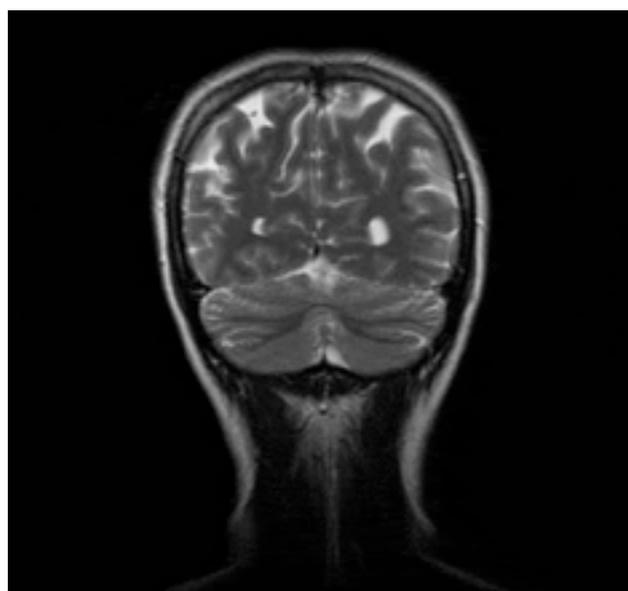


# Les mathématiques et le traitement de l'image



**Kévin Polisano**

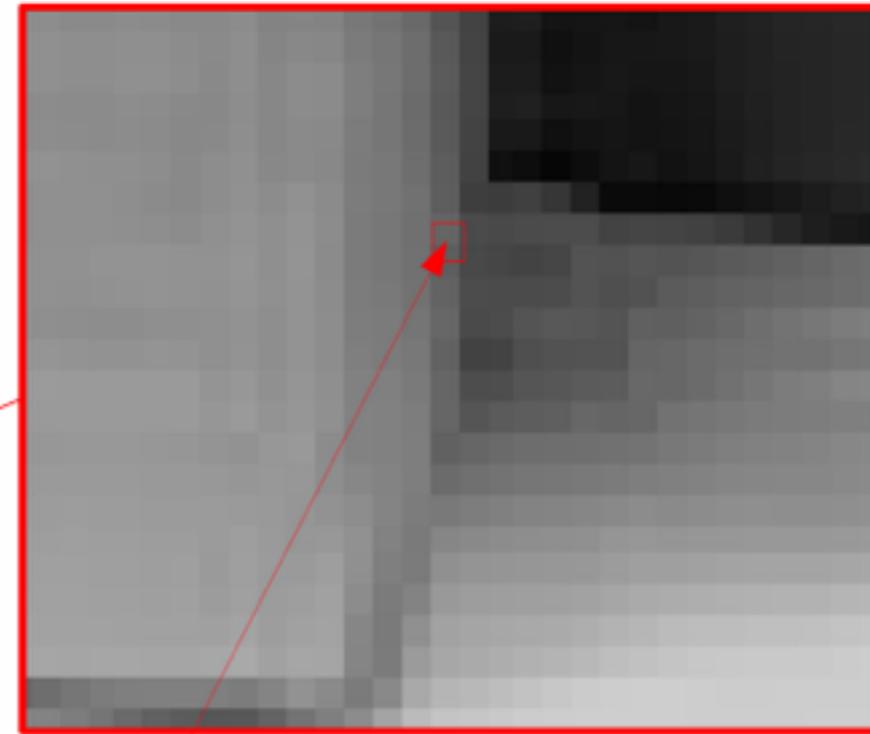
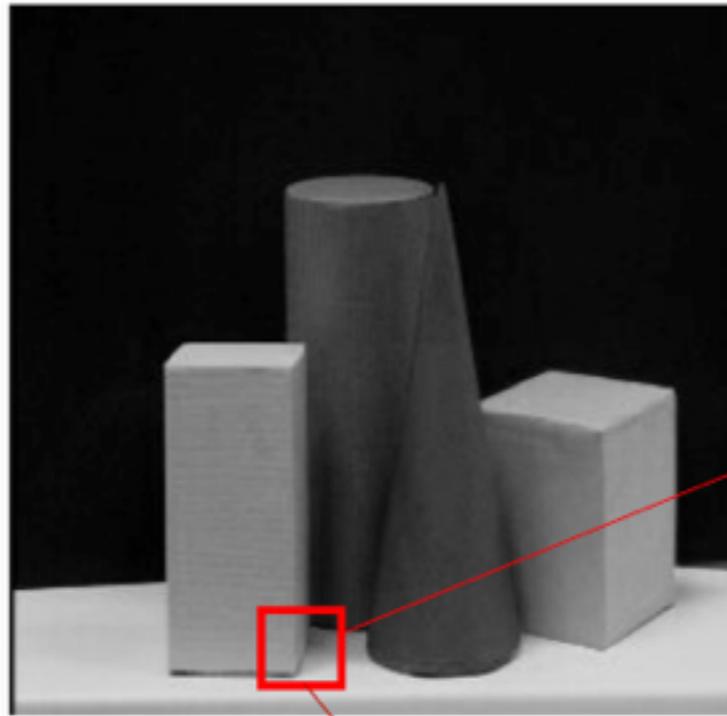
[ Doctorant au Laboratoire  
Jean Kuntzmann et CNRS ]



**N'hésitez pas à lever la main  
pour m'interrompre !**

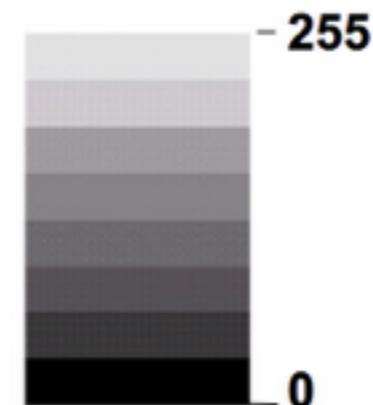


# Qu'est-ce qu'une image numérique ?

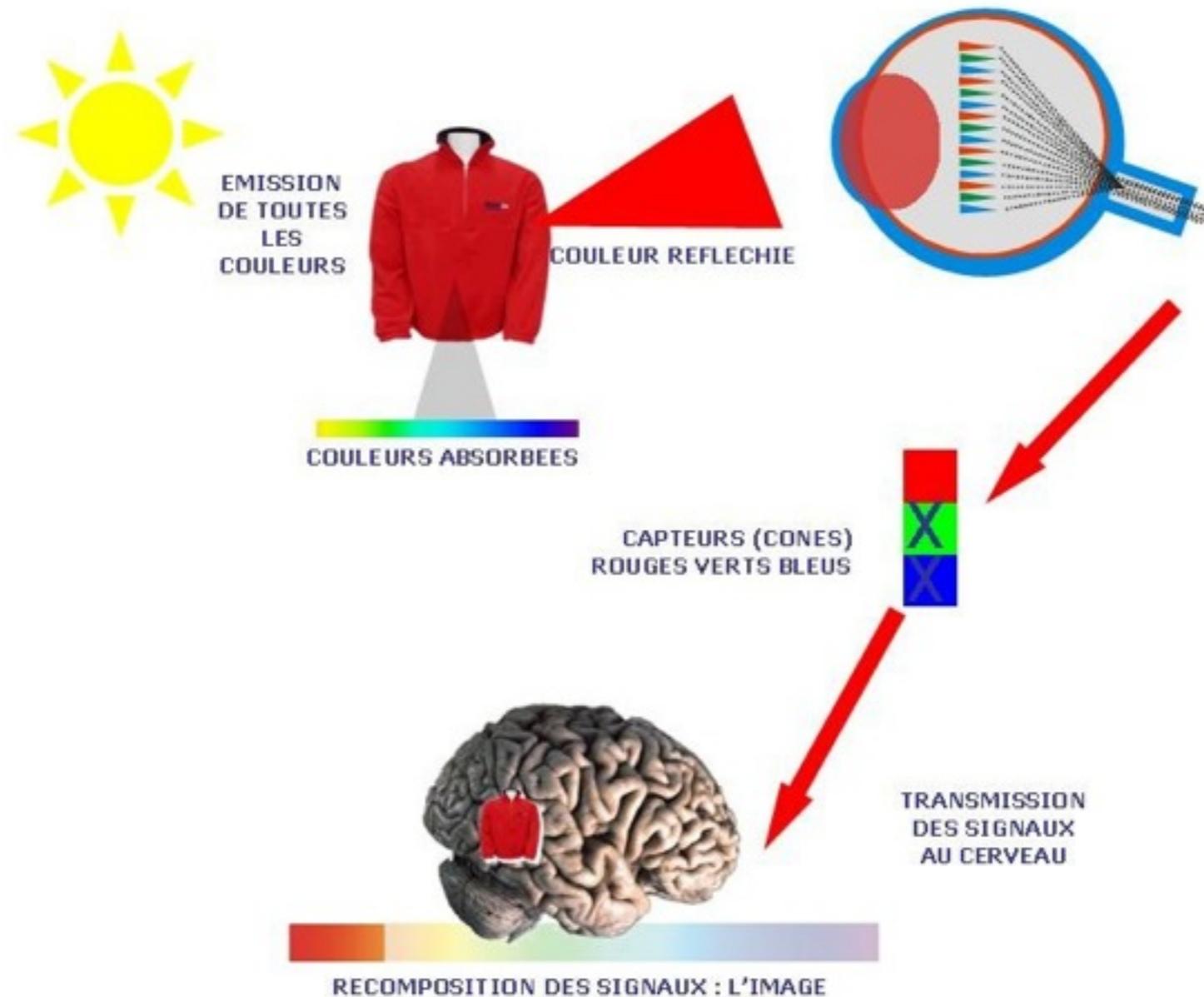


**Pixel (PICTure ELement)**

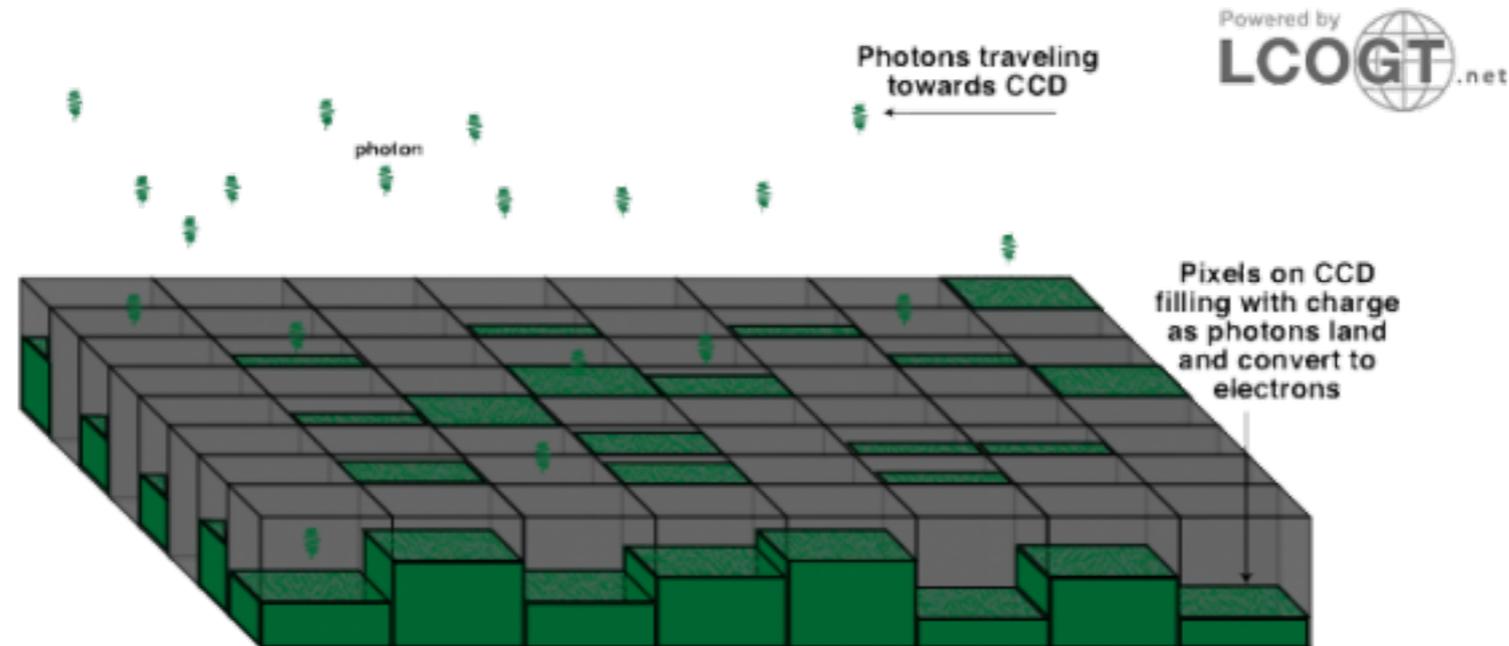
123	119	113	94	73	70	75	76	67
124	121	114	99	76	75	67	70	76
125	123	117	103	74	76	75	76	83
127	124	120	98	66	67	84	88	87
129	128	122	102	77	78	79	82	83
130	129	125	102	85	91	85	87	91
123	119	113	94	73	70	71	73	67
127	121	114	99	76	75	67	73	75
130	123	117	103	74	76	77	75	81
127	124	120	98	66	67	84	88	85
125	128	122	102	77	78	89	87	87



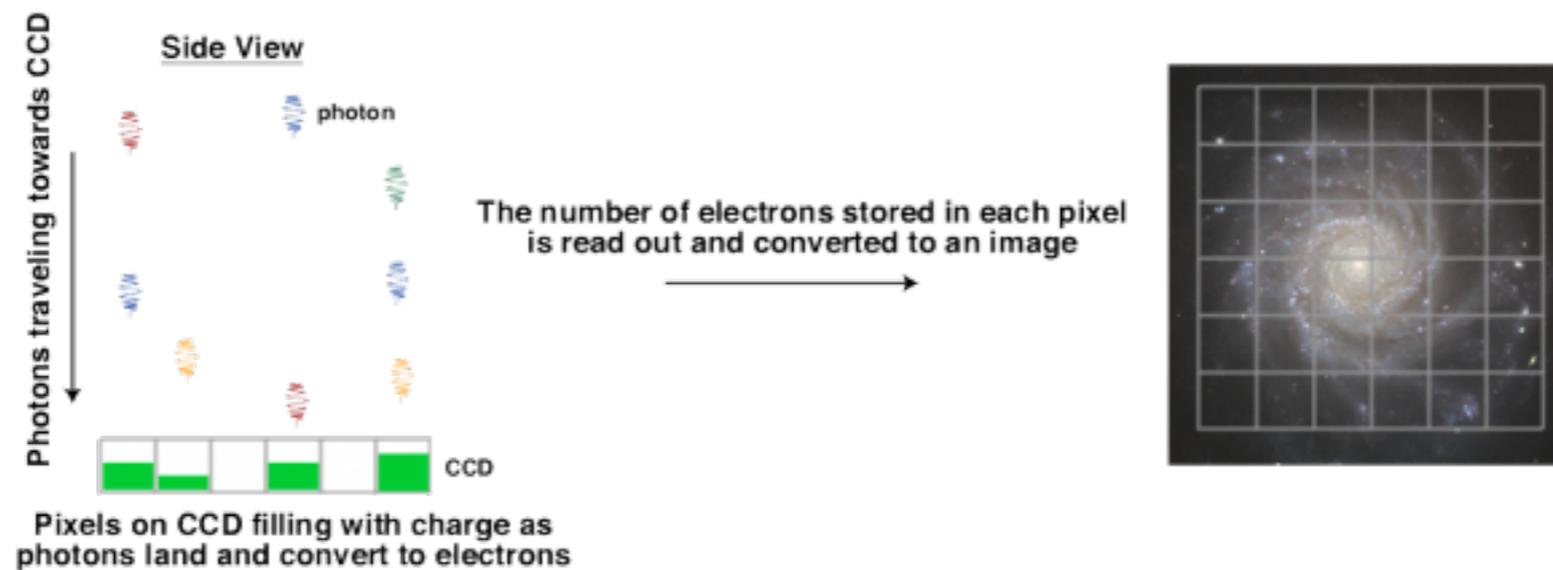
# Comment l'oeil perçoit une image ?



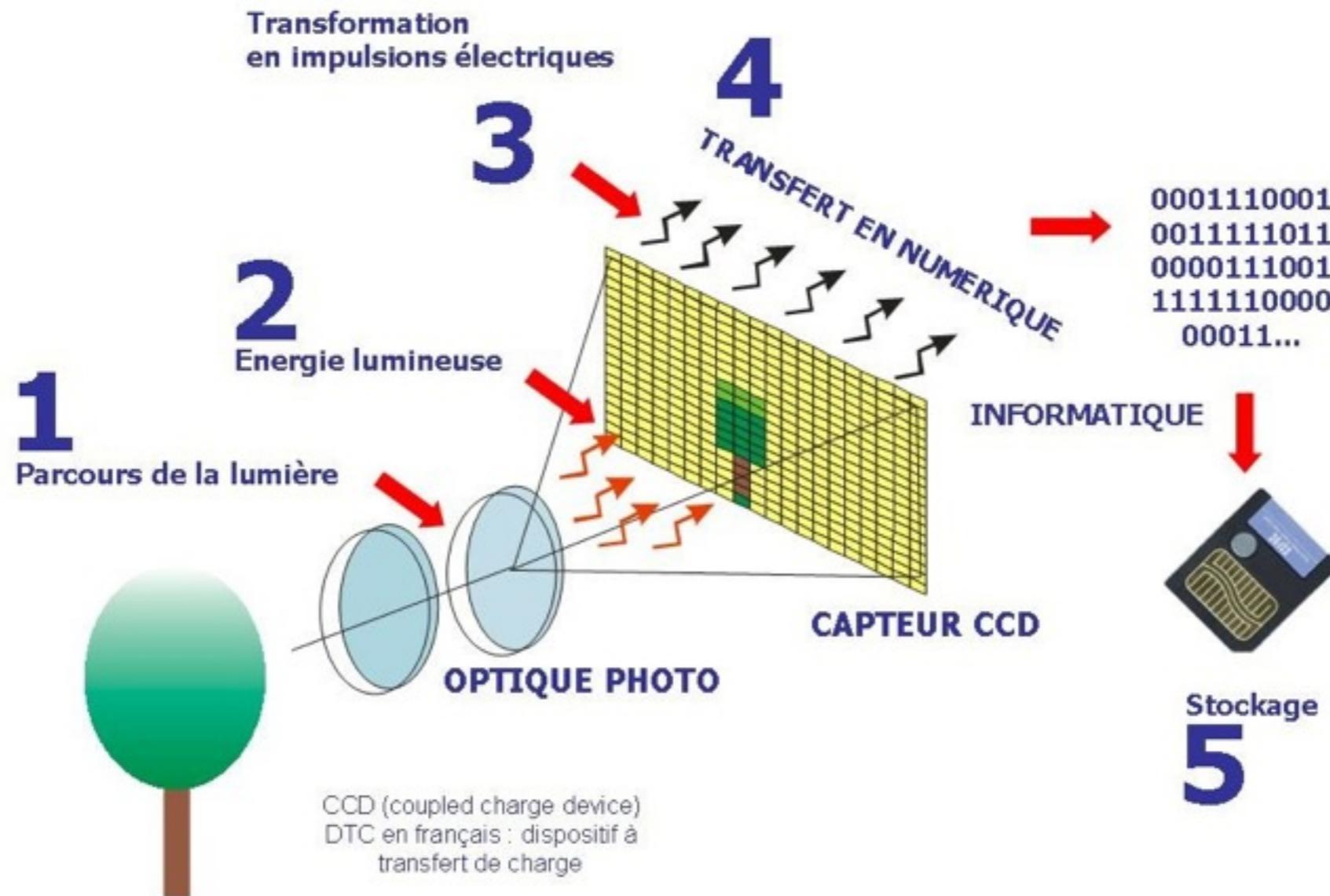
# Comment un appareil photo numérique forme-t-il une image ?



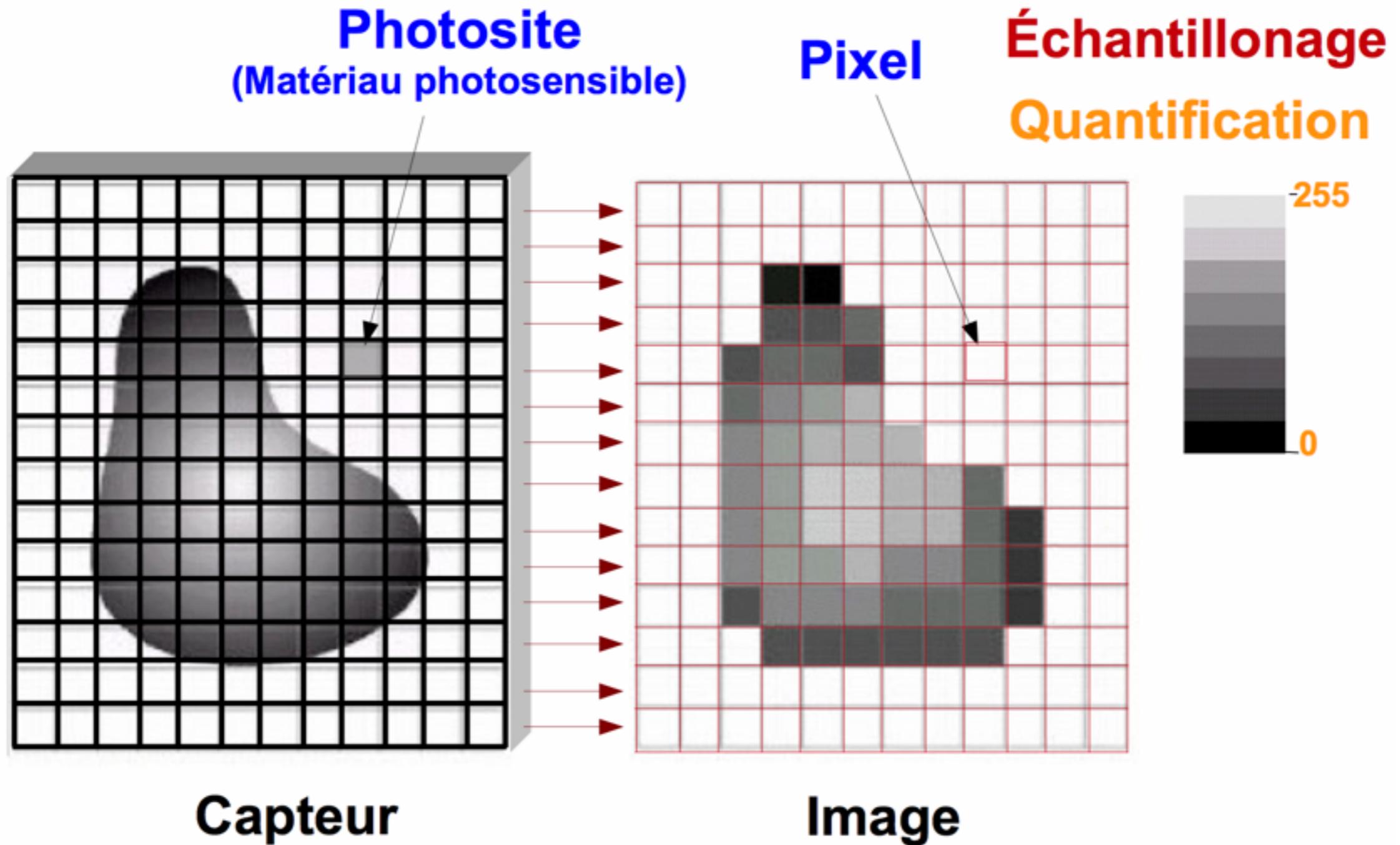
*Please note:* This diagram is representational only and not to scale. There are actually 2048 pixels along each side of the CCD in the Merope camera on Faulkes Telescope North. Each pixel is 13.5 micrometers on a side.



# Comment un appareil photo numérique forme-t-il une image ?



# Comment un appareil photo numérique forme-t-il une image ?



# Prise en compte de la couleur

**LCD**  
Liquid Cristal Display  
(affichage à cristaux liquides)

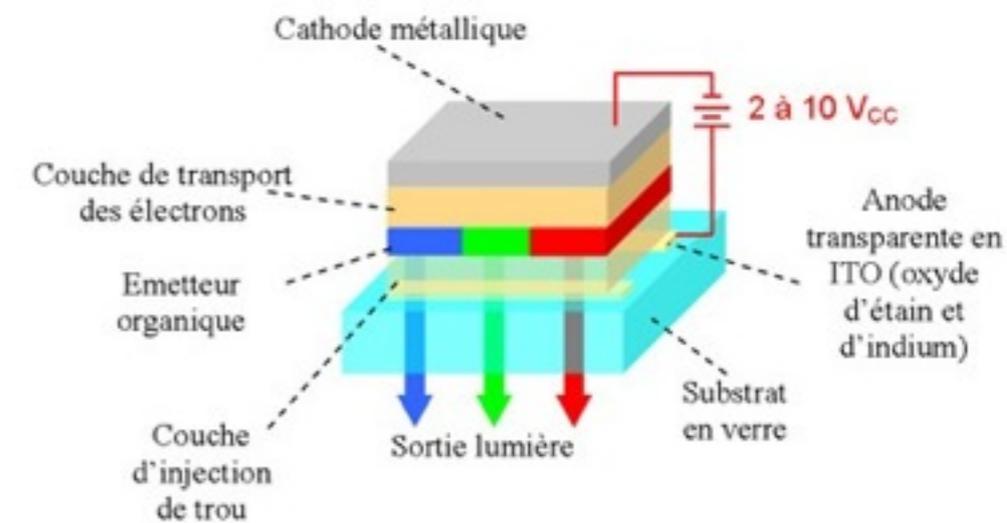
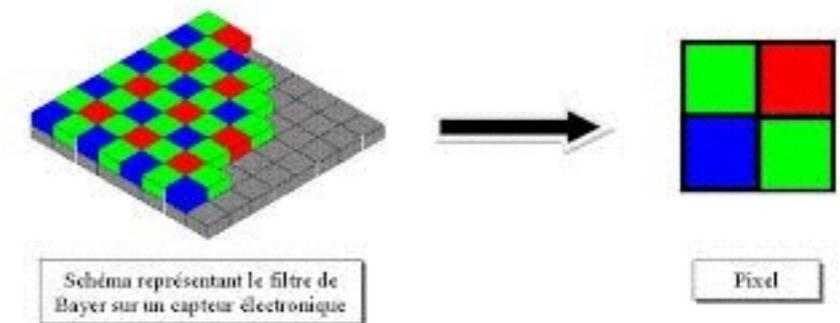
1/. L'écran est divisé en petites unités lumineuses : les pixels.

2/. Chaque pixel est subdivisé en trois rectangles (appelés sous-pixels). La seule différence entre ces sous-pixels provient de la couleur du filtre qui le recouvre : rouge, vert ou bleu.

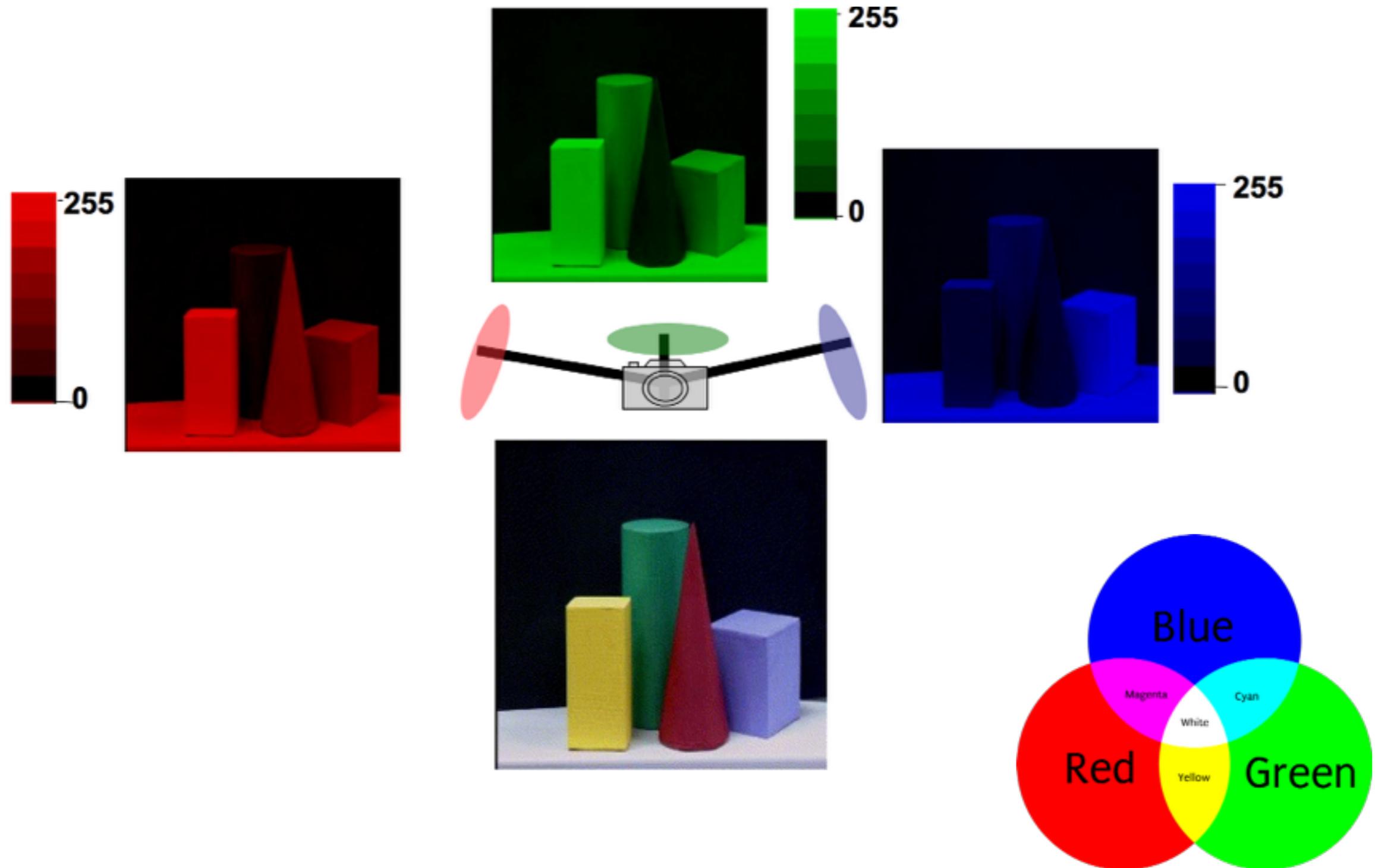
3/. Un tube fluorescent, ou un panneau de diodes électroluminescentes (DEL), placé à l'arrière de l'écran, produit de la lumière blanche. Pour chaque sous-pixel, une couche de cristaux liquides est placée entre deux polariseurs. L'ensemble joue le rôle de store : la quantité de lumière qu'il laisse passer varie en fonction de la tension électrique appliquée aux cristaux liquides. Cette lumière est ensuite colorée en traversant un filtre correspondant à la couleur du sous-pixel : rouge, vert ou bleu.

Nos yeux ne distinguent pas les différents sous-pixels, ils mélangent les lumières colorées primaires (rouge, verte et bleue) provenant de chacun d'eux. Cela suffit pour reproduire pratiquement toutes les couleurs perçues par l'oeil humain : c'est la **synthèse additive des couleurs**.

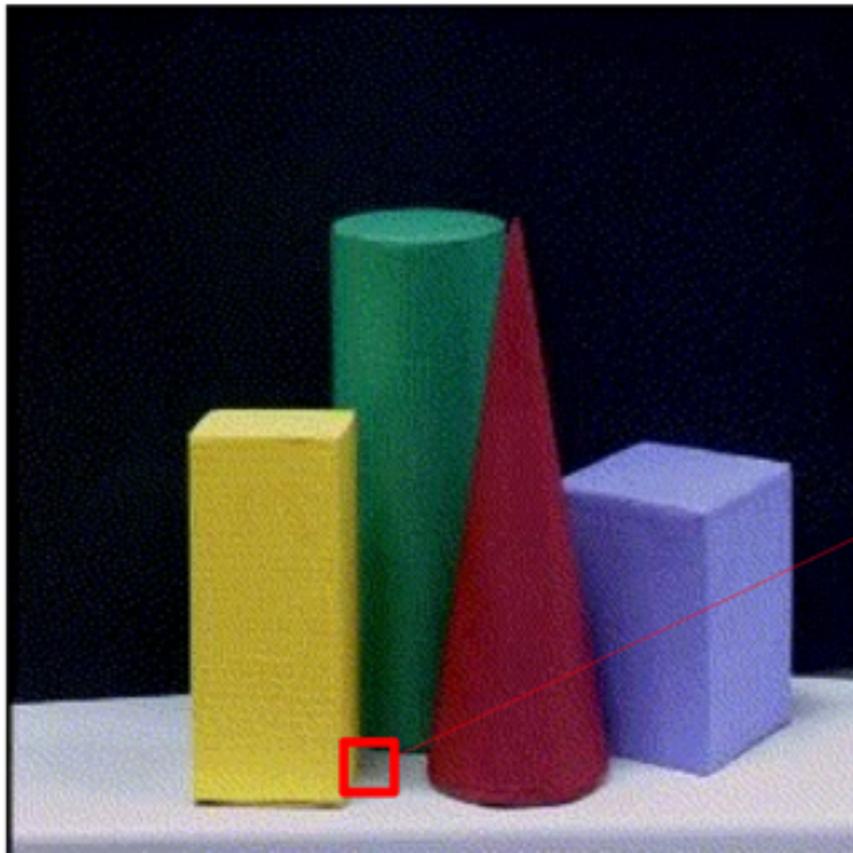
Cristaux liquides « ouverts »  
Cristaux liquides « fermés »  
Tubes fluorescents ou DEL  
Polariseurs  
Filtres  
Vitre de protection



# Formation de l'image couleur par synthèse additive



# Représentation de l'image couleur en mémoire



[ x: 103 , y: 226 , z: 0 ]					
146	109	109	73	73	73
146	109	109	73	73	73
85	0	85	0	85	85
182	109	109	73	73	73
146	109	73	73	73	73
85	0	85	85	85	85
146	146	109	73	109	73
146	146	73	109	73	109
85	0	0	85	85	85
182	109	73	73	73	73
146	109	73	73	73	109
0	85	0	85	0	85
146	146	109	73	109	109
146	109	109	73	109	73
85	85	0	85	85	85
146	109	109	109	73	109
146	109	109	109	73	109
85	0	85	85	85	85

# Effet de l'échantillonnage



10x7 pixels



38x25 pixels

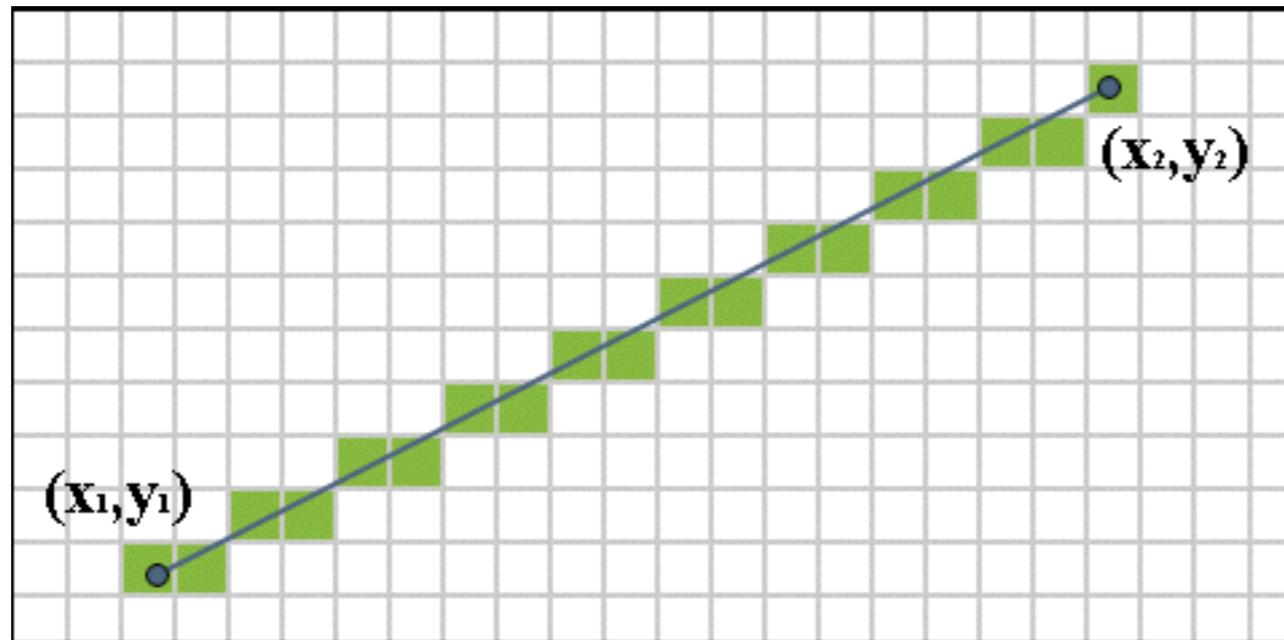


150x100 pixels

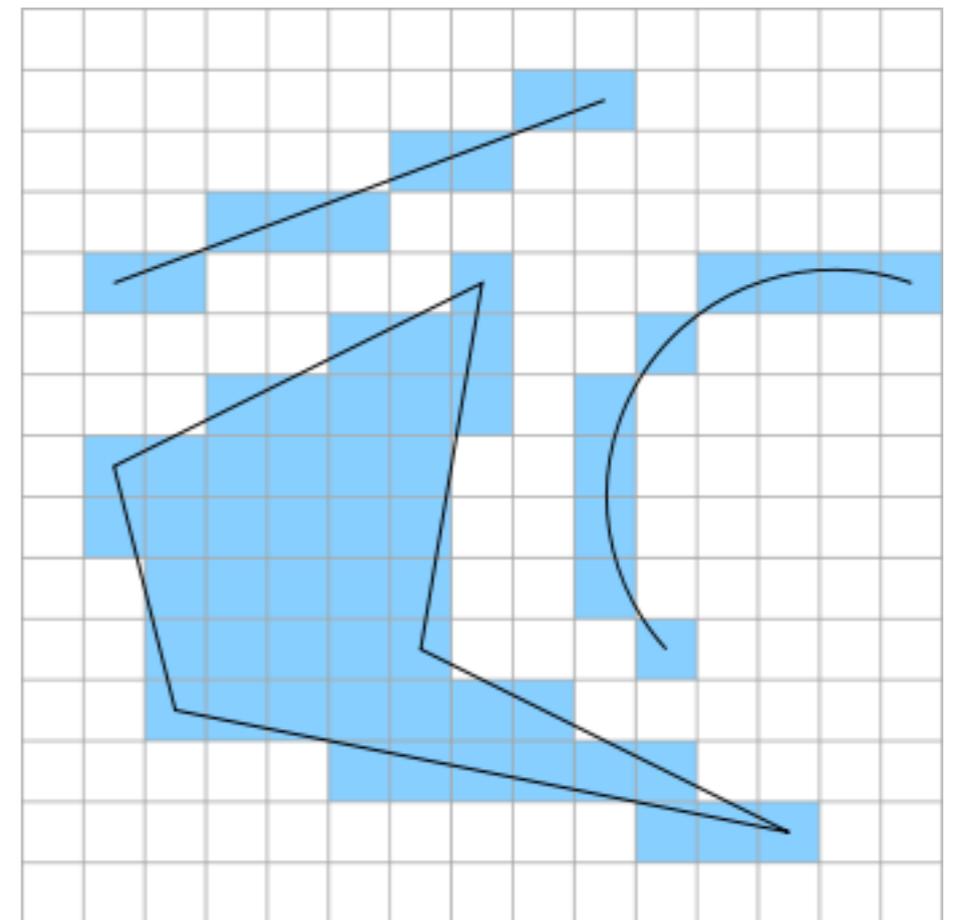


600x400 pixels

# Primitives graphiques de base

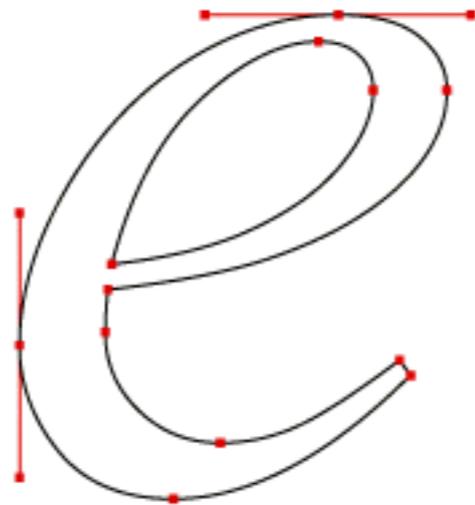


- Tracé de droite discrète  
(algorithme de Bresenham)
- Tracé de courbes
- Remplissage de polygones  
(balayage horizontal)

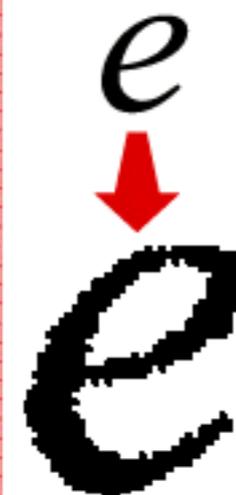
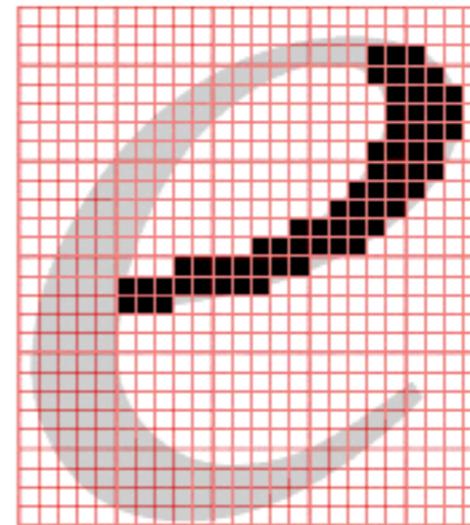


# Bitmap vs. Vectoriel

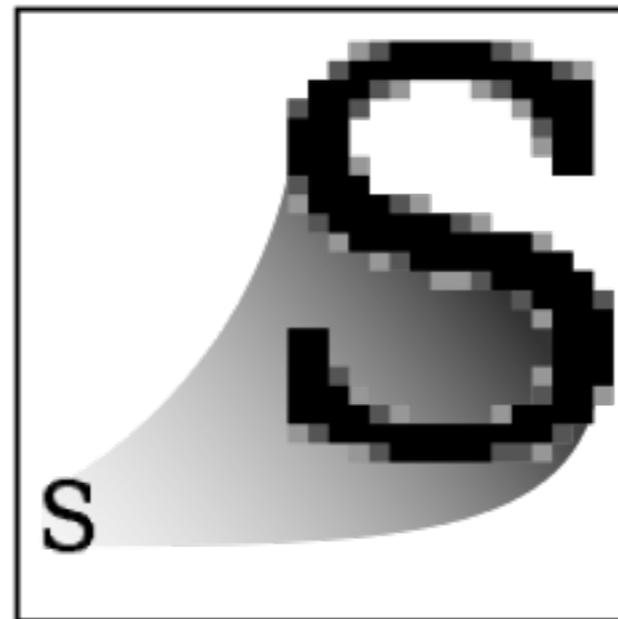
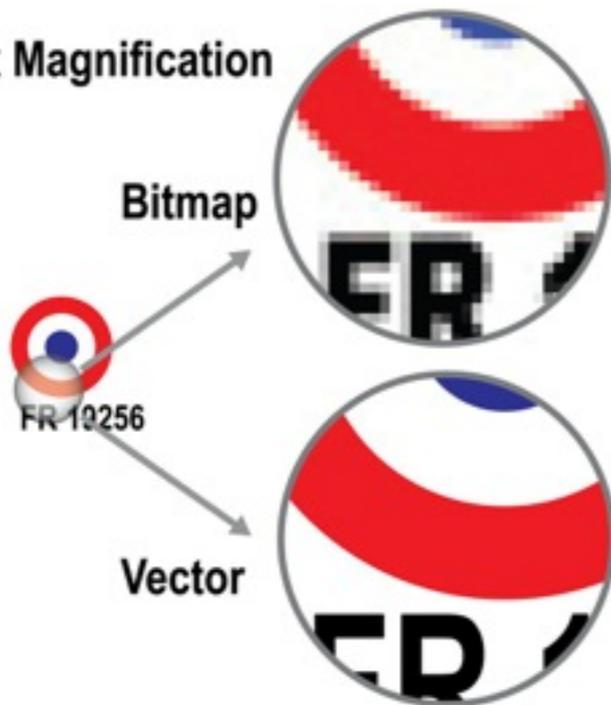
VECTOR GRAPHICS



BITMAPMED (RASTER) GRAPHICS



5x Magnification



Raster  
.jpeg .gif .png



Vector  
.svg

# Compression d'image JPEG



- Réduire la taille de stockage de l'image
- Tout en conservant une bonne qualité visuelle

→ Suppression des détails (« hautes fréquences »)



Q=100



Q=25

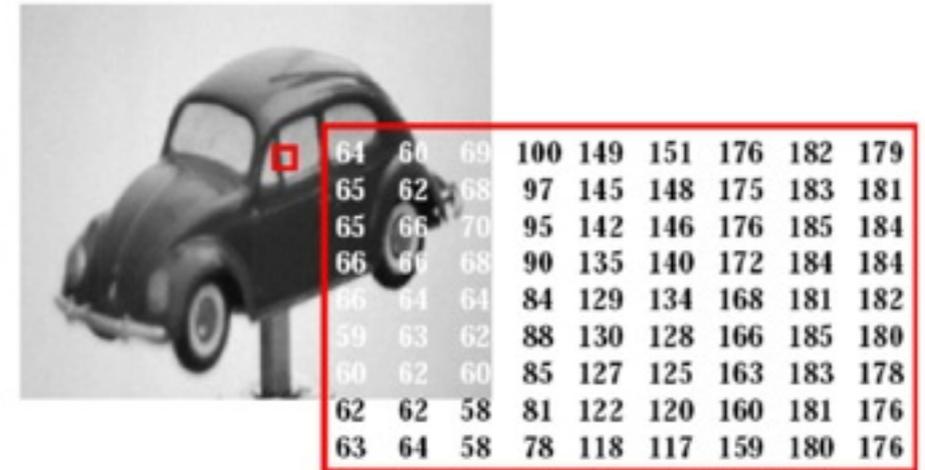
4,3% de taille initiale



Q=1

# En quoi consiste le traitement d'image ?

→ Une image est un tableau à deux dimensions (2D)



## ■ Amélioration de l'image

- Contraste, luminosité, etc
- Filtrage : lissage, débruitage
- Warping, morphing
- Restauration, inpainting
- Vectorisation

## ■ Extraction d'information

- Détection de contours
- Détection de points d'intérêts
- Recherche d'objets
- Segmentation d'image
- Classification de textures

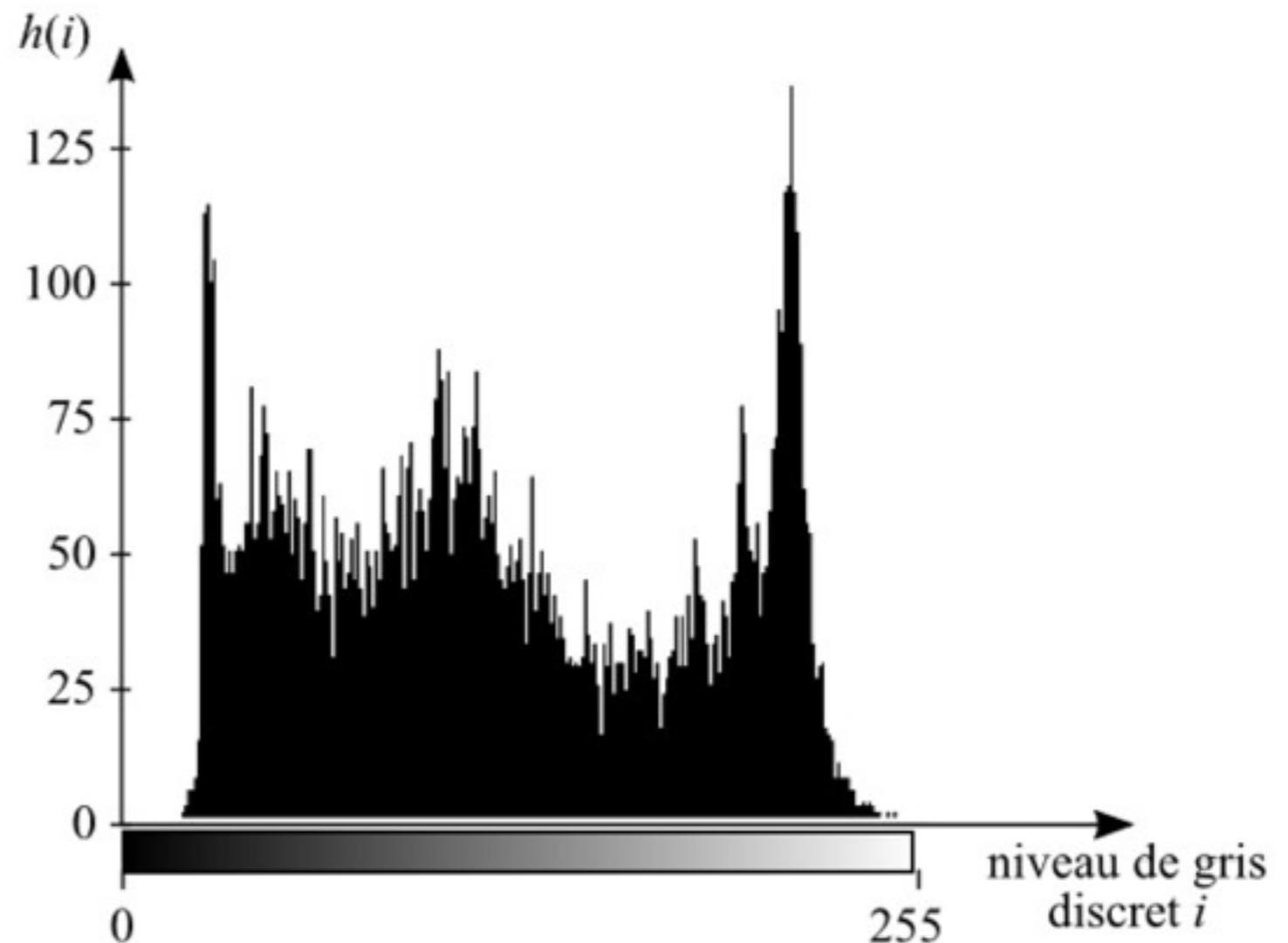
# Utilisation de l'histogramme d'une image

- ✿ *L'histogramme d'une image en niveaux de gris dénombre les occurrences de chacun des niveaux.*

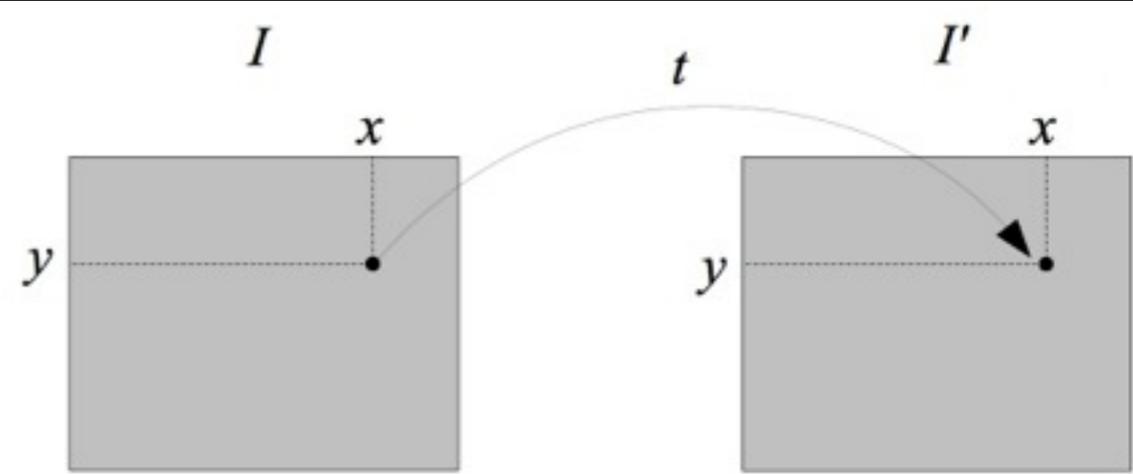
$h(i)$  = nombre de pixels dans l'image ayant le niveau de gris  $i$ .



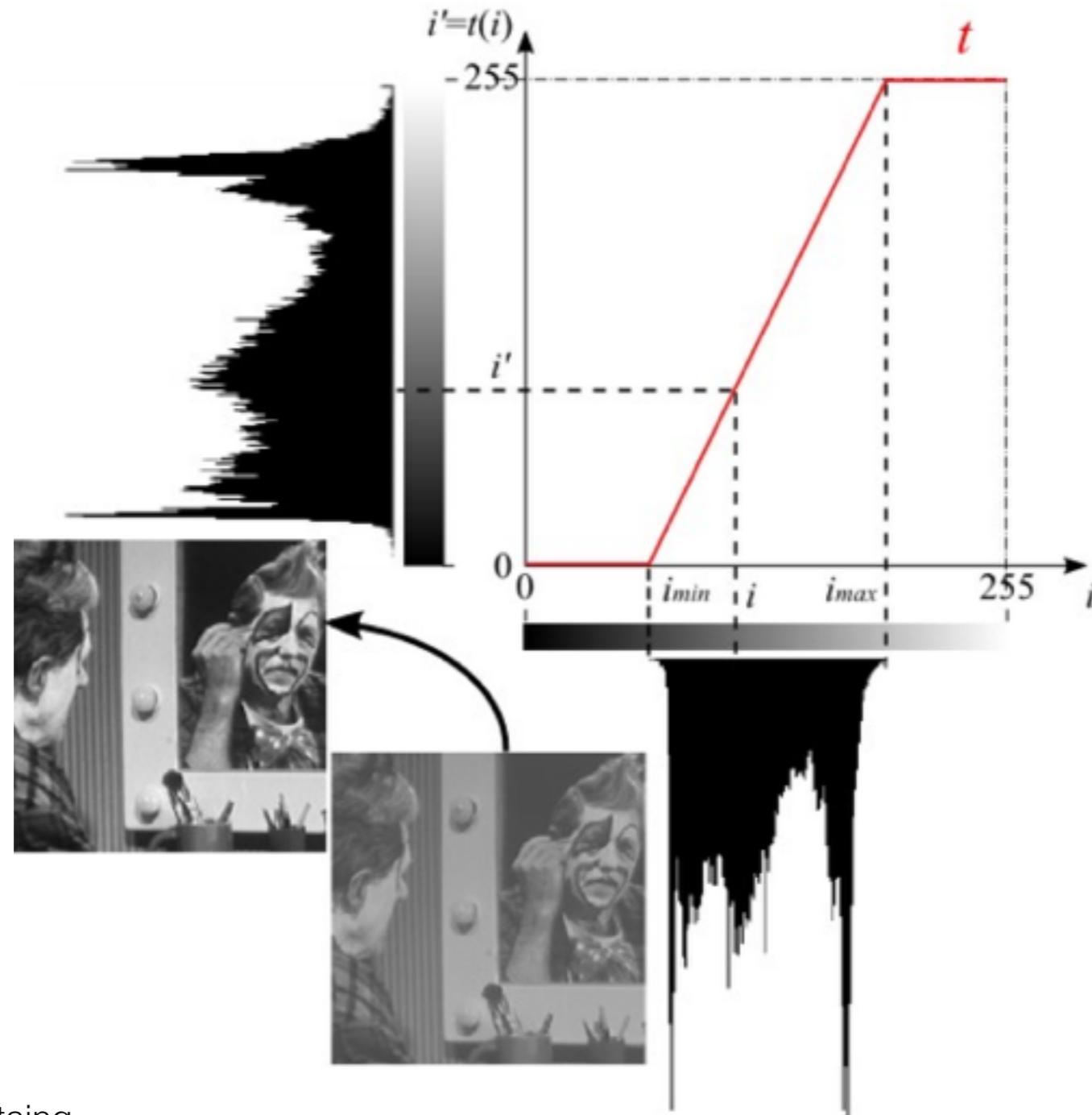
$I_{100 \times 100}$



$$I(x, y) \xrightarrow{t} I'(x, y) = t(I(x, y))$$



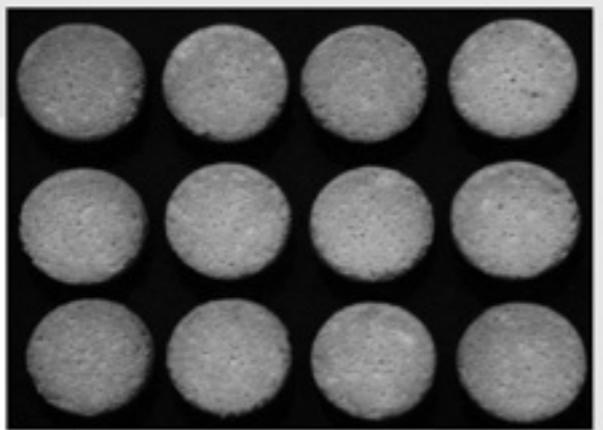
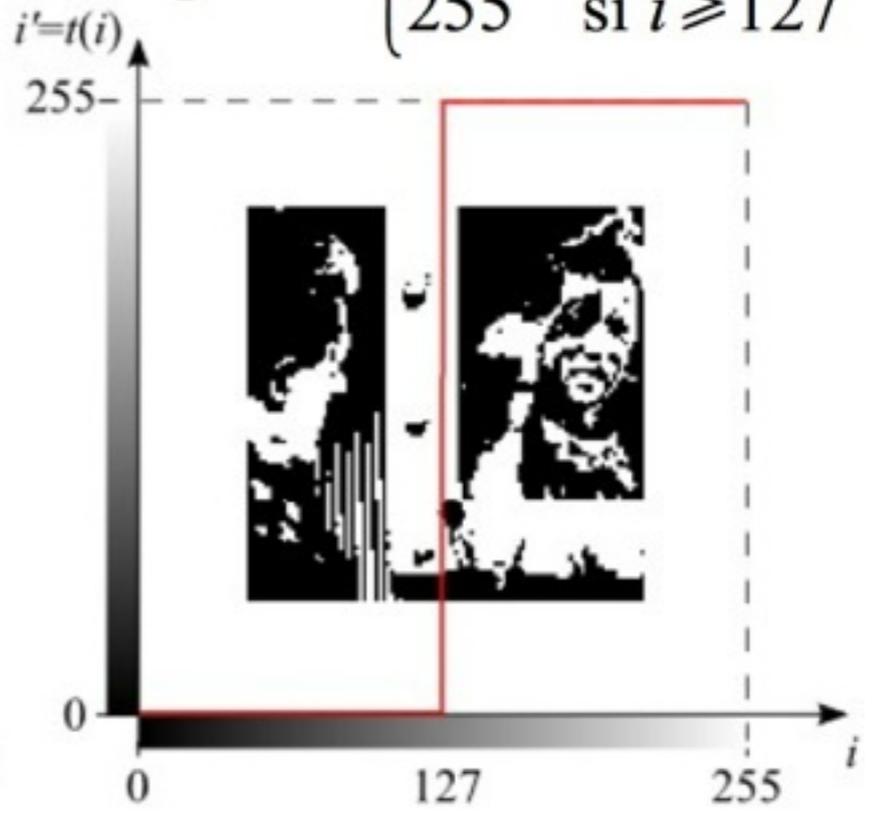
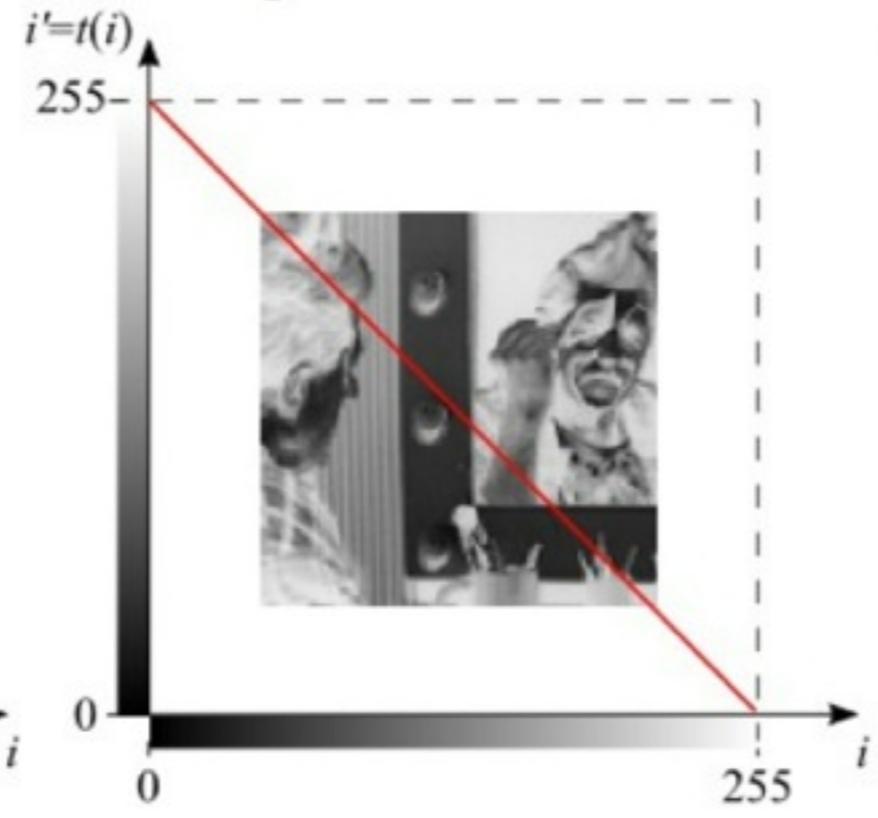
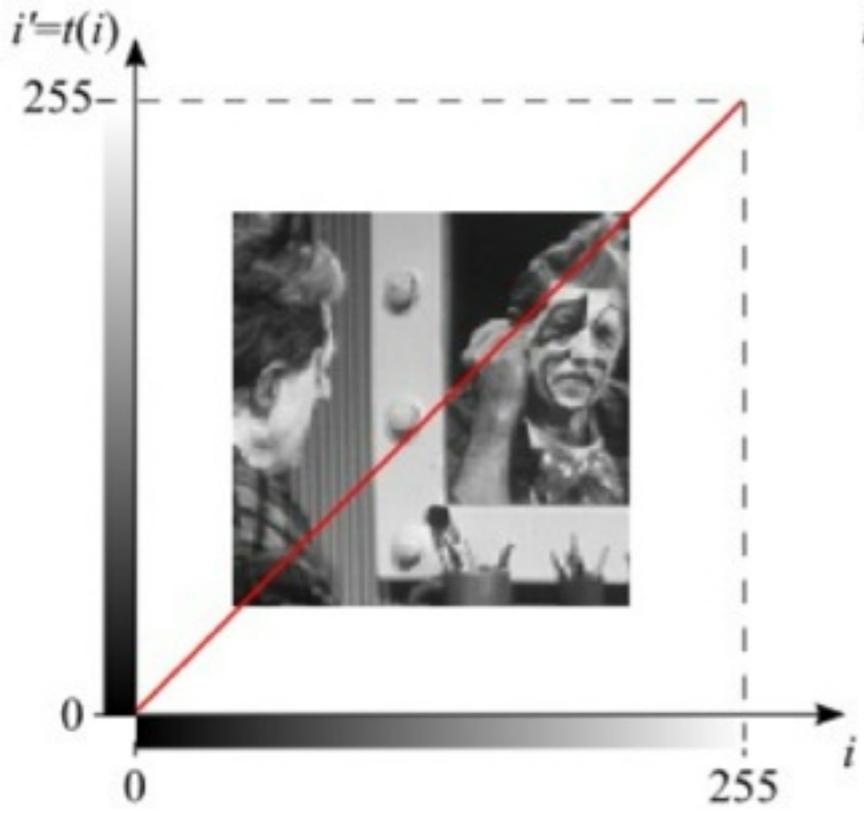
## Transformation ponctuelle



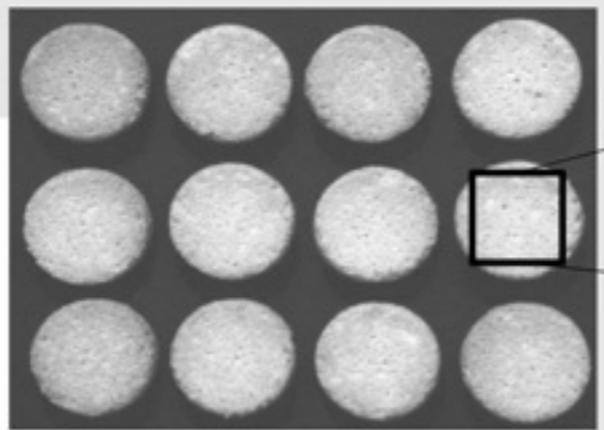
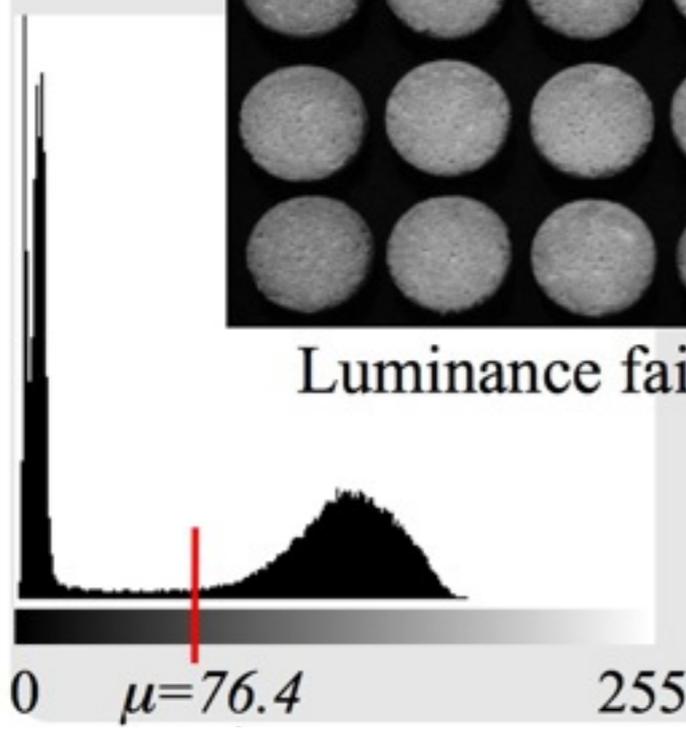
identité  $i'=i$

négatif  $i'=255-i$

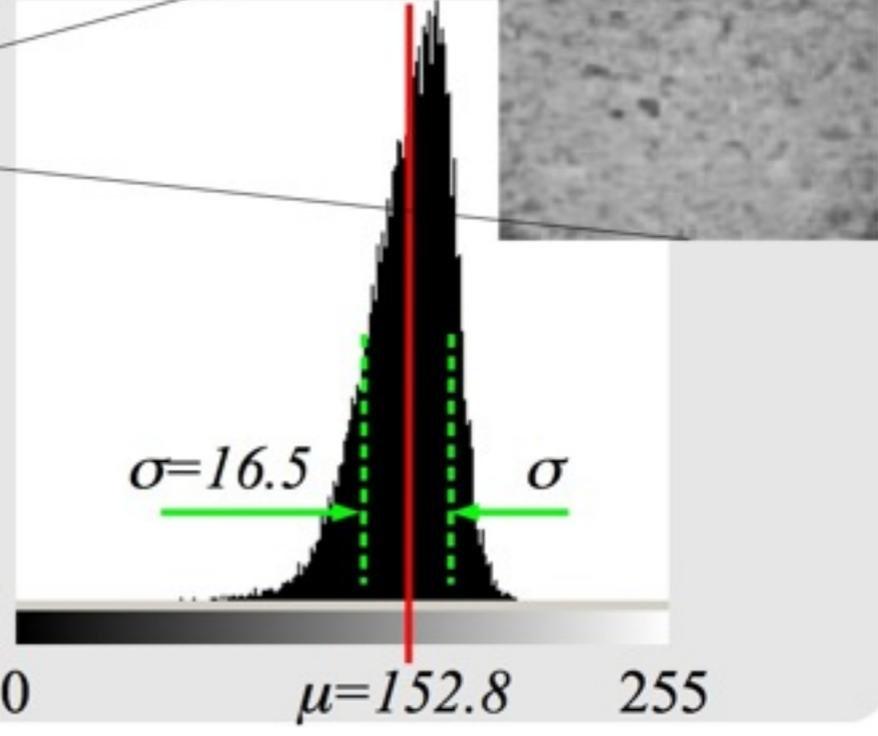
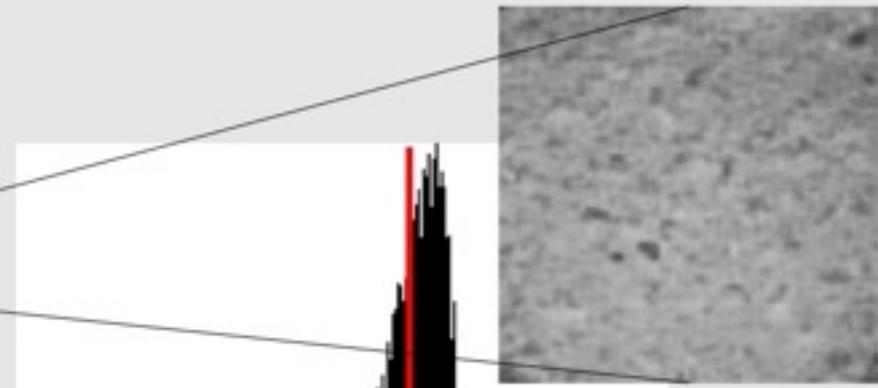
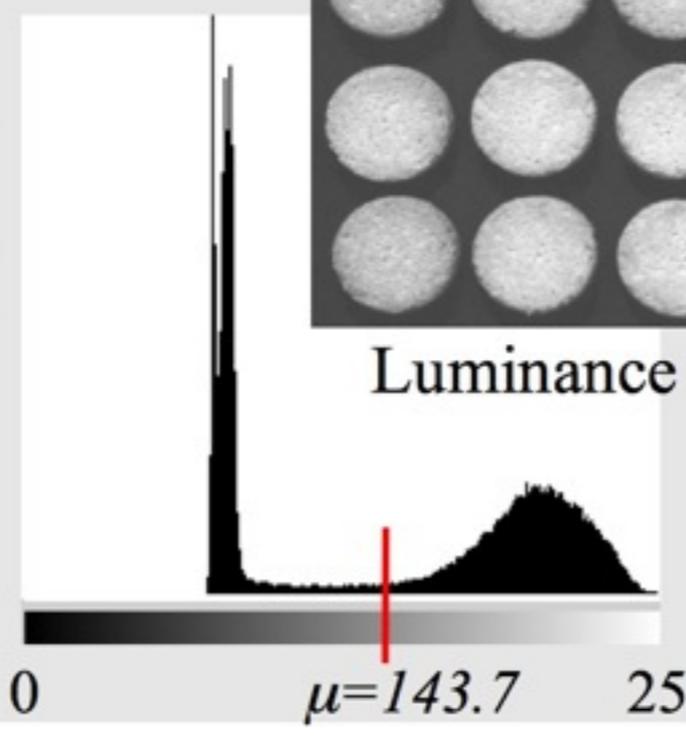
seuillage  $i' = \begin{cases} 0 & \text{si } i < 127 \\ 255 & \text{si } i \geq 127 \end{cases}$



Luminance faible

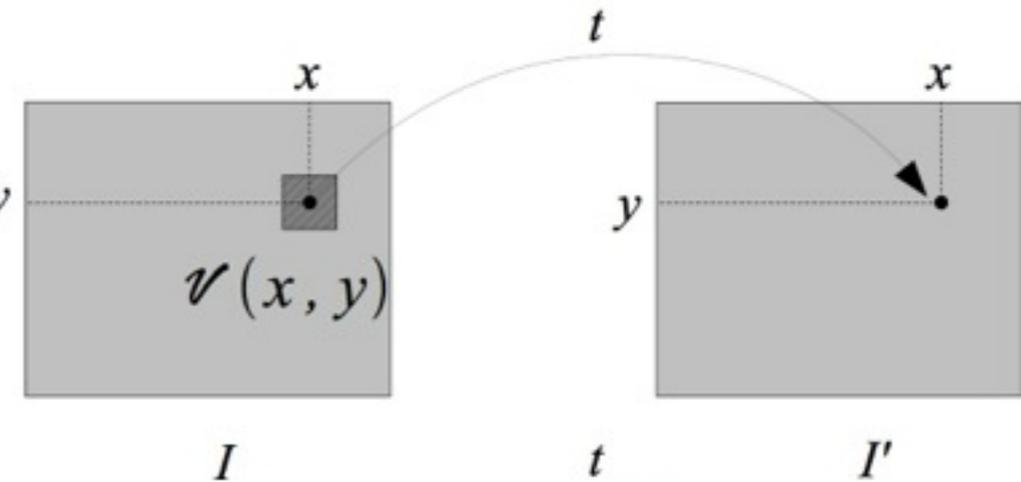


Luminance élevée

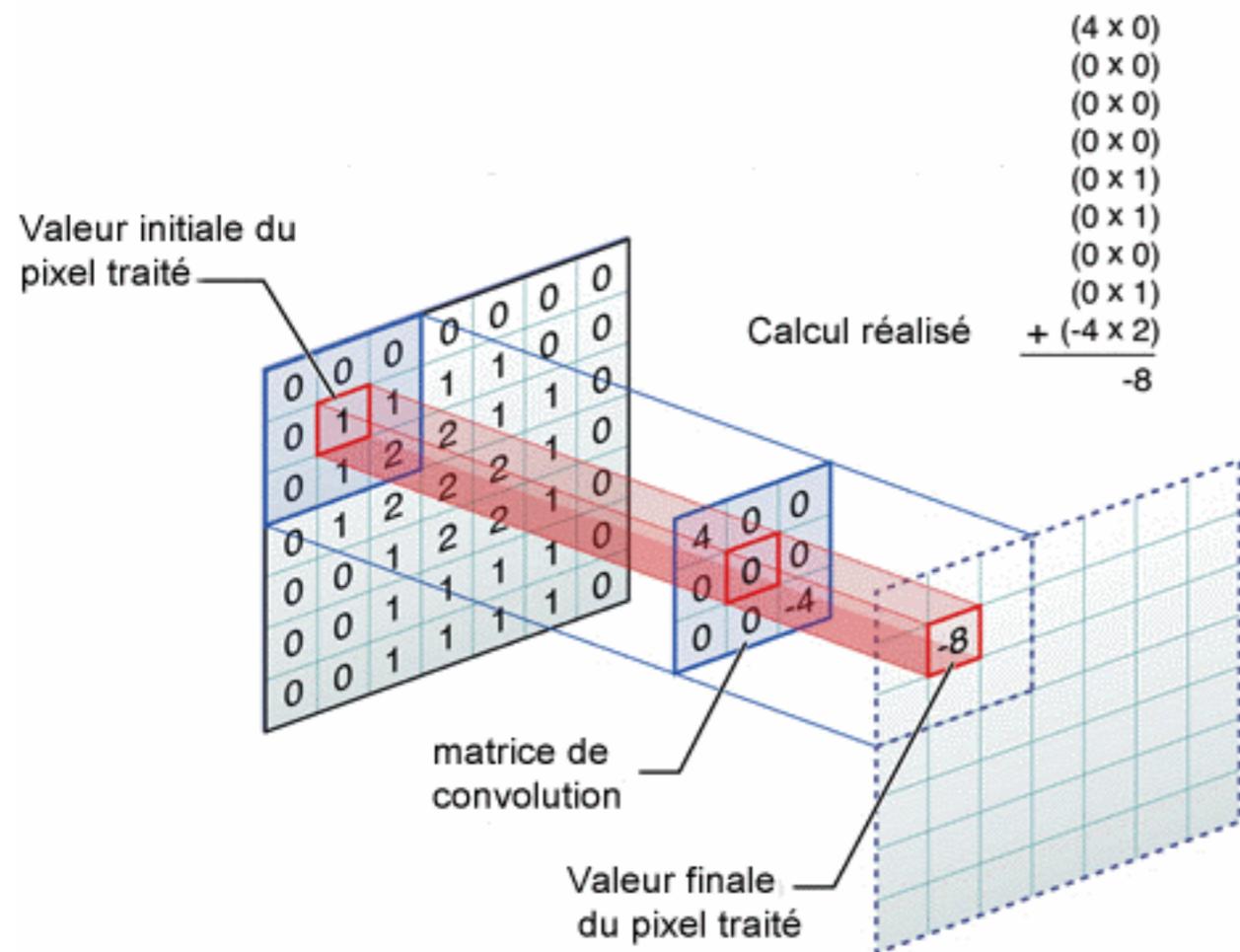


# Filtrage linéaire d'une image

$$I(x, y) \xrightarrow{t} I'(x, y) = t(I(\mathcal{V}(x, y)))$$



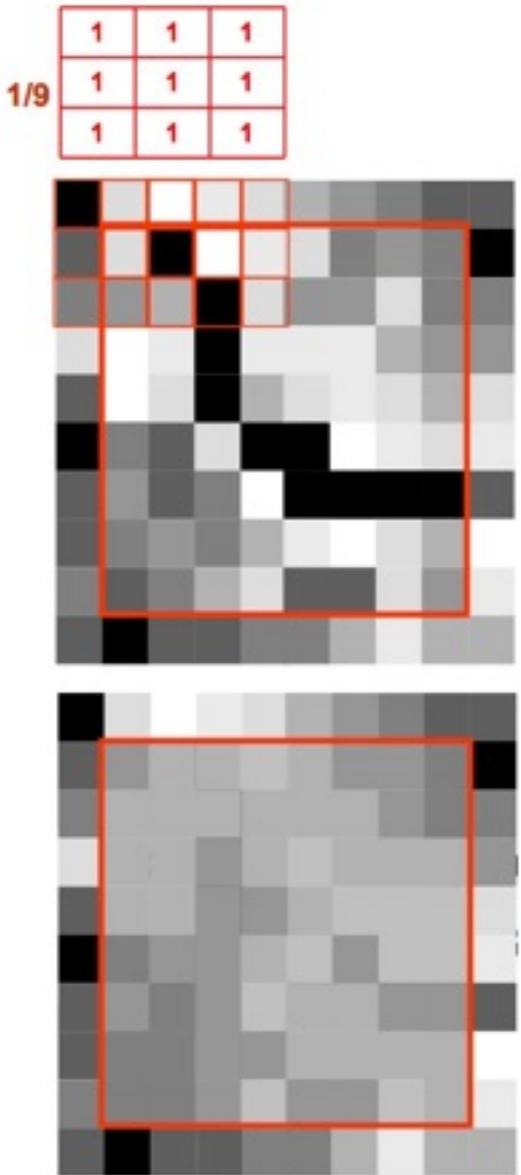
## Transformation locale



On transforme une image en tenant compte des pixels voisins (**voisinage**)

La transformation consiste à appliquer un **masque** (appelé aussi matrice de convolution) à ce voisinage de pixels

# Le filtre **moyenneur**



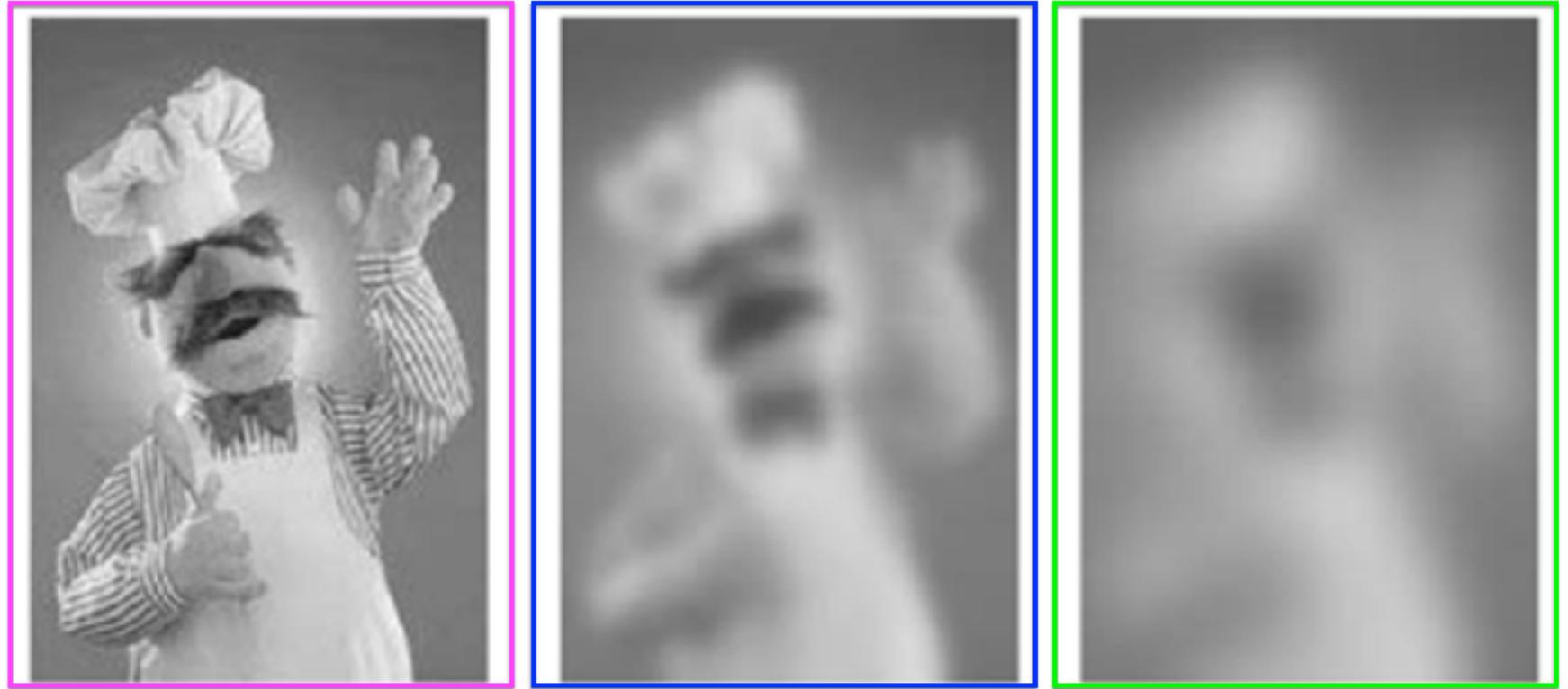
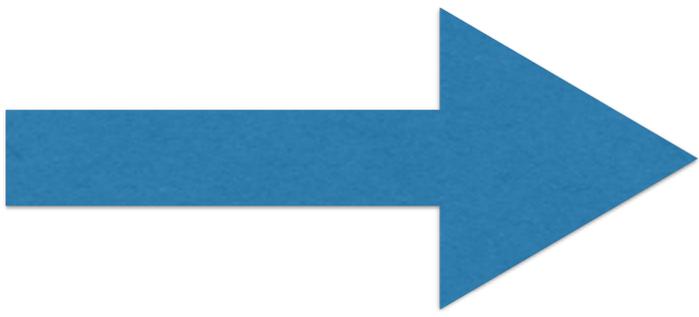
0	5	7	6	5	4	3	2	1	1
1	5	0	7	6	5	2	3	2	0
2	3	4	0	5	3	3	5	2	2
5	7	6	0	6	6	6	4	3	3
1	7	5	0	4	5	6	5	4	5
0	2	1	5	0	0	7	6	5	6
1	3	1	2	7	0	0	0	0	1
1	2	3	2	4	6	7	5	4	7
2	1	2	3	5	1	1	5	2	6
1	0	1	1	2	2	3	6	4	4

0	5	7	6	5	4	3	2	1	1
1	3	4	4	5	4	3	3	2	0
2	4	4	4	4	4	4	3	2	2
5	4	4	3	4	5	4	4	4	3
1	4	4	3	3	4	5	5	5	5
0	2	3	3	4	5	3	5	5	6
1	3	2	3	5	4	4	3	3	1
1	2	2	3	3	4	4	4	4	7
2	2	2	3	5	3	3	5	4	6
1	0	1	1	2	2	3	6	4	4

← Bords non traités

- Masque 3x3 :  $\frac{1}{9} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$
- Masque 5x5 :  $\frac{1}{25} \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix}$
- Masque 11x11: etc

Effet floutage



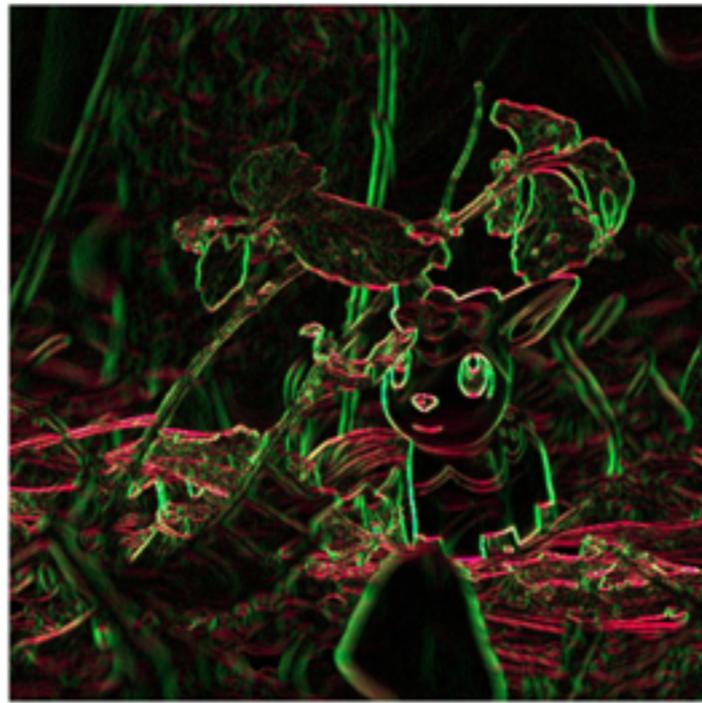
# Exemple d'applications des filtres moyenneurs

- **Effet de flou** (« bokeh » dans le jargon de la photographie)



- **Débruitage** de l'image





# Le filtre de Sobel

- Masque horizontal :

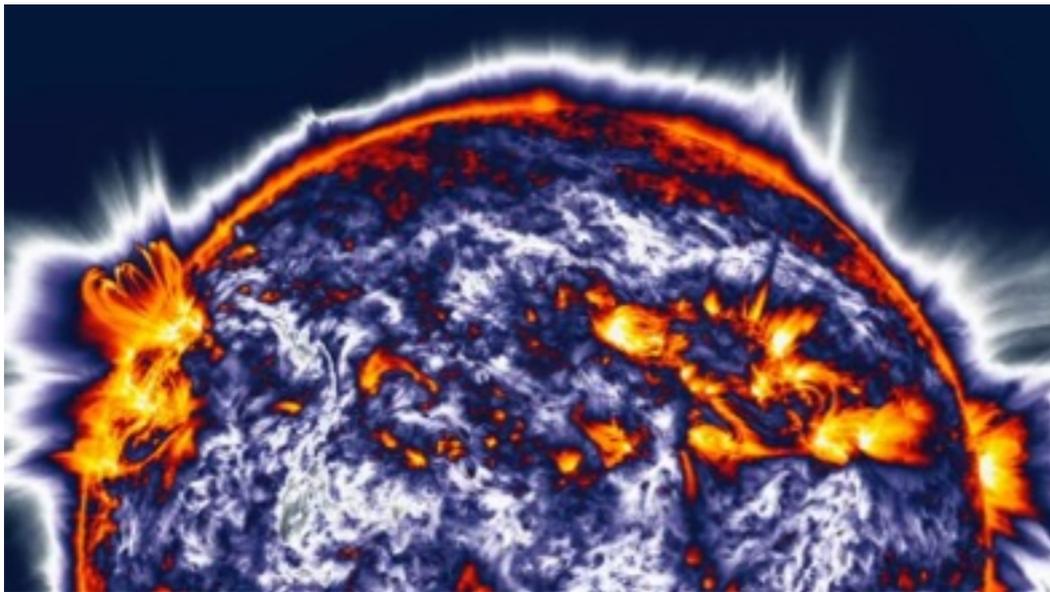
$$G_x = \begin{pmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{pmatrix} * I$$

- Masque vertical :

$$G_y = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix} * I$$

- Image des contours :

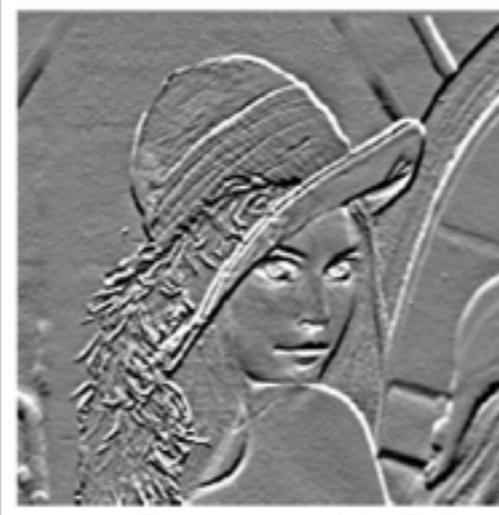
$$G = \sqrt{G_x^2 + G_y^2}$$



Original



Gradient horizontal (Sobel)



Gradient vertical (Sobel)



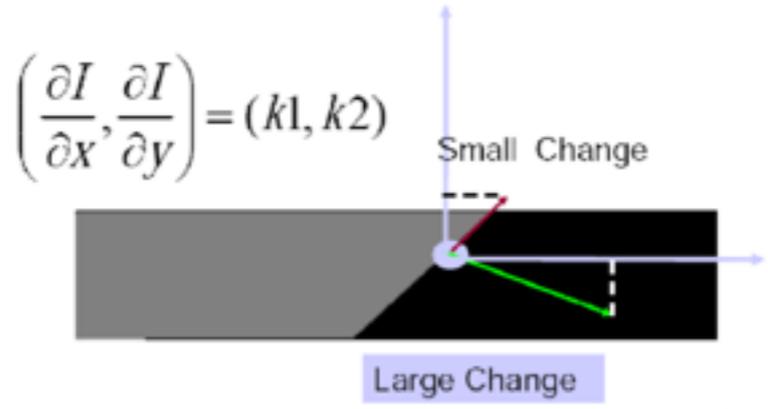
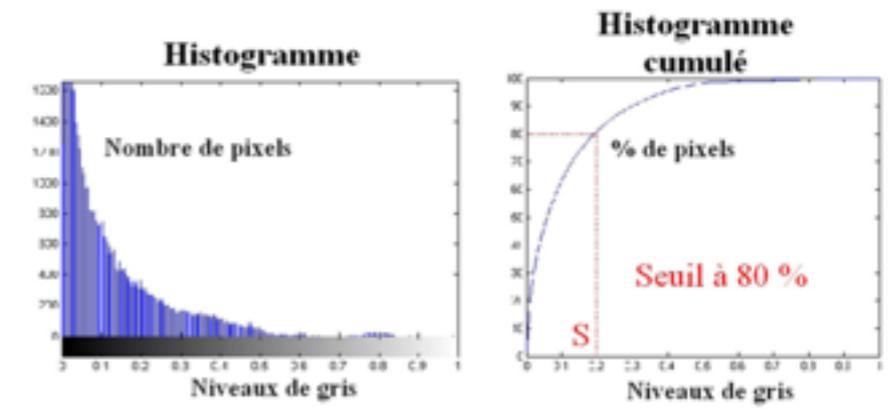
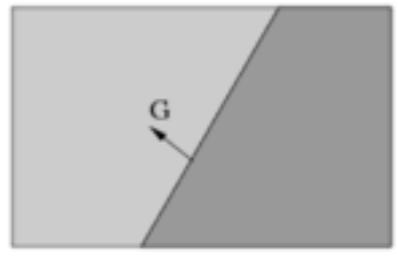
Module du gradient de Sobel

# Vous avez dit à quoi ça sert les maths ?

$$\nabla I = \left( \frac{\partial I(x,y)}{\partial x}, \frac{\partial I(x,y)}{\partial y} \right)^t$$

$$m = \left( \frac{\partial I(x,y)}{\partial x}^2 + \frac{\partial I(x,y)}{\partial y}^2 \right)^{1/2}$$

$$\phi = \arctan \left( \frac{\partial I(x,y)}{\partial y} / \frac{\partial I(x,y)}{\partial x} \right)$$



dérivées secondes

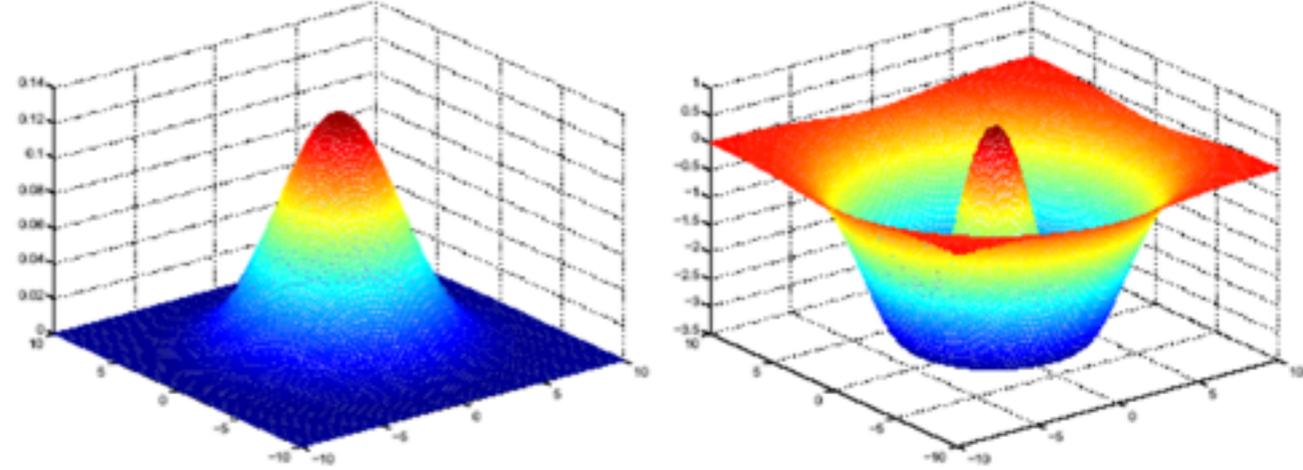
$$\begin{pmatrix} \frac{\partial^2 I}{\partial x^2} & \frac{\partial^2 I}{\partial x \partial y} \\ \frac{\partial^2 I}{\partial y \partial x} & \frac{\partial^2 I}{\partial y^2} \end{pmatrix}$$

laplacien de l'image : somme des dérivées secondes non mixtes

$$\Delta I = \nabla^2 I = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}$$

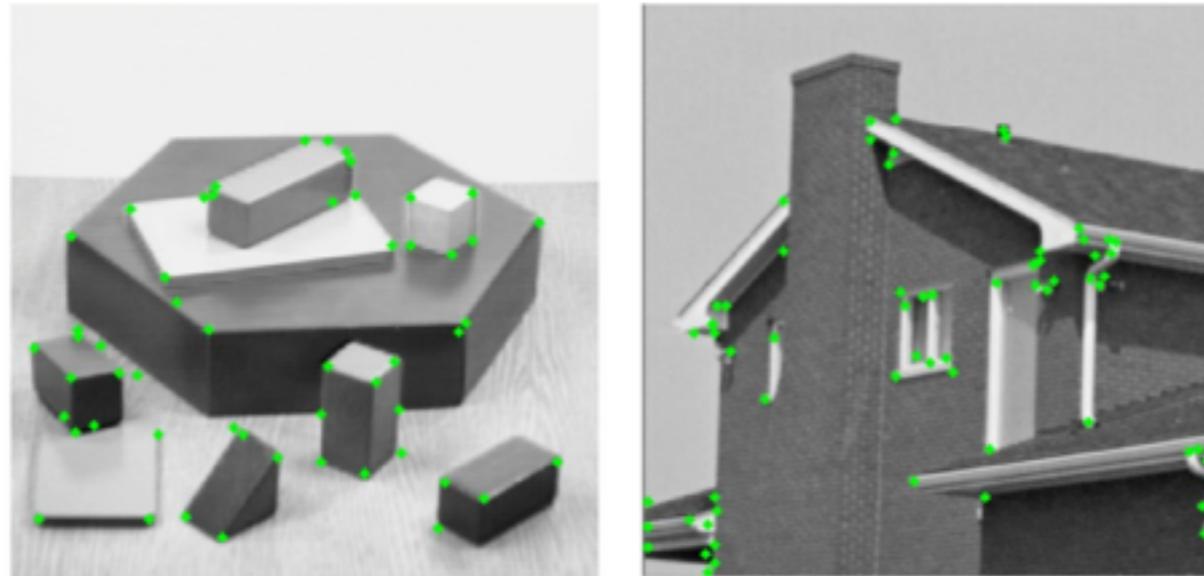


Filtre gaussien et son Laplacien

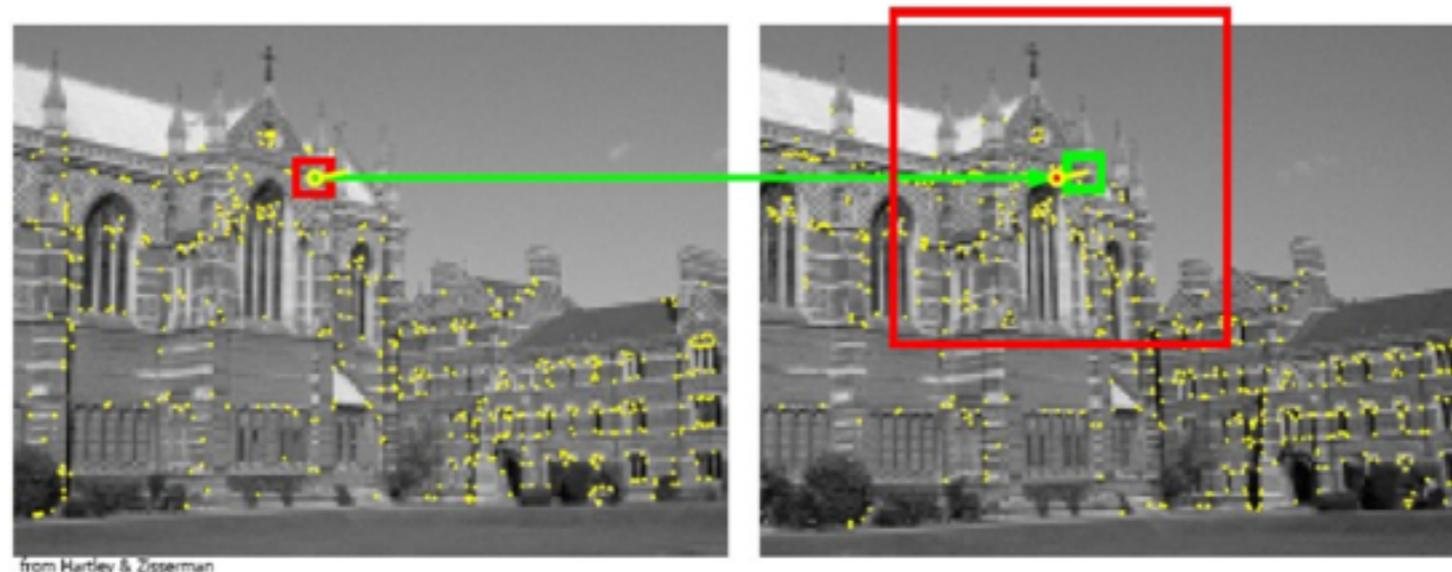


# Détection de points d'intérêts

- Détecter des **coins** (détecteur de Harris)



- Chercher à associer les coins de 2 images



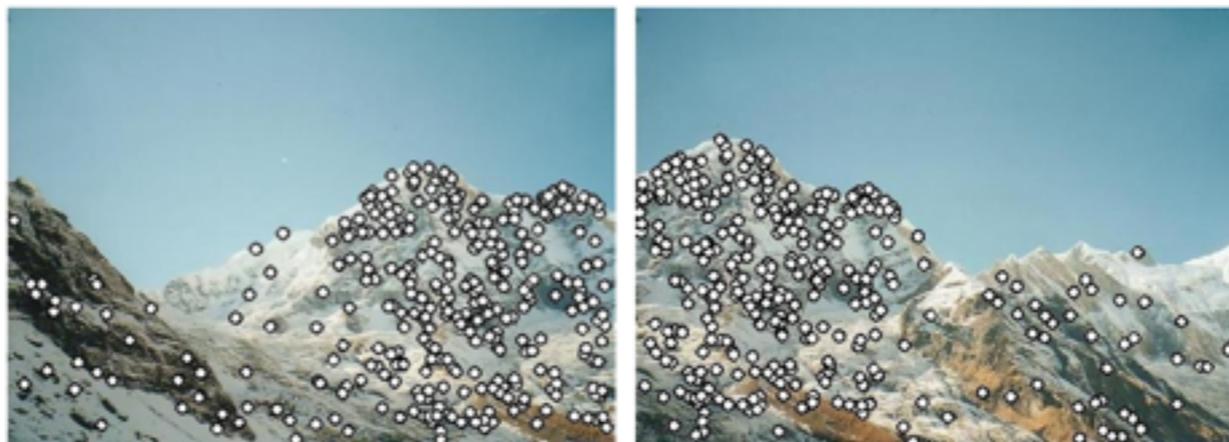
# Exemples d'application des points d'intérêts

## ■ Reconstruction de **panorama**

**1)** Aligner les clichés photographiés « en décalé »



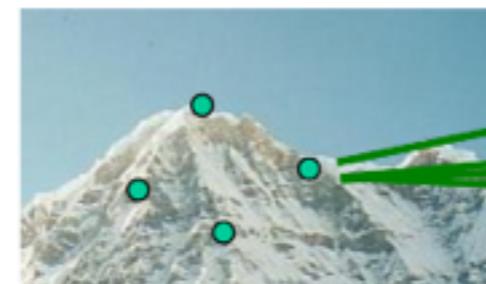
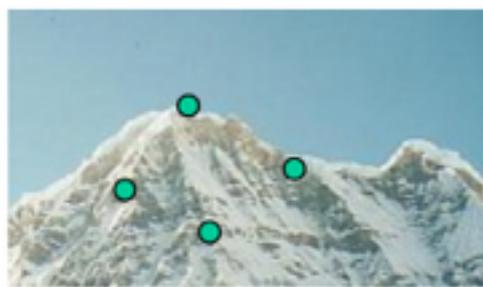
**2)** Recherche les points d'intérêts dans les 2 images



# Exemples d'application des points d'intérêts

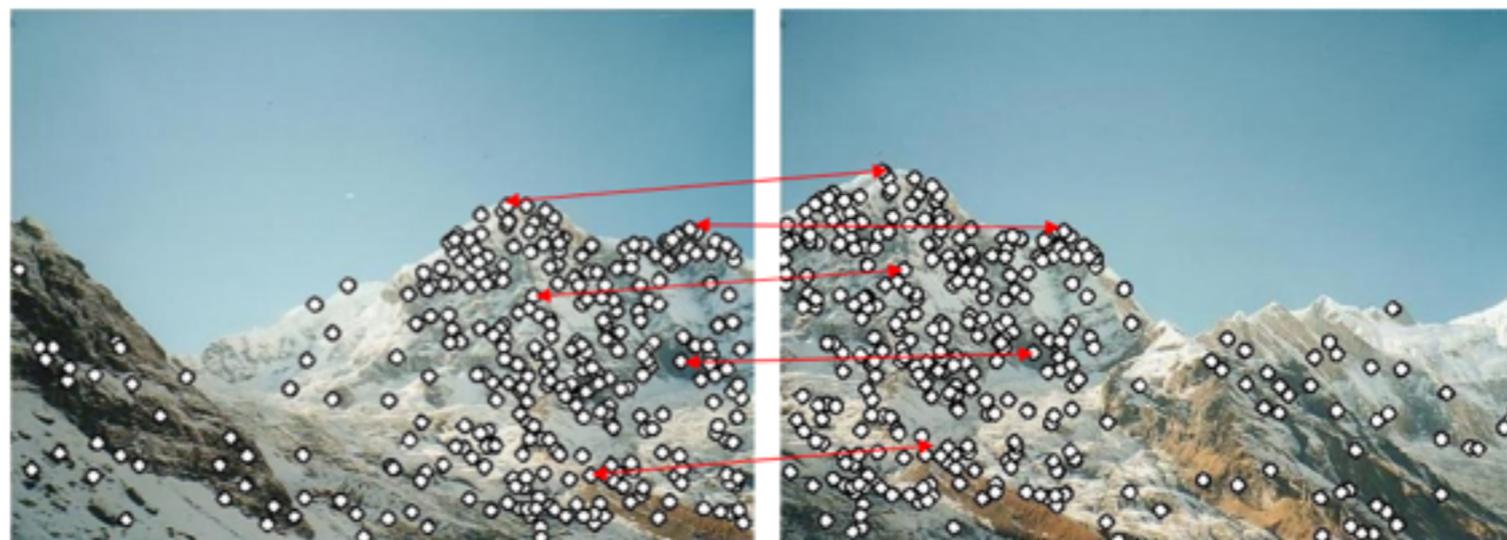
- Reconstruction de **panorama**

## 3) Faire correspondre les points d'intérêts



*Situation impossible*

*Chercher le point d'intérêt correspondant*



# Exemples d'application des points d'intérêts

- Reconstruction de **panorama**

4) Superposer les 2 images



# Exemples d'application des points d'intérêts

- Fonction **panorama** de vos smartphones



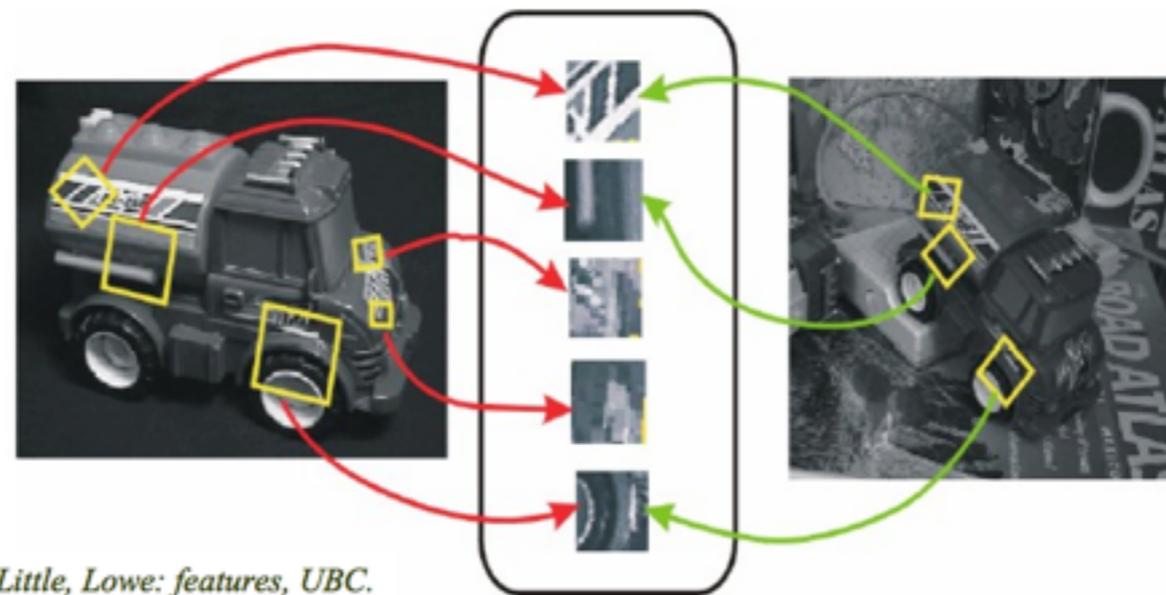
# Exemples d'application des points d'intérêts

- Utilisation des panoramas dans **Street View**

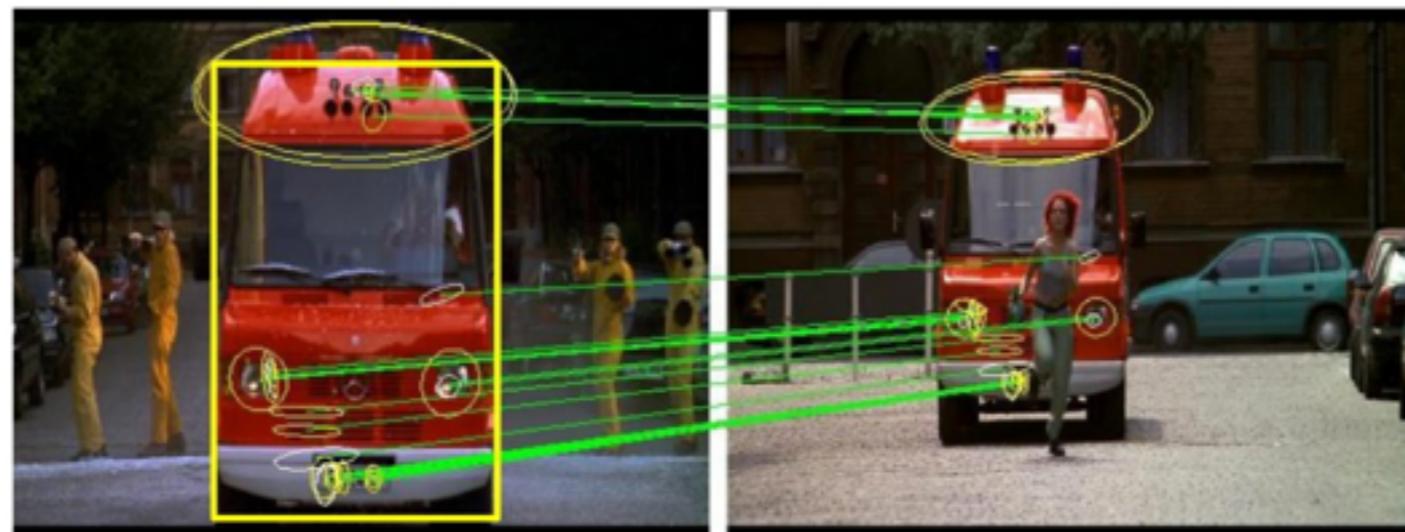


# Exemples d'application des points d'intérêts

- Reconnaissance d'objets dans une image



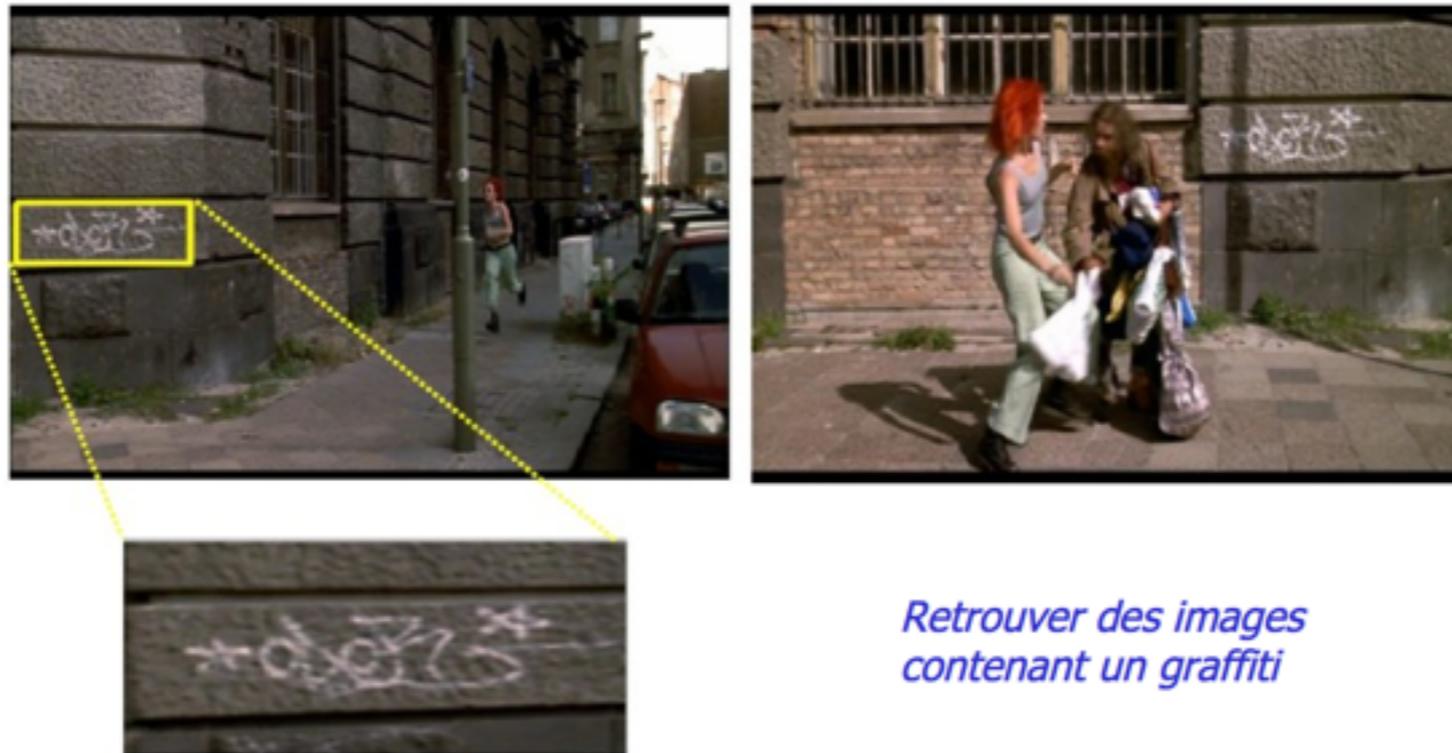
Source : Jim Little, Lowe: features, UBC.



*Ici, c'est le même autobus malgré la taille différente et le fait qu'il soit en partie caché sur une image...*

# Exemples d'application des points d'intérêts

- Reconnaissance **d'objets** dans une image



*Des scènes sont différentes (angle de vue, zoom, lumière, ...), mais on veut identifier des éléments semblables...*

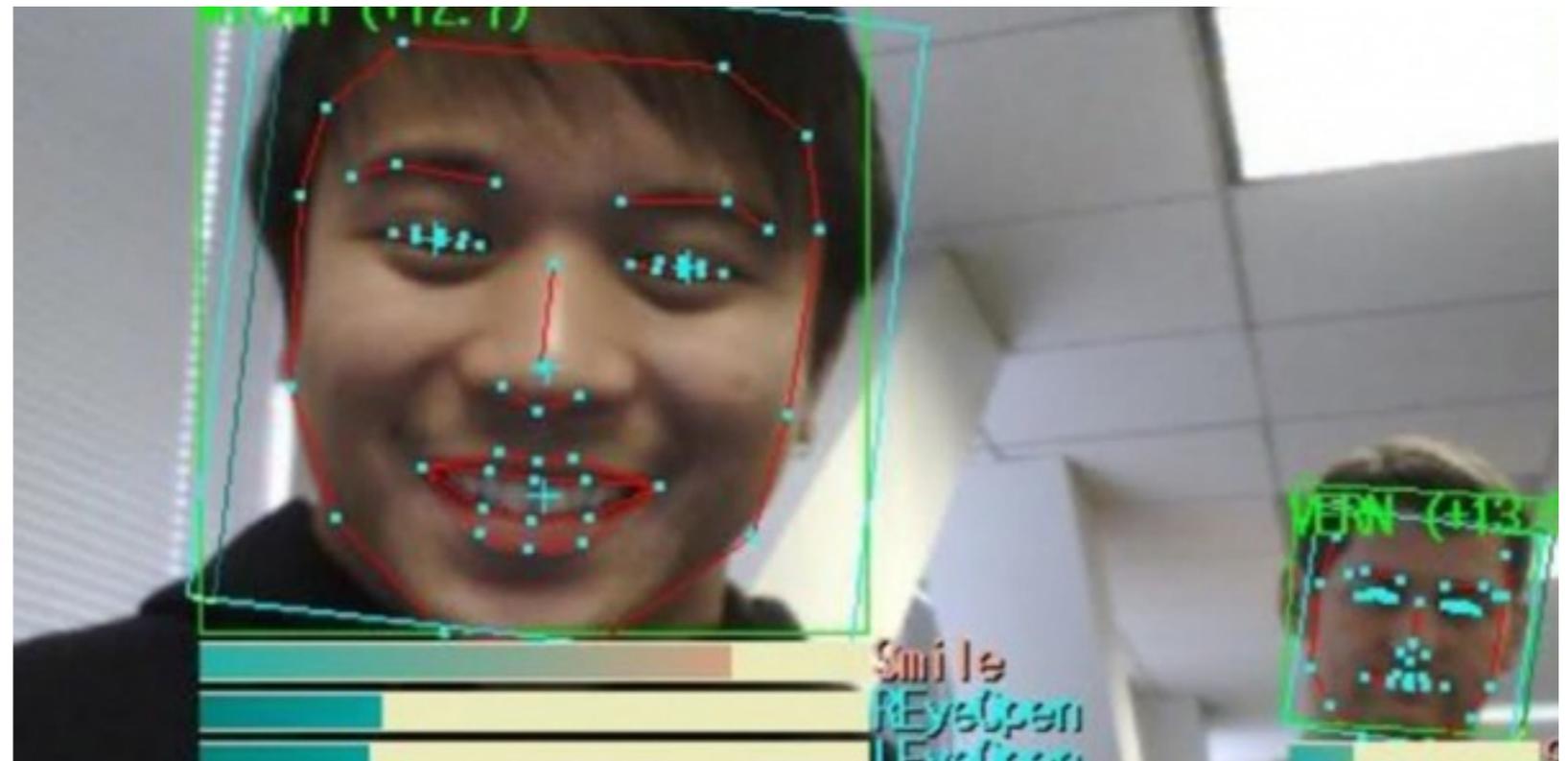
# Exemples d'application des points d'intérêts

- Reconnaissance de visages dans une image

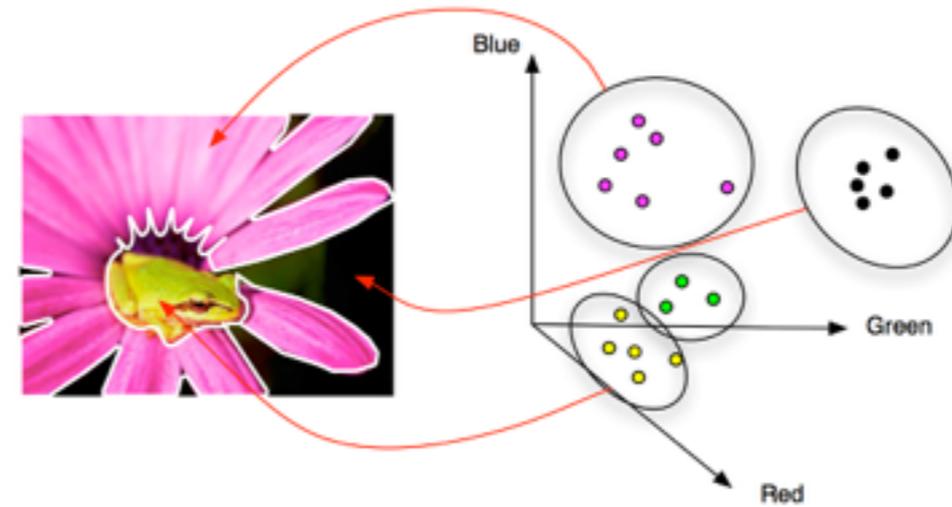
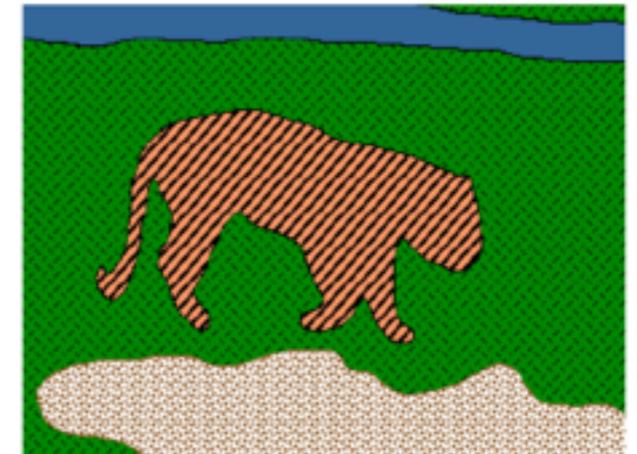
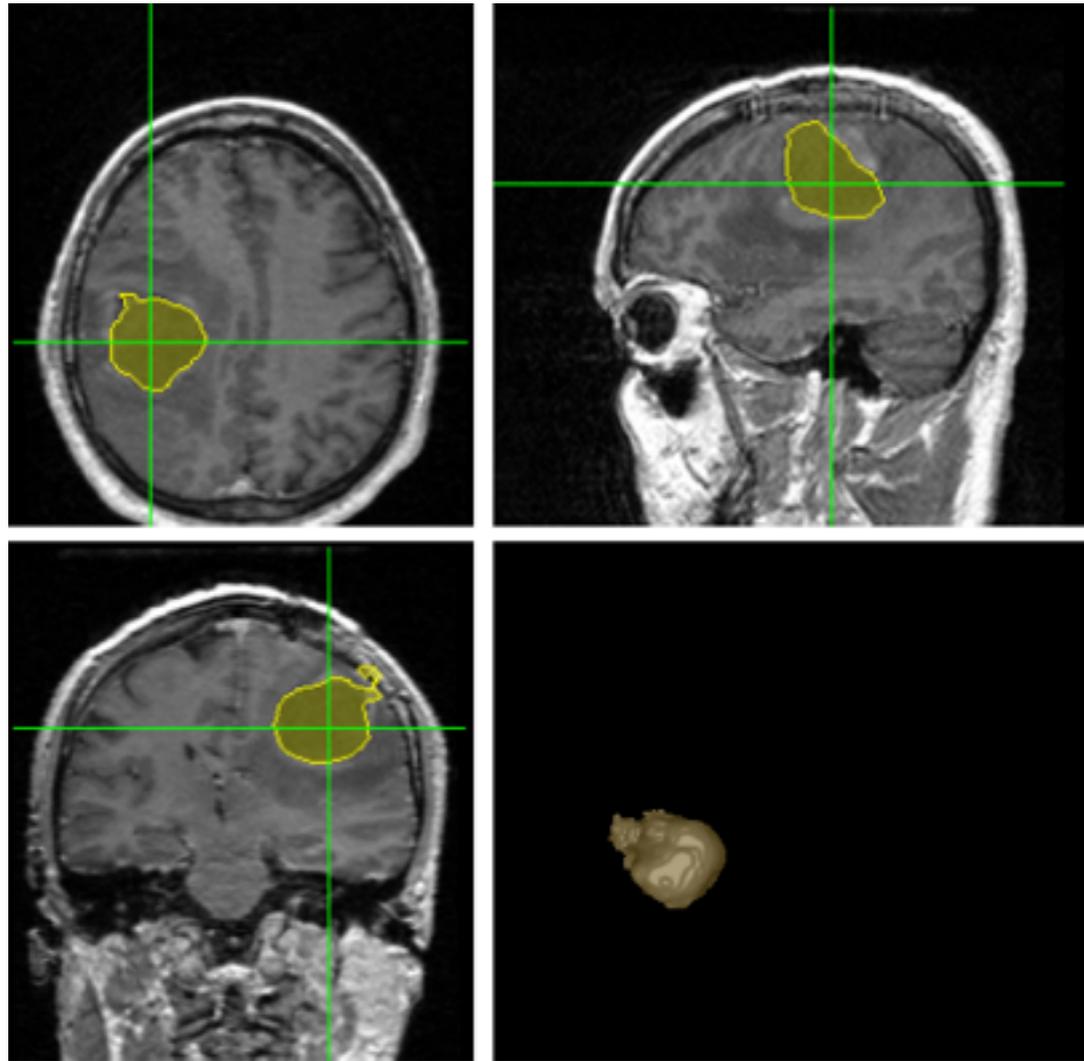


Application de masques spécifiques propres aux caractéristiques des visages

Détection des visages en temps réel (algorithme de Viola et Jones)



# Segmentation d'image



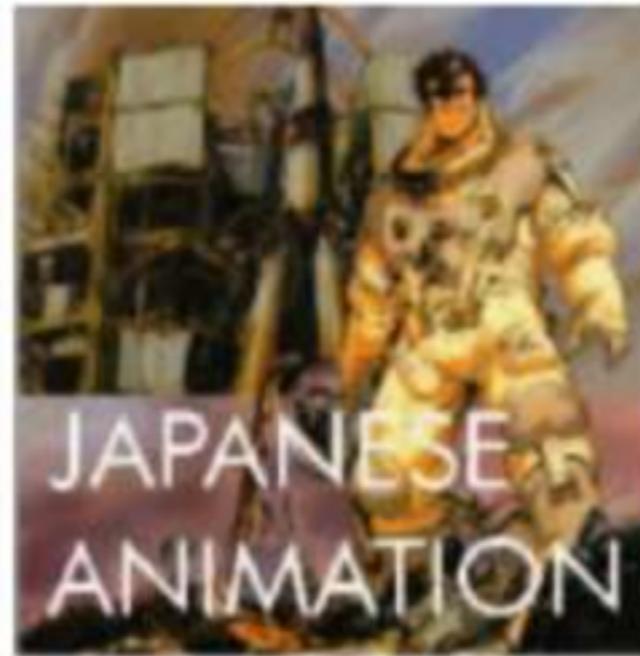
# Restauration d'image

- Restaurer des **données manquantes** dans une image



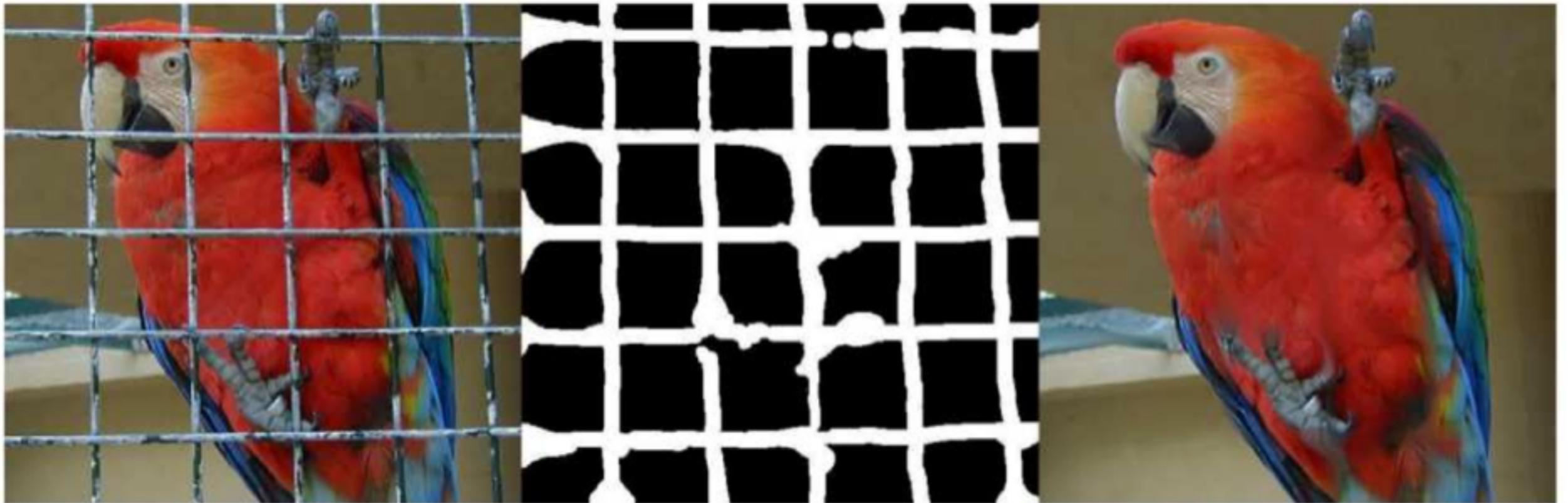
# Restauration d'image

- **Supprimer du texte** ou un logo superposé à une image



# Restauration d'image

- Supprimer un objet dans une image



# Restauration d'image

- Supprimer un objet dans une image



# Restauration d'image

- Reconstituer une image à partir de très peu de pixels



**98% des pixels ont  
été supprimés !**

De l'image à la vidéo il n'y a qu'un pas (d'échantillonnage)

Illustration avec un exemple de cinéma de poche :  
le flipbook

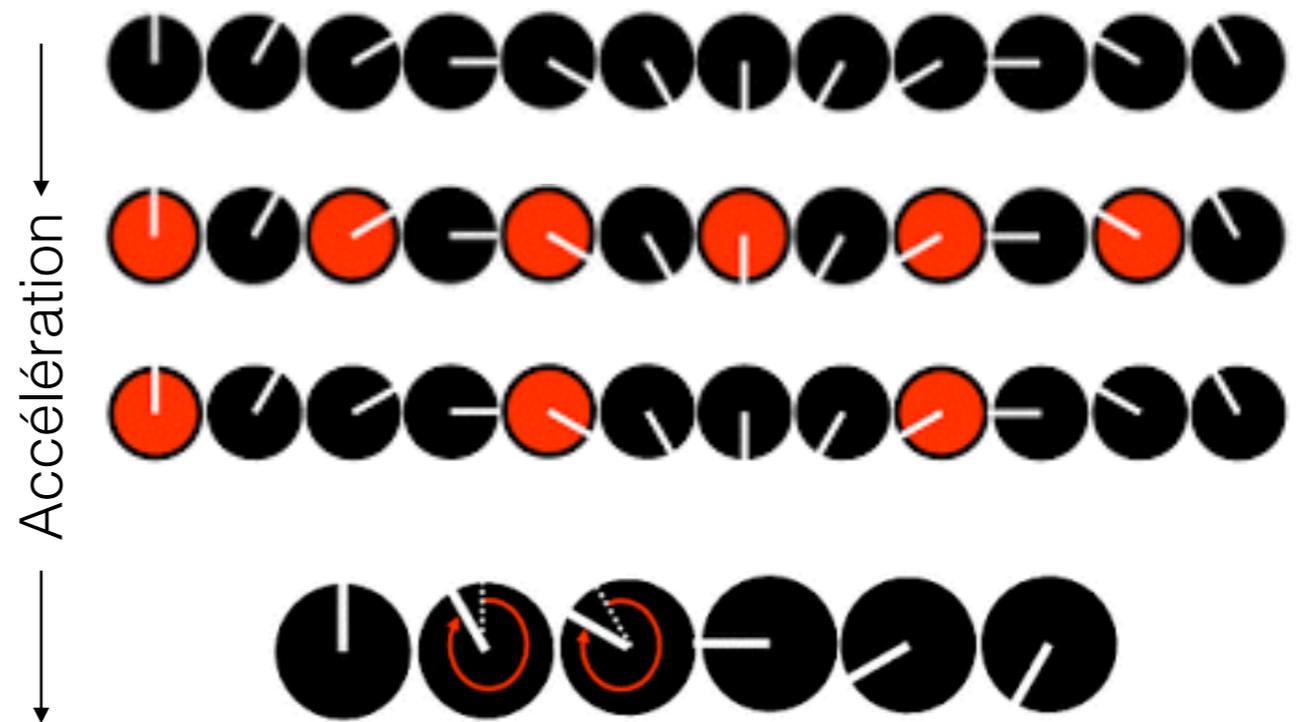
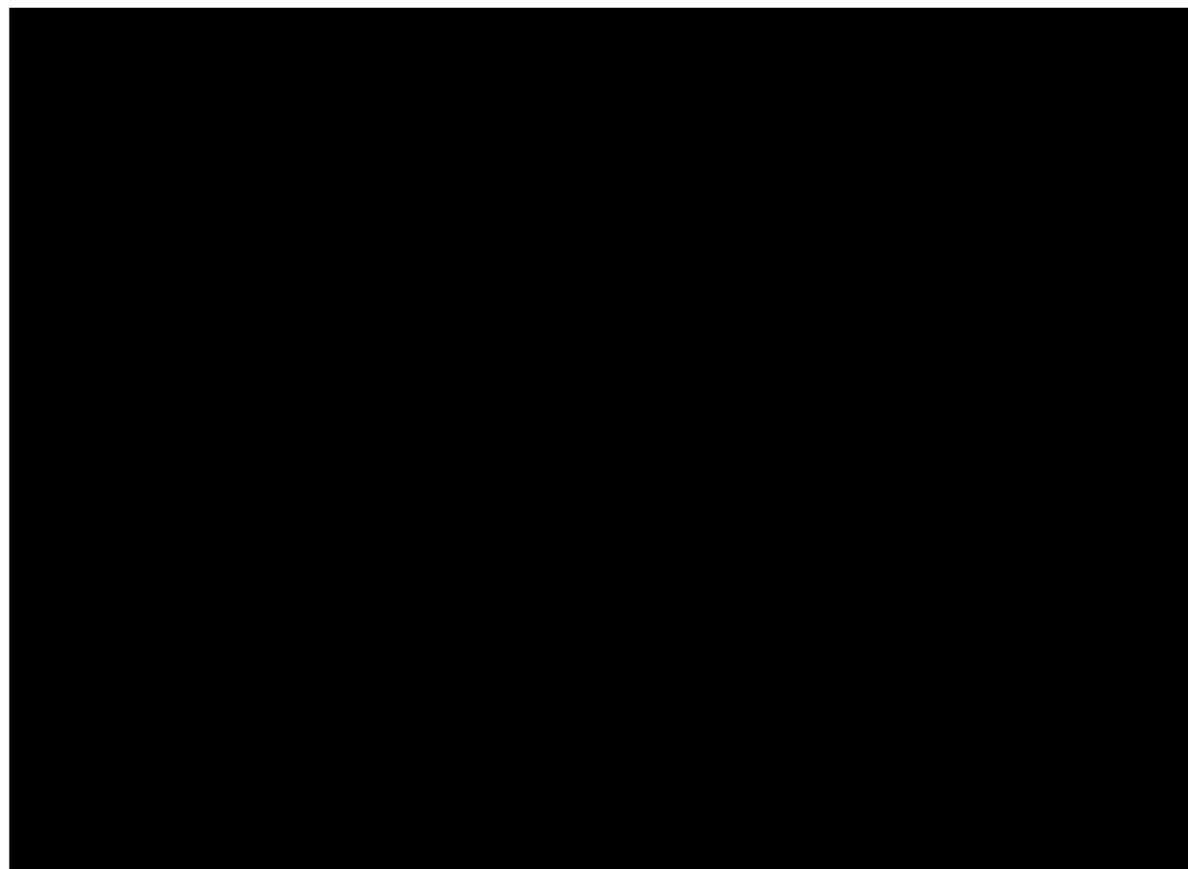
DBZ Animation  
Episode 3

# Quand l'oeil nous joue des tours ...

**Persistance rétinienne = 1/25 secondes**



Fréquence d'affichage au cinéma  
=  
24 images par seconde



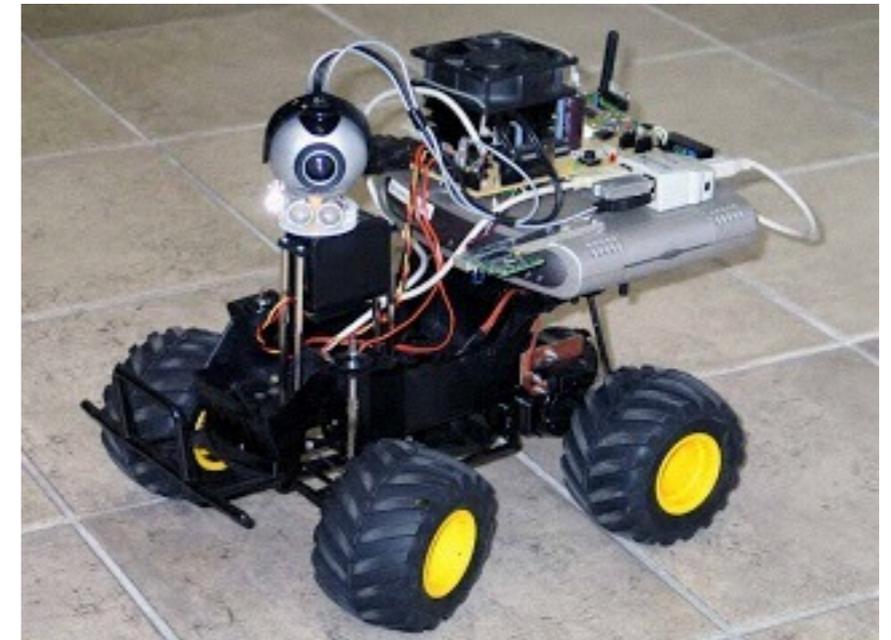
**Impression que la roue recule !**

Illustration de l'effet stroboscopique

# Vision par ordinateur

**Objectif : apprendre aux machines (ou robots)**

- à reconnaître des images dans un flux vidéo
- à analyser et extraire l'information de celles-ci
- à émettre une action en conséquence



# Vision par ordinateur

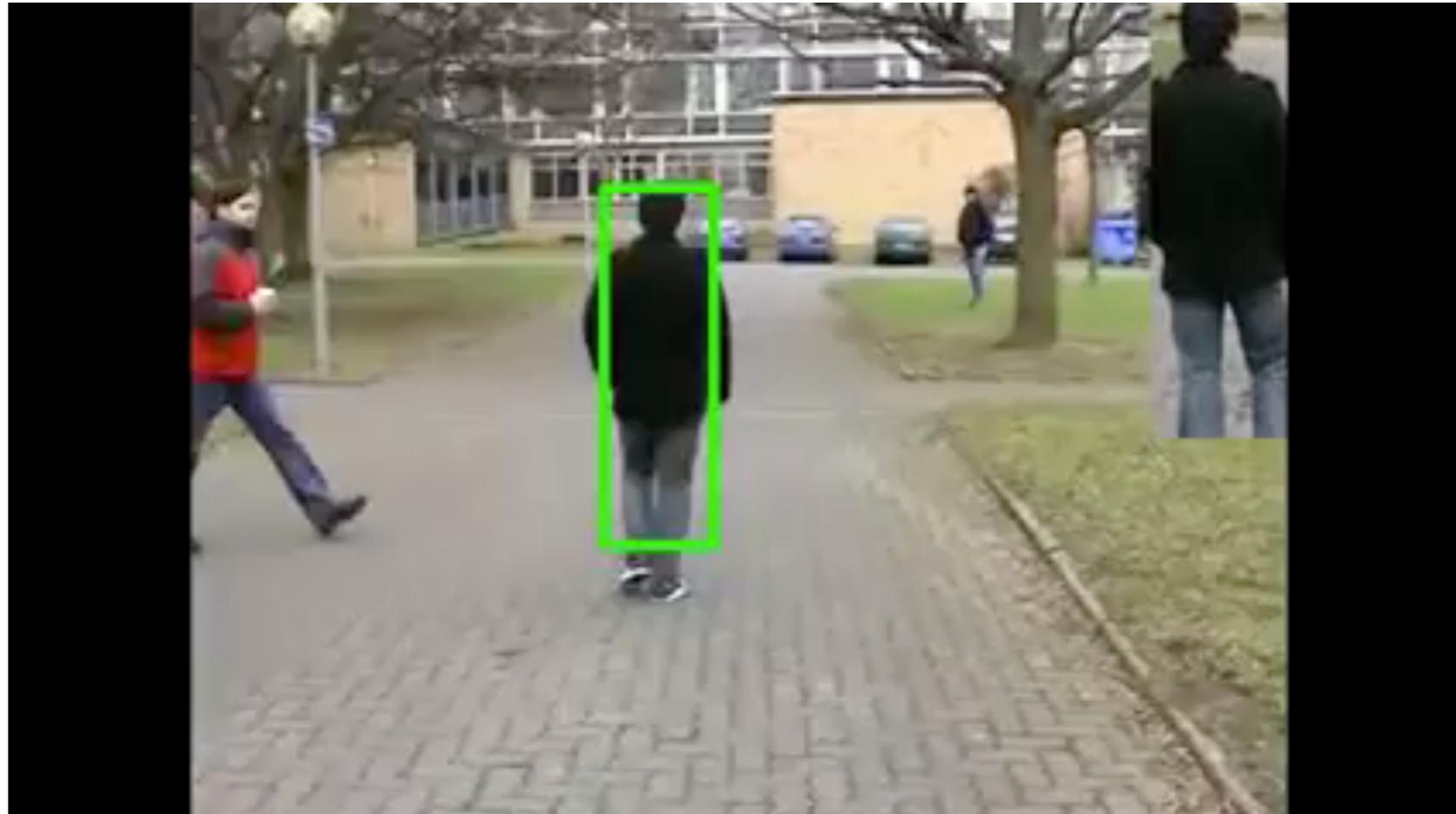
## Exemples d'applications

- ★ Procédés de contrôle, p. ex. dans la robotique industrielle
- ★ Navigation, p. ex. dans un véhicule autonome ou un robot mobile
- ★ Détection d'événements, p. ex. pour la surveillance ou le comptage automatique de personnes
- ★ Organisation d'informations, p. ex. pour indexer des bases de données d'images et de suites d'images
- ★ Modélisation d'objets ou d'environnements, p. ex. dans l'analyse d'image médicale ou la topographie
- ★ Interaction, p. ex. en tant que dispositif d'entrée pour une interaction homme-machine
- ★ Inspection automatique, p. ex. dans des domaines de production

# Vision par ordinateur

## Exemples d'applications

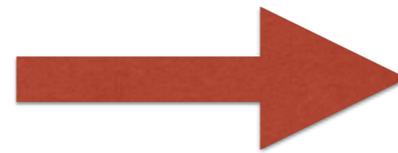
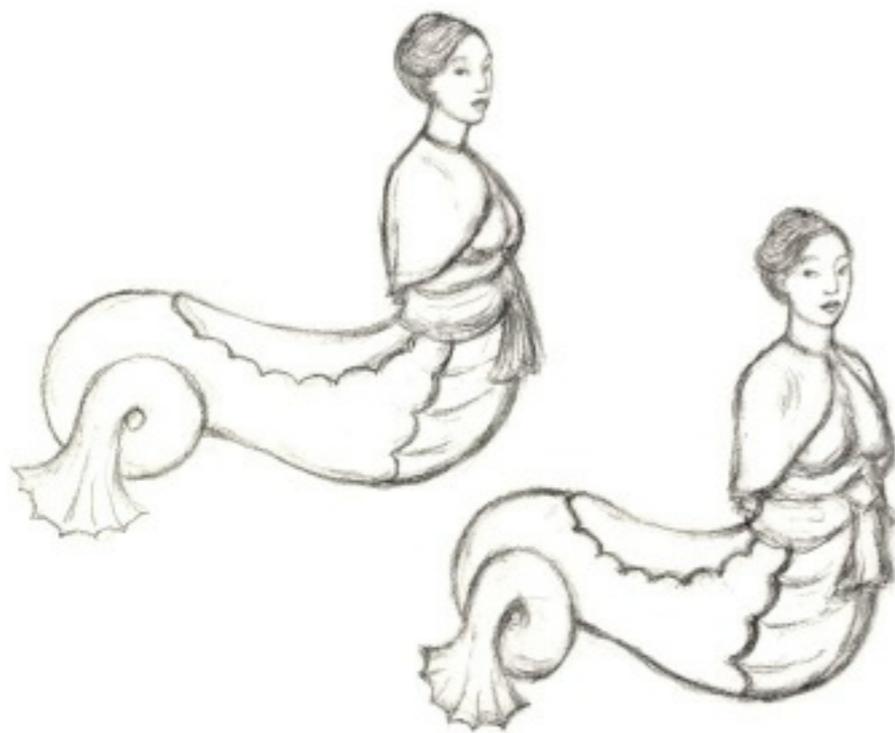
- ★ Procédés de contrôle, p. ex. dans la robotique industrielle
- ★ Navigation, p. ex. dans un véhicule autonome ou un robot mobile
- ★ Détection d'événements, p. ex. pour la surveillance ou le comptage automatique de personnes
- ★ Organisation d'informations, p. ex. pour indexer des bases de données d'images et de suites d'images
- ★ Modélisation d'objets ou d'environnements, p. ex. dans l'analyse d'image médicale ou la topographie
- ★ Interaction, p. ex. en tant que dispositif d'entrée pour une interaction homme-machine
- ★ Inspection automatique, p. ex. dans des domaines de production



# Warping et Morphing

**Définition : Transformation continue d'une image initiale vers une image finale**

Transformation = Warping + Morphing  
(déformation) (fendu)



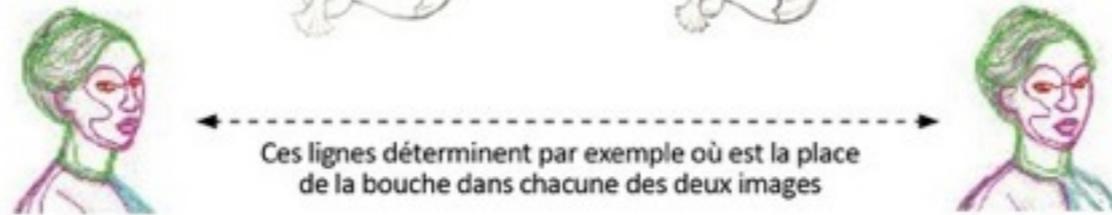
# Warping et Morphing

**Exemple de morphing:**

Un dessinateur crée d'abord deux dessins



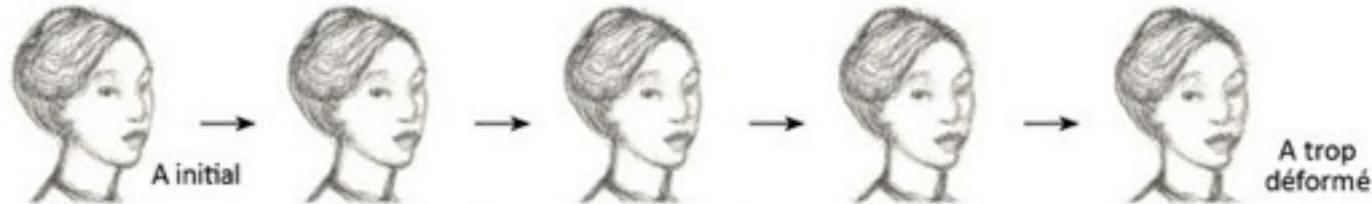
L'animateur pose des contraintes de déformation entre A et B



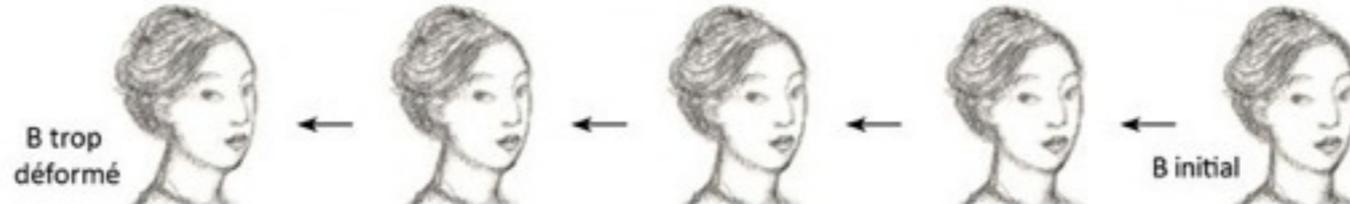
Entre les deux, le logiciel calcule un chemin de transformation continue qui sert de base au warping



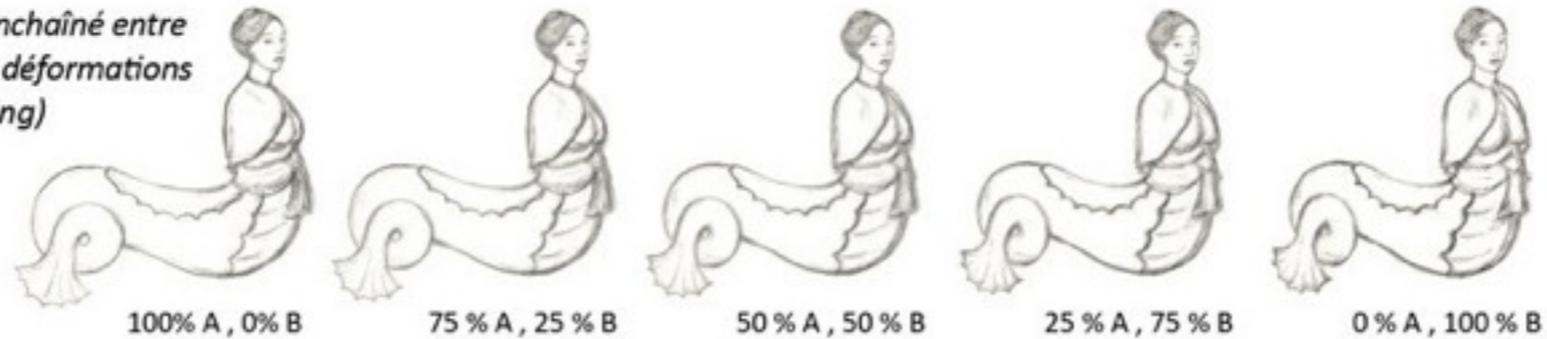
Déformation (warping) du dessin A



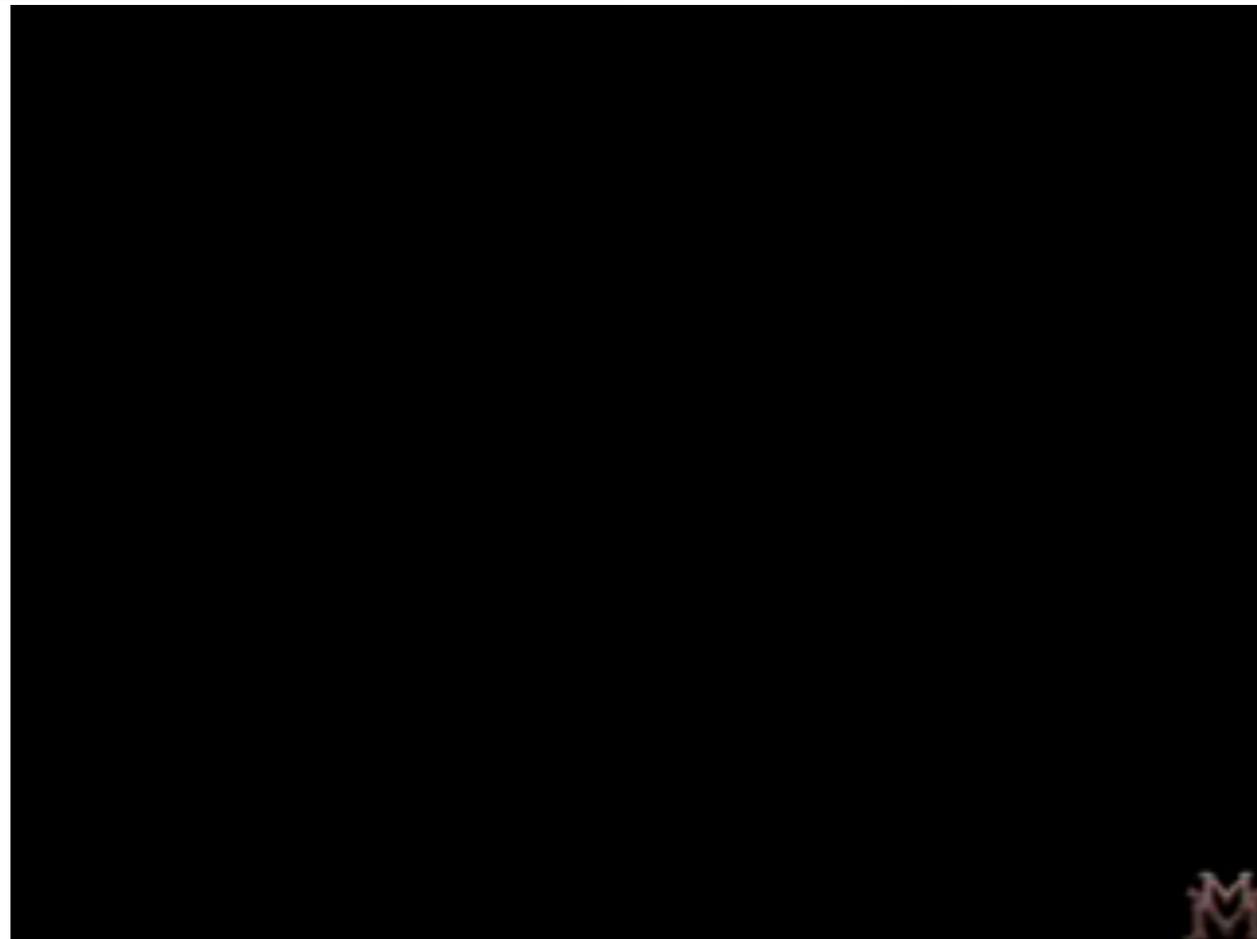
Déformation (warping) du dessin B



Fondu enchaîné entre les deux déformations (morphing)



# Warping et Morphing



# Comment perçoit-on la 3D ?

**Alignez votre index  
le long de cette barre**



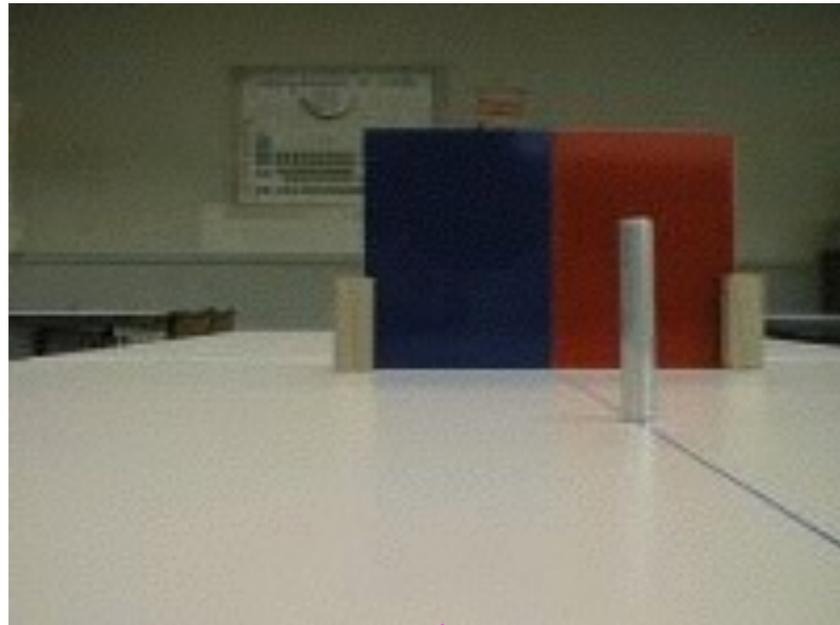
**Et fermez simultanément  
l'oeil droit puis l'oeil gauche**



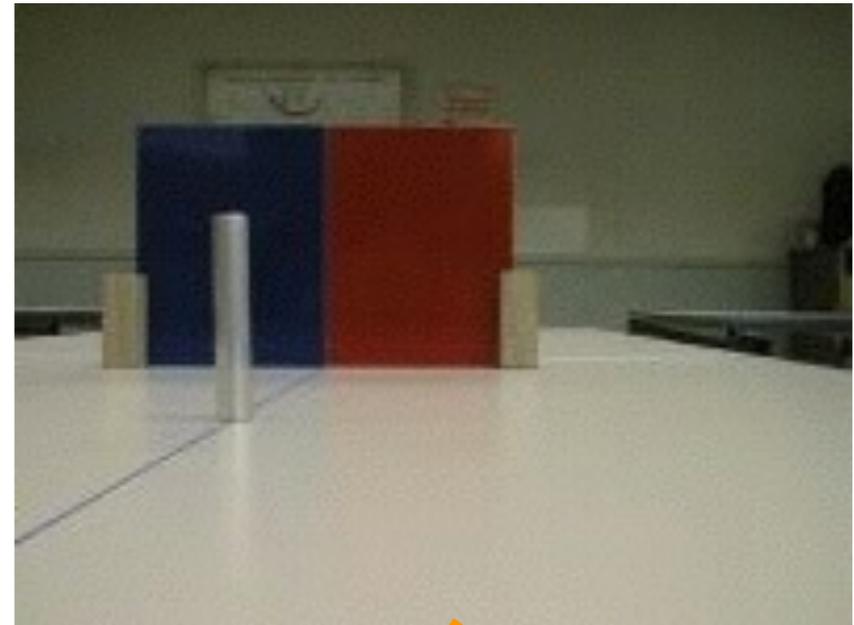
Que constatez-vous ?

# Comment perçoit-on la 3D ?

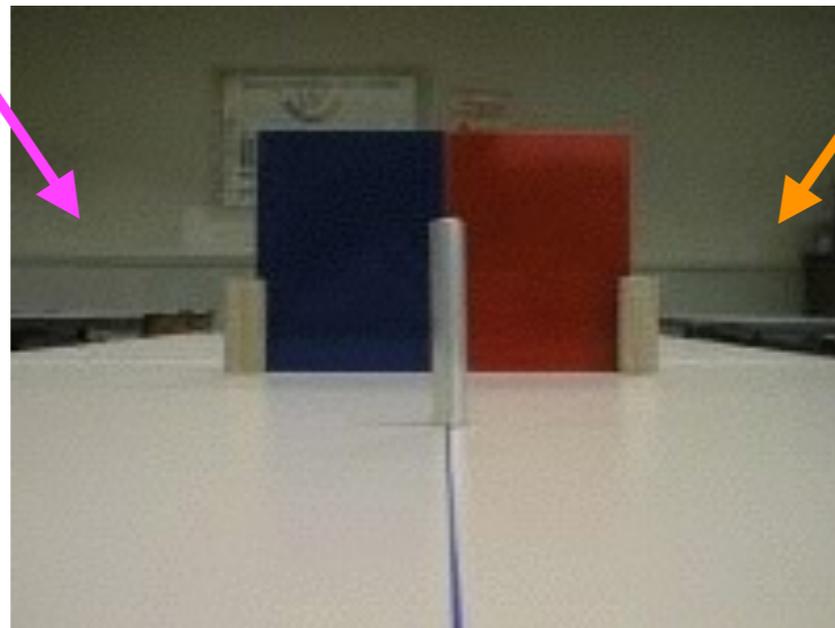
Vue de l'oeil gauche



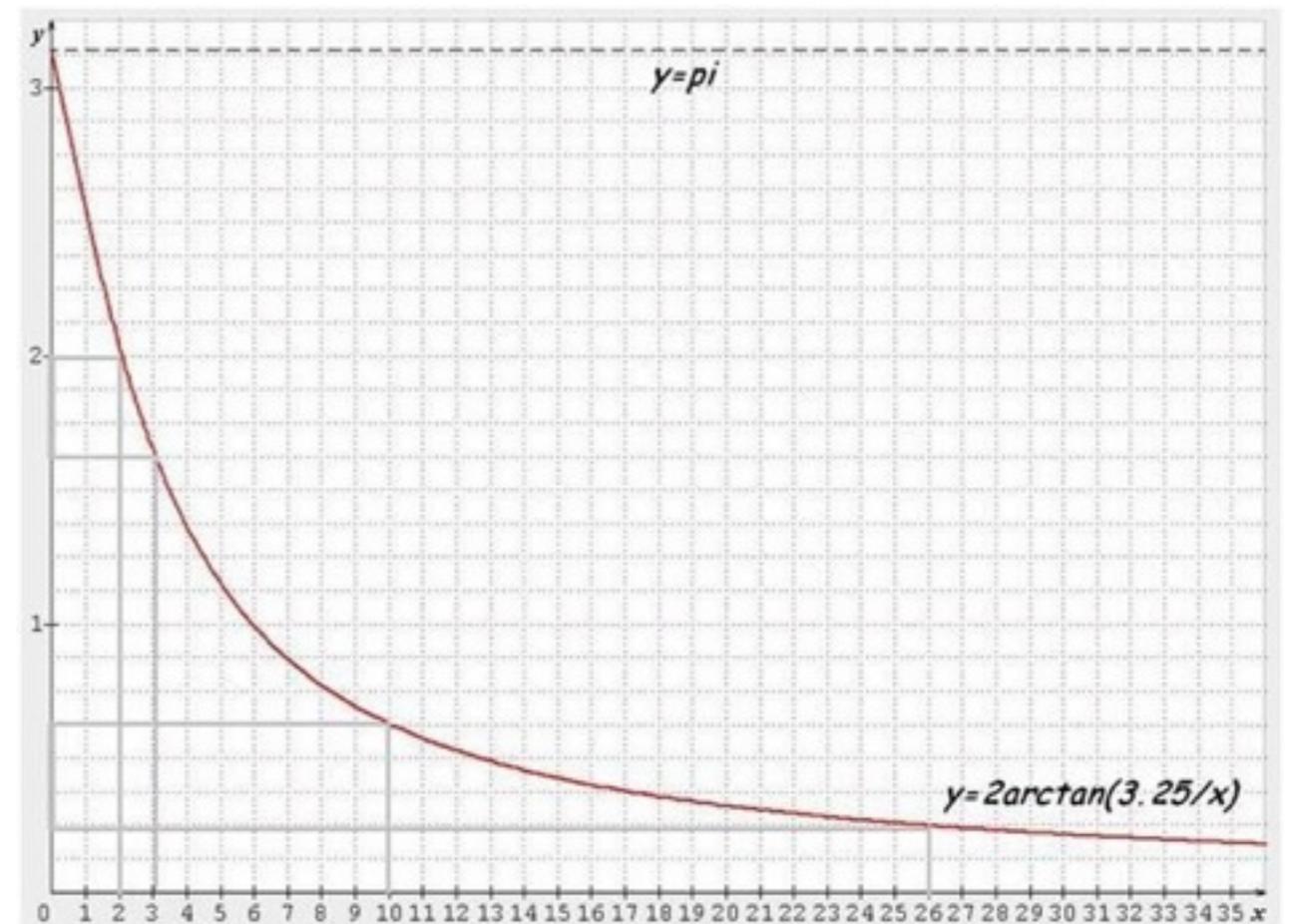
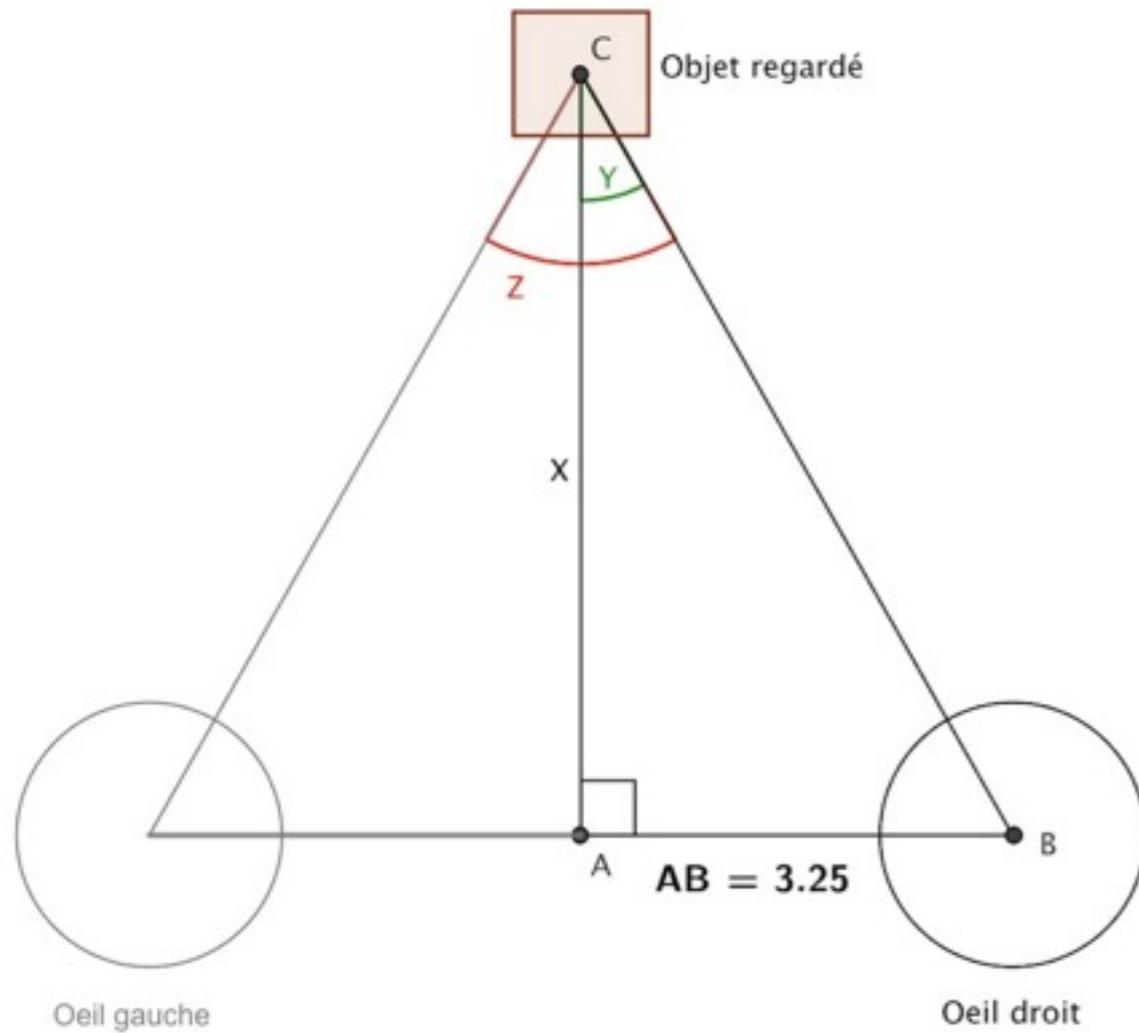
Vue de l'oeil droit



Vue des 2 yeux



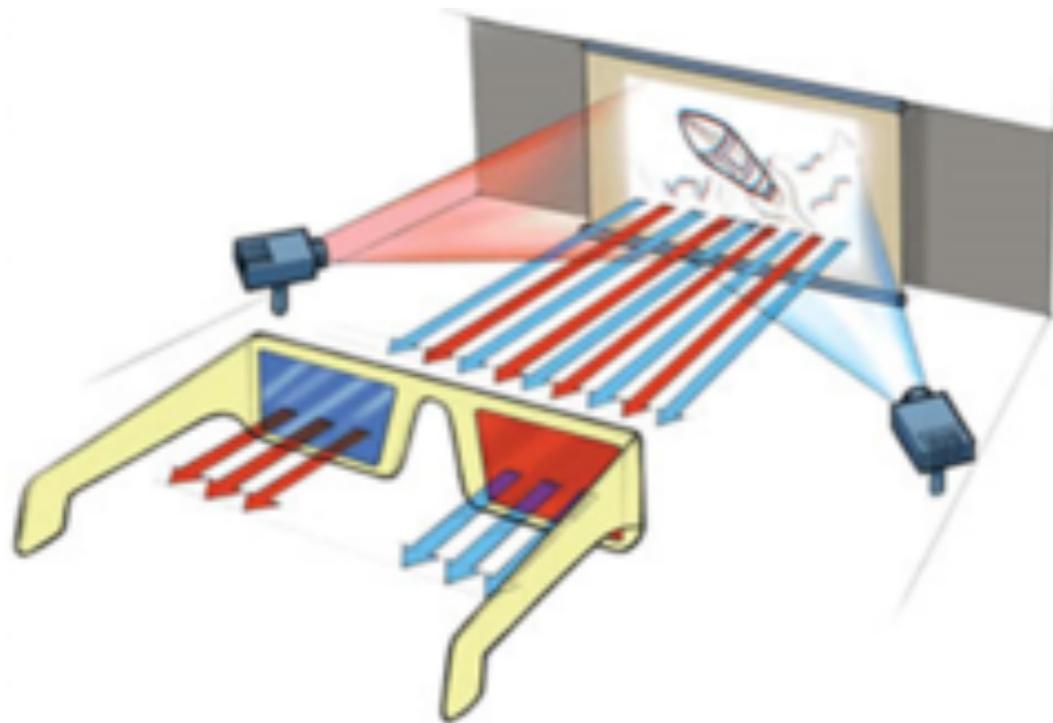
# Comment perçoit-on la 3D ?



**L'angle de parallaxe  $z$  varie en fonction de la profondeur de l'objet observé et participe à la vision en 3D**

# Le cinéma 3D recrée le principe de la vision binoculaire

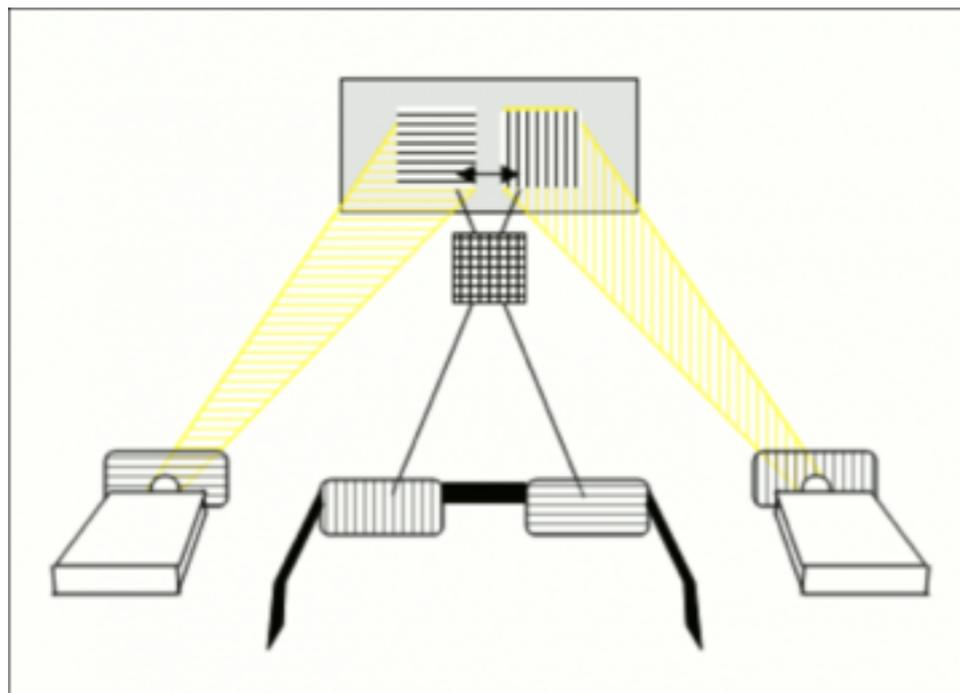
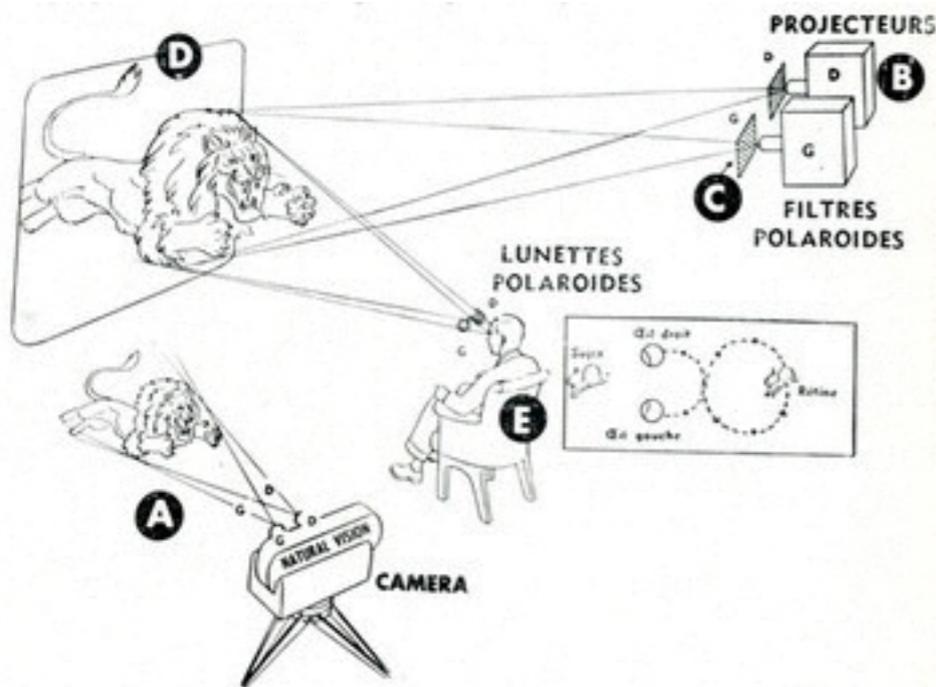
## ■ Projection **anaglyphe**



Les images **rouge** et **cyan** sont « décalées »  
Un **filtre** est disposé devant chaque oeil  
pour laisser passer la **couleur complémentaire**  
Le cerveau fusionne enfin les deux images

# Le cinéma 3D recrée le principe de la vision binoculaire

## ■ Projection polarisée passive

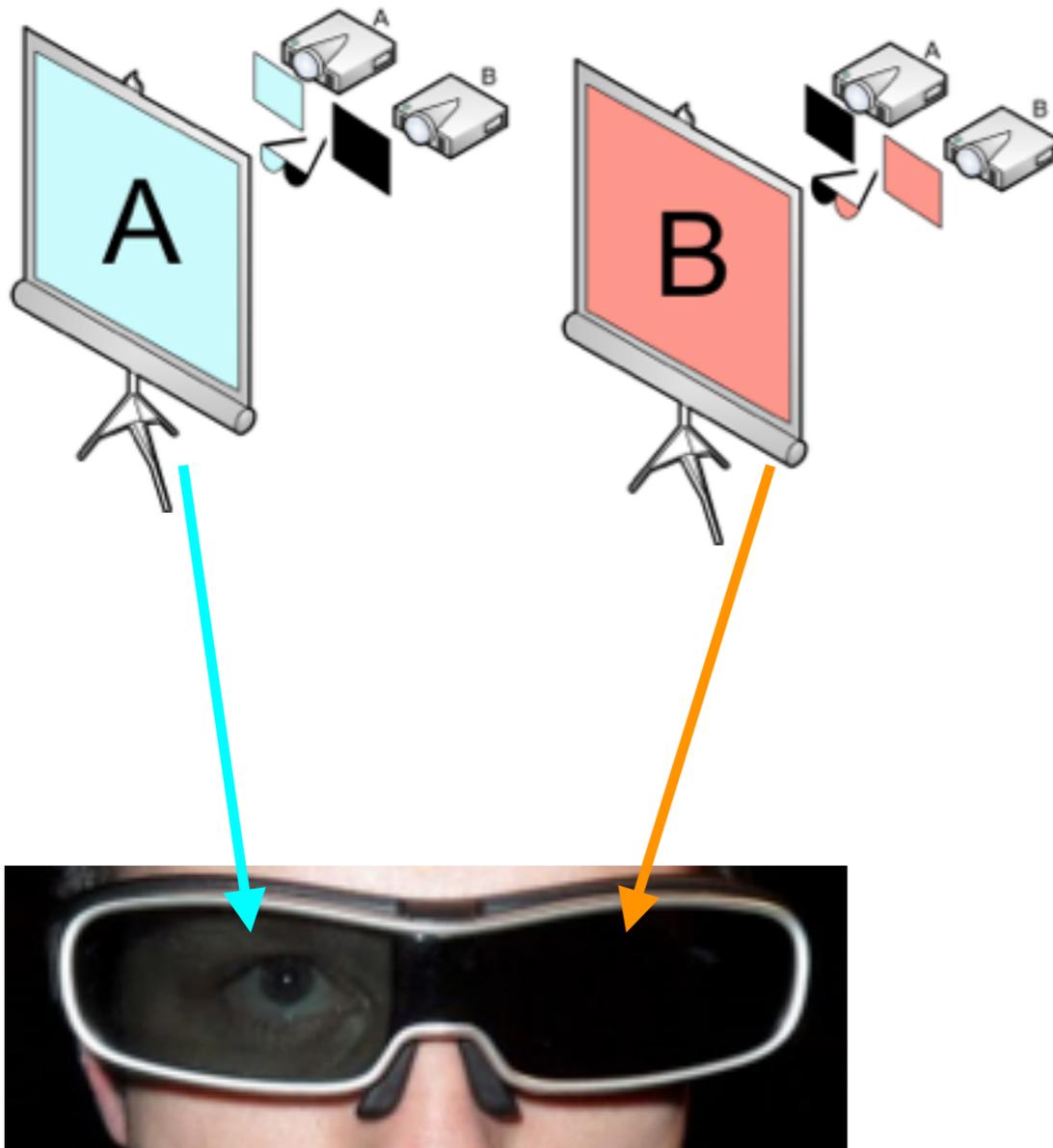


## Etapes de la reconstitution 3D

- **A** : 2 caméras espacées de 6,5cm filment la même scène
- **B,C** : 2 projecteurs projettent les 2 films avec une polarisation différente via les filtres polaroïdes
- **D** : l'écran métallisé réfléchit les ondes polarisées vers le spectateur
- **E** : Les verres des lunettes polarisées du spectateur filtre les ondes correspondantes et le cerveau reconstitue la scène 3D.

# Le cinéma 3D recrée le principe de la vision binoculaire

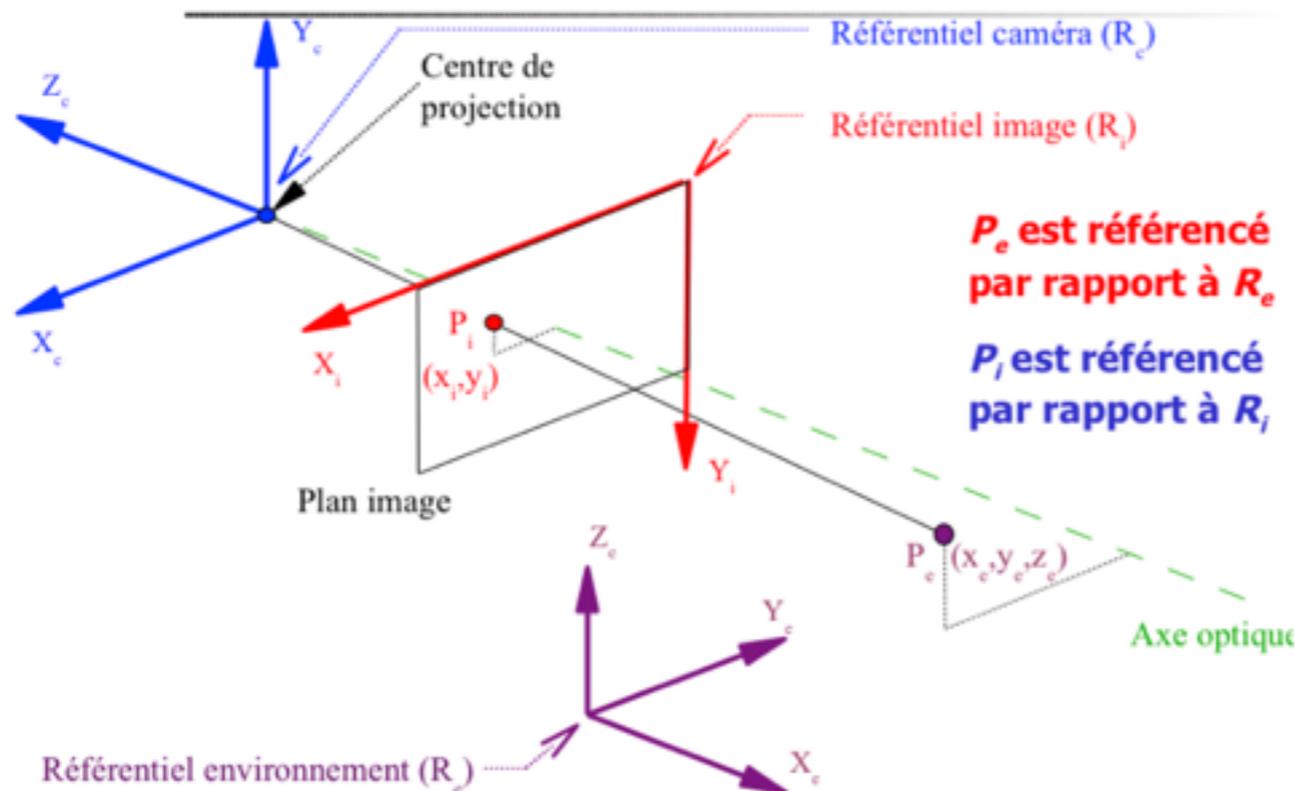
## ■ Projection **alternée active**



## Etapes de la reconstitution 3D

- L'écran reçoit alternativement une image de chaque projecteur A et B à une fréquence de 120 Hz.
- Les verres des lunettes du spectateur s'opacifie également alternativement de manière synchronisée avec les projecteurs.
- L'oeil droit et l'oeil gauche voient donc respectivement les images provenant des projecteurs A et B

# Du monde réel 3D à l'image 2D



Source : Jacques-André Landry, Vision robotique, Ecole de Technologie Supérieure (Montréal, Canada)

Trois systèmes de coordonnées distincts

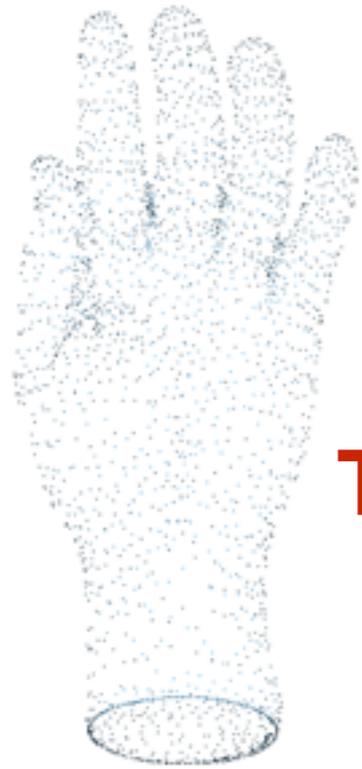
- ▶ l'environnement (ou le monde ou la scène)  $R_e$ 
  - ▶ système de réfère connu. Tout (caméra, objet, etc. ) est défini par rapport à ce système de coordonnées
- ▶ la caméra  $R_c$ 
  - ▶ système dans lequel s'effectue la projection des points 3D vers 2D
- ▶ l'image  $R_i$ 
  - ▶ système établissant la référence aux pixels

$$P_e(x_e, y_e, z_e) \rightarrow P_c(x_c, y_c, z_c) \rightarrow P_i(x_i, y_i)$$

scène  $\rightarrow$  caméra  $\rightarrow$  image

$$\begin{bmatrix} x_i \cdot w \\ y_i \cdot w \\ w \end{bmatrix} = \begin{bmatrix} k_x & 0 & u_o \\ 0 & k_y & v_o \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1/f & 0 \end{bmatrix} \cdot \begin{bmatrix} R & T \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_e \\ y_e \\ z_e \\ 1 \end{bmatrix}$$

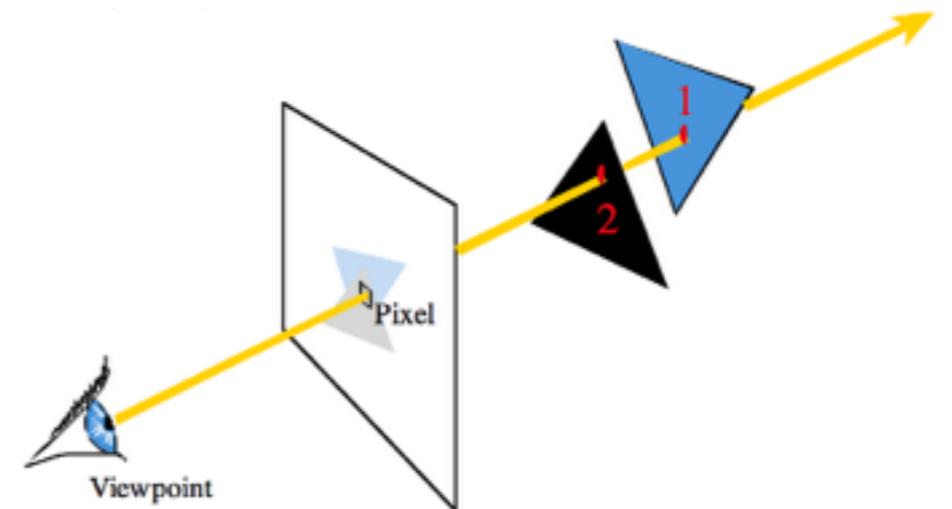
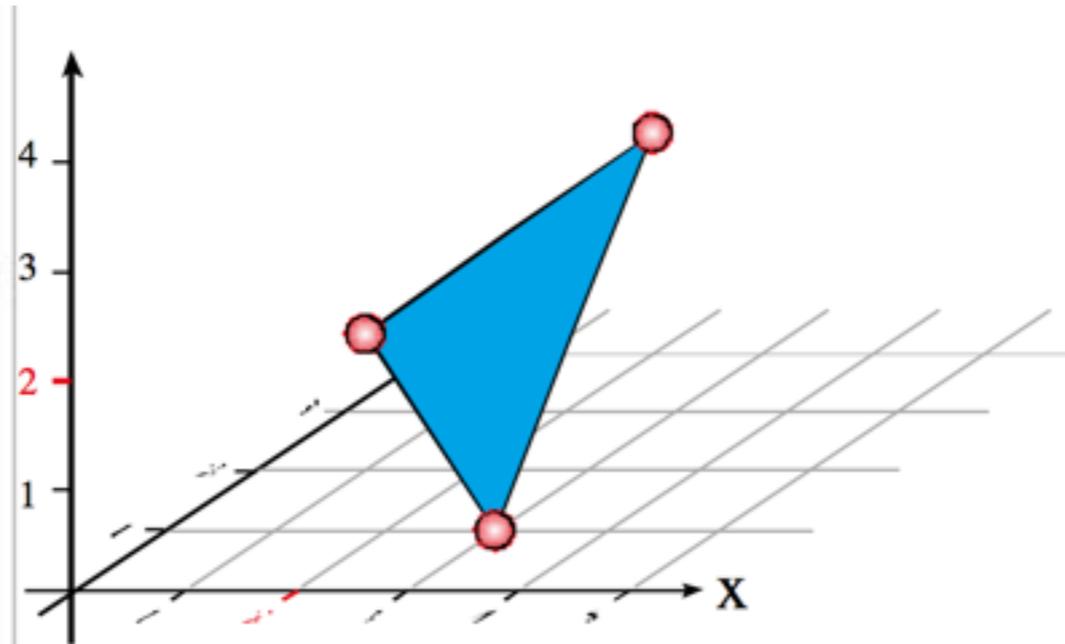
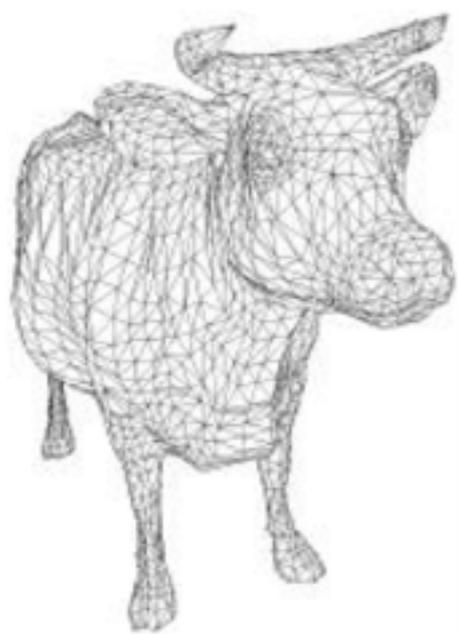
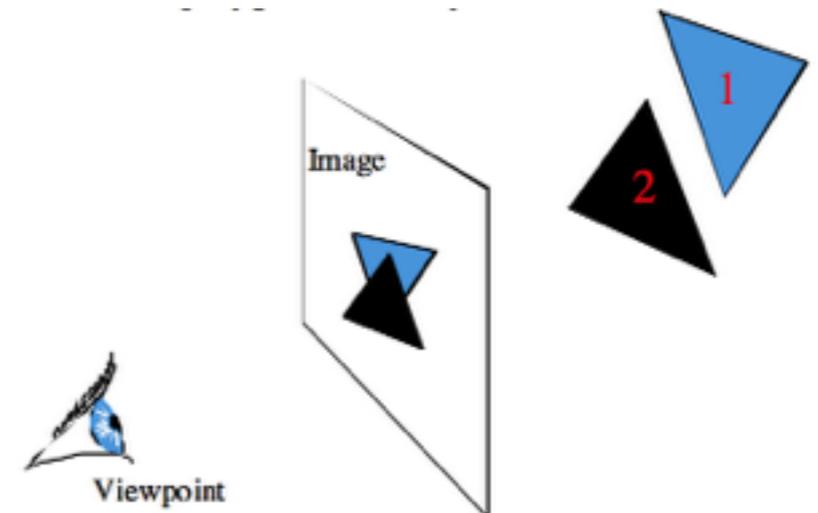
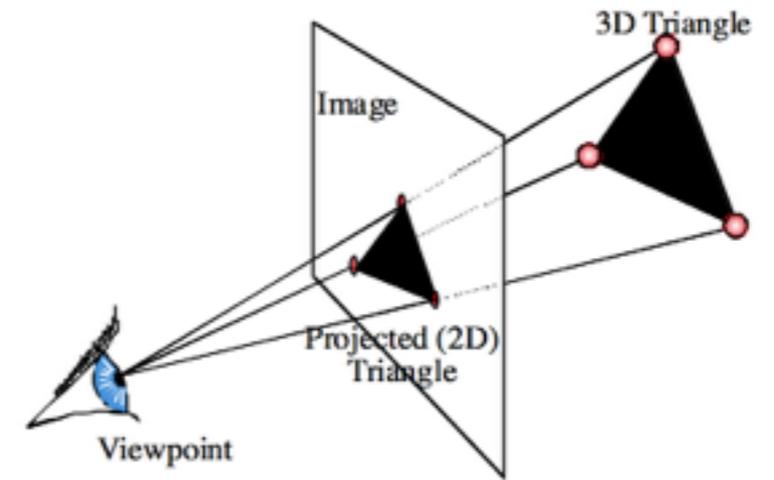
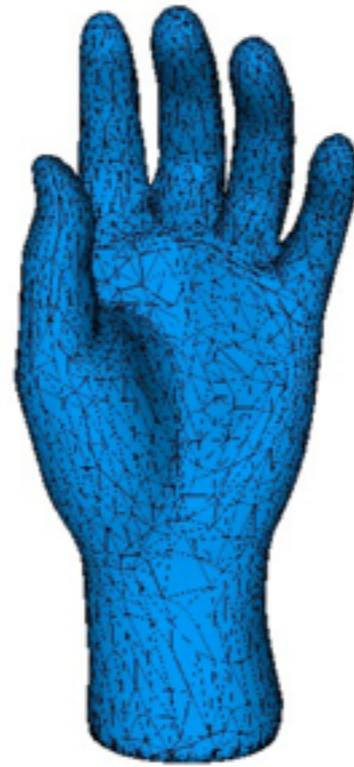
# Création du monde virtuel 3D



**Maillage**



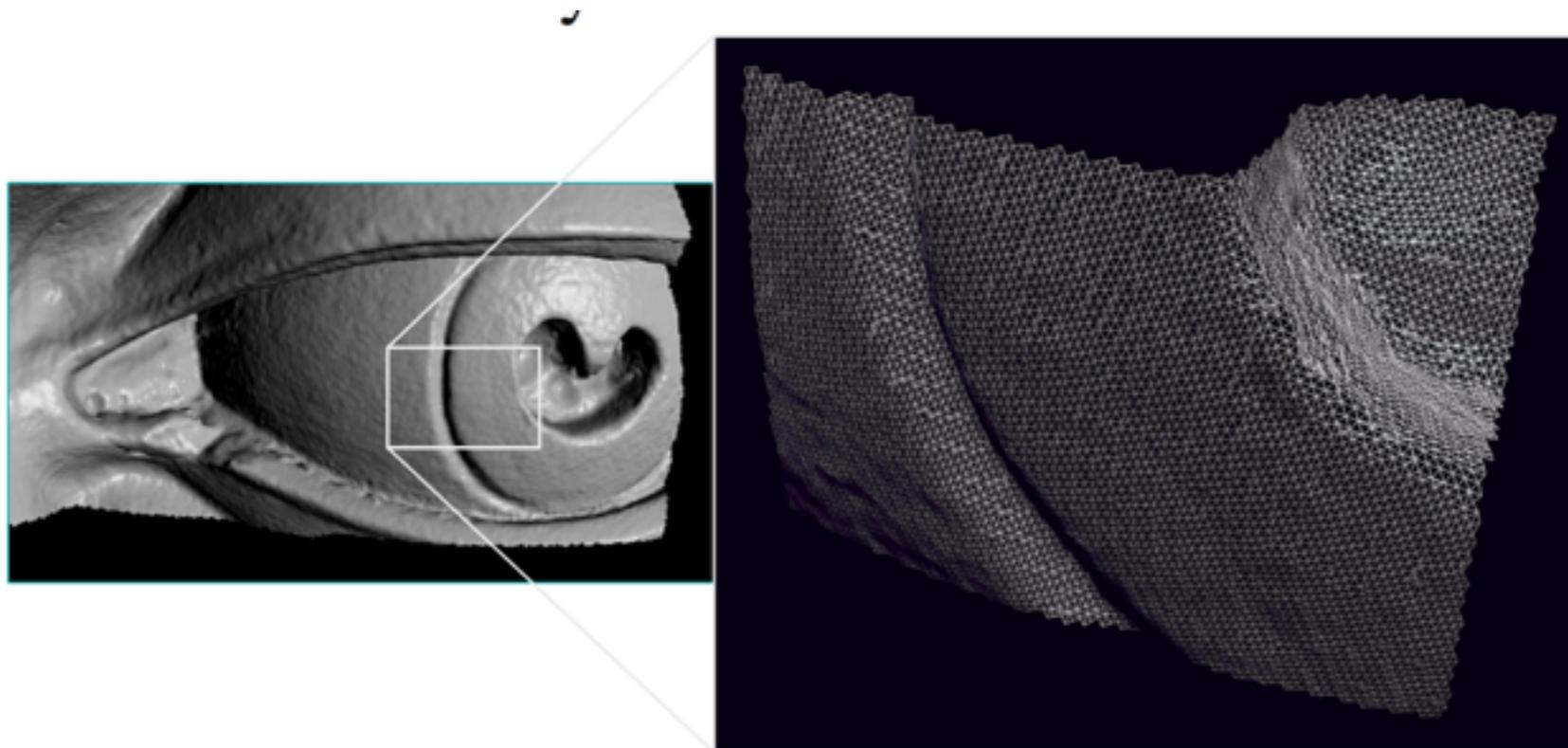
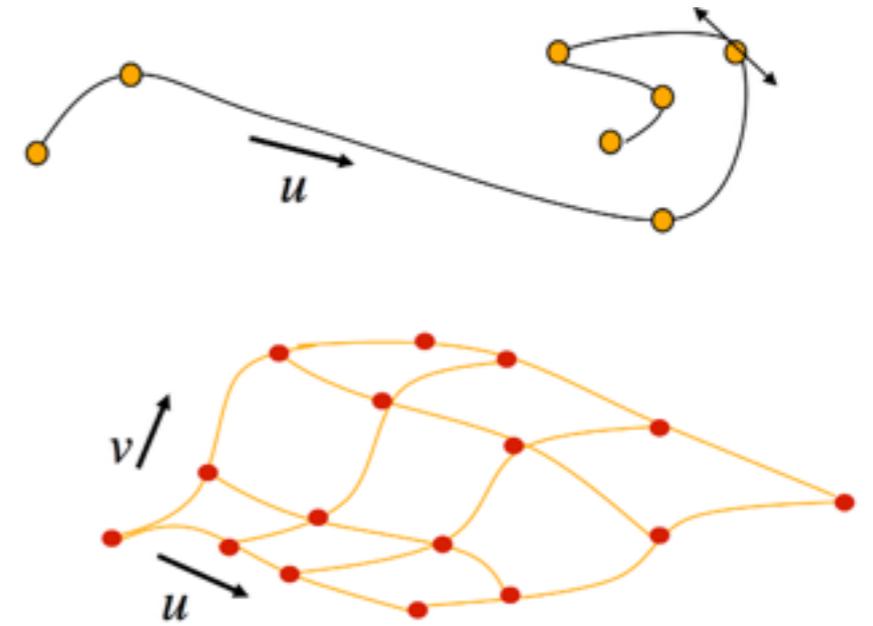
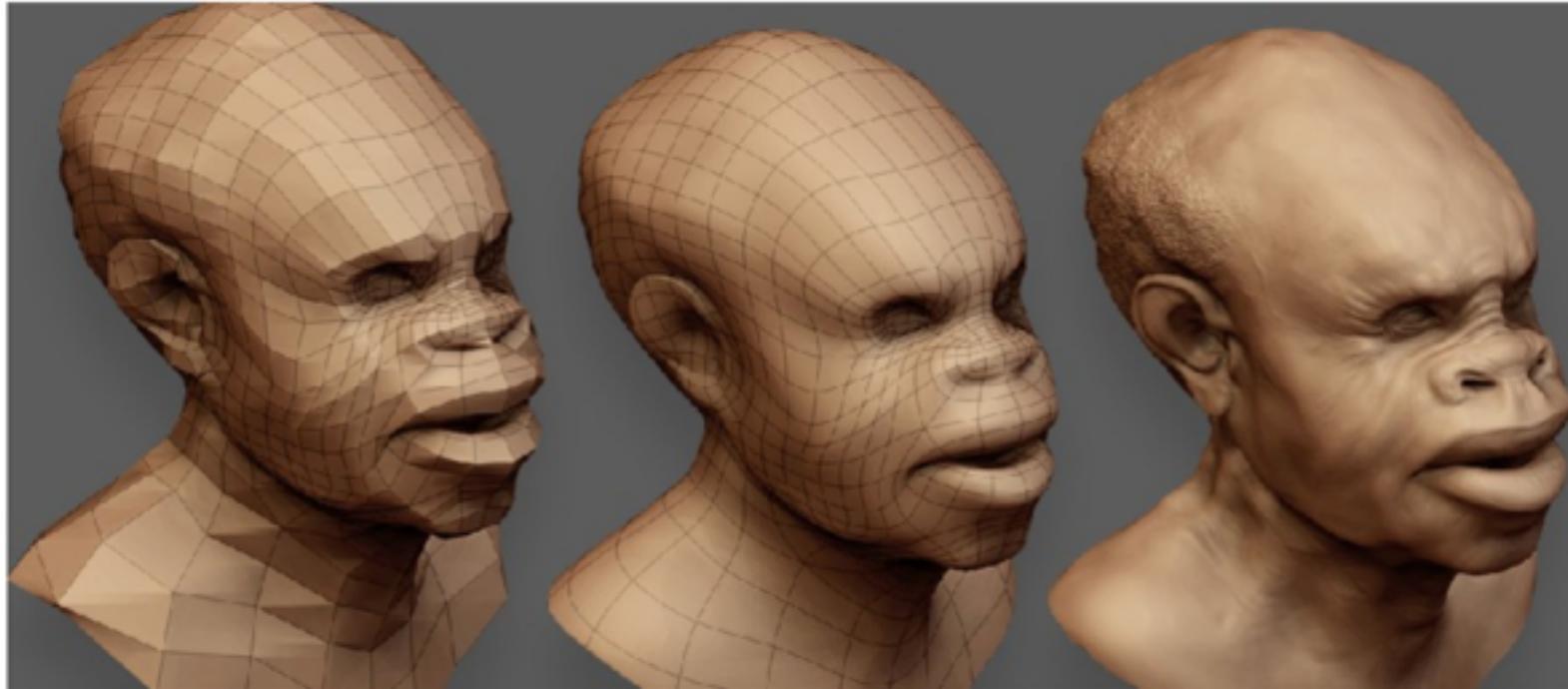
**Triangulation**



# Création du monde virtuel 3D



# Création du monde virtuel 3D



**Subdivision  
+  
Lissage  
=  
Graphismes  
réalistes**

# Création du monde virtuel 3D

Texture

Lumière

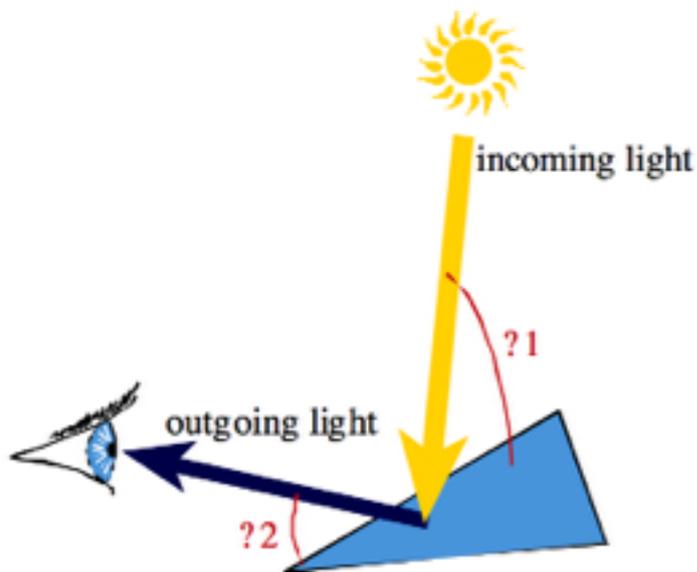
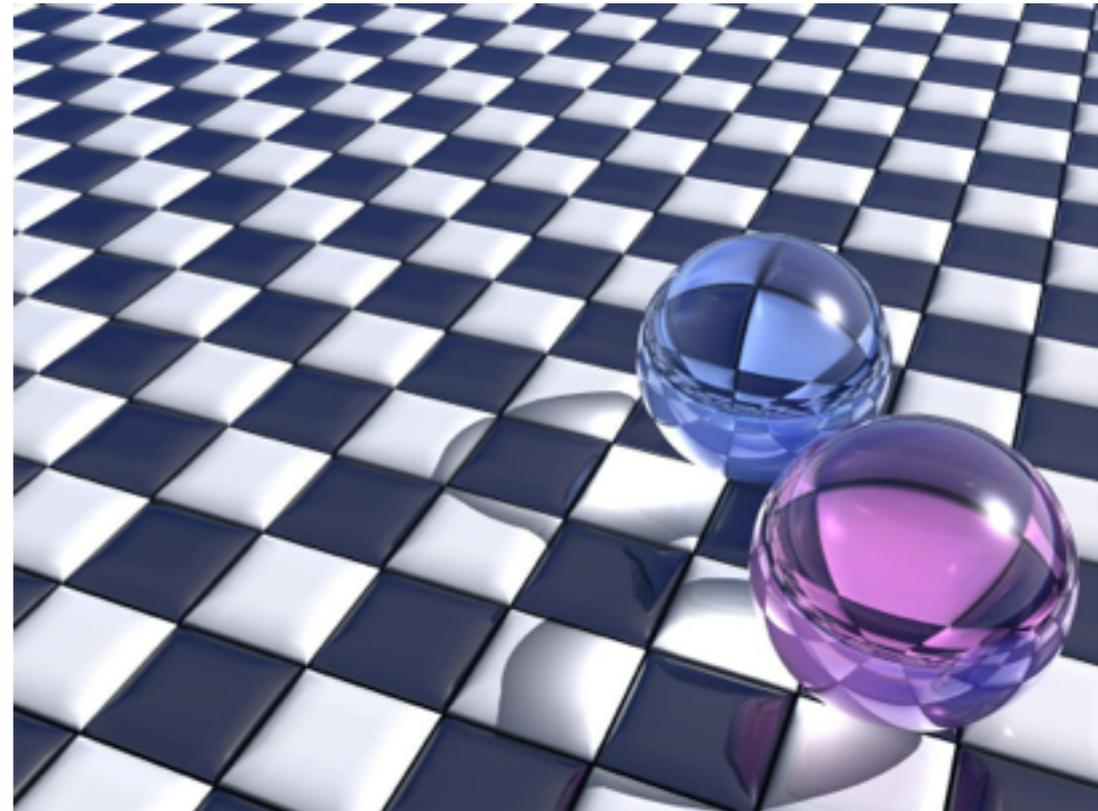
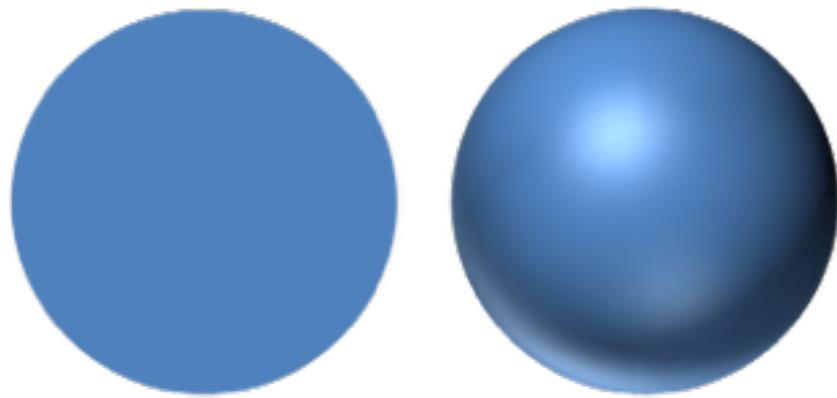


Ombres

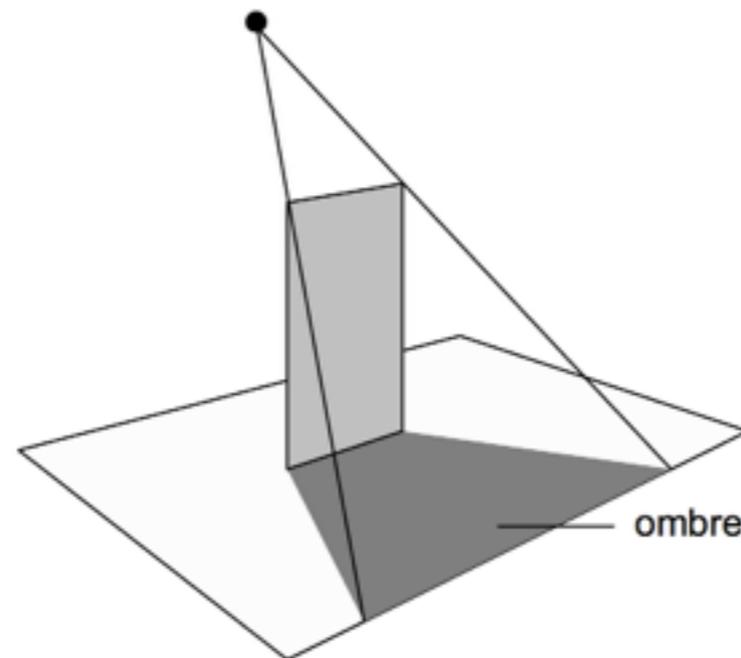


# Création du monde virtuel 3D

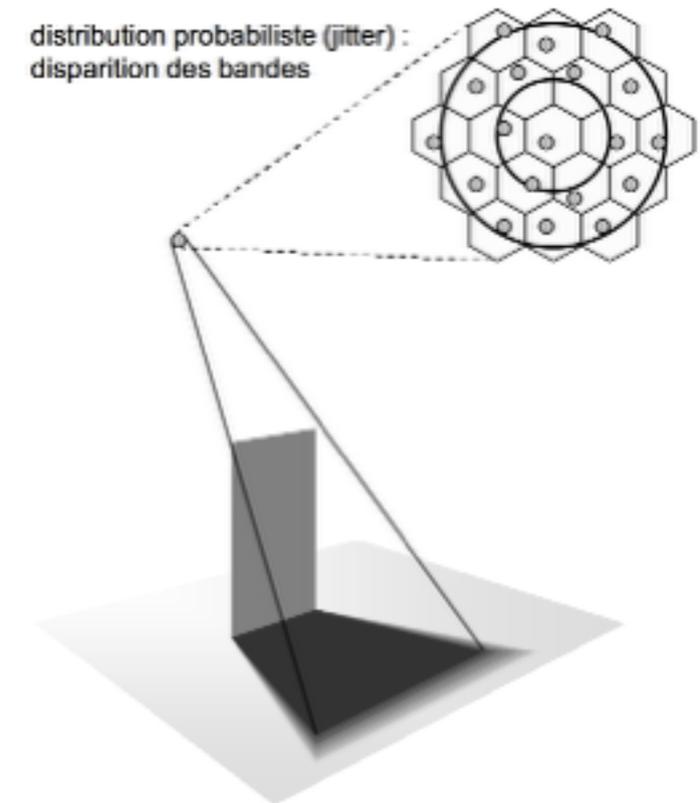
## Luminosité et 3D



lumière ponctuelle



distribution probabiliste (jitter) :  
disparition des bandes



# Création du monde virtuel 3D

## Texture et mapping

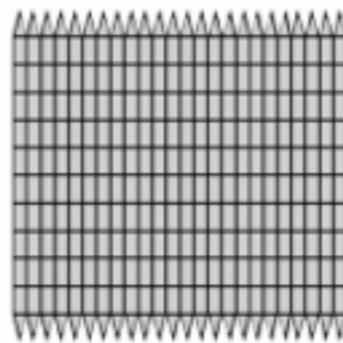
**3-D Model**



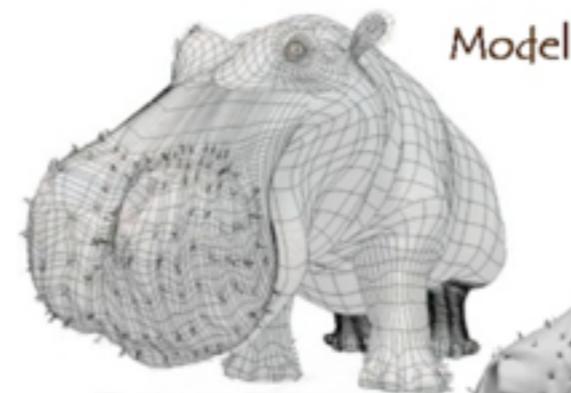
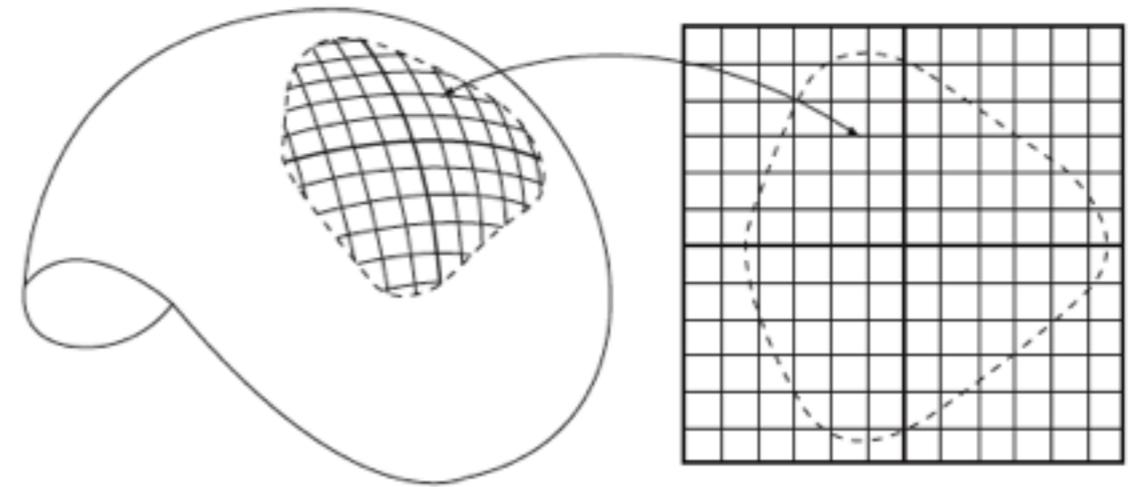
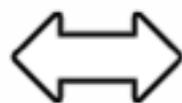
$$p = (x, y, z)$$



**UV Map**



$$p = (u, v)$$



Model + Shading



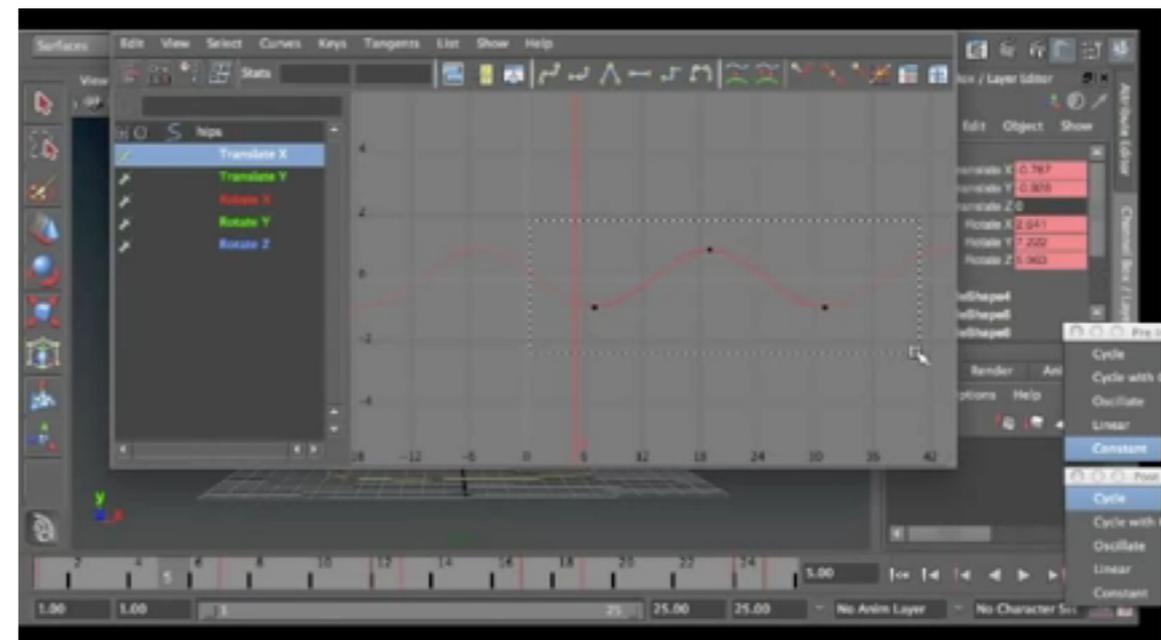
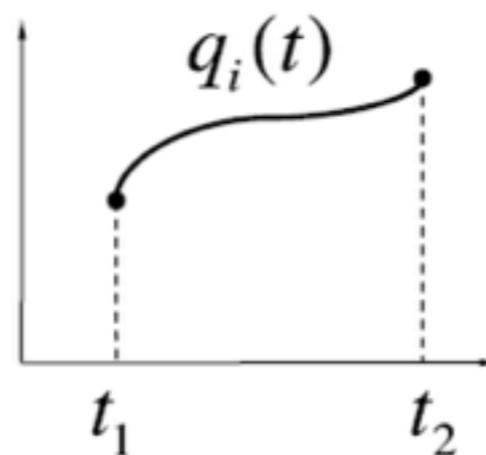
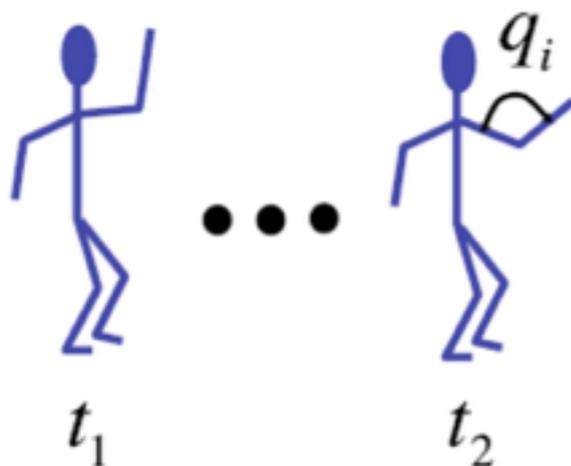
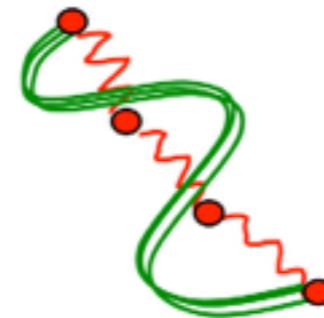
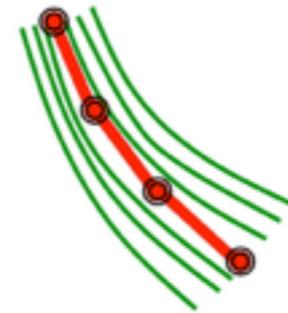
Model + Shading + Textures



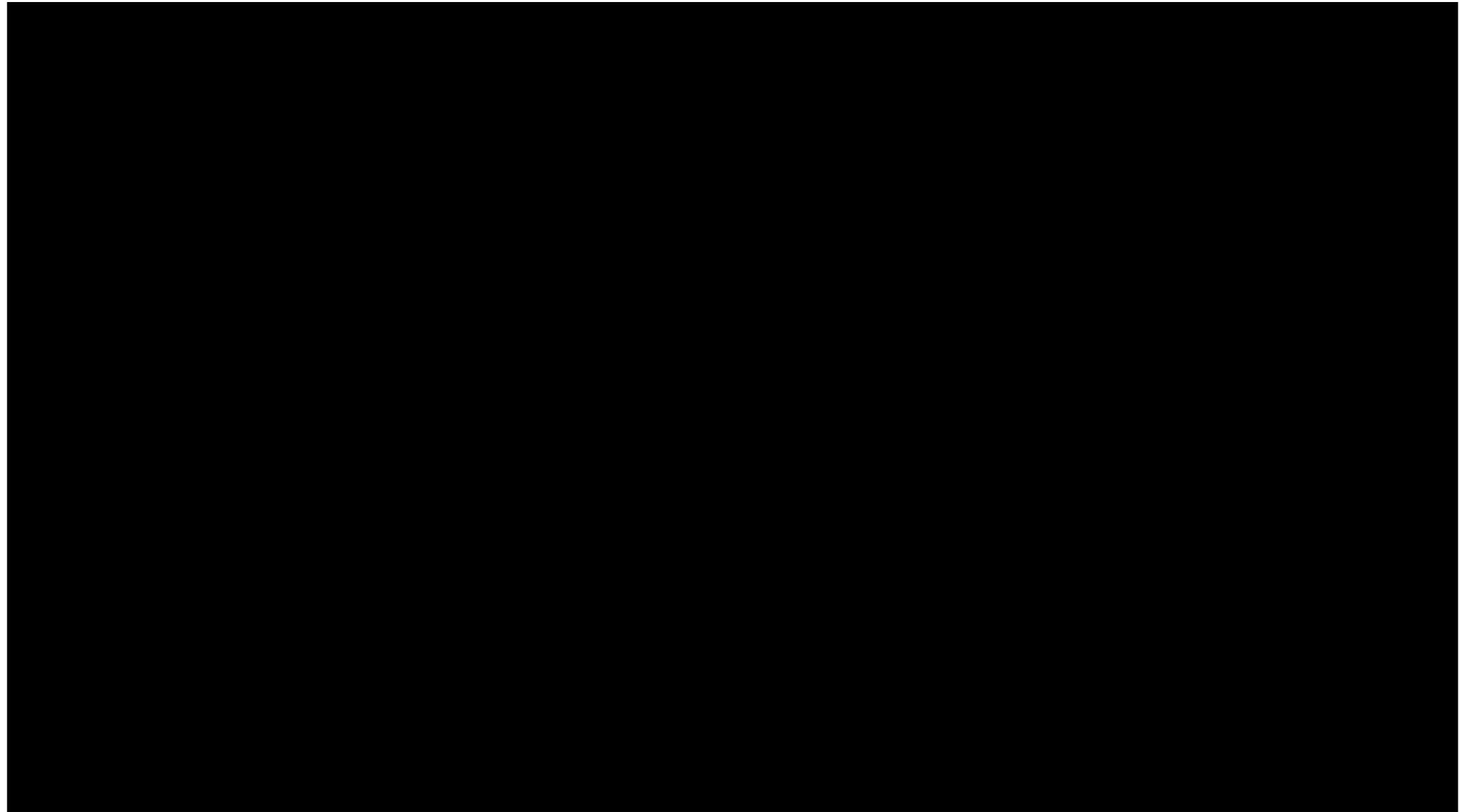
At what point do things start looking real?

For more info on the computer artwork of Jeremy Birn see <http://www.3drender.com/jbirn/productions.html>

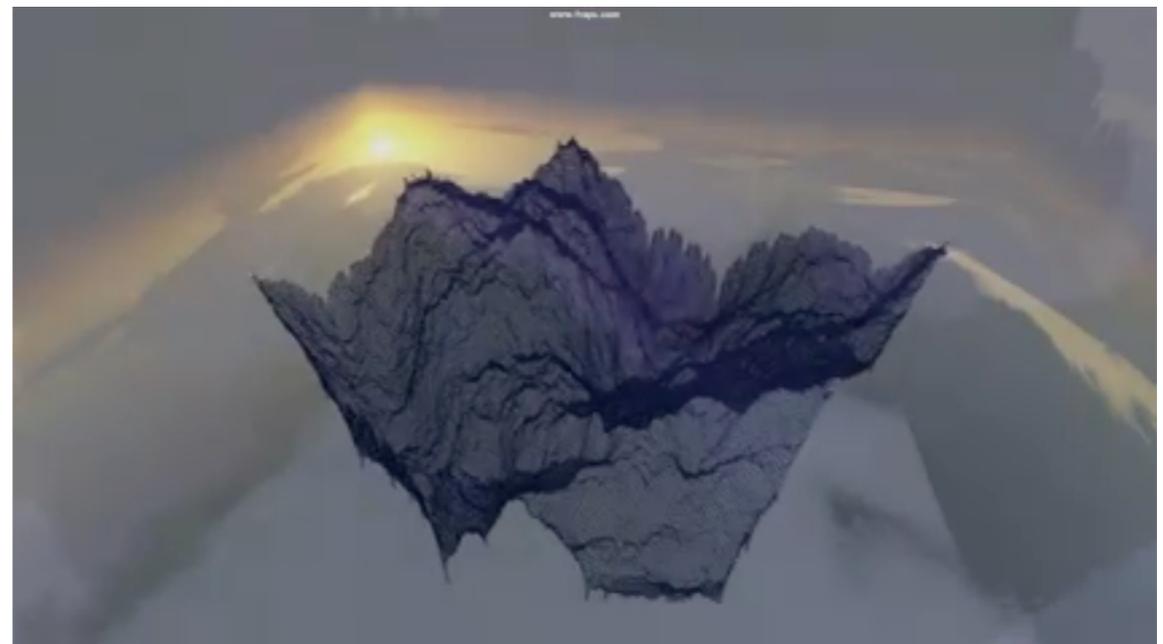
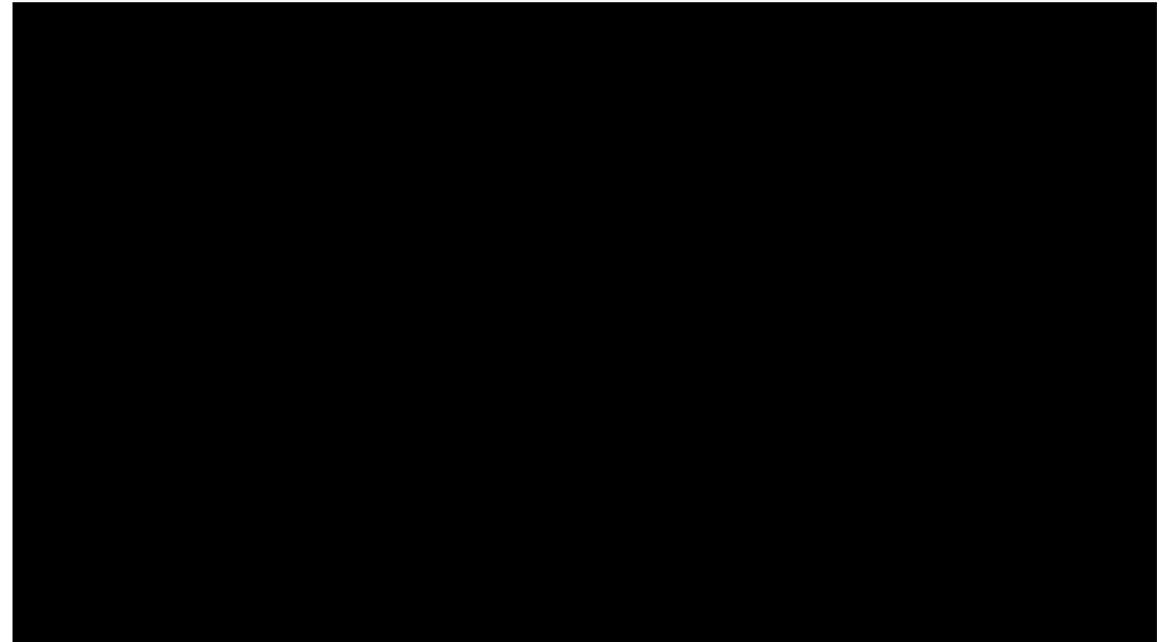
# Animation de la scène 3D



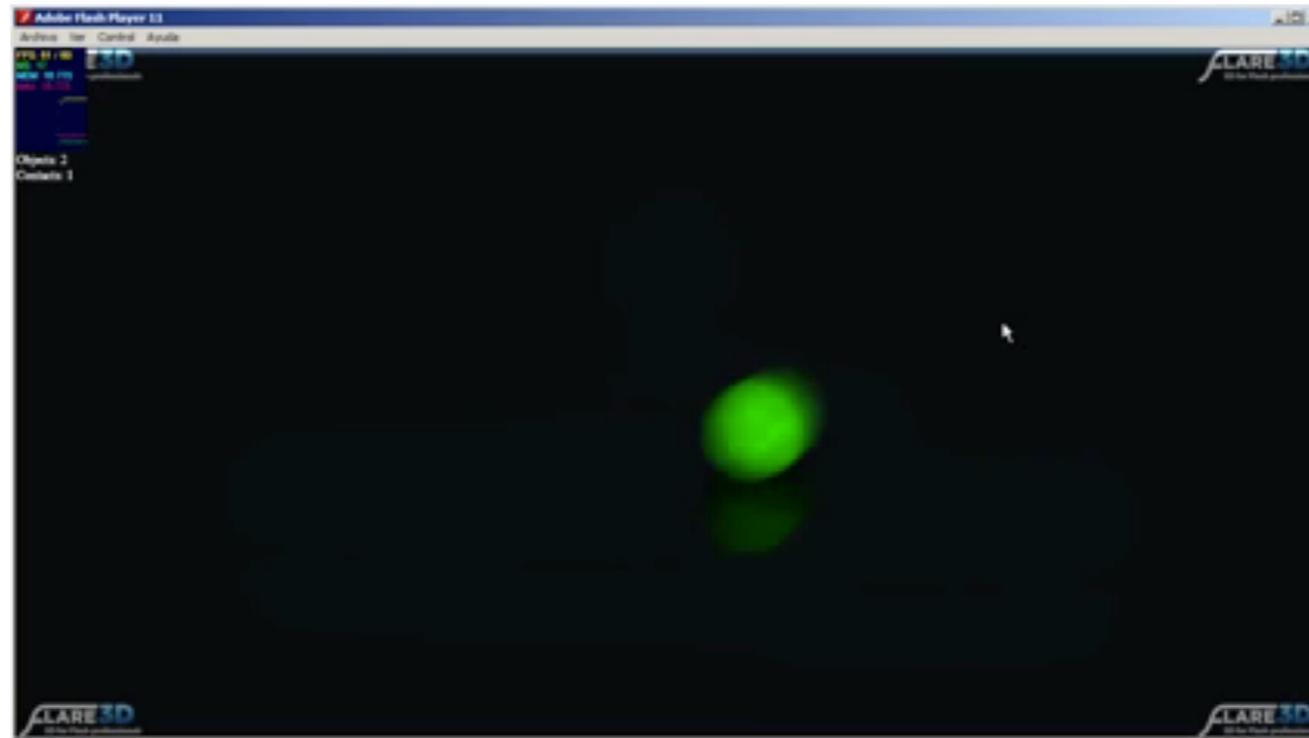
# Développement 3D d'une créature de jeux vidéo



# Animation du décor par simulation de modèles physiques



# Interaction des objets dans la scène 3D



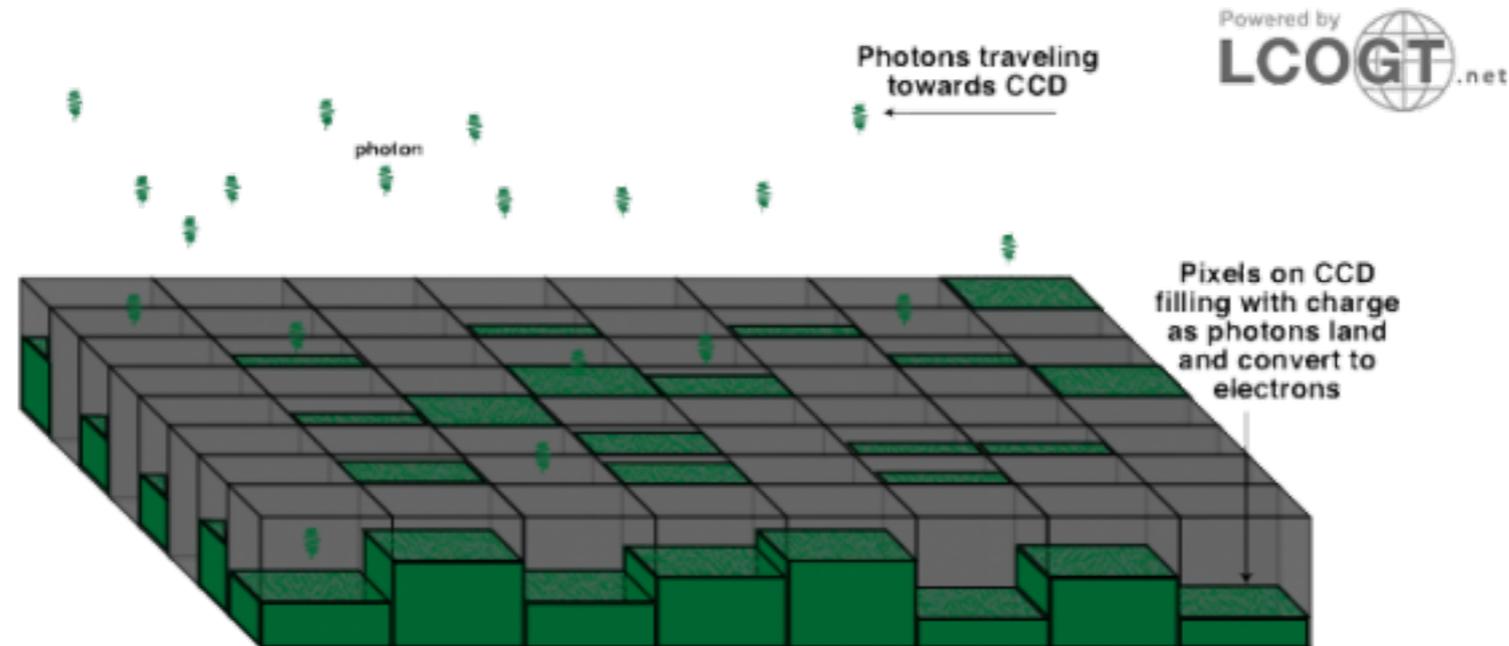
**MERCI DE VOTRE ATTENTION !**



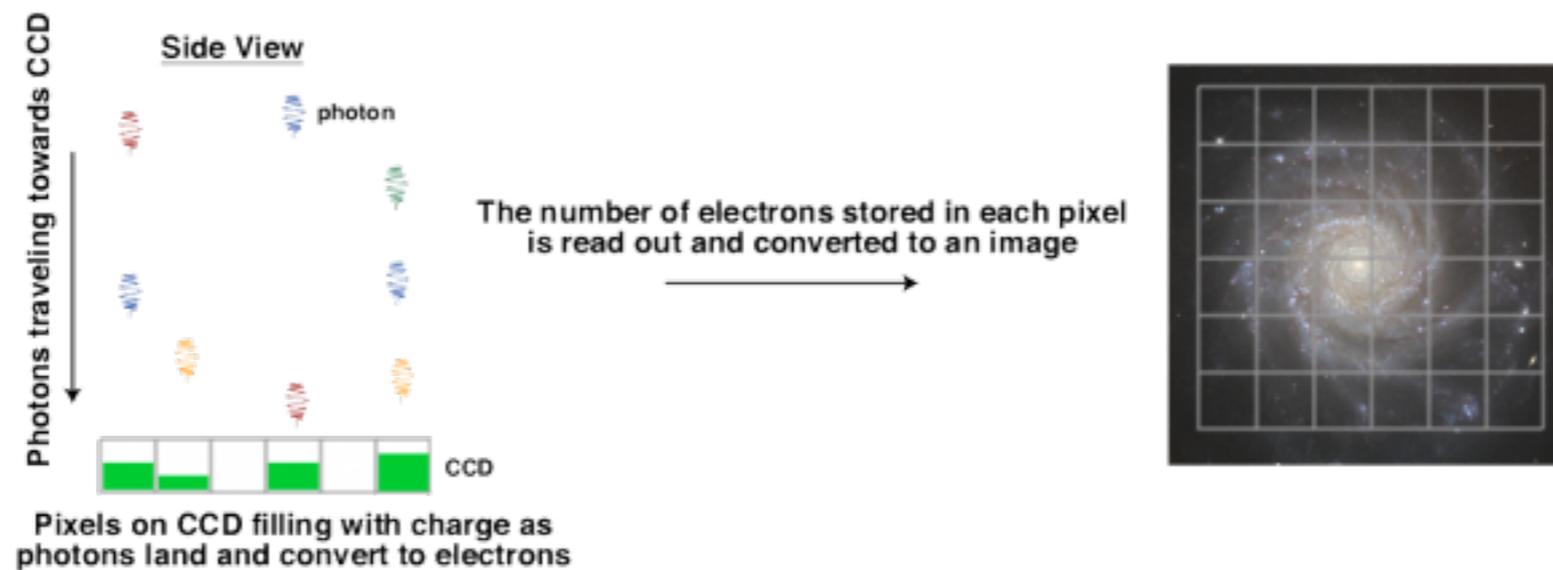
**Email : [kevin.polisano@gmail.com](mailto:kevin.polisano@gmail.com)**

**Site web : <http://kevin.polisano.free.fr>**

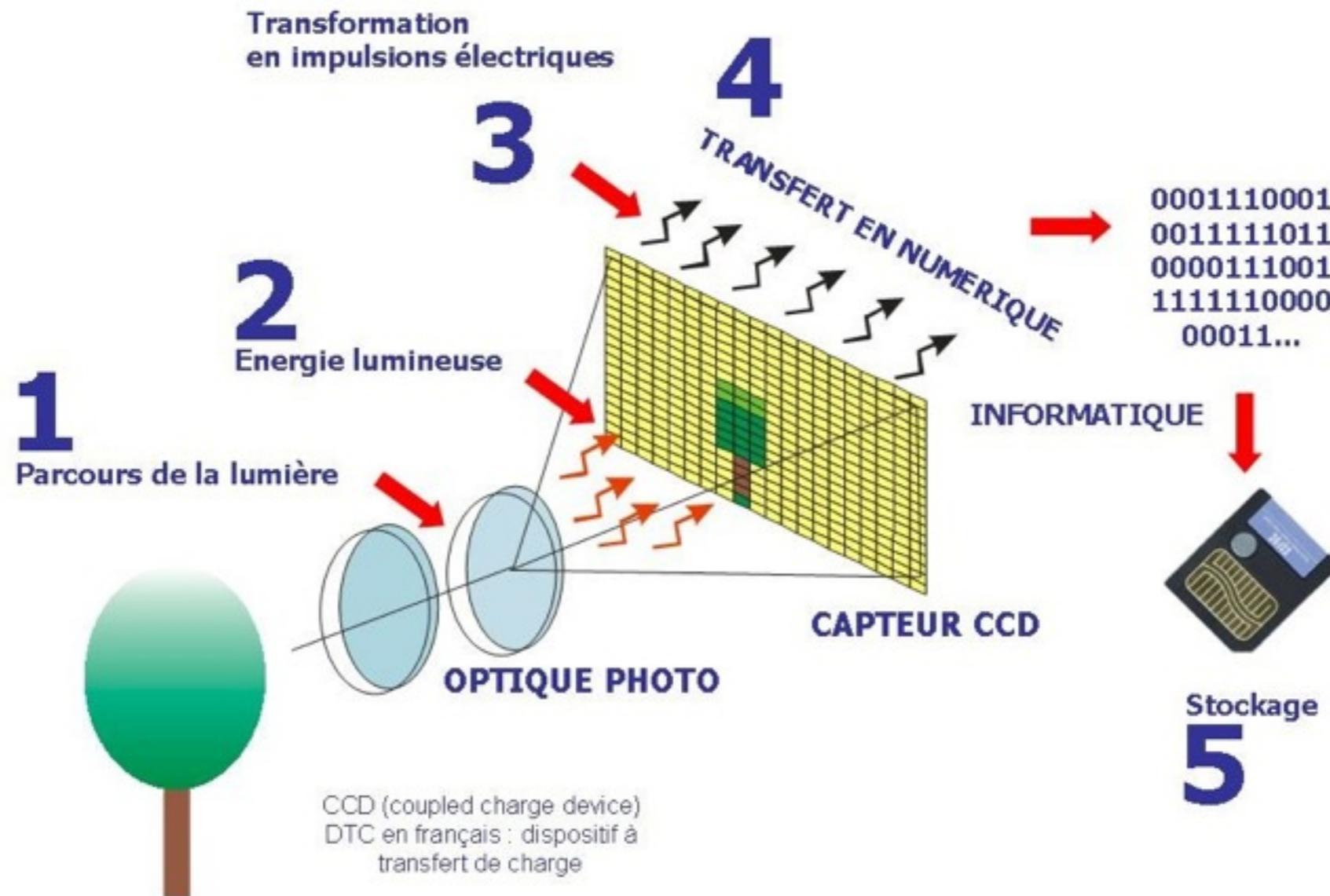
# Comment un appareil photo numérique forme-t-il une image ?



*Please note:* This diagram is representational only and not to scale. There are actually 2048 pixels along each side of the CCD in the Merope camera on Faulkes Telescope North. Each pixel is 13.5 micrometers on a side.



# Comment un appareil photo numérique forme-t-il une image ?



# Comment un ordinateur lit-il les nombres en binaire ?

Équivalent en décimal si le bit est allumé

128	64	32	16	8	4	2	1
0	0	0	0	0	0	0	0
$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$

Un octet (8 bits)

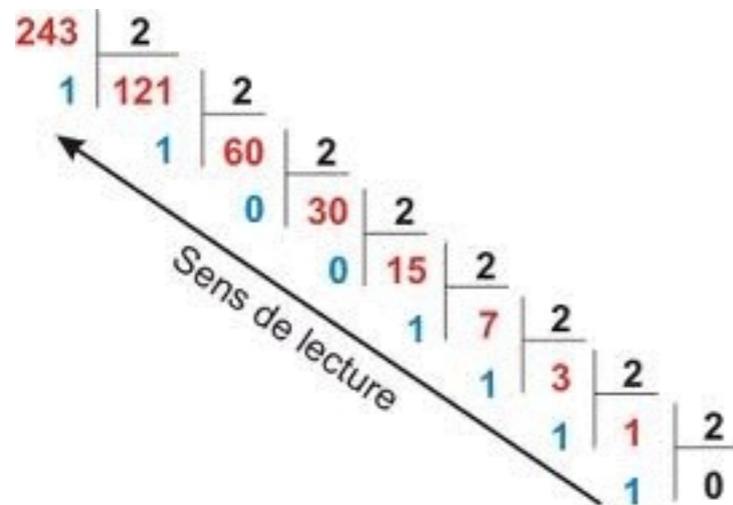
Équivalent en puissance de deux du nombre décimal

L'AE N°100 de 1971 - ISBN 2-901-047

## Décomposition sur la base décimale

$$\begin{aligned}
 243 &= 2 \times 100 + 4 \times 10 + 3 \times 1 \\
 &= 2 \times 10^2 + 4 \times 10^1 + 3 \times 10^0
 \end{aligned}$$

- Image dimension **1780x1335**
- $1780 \times 1335 = 2\,376\,300$  pixels
- 1 pixel = 3 octets (RGB)
- Taille totale = 7 128 900 octets
- 1Ko=1024 octets et 1Mo=1024 Ko
- $7\,128\,900 / 1024 = 6961,8$  Ko et  $6961,8 / 1024 = \mathbf{6,8\ Mo}$

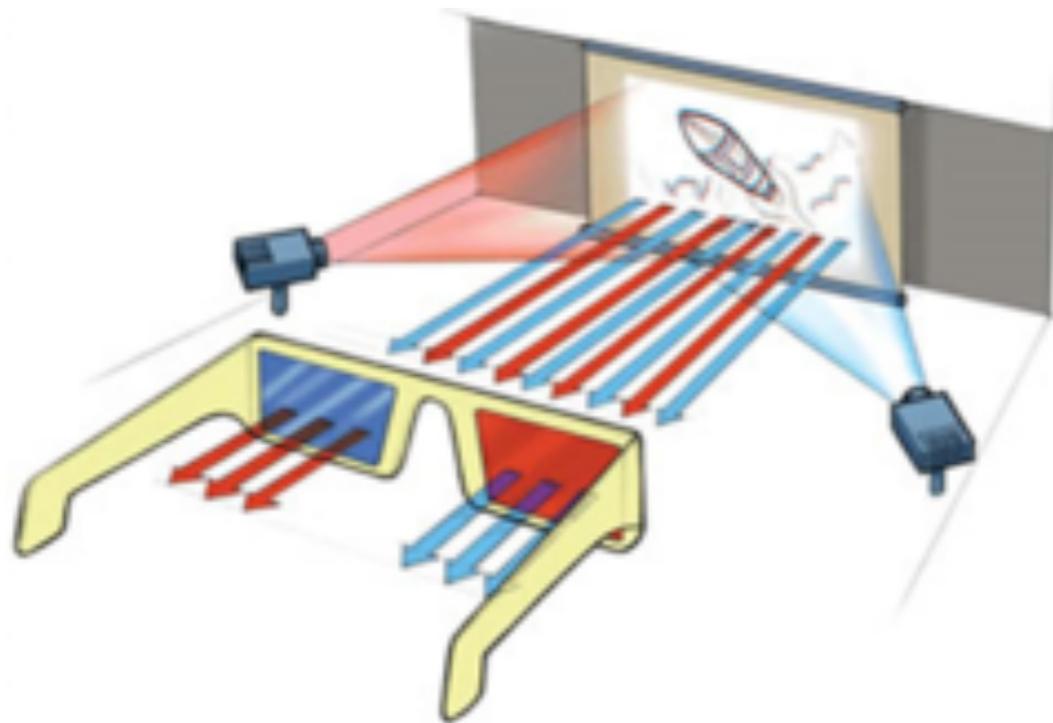


## Décomposition sur la base binaire

$$\begin{aligned}
 243 &= 1 \times 128 + 1 \times 64 + 1 \times 32 + 1 \times 16 + 0 \times 8 + 0 \times 4 + 1 \times 2 + 1 \times 1 \\
 &= 1 \times 2^7 + 1 \times 2^6 + 1 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0
 \end{aligned}$$

# Le cinéma 3D recrée le principe de la vision binoculaire

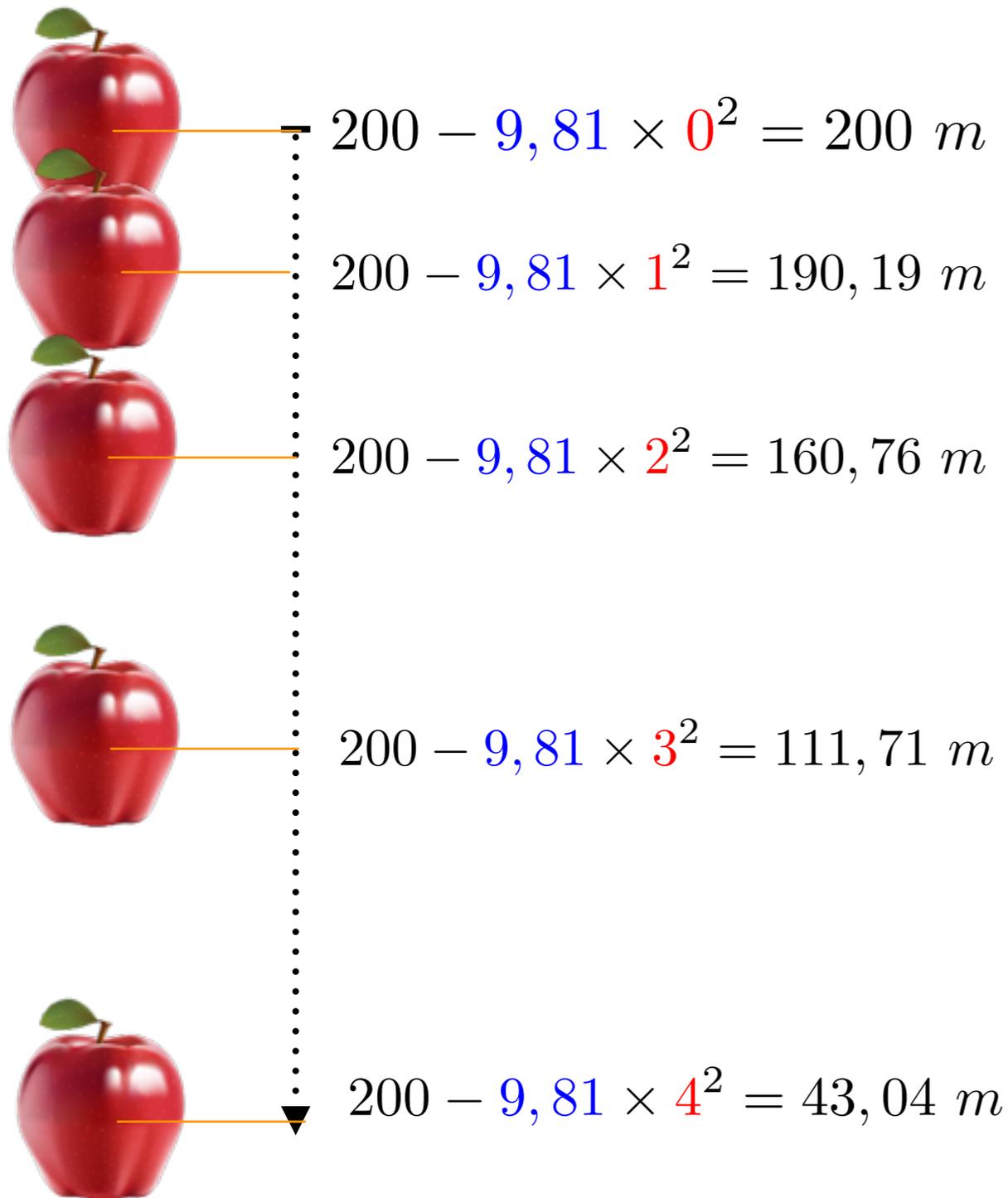
## ■ Projection **anaglyphe**



Les images **rouge** et **cyan** sont « décalées »  
Un **filtre** est disposé devant chaque oeil  
pour laisser passer la **couleur complémentaire**  
Le cerveau fusionne enfin les deux images

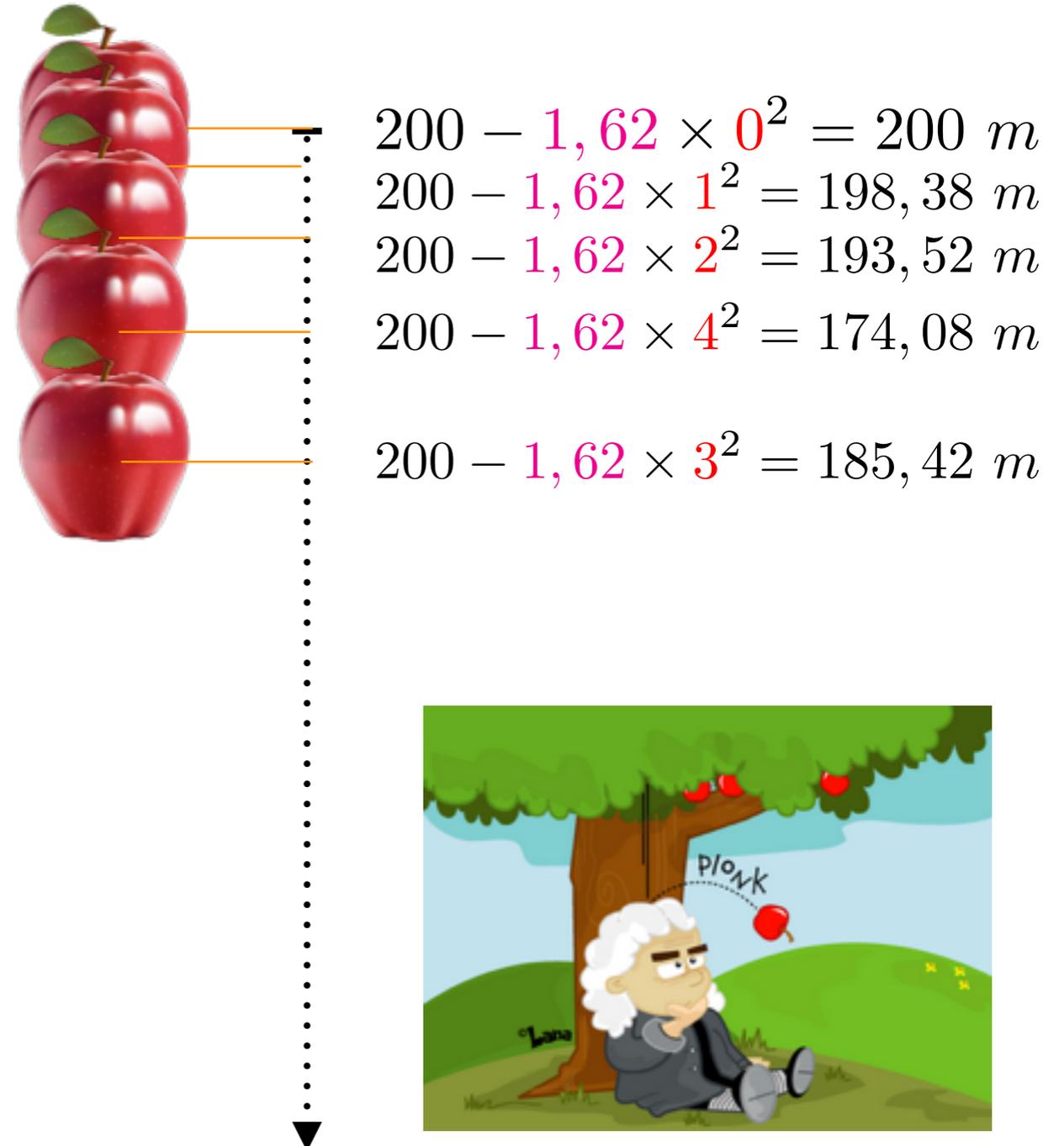
# Comment la gravité affecte le mouvement des objets ?

## Pomme tombant sur la Terre

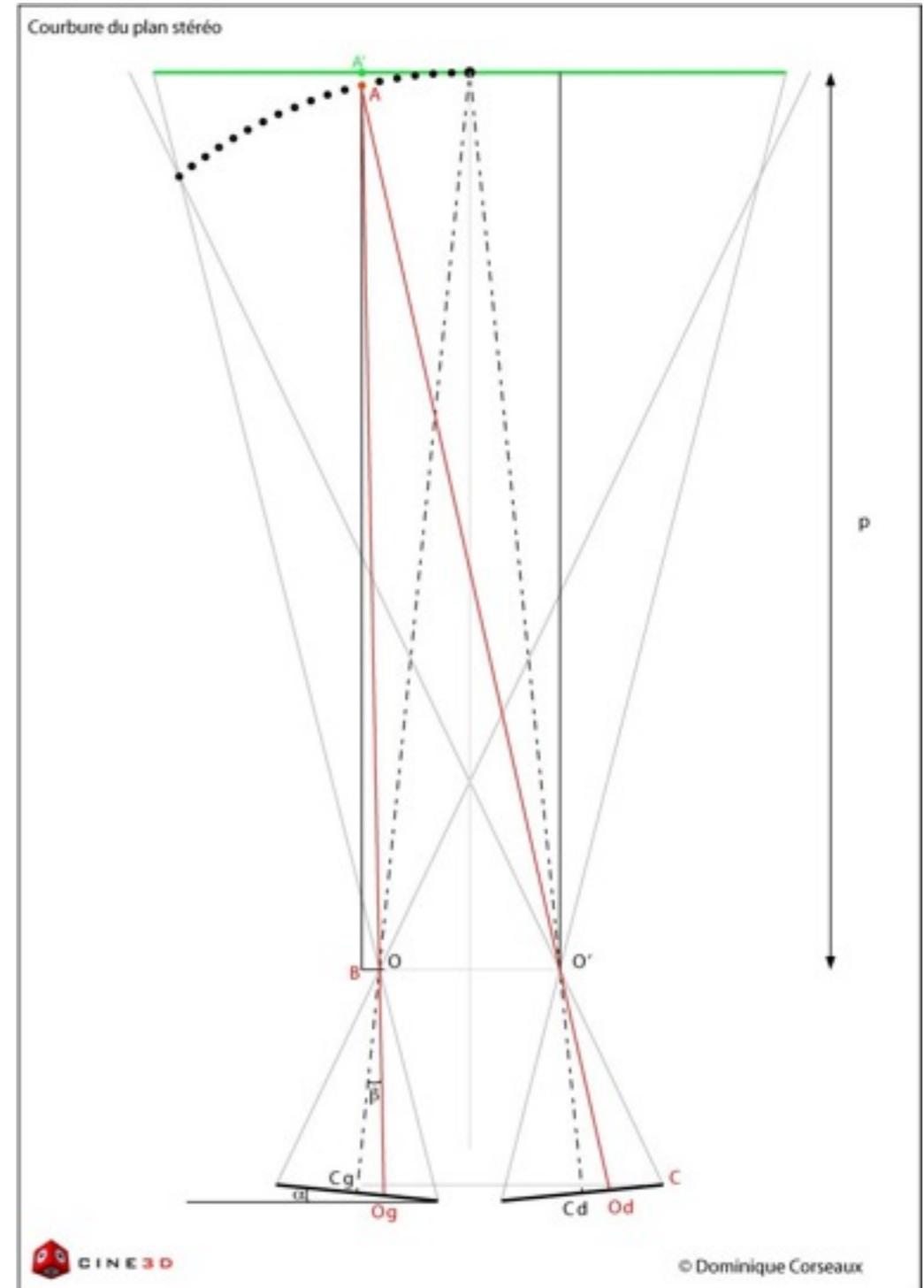
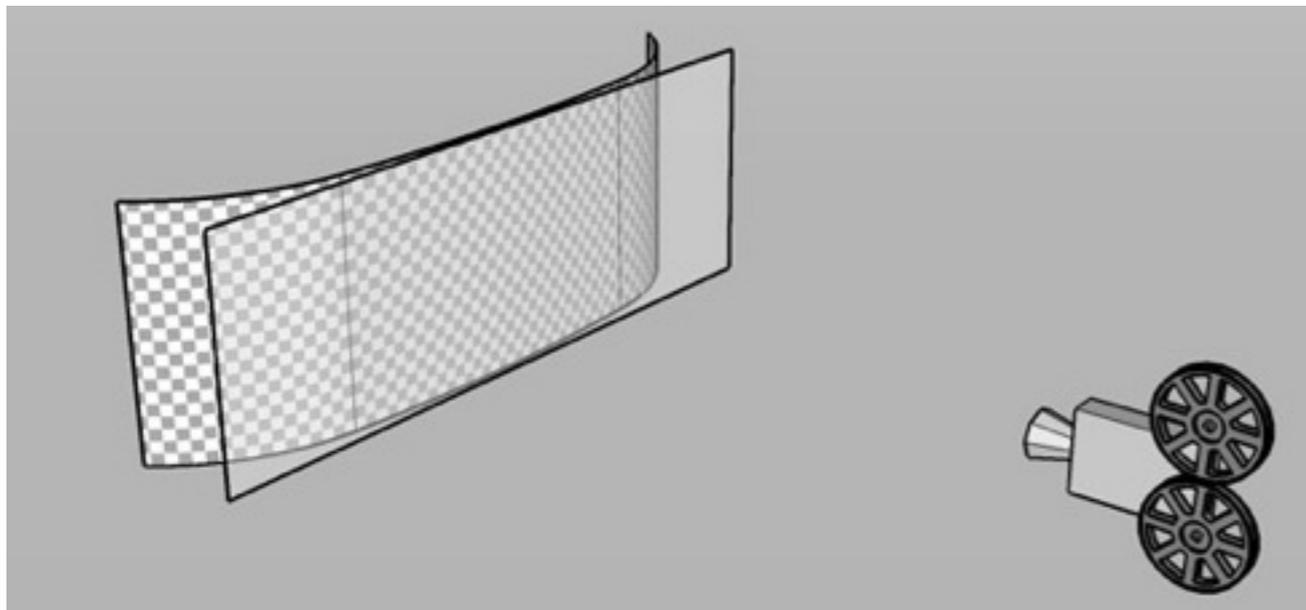
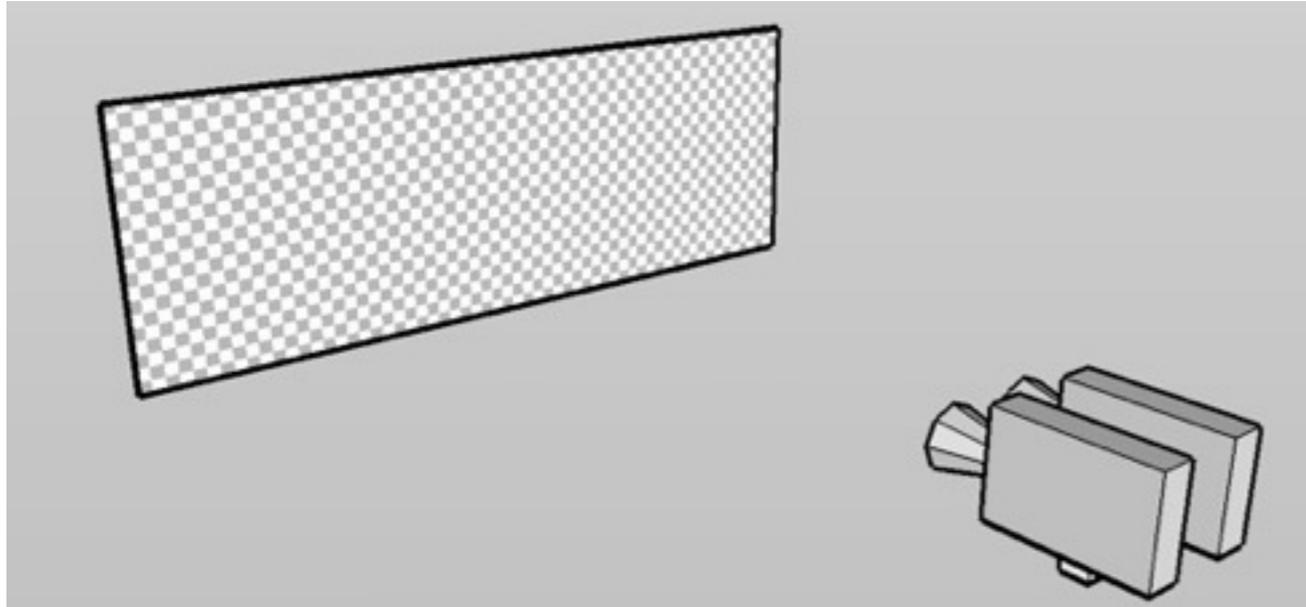


$$z = z_0 - gt^2$$

## Pomme tombant sur la Lune

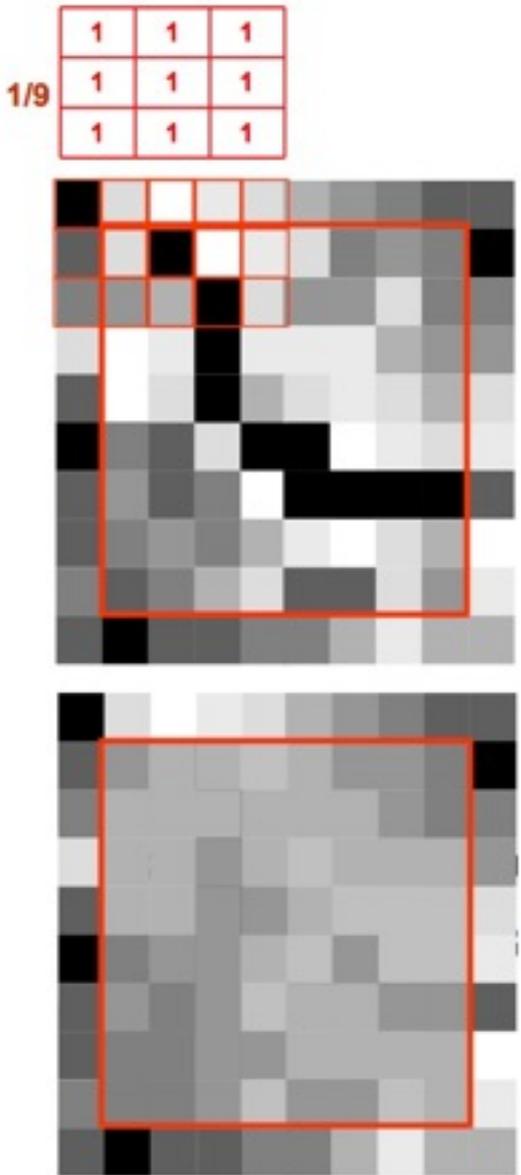


# Pourquoi observe-t-on une déformation de l'image sur les bords de l'écran au cinéma 3D ?



Source : <http://www.cine3d.ch/cine3d/>

# Le filtre **moyenneur**



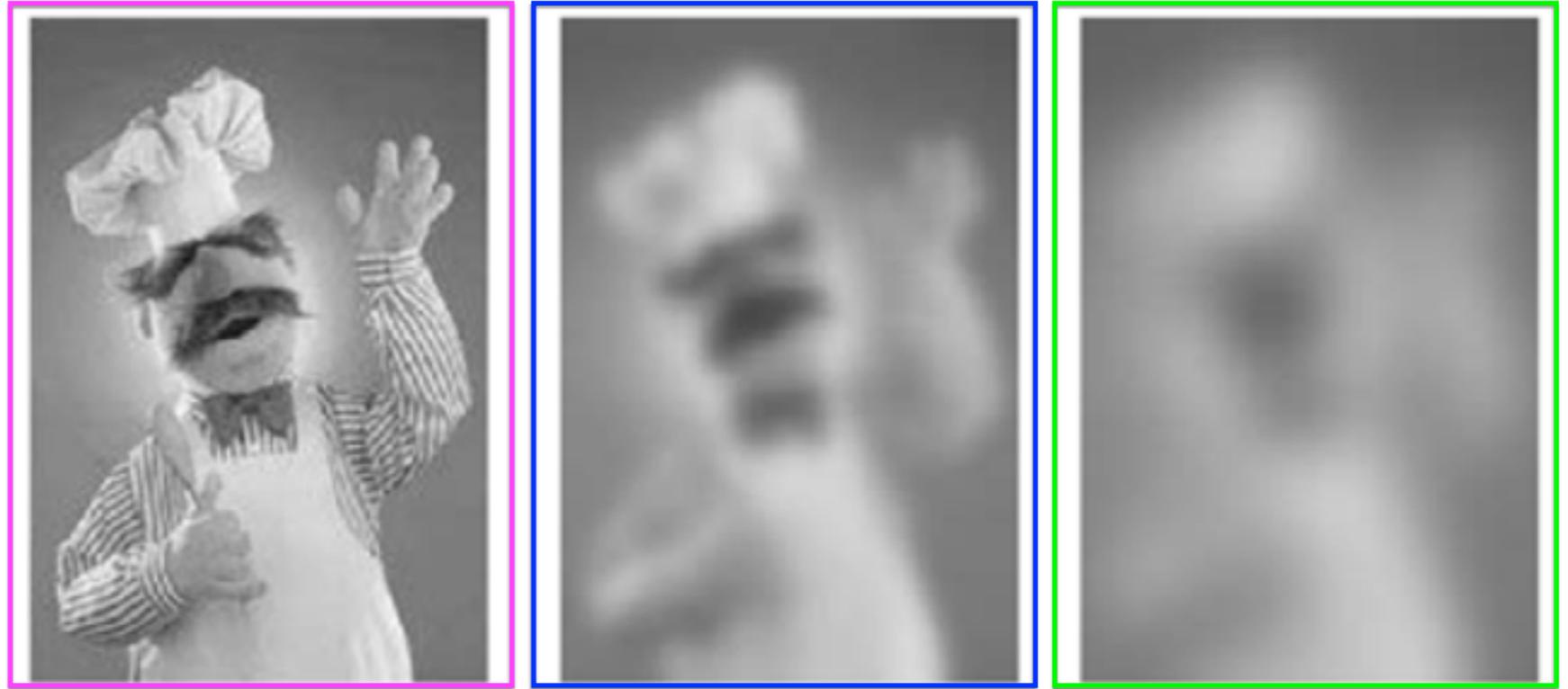
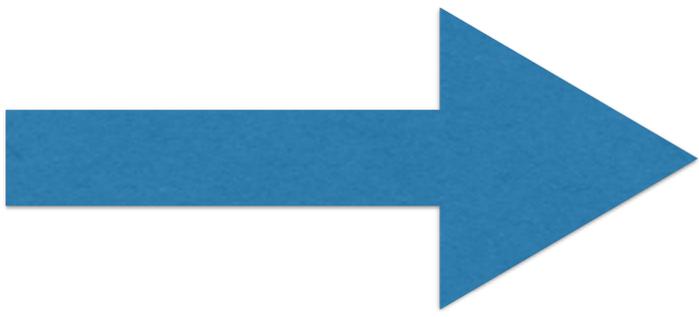
0	5	7	6	5	4	3	2	1	1
1	5	0	7	6	5	2	3	2	0
2	3	4	0	5	3	3	5	2	2
5	7	6	0	6	6	6	4	3	3
1	7	5	0	4	5	6	5	4	5
0	2	1	5	0	0	7	6	5	6
1	3	1	2	7	0	0	0	0	1
1	2	3	2	4	6	7	5	4	7
2	1	2	3	5	1	1	5	2	6
1	0	1	1	2	2	3	6	4	4

0	5	7	6	5	4	3	2	1	1
1	3	4	4	5	4	3	3	2	0
2	4	4	4	4	4	4	3	2	2
5	4	4	3	4	5	4	4	4	3
1	4	4	3	3	4	5	5	5	5
0	2	3	3	4	5	3	5	5	6
1	3	2	3	5	4	4	3	3	1
1	2	2	3	3	4	4	4	4	7
2	2	2	3	5	3	3	5	4	6
1	0	1	1	2	2	3	6	4	4

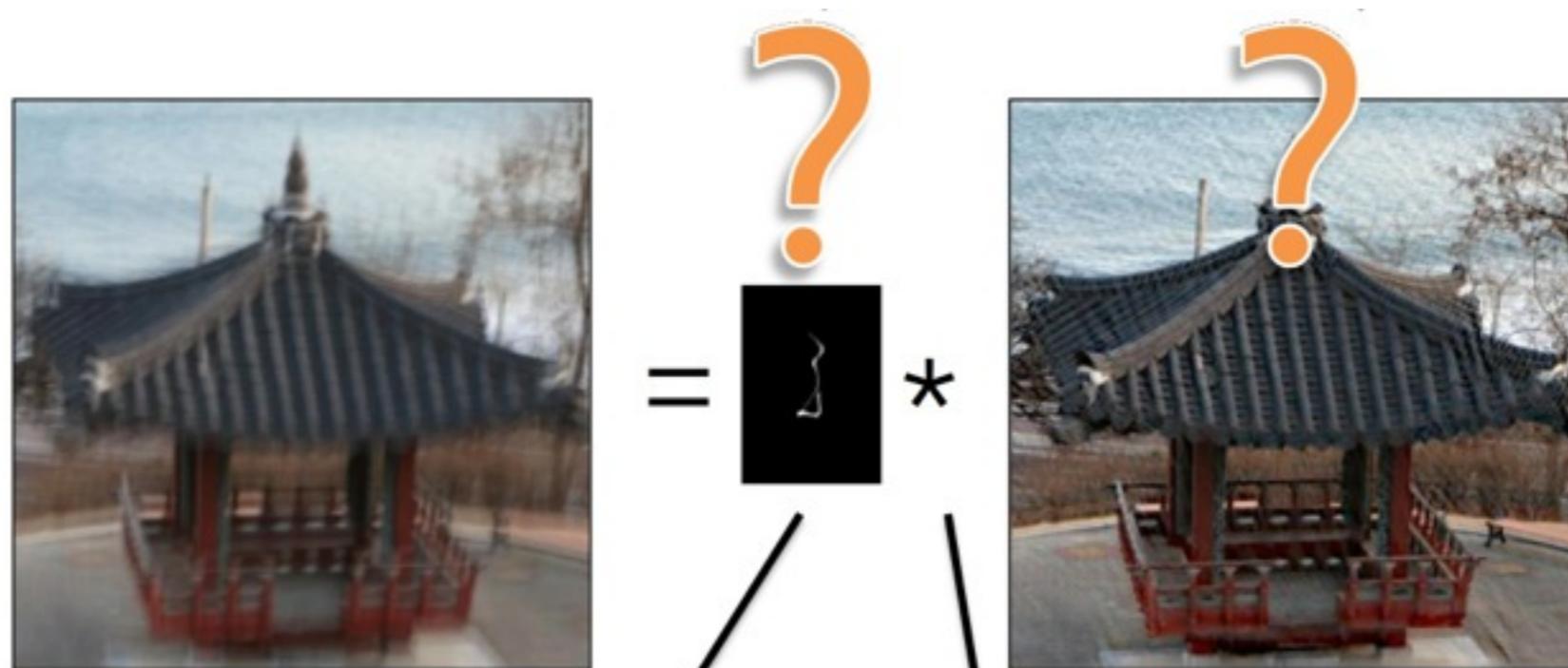
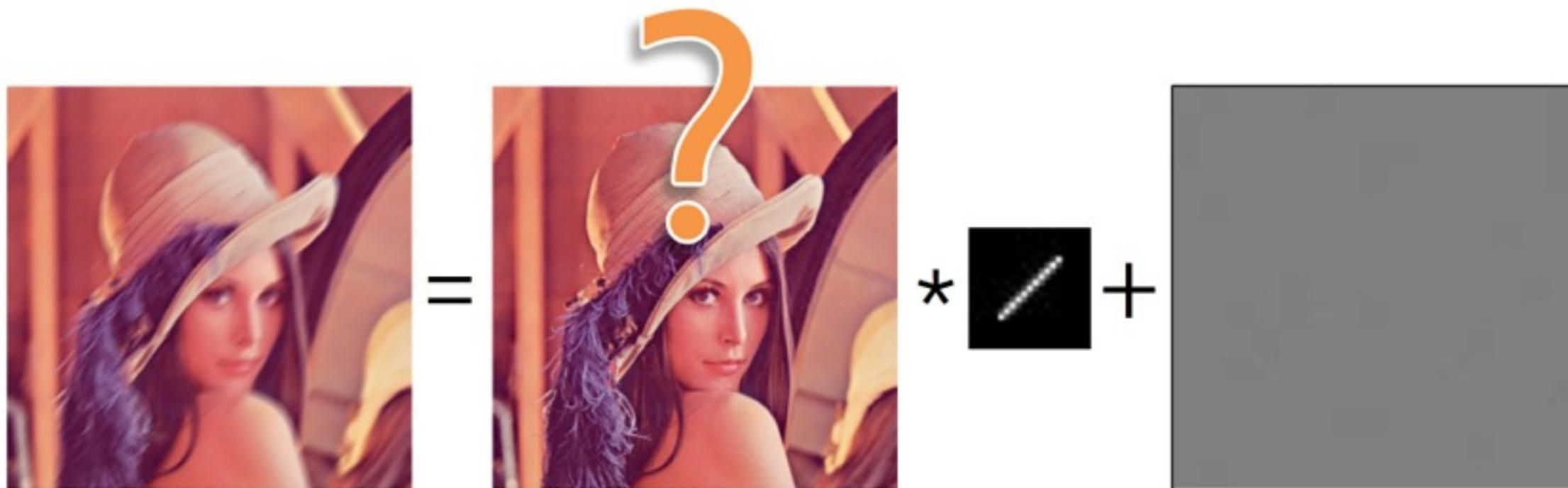
← Bords non traités

- Masque 3x3 :  $\frac{1}{9} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$
- Masque 5x5 :  $\frac{1}{25} \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix}$
- Masque 11x11: etc

Effet floutage



# Est-il possible de déflouter une image ?



Masque de floutage

Convolution

**N'hésitez pas à me contacter !**



**Email : [kevin.polisano@gmail.com](mailto:kevin.polisano@gmail.com)**

**Site web : <http://kevin.polisano.free.fr>**