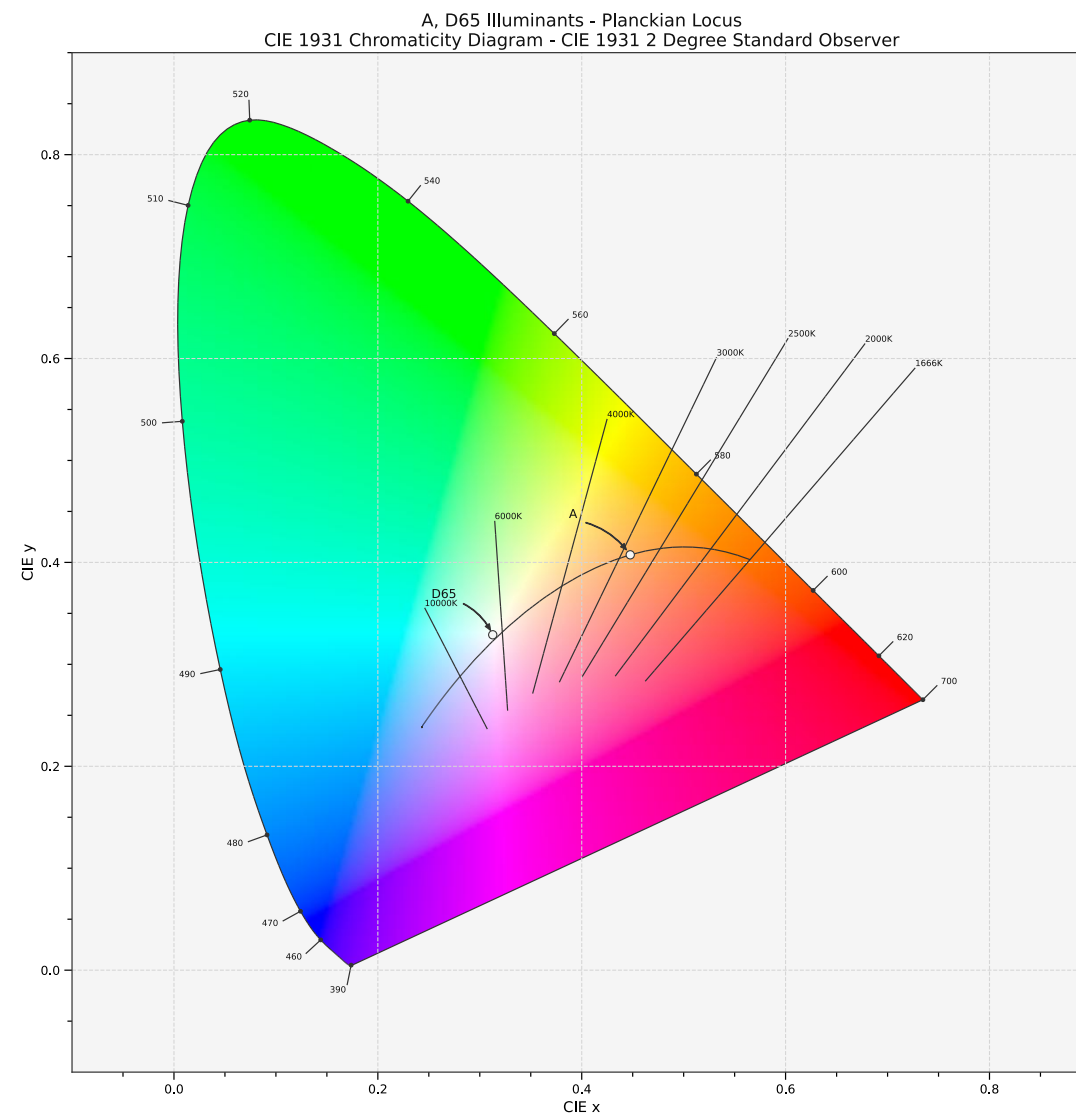


# Colorimétrie

Diagramme de chromaticité CIE 1931 xy

$$\begin{cases} x = \frac{X}{X + Y + Z} \\ y = \frac{Y}{X + Y + Z} \end{cases}$$

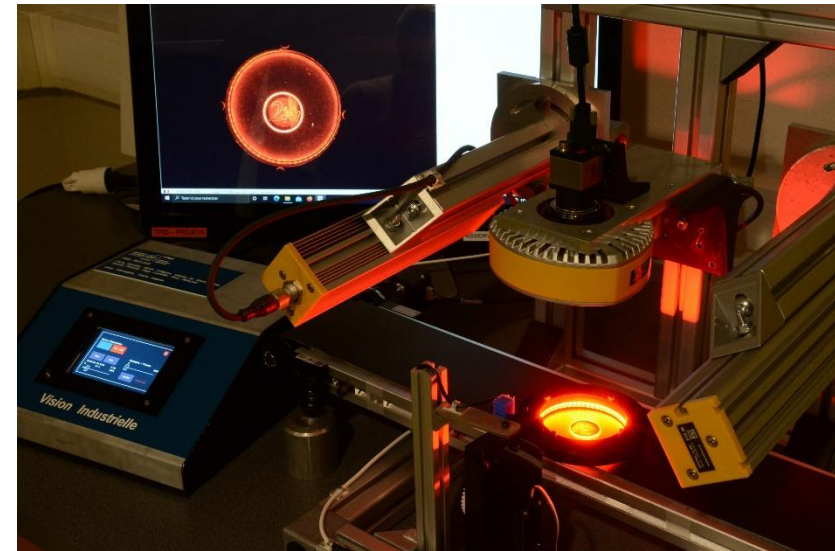
Les coordonnées (x,y) définissent la couleur  
de la source échantillon



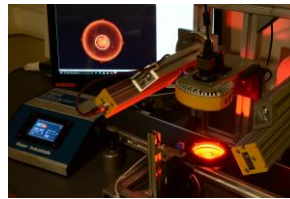


# Objectif optique

Créer une image exploitable

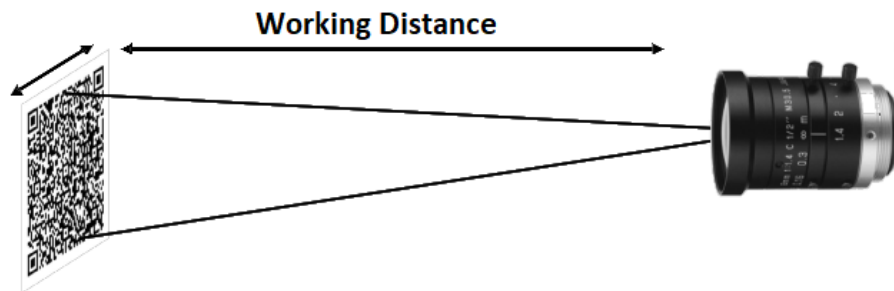
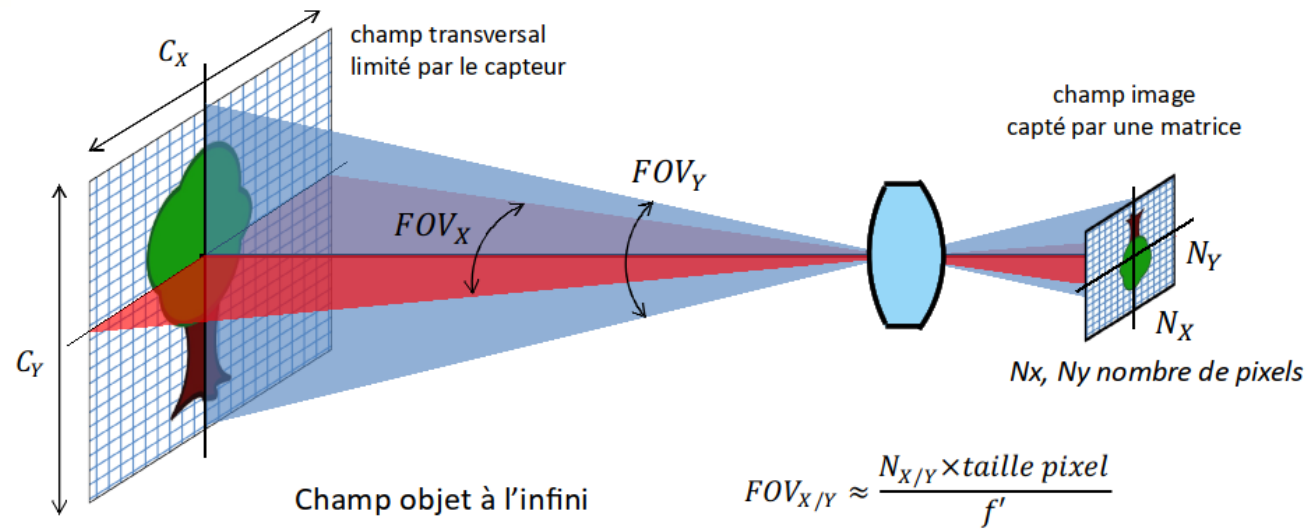


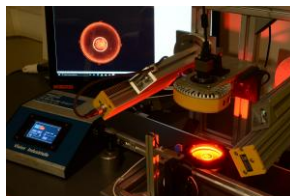




# Objectif optique

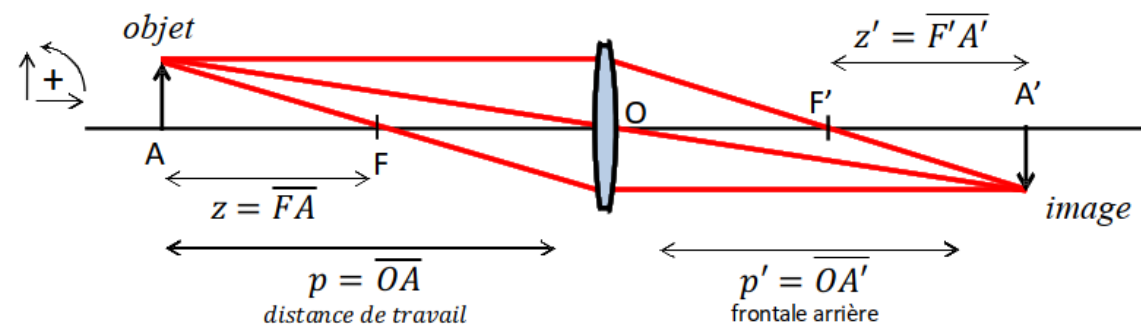
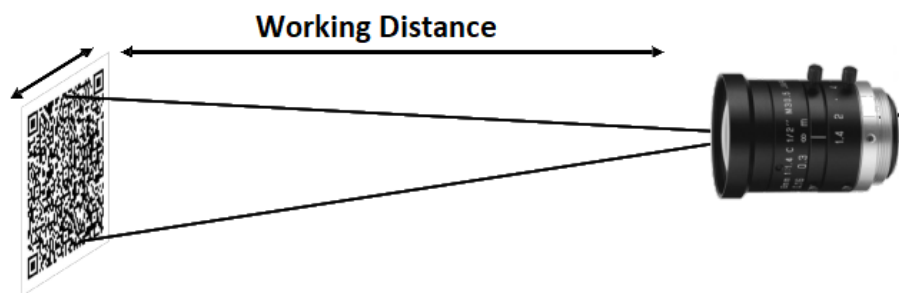
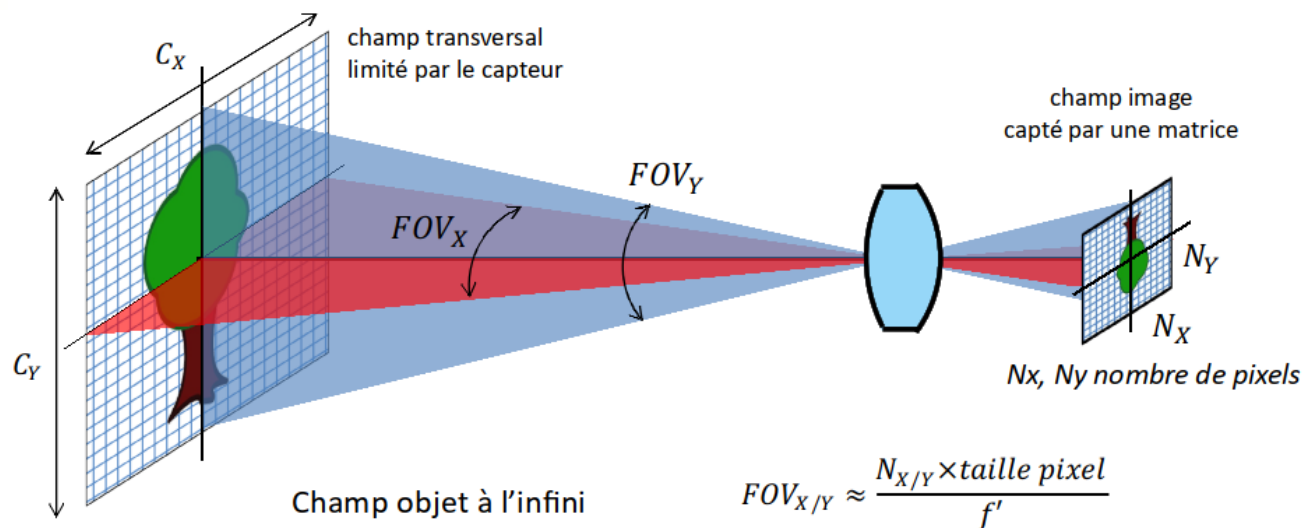
Créer une image





# Objectif optique

## Créer une image



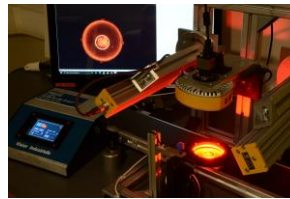
taille image

grandissement transversal

$$g_y = \frac{y'}{y} = \frac{p'}{p} = -\frac{z'}{f'} = -\frac{f}{z}$$

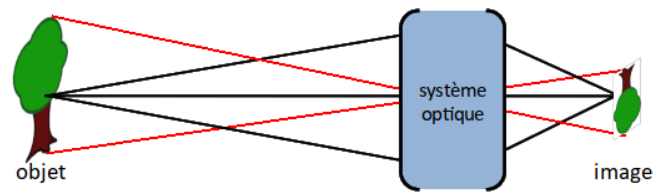
grandissement longitudinal

$$g_z = \frac{\delta p'}{\delta p} = (g_y)^2$$



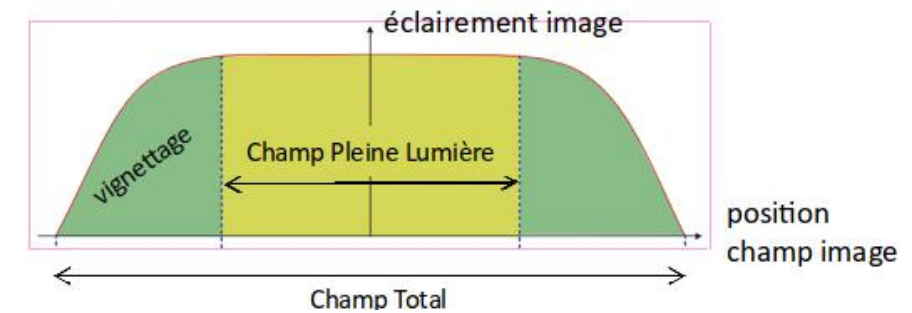
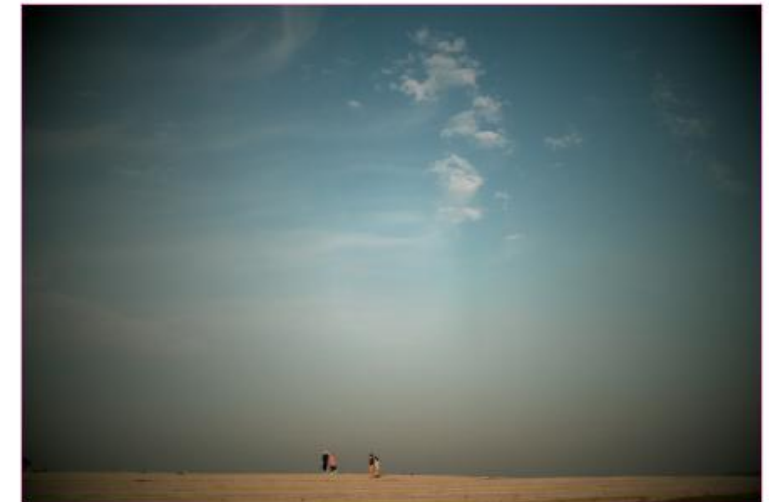
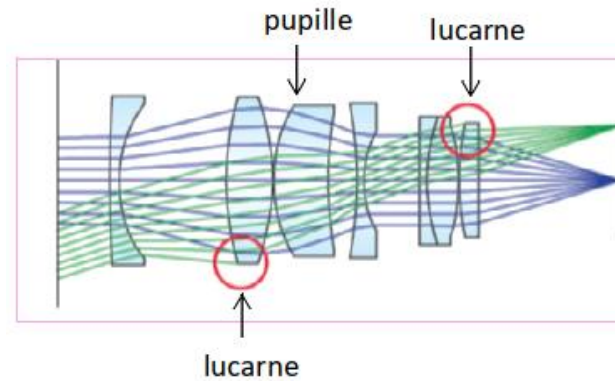
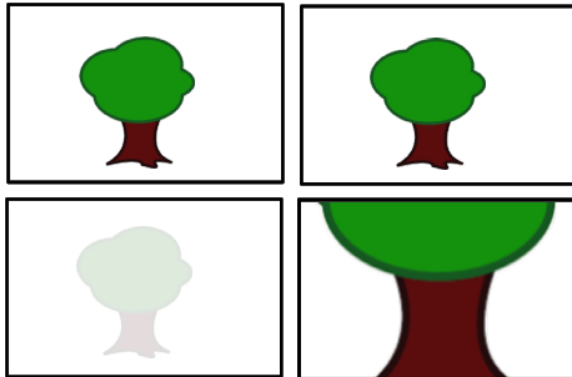
# Objectif optique

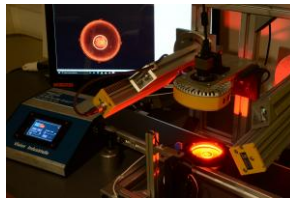
Créer une image



**Ouverture**  
flux collecté

**Champ**  
taille max objet capté





# Objectif optique

## Créer une image

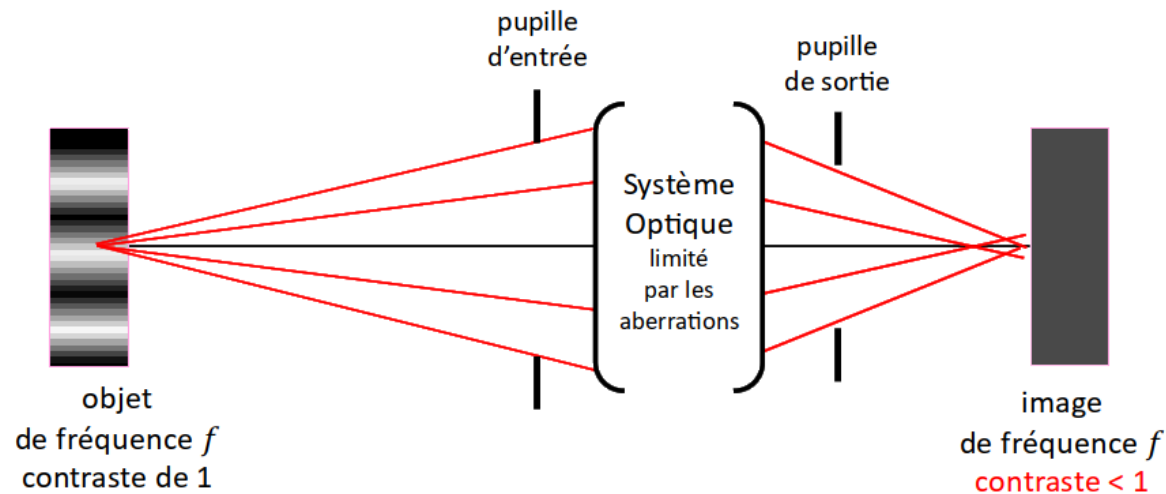
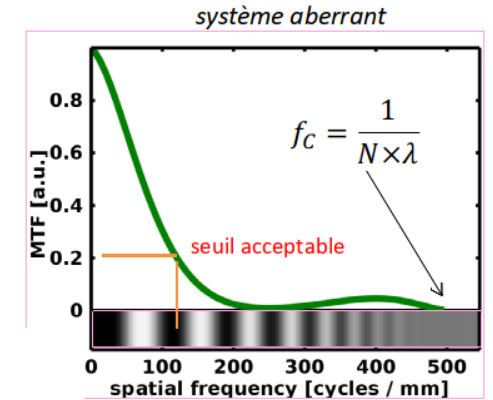
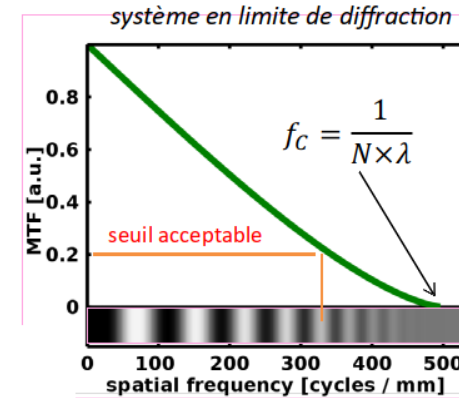
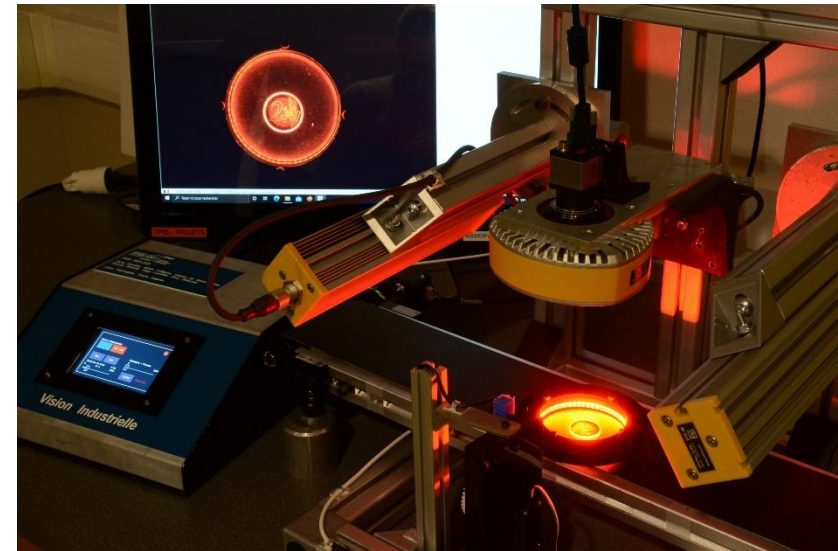


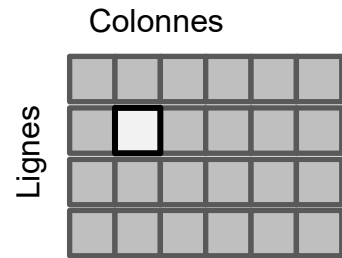
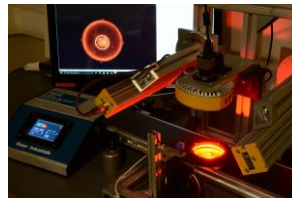
Image mire et profil

# Caméra numérique

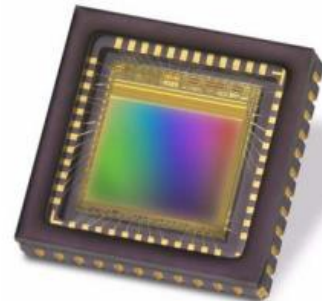
Echantillonnage / Quantification  
Colorimétrie







IDS UI-1240SE-C-HQ



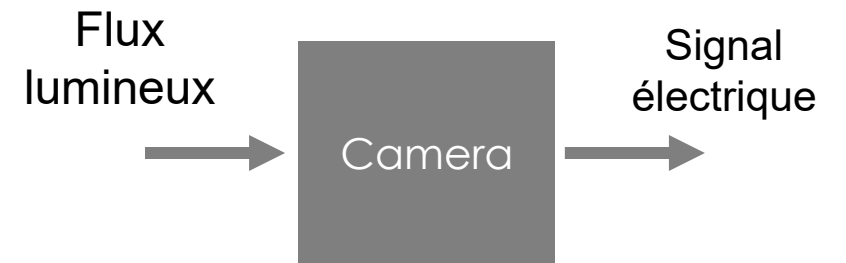
e2v sensor EV76C560ACT

# Caméra numérique

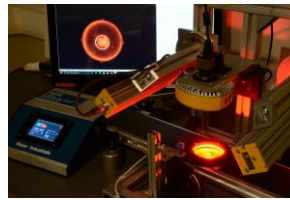
## Matrice de pixel

Camera

Système qui transforme un **flux lumineux** en un **signal électrique mesurable**

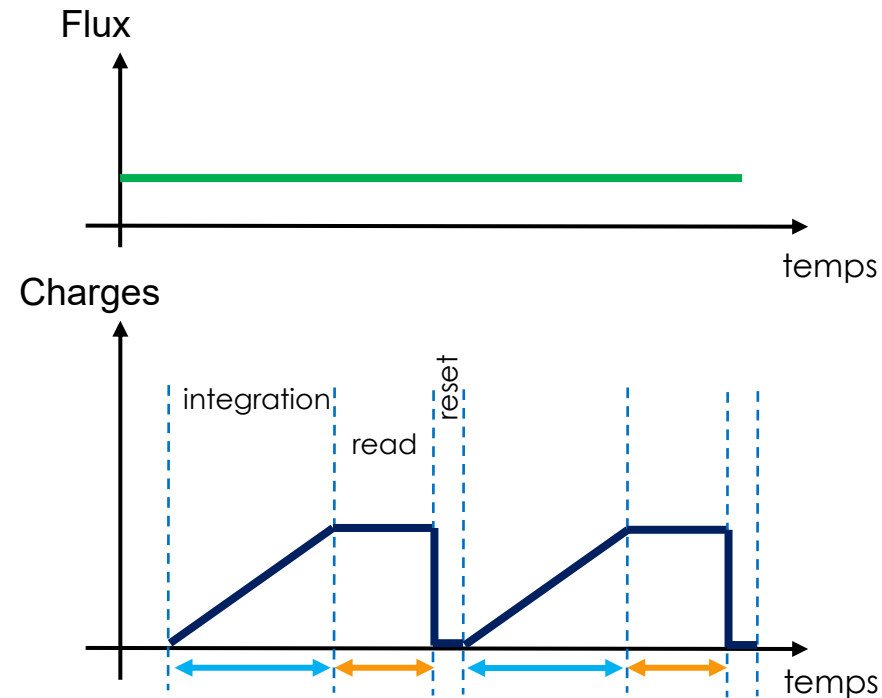
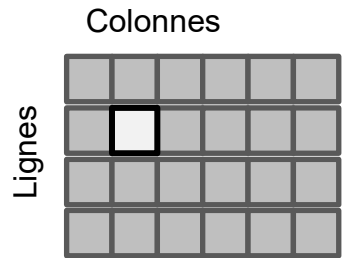


*Taille d'un pixel de l'ordre de 2 à 10  $\mu\text{m}$*



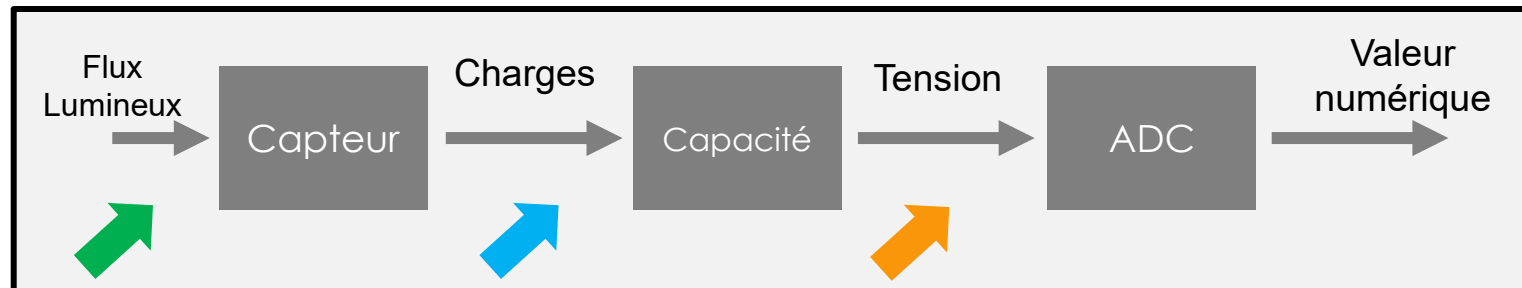
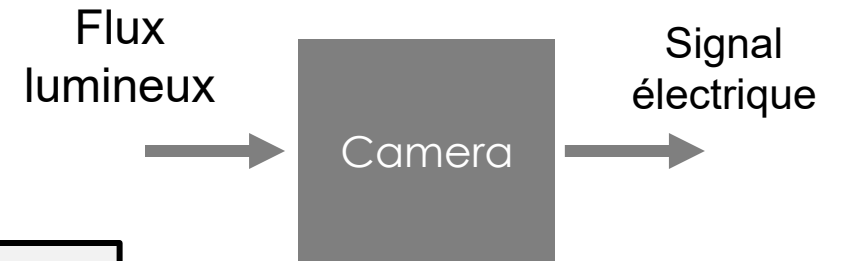
# Caméra numérique

## Matrice de pixel

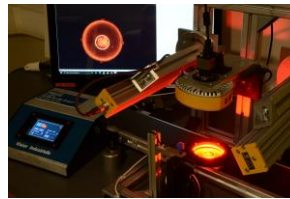


Camera

Système qui transforme un **flux lumineux** en un **signal électrique mesurable**

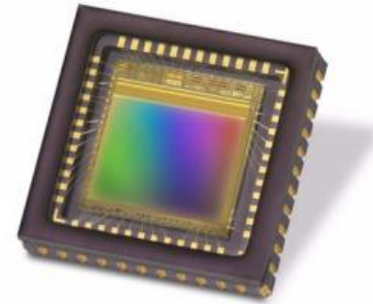


Taille d'un pixel de l'ordre de 2 à 10  $\mu\text{m}$

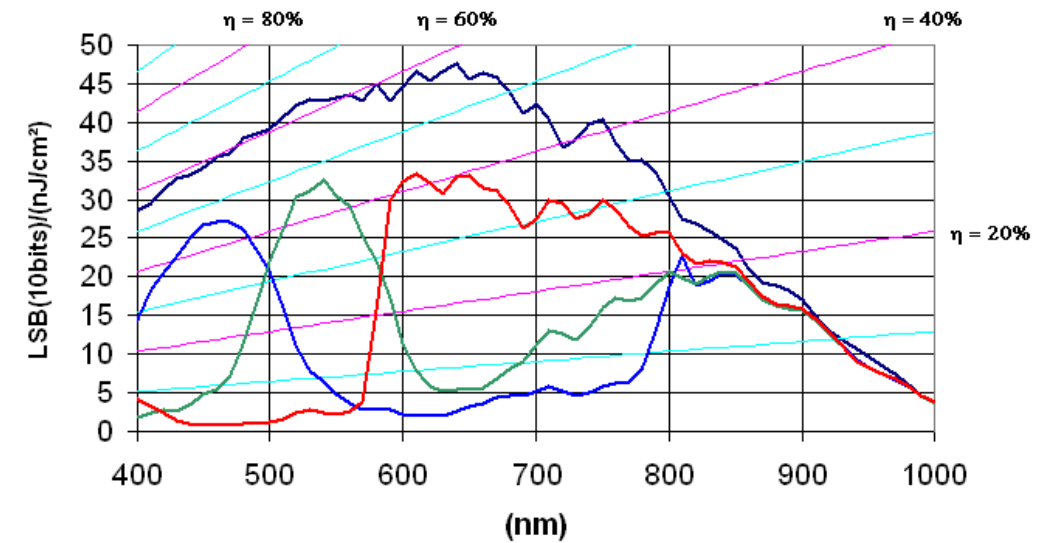
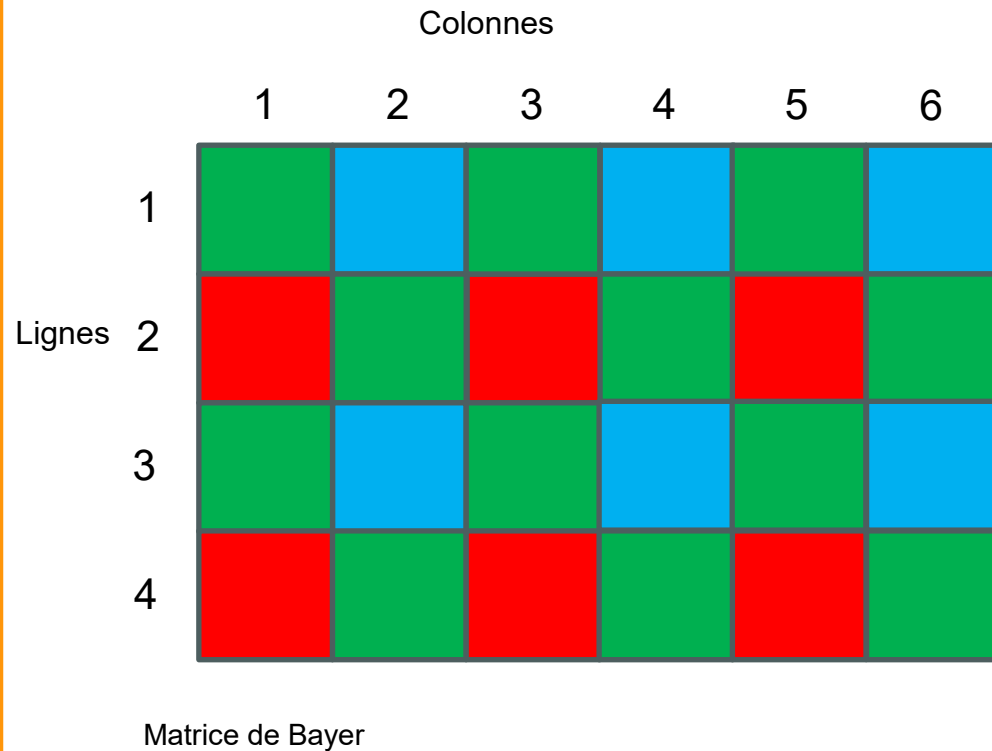


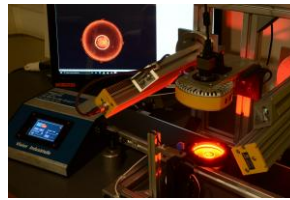
# Caméra numérique

## Réponse spectrale



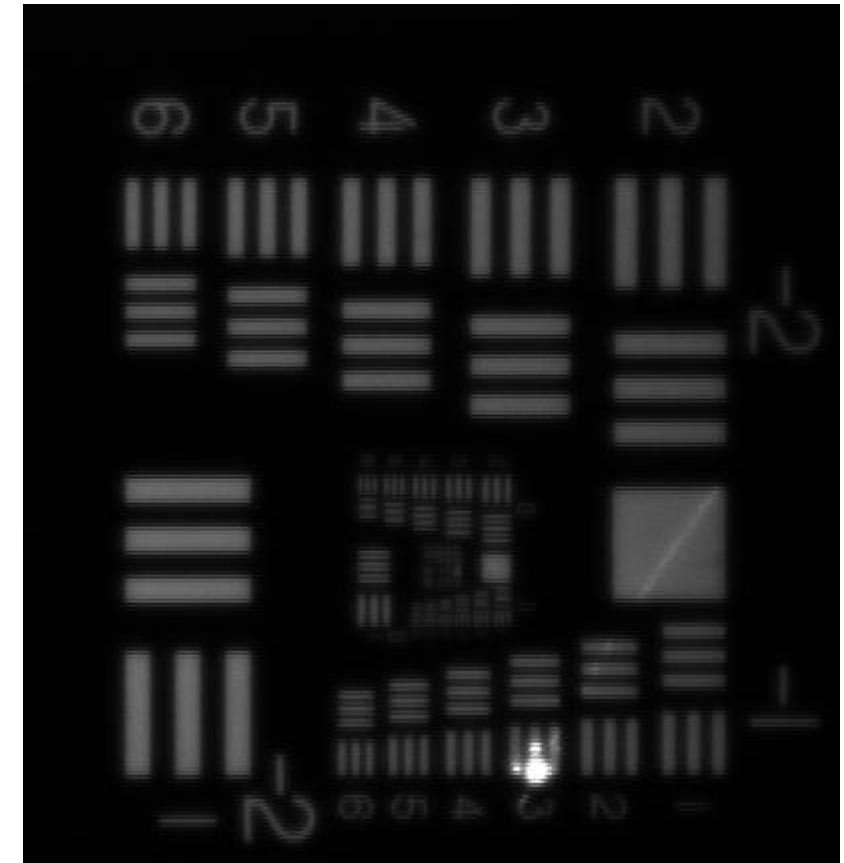
e2v sensor EV76C560ACT





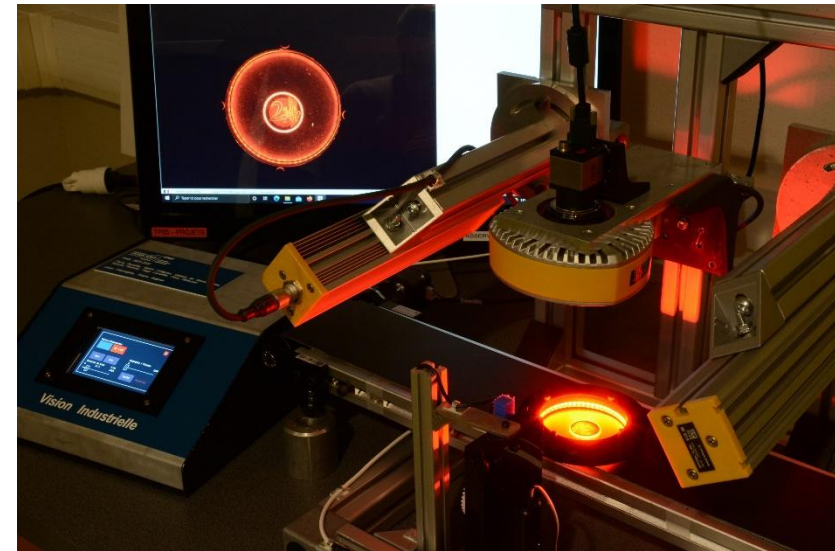
# Vision Industrielle

## Résolution

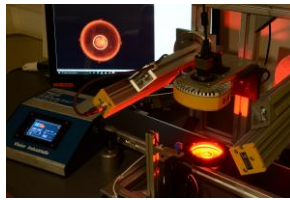


# Traitement d'image

Pré-traitement / Segmentation / Classification







# Traitement d'images

## Objectif

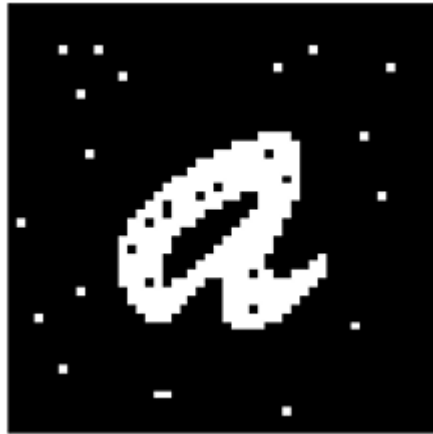


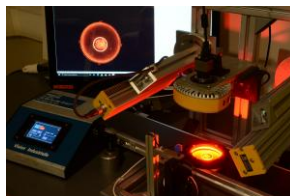
Image brute 'RAW' / Caméra

- **Bruitée**
- Mauvais contraste
- Eclairage non uniforme
- ...



Image souhaitée / Contours bien définis

- Zones homogènes
- Transitions nettes



# Traitement d'images

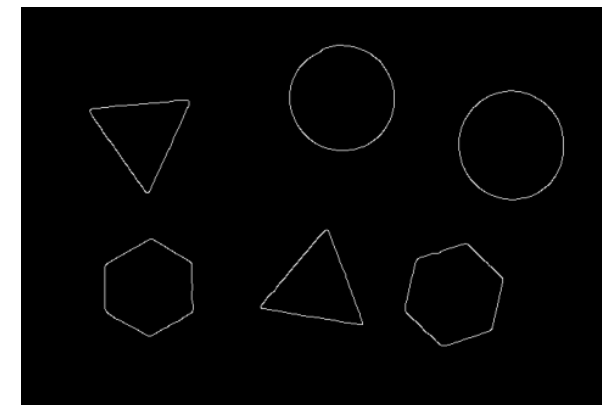
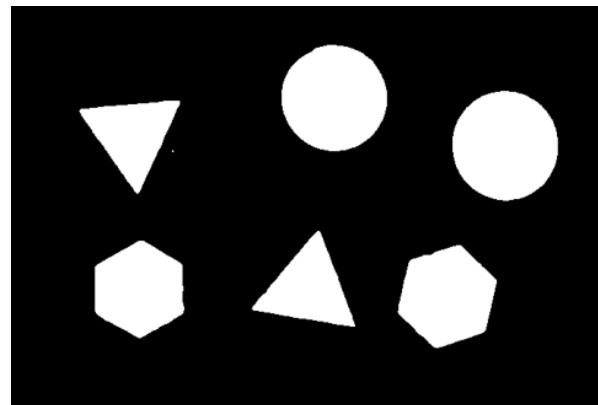
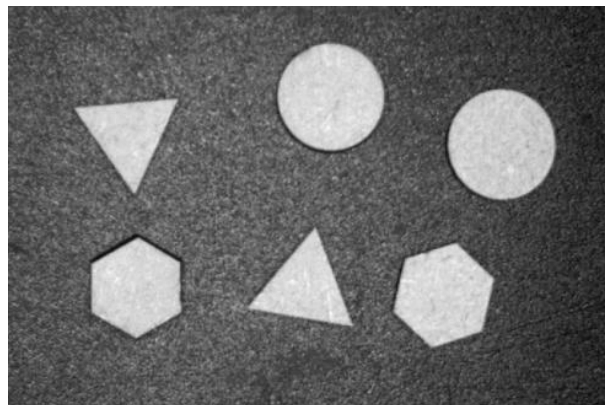


Image brute 'RAW' / Caméra

- **Bruitée**
- Mauvais contraste
- Eclairage non uniforme
- ...

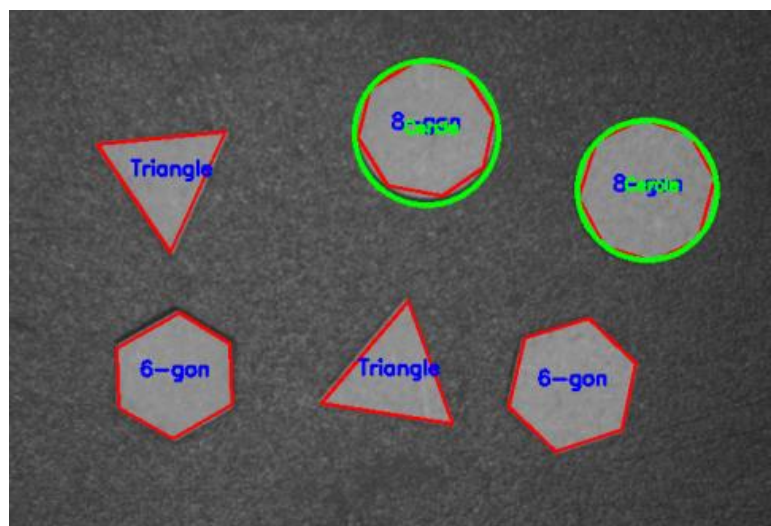
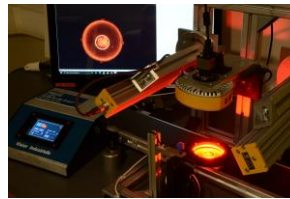


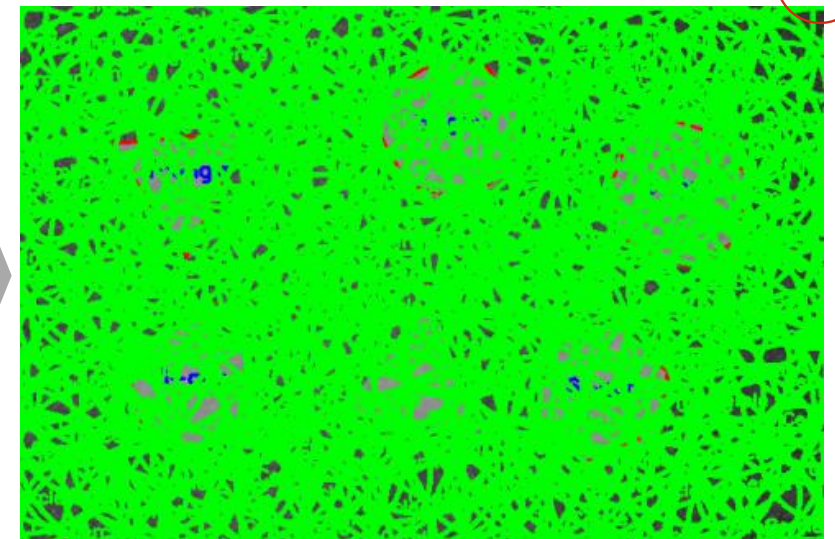
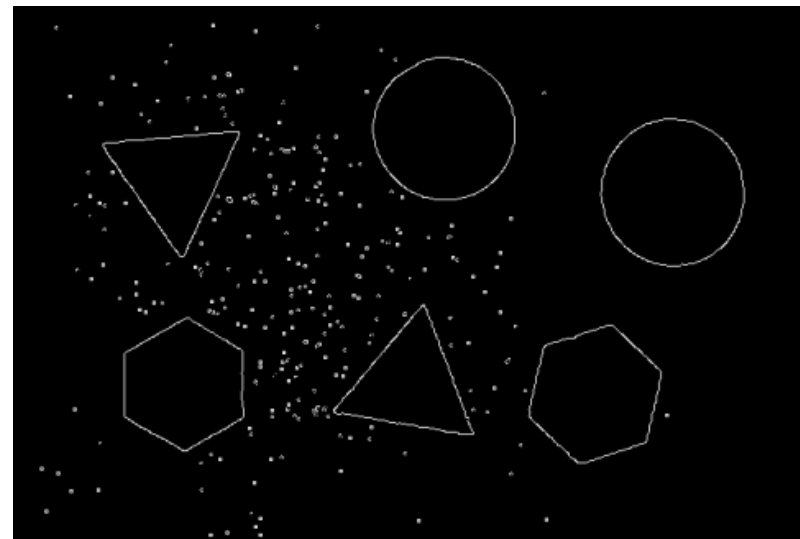
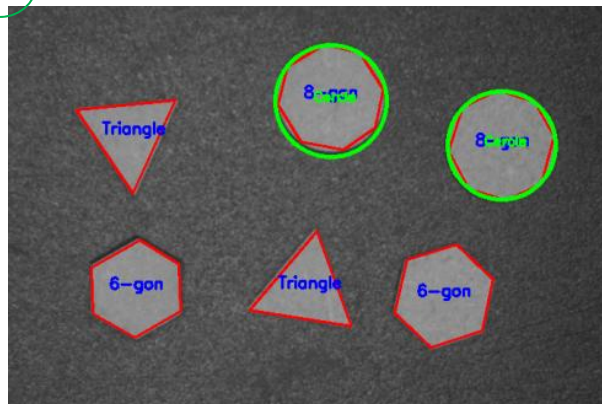
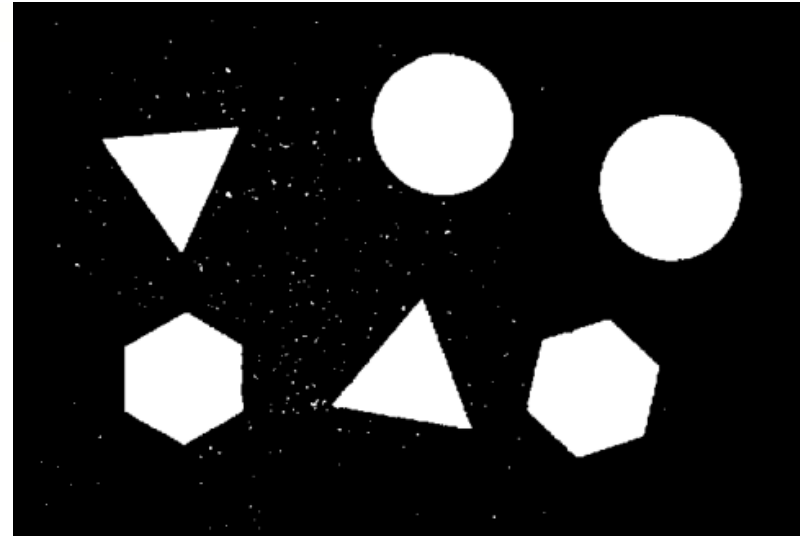
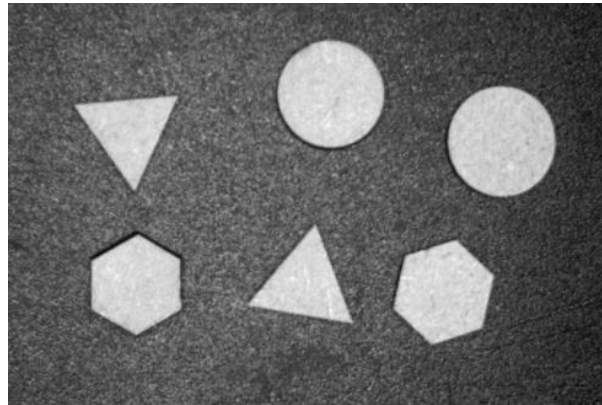
Image souhaitée / Contours bien définis

- Zones homogènes
- Transitions nettes

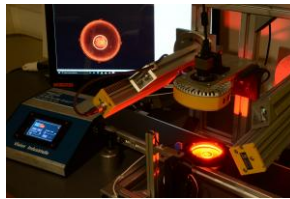


# Traitement d'images

Mauvais traitement

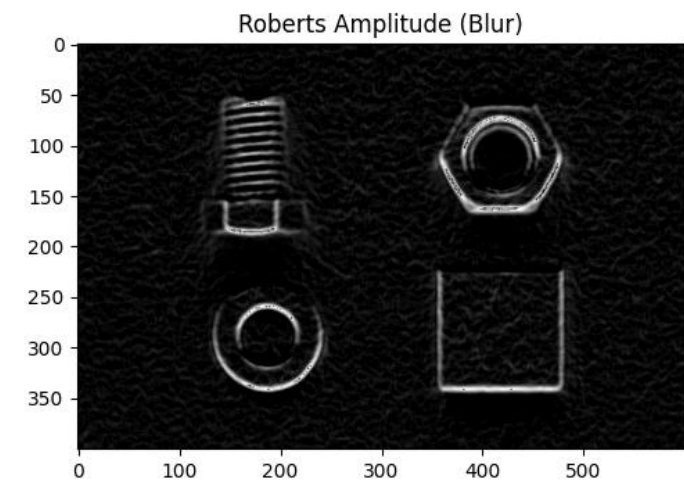
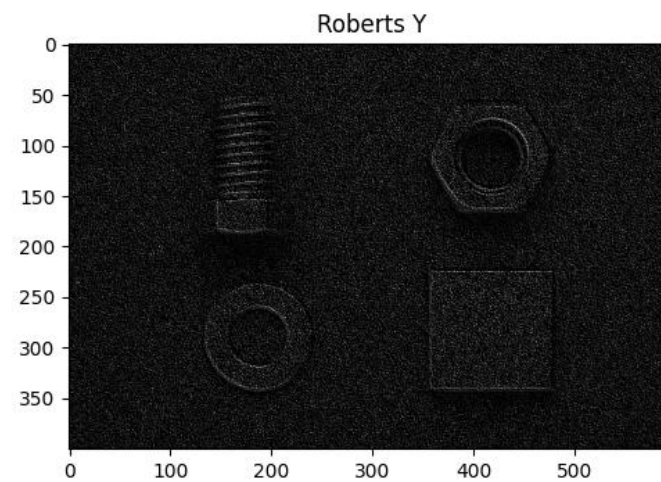
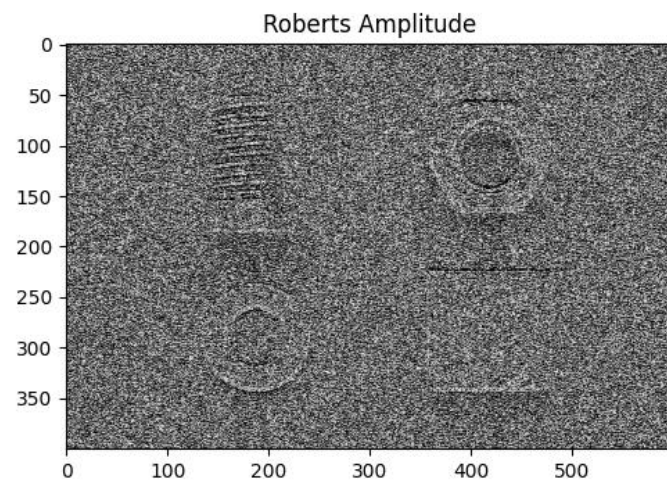
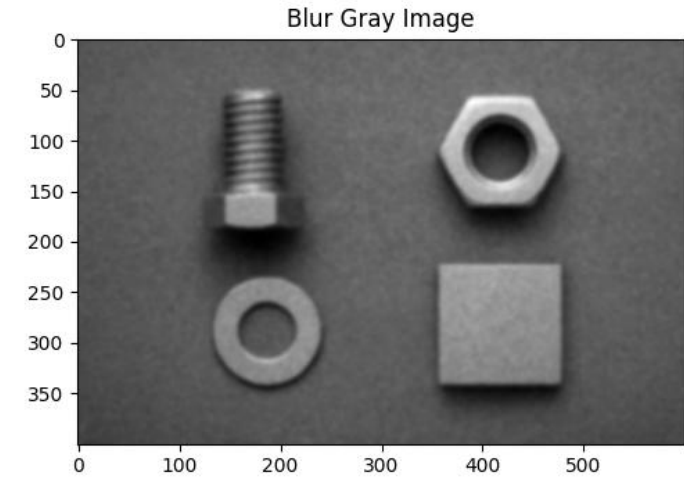
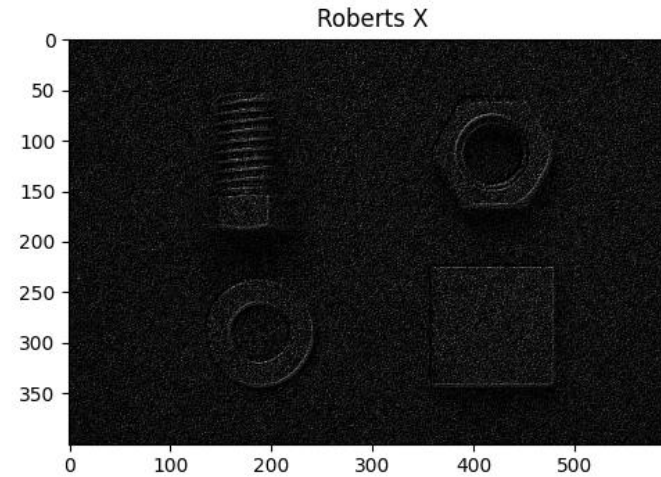
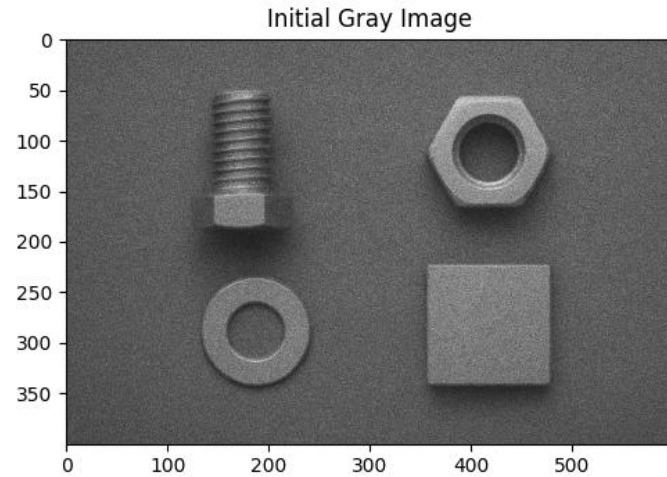


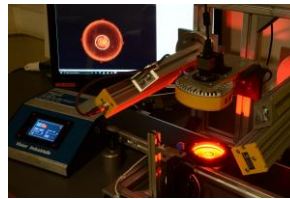




# Traitement d'images

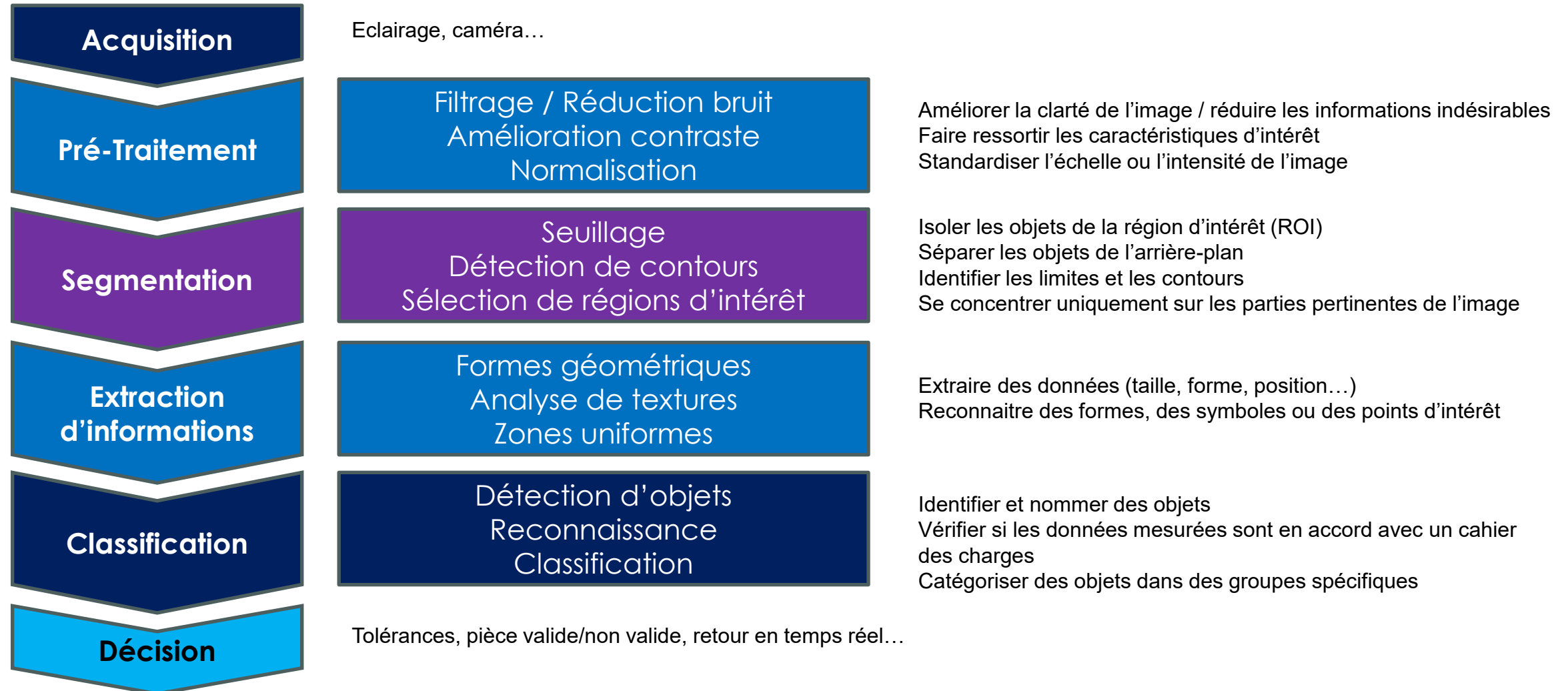
## Exemple industriel



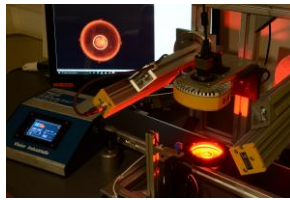


# Traitement d'images

## Objectif







# Traitement d'images

## Images numériques

### Image continue

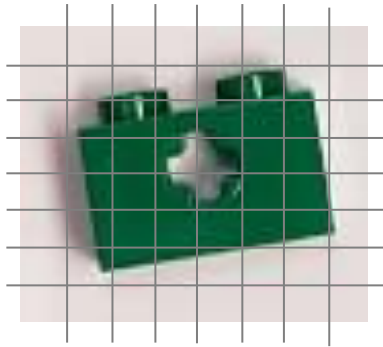
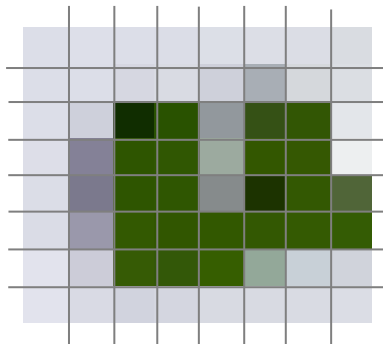


Image numérique : projection sur une matrice d'une image continue



8 x 8 grid



16 x 16 grid

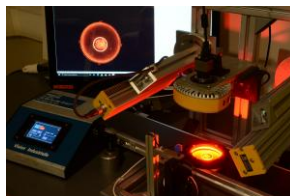


32 x 32 grid

### Image numérique

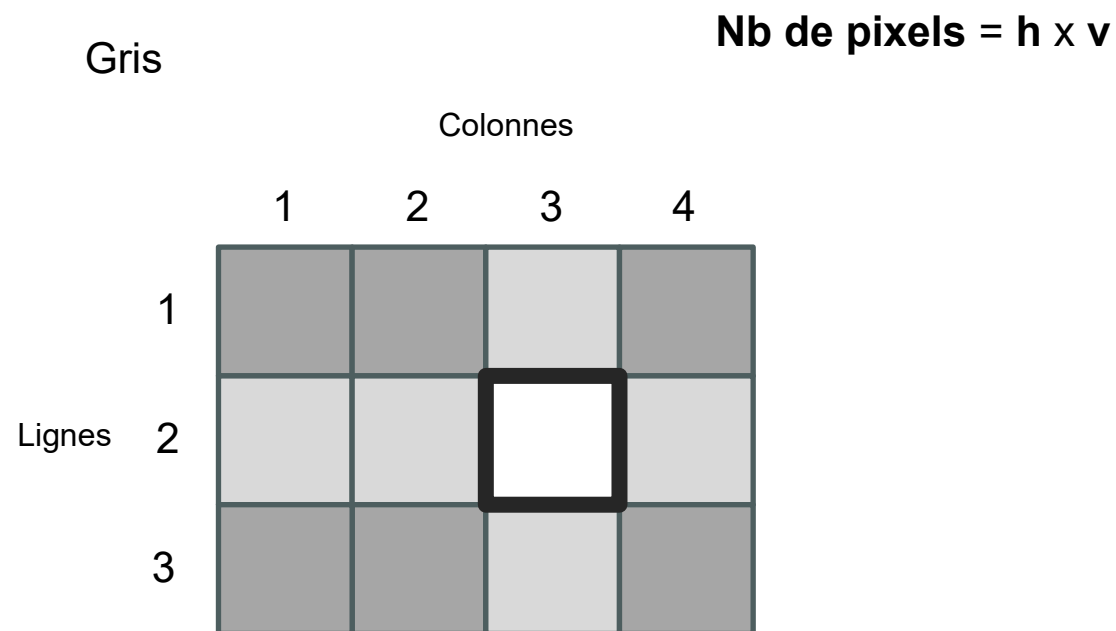
Représentation d'une **image**  
**sous forme numérique**

*Pour être **sauvegardée**, **traitée**  
et **affichée** par des ordinateurs  
ou des systems numériques.*

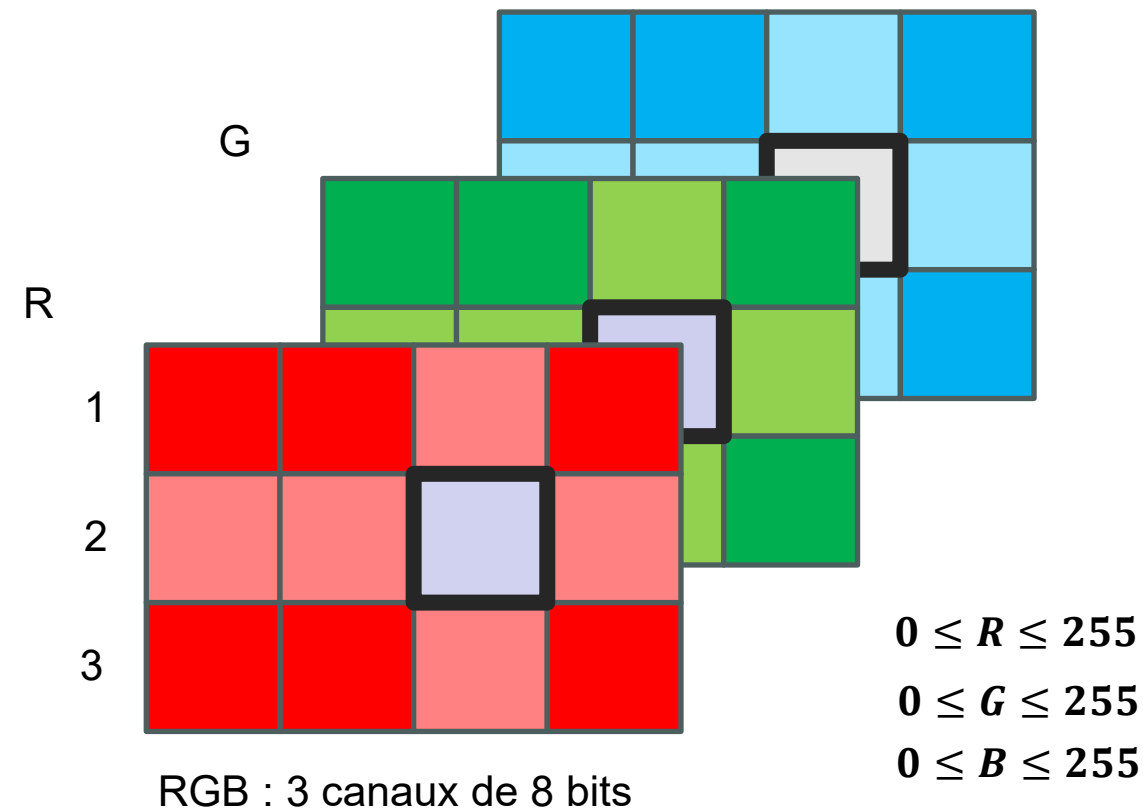


# Traitement d'images

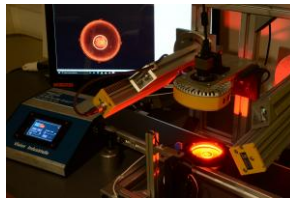
Images numériques / Gris ou RGB



Chaque pixel est converti sur **n bits**.



R=200, G=100, B=50



# Traitement d'images

OpenCV

## Open Source Computer Vision

Une bibliothèque de **traitement d'images**  
et de **Machine learning**

*Développés sur de multiple environnement,  
comme Python, C++, Java, and MATLAB*

Traitement d'images

*Filtrage, detection de contours, transformations...*

Reconnaissance

*Détection d'objets dans des images et des vidéos*

Algorithmes Vidéo

*Suivi de mouvement, Reconstruction 3D...*

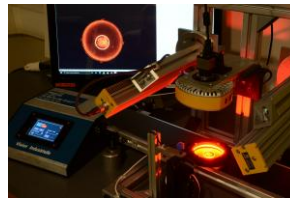
Machine Learning

*Classification d'images, Reconnaissance de formes*



<https://opencv.org>

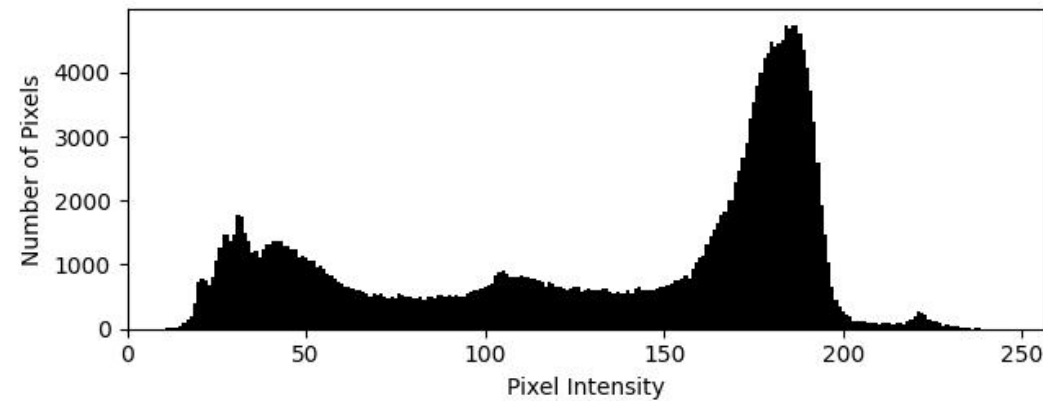
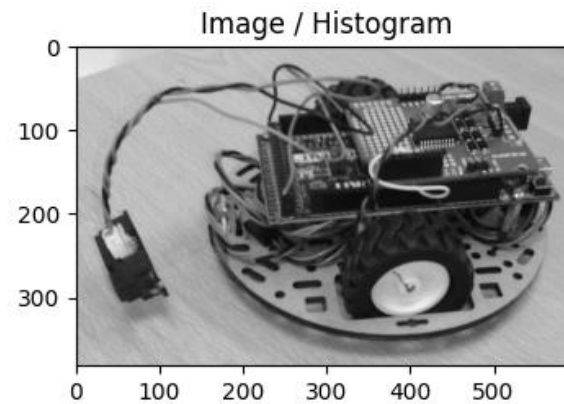




# Traitement d'images

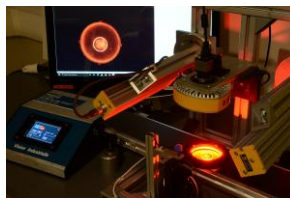
Filtrage par TF

## Acquisition



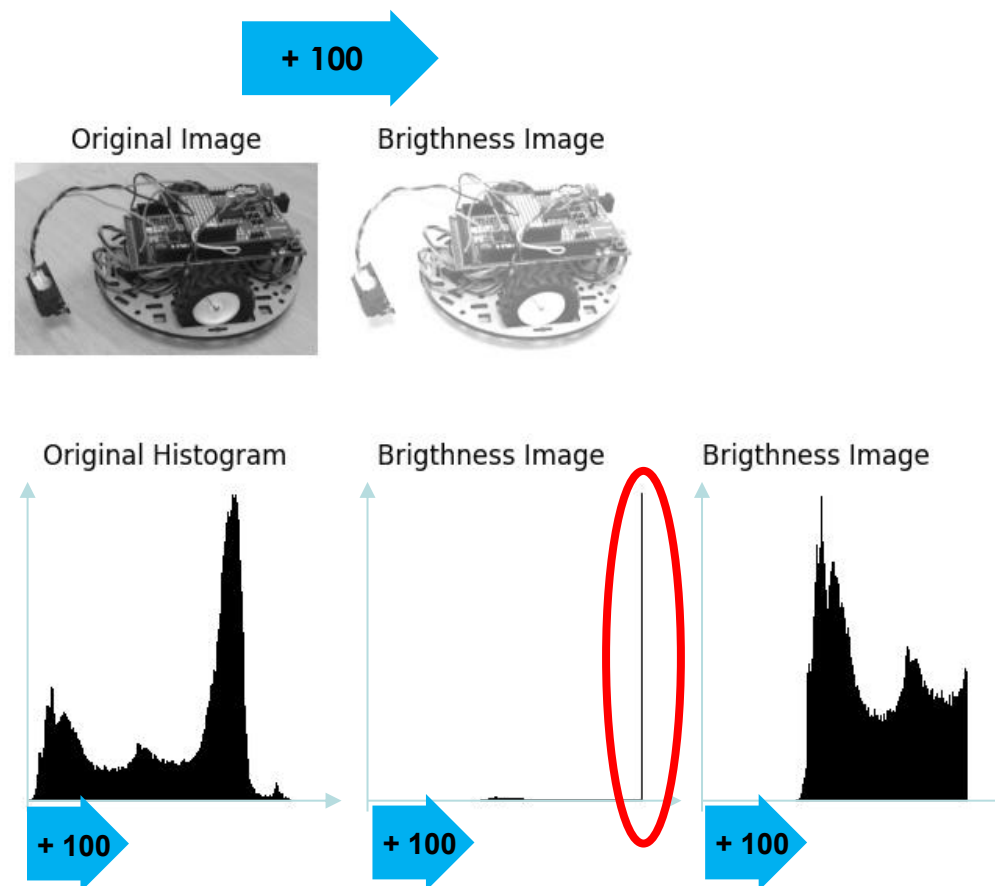
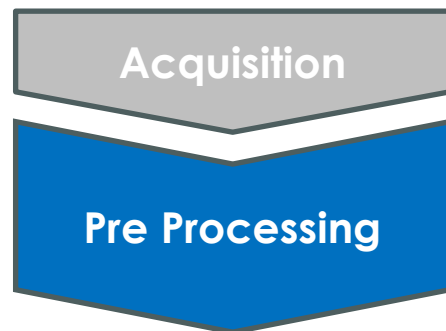
## Histogramme

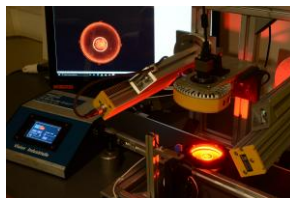
**Représentation graphique**  
montrant la **distribution des**  
**valeurs de niveaux de gris** des  
pixels de l'image



# Traitement d'images

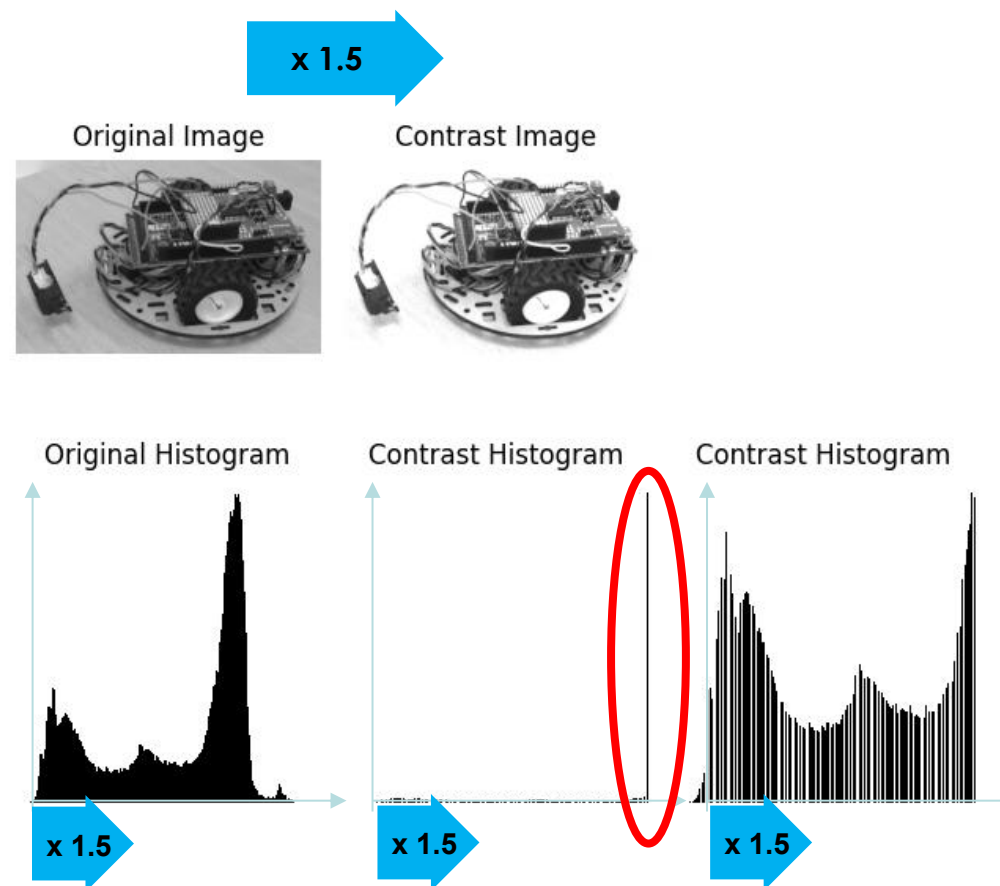
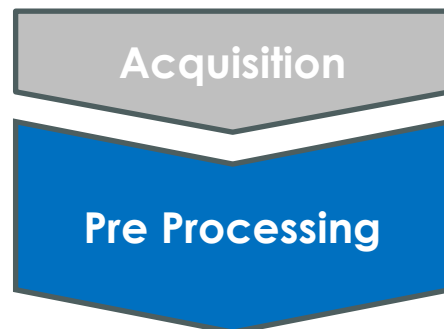
## Amélioration de l'image



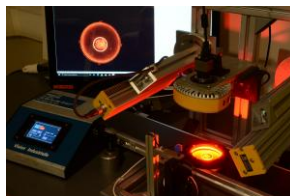


# Traitement d'images

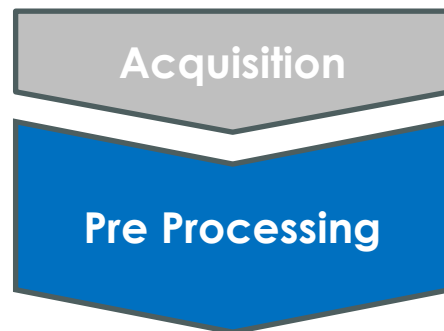
## Amélioration de l'image







# Traitement d'images



kernel

-1	0	-2
1	5	1
-2	0	-1

original image

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

Filtrage / Convolution

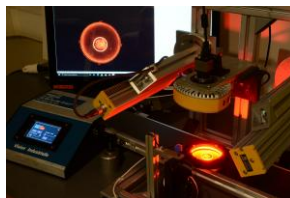
5	8	4	2	3	1	5
9	5	1	8	7	6	2
5	7	1	5	6	8	7
5	8	2	8	4	3	3
5	6	6	7	2	5	1

filtered image

				4		

$$R = -8 + 0 - 12 + 5 + 30 + 8 - 16 + 0 - 3$$

$$R = 4$$



# Traitement d'images

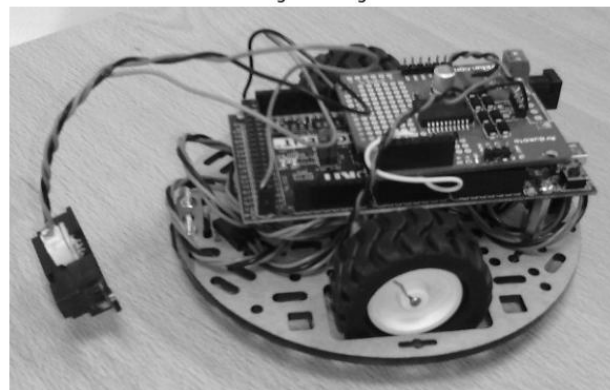
## Filtrage / Convolution

Suppression de détails insignifiants

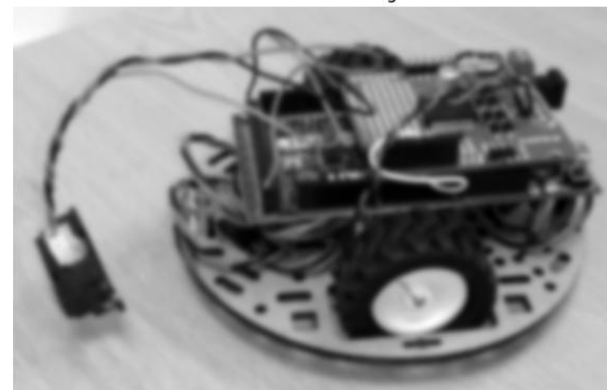
Acquisition

Pre Processing

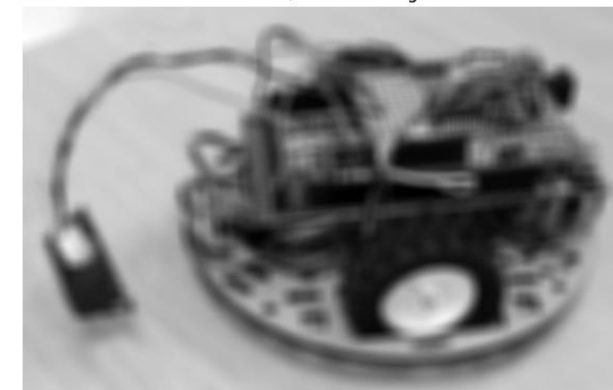
Original Image



Gaussian Blur Image



Median/Box Blur Image

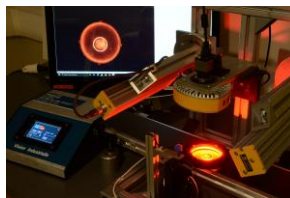


1	4	7	4	1
4	16	26	16	4
7	26	41	26	7
4	16	26	16	4
1	4	7	4	1

**Gaussian Kernel**  
(x 1/273)

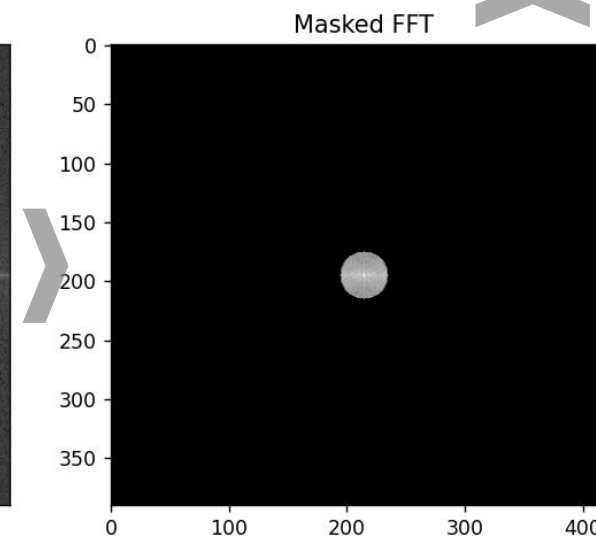
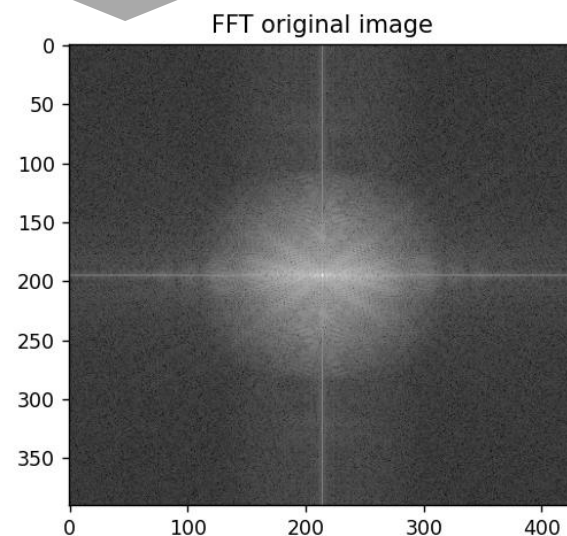
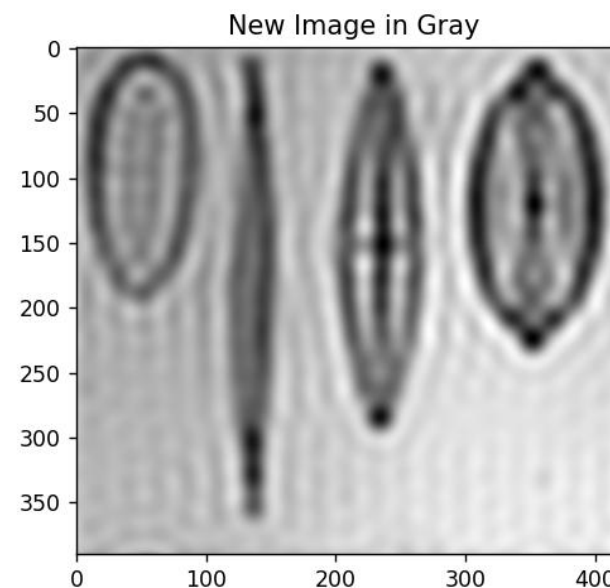
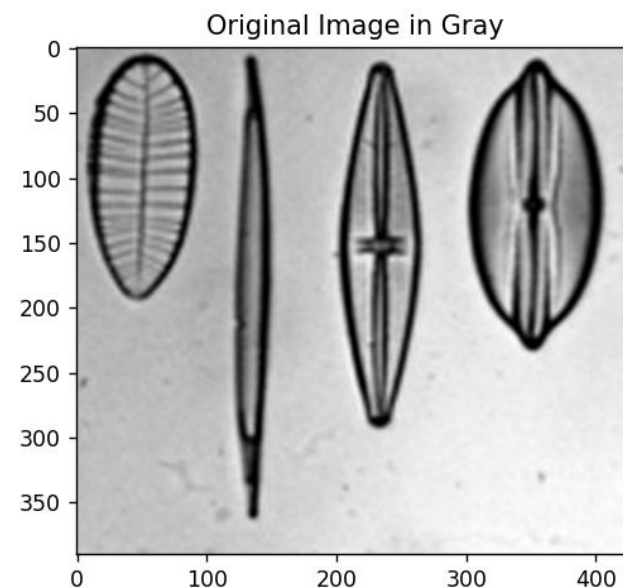
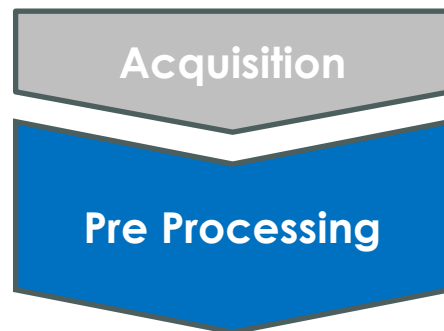
**Mean Kernel** ( x 1/(N\*M) )

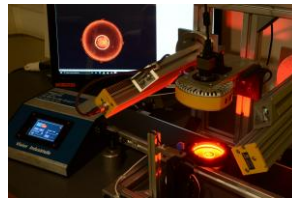
1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9



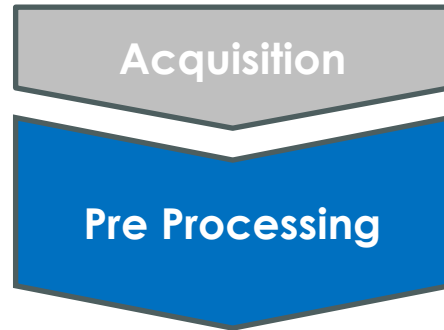
# Traitement d'images

Filtrage par TF







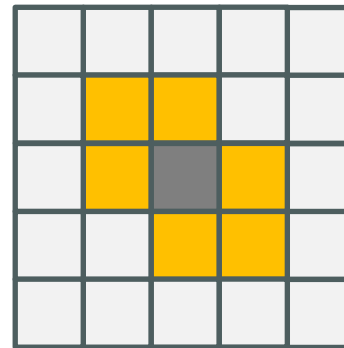
# Traitement d'images



kernel

0	1	0
1	1	1
0	1	0

 Pixels originaux  
 Pixels retirés

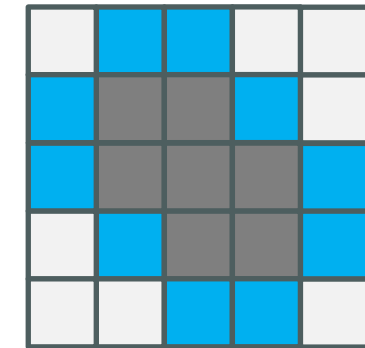


Erosion

**Réduire le premier plan** en retirant progressivement les pixels le long des contours des objets

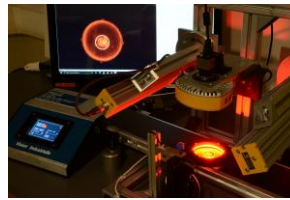
Erosion / Dilatation

 Pixels ajoutés



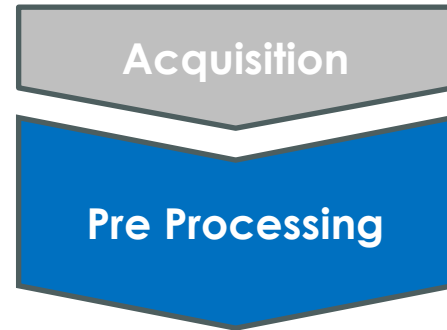
Dilatation

**Étendre le premier plan** en ajoutant des pixels le long des contours des objets



# Traitement d'images

## Erosion / Dilatation



Eroded Image



Original Image



Dilated Image



kernel

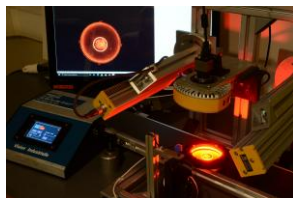
0	1	0
1	1	1
0	1	0

### Erosion

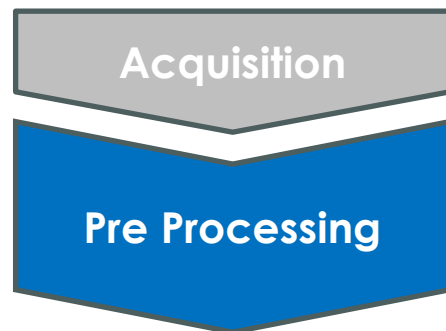
**Réduire le premier plan** en retirant progressivement les pixels le long des contours des objets

### Dilatation

**Étendre le premier plan** en ajoutant des pixels le long des contours des objets



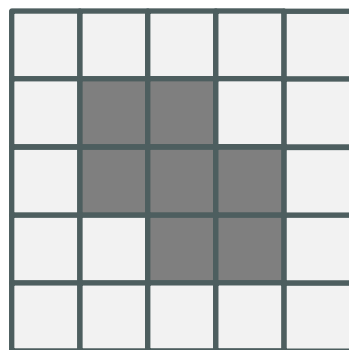
# Traitement d'images



kernel

0	1	0
1	1	1
0	1	0

Original pixels  
Removed pixels

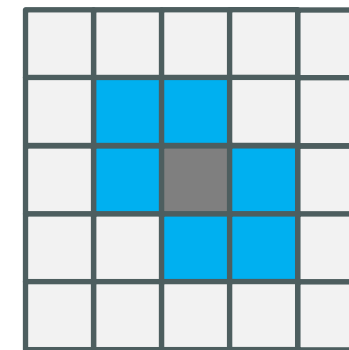
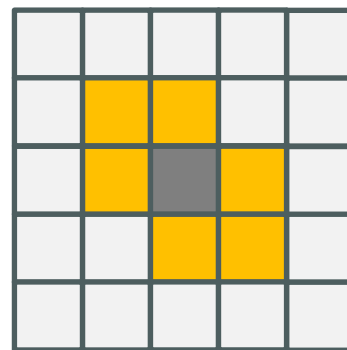


Ouverture

**Erosion** puis **Dilatation**

Retire des petits objets

Added pixels

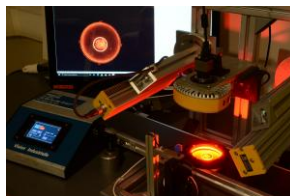


Fermeture

**Dilatation** puis **Erosion**

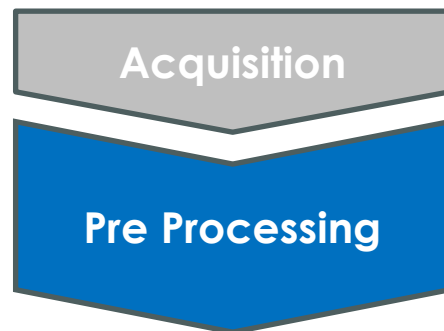
Remplit des petites zones





# Traitement d'images

## Ouverture / Fermeture



Opening Image



Original Image



Closing Image



kernel

0	1	0
1	1	1
0	1	0

Ouverture

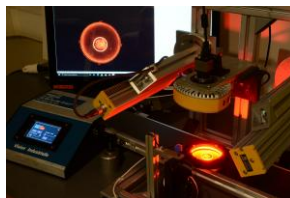
**Erosion** puis **Dilatation**

Retire des petits objets

Fermeture

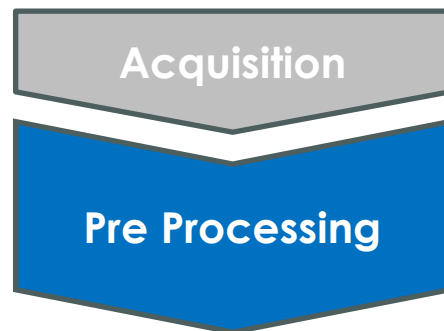
**Dilatation** puis **Erosion**

Remplit des petites zones



# Traitement d'images

## Ouverture / Fermeture



Opening Image



Original Image



Closing Image



kernel

0	1	0
1	1	1
0	1	0

Ouverture

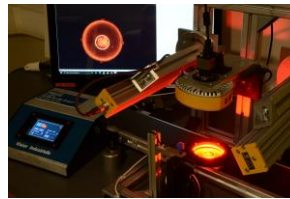
**Erosion** puis **Dilatation**

Retire des petits objets

Fermeture

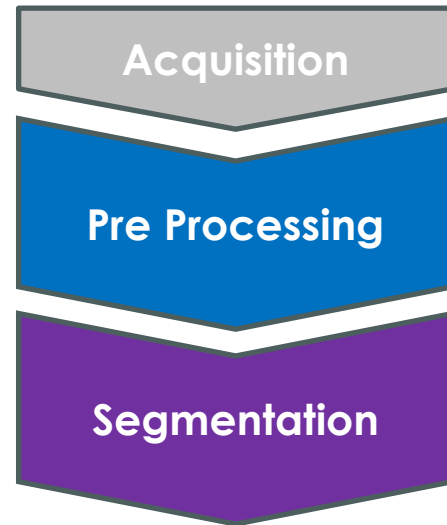
**Dilatation** puis **Erosion**

Remplit des petites zones



# Traitement d'images

Gradient



kernel

0	1	0
1	1	1
0	1	0

Original Image



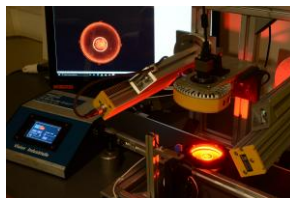
Gradient Image



Gradient

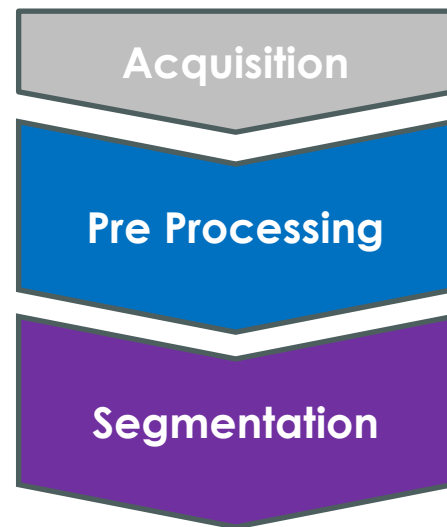
**Difference** entre une **dilatation** et une **érosion**

*Classification des pixels : **scène** (background) ou **objets** (foreground) ?*



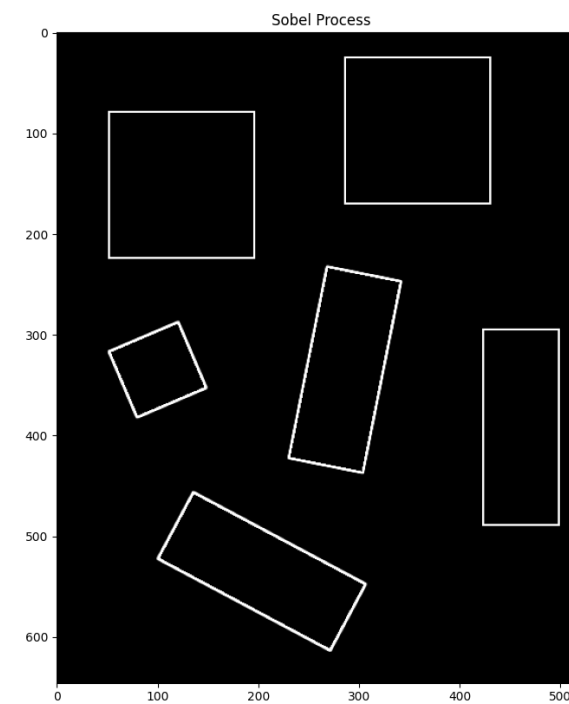
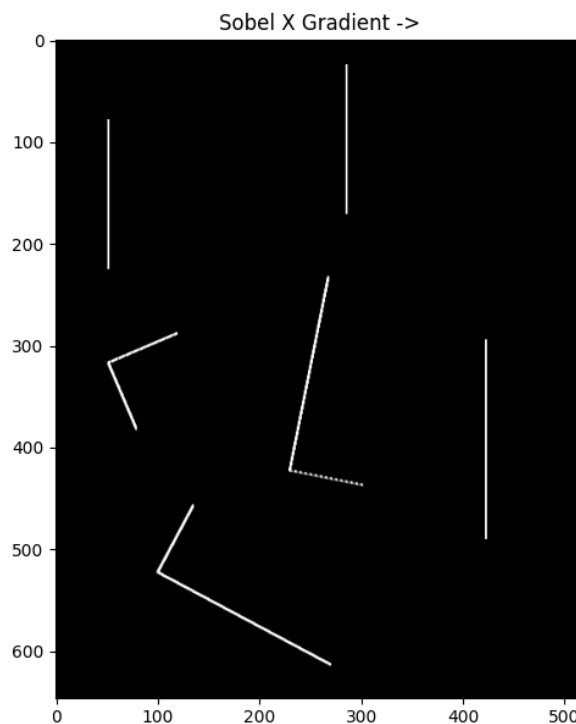
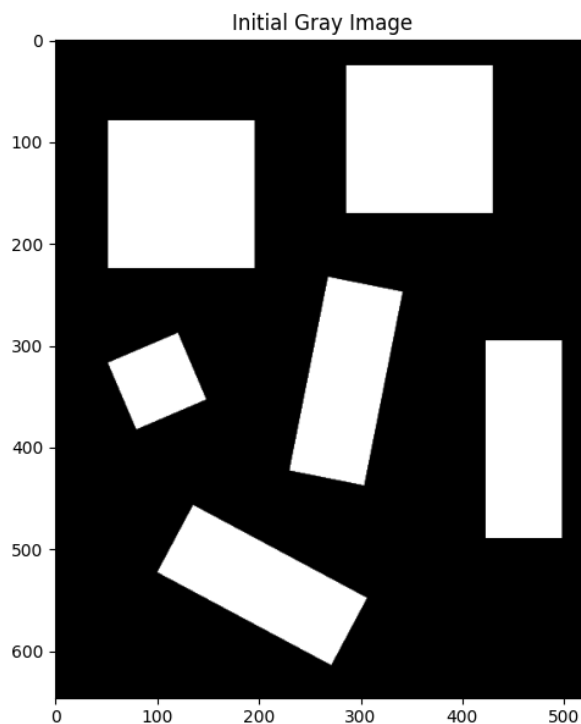
# Traitement d'images

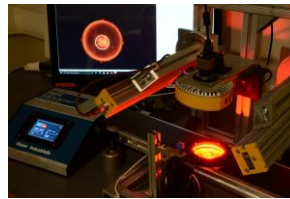
## Opérateur de Sobel



kernel

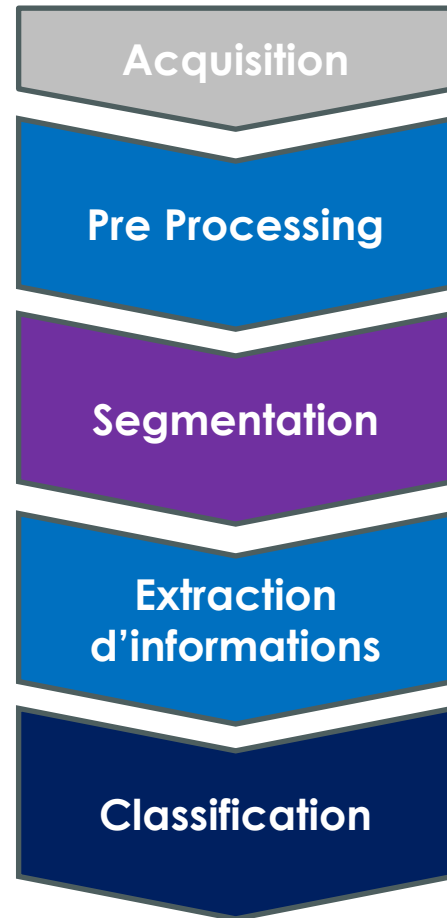
-1	0	1
-2	0	2
-1	0	1

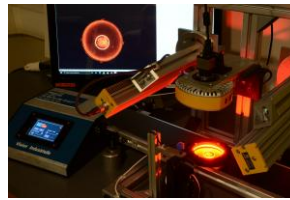




# Traitement d'images

## Méthode de Watershed





# Traitement d'images

## Méthode de Watershed

