

# Outils d'imagerie pour la robotique

## Analyse d'images et de vidéos

Florent Grélard  
[florent.grelard@u-bordeaux.fr](mailto:florent.grelard@u-bordeaux.fr)

TS341 | Option Robotique, 2022 – 2023

## Analyse d'images

Jusqu'ici nous avons utilisé des filtres, ou des régions, pour traiter l'image :

$$F : I \rightarrow I'$$

L'analyse d'images :

$$F : I \rightarrow \text{caractérisation}$$

↳ des mesures, de la détection, de la reconnaissance de formes...

# CARACTÉRISATION ET MESURES

## Position dans la chaîne de traitement

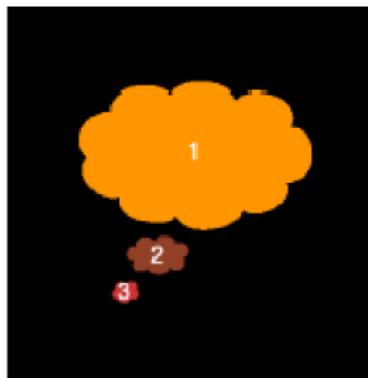


On s'intéresse ici aux méthodes de **caractérisation quantitative** dans une image segmentée.

## Composantes connexes (1/2)

Une **composante connexe** est un groupe de pixels connectés entre eux.

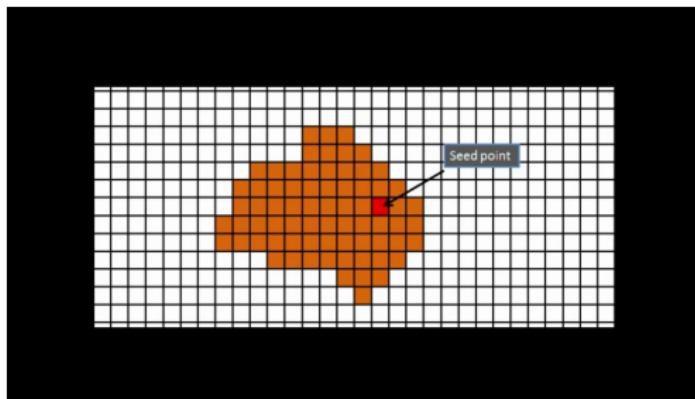
**Intérêt :** séparer les différents objets présents dans l'image.



## Composantes connexes (2/2)

Principe de la détermination des composantes connexes :

1. Initialisation sur un pixel blanc aléatoire (seed)
2. **Propagation** à ses voisins (4-voisins ou 8-voisins)
3. Si ses voisins sont objet, ils sont ajoutés à une **liste**
4. Répétition des étapes 2 et 3 pour les pixels dans la liste



## Boîte englobante et enveloppe convexe

La **boîte englobante** est le plus petit rectangle qui contient l'objet.

L'**enveloppe convexe** est le plus petit polygone convexe qui contient l'objet.



(a) Boîte englobante



(b) Enveloppe convexe

## Mesures géométriques (1/2)

Permettent de **caractériser** un objet (ex : reconnaissance geste de la main).

- **Aire** : nombre de pixels blancs (bonne approximation)
- **Périmètre** : nombre de pixels blancs dans l'image de contour (mauvaise approximation)
- **Compacité** : mesure si une forme est allongée ou a beaucoup de plis (<1) ou est proche d'un cercle (1).

$$\text{Calcul} = \frac{4\pi \text{ Aire}}{\text{Périmètre}^2}$$



## Mesures géométriques (2/2)

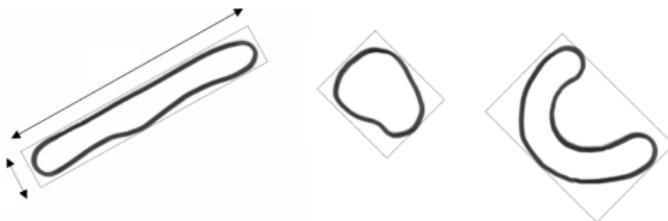
- **Barycentre** : centre de la boîte englobante ou de l'enveloppe convexe
- **Circularité** : mesure à quel point l'objet ressemble à un cercle.

Calcul :

$$\frac{4\pi \text{ Aire}}{\text{Périmètre de l'enveloppe convexe}^2}$$

- **Élongation** :

$$\frac{\text{longueur boîte englobante}}{\text{largeur boîte englobante}}$$

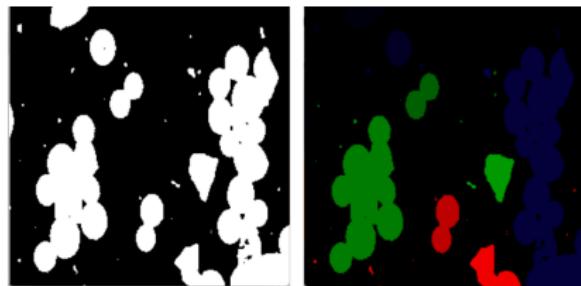


high elongation

low elongation

## Exemple

Exemple : stades cellulaires (cellules de levure)



1. Extraction des **composantes connexes**
2. Calcul d'**aire** : on ne conserve que les composantes connexes suffisamment grandes (seuil)
3. Calcul de **compacité** : permet de distinguer la phase cellulaire (division ou repos)

# DÉTECTION D'OBJETS

## Détection et reconnaissance (1/2)

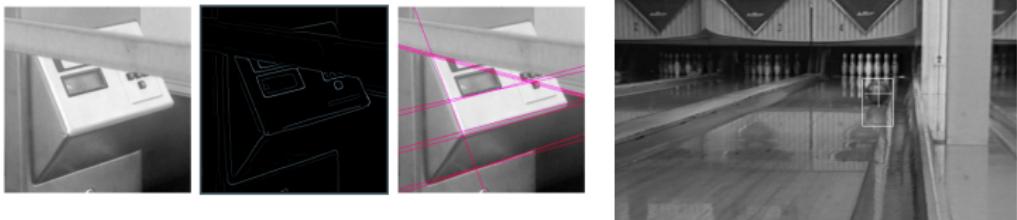
Deux familles d'algorithmes :

- la **détection** : identifier un type d'objets. Exemple : détecter tous les visages
- la **reconnaissance** : identifier un objet. Exemple : reconnaître le visage d'une personne

Pour la **détection** d'objets spécifiques :

1. Détection de **blobs** (= formes quelconques dans l'image)
2. Détection de **lignes et d'ellipses**
3. Détection de **points d'intérêt**

## Détection et reconnaissance (2/2)



## Détection de blobs (1/2)

**Blobs** : régions partageant des caractéristiques communes (couleur, forme) et différentes des régions voisines.



## Détection de blobs (2/2)

Idée de l'algorithme :

1. Générer plusieurs images binaires avec différents seuils
2. Pour chaque image, extraire les **composantes connexes**
3. Fusion des composantes connexes suffisamment  
“proches” → **blobs**. Utilise un critère de caractéristique  
géométrique, de couleur, de distance...

# DÉTECTION DE LIGNES

## Transformée de Hough

Transformée de Hough : permet de détecter les **lignes** dans une image.

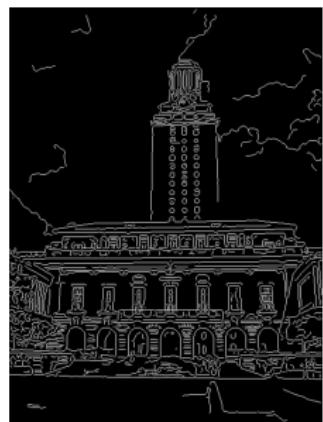
Intérêt : beaucoup d'objets sont caractérisés par leurs arêtes :



## Transformée de Hough

Image de contours : pourquoi est-ce insuffisant ?

- information **partielle** : points manquant dans une ligne
- information **superflue** : points n'appartenant pas à des lignes
- bruit



## Transformée de Hough

La **transformée de Hough** détecte les lignes majoritaires dans une **image de contour**.

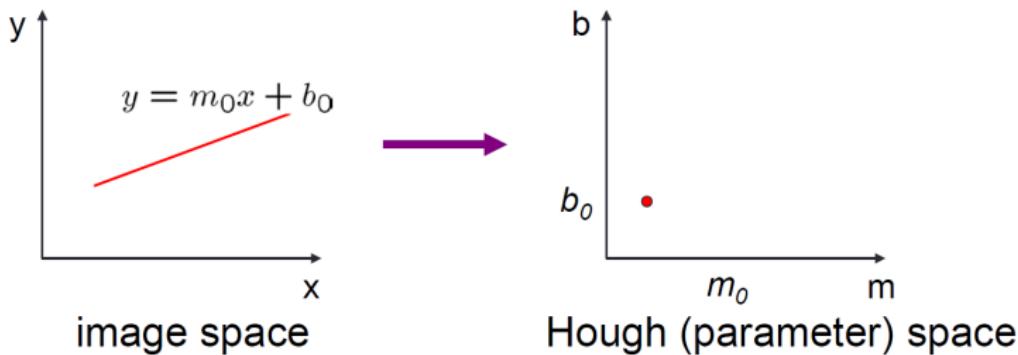
**Principe** : système de vote :

1. ajuster toutes les droites pouvant passer par un point de contour
2. choisir les droites les plus fréquentes

↳ On cherche des lignes dans l'espace de l'image  $(x, y)$ , satisfaisant  $y = mx + b$ .

## Transformée de Hough

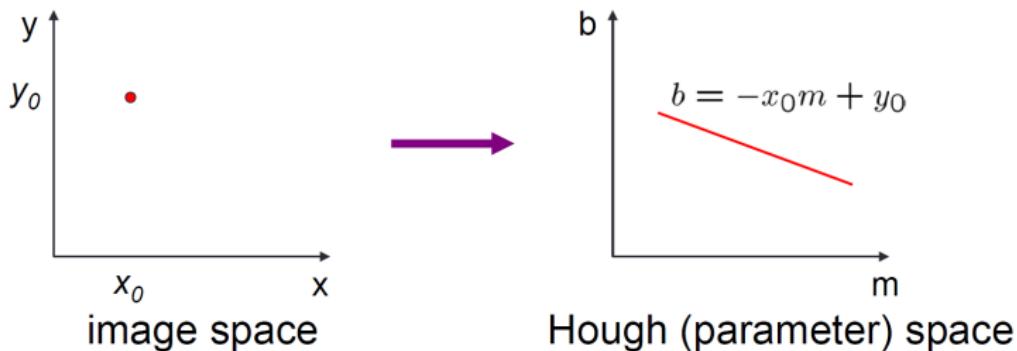
Idée : chercher l'information dans l'espace de paramètres  $(m, b)$ .



Une **ligne** dans l'espace image correspond à un **point** dans l'espace  $(m, b)$ .

## Transformée de Hough

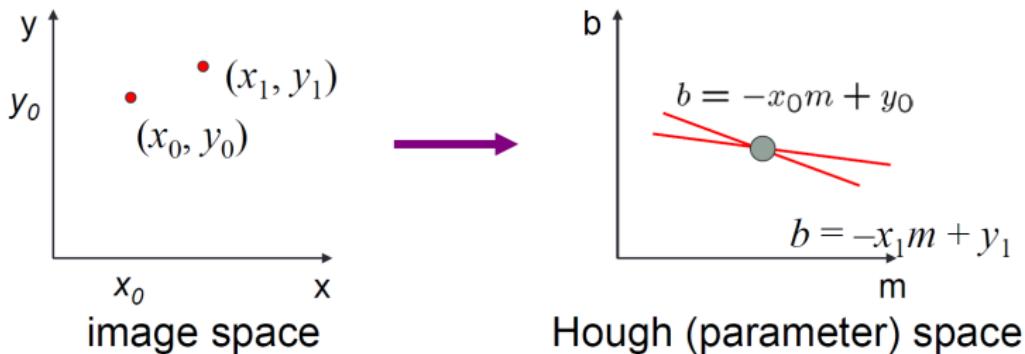
Idée : chercher l'information dans l'espace de paramètres  $(m, b)$ .



Un **point** dans l'espace image correspond à une **ligne** dans l'espace  $(m, b)$ .

# Transformée de Hough

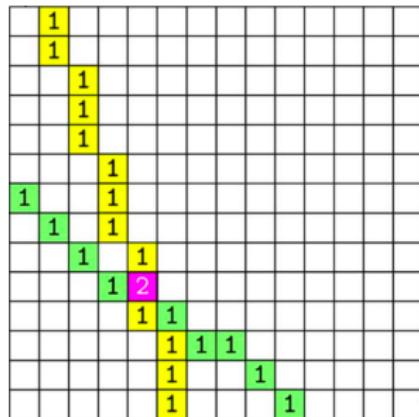
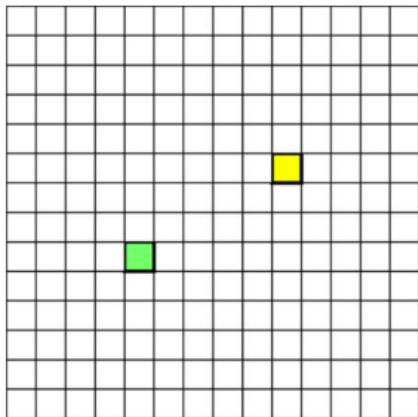
Idée : chercher l'information dans l'espace de paramètres  $(m, b)$ .



La ligne passant par  $p_0$  et  $p_1 \rightarrow$  intersection entre  $-x_0 m + y_0$  et  $-x_1 m + y_1$ .

## Transformée de Hough

Discrétisation de l'espace ( $m, b$ ) : **vote** :

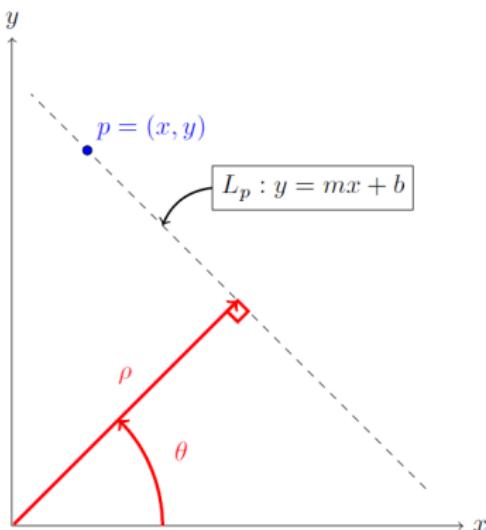


Problèmes : lignes verticales, espace ( $m, b$ ) non borné.

# Transformée de Hough : espace polaire

En pratique : on utilise l'espace polaire  $(\rho, \theta)$  :

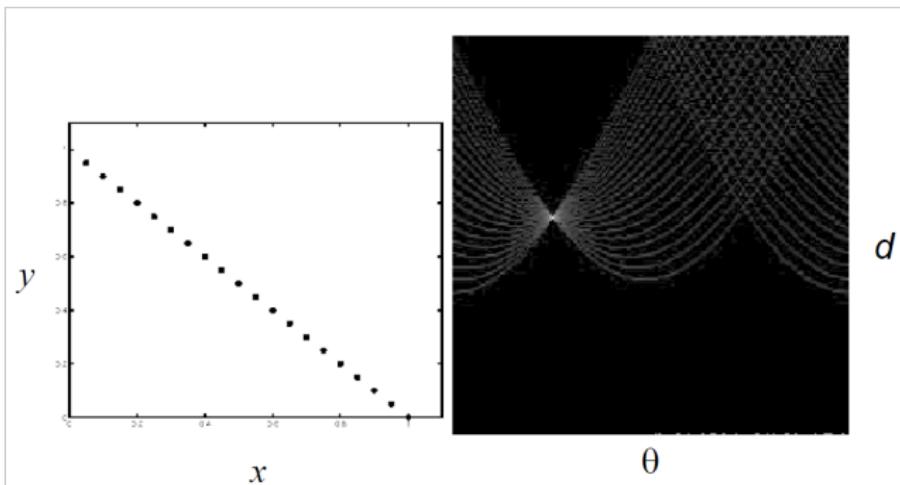
$$y = mx + b \Leftrightarrow \rho = x \cos \theta + y \sin \theta.$$



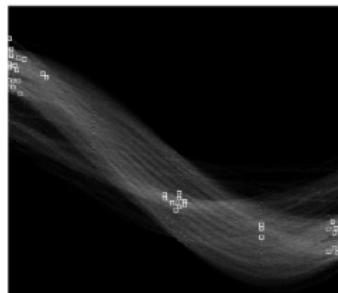
## Transformée de Hough : espace polaire

Intérêt : l'espace  $(\rho, \theta)$  est borné.

- Un **point** en  $(x, y)$  devient une **sinusoïde** en  $(\rho, \theta)$
- Une **ligne** en  $(x, y)$  reste un **point** en  $(\rho, \theta)$



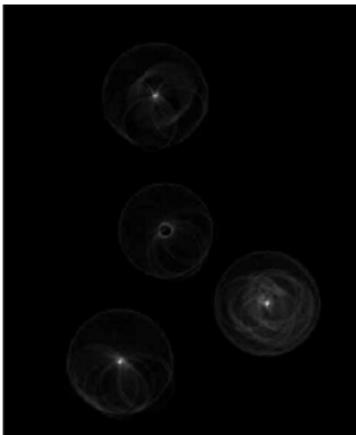
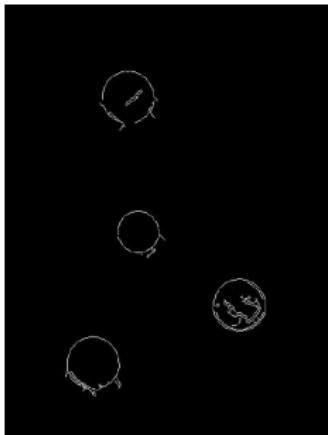
## Transformée de Hough : application



## Transformée de Hough : ellipses

La transformée de Hough peut être étendue à la détection de cercles ou d'ellipses :

Equation du cercle :  $(x - x_C)^2 + (y - y_C)^2 = r^2 \rightarrow$  espace à 3 paramètres.

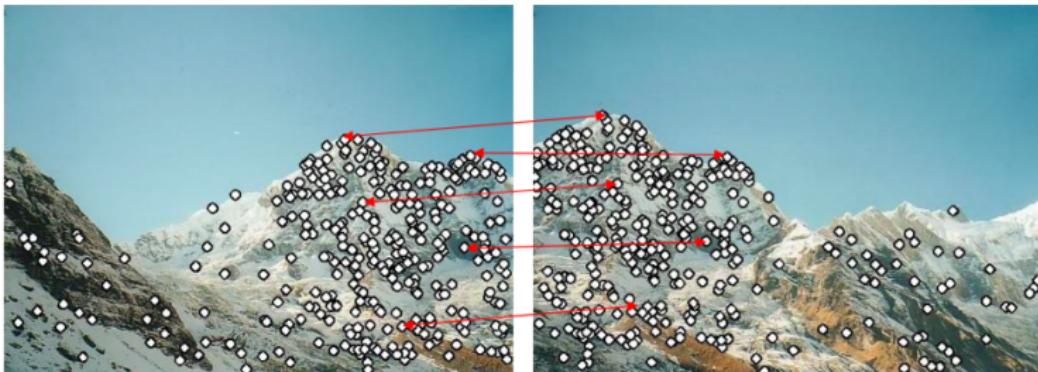


# DÉTECTION DE POINTS D'INTÉRÊT

## Détection de points d'intérêt

Intérêt : appariement de photographies

- Détection de coins
- Détection de points saillants

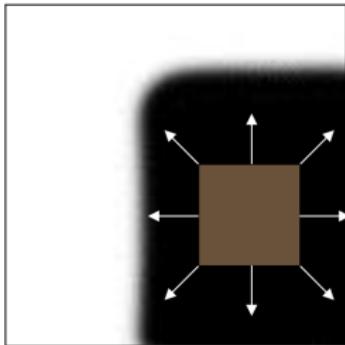


## Détection de coins

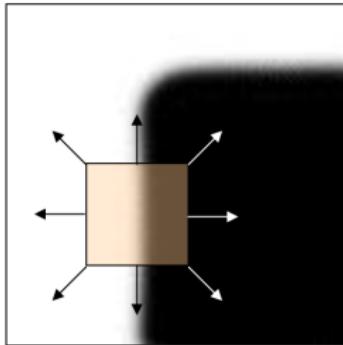
Zone homogène : pas de changement d'intensité

Contour : pas de changement le long du contour

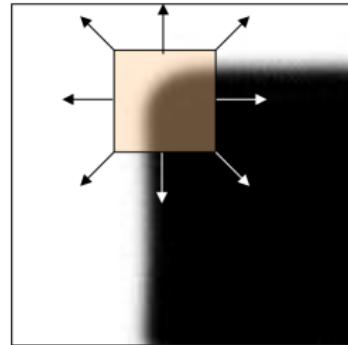
Coin : changement significatif dans au moins 2 directions



(a) Zone homogène



(b) Contour

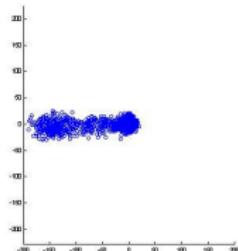
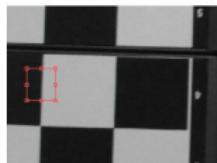


(c) Coin

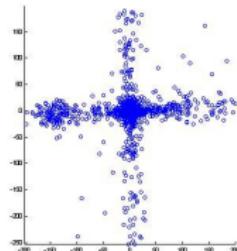
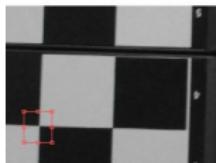
## Détection de coins

L'algorithme de Harris utilise les gradients horizontaux et verticaux  $I_x$  et  $I_y$ .

Valeurs de  $I_y$  en fonction de  $I_x$  :



(a) Contour



(b) Coin

## Algorithme de Harris

**Idée :** Discriminer entre **coin** et **non-coin** par la **dispersion** des valeurs de gradient.

Algorithme de Harris :

1. Calcul des gradients  $I_x$  et  $I_y$  par **filtres de Sobel**
2. Calcul de la **matrice de covariance** des valeurs de gradients
3. Un **coin = valeurs propres similaires** dans la matrice de covariance

## Algorithme de Harris

Application :



## Détection de points saillants

Détection de **points saillants** dans l'image : points avec des variations de contraste dans un voisinage.

**Problème** : entre deux images : variations en fonction de l'échelle, de la rotation, de l'illumination.

---

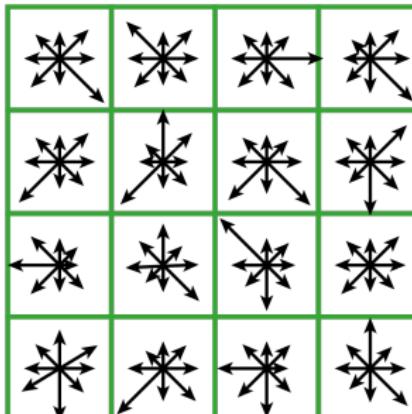
Méthode **SIFT** (*Scale Invariant Feature Transform*) pour pallier ces problèmes.

Points saillants = points avec de fortes variations de **contraste**, **d'échelle** et **d'orientation**.

## Méthode SIFT

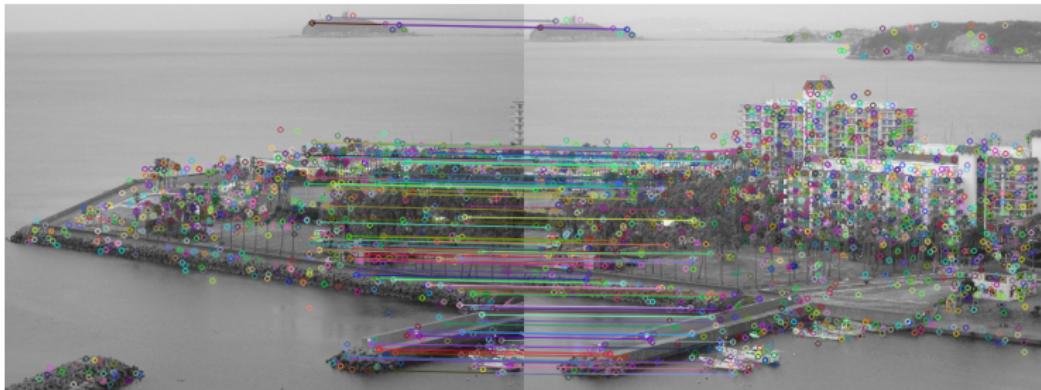
A chaque point saillant : descripteur de 128 valeurs permettant l'appariement.

Le descripteur reflète l'orientation des gradients dans un voisinage.



## Méthode SIFT

Application pour un panorama :



# Sommaire

Objectifs et organisation du cours

Analyse d'image

Analyse de vidéos

- Définitions
- Soustraction de fond
- Suivi couleur

## Analyse de vidéos : exemples (1/2)

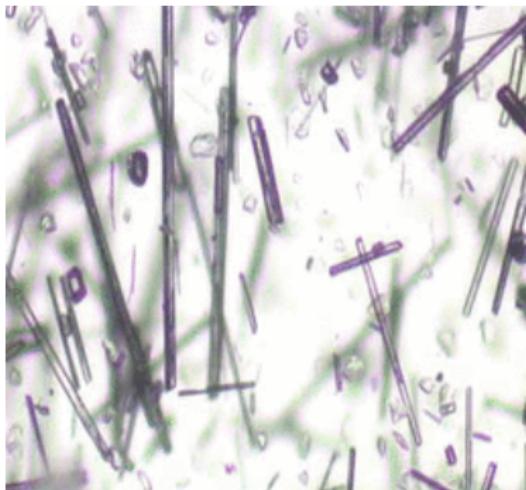
### 1. Reconnaissance d'activités

**Comportement** : analyse et prédition d'activités humaines



### 2. Estimation de mouvements

**Biologie** : mobilité des composants cellulaires

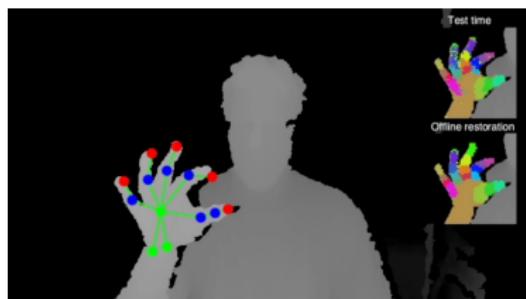


## Analyse de vidéos : exemples (2/2)

### 3. Reconnaissance de gestes

Comportement :

Reconnaissance de gestes de la main



### 4. Interaction

Art : Danse et réalité mixte

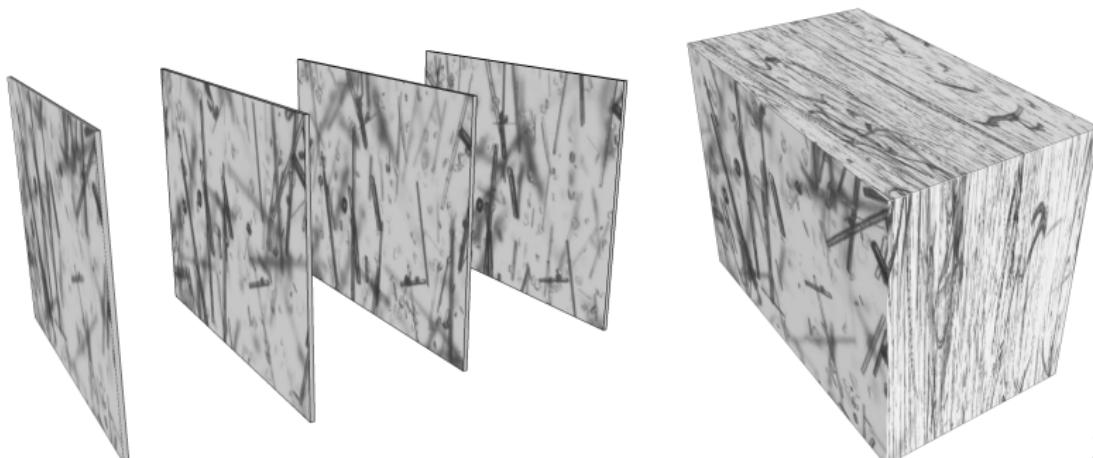


# Vidéo

**Vidéo** : succession d'images dans le temps.

**Fréquence** : nombre d'images par seconde (fps, Hz).

Lien de **causalité** dans la dimension temporelle.



## Analyse de vidéos (1/3)

L'analyse de vidéos est une branche de l'analyse d'image.

**Contrainte :** temps réel.

**Information supplémentaire :** prise en compte de la dimension temporelle :

1. Analyse du mouvement
2. **Déetecter / Reconnaître** ce qui apparaît ou disparaît dans la vidéo

### Analyse de vidéos (2/3)

Détection d'objets en vidéo basée sur :

- la couleur ou la texture (fond / objets)
- la forme (objets)
- le mouvement (fond / objets)

Généralement : **combinaison** de différents critères



# Analyse de vidéos (3/3)

Quelques algorithmes couramment utilisés en traitement de vidéo :

- Soustraction de fond (avant-plan/arrière-plan)
  1. Algorithme par différence pixel à pixel
  2. Algorithme par mélange de gaussiennes
- Détection d'objets :
  1. Seuillage
  2. Détection de contours
  3. Suivi de couleur
  4. Détection de blobs
- Caractérisation d'objets : mesures

# SOUSTRACTION DE FOND

## Soustraction de fond (1/2)

**Objectif :** différencier l'arrière-plan de l'avant-plan.

**Idée :** Apprendre le fond.

**Exemple :** remplacement de fond :



### Soustraction de fond (2/2)

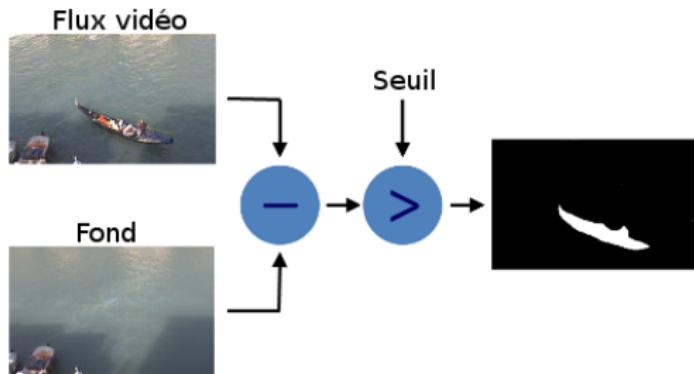
Deux algorithmes :

1. Différence pixel à pixel
2. Mélange de gaussiennes

## Différence pixel à pixel (1/4)

Idée : Les objets bougent, le fond reste inchangé.

Algorithme par différence pixel à pixel : seuillage de la différence entre une image **avec objet** et une image **sans objet** (image de fond)



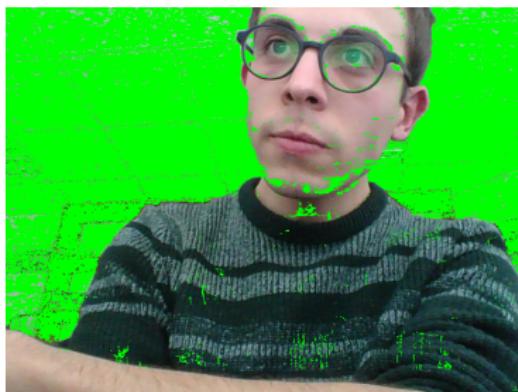
## Différence pixel à pixel (2/4)

Algorithme :

1. `seuil`  $\leftarrow$  seuil de couleur
2. `fond`  $\leftarrow$  image de fond
3. Pour chaque pixel de l'image de la webcam :
  - 3.1 Récupérer sa couleur  $C$
  - 3.2 Récupérer la couleur correspondante  $C_f$  de l'image `fond`
  - 3.3 Si la différence entre la couleur  $C$  et  $C_f$  est inférieure à `seuil`
    - Alors : il s'agit d'un pixel de fond
    - Sinon : il s'agit d'un pixel de l'objet

### Différence pixel à pixel (3/4)

Nécessite de posséder ou d'enregistrer **une image sans objet**.



### Différence pixel à pixel (4/4)

Limites :

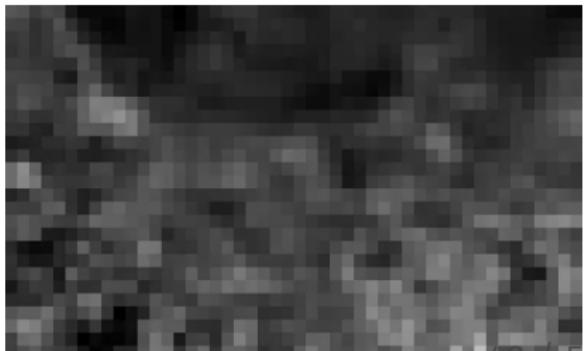
- Couleur objet/fond proche
- Variation de luminosité
- Mouvement du fond
- Variation intrinsèque de la caméra



### Mélange de gaussiennes

**Enjeu :** déterminer et **mettre à jour** l'image de fond régulièrement pour pallier les problèmes de luminosité

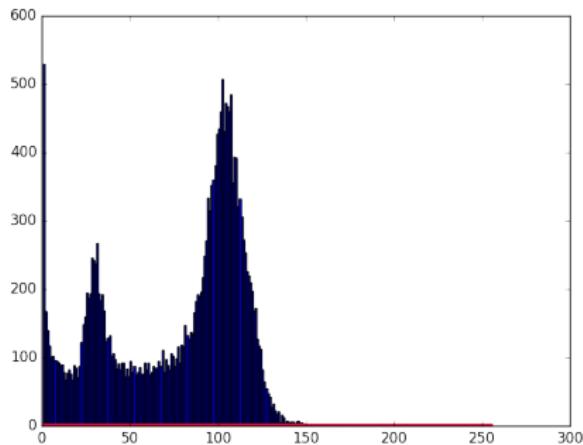
**Idée :** un pixel de fond a peu de variations d'intensité, alors qu'un pixel objet a des variations importantes.



# Mélange de gaussiennes

Principe : pour chaque pixel :

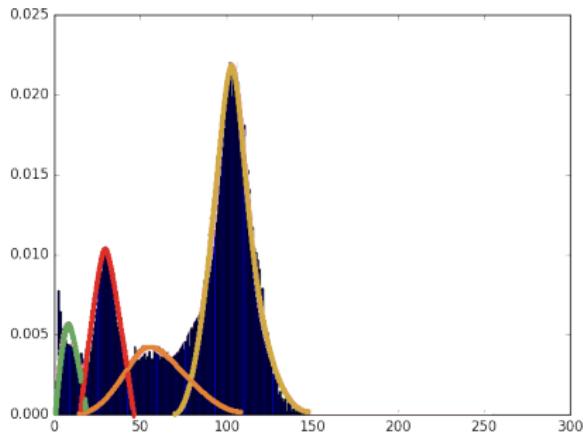
1. Calculer l'histogramme des intensités au cours du temps



## Mélange de gaussiennes

Principe : pour chaque pixel :

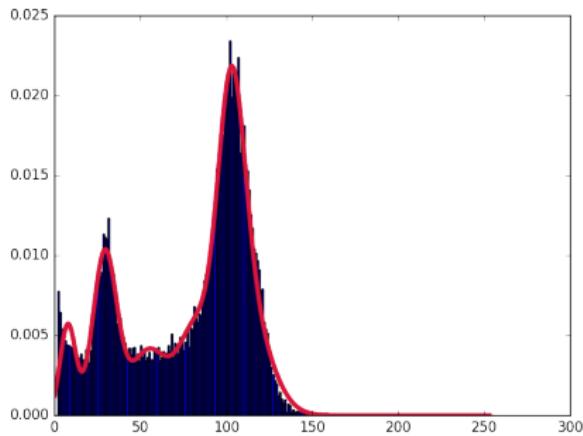
1. Calculer l'histogramme des intensités au cours du temps
2. Modéliser chaque histogramme par une **mélange de gaussiennes** (algorithme d'espérance-maximisation)



## Mélange de gaussiennes

Principe : pour chaque pixel :

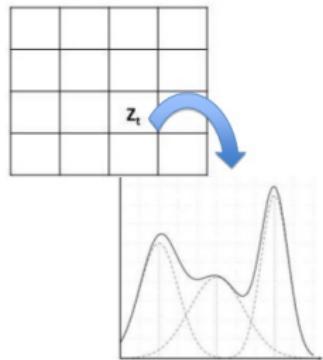
1. Calculer l'histogramme des intensités au cours du temps
2. Modéliser chaque histogramme par une **mélange de gaussiennes** (algorithme d'espérance-maximisation)



## Mélange de gaussiennes

Classification :

1. Pixel de fond : mélange de gaussiennes avec un **écart-type faible** (peu de variations au cours du temps)
2. Pixel d'objet : mélange de gaussiennes avec au moins une gaussienne d'**écart-type important** (fortes variations au cours du temps)



## Mélange de gaussiennes

Application : suivi d'objets



## Mélange de gaussiennes

Avantages :

1. S'adapte aux changements progressifs de luminosité
2. L'image de fond est recalculée fréquemment ⇒ pas besoin d'enregistrer une image de fond

Limites :

1. Ne fonctionne pas avec les changements brusques de luminosité
2. Objet immobile dans l'image

# Suivi de couleur

1. On enregistre la couleur à suivre avec un clic de souris
2. On cherche dans l'image du flux vidéo la couleur la plus proche



# BILAN

# Bilan

Les algorithmes de traitement et d'analyse d'image sont applicables aux vidéos :

- Correction de contraste
- Filtres
- Détection d'objets
- Mesures

Attention toutefois au **temps de calcul** (généralement 24 images par seconde)!

# Bilan

Limites des opérations locales :

- Images complexes : on ne peut pas toujours réduire l'analyse à des traitements locaux.
- Les *a priori* dont découle l'analyse sont parfois erronés

⇒ Méthodes d'*apprentissage* : laisser le soin à l'ordinateur de trouver les coefficients d'un filtre pour **reconnaître** une forme.